

01  
31



**UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO**

FACULTAD DE INGENIERIA

SISTEMA INTEGRAL DE INFORMACION DOCENTE  
DE LA FACULTAD DE DERECHO BAJO UN  
AMBIENTE CLIENTE/SERVIDOR.

**T E S I S**

QUE PARA OBTENER EL TITULO DE

**INGENIERO EN COMPUTACION**

P R E S E N T A N :

LEON MARTINEZ, JORGE  
LOPEZ ALBARRAN, CARLOS  
MARTINEZ JORGE ARTURO  
RIVERA DIAZ LETICIA  
SALVADOR HERNANDEZ VERONICA

DIRECTOR DE TESIS: M.I. JUAN CARLOS ROA BEIZA



MEXICO, D. F.

ABRIL 1997.

**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.



**EL PRESENTE TRABAJO SE IMPRIMIÓ CON EL APOYO DE  
LA COMISIÓN NACIONAL DEL AGUA**

---

## AGRADECIMIENTOS

A la Universidad Nacional Autónoma de México el habernos abierto sus puertas para llevar a cabo nuestros estudios dentro de esta Máxima Casa de Estudios y por habernos brindado todo el apoyo para lograr nuestros objetivos.

A la Facultad de Ingeniería el habernos proporcionado el conocimiento necesario para formarnos como buenos profesionistas.

A la Facultad de Derecho por habernos brindado todo el tiempo, documentación, información e infraestructura necesarios, que nos permitieron desarrollar nuestro trabajo de tesis.

Al Programa de Apoyo a la Titulación de la Facultad de Ingeniería y en particular al M. I. Juan Carlos Roa el habernos apoyado en la elaboración de la presente tesis y lograr nuestro objetivo de titulación.

---

**A mis padres Martha Yolanda y Jorge**

Por su esfuerzo, disciplina e infinito amor.

**A mis hermanos Mauricio, Hortensia y Cesar**

Por su compañía y apoyo moral que siempre me han brindado.

**A mi esposa Gabriela**

Por su maravilloso amor, ternura, comprensión y sobre todo por desear compartir el resto de nuestras vidas juntos, gracias.

**A todos mis Amigos**

Por su afecto incondicional.

**JORGE**

---

### **A mi esposa Rocío**

Desde que la conocí todo tiene sentido, la mejor etapa de mi existencia es la más reciente debido a ella. Gracias por ceder gran parte de nuestro tiempo para lograr este objetivo que es mutuo. Todo mi amor para la que siempre será la única en mi vida.

### **A mi mamá Carmen**

Por estar siempre conmigo, por sus desvelos y su amor que me hicieron conseguir la primera meta importante que me tracé en la vida.

### **A mi papá Martín**

Porque el carácter y la integridad no solo se aprenden sino sobre todo se heredan, por siempre voy a estar orgulloso de esta semejanza, y los seguiré canalizando para seguir consiguiendo objetivos como este.

### **A mis hermanos Jorge y Claudia**

Compartimos desde nuestros juegos hasta nuestros problemas, siempre juntos. Les debo mucho y nunca lo olvidaré.

### **A Juan Luis y a Jorge**

Por haberme brindado su amistad y su confianza, dos cosas invaluable para mí.

### **A Angel**

Por ser mi amigo desde hace muchos años, a él le debo gran parte de lo que ahora agradezco.

Hay ocasiones en que la existencia sólo dura unos instantes, y es cuando se aprende la importancia de disfrutar tan efímeros momentos. Gracias Iván por haberme dado la oportunidad de ser padre. No importa el tiempo que te tuvimos siempre estás con nosotros. R. y C.

**CARLOS**

---

**A Dios:**

*Porque nunca se apartó de mi lado.*

**A mi mamá Carolina:**

*Por haberme dado una educación y enseñarme a trabajar.*

**A mis hermanos:**

*Sonia, Carlos, y David por siempre animarme para continuar adelante.*

**A mi Tía:**

*Porque de una u otra manera me apoyó.*

**A mis amigos:**

*Por apoyarme y saber que siempre contaré con ellos.*

**A mis compañeros de tesis:**

*Que me enseñaron que debo trabajar muy duro si quiero lograr algo.*

*Y sobre todo a una persona especial que no hace falta mencionar, para ti con todo mi amor.*

**JORGE ARTURO**

---

**Dedico este trabajo con mucho cariño, fruto de un verdadero esfuerzo de todos nosotros:**

**A "Dios" por haberme dado el regalo más hermoso, "Mi vida" y le agradezco el haberme acompañado en cada momento.**

**A mi "Madre" que siempre me ha enseñado con su ejemplo que las cosas no son fáciles y que siempre hay que salir adelante con los problemas que se nos presentan en el transcurso de nuestra vida. Gracias mamá por estar conmigo siempre que te necesito.**

**A mi "Padre" que me ha enseñado con su ejemplo a ser responsable y a tomar las cosas de una manera optimista. Gracias papá por ser un amigo y un verdadero padre.**

**A mi "Esposo" que con su apoyo y amor me ha ayudado a conseguir las metas que me he propuesto. Gracias Francisco por haber elegido ser parte de mi vida.**

**A mis "Hijas: Leslie y Jazmín", aquellas pequeñinas que nunca me han exigido nada más que un poco de cariño y que desde su nacimiento me han motivado para salir adelante.**

**A la Comisión Nacional del Agua, por haberme apoyado para cumplir con este propósito.**

**A todas las personas que de alguna manera me apoyaron y alentaron a no desfallecer en los momentos difíciles. A todos mis amigos gracias por su comprensión y cariño.**

**LETICIA**

---

**A Dios:**

**Por haberme dado la vida y la dicha de tener a mis padres.**

**A mis padres Marcelina y Martín:**

**Por su cariño, comprensión y paciencia, porque gracias a su esfuerzo logré una carrera profesional.**

**A mis hermanos y sobrinos:**

**Por su cariño y con el ferviente deseo de que lleguen a obtener una carrera profesional.**

**A mis amigos:**

**Por su amistad y su confianza.**

**A Jossue:**

**Por haberme enseñado a ver la vida de una forma diferente.**

**VERÓNICA**

# CONTENIDO

<b>1 CONCEPTOS</b>	<b>1</b>
<b>1.1 METODOLOGÍAS DE DISEÑO</b>	<b>3</b>
1.1.1 Estructurado.	3
1.1.1.1 Análisis.	4
1.1.1.1.1 Diagrama de flujo de datos.	5
1.1.1.1.2 Diccionario de datos.	10
1.1.1.2 Diseño.	11
1.1.2 Tiempo Real.	12
1.1.3 Orientado a Objetos.	14
1.1.3.1 Modelo de objetos.	15
1.1.3.2 Conceptos.	16
1.1.3.3 Herramientas.	17
<b>1.2 MODELOS DE BASES DE DATOS</b>	<b>19</b>
1.2.1 Introducción.	19
1.2.2 Modelo Jerárquico	20
1.2.2.1 Jerarquías balanceadas y binarias.	21
1.2.2.2 Correspondencia simple y compleja.	22
1.2.3 Modelo de Red.	22
1.2.3.1 Estructuras de red simples y complejas.	24
1.2.3.2 Descomposición.	24
1.2.4 Modelo Relacional.	24
1.2.4.1 Herramientas.	25
1.2.5 Modelo Objeto.	27
<b>1.3 ARQUITECTURAS</b>	<b>29</b>
1.3.1 Arquitectura Centrada Mainframe.	29
1.3.2 Arquitectura Centrada PC-Servidor.	30
1.3.3 Arquitectura Punto a Punto.	32
1.3.4 Arquitectura Cliente/Servidor.	32
<b>1.4 AMBIENTE CLIENTE/SERVIDOR</b>	<b>37</b>

1.4.1 Cliente.	37
1.4.1.1 Características.	37
1.4.1.2 Funciones.	38
1.4.1.3 Sistemas Operativos.	39
1.4.1.4 Plataformas.	40
1.4.1.5 Acceso a las Bases de Datos.	40
1.4.2 Servidor.	41
1.4.2.1 Características.	41
1.4.2.1.1 Multiprocesamiento.	42
1.4.2.1.2 Multitulos.	42
1.4.2.1.3 Arreglos de discos.	43
1.4.2.2 Clasificación de Servidores.	43
1.4.2.2.1 Servidor de archivos.	43
1.4.2.2.2 Servidor de aplicaciones.	43
1.4.2.2.3 Servidor de bases de datos.	44
1.4.2.3 Requerimientos.	44
1.4.2.3.1 Independencia de plataforma.	44
1.4.2.3.2 Transacciones.	45
1.4.2.3.3 Conectividad.	45
1.4.2.3.4 Procedimientos almacenados (stored procedures).	46
1.4.2.3.5 Disparos (triggers).	46
1.4.2.3.6 Optimizador.	46
1.4.2.3.7 Herramientas de prueba y diagnóstico.	46
1.4.2.3.8 Confiabilidad.	47
1.4.2.3.9 Mecanismos de respaldo y recuperación.	47
1.4.3 Comunicación Cliente/Servidor.	47
1.4.3.1 Modelo Conversacional.	47
1.4.3.2 Modelo de Mensajes.	48
1.4.3.3 RPCs.	48
1.4.3.4 SQL.	49
<b>1.5 REDES</b>	<b>50</b>
1.5.1 Topologías.	51
1.5.1.1 Estrella.	51
1.5.1.2 Bus.	52
1.5.1.3 Anillo.	53
1.5.1.4 Malla.	54
1.5.1.5 Soporte (backbone).	54
1.5.1.6 Estrella jerárquica.	55
1.5.2 Medios de Transmisión.	56
1.5.2.1 Par trenzado.	56
1.5.2.2 Cable coaxial.	56
1.5.2.3 Fibra óptica.	56
1.5.2.4 Inalámbrica.	57
1.5.2.5 Microondas.	58
1.5.3 Modelos de Interconexión.	59
1.5.3.1 Modelo de referencia OSI (Open System Interconnection).	59
1.5.3.2 SNA.	61
1.5.3.3 TCP/IP.	64
1.5.3.4 DNA.	65
1.5.3.5 Netware.	66
1.5.3.6 LAN Manager.	67
1.5.4 Protocolos de Control de Acceso.	68
1.5.4.1 Estándar IEEE 802.	68

1.5.4.1.1 Estándar IEEE 802.3, CSMA/CD.	68
1.5.4.1.2 Estándar IEEE 802.4, token bus.	71
1.5.4.1.3 Estándar IEEE 802.5, token ring.	73
1.5.4.2 Otros protocolos.	76
1.5.4.2.1 FDDI.	76
1.5.4.2.2 ATM.	79
1.5.4.2.3 Frame relay.	84
<b>1.6 HERRAMIENTAS DE DESARROLLO</b>	<b>87</b>
1.6.1 Servidores de Bases de Datos (Back-Ends o DBMS).	87
1.6.1.1 Sybase.	87
1.6.1.2 Informix.	89
1.6.1.3 Oracle.	89
1.6.1.4 Pruebas de desempeño (benchmark).	89
1.6.1.4.1 Transaccion aleatoria de escritura mezclada(Random Write Transaction Mix).	90
1.6.1.4.2 Lectura aleatoria unica(Single Random Read).	91
1.6.1.4.3 Transaccion aleatoria de lectura mezclada(Random Read Transaction Mix).	92
1.6.1.4.4 Recuperación de B.LOBs ( Binary Large Objects).	94
1.6.1.4.5 Consultas Ad hoc.	95
1.6.1.4.6 Cargar e indexar.	95
1.6.1.4.7 Exportar.	96
1.6.2 Herramienta de Desarrollo para el Usuario Final o Front-End.	97
1.6.2.1 Funcionamiento de los front-ends.	99
1.6.2.2 Herramientas para el desarrollo de aplicaciones.	101
1.6.2.3 Evaluando herramientas para el desarrollo de aplicaciones.	102
1.6.2.3.1 Delphi 2.0 (Borland).	103
1.6.2.3.2 OMNIS 7.3.0 (Biyth Software).	106
1.6.2.3.3 Powerbuilder 5.0 (PowerSoft; Sybase).	111
1.6.2.3.4 SQL Windows 5.0 (Grupta).	116
1.6.2.3.5 Visual Basic 4.0 (Microsoft).	119
1.6.2.4 Análisis de las comparaciones.	124
<b>1.7 NORMATIVIDAD</b>	<b>126</b>
1.7.1 Software.	126
1.7.1.1 Bases de datos.	126
1.7.1.2 Herramientas de desarrollo.	127
1.7.1.3 Programas.	128
1.7.1.4 Reportes.	128
1.7.1.5 Sistemas operativos.	129
1.7.1.6 Interfaces de usuario graficas (GUI).	130
1.7.2 Hardware.	130
1.7.2.1 Equipos de usuario final.	130
1.7.2.2 Servidores.	131
1.7.2.3 Redes.	131
1.7.3 Generales.	131
1.7.3.1 Documentación.	132
1.7.3.2 Manuales de usuario.	132
1.7.3.3 Entradas/salidas de información por área.	132
1.7.3.4 Respaldos periódicos.	133
1.7.3.5 Seguridad.	133
1.7.3.6 Reportes de fallas.	133
1.7.3.7 Bitácora por equipos.	134
<b>2 PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCION</b>	<b>135</b>

<b>2.1 PROBLEMÁTICA ACTUAL</b>	<b>137</b>
2.1.1 Áreas de la Institución.	140
<b>2.2 ORGANIGRAMA DE LA SITUACION ACTUAL</b>	<b>144</b>
<b>2.3 REQUERIMIENTOS DEL USUARIO</b>	<b>150</b>
<b>2.4 ESTRATEGIA DE SOLUCION</b>	<b>157</b>
2.4.1 Análisis.	157
2.4.1.1 Investigación inicial	158
2.4.1.1.1 Evaluar los requerimientos del proyecto.	158
2.4.1.1.2 Preparar la conducción de la investigación inicial.	159
2.4.1.1.3 Analizar los datos colectados.	159
2.4.1.2 Estudio de viabilidad.	159
2.4.1.2.1 Conducir entrevistas a los usuarios.	160
2.4.1.2.2 Revisar el sistema actual.	160
2.4.1.2.3 Definir los objetivos del sistema.	161
2.4.1.2.4 Identificar alternativas.	161
2.4.1.2.5 Evaluar costos y beneficios.	161
2.4.1.3 Definición de requerimientos.	162
2.4.1.3.1 Conducir entrevistas para definición de requerimientos.	162
2.4.1.3.2 Definir los requerimientos del sistema.	162
2.4.1.3.3 Definir los requerimientos del cliente y del servidor.	162
2.4.1.3.4 Definir los requerimientos sobre los servicios de la red.	163
2.4.1.3.5 Definir los requerimientos de implementación.	163
2.4.1.3.6 Identificar alternativas.	164
2.4.1.3.7 Construir el modelo de requerimientos.	164
2.4.2 Diseño.	164
2.4.2.1 Diseño externo	164
2.4.2.1.1 Revisar los requerimientos de definición de datos.	165
2.4.2.1.2 Definir subsistemas.	165
2.4.2.1.3 Definir las funciones cliente/servidor y asignar actividades a procesadores.	165
2.4.2.1.4 Definir sistema de controles de seguridad	166
2.4.2.1.5 Evaluar alternativas de diseño.	166
2.4.2.1.6 Terminación preliminar del modelo de diseño	166
2.4.2.2 Diseño interno.	167
2.4.2.2.1 Diseño de la arquitectura del sistema.	167
2.4.2.2.2 Especificar estructuras de datos	168
2.4.2.2.3 Diseñar la base de datos.	168
2.4.2.2.4 Consideraciones específicas del diseño.	168
2.4.2.2.5 Definir planes de implementación y pruebas.	169
2.4.3 Implementación.	169
2.4.3.1 Revisar el diseño interno de datos.	169
2.4.3.2 Extraer componentes reutilizables.	170
2.4.3.3 Conducir las pruebas de unidad e integridad.	170
2.4.3.4 Preparar manual de usuario.	171
2.4.3.5 Preparar documentación técnica	171
2.4.3.6 Instalar el sistema cliente/servidor.	172
2.4.3.7 Instalar el DEMS.	172
2.4.3.8 Crear la base de datos.	173
2.4.3.9 Instalar la aplicación.	173
2.4.3.10 Verificar y validar aplicaciones.	173
<b>2.5 ALTERNATIVAS DE SOLUCION</b>	<b>174</b>
2.5.1 Realizar un Mantenimiento Mayor a los Programas Existentes.	175

2.5.1.1 Descripción.	175
2.5.1.2 Equipo y recursos disponibles.	176
2.5.1.3 Ventajas	176
2.5.1.4 Desventajas.	176
2.5.2 Actualizar la versión de Paradox y Mantener la Red Netware.	177
2.5.2.1 Descripción.	177
2.5.2.2 Equipo y recursos disponibles	177
2.5.2.3 Ventajas.	178
2.5.2.4 Desventajas	178
2.5.3 Utilizar SqlAnware como Base de Datos, algún Software Cliente y una Red Punto a Punto.	179
2.5.3.1 Descripción	179
2.5.3.2 Equipo y recursos disponibles.	179
2.5.3.3 Ventajas.	180
2.5.3.4 Desventajas.	180
2.5.4 Diseñar e Implementar un Nuevo Sistema Gráfico bajo un Ambiente Cliente/Servidor.	181
2.5.4.1 Descripción	181
2.5.4.2 Equipo y recursos disponibles.	181
2.5.4.3 Ventajas.	182
2.5.4.4 Desventajas.	183
<b>2.6 DIAGRAMAS DE FUNCIONALIDAD</b>	<b>184</b>
2.6.1 Técnicas de Simplificación de Operaciones.	184
2.6.2 Diagramas	187
2.6.2.1 Diagrama de inasistencia de personal docente.	188
2.6.2.2 Diagrama de registro de tesis.	195
2.6.2.3 Diagrama de examen profesional.	197
2.6.2.4 Diagrama de exámenes de oposición abiertos.	200
2.6.2.5 Diagrama de exámenes de oposición cerrados.	203
2.6.2.6 Diagrama de rectificación de calificaciones.	205
2.6.2.7 Diagrama de generación de contratos.	207
<b>2.7 DEFINICION DE REQUERIMIENTOS</b>	<b>209</b>
<b>3 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA</b>	<b>215</b>
<b>3.1 MODELO AMBIENTAL</b>	<b>217</b>
3.1.1 Declaración de propósitos	217
3.1.2 Diagrama de Contexto.	218
3.1.3 Lista de Acontecimientos.	219
3.1.3.1 Inasistencias del personal docente.	219
3.1.3.2 Registro de tesis y examen profesional.	222
3.1.3.3 Exámenes de oposición.	226
<i>Para examen de oposición cerrado.</i>	228
3.1.3.4 Rectificación de calificaciones.	229
3.1.3.5 Generación de contratos.	230
<b>3.2 DIAGRAMA DE FLUJO DE DATOS</b>	<b>234</b>
3.2.1 Nivel Superior.	236
3.2.2 Inasistencias del Personal Docente.	237
3.2.2.1 Inasistencia profesores de asignatura.	238
3.2.2.2 Inasistencia profesores de carrera.	239
3.2.2.3 Inasistencia profesores de extraordinarios y acuerdo de pasantes.	240
3.2.3 Registro de Tesis y Examen Profesional.	241
3.2.3.1 Registro de tesis.	242
3.2.3.1.1 Alta, baja o cambio de tesis.	243
3.2.3.2 Examen profesional.	244

3.2.4 Exámenes de Oposición.	245
3.2.4.1 Exámenes de oposición abierto	246
3.2.4.2 Exámenes de oposición cerrado	247
3.2.5 Rectificación de Calificaciones.	248
3.2.6 Generación de Contratos	249
<b>3.3 DICCIONARIO DE DATOS</b>	<b>250</b>
<b>3.4 NORMALIZACIÓN DE LA BASE DE DATOS</b>	<b>266</b>
3.4.1 Primera Forma Normal.	267
3.4.2 Segunda Forma Normal.	268
3.4.3 Tercera Forma Normal.	269
<b>3.5 DIAGRAMA ENTIDAD-RELACION</b>	<b>271</b>
3.5.1 Módulo de Inasistencias del Personal Docente.	272
3.5.2 Módulo de Registro de Tesis y Examen Profesional.	273
3.5.3 Módulo de Exámenes de Oposición.	274
3.5.4 Módulo de Rectificación de Calificaciones.	275
3.5.5 Módulo de Generación de Contratos.	276
3.5.7 Especificaciones de Entidades	277
3.5.8 Especificaciones de Entidades-Asociativas.	285
<b>3.6 DEFINICIÓN DE LA BASE DE DATOS EN SYBASE</b>	<b>289</b>
3.6.1 Generación de las Bases de Datos en Sybase.	289
3.6.2 Implementación Física de la Base de Datos.	292
3.6.3 Generación del Modelo de Datos Lógico y Físico.	301
3.6.4 Migración de datos	311
<b>3.7 GENERACIÓN DE LAS APLICACIONES GRÁFICAS EN POWERBUILDER</b>	<b>315</b>
<b>4 CONCLUSIONES</b>	<b>357</b>
<b>BIBLIOGRAFIA</b>	<b>363</b>
<b>APENDICES</b>	
<b>CODIGO DE LA BASE DE DATOS</b>	<b>A-1</b>
<b>MANUAL DE USUARIO</b>	<b>B-1</b>
<b>SYBASE SQL SERVER 11</b>	<b>C-1</b>
<b>POWERBUILDER 5</b>	<b>D-1</b>
<b>ERWIN 2.1 GRX</b>	<b>E-1</b>
<b>GLOSARIO</b>	<b>G-1</b>

# INTRODUCCION

**En la actualidad la Facultad de Derecho cuenta con algunos sistemas aislados que le ayudan a llevar adelante sus actividades esenciales, sin embargo, dichos sistemas o bien trabajan en una PC (standalone) o bien en una red local aislada, lo cual limita la información a un grupo reducido.**

**Para poder compartir información que generan algunas áreas se deben de realizar actualizaciones constantes de la información de máquina a máquina.**

**Los sistemas en su mayoría se desarrollaron en Paradox que no es propiamente un servidor de base de datos, propiciando una sobrecarga en la red, no contempla mecanismos de respaldo y recuperación, conectividad, registro de transacciones, de diagnostico, su confiabilidad y rendimiento se ven decrementados mientras se incrementa su uso y la cantidad de datos a manejar.**

Cada día las diferentes áreas de esta facultad exigen más y mejor información de otras áreas con la mayor rapidez posible mediante un acceso cada vez más sencillo para la explotación de la misma, haciendo más eficientes las actividades propias de esta facultad.

Debido a tales exigencias el centro de cómputo en un primer esfuerzo buscara la integración de toda la infraestructura de cómputo disponible, es decir, conectar a una red todos aquellos equipos que generen información de interés a otras áreas, e interconectar las redes que existan. En forma paralela se deberá organizar y centralizar la información buscando la unificación de la misma para todas las áreas. Así mismo se rediseñarán las aplicaciones existentes y desarrollarán nuevas aplicaciones que permitan compartir toda la información de la facultad.

Analizando los recursos de cómputo disponibles logramos observar que la facultad cuenta con más de 400 computadoras personales (en su mayoría con procesador 486 y pentium) distribuidas en todas las áreas, 2 servidores UNIX, 8 redes locales con capacidad para conectar 352 nodos.

En base a todo lo anterior se decidió adoptar el modelo Cliente/Servidor. Se cuenta con redes, servidores de UNIX que pueden alojar cualquiera de los 3 mejores manejadores de datos (Oracle, Sybase e Informix), PCs con capacidad para trabajar con cualquier software front-end en un ambiente gráfico ameno y sencillo para el usuario final.

El sistema que se desea elaborar se basará en el manejador de base de datos Sybase como servidor y se utilizará PowerBuilder como front-end gráfico para las aplicaciones cliente.

La estructura de la base de datos que se creará servirá de base para todo el sistema integral, sin embargo solo se desarrollará la parte académica como parte de esta propuesta.

## Introducción

---

En el Capítulo Uno se habla de métodos, modelos y arquitecturas lo cual nos sirve para conocer los caminos de como se puede llegar a un buen desarrollo de la aplicación que necesita la facultad y si se cuenta con el equipo idoneo para llevarlo a cabo. También se definen detalladamente los tres elementos básicos de una arquitectura cliente/servidor como son la red, el servidor y el cliente, así como el software de desarrollo con que se cuenta.

La etapa de investigación inicial, estudio de viabilidad y definición de requerimientos se exponen en el Capítulo Dos. En esta sección se analiza el funcionamiento de las diversas áreas con las cuales se va a trabajar y su problemática actual, los caminos viables para resolver la situación y la especificación de lo que se va a desarrollar.

Por último el análisis estructurado con sus herramientas de muchos conocidos como el modelo ambiental, el modelo conductual y el modelo de datos se aplican para el desarrollo de nuestra aplicación en el Capítulo Tres. En los subcapítulos finales se muestra la definición física de la base de datos así como la generación de las aplicaciones con el software PowerBuilder.

**CAPITULO**

**1**

# **CONCEPTOS**

# 1.1

## METODOLOGÍAS DE DISEÑO

En la década de los años setentas surge la necesidad de una herramienta capaz de modelar los sistemas de computación, debido a que cada vez eran más complejos los sistemas computacionales y como consecuencia la forma de realizarlos.

El desarrollador de sistemas necesita de una herramienta, para plasmar su idea o solución a una lista de requisitos que debe contemplar su construcción, comúnmente llamada modelado.

El modelado es una simulación de lo que el sistema debe de realizar por medio del cual se pueden llevar a cabo pruebas sin hacer un gran gasto, los modelados surgen de la aplicación de un método. A continuación se describen métodos trascendentes:

### *1.1.1 Estructurado.*

El método estructurado se divide principalmente en dos etapas, el análisis y el diseño.

### 1.1.1.1 Análisis.

El análisis de un sistema es el procedimiento intermedio entre el problema que plantea un cliente (entiéndase cliente como cualquier persona o grupo de personas que requieren de un sistema computarizado) y el diseño del mismo sistema, por lo que en un principio la interacción entre el cliente y el analista es esencial.

La tarea esencial del analista es entender exactamente lo que requiere el cliente, si hay confusiones o malos entendidos por muy bueno que sea el diseño y desarrollo el sistema va a fracasar al no cumplir debidamente con sus necesidades. El análisis es la base, es esencial, de todo el desarrollo de un sistema la comunicación es un requisito fundamental, ya que de ella dependerá el realizar un buen trabajo o un mal trabajo.

El análisis de sistemas puede ser dividido por etapas; la primera es la del planteamiento del problema, en la cual el analista debe de identificar a la persona o personas que más estén relacionadas con el trabajo que se quiere computarizar ya que de ello dependerá que la información que se les pida sea lo mas clara posible. Ya identificadas estas personas, se pueden separar por niveles jerárquicos de la organización para tener un contexto más amplio.

El cliente (persona(s) identificada(s)) da a conocer las necesidades que el requiere por primera vez, ya planteado el problema, el analista se adjudica el papel de periodista para cuestionar las veces que sean necesarias al cliente, no importando que llegue a veces a redundar, es mejor que caer en confusiones.

Una segunda etapa es la del **modelado** en la cual el analista usa alguna de las metodologías o mezcla de ellas para representar lo que ha entendido que requiere el cliente.

El modelado del sistema debe ser tan sencillo que el propio cliente sin necesidad de tener conocimiento de la metodología pueda entender lo que esta plasmado en el papel.

El modelado contempla varios fines como por ejemplo, el cliente a través del modelado que comúnmente es gráfico, si lo comprende puede descubrir algunos errores o enfatizar otros

puntos. Con estas acciones el modelado se va refinando hasta que el cliente este convencido de que es lo que quiere y lo que necesita. Estas correcciones son factibles porque hasta este momento no se ha invertido en equipo de cómputo o software de aplicación.

Otro fin que busca el modelado es conocer exactamente cuales son las características más importantes que requiere el sistema, como también el descartar la información que no es necesaria para el desarrollo del sistema. Esto sucede a raíz de toda la gama de información que nos proporciona el cliente.

Por último el modelado tiene como otra de sus características el estar bien documentado para ser comprendido claramente por el cliente, además de los diseñadores y programadores que se encargarán de llevar a cabo la construcción del sistema.

El modelado es el resultado del trabajo del analista, por lo que muchas personas se han dedicado a tratar de crear o perfeccionar herramientas que nos representen sistemas en forma óptima.

Su principal característica de este método es su representatividad del flujo de información y del contenido de la información. Mediante este estilo se puede imaginar que la información se va modificando conforme va circulando en los diagramas.

DeMarco fue uno de los iniciadores en conjuntar las herramientas para el análisis estructurado aplicado a los sistemas de computación [DEM79].

A principios de los 80s la notación básica de DeMarco tuvo muchas criticas y debido a ello otras personas colaboraron con el modelado, anexando a la notación básica otras herramientas. Estas anexos fueron hechos por Page-Jones [PAG88], Ward y Mellor [WAR85], Gane y Sarson [GAN78], entre otros.

#### **1.1.1.1 Diagrama de flujo de datos.**

El Diagrama de Flujo de Datos (DFDs) es tal vez la herramienta más usada del análisis estructurado, esta herramienta permite conectar mediante los flujos de información los

procesos o transformaciones con almacenes de información, así como con entidades externas al sistema.

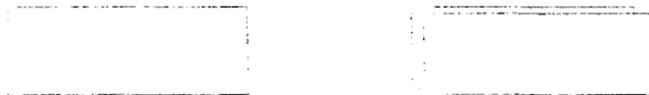
Cuando los datos no son lo más relevante que busca el sistema sino los procesos o funciones operacionales es de gran importancia esta herramienta. A esta característica de los DFDs comúnmente en los textos se le nombra **Orientado a Funciones**.

Edward Yourdon [YOUS89], define al DFD como "un modelo gráfico teórico descendente y por partes".

La anterior definición se irá comprendiendo ampliamente conforme se vayan conociendo los elementos y reglas de los DFDs.

DeMarco [DEM79] da a conocer al mundo entero la notación básica de los DFDs, la cual consistía de:

- Un rectángulo que representa una entidad externa al sistema, es decir, reside fuera de los límites del sistema; son con los que el sistema se comunica. Generalmente las entidades son personas u organizaciones y hasta a veces suelen ser otros sistemas.



**Ilustración 1 Entidades en un DFD.**

- Un círculo por medio del cual se representan las transformaciones o procesos que se aplican a la información que entra, otras veces se representa por un rectángulo con los vértices redondeados.



**Ilustración 2 Procesos en un DFD.**

- Dos líneas paralelas las cuales indican que paquetes de datos están en reposo, comúnmente a este elemento se le conoce como almacén.



**Ilustración 3 Almacenes en un DFD.**

- Una flecha que representa el flujo de la información a través del DFD. Este elemento conecta a los demás elementos antes mencionados como son entidades, procesos y almacenes.



**Ilustración 4 Flujos en un DFD.**

El flujo cuenta de las siguientes características:

- Muestra datos en movimiento.
- Solamente contiene datos.
- No es una señal o pulso.

- Contiene siempre un nombre.

Un flujo muestra el movimiento de los datos entre procesos, entrada y salida de almacenes y entidades. La dirección del flujo es indicado por una flecha. Por definición el flujo debe tener una etiqueta que representa el significado de los datos .

Hay diferentes tipos de flujos como son:

- Flujo divergente: el cual puede tener dos significados, el primero que de un mismo conjunto de datos se hacen copias y es mandado a diferentes lugares como procesos o almacenes; el segundo es que un mismo conjunto de datos es dividido en varios paquetes elementales que requieren ciertos procesos o almacenes.
- Flujo convergente: es cuando varios conjuntos de datos elementales se unen para formar un conjunto más complejo.
- Flujo de diálogo: es cuando el flujo contiene dos cabezas, esto significa que no solo es un conjunto de datos sino dos, como cuando dos personas se comunican, emiten y reciben mensajes en este caso datos.

El proceso cuenta con las siguientes características:

- Transformación de datos.
- Función específica.
- No redundante.

El símbolo de proceso representa la transformación de una entrada de información dando una salida. Los DFDs pueden estar ordenados jerárquicamente, como especie de árbol, entonces, un símbolo de proceso puede estar subdividido en varios procesos y demás herramientas.

Los almacenes se requieren para mantener la información en reposo por algún rango de tiempo. Puede ser usado para mostrar almacenamientos físicos.

Las entidades o terminadores cuentan con las siguientes características:

- Fuera del sistema a ser estudiado.
- Define el contexto del sistema.
- Es el originador y receptor de datos.
- Solamente aparece en el diagrama de contexto.

Una entidad es usada para denotar la fuente o destino de los datos usados o creados. Las entidades definen los límites del trabajo de investigación.

Los DFDs son comúnmente usados por niveles desde lo más general hasta lo más detallado. Al nivel más general se nombra Diagrama de Contexto donde solo se dibuja un símbolo de proceso que representa todo el sistema, que posteriormente se subdividirá en otros procesos obteniendo varios niveles, y las entidades que estén relacionadas con el sistema.

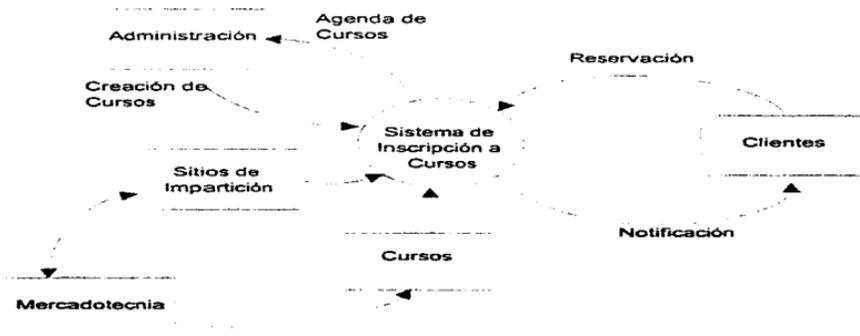


Ilustración 5 Diagrama de contexto.

Hay una herramienta llamada miniespecificaciones por medio de la cual se obtienen las características detalladas de los procesos en un DFD, para llevar a cabo está función, se hace uso de gráficas, tablas de decisión, "español estructurado" (lenguaje narrativo estructurado), diagramas de flujo (no DFDs).

Algunos autores mencionan que las miniespecificaciones son una metodología complementaria a los DFDs, otros la hacen parte fundamental de los DFDs.

Es importante resaltar que los DFDs no están estrictamente diseñados para representar solo sistemas de procesamiento de información, los DFDs también pueden modelar distintas organizaciones, tan complejas como sean.

#### 1.1.1.1.2 Diccionario de datos.

El diccionario de datos se usa comúnmente como un complemento a los demás modelados como el antes mencionado y otros. Este modelado no es gráfico como son la mayoría.

Como su nombre lo indica es un diccionario donde uno puede ir a buscar cualquier parte de un sistema que no se ha entendido o representado por medio de una herramienta gráfica.

Edward Yourdon define a esta herramienta como "una lista organizada de todos los datos pertinentes al sistema, con definiciones precisas y rigurosas, para que tanto el cliente como el analista tengan un entendimiento común de todas las entradas, salidas, etc.".

Las características del diccionario de datos son las siguientes:

- Define las etiquetas que se usan en los flujos y almacenes.
- Los datos no deben ser definidos en forma ambigua.
- El diccionario no debe contener información redundante.
- Ayuda a mantener estándares y nombres únicos.

El diccionario de datos también usa una notación especial como se muestra a continuación:

=	esta conformado de
+	y / junto con
{}	iteraciones de
[/]	selecciona uno de
()	opcional
**	comentario

A continuación se aclaran algunas características de la notación.

"+" no tiene significado matemático.

"Selecciona uno de" puede aplicarse a solo una de muchas opciones.

"Iteraciones de" no tiene límite, esto es repetitivo desde cero hasta el infinito.

El límite de iteraciones puede ser mostrada usando prefijos o sufijos.

### 1.1.1.2 Diseño.

Así como el análisis nos sirve para saber que hay que realizar en un sistema, el diseño nos sirve para saber como hay que desarrollar el sistema.

Los objetivos del diseño más relevantes, deben de encaminar al diseñador a entregar las funciones necesariamente requeridas por el usuario. Para ello se cuentan con tres objetivos [GAN78]: desempeño (Performance), control y adaptabilidad (changeability).

- **Desempeño.** En el desempeño se estudian las transacciones que se llevan a cabo por el sistema en un tiempo determinado (throughput), el tiempo de ejecución y tiempo de respuesta.

- **Control.** Dependiendo de la naturaleza de los sistemas y la cantidad de dinero que se asigne al sistema en construcción, el diseñador incrementará los niveles de control. Por supuesto que mientras más crítica sea la función del sistema, los procesos de control serán más robusto, para abatir errores que puedan surgir.
- **Adaptabilidad.** Comúnmente se considera que un sistema será estático, pero en la realidad todos los sistemas son susceptibles a cambios, por lo cual los sistemas deben de tener un buen grado de cohesión, es decir que cada modulo solamente lleve a cabo una función específica, y un grado mínimo de acoplamiento para que la dependencia entre módulos sea muy reducida y los cambios al sistema no sean muy drásticos.

### ***1.1.2 Tiempo Real.***

Ward y Mello [WAR85] contribuyeron con notación para los sistemas en tiempo real creando los diagramas de transición de estados, estos modelados son frecuentemente usados en el análisis para sistemas en **tiempo real**. James Martin [MAR67] define a estos sistemas como "aquel que controla un ambiente recibiendo datos, procesándolos y devolviéndolos con la suficiente rapidez como para influir en dicho ambiente en ese momento".

La notación de este modelado es muy simple solo usa estados y flujos; los flujos son la vía por medio de la cual se hacen los cambios de estados. Los estados son representados por rectángulos y los flujos, similar al modelado anterior, con flechas. Los Diagramas de Transición de Estados (DTE) contienen otros elementos, los cuales no tienen en si una notación, estos son los nombrados condición y acción.

Todos los sistemas en tiempo-real tienen como característica única que ellos esperan a que ocurra un evento, lo cual es novedoso porque los demás modelados que se describen se enfocan a sistemas que van realizando procesos.

Los estados son aquellas instancias en las que se puede encontrar un sistema, es decir, se deben de analizar e identificar cuales son todas las posibilidades que se puedan dar en un sistema de tiempo real.

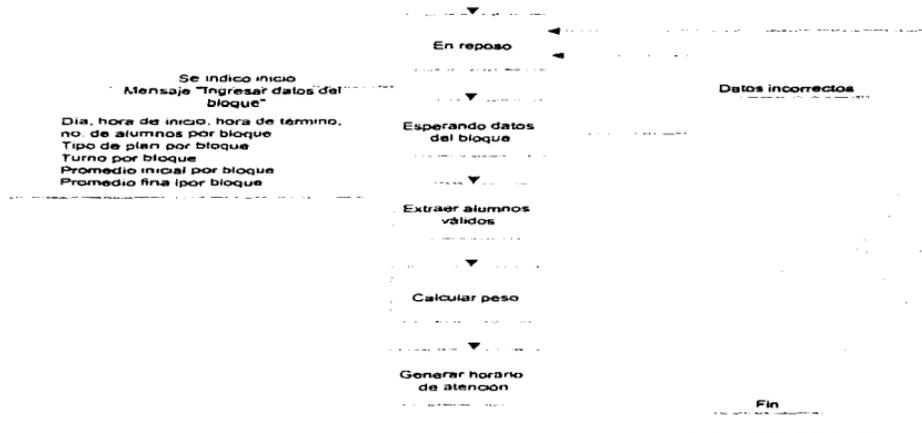


Ilustración 6 Diagrama de transición de estados.

Cada estado esta conectado con uno o varios estados, a los cuales puede estar cambiando, sin embargo, se especifican por medio de reglas que validan los cambios del sistema.

Los Diagramas de Transición de Estados, como todo, tienen un inicio y un fin, el estado de inicio tiene un flujo de entrada que no esta conectado a ningún otro estado; el estado final es aquel que no tiene flujos de salida. No necesariamente solo debe haber un estado final, sino varios. Independientemente del número de estados finales que se tengan, el sistema puede tener solo un estado final en un proceso.

Como se menciono anteriormente hay otros dos elementos acción(es) y condición(es). Las condiciones son los acontecimientos externos que detectan los sistemas, es decir, que dependiendo de lo que se detecta se da una solución (acción) y las acciones son las funciones que se toman cuando se cambia de estado, es decir, respuestas.

Sistemas muy complejos pueden ser representados mediante este modelado, para ello cuenta también con la flexibilidad de ir formando niveles en forma jerárquica.

Este tipo de diagramación es muy usado por los ingenieros electrónicos para la elaboración de sistemas digitales, como cajeros automáticos, controladores (vías férreas, letreros digitales), etc.

### ***1.1.3 Orientado a Objetos.***

En el año de 1967 con la necesidad de programar simulaciones discretas sobre problemas del mundo real surgió el lenguaje SIMULA 67, creado en Noruega, el cual integra por primera vez los conceptos de objetos, clases y herencia.

En la década de los 70s la compañía XEROX desarrollo SMALLTALK basándose en los conceptos manejados por SIMULA 67 dándole nacimiento al paradigma Orientado a Objetos y posteriormente en los 80s aparecieron lenguajes muy conocidos como C++ y Eiffel.

Explicando lo que es "orientado a objetos" podemos mencionar que se trata de realizar una abstracción del mundo real mediante objetos más que de operaciones. Los métodos anteriores se enfocaban principalmente en lo que se debería de hacer, es decir, en las operaciones mientras que este método se basa en los objetos .

Análisis orientado a objetos es:

"Un método de análisis que examina requerimientos desde la perspectiva de las clases y objetos encontrados en el vocabulario del dominio del problema" [BOO94].

Diseño orientado a objetos es:

“Un método de diseño que abarca el proceso de descomposición orientado a objetos y una notación para representar modelos lógicos y físicos así como modelos estáticos y dinámicos del sistema a desarrollar” [BOO94].

Hoy en día no solo contamos con lenguajes orientados a objetos sino también con aplicaciones de desarrollo rápido (RAD) y bases de datos orientados a objetos.

### 1.1.3.1 Modelo de objetos.

El modelo de objetos abarca los principios de abstracción, encapsulación, modularidad, jerarquía, además de tipificación, concurrencia y persistencia.

Los cuatro elementos principales de Modelo de objetos son[BOO94]:

- **Abstracción.** Denota las características esenciales de un objeto que lo distinguen de otras clases de objetos proporcionando también los límites conceptuales, relativos a la perspectiva de quien lo observe.

Por ejemplo si observaran un carro dos personas, una de ellas mecánico y otra un joven. El primero al observar el objeto carro pensaría en el motor, su sistema de frenos, etc; y el segundo le tomaría más importancia al chasis, color, tipo de rines, etc. Con lo anterior nos podemos percatar que aunque sea un mismo objeto diversas personas lo podrían conceptualizar de diferente forma.

- **Encapsulación.** La abstracción y encapsulación son complementarios, la abstracción se enfoca en la conducta de un objeto mientras la encapsulación se enfoca en la implementación que da origen a su conducta. La encapsulación se logra mediante el ocultamiento de la información. Comúnmente, la estructura de un objeto es oculta así como la implementación de sus métodos.

- **Modularidad.** Consiste en particionar un programa en diversos componentes individuales, buscando reducir la complejidad del programa mismo. La modularidad nos permite la reutilización de código lo cual en nuestros tiempos es indispensable.
- **Jerarquía.** Es un rango u ordenamiento de abstracciones. Las 2 importantes jerarquías en un sistema complejo es su estructura de clases y su estructura de objetos.

### 1.1.3.2 Conceptos.

- **Objeto:** Es un componente del mundo real que se transforma en dominio del software. En el contexto de un sistema basado en computadora, un objeto es un procedimiento o consumidor de información o un elemento de información. Un objeto consta de una estructura de datos privada y procesos (métodos) los cuales transforman la estructura de datos. Las operaciones contienen el control y las construcciones procedurales que pueden ser llamadas mediante un mensaje.
- Un objeto tiene además una interfaz por medio de la cual se mueven los mensajes y especifican que operaciones del objeto se desean. El objeto que recibe un mensaje determina como se implementa la operación.
- **Mensaje.** Para realizar algo con un objeto, necesitamos mandarle un requerimiento, este requerimiento invoca una operación. La operación ejecuta el método apropiado y opcionalmente regresa una respuesta. El mensaje que constituye el requerimiento contiene el nombre del objeto, el nombre de la operación y algunas veces un grupo de parámetros[MAR92].
- **Clase:** Es un conjunto de objetos que tienen las mismas características.
- **Instancia:** Un objeto individual es una instancia de una clase más amplia.

- **Herencia:** Todos los objetos son miembros de una clase más amplia y heredan la estructura de datos privada y sus operaciones.
- **Polimorfismo:** Uno de los principales objetivos del diseño orientado a objetos es la reutilización de código. Sin embargo, algunas operaciones a veces son muy particulares. Por ejemplo, si contamos en nuestro diseño con una clase EMPLEADOS y se le define una operación de jubilación automáticamente todas las subclases heredarán dicha operación. Pero una organización puede tener diferentes métodos uno para jubilar a un ejecutivo y otro para jubilar un empleado en general. Aunque los métodos difieran logran el mismo propósito. Es decir, polimorfismo es la característica que permite una diferentes implementaciones, de una misma operación dependiendo del tipo de objeto.

La clase más amplia se denomina superclase y las clases específicas se les conoce como subclases.

### 1.1.3.3 Herramientas.

Grandes diseños vienen de grandes diseñadores, no de grandes herramientas. Las herramientas simplemente facilitan la individualidad, liberando a la persona para concentrar en la genuina creatividad de aspectos del análisis o diseño. Así hay algunas cosas que las herramientas pueden hacer bien y algunas otras que pueden hacer del todo bien

- **Diagramas de Clases.** Un diagrama de clases es usado para mostrar la existencia de clases y sus relaciones en la vista lógica de un sistema. Un único diagrama de clases representa una vista de la estructura de clases de un sistema. Durante el análisis, usamos diagramas de clases para indicar los roles comunes y responsabilidades de las entidades que proporcionan la conducta del sistema. Durante el diseño usamos diagramas de clases para captar la estructura de las clases que forman la arquitectura del sistema.

- **Diagramas de Objetos.** Un diagrama objeto es usado para mostrar la existencia de objetos y sus relaciones en el diseño lógico de un sistema. Un diagrama de objetos representa una instancia de eventos sobre una cierta configuración de objetos. Durante el análisis se usan diagramas para indicar la semántica de escenarios primarios y secundarios que proporcionen una línea de conducta del sistema. Durante el diseño se usan para ilustrar la semántica de mecanismos en el diseño lógico de un sistema.
- **Diagramas de Transición de Estado.** Es usado para mostrar el estado de una clase dada, los eventos que causan una transición de un estado a otro y las acciones que resulten de un cambio de estado.
- **Diagramas de Interacción.** Es usado para trazar la ejecución de un escenario en el mismo concepto como un diagrama de objetos. Este diagrama es simplemente otro camino de representación de un objeto. La ventaja de usar un diagrama de Interacción es que facilita la lectura de paso de mensajes en cierto orden, mientras que los diagramas de objetos son mejores para escalar muchos objetos con invocaciones complejas y permite la inclusión de otra información como valores de atributos, roles, flujo de datos y visibilidad.
- **Diagramas de Módulos.** Este es usado para mostrar la colocación de clases y objetos a módulos en el diseño físico de un sistema. Durante el desarrollo se utilizan estos diagramas para indicar los niveles o capas físicas y particionar la arquitectura.
- **Diagramas de Procesos.** Es útil para mostrar la colocación de procesos a procesadores en el diseño físico de un sistema. Durante el desarrollo se utilizan para indicar la colección física de procesadores y dispositivos que sirven como plataforma para ejecutar los sistemas.

# 1.2

## MODELOS DE BASES DE DATOS

De los 4 modelos de bases de datos más importantes, 3 de ellos han sido muy usados en su tiempo: jerárquico, red y relacional. Debemos aclarar que el relacional hoy en día es el más popular. El cuarto modelo, el orientado a objetos está en sus primeros pasos. Aunque sabemos del gran auge que están teniendo las aplicaciones orientadas a objetos en las bases de datos todavía no se ha dado. Contamos actualmente con varias bases de datos orientadas a objetos sin embargo el mercado continúa siendo dominado por las relacionales.

### *1.2.1 Introducción.*

Las bases de datos se dividen de acuerdo al modelo. Hoy en día diferentes bases de datos (diferentes marcas) pueden intercambiar información, lo cual en el pasado era casi un sueño o muy complejo de realizar. Las bases de datos en los últimos años han tenido un gran crecimiento, sus interfaces anteriormente llegaban a ser hasta complejas hoy en día las más importantes tienen interfaces gráficas muy amigables.

Una base de datos puede definirse como una colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es la de servir a una aplicación o más, de la mejor manera posible; los datos se almacenan de modo que resulten independientes de los programas que los usan; se emplean métodos bien determinados para incluir datos nuevos y para modificar o extraer los datos almacenados. Se dice que un sistema comprende una colección de base de datos cuando estas son totalmente independientes desde el punto de vista estructural [MAR77].

Estos modelos son lógicos y se basan en registros, son usados para especificar la estructura lógica de la base de datos y una detallada descripción de la implementación.

### ***1.2.2 Modelo Jerárquico.***

En un modelo jerárquico, los elementos de datos tiene una relación Maestro/esclavo (Master/Slave); cada maestro puede tener muchos esclavos, pero cada esclavo puede tener sólo un maestro.

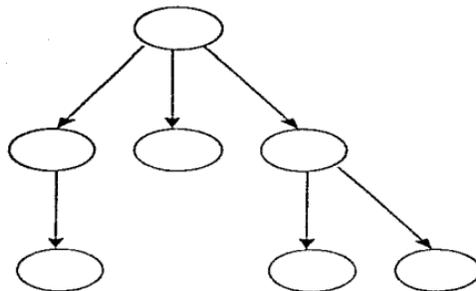


Ilustración 7 Modelo jerárquico.

Los árboles están compuestos por una jerarquía de elementos llamados nudos, el nivel más alto de la jerarquía tiene un solo nudo llamado raíz.

Con excepción de la raíz, todo nudo está relacionado con otro nudo de nivel más alto llamado Padre. Ningún elemento puede tener más de un padre, sin embargo, todo elemento puede tener uno o más elementos de nivel más bajo llamados hijos. Los elementos que se encuentran en las puntas de las ramas, es decir, que no tienen hijos se les llaman hojas.

Los árboles se utilizan tanto para las descripciones lógicas, donde describen las relaciones que existen entre tipos de segmentos o tipos de registro; como para las descripciones físicas donde se les emplea para describir conjuntos de apuntadores y relaciones entre entradas en los índices.

### 1.2.2.1 Jerarquías balanceadas y binarias.

En una jerarquía balanceada todos los nudos deben tener el mismo número de ramas, el árbol empieza por la raíz en forma descendente y va avanzando de izquierda a derecha.

Es fácil implementar una organización física de datos para los árboles que tienen un número fijo de ramas por nudo que para los que tienen números variables.

La mayoría de las organizaciones de datos por supuesto que no encajan en una estructura jerárquica balanceada sino que requiere números de ramas diferentes en los diversos nudos.

Los algoritmos de indexación y exploración manejan este tipo de estructuras.

La jerarquía binaria es una categoría especial de la estructura jerárquica balanceada, que admite sólo dos nudos por rama. Las jerarquías binarias al igual que otras jerarquías balanceadas son principalmente utilizadas en la representación física de los datos y no en las lógicas.

### **1.2.2.2 Correspondencia simple y compleja.**

Las estructuras jerárquicas implican por lo general que hay una correspondencia simple de hijo a padre, es decir, cada hijo tiene sólo un padre. Sin embargo la correspondencia inversa es compleja, de uno a muchos, porque cada padre puede tener varios hijos.

Ocasionalmente hay correspondencia simple en los dos sentidos.

### ***1.2.3 Modelo de Red.***

El modelo de red es más flexible que el jerárquico, permite a los esclavos relacionarse con varios maestros.

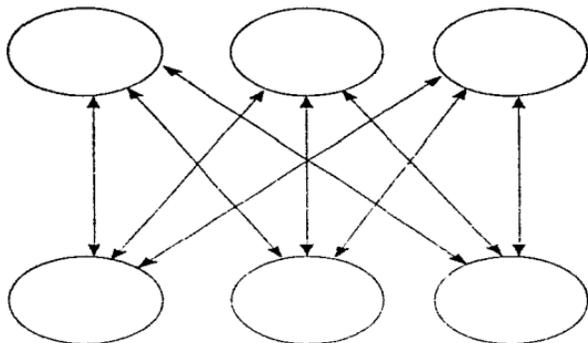


Ilustración 8 Modelo de red.

Una desventaja de éste modelo es que algunas estructuras de red comienzan a tomar una apariencia de "telaraña", con apuntadores que salen en todas direcciones, por lo cual podría llegar a ser compleja y difícil de modificar.

Si en una relación entre datos un hijo tiene más de un padre, la relación no puede ser descrita por medio de un árbol o estructura jerárquica, se le describe en cambio, como una estructura de red.

En una estructura de red cualquier elemento puede vincularse con cualquier otro elemento. Al igual que la estructura jerárquica la estructura de red puede ser descrita en términos de padres e hijos, y dibujada de cierta forma que los padres siempre queden arriba de los hijos. Pero en la estructura de red, un hijo puede tener más de un padre.

En algunas estructuras de red es natural referirse a niveles, como ocurre con las estructuras jerárquicas.

### **1.2.3.1 Estructuras de red simples y complejas.**

En una buena cantidad las estructuras de red que representan relaciones entre tipos de registros o tipos de agregados de datos, la correspondencia entre padres e hijos es similar a la que existe en un árbol, la correspondencia de hijo a padre es simple y la de padre a hijo compleja.

Sin embargo en un modelo de red es común que la estructura sea compleja, es decir, que los hijos tengan muchos padres, así como los padres tengan muchos nudos hijos.

La razón que justifica el patentizar la diferencia entre estructuras simples y complejas reside en el hecho de que las últimas requieren métodos más elaborados para su representación física.

### **1.2.3.2 Descomposición.**

Las estructuras de red se pueden reducir a una forma más simple introduciéndose algo de redundancia. En algunos casos la redundancia involucrada en la transformación no es mucha y se tolera sin problemas.

Una estructura de red compleja no es tan fácil de convertir en una estructura de árbol, se requieren dos arboles para ser representada.

En general cada relación con correspondencia compleja en los dos sentidos tiene que ser reemplazada por dos vinculaciones del tipo de estructura jerárquica.

### **1.2.4 Modelo Relacional.**

El modelo de base de datos relacional ha recibido considerable atención en los últimos años. Ofrece muchas ventajas sobre los modelos jerárquico y de red.

El modelo relacional es una estructura lógica en vez de física. A continuación se mencionarán algunas ventajas de este modelo[DEI90]:

- La representación tabular usada en el esquema relacional es fácil de comprender por los usuarios y fácil de implementar en el sistema físico de bases de datos.
- Es relativamente fácil convertir casi cualquier otro tipo de estructura o modelo de base de datos al esquema relacional, por lo que este modelo puede considerarse como una forma de representación universal.
- Las operaciones de proyección y reunión son fáciles de implementar, por lo que la creación de nuevas relaciones necesarias para aplicaciones particulares resulta fácil de realizar.
- El control de acceso a datos sensibles es de implementación inmediata. Los datos sensibles se colocan simplemente en relaciones separadas y el acceso a estas relaciones se controlan mediante algún tipo de autoridad o esquema de acceso.
- Las búsquedas pueden ser mucho más rápidas que en los sistemas que deben seguir una cadena de apuntadores.
- Las estructuras relacionales son mucho más fáciles de modificar que las estructuras jerárquicas o de red. En ambientes donde la flexibilidad es importante, esto es primordial.
- La claridad y visibilidad de la base de datos mejora con la estructura relacional. Es mucho más fácil buscar datos tabulados que desarrollar posibles interconexiones arbitrarias y complejas de elementos de datos dentro de un mecanismo de apuntadores.

#### **1.2.4.1 Herramientas.**

- **Tablas o relaciones.** Las tablas bidimensionales han sido una herramienta muy fácil para representar datos de los usuarios. Las tablas deberán organizarse de forma que no se pierda ninguna de las relaciones existentes entre datos. Las tablas son matrices rectangulares que pueden ser descritas matemáticamente.



Ilustración 9 DER compuesto de tablas.

- **Atributos.** A las columnas de las tablas se les conoce como atributos de las entidades y contienen valores. Date les nombra dominios[DATS1].
- **Tupla.** Es un renglón de la tabla. Al número de tuplas de una tabla se le nombra cardinalidad.
- **Llaves o claves.** Por lo general en una tabla o relación existe un atributo cuyos valores son únicos, por lo cual, se pueden usar para identificar cada una de las tuplas de esa tabla, ha este tipo de atributos se les conoce como llaves primarias. No siempre en una relación existe un solo atributo que identifique a cada tupla, por lo que para identificar a cada tupla se requieren conjuntar más de un atributo formando la llave primaria.

- **Llave foránea.** La llave foránea es un atributo que contiene una "x tabla", en la cual ese atributo no es llave primaria sin embargo en otra tabla y de la misma base de datos si actúa como llave primaria.
- **Esquemas.** Es la descripción lógica de la base de datos, es un diagrama de los tipos de datos que se usan. Proporcionan los nombres de las entidades y sus atributos y especifica las relaciones que existen entre ellos. Es un marco en el que se inscriben los valores de los ítems de datos [MAR77].
- **Instancias.** Las bases de datos cambian con el tiempo. La información almacenada en la base de datos en algún instante es llamada instancia.
- **Normalización.** Es un proceso de paso a paso que permite reemplazar relaciones entre datos. Las técnicas de normalización han sido creadas por E. F. Codd.
- **Álgebra relacional.** Es un lenguaje de consulta procedural [KOR88]. Hay 5 operaciones fundamentales: selección, proyección, producto cartesiano, unión y diferencia. Todas estas operaciones crean una nueva relación o tabla como resultado. Adicionalmente se contemplan las operaciones: intersección, reunión y división. Estas 3 últimas operaciones pueden ser definidas en términos de las operaciones fundamentales.

### ***1.2.5 Modelo Objeto.***

El modelo orientado a objetos de una base de datos toma la idea de una base de datos inteligente para su conclusión lógica. Los datos son accedidos solo con los métodos que se encuentran almacenados en la base de datos. Estos métodos siempre están listos para realizar la acción correspondiente en el momento que se le requiera. Los datos de todos los objetos están encapsulados. Los datos están generalmente activos más que pasivos.

El análisis, diseño, generación de código y generación de bases de datos puede realizarse usando el mismo modelo conceptual[MAR92].

El empleo de un único modelo conceptual para el análisis, diseño programación, y bases de datos nos da las siguientes ventajas:

- Alta productividad.
- Pocos errores.
- Mejor comunicación entre usuarios, analistas y desarrolladores
- Mejor calidad.
- Mayor flexibilidad.
- Gran inventiva.

El mundo no es una colección de tablas. Un modelo común del mundo es el ensamble por jerarquías. Por ejemplo, una parte esta compuesta de subpartes, que a la vez se pueden dividir. Esto es difícil de representar y manipular usando tablas.

La primera base de datos orientada a objetos OODB, surgió en los últimos años de los 80s y actualmente hay por lo menos una decena de ellas.

Aunque la iniciativa privada se este interesando por este tipo de bases de datos y no dudemos que algunos estén desarrollando aplicaciones, el "boom" todavía no se ha dado y el modelo relacional sigue siendo el más usado y con mayores productos

# 1.3

## ARQUITECTURAS

Todavía hoy en día en el mundo hay una gran cantidad de terminales conectadas a Mainframes y una explosiva expansión de redes de área local que naturalmente difieren con conceptos de la Arquitectura Cliente-Servidor, sin embargo muy a menudo son confundidos por lo que a continuación se detallan características de las arquitecturas Céntrica PC-Servidor y Céntrica Mainframe.

### *1.3.1 Arquitectura Céntrica Mainframe.*

En el ambiente Céntrico Mainframe, los usuarios interactúan con aplicaciones que corren sobre Mainframes a través de terminales o emuladores de terminal basados en PC, los cuales tiene las siguientes características:

Pueden presentar interfaces propietarios. Comúnmente las terminales o emuladores de terminal presentan una interface de usuario en modo texto y no gráfica. Recientemente, la industria ha presentado terminales X-Window que hacen más amigable la interacción con los Mainframes.

En las terminales lo que se tecldea es transmitido al Mainframe, por lo que no se necesita de inteligencia local, delegando de esta forma el procesamiento que se requiera al Mainframe. Las terminales solo deben ser capaces de realizar la transmisión de comandos o queries que el usuario exija.

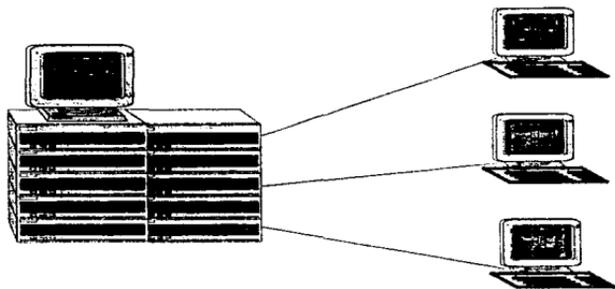


Ilustración 10 Arquitectura Céntrica Mainframe.

Las terminales son conectadas en forma directa a controladores, mientras los emuladores de terminales en PC son conectadas a través de modems o redes.

La sobrecarga de trabajo excesiva sobre los recursos, bastante caros, de un Mainframe como es el compartir tiempo de CPU y sistemas de almacenamiento, constituyen su principal desventaja.

En los sistemas de Mainframe, el control administrativo es tan bueno como comprensivo sea el sistema administrador del Mainframe.

### ***1.3.2 Arquitectura Céntrica PC-Servidor.***

En la arquitectura Céntrica PC-Servidor, las PCs comparten aplicaciones y datos residentes sobre uno o más servidores basados en PC. Este ambiente proporciona flexibilidad a

usuarios individuales, pero el control administrativo es muy pobre. Estos sistemas tienen las siguientes características:

Comúnmente, el Servidor basado en PC es usado para compartir impresoras, compartir aplicaciones comunes y datos (archivos solamente). El servidor de archivos proporciona un rango de servicios para compartir datos con una o más aplicaciones.

Usualmente todos los comandos o queries son procesados en PC. Por lo que debe contener la PC una gran cantidad de memoria para correr aplicaciones sofisticadas, por lo que va en contra de los descos corporativos para proporcionar el mejor costo-beneficio para los usuarios.

Además las aplicaciones basadas en PC pueden interactuar con el servidor de archivos accediendo datos. La transportación de archivos de datos del Servidor de archivos hacia alguna PC local causa tráfico sobre la red. Esto es particularmente cierto para los programas de bases de datos basados en una red de PCs (DBASE, FOXPRO, PARADOX).

En conjunto, estas desventajas de las aplicaciones basadas en PC forman el "Talón de Aquiles", punto débil de los sistemas Céntricos PC-Servidor.

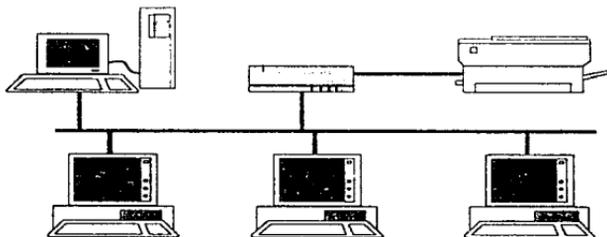


Ilustración 11 Arquitectura Céntrica PC-Servidor.

### ***1.3.3 Arquitectura Punto a Punto.***

Todas las computadoras que participan en la arquitectura punto a punto, son similares y pueden requerir o proporcionar servicios a o de cualquier otra computadora. Esto es lo último en la distribución de procesamiento.

También comparten los recursos, una sencilla arquitectura punto a punto puede actuar como un cliente para otros servidores y como un servidor para otros clientes (incluyéndose), pero en una arquitectura inteligente de punto a punto un servidor puede distribuir la carga de trabajo entre diferentes servidores. Idealmente el ambiente punto a punto debe proporcionar transparentemente el procesamiento cooperativo entre aplicaciones posiblemente residentes sobre una gran variedad de hardware y software.

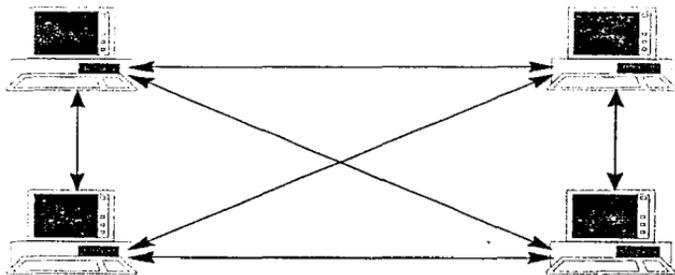


Ilustración 12 Arquitectura Punto a Punto.

### ***1.3.4 Arquitectura Cliente/Servidor.***

Como primer paso se requiere conocer el termino Cliente/Servidor, el cual nos ayudará a comprender su significado. cada autor contempla su propio concepto sin embargo todos tienden a coincidir. A continuación se listan algunas definiciones:

"Cliente/Servidor es un diseño de aplicación enfocada en la descomposición de un sistema de información dentro de un número pequeño de funciones del servidor, ejecutándose sobre una o más plataformas, que proporcionan servicios de uso común a un número grande de funciones cliente, que están interconectadas y realizan trabajos definidos sobre los servicios comunes proporcionados por las funciones servidor"[VAU94].

"En la computación Cliente/Servidor uno o más clientes y uno o más servidores bajo un sistema operativo y sistemas de comunicación permiten, un sistema de computo distribuido. A este compuesto se le suele llamar Sistema Cliente /Servidor"[SIN92].

"Básicamente, una arquitectura Cliente/Servidor es un enfoque computacional que separa procesos sobre plataformas separadas interactuando una con otra que permite compartir recursos tomando la mejor ventaja de los diferentes dispositivos"[BOC94].

Resumiendo Cliente/Servidor es un modelo de cómputo, para crear sistemas de información usando diferentes tipos de tecnologías, que permite la distribución de los recursos dividiendo las tareas de la aplicación.

- Una arquitectura Cliente/Servidor consiste de un proceso cliente y un proceso servidor que pueden interactuar.
- La parte del cliente y la parte del servidor pueden operar sobre plataformas diferentes.
- Cualquiera de las dos partes Cliente o Servidor puede ser actualizada sin tener que actualizar la otra parte.
- La Arquitectura Cliente/Servidor incluye la capacidad de comunicación por red.
- El Servidor es capaz de atender los requerimientos de múltiples usuarios y los clientes pueden ser capaces de acceder a múltiples servidores.
- Una parte significativa de la aplicación reside en el cliente.

- La acción o requerimiento es usualmente iniciado por el cliente, no por el servidor. Sin embargo los servidores de bases de datos pueden ejecutar acciones como los triggers y procedimientos almacenados.
- Las interfaces de usuarios gráficas (GUIs) son muy amigables y residen generalmente en el cliente.
- La capacidad de un lenguaje de consulta estructurado (SQL) es característica de la mayoría de las sistemas bajo Cliente/Servidor.
- El servidor de bases de datos debe proporcionar seguridad a los datos.

La arquitectura Cliente-Servidor establece una división óptima de trabajo entre las aplicaciones Cliente y los Servidores de bases de datos. Las aplicaciones Cliente interactúan y presentan información a los usuarios.

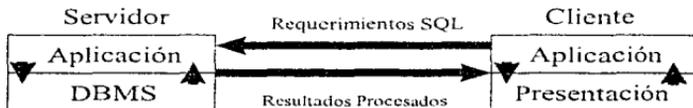


Ilustración 13 Arquitectura Cliente/Servidor.

Los Servidores de bases de datos llevan a cabo la manipulación de los datos a una gran velocidad, protegen la integridad de los datos. Da el control de los datos bajo ambientes distribuidos al administrador del servidor de bases de datos y al usuario final la flexibilidad para el uso de archivos o herramientas de su elección.

Se integran sistemas operativos heterogéneos, dispositivos de almacenamiento de información, redes y aplicaciones, esto es posible gracias a la nueva generación de poderosas workstation y PCs.

Cliente-Servidor es la forma de emprender el procesamiento de gran cantidad de información distinguiéndose por la incorporación de potentes equipos de cómputo, interoperabilidad entre plataformas y software, y fácil acceso a datos.

Cliente-Servidor es un proceso mejorado y de diferente uso de la computación.

Lo básico de la arquitectura Cliente-Servidor depende de dos clases de plataforma:

- Cliente y Servidor, y
- una interconexión de red (llamada Middleware).

El término Cliente-Servidor ha sido popularmente extendido principalmente a los sistemas multi-plataforma donde los recursos son compartidos.

Dos décadas atrás, en la industria del procesamiento de datos se usaron los poderosos mainframes, los usuarios compartían tiempo de CPU y la facilidad del almacenamiento o de información.

El acceso a los mainframes fue rigurosamente controlado por los administradores de los sistemas de información y solamente los métodos de acceso eran directos mediante tarjetas o terminales tontas. Nos ha tocado ver la evolución de la recuperación de la información así como las metodologías de análisis y técnicas difíciles y restringidas de los recursos centralizados e interfaces de usuario hostiles.

Cuando las PCs llegaron a la industria de la computación en la década anterior, su desarrollo fue fenomenal por su forma de uso tan amigable.

Las PCs sin embargo fueron incapaces de manejar grandes cantidades de procesamiento de datos que requiere una empresa grande. Las redes de área local (LANs) proporcionan la solución para conectar PCs y mainframes.

Con la presencia de las PCs, estas fueron usadas como tipo de terminal amigable con el usuario para conectarse a un host.

En estos últimos años hemos visto como las LAN han evolucionado. Hoy en día hay gran cantidad de LANs y PCs instaladas en corporaciones, trabajando con procesadores de palabra, hojas de cálculo y bases de datos.

Mientras las PCs han alentado el desarrollo de mucho software, las LANs han permitido compartir archivos y diversos periféricos en una empresa. También hoy en día, nos encontramos instaladas una gran cantidad de estaciones de trabajo con UNIX.

La arquitectura Cliente-Servidor divide una aplicación en diferentes procesos operativos sobre máquinas separadas conectadas a una red.

# 1.4

## AMBIENTE CLIENTE/SERVIDOR

### *1.4.1 Cliente.*

Al software para Cliente se le llama "Front-end" puede estar en los más diversos ambientes como UNIX, MSDOS, WINDOWS, MACs, etc., permite al usuario final interactuar con la computadora y contiene al menos alguna aplicación lógica. La mayoría del software Cliente proporciona la característica popular de una interfaz de usuario gráfica (GUI) lográndose un uso amigable.

#### **1.4.1.1 Características.**

Técnicamente hablando la característica más importante de un Front-end de un sistema Cliente/Servidor es que le requiere un servicio al software del Servidor también llamado "Back-end".

Un Front-end puede encontrarse en una amplia gama de software. Puede ser tan sencillo como una herramienta existente de usuario final como una hoja de calculo o tan complejo como un programa en C o C++.

Muchas de las empresas que desarrollaban 4gl actualmente desarrollaron sus productos para que trabajaran como Front-end en un ambiente Cliente/Servidor.

Algunos sistemas Cliente/Servidor son tan simples que la aplicación lógica meramente consiste de una instrucción de SQL la cual accede a la información de un Servidor de Bases de Datos[BOC'94].

La mayoría de los Front-end contienen un GUI para facilitar su uso pero no es necesario ya que existen aplicaciones que corren en modo texto.

Muchos de los Front-ends trabajan con una amplia variedad de Back-ends. Sin embargo no todos los Front-ends trabajan con todos los Back-ends. Algunas compañías de software ofrecen tanto el Front-end como el Back-end y además permiten a otras compañías del ramo desarrollar otros Front-ends para su Back-end.

Los Front-ends continúan incorporando características orientadas a objetos.

Por último hay que recordar que un Front-end además de ser el software que permite interactuar al usuario final con la máquina, es el que hace un requerimiento al Back-end.

### **1.4.1.2 Funciones.**

Uno de los usos populares de los sistemas Cliente/Servidor es permitir a los usuarios finales el acceso a los datos de un mainframe, que frecuentemente tiene la función de servidor en esta clase de arquitecturas. Muchas herramientas de consulta (query) como un propósito, acceden a los datos de un Servidor.

Bases de Datos Relacionales son instaladas en los servidores (mainframe, workstation, etc) donde pueden ser accedidas fácilmente usando SQL.

---

Algunas herramientas query son propietarias, es decir, son desarrolladas específicamente para una base de datos en particular, sin embargo también hay otras herramientas query que pueden acceder a la información de diferentes sistemas manejadores de bases de datos relacionales (RDBMS) y a SQL se le debe esta significativa contribución[BER92].

La gran mayoría de herramientas proporcionan acceso a bases de datos a través de SQL.

### 1.4.1.3 Sistemas Operativos.

El primer propósito de un sistema operativo es proporcionar a las aplicaciones los recursos de hardware de los equipos de cómputo como memoria, almacenamiento en disco, etc., y manejo con los dispositivos externos como impresora, scanner, etc.

La transportabilidad de nuestras aplicaciones en un Front-end depende primordialmente de los sistemas operativos en los que sea capaz de trabajar.

Actualmente nos encontramos en el mercado con una gran gama de sistemas operativos y en gran medida las empresas cuentan con diversos equipos, teniendo como consecuencia, también una gran diversidad de sistemas operativos.

En las empresas con la diversidad de plataformas, antes mencionada, no les conviene "casarse" con un único sistema operativo y mucho menos propietario, por lo que se recomienda desarrollar las aplicaciones cliente en un Front-end que trabaje sobre una gran variedad de sistemas operativos. esta característica nos ahorrara tiempo de trabajo ya que al exportar nuestra aplicación de un sistema operativo a otro serán mínimas las modificaciones o quizás nulas.

Entrando en detalles con los sistemas operativos debemos de tener en cuenta cuantas diferentes configuraciones de hardware pueden soportar dichos sistemas operativos.

Dando algunos ejemplos, el MS-DOS sólo puede correr en PC's, Windows también fue hecho para trabajar en PC's, a diferencia de UNIX que puede trabajar en la gran mayoría de

computadoras ya sea PC's, WorkStations, Minis, etc. dándole la característica de portabilidad.

También hay que tomar en cuenta que el sistema operativo donde corra nuestra aplicación incluya interfaces de usuario gráficas como Windows, OS/DOS Warp, Solaris, etc., con el fin de facilitar el manejo de la aplicación al usuario final[DAW93].

#### **1.4.1.4 Plataformas.**

Sin lugar a dudas el primer problema a resolver cuando deseamos realizar cambios de sistemas de cómputo buscando mejoras, es la compra de nuevo hardware lo cual redundará en llevar a cabo grandes inversiones, sin embargo una de las grandes ventajas de la Arquitectura Cliente/Servidor es la de integrar las diversas modalidades de hardware con que se cuentan, desde una PC con procesador 8088, hasta una moderna WorkStation con procesador RISC.

Aunque hemos dicho que un software cliente puede instalarse desde un equipo bastante limitado como un 8088 mediante una interfaz de usuario de tipo carácter, los software cliente en el mercado actual necesitan de un equipo más avanzado debido a que integran la mayoría de las veces una interfaz de usuario gráfica, la cual resulta más sencilla para el aprendizaje de los usuarios.

La arquitectura Cliente/Servidor ha sido la punta de lanza para que empresas de cómputo tradicionalmente propietarias se abran a la gran gama de productos que puedan ejecutarse en sus plataformas. Con esta apertura la mayoría de las empresas dedicadas a la fabricación de equipos minis o mainframes han logrado recuperar mercado y prestigio.

#### **1.4.1.5 Acceso a las Bases de Datos.**

Uno de los usos más populares de los sistemas Cliente/Servidor es el de permitir a los usuarios finales el acceso a los datos de un gran almacén como el de una base de datos. Muchas herramientas query ayudan a llevar a cabo este propósito.

El desarrollo de las herramientas query en primera instancia se pensaba que iban en contra de la producción de bases de datos sobre mainframes, sin embargo actualmente es frecuente que se les instale un manejador de bases de datos relacional (RDBMS) donde se le pueda acceder fácilmente por medio de SQL o herramientas generadoras de SQL(COD90).

Mientras algunos productos de software para queries son desarrollados por la misma compañía que desarrolla el RDBMS y específicamente accede datos de esa base de datos hay otros que acceden a otros RDBMS, lo cual es un gran beneficio por que de esta manera no dependo exclusivamente de un RDBMS y la competitividad los hace mejor cada día.

También se tiene la posibilidad de tener diferentes bancos de datos por seguridad, rapidez, etc., los cuales pueden ser manejados por diferentes RDBMS y con una misma herramienta query es posible obtener información de cualquier RDBMS gracias a la gran contribución hecha por SQL.

Además de los RDBMS también existen varias bases de datos desarrollados para trabajar sobre PCs, las cuales no suelen ser sistemas relacionales, pero por lo general permiten a los usuarios expresar un requerimiento en términos de SQL y posteriormente el software convierte el requerimiento en el formato que la base de datos pueda entender.

### ***1.4.2 Servidor.***

En la arquitectura Cliente/Servidor el servidor back-end maneja los recursos de información, es decir, almacena recupera y protege los datos. El servidor revisa los datos requeridos y si es necesario, genera y manda los datos solicitados a un cliente u otro servidor.

#### **1.4.2.1 Características.**

Cuando se elige una máquina como servidor deben de contemplarse algunas características técnicas como el multiprocesamiento, multihilos y arreglos de discos. Hay que tomar en

cuenta que la mayoría de las veces, si no es que siempre, los servidores juegan un papel crítico, donde una falla podría ser catastrófica para cualquier tipo de empresa, es por esto que además se deben de contemplar redundancia de algunos componentes tales como manejadores de discos, fuentes de energía, discos espejo, etc.

#### **1.4.2.1.1 Multiprocesamiento.**

Los fabricantes están incluyendo múltiples procesadores en su hardware para incrementar la velocidad de procesamiento. Varios procesadores nos permiten realizar el multiprocesamiento simétrico o funcional.

Con el multiprocesamiento simétrico se puede asignar dinámicamente una tarea a un procesador. Los recursos de procesamiento son maximizados. Cuando hay una tarea por realizar un procesador desocupado la lleva a cabo.

El multiprocesamiento funcional permanentemente asigna un conjunto de tareas a un procesador. Consecuentemente, un procesador puede "flojear" cuando otro puede estar sobrecargado de trabajo.

#### **1.4.2.1.2 Multihilos.**

Un hilo es la unidad más pequeña de ejecución que el sistema puede listar. Cada hilo consiste de una pila, un apuntador, una prioridad, el estado del CPU y una entrada en la lista de espera del sistema.

Un hilo puede ser bloqueado, agendado para ejecución o ejecutado. Los hilos se comunican a través de mensajes y compiten por adueñarse de varios semáforos los cuales controlan la asignación de recursos de cómputo.

Los sistemas operativos antiguos llevaban a cabo la multitarea para crear múltiples procesos, pero esto desarrollaba overhead, sin embargo con el ambiente de multihilos, un proceso es dividido en tareas independientemente ejecutables(hilos). Estos hilos

---

colectivamente realizan el trabajo permitiendo a las aplicaciones realizar muchas tareas simultáneamente[DAW93].

#### **1.4.2.1.3 Arreglos de discos.**

Los arreglos de discos son de gran utilidad para el usuario, debido a que pueden recuperarse de alguna falla reemplazando el manejador de disco dañado estando el servidor en línea.

### **1.4.2.2 Clasificación de Servidores.**

El servidor fue desarrollado para compartir periféricos costosos como impresora láser, lectores de CD ROM, etc. Además de los periféricos, también se desarrolló para compartir los datos y en cuanto a esto se pueden clasificar los servidores como[BOC94]:

#### **1.4.2.2.1 Servidor de archivos.**

Los servidores de archivos están muy orientados a I/O (entradas y salidas). Estos servidores colocan gran cantidad de información en sus sistemas de almacenamiento y pasan los datos sobre una red. Cuando los datos de un archivo son solicitados, el servidor transmite todos los registros al cliente, saturando de esta forma las redes.

Debido a que el servidor manda todo el archivo del cual se requiera alguna información y no solamente los registros específicos éste debe de tener una gran capacidad y un muy rápido acceso al disco duro.

#### **1.4.2.2.2 Servidor de aplicaciones.**

Este tipo de servidor se caracteriza por prestar servicios similares a los que presta un mainframe. Cuando las aplicaciones de un mainframe son cambiadas a otra plataforma tratando de reducir los costos, una de las opciones consiste en instalar las aplicaciones sobre una máquina menos grande que corra el mismo software.

#### **1.4.2.2.3 Servidor de bases de datos.**

Este es el uso más típico de servidor. Las aplicaciones corren sobre los clientes. El servidor de bases de datos acepta requerimientos y pasa el resultado al cliente.

La función de manejo de datos es del servidor y el cliente se encarga de ejecutar una aplicación específica.

El servidor de base de datos mejora el manejo de requerimientos y solo envía al cliente los datos que satisfacen el requerimiento. Esto es más eficiente en términos de la carga de la red que una arquitectura file server.

Mediante una herramienta llamada SQL se pueden procesar los registros en conjunto, una aplicación con una sencilla instrucción de SQL puede recuperar o modificar un conjunto de registros del servidor.

#### **1.4.2.3 Requerimientos.**

Un elemento básico y fundamental en un servidor en el ambiente cliente servidor se llama base de datos, como anteriormente lo comentamos es el elemento que se encarga de almacenar, recuperar, actualizar y eliminar los datos. Otras de sus funciones pueden ser las de realizar transacciones en línea, integridad, y contemplar los procedimientos de recuperación de la información ante una catástrofe.

Cuando seleccionamos una base de datos debemos tomar en cuenta la flexibilidad para emigrar y tener cierta independencia con respecto a la compañía dueña de la base de datos. Para ello se deben de contemplar los puntos que a continuación serán detallados[DAW93].

##### **1.4.2.3.1 Independencia de plataforma.**

La independencia de plataforma es un gran beneficio de la arquitectura cliente/servidor. Actualizar el hardware puede ser tan sencillo como respaldar los datos y posteriormente restablecerlos sobre una mejor máquina corriendo la misma base de datos. Es decir, cuando

migramos de una a otra computadora nuestro servidor de bases de datos debe tener la capacidad de funcionar tan bien como lo venía haciendo antes de llevar a cabo el cambio.

El software debe ser compatible entre plataformas, los datos, programas y front-ends deben necesitar tan solo unas modificaciones sencillas.

#### **1.4.2.3.2 Transacciones.**

Las transacciones es un requerimiento adicional a la base de datos. Las transacciones las genera el cliente y se las manda al servidor para su procesamiento. Frecuentemente las transacciones utilizan dos o más tablas que pueden residir sobre diferentes máquinas. Cuando el sistema llegue a tener una falla, el servidor de base de datos debe ser capaz de regresar o deshacer las transacciones procesadas (rollback) hasta antes del último respaldo (commit).

Una transacción es una o más operaciones que fueron ejecutadas para completar una tarea. Para que una transacción se considere exitosa, todas las operaciones deben de llevarse a cabo. Si cualquier operación de una transacción no se llevo a cabo, las demás operaciones de la transacción deben deshacerse.

Mientras se estén ejecutando las operaciones de una transacción, el sistema conserva un diario (log) de trabajo. Cuando la transacción termino satisfactoriamente el sistema guarda (commit) todos los cambios permanentemente. Si la transacción no termino satisfactoriamente, el sistema usa el log para restablecer (rollback) la base de datos al estado anterior a su ejecución.

#### **1.4.2.3.3 Conectividad.**

El servidor debe de proporcionar acceso a una gran variedad de fuentes de datos, es decir no se debe de "casar" con un servidor propietario, sino todo lo contrario nuestro servidor debe ser abierto. La mayoría de las bases de datos usan SQL como el lenguaje de acceso.

#### **1.4.2.3.4 Procedimientos almacenados (stored procedures).**

Son una colección de instrucciones SQL que son compiladas y almacenadas en el servidor de bases de datos. Cuando un comando SQL es mandado a la base de datos, el servidor analiza el comando, revisa la sintaxis, revisa la protección y genera la ejecución.

Los stored procedures permiten desarrollar queries y otros grupos de instrucciones. También pueden ser usados para forzar las reglas del negocio y la integridad de los datos. Aceptan parámetros, pueden ser utilizados por múltiples aplicaciones con gran variedad de datos y se ejecutan con gran rapidez.

#### **1.4.2.3.5 Disparos (triggers).**

Los triggers son procedimientos almacenados especiales que son invocados automáticamente por el servidor de bases de datos. Estos están asociados con tablas específicas, y se ejecutan cuando hay un intento por modificar los datos de dichas tablas. Pueden llevar a cabo validaciones complejas sobre los datos debido a que usan el poder de SQL.

#### **1.4.2.3.6 Optimizador.**

Un robusto servidor debe proporcionar un optimizador, el cual analiza una instrucción SQL y determina el camino más eficiente para procesar el requerimiento. Estos analizan los índices estadísticos de distribución y el tamaño de las tablas para determinar el proceso menos costoso.

#### **1.4.2.3.7 Herramientas de prueba y diagnóstico.**

Un área que continuamente necesita atención es la de prueba y diagnóstico. Cuando los datos y/o índices se corrompen por errores de sistema o sectores dañados del disco, las utilerías que diagnostican los problemas y ofrecen una recuperación son esenciales.

#### **1.4.2.3.8 Confiabilidad.**

Los servidores son una pieza central para los usuarios, si llegara a tener una falla por algún período de tiempo sería fatal para algunas empresas, por lo que se recomienda contemplar recursos extras como procesadores, fuentes de poder, manejadores de discos, discos espejo, etc., para tener un servidor tolerante a fallas.

#### **1.4.2.3.9 Mecanismos de respaldo y recuperación.**

Cuando se llevan a cabo procesos críticos de importancia para cualquier tipo de empresa ya sea pública o privada el prevenir pérdida de información es necesario por lo que los mecanismos de respaldo y recuperación, comúnmente en cintas o sobre otro servidor, son indispensables. Estos se deben llevar periódicamente dependiendo de la necesidad. diario, cada x días, cada semana, cada mes, etc..

### ***1.4.3 Comunicación Cliente/Servidor.***

Diferentes tipos de comunicación son usados por los sistemas cliente/servidor, los 3 más importantes son el conversacional, llamada de procedimiento remoto(RPC) y mensajes. SQL es de alto nivel y es usualmente construido sobre uno de los 3 anteriores (frecuentemente sobre RPC)[BOC94].

#### **1.4.3.1 Modelo Conversacional.**

Un ejemplo del modelo conversacional se encuentra en el APPC de IBM. En SNA, el término "conversation" se refiere a la comunicación entre procesos durante una sesión.

El modelo conversacional es particularmente apropiado para situaciones que involucran muchas interacciones por transacción entre 2 puntos, proporciona un control estricto para interacciones detalladas. Proporciona e implementa rutinas que permite ejecuciones traslapadas(overlap).

### **1.4.3.2 Modelo de Mensajes.**

Algunos productos que usan este modelo son Windows de Microsoft, Presentation Manager y CICS. Con Windows y Presentation Manager los mensajes son usados entre las interfaces de usuario y procesos windows. El concepto de mensajes es familiar para cualquiera que entienda los fundamentos de mensajes electrónicos( E-mail).

Usando mensajes, si un cliente requiere datos de un servidor de base de datos, el cliente formatea el requerimiento en un mensaje y a su vez el servidor proporciona la respuesta en la forma de un mensaje también.

Como lleguen los mensajes, son colocados en una cola; en cuanto se este listo para procesar un mensaje, este es borrado de la cola. Cuando los mensajes entran a la cola pueden ser ordenados de una de las tres formas siguientes: FIFO (primero que entra es el primero que sale) LIFO (ultimo que entra primero que sale) o de acuerdo a la prioridad asignada al mensaje.

Un cliente puede mandar un mensaje a cualquier número de servidores. Los manejadores de colas son necesarios en ambos, tanto el cliente como el servidor.

Los mensajes han sido desarrollados por sistemas de procesamiento de transacciones en línea como IMS y CICS.

### **1.4.3.3 RPCs.**

Con RPCs, el procedimiento o subrutina se localiza sobre otra máquina en la red. Un RPC permite a un programa o un proceso sobre una computadora ejecutar un proceso sobre un sistema diferente que se encuentra en otra computadora remota. Los parámetros que sean necesarios son pasados cuando se llama al procedimiento.

El procedimiento remoto tiene código que entiende que hacer con la llegada de mensajes, como responder a estos, y como obtener los resultados requeridos para la llamada del procedimiento.

Hay una gran variedad de diferentes implementaciones de RPCs. Las herramientas generalmente proporcionan un lenguaje y un compilador que facilita la creación de código fuente portable así como un Run Time. El desarrollador de aplicación usa el lenguaje proporcionado por la herramienta RPC para crear una especificación describiendo aspectos de funcionalidad mientras el Run Time manipula el ambiente mandando y recibiendo datos a través de la red.

#### **1.4.3.4 SQL.**

SQL fue introducido por IBM, esta muy relacionado con las bases de datos relacionales las cuales fueron introducidas por el DR. E. F. Codd en 1970. Sql fue propuesto por primera vez en 1974[COD90].

SQL no es un lenguaje de programación completo. Este es usado principalmente como un lenguaje de definición de datos (DDL) y como un lenguaje de manipulación de datos (DML).

Algunas instrucciones SQL también sirven para la integridad de datos, seguridad, recuperación y administración de bases de datos.

# 1.5

## REDES

Las redes de área local (LAN) son en la actualidad productos comerciales que son utilizados en gran medida por diferentes dependencias públicas y empresas privadas en nuestro país y en la mayor parte del mundo. Las redes han tenido gran auge debido a que tienen la característica de compartir recursos como impresoras, scanners, medios de almacenamiento, procesadores, software, etc., además de compartir e intercambiar información con varios usuarios como las bases de datos y el correo electrónico (mail) sucesivamente.

También las redes cuentan con la capacidad de uso del concepto de distribución como es en el control de procesos de tiempo real, las estructuras de procesamiento en paralelo y las mismas bases de datos.

Después de haber mencionado algunas de sus características es importante mencionar que el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) es una organización que entre sus funciones está la de desarrollar normas en diferentes áreas y las LAN no han sido la

excepción. La IEEE ha desarrollado la norma 802 que se ha convertido en el estándar para el desarrollo de redes de área local (a nivel software).

La IEEE define a la red de área local como:

*Un sistema que comunica datos que permite un número de dispositivos independientes conectados directamente entre sí, con un margen de distancia moderada sobre canales físicos de comunicación. [MAR89].*

A continuación se presentan algunos conceptos generales sobre las LAN.

### ***1.5.1 Topologías.***

La topología de una red se refiere a la configuración física en que los dispositivos son conectados mediante un canal (usualmente cable). Las tres principales topologías son estrella, bus y anillo.

#### **1.5.1.1 Estrella.**

La configuración en estrella se caracteriza por un controlador central, al cual todos los nodos son directamente conectados. Todas las transmisiones de un dispositivo a otro pasan necesariamente por el controlador central el cual es responsable por todo manejo y control de la comunicación. Cuando un dispositivo quiere comunicarse con otro dispositivo el controlador central funciona como un conmutador (switch) cerrando un circuito. La gran desventaja de esta topología es depender en su totalidad del nodo central por lo que al fallar éste, falla toda la red. Esta topología es usada en gran escala en los sistemas telefónicos.

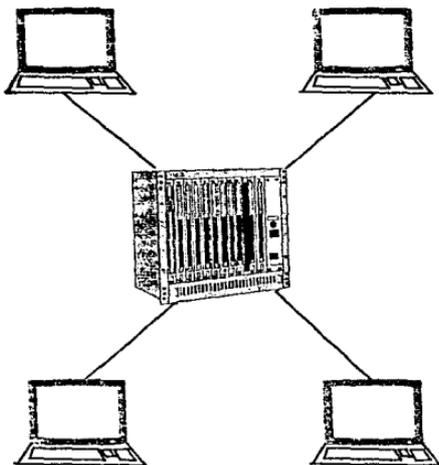


Ilustración 14 Topología estrella.

### 1.5.1.2 Bus.

En la topología de bus, cada dispositivo está directamente conectado a un canal de comunicación común. Las señales que son transmitidas sobre el canal son llamadas mensajes. Todos los mensajes pasan a lo largo del canal, por lo cual todas las computadoras se dan cuenta de lo que se transmite sobre el canal. El mensaje contiene una dirección la cual indica a que dispositivo se traslada en caso de que el nodo al que pasa el mensaje es el indicado por la dirección se procesa en caso contrario se ignora y el mensaje sigue su camino.

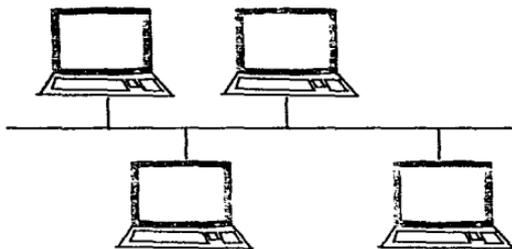


Ilustración 15 Topología bus.

### 1.5.1.3 Anillo.

En una configuración de anillo las computadoras se conectan entre sí, como su nombre lo indica. Los mensajes que se mandan alrededor del anillo deben estar específicamente dirigidos al nodo destino, debido a que las computadoras conectadas tienen acceso al mensaje, de similar forma que la configuración de bus.

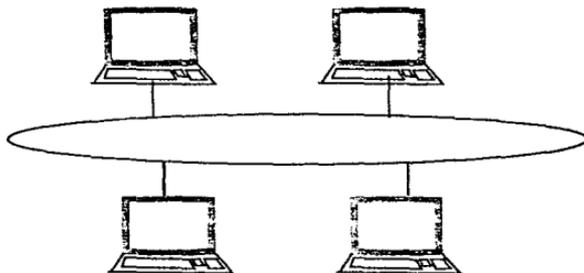


Ilustración 16 Topología anillo.

#### 1.5.1.4 Malla.

La topología se puede llevar a cabo en forma local sin embargo con frecuencia este tipo se utiliza en las redes amplias o metropolitanas (MAN, WAN) que necesitan conectar diferentes puntos en forma remota. Se utilizan ruteadores para seleccionar el mejor trayecto de un punto a otro punto a través de la malla.

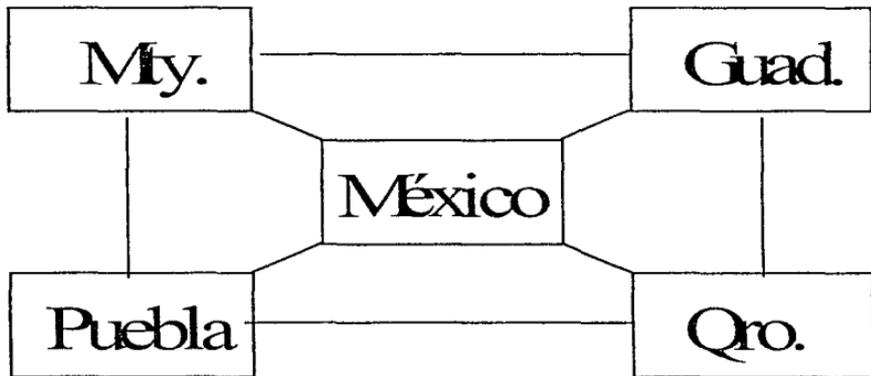


Ilustración 17 Topología malla.

#### 1.5.1.5 Soporte (backbone).

Los ruteadores o puentes administran el flujo de tráfico entre las subredes y el soporte. Se encuentran por lo general en grandes consorcios de oficinas o campus (el de C.U. es un ejemplo).

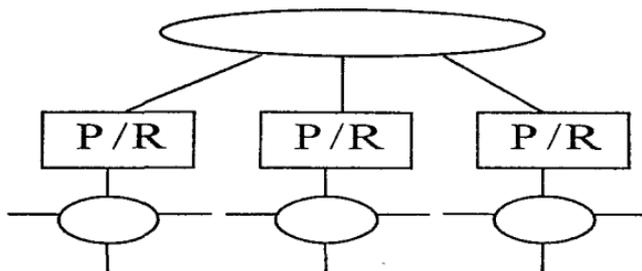


Ilustración 18 Topología soporta.

#### 1.5.1.6 Estrella jerárquica.

Los concentradores, de algunas áreas o departamentos a los cuales se conectan las computadoras en forma de estrella, se conectan a un concentrador central a su vez, también en forma de estrella el cual administra el tráfico entre ellos.

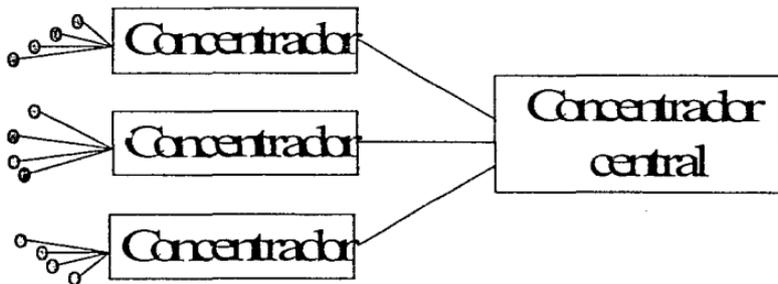


Ilustración 19 Topología estrella jerárquica.

## ***1.5.2 Medios de Transmisión.***

Para la implantación de LAN hoy en día son usados con más frecuencia tres medios de transmisión: par trenzado, cable coaxial y fibra óptica, sin embargo hoy en día hay otros medios como los inalámbricos.

### **1.5.2.1 Par trenzado.**

Un par trenzado consiste de dos alambres de cobre aislados y entrelazados. Una cantidad de pares trenzados pueden ser agrupados y cubiertos por una envoltura protectora, sin formar entre sí alguna interferencia. Este tipo de cableado es frecuentemente usado por las compañías de teléfonos su transmisión llega a 10 millones de bits por segundo (10Mbps). En este tipo de cableado se requiere una resistencia de 100 ohms (10BaseT), [SAN93].

### **1.5.2.2 Cable coaxial.**

El cable coaxial consiste de un conductor de cobre envuelto por un material aislante. El cable coaxial contiene una segunda capa, que rodea al aislante, comúnmente un conductor en forma de malla entrelazada. Por último se cuenta con una capa protectora de material no conductor. El cable coaxial es más vulnerable a la interferencia que el par trenzado, pero es capaz de soportar 100 Mbps. El cable coaxial ha sido usado principalmente por la industria de la televisión, para transmisión de sus programas. El cable coaxial que es usado debe tener una resistencia de 50 ohms. Hay dos tipos de este cable el delgado y el grueso la forma es semejante, mientras que en el alcance varían en el primero la transmisión sin ningún error puede llegar hasta 280m (10Base2), mientras que el segundo hasta 500m (10Base5), [SAN93].

### **1.5.2.3 Fibra óptica.**

La fibra óptica puede transportar señales utilizando rayos de luz modulada (pulsos). Una fibra óptica consiste de un cilindro de vidrio delgadísimo llamado núcleo (core), rodeada por una capa de vidrio y al igual que los demás con una capa protectora. La fibra óptica soporta un ancho de banda muy grande debido a que la luz tiene una frecuencia de 10<sup>14</sup>Mhz, la transmisión alcanza 565Mbps comercialmente hablando, pero se ha comprobado la transmisión de 200,000Mbps. Las señales transmitidas por la fibra óptica no están sujetas a interferencias eléctricas, sin embargo todavía los costos son extremadamente más caros que los cables eléctricos anteriormente mencionados. [TAN91].

#### **1.5.2.4 Inalámbrica.**

Este medio de transmisión se puede dividir en 2 grandes ramas:

- LAN inalámbrica y
- Móvil inalámbrica

La diferencia crítica radica en las facilidades de transmisión mientras que la comunicación LAN inalámbrica utiliza transmisores y receptores que se encuentran dentro de una área limitada, la móvil inalámbrica involucra a terceros como compañías de telecomunicaciones; los usuarios que se encuentran fuera de su área necesitan utilizar radios, celulares y/o estaciones de satélite.

El transmisor y receptor (transceptor) inalámbrico de una LAN se localiza en algún lugar fijo el cual recibe las señales de las estaciones de trabajo. Este medio de transmisión es recomendado cuando un equipo de trabajo requiere establecerse temporalmente en algún sitio y una vez terminado el proyecto trasladarse a otro lugar.

Una LAN inalámbrica requiere un equipo transceptor conectado a por lo menos un servidor mediante algún cableado (ethernet estándar), y el propio transceptor recibe y envía señales (datos) a las estaciones.

Se cuentan con 3 diferentes formas de transmisión inalámbrica:

- Luz infrarroja
- Radio de frecuencia simple
- Radio de espectro expandido

### 1.5.2.5 Microondas.

Este medio de transmisión también es inalámbrico. Como sabemos la comunicación por medio de microondas no requiere un medio físico, es posible transmitir a grandes distancias. Los enlaces por medio de microondas pueden ser los siguientes:

- Enlaces por medio de satélite.
- Enlaces de edificio a edificio en área metropolitana.
- Enlaces a través de suelos difíciles para tendido de cable.

La configuración de este medio de transmisión requiere de por lo menos dos antenas bidireccionales las cuales enfocan haces de energía electromagnética u ondas de radio de uno a otro punto, es decir la señal no se transmite de un lugar a muchos tan solo de punto a punto. Las antenas deben ser altas para que la transmisión no encuentre obstáculo alguno. La distancia entre las antenas no deben rebasar las 30 millas. Hoy en día se cuenta con rangos de frecuencia de 2 a 5 Ghz con el ancho de banda más alto[SHE95].

La comunicación por satélite transmite señales desde transeceptores ubicados en tierra a los ubicados en el espacio. Los satélites se sincronizan con la órbita de la tierra logrando un lugar "fijo" con respecto a la tierra. Las antenas consecuentemente se orientan al satélite para poder transmitir la señal por medio de haces de microondas. En cuestión de transmisión satelital se cuentan con 3 rangos de frecuencia[SHE95]:

- **Banda C.** Las frecuencias de los enlaces ascendentes son del orden de los 6 Ghz y de los descendentes son del orden de los 4 Ghz.
- **Banda Ku.** Las frecuencias de los enlaces ascendentes son del orden de los 14 Ghz y de los descendentes son del orden de los 11 Ghz.
- **Banda Ka .** Las frecuencias de los enlaces ascendentes son del orden de los 30 Ghz y de los descendentes son del orden de los 20 Ghz.

### ***1.5.3 Modelos de Interconexión.***

Las redes se organizan generalmente en niveles, evitando la complejidad en su diseño, cada nivel se construye sobre su predecesor y el objetivo de cada nivel es el de ofrecer sus servicios al nivel superior[TAN91].

#### **1.5.3.1 Modelo de referencia OSI (Open System Interconnection).**

El modelo de referencia OSI fue desarrollado por la Organización Internacional de Estándares (ISO). El propósito principal de este modelo es el de proporcionar las bases para el desarrollo coordinado de estándares con objeto de hacer flexible la interconexión de sistemas facilitando el intercambio de datos. Las reglas que se aplican para establecer las capas [TAN91], son las siguientes:

- Una capa se creará en situaciones en donde se necesita un nivel diferente de abstracción.
- Cada capa deberá efectuar una función bien definida.
- La función que realizará cada capa deberá seleccionarse con la intención de definir protocolos normalizados internacionalmente.
- Los límites de las capas deberán seleccionarse tomando en cuenta la minimización del flujo de información a través de las interfaces.

- El número de capas deberá ser lo suficientemente grande para que funciones diferentes no tengan que ponerse juntas en la misma capa y, por otra parte, también deberá ser lo suficientemente pequeño para que su arquitectura no llegue a ser difícil de manejar.

El modelo OSI esta compuesto por siete capas como se muestran a continuación:

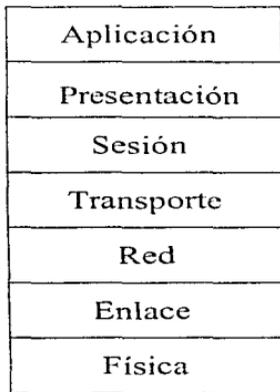


Ilustración 20 Modelo OSI.

Cada capa tiene sus funciones correspondientes [DE190].

- **Capa Física.** Esta capa se responsabiliza por la transmisión de flujo de bits a través de un medio de transmisión físico. Esto involucra una conexión entre dos máquinas.
- **Capa de Enlace.** La capa de enlace es responsable de proporcionar confiabilidad en la transmisión de datos de un nodo a otro y de proteger las capas más altas a

través del medio físico de transmisión, es decir, que la transmisión de tramas a través de un cable (o cualquier medio físico) sea libre de errores.

- **Capa de Red.** Esta capa tiene que ver con el ruteo de los datos de un punto de la red a otro. Esta es responsable por establecer, mantener y terminar la conexión de red entre dos usuarios y para transferir datos a lo largo de la conexión. Para conectar a dos usuarios puede haber muchas rutas, de las cuales para el establecimiento de la conexión solo se elige una en particular.
- **Capa de Transporte.** Cuando hay alguna conexión la capa de transporte es responsable de aceptar los datos de la capa superior, pasarlos a la capa de red y asegurar que lleguen correctamente a su destino.
- **Capa de Sesión.** Esta capa permite que los usuarios de diferentes máquinas puedan establecer dialogo entre ellos.
- **Capa de Presentación.** Esta capa se encarga de la sintaxis y la semántica de la información que se transmite, y puede incluir la codificación, conversión, compresión y expansión de datos.
- **La capa de Aplicación.** Contiene una gran variedad de protocolos que se necesitan para el intercambio de información o transferencia de archivos.

### 1.5.3.2 SNA.

El SNA de IBM es una arquitectura jerárquica, consiste de 7 niveles, cada nivel lleva a cabo una función específica. Los 7 niveles son organizados verticalmente, en orden ascendente. Los niveles son:

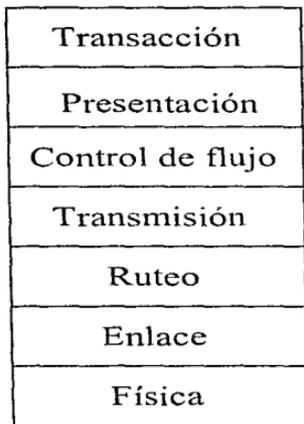


Ilustración 21 Modelo SNA.

Cada capa o nivel en la jerarquía del SNA participa en lo siguiente:

- Realiza servicios para el nivel alto inmediato.
- Requiere servicios del nivel bajo inmediato
- Se comunica con el correspondiente nivel en otra red SNA.
- Los cambios a algún nivel no afectaran a otros niveles.

Las capas o niveles del SNA se pueden dividir en dos grupos, el primero en el cual se consideran las 4 capas superiores: transmisión, flujo de datos, presentación y transacción que proporcionan las funciones.

La parte inferior se conforma con las capas: física, de enlace y ruteo, definen el control de ruteo y proporcionan las funciones y servicios del mismo.

Una SNA esta constituida por máquinas denominadas nodos. Estos nodos se clasifican en terminales; controladores, que son los que supervisan el funcionamiento de los periféricos; procesadores frontales, que reducen la carga de trabajo a los CPUs principales; los Hosts [TAN91].

Cada uno de los nodos contienen una o más unidades direccionables de red (NAU), estas son módulos de software que permiten a los procesos utilizar la red. Para que un proceso utilice la red se debe de conectar directamente a una NAU.

La estructura del modelo OSI y del modelo SNA son muy similares, ambos consisten de 7 niveles. Los niveles de ambos modelos tienen propiedades. Cada nivel o capa realiza una función específica, etc. Sin embargo el propósito del modelo OSI es diferente al del modelo SNA. El objetivo del modelo OSI es establecer reglas y orden en el mundo de la comunicación entre redes generando un estándar. Ahora bien, el modelo SNA esta diseñado para intercambiar información entre nodos de red que pertenecen a una única arquitectura. Esta única arquitectura es sobre la cual IBM construye varios de sus programas. La existencia de una única arquitectura permite a IBM conjuntar a la medida su software y hardware para lograr máxima eficiencia.

IBM reconoce la importancia del modelo OSI como un estándar común para la interconexión de sistemas heterogéneos.

Aunque OSI y SNA tienen el mismo número de capas no quiere decir que las capas correspondientes de ambos realicen la misma función.

### 1.5.3.3 TCP/IP.

Mientras que el modelo SNA fue desarrollado para satisfacer los requerimientos de red de los usuarios de IBM específicamente, investigaciones importantes se han hecho para desarrollar tecnologías que permitan interconectar redes heterogéneas.

La mayoría de estas investigaciones fueron financiadas por varias agencias del gobierno de los Estados Unidos. Estas organizaciones fueron las primeras en realizar y apreciar la importancia y potencialidad de la interconexión de redes.

Esta tecnología conocida como TCP/IP (Transmission Control Protocol/Internet Protocol) incluye un conjunto de estándares de red que especifica los detalles de comunicación.

TCP/IP forma la base tecnológica de la gran red Internet que conecta a la mayoría de los mejores centros de investigación, universidades y agencias gubernamentales y actualmente también a empresas privadas.

TCP/IP es una arquitectura de comunicación con 4 capas o niveles: Servicios de Red; Internet Protocol; TCP UDP; y de Aplicación.

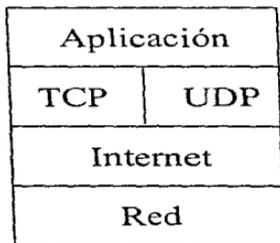


Ilustración 22 Modelo TCP/IP.

### 1.5.3.4 DNA.

En 1974 Digital Equipment Corporation (DEC) empezó el desarrollo de una arquitectura de red llamada Digital Network Architecture (DNA), al igual que SNA es propietaria.

DNA fue creada como un producto de red punto a punto. Esto significa que ningún nodo es controlador o propietario de la red, esto mismo, hace que la red sea muy flexible, sin embargo la administración suele ser muy compleja.

El DNA ha pasado por 5 fases, la última fue anunciada en diciembre de 1987 [BER92], y es muy similar al modelo OSI completamente compatible con las primeras 5 etapas del mismo OSI.

A continuación se muestran las capas que la conforman [SHL95]:

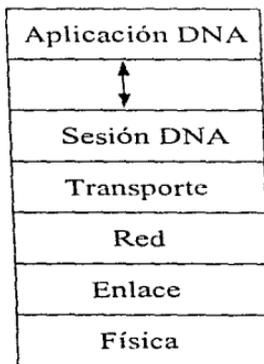


Ilustración 23 Modelo DNA.

### 1.5.3.5 Netware.

El modelo Netware esta constituido por 6 niveles o capas que a continuación se describen[SHL95]:

- **Física.** por medio de la cual se lleva la transmisión de la información de un equipo fisico de la red a otro.
- **Interfaz abierta de enlace de datos y especificaciones de la interfaz del controlador de red.** Esta capa como se muestra en la figura se divide en 3 partes la ODI y NDIS corresponden a la capa de enlace del modelo OSI, su función consiste en definir protocolos que envíen y reciban información entre unidades conectadas.
- **IPX (Internetwork Packet Exchange).** Similar a la capa de red del modelo OSI, el cual se encarga de definir los protocolos sin conexión que encaminan los datos en forma dinámica.
- **SPX (Sequenced Packet Exchange).** Comparando con el modelo OSI esta capa correspondería a la de transporte. Esta capa se encarga de que la información transmitida de un punto a otro llegue en el mismo orden en que fueron enviadas.
- **Conductos nominados y NetBIOS.** Mientras los conductos nominados se encargan de ejecutar los servicios de nivel de sesión en redes netware, NetBIOS se encarga de la gestión de la sesión de comunicaciones, la denominación de nodos, la difusión de los nombres y emplazamientos de los nombres de los servidores y la transmisión de los datagramas sin conexión.
- **NCP (Netware Core Protocol).** Este nivel se encarga de proporcionar acceso a los archivos, servicios de impresión, asignación de recursos, gestión, seguridad de la red además de la comunicación entre servidores.

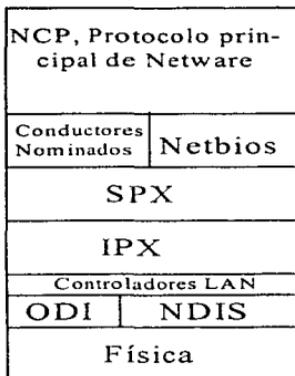


Ilustración 24 Modelo Netware.

### 1.5.3.6 LAN Manager.

LAN Manager utilizado por los productos del propio Microsoft para conexión en red y siendo en estos momentos Windows NT su principal impulsor. Fue creado por IBM en unión con Microsoft, sin embargo, IBM se retiró hace tiempo de esta asociación. A continuación se muestran las capas de este modelo y una comparación con el OSI[SHL95]:

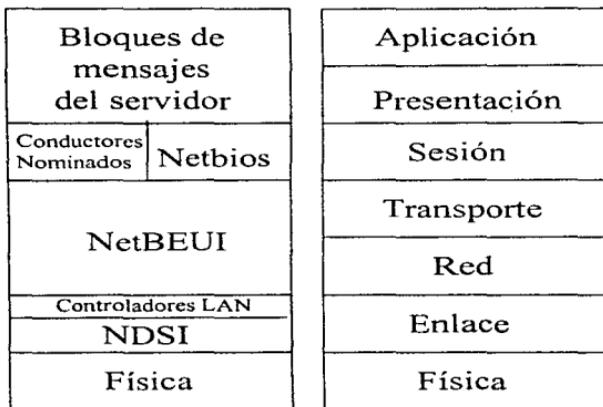


Ilustración 25 Modelo LAN Manager vs OSI.

### ***1.5.4 Protocolos de Control de Acceso.***

#### **1.5.4.1 Estándar IEEE 802.**

Debido a la gran variedad de protocolos de acceso que pueden construirse y los cuales pueden ser completamente incompatibles en 1980 en los Estados Unidos el Instituto de Ingenieros Electricistas y Eléctricos mejor conocido como el IEEE se dio a la tarea de normalizar un solo protocolo de acceso denominándole "802". Esta normalización trajo como consecuencia el gran desarrollo y auge de las redes locales.

##### **1.5.4.1.1 Estándar IEEE 802.3, CSMA/CD.**

Protocolo de Acceso: CSMA/CD [TAN91]. Los protocolos son usados en las topologías de bus y anillo debido a que los dispositivos están conectados a un mismo medio de

transmisión, por lo cual si no existieran dichos protocolos, en el caso de que dos o más computadoras transmitan al mismo tiempo la interferencia en el medio puede causar que una o varias transmisiones resulten dañadas. Es necesario seguir un protocolo de transmisión para la seguridad de la información.

Este protocolo propone que las computadoras conectadas a la LAN deben escuchar el medio físico antes de transmitir y esperar si existe una señal en el bus. CSMA/CD quiere decir Acceso Múltiple por Sensado de Portadora con Detección de Colisión (Carrier Sense Multiple Access / Collision Detection). En este protocolo, las computadoras escuchan el bus mientras transmiten para detectar una posible colisión, en caso de ser positiva la colisión abortan la transmisión y esperan un tiempo aleatorio antes de volver a sensar el bus. Para mayor explicación contemplemos lo siguiente: Si en un tiempo  $t_1$  una computadora A inicia una transmisión, poco tiempo después en  $t_2$  la computadora B desea enviar información pero encuentra el bus ocupado y espera. Más adelante, en  $t_3$ , otra computadora C también desea transmitir y encuentra ocupado el bus por la computadora A, por lo que también espera. En  $t_4$ , A termina de transmitir y el bus queda libre, esto lo notan las computadoras B y C e inician su respectiva transmisión. Como se ha mencionado anteriormente en CSMA/CD, la computadora que transmite también escucha la señal en el cable. Si lo que recibe es lo que está enviando todo va bien, en caso contrario, supone que ha ocurrido una colisión. Al detectar la colisión, la computadora transmisora aborta su información y en su lugar envía una pequeña señal reforzando la colisión para garantizar que las demás computadoras involucradas en la interferencia también puedan detectar la colisión, después de enviar la señal espera durante un intervalo de tiempo aleatorio y vuelve a intentar la transmisión de su información escuchando en el bus.

#### ***Formato de la trama.***

La información que se transmite a través de la red debe de contener además algunos campos a este conjunto se le llama trama. La trama no debe ser menor a 64 bytes eliminado el preámbulo y el delimitador de inicio de trama. La trama contiene lo siguiente:

Preámbulo 7	SFD 1	Destino 2 o 6	Fuente 2 o 6	Longitud 2	Info	FCS 4
----------------	----------	------------------	-----------------	---------------	------	----------

Ilustración 26 Trama CSMA/CD.

- **Preámbulo (7 bytes).** Es un campo de 7 bytes con el código 10101010. Al transmitir este byte en codificación Manchester, se genera una señal cuadrada que sirve para sincronizar a los receptores en la red.
- **Delimitador de inicio de trama (SFD: Start Frame Delimiter, 1 byte).** Un byte formado por el patrón 10101011. El último par de bits interrumpe la onda cuadrada formada por el preámbulo y los primeros bits de este byte. Esta interrupción sirve para indicar dónde en realidad inician los campos con información útil.
- **Dirección Destino (2 o 6 bytes).** Cada computadora en la red tiene un identificador único que es su dirección. Este campo contiene la dirección del sitio al que se envía la trama. Se utilizan 6 bytes para los campos de direcciones. Con esto de 2<sup>48</sup> valores con la intención de garantizar que ningún otro dispositivo en el mundo pueda tener una misma dirección.
- **Dirección Fuente (2 o 6 bytes).** Contiene la dirección de la computadora que generó la trama.
- **Tipo de trama o longitud (2 bytes).** En la mayor parte de los casos esto permite identificar cómo interpretar el campo de datos, pues indica el tipo de protocolo de capas superiores que lleva la trama. También este campo puede indicar el número de bytes válidos en el campo de información, tomando valores de 0 a 1500.
- **Información.** En este campo viaja la información que se transfiere de un sitio a otro. Como se ha mencionado anteriormente las tramas deben tener un tamaño mínimo de 64 bytes. Para determinar el tamaño de la trama se toman en cuenta todos los campos excepto el preámbulo y el delimitador, por lo que para LAN

con direcciones de 6 bytes, el tamaño mínimo del campo de información debe ser de 46 bytes. En caso de que la computadora transmisora quiera enviar menos de 46 bytes, se deben agregar caracteres de relleno (padding).

- **Secuencia de Verificación de la trama (FCS: Frame Check Sequence, 4 bytes).** El último campo es un código cíclico de redundancia (CRC) de 32 bits, calculado con los campos de dirección fuente y destino, tipo o longitud e información para tratar de predecir la integridad de la trama: la computadora calcula el CRC y lo añade a la trama cuando lo envía por la red, la computadora destino recibe la trama y repite el cálculo, si el valor calculado es diferente al recibido, se supone que la trama ha sido alterada por el ruido en el medio por lo que se descarta.

CSMA/CD ha sido apoyado desde sus inicios por Xerox, Intel y DEC.

#### **1.5.4.1.2 Estándar IEEE 802.4, token bus.**

El paso de testigo en bus trabaja de la siguiente forma: el token es transmitido de un equipo al siguiente. Cuando un equipo recibe el token este puede transmitir datos en un tiempo límite. Una vez terminada su transmisión o terminado su tiempo para el token al siguiente equipo. Si el equipo que recibe el token no tiene datos que transferir pasa el token de inmediato al siguiente equipo.

Hay que tomar en cuenta que existen en token bus prioridades por lo que las tramas intrascendentes esperan mientras que las importantes terminan su transmisión.

Aunque la red físicamente es un bus, lógicamente se configura como un anillo para el paso de testigo de un equipo al otro.

#### ***Formato de la trama.***

Preámbulo 1-n	SD 1	FC 1	Destino 2o6	Fuente 2o6	Info 0-819	FCS 4	FD 1
------------------	---------	---------	----------------	---------------	---------------	----------	---------

Ilustración 27 Trama token bus.

- Preámbulo (1-n bytes). Cada trama comienza con un preámbulo que es usado por la estación receptora para la sincronización. La longitud varía dependiendo del método de modulación.
- Delimitador inicial (SD: Starting Delimiter, 1 byte). Su longitud es de un byte y contiene una señal siempre distintiva de un dato.
- Control de Trama (FC: Frame Control, 1 byte). Este identifica el tipo de trama que esta siendo enviada
- Dirección destino (DA: Destination Address, 2 o 6 bytes). La dirección destino puede ser tan solo una dirección o un grupo de direcciones.
- Dirección Origen (SA: Source Address, 2 o 6 bytes). La dirección origen a diferencia de la destino es única. Ambas direcciones pueden ser de 2 o 6 bytes.
- Campo de Información (IF: Information Field, 0-819 Bytes). Este campo puede contener: una unidad de enlace (LLC), controlador de señal, manejador de datos o datos de propósito especial.
- Secuencia de Verificación de Trama (FCS: Frame Check Sequence, 4 bytes). Cuando el equipo emisor conjunta una trama realiza una revisión redundante cíclica (CRC), esta fórmula fue descrita por la IEEE la cual requiere 4 bytes para el resultado. El equipo emisor almacena este resultado en este campo. Cuando el equipo receptor recibe la trama realiza la misma fórmula que el emisor y el resultado lo compara con el de este campo (FCS) Si los resultados no son iguales, el equipo receptor asume que la transmisión tuvo algún error y puede nuevamente requerir que la trama sea retransmitida.

- Delimitador final (ED: Ending Delimiter, 1 byte). Es similar al delimitador inicial y marca el fin de la trama.

#### 1.5.4.1.3 Estándar IEEE 802.5, token ring.

Protocolo de Acceso: Paso de Testigo (Token Ring), [MAR89]. Hoy en día hay diversas técnicas para que en una topología en anillo se pueda controlar el acceso al medio, pero la más conocida se le denomina Token Ring. La IEEE en 1989 (revisión) dio a conocer el estándar 802.5 describiendo las características para una LAN del tipo Token Ring como: la topología en anillo, el cable par trenzado, aunque el uso de otros medios esta sujeto a consideración, el tipo de codificación Manchester diferencial, con velocidad de 4 a 16 Mbps y protocolo de acceso paso de token.

Las unidades de datos son transmitidas de una computadora a la próxima en una secuencia física a lo largo del anillo. Cada computadora transmite la unidad de datos a la próxima computadora actuando como un repetidor. La transmisión de unidades de datos esta controlada por un token. Cuando una computadora recibe el token, puede transmitir unidades de datos mientras un tiempo limite esta corriendo. Un token con una configuración que indica libre para transmitir es llamado free token (testigo libre). Cuando una computadora recibe un free token y tiene unidades de datos para transmitir, cambia la configuración del token de free token a busy token (testigo ocupado) e incluye el propio busy token con la unidad de datos que transmita. La unidad de datos viaja de computadora en computadora alrededor del anillo. Cada computadora que recibe una unidad de datos realiza un chequeo, en esta verificación se analiza el campo de control (FC) en caso de tratarse de una unidad de datos de tipo MAC, la información será procesada por todas las computadoras del anillo, si es de tipo LLC y coincide con la dirección destino es procesada. En caso de que no vaya dirigida a la computadora que en instancia tenga el busy token, lo retransmite a la siguiente computadora, y así sucesivamente hasta que encuentre su destino. Para evitar que una computadora transmita indefinidamente, se define un tiempo máximo de posesión del token, después del cual deberá emitirse un nuevo token que proporcione la oportunidad de transmitir tramas a las demás computadoras. Cuando una unidad de datos

retorna a la computadora que originalmente lo mando, esta remueve la unidad de datos de la red y manda un free token a través de la red a la siguiente computadora.

Dos condiciones de error pueden seriamente afectar la operación de una red mediante esta técnica: la pérdida del token y la persistencia del busy token. Para la detección y corrección de estas fallas, una de las computadoras conectada a la red debe de tener la función de Monitor Activo. La computadora asignada como Monitor Activo continuamente monitorea la red en caso de que el período de tiempo predeterminado transcurre y no se detecta inicio de token, el monitor asume que el token ha sido perdido e inicia un nuevo token. Para verificar una circulación indefinida de busy token, el monitor utiliza un bit de la trama conocido como bit M. Si el recibe una trama con el bit M=1, determina que la computadora que transmite no eliminó la trama del anillo y por lo tanto no liberó un nuevo token, por lo cual el Monitor Activo realiza estas dos ultimas funciones.

**Formato de la trama.**

A lo largo de la explicación del protocolo por paso de testigo se ha hablado mucho sobre las unidades de datos, la cual es sinónimo de trama, el cual contiene un encabezado que consiste de 5 partes (Delimitador de inicio, Control de Acceso, Control de Trama, Dirección destino y fuente), un campo de información y una secuencia de fin de trama que consiste de tres partes (Secuencia de Verificación de trama, Delimitador final, Estado de trama). Todas las partes de la trama serán detalladas a continuación.

SD 1	AC 1	FC 1	Destino 206	Fuente 206	Info	FCS 4	ED 1	FS 1
---------	---------	---------	----------------	---------------	------	----------	---------	---------

Ilustración 28 Trama token ring.

- Delimitador inicial (SD: Starting Delimiter, 1 byte). Es una señal única que identifica el inicio de la trama y esta formado por la siguiente secuencia de bits: "JKOJK000".

- **Control de Acceso (AC: Access Control, 1 byte).** Este byte esta formado por los siguientes bits "PPPTMRRR". Los bits PPP indican la prioridad del token o de la trama. El bit T tiene un valor de "0" en caso de tratarse de un free token. Cuando una computadora desea transmitir espera el free token y cambia el valor del bit a "1". El bit M se transmite con un valor de "0". Cuando el monitor activo lo retransmite, le cambia el valor a "1". Esto ayuda a la detección y corrección de situaciones en las que una trama de alta prioridad se transmite indefinidamente. Los bits RRR sirven para reservar la prioridad del siguiente free token.
- **Control de Trama (FC: Frame Control, 1 byte).** Esta formado de la siguiente cadena de bits "FFZZZZZZ". Los bits FF indican el tipo de trama (00=MAC, 01=LLC), los bits ZZZZZZ identifican un subtipo de la trama.
- **Dirección destino (DA: Destination Address, 2 o 6 bytes).** Todas las direcciones de la red deberán tener la misma longitud. El primer bit indica si se trata de una dirección individual o de grupo, mientras que el segundo bit, para el caso de direcciones de 6 bytes indica si se trata de una dirección local o universal.
- **Dirección Origen (SA: Source Address, 2 o 6 bytes).** Debe tener la misma forma y longitud que la dirección destino, y el primer bit debe valer "0".
- **Campo de Información (IF: Information Field).** Contiene la información destinada a las capas MAC o LLC. La longitud máxima es de 4.488Kbytes en el caso de una red que soporte 4Mbps y de 17.984Kbytes para una red diseñada para 16 Mbps.
- **Secuencia de Verificación de Trama (FCS: Frame Check Sequence, 4 bytes).** Este campo contiene información para detectar errores de transmisión. La secuencia se calcula a partir del FC, DA, SA y de información.
- **Delimitador de final (ED: Ending Delimiter, 1 byte).** Este byte contiene la siguiente cadena de bits "JK1JK11E". El bit 1 puede usarse para determinar el fin de transmisión de una estación. Para la transmisión de una serie de tramas el bit

I=1 en la primera trama e intermedias; I=0 cuando se transmite una sola trama o cuando se transmite la última trama. El bit E se transmite con valor igual a"0", cuando se detecta un error el bit E=1.

- Estado de la Trama (FS: Frame Status, 1 byte). Este byte contiene los bits de reconocimiento de la dirección y la trama copiada que son usados para indicar si la trama fue o no fue exitosamente recibida por la computadora destino.

### 1.5.4.2 Otros protocolos.

#### 1.5.4.2.1 FDDI.

La FDDI (Fiber Distributed Data Interface) [SAN93] es un estándar que fue desarrollado por el Instituto Nacional de Estándares de América (ANSI: American National Standards Institute). Esta basado para el uso de fibra óptica, control de acceso por paso de testigo y con capacidad para transportar 100Mbps. Con en el FDDI se busco principalmente:

- La Interconexión de mainframes y dispositivos de gran almacenamiento de datos, motivado por la necesidad de transferencia de datos a gran escala.
- Una alta rapidez en redes de oficina, debido al gran incremento de uso de imágenes y gráficos. Una típica transacción de procesamiento de datos es del orden de los 500 bits, mientras que una imagen puede requerir la transmisión de 500 000 bits o más.
- Una red como espina dorsal con alta capacidad que pueda servir para interconectar redes locales de menos capacidad.

#### Protocolo de Acceso: MAC

El acceso al medio está controlado por la posesión de un privilegio denominado token. Si ninguna estación en la red quiere transmitir, el token circula sobre el anillo. Cuando una estación quiere transmitir, espera el token, lo remueve del anillo y transmite su trama. La

alta velocidad de transmisión de FDDI, el tamaño máximo que pueden tener las tramas y la extensión física que puede alcanzar la red hacen que el tiempo de transmisión de una trama puede ser pequeño en comparación al retardo de propagación de la trama alrededor del anillo. Por esta razón para aumentar la eficiencia en el uso de la red, la estación que transmite pone en circulación un nuevo token inmediatamente después de terminar una transmisión.

A este esquema básico se agregan otros mecanismos de asignación del ancho de banda que permiten transmitir varias tramas consecutivas y multiplexar tráfico continuo y en ráfagas. Dado que en FDDI la estación que transmite no espera el regreso de su trama para transmitir un nuevo token, no puede utilizarse el esquema de prioridades de 802.5, basado en la reservación del siguiente token a una determinada prioridad utilizando el encabezado de la trama actualmente en circulación. En su lugar se utiliza una variante del algoritmo de asignación de capacidad del 802.4 (Token Bus).

A diferencia de 802.5, en FDDI no hay una estación particular encargada de monitorear el anillo. Esta función se distribuye entre todas las computadoras y tiene como objetivo detectar y corregir una condición inválida de funcionamiento en el anillo. La existencia de esta condición es detectada por las computadoras cuando no ven circular un token en un determinado intervalo de tiempo. Para corregir esta condición, las computadoras de FDDI ejecutan un algoritmo distribuido de elección, y la computadora ganadora inserta en el anillo un nuevo token. Si la condición inválida no puede resolverse debido a una ruptura en el anillo, las computadoras invocan un proceso de guía que finaliza hasta que el anillo se restaura.

#### ***Formato de la trama.***

Las tramas para este estándar constan de los siguientes campos:

Preambulo 16	SD 2	FC 2	Destino 4o12	Fuente 4o12	Info	FCS 8	ED 1o2	FS 1
-----------------	---------	---------	-----------------	----------------	------	----------	-----------	---------

Ilustración 29 Trama FDDI.

- Preambulo (P: Preamble, 16 símbolos idle o 64 bits). Es usado para sincronizar el reloj de cada computadora en la red con la transmisión. La trama envía este campo con 16 símbolos idle, los nodos que repiten la trama alrededor del anillo pueden cambiar la longitud de este campo de acuerdo a sus requerimientos de sincronización.
- Delimitador de inicio (SD: Starting Delimiter, 2 símbolos idle). Es una señal que identifica el comienzo de la trama. Consta de un símbolo J y otro K.
- Control de la trama (FC: Frame Control). Sirve para identificar el tipo de trama. Este tiene el siguiente formato de bits CLFFZZZZ, donde bit C especifica si se trata de una trama asincrónica o síncrona, el bit L especifica si las direcciones tienen una longitud de 16 o 48 bits, los bits FF indican si la trama es un LLC o MAC, y los bits ZZZZ proporcionan el control de información para tramas de MAC.
- Dirección destino (DA: Destination Address, 4 o 12 símbolos idle). El anillo puede contener una mezcla de computadoras usando direcciones de 16 bits (4 idle) o 48 bits (12 idle). La dirección destino puede ser individual, de grupo o para todos. Especifica la estación a la cual va dirigida la trama.
- Dirección fuente (SA: Source Address, 4 o 12 símbolos idle). Identifica la computadora que envió la trama.
- Información (IF: Information Field, 0-n símbolos idle). Contiene datos del usuario LLC o información de control MAC. La longitud máxima de una trama es de 4500 bytes.

- Secuencia de verificación de trama (FCS: Frame Check Sequence, 8 símbolos idle). Este campo contiene un valor de 32 bits. El valor se calcula tomando como base los siguientes campos: Control de Trama, Dirección Destino, Dirección Fuente e Información. La computadora que recibe la trama realiza el mismo cálculo y compara con el valor almacenado en FCS. Si al comparar no hay similitud la transmisión de la trama es considerada errónea.
- Delimitador de fin (ED: Ending Delimiter, 1 o 2 símbolos idle). Identifica el fin de una trama. Consiste de uno o dos símbolos de terminación (T).
- Estado de la trama (FS: Frame Status, 1 símbolo idle). Como su nombre lo indica contiene información, algunas banderas, sobre la trama como son si hubo detección de algún error, si las direcciones se reconocieron y si la trama fue copiada con éxito.

#### 1.5-4.2.2 ATM.

ATM (Asynchronous Transfer Mode - Modo de Transferencia Asíncrono) es una tecnología de transmisión de datos de alta velocidad (voz, datos, facsimil, vídeo en tiempo real, sonido e imágenes) en cables de fibra óptica.

Con ATM se pueden manejar aplicaciones en tiempo real, debido a su alto ancho de banda, su capacidad para dedicar un cierto ancho de banda a una aplicación y sus paquetes de tamaño fijo (celdas).

ATM puede utilizarse como un concentrador dentro de una organización que transmite rápidamente paquetes de un nodo a otro o puede servir como un dispositivo de comunicación de área extensa, que transmite celdas ATM entre LANs remotas a velocidades altas. ATM proporciona conexiones muchos a muchos y los nodos pueden transmitir simultáneamente.

Un conmutador ATM simplemente retransmite celdas. Examina la cabecera e inmediatamente empieza a enviar la celda. Se elimina el tiempo consumido por los métodos de almacenaje y de envío que usan los ruteadores.

Una red ATM contiene conmutadores ATM, los cuales son generalmente dispositivos multipuertos que realizan conmutación de celdas. Cuando una celda llega a un puerto, el conmutador ATM examina la información del destino de la celda y la envía por el puerto de salida apropiado.

Una razón para las altas velocidades de conmutación de los conmutadores ATM es que realizan sus operaciones de conmutación mediante hardware. Los conmutadores ATM evitan el nivel de red del protocolo OSI. ATM solo coloca la información en la celda y la envía. ATM es muy rápido debido a que no se realiza la verificación de errores. La estación receptora es responsable de la seguridad de todo lo que recibe de un emisor.

ATM es un protocolo de transporte que opera aproximadamente en el subnivel de control de acceso al medio (MAC) del modelo OSI.

En ATM no existen normas que limiten la velocidad de transmisión por lo que las velocidades de transferencia de ATM son escalables, dependen de la capacidad del nivel físico.

Originalmente se definió ATM como parte de la Red Digital de Servicios Integrados de Banda ancha (B-ISDN Broadband-Integrate Services Digital Network) que desarrollo el CCITT.

En 1991 se formo el Forum ATM el cual definió dos métodos de interfaz física:

- Interfaz de usuario a red (UNIs, User-to-Network Interfaces). El UNI es el punto de conexión de las estaciones finales a una red ATM.
- Interfaces de red a red (NNIs, Network-to-Network Interface) Consisten en interfaces de interoperabilidad de conmutadores ATM.

***El modelo de referencia B-ISDN.***

El nivel físico define las interfaces eléctricas y físicas, las velocidades de las líneas y otras características físicas. Lo interesante es que no se define ningún medio físico, pero se recomienda FDDI (100 Mb/s), canal de fibra (155 Mb/s), OC3 SONET (155 Mb/s) y T3 (45 Mb/s).

El nivel ATM define el formato de celda, el canal virtual, el trayecto y el control de errores. Las celdas tienen una longitud de 53 bytes, 48 de ellos los ocupa el campo de datos y los otros 5 la información de la cabecera.

La información contenida en la cabecera se divide en:

- **Control de flujo genérico** (GFC, Generic Flow Control), establece un modo para que múltiples estaciones de trabajo usen la misma interfaz de red de usuario (UNI, User Network Interface).
- **Identificador de Trayecto Virtual** (VPI, Virtual Path Identifier), identifica los trayectos virtuales entre usuarios, o usuarios y redes.
- **Identificador de Canal Virtual** (VCI, Virtual Channel Identifier), identifica los canales virtuales entre usuarios, o usuarios y redes.
- **Indicador de Tipo de Campo de Dato** (PTI, Payload Type Indicator), indica el tipo de información en el área de campo de datos, tales como información de usuario, red o gestión.
- **Prioridad de Pérdida de Celdas** (CLP, Cell Loss Priority), define como eliminar ciertas celdas si se produce la congestión en la red. Los campos mantienen un valor de prioridad, un 0 indica que la celda no puede suprimirse.
- **Control de Error de Encabezamiento** (HEC, Header Error Control), proporciona el control para la detección de errores y la corrección de un único error en un bit.

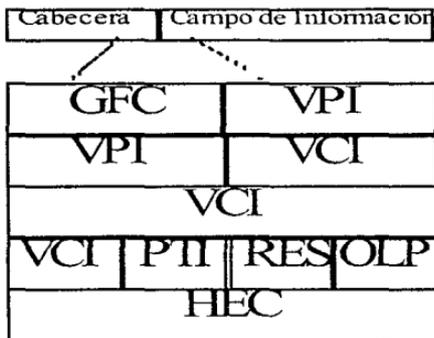


Ilustración 30 Campos de cabecera de la celda ATM.

El nivel de adaptación ATM (AAL, ATM Adaptation Layer) define el proceso de conversión de información de los niveles superiores en las celdas ATM, es decir, divide los paquetes de los niveles superiores en celdas de 48 bytes para su transporte. Este nivel se divide en dos subniveles: El subnivel de convergencia (CS, Convergence Sublayer) que acepta los datos del nivel más alto y los pasa al subnivel de segmentación y reensamblado (SAR, Segmentation and Reassembly), que es el responsable de la segmentación de la información en celdas ATM de 53 bytes. Si llegan celdas, el SAR reensambla los datos y los pasa al nivel superior. Existen varios tipos de AAL:

- Tipo 1 es un servicio de capacidad constante isócrono para las aplicaciones de audio y vídeo.
- Tipo 2 es una aplicación isócrona de capacidad variable como vídeo comprimido.
- Tipo 3-4 soporta ráfagas de datos de capacidad variable tipo LAN que soporta interfaces Frame Relay e SMDS.

- Tipo 5 soporta un subconjunto de funciones del tipo 3-4, proporciona modo de mensajes y operaciones no seguras.

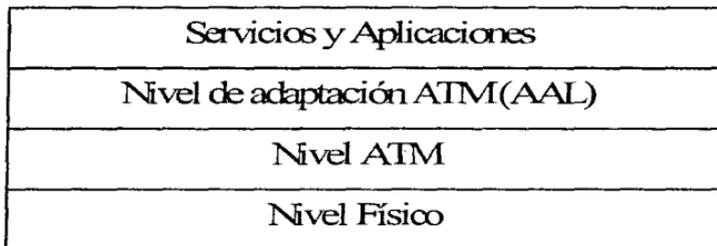


Ilustración 31 Modelo de referencia B-ISDN.

#### *Clases de servicios.*

- Clase A, es un servicio orientado a la conexión que proporciona una capacidad constante. Las compensaciones de sincronización lo hacen adecuado para aplicaciones de video y voz.
- Clase B, es un servicio orientado a la conexión y se sincroniza para la transmisión de capacidades variables de voz y video.
- Clase C, es un servicio orientado a la conexión de capacidad variable sin sincronismos adecuados para los servicios como X.25, Frame Relay y TCP/IP.
- Clase D, es un servicio sin conexión, con un tráfico de datos de velocidad variable que no requiere relaciones de sincronización entre los nodos finales.

#### 1.5.4.2.3 Frame relay.

Frame relay constituye un método de comunicación orientado a paquetes para la conexión de sistemas informáticos. Se utiliza principalmente para la interconexión de LANs y WANs sobre redes públicas o privadas y ofrece un ancho de banda entre 56 Kb/s y 1.544 Mb/s.

Frame Relay proporciona circuitos virtuales permanentes a través de una trayectoria predefinida que conecta dos puntos finales. Los canales permanecerán activos continuamente y garantizan las especificaciones que se han negociado con el cliente. Frame relay evita el control de flujo y la gestión de errores dentro de la propia red, cuando estos últimos se presentan Frame Relay asume que los nodos finales son máquinas programables que pueden realizar su propia gestión de errores. En Frame Relay los nodos intermedios (conmutadores) simplemente realiza una retransmisión de tramas a través de una trayectoria predefinida.

##### *Servicios de red.*

- **Mensajes de estado de los circuitos virtuales.** Este servicio proporciona la comunicación entre la red y el cliente. Asegura que el PVC exista e informa sobre los PVCs eliminados.
- **Multidistribución.** Este servicio es opcional y permite a un usuario enviar tramas a múltiples destinos.
- **Direccionamiento Global.** Este servicio opcional concede a una red Frame Relay capacidades semejantes a las de una LAN.
- **Control simple de flujo.** Es un servicio opcional que proporciona un control de flujo XON/XOFF a los dispositivos que así lo requieran.

Cuando una red Frame Relay alcanza la congestión, las tramas pueden descartarse arbitrariamente (los nodos finales son los responsables de la retransmisión de las mismas) o bien el rechazo se puede efectuar según las preferencias del cliente.

Una red Frame Relay conecta dos LANs sobre una red pública de conmutación de paquetes. El proceso es muy simple: la trama procedente de la LAN se sitúa en una trama Frame Relay y se distribuye a través del sustrato de red hacia el destino. Técnicas de multiplexión estadística entrelazan los datos procedentes de múltiples orígenes en la ubicación del cliente sobre una única línea.

**Formato de la trama.**

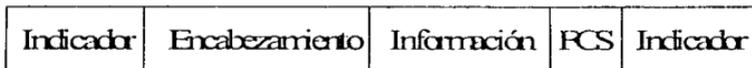


Ilustración 32 Trama frame relay.

La estructura de una trama Frame Relay está compuesta de dos campos en los extremos llamados indicadores que delimitan la trama mediante una secuencia especial de bits.

Siguiendo al campo principal se encuentra el encabezamiento que mantiene la información sobre direcciones y control de congestión. Este campo contiene la siguiente información:

- **Identificador de conexión de enlace de datos (DLCI, Data Link Connection Identifier)**, que contiene el número de identificación para la conexión lógica multiplexada dentro del canal.
- **Capacidad de elección de descarte (DE, Discart Eligibility)**, que establece las prioridades si una trama puede descartarse durante un congestiónamiento.
- **Notificación de congestión explícita hacia delante (FECN, Forward Explicit Congestion Notification)**, informa al ruteador que se recibe la trama que se ha experimentado congestión en el trayecto que ha atravesado ésta.
- **Notificación de congestión explícita hacia atrás (BECN, Backward Explicit Congestion Notification)**, añade información a las tramas que viajan en sentido

contrario para ayudar a los protocolos con el fin de emprender la acción adecuada para el control de flujo.

El siguiente campo corresponde a la información.

Después se añade un campo para la secuencia de verificación de trama (FCS, Frame Check Sequence), se realiza un cálculo con cada trama recibida y se compara con el campo FCS, que calculó el emisor. El paquete se elimina si no hay coincidencia y las estaciones finales deben de solucionar su pérdida.

# 1.6

## HERRAMIENTAS DE DESARROLLO

### *1.6.1 Servidores de Bases de Datos (Back-Ends o DBMS).*

Hoy en día existe una gran variedad de servidores de bases de datos que corren no nada más en UNIX sino también en otras plataformas como son DOS, OS/DOS, Netware, Windows NT, etc.

De esa gran variedad de servidores o manejadores de bases de datos solo hablaremos de 3 de los mejores que son: Oracle, Sybase e Informix además de mostrar los resultados de unas pruebas (Benchmark).

#### **1.6.1.1 Sybase.**

SYBASE SQL Server, tiene más herramientas de terceros que cualquier otra base de datos, es considerada como una de las líderes. SQL Server originalmente corre para plataformas UNIX.

Contiene procedimientos almacenados (stored procedures) y disparadores (triggers).

Microsoft se asoció con SYBASE, en 1992 ambos el de SQL Server de Microsoft y el de SYBASE contenían las mismas características, guardando su distancia uno para ambiente OS/DOS y el otro UNIX. Sin embargo, pasado un tiempo de armonía se dividieron y hoy en día cada una de estas bases de datos es desarrollada por sus propios programadores.

SYBASE es frecuentemente usado en grandes sistemas, los cuales requieren gran velocidad de procesamiento.

SQL Server es una arquitectura multihilos con su propio Kernel que tiene integrado un monitor de control de SQL.

Parte de SYBASE, Transact SQL es un conjunto de extensiones para ANSI-estándar SQL que es usado para escribir stored procedures, los cuales son compilados y almacenados en diccionarios de datos.

El producto integra funciones de seguridad que permiten a los administradores de sistemas, dueños de bases de datos dar o denegar acceso a usuarios específicos o grupos de usuarios.

SQL Server puede rápidamente recuperarse de fallas del sistema o caídas de red. El producto utiliza un Log de transacciones al cual se escribe cualquier modificación que sufra la base de datos.

SYBASE también proporciona la opción de generar discos espejos.

El API de SQL Server permite a los desarrolladores de Software construir aplicaciones cliente que trabajen con SQL Server y que además proporcionan una interface transparente. DB-Library, es un conjunto de funciones creadas en C que permiten realizar comandos de SQL para recuperar y actualizar datos. Las rutinas de DB-Library pueden ser usadas con C

---

o Cobol. DB-Library es una biblioteca con ligas estáticas bajo DOS y una biblioteca con ligas dinámicas bajo Windows y OS 2. Las aplicaciones pueden también ser desarrolladas usando ODBC's de Microsoft.

### **1.6.1.2 Informix.**

La base de datos Informix Online es considerada una de las mejores y frecuentemente es usada en sistemas de procesamiento con transacciones de misión crítica.

Informix Online opera bajo sistemas operativos UNIX. La compañía, desde que fue formada en 1989, se ha dedicado a los sistemas abiertos.

Informix Online proporciona alto performance y alta disponibilidad de información. Incluye características adicionales como disco espejo y mecanismos de recuperación rápidos.

### **1.6.1.3 Oracle.**

Un gran número de características hacen de Oracle un fuerte contendiente en el mercado de los servidores de bases de datos.

Primero, Oracle ha sido siempre conocido por su portabilidad. Esta base de datos corre sobre más hardware, software y redes que cualquier otra base de datos relacional.

Segundo, Oracle es soportado por más del 70% de front-ends que hay en el mercado.

### **1.6.1.4 Pruebas de desempeño (benchmark).**

A continuación mostraremos los resultados de pruebas realizadas a las 3 bases de datos más robustas Oracle, Sybase e Informix.

Para evaluar estas bases de datos se utilizó el benchmark AS'AP(ANSI-SQL Standard Scalable and Portable) para sistemas de bases de datos relacionales desarrollado originalmente por la Universidad de Cornell. Este conjunto de pruebas cubre un amplio espectro de las operaciones típicas de bases de datos[BR194].

Las pruebas que se realizaron son:

#### 1.6.1.4.1 Transacción aleatoria de escritura mezclada(Random Write Transaction Mix).

Esta prueba accede 6 tablas en la base de datos, simultáneamente hay una dura carga de trabajo y transacciones de lectura y escritura. Esta prueba realiza funciones simultáneas para borrar, insertar, seleccionar y actualizar al servidor de base de datos.

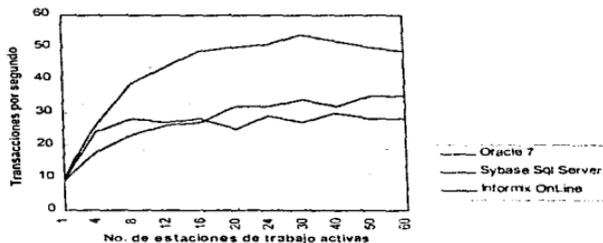


Ilustración 33 Transacción aleatoria de escritura mezclada.

En esta prueba se realizan 5 diferentes tipos de ejecuciones:

La primera transacción actualiza un campo entero en una tabla con 7 millones de renglones a través de la llave primaria usando un operador "entre" (between). La segunda transacción es una operación join (unión) entre dos tablas de un millón de renglones. La tercera transacción actualiza un campo entero en una tabla de mil renglones e incluye un almacenamiento de la actualización en una tabla vacía. La cuarta actualiza una tabla con 2 millones de registros con valores definidos y la quinta mueve 5 millones de registros uno por uno a una tabla vacía.

La mejor base de datos de esta prueba fue Oracle 7 siguiendo Sybase que implemento estas pruebas con stored procedures accediendo mediante llamadas a procedimientos remotos (RPCs). Al último llegó Informix utilizando bloqueos a nivel registro sobre todas las tablas con las que se trabajo..

#### **1.6.1.4.2 Lectura aleatoria única(Single Random Read).**

Esta prueba se basa en la lectura sencilla de un registro mediante la llave primaria, mostrando el número máximo de recuperaciones concurrentes que los sistemas puedan manipular.

En esta prueba cada workstation (con su manejador de base de datos) selecciona un renglón aleatoriamente de una tabla única. Las workstations repiten el proceso a la velocidad máxima que soporta la base de datos.

El mejor en esta prueba fue Sybase SQL Server debiéndose a sus índices cluster y al uso de stored procedures. Oracle 7 llegó en segundo lugar no utilizó stored procedure debido a la simplicidad de la transacción pero abrió y mantuvo un cursor. Oracle tiene la capacidad para compartir cursores a través de múltiples clientes, esto permitió a los clientes ejecutar las transacciones sin tener que optimar las instrucciones SQL.

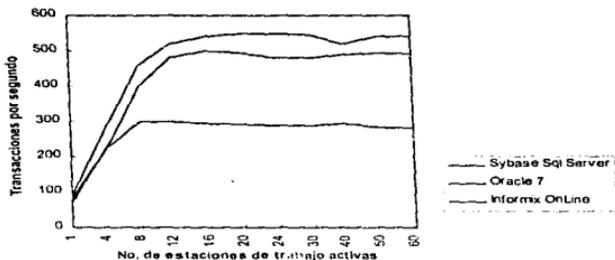


Ilustración 34 Lectura aleatoria única.

#### 1.6.1.4.3 Transacción aleatoria de lectura mezclada(Random Read Transaction Mix).

Esta prueba accede 5 tablas de la base de datos, simula una carga de trabajo mixta de consultas (queries) de solo lectura.

Fue diseñada esta prueba para dificultar la capacidad de recuperación de datos de la base de datos. Durante la prueba cada estación aleatoriamente seleccionó y ejecuto una serie de queries de un conjunto de 5 tipos de query.

El primer query es una lectura de registro mediante la llave primaria. La segunda es un join sobre la llave primaria entre dos tablas con un millón de registros. La tercera es una selección (select) de una tabla con 7 millones de registros usando valores definidos. El cuarto query es un join entre una tabla con un millón de registros y otra con 2 millones usando un operador between como restricción y un join sobre un campo carácter. El quinto

query también es un join entre dos tablas una con un millón de registros y la otra con 5 millones de registros usando valores definidos y es también sobre un campo carácter.

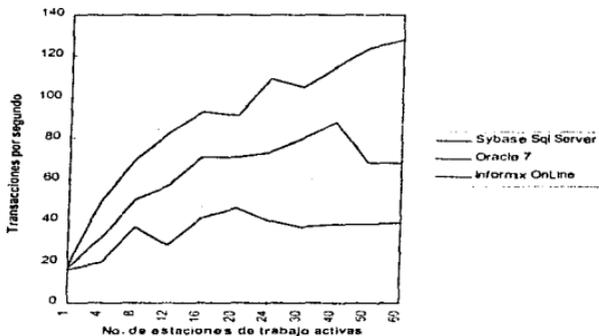


Ilustración 35 Transacción aleatoria de lectura mezclada.

El ganador por ventaja clara fue nuevamente Sybase SQL Server, su desempeño puede ser atribuible al uso de índices cluster y stored procedures (usando RPCs), le siguieron en orden descendente Oracle e Informix.

GRAFICA Sybase, Oracle, Informix.

#### 1.6.1.4.4 Recuperación de BLOBs ( Binary Large Objects).

Los BLOBs son estructuras utilizadas comúnmente para almacenar imágenes. La prueba de recuperación de BLOBs mide que tan rápido el cliente puede recuperar sus estructuras grandes, probando además que tan bien la base de datos hace uso de la red.

Los 3 productos ofrecen métodos para la obtención de las imágenes como BLOBs en una única llamada a la red. Esta prueba uso una tabla con 5000 imágenes diferentes en formato .GIF entre 20 y 150 Kb de tamaño por cada una. La mayoría estaba en el rango de los 70 Kb.

Oracle en esta prueba fue claramente el ganador con un promedio de transferencia máxima de 1.61 Mbps. Sybase SQL Server llegó en segundo con un promedio de transferencia poco abajo de 1 Mbps. De Informix no se lograron obtener resultados.

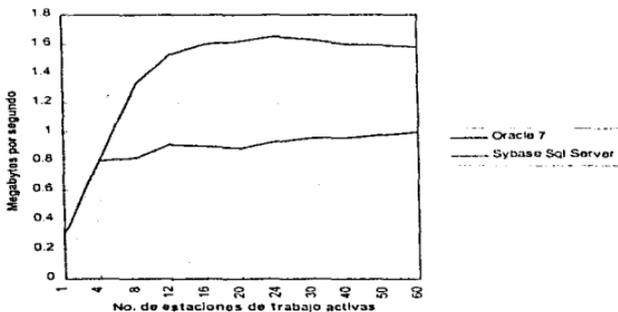


Ilustración 36 Recuperación de BLOBs.

#### 1.6.1.4.5 Consultas Ad hoc.

Esta prueba mide la efectividad del producto en un ambiente de toma de decisiones. El tiempo de respuesta es una medida importante en el mundo real en el cual un usuario requiere de forma urgente ver los resultados.

Esta prueba consiste de 34 queries que utilizan funciones como: selects, joins, proyecciones, adiciones, clasificaciones y subqueries.

En esta prueba gana Oracle pero Sybase llevo muy cerca y con más de un minuto de diferencia acabó Informix.

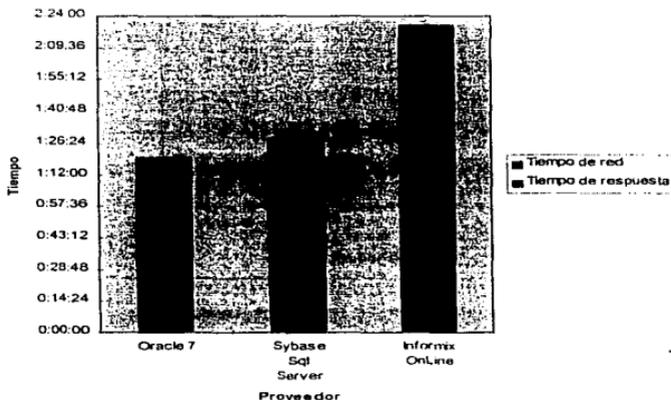


Ilustración 37 Consultas ad hoc.

#### 1.6.1.4.6 Cargar e indexar.

Esta prueba mide que tan rápido la base de datos puede importar 18.11 millones de registros y crear 33 índices, las tablas fueron cargadas consecutivamente

Oracle demostró ser muy rápida en esta prueba, seguida de Sybase, mientras Informix se quedó nuevamente al último.

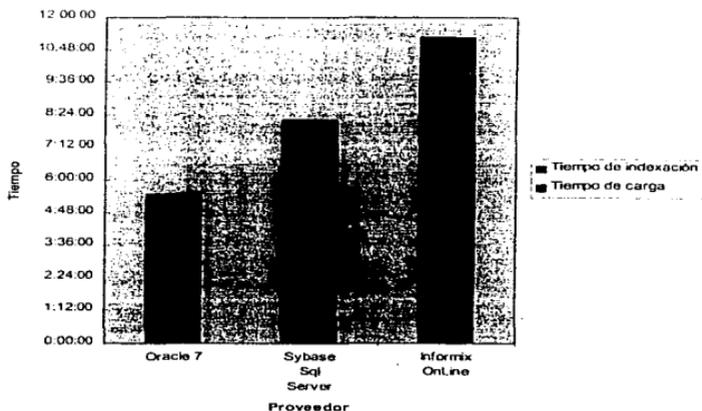


Ilustración 38 Cargar e indexar.

#### 1.6.1.4.7 Exportar.

Esta prueba mide que tan rápido una base de datos puede exportar una tabla con un millón de registros a un formato texto y delimitando cada campo con una coma de ASCII. La exportación se realizó a un disco local del servidor.

Oracle llegó nuevamente en primer lugar sin embargo Informix por primera vez llegó en segundo lugar y al último Sybase.

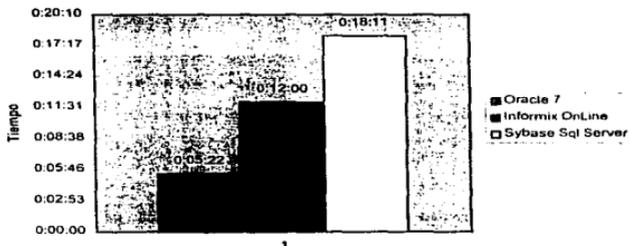


Ilustración 39 Tiempos de exportación.

## 1.6.2 Herramienta de Desarrollo para el Usuario Final o Front-End.

Una base de datos cliente/servidor es una combinación de hardware y software 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 software de desarrollo, -también llamados front- ends-, disponibles por parte de terceros.

Se puede clasificar a los front-ends en cuatro grandes categorías: add-ons a productos ya existentes, herramientas de desarrollo de aplicaciones, reportadores y herramientas de análisis e integración de datos.

A menudo es difícil determinar a qué categoría pertenece cierto producto, ya que no es raro que un front-end tenga características que lo hagan estar en más de una clasificación; por ejemplo, un buen programa de análisis de datos puede tener un buen reporteador.

Para hablar sobre las distintas clasificaciones, mencionaremos que los add-ons a productos ya existentes son módulos que permiten que una aplicación de PC, consulte al servidor de bases de datos. Ejemplos de add-ons son los existentes para DBASE, Paradox, Access, Superbase, Q&A, Advanced Revelation, DataEase, Clarion, y hojas de cálculo tales como Lotus 1-2-3 o Excel, etc.

Las herramientas de desarrollo de aplicaciones son usadas principalmente por los programadores y están diseñadas para facilitar el proceso de creación de aplicaciones front-end a la medida.

Los reporteadores también permiten hacer consultas no planeadas a la base de datos; facilitan la creación de consultas y reportes al back-end a los no programadores. Ejemplos de reporteadores son R&R de Concentric Data Systems, Crystal Reports, Report Smith e Infomaker

Se mostrará las características más relevantes de las herramientas front-end de desarrollo más utilizadas en nuestros días y sobre 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 utilizará en la Facultad de Derecho), ejecutándose en plataforma Unix. Para la elección del producto adecuado es importante la facilidad de aprendizaje y uso, poder de la herramienta y recursos que consume.

Las herramientas de análisis e integración de datos están diseñadas para que los tomadores de decisiones examinen los datos a partir de diferentes fuentes para así construir cuadros de decisiones complejas. Como ejemplo tenemos LightShip, InfoAlliance y Forest & Trees.

A pesar de que las aplicaciones front-end están disponibles para casi todas las plataformas, la mayoría soporta, principalmente, el ambiente DOS, Windows y OS/2.

Para la elección de un front-end adecuado a nuestras necesidades se requiere de algunos puntos de criterio. Describir todos los productos existentes en el mercado llevaría mucho tiempo, por lo que solo se mostrarán las características de los más comunes: Delphi, Omnis 7<sup>1</sup>, PowerBuilder, SQL Windows y Visual Basic. Estas herramientas cumplen con los siguientes criterios:

- Son herramientas de desarrollo visual.
- Están orientadas a ambientes cliente/servidor.
- Corren bajo la plataforma Windows 3.x (aunque pueden correr bajo cualquier otra)
- Permiten la conexión a distintos servidores de datos.

### 1.6.2.1 Funcionamiento de los front-ends.

Las aplicaciones cliente se ven y se ejecutan igual que cualquier otra aplicación que el usuario 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 un front-end de un servidor remoto de bases de datos se da cuando tiene que dar tanto su clave como su password para entrar en sesión con dicho servidor.

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

- (Cliente) El usuario crea su consulta (query).
- (Cliente) El front-end 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 al front-end.
- (Cliente) El front-end recibe la respuesta y la formatea para su presentación al usuario.
- (Cliente) El usuario visualiza y/o manipula los datos y/o re inicia 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.
- Cuando el cliente es un add-on a una aplicación ya existente, la secuencia de eventos para el procesamiento de una consulta se complica un poco.
- (Cliente) El usuario crea su consulta (query) en el lenguaje nativo de la aplicación.
- (Cliente) El add-on traduce la consulta a 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 al front-end.
- (Cliente) El front-end recibe la respuesta y la traduce al formato nativo de la aplicación.
- (Cliente) El usuario visualiza y/o manipula los datos en el lenguaje nativo de la aplicación.

Todas las traducciones que se requieren con un add-on requieren mayor capacidad de memoria y de proceso en el cliente, por lo que es normal requerir mayores recursos de hardware.

No todas las aplicaciones front-end pueden acceder todas las marcas de bases de datos. Lo anterior es resultado de los diferentes "dialectos" de SQL que existen y de la diversidad de protocolos de comunicación. SQL no es tan estándar; cada proveedor de bases de datos le agrega extensiones únicas o ciertas interpretaciones de SQL, que hacen que cada versión sea ligeramente incompatible con otras versiones de proveedores distintos. Por lo que respecta a las comunicaciones, cada DBMS utiliza un protocolo de comunicaciones distinto para enlazar a los clientes con el servidor de bases de datos, por lo que se requieren APIs (application programming interfaces) apropiadas en el software cliente para poder "platicar" con el driver de comunicaciones del DBMS.

### **1.6.2.2 Herramientas para el desarrollo de aplicaciones.**

Escribir programas en lenguajes de tercera generación no es la manera más fácil de desarrollar la mayoría de las aplicaciones front-end. Usando una herramienta de desarrollo de aplicaciones, se simplifica de manera significativa el proceso de creación de una aplicación cliente/servidor. Una herramienta de desarrollo de aplicaciones es un paquete de software especialmente diseñado para crear aplicaciones front-end.

Estas herramientas se hacen cargo de incluir en el código del programador algunas rutinas de bajo nivel que permiten la interacción con el hardware o con el servidor de bases de datos. De esta forma, el desarrollador de aplicaciones tiene más tiempo para concentrarse en el diseño de la misma aplicación, y en la interfaz con el usuario, reduciendo el tiempo total de desarrollo y, por tanto, reduciendo costos.

Cada proveedor de bases de datos cliente/servidor tiene su propio conjunto de herramientas (o "toolkit") para crear aplicaciones, por ejemplo, Sybase APT Workbench, Oracle SQL\*Forms o INGRES/Tools. Sin embargo, estas herramientas normalmente están restringidas a la creación de aplicaciones para el proveedor que las comercializa.

Es creciente el número de terceros que han venido generando paquetes de desarrollo que pueden conectarse a más de un servidor de bases de datos. La mayoría corren sobre Windows lo cual, desafortunadamente, puede constituirse en un problema ya que muchas organizaciones tendrían que actualizar sus PC's para poder correr Windows. Claro está que dicha actualización tendría que justificarse con algo más que correr una sola aplicación.

### **1.6.2.3 Evaluando herramientas para el desarrollo de aplicaciones.**

Una primera e importante consideración es el sistema operativo y el hardware con el que cuenta la organización, particularmente si el ambiente de cómputo es heterogéneo, con PC's y sistemas UNIX que requieran tener acceso a la base de datos. Para la Facultad de Derecho esto no es ningún problema ya que su ambiente de cómputo es heterogéneo y cuenta además con sistemas UNIX; sin embargo hay que considerar los recursos de hardware que requiere la herramienta de desarrollo; muchas de ellas corren en una PC con procesador mínimo 80386, y con un mínimo de 8 MB de RAM.

También es necesario pensar que el tipo de interfaz de usuario que se pretende utilizar es gráfico para sistemas PC, con plataforma de Microsoft Windows 3.x y Windows 95.

En un ambiente gráfico, las herramientas para desarrollo de aplicaciones adquieren especial importancia, ya que son las responsables de manejar los detalles para que la aplicación corra en una interfaz gráfica en particular; esto se traduce en que, por ejemplo, el programador no tenga que aprender a crear pantallas de usuario en dicha interfaz.

Es importante elegir una herramienta de desarrollo que los programadores del equipo de desarrollo ya conozcan, ya que tener que aprender un lenguaje incrementa el tiempo que un programador necesita para diseñar y desarrollar la aplicación. También puede darse el caso de que la herramienta de desarrollo utilice su propio lenguaje de programación, y que no tenga interfaz con ningún 3GL.; en este caso, puede ser que dicho lenguaje "propietario" de

la herramienta sea muy poderoso y que se puedan crear aplicaciones muy complejas con él, pero al equipo de desarrollo le tomará un tiempo adicional aprenderlo.

Considerar si la aplicación final creada con la herramienta en cuestión puede ser compilada para que corra por sí misma, o si será necesario comprar una copia de una versión "recortada" de la herramienta, usualmente llamada versión *run-time*, para cada usuario final. Si este fuera el caso, el costo final de la aplicación se vería seriamente afectado, por lo que es preferible que la herramienta para desarrollar aplicaciones genere ejecutables, lo que facilita la distribución de aplicaciones.

Asegurarse que la herramienta para desarrollo de aplicaciones soporte los otros servidores de bases de datos que puedan existir en el ambiente de cómputo, es decir, si es multi plataforma.

Y finalmente tomar en cuenta la importancia del soporte técnico del producto. Cabe resaltar que no importan cuán bueno sea un producto, si no cuenta con el soporte técnico adecuado, puede resultar una mala adquisición.

#### **1.6.2.3.1 Delphi 2.0 (Borland).**

##### ***Requerimientos del sistema:***

Windows 95 or Windows NT, 386 CPU, 8MB de RAM, 35MB de espacio en disco duro y unidad de CD-ROM.

Utiliza como lenguaje de programación, Object Pascal, por lo que muchos programadores pueden utilizarlo sin mucho entrenamiento. Delphi es la primera herramienta en ofrecer un alto desempeño en código nativo compilado con rapidez de ejecución y con la capacidad de acceder a bases de datos para cliente/servidor. Tiene una alta productividad, ya que permite re utilizar el código logrando un producto sumamente competitivo.

##### ***Desventajas.***

Delphi cuenta con un punto negativo en relación a la desaparición de la compañía Borland, así como el tipo de ambiente que es (Low End Client) lo que lo hace poco competitivo en el desarrollo de aplicaciones de gran tamaño.

***Observaciones.***

Delphi es la más novedosa herramienta en el mercado, y presenta características que ningún otro posee.

**Utillerías extra disponibles con el producto.**

Un kit de desarrollo para un servidor de base local. El reporteador Report Smith SQL. Herramientas para desarrollos en equipo. Constructor de consultas visuales. Una librería de componentes visuales (VCL), con código.

***Servidor de DB SQL local.***

Incluye una versión individual del Engine SQL de Borland, Interbase.

***Reporteador.***

Report Smith es un producto adquirido por Borland con el objeto de ofrecerlo junto a sus herramientas de desarrollo. Es uno de los mejores en el mercado, sobresaliendo por el tamaño (más de 10 Mb.) que ocupa.

**Interface con el programador.**

***CUA.***

Se ajusta a la interface de Windows, con controles 3-D, siendo uno de los que tiene la interfaz más intuitiva de los productos en el mercado.

***Modo de Programación.***

Delphi introduce el concepto de "lenguaje de dos vías", en el cual la programación visual va junto a la programación normal, de forma que existen 2 formas diferentes pero equivalentes de desarrollar una aplicación.

#### **Lenguaje de programación .**

Object Pascal. Es similar a Borland Pascal, pero ha sido escogido por su alto desempeño en el desarrollo visual. Dentro de sus características incluye Exception Handling, información a tiempo de corrida, y métodos de tablas virtuales, lo que lo hace uno de los mejores lenguajes orientados a objetos que existen.

Es un dialecto de Pascal, altamente compatible con Borland Pascal 7. El lenguaje es estructurado orientado a objetos, es decir, incorpora las características de programación de un lenguaje 4GL y el desempeño y potencia de un lenguaje 3GL. Además cuenta con un depurador gráfico que permite la rápida localización de problemas en tiempo de ejecución.

#### ***Tipo de código generado.***

Código ejecutable autónomo de 32 o 16 bits, de forma que es posiblemente el más rápido generado por herramientas visuales de la actualidad. No es necesario contar con intérpretes de DLLs. Cuenta con generación de errores de sintaxis, de ligado o de ejecución.

#### ***POO.***

Object Pascal soporta todos los puntos básicos de la POO, además de aspectos avanzados como la excepción de errores, lo cual lo hace uno de los más completos en el mercado.

#### ***POE.***

Facilidad de crear nuevos eventos en clases creadas por el programador, además de soporte a todos los eventos de Windows

#### ***Biblioteca o repositorio de componentes.***

Cuenta con una Librería de Componentes Visuales de 32 bits (VCL32) que ya tiene implementados (aproximadamente 100) o se puede ir añadiendo componentes del desarrollador. Algo importante que hay que resaltar aquí es que provee del código fuente para los VCL.

***Distribución de aplicaciones.***

Es posiblemente el más simple de distribuir, al crear ejecutables autónomos sin necesidad de run engines o librerías.

**Capacidades de interconexión.**

***Método de conexión a servidor SQL.***

La edición Client/Server cuenta con drivers nativos para los más conocidos DBMS del mercado, así mismo como conectividad a través de IDAPI, el estándar de conectividad impulsado por Borland.

También es posible conectarse a un servidor por medio de ODBC.

***Facilidad de migración.***

Muy fácil, gracias al concepto de alias (usado extensivamente en Paradox), es posible que no se necesite ni recompilar el código para migrar de una fuente local a una remota.

**Desarrollo de aplicaciones con interconexión.**

Soporte para OLE, a nivel servidor y cliente, con soporte para automatización, DDE (Dynamic Data Exchange), DLLs (Dynamic Link Libraries), VBX (Visual Basic Controls) y OCX (OLE Custom Control).

**1.6.2.3.2 OMNIS 7 3.0 (Biyth Software).**

Este producto se distribuye únicamente en CD-ROM, y es Multiplataforma.

**Requerimientos de la plataformas de Desarrollo.**

En una PC compatible, configuración mínima procesador 386, 8 MB de memoria y Windows 3.1. Idealmente una 486 con 16 MB de memoria.

**Ventajas.**

Permite desarrollar aplicaciones en una plataforma, y posteriormente migrarlas a toda una variedad de plataformas diferentes sin necesidad de modificar el código.

Proporciona una múltiple variedad de formatos para la elaboración de gráficos; éstos pueden ser escogidos y modificados por el usuario de acuerdo a sus preferencias y necesidades.

**Desventajas.**

La gran cantidad de operaciones y beneficios que proporciona OMNIS 7, hacen que éste sea una herramienta que consuma bastantes recursos físicos para que pueda trabajar adecuadamente.

No posee una interface común de usuario. Ya que en la manipulación de ventanas cuando están activas varias al mismo tiempo, algunas teclas iguales tienen un significado diferente para cada ventana.

Presenta algunas inconsistencias en cuando al manejo de su ayuda y ésta no es muy completa. No siempre es factible obtener la información deseada de las aplicaciones que se realizan, ya que en ocasiones marca errores irreconocibles.

Salva automáticamente todas las modificaciones que se realicen, lo cual a veces presenta problemas, ya que no siempre desean guardarse todos los cambios que se hacen.

Es extremadamente difícil de instalar. Quizás su mejor plataforma no es PC, ni de desarrollo ni de ejecución.

***Observaciones.***

Perteneciente a una categoría diferente, no parece ser competencia para los demás productos.

**Utilerías extra disponibles con el producto.**

***Reporteador.***

Cuenta con dos reporteadores. Reports y Ad Hoc Reports, los dos pueden ser utilizados dentro de OMNIS7.

Reports cuenta con varias opciones para darle presentación a los datos, en forma horizontal, vertical o en paquete. Por medio de una barra de herramientas se puede diseñar la presentación del reporte. Se pueden poner varios encabezados, pie de página, número de página, fecha, sacar subtotales, totales, así como seleccionar el orden en que se quiera que aparezcan los datos, y realizar la selección de los datos que se necesitan. Los reportes pueden ser vistos antes de ser impresos, desplegarlos en la pantalla o mandarlos a un archivo o al clipboard de Windows.

Ad Hoc Reports es un generador de uso rápido para realizar reportes y query fácilmente, directamente con OMNIS7. Para realizar el reporte por medio de una ventana se seleccionan las tablas y los campos que se van a ocupar de cada una, posteriormente se la da presentación y se ejecuta la aplicación: este reporteador no corre consultas que sean muy complicadas, además de que falla cuando es mucha la información a reportar.

**Interface con el programador.**

*CUA.*

Difiere significativamente del CUA de Windows, lo que lo hace difícil de aprender y usar.

***Modo de programación.***

OMNiS7 provee dos formas para construir la aplicación del cliente: *Perform SQL command* y *SQL scripts*.

*Perform SQL* permite una sola línea para declarar la sesión de SQL.

SQL Scripts se utiliza para realizar declaraciones que sean más de una línea. OMNiS7 cuenta con un *Buffer SQL*, un área de memoria que contiene una sola declaración que se construye con una serie de comandos.

#### **Lenguaje de programación.**

No es similar al de otro producto (quizás un poco a PowerBuilder), tipo de lenguaje 4GL similar a C++, El tipo de código generado es de intérprete, por lo que requiere de un Run-Engine.

#### **Debugger.**

El Debugger de OMNiS7 es muy completo, permitiendo correr la aplicación paso a paso, inspeccionar los procedimientos, evaluar y alterar los campos y las variables.

#### **POE.**

Utiliza eventos propios, que permiten la portabilidad.

#### **Biblioteca o repositorio de componentes.**

Dentro de Omnis todo se realiza mediante una biblioteca de componentes, ya sean Menús, Pantallas, Reportes u otros.

#### **Distribución de aplicaciones.**

Se debe copiar el fuente de la aplicación a la plataforma donde se desee correr, y contar con el intérprete para dicha plataforma.

#### **Programación en grupo.**

No cuenta con un soporte formal para programación en grupo.

**Interconexión.**

*Método de conexión a servidor SQL.*

Hay dos formas de conectarse al servidor, por medio de SQL OBJECT BROWSE y LOGON MANAGER. SQL OBJECT BROWSE se selecciona al abrir la conexión al servidor por medio de iconos y solamente se selecciona el servidor a conectarse y no es necesario darle todos los parámetros, mientras que con LOGON MANAGER aparece una ventana donde se seleccionan los parámetros y cuántos registros, como máximo, se van a traer después de realizar un query.

Por ser multi plataforma, todas sus conexiones son por ODBC. Carece de drivers nativos.

*Facilidad de migración.*

Es capaz no sólo de migrar de servidor, sino también variar sin necesidad de modificaciones del programa la plataforma del cliente, por lo que la hace una de las mejores.

*Desarrollo de aplicaciones con interconexión.*

Soporte para OLE 2.0, a nivel cliente, sin soporte para automatización y DDE (Dynamic Data Exchange).

**Integración con herramientas de terceros.**

**CASEs.**

Cuenta con integración de SilverRun, un CASE capaz de interactuar en ambas vías con Omnis 7. Es uno de los mejores CASES que hay. Su integración es muy estrecha.

*Escalabilidad.*

La capacidad de cambiar la plataforma de hardware de cliente lo hace muy escalable.

***Manejo de gráficos.***

Las gráficas de OMNIS son de gran calidad, y se podrán seleccionar diversas características para realizarlas. Su presentación es en pay, barras y líneas donde se le podrán adicionar diversos atributos como colores, tercera dimensión, rayado en dirección horizontal, vertical, con diferentes rangos, etc.

**1.6.2.3.3 Powerbuilder 5.0 (PowerSoft; Sybase).**

PowerBuilder es quizá el mejor front-end que existe en la actualidad, ya que se presenta como un producto que permite el desarrollo de aplicaciones robustas, bajo ambientes multi plataforma, provee de un optimizador de código, posibilidad de distribución de objetos en un ambiente de red y drives nativos para diferentes bases de datos.

***Plataformas de desarrollo soportadas.***

Procesador 486 o superior, MS-DOS o PC-DOS versión 3.3 o mayor, 12MB RAM, Microsoft Windows 3.x, Windows NT, Windows95; 32MB de espacio en disco duro; unidad de CD-ROM.

PowerBuilder posee un soporte completo para ambientes Windows de 16 y 32 bits en plataformas Intel, incluyendo Microsoft Windows 3.x, Windows NT, Windows95, Win OS/2, Mac y UNIX (Motif y Open Look).

***Ventajas.***

PowerSoft ha sido comprado recientemente por Sybase, por lo que la versión 5.0 provee de herramientas y drivers mejorados para este DBMS. Además, permite el fácil desarrollo y distribución de aplicaciones en varias plataformas a nivel corporativo.

***Desventajas.***

Su ambiente de programación difiere del normal. Asimismo, su futuro es incierto al haber cambiado de dueño.

***Observaciones.***

Posiblemente es la herramienta de high-end más exitosa del mercado, con muchos desarrollos y herramientas de terceros. Definitivamente sería la mejor opción.

***Utillerías extra disponibles con el producto.***

PowerBuilder ofrece una familia de herramientas de desarrollo escalable que incrementan la productividad de las aplicaciones. La serie incluye PowerBuilder Enterprise, PowerBuilder Team/PDBS, PowerBuilder Desktop, Infomaker y PowerBuilder Library for Lotus Notes.

***Servidor de DB SQL local.***

Incluye el Engine más popular en el mercado, Watcom SQL en su versión, solo que soporta DB de hasta 5 Mb.

***3GL.***

Incluye Watcom C++, que permite hacer módulos externos y ligarlos a la aplicación.

***Reporteador.***

PowerBuilder, a través de Infomaker, trae consigo un impresionante arreglo de tipos de reportes: formas libres, tabulares, control break, crosstab, etiquetas, compuesto, y otros, con el acceso más completo a la información y manejo de herramientas para usuarios finales y desarrolladores.

Infomaker habilita la creación de representaciones, reportes de alta calidad y fáciles definiciones de consultas sin necesidad de programar.

Las consultas se realizan a través de un constructor gráfico y un quickselect multitabla. Sólo hay que salvar las consultas como objetos y entonces usarlos como fuentes de datos para una gran variedad de reportes.

#### ***CUA.***

Sin ajustarse mucho a los estándares de Windows, es relativamente fácil de usar.

#### ***Modo de programación.***

PowerBuilder puede definir, compilar y corregir una clase integradas de C++ basadas en el compilador Watcom C/C++ de alto rendimiento, aparte del 4GL nativo que incluye.

#### ***Lenguaje de programación.***

PowerBuilder ofrece un extenso lenguaje orientado a objetos que provee acceso a miles de funciones. Los desarrolladores pueden escribir sus propias funciones o utilizar las ya existentes escritas en C o en otros lenguajes. Han sido incluidos un compilador manejador y un debugger completamente novedoso.

#### ***Tipo.***

Lenguaje 4GL similar a C++.

#### ***Tipo de Código Generado.***

Código P

#### ***POO.***

A través de Watcom C++ incluye toda la gama de programación orientada a objetos, pero es en esta herramienta aparte, y en 3GL.

PowerBuilder soporta la definición de clases para modelados visuales y objetos no visuales. Además, también provee soporte para otras características de la POO, incluyendo herencia.

encapsulamiento de datos y procesos lógicos, y polimorfismo. Estas capacidades aseguran consistencia en aplicaciones, incrementando la productividad y minimizando costos.

Es posible usar ventanas de PowerBuilder, menús y objetos creados por los usuarios para definir objetos ancestro con atributos de encapsulamiento, eventos y funciones. Entonces es posible heredar esos objetos para crear objetos descendentes.

#### ***POE.***

Todo el ambiente de la aplicación se basa en los eventos de Windows.

#### ***Biblioteca o repositorio de componentes.***

PowerBuilder provee una biblioteca de objetos centralizada y administrador de código fuente, además, una aplicación de administración de configuración e interfaces para los más populares programas de administración de versiones de terceros.

Tiene capacidad de Bitácora del uso del Repositorio de componentes, Project Painter para mantenimiento y generación de configuración de aplicaciones.

#### ***Distribución de aplicaciones.***

Genera disco de instalación, mismos que pueden distribuirse libre de regalías. Además, con el producto PowerBuilder Assistant, se cuenta con soporte para lenguajes múltiples en tiempo de corrida.

#### ***Programación en grupo.***

Cuenta con amplias herramientas para desarrollo en grupo.

#### ***Método de conexión a servidor sql.***

Incluye el Watcom SQL ODBC driver que soporta conexiones con bases de datos Watcom SQL creadas con Watcom SQL, PowerBuilder o Infomaker. Este driver es multi-tier, el cual procesa funciones de ODBC pero manda las instrucciones SQL dependiendo de la base

de datos usada para que se procesen. De esta forma el código está separado del lenguaje de transacción, optimando la programación y el aprendizaje de la herramienta

***Facilidad de migración.***

Con Watcom SQL como servidor local, es muy fácil desarrollar en él para después migrar a otro servidor SQL, ya sea de Watcom u otros.

***Control del back-end.***

Es posible la creación de la base de datos desde PowerBuilder.

***Desarrollo de aplicaciones con interconexión.***

Soporte para OLE 2.0, a nivel servidor y cliente, con soporte para automatización, DDE, DLLs, VBXs y COX.

***Aplicaciones de terceros.***

***CASEs.***

Existe una herramienta de ERIWIN diseñada exclusivamente para PowerBuilder.

***3GLs.***

Directamente cuenta con un lenguaje de programación incluido, el Watcom C++.

***Escalabilidad.***

El CODE (Client/Server Open Development Environment) expande la tecnología de los productos de PowerSoft que cubren varios aspectos como son llamadas remotas a procedimientos y procesos de transacciones de modelo de datos y pruebas automatizadas.

***Manejo de gráficos.***

La ingeniería gráfica de PowerBuilder provee de gráficos de dos y tres dimensiones, de pastel, de barras, columnas, líneas, scatter y gráficas de área.

#### **1.6.2.3.4 SQL Windows 5.0 (Grupa).**

##### ***Requerimientos del producto.***

Procesador 486 o superior, con memoria de 12 MB (mínimo) hasta 16 a 32 MB para una óptima ejecución.

En disco 28 MB de espacio en disco duro para standalone y 45 MB de memoria para SQL Windows Corporativo.

Sólo Windows actualmente, pero se espera una versión para Windows 95, ampliamente integrada a ese nuevo sistema operativo.

##### ***Ventajas.***

Permite conectarse con fuentes de datos, manipular datos y generar aplicaciones enteras, todo sin escribir una línea de código. Con los Quick Objects, su librería de controles, es posible el generar aplicaciones robustas muy rápidamente. Quizás es el ambiente con mejor relación poder/facilidad.

Tiene una alianza estratégica con Microsoft, por lo que sus productos pueden llegar a ser los más conocidos del mercado.

##### ***Desventajas.***

Requerimientos de hardware sumamente elevados, que lo hacen poco competitivo. En el futuro se basarán en controles OCX, que han sido vistos como grandes, lentos e inestables.

##### ***Observaciones.***

Buscando desbancar a PowerBuilder, SQL Windows es una herramienta que cada vez gana más popularidad, gracias a sus Quick Objects. Sin embargo, la alianza con Microsoft puede

no ser del todo buena, al obligarles a usar controles OCX. También requiere una fuerte inversión en hardware.

**Utilerías extra disponibles con el producto.**

Team Windows, que es un programa de control de trabajo en grupo, incluyendo control de versiones y repositorio de componentes, herramientas de control y monitores de la red y del DBMS, compilador de SQL Windows a C, mismo que mejora enormemente el desempeño, y Report Windows, un pequeño reporteador para usar dentro de las aplicaciones.

***Servidor de DB SQL local.***

Incluye el Engine de la compañía, SQL Base, en su versión monousuario, que soporta DB de hasta 5 Mb.

***Reporteador.***

Quest es una herramienta de búsquedas y reportes orientada para personal directivo, la cual tiene una interface simple, pero una poderosa capacidad.

***CUA.***

No es intuitiva, pero después de usarla un tiempo es fácil aprender su funcionamiento.

***Modo de programación.***

Principalmente visual, quizás es la herramienta que más permite hacer sin escribir código. En caso de necesitar hacerlo, es en forma jerárquica.

***Lenguaje de programación.***

Trata de ser un ambiente de desarrollo completamente visual, permitiendo incluso crear nuevos componentes heredados de otros. También tiene un poderoso 4GL.

***Tipo.***

Lenguaje 4GL similar a C++.

***Tipo de código generado.***

Código P, requiere de un Run-Engine.

También es posible "compilarlo" a C++, con una herramienta extra.

***POO.***

Basándose principalmente en programación visual, tiene la capacidad de POO. Es posible construir componentes nuevos.

***POE.***

Soporta los de Windows, pero es posible, con un 3GL, construir Quick Objects que respondan a otros eventos.

***Biblioteca o repositorio de componentes.***

Tiene la capacidad de programación en grupo, siendo uno de los más capaces para esta labor.

***Distribución de aplicaciones.***

Cuenta con un generador de discos de instalación.

***Programación en grupo.***

Cuenta con amplias herramientas para desarrollo en grupo. Estas se incluyen en las versiones corporativa y empresarial de sus herramientas.

***SQL Windows capacidades de interconexión a servidor SQL.***

Soporta, a través de repositorios especiales, drivers nativos para Sybase, Oracle y otras. Además puede conectarse por ODBC.

***Facilidad de migración.***

Muy simple, al poder desarrollar en un servidor local y luego migrar a uno remoto.

***Control del back-end.***

Es posible la creación de la base de datos desde PowerBuilder.

***Desarrollo de aplicaciones con interconexión.***

Quizás la más alta integración con aplicaciones OLE y DDE.

***Enlace de productos de terceros.***

Puede conectarse a muchas aplicaciones, con repositorios especiales, como la de Lotus Notes.

***Aplicaciones de terceros.***

***CASEs.***

Existe una herramienta de ERWIN diseñada exclusivamente para SQL Windows y SQL Base.

Además con PVCS, que es una herramienta muy popular de control de versiones.

**1.6.2.3.5 Visual Basic 4.0 (Microsoft).**

***Plataformas de desarrollo soportadas.***

Windows 95 o Windows NT (para la versión de 32-bits), 486 CPU, 8MB de RAM (16MB de RAM para Windows NT), 20MB a 80MB de espacio en disco, unidad de CD-ROM.

Sólo puede implantarse en plataformas PC, cuyos requerimientos mínimos son procesador 386SX con 4MB de Memoria, y Windows 3.x (Requiere instalar archivos del sistema).

El run-time de Visual Basic 3 consiste en un conjunto de DLL'S, los cuales van de 350Kb a 2Mb, dependiendo de lo que requiera el programa.

***Ventajas.***

Visual Basic es el producto que permitió la programación para la plataforma Windows a miles de programadores, fue el primero en su tipo y por ello es el producto más popular en el mercado. Posee una cantidad increíble de add-ons y librerías, ya sea de Microsoft o de terceros. El ser un producto Microsoft asegura una perfecta integración con Windows. El dialecto de Basic que utiliza es simple de aprender y de usar. Un tamaño y requerimientos reducidos ayudan a programar rápidamente sin requerimientos de hardware altos. La mayor parte de los demás productos ofrecen, aunque sólo parcialmente, un camino de migración de Visual Basic a ellos. Los controles VBX, que se empezaron a usar con Visual Basic, son ahora el mayor estándar del mercado.

***Desventajas.***

Este producto no está diseñado para desarrollo de aplicaciones cliente/servidor, aunque éste fue el uso que se le dio desde un principio. Realmente está orientado para aprender programación en Windows, o prototipos de aplicaciones más grandes, donde la versión real del programa será realizada en un lenguaje de programación más formal (C o Pascal). Es por ello que sólo tiene facilidades básicas de debug y distribución de aplicaciones, y carece por completo de facilidades de control de versiones, programación en grupo y otras características avanzadas de desarrollo. Actualmente, Visual Basic tiene un desempeño bajo y características inferiores a cualquier producto posterior nativo de PC. Esto incluye los controles VBX o el uso de 1001 DLL's diferentes como run-time. Basic, aunque fácil de aprender, no es un lenguaje óptimo para un desarrollo profesional. Como cualquier otro producto Microsoft, presenta problemas o "cualidades" las cuales son difíciles de ver y controlar. Depende demasiado de add-ons para desarrollos avanzados, cada uno de los cuales aumenta la posibilidad de que el producto se comporte de forma inesperada.

***Observaciones.***

La utilidad actual de Visual Basic es como base de comparación para productos posteriores, ya que es el producto más conocido externa e internamente. La versión 4.0 incluye cambios fuertes al cambiar de controles VBX a controles OCX (usados por primera vez en Access), y va a traer cambios en la sintaxis del Basic.

#### **Utilerías extra disponibles con el producto.**

Incluye el "engine" de Access 1.1 (pero lleno de trucos y trampas) junto con dos aplicaciones de ejemplo que permiten una administración básica de las mismas, ODBC (también con un pequeño "bug" incluido), un ejemplo de una aplicación de distribución, soporte para Pen for Windows, Mapi (limitado) y Telecomunicaciones (aún más limitado). También incluye una versión "crippleware" (limitada) de Crystal Reports for Visual Basic (no opera bien con servidores de bases de datos remotos). Además incluye información (pobre) y ejemplos sobre cómo crear controles VBX con Microsoft Quick C.

#### ***Servidor de BD SQL local.***

Las bases de datos de Access responden a algunos comandos de SQL (los de consulta), pero no proveen el desempeño o portabilidad de éstas.

#### ***Reporteador.***

El Crystal Reports para Visual Basic es un reporteador simple, pero popular, el cual permite el desarrollo de reportes básicos para bases de datos Access, Paradox, Xbase y Btrieve. Permite el acceso de otros tipos de manejadores o de bases de datos remotas por medio de ODBC, pero su rendimiento en este último caso es muy deficiente.

#### **Interface con el programador.**

#### ***CUA.***

Se ajusta a la interfaz de Windows, con controles 3-D como agregados en la versión profesional, siendo la base de las interfaces de la mayoría de los productos actuales. Sin

embargo, al ampliar el número de controles o de formas, la facilidad de uso sufre una degradación importante.

***Modo de programación.***

Visual Basic es el primer lenguaje de "programación visual", en el cual se le da una forma visible a cada una de las partes de la aplicación, y en donde el código está limitado a estos "componentes". El resultado es que la distribución y apariencia de los componentes toman precedencia a la creación de un código orientado a eventos delimitado a ellos.

***Tipo de lenguaje de programación.***

Es un dialecto de Basic, supuestamente compatible con Qbasic, de una forma limitada.

***Tipo de código generado.***

Código P. Es un código compactado y verificado por sintaxis, el cual es interpretado a tiempo de corrida, restándole eficiencia, velocidad y detección de errores, pero facilitando el desarrollo.

***POO.***

Mínimo, se utiliza la sintaxis de objeto componente, pero no el uso real de métodos, herencia, encapsulamiento o polimorfismo.

***POE.***

Limitada, se está sujeto a los eventos y parámetros predefinidos en Visual Basic, y carece de respuesta a eventos dentro de métodos.

***Biblioteca o repositorio de componentes.***

Incluye una librería limitada de controles VBX, otra más útil en la versión profesional. La creación de controles VBX puede ser difícil por la falta de información, y debe realizarse en un lenguaje serio de programación.

***Distribución de aplicaciones.***

Presenta dificultades, por el gran número de DLL's que deben ser incluidos, además de falta en control de versiones.

***Método de conexión a servidor SQL.***

La edición profesional incluye ODBC versión uno con drivers para Microsoft SQL Server y Oracle 6. Para poder usar el primero con Sybase SQL Server, es necesario realizar modificaciones en el servidor. También existe un agregado que permite conexión nativa a Microsoft SQL Server (y a Sybase con las mismas condiciones que con ODBC).

***Facilidad de migración.***

Fácil, pero es necesario modificar el código y recompilar. Así mismo la sintaxis del SQL de Access presenta diferencias con el SQL estándar.

***Desarrollo de aplicaciones con interconexión.***

Soporte para OLE 2.0, a nivel servidor y cliente, con soporte para automatización, DDE (Dynamic Data Exchange), DLL's (Dynamic Link Libraries), y VBX's (Visual Basic Controls). Se vende por separado un conjunto de herramientas y documentación para programar aplicaciones de integración de Microsoft Office (Word, Excel, Access, Mail, PowerPoint y Proyecto).

***Programación en grupo.***

No cuenta con soporte para programación en grupo.

***Aplicaciones de terceros.***

Visual Basic tiene la mayor cantidad de herramientas de terceros disponibles actualmente, la mayoría de ellas son add-ons para tratar de paliar las limitaciones inherentes de Visual Basic.

**CASEs.**

Por la falta de soporte a programación avanzada o administración de bases de datos del mismo Visual Basic, no hay gran número de Cases que soporten o aprovechen todo su potencial con él.

**Escalabilidad.**

No existe.

**Manejo de gráficos.**

Se realizan por medio de un control VBX incluido, y por gran número de herramientas de terceros. Tienen calidad, pero es muy limitada la importación, exportación y manejo.

### **1.6.2.4 Análisis de las comparaciones.**

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

Tras la evaluación, las conclusiones sobre cada herramienta son:

**Delphi.**

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 de Borland en herramientas de desarrollo se observa en todo su esplendor con este producto.

**Power Builder.**

Esta herramienta cruza un momento difícil de su historia, pero 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:**

Pese a que promete ser un ambiente de desarrollo completo, al menos lo que pudimos 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.

#### **SQL Windows.**

Esta herramienta promete ser el futuro para el desarrollo de aplicaciones cliente/servidor, gracias a una facilidad de desarrollo inigualable y alianzas estratégicas. De todas formas, es necesario mantenerse atento a la próxima versión, que con su promesa de apoyo a Windows 95, puede implicar un cambio importante respecto a versiones anteriores. También hay que considerar los altos requerimientos en memoria.

#### **Visual Basic.**

Esta herramienta inició 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 como se menciono anteriormente esta supeditada a los recursos existentes dentro de la Facultad de Derecho, tanto de hardware, software y personal de desarrollo. Los candidatos mas viables son Delphi y PowerBuilder, y dentro de estos últimos se encuentra PowerBuilder, ya que al ser adquirido por Sybase presenta una mejor integración con este producto.

# 1.7

## NORMATIVIDAD

En cualquier organización las reglas son fundamentales ya que define a la misma, por lo que a continuación desglosamos este punto en software y hardware.

### *1.7.1 Software.*

#### **1.7.1.1 Bases de datos.**

El primer sistema que se desarrolló en la facultad de Derecho fue el de reinscripciones de alumnos en ese entonces se evaluaron diversas bases de datos y se declinó por PARADOX 3.5.

Hoy en día en la Universidad Nacional Autónoma de México ha tomado como norma el manejador de bases de datos SYBASE, podemos mencionar algunas dependencias como DGAE (Dirección General de Asuntos Escolares), DCAA (Dirección de Cómputo para la

Administración Académica), DGP (Dirección General de Personal), etc., por lo cual se desea que la Facultad de Derecho se incluya en esta norma..

### **1.7.1.2 Herramientas de desarrollo.**

PARADOX además de su base de datos, contiene un lenguaje de desarrollo estructurado (PAL) con el cual se han creado las aplicaciones (Scripts) entre las cuales podemos mencionar:

- Sistema de Inscripciones.
- Sistema de Control de Asistencia Docente.
- Sistema de Exámenes Profesionales.
- Sistema de Seminarios.
- Sistema de Control del Alumnado (Directorio).
- Sistema de Exámenes de Oposición.
- Directorio de Profesores.
- Sistema de Control de Acceso a Estacionamiento.
- Sistema de Inventario de Equipo de Cómputo.
- Sistema Automático de Insaculación.
- Sistema de Control de Usuarios del Centro de Cómputo.
- Sistema de Control de Servicios del Centro de Cómputo.

La versión de la herramienta no se ha actualizado hasta nuestros días.

### **1.7.1.3 Programas.**

El desarrollo de los programas es competencia exclusiva del personal propio de esta facultad, debido a los altos costos externos de manufactura.

Todo el software desarrollado por personal de esta facultad pasa a ser propiedad de nuestra casa de estudios.

Todo programa debe estar debidamente documentado. Nombres de variables, procedimientos, etc., deben ser mnemónicos. También deben contener un encabezado que indique que se lleva a cabo, quien lo desarrollo, fecha de creación. En caso de haber sufrido modificaciones mencionar quien las realizó y en que fecha.

Cada vez que se realiza una actualización se hace un respaldo del programa original como prevención a cualquier eventualidad.

Los programas no deben ser mayores a 200 líneas, es decir muy extensos. Debe ser modular. Las variables globales deben ser mínimas.

Los módulos deben caracterizarse por ser reutilizables.

Mientras que el código sea óptimo mejor será el programa.

Para liberar un sistema debe ser sometido a pruebas por el propio desarrollador, por terceras personas y finalmente por el usuario. En caso de haber sufrido fallas será perfeccionado y nuevamente pasara por las pruebas antes mencionadas.

La mayoría de las funciones deben ser introducidas a una biblioteca para la cual hay un control.

### **1.7.1.4 Reportes.**

La letra del contenido de cualquier reporte debe ser mínimo de 12 puntos La impresión debe ser en hoja blanca tamaño carta. Debe contener siempre un encabezado con las siguientes características:

- Escudos, tanto de la Universidad como el de la Facultad.
- Leyendas: UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE DERECHO.

- Título del reporte.
- Fecha y hora de emisión.

Además deben tener:

- Número de página.
- Posibilidad de salida tanto a pantalla como a impresora.
- Un documento que contenga lo siguiente:

Quien lo creó y fecha de creación.

Para que área(s) fue creado.

Modificaciones que ha sufrido con su respectiva fecha.

### **1.7.1.5 Sistemas operativos.**

El sistema operativo como norma en esta institución es el MS DOS de Microsoft para usuarios finales debido a la gran gama de aplicaciones de terceros que hay para este y su amplia difusión internacional.

En cambio para servidores se ha normado UNIX como plataforma, debido a su gran robustez y además a su cualidad de ser abierto.

### **1.7.1.6 Interfaces de usuario gráficas (GUI).**

Los GUIs han tomado un gran auge en estos tiempos por ser amigables e intuitivos por lo cual la facultad no ha sido ajena a estas herramientas y se ha elegido Windows de Microsoft como plataforma para aplicaciones de usuario finales por su gran uso y difusión a diferencia OS/2, Next Step, DESQview o el de Machintosh.

Para la administración de servidores los cuales trabajan bajo sistema operativo UNIX (Solaris) el mismo proveedor del equipo incluye una interface de usuario gráfica llamada OPEN LOOK.

## **1.7.2. Hardware.**

### **1.7.2.1 Equipos de usuario final.**

Los equipos que se han estandarizado para usuarios finales son computadoras personales compatibles con las siguientes características:

- Procesador 486 o superior.
- Velocidad 66Mhz o superior.
- Memoria 4 Mb o más.
- Capacidad de almacenamiento de 500 Mb o mayor.
- Monitor VGA.

- Unidad de diskettes de 3 ½ .
- Mouse.
- Impresora de matriz o láser.

### **1.7.2.2 Servidores.**

Los equipos que funcionan como servidores deben ser WorkStation y contar con las siguientes características:

- Unidad de CDRom.
- Unidad de Cinta de 8 o 4 mm..
- Capacidad de almacenamiento del orden de Gigabytes.
- Unidad de diskettes de 3 ½ .

### **1.7.2.3 Redes.**

- Tarjetas: Ethernet de 16 bits o mayor, para cable coaxial y/o par trenzado.
- Cableado: Cable coaxial delgado o par trenzado UTP nivel 5.
- Topología: Estrella y/o bus.
- Interconexión mediante IPX y TCP/IP.

### **1.7.3 Generales.**

### **1.7.3.1 Documentación.**

Documentación de programas. Se cuenta con información que describe en forma global el funcionamiento de cada modulo y en forma integral al o los sistemas.

Algunos de los puntos que contiene dicha documentación es:

- Nombre del sistema y/o modulo en particular.
- Objetivo o función.
- Código fuente.
- Notas importantes a cerca de su uso o manejo.
- Inclusión de un manual del sistema para los desarrolladores o bien para su mantenimiento posterior.

### **1.7.3.2 Manuales de usuario.**

Se debe de tener un manual sobre el uso y capacidad de los sistemas en un lenguaje sencillo, apoyándose con ilustraciones del propio sistema que definan claramente la forma de trabajar. Así mismo debe de contener un apartado con soluciones a los problemas más frecuentes sobre el uso del sistema. Además, se proporcionará el nombre y ubicación de la(s) persona(s) que pueden brindar apoyo ante alguna eventualidad mayor.

### **1.7.3.3 Entradas/salidas de información por área.**

- Sría. General.

- Sría. Exámenes Profesionales.
- Sría. Administrativa (Personal Docente).
- Sría. Académica.
- Sría. Asuntos Escolares (Control de Actas).
- Dpto. de Mesa de Firmas.

#### **1.7.3.4 Respaldos periódicos.**

Se realizarán respaldos de los datos generados durante una jornada de trabajo; los cuales pueden ser semanales o mensuales, e inclusive diarios dependiendo de la importancia de los mismos.

#### **1.7.3.5 Seguridad.**

En este rublo se contempla la creación de grupos de trabajo, de asignación de privilegios y recursos, control de acceso, cambio periódico de contraseñas, respaldos, etc.

#### **1.7.3.6 Reportes de fallas.**

Equipos. Las fallas presentadas en el equipo deberán ser reportadas a la brevedad posible y se acompañara de un reporte de fallas que le será entregado por parte del centro de computo. El equipo será revisado en el lugar de la falla y en caso de ser necesario será retirado del lugar y si es posible se proporcionará un equipo similar durante el período de reparación del equipo reportado.

Sistemas. En el caso de presentarse fallas durante el uso de los sistemas y una vez leído el manual del usuario y consultar el apartado referente "*Solución a los problemas mas frecuentes*", se solicitará el soporte necesario para poder reparar la falla. Si se trata de un error en la programación esta deberá ser solucionada a la mayor brevedad posible y así mismo realizar los cambios necesario en los módulos que se verán afectados y en los manuales del usuario.

### **1.7.3.7 Bitácora por equipos.**

Se deberá de llevar la historia de operación de los equipos, para programar los periodos de mantenimiento preventivo y/o determinar su tiempo de vida útil.

De esta misma forma el expediente de cada equipo deberá de ser único, para evitar pérdida de información, garantías, reparaciones, configuración, software instalado, equipo auxiliar, ubicación, números de inventario y cualquier otra característica que ayude a su mejor control.

**CAPITULO**

**2**

**PLANTEAMIENTO  
DEL PROBLEMA Y  
PROPUESTA DE  
SOLUCION**

## 2.1

### PROBLEMÁTICA ACTUAL

La Facultad de Derecho requiere controlar alrededor de 10000 alumnos, más de 700 profesores de licenciatura, 40 de posgrado y cerca de 60 del sistema abierto. Un plan de estudios de 38 materias para el plan anterior y 76 materias para el nuevo plan. Y cuenta con varias secretarías que llevan a cabo el control de estos recursos.

El sistema de información docente tema de la presente tesis es parte del proyecto que en su conjunto se desarrolla en la Facultad de Derecho, cuyo objetivo es poder resolver los problemas que actualmente se vienen presentando en las secretarías general, académica y administrativa. Entre los cuales podemos mencionar los siguientes:

- Reducir el tiempo de elaboración de los horarios semestrales y administración general de los profesores.
- Mantener la integridad de la información.
- Obtener el control de la cantidad de información que actualmente se maneja.

Por lo tanto se requiere de un sistema capaz y eficiente para poder llevar a cabo las tareas del proceso de administración de los profesores.

En la actualidad la Facultad de Derecho cuenta con algunos sistemas aislados que le ayudan a llevar adelante sus actividades esenciales, sin embargo, dichos sistemas o bien trabajan en stand-alone o bien en una red local aislada, lo cual limita la información a un grupo reducido.

Para poder compartir información que generan algunas áreas se deben de realizar actualizaciones constantes de la información de máquina a máquina, lo cual resulta ser una tarea tediosa y sobre todo actualmente no práctica.

Los sistemas en su mayoría se desarrollaron en Paradox que no es propiamente un servidor de base de datos, propiciando una sobrecarga en la red, no contempla mecanismos de respaldo y recuperación, de conectividad, de registro de transacciones, de diagnóstico, su confiabilidad y rendimiento se ven decrementados mientras se incrementa su uso y la cantidad de datos a manejar.

Con el crecimiento de la facultad, las actividades se han multiplicado los servicios se han tenido que mejorar y lo que en algún tiempo fue suficiente, actualmente ya no lo es. Se ha invertido en equipo de cómputo, desde computadoras personales hasta servidores Unix, además se cuenta desde hace tiempo con el manejador de base de datos Sybase System 10 con capacidad para 32 conexiones, se ha generalizado el uso del ambiente gráfico Microsoft Windows 3.x, y se ha requerido acceder desde cualquier parte de la facultad o inclusive fuera de ella a la información que se maneja, principalmente de tipo escolar y académica.

Dado que se contaba con una infraestructura primaria basada en una red local con un servidor Netware 3.x y con una base de datos relacional Paradox 3.5, bajo un sistema operativo MS-DOS, en la actualidad este tipo de configuración resulta insuficiente por sus características.

Los programas de los sistemas con que se controlan varias de las actividades de esta institución, se almacenan en scripts, los cuales son interpretados por el run-time de Paradox.

Como se trata de un interprete en tiempo de ejecución, el desempeño de la aplicación resulta ser muy bajo dadas las condiciones de respuesta requeridas actualmente.

Pero no solo es la interpretación de código, aunado a esto es necesario contar con mecanismos de compartición de datos, es decir, elementos que nos permitan administrar los recursos en un ambiente multiusuario, tanto de usuario como de integridad de datos. Paradox solo provee de recursos de compartición de datos básicos, los cuales resultan difíciles de controlar, por lo que las aplicaciones se vuelven complejas y extremadamente lentas en desarrollarse y si se requiere de un cambio en la estructura de la tabla, ésta deberá esperar hasta que se analicen todos los recursos asociados y conocer como afectará el cambio al sistema en general.

Por otra parte los sistemas actuales carecen de documentación por parte del equipo de desarrollo que las implemento en un principio; se carece de formalidad en la manera de programar; no se llevo a cabo una actualización de la versión de Paradox; no se diseñaron bajo un ambiente de red y solo se agregaron a un servidor de archivos, por lo que si dos o mas usuarios comparten el mismo programa o la misma tabla al mismo tiempo alguno de ellos tomara el control de la situación y los demás se verán obligados a renunciar a dicho recurso. Este ultimo problema no ha sido muy grave ya que hasta el momento los usuarios han tenido la precaución de saber que otro usuario esta ocupando o no el recurso que se desea utilizar, o bien se ha optado por solicitarle a la persona que se hace cargo de dicho recurso la información que se requiera, generándose tiempos de respuesta grandes.

Dado que la información se encuentra semi-distribuida y en sistemas semi-monousuario (las aplicaciones se hayan en una red, pero solo las pueden acceder un usuario a la vez), esto ha propiciado que las actividades se vean obstaculizadas por el tiempo disponible de las personas, creándose un esquema de atraso en la obtención y generación de información, repercutiendo grandemente en cada una de las actividades de la institución y generando un alto grado de ineficacia operativa.

Por otra parte el diseño de un sistema integral no fue planeado.

Los problemas antes planteados pueden resumirse de la siguiente manera:

- **Redundancia innecesaria.** En el sistema tradicional que se maneja actualmente los datos son requeridos por diferentes aplicaciones y se registran en varios archivos en forma repetida. La redundancia conduce a problemas que tiene que ver con la integridad de la información. Cuando existe redundancia, se requieren múltiples procesos de entrada, actualizaciones e informes. Además de la redundancia de datos es común que exista redundancia de procesos.
- **Falta de integridad.** Es una de las consecuencias de la redundancia. Cuando se realizan cambios en algún campo de un archivo, se tiene que actualizar este dato en todos los archivos que contengan el mismo campo.
- **Datos compartidos de forma limitada.** No se puede compartir información entre diferentes archivos de datos, ya que están instalados como unidades separadas.
- **Disponibilidad de datos limitada.** Los datos no están disponibles a la persona autorizada en cualquier momento.
- **Inconsistencia de datos.** Existe un desorden en el universo de datos, hay varias versiones de datos.
- **Inconsistencia de representaciones y códigos.** En alguna tabla un campo clave es de diferente tipo o de diferente longitud de los otros campos en otras tablas.
- **Falta de seguridad.**
- **Documentación deficiente o inexistente.**

### ***2.1.1 Áreas de la Institución.***

Es importante mencionar las áreas de las cuales esta compuesta la facultad de derecho, así como de las funciones que desempeña.

- **Secretaría General:** coordina actividades como calendario escolar, asignación de profesores, proceso de titulación,
- **Secretaría Académica:** controla las actividades propias del personal académico de la Facultad.
- **Secretaría de Asuntos Escolares:** controla las operaciones propias de los alumnos, en lo referente al control escolar, emisión de cualquier documentación durante y después de inscrito un alumno.
- **Secretaría Administrativa:** lleva el control de las actividades propias de la administración de personal adjudicado a esta institución tales como: contrataciones de personal académico, de confianza, o sindicalizado; nómina, contabilidad, etc.
- **División de Universidad Abierta:** lleva el control escolar en lo referente al sistema de universidad abierta de la facultad.
- **Centro de Cómputo:** Lleva el control y administración de los recursos de cómputo con que cuenta la Facultad así como de promover su uso y mantenimiento.
- **Biblioteca:** controla todos los aspectos referentes al material de tipo bibliográfico asignado a la Facultad.

Cada día las diferentes áreas de esta facultad exigen más y mejor información de otras áreas con la mayor rapidez posible mediante un acceso cada vez más sencillo para la explotación de la misma, haciendo más eficientes las actividades propias de esta facultad.

Debido a tales exigencias el centro de cómputo en un primer esfuerzo buscará la integración de toda la infraestructura de cómputo disponible, es decir, conectar a una red todos aquellos equipos que generen información de interés a otras áreas, e interconectar las redes que existan.

No existe hasta este momento un plan modular de trabajo, en el desarrollo de las aplicaciones por lo que será necesario crearlo, así mismo no existe un proyecto de actualización y/o modificación de los sistemas actuales.

Debido a que se carece de documentación de los sistemas actuales, será necesario realizar todo el proceso de análisis y diseño de un sistema integral que se ajuste a las necesidades actuales y establezca un proyecto de desarrollo a futuro.

En forma paralela se deberá organizar y centralizar la información buscando la unificación de la misma para todas la áreas.

Analizando los recursos de cómputo disponibles logramos observar que la facultad cuenta con más de 400 computadoras personales (en su mayoría con procesador 486 y pentium) distribuidas en todas las áreas, 2 servidores UNIX, 8 redes locales con capacidad para conectar 352 nodos.

Se cuenta con redes, servidores de UNIX que pueden alojar cualquiera de los 3 mejores manejadores de datos (Oracle, Sybase e Informix) e inclusive se cuenta con una licencia de uno de ellos, PCs con capacidad para trabajar con cualquier software front-end en un ambiente gráfico ameno y sencillo para el usuario final.

Para poder lograr el objetivo es necesario contar con las herramientas adecuadas. Como se puede apreciar en los párrafos anteriores se cuenta con toda la infraestructura necesaria y los recursos suficientes para poder lograr la meta fijada, sin embargo hasta este momento no se ha realizado.

Los paradigmas y herramientas nos ayudan a implementar las propiedades de una base de datos para la automatización de actividades. Sin embargo, se requieren conocimientos e ingenio para alcanzar un grado de funcionalidad y eficiencia adecuado en cualquier sistema.

En cuanto a las herramientas, se requiere de alguna que nos permitan trabajar con los datos, que contenga rutinas para poder definir, crear, acceder, respaldar, recuperar y administrar la base de datos garantizando seguridad, integridad y protección de los datos, así como el

sincronizar el acceso de múltiples aplicaciones. Esto también nos impone cierta disciplina, tanto para modelar la base de datos como para poder accederla. Por seguridad debe de entenderse que sólo podrán acceder la base de datos los usuarios autorizados y cada uno verá solo la parte que le corresponde; que los valores almacenados son correctos y que la base de datos es consistente (integridad); que en caso de corrupción de la base de datos es posible recuperar los datos correctos (protección); que varias aplicaciones accedan a la base de datos de manera concurrente.

Cabe resaltar que aunque se cuente con las herramientas necesarias para implementar las propiedades anteriormente mencionadas a un sistema de bases de datos, es necesario saber aplicarlas, siendo entonces los administradores de la base de datos y de desarrollar las aplicaciones los que tienen que hacer buen uso de estas rutinas para obtener los beneficios propios de un DBMS.

## 2.2

# ORGANIGRAMA DE LA SITUACION ACTUAL

La popularidad de las redes es indiscutible y en parte se debe a su buena habilidad para interconectar casi cualquier tipo de equipo, localmente o a grandes distancias. La primera generación de conectividad fue el hecho de crear las redes locales. Ahora, tal vez la segunda generación de conectividad se refiere a la habilidad de poder conectar redes con redes.

El departamento de Redes y Soporte Técnico y el de Desarrollo y Mantenimiento de Sistemas del Centro de Cómputo de la Facultad de Derecho dirige sus esfuerzos para emigrar sus sistemas a una arquitectura cliente/servidor.

Para lograr esta meta se ha ampliado notablemente la infraestructura de red y se ha estado centralizando la información, para lograr que ésta sea única y que cada área se responsabilice por la actualización de su información. En este caso, el banco de profesores será el mismo para todos los sistemas que requieran información de los académicos evitándose con esto la redundancia de los datos.

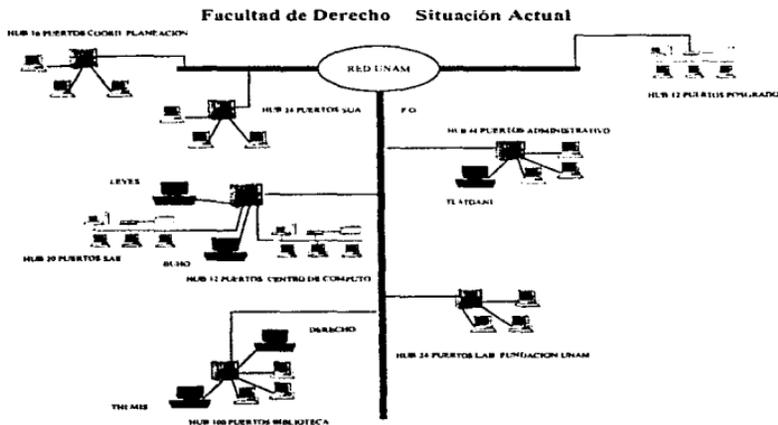


Ilustración 40 Situación actual de la red.

En la **Ilustración 40** se muestra el esquema de la situación actual de la red de la Facultad. Actualmente la Facultad cuenta con 2 servidores UNIX y 8 redes locales interconectadas, con capacidad para conectar 352 computadoras personales a la red. Se tiene un segmento principal (backbone) de fibra óptica del cual se desprenden varias conexiones a los hubs. A cada uno de estos hubs se conectan los servidores y las computadoras personales. Para las áreas de Planeación, Posgrado, Administrativa, Sistema de Universidad Abierta, Biblioteca y Laboratorios de Fundación UNAM la topología empleada es en estrella (se tiene un controlador central al cual todos los nodos se conectan), mientras que para el Sistema de Administración Académica y Centro de Cómputo se tiene una combinación de topologías

en estrella y bus (se trata de un canal de comunicación común al cual todos los dispositivos se conectan).

El problema para este tipo de conexiones es el congestionamiento ya que al acceder desde cualquier computadora (cliente) a un determinado servidor se tiene que llevar a cabo un recorrido tal que en un determinado momento se puede llegar a saturar la línea. Como consecuencia se tiene una comunicación poco eficiente y bastante lenta en horas pico.

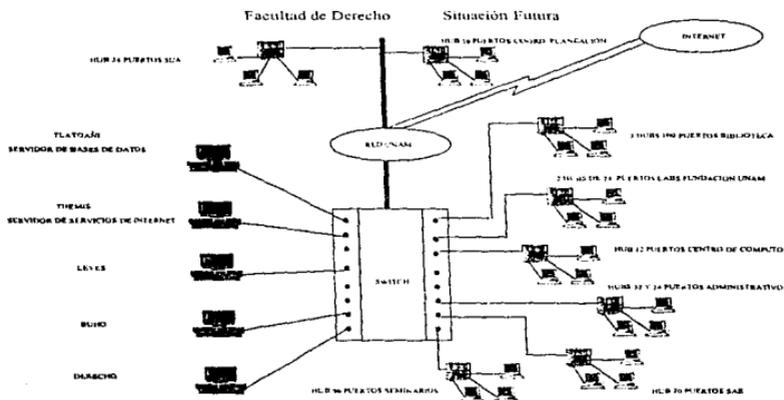


Ilustración 41 Propuesta nueva de diseño de la red.

En la Ilustración 41 se muestra el esquema de la situación propuesta de la red de la Facultad con un enfoque distribuido. En este caso se contará con servidores UNIX y con 8 redes locales con capacidad para conectar 372 computadoras personales a la red. Se

mantendrá el backbone principal y se introducirá un switch general que permita distribuir la carga. Cualquier petición que salga de la computadora se va al Hub y de ahí al switch, una vez aquí se podrá acceder a cualquier servidor.

En este esquema las conexiones son en estrella de tal manera que el acceso es mucho más rápido y eficiente y como consecuencia el balanceo de carga se reducirá. En los servidores se pueden utilizar cualquiera de los siguientes manejadores de bases de datos : Oracle, Sybase e Informix. Las computadoras personales tienen capacidad para trabajar con cualquier software para ambientes gráficos. En este caso se utilizará Sybase para el servidor y PowerBuilder para las computadoras personales.

Los servidores Tlatoani (132.248.84.2) y Themis (132.248.84.1) utilizan el sistema operativo SOLARIS 2.5, Buho y Leyes utilizan NETWARE 3.11, mientras que Derecho utiliza NETWARE 3.12.

Tlatoani será el servidor de base de datos, Themis brinda servicios de Internet (correo, web, FTP, telnet, gopher, wise, news, etc.).

En la **Ilustración 42** se muestra el esquema general de los "sistemas" actuales relacionados con el sector académico de la Facultad de Derecho: Control de Acceso, Contratación y Directorio de Profesores, Seminarios, Control de Asistencia Docente, Titulación, Exámenes de Oposición e Incentivos.

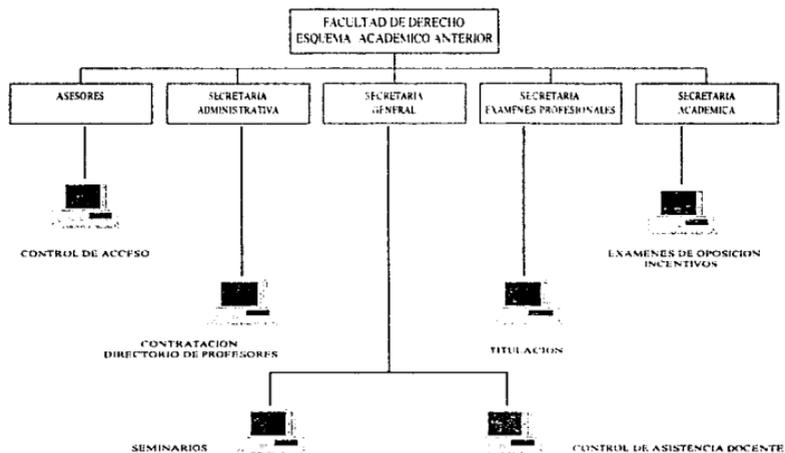


Ilustración 42 Esquema actual de sistemas.

La **Ilustración 43** muestra el diagrama del Sistema Integral del Personal Docente a implantar: Inasistencias del Personal Docente, Registro de Tesis y Examen Profesional, Exámenes de Oposición, Rectificación de Calificaciones.

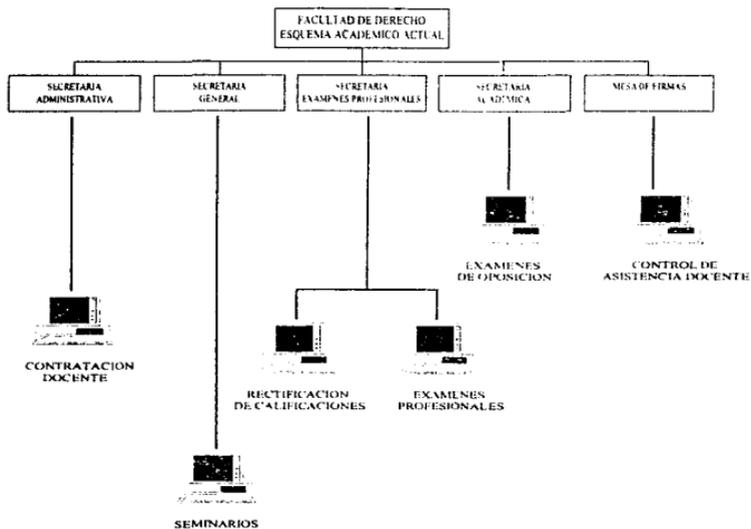


Ilustración 43 Propuesta de solución en sistemas.

## 2.3

# REQUERIMIENTOS DEL USUARIO

En la problemática actual se trata de dar a conocer la necesidad de contar con una modificación de manera radical al sistema integral de información de la Facultad.

Para ello es importante llevar a cabo un proceso de entrevistas con los usuarios y conocer sus requerimientos.

En nuestro caso es necesario tomar en cuenta las características propias de cada actividad, las relaciones que tienen con las demás áreas y el flujo de información.

El proyecto que se pretende desarrollar toma en cuenta la entidad profesor, por lo que se requiere analizar los aspectos y atributos que giran en torno a dicho objeto.

La entidad profesor es una de las más importantes dentro de un proceso de control escolar, de la Facultad de Derecho.

El objeto profesor se controla desde su creación, es decir, desde su registro y permanece dentro del sistema como parte de un proceso histórico.

Son tres las áreas que se ven involucradas o bien hacen uso principalmente del objeto profesor: la secretaría administrativa, la secretaría académica y la secretaría general.

Las secretarías anteriormente mencionadas necesitan conocer de manera clara y oportuna datos a cerca de cada profesor que ha pasado por la Facultad de Derecho, tal como nombres completos, domicilio o manera de localizar a un profesor, su curriculum académico, su asistencia, su porcentaje de calificación, es decir todo un expediente para poder tomar decisiones como son la contratación periódica del profesor, evaluar su capacidad de enseñanza, proponerlos como candidatos para ocupar las plazas académicas, promoverlos para mejores puestos, etc.

En primer lugar la secretaría administrativa recibe la contratación del profesor, por lo tanto dentro de esta área se realizan los procesos de altas, bajas y cambios de profesores.

El proceso es como sigue: se reciben los datos generales del profesor y se procede a su captura. Los datos son controlados a través del RFC de cada profesor.

En este mismo punto se incluyen datos como fecha de contratación, nombre desglosado, fecha de nacimiento, grado académico, categoría, status, domicilio particular, domicilio del trabajo y número de expediente.

Adicionalmente se requieren datos de carácter académico, lo que se encargará de hacer la secretaría académica y se tomarán todos estos datos para formar las plantillas de profesores que impartirán clases durante el semestre.

Con estos antecedentes se entrevisto a los usuarios para conocer sus preferencias y necesidades, encontrándose lo siguiente:

En la secretaría administrativa se cuenta con una persona que lleva el control de los datos generales de cada profesor, entre sus necesidades principales son el contar con un banco de

datos único de profesores que laboran o que laboraron dentro de la Facultad de Derecho, con información completa como la mencionada anteriormente, contar con acceso a los datos que generan otras áreas y que se relacionen con los profesores, establecer un mecanismo de reportes de diversa índole, contar con servicios de búsqueda de información y la posibilidad de poder agrupar la información para una mejor consulta de la misma.

Adicionalmente se solicita una interfaz amigable, sencilla, rápida para poder llevar a cabo todo tipo de controles y accesos.

Así mismo se requiere que exista una mayor coordinación con esta secretaría, debido a que comúnmente no se tiene acceso a la información desde otras dependencias, cayendo en una centralización de datos y actividades.

Se requiere de una mayor interrelación de datos y un mejor conocimiento de los existentes para lograr una mayor coordinación entre los departamentos, evitar trabas en el flujo de datos y agilizar los trámites a nivel administrativo.

La Secretaría Académica entre sus objetivos están el poder llevar a cabo un mejor cumplimiento de las actividades académicas asignadas a los profesores de medio y tiempo completo en base al Estatuto de Personal Académico de la Universidad Nacional Autónoma de México. Promover y estimular sus investigaciones para la elaboración de libros de texto, publicaciones, círculos de estudio, impartición de cursos, intercambio académico, etc.

Por otra parte lleva el control de los profesores en el aspecto de superación académica, es decir, preparar los exámenes de oposición para mejoramiento en la calidad del profesor y por lo tanto de la enseñanza; así mismo se encarga de evaluar periódicamente a los profesores y promoverlos para categorías superiores.

Llevar un control riguroso de las actividades de docencia y de investigación que están llevando a cabo los maestros de carrera, a fin de contar con un banco de datos que permita identificar los avances en los proyectos o líneas de investigación y poder determinar los medios para apoyarlos en las mismas.

Para lograr este objetivo es necesario contar con un banco de datos general y único donde se pueda consultar los antecedentes académicos y docentes de los profesores, así mismo poder llevar un control de las actividades extracurriculares que desempeñan actualmente.

Se requiere de un sistema que permita llevar un control de los informes anuales de los profesores de carrera, así como de su plan de trabajo y poder establecer un mecanismo de evaluación. De conformidad a los resultados obtenidos, se identificarán los aspectos menos desarrollados por el personal docente de carrera a fin de instrumentar los apoyos y mecanismos necesarios para elevar su rendimiento.

De esta forma también se pretende elevar el nivel y formación de cada profesor llevando a cabo cursos permanentes de informática aplicada al derecho, actualización docente en los temas de las diversas áreas del derecho, en la didáctica y pedagogía aplicada al derecho y formación docente, principalmente en las materias del nuevo plan. Se promoverán concursos de oposición abiertos y cerrados, en todas las materias.

En otro sentido es necesario contar con acceso a toda la información referente a profesores y la manera de que ellos mismos puedan establecer contacto con otras personas a fin de establecer mecanismos de intercambio de información y promover el intercambio académico a distancia.

La Secretaría General entre sus diversos objetivos están el llevar el control y la elaboración de los horarios que cada semestre son necesarios dentro de la Licenciatura en Derecho. En este punto cabe aclarar que en la actualidad se trabaja con dos planes de estudio y estarán vigentes 3 años más.

Generar los horarios cada semestre, de manera que los profesores de carrera y los de mayor antigüedad se vean beneficiados al poder cumplir con sus clases en horas seguidas o consecutivas, en el horario más favorable para ellos y de acuerdo a los requerimientos de la facultad.

En este punto hay que tomar en cuenta que la elaboración del horario correspondiente a cada semestre deberá cumplir con el objetivo planteado en el párrafo anterior. Y tomar en cuenta los siguientes puntos:

- Llevar un control estricto a cerca de la seriación y del manejo de dos planes de estudio, siendo el plan viejo de cuatro materias y el nuevo de ocho por semestre. La última generación con el plan anterior está en el décimo semestre de la licenciatura.
- Tomar en cuenta que se requerirá sustituir a algún profesor en alguna materia.
- Que el sistema propondrá un grupo de académicos para la impartición de la asignatura y que la elección responderá al currículum y a la preparación dentro del campo docente.

Otra de las actividades de la Secretaría General es la titulación de los alumnos donde establece los mecanismos de control para el desarrollo de un tema de seminario y su conclusión con la presentación del examen profesional.

Para ello es necesario tener controlados los temas que generan cada uno de los 17 seminarios con que cuenta la Facultad de Derecho. Es necesario que se lleve un control de los alumnos que están inscritos en cada seminario, del tema de tesis que se está desarrollando, del avance del mismo y del director de la tesis. Un control adicional sobre los temas de tesis es que antes de aprobar un tema este no deberá ser un "fusil" de alguna otra propuesta anteriormente registrada y/o concluida. En este punto cabe destacar que será necesario contar con un banco de datos histórico para solventar dicha necesidad.

En este rubro es necesario llevar un control del avance de los trámites que son necesarios y de la documentación que se requiere para poder llevar a cabo el registro del tema de tesis, ya que muchos de los temas son abandonados o bien el alumno registra otro tema de tesis, siendo este último el que se concluye.

Una vez terminado el proceso de registro de tesis, se requiere que el seminario apruebe junto con el director de la misma su impresión y emitir un comunicado a la Secretaría de Exámenes Profesionales para que asigne el aula, la hora y los sinodales, que examinarán al sustentante.

En este último punto se requiere que la selección sea automática, que además sea de manera aleatoria y que el grupo de sinodales que examinan sean del mismo seminario del cual surja la tesis. Que tome en cuenta el número de exámenes profesionales en los cuales ha participado cada uno de los sinodales, de tal manera que la distribución de los exámenes por cada sinodal sea similar entre los miembros de cada seminario. Y finalmente que el cargo de cada sinodal será asignado en base a la antigüedad y categoría del mismo dentro de la Facultad de Derecho.

Otras de las actividades que coordina la Secretaría General es el control de las asistencias de los profesores a las diversas asignaturas de las cuales es titular. Y no solo de los profesores de asignatura, sino además de los profesores de medio tiempo, tiempo completo y jefes de seminario.

En este aspecto es necesario contar con un control histórico de las faltas en que ha incurrido cada profesor, en que asignatura, en que fecha y cual fue el motivo por el cual faltó. En este sentido se establecer un menú de reportes y consultas de diversa índole, por ejemplo: asignaturas a las cuales ha faltado el profesor durante un determinado periodo, cual es su porcentaje de asistencia, que días a faltado y cual fue el motivo, comparativo de asistencia de los diferentes profesores que imparten la asignatura, comparativo con los profesores del mismo seminario o de toda la Facultad, reportes de asistencia por semestres, etc.

En cada una de las entrevistas llevadas a cabo con las diferentes secretarías involucradas, se pudo concluir que uno de los puntos que se mencionaba con mayor frecuencia era la implementación de un esquema de interacción con la computadora que les proporcione de una manera fácil y sencilla la información que necesiten, ya que en la mayoría de los casos se carece de una preparación en el campo computacional.

Otros de los puntos relevantes es la posibilidad de tener acceso a información generada por otras áreas.

## 2.4

# ESTRATEGIA DE SOLUCION

### *2.4.1 Análisis.*

El proceso del análisis esta dividido en 3 áreas:

- Investigación inicial,
- Estudio de viabilidad y
- Definición de requerimientos.

Este proceso contempla proporcionar el manejo con suficientes datos para permitir una decisión inteligente apreciando la justificación económica y la viabilidad técnica del desarrollo e implementación del proyecto propuesto, junto con la definición funcional, modelo ambiental, modelo conceptual y modelo de datos.

### **2.4.1.1 Investigación inicial.**

Lo primero en el proceso de análisis es la investigación inicial, cuyo resultado será la identificación de necesidades, en este caso el analista tendrá contacto con el cliente o usuario el cual puede ser un técnico, sin embargo en general resulta ser personas totalmente ajenas al área de sistemas por lo cual se debe de buscar el mejor entendimiento posible.

La investigación inicial es la base de un sistema que se va a desarrollar por lo que si llegara haber alguna confusión influiría en todas las etapas del sistema.

El analista se asegura de distinguir entre lo que "necesita" el cliente (elementos críticos para la realización) y lo que el cliente "quiere" (elementos deseables pero no esenciales) [PRE93].

La investigación inicial permite manejar el potencial estimado del sistema propuesto en términos de costos y beneficios. Este proporciona un estado de problemas existentes, costos de proyectos y esquemas para desarrollo y operación del sistema propuesto y produce un plan de trabajo con un estudio de viabilidad. La investigación inicial se divide en 3 actividades:

#### **2.4.1.1.1 Evaluar los requerimientos del proyecto.**

El objetivo de esta actividad es evaluar el problema, necesidad y validar que los costos y beneficios indicados en los requerimientos del proyecto son realizables. Los pasos de la actividad son:

- Evaluar el tiempo de desarrollo del proyecto propuesto para que éste no interrumpa las actividades de la empresa.
- Evaluar los conflictos potenciales que pueden afectar hostilmente la implementación exitosa del sistema propuesto.

- Evaluar si los requerimientos del proyecto son compatibles con el plan y objetivos de la empresa.

#### **2.4.1.1.2 Preparar la conducción de la investigación inicial.**

El objetivo de esta actividad es para organizar entrevistas y preparar materiales para ser usados en la conducción de estas actividades y son:

- Establecer listas para conducir las entrevistas.
- Identificar las unidades individuales y organizacionales para ser entrevistadas.
- Formular las preguntas acerca del problema.
- Listar documentación existente para obtener información.
- Llevar a cabo las entrevistas.
- Colectar información validada.

#### **2.4.1.1.3 Analizar los datos colectados.**

El objetivo de esta actividad es analizar los datos recolectados en orden para hacer determinaciones preliminares relativas a los costos del proyecto, los pasos son:

- Analizar los datos que se necesitan.
- Analizar los datos con problemas.
- Analizar anticipadamente los beneficios.

#### **2.4.1.2 Estudio de viabilidad.**

Cuando una empresa necesita un sistema la primera consideración ,por lo general, es la del costo. con el cual muchos clientes desisten del proyecto. Es por ello que el analista debe de

realizar el estudio e indicarle detalladamente al cliente además de su costo, los beneficios que obtendría. Por muy caro que sea el sistema si en un tiempo razonable para la empresa se recupera la inversión (podría incrementar su producción, mejorar el servicio al cliente, tomar decisiones rápidamente, etc.) sería viable el sistema.

El estudio de viabilidad esta comprometido con proporcionar al administrador suficientes datos para permitir decisiones inteligentes valorando la justificación económica y viabilidad técnica del proyecto propuesto. La documentación se enfoca sobre la definición de características del sistema actual y los requerimientos del sistema propuesto. El estudio de viabilidad se divide en 5 actividades:

#### **2.4.1.2.1 Conducir entrevistas a los usuarios.**

El objetivo de esta actividad es recoger información que pueda hacer posible el procesamiento y el control del usuario para evaluar los costos y beneficios del proyecto propuesto. Los pasos son:

- Identificar el manejo de operaciones y personal responsable de las áreas que intervienen en el sistema para ser entrevistado.
- Preparar cuestionarios para entrevistas.
- Listar y conducir las entrevistas de estudio de viabilidad.
- Recolectar información.

#### **2.4.1.2.2 Revisar el sistema actual.**

El objetivo de esta actividad es ganar entendimiento de lo que el sistema actual realiza y como lo hace. Los pasos son:

- Revisar las funciones organizacionales y sus relaciones.

- Revisar los tipos y volúmenes de entradas y salidas.
- Revisar el ambiente técnico (hardware, plataformas, facilidades de red, y objetos potencialmente reutilizables).
- Analizar los procedimientos de control (acceso a red, acceso a base de datos, backup y recuperación, y programa de control de procedimientos).
- Identificar componentes los cuales pueden ser reutilizados en el sistema propuesto.

#### **2.4.1.2.3 Definir los objetivos del sistema.**

El objetivo de esta actividad es resumir los requerimientos del sistema propuesto, los pasos a seguir son:

- Identificar los objetivos funcionales.
- Identificar objetivos de arquitectura.

#### **2.4.1.2.4 Identificar alternativas.**

El objetivo de esta actividad es para identificar soluciones alternativas que puedan satisfacer los requerimientos del usuario.

- Identificar alternativas para satisfacer los objetivos funcionales.
- Identificar alternativas para satisfacer los objetivos de arquitectura.

#### **2.4.1.2.5 Evaluar costos y beneficios.**

El objetivo de esta actividad es comparar costos y beneficios para determinar la viabilidad económica de cada alternativa de desarrollo, los pasos son:

- Evaluar beneficios para ser derivados del sistema propuesto.
- Revisar los costos operativos corrientes y el desarrollo de éstos.
- Comparar desarrollo y costos de operación para identificar beneficios.

### **2.4.1.3 Definición de requerimientos.**

El objetivo de la fase de definición de requerimientos es definir los requerimientos físicos y funcionales del sistema propuesto. La definición de requerimientos se divide en 7 actividades.

#### **2.4.1.3.1 Conducir entrevistas para definición de requerimientos.**

- Lista de entrevistas para definición de requerimientos.
- Conducir entrevistas para definición de requerimientos.
- Recolectar documentación validada.

#### **2.4.1.3.2 Definir los requerimientos del sistema.**

El objetivo es definir las funciones para realizar el sistema, los pasos son:

- Investigar y llevar a cabo los diagramas de funcionalidad o de operación.
- Definir el modelo ambiental.
- Desarrollar los diagramas de flujo de datos.
- Desarrollar el diccionario de datos.
- Definir requerimientos de interface.

#### **2.4.1.3.3 Definir los requerimientos del cliente y del servidor.**

El objetivo es definir los requerimientos para establecer el ambiente Cliente/Servidor el cual pueda soportar los requerimientos arquitectónicos y funcionales del sistema. Los pasos

- Definir requerimientos de hardware del cliente.
- Definir requerimientos de software del cliente.
- Definir requerimientos de hardware del servidor.
- Definir requerimientos de software del servidor.

#### **2.4.1.3.4 Definir los requerimientos sobre los servicios de la red.**

- Definir tiempos de respuesta.
- Definir requerimientos de salida.
- Definir los requerimientos de confiabilidad y disponibilidad.
- Definir topología y componentes de la red.

#### **2.4.1.3.5 Definir los requerimientos de implementación.**

El objetivo es definir los requerimientos físicos que deben ser satisfechos en la implementación del sistema propuesto, los pasos son:

- Definir los requerimientos del performance del sistema propuesto, requerimientos de tiempo, requerimientos de contingencias ante fallas, y requerimiento de interface.
- Definir los requerimientos de ambiente del sistema propuesto(e.j. requerimientos de equipo, soporte del software, de interface, seguridad y privacidad, control.).

- Definir los requerimientos organizacionales del sistema propuesto (ej. cambios de posición, capacitación, cambios en la clasificación del trabajo).
- Definir los requerimientos de desarrollo del sistema propuesto ( herramientas CASE, capacitación y tiempo).

#### **2.4.1.3.6 Identificar alternativas.**

El objetivo es identificar cada alternativa práctica para diseñar el sistema, los pasos son:

- Identificar los paquetes de software potenciales, redes y/o computadoras que puedan satisfacer los requerimientos.
- Conducir investigación y contactar vendedores para información adicional.
- Analizar el potencial de cada alternativa.
- Establecer criterios para evaluar alternativas.

#### **2.4.1.3.7 Construir el modelo de requerimientos.**

El objetivo de esta actividad es construir un modelo que extrapole requerimientos de los resultados de las actividades realizadas durante la definición de requerimientos.

### ***2.4.2 Diseño.***

La función del diseño es completada en 2 fases:

- Diseño externo y
- Diseño interno.

#### **2.4.2.1 Diseño externo.**

La fase de diseño externo proporciona la transición del modelo de requerimientos a un conjunto de modelos de diseño orientado-a-usuarios. Las actividades realizadas están enfocadas a la capacidad de las organizaciones de usuarios para monitorear el diseño interno y la implementación subsecuente del sistema para asegurar que los requerimientos hayan sido conocidos. El diseño externo se divide en 6 actividades principales.

#### **2.4.2.1.1 Revisar los requerimientos de definición de datos.**

Los pasos son:

- Revisar la definición de requerimientos de los resúmenes de entrevistas.
- Revisar los documentos soportados y otros recolectados durante la definición de requerimientos.

#### **2.4.2.1.2 Definir subsistemas.**

El objetivo es diseñar los subsistemas y asignar actividades a cada subsistema. Los pasos son:

- Definir las funciones de la interface.
- Definir los subsistemas funcionales.

#### **2.4.2.1.3 Definir las funciones cliente/servidor y asignar actividades a procesadores.**

El objetivo es definir las funciones del ambiente cliente/servidor y asignar actividades a los procesadores. Los pasos son:

- Asignar las funciones a los clientes.
- Asignar las funciones a los servidores.

#### **2.4.2.1.4 Definir sistema de controles de seguridad.**

Los pasos son:

- Definir los métodos de seguridad física.
- Definir los métodos de control procedural.

#### **2.4.2.1.5 Evaluar alternativas de diseño.**

Los pasos son:

- Evaluar alternativas para satisfacer requerimientos de performance.
- Evaluar alternativas para satisfacer los requerimientos de datos.
- Evaluar alternativas para satisfacer los requerimientos del medio.
- Evaluar alternativas para satisfacer los requerimientos organizacionales.

#### **2.4.2.1.6 Terminación preliminar del modelo de diseño.**

La última actividad del diseño externo es la construcción de un modelo de diseño preliminar. El modelo considera los requerimientos para la metodología estructurada y la arquitectura cliente/servidor. Los aspectos del modelo muestran la distribución de la

conducta específica en el uso de descripciones para formar subsistemas. Los aspectos cliente/servidor del modelo muestran como las actividades de subsistemas son asignadas a los clientes y servidores. Los segmentos de modelos estructurado y cliente/servidor están combinados para crear un modelo preliminar de diseño íntegro para el sistema cliente/servidor a ser implementado. Las salidas de esta actividad son descripciones alto-nivel (High-level) de todos los subsistemas y esquemas de cada subsistema mostrando la ubicación de funciones.

#### **2.4.2.2 Diseño interno.**

El diseño interno establece la estructura para la implementación y pruebas del sistema. Durante esta fase, los documentos están produciendo una especificación precisa de como los elementos funcionales definidos durante el diseño externo pueden ser transformados en una arquitectura física para un sistema. Todas las especificaciones necesarias para código de programas de aplicación están incluidas en los campos entregados. El diseño interno se divide en 5 actividades .

##### **2.4.2.2.1 Diseño de la arquitectura del sistema.**

El objetivo es diseñar los elementos técnicos que deben ser integrados en el diseño del sistema. Los pasos son:

- Diseñar la arquitectura del sistema.
- Diseñar la arquitectura cliente/servidor.
- Diseñar la arquitectura de recursos compartidos.
- Definir la arquitectura de base de datos relacional.
- Diseñar la arquitectura de procesamiento distribuido.

#### **2.4.2.2.2 Especificar estructuras de datos.**

El objetivo es terminar las especificaciones de datos para proporcionar entradas y salidas al sistema. Los pasos son:

- Especificar transacciones.
- Especificar presentaciones.
- Especificar reportes.
- Preparar salidas indexadas.

#### **2.4.2.2.3 Diseñar la base de datos.**

El objetivo es diseñar los requerimientos de la base de datos para implementar el diseño del sistema.

- Transformar funciones en un diseño de base de datos.

#### **2.4.2.2.4 Consideraciones específicas del diseño.**

El objetivo es definir las consideraciones especiales pertinentes para el diseño del sistema. Los pasos son:

- Terminar los procedimientos de control de usuarios.
- Terminar los procedimientos de control de acceso.
- Terminar los procedimientos de mantenimiento de archivos.

- Terminar los procedimientos de control de recuperación.
- Preparar los manuales de procedimientos.

#### **2.4.2.2.5 Definir planes de implementación y pruebas.**

El objetivo es desarrollar planes de trabajo para la implementación y pruebas del sistema. Los pasos son:

- Definir planes para extraer componentes reutilizables.
- Definir planes para la creación de nuevos componentes.
- Desarrollar el plan para implementar el sistema cliente/servidor.
- Definir el plan para instalar el sistema manejador de base de datos (DBMS).
- Definir el plan de trabajo para crear la base de datos
- Desarrollar el plan para instalar la aplicación.
- Desarrollar los planes de trabajo para probar la red, aplicación y DBMS.

### ***2.4.3 Implementación.***

El diseño interno establece la estructura para la implementación y prueba del sistema. Durante esta fase los elementos especificados en el modelo del diseño son transformados en una arquitectura física para un sistema estructurado bajo cliente/servidor. El proceso de implementación se divide principalmente en 10 actividades.

#### **2.4.3.1 Revisar el diseño interno de datos.**

El objetivo es revisar las especificaciones de mantenimiento almacenadas en el repositorio.

Los pasos son:

- Revisar el modelo de diseño interno.
- Revisar la documentación soportada y otros documentos recolectados durante la fase del diseño externo.

### **2.4.3.2 Extraer componentes reutilizables.**

El objetivo de esta actividad es reducir costos de implementación por utilizar código escrito que sea reutilizable. Los pasos son:

- Extraer componentes reutilizables del sistema actual.
- Generalizar los componentes extraídos.
- Personalizar los componentes generalizados.

### **2.4.3.3 Conducir las pruebas de unidad e integridad.**

El objetivo es conducir las pruebas de unidad para asegurar que las operaciones individuales y métodos se comporten como se espera y conforme a las especificaciones de diseño, y conducir la integración de pruebas. Los pasos son:

- Actualizar el plan de pruebas de unidad.
- Actualizar el plan de pruebas de integración.

- Crear especificaciones de prueba.
- Conducir pruebas.
- Corregir errores e iterar.

#### **2.4.3.4 Preparar manual de usuario.**

El objetivo es preparar el manual de usuario. El contenido del manual esta dividido en :

- Información General (información que explica el propósito de la guía de usuario y los términos y referencias que son sustantivas para la interpretación por el usuario.).
- Resumen del sistema( ej. explica las operaciones del sistema, la configuración del sistema, y las capacidades de ejecución).
- Procedimientos de consulta y actualización (el cual incluye escenarios e instrucciones paso-paso para consultar las bases de datos.).

#### **2.4.3.5 Preparar documentación técnica.**

El objetivo es recolectar y organizar la documentación necesaria para las operaciones personales para el soporte del sistema. Los pasos son:

- Preparar documentación de operaciones que proporcione el equipo de operaciones con la información requerida para soportar la aplicación.
- Preparar documentación del sistema que proporcione soporte al equipo con la información necesaria para mantener y/o mejorar la aplicación.

### **2.4.3.6 Instalar el sistema cliente/servidor.**

El objetivo es implementar el diseño del sistema cliente/servidor. Los pasos son:

- Adquirir los componentes de red.
- Preparar para instalación.
- Instalar el servidor de red, cables y software.
- Instalar y probar el sistema operativo de la red, utilerías y directorios de usuario.
- Convertir los sistemas de usuario simple a sistemas multiusuario.
- Capacitar usuarios.

### **2.4.3.7 Instalar el DBMS.**

El objetivo es instalar el DBMS seleccionado para facilitar la creación y mantenimiento de la base de datos.

- Instalar el engine del DBMS.
- Instalar el subsistema de herramientas de definición.
- Instalar la interfaz de procesamiento.
- Instalar las herramientas de desarrollo de aplicaciones.
- Instalar el diccionario de datos.
- Instalar el administrador de datos.

#### **2.4.3.8 Crear la base de datos.**

El objetivo es implementar la base de datos del diseño. Los pasos son:

- Definir la estructura de la base de datos en el DBMS.
- Asignar espacio.
- Crear los datos de la base de datos.

#### **2.4.3.9 Instalar la aplicación.**

El objetivo es instalar la aplicación en el ambiente cliente/servidor. Los pasos:

- Crear discos.
- Cargar archivos sobre el servidor.
- Cargar archivos sobre los clientes.

#### **2.4.3.10 Verificar y validar aplicaciones.**

El objetivo por último es verificar y validar que las aplicaciones reúnan los requerimientos del sistema.

## 2.5

# ALTERNATIVAS DE SOLUCION

Para cubrir los requerimientos descritos en los puntos anteriores es necesario desarrollar un modelo de información que permita explotar los datos, compartir la información y tener un control amplio sobre los mismos.

Para poder evaluar las alternativas se debe de pensar en la infraestructura existente, el tiempo de respuesta que se llevaría el realizar las modificaciones necesarias, los recursos humanos y físicos con que se cuenta, las necesidades de crecimiento, la facilidad de mantenimiento y su funcionalidad.

Debido al tipo de información que se maneja es necesario contar con elementos de seguridad para casi cualquier eventualidad que implique algún tipo de pérdida de datos.

A continuación se presentan las diferentes alternativas a tomar en cuenta analizadas en el presente proyecto:

- Realizar un mantenimiento a los programas existentes.
- Actualizar la versión de Paradox y mantener la red Netware.
- Utilizar SqlAnyware como base de datos, algún software cliente y una red punto a punto (p.e. windows para trabajo en grupo)
- Diseñar e implementar un nuevo sistema bajo un ambiente cliente/servidor en un ambiente gráfico para un ambiente multiplataforma.

### ***2.5.1 Realizar un Mantenimiento Mayor a los Programas Existentes.***

#### **2.5.1.1 Descripción.**

El objetivo es lograr que los programas actuales (scripts) que se tienen en paradox puedan ejecutarse en un ambiente de red, para ello es necesario agregar líneas de bloqueo y desbloqueo para las aplicaciones existentes.

Es necesario contar con una centralización completa de la información actual. Adicionalmente será necesario realizar una depuración exhaustiva de los programas existente para reducir el tiempo de ejecución de los mismos, tratando de agilizar los procesos actuales.

Será necesario realizar un diagnóstico de los nuevos procesos a ser automatizados e integrar a la mayor cantidad de gente para que intervenga en dichas actividades para un mejor manejo de la información.

Establecer los mecanismos necesarios para que los usuarios tengan acceso a la información que se encuentra dentro de la red, es decir, otorgarles una clave, permisos, configurar su equipo, instalar el software necesario, capacitarlo sobre su uso, etc.

### **2.5.1.2 Equipo y recursos disponibles.**

Los recursos materiales necesarios para llevar a cabo este cambio son pocos o casi nulos. Pero no es así, en los recursos humanos, ya que se requiere de varias personas para poder llevar a cabo el desarrollo de las nuevas aplicaciones en un tiempo de respuesta relativamente corto, además de que se requiere de capacitarlos ya que en la actualidad existe en el mercado poca gente que sabe utilizar Paradox para DOS, incrementándose dramáticamente la curva de aprendizaje de los desarrolladores.

### **2.5.1.3 Ventajas.**

La inversión es baja, no se requiere ningún desembolso grande ya que la capacitación correría por parte del personal que ya conoce el sistema y Paradox, no se compraría ningún software de desarrollo, de cliente o de red.

Los recursos que consume las aplicaciones clientes son muy básicas que cualquier equipo existente actualmente cubre por mucho sus especificaciones.

### **2.5.1.4 Desventajas.**

De las personas que participan en el proyecto actualmente solo dos conocen el funcionamiento de Paradox para DOS 3.5; por lo que sumado al tiempo de desarrollo se debe agregar el tiempo de capacitación para desarrolladores.

El ambiente de desarrollo es pesado y difícil de encontrar los errores existentes. Se deben de programar muchas funciones de tipo estándar, por lo que se deben de realizar primero la programación de utilerías básicas para las aplicaciones generales.

Con la experiencia de los sistemas actuales, no se sugiere la utilización de Paradox para DOS en el desarrollo de aplicaciones, ya que son lentas.

La interacción con las aplicaciones de oficina como un procesador de palabras o una hoja de cálculo, es difícil para usuarios no experimentados y en el caso de que se realizara una conexión a las bases de datos esta carecería de seguridad ya que cualquier usuario podría borrar o alterar el contenido de las bases de datos a las cuales tiene acceso.

No se cuenta con un esquema de seguridad para el control de la información de manera efectiva, solo se cuenta con seguridad de acceso a nivel archivos.

Con respecto a la integridad de datos, se cuenta con un esquema básico de validación de datos, la integridad referencial se limita al manejo de llaves primarias.

Se invierte mucho en horas-hombre para el desarrollo.

La elaboración de los reportes es precaria, por lo que para los actuales momentos no es suficiente los reportes que se pueden elaborar con el reporteador incluido.

### ***2.5.2 Actualizar la versión de Paradox y Mantener la Red Netware.***

La versión existente en el mercado de paradox es la 7.0 para ambiente gráfico windows.

#### **2.5.2.1 Descripción.**

Las aplicaciones se desarrollaran en ambiente gráfico. Se actualizará el formato de la versión anterior ya que Paradox 7.0 tiene un formato de almacenamiento diferente.

#### **2.5.2.2 Equipo y recursos disponibles.**

Para los desarrolladores se requiere computadoras lo mas veloces posible (pentium a 120 Mhz), de 12 a 16 Mb de memoria RAM, cerca de 40 Mb de espacio en disco y Windows95.

Los equipos para las aplicaciones cliente requiere de utilizar Windows95 con 8 Mb de RAM mínimo. Se requiere comprar Paradox 7.0 y obtener una licencia de uso de aproximadamente 20 usuarios.

### **2.5.2.3 Ventajas.**

Se tiene un ambiente de desarrollo gráfico y rápido análogo a los ya existentes por lo cual se asegura su permanencia en el mercado, por un largo tiempo.

Como existe una rotación permanente de personal que se encarga de desarrollar las aplicaciones y dar mantenimiento a los sistemas, es importante notar que el mercado de trabajo actualmente provee de desarrolladores para ambientes gráficos, por lo que pasar de un software cliente a otro es relativamente fácil y el tiempo de adaptación para el ambiente de desarrollo es corto.

Se tiene un grado mayor de seguridad en las bases de datos, mediante la utilización de claves para las tablas.

### **2.5.2.4 Desventajas.**

Para el manejo de grandes volúmenes de datos el motor de tiempo de corrida de Paradox se vuelve lento y se requiere de más memoria para obtener un tiempo de respuesta aceptable.

Se requiere comprar dos o tres licencias de Paradox para desarrollo y cerca de veinte licencias de uso para el motor de tiempo de corrida.

La seguridad se limita al uso de las bases de datos y no por el uso de la información.

El control de seguridad de los archivos que contienen las bases de datos sigue siendo a nivel archivo, por lo que la seguridad es baja.

Netware no ofrece un control sobre los atributos de la base de datos, es decir a nivel información, sino que simplemente a nivel archivo de datos.

El equipo que contiene la base de datos deberá ser bastante poderoso para soportar la carga de petición de archivos y al propio sistema operativo de red (Netware):

Mediante el uso de Netware como sistema operativo de red, combinado con un manejador de bases de datos como Paradox no evitarían la sobrecarga del canal físico de comunicación y como ya se menciona el canal no será ocupado solo por aplicaciones de la base de datos.

### ***2.5.3 Utilizar SqlAnyware como Base de Datos, algún Software Cliente y una Red Punto a Punto.***

#### **2.5.3.1 Descripción.**

Se utilizará SqlAnyware como administrador de la base de datos, dejándole toda la carga de seguridad e integridad de datos. Se desarrollaran las aplicaciones cliente en cualquier programa de desarrollo visual, tal como Visual Basic, PowerBuilder o Delphi. Se instalará una red punto a punto con ambiente gráfico windows para trabajo en grupo y se crearán grupos de trabajo para un mayor control sobre el acceso a los recursos.

#### **2.5.3.2 Equipo y recursos disponibles.**

Se requiere de que todos los equipos de cómputo cuenten con Windows para trabajo en grupo y que estos se encuentren interconectados por medio de la red.

El equipo donde se instalará SqlAnyware tiene que ser lo mas veloz posible, con una cantidad de memoria de al menos 16 Mb de RAM y de preferencia que sea una máquina dedicada como servidor de bases de datos.

Si se quisiera tener mayor control sobre los accesos compartidos será necesario contar con Windows NT, por lo que se requerirá de un equipo con mejores características y más memoria que el administrador de bases de datos.

### **2.5.3.3 Ventajas.**

Si se utiliza windows para trabajo en grupo éste viene instalado de fábrica en todos los equipos actuales por lo que no se requerirá de licencias o de algún desembolso por éste concepto.

Todos los equipos están interconectados físicamente por lo que no hay ningún costo por concepto de interconexión.

Con SqlAnyware se podrán implementar fácilmente medidas de seguridad e integridad de datos. Se evitará la sobrecarga del canal por evitar las transmisión de información innecesaria.

Se podrá utilizar cualquier software de desarrollo visual independientemente de la plataforma en que se encuentre, ya que SqlAnyware contiene los controladores necesarios para cualquier herramienta visual.

### **2.5.3.4 Desventajas.**

SqlAnyware se vende por número de licencias por lo que la compra de una versión con 20 licencias resulta un poco elevado. El desempeño de SqlAnyware para una organización de tamaño pequeño o mediano es bueno pero cuando se maneja un gran volumen de datos y con muchos usuarios al mismo tiempo se vuelve lento y la seguridad disminuye dramáticamente.

La seguridad por parte de windows para trabajo en grupo es solo a nivel acceso por lo que si algún usuario quiere ver los archivos de la base de datos este podrá ocasionar algún daño a los datos.

## ***2.5.4 Diseñar e Implementar un Nuevo Sistema Gráfico bajo un Ambiente Cliente/Servidor.***

### **2.5.4.1 Descripción.**

Al igual que otras propuestas mencionadas la información se centralizaría, en éste caso en una verdadera base de datos relacional, como puede ser Oracle, Sybase o Informix. El servidor en donde se instalaría la base de datos debería de tener gran capacidad de proceso y de almacenamiento por lo que se sugiere una computadora con procesador RISC y sistema operativo UNIX que se caracterizan por ser muy robustos.

Los clientes deben de tener interfaz gráfica para que el manejo de las aplicaciones sea lo más sencillo, el equipo debe ser de lo más común en la facultad para evitar comprar equipos de computo que incrementen el costo del sistema.

Para la conexión del cliente con el servidor se sugiere sea un manejador de red (drive) del propietario de la base de datos elegida, en lugar de utilizar librerías ODBC (conexión de base de datos abiertas), para lograr una mayor rapidez.

### **2.5.4.2 Equipo y recursos disponibles.**

La facultad de derecho cuenta actualmente con 2 Workstation marca SUN modelos SpareStation y UltraSpare, los cuales trabajan con procesadores RISC y sistema operativo Solaris (Unix de SUN). La UltraSpare que sería la seleccionada tiene gran capacidad 6 Gbytes en disco duro, a 143 Mhz y 64 Mbytes de Memoria.

En la facultad de derecho se cuentan con más de 400 computadoras personales la mayoría con procesador 486 y pentium compatibles con IBM, con monitor VGA a color, mínimo 4 Mbytes en RAM y son utilizadas en gran medida por el personal de esta dependencia. Como en la mayoría de la universidades y el país la interface gráfica más usada es Windows de Microsoft. Este tipo de computadoras y la interface gráfica se sugieren para utilizarse como maquinas cliente.

La facultad adquirió una licencia del manejador de base de datos Sybase SQL Server 11 con permiso para 32 conexiones a la vez, donde actualmente se corren ya algunas aplicaciones, por lo que es deseable que el sistema utilice el propio Sybase.

También se cuenta con una licencia de desarrollo de PowerBuilder Enterprise 5.0 la cual fue donada por la Dirección General de Administración Escolar. Aunque se había pensado utilizar otro front-end como Delphi de Borland, podemos decir que ambos front-ends son los mejor evaluados la elección de cualquiera de los 2 será acertada.

Al comprar la licencia de Sybase también se contó con los manejadores de red (drivers) nativos denominados dlibrary que conectan al servidor de base de datos (Sybase) con el cliente.

### **2.5.4.3 Ventajas.**

Entre las ventajas más importantes esta la centralización de la información en una base de datos robusta.

Con la arquitectura cliente/servidor podemos asegurar la menor carga en la red debido a que el cliente solo envía su petición al servidor, este la procesa y devuelve la respuesta al cliente.

La interface gráfica Windows resulta ser muy familiar para los usuarios del sistema por lo cual para ellos será fácil el manejo de la misma y no se capacitara en este ámbito.

En cuanto a invertir recursos en software podemos mencionar que lo que se requiere se tiene por lo que no se gastará ni un centavo más en este rubro.

En equipos PC que servirán como clientes tampoco hace falta una inversión en nuevo equipo o ampliaciones ya que el actual equipo puede ejecutar satisfactoriamente aplicaciones desarrolladas en PowerBuilder.

En lo que se refiere a comunicaciones las diversas áreas de la Facultad de Derecho involucradas en el alcance del sistema cuentan con las conexiones necesarias a la red de la facultad.

#### **2.5.4.4 Desventajas.**

La única desventaja que posee esta propuesta es la falta de conocimiento del front-end PowerBuilder, con el cual se desarrollarán las aplicaciones, pero contamos con los manuales y conocimientos esenciales para llevar a cabo este desarrollo. Sin embargo como se tiene cierto conocimiento en herramientas de desarrollo visual las diferencias esenciales con PowerBuilder son pocas.

## 2.6

# DIAGRAMAS DE FUNCIONALIDAD

Un sistema o procedimiento es un conjunto de operaciones o de etapas que en forma cronológica se establecen para llevar a cabo un determinado tipo de trabajo.

Los diagramas de funcionalidad u operacionales son diagramas que representan en forma gráfica la secuencia en que se realizan las operaciones de un determinado procedimiento y/o el recorrido de las formas o materiales. Con el fin de tener una visión de conjunto de un sistema o procedimiento se utilizan los diagramas de flujo donde se muestran las unidades orgánicas o puestos que intervienen en el procedimiento, las operaciones que realizan, la secuencia de las mismas y el equipo utilizado en cada caso.

### *2.6.1 Técnicas de Simplificación de Operaciones.*

El análisis del sistema o procedimiento implica descomponerlo en sus partes para conocer la naturaleza, los objetivos, los responsables de su ejecución, el lugar y tiempo en que se debe realizar.

Para analizar la información que permita después llevar a cabo un proceso de simplificación de operaciones se debe buscar las respuestas a las siguientes preguntas:

- **¿Qué trabajo se hace?** Se pretende determinar el propósito y objetivo de la naturaleza o tipo de labores que se realizan en la unidad orgánica y los resultados que se obtienen.
- **¿Quién lo hace?** Se intenta definir qué unidades orgánicas o personas intervienen en el procedimiento y además las aptitudes del personal para la realización de una tarea específica y la actitud que adopta en la relación de la misma.
- **¿Cómo se hace?** Se busca determinar los medios con los que se cuenta para realizar la tarea, que comprende los métodos y técnicas aplicados y adaptados a la dependencia y a los equipos e instrumentos utilizados.
- **¿Cuándo se hace?** Se determina el orden o secuencia en que se desarrollan las actividades de la unidad orgánica, es decir, la ubicación geográfica y domicilio de las oficinas, la funcionalidad de éstas y la distribución del espacio de las mismas.
- **¿Porqué se hace?** Se busca la justificación de la existencia de ese trabajo o de su procedimiento. Se pretende conocer los objetivos de las acciones que integran el trabajo o procedimiento al igual que la primera pregunta.

El analista establece como resultado no sólo la descripción de las operaciones del sistema o procedimiento en forma más precisa y definida, sino también la posibilidad de mejorarlo y para ello se tienen las siguientes alternativas:

- **Eliminar el trabajo innecesario.** Se parte del principio de que todo sistema o procedimiento es perfectible y por ello es necesario mejorarlo, eliminando esfuerzos y tareas innecesarias. Por ejemplo, el exceso en el llenado de formas, de firmas, registros, etc.
- **Modificar las operaciones o partes de ellas.** Es conveniente subdividir el trabajo pero no tan detalladamente que haga ineficiente el procedimiento o sistema, al someterlo a repetidas revisiones que repercutan en pérdida de tiempo y energía. Por ejemplo se pueden cambiar dos operaciones en una sola o bien modificar el procedimiento al incorporar operaciones que lo hacen más lógico, completo y congruente.
- **Cambiar el orden de las operaciones.** El simple cambio en el orden de las operaciones puede hacer más eficiente un procedimiento; por ejemplo eliminar el desplazamiento inútil de documentos.
- **Simplificar las operaciones necesarias.** Aquí es necesario estudiar más detalladamente cada una de las operaciones del procedimiento para determinar posibilidades de simplificación. Pueden aplicarse los estudios de tiempo y movimientos, de esquemas del lugar de trabajo, etc.
- **Modificar las actividades de los puestos y equilibrar cargas de trabajo.** Cuando se observa en algunos puestos que se consignan los asuntos por negligencia, falta de adiestramiento del personal o mala distribución de la carga de trabajo. Es necesario en los dos primeros casos controlar, motivar y capacitar al personal y en el último caso, replantear las actividades entre los empleados que participan en los procedimientos.

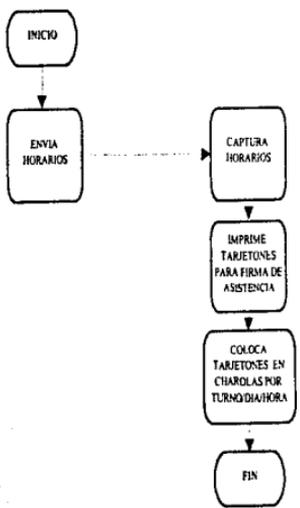
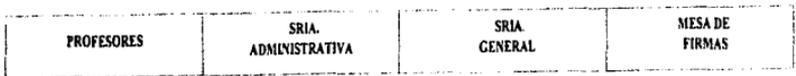
Para llevar a cabo la simplificación de operaciones se necesita establecer un programa que pueda contener los temas siguientes:

- **Seleccionar un trabajo importante.** El seleccionar un trabajo requiere de una consideración detenida, ya que sólo un trabajo de gran volumen o técnica difícil ofrecerá la oportunidad para efectuar una simplificación de trabajo considerable.
- **Dividirlo.** Es el proceso de clasificar los detalles o componentes de un trabajo, sistema o procedimiento, de tal manera que pueda examinarse en detalle. Para ello, es posible utilizar un diagrama.
- **Hacer preguntas sobre los detalles, con una mente abierta.** Constituye un requisito absolutamente necesario para la preparación de los diagramas, ya que sólo un completo conocimiento puede producir una visión total de las operaciones para pasar luego al análisis que conduce a las mejoras.
- **Desarrollar las mejoras que se propondrán.** Constituye el resultado final de la división, cuadros e interrogatorios y requiere una buena dosis de ingenio, imaginación y lógica.
- **Instalar las mejoras.** Requiere la participación y cooperación de los empleados para que se pueda instalar la o las mejoras del sistema o procedimiento.

### ***2.6.2 Diagramas.***

Para el sistema que se pretende desarrollar en la presente tesis se tienen los siguientes diagramas de funcionalidad o flujo de operaciones. Las diagramas muestran el flujo de actividades que deben realizar las áreas para obtener su información así mismo las demás áreas involucradas para cada caso.

2.6.2.1 Diagrama de inasistencia de personal docente.

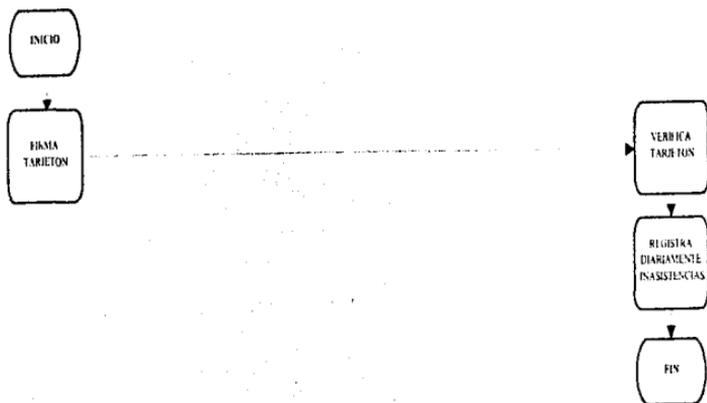


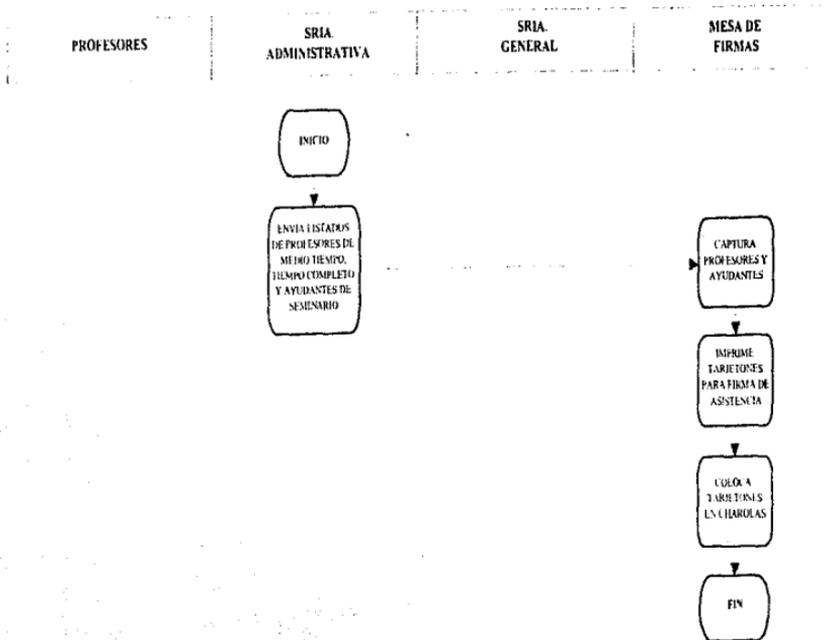
PROFESORES

SRIA.  
ADMINISTRATIVA

SRIA.  
GENERAL

MESA DE  
FIRMAS



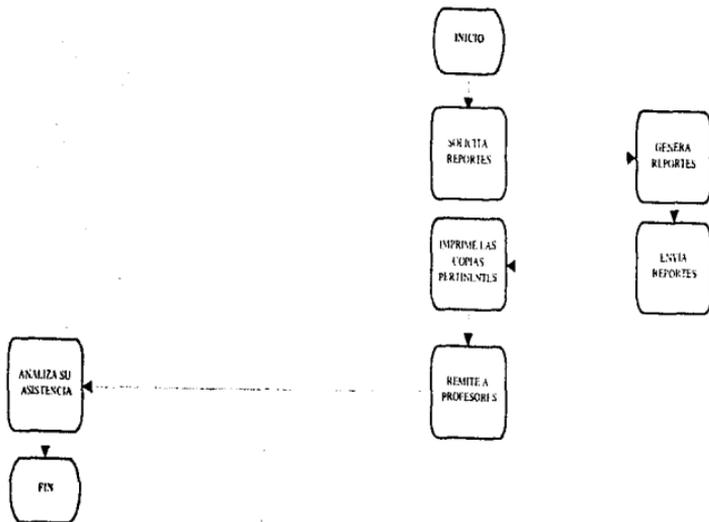


PROFESORES

SRIA  
ADMINISTRATIVA

SRIA  
GENERAL

MESA DE  
FIRMAS



MESA DE FIRMAS

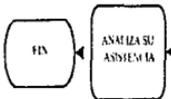


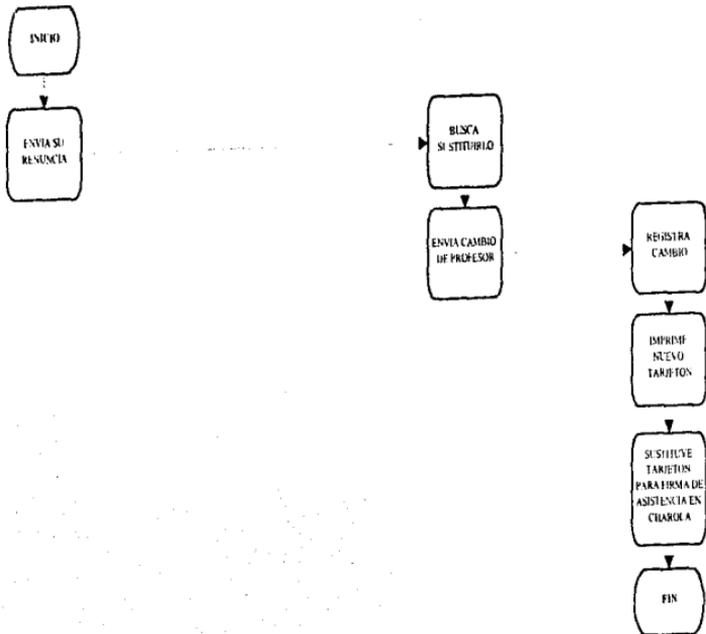
SRIA GENERAL



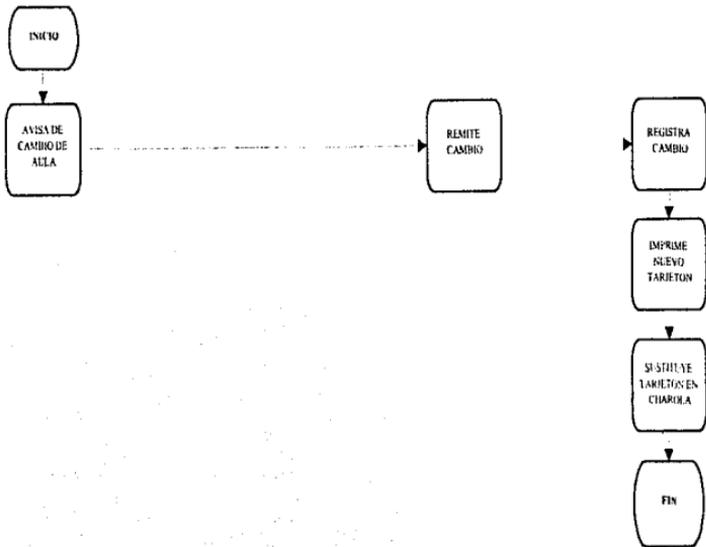
SRIA ADMINISTRATIVA

PROFESORES

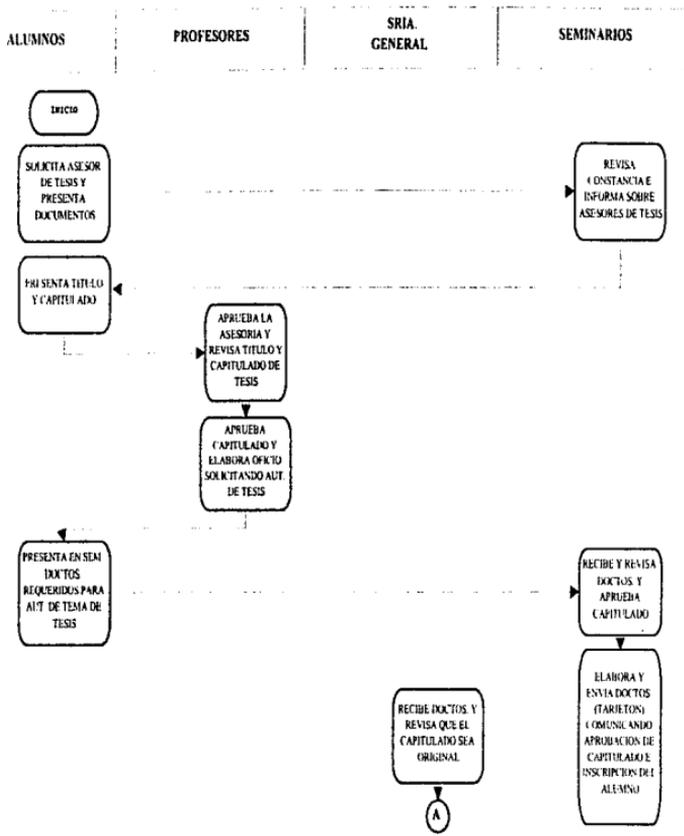


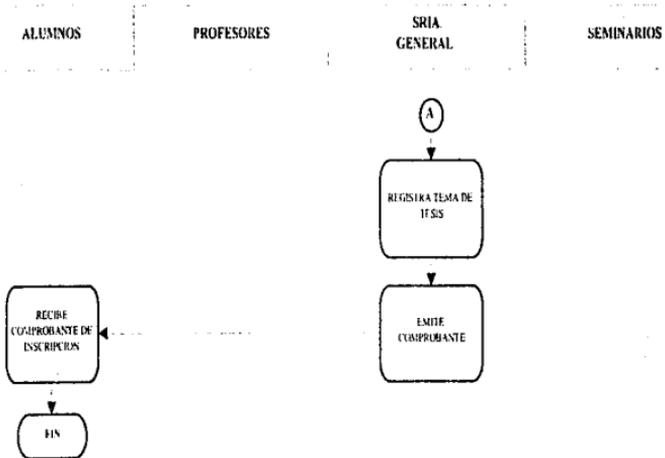


PROFESORES	SRIA. ADMINISTRATIVA	SRIA. GENERAL	MESA DE FIRMAS
------------	-------------------------	------------------	-------------------

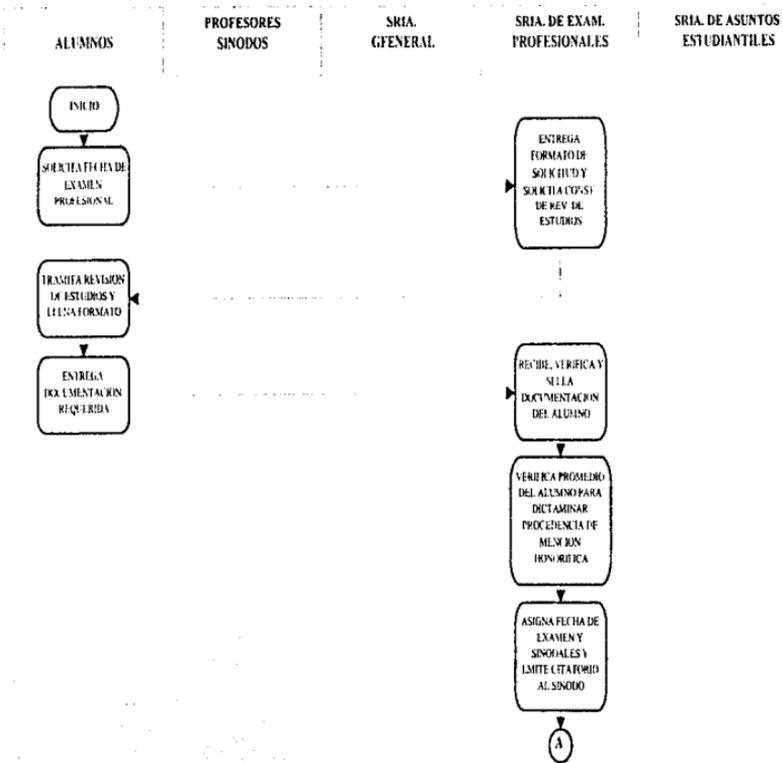


### 2.6.2.2 Diagrama de registro de tesis.





### 2.6.2.3 Diagrama de examen profesional.



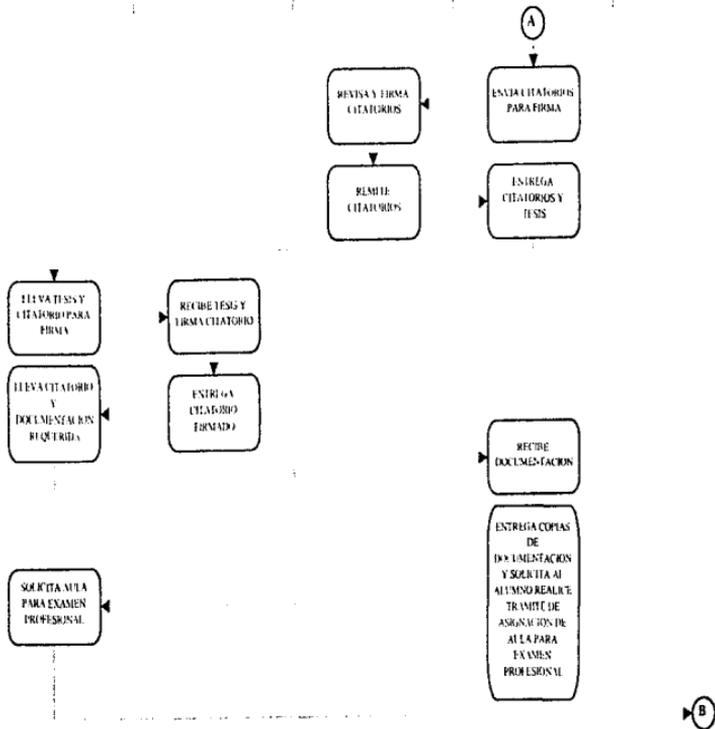
SRIA. DE ASUNTOS ESTUDIANTILES

SRIA. DE EXAM. PROFESIONALES

SRIA. GENERAL

PROFESORES SINODOS

ALUMNOS



ALUMNOS

PROFESORES  
SINODOS

SRIA.  
GENERAL

SRIA. DE EXAM.  
PROFESIONALES

SRIA. DE ASUNTOS  
ESTUDIANTILES

ENTREGA COPIA Y  
COMPROBANTE

ASIGNA AULA Y  
ENTREGA  
COMPROBANTE

RECIBE  
COMPROBANTE Y  
PUBLICA FECHA Y  
LUGAR DEL  
EXAMEN  
PROFESIONAL

EL DIA DEL  
EXAMEN  
PROFESIONAL  
PREPARA EL AULA  
Y ENVIA ACTAS Y  
CONSTANCIAS AL  
SINADO

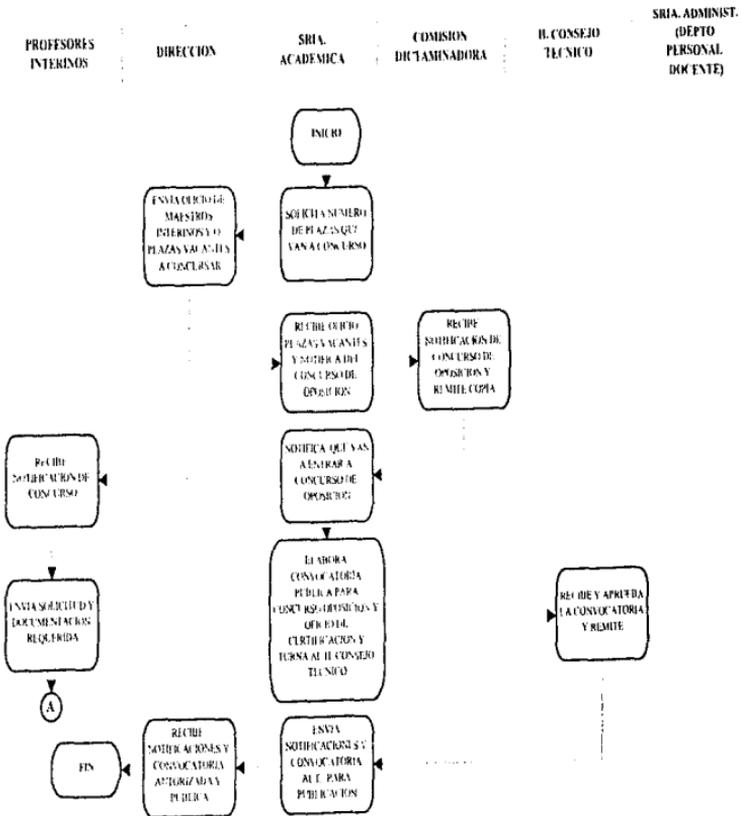
DICTAMINAN  
EMPLEAN  
RESULTADO

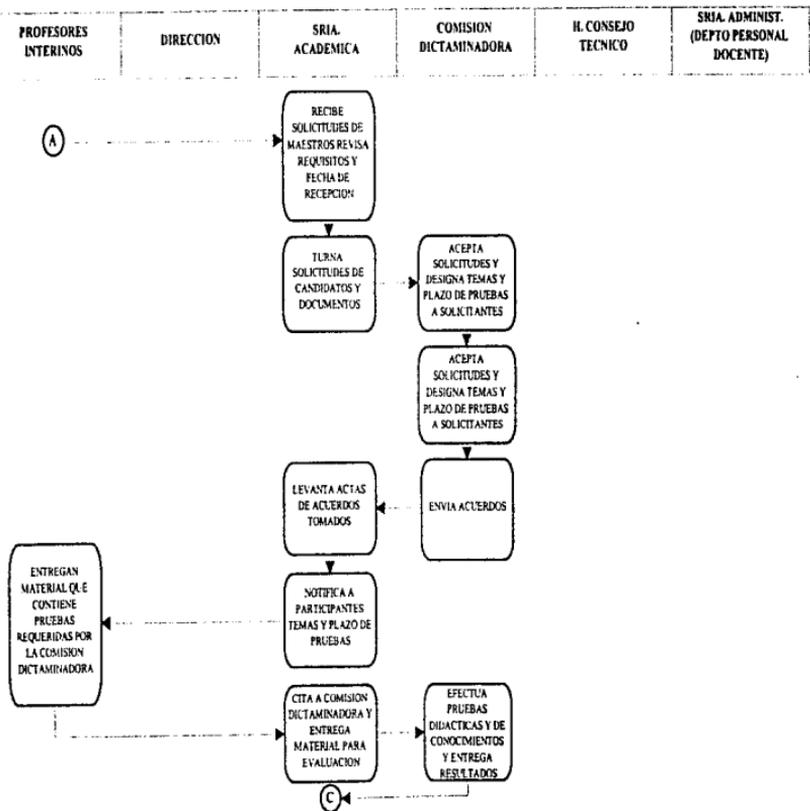
REGISTRA  
RESULTADO Y  
SINODO PRESENTE

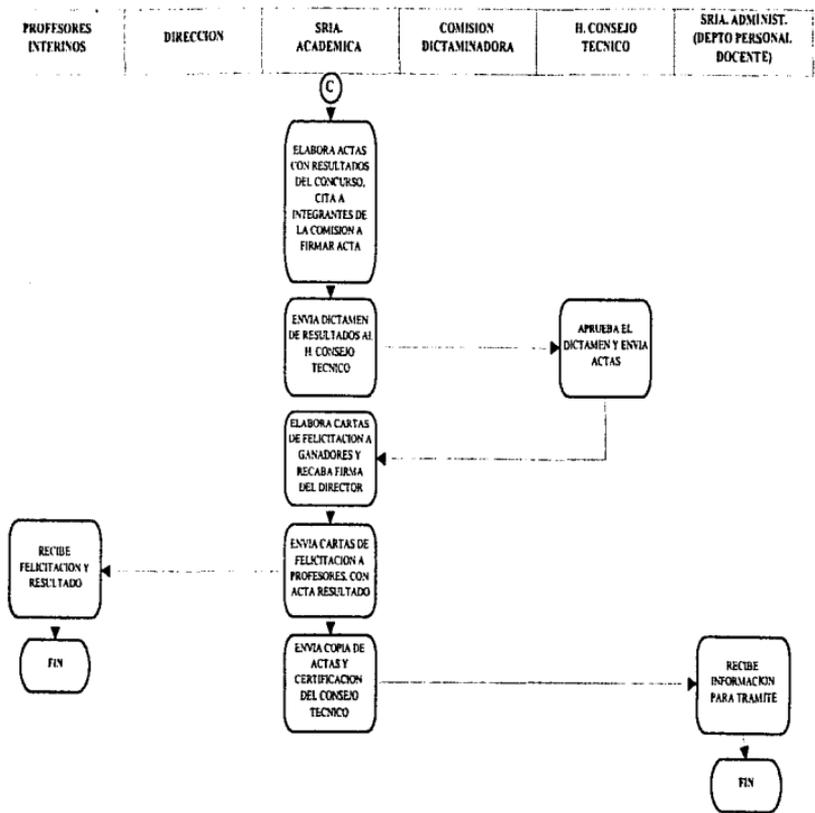
FIN

Planteamiento del Problema y Propuesta de Solución

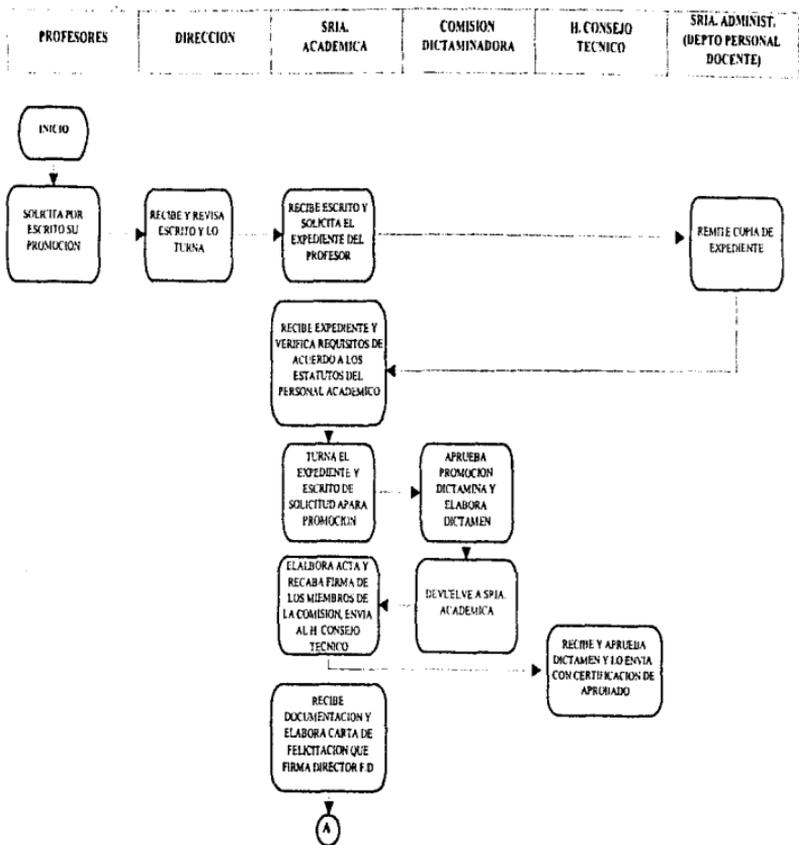
2.6.2.4 Diagrama de exámenes de oposición abiertos.

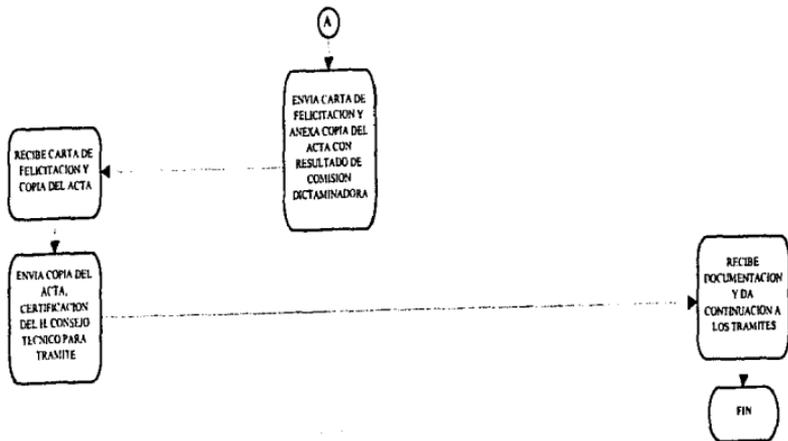




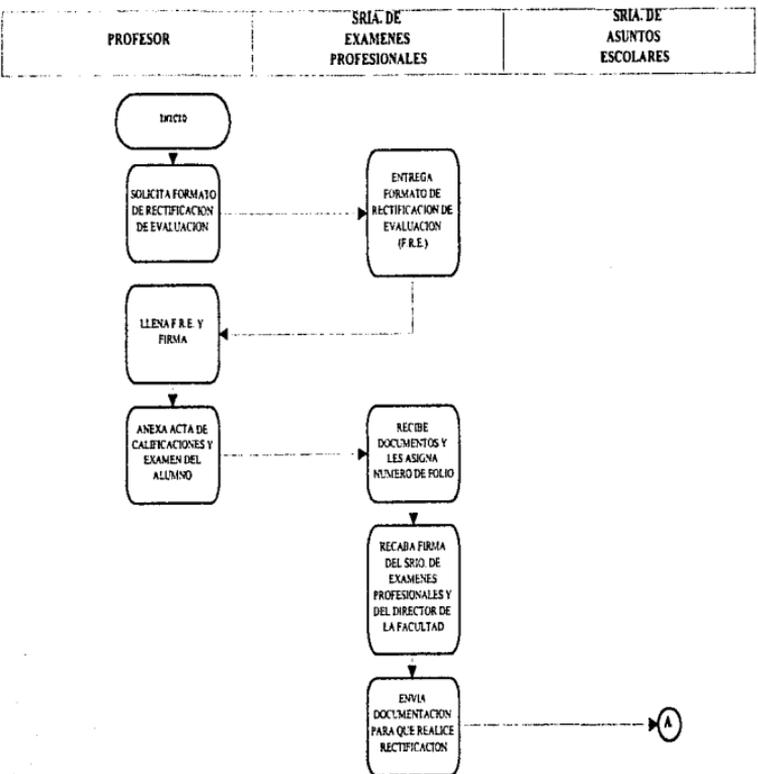


## 2.6.2.5 Diagrama de exámenes de oposición cerrados.





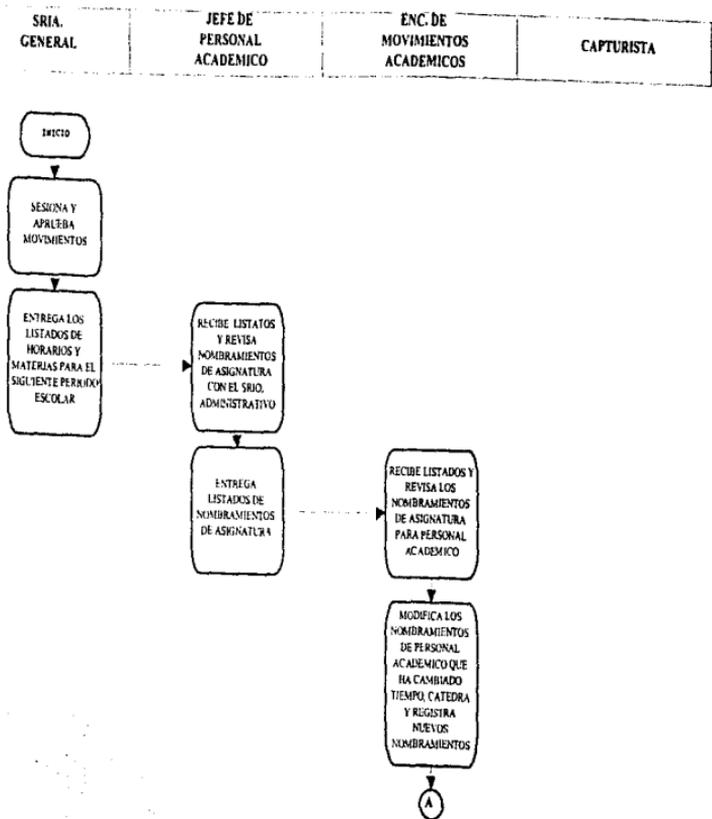
## 2.6.2.6 Diagrama de rectificaci3n de calificaciones.



PROFESOR	SERIA DE EXAMENES PROFESIONALES	SERIA DE ASUNTOS ESCOLARES
----------	---------------------------------	----------------------------



## 2.6.2.7 Diagrama de generación de contratos.



SRIA. GENERAL	JEFE DE PERSONAL ACADEMICO	ENC. DE MOVIMIENTOS ACADEMICOS	CAPTURISTA
------------------	----------------------------------	--------------------------------------	------------



## 2.7

# DEFINICION DE REQUERIMIENTOS

Una de las principales necesidades es poder compartir información y contar con una interfaz sencilla y amigable.

Este subcapítulo nos ayuda a poder llevar a cabo una adecuada selección de los requerimientos físicos y funcionales de un sistema de información eficiente que permita satisfacer las necesidades planteadas en este proyecto.

En la estrategia de solución se planteo la manera de como poder solucionar el problema; y mas aún se analizaron varias alternativas para obtener una solución efectiva, en base a dicho análisis se tomo la decisión de desarrollar el sistema en el manejador de base de datos Sybase y utilizar Powerbuilder como interfaz con el usuario.

Esta solución se determino a partir de proponer la arquitectura cliente/servidor como espina dorsal de dicha alternativa.

Las estadísticas analizadas del desempeño de Sybase comparadas anteriormente con Oracle e Informix principalmente demuestran que se trata de un producto con un alto nivel de ejecución y seguridad, aspectos importantes en la elección de un producto.

La metodología estructurada utilizada para el desarrollo del sistema proveerá de un alto grado de eficiencia y fiabilidad, aunado a las ventajas de trabajar con un producto de desarrollo visual como lo es PowerBuilder reduciendo enormemente el tiempo de desarrollo y dejando mas espacio para el análisis y diseño. Estas dos ultimas etapas solo se presentaran durante el inicio de vida del sistema y posteriormente se tomara todo el tiempo restante para poder desarrollar y mantener lo ya existente, por lo cual a mediano plazo se verán reflejados los beneficios.

Para el usuario promedio contar con una interface gráfica y que permita interactuar con los diferentes productos existentes en el mercado para la misma plataforma, representa un ahorro enorme en el desarrollo de sus actividades.

Los sistemas desarrollados contarán con dos niveles de seguridad. El primero respecto a la integridad de los datos de la cual se encarga el servidor de la base de datos y controlada por el Administrador de Sybase, en este nivel se contempla la posible pérdida y recuperación de los datos durante una eventual caída del sistema o bien cuando algún proceso no fue concluido normalmente.

El segundo se refiere a la seguridad correspondiente a cada usuario y contempla el uso de claves de usuario, contraseñas y derechos sobre cada objeto de la base de datos. Evitando así un mal uso o empleo de la información contenida. Este nivel de seguridad debe estar reflejado en las actividades propias de cada área, por lo que el sistema en general no tendrá disponibles todas las opciones del sistema y solo hasta que el usuario se identifique y proporcione su contraseña le serán asignados los derechos sobre cada modulo del sistema y sobre alguna información en particular.

Este ultimo nivel será definido en base al diagrama de funcionalidad e implementado durante la generación de las aplicaciones gráficas.

Por otra parte los requerimientos necesarios para poder implementar el sistema a los usuarios son:

- Una conexión a red.
- Una computadora personal con 8 Mb de memoria RAM.
- Ambiente gráfico windows versión 3.x o superior.
- Contar con mouse (preferentemente).

Cabe aclarar que los equipos con que se cuenta actualmente superan los requerimientos óptimos necesarios para poder utilizar el sistema.

Se pretende conjuntar la actividad de seminario con la de titulación, ya que actualmente son dos las personas encargadas de llevar a cabo dichas actividades; y durante el análisis se observó que no es conveniente el dividir un proceso tan relacionado como éste.

Así mismo se llevará a cabo un proceso de capacitación en cuanto al uso y cambios en el control de sus actividades por medio del sistema, indicando la importancia de la veracidad y responsabilidad con que se usen los datos a los que se tiene acceso.

Se contará con herramientas visuales de apoyo a la administración de Sybase tal como SQL Server Manager, Erwin para el diseño de los diagramas entidad - relación y generación de bases de datos y tablas en Sybase. Cada uno de ellos proporcionará una manera más ágil para poder desarrollar las aplicaciones del sistema.

SQL Server Manager, permitirá administrar la base de datos Sybase en aspectos tales como tablas, claves y atributos de usuarios, grupos de usuarios, dispositivos de almacenamiento, dispositivos de almacenamiento temporal, dispositivos de respaldo y seguridad.

Erwin es una herramienta CASE para el diseño de estructuras de bases de datos en ambientes cliente/servidor para diferentes manejadores de bases de datos..

Se han analizado por separado las características y requerimientos del hardware necesario para la ejecución satisfactoria de ambos productos; ambos requerimientos se tienen cubiertos con la infraestructura existente de la Facultad de Derecho, así que solo queda definir el desarrollo del sistema.

Los módulos con que constará el sistema de académicos serán los siguientes:

- Inasistencias del personal docente.
- Registro de tesis y examen profesional.
- Exámenes de oposición.
- Rectificación de calificaciones.
- Generación de contratos.

El módulo de inasistencias del personal docente deberá ser capaz de llevar el control de asistencia de cada profesor - asignatura - grupo, permitir consultar los horarios del semestre actual o bien de anteriores, contar con reportes de estadísticas por periodo, por profesor, por asignatura, etc.

Para el módulo de registro de tesis y examen profesional es necesario tomar en cuenta varios aspectos tales como que un tema de tesis no se debe de repetir y proporcionar una lista de los temas parecidos para que el usuario decida si se acepta la propuesta de tesis o no. Revisar si un alumno no ha registrado un tema de tesis anteriormente o bien si ya cumplió con todos los requisitos necesarios para propener un tema específico. Una vez registrado el tema el alumno regresara para que le sean asignados los sinodales y fecha correspondientes a su examen profesional. La asignación deberá ser automática, pero contemplara la posibilidad de hacerlo de manera manual para el personal autorizado.

En lo concerniente a exámenes de oposición se llevará el control de comisiones dictaminadoras por área, es decir, se integrarán comisiones revisoras que se encargarán de

asignar y evaluar algún tipo de trabajo a cada uno de los participantes del concurso. Así mismo se registrará el resultado y se emitirán los documentos necesarios para su certificación oficial.

La rectificación de calificaciones es el resultado del recurso de inconformidad al cual recurren los alumnos para pedir la revisión de una evaluación asignada en un período ordinario o en un examen extraordinario. También el motivo puede ser una calificación asentada erróneamente por lo cual el profesor solicita la rectificación.

Finalmente en el módulo de generación de contratos se elaborarán los contratos que se envían a la Dirección General de Personal por cada una de las clases impartidas. Cabe mencionar que también se generan los contratos de los profesores que tienen plazas de medio tiempo y tiempo completo.

**CAPITULO**

**3**

**DISEÑO E  
IMPLEMENTACIÓN  
DEL SISTEMA**

## 3.1

# MODELO AMBIENTAL

Este modelo define las interfaces entre el sistema y el resto del universo, es decir, modela el exterior del sistema. Consta de tres componentes:

- La declaración de propósitos que se trata de una declaración textual breve y concisa del propósito del sistema.
- El diagrama de contexto que es un caso especial del diagrama de flujo de datos, en donde en una sola burbuja se representa todo el sistema.
- La lista de acontecimientos que es una lista narrativa de los "estímulos" que ocurren en el mundo exterior a los cuales el sistema debe responder.

### *3.1.1 Declaración de propósitos.*

“El propósito del sistema de información docente es lograr la integración de la información académica, buscando eliminar la información redundante y compartir los recursos de red, programas y bases de datos, para simplificar actividades y agilizar trámites.”

### 3.1.2 Diagrama de Contexto.



### ***3.1.3 Lista de Acontecimientos.***

La mayoría están dirigidos por flujo. Los acontecimientos se listan a continuación; los temporales se marcan con una "T" luego de su descripción.

#### **3.1.3.1 Inasistencias del personal docente.**

El registro diario de inasistencia se realiza en base a la asignación del profesor, es decir, medio tiempo, tiempo completo o seminario, o bien en exámenes finales, exámenes profesionales, periodos de exámenes extraordinarios o acuerdo de pasantes.

Se contempla además la captura de inasistencias justificadas que están respaldadas por un documento.

Registan los cambios de profesores que se den a lo largo del semestre en la tabla de clases.

Se realizan calendarios de control de asistencia.

Impresión de tarjetones para control de asistencia.

Se generan reportes diarios, semanales, mensuales, semestrales, medio tiempo, tiempo completo o seminarios.

A continuación presentamos el listado de acontecimientos relacionados a este proceso.

- Mesa de Firmas solicita horarios del semestre correspondiente a Secretaría General (T).
- Mesa de Firmas solicita lista de profesores de medio tiempo, tiempo completo y ayudantes de seminario a Secretaría Administrativa (Departamento de Control Docente) (T).

- Mesa de Firmas solicita plantilla de profesores y fechas de exámenes extraordinarios (T).
- Mesa de Firmas solicita plantilla de profesores y fechas de exámenes por acuerdo de pasantes (T).
- Mesa de Firmas imprime tarjetones para control de asistencia en base a los horarios del semestre correspondiente (T).
- Mesa de Firmas imprime tarjetones para control de asistencia para profesores de medio tiempo, tiempo completo y seminarios (T).
- Mesa de Firmas registra inasistencia diaria de profesores con grupo durante el período de clases.
- Mesa de Firmas registra inasistencia diaria de profesores de medio tiempo y tiempo completo durante todo el semestre.
- Mesa de Firmas registra inasistencia diaria de ayudantes de seminario durante todo el semestre.
- Mesa de Firmas registra inasistencia de profesores con grupo durante el período de exámenes finales.
- Mesa de Firmas registra inasistencia de profesores con grupo durante el período de exámenes extraordinarios y acuerdo de pasantes.
- Mesa de Firmas registra los cambios de profesores que se dan durante el semestre correspondiente.
- Mesa de Firmas localiza profesores.
- Mesa de Firmas solicita impresión de reportes de asistencia por profesor semanales (T).

- Mesa de Firmas solicita impresión de reportes de asistencia por profesor mensuales (T).
- Mesa de Firmas solicita impresión de reportes de asistencia por profesor semestrales (T).
- Mesa de Firmas solicita impresión de reportes de asistencia de profesor diarios por asignatura (T).
- Mesa de Firmas solicita impresión de reportes de asistencia de profesor semanales por asignatura (T).
- Mesa de Firmas solicita impresión de reportes de asistencia de profesor mensuales por asignatura (T).
- Mesa de Firmas solicita impresión de reportes de asistencia de profesor semestrales por asignatura (T).
- Mesa de Firmas solicita impresión de reportes de asistencia de profesores de medio tiempo y tiempo completo diarios (T).
- Mesa de Firmas solicita impresión de reportes de asistencia de profesores de medio tiempo y tiempo completo semanales (T).
- Mesa de Firmas solicita impresión de reportes de asistencia de profesores de medio tiempo y tiempo completo mensuales (T).
- Mesa de Firmas solicita impresión de reportes de asistencia de profesores de medio tiempo y tiempo completo semestrales (T).
- Mesa de Firmas solicita impresión de reportes de asistencia de ayudantes de seminario diarios (T).
- Mesa de Firmas solicita impresión de reportes de asistencia de ayudantes de seminario semanales (T).

- Mesa de Firmas solicita impresión de reportes de asistencia de ayudantes de seminario mensuales (T).
- Mesa de Firmas solicita impresión de reportes de asistencia de ayudantes de seminario semestrales (T).
- Mesa de Firmas solicita impresión de reportes de asistencia de profesores en exámenes extraordinarios (T).
- Mesa de Firmas solicita impresión de reportes de asistencia de profesores en exámenes por acuerdo de pasantes (T).

### **3.1.3.2 Registro de tesis y examen profesional.**

En este punto se divide la lista de acontecimientos en dos partes, la primera propia del registro de tesis y la segunda de examen profesional.

La inscripción a seminario para registro de tesis se basa en la situación académica del alumno, es decir, se verifica que los alumnos hayan concluido todas las materias, que no tenga materias reprobadas, que por lo menos tenga el 90% de las materias aprobadas, que aún esté activa su inscripción en la Facultad.

Existe lo que se llama registro condicionado provisional donde el alumno tiene que concluir las materias pendientes en el siguiente período inmediato.

Cuando un alumno debe materias, el sistema no permitirá su inscripción.

Se realizarán diferentes tipos de búsquedas con el objetivo de detectar temas duplicados.

Control sobre temas cancelados, si se requiere modificar el nombre del tema o cambiar de provisional a definitivo, se asignará un nuevo número de folio.

Se verifica también, que haya correspondencia entre materia y tema de tesis.

Se debe controlar que no se inscriban dos veces los alumnos y se tiene que cancelar el registro anterior antes de procesar uno nuevo.

Se permite que un maestro tenga un mismo tema de tesis, lo que no es permitido es que dos maestros tengan a su cargo un solo tema de tesis.

El maestro asesor de tesis deberá ser profesor de la Facultad. El tema de tesis deberá ser autorizado por el Director del Seminario. El asesor de tesis deberá ser designado por el Director de Seminario. El tiempo límite de elaboración de tesis deberá ser 6 meses.

Los alumnos presentarán al Seminario informe sobre el desarrollo de la tesis.

Se deben generar reportes de alumnos inscritos, temas de tesis registrados y sinodales.

*En base a lo anterior el listado de acontecimientos es como sigue:*

- Alumno solicita director de tesis a Seminario.
- Alumno entrega tema y capitulado al director de tesis.
- Director de tesis analiza y aprueba tema y capitulado de tesis.
- Director de tesis emite oficio de aprobación de tema y capitulado de tesis.
- Seminario recibe tema de tesis y capitulado y emite visto bueno.
- Alumno solicita a Secretaría General registro de tema de tesis.
- Secretaría General verifica situación escolar del alumno.
- Secretaría General revisa que el tema y capitulado sean originales.
- Secretaría General registra tema de tesis.
- Secretaría General entrega al alumno comprobante de inscripción.

- Alumno solicita a Secretaría General modificaciones relacionadas a su tema de tesis.
- Secretaría General cancela registro de tesis a los alumnos que no hayan cumplido con los requisitos.

Para la segunda lista de acontecimientos se necesita que la propuesta del Jurado deberá ser designada por la Secretaría de Exámenes Profesionales. El Jurado deberá contener como mínimo 3 maestros especialistas en la materia de la tesis.

Los comprobantes del voto aprobatorio de la tesis, de la Biblioteca de la Facultad y la Central de no adecuado de libros deberán entregarse cuando menos un día antes de la celebración del examen profesional.

El alumno deberá confirmar la asistencia de sus sinodales, sino le será cancelado el examen. El alumno deberá presentarse media hora antes del examen y esperará a los sinodales.

*Los puntos siguientes forman el listado de acontecimientos relativos al proceso de exámenes profesionales.*

- Alumno solicita fecha de examen profesional.
- Secretaría de Exámenes Profesionales entrega solicitud de fecha de examen al alumno.
- Secretaría de Exámenes Profesionales solicita revisión de estudios al alumno.
- Alumno tramita la constancia de revisión de estudios.
- Alumno llena solicitud de examen profesional.
- Alumno entrega solicitud de examen profesional, constancia de revisión de estudios y ejemplares del trabajo de tesis a Secretaría de Exámenes Profesionales.

- **Secretaría de Exámenes Profesionales** recibe, verifica y sella constancia de revisión de estudios y ejemplares de tesis.
- **Secretaría de Exámenes Profesionales** verifica promedio del alumno para dictaminar procedencia a mención honorífica.
- **Secretaría de Exámenes Profesionales** asigna fecha de examen profesional y sinodales.
- **Secretaría de Exámenes Profesionales** emite citatorio al sínodo.
- Alumno se encarga de llevar citatorios al sínodo y de recabar firmas.
- Alumno lleva citatorios firmados a **Secretaría de Exámenes Profesionales**.
- **Secretaría de Exámenes Profesionales** recibe documentación y entrega copias de documentación.
- **Secretaría de Exámenes Profesionales** solicita al alumno realice trámites de asignación de aula.
- Alumno solicita aula para realizar examen profesional a **Secretaría de asuntos estudiantiles**.
- Alumno entrega copia de comprobante de asignación de aula a la **Secretaría de Exámenes Profesionales**.
- **Secretaría de Exámenes Profesionales** recibe comprobante y publica fecha y lugar del examen profesional.
- **Secretaría de Exámenes Profesionales** envía actas y constancias al sínodo.
- Sínodo dictamina y envía resultado a **Secretaría de Exámenes Profesionales**.
- **Secretaría de Exámenes Profesionales** registra el resultado del examen profesional.

- Secretaría de Exámenes Profesionales registra asistencia del sínodo.
- Secretaría de Exámenes Profesionales emite reportes mensuales de estadísticas de exámenes profesionales presentados (T).

### **3.1.3.3. Exámenes de oposición.**

Los profesores interinos con antigüedad mayor a 1 año en la Facultad deberán presentarse a los concursos de oposición.

Los profesores que no participen en el concurso o no sean seleccionados no tendrán derecho a que se les asigne grupo.

Solamente podrán participar aquellos profesores que tengan título de licenciado en Derecho, que demuestren aptitud para la docencia y que efectúen las pruebas que señala la Comisión y sean aprobadas por el Consejo Técnico.

Si un participante está inconforme del resultado podrá interponer inconformidad ante el Consejo Técnico, siendo la Secretaría General quien realizará el proceso hasta su término.

Los tipos de exámenes de oposición se dividen en dos categorías: abiertos y cerrados en cada uno de ellos se describe su listado de acontecimientos.

*Para el examen de oposición abierto:*

- Secretaría Académica solicita al Departamento de personal docente el número de plaza que va a concursar.
- Departamento de Personal Académico envía a Secretaría Académica oficio de los maestros interinos y/o plazas vacantes a concursar.

- Secretaría Académica recibe plazas vacantes y notifica a la Comisión Dictaminadora correspondiente que abrirá concurso de oposición en el área de su disciplina.
- La Comisión Dictaminadora recibe notificación de concursos de oposición.
- Secretaría Académica notifica a los maestros interinos que sus horas de clase van a entrar a concurso de oposición.
- Secretaría Académica elabora convocatoria pública para concurso y oficio de certificación para su publicación, la turna a Consejo Técnico.
- Consejo Técnico recibe y aprueba la convocatoria y envía a Secretaría Académica.
- Secretaría Académica envía las notificaciones y convocatorias autorizadas a la Dirección de Asuntos de Personal Académico para su publicación.
- Secretaría Académica recibe la solicitud de los maestros interesados en el concurso, revisa los requisitos que marca la convocatoria y las fechas de recepción.
- Secretaría Académica supervisa que los maestros interinos participen en el concurso.
- Secretaría Académica turna a la Comisión Dictaminadora correspondiente, las solicitudes de candidatos y los documentos presentados por los mismos.
- Comisión Dictaminadora acepta las solicitudes y designa a los concursantes temas y plazo de las pruebas correspondientes.
- Secretaría Académica levanta acta de los acuerdos tomados por la Comisión.
- Secretaría Académica notifica a los participantes temas y plazo de pruebas.

- **Secretaría Académica cita a la Comisión Dictaminadora y entrega material para la evaluación a cada miembro de la comisión.**
- **Comisión Dictaminadora efectúa pruebas didácticas y de conocimientos a los candidatos y entrega a la Secretaría Académica los resultados.**
- **Secretaría Académica elabora las actas con los resultados correspondientes al concurso (dictamen) y cita a los integrantes de la comisión para firmar actas.**
- **Secretaría Académica envía dictamen de resultados al H. Consejo Técnico de la Facultad para su aprobación.**
- **Consejo Técnico de la Facultad aprueba el dictamen de las comisiones y lo envía a Secretaría Académica.**
- **Secretaría Académica elabora cartas de felicitación a profesores que ganaron concurso y recaba firma del director (T).**
- **Secretaría Académica envía al profesor carta de felicitación acompañada de una copia del acta con el resultado de la comisión respectiva.**
- **Secretaría Académica envía copia de las actas y la certificación del Consejo Técnico a la Secretaría Administrativa de la Facultad para los trámites correspondientes.**

*Para examen de oposición cerrado.*

- **Profesor solicita por escrito a la Dirección de la Facultad su promoción por ser profesor definitivo.**
- **Dirección recibe, revisa escrito y lo turna a la Secretaría Académica.**
- **Secretaría Académica solicita al Departamento de Personal Académico de la Facultad el expediente del profesor.**

- Secretaría Académica verifica que los requisitos estén de acuerdo a los estatutos del personal académico.
- Secretaría Académica turna el expediente y la solicitud a la Comisión Dictaminadora para la evaluación de su promoción.
- Comisión Dictaminadora del personal académico aprueba promoción, determina pruebas y/o elabora dictamen y lo devuelve a la Secretaría Académica.
- Secretaría Académica elabora el acta y recaba firmas de la Comisión y lo envía al H. Consejo Técnico de la Facultad, entrega el expediente del profesor al Departamento de personal académico.
- Consejo Técnico recibe y aprueba dictamen y lo envía a la Secretaría Académica con la certificación de aprobado.
- Secretaría Académica recibe documentación y elabora carta de felicitación que firma el director de la Facultad (T).
- Secretaría Académica envía al profesor carta de felicitación y anexa una copia del acta con el resultado de la Comisión Dictaminadora respectiva.
- Profesor recibe carta de felicitación y copia del acta.
- Secretaría Académica envía copia del acta y certificación del H. Consejo Técnico a la Secretaría Administrativa de la Facultad para el trámite correspondiente.

#### **3.1.3.4 Rectificación de calificaciones.**

La solicitud de rectificación de calificación deberá ser tramitada por el profesor titular de la asignatura. No será atendida ninguna solicitud de rectificación de calificación hecha por alumnos.

La rectificación de calificación deberá basarse en las normas establecidas por la Legislación Universitaria vigente.

Para la rectificación de calificación de exámenes extraordinarios, la solicitud deberá ser firmada también por los dos profesores que efectuaron el examen.

A continuación se describe el listado de acontecimientos:

- Profesor pide formato de rectificación de evaluación en la Secretaría de Exámenes Profesionales.
- Secretaría de Exámenes Profesionales entrega solicitud al profesor.
- Profesor llena la solicitud, firma y entrega acta de calificaciones y examen del alumno a Secretaría de Exámenes Profesionales.
- Secretaría de Exámenes Profesionales recibe documentos y asigna número de folio.
- Secretaría de Exámenes Profesionales firma documentos, recaba firma de VO. BO. En solicitud del director de la Facultad.
- Secretaría de Exámenes Profesionales envía la documentación a la Secretaría de Asuntos Escolares para hacer la rectificación correspondiente.
- Secretaría de Asuntos Escolares recibe documentación y elabora acta de rectificación.
- Secretaría de Asuntos Escolares archiva documentación que entregará posteriormente a Dirección General de Administración Escolar para que realice el trámite de rectificación.
- Secretaría de Asuntos Escolares entrega al alumno, constancia de rectificación de calificación de examen.

### **3.1.3.5 Generación de contratos.**

Cada vez que inicia un nuevo semestre para los profesores de asignatura es necesario generarles un contrato para cada materia que imparten, para lo cual se sigue todo un procedimiento que lleva desde la generación de la plantilla del nuevo semestre hasta la sustitución de algún profesor por otro generando así, una baja y una nueva alta en una materia dada.

El kardex de control de personal académico, el listado de nombramientos y la lista original entregada por Consejo Técnico deberán contener la misma información.

El profesor de nuevo ingreso deberá entregar original de los documentos solicitados, con dos copias legibles.

No deberá proceder ningún movimiento de personal sin la aprobación del Director de la Facultad y el H. Consejo Técnico.

Todo movimiento de personal deberá tramitarse ante Dirección General de Personal, una vez que ha sido aprobado.

Este proceso se encarga de elaborar e imprimir los contratos a partir de los datos laborales y los horarios.

Es posible la cancelación de contratos cuando por alguna razón el profesor es dado de baja.

A continuación se presenta la lista de acontecimientos para este proceso.

- Consejo Técnico sesiona y aprueba nombramientos de profesores de asignatura con base en el Estatuto de Personal Académico.
- Consejo Técnico entrega listados de horarios y materias por impartir para el siguiente período escolar o acta de aprobación para los profesores de nuevo ingreso al Jefe de personal académico.
- Jefe de personal académico le solicita al profesor su documentación completa para formar su kardex.

- Jefe de personal académico modifica los nombramientos del personal académico que han cambiado su tiempo de cátedra y registra los nuevos nombramientos o movimientos:
  - a) Nuevo ingreso.
  - b) Por reingreso.
  - c) Por reanudación.
  - d) Por prórroga
  - e) Por aumento o disminución de horas.
  - f) Licencia con o sin sueldo.
  - g) Licencias por año sabático.
  - h) Por renuncia.
  - i) No reanudó licencia.
  - j) Término del contrato.
  - k) Por rescisión.
  - l) Por jubilación.
  - m) Defunción.
  - n) Otro nombramiento dentro de la Facultad o de la U.N.A.M.
- Director de la Facultad da Vo.Bo. en el caso de nuevo ingreso.
- Secretaría Académica tramita el movimiento de personal generado ante la Dirección General de Personal.
- Secretaría Académica solicita los siguientes documentos: minuta 8 con número de control, minuta 6 con credencial del profesor y minuta 7 con número de expediente del profesor a la Dirección General de Personal, para el caso de nuevo ingreso.

- **Jefe de personal académico** llena forma única para dar aviso de baja de personal académico incluyendo en forma breve motivo y proporciona su VO. BO. enviando éste a la Secretaría Administrativa.
- **Secretaría Administrativa** revisa y firma de VO. BO.; envía al director de la Facultad.
- **Director de la Facultad** autoriza la baja de nombramiento de personal académico y envía a Secretaría Académica.
- **Secretaría Académica** registra en kardex y envía la forma única indicando baja a la Dirección General de Personal.
- **Dirección General de Personal** recibe documentación y envía notificación de aceptación de documentos.
- **Secretaría Administrativa** archiva documentación en kardex.

## 3.2

# DIAGRAMA DE FLUJO DE DATOS

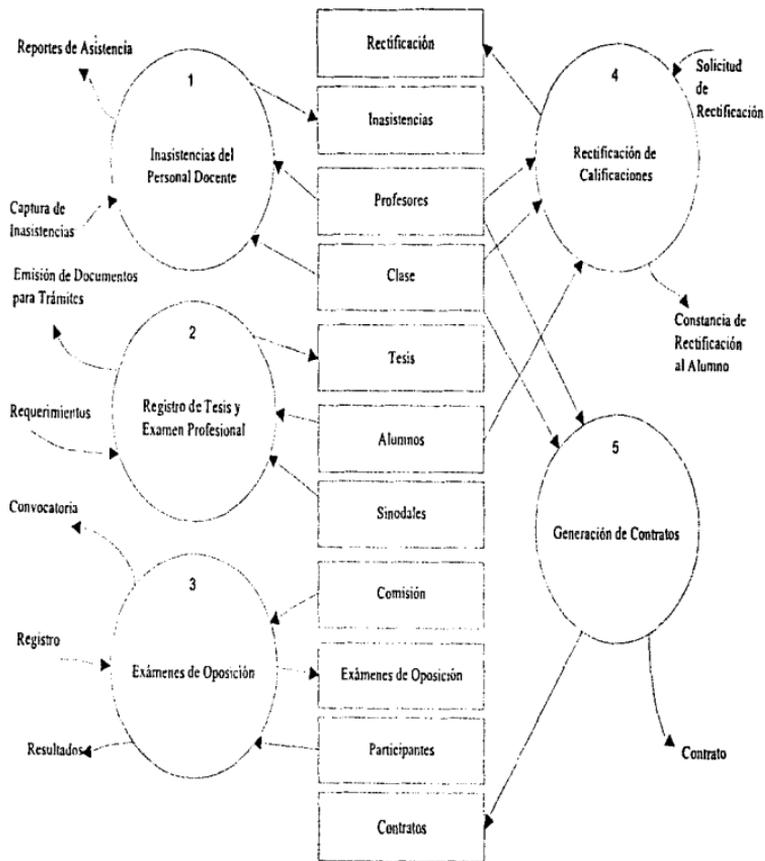
Hoy en día la información es un factor sumamente importante a nivel operativo y funcional en cualquier tipo de corporación.

Tanto los programas de aplicación como las técnicas empleadas para la correcta explotación de la información han estado evolucionando constantemente. Actualmente, existen diferentes técnicas o herramientas que nos ayudan a manipular los datos de una forma eficiente.

Una de las herramientas más frecuentemente utilizadas para el análisis estructurado es el diagrama de flujo de datos (DFD). Esta es una herramienta que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por conductos y tanques de almacenamiento de datos. Consta de cuatro componentes:

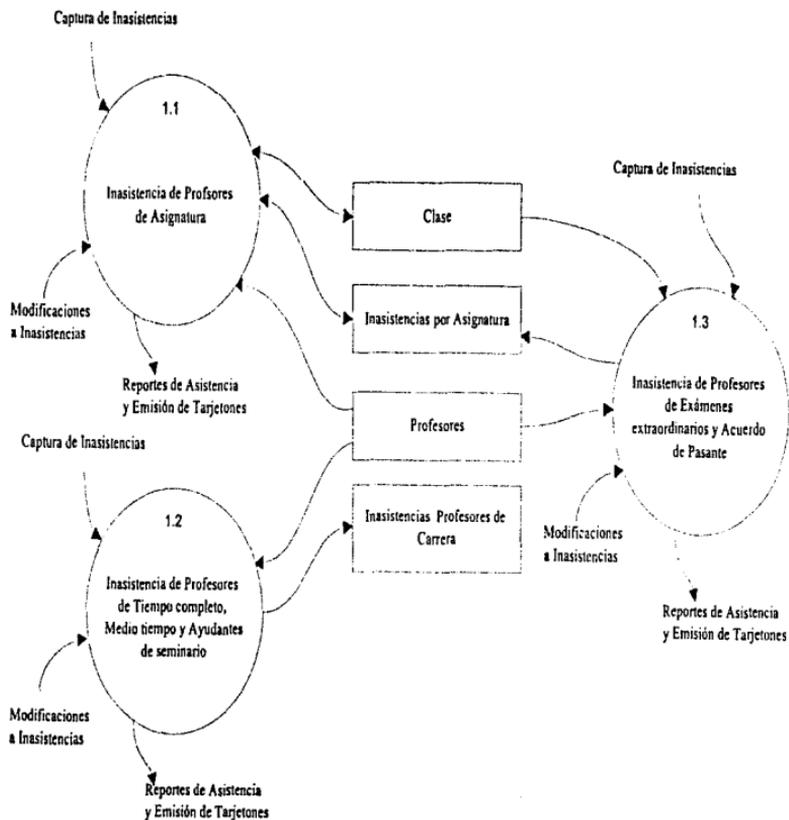
- El proceso que muestra cómo es que una o más entradas se transforman en salidas. Se representa por medio de un círculo.

- El flujo que se usa para describir el movimiento de bloques o paquetes de información de una parte del sistema a otra. Se representa con una flecha que entra o sale de un proceso.
- El almacén que se utiliza para modelar una colección de paquetes de datos en reposo. Se denota por un rectángulo con los vértices curvados.
- El terminador que representa entidades externas con las cuales el sistema se comunica. Se representa como un rectángulo.

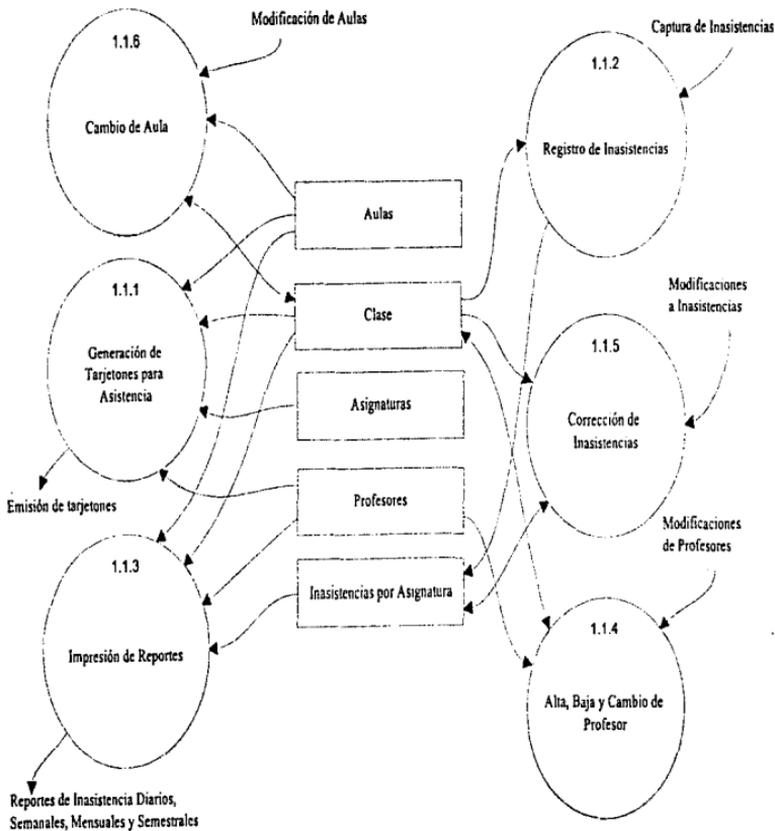


### 3.2.1 Nivel Superior.

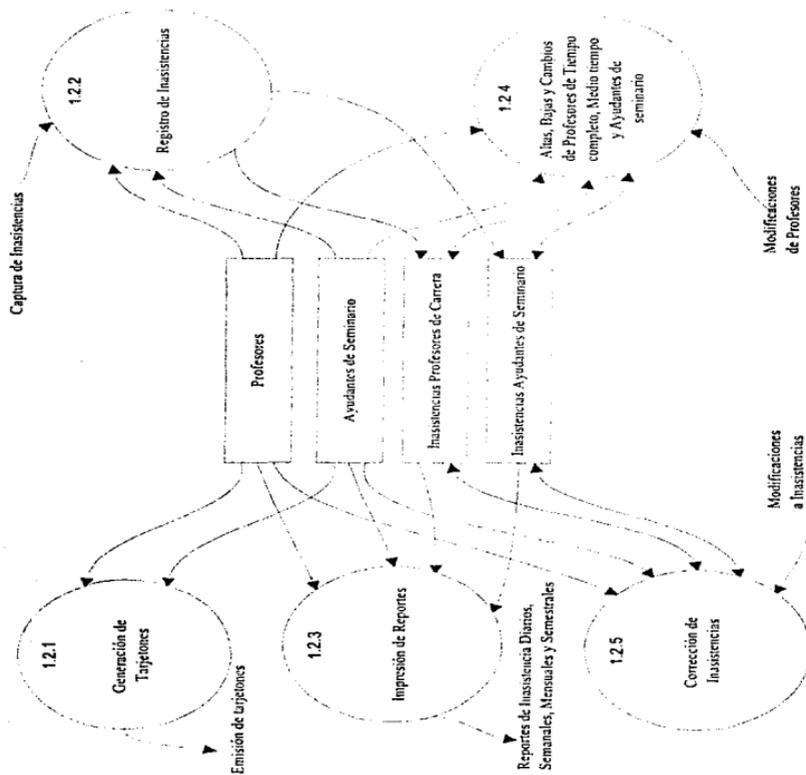
### 3.2.2 Inasistencias del Personal Docente.



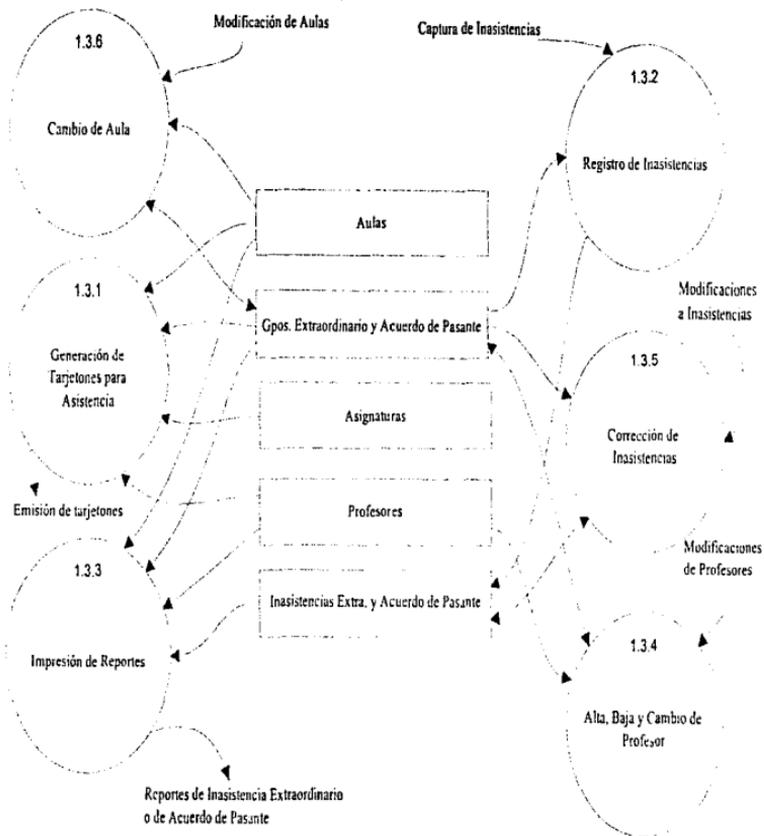
### 3.2.2.1 Inasistencia profesores de asignatura.



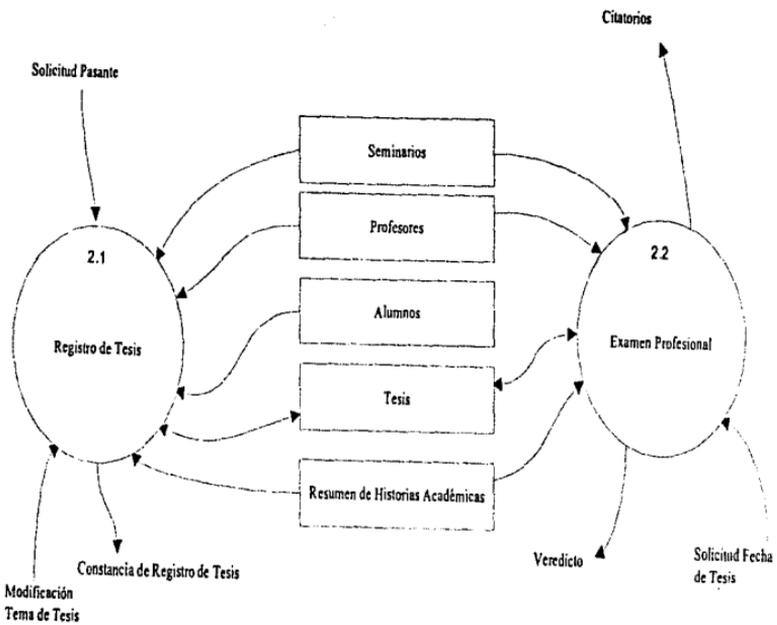
## 3.2.2.2 Inasistencia profesores de carrera.



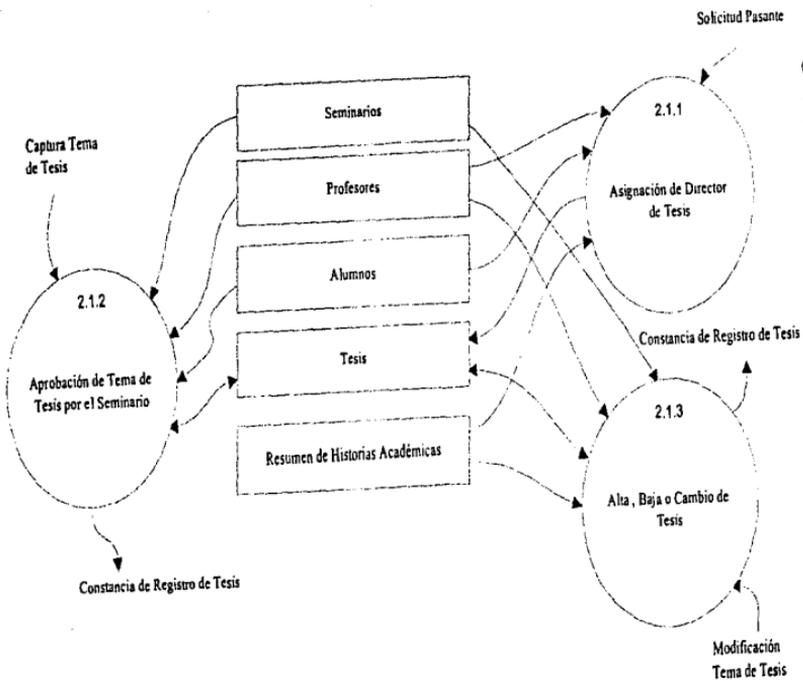
### 3.2.2.3 Inasistencia profesores de extraordinarios y acuerdo de pasantes.



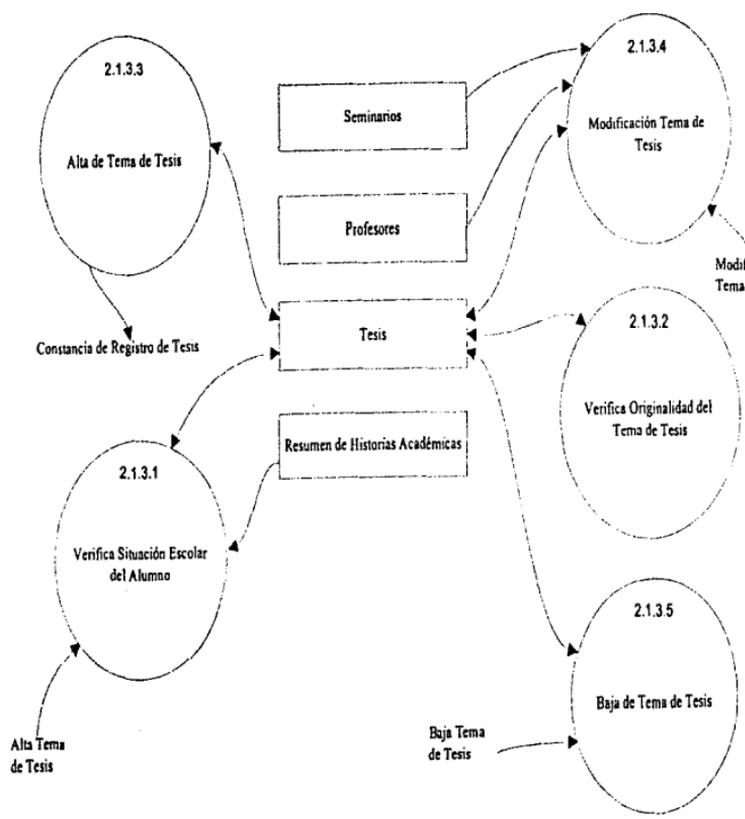
### 3.2.3 Registro de Tesis y Examen Profesional.



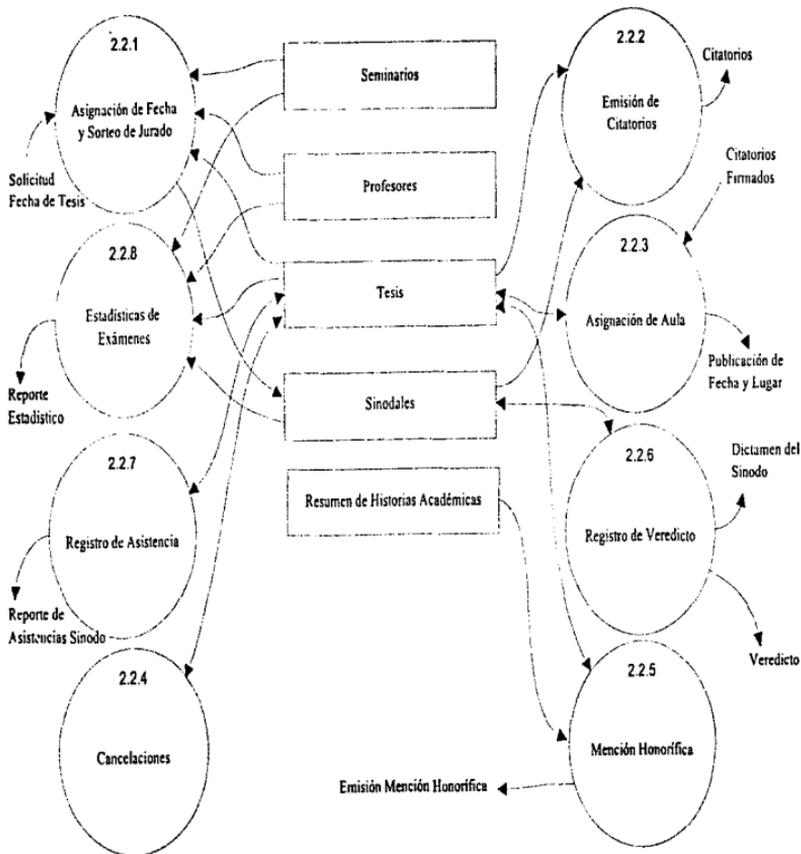
### 3.2.3.1 Registro de tesis.



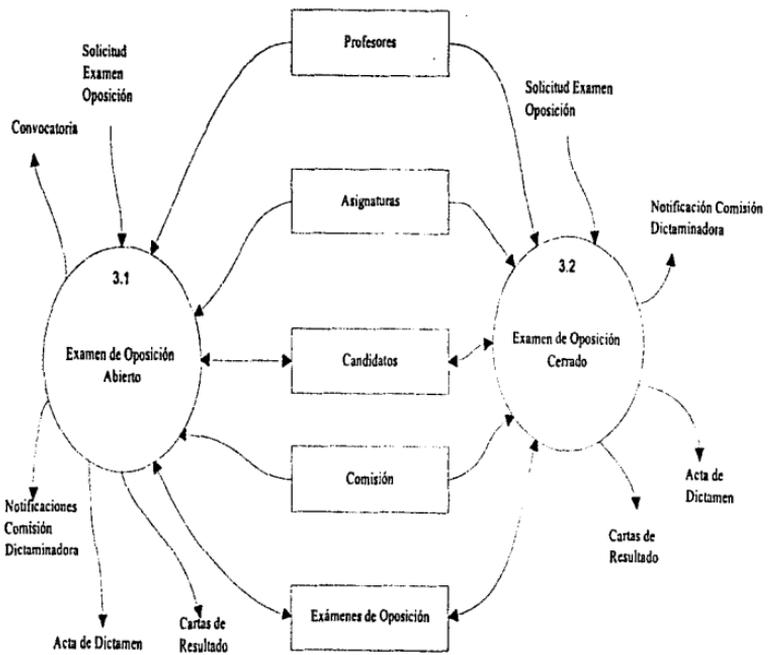
### 3.2.3.1.1 Alta, baja o cambio de tesis.



3.2.3.2 Examen profesional.

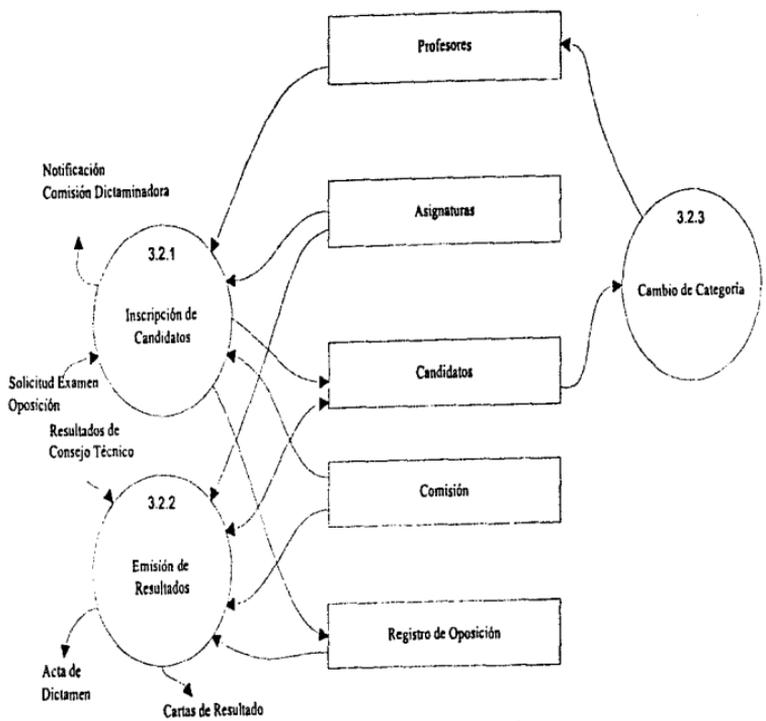


### 3.2.4 Exámenes de Oposición.

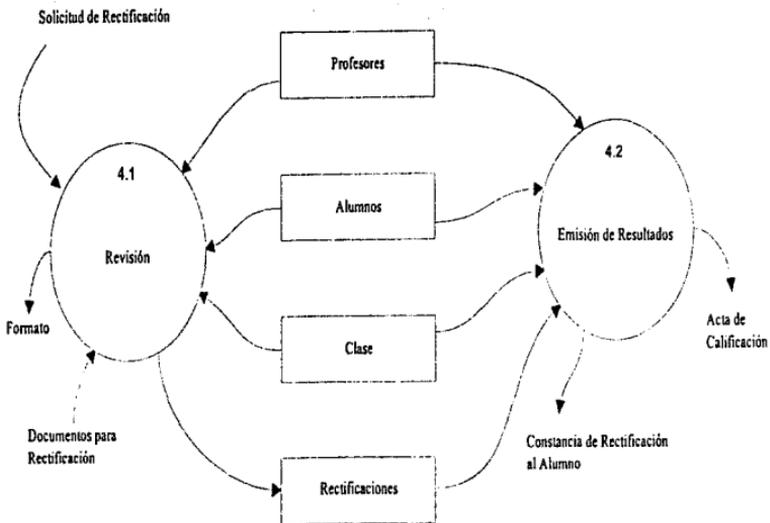




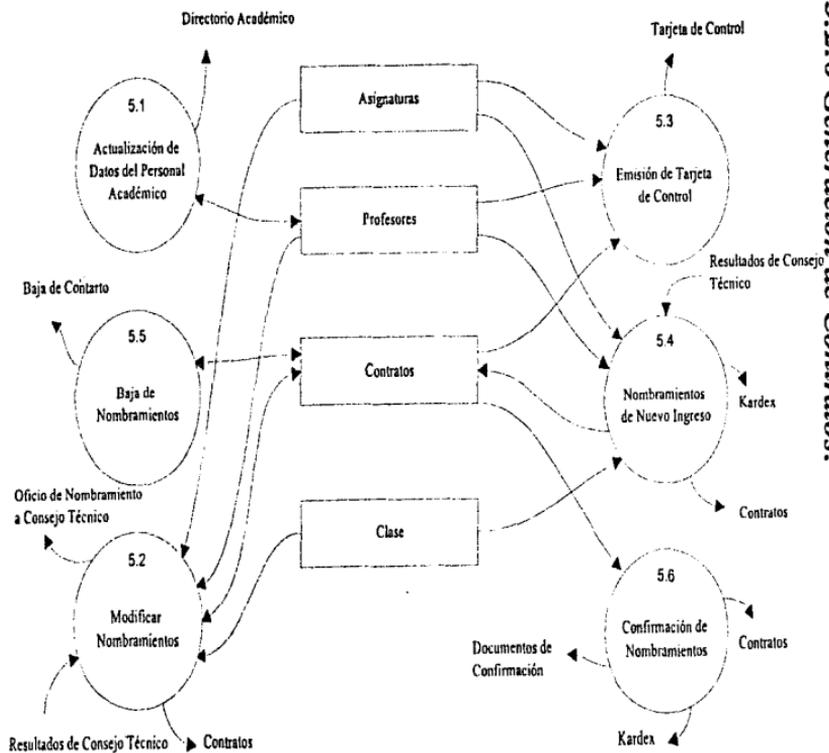
### 3.2.4.2 Exámenes de oposición cerrado.



### 3.2.5 Rectificación de Calificaciones.



### 3.2.6 Generación de Contratos.



## 3.3

# DICCIONARIO DE DATOS

Utilizando la metodología de Edward Yourdon para el análisis estructurado los almacenes y flujos deberán ser descritos en una forma sencilla, pero detallada, para lograr una homogeneidad en las descripciones de los datos se hace uso de un conjunto de herramientas denominadas "Diccionario de Datos".

Estas herramientas fueron explicadas en el primer capítulo de "Conceptos" en la parte de "Metodologías de Diseño" por lo cual lo damos por entendido

CONCEPTO	DEFINICIÓN
acta de acuerdo con la comisión.	= *acuerdo que celebra la comisión de examen de oposición con la secretaría académica*  **
acta de calificación.	= *documento oficial donde se asienta la calificación que obtiene un alumno en el curso de una asignatura*

	<p>nombre plantel + clave plantel + periodo + nombre asignatura + clave asignatura + grupo + nombre profesor + folio rectificación + {número de cuenta + carrera + nombre del alumno + [{"MB"}, {"B"}, {"S"}] {"NA"} [{"NP"}];</p>
<b>acta de dictamen.</b>	<p>= "documento donde se asientan los resultados finales del concurso de oposición"</p> <p>nombre comisión + asignatura + fecha de dictamen + [rfe + nombre profesor + [{"vencedor"}   {"pto"}   {"no apto"}   {"deserto"}];</p>
<b>alta tema de tesis.</b>	<p>= *registro de algún título de tesis*</p> <p>**</p>
<b>alumno.</b>	<p>= *persona que esta inscrita en la licenciatura de derecho de esta facultad*</p> <p>@no_de_cuenta + nombre_alumno + plantel + carrera + año_de_ingreso + nacionalidad + causa_de_ingreso + causa_exalumno + sexo + fecha_de_nacimiento + fecha_de_movimiento + marca_de_inscrito + acasa_calle_y_numero + acasa_colonia + acasa_delegación_o_población + acasa_ciudad + acasa_estado + acasa_cp + acasa_teléfono + atrabajo_calle_y_número + atrabajo_colonia + atrabajo_delegación_o_población + atrabajo_ciudad + atrabajo_estado + atrabajo_cp + atrabajo_teléfono + ateléfono_celular + acorreo_electrónico</p>

<b>ALUMNOS.</b>	= {alumno}
<b>asignatura.</b>	= *almacén de materias válidas para el plan de estudios de la facultad de derecho*  @clave_de_asignatura + nombre_de_la_asignatura + créditos + horas_u_la_semana + semestre + área + clave_de_plantel + plan + división
<b>ASIGNATURAS.</b>	= {asignatura}
<b>aula.</b>	= *salón asignado a la impartición de clases de las asignaturas del plan de estudios*  @clave_aula + nombre del aula + capacidad
<b>AULAS.</b>	= {aula}
<b>aviso a comisión.</b>	= *se le informa a una comisión de examen de oposición sobre un próximo concurso*  **
<b>baja de contrato.</b>	= *documento oficial firmado por el director de la facultad donde se da de baja a un profesor explicando brevemente la causa y se remite a la dirección general de personal*  **
<b>baja tema de tesis.</b>	= *eliminación de algún título de tesis*  **

<b>captura de inasistencias.</b>	= *registro de la falta de profesor*  **
<b>captura tema de tesis.</b>	= *registro del título de tesis que sobre el cual el alumno desea trabajar*  **
<b>carta de resultado.</b>	= *documento que se le entrega al o a los profesores que participaron en el concurso de oposición*  **
<b>citatorios firmados.</b>	= *contiene la misma información que citatorios salvo que este además tiene la rubrica del profesor*  **
<b>citatorios.</b>	= *solicitud de presencia hacia un profesor para participar como sinodo en un examen profesional en una fecha determinada*  folio tesis + nombre sustentante + número de cuenta + tema de tesis + fecha de examen + horario + lugar + {sinodales}
<b>clase.</b>	= *almacén de grupos de un semestre que se le asigna a un profesor*  @rfc + @clave_de_grupo + @fecha_inicio +  fecha_fin
<b>CLASES.</b>	= {clase}

<b>comisión.</b>	=*grupo de profesores especialistas que evalúan y dictaminan el resultado de un profesor de asignatura no definitivo en su examen de oposición*  @clave_de_comisión + nombre_de_comisión
<b>comisionado.</b>	= *profesor que forma parte de una comisión dictaminadora*  @rfe + @clave_de_comisión
<b>COMISIONADOS.</b>	= {comisionado}
<b>COMISIONES.</b>	= {comisión}
<b>constancia de rectificación al alumno.</b>	= *documento que ampara al alumno sobre una rectificación de calificación*  **
<b>constancia de registro de tesis.</b>	= *documento emitido al alumno después de ser aprobado su tema de tesis por parte de secretaría general*  nombre de seminario + nombre de director de seminario + director de tesis + tema de tesis + nombre del alumno
<b>contrato.</b>	= *documento oficial convenido entre el profesor y la universidad para el pago de honorarios*  @rfe + @clave_de_categoria + @clave_de_grupo + código_programático + no_de_registro_de_plaza + personal + movimiento + causa +

	<p>tipo_de_profesor_en_contrato + no_de_relación +  no_de_horas_teóricas + no_de_horas_prácticas +  fecha_de_inicio + fecha_de_fin + sustituye_a +  observaciones_en_contrato</p>
<b>CONTRATOS.</b>	= {contrato}
<b>contratos.</b>	<p>= *reporte impreso del contrato que se genera a los profesores cada periodo por las clases impartidas*</p> <p>{rfc + clave_de_categoria + clave_de_grupo +  código_programático + no_de_registro_de_plaza +  personal + movimiento + causa +  tipo_de_profesor_en_contrato + no_de_relación +  no_de_horas_teóricas + no_de_horas_prácticas +  fecha_de_inicio + fecha_de_fin + sustituye_a +  observaciones_en_contrato}</p>
<b>convocatoria.</b>	<p>= *documento público por medio del cual se informa a los profesores que habrá concurso de oposición en la asignatura determinada*</p> <p>••</p>
<b>dictamen del sínodo.</b>	<p>= *calificación que emite el sínodo sobre el examen profesional*</p> <p>••</p>
<b>directorio académico.</b>	<p>= *información académica y personal de los profesores que forman parte de la planta docente de la Facultad de Derecho*</p>

	**
<b>documentos a comisión dicraminadora.</b>	= *documentos que se turnan a la comisión dictaminadora para la promoción de un profesor . estos documentos son el expediente y la solicitud por escrito de un profesor *
	**
<b>documentos de confirmación.</b>	= *oficio firmado por un profesor para confirmar su nombramiento*
	**
<b>documentos para rectificación.</b>	= *documentos varios que se requieren para realizar una rectificación: formato lleno, acta de calificaciones y exámenes del alumno*
	**
<b>emisión de documentos para trámites.</b>	= *documentos que emiten las instancias correspondientes para que el alumno realice su registro de tesis y examen profesional*
	**
<b>emisión mención honorífica.</b>	= *reconocimiento al titulado por su gran desempeño en la carrera y examen profesional*
	**
<b>examen de oposición.</b>	= *convocatoria de inscripción de profesores de asignatura no definitivos a examen de oposición de alguna asignatura*

	@periodo + @clave_de_asignatura + clave_de_comision + fecha_de_convocatoria + fecha_de_dictamen + número_de_plazas
<b>EXAMENES DE OPOSICION.</b>	= {examen de oposición}
<b>formato.</b>	= *hoja que se entrega al profesor que necesita realizar una rectificación*  **
<b>historial académico.</b>	= *es el reporte que refleja la situación académica del alumno emitido por la secretaría de asuntos escolares*  nombre del alumno + número de cuenta + plantel + clave plantel + año de ingreso + carrera + clave carrera + periodo + créditos obligatorios + créditos optativos + total de créditos + asignaturas aprobadas + asignaturas no aprobadas + total de asignaturas cursadas + promedio + {clave plantel + clave asignatura + créditos asignatura + nombre asignatura + calificación + tipo de examen + periodo + folio acta + grupo + número ordinario + número extraordinario}
<b>inasistencia.</b>	= *almacén de control de faltas del personal docente*  @rfc + @clave_de_grupo + @fecha_de_falta + tipo_de_falta
<b>INASISTENCIAS.</b>	= {inasistencia}

<p><b>kardex</b></p>	<p>= *documento impreso donde se registran generales del profesor y contratos convenidos*</p> <p>fotografía + nombre profesor + rfc + lugar nacimiento + fecha nacimiento + nacionalidad + estado civil + dirección domicilio + teléfono domicilio + dirección oficina + teléfono oficina + título profesional + grados académicos + cédula profesional + primer nombramiento + número expediente + {año + periodo + categoría + número horas + sueldo + fecha alta contrato + fecha límite contrato + fecha baja contrato + fecha inicio licencia + fecha fin licencia + ["licencia con sueldo" ] "licencia sin sueldo"} + horario + [{"asignatura " ] "seminario"} + grupo + aula}</p>
<p><b>listado de horarios.</b></p>	<p>= *reporte de horarios por plan y por semestre válidos para las reinscripciones , enviados a la secretaría de asuntos escolares*</p> <p>{plan + semestre + asignatura + grupo + profesor + horario + aula}</p>
<p><b>modificación de aulas.</b></p>	<p>= *cambio de aula de algún grupo*</p> <p>**</p>
<p><b>modificación tema de tesis.</b></p>	<p>= *cambio al título de tesis*</p> <p>**</p>
<p><b>modificaciones a inasistencias.</b></p>	<p>= *corrección a alguna inasistencia de profesor*</p>

	**
<b>modificaciones de profesores.</b>	= "cambio de profesor a una clase" **
<b>notificación dictaminadora.</b>	<b>comisión</b> = "documento donde se le informa a la comisión dictaminadora que se abrirá concurso de oposición en el área de su disciplina" **
<b>notificación del tema.</b>	= "documento por medio del cual se le informa al participante de un examen de oposición el trabajo a desarrollar" **
<b>número de plazas.</b>	= "el total de plazas que entrarán a concurso en un examen de oposición" **
<b>oficio de nombramientos a consejo técnico.</b>	= "lista de profesores candidatos a ser contratados para que en sesión de consejo técnico apruebe nombramientos" **
<b>profesor.</b>	= "almacén de catedráticos que imparten alguna asignatura en un semestre dado"  @rfc + nombre + apellido_paterno + apellido_materno + grado + nacionalidad + sexo +

	estado_civil + fecha_de_nacimiento + cédula_profesional + profesor_activo + tipo_de_profesor + observaciones + casa_calle_y_número + casa_colonia + casa_delegación_o_población + casa_ciudad + casa_estado + casa_cp + casa_teléfono + trabajo_calle_y_número + trabajo_colonia + trabajo_delegación_o_población + trabajo_ciudad + trabajo_estado + trabajo_cp + trabajo_teléfono + telefono_celular + correo_electrónico + fecha_ingreso
<b>PROFESORES.</b>	= {profesor}
<b>publicación de fecha y lugar.</b>	= *información al público del lugar y fecha del examen profesional*  **
<b>rectificación.</b>	= *actualización de la calificación asentada en actas*  @rfc + @clave_de_grupo + @no_de_cuenta + no_de_folio + folio_de_acta + evaluación_pasada + evaluación_presente + renglón_del_acta
<b>RECTIFICACIONES.</b>	= {rectificación}
<b>registro.</b>	= *inscripción para participar en un examen de oposición*  **
<b>reporte de asistencias sínodo.</b>	= *listado de asistencia de los profesores que

	formaron el sínodo en examen profesional*  **
<b>reporte de situación contractual.</b>	= *diversos tipos de listados con los cuales se pretende sacar estadísticas por medio de los contratos de los profesores*  **
<b>reporte estadístico.</b>	= *diversos tipos de reportes que genera el sistema de exámenes profesionales, por ejemplo: no. de tesis por seminario, no. de tesis por profesor, alumnos con mención honorífica, etc.*  **
<b>reportes de asistencia.</b>	= *listado de faltas de profesores que se emite periódicamente -diario, semanal, mensual, semestral y anual- por parte del departamento de mesa de firmas y es enviado al secretario general*  {clave asignatura + nombre asignatura + grupo + nombre profesor + número de días de clase + número de clases no impartidas + porcentaje de asistencia}
<b>reportes de inasistencia diarios, semanales, *mensuales y semestrales.</b>	= *listados de porcentaje de asistencia de profesores por asignatura según el periodo*  **
<b>reportes de inasistencia extraordinarios o de acuerdo de pasantes.</b>	= *listado de asistencia de profesores por asignatura en exámenes extraordinario o acuerdo de pasante*  **

	<p>**</p>
<b>requerimientos.</b>	<p>= *documentos que necesita el alumno en conjunto para registra su tesis y poder realizar su examen profesional*</p> <p>**</p>
<b>resultados de consejo técnico.</b>	<p>= *documento generado por el h. consejo técnico donde se aprueban nombramientos de profesores*</p> <p>**</p>
<b>resultados.</b>	<p>= *es el dictamen final de los exámenes de oposición emitido por las comisiones dictaminadoras, enviado a secretaria académica*</p> <p>["vencedor"   "apto"   "no apto"   "desertó"]</p>
<b>resumen de historia académica.</b>	<p>= *expediente del alumno donde está asentada la evaluación de las asignaturas que ha cursado el alumno*</p> <p>@no_de_cuenta + no_mh + no_b + no_s + no_scis + no_siete + no_ocho + no_nueve + no_diez + no_na + no_np no_revalidadas + no_acreditadas + no_aprobadas_cn_ordinario + no_aprobadas_cn_extraordinario + no_reprobadas_ordinarios + no_reprobadas_extraordinario + no_creditos_obligatorios_acumulados + no_creditos_optativos_acumulados + no_covalidadas</p>

<b>RESUMEN DE HISTORIAS ACADEMICAS.</b>	= {resumen de historia académica}
<b>seminario.</b>	= *área específica del conocimiento del derecho donde se generan tesis relacionadas con el área*  @clave_seminario + nombre_seminario + director
<b>SEMINARIOS.</b>	= {seminario}
<b>sinodo.</b>	= *profesor que forma parte de un jurado que evalúa a un alumno en su examen profesional*  @no_registro + @ rfc + función + asistencia
<b>SINODOS.</b>	= {sinodo}
<b>solicitud de examen de oposición.</b>	= *solicitudes que envían los profesores interesados en llevar a cabo el examen de oposición*  rfc + nombre profesor + clave asignatura + nombre asignatura
<b>solicitud de rectificación.</b>	= *hoja que llena el profesor como requisito para llevar a cabo una rectificación a calificación del alumno*  numero de cuenta + asignatura + grupo + nombre del profesor + periodo + calificación correcta
<b>solicitud fecha de tesis.</b>	= *fecha para examen profesional que solicita el tesista una vez terminado el trabajo escrito*  **

<b>solicitud pasante.</b>	= "petición del alumno para realizar su tesis"  **
<b>tarjeta de control.</b>	= *documento impreso donde se asientan algunos datos del profesor contratado para su reconocimiento y control*  {rfc + nombre profesor + clave asignatura + nombre asignatura + horario}
<b>tarjetones.</b>	= *hoja impresa con datos del profesor donde éste firma los días que le corresponda dar clases*  {nombre profesor + nombre asignatura + grupo + horario + aula}
<b>temas.</b>	= *título del trabajo que el participante de un examen de oposición debe de desarrollar*  **
<b>TESIS.</b>	= {tesis}
<b>tesis.</b>	= *tema desarrollado por un alumno, que ha cumplido con el 100 % de créditos, como requisito indispensable para obtener su título*  @no_registro + rfc + no_de_cuenta + clave_seminario + fecha_de_registro + tema + aprobación_de_tema + fecha_examen + hora_examen + aula + mención_honorífica + resultado

**veredicto.**

= "resultado que emiten los sinodos una vez que el sustentante realiza su examen profesional"

["aprobado" | "no aprobado"]

## 3.4

# NORMALIZACIÓN DE LA BASE DE DATOS

Edgar F. Codd [COD90] desarrolló una teoría de normalización. Toda relación de base de datos esta en una particular forma normal si y sólo si cumple con algunas reglas.

Codd definió algunas formas normales aunque originalmente fueron la primera (1FN), la segunda (2FN) y la tercera (3FN) debido a ciertas insuficiencias en estas formas normales, Codd en compañía de Boyce perfeccionaron la 3FN y a esta nueva regla le nombraron Forma Normal de Boyce y Codd (FNBC). Tiempo después Fagin desarrollo la cuarta forma normal (4FN) y por último la quinta forma normal (5FN) o también llamada FN/PR.

Estas formas o reglas están hechas de tal forma que si se cumple la 1FN, las siguientes formas cumplen con la 1FN.

La 2FN es más deseable que la 1FN, a su vez la 3FN es más deseable que la 2FN y así sucesivamente. Por lo menos debe cumplirse con la 3FN para diseñar una base de datos.

### 3.4.1 Primera Forma Normal.

Una relación está en primera forma normal si y sólo si todos los dominios subyacentes solo contienen valores atómicos.

Grupo	Nombre prof.	Ap. paterno p. prof.	Ap. materno prof.	Nacionalidad	Grado
0001	Ignacio	Burgoa	Onhueia	Mexicana	Dr.
0007	Ignacio	Galindo	Garfias	Mexicana	Dr.
Sexo	Estado civil	Fecha de nac.	Cédula profes.	Profesor activo	Tipo profesor
M	Casado	13/03/1918		Si	Emérito
M	Casado	07/11/1913		Si	Emérito
Dirección part.	Dirección laboral	Localizador	Antigüedad	Nomb. asignat.	Créditos
			49	Amparo II	6
			40	Der. Sucesono	6
Horas semana	Semestre	Area	Plantel	Plan	División
3	7	6	Fac. de Derecho	1994	Licenciatura
3	7	8	Fac. de Derecho	1994	Licenciatura
Inicio clase	Fin de clase	Días	Aula	Cupo máximo	Lugares ocup.
08:30	10:00	MJ	D-306	90	83
09:00	11:00	LMV	E-104	90	87
No. rectificaciones	Porc. de asistencia	Promedio calif.			
0	93	7.80			
1	95	8.10			

Figura A. Tabla de Grupos en primera forma normal

En la intersección de una fila x con una columna y debe haber sólo un valor indivisible.

### 3.4.2 Segunda Forma Normal.

Codd define a la segunda forma normal 2FN como: " Una relación esta en segunda forma normal si y sólo si esta en 1FN y cada atributo no primo es completamente dependiente de la llave primaria".

Los atributos no primos son todos aquellos que no forman parte de la llave primaria.

A continuación se muestran las tablas resultantes de la 2FN:

Cve. Asignatura	Grupo	RFC profesor	Ap. paterno prof.	Ap. materno prof.	Nombre prof.
PK	PK	PK			
1704	0001	BUOI180313	Burgoa	Orihuela	Ignacio
1706	0007	GAGI111107	Galindo	Garfias	Ignacio
Sexo	Estado civil	Fecha de nac.	Cédula profes.	Profesor activo	Tipo profesor.
M	Casado	13/03/1918		Si	Emérito
M	Casado	07/11/1913		Si	Emérito
Dirección part.	Dirección labóral	Localizador	Grado	Nomb Asignat.	Créditos
			Dr.	Amparo II	6
			Dr.	Der. Sucesorio	6
Horas semana	Semestre	Area	Plantel	Plan	División
3	7	6	Fac. de Derecho	1994	Licenciatura
3	7	8	Fac. de Derecho	1994	Licenciatura
Inicio clase	Fin de clase	Días	Aula	Cupo máximo	Lugares ocup.
08:30	10:00	MJ	D-306	90	83
09:00	10:00	LMV	E-104	90	87
No rectificaciones	Porc. de asistencia	Promedio calif.	Nacionalidad	Antigüedad	
0	93	7.80	Mexicana	49	
1	95	8.10	Mexicana	40	

Figura B. Tabla de Grupos en segunda forma normal

### 3.4.3 Tercera Forma Normal.

Definición de Codd para esta forma: " Una relación esta en tercera forma normal si y sólo si esta en segunda forma normal y todo atributo no primo es dependiente no transitivamente de la llave primaria".

Como podemos observar en la tabla anterior los datos de profesor no dependen directamente de la llave de la tabla de **Grupos** por lo cual se crea una nueva relación independiente cuya llave es el RFC del profesor.

Otros datos como los referentes a asignaturas tampoco dependen directamente de la llave primaria por lo que sucede lo mismo del punto anterior creándose una nueva tabla.

Cve. Asignatura	Grupo	RFC	Días	Aula	Cupo máximo
PK	PK	PK			
1704	0001	BUO1180313	MJ	D-306	90
1706	0007	GAG111107	LMV	E-104	90
Lugares ocup.	Inicio clase	Fin de clase	No rectificaciones	Porc. de asistencia	Promedio calif.
83	08:30	10:00	0	93	7.80
87	09:00	10:00	1	95	8.10

Figura C. Tabla de Grupos en tercera forma normal

RFC	Ap. paterno prof.	Ap. materno prof.	Nombre prof.	Nacionalidad	Sexo
PK					
BUO1180313	Burgoa	Orihuela	Ignacio	Mexicana	M
GAG111107	Galindo	Garfias	Ignacio	Mexicana	M
Estado civil	Fecha de nac.	Cédula profes.	Profesor activo	Tipo profesor	Dirección part.
Casado	13/03/18		Si	Emérito	
Casado	07/11/11		Si	Emérito	
Dirección laboral	Localizador	Grado	Antigüedad		
		Dr.	49		
		Dr.	40		

Figura D. Tabla de Profesores

Cve. asignatura	Nomb. Asignat.	Créditos	Plánetel	Plan	División
PK					
1704	Amparo II	6	Fac. de Derecho	1994	Licenciatura
1706	Der. Sucesorio	6	Fac. de Derecho	1994	Licenciatura
Horas semana	Semestre	Area			
3	7	6			
3	7	8			

Figura E. Tabla de Asignaturas

## 3.5

# DIAGRAMA ENTIDAD- RELACION

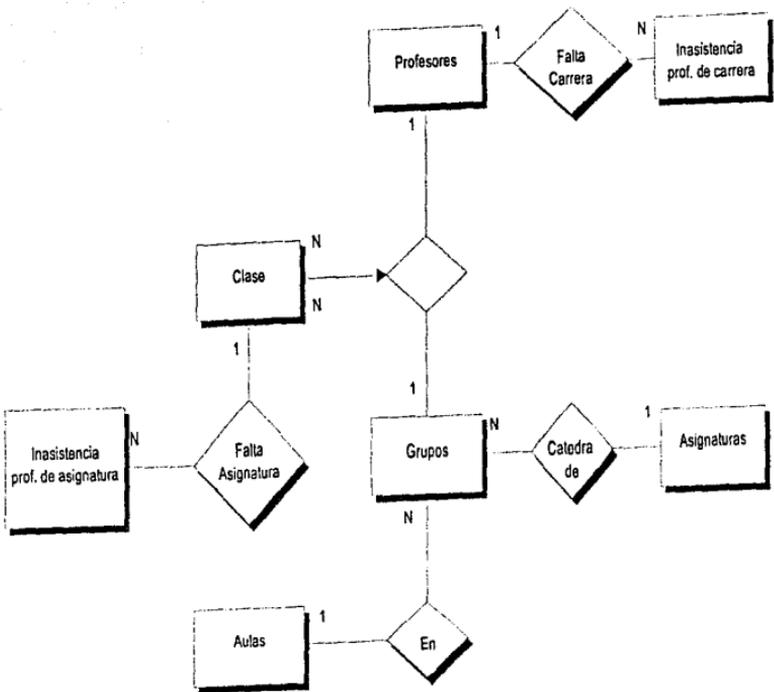
En este capítulo mostramos gráficamente la base de datos mediante el modelado Entidad-Relación previamente definido en el primer capítulo.

Empezaremos mostrando modularmente el diagrama Entidad-Relación y al final los integraremos para formar el diagrama del sistema completo.

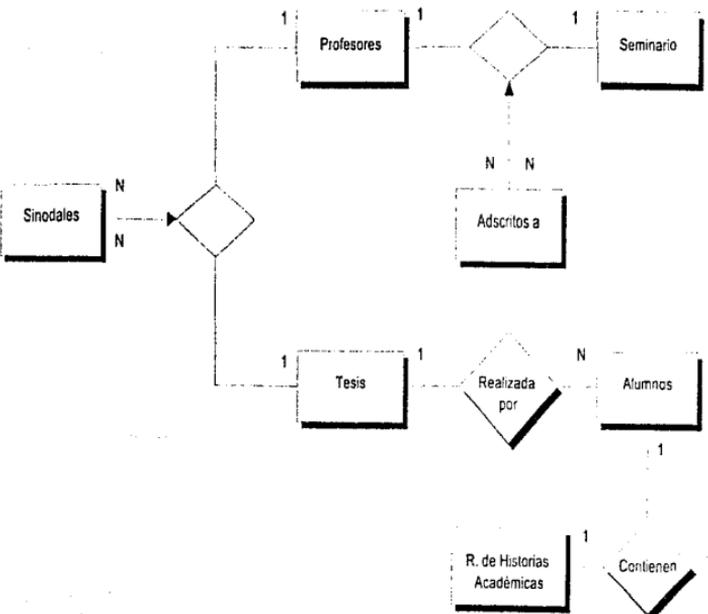
Para desarrollar el diagrama Entidad-Relación se utilizó la herramienta CASE ERwin versión 2.1, este CASE no utiliza exactamente la misma nomenclatura que definimos en el primer capítulo, sin embargo el concepto es el mismo.

La nomenclatura que varía es la de las relaciones o entidades asociativas la que comúnmente es representada mediante un rombo, en ERWin son representadas mediante rectángulos con vértices ligeramente redondeados además les pintamos un fondo gris para poderlos diferenciar de las entidades.

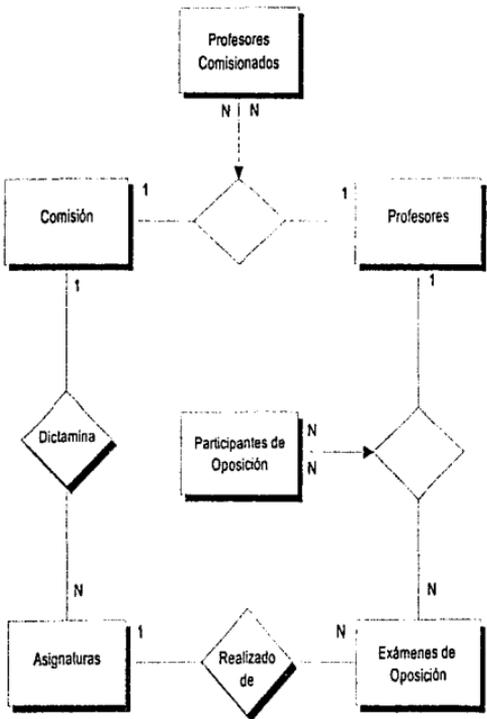
3.5.1 Módulo de Inasistencias del Personal Docente.



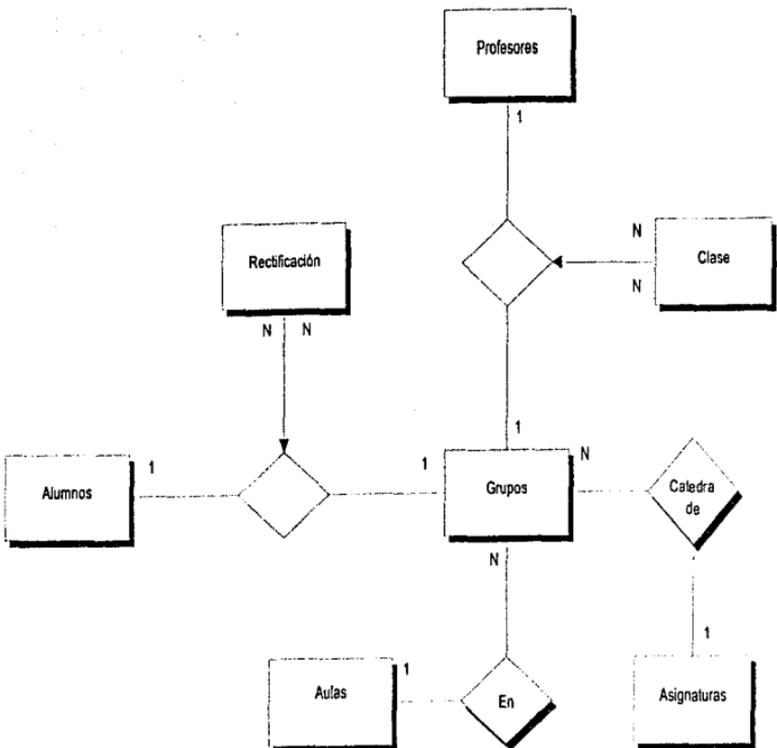
### 3.5.2 Módulo de Registro de Tesis y Examen Profesional.



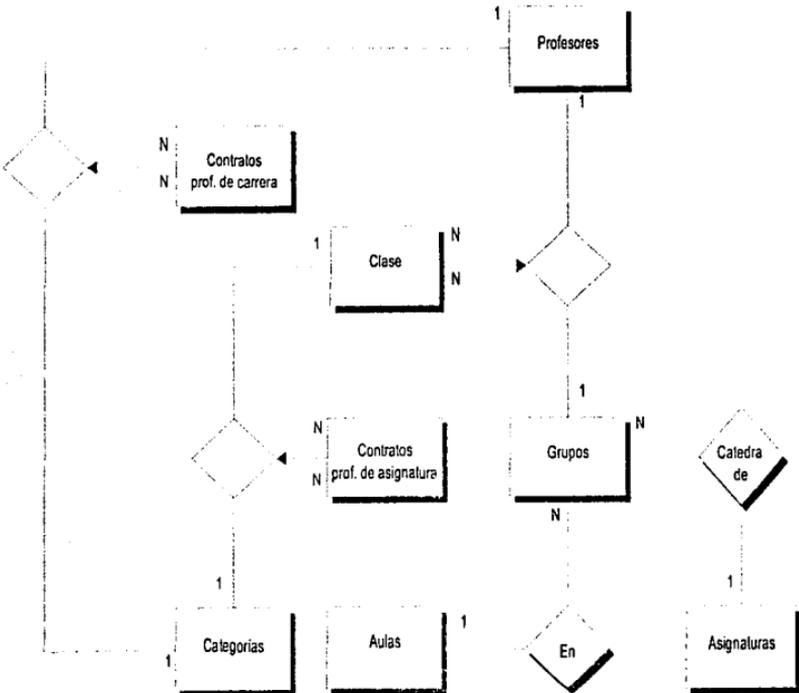
### 3.5.3 Módulo de Exámenes de Oposición.



### 3.5.4 Módulo de Rectificación de Calificaciones.



### 3.5.5 Módulo de Generación de Contratos.



### 3.5.7 Especificaciones de Entidades.

Entidad		Profesores	
Significado		Almacén que contiene la información personal de los catedráticos de la Facultad de Derecho.	
#	Nombre	Llave Tipo	Long. Nulo Descripción
1	RIC	PK char	13 No El número del registro federal de causantes, lo utilizamos como identificador único
2	Nombre	varchar	20 Nombre(s) de pila del profesor.
3	Apellido_paderno	varchar	25 Primer apellido del profesor comúnmente heredado del padre
4	Apellido_materno	varchar	25 Segundo apellido del profesor comúnmente heredado de la madre.
5	Grado	char	4 Grado académico, puede ser Lic., Mtro. o Dr.
6	Nacionalidad	char	3 Comúnmente lugar de origen. Puede ser MEX, USA, GER, BRT, SPN, ARG, BRG.
7	Sexo	char	1 M: masculino, F: femenino
8	Estado_civil	char	1 S: soltero, C: casado D: divorciado, V: viudo U: unión libre, etc.
9	Fecha_de_nacimiento	smaltdatotime	6
10	Cedula_profesional	char	9 Identificador alfanumérico que se recibe cuando se logra el grado de licenciatura.
11	Profesor_activo	char	1 N: no, S: si. Profesor que esta impartiendo clase el periodo actual.
12	Tipo_de_profesor	char	2 Tc: tiempo completo, Mt: medio tiempo, Ad: asignatura definitivo, etc.
13	Observaciones	varchar	255 Comentarios diversos sobre el profesor.
14	Casa_calle_y_numero	varchar	30 Domicilio particular del profesor.
15	Casa_colonia	varchar	30
16	Casa_delegación_o_población	char	18
17	Casa_ciudad	char	20
18	Casa_estado	tinyint	2
19	Casa_CP	char	5
20	Casa_telefono	char	13
21	Trabajo_calle_y_numero	varchar	30 Domicilio laboral del profesor.
22	Trabajo_colonia	varchar	30
23	Trabajo_delegación_o_población	char	18
24	Trabajo_ciudad	char	20
25	Trabajo_estado	tinyint	2
26	Trabajo_CP	char	5
27	Trabajo_telefono	char	13
28	Telefono_celular	char	13
29	Correo_electronico	varchar	30 Clave y dominio en donde puede recibir recados a través de una computadora.
30	Fecha_ingreso	smaltdatotime	6 La fecha en que fue contratado por primera vez para impartir cátedra.

Entidad		Inasistencias de profesores de carrera.				
Significado		Almacén histórico que contiene las faltas de los profesores de medio tiempo y tiempo completo de la Facultad.				
#	Nombre	Llave	Tipo	Long.	Nulo	Descripción
1	Rfc	PK	char	13	No	El número del registro federal de causantes, lo utilizamos como identificador único.
2	Fecha_falta	PK	smaldateime	6	No	Día en que no asistió a la facultad a cumplir con sus funciones encomendadas.
3	Tipo_de_inasistencia		tinyint	1		Causa que puede ser: No asistió, falta por enfermedad, falta justificada, etc.

Entidad		Grupos				
Significado		Almacén que contiene la información histórica de los grupos de las diferentes divisiones y periodos.				
#	Nombre	Llave	Tipo	Long.	Nulo	Descripción
1	Clave_de_grupo	PK	char	13	No	Identificador único de algún periodo y de alguna división.
2	Periodo		char	5		Año y periodo del semestre. Ejemplo 19971, 19972, 19981, 19982, etc.
3	Tipo		char	1		O: ordinario, E: extraordinario, A: acuerdo de pasantes.
4	División		char	1		L: licenciatura, P: posgrado, S: sistema de universidad abierta.
5	Grupo		char	2		Número de grupo de una misma materia. 01,02,03, 20.
6	Clave_de_asignatura	FK	char	4		Identificador de una asignatura.
7	Cupo_maximo		tinyint	3		Número tope de alumnos que pueden inscribirse a un grupo.
8	Lugares_ocupados		tinyint	3		Número de alumnos que se inscriben a un grupo.
9	Clave_aula	FK	char	8		Identificador único de una aula.
10	Hora_inicio		smaldateime	6		Hora de inicio de la clase.
11	Hora_fin		smaldateime	6		Hora de terminación de la clase.
12	Dias		tinyint	2		Días de la semana que tiene clase el grupo. 1.LMIV, 2.LMV,3.MJ, etc.

Entidad		Asignaturas			
Significado		Almacén que contiene las asignaturas de viejo y nuevo plan.			
#	Nombre	Llave	Tipo	Long.	Nulo Descripción
1	Clave_de_asignatura	PK	char	4	No Identificador de una asignatura.
2	Nombre_de_la_asignatura		varchar	40	Nombre de la asignatura.
3	Clave_de_seminario	FK	tinyint	2	Identificador de seminario. 1. Penal, 2. Civil, 3. Constitucional, etc.
4	Tipo_plan		char	1	N. Plan actual de 1994, V. Plan anterior de 1968.
5	Creditos		tinyint	2	Número de créditos por materia.
6	Horas_a_la_semana		tinyint	2	Horas obligatorias de impartición de cátedra a la semana.
7	Semestre		tinyint	2	Número de semestre en que se debe de dar la asignatura.
8	Clave_de_plantel		tinyint	1	Plantel donde se imparte la asignatura.
9	Division		char	1	Licenciatura, P posgrado, S sistema de universidad abierta.

Entidad		Aulas			
Significado		Almacén que contiene los salones en los cuales se puede impartir cátedra.			
#	Nombre	Llave	Tipo	Long.	Nulo Descripción
1	Clave_aula	PK	char	8	No Identificador único de una aula.
2	Nombre_del_aula		varchar	40	Nombre del aula.
3	Capacidad		tinyint	3	Número de pupitres que contiene cada aula.

Entidad		Inasistencia profesores de asignatura.				
Significado		Almacén que contiene las faltas en las que incurrió un profesor.				
#	Nombre	Llave	Tipo	Long.	Nulo	Descripción
1	R/c	PK	char	13	No	El número del registro federal de causantes, lo utilizamos como identificador único.
2	Clave_de_grupo	PK	char	13	No	Identificador único de algún periodo y de alguna división.
3	Fecha_de_inicio	PK	smaldateime	6	No	Primer día que imparte la cátedra el profesor en ese grupo.
4	Fecha_falta	PK	smaldateime	6	No	Día en que no asistió a la facultad a cumplir con sus funciones encomendadas.
5	Tipo_de_inasistencia		tinyint	1		Causa que puede ser: No asistió, falta por enfermedad, falta justificada, etc.

Entidad		Seminario.				
Significado		Áreas de estudio específico en derecho.				
#	Nombre	Llave	Tipo	Long.	Nulo	Descripción
1	Clave_de_seminario	PK	tinyint	2	No	Identificador de seminario. 1: Penal, 2: Civil, 3: Constitucional, etc.
2	Nombre_del_seminario		char	25		Nombre oficial del seminario.
3	Director_del_seminario		char	30		Persona que preside el seminario.

Entidad		Tesis			
Significado					
Almacén histórico que contiene la información de los exámenes profesionales.					
#	Nombre	Llave	Tipo	Long. Nulo Descripción	
1	No_registro	PK	int	7 No	Identificador único de un examen profesional.
2	Rlc	FK	char	13	El número del registro federal de causantes, del director de la tesis.
3	Clave_de_seminario	FK	tinyint	2	Identificador de seminario. 1.Penal, 2.Civil,3.Constitucional, etc.
4	No_de_cuenta	FK	char	9	Identificador único de un alumno de la universidad.
5	Fecha_de_registro		smaldateime	6	Fecha en la que se registro el tema de tesis en Sria. Gral
6	Tema		varchar	200	Título de la tesis.
7	Aprobación_de_tema		bit	1	Bandera que sirve para que se siga el tramite de titulación.
8	Fecha_examen		smaldateime	6	Fecha en que se llevará a cabo el examen profesional.
9	Hora_examen		smaldateime	6	Hora en que se llevará a cabo el examen profesional.
10	Aula_o_auditorio		varchar	10	Lugar en que se llevará a cabo el examen profesional.
11	Mención_honorifica		bit	1	Bandera que indica si el alumno obtuvo la mención.
12	Resultado		char	1	Resolución que dictó el jurado. Aprobado, suspendido, cancelado.

Entidad		Alumnos			
Significado					
Almacén que contiene la información personal de los alumnos de la Facultad de Derecho.					
#	Nombre	Llave Tipo	Long.	Nulo Descripción	
1	No_de_cuenta	PK char	9	No	Identificador único de un alumno de la universidad.
2	Nombre_alumno	varchar	40		Nombre completo del alumno como lo manda DGAE.
3	Plantel	tinyint	1		Plantel donde se imparte la asignatura.
4	Carrera	tinyint	1		Carrera a la que está inscrito en la facultad.
5	Año_de_ingreso	char	4		En que año ingresó a la carrera.
6	Nacionalidad	char	3		Comunión Lugar de origen. Puede ser MEX, USA, GER, BRT, SPN, ARG, BR5.
7	Causa_de_ingreso	tinyint	2		Categoría con la cual ingresó el alumno a la carrera.
8	Causa_exalumno	tinyint	2		Categoría que por diversas causas se le asignan a un exalumno.
9	Sexo	char	1		M: masculino, F: femenino
10	Fecha_de_nacimiento	smalldatetime	6		
11	Fecha_de_movimiento	smalldatetime	6		Fecha en la cual registro su última inscripción.
12	Marca_de_inscrito	char	1		Bandera que se activa si el alumno está inscrito en el periodo actual.
13	ACasa_calle_y_numero	varchar	30		Domicilio particular del alumno.
14	ACasa_colonia	varchar	30		
15	ACasa_delegación_o_poblac.	char	18		
16	ACasa_ciudad	char	20		
17	ACasa_estado	tinyint	2		
18	ACasa_CP	char	5		
19	ACasa_telefono	char	13		
20	ATrabajo_calle_y_numero	varchar	30		Domicilio laboral del alumno.
21	ATrabajo_colonia	varchar	30		
22	ATrabajo_delegación_o_poblac.	char	18		
23	ATrabajo_ciudad	char	20		
24	ATrabajo_estado	tinyint	2		
25	ATrabajo_CP	char	5		
26	ATrabajo_telefono	char	13		
27	Telefono_celular	char	13		
28	Correo_electronico	varchar	30		Clave y dominio en donde puede recibir recados a través de una computadora.

Entidad		Resumen de historias académicas.			
Significado		Almacén histórico que contiene la información académica de los alumnos de la Facultad de Derecho.			
#	Nombre	Llave Tipo	Long.	Nulo Descripción	
1	No_de_cuenta	PK char	9	No	Identificador único de un alumno de la universidad.
2	MB	tinyint	2		Número de asignaturas que ha acreditado con MB en su carrera.
3	B	tinyint	2		Número de asignaturas que ha acreditado con B en su carrera.
4	S	tinyint	2		Número de asignaturas que ha acreditado con S en su carrera.
5	Seis	tinyint	2		Número de asignaturas que ha acreditado con seis en su carrera.
6	Siete	tinyint	2		Número de asignaturas que ha acreditado con siete en su carrera.
7	Ocho	tinyint	2		Número de asignaturas que ha acreditado con ocho en su carrera.
8	Nueve	tinyint	2		Número de asignaturas que ha acreditado con nueve en su carrera.
9	Diez	tinyint	2		Número de asignaturas que ha acreditado con diez en su carrera.
10	NA	tinyint	2		Número de asignaturas que no ha acreditado en su carrera.
11	NP	tinyint	2		Número de asignaturas que no ha presentado en su carrera.
12	Revalidades	tinyint	2		Total de asignaturas que ha revalidado.
13	Acreditadas	tinyint	2		Total de asignaturas que ha acreditado.
14	AprobOrd	tinyint	2		Total de asignaturas que ha aprobado en ordinario.
15	AprobExtra	tinyint	2		Total de asignaturas que ha aprobado en extraordinario.
16	ReproOrd	tinyint	2		Total de asignaturas que ha reprobado en ordinario.
17	ReproExtra	tinyint	2		Total de asignaturas que ha reprobado en extraordinario.
18	CredOblAcum	smallint	3		Porcentaje de créditos obligatorios acumulados.
19	CredOptaAcum	smallint	3		Porcentaje de créditos optativos acumulados.
20	Covalidadas	tinyint	2		Número de materias covalidadas.

Entidad		Exámenes de oposición.			
Significado		Almacén histórico que contiene los exámenes de oposición.			
#	Nombre	Llave Tipo	Long.	Nulo	Descripción
1	Periodo	PK char	5	No	Semestre en que se convocó a examen de oposición.
2	Clave_de_asignatura	PK char	4	No	Identificador de una asignatura.
3	Clave_de_comisión	FK tinyint	2		Identificador de una comisión.
4	Fecha_de_convocatoria	smaldateime	6		Día en que fue publicada la convocatoria.
5	Fecha_de_diciamen	smaldateime	6		Día en que el consejo técnico aprobó los resultados de los alumnos.
6	No_de_plazas	tinyint			Número de plazas que concursan en un examen de oposición.

Entidad		Comisión.			
Significado		Grupo de profesores especialistas en la materia.			
#	Nombre	Llave Tipo	Long.	Nulo	Descripción
1	Clave_de_comisión	PK tinyint	2	No	Identificador de una comisión.
2	Nombre_de_comisión	char	18		

### 3.5.8 Especificaciones de Entidades-Asociativas.

Diseño e Implementación del Sistema

Entidad Asociativa	Clase					
Significado	Almacén que contiene la información de los grupos que imparten los profesores cada semestre.					
Entidades Participantes	Profesores y Grupos.					
#	Nombre	Llave	Tipo	Long.	Nul	Descripción
1	Rlc	PK	char	13	No	El número del registro federal de causantes, lo utilizamos como identificador único.
2	Clave_de_grupo	PK	char	13	No	Identificador único de algún periodo y de alguna división
3	Fecha_inicio	PK	smaldateime	6	No	Día en que empezó a impartir cátedra al grupo designado.
4	Fecha_fin		smaldateime	6		Día en que terminó de impartir cátedra al grupo designado

Entidad Asociativa	Sinodales					
Significado	Almacén que contiene la información de los profesores que forman un sínodo para una tesis.					
Entidades Participantes	Profesores y Tesis.					
#	Nombre	Llave	Tipo	Long.	Nul	Descripción
1	Rlc	PK	char	13	No	El número del registro federal de causantes, lo utilizamos como identificador único
2	No_registro	PK	int	7	No	Identificador único de un examen profesional.
3	Funcion		inyint	1		1 presidente, 2 secretario, 3 vocal, 4. suplente uno, suplente dos.
4	Asistencia		bit	1		Banderita que indica si asistió o no al sínodo al examen profesional.

Entidad Asociativa	Adscritos a seminario.				
Significado	Almacén que contiene la información de los profesores que están inscritos a un seminario.				
Entidades Participantes	Profesores y Seminario.				
#	Nombre	Llave Tipo	Long.	Nut	Descripción
1	Rfc	PK char	13	No	El número del registro federal de causantes, lo utilizamos como identificador único.
2	Clave_de_seminario	PK tinyint	2	No	Identificador de seminario. 1.Penal, 2.Civil,3.Constitucional, etc
3	Fecha de ingreso	smalldate/time	6		Día en que el profesor fue aceptado en el seminario indicado.

Entidad Asociativa	Comisionado				
Significado	Almacén que contiene la información de los profesores que forman parte de una comisión.				
Entidades Participantes	Profesores y Comisión.				
#	Nombre	Llave Tipo	Long.	Nut	Descripción
1	Rfc	PK char	13	No	El número del registro federal de causantes, lo utilizamos como identificador único.
2	Clave_de_comisión	PK tinyint	2	No	Identificador de una comisión.

Entidad Asociativa	Candidatos a oposición					
Significado	Almacén que contiene la información de los profesores que participan den un examen de oposición.					
Entidades Participantes	Profesores y Exámenes de Oposición					
#	Nombre	Llave	Tipo	Long.	Nul.	Descripción
1	Rfc	PK	char	13	No	El número del registro federal de causantes, lo utilizamos como identificador único.
2	Clave_de_asignatura	PK	char	4	No	Identificador de una asignatura.
3	Periodo	PK	char	5	No	Semestre en que concursa.
4	Resultado		tinyint	1		Calificación que obtiene el profesor concursante: Vencedor, Apto, No apto, Deserto.
5	Tema_asignado		varchar	100		Investigación que se le solicita al profesor concursante

Entidad Asociativa	Rectificación					
Significado	Almacén que contiene la información de los alumnos inconformes con una evaluación final.					
Entidades Participantes	Alumnos y Clase					
#	Nombre	Llave	Tipo	Long.	Nul.	Descripción
1	Clave_de_grupo	PK	char	13	No	Identificador único de algún periodo y de alguna división.
2	No_de_cuenta	PK	char	9	No	Identificador de una asignatura.
3	Rfc	FK	char	13		El número del registro federal de causantes, lo utilizamos como identificador único.
4	No_de_folio		int	8		Número asignado a la rectificación.
5	Folio_de_acta		int	8		Número consecutivo de acta a rectificar
6	Evaluación_pasada		char	2		Calificación que se solicita rectificar.
7	Evaluación_presente		char	2		Se ratifica la calificación o se modifica.
8	Ranglon_del_acta		tinyint	3		Ranglón correspondiente a la corrección del acta

Entidad Asociativa		Contratos profesor de asignatura.		
Significado		Almacén que contiene la información de los contratos celebrados con profesores de asignatura para cátedra.		
Entidades Participantes		Categoría y Clase		
#	Nombre	Llave Tipo	Long. N°	Descripción
1	Rlc	PK char	13	No El número del registro federal de causantes, lo utilizamos como identificador único.
2	Clave_de_grupo	PK char	13	No Identificador único de algún periodo y de alguna división.
3	Fecha_inicio	PK smalldatetime	6	No Día en que empezó a impartir cátedra al grupo designado.
4	Clave_de_categoria	PK char	8	No Tipo de contratación del profesor.
5	Código_programatico	char	14	Código que hace referencia a la partida presupuestal.
6	No_de_oficio	smallint	8	Número de oficio con el cual se envía el contrato a la Dir. Gral de Personal.
7	No_de_registro_de_plaza	char	18	Identificador de la plaza que tiene dada de alta DGP.
8	No_de_expediente	smallint	8	Expediente del profesor.
9	Tipo_de_personal	char	1	A. administrativo, I. investigador, D. docente.
10	Movimiento	tinyint	1	Puede ser una alta, una baja, una licencia, etc.
11	Causa	tinyint	2	El motivo del movimiento.
12	Tipo_de_profesor_en_contrato	tinyint	2	Categoría del profesor.
13	No_de_relacion	char	18	Campo no usado actualmente, pero puede ser activado en cualquier momento
14	No_de_horas_teoricas	decimal	4	Número de horas teóricas por las que se contrata.
15	No_de_horas_prácticas	decimal	4	Número de horas prácticas por las que se contrata.
16	Fecha_de_inicio	smalldatetime	6	Día a partir del cual el contrato entra en vigor.
17	Fecha_de_fin	smalldatetime	6	Día a partir del cual expira el contrato.
18	Sustituye_a	char	40	Profesor al cual se sustituye en la plaza.
19	Observaciones_en_contrato	char	100	Diversos comentarios.

## 3.6

# DEFINICIÓN DE LA BASE DE DATOS EN SYBASE

### *3.6.1 Generación de las Bases de Datos en Sybase.*

Dentro de la generación de la base de datos y la creación de las tablas que describirán completamente la forma en que operará el sistema es necesario tomar aspectos que garanticen de manera confiable y segura que los datos que se almacenaran son los correctos, estos aspectos son:

#### **Integridad de datos.**

Para preservar la consistencia en los datos que se almacenan en la base de datos es necesario implementar una serie de reglas que restrinjan los valores que pueden tomar los diferentes campos dentro de la base de datos. Los aspectos a considerar dentro de la integridad de datos son:

## **Datos requeridos.**

Algunas columnas en la base de datos deben de contener algún valor en todos los registros.

Cuando un campo es declarado NOT NULL, se evita que ese campo quede vacío y por lo tanto sea requerido al usuario como condición para que pueda ser agregado el nuevo registro o bien sea actualizado.

## **Validación.**

Cada campo de la base de datos tiene un dominio, es decir un conjunto de valores para los cuales el contenido del campo es válido. Dentro de Sybase la creación de RULES permite la validación de los valores para un campo determinado.

## **Integridad de la entidad.**

Las llaves primarias de una tabla son utilizadas para garantizar la inclusión de un valor único en cada registro, los demás campos podrán contener distintos valores. Para la base de datos es importante tener llaves primarias ya que a través de ellas se podrá identificar de manera única el registro dentro de la base de datos.

## **Integridad referencial.**

Una llave foránea es una liga entre bases de datos que liga cada registro con la llave foránea de la base de datos hija, con el registro de la base de datos padre que contiene el valor idéntico para su llave primaria.

Dentro de esta parte es necesario implementar diversas medidas que garanticen la existencia de la entidad padre para cualquier actualización o inserción de la entidad hija.

## **Reglas de transacciones.**

Cuando se actualiza una base de datos es necesario tomar en cuenta las reglas de transacciones que rigen el manejo de información. Es decir, cada vez que se agrega o se

actualiza información es necesario verificar la información que se agrega, por lo que se rechaza la información que no es válida.

Las reglas de transacciones son implementadas por los TRIGGERS, para los diferentes movimientos de una base de datos: inserción, actualización y eliminación.

## Consistencia.

La consistencia se refiere a la necesidad de que exista un registro en la entidad padre para todos los registros que se agregan en la entidad hija. Cuando se desee eliminar un registro en la entidad padre es necesario que la entidad hija no contenga ningún registro que sea asociado a la entidad padre.

Una vez definidos los aspectos a considerar es necesario tomar en cuenta los tipos de datos con que se cuenta en Sybase a continuación se describen dichos tipos:

Tipo de datos	Sinónimos	Rango
<b>Número exacto</b>		
tinyint		0 to 255
smallint		-2 <sup>15</sup> (-32,767)
int	integer	-2 <sup>31</sup> (-2,147,483,648 ) to 2 <sup>31</sup> -1 (2,147,483,647)
numeric (p,s)		-10 <sup>m</sup> to 10 <sup>m</sup> -1
decimal (p,s)	dec	-10 <sup>m</sup> to 10 <sup>m</sup> -1
<b>Approximate numeric</b>		
float (precision)		depende del equipo
double precision		depende del equipo
real		depende del equipo
<b>Money</b>		
smallmoney		-214,748,3648 to 214,748,3647
money		-922,337,203,685,477,5808 to 922,337,203,685,477,5807
<b>Fecha/Hora</b>		
smalldatetime		January 1, 1900 to June 6, 2079
datetime		January 1, 1753 to December 31,9999
<b>Character</b>		
char(n)	character	255 caracteres o menos
varchar(n)	char[acter]varying	255 caracteres o menos
nchar(n)	national	255 caracteres o menos
nvarchar(n)	nchar varying,	255 caracteres o menos
<b>Binary</b>		
binary(n)		255 Bytes o menos
varbinary		255 Bytes o menos

<b>Bit</b>	
bit	0 or 1
<b>Texto e Imagen</b>	
text	2 <sup>n</sup> - 1 (2,147,483,647) Bytes o menos
image	2 <sup>n</sup> - 1 (2,147,483,647) Bytes o menos

A continuación se describen cada una de las entidades con sus campos y los tipos utilizados, así como los campos que forman la llave primaria y los campos que son llaves foráneas.

Ahora bien debemos tomar en cuenta las estructuras que podemos utilizar dentro de Sybase y cuya implementación es a través de SQL.

Si bien, la implementación y generación de las estructuras fue realizada con las herramienta gráfica Erwin y SQL Server Manager se incluye en el **Apéndice A** el listado completo de la sintaxis utilizada.

### ***3.6.2 Implementación Física de la Base de Datos.***

La generación de la base de datos "Derecho" en Sybase fue utilizando una herramienta gráfica denominada Sybase SQL Server Manager incluida en la versión 11 de Sybase.

El administrador del servidor SQL es una herramienta de administración de sistemas y bases de datos para el servidor SQL. Consiste en una colección de características e interfaces visuales que hacen fácil la ejecución de tareas.

El administrador del servidor de SQL realiza tareas tales como:

- Administrar servidores SQL.
- Administrar los recursos físicos del SQL.
- Administrar los objetos de las bases de datos y las bases de datos.

- Administrar el acceso a los servidores y a los objetos de las bases de datos.

y permite ejecutar fácilmente tareas complejas como:

- La representación jerárquica de objetos en bases de datos y servidores SQL.
- La navegación entre objetos relacionados.
- La creación de nuevos objetos de bases de datos y del servidor SQL.

Lo primero que se definió fueron los tamaños de los dispositivos para el manejo del servidor Sybase, para ello se debe de especificar el nombre físico, lógico, tamaño y tipo del mismo.

Pero antes hay que conectarse al servidor Sybase donde deseamos que nuestra base de datos se encuentre. Es importante hacer notar que la creación de dispositivos es una tarea que debe de realizar el administrador del servidor de base de datos, por lo que se requiere utilizar un nombre de usuario con tal rango o privilegios.

*Conexión al servidor SQL:*

- Dar click en el icono de conexión del servidor SQL.
- Elegir un servidor para conectarse.
- Si la caja del nombre del servidor está vacía o se quiere cambiar a un servidor diferente, seleccionarlo de la lista de nombres de servidores.
- Introducir el nombre de usuario en la caja de login para conectarse al servidor SQL.
- Introducir el password.
- Click OK.

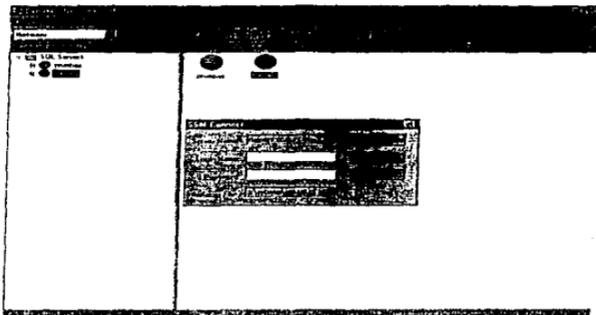


Ilustración 44 Conexión al servidor de base de datos.

Una vez establecida la conexión al servidor se procede a crear el dispositivo de la base de datos, para ello se requiere ciertos conocimientos del sistema operativo sobre el cual correrá el servidor Sybase.

*Creación de un dispositivo de la base de datos:*

- Dar click en el icono del servidor SQL.
- Elegir las opciones Server/Create/Database Device.
- Introducir el nombre del dispositivo.
- Introducir el nombre completo del dispositivo físico. Un host UNIX puede ser el nombre de la partición de un disco duro o un archivo de sistema operativo.
- Introducir el número del controlador del disco o aceptar el de default.
- Introducir el tamaño en megabytes si el dispositivo físico es un archivo. Si el dispositivo físico es una partición dura, introducir el tamaño de la partición.

- El administrador del servidor SQL proporciona el próximo número secuencial disponible para el número de dispositivo virtual.
- Si se desea el dispositivo puede ser el de default, verificar la opción Default Device.
- Si se requiere reflejar el dispositivo, introduce el nombre del dispositivo físico reflejado en la opción Mirror Name y elegir serial o paralelo para definir como se va a escribir la ocurrencia.
- Elegir la opción Create.

Como se puede observar la creación de un dispositivo de la base de datos es una tarea importante ya que se está definiendo el tamaño de la base de datos a contener; aunque posteriormente se puede incrementar el espacio designado a la base de datos es conveniente tener toda la base de datos en un solo segmento, ya que con eso se evita la fragmentación en diferentes dispositivos. Para nuestro caso solo se pueden definir dispositivos menores a 1GB por lo que se definieron dos dispositivos denominados "data1" y "data2". Finalmente haciendo cálculos aproximados sobre la cantidad de información y su ritmo de crecimiento se tomó la decisión de ocupar un solo dispositivo (data1) con un espacio de 1GB; lo que nos lleva a una duración de la información sobre este esquema de cerca de 10 años, es decir, que tendrán que pasar 10 años para que se acabe el espacio destinado. Sin embargo como se mencionó anteriormente se definieron dos dispositivos, el segundo llamado "data2" y también con un tamaño de 1GB, se utiliza temporalmente para mantener funcionando algunas aplicaciones que ya existen y que forman parte del modelo global de información de la facultad de derecho.

Una vez definidos los dispositivos se procede a crear la base de datos, que en nuestro caso se llamará "DERECHO", y que para ello se debe el dueño, el tamaño la base de datos, en que dispositivo se encuentra, si es de datos o logs, el tamaño de buffer a utilizar, etc.

*Creación de una base de datos:*

- Elegir las opciones del menú **Server/Create/Database**.
- Introducir el nombre de la base de datos.
- Introducir el "dueño" de la base de datos.
- Si se está creando la base de datos, ésta puede ser restaurada de un backup (esto no es nuestro caso).
- Para crear un usuario en la base de datos verificar la opción **Guest User**.
- Reservar el espacio en uno o más dispositivos para almacenar la base de datos.
  - Seleccionar el nombre de un dispositivo.
  - Introducir el tamaño del espacio que se requiere reservar. Si no se introduce ningún tamaño, el servidor SQL reserva 2 MB o el tamaño del modelo de la base de datos, dependiendo cual es el más grande.
  - Si se tiene el espacio limitado y se debe poner el log de transacciones y los datos en el mismo dispositivo verificar la opción **With Override** para almacenar el log y los datos en fragmentos separados del dispositivo.
- Si esta almacenando el log de transacciones en un dispositivo por separado y esta reservando el espacio para el log y no para la base de datos, seleccionar la opción **Log**.
- Dar click en **Add** para dar de alta la reservación del dispositivo. Para cambiarla, seleccionar un registro del dispositivo y cambiar los datos en las cajas de edición. Dar click en **Change** para ejecutar la función. Para deshacer la reservación, seleccionar el registro del dispositivo y dar click en **Remove**.
- Para especificar el tamaño del buffer del log, se selecciona un valor de la lista de tamaños de buffers de log. Los valores disponibles en la lista reflejan el tamaño

del buffer que existe en el log cache actual. Esta opción está disponible en la versión 11.0.

- El tamaño del buffer del log determina el tamaño de I/O que puede ejecutar el servidor SQL para el buffer de la memoria, en la memoria cache designada para la I/O del log de transacciones.
- Dar click en Create.

Cabe mencionar que se sugiere una tamaño de logs igual a 20 o 25% del tamaño de la base de datos, esto es para un mejor rendimiento en el manejo de las transacciones.

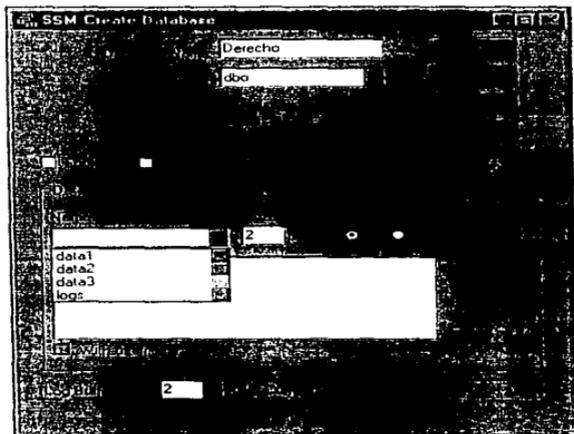
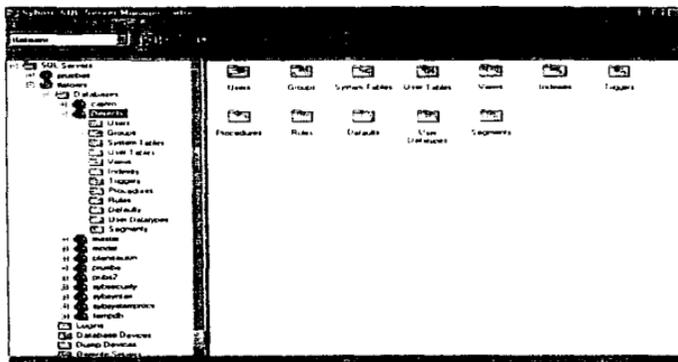


Ilustración 45 Creación de la base de datos.

Una vez creada la base de datos se puede consultar la estructura tal como aparece en la figura siguiente, es decir, se muestra el contenido de las diferentes estructuras con las cuales

se llevará a cabo el control de la integridad y el manejo de los datos a través de las aplicaciones cliente.

Como se puede ver se tienen varias carpetas que facilitan la visualización de las diferentes



**Ilustración 46** Las diferentes estructuras que se crean para una mejor administración de la base de datos.

estructuras creadas para una mejor administración de la base de datos.

**USERS**, contiene a todos los usuarios que tienen acceso a la base de datos.

**GROUPS**, contiene a los diferentes conjuntos de usuarios en que se divide para un mejor control de los mismos.

**SYSTEM TABLES**, contiene las tablas necesarias para el sistema.

**USER TABLES**, contiene las tablas definidas por el usuario.

**VIEWS**, contiene las vistas definidas por el usuario.

**INDEXES**, contiene los índices generados por el usuario.

**TRIGGERS**, contienen los procedimientos para reforzar la integridad referencial en operaciones de inserción, borrado, y actualización.

**PROCEDURES**, contiene procedimientos para poder implementar una serie de acciones a nivel SQL y de control de flujo.

**RULES**, contiene las definiciones de los valores que puede aceptar un campo de la base de datos.

**DEFAULTS**, contiene las definiciones de los valores por omisión que pueden adoptar algunos campos de las tablas.

**USER DATATYPES**, contiene los tipos de datos definidos por el usuario.

**SEGMENTS**, contiene las áreas que esta utilizando la base de datos y pueden ser desde un solo dispositivo o más.

Ya que las carpetas que contendrán las diferentes estructuras a utilizar son fijadas, es necesario generar dichas estructuras.

Para nuestro caso solo se crearon reglas, defaults, tablas, triggers e índices.



### ***3.6.3 Generación del Modelo de Datos Lógico y Físico.***

Como se mencionó anteriormente el diseño de la base de datos se realizó con Erwin/ER, que es un software de desarrollo visual para la creación de modelos de datos en el diseño de base de datos dentro de esquemas de organización de información. Permitiendo el desarrollo, análisis y generación de documentación en lapsos cortos de tiempo y durante el ciclo de desarrollo. Utiliza el esquema IDEF1X desarrollado por la fuerza aérea estadounidense.

Este software solo permite el desarrollo del esquema de datos físico a partir de la definición del esquema lógico por parte del usuario; por lo que no se requiere escribir ni una sola instrucción a nivel SQL para crear una nueva base de datos, incluyendo tablas, índices, procedimientos almacenados, triggers de integridad referencial y demás componentes que permiten un mejor manejo de los datos.

Adicionalmente Erwin puede conectarse a un esquema de base de datos existente, extraer el modelado y generar el modelo lógico, donde se puede agregar, eliminar o rediseñar el esquema recuperado y volverlo a desarrollar dentro de la misma plataforma o una diferente.

Dentro del modelado lógico de datos para nuestro proyecto se utilizaron las diferentes herramientas con las que cuenta ERwin, que a continuación se mencionan:

-  Selector.
-  Manipulador de atributos.
-  Entidad dependiente.
-  Entidad Independiente.
-  Relación completa.
-  Relación Incompleta.
-  Relación que identifica.
-  Relación que no identifica.
-  Relación muchos a muchos.
-  Bloque de texto.

A continuación describiremos como hicimos uso de ERwin/ER para el desarrollo del modelos de datos lógico y posteriormente el modelo físico de datos.

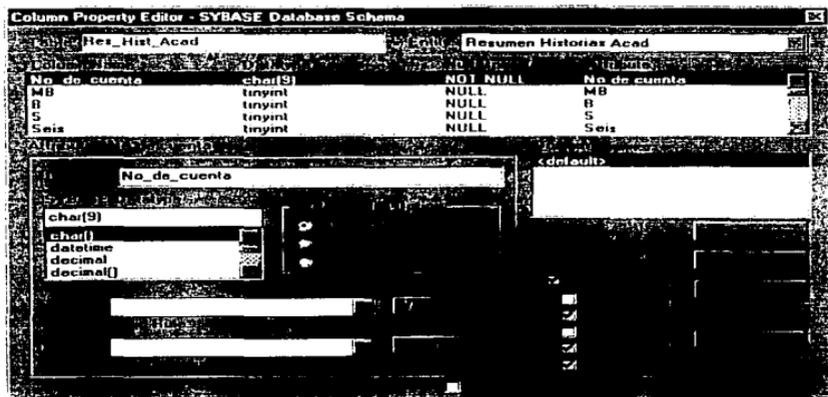
Lo primero que se desarrolla son las entidades independientes, con sus atributos, los campos que forman la llave principal y los campos que no son llave.



Ilustración 48 : Definición de una entidad independiente y sus atributos.

Una vez definidas cada una de las entidades independientes se procede a definir las entidades dependientes, de la misma manera que las independientes solo que hay que cambiar el tipo.

Una vez terminada la definición de las entidades se debe de definir el esquema de la base de datos, es decir los tipos, nombres, reglas y valores por default que adoptará cada uno de los campos de las entidades.



**Ilustración 49** Establecimiento de los atributos para los campos de cada entidad.

Cabe mencionar que dentro de la definición de los atributos de los campos de cada entidad es necesario establecer que tipo de diccionario se empleará para el diseño lógico, para nuestro caso es Sybase System 10, ya que en base a este diccionario se podrán utilizar los tipos de datos y demás estructuras para el manejo de los datos.

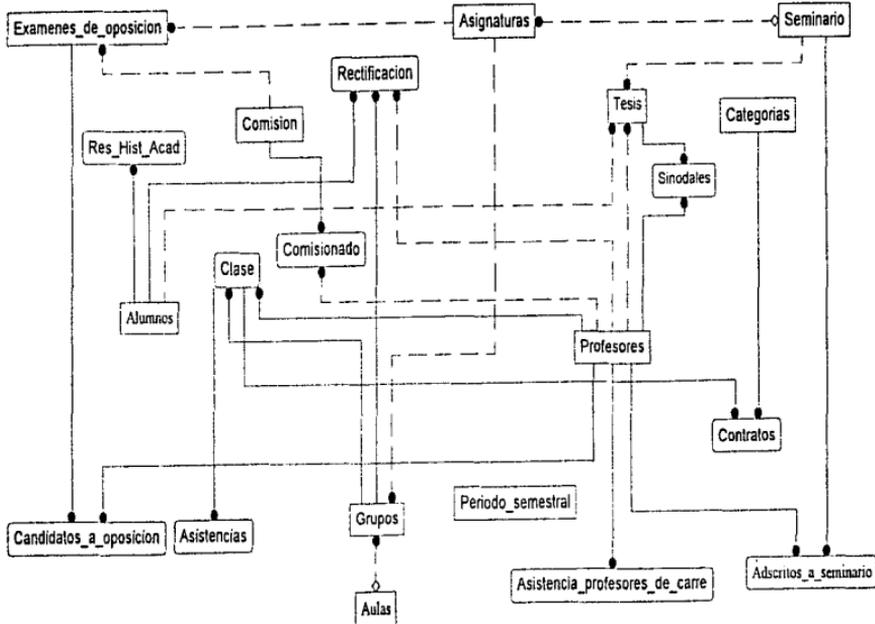
Posteriormente hay que definir el tipo de relación que existe entre las entidades ya sea que identifica o que no identifica.



Si se actualiza un registro en la entidad hija	Se restringe a que exista en la entidad padre el registro que los relaciona.
Si se borra un registro de la entidad padre	Se restringe a que no tenga registros que lo relacionen en la entidad hija.
Si se agrega un registro en la entidad padre	No hay ninguna restricción.
Si se actualiza un registro de la entidad padre	Se actualizan de manera automática todos los registros en la entidad hija que estén relacionados con la entidad padre.

De esta manera se van estableciendo las relaciones entre las diferentes entidades que forman parte de nuestro diseño lógico de los datos.

En el siguiente esquema se muestra el modelo lógico de datos que se desarrollo para el proyecto que comprende nuestro trabajo. Este esquema es solo una representación a nivel entidad, las líneas continuas representan una relación que identifica a la entidad hija, las líneas discontinuas representan una relación que no identifica y ambas incluyen una llave foránea en la entidad hija.



El siguiente esquema muestra el diagrama a nivel físico del modelo de datos lógico, es decir, es el modelo tal y como se va a generar en la base de datos ya definida.

Como se podrá observar en el esquema se tienen una serie de anotaciones que se utilizan para describir de una manera física el modelo de datos. Una de las nomenclaturas para el manejo de integridad referencial en Erwin/ER es la definición de las relaciones.

Cuadro descriptivo de las relaciones:

<b>Tipo de relación</b>	<b>Descripción</b>
<b>U:R</b>	Update : Restricted
<b>I:R</b>	Insert : Restricted
<b>D:R</b>	Delete : Restricted
<b>U:C</b>	Update : Cascade
<b>I:C</b>	Insert : Cascade
<b>D:C</b>	Delete : Cascade

Además se utiliza una numeración en las relaciones existentes, p.e. R\_6, R\_7, etc. esto es para un mejor control en cada una de las relaciones de nuestro modelo de datos.



Una vez concluidas todas las relaciones con sus respectivas entidades se puede generar el código para el modelo físico, para ello se establece la conexión adecuada para la plataforma en que se va a implementar de manera real, es decir, se elige la plataforma destino que para nuestro caso será Sybase. Esto se logra accediendo al menú Server y la opción Target Server.



Ilustración 51 Opciones para la generación del modelo físico.

Posteriormente se elige la opción Server/SYBASE Connection para establecer la conexión física al servidor Sybase. Dentro de esta opción es importante decir que la clave de usuario que se utilice para generar el esquema físico debe de tener derechos de administrador sobre el servidor Sybase. Si se cuenta con los permisos suficientes entonces se procede a generar el esquema a través de la opción Server/SYBASE Schema Generation. A continuación se presenta una figura con las opciones que pueden emplearse en la generación del modelo de datos físico.

Como se puede observar se presentan diferentes opciones entre ellas la generación de las tablas, la generación de las llaves tanto primarias como foráneas, la sentencia de vinculación de las llaves, la creación de la reglas, los defaults, los procedimientos almacenados, los triggers, etc.

Aquí mismo se puede generar un reporte con las sentencias SQL en un archivo de texto y poderse ejecutar en el servidor como un archivo batch. Este archivo se puede consultar en el Apéndice A.

Finalmente se muestran los resultados de la creación de las tablas y triggers para la base de datos "Derecho", en pantallas de Sybase SQL Server Manager.

Los nombres de los triggers están basados en el nombre de la tabla y el tipo de movimiento que se pretende utilizar, p.e. ID\_Alumnos, se refiere al trigger de la tabla Alumnos en la

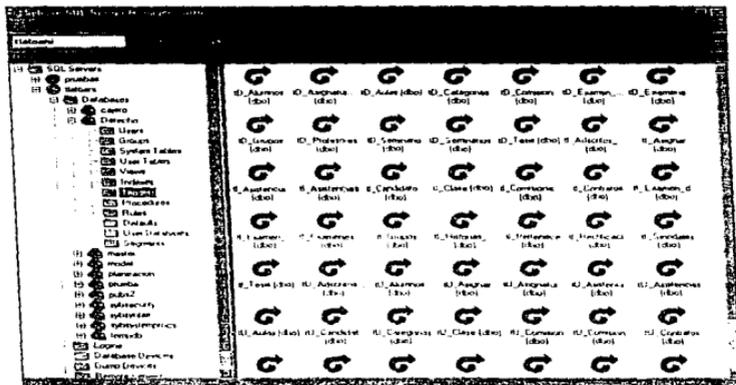


Ilustración 52 Pantalla que muestra la designación de Triggers en los procesos de inserción, actualización y eliminación de registros para cada una de las tablas.

modalidad de eliminación de registros.

Hasta este momento se ha generado la estructura completa y su implementación física de acuerdo a las especificaciones dentro del diagrama entidad-relación y diccionario de datos desarrollado con anterioridad.

### ***6.3.4 Migración de datos.***

Como se mencionó en la problemática actual, se contaba con un sistema atrasado y con el conjunto de datos distribuidos en pc's independientes.

Uno de los logros que se habían obtenido hasta ese momento era la centralización de manera parcial de los datos, al menos de los más importantes o de los que la mayoría de la gente utilizaba con mayor frecuencia.

Dado que todo esquema de cambio de plataforma o mejora de esquema de manejo de la información debe contemplar que si existe un sistema previo deben de rescatarse en la mayor cantidad posible los datos ya existentes para no empezar de nuevo, nuestro caso no fue la excepción por lo cual se procedió a una migración de datos.

PowerBuilder cuenta con una utilidad llamada "Pipeline", la cual permite exportar los datos contenidos en una tabla origen a otra tabla contenida en una plataforma igual o diferente a la existente.

Para nuestro caso se exportarán los datos de Paradox 3.5 a Sybase System 11.

Lo primero que se hace es preparar ambas tablas con los mismos campos que utilizarán y con los tipos de datos de preferencia de la tabla destino.

Después se establece la conexión entre el origen y destino de los datos, así como el origen de los datos.

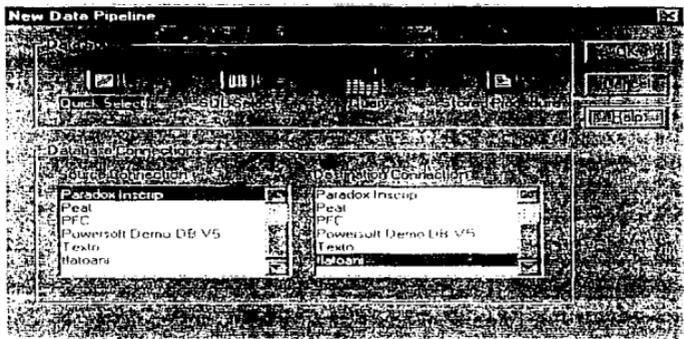


Ilustración 53 Conexión el entre el origen y destino de los datos.

Existen cuatro tipos de fuente de datos:

**Quick Select**, los datos están en forma de tablas y están relacionados por medio de una llave, y donde solo nos interesa obtener los datos de la tabla origen.

**SQL Select**, cuando se quiere un mejor control de los datos a través de una sentencia **Select** que relaciona a tablas que no contienen llaves que las unan.

**Query**, cuando los datos son definidos a través de una consulta.

**Stored Procedure**, cuando los datos están definidos por medio de una procedimiento almacenado.

Una vez seleccionado el origen de los datos así como la conexión origen y destino, se elige la tabla, las columnas que se van a exportar y sobre estas últimas algún tipo de ordenamiento, y algún criterio de selección.

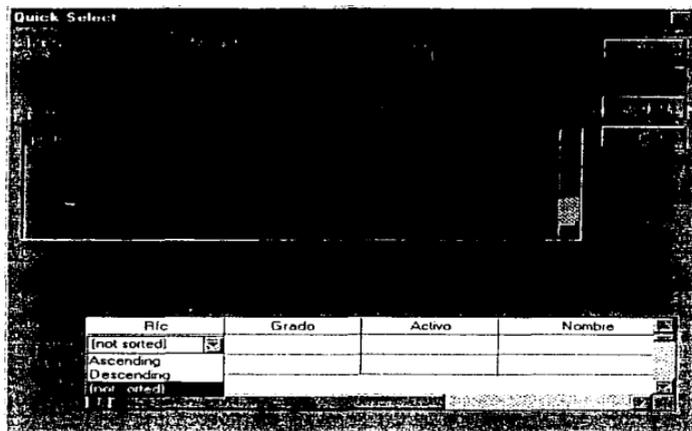


Ilustración 54 Selección de tablas y columnas a exportar.

Y por último se elige el tipo de operación que se realizará con los datos obtenidos de la tabla origen entre algunos de ellos están:

- Crear la tabla y agregarle los registros del origen.
- Reemplazar la tabla existente y agregar los registros del origen.
- Reemplazar los registros existentes por los del origen.
- Abrir la tabla destino y agregar los registros del origen.
- Actualizar los registros existentes en el destino.

Además se puede elegir el aborto de la operación después de cierto número de errores, la ejecución de la sentencia commit después de un número de registros operados.

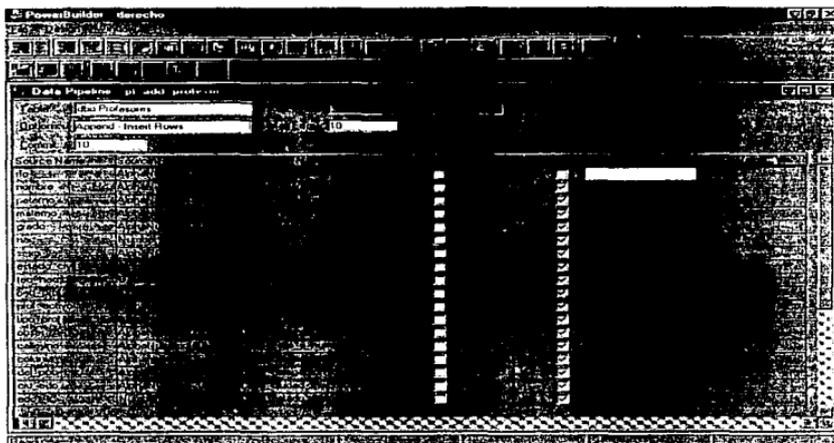


Ilustración 55 Exportación de datos.

Es importante mencionar que en nuestro caso debido a que se realizó un cambio en la estructura de la información en general, las tablas origen y destino no correspondían de manera exacta, por lo que se tuvo que preparar la información previamente a la migración.

Con este procedimiento de migración de datos en la plataforma existente se evitó tener que pasar los datos en formato ASCII y utilizar alguna otra herramienta para exportarlos y evitar problemas de conversión de datos.

## 3.7

# GENERACIÓN DE LAS APLICACIONES GRÁFICAS EN POWERBUILDER

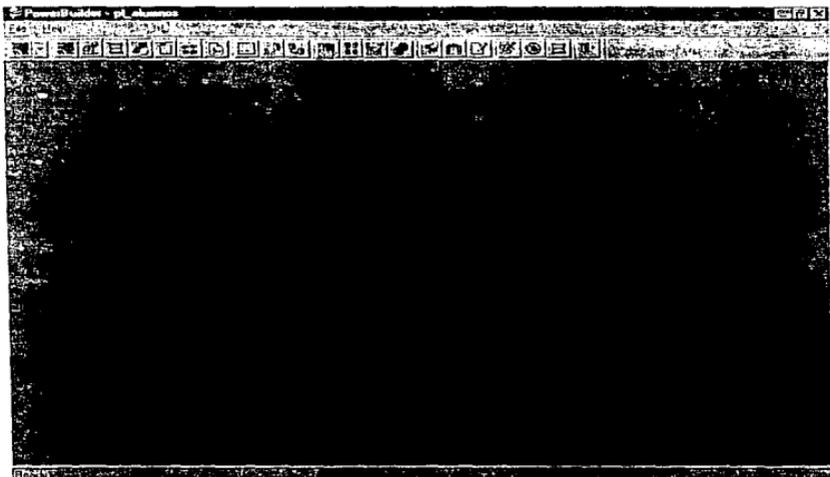
Se puede ver lo que hace un programa realizado en un lenguaje tradicional leyendo el programa. Todas las operaciones relativas a la presentación y uso del formato se encuentran en el programa. Las sentencias, que se pueden imprimir y leer, describen todas las operaciones.

Una aplicación PowerBuilder no tiene un único lugar donde se pueda leer. No existe un listado que se pueda imprimir y leer para comprender la operativa de la misma.

Con PowerBuilder normalmente se realiza poca programación. Lo que se programa son los procedimientos de los eventos asociados a un objeto.

La mayor parte del trabajo consiste en crear objetos para la aplicación, no en programar cómo reaccionarán los objetos. Las herramientas del entorno de desarrollo PowerBuilder permiten crear los objetos y escribir los procedimientos.

Al iniciar una sesión en PowerBuilder se abre una Ventana como aparece en la **Ilustración 56**.

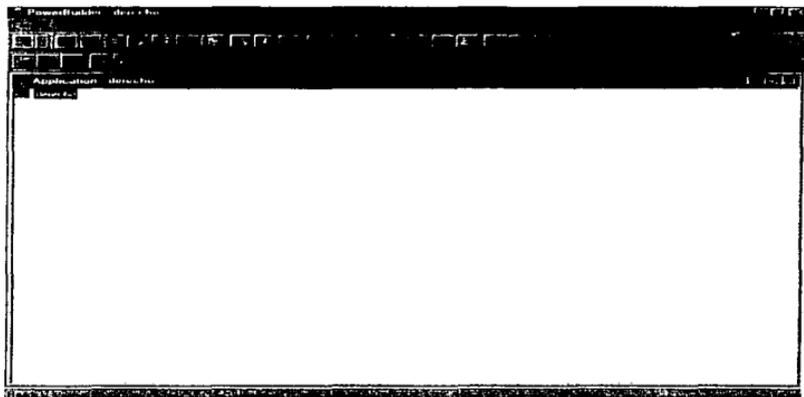


**Ilustración 56** Ventana de inicio de sesión en PowerBuilder.

PowerBuilder tiene una serie de Painters. Cada painter es una herramienta con un propósito particular. Cada icono situado en el PowerBar permite el acceso a un painter diferente.

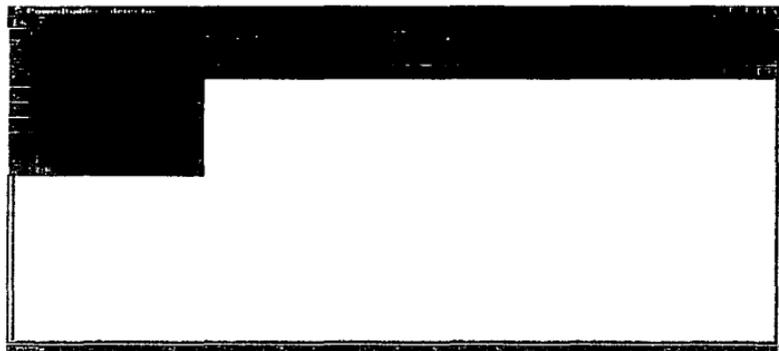
El marco general para cualquier cosa que se construya con PowerBuilder es la aplicación. La aplicación que se construya puede contener muchos objetos, todos estos objetos se encuentran dentro de una aplicación.

Para iniciar el painter de aplicación hay que hacer clic en el icono de Application en el PowerBar. En la **Ilustración 57** se muestra una painter de aplicación en ejecución.



**Ilustración 57** Painter de aplicación.

Para crear una nueva aplicación seleccione la opción New del menú File.



**Ilustración 58** Opciones de archivo del Painter de Aplicación.

Una vez realizada esta operación aparecerá el selector de librerías de aplicaciones.

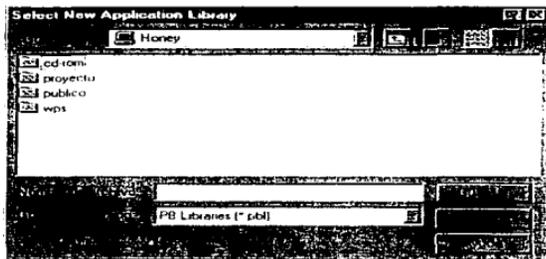


Ilustración 59 Ventana de selección de librerías para nuevas aplicaciones.

Es necesario navegar entre las unidades y subdirectorios existentes hasta localizar el lugar en que se quiera realizar la aplicación. Esto se consigue haciendo doble clic en el subdirectorio deseado.

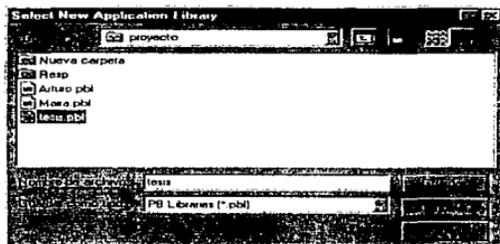


Ilustración 60 Librería seleccionada para incluir nueva aplicación.



Para abrir una aplicación ya existente seleccione la opción Open del menú File.

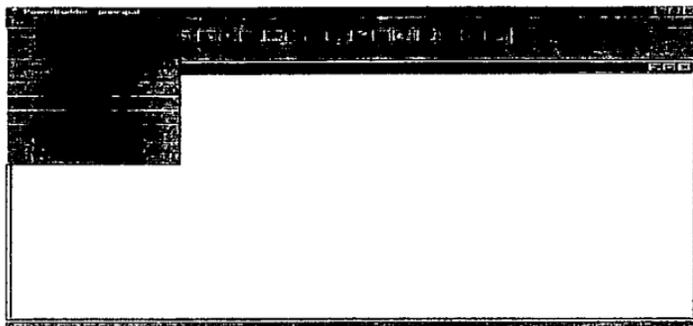


Ilustración 63 Opciones de archivo del Painter de aplicación.

De la misma manera que para crear aplicaciones se selecciona la librería.



Ilustración 64 Ventana de selección de librerías.

Seleccionar, de las aplicaciones disponibles en la ventana de selección, la aplicación que se quiera abrir haciendo doble clic en el nombre o seleccionando la aplicación y después presionando OK.

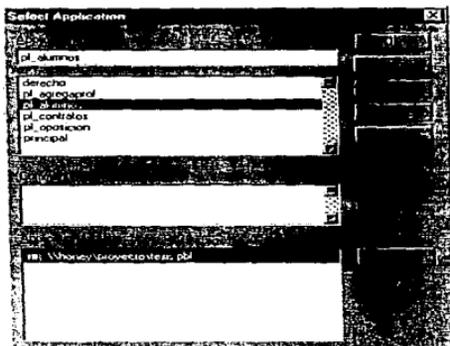


Ilustración 65 Ventana de selección de aplicación.

Al igual que sucede cuando se crea una aplicación nueva, el painter de aplicación ahora tiene en uso la que fue seleccionada y su nombre aparece en la parte superior de la pantalla.

Otro painter con el que cuenta PowerBuilder es el painter de DataWindow. Las DataWindow presentan, manipulan, actualizan e imprimen informes de datos. Una DataWindow automatiza la interfaz a la base de datos fuente. También automatiza la interfaz entre una aplicación en ejecución y el usuario.

Para crear un objeto DataWindow es necesario hacer clic en el icono DataWindow, esta acción arrancará el painter de DataWindow con la ventana de selección de DataWindow.

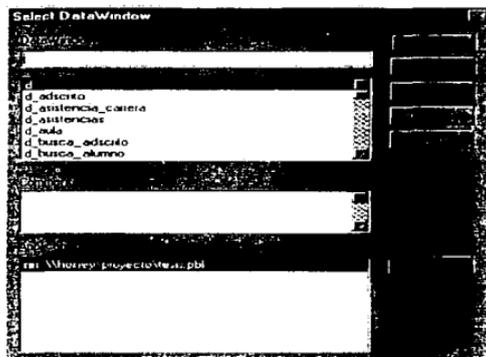


Ilustración 66 Ventana de selección de DataWindow.

Al hacer clic en el botón New aparecerá la ventana New DataWindow.

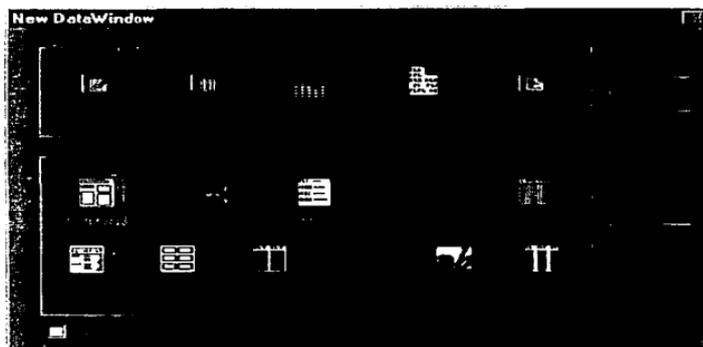


Ilustración 67 Ventana de fuente de datos y estilo de presentación de la nueva DataWindow.

Se selecciona la fuente de datos y el estilo de presentación de las opciones mostradas en la ventana. Para este ejemplo seleccionaremos la fuente de datos SQL Select y el estilo Freeform .

Una vez hecho esto, se hace clic en el botón OK para iniciar el diseño de esta DataWindow.

La siguiente ventana que aparece sirve para seleccionar tablas, y en ella se muestran las tablas existentes ya creadas con anterioridad.

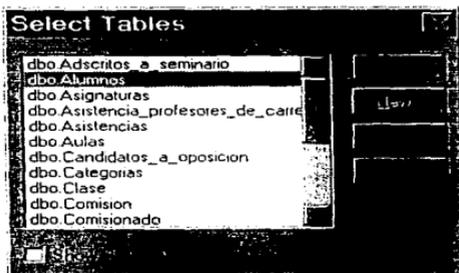


Ilustración 68 Ventana de selección de tablas.

Para seleccionar marque una o más de las tablas haciendo clic sobre ellas y posteriormente haga clic en el botón Open. Para este ejemplo solo seleccionaremos la tabla de alumnos.

Una vez hecho esto en el painter de DataWindow se muestra la tabla seleccionada y el detalle de todos sus campos.

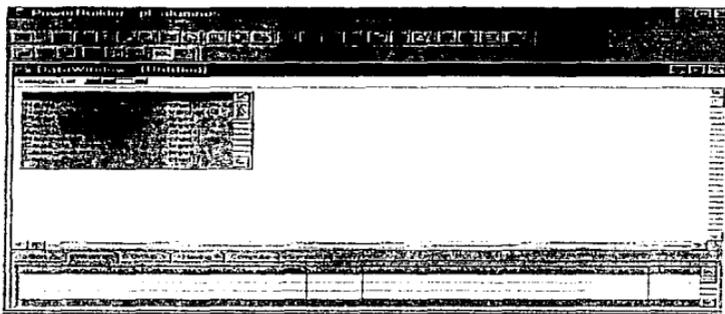


Ilustración 69 Despliegue de tablas seleccionadas con el detalles de sus campos.

Seleccione las columnas con las que se desea trabajar haciendo clic sobre ellas, observe que cada nombre de columna aparece en la parte superior de la ventana cuando se selecciona la columna de la tabla. El orden de hacer clic en los nombres es importante. De esta forma se está determinando el orden en el que se mostrarán las columnas en el objeto DataWindow una vez finalizado.

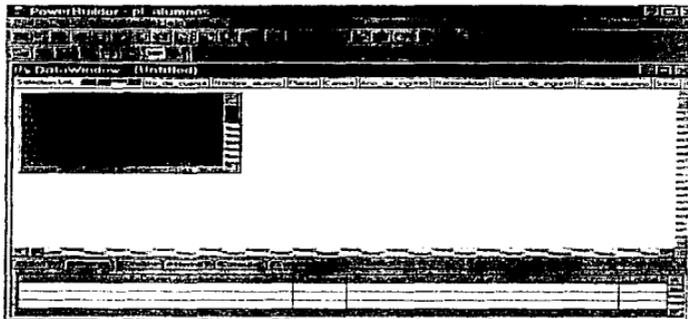


Ilustración 70 Selección de columnas de la tabla Alumnos.

Si se requiere restringir o modificar la selección SQL del objeto DataWindow, se selecciona una de las opciones de la caja de herramientas de SQL. Vamos a utilizar la opción Where para limitar al momento de ejecutar esta DataWindow dentro de una Window a que sea necesario capturar un número de cuenta para poder capturar o modificar el resto de las columnas. Se selecciona la columna No\_de\_cuenta.

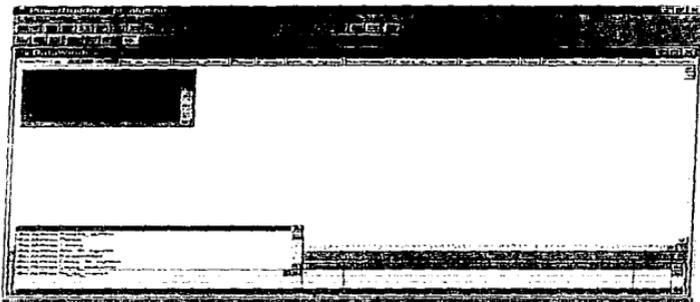


Ilustración 71 Selección de columna de la opción Where de la caja de herramientas de SQL.

Se selecciona el operador "=" y se iguala a una variable "no\_cta".

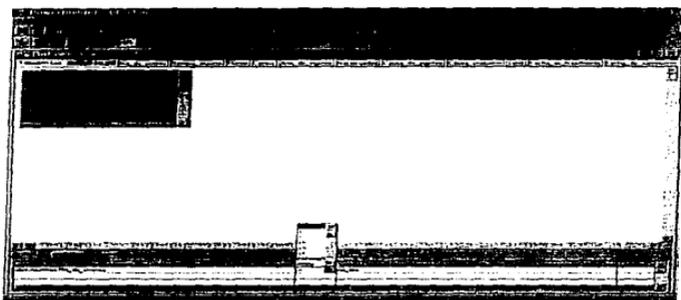


Ilustración 72 Selección de operadores de la opción Where de la caja de herramientas de SQL.

Posteriormente se declara la variable seleccionando la opción Retrieval Arguments del menú Design, donde se escribe el nombre de la variable y su tipo. Al terminar se hace clic en el botón OK.

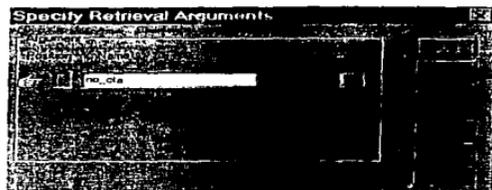


Ilustración 73 Ventana de declaración de variables.

Ahora que las columnas están seleccionadas y la variable declarada haga clic en el botón de Select. La pantalla de painter de DataWindow aparece con las columnas y los nombres de estas.



Ilustración 74 Painter de DataWindow que despliega las selecciones hechas.

Para cambiar la presentación original del objeto DataWindow se arrastran las columnas en el lugar deseado así como los nombres de las columnas. Para poder modificar el texto del nombre de las columnas es necesario colocarse en el objeto a modificar y presionar el botón derecho del ratón seleccionando la opción Properties.

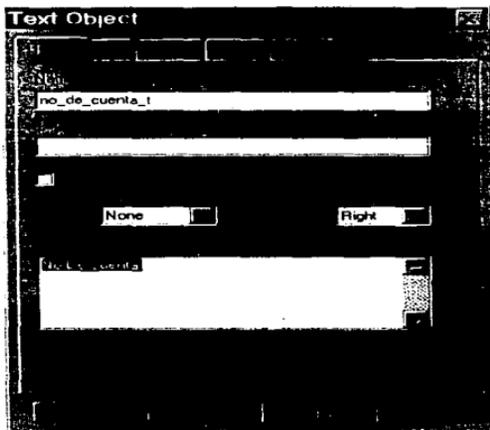


Ilustración 75 Ventana de definición de texto.

Una vez acomodadas las columnas y los textos modificados, nos da como resultado la Ilustración 76.

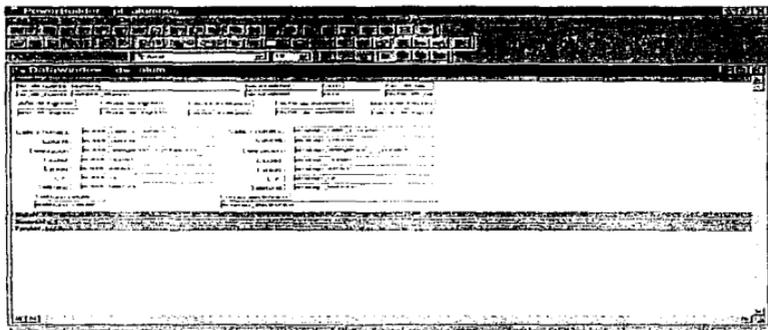


Ilustración 76 Painter de DataWindow con textos y columnas ordenadas.

Para cambiar el color del frente y el fondo de las columnas y textos es necesario seleccionarlos en conjunto o en particular y cambiar colores presionando los botones Background o Foreground del PainterBar.



Ilustración 77 Definición de colores de frente y fondo.

Para incluir mas textos se hace clic en el botón Text del PainterBar, y se hace clic en el lugar de la pantalla donde se desee dejar.



Ilustración 78 Opción de texto del PainterBar.

Ya creados estos textos se modifica su contenido y se cambian sus colores.



Ilustración 79 Textos creados y modificados en contenido y diseño.

Para modificar el orden de tabulación de los objetos mostrados en la DataWindow es necesario seleccionar la opción Tab Order del menú Design.



Ilustración 80 Selección para editar el orden de tabulación.

Una vez hecho esto, todas las columnas aparecen con un número que significa el orden de tabulación al momento de la ejecución. Las columnas que aparecen con "0" no son susceptibles a cambios. Se escribe el orden que se desee en estos números. Es necesario poner el número mas pequeño en la columna No\_de\_cuenta, ya que aquí es donde está la restricción de entrada definida anteriormente.



Ilustración 81 Definición de orden de tabulación.

Para desactivar la opción de orden de tabulación se selecciona nuevamente la opción Tab Order del menú Design. Ahora, para hacer que al momento de la ejecución la información de la tabla pueda ser modificada, se selecciona la opción Update Properties del menú Rows.



Ilustración 82 Selección para editar las propiedades de actualización.

La ventana que aparece está desactivada.

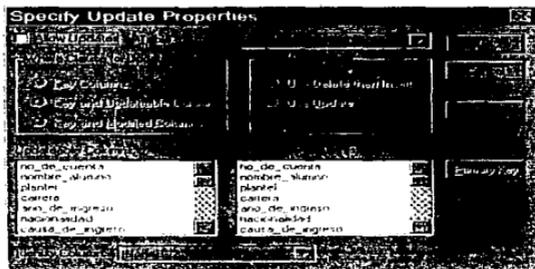


Ilustración 83 Ventana de especificación de propiedades de actualización.

Para activarla se hace clic en la opción Allow Updates y se marcan las columnas que deban actualizarse haciendo clic con el ratón en estas, de la misma manera se marca la columna llave. Si no esta marcada seleccione la tabla que se modificará en la opción Table to Update.



Ilustración 84 Propiedades de actualización definidas.

Terminada esta operación el botón OK se activará y podrá ser presionado para guardar los cambios.

El objeto DataWindow está terminado y solo falta ser salvado, seleccione la opción Save del menú File y escriba el nombre del objeto DataWindow.

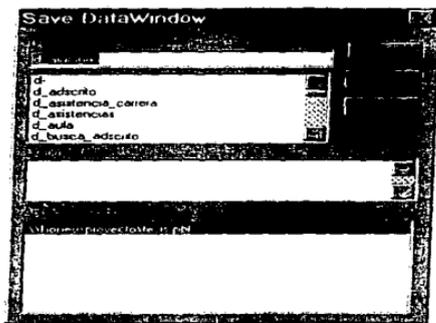


Ilustración 85 Ventana para salvar la DataWindow creada.

Ahora se presiona OK y la DataWindow quedará salvada.

Ya creado el objeto DataWindow es necesario incorporar a una ventana un control DataWindow y asociarle el objeto ya creado d\_ alumnos.

Para poder realizar esto es necesario conocer otro nuevo painter que es el painter de Window. Se necesita crear una nueva Window, y esto se consigue haciendo clic en el botón Window del PowerPanel lo que abrirá la ventana de selección de Windows y posteriormente haciendo clic en el botón New -si lo que se desea es abrir una Window existente solo haga doble clic en la Window deseada-.



Ilustración 86 Selector de ventanas.

Esta acción crea una nueva ventana vacía. Seleccione la opción DataWindow del menú Controls.

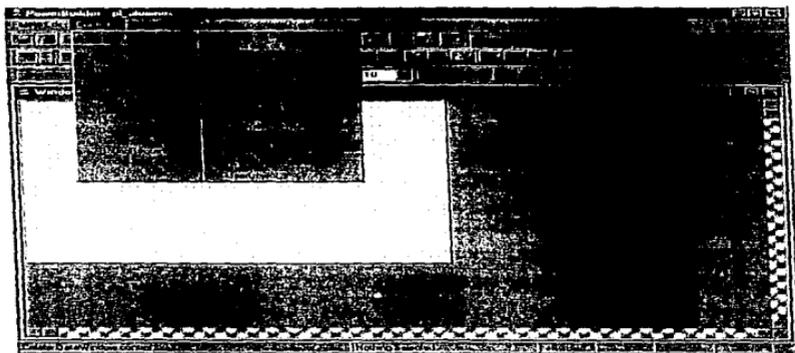


Ilustración 87 Opciones del menú Controls.

El aspecto del cursor cambiará al de una cruz cuadrada. Situando el cursor en la esquina superior izquierda de la posición en donde se desea situar el control DataWindow y haciendo clic, aparece el correspondiente control DataWindow.

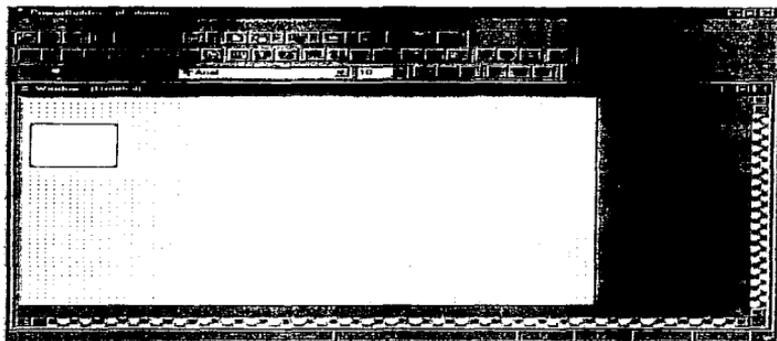


Ilustración 88 Painter de Window con la sección del control DataWindow creada.

Situando el cursor dentro del área del control, haciendo clic y manteniendo pulsado el botón izquierdo del ratón, es posible arrastrar el control hasta la posición deseada.

Situando el cursor en alguna de las esquinas u orillas del control, el cursor cambiará de aspecto a una flecha bidireccional, y haciendo clic en el botón izquierdo del ratón y manteniéndolo pulsado, puede modificar el tamaño del control al arrastrar el ratón.

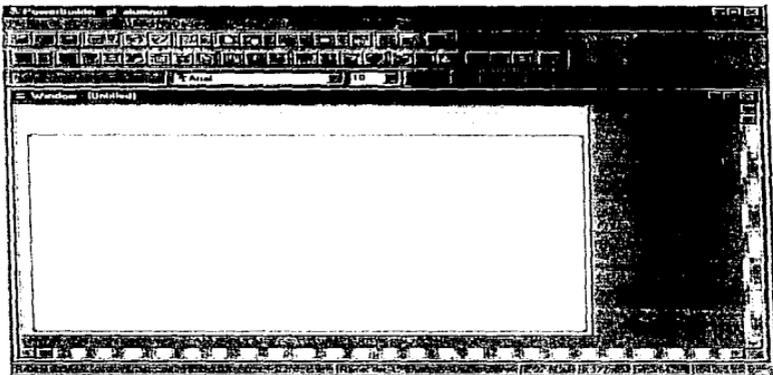


Ilustración 89 Definición de tamaño y posición de la sección del control DataWindow.

Para asociar un objeto DataWindow, se hace clic con el botón derecho del ratón dentro del área que ocupa el control, y entonces se selecciona la opción Properties del menú.



Ilustración 90 Selección de las propiedades de la sección del control DataWindow.

En las propiedades del control se escribe el nuevo nombre del control dentro de la opción Name, en la opción DataWindow Object Name se escribe el nombre del objeto DataWindow. Posteriormente se hace clic en el botón OK.



Ilustración 91 Ventana de propiedades del control DataWindow.

Una vez hecho esto se asocia el objeto DataWindow d\_alumnos al correspondiente control, y de esta manera ya es visible en el painter de Window.

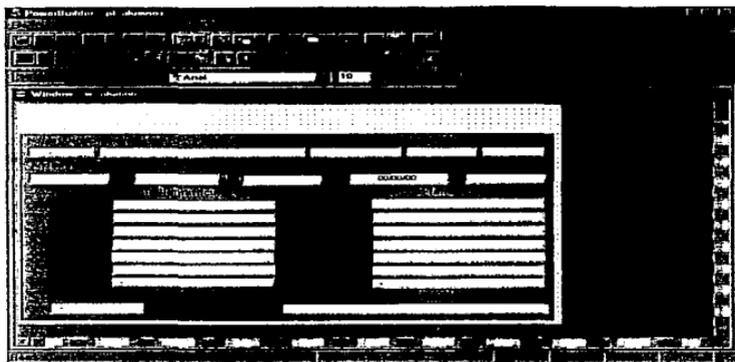


Ilustración 92 Painter de Window con el control DataWindow creado.

Para agregar un texto a la ventana se actúa de la misma manera que en el painter de DataWindow. Eligiendo el botón de texto del PainterBar y haciendo clic en el lugar donde se desee colocar dicho texto y escribiendo la leyenda. También cambiar el color de frente o fondo es el mismo procedimiento.

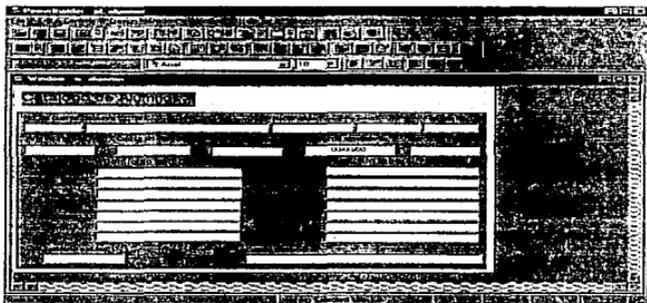


Ilustración 93 Inserción de texto en el Painter de Window.

Para cambiar el color de fondo de la ventana al mismo que tiene el control DataWindow se elige la opción Propiedades de menú que resulta de hacer clic con el botón derecho del ratón y cambiando el color en la opción Window Color en la ventana que aparece.



Ilustración 94 Propiedades de la ventana donde se selecciona el color de esta.

Con esta acción la ventana cambia al color seleccionado.

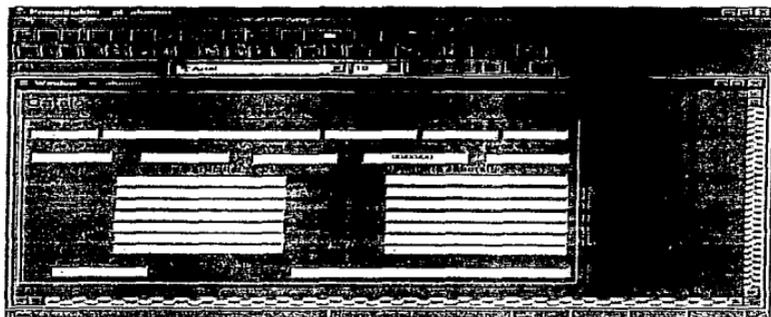


Ilustración 95 Ventana con el color definido.

Se necesita incluir botones de control en la ventana para poder manipular la información grabada en la tabla de alumnos. Haciendo clic en el ícono CommandButton la forma del cursor cambia a una cruz cuadrada.

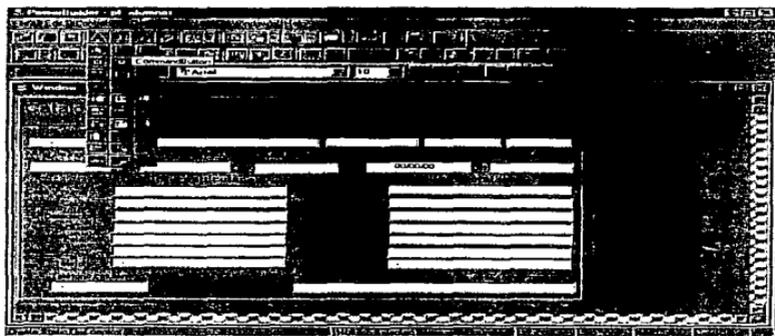


Ilustración 96 Opción CommandButton del PainterBar.

Haciendo clic aparece el botón en el lugar de la ventana donde se desea.

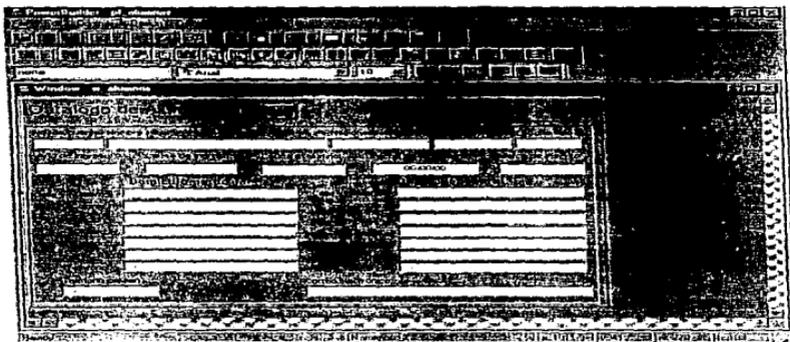


Ilustración 97 Botón recién creado dentro de la ventana.

Colocando el cursor dentro del área del control y presionando el botón derecho del ratón se elige la opción Properties. Dentro de la ventana de CommandButton en la opción Name se escribe el nombre con el que se va a identificar el control, y en la opción Text el texto que va a aparecer en el botón -el carácter "&" se utiliza para subrayar la letra que precede con el fin de activar el acceso rápido-.

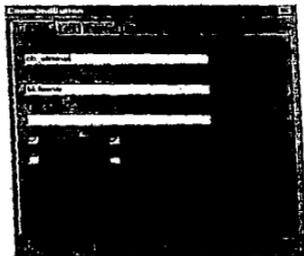


Ilustración 98 Ventana de propiedades del CommandButton.

Después de haber hecho esto se hace clic en el botón OK y ya en el painter de Window los cambios quedan registrados.

Para agregar más botones de control a la ventana basta con presionar el botón derecho del ratón dentro del área del botón ya creado y seleccionar la opción Duplicate del menú.

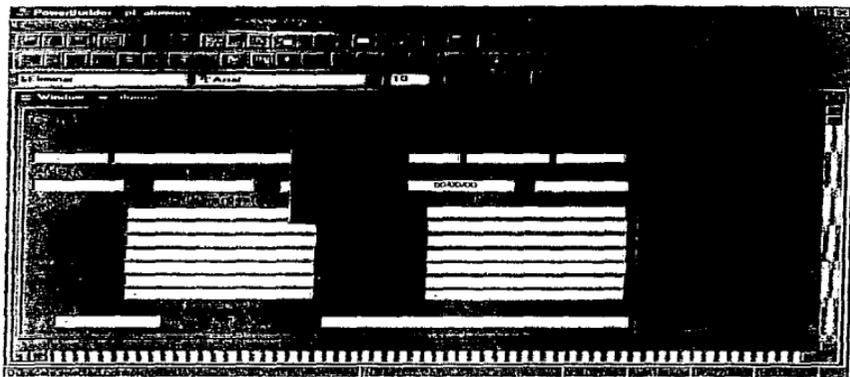


Ilustración 99 Opción para duplicar el CommandButton.

Al hacer esto aparece otro botón, el cual se coloca en el lugar que el usuario designe para él, esto se hace ubicando el cursor en el área del nuevo botón y haciendo clic y manteniéndolo presionado se arrastra el objeto hasta dicho lugar.

Se actúa de la misma forma para crear los demás botones de control que se necesiten. El procedimiento para modificar el nombre y el desplegado de cada botón es el mismo que se aplicó para el primer botón. De tal manera que la ventana queda como en la **Ilustración 100**.

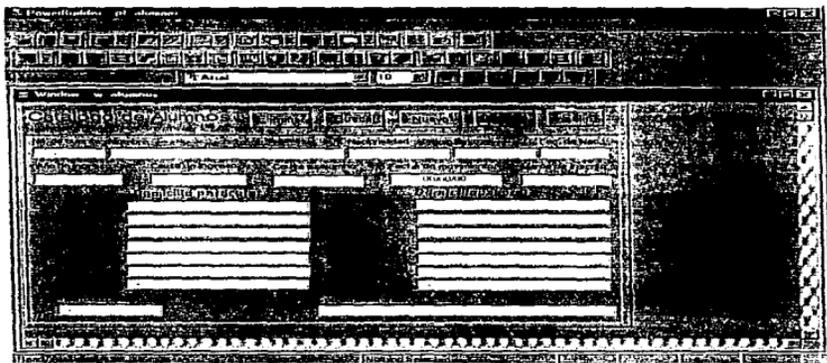


Ilustración 100 Botones creados y definidos en la ventana.

Colocando el cursor en el área del botón Nuevo y haciendo clic en el botón derecho del ratón se elige la opción Script del menú. En ese momento aparece el painter del script el cual está vacío. Se escriben las siguientes líneas:

```
dw_alumnos.reset()
```

```
dw_alumnos.insertrow(0)
```

La primera línea borra cualquier dato que aparezca en el control DataWindow y la segunda reemplaza con un registro en blanco. Se cierra el painter del Script guardando las modificaciones, en ese momento PowerBuilder ejecuta la compilación de las líneas del Script, si no hay errores, el sistema regresa al painter de Window, si hay errores, el sistema pide corregirlos. De manera que al hacer clic en el botón Nuevo se limpian las columnas del control DataWindow y lo deja listo para otro registro.

Editando ahora el script del botón Eliminar se escriben las siguientes líneas:

```
dw_alumnos.deleteRow(0)
```

```
dw_alumnos.update()
```

```
commit;
```

La primera línea borra el registro y la segunda lo actualiza, lo que hace la tercera línea es indicar que la transacción ha sido completada y que se registre el cambio definitivamente. El botón Eliminar borra registros de la tabla Alumnos.

En el script del botón Guardar se escriben las siguientes líneas:

```
dw_alumnos.update()
```

```
commit;
```

La primera línea actualiza el alta o los cambios hechos en el registro y la segunda confirma la transacción y la hace definitiva.

Para el botón Salir solo se escribe la siguiente línea:

```
close (parent)
```

Esta línea cierra la aplicación del objeto ascendente del botón salir, en este caso, la ventana `w_alumnos`, durante la ejecución.

Colocando ahora el cursor en un lugar libre de objetos de la ventana y editando su script, de la misma manera que en los botones, se escriben las siguientes líneas:

```
dw_alumnos.settrans(sqlca)
```

```
cb_nuevo.triggerevent(Clicked!)
```

La primera sentencia asocia el objeto transacción por defecto; `SQLCA`, al objeto `DataWindow dw_alumnos`. La segunda sentencia realiza un clic automático en el botón Nuevo, ejecutando las instrucciones que contiene este.

Colocando el cursor en cualquier lugar del área del control DataWindow y editando su script escribimos las siguientes líneas:

```
string ls_no_de_cuenta

if GetColumnName()=="no_de_cuenta" then

    ls_no_de_cuenta = GetText()

    retrieve(ls_no_de_cuenta)

    if RowCount()<1 then

        cb_nuevo.triggerevent(clicked!)

        setColumn("no_de_cuenta")

        SetText(ls_no_de_cuenta)

        setColumn(1)

    end if

end if
```

Este procedimiento inicia declarando una variable `-ls_no_de_cuenta-`, posteriormente la decisión exige sea capturado algún dato en la columna `no_de_cuenta`, una vez dentro de esta decisión el dato es asignado a la variable declarada y se hace una recuperación de dicho registro. De no haberse recuperado ningún registro es como se entra a la segunda decisión, en la cual se hace un clic automático en el botón Nuevo que limpia totalmente el control DataWindow, por lo que las siguientes líneas asignan el valor de la variable a la columna `no_de_cuenta` y las demás columnas quedan liberadas para poder ser escritas en ellas.

El único botón que falta por describir es el llamado Buscar, la línea que se escribe en su script es la siguiente:

`open(w_busca_alumno)`

La razón por la cual este botón se dejó al último es que la sentencia asociada a dicho objeto abre otra ventana llamada `w_busca_alumno`. Los procedimientos de creación de esta ventana, desde el objeto `DataWindow` hasta la misma ventana, son similares a los utilizados en la ventana `w_alumnos` por lo que solo se describirán los controles aún no mencionados. La ventana que se solicita abrir se observa en la **Ilustración 101**.

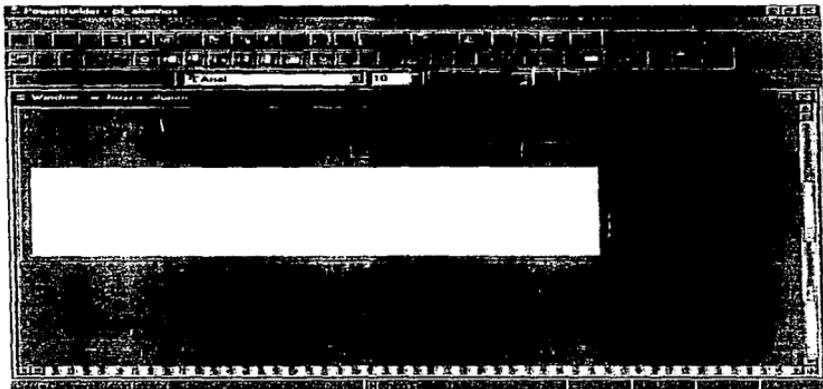


Ilustración 101 Ventana `w_busca_alumno`.

Se hace clic en el icono `SingleLineEdit` del `PainterBar`.

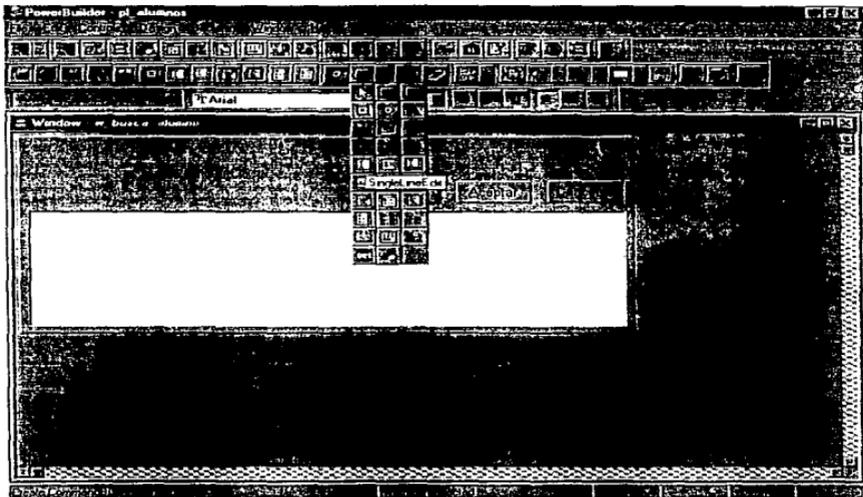


Ilustración 102 Opción SingleLineEdit del PainterBar.

Se coloca el cursor en la ventana y se hace clic. Se modifica la posición y el tamaño, y en la parte superior se le agrega el texto Buscar a:

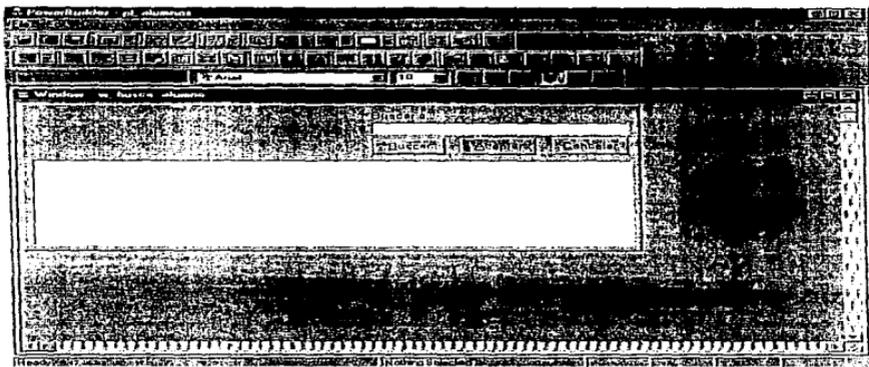


Ilustración 103 SingleLineEdit ubicado en la ventana.

Es necesario incluir dos controles más a la ventana. Estos se obtiene haciendo clic en el icono GroupBox del PainterBar.

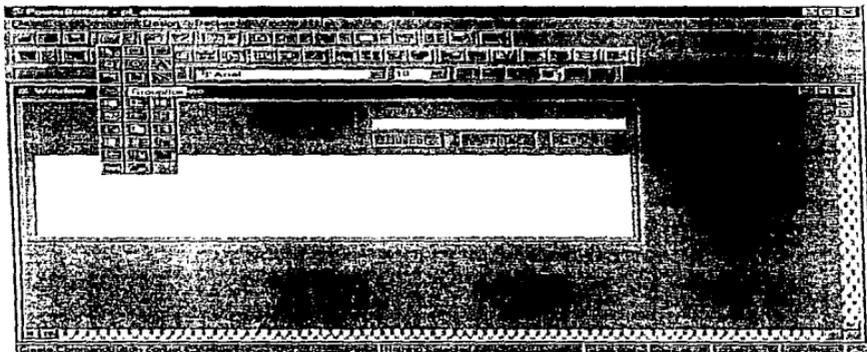


Ilustración 104 Opción GroupBox del PainterBar.

Posteriormente se hace clic en el lugar de la ventana donde se quiera colocar el GroupBox, se modifica el tamaño y la posición. Ahora se ubica el cursor en el área del GroupBox y se hace clic con el botón derecho del ratón eligiendo la opción Duplicate del menú, creándose de esta manera otro control igual que se coloca en la posición que se quiera designar para él.

En la ventana de propiedades de cada uno de los controles se define el nombre del objeto y el texto a desplegar.

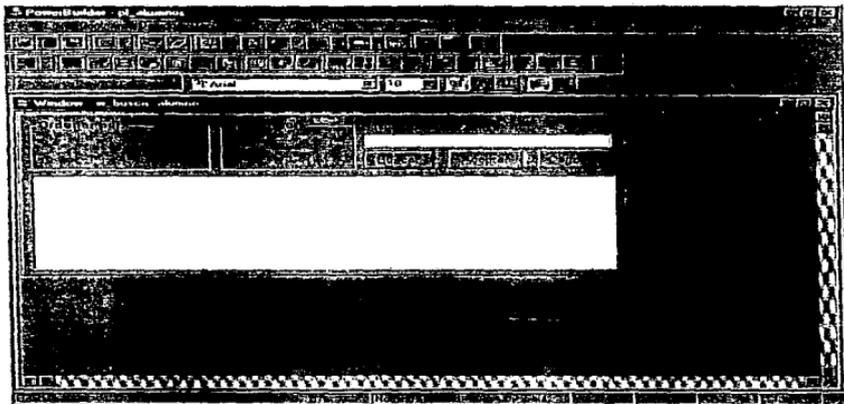


Ilustración 105 GroupBox definidos en la ventana.

Ahora se hace clic el icono RadioButton del PainterBar .

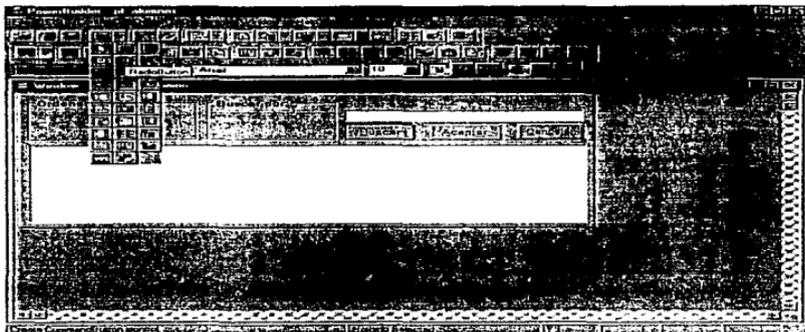


Ilustración 106 Opción RadioButton del PainterBar.

Se coloca el cursor en el GroupBox llamado Ordenado por y se hace clic, se crea otro RadioButton para el mismo GroupBox y otros dos para el otro GroupBox. En la ventana de propiedades de cada uno de los RadioButton se cambia el nombre y el texto a desplegar.

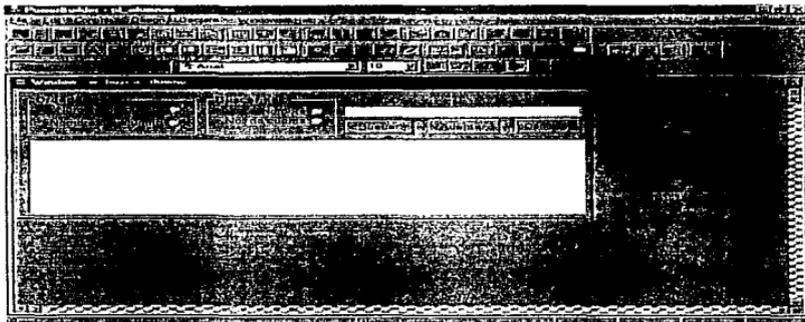
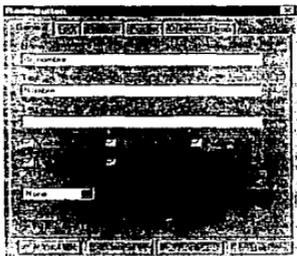


Ilustración 107 RadioButton definidos en la ventana.

Colocando el cursor en el RadioButton llamado Nombre se presiona el botón derecho del ratón y se elige la opción Properties del menú. Una vez dentro de la ventana de propiedades del control RadioButton se hace clic en la opción Checked y se presiona el botón OK.



**Ilustración 108** Ventana de propiedades del RadioButton (b) nombre con la opción Checked activada. Al hacer esto queda la opción Nombre marcada, y cada que se inicie la ejecución de la ventana w\_busca\_alumno este botón así aparecerá, hasta que se haga clic en el otro botón del GroupBox.



**Ilustración 109** Painter de Window con el RadioButton Nombre marcado.

Para poder completar esta ventana es necesario definir los procedimientos asociados a cada objeto. Colocando el cursor en el área descubierta de la ventana (zona punteada) se abre el painter del script, en el cual se escribe la siguiente línea:

```
d_busca_alumno.settrans(sqlca)
```

Esta sentencia asocia el objeto transacción por defecto; SQLCA, al objeto DataWindow `d_busca_alumnos`.

Ahora se edita el script del control DataWindow y se escribe el siguiente procedimiento:

```
if RowCount()>0 then
  if IsValid(w_alumnos) then
    w_alumnos.dw_alumnos.retrieve(GetItemString(getrow(),"no_de_cuenta"))
  elseif IsValid(w_rectificacion) then
    w_rectificacion.dw_rectificacion.Object.no_de_cuenta[1,1]=GetItemString(getrow(),"no_de_cuenta")
  elseif IsValid(w_titulacion) then
    w_titulacion.dw_tesis.Object.no_de_cuenta[1,1]=GetItemString(getrow(),"no_de_cuenta")
    w_titulacion.dw_tesis.Object.nom_alumno.text=GetItemString(getrow(),"nombre_alumno")
  end if
end if
close (parent)
```

La primera decisión de este procedimiento requiere de que exista por lo menos un registro en la tabla de Alumnos. Una vez adentro, existen tres decisiones más, la primera regresa el registro del número de cuenta seleccionado al control DataWindow `dw_alumnos` de la ventana `w_alumnos` al hacer doble clic en algún registro, si dicha ventana se encuentra abierta en ese momento. La segunda y tercera decisiones regresan campos específicos si es

que las ventanas por las que preguntan dichas decisiones están abiertas. Este procedimiento se creó de esta manera para que la ventana `w_busca_alumno` pueda ser llamada para recuperar datos por ventanas diferentes.

En el script asociado al botón `Buscar` se escriben las siguientes líneas:

```
d_busca_alumno.reset()

if rb_nombre.Checked then

    d_busca_alumno.retrieve(sle_1.text,"")

else

    d_busca_alumno.retrieve("",sle_1.text)

end if

if d_busca_alumno.RowCount()<1 then

    MessageBox("Error ", "No se encontro ningun registro")

end if
```

La primera línea limpia el control `DataWindow`. La primera decisión pregunta si el `RadioButton` `rb_nombre` tiene la condición `Checked`, de ser afirmativo se despliegan en el control `DataWindow` los registros encontrados, los cuales fueron buscados de acuerdo con lo que se capturó en el `SingleLineEdit`, ya sea parte del nombre o el nombre completo. De no cumplirse esta condición, significa que el `RadioButton` que se encuentra en estado de `Checked` es el de número de cuenta y que va a buscar por la parte o número de cuenta completo que se capturó en la `SingleLineEdit`. La segunda decisión revisa si se han encontrado registros con estos parámetros de búsqueda, y de no ser así, despliega un mensaje de error.

Por lo que es posible hacer que se despliegue un conjunto de registros menor o igual al total de la tabla Alumnos solo escribiendo una parte o todo el nombre o número de cuenta en el SingleLineEdit, es debido a la cláusula Where definida en el diseño del objeto DataWindow. Esta selección declara dos variables, en la primera posición se limita al campo nombre\_alumno a lo que se le asigne a la variable Nom\_bus, en la segunda posición se limita al campo no\_de\_cuenta a lo que se le asigne a la variable Num\_etc. Las variables se completan con el carácter ""a" por si solamente se captura una parte del campo solo se busca esa parte del campo indistintamente del resto.

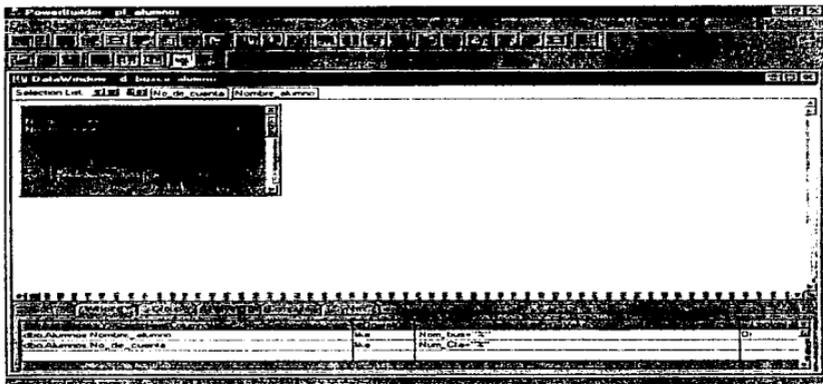


Ilustración 110 Opción Where de la caja de herramientas de SQL del objeto DataWindow d\_busca\_alumno.

En el script del botón Aceptar se escribe la siguiente línea:

```
d_busca_alumno.triggerevent(doubleclicked!)
```

Esta línea toma el registro que está marcado en el control DataWindow y ejecuta un doble clic sobre él.

En el script asociado al botón cancelar se escribe la siguiente línea:

```
close (parent)
```

Esta línea cierra la aplicación del objeto ascendente del botón **Cancelar**, en este caso, la ventana `w_busca_alumno`, durante la ejecución.

En los scripts asociados a los **RadioButtons** **Número de cuenta** y **Nombre del alumno** se escriben las siguientes líneas:

**Número de cuenta:**

```
if checked then  
    d_busca_alumno.setsort("no_de_cuenta")  
    d_busca_alumno.sort()  
end if
```

**Nombre del alumno:**

```
if checked then  
    d_busca_alumno.setsort("nombre_alumno")  
    d_busca_alumno.sort()  
end if
```

En el primer procedimiento se pregunta si está marcado el **RadioButton** **Número de cuenta**, de ser así, se realiza un ordenamiento por este campo.

En el segundo procedimiento se pregunta si está marcado el RadioButton Nombre del alumno, de ser así, se realiza un ordenamiento por este campo.

Para salvar esta o cualquier ventana se elige la opción Save del menú File y se escribe el nombre de la ventana.

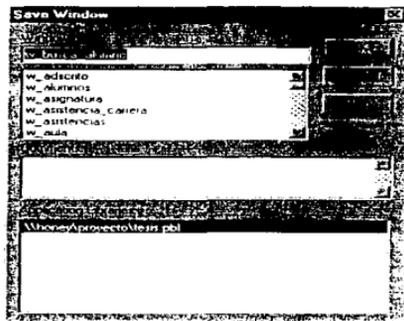


Ilustración 111 Ventana para salvar la Window creada.

Para poder ejecutar esta aplicación se necesita regresar al painter de aplicación que se eligió con anterioridad. Se hace clic en el icono Application para situarnos en el painter y ahora se hace clic en el icono Script del PainterBar. Una vez en el script de la aplicación se escribe la siguiente línea:

```
Open (w_alumnos)
```

Al momento de salvar el script es compilado indicando si existen errores. Ya salvado y cerrado el script se elige el icono Run del PowerBar, al hacer clic sobre él se inicia la ejecución de la aplicación creada.

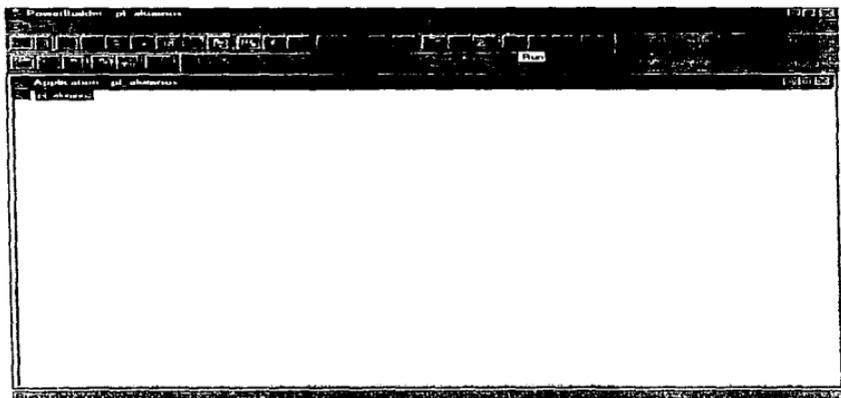


Ilustración 112 Opción Run del PowerBar.

# CONCLUSIONES

Actualmente, la tecnología se ha estado desarrollando de manera impresionante y aunque la industria de computación no está incrementándose en la misma proporción que en años anteriores, cada vez más van surgiendo equipos de cómputo con gran capacidad y velocidad de procesamiento. El área de comunicaciones ha crecido de tal manera que ahora cualquier persona desde un determinado punto se puede comunicar con muchas otras vía red, de esta misma forma se pueden compartir recursos e información, evitando así la redundancia de datos y la pérdida de información. La adecuada administración o uso de los recursos hace óptimo y amigable el desempeño de los sistemas tanto educacionales como de la industria de las grandes empresas.

La arquitectura cliente/servidor ha abierto la pauta a muchas empresas para migrar sus sistemas, con la firme intención de poder mejorar su desempeño laboral.

El nivel de seguridad actual en los sistemas es una gran ventaja ya que controla el acceso a información de tipo confidencial y además, se restringen los privilegios de acceso y actualización a cualquier base de datos.

En la Facultad de Derecho se maneja mucha información de los estudiantes y el personal docente (domicilios, contratos, exámenes profesionales, exámenes de oposición, asistencia de personal académico, etc.) esta información hasta el momento se maneja de manera aislada en computadoras personales o, en algunos casos, solo en papel. La información se procesa de manera independiente en cada una de las áreas y para que las demás áreas estén actualizadas con dicha información tienen que esperar a ser enteradas para poder poner al tanto su archivo independiente. Debido a la lentitud de esta forma de trabajar y el margen de error que esto provoca, fue necesario concebir un sistema integral que uniera todas estas áreas independientes con una sola fuente de datos y así garantizar la integridad de la información y aumentar la velocidad de los trámites.

Esta tesis fue creada debido a esas limitaciones. Las necesidades que se manifiestan en algunas de sus páginas son el reflejo de las mismas necesidades que tiene esta Facultad. Se investigaron estas necesidades, se analizaron, se les dio el trato y la prioridad que merecían,

se resolvieron dentro de las capacidades técnicas de los equipos y las limitaciones de tiempo, y se llegó a un trabajo con la capacidad suficiente para manejar toda la información de forma adecuada y subsanar las limitaciones administrativas que la falta de un sistema de apoyo provocan.

En un principio se planteó la necesidad de poder tener un sistema que conjugara la capacidad de tratar los datos de una manera organizada y confiable.

Dadas la situación actual en la cual se encontraban los sistemas de la Facultad de Derecho y por nuestra experiencia dentro de la misma institución, todo indicaba a que se debía realizar una análisis profundo para encontrar un esquema que resolviera el problema planteado en este trabajo.

Sin embargo dado que se carecía de todo tipo de documentación y antecedentes que apoyaran en algo el análisis del sistema, nos dimos a la tarea de llevar a cabo de manera práctica el uso de una metodología. Sobre el tema de modelado de sistema existe mucha información, en este trabajo se utilizó la metodología de Yourdon.

El desarrollo del esquema lógico de datos se realizó en base a la metodología mencionada, por el conocimiento de ésta, pero dado a lo extenso de la misma solo se utilizaron algunos aspectos y se complementaron con el uso de herramientas especializadas, ambos puntos fueron complementados en gran medida con las bases teóricas.

De ahí que la implementación física de un esquema de datos lógico se puede resolver de manera más práctica cuando se cuenta con herramientas de desarrollo que faciliten su creación, sin embargo no libera al analista de su responsabilidad.

En la actualidad se requiere de mayor rapidez para desarrollar aplicaciones, dado que el tiempo del ciclo de vida de un sistema se ve afectado por los cambios constantes que demanda una sociedad tan dinámica. Debido a esta necesidad de respuesta rápida y confiable se han puesto a disposición de los analistas y desarrolladores de sistemas una cantidad tan variada de herramientas visuales que facilitan y apoyan la implementación y desarrollo de sistemas completos en un lapso de tiempo relativamente corto.

Con la tecnología de la programación orientada a objetos, que permite la reutilización de código, el polimorfismo, los eventos, métodos, herencia y encapsulación, hace posible que surjan nuevas formas y maneras de analizar los problemas y en consecuencia presentar la solución.

Con el presente trabajo, logramos adquirir grandes experiencias al conjuntar tres tipos de programas diferentes: un sistema operativo (Unix, Sun Solaris 2.5.1), un manejador de bases de datos (Sybase system 11) y un desarrollador de aplicaciones gráfico (PowerBuilder 5.0), todos ellos en un ambiente de red. El primero corriendo casi en cualquier plataforma de hardware siguiendo la filosofía de los sistemas abiertos, el segundo, un manejador que tiene la capacidad para interactuar con otros manejadores de bases de datos, y el último que con su interfaz de usuario gráfica hace mas amigable la interacción con el usuario y que permite la reutilización de las aplicaciones en un ambiente multiplataforma.. También utilizamos herramientas de apoyo en el análisis y desarrollo del modelo de datos. Finalmente al conjuntar todos éstos elementos podemos decir que realizamos un ejemplo de interoperabilidad.

Es evidente que logramos integrar en una aplicación varios conceptos tales como: arquitectura cliente/servidor, redes, ambientes gráficos y bases de datos. Sin embargo esto no se hubiera logrado sin contar con los conocimientos adquiridos en la Facultad de Ingeniería los cuales fueron la base fundamental a partir de la cual, logramos resolver un problema con nuevos esquemas de modernización, rapidez y automatización de procedimientos.

Pero los logros no solo fueron a nivel de integración de conocimientos ya que nuestra tarea no se limita solo a poder utilizar las diferentes tecnologías y esquemas de desarrollo de sistemas, sino a lograr que estas herramientas sirvas como medio para lograr un mejor avance en las actividades o funciones de una empresa.

Tales avances se ven reflejados en lo que hoy en las actividades de las áreas administrativas de la Facultad de Derecho, pues se tiene un mejor control de acceso y actualización de la información, traduciéndose en un mejor desempeño de las funciones y mejor toma de

de igiones, por ejemplo se tiene mayor control sobre la asistencia de los profesores a las asignaturas de la carrera, se cuenta con un banco de datos del personal académico, confiable y actualizado. Se mejoró la asignación de sinodales para los exámenes profesionales, se disminuyó la carga de trabajo en algunas áreas por evitar la duplicación de información; se acabó con la pérdida de información por carecer de un esquema de respaldo; se consiguieron simplificar procesos redundantes y realizar de manera más rápida y amigable actividades que antes resultaban tediosas y repetitivas.

Esto solo es el principio pues quedan algunas actividades que aún no han sido actualizadas; sin embargo las pautas y los principios están presentes.

# BIBLIOGRAFIA

- [ADA94] Adaptec  
Tecnología de I/O  
Adaptec, Inc., 1994
- [BER92] Alex Berson  
Client/Server Architecture  
McGraw-Hill, 1992
- [BOC94] Barbara Bochensky  
Implementing Production-Quality Client/Server Systems.  
Wiley, 1994
- [BOO94] Grady Booch  
Object Oriented Analysis and Design with Applications  
Benjamin/Cummings Publishing, 1994
- [BRI94] Brian Butler  
The Great Leap Forward (SQL Databases)  
PC Magazine vol. 13 no. 17, 1994
- [COD90] E. F. Codd  
The Relational Model for Database Management  
Addison-Wesley, 1990
- [DAT81] C. J. Date  
An Introduction to Database Systems  
Addison-Wesley, 1981
- [DAW93] Dawna Travis Dewire  
Client/Server Computing  
McGraw-Hill, 1993
- [DEI90] Harvey M. Deitel  
Operating Systems  
Addison-Wesley, 1990
- [DEM79] Tom de Marco  
Structured Analysis and Systems Specification  
Prentice-Hall, 1979
- [FRE93] Alan Freedman  
Diccionario de Computación  
McGrawHill, 1993

## Bibliografía

---

- [GAN78] Chris Gane and Trish Sarson  
Structured Systems Analysis  
Prentice-Hall, 1978
- [GAR95] Jeff Garbus, David Solomon, & Brian Tretter  
SYBASE DBA Survival Guide  
Sams Publishing, 1995.
- [KOR88] Henry F. Korth and Abraham Silberschatz  
Database Systems Concepts  
McGraw-Hill, 1988
- [MAR67] James Martin  
Design of Real-Time Computer Systems  
Prentice-Hall, 1967
- [MAR77] James Martin  
Organización de las Bases de Datos  
Prentice-Hall, 1977
- [MAR89] James Martin  
Local Area Networks  
Prentice-Hall, 1989
- [MAR92] James Martin and James J. Odell  
Object Oriented Analysis and Design  
Prentice-Hall, 1992
- [ORF95] Robert Orfali, Dan Harkey y Jeri Edwards  
Essential Client/Server Survival Guide  
Byte, Abril 1995
- [PAG88] Meilir Page-Jones  
The Practical Guide to Structured Systems Design  
Prentice-Hall, 1988
- [PAU85] Paul Ward and Steve Mellor  
Structured Development for Real-Time Systems  
Yourdon Press, 1985
- [POS96] Powersoft  
Connecting to your Database  
Powersoft, 1996

- [POW96] Powersoft  
Getting Started PowerBuilder 5.0  
Powersoft, 1996
- [RAN96] Rankins Ray and Garbus Jeffrey R.  
Sybase SQL Server 11  
Sams Publishing, 1996
- [SAN93] Sander C. Ronald  
Local Area Networking and LAN Integration  
Sander Group Inc., 1993
- [SHE95] Sheldon Tom  
LAN TIMES Encyclopedia of Networking  
McGraw-Hill, 1995
- [SHL95] Sheldon Tom  
LAN TIMES Guide to interoperability  
McGraw-Hill, 1995
- [SIN92] Sinha Alok  
Client/server computing  
Communications of de ACM, julio 1992
- [SUN96] Sun  
La red Internet  
Sun, Febrero 1996
- [TAN91] Tanembaum Andrew S.  
Redes de Ordenadores  
Prentice-Hall, 1991
- [VAU94] Larry T. Vaughn  
Client/Server System Design and Implementation  
McGraw-Hill, 1994
- [WOO95] Wood Charles  
Using PowerBuilder 4  
Que, 1995
- [YOU89] Edward Yourdon  
Modern Structured Analysis  
Prentice-Hall, 1989

**APENDICE A**  
**Código de la Base de**  
**Datos**

```

CREATE DEFAULT Def_carrera
AS 21
go

CREATE DEFAULT Def_Falta
AS 2
go

CREATE DEFAULT Def_Grado
AS 'LIC'
go

CREATE DEFAULT Def_Nac
AS 'MEX'
go

CREATE DEFAULT Def_Plan
AS 'N'
go

CREATE DEFAULT Def_Plantel
AS 7
go

CREATE TABLE Categorías
(Clave_de_categoria char(8) NOT NULL,
Categoría char(40) NULL,
Nivel char(25) NULL,
Sueldo_numero money NULL,
Sueldo_en_letra char(50) NULL,
PRIMARY KEY (Clave_de_categoria)
)
go

exec sp_primarykey Categorías,
Clave_de_categoria
go

CREATE TABLE Seminario
(Clave_de_seminario tinyint NOT NULL,
Nombre_del_seminario char(25) NULL,
Director_del_seminario char(30) NULL,
PRIMARY KEY (Clave_de_seminario)
)
go

exec sp_primarykey Seminario,
Clave_de_seminario
go

CREATE TABLE Profesores
(Rfc char(13) NOT NULL,
Nombre varchar(20) NULL,
Apellido_paterno varchar(25) NULL,
Apellido_materno varchar(25) NULL,
Grado char(4) NULL,
Nacionalidad char(3) NULL,
Sexo char(1) NULL,
Estado_civil char(1) NULL,
Fecha_de_nacimiento smalldatetime NULL,
Cedula_profesional char(9) NULL,
Profesor_activo char(1) NULL,
Tipo_de_profesor char(2) NULL,
Observaciones varchar(255) NULL,
Casa_calle_y_numero varchar(30) NULL,
Casa_colonia varchar(30) NULL,
Casa_delegacion_o_poblacion char(18) NULL,
Casa_ciudad char(20) NULL,
Casa_estado tinyint NULL,
Casa_CP char(5) NULL,
Casa_telefono char(13) NULL,
Trabajo_calle_y_numero varchar(30) NULL,
Trabajo_colonia varchar(30) NULL,
Trabajo_delegacion_o_poblacion char(18) NULL,
Trabajo_ciudad char(20) NULL,
Trabajo_estado tinyint NULL,
Trabajo_CP char(5) NULL,
Trabajo_telefono char(13) NULL,
Telefono_celular char(13) NULL,
Correo_electronico varchar(30) NULL,
Fecha_ingreso smalldatetime NULL,
PRIMARY KEY (Rfc)
)
go

exec sp_primarykey Profesores,
Rfc
go

exec sp_hindefault Def_Grado, 'Profesores.Grado'
exec sp_hindefault Def_Nac, 'Profesores.Nacionalidad'
go

CREATE TABLE Aulas
(Clave_aula char(8) NOT NULL,
Nombre_del_aula varchar(40) NULL,
Capacidad tinyint NULL,
PRIMARY KEY (Clave_aula)
)
go

exec sp_primarykey Aulas,
Clave_aula
go

CREATE TABLE Alumnos
(No_de_cuenta char(9) NOT NULL,
Nombre_alumno varchar(40) NULL,
Plantel tinyint NULL,
Carrera tinyint NULL,
Año_de_ingreso char(4) NULL,
Nacionalidad char(3) NULL,
Causa_de_ingreso tinyint NULL,
Causa_exalumno tinyint NULL,
Sexo char(1) NULL,
Fecha_de_Nacimiento smalldatetime NULL,
Fecha_de_movimiento smalldatetime NULL,
Marca_de_inscrito char(1) NULL,
ACasa_calle_y_numero varchar(30) NULL,
ACasa_colonia varchar(30) NULL,

```

## Apéndice A

```

ACasa_delegacion_o_poblacion char(18) NULL,
ACasa_ciudad varchar(20) NULL,
ACasa_estado tinyint NULL,
ACasa_CP char(5) NULL,
ACasa_telefono char(13) NULL,
ATrabajo_calle_y_numero varchar(30) NULL,
ATrabajo_colonia varchar(30) NULL,
ATrabajo_delegacion_o_poblacion char(18) NULL,
ATrabajo_ciudad varchar(20) NULL,
ATrabajo_estado tinyint NULL,
ATrabajo_CP char(5) NULL,
ATrabajo_telefono char(13) NULL,
ATelefono_celular char(13) NULL,
ACorreo_electronico varchar(30) NULL,
PRIMARY KEY (No_de_cuenta)
)
go

exec sp_primarykey Alumnos,
No_de_cuenta
go

exec sp_bindefault Def_Plantel, 'Alumnos Plantel'
exec sp_bindefault Def_Carrera, 'Alumnos Carrera'
go

CREATE TABLE Asistencias
(Rfc char(13) NOT NULL,
Clave_de_grupo char(13) NOT NULL,
Fecha_de_falta smalldatetime NOT NULL,
Tipo_de_falta tinyint NULL,
PRIMARY KEY (Rfc, Clave_de_grupo,
Fecha_de_falta),
FOREIGN KEY (Clave_de_grupo)
REFERENCES Grupos,
FOREIGN KEY (Rfc)
REFERENCES Profesores
)
go

exec sp_primarykey Asistencias,
Rfc,
Clave_de_grupo,
Fecha_de_falta
go

exec sp_bindefault Def_Falta, 'Asistencias Tipo_de_falta'
go

CREATE TABLE Periodo_semestral
(Periodo_sem char(5) NOT NULL,
Fecha_inicio smalldatetime NOT NULL,
Fecha_fin smalldatetime NOT NULL,
PRIMARY KEY (Periodo_sem)
)
go

exec sp_primarykey Periodo_semestral,
Periodo_sem
go

```

```

CREATE TABLE Contratos
(Rfc char(13) NOT NULL,
Clave_de_categoria char(8) NOT NULL,
Clave_de_grupo char(13) NOT NULL,
Codigo_programatico char(14) NULL,
No_de_oficio smallint NULL,
No_de_registro_de_plaza char(18) NULL,
No_de_exposante smallint NULL,
Personal char(18) NULL,
Movimiento tinyint NULL,
Causa tinyint NULL,
Tipo_de_profesor_en_contrato tinyint NULL,
No_de_relacion char(18) NULL,
No_de_horas_teoricas decimal(2,2) NULL,
No_de_horas_practicas decimal(2,2) NULL,
Fecha_de_inicio smalldatetime NULL,
Fecha_de_fin smalldatetime NULL,
Sustituye_a char(40) NULL,
Observaciones_en_contrato char(100) NULL,
PRIMARY KEY (Rfc, Clave_de_categoria, Clave_de_grupo),
FOREIGN KEY (Clave_de_grupo)
REFERENCES Grupos,
FOREIGN KEY (Clave_de_categoria)
REFERENCES Categorias,
FOREIGN KEY (Rfc)
REFERENCES Profesores
)
go

exec sp_primarykey Contratos,
Rfc,
Clave_de_categoria,
Clave_de_grupo
go

CREATE TABLE Adscritos_a_seminario
(Rfc char(13) NOT NULL,
Clave_de_seminario tinyint NOT NULL,
PRIMARY KEY (Rfc, Clave_de_seminario),
FOREIGN KEY (Rfc)
REFERENCES Profesores,
FOREIGN KEY (Clave_de_seminario)
REFERENCES Seminario
)
go

exec sp_primarykey Adscritos_a_seminario,
Rfc,
Clave_de_seminario
go

CREATE TABLE Exámenes_de_oposición
(Periodo char(5) NOT NULL,
Clave_de_asignatura char(4) NOT NULL,
Clave tinyint NOT NULL,
Fecha_de_convocatoria smalldatetime NULL,
Fecha_de_diciamen smalldatetime NULL,
No_de_plazas tinyint NULL,
PRIMARY KEY (Periodo, Clave_de_asignatura),

```

```

FOREIGN KEY (Clave_de_asignatura)
REFERENCES Asignaturas.
FOREIGN KEY (Clave)
REFERENCES Comision
)
go

exec sp_primarykey Exámenes_de_oposición,
Periodo,
Clave_de_asignatura
go

CREATE TABLE Tesis
(No_registro int NOT NULL,
Rfc char(13) NOT NULL,
No_de_cuenta char(9) NOT NULL,
Fecha_de_registro smalldatetime NULL,
Tema varchar(200) NULL,
Aprobacion_de_tema bit NOT NULL,
Fecha_Examen smalldatetime NULL,
Hora_examen smalldatetime NULL,
Aula varchar(10) NULL,
Mencion_Honorifica bit NOT NULL,
Resultado char(1) NULL,
PRIMARY KEY (No_registro),
FOREIGN KEY (No_de_cuenta)
REFERENCES Alumnos,
FOREIGN KEY (Rfc)
REFERENCES Profesores
)
go

exec sp_primarykey Tesis,
No_registro
go

CREATE TABLE Sinodales
(No_registro int NOT NULL,
Rfc char(13) NOT NULL,
Funcion tinyint NULL,
Asistencia bit NOT NULL,
PRIMARY KEY (No_registro, Rfc),
FOREIGN KEY (Rfc)
REFERENCES Profesores,
FOREIGN KEY (No_registro)
REFERENCES Tesis
)
go

exec sp_primarykey Sinodales,
No_registro,
Rfc
go

CREATE TABLE Asignaturas
(Clave_de_asignatura char(4) NOT NULL,
Nombre_de_la_asignatura varchar(40) NOT NULL,
Clave_de_seminario tinyint NULL,
Tipo_Plan char(1) NULL,
Creditos tinyint NULL,

Horas_a_la_semana tinyint NULL,
Semestre tinyint NULL,
Clave_de_plantel tinyint NULL,
Division char(1) NULL,
PRIMARY KEY (Clave_de_asignatura),
FOREIGN KEY (Clave_de_seminario)
REFERENCES Seminario
)
go

exec sp_primarykey Asignaturas,
Clave_de_asignatura
go

exec sp_bindefault Def_Plantel, 'Asignaturas Clave_de_plantel'
go

CREATE TABLE Res_Hist_Acad
(No_de_cuenta char(9) NOT NULL,
MB tinyint NULL,
B tinyint NULL,
S tinyint NULL,
Ses tinyint NULL,
Siete tinyint NULL,
Ocho tinyint NULL,
Nueve tinyint NULL,
Diez tinyint NULL,
NA tinyint NULL,
NP tinyint NULL,
Revalidadas tinyint NULL,
Acreditadas tinyint NULL,
AprobOrd tinyint NULL,
AprobExtra tinyint NULL,
ReproOrd tinyint NULL,
ReproExtra tinyint NULL,
CredObtAcum smallint NULL,
CredOptAcum smallint NULL,
Covalidadas tinyint NULL,
PRIMARY KEY (No_de_cuenta),
FOREIGN KEY (No_de_cuenta)
REFERENCES Alumnos
)
go

exec sp_primarykey Res_Hist_Acad,
No_de_cuenta
go

CREATE TABLE Grupos
(Clave_de_grupo char(13) NOT NULL,
Periodo char(5) NULL,
Tipo char(1) NULL,
Division char(1) NULL,
Grupo char(2) NULL,
Clave_de_asignatura char(4) NOT NULL,
Cupo_maximo tinyint NULL,
Lugares_ocupados tinyint NULL,
Clave_aula char(8) NULL,
Hora_inicio smalldatetime NULL,
Hora_fin smalldatetime NULL,

```

## Apéndice A

```

Dias          tinyint NULL,
PRIMARY KEY (Clave_de_grupo),
FOREIGN KEY (Clave_grupo)
REFERENCES Aulas,
FOREIGN KEY (Clave_de_asignatura)
REFERENCES Asignaturas
)
go

exec sp_primarykey Grupos,
Clave_de_grupo
go

CREATE TABLE Comisionada
(Clave          tinyint NOT NULL,
Rfc            char(13) NOT NULL,
PRIMARY KEY (Clave, Rfc),
FOREIGN KEY (Clave)
REFERENCES Comision,
FOREIGN KEY (Rfc)
REFERENCES Profesores
)
go

exec sp_primarykey Comisionada,
Clave,
Rfc
go

CREATE TABLE Candidatos_a_oposicion
(Clave_de_asignatura char(4) NOT NULL,
Rfc                char(13) NOT NULL,
Periodo            char(5) NOT NULL,
Resultado          tinyint NULL,
Tema_Asignado     varchar(100) NULL,
PRIMARY KEY (Clave_de_asignatura, Rfc, Periodo),
FOREIGN KEY (Periodo, Clave_de_asignatura)
REFERENCES Exámenes_de_oposicion,
FOREIGN KEY (Rfc)
REFERENCES Profesores
)
go

exec sp_primarykey Candidatos_a_oposicion,
Clave_de_asignatura,
Rfc,
Periodo
go

CREATE TABLE Asistencia_profesores_de_carre
(Rfc char(13) NOT NULL,
Fecha_Falta smalldatetime NOT NULL,
Tipo_de_inasistencia tinyint NULL,
PRIMARY KEY (Rfc, Fecha_Falta),
FOREIGN KEY (Rfc)
REFERENCES Profesores
)
go

exec sp_primarykey Asistencia_profesores_de_carre,

```

```

Rfc,
Fecha_Falta
)
go

exec sp_bindefault Def_Falta,
'Asistencia_profesores_de_carre Tipo_de_inasistencia'
go

CREATE TABLE Comision
(Clave          tinyint NOT NULL,
Nombre_de_comision char(18) NULL,
PRIMARY KEY (Clave)
)
go

exec sp_primarykey Comision,
Clave
go

CREATE TABLE Rectificacion
(Rfc char(13) NOT NULL,
Clave_de_grupo char(13) NOT NULL,
No_de_cuenta   char(9) NOT NULL,
No_de_folio    int NULL,
Folio_del_acta int NULL,
Evaluacion_pasada char(2) NULL,
Evaluacion_presente char(2) NULL,
Renglon_del_acta tinyint NULL,
PRIMARY KEY (Clave_de_grupo, No_de_cuenta),
FOREIGN KEY (No_de_cuenta)
REFERENCES Alumnos,
FOREIGN KEY (Rfc)
REFERENCES Profesores,
FOREIGN KEY (Clave_de_grupo)
REFERENCES Grupos
)
go

exec sp_primarykey Rectificacion,
Clave_de_grupo,
No_de_cuenta
go

CREATE TABLE Clase
(Rfc char(13) NOT NULL,
Clave_de_grupo char(13) NOT NULL,
Fecha_inicio   smalldatetime NOT NULL,
Fecha_fin      smalldatetime NULL,
PRIMARY KEY (Rfc, Clave_de_grupo, Fecha_inicio),
FOREIGN KEY (Rfc)
REFERENCES Profesores,
FOREIGN KEY (Clave_de_grupo)
REFERENCES Grupos
)
go

exec sp_primarykey Clase,
Rfc,
Clave_de_grupo,
Fecha_inicio

```

```

go
exec sp_foreignkey Asistencias, Grupos,
  Clave_de_grupo
go
exec sp_foreignkey Asistencias, Profesores,
  Rfc
go
exec sp_foreignkey Contratos, Grupos,
  Clave_de_grupo
go
exec sp_foreignkey Contratos, Categorías,
  Clave_de_categoria
go
exec sp_foreignkey Contratos, Profesores,
  Rfc
go
exec sp_foreignkey Adscritos_a_seminario, Profesores,
  Rfc
go
exec sp_foreignkey Adscritos_a_seminario, Seminario,
  Clave_de_seminario
go
exec sp_foreignkey Exámenes_de_oposición, Asignaturas,
  Clave_de_asignatura
go
exec sp_foreignkey Exámenes_de_oposición, Comisión,
  Clave
go
exec sp_foreignkey Tesis, Alumnos,
  No_de_cuenta
go
exec sp_foreignkey Tesis, Profesores,
  Rfc
go
exec sp_foreignkey Sinodales, Profesores,
  Rfc
go
exec sp_foreignkey Sinodales, Tesis,
  No_registro
go
exec sp_foreignkey Asignaturas, Seminario,
  Clave_de_seminario
go
exec sp_foreignkey Res_Hist_Acad, Alumnos,
  No_de_cuenta

```

```

go
exec sp_foreignkey Grupos, Aulas,
  Clave_aula
go
exec sp_foreignkey Grupos, Asignaturas,
  Clave_de_asignatura
go
exec sp_foreignkey Comisionado, Comisión,
  Clave
go
exec sp_foreignkey Comisionado, Profesores,
  Rfc
go
exec sp_foreignkey Candidatos_a_oposición,
  Exámenes_de_oposición,
  Periodo,
  Clave_de_asignatura
go
exec sp_foreignkey Candidatos_a_oposición, Profesores,
  Rfc
go
exec sp_foreignkey Asistencia_profesores_de_carre, Profesores,
  Rfc
go
exec sp_foreignkey Rectificación, Alumnos,
  No_de_cuenta
go
exec sp_foreignkey Rectificación, Profesores,
  Rfc
go
exec sp_foreignkey Rectificación, Grupos,
  Clave_de_grupo
go
exec sp_foreignkey Clase, Profesores,
  Rfc
go
exec sp_foreignkey Clase, Grupos,
  Clave_de_grupo
go
create trigger tD_Categorías on Categorías for DELETE as
/* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
/* DELETE trigger on Categorías */
begin
declare @termo int,
        @errmsg varchar(255)
/* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */

```

## Apéndice A

```

/* Categorías Es parte de Contratos ON PARENT
DELETE RESTRICT */
if exists (
  select * from deleted,Contratos
  where
    /* %JoinFKPK(Contratos,deleted,"="," and" *)
    Contratos.Clave_de_categoria =
deleted.Clave_de_categoria
)
begin
  select @errno = 30001,
         @errmsg = "Cannot DELETE "Categorías" because
"Contratos" exists"
  goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
return
error:
  raiserror @errno @errmsg
  rollback transaction
end
go

create trigger tU_Categorías on Categorías for UPDATE as
/* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
/* UPDATE trigger on Categorías */
begin
  declare @numrows int,
          @nullent int,
          @insClave_de_categoria char(8),
          @errno int,
          @errmsg varchar(255)

  select @numrows = @@rowcount
  /* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
  /* Categorías Es parte de Contratos ON PARENT UPDATE
  CASCADE */
  if
    /* %ParentPK(" or".update) */
    update(Clave_de_categoria)
  begin
    if @numrows = 1
    begin
      select @insClave_de_categoria =
inserted.Clave_de_categoria
      from inserted
      update Contratos
      set
        /* %JoinFKPK(Contratos,@ins,"="," and" *)
        Contratos.Clave_de_categoria =
@insClave_de_categoria
      from Contratos,inserted,deleted
      where
        /* %JoinFKPK(Contratos,deleted,"="," and" *)
        Contratos.Clave_de_categoria =
deleted.Clave_de_categoria
    end
  else
  begin

```

```

  select @errno = 30006,
         @errmsg = "Cannot cascade "Categorías" UPDATE because
more than one row has been affected."
  raiserror @errno @errmsg
  end
end
/* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
return
error:
  raiserror @errno @errmsg
  rollback transaction
end
go

create trigger tD_Seminario on Seminario for DELETE as
/* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
/* DELETE trigger on Seminario */
begin
  declare @errno int,
          @errmsg varchar(255)
  /* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
  /* Seminario Esta firmado por Asignaturas ON PARENT DELETE
  RESTRICT */
  if exists (
    select * from deleted,Asignaturas
    where
      /* %JoinFKPK(Asignaturas,deleted,"="," and" *)
      Asignaturas.Clave_de_seminario = deleted.Clave_de_seminario
    )
  begin
    select @errno = 30001,
           @errmsg = "Cannot DELETE "Seminario" because
"Asignaturas" exists"
    goto error
  end
  /* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
  /* Seminario Contiene Adscritos_a_seminario ON PARENT
  DELETE RESTRICT */
  if exists (
    select * from deleted,Adscritos_a_seminario
    where
      /* %JoinFKPK(Adscritos_a_seminario,deleted,"="," and" *)
      Adscritos_a_seminario.Clave_de_seminario =
deleted.Clave_de_seminario
    )
  begin
    select @errno = 30001,
           @errmsg = "Cannot DELETE "Seminario" because
"Adscritos_a_seminario" exists"
    goto error
  end
  /* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
  return
error:
  raiserror @errno @errmsg
  rollback transaction
end
go

create trigger tU_Seminario on Seminario for UPDATE as

```

```

/* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
/* UPDATE trigger on Seminario */
begin
  declare @numrows int,
          @nullent int,
          @valident int,
          @insClave_de_seminario tinyint,
          @erro int,
          @errmsg varchar(255)

  select @numrows = @@rowcount
  /* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
  /* Seminario Esta formado por Asignaturas ON PARENT
  UPDATE CASCADE */
  if
    /* *ParentPK( " or ".update) */
    update(Clave_de_seminario)
  begin
    if @numrows = 1
    begin
      select @insClave_de_seminario =
inserted.Clave_de_seminario
      from inserted
      update Asignaturas
      set
        /* *JoinFKPK(Asignaturas,@ins," = ","") */
        Asignaturas.Clave_de_seminario =
@insClave_de_seminario
      from Asignaturas,inserted,deleted
      where
        /* *JoinFKPK(Asignaturas,deleted," = "," and") */
        Asignaturas.Clave_de_seminario =
deleted.Clave_de_seminario
      end
    else
    begin
      select @erro = 30006,
             @errmsg = "Cannot cascade "Seminario" UPDATE
because more than one row has been affected."
      raiserror @erro @errmsg
    end
  /* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
  /* Seminario Contiene Adscritos_a_seminario ON PARENT
  UPDATE CASCADE */
  if
    /* *ParentPK( " or ".update) */
    update(Clave_de_seminario)
  begin
    if @numrows = 1
    begin
      select @insClave_de_seminario =
inserted.Clave_de_seminario
      from inserted
      update Adscritos_a_seminario
      set
        /* *JoinFKPK(Adscritos_a_seminario,@ins," = ","") */
        Adscritos_a_seminario.Clave_de_seminario =
@insClave_de_seminario

```

```

      from Adscritos_a_seminario,inserted,deleted
      where
        /* *JoinFKPK(Adscritos_a_seminario,deleted," = "," and") */
        Adscritos_a_seminario.Clave_de_seminario =
deleted.Clave_de_seminario
      end
    else
    begin
      select @erro = 30006,
             @errmsg = "Cannot cascade "Seminario" UPDATE because
more than one row has been affected."
      raiserror @erro @errmsg
    end
  /* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
  return
error:
  raiserror @erro @errmsg
  rollback transaction
end
go

create trigger ID_Profesores on Profesores for DELETE as
/* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
/* DELETE trigger on Profesores */
begin
  declare @erro int,
          @errmsg varchar(255)
  /* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
  /* Profesores Es Simodales ON PARENT DELETE RESTRICT */
  if exists (
    select * from deleted,Simodales
    where
      /* *JoinFKPK(Simodales,deleted," = "," and") */
      Simodales.Rtc = deleted.Rtc
  )
  begin
    select @erro = 30001,
           @errmsg = "Cannot DELETE "Profesores" because
"Simodales" exists."
    goto error
  end
  /* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
  /* Profesores Realiza una Rectificacion ON PARENT DELETE
  RESTRICT */
  if exists (
    select * from deleted,Rectificacion
    where
      /* *JoinFKPK(Rectificacion,deleted," = "," and") */
      Rectificacion.Rtc = deleted.Rtc
  )
  begin
    select @erro = 30001,
           @errmsg = "Cannot DELETE "Profesores" because
"Rectificacion" exists."
    goto error
  end
  /* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
  /* Profesores Imparte una Clase ON PARENT DELETE
  RESTRICT */

```

```

if exists (
  select * from deleted.Clase
  where
    /* %JoinFKPK(Clase,deleted," = "," and") */
    Clase Rfc = deleted.Rfc
)
begin
  select @ierrno = 30001,
         @ierrmsg = 'Cannot DELETE "Profesores" because
"Clase" exists.'
  goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
/* Profesores Contiene Tesis ON PARENT DELETE
RESTRICT */
if exists (
  select * from deleted.Tesis
  where
    /* %JoinFKPK(Tesis,deleted," = "," and") */
    Tesis.Rfc = deleted.Rfc
)
begin
  select @ierrno = 30001,
         @ierrmsg = 'Cannot DELETE "Profesores" because
"Tesis" exists.'
  goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
/* Profesores Tiene Asistencia_profesores_de_carre ON
PARENT DELETE RESTRICT */
if exists (
  select * from deleted.Asistencia_profesores_de_carre
  where
    /* %JoinFKPK(Asistencia_profesores_de_carre,deleted," = ","
and") */
    Asistencia_profesores_de_carre.Rfc = deleted.Rfc
)
begin
  select @ierrno = 30001,
         @ierrmsg = 'Cannot DELETE "Profesores" because
"Asistencia_profesores_de_carre" exists.'
  goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
/* Profesores Se encuentra en Comisionado ON PARENT
DELETE RESTRICT */
if exists (
  select * from deleted.Comisionado
  where
    /* %JoinFKPK(Comisionado,deleted," = "," and") */
    Comisionado.Rfc = deleted.Rfc
)
begin
  select @ierrno = 30001,
         @ierrmsg = 'Cannot DELETE "Profesores" because
"Comisionado" exists.'
  goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */

```

```

/* Profesores Debe ser Candidatos_a_oposicion ON PARENT
DELETE RESTRICT */
if exists (
  select * from deleted.Candidatos_a_oposicion
  where
    /* %JoinFKPK(Candidatos_a_oposicion,deleted," = "," and") */
    Candidatos_a_oposicion.Rfc = deleted.Rfc
)
begin
  select @ierrno = 30001,
         @ierrmsg = 'Cannot DELETE "Profesores" because
"Candidatos_a_oposicion" exists.'
  goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
/* Profesores Tiene Asistencias ON PARENT DELETE
RESTRICT */
if exists (
  select * from deleted.Asistencias
  where
    /* %JoinFKPK(Asistencias,deleted," = "," and") */
    Asistencias.Rfc = deleted.Rfc
)
begin
  select @ierrno = 30001,
         @ierrmsg = 'Cannot DELETE "Profesores" because
"Asistencias" exists.'
  goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
/* Profesores Se encuentra Adscritos_a_seminario ON PARENT
DELETE RESTRICT */
if exists (
  select * from deleted.Adscritos_a_seminario
  where
    /* %JoinFKPK(Adscritos_a_seminario,deleted," = "," and") */
    Adscritos_a_seminario.Rfc = deleted.Rfc
)
begin
  select @ierrno = 30001,
         @ierrmsg = 'Cannot DELETE "Profesores" because
"Adscritos_a_seminario" exists.'
  goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
/* Profesores Es Contratado ON PARENT DELETE RESTRICT */
if exists (
  select * from deleted.Contratos
  where
    /* %JoinFKPK(Contratos,deleted," = "," and") */
    Contratos.Rfc = deleted.Rfc
)
begin
  select @ierrno = 30001,
         @ierrmsg = 'Cannot DELETE "Profesores" because
"Contratos" exists.'
  goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:11 1997 */
return

```



## Apéndice A

```

end
else
begin
select @ermo = 30006,
        @errmsg = 'Cannot cascade "Profesores" UPDATE
because more than one row has been affected.'
raiserror @ermo @errmsg
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Profesores Tiene Asistencia_profesores_de_carre ON
PARENT UPDATE CASCADE */
if
/* %ParentPK(" or",update) */
update(Rfc)
begin
if @numrows = 1
begin
select @insRfc = inserted.Rfc
from inserted
update Asistencia_profesores_de_carre
set
/* %JoinFKPK(Asistencia_profesores_de_carre,@ins,
" ","") */
Asistencia_profesores_de_carre.Rfc = @insRfc
from Asistencia_profesores_de_carre,inserted,deleted
where
/*
%JoinFKPK(Asistencia_profesores_de_carre,deleted," ","
and") */
Asistencia_profesores_de_carre.Rfc = deleted.Rfc
end
else
begin
select @ermo = 30006,
        @errmsg = 'Cannot cascade "Profesores" UPDATE
because more than one row has been affected.'
raiserror @ermo @errmsg
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Profesores Se encuentra en Comisionado ON PARENT
UPDATE CASCADE */
if
/* %ParentPK(" or",update) */
update(Rfc)
begin
if @numrows = 1
begin
select @insRfc = inserted.Rfc
from inserted
update Comisionado
set
/* %JoinFKPK(Comisionado,@ins," ","") */
Comisionado.Rfc = @insRfc
from Comisionado,inserted,deleted
where
/* %JoinFKPK(Comisionado,deleted," "," and") */
Comisionado.Rfc = deleted.Rfc
end
end

```

```

else
begin
select @ermo = 30006,
        @errmsg = 'Cannot cascade "Profesores" UPDATE because
more than one row has been affected.'
raiserror @ermo @errmsg
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Profesores Debe ser Candidatos_a_oposicion ON PARENT
UPDATE CASCADE */
if
/* %ParentPK(" or",update) */
update(Rfc)
begin
if @numrows = 1
begin
select @insRfc = inserted.Rfc
from inserted
update Candidatos_a_oposicion
set
/* %JoinFKPK(Candidatos_a_oposicion,@ins," ","") */
Candidatos_a_oposicion.Rfc = @insRfc
from Candidatos_a_oposicion,inserted,deleted
where
/* %JoinFKPK(Candidatos_a_oposicion,deleted," "," and") */
Candidatos_a_oposicion.Rfc = deleted.Rfc
end
else
begin
select @ermo = 30006,
        @errmsg = 'Cannot cascade "Profesores" UPDATE because
more than one row has been affected.'
raiserror @ermo @errmsg
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Profesores Tiene Asistencias ON PARENT UPDATE CASCADE
*/
if
/* %ParentPK(" or",update) */
update(Rfc)
begin
if @numrows = 1
begin
select @insRfc = inserted.Rfc
from inserted
update Asistencias
set
/* %JoinFKPK(Asistencias,@ins," ","") */
Asistencias.Rfc = @insRfc
from Asistencias,inserted,deleted
where
/* %JoinFKPK(Asistencias,deleted," "," and") */
Asistencias.Rfc = deleted.Rfc
end
else
begin
select @ermo = 30006,

```

```

        @errmsg = 'Cannot cascade "Profesores" UPDATE
because more than one row has been affected.'
        raiserror (@errmsg)
    end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Profesores Se encuentran Adscritos_a_seminario ON
PARENT UPDATE CASCADE */
if
/* %ParentPK(" or",update) */
update(Rfc)
begin
if @numrows = 1
begin
select @insRfc = inserted.Rfc
from inserted
update Adscritos_a_seminario
set
/* %JoinFKPK(Adscritos_a_seminario,@ins," ",")
*/
Adscritos_a_seminario.Rfc = @insRfc
from Adscritos_a_seminario,inserted,deleted
where
/* %JoinFKPK(Adscritos_a_seminario,deleted," ",
and") */
Adscritos_a_seminario.Rfc = deleted.Rfc
end
else
begin
select @errno = 30006,
@errmsg = 'Cannot cascade "Profesores" UPDATE
because more than one row has been affected.'
raiserror (@errmsg)
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Profesores Es Contratos ON PARENT UPDATE
CASCADE */
if
/* %ParentPK(" or",update) */
update(Rfc)
begin
if @numrows = 1
begin
select @insRfc = inserted.Rfc
from inserted
update Contratos
set
/* %JoinFKPK(Contratos,@ins," ",") */
Contratos.Rfc = @insRfc
from Contratos,inserted,deleted
where
/* %JoinFKPK(Contratos,deleted," ",
and") */
Contratos.Rfc = deleted.Rfc
end
else
begin
select @errno = 30006,
@errmsg = 'Cannot cascade "Profesores" UPDATE
because more than one row has been affected.'

```

```

        raiserror (@errmsg)
    end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
return
error:
raiserror (@errmsg)
rollback transaction
end
go

create trigger U_Aulas on Aulas for DELETE as
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* DELETE trigger on Aulas */
begin
declare @errno int,
@errmsg varchar(255)
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Aulas Alberga a Grupos ON PARENT DELETE RESTRICT */
if exists (
select * from deleted,Grupos
where
/* %JoinFKPK(Grupos,deleted," ",
and") */
Grupos.Clave_aula = deleted.Clave_aula
)
begin
select @errno = 30001,
@errmsg = 'Cannot DELETE "Aulas" because "Grupos"
exists.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
return
error:
raiserror (@errmsg)
rollback transaction
end
go

create trigger U_Aulas on Aulas for UPDATE as
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* UPDATE trigger on Aulas */
begin
declare @numrows int,
@nullent int,
@valident int,
@insClave_aula char(8),
@errno int,
@errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Aulas Alberga a Grupos ON PARENT UPDATE CASCADE */
if
/* %ParentPK(" or",update) */
update(Clave_aula)
begin
if @numrows = 1
begin
select @insClave_aula = inserted.Clave_aula

```

```

from inserted
update Grupos
set
/* %JoinFKPK(Grupos,@ins,"=","") */
Grupos.Clave_aula=@insClave_aula
from Grupos,inserted,deleted
where
/* %JoinFKPK(Grupos,deleted,"=","") and" */
Grupos.Clave_aula=deleted.Clave_aula
end
else
begin
select @errno=30006,
@errmsg="Cannot cascade "Aulas" UPDATE
because more than one row has been affected."
raiserror @errno,@errmsg
end
/* Erwin BuiltIn Sun Feb 02 18:41:12 1997 */
return
error:
raiserror @errno,@errmsg
rollback transaction
end
go

create trigger tD_Alumnos on Alumnos for DELETE as
/* Erwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* DELETE trigger on Alumnos */
begin
declare @errno int,
@errmsg varchar(255)
/* Erwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Alumnos Tiene un Res_Hist_Acad ON PARENT
DELETE RESTRICT */
if exists (
select * from deleted,Res_Hist_Acad
where
/* %JoinFKPK(Res_Hist_Acad,deleted,"=","") and" */
Res_Hist_Acad.No_de_cuenta=deleted.No_de_cuenta
)
begin
select @errno=30001,
@errmsg="Cannot DELETE "Alumnos" because
"Res_Hist_Acad" exists."
goto error
end
/* Erwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Alumnos Solicita una Rectificacion ON PARENT
DELETE RESTRICT */
if exists (
select * from deleted,Rectificacion
where
/* %JoinFKPK(Rectificacion,deleted,"=","") and" */
Rectificacion.No_de_cuenta=deleted.No_de_cuenta
)
begin
select @errno=30001,
@errmsg="Cannot DELETE "Alumnos" because
"Rectificacion" exists."

```

```

goto error
end
/* Erwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Alumnos Inscribe Tesis ON PARENT DELETE RESTRICT */
if exists (
select * from deleted,Tesis
where
/* %JoinFKPK(Tesis,deleted,"=","") and" */
Tesis.No_de_cuenta=deleted.No_de_cuenta
)
begin
select @errno=30001,
@errmsg="Cannot DELETE "Alumnos" because "Tesis"
exists."
goto error
end
/* Erwin BuiltIn Sun Feb 02 18:41:12 1997 */
return
error:
raiserror @errno,@errmsg
rollback transaction
end
go

create trigger tU_Alumnos on Alumnos for UPDATE as
/* Erwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* UPDATE trigger on Alumnos */
begin
declare @numrows int,
@nullent int,
@valident int,
@insNo_de_cuenta char(9),
@errno int,
@errmsg varchar(255)

select @numrows=@@row_count
/* Erwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Alumnos Tiene un Res_Hist_Acad ON PARENT UPDATE
CASCADE */
if
/* %ParentFK("ur",update) */
update(No_de_cuenta)
if @numrows=1
begin
select @insNo_de_cuenta=inserted.No_de_cuenta
from inserted
update Res_Hist_Acad
set
/* %JoinFKPK(Res_Hist_Acad,@ins,"=","") */
Res_Hist_Acad.No_de_cuenta=@insNo_de_cuenta
from Res_Hist_Acad,inserted,deleted
where
/* %JoinFKPK(Res_Hist_Acad,deleted,"=","") and" */
Res_Hist_Acad.No_de_cuenta=deleted.No_de_cuenta
end
else
begin
select @errno=30006,

```

```

@errmsg = 'Cannot cascade "Alumnos" UPDATE
because more than one row has been affected'
raiserror @errno @errmsg
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Alumnos Solicita una Rectificacion ON PARENT
UPDATE CASCADE */
if
/* %ParentPK(" or",update) */
update(No_de_cuenta)
begin
if @numrows = 1
begin
select @insNo_de_cuenta = inserted.No_de_cuenta
from inserted
update Rectificacion
set
/* %JoinFKPK(Rectificacion,@ins," ",") */
Rectificacion_No_de_cuenta = @insNo_de_cuenta
from Rectificacion,inserted,deleted
where
/* %JoinFKPK(Rectificacion,deleted," ",and") */
Rectificacion_No_de_cuenta = deleted.No_de_cuenta
end
else
begin
select @errno = 30006,
@errmsg = 'Cannot cascade "Alumnos" UPDATE
because more than one row has been affected'
raiserror @errno @errmsg
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Alumnos Inscribe Tesis ON PARENT UPDATE
CASCADE */
if
/* %ParentPK(" or",update) */
update(No_de_cuenta)
begin
if @numrows = 1
begin
select @insNo_de_cuenta = inserted.No_de_cuenta
from inserted
update Tesis
set
/* %JoinFKPK(Tesis,@ins," ",") */
Tesis.No_de_cuenta = @insNo_de_cuenta
from Tesis,inserted,deleted
where
/* %JoinFKPK(Tesis,deleted," ",and") */
Tesis.No_de_cuenta = deleted.No_de_cuenta
end
else
begin
select @errno = 30006,
@errmsg = 'Cannot cascade "Alumnos" UPDATE
because more than one row has been affected.'
raiserror @errno @errmsg
end
end

```

```

end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
return
error.
raiserror @errmsg @errmsg
rollback transaction
end
go
create trigger If_Asstencias on Asistencias for INSERT as
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* INSERT trigger on Asistencias */
begin
declare @numrows int,
@nullent int,
@valident int,
@errmsg int,
@errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Grupos Es parte de una Asistencias ON CHILD INSERT
RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave_de_grupo)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Grupos
where
/* %JoinFKPK(inserted,Grupos) */
inserted.Clave_de_grupo = Grupos.Clave_de_grupo
/* %NotNullFK(inserted," is null",select @nullent = count(*) from
inserted where "", and") */

if @valident + @nullent != @numrows
begin
select @errno = 30002,
@errmsg = 'Cannot INSERT "Asistencias" because "Grupos"
does not exist'
goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Profesores Tiene Asistencias ON CHILD INSERT RESTRICT */
if
/* %ChildFK(" or",update) */
update(Rfc)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotNullFK(inserted," is null",select @nullent = count(*) from
inserted where "", and") */

if @valident + @nullent != @numrows

```

## Apéndice A

```

begin
select @errno = 30002,
        @errmsg = 'Cannot INSERT "Asistencias" because
"Profesores" does not exist'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
return
error:
raiserror @@errno @@errmsg
rollback transaction
end
go

create trigger tU_Asistencias on Asistencias for UPDATE as
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* UPDATE trigger on Asistencias */
begin
declare @numrows int,
        @nullent int,
        @valident int,
        @insRfc char(13),
        @insClave_de_grupo char(13),
        @insFecha_de_falta smalldatetime,
        @errno int,
        @errmsg varchar(255)

select @numrows = @@@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Grupos Es parte de una Asistencias ON CHILD
UPDATE RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave_de_grupo)
begin
select @nullent = 0
select @valident = count(*)
from inserted.Grupos
where
/* %JoinFKPK(inserted.Grupos) */
inserted.Clave_de_grupo = Grupos.Clave_de_grupo
/* %NotnullFK(inserted," is null","select @nullent =
count(*) from inserted where"," and") */

if @valident + @nullent != @numrows
begin
select @errno = 30007,
        @errmsg = 'Cannot UPDATE "Asistencias" because
"Grupos" does not exist'
goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Profesores Tiene Asistencias ON CHILD UPDATE
RESTRICT */
if
/* %ChildFK(" or",update) */
update(Rfc)
begin

```

```

select @nullent = 0
select @valident = count(*)
from inserted.Profesores
where
/* %JoinFKPK(inserted.Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotnullFK(inserted," is null","select @nullent = count(*) from
inserted where"," and") */

if @valident + @nullent != @numrows
begin
select @errno = 30007,
        @errmsg = 'Cannot UPDATE "Asistencias" because
"Profesores" does not exist'
goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
return
error:
raiserror @@errno @@errmsg
rollback transaction
end
go

create trigger tI_Contratos on Contratos for INSERT as
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* INSERT trigger on Contratos */
begin
declare @numrows int,
        @nullent int,
        @valident int,
        @errno int,
        @errmsg varchar(255)

select @numrows = @@@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Grupos Forma un Contratos ON CHILD INSERT RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave_de_grupo)
begin
select @nullent = 0
select @valident = count(*)
from inserted.Grupos
where
/* %JoinFKPK(inserted.Grupos) */
inserted.Clave_de_grupo = Grupos.Clave_de_grupo
/* %NotnullFK(inserted," is null","select @nullent = count(*) from
inserted where"," and") */

if @valident + @nullent != @numrows
begin
select @errno = 30002,
        @errmsg = 'Cannot INSERT "Contratos" because "Grupos"
does not exist'
goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */

```

```

/* Categorías Es parte de Contratos ON CHILD INSERT
RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave_de_categoria)
begin
select @iNULL = 0
select @iVALID = count(*)
from inserted,Categorias
where
/* %JoinFKPK(inserted,Categorias) */
inserted.Clave_de_categoria =
Categorias.Clave_de_categoria
/* %NoNullFK(inserted," is null","select @iNULL =
count(*) from inserted where"," and") */

if @iVALID + @iNULL != @iNUMROWS
begin
select @iERRNO = 30002,
@iERRMSG = 'Cannot INSERT "Contratos" because
"Categorias" does not exist.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Profesores Es Contratos ON CHILD INSERT
RESTRICT */
if
/* %ChildFK(" or",update) */
update(Rfc)
begin
select @iNULL = 0
select @iVALID = count(*)
from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NoNullFK(inserted," is null","select @iNULL =
count(*) from inserted where"," and") */

if @iVALID + @iNULL != @iNUMROWS
begin
select @iERRNO = 30002,
@iERRMSG = 'Cannot INSERT "Contratos" because
"Profesores" does not exist.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
return
error:
raiserror @iERRNO @iERRMSG
rollback transaction
end
go

create trigger tU_Contratos on Contratos for UPDATE as
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* UPDATE trigger on Contratos */
begin

```

```

declare @iNUMROWS int,
        @iNULL int,
        @iVALID int,
        @iINSRfc char(13),
        @iINSClave_de_categoria char(8),
        @iINSClave_de_grupo char(13),
        @iERRNO int,
        @iERRMSG varchar(255)

select @iNUMROWS = @@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Grupos Forma un Contratos ON CHILD UPDATE RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave_de_grupo)
begin
select @iNULL = 0
select @iVALID = count(*)
from inserted,Grupos
where
/* %JoinFKPK(inserted,Grupos) */
inserted.Clave_de_grupo = Grupos.Clave_de_grupo
/* %NoNullFK(inserted," is null","select @iNULL = count(*) from
inserted where"," and") */

if @iVALID + @iNULL != @iNUMROWS
begin
select @iERRNO = 30003,
@iERRMSG = 'Cannot UPDATE "Contratos" because "Grupos"
does not exist.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Categorías Es parte de Contratos ON CHILD UPDATE
RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave_de_categoria)
begin
select @iNULL = 0
select @iVALID = count(*)
from inserted,Categorias
where
/* %JoinFKPK(inserted,Categorias) */
inserted.Clave_de_categoria = Categorias.Clave_de_categoria
/* %NoNullFK(inserted," is null","select @iNULL = count(*) from
inserted where"," and") */

if @iVALID + @iNULL != @iNUMROWS
begin
select @iERRNO = 30007,
@iERRMSG = 'Cannot UPDATE "Contratos" because
"Categorias" does not exist.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Profesores Es Contratos ON CHILD UPDATE RESTRICT */
if

```

## Apéndice A

```

/* %ChildFK(" or" update) */
update(Rfc)
begin
select @nnullent = 0
select @nvalident = count(*)
from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotnullFK(inserted," is null","select @nnullent =
count(*) from inserted where"," and") */
if @nvalident + @nnullent != @numrows
begin
select @erromo = 30007,
@errmsg = 'Cannot UPDATE "Contratos" because
"Profesores" does not exist.'
goto error
end
end
/* ERwin Built-in Sun Feb 02 18:41:12 1997 */
return
error:
raiserror @erromo @errmsg
rollback transaction
end
go

create trigger tL_Adscritos_a_seminario on
Adscritos_a_seminario for INSERT as
/* ERwin Built-in Sun Feb 02 18:41:12 1997 */
/* INSERT trigger on Adscritos_a_seminario */
begin
declare @numrows int,
@nnullent int,
@nvalident int,
@erromo int,
@errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin Built-in Sun Feb 02 18:41:12 1997 */
/* Profesores Se encuentra Adscritos_a_seminario ON
CHILD INSERT RESTRICT */
if
/* %ChildFK(" or" update) */
update(Rfc)
begin
select @nnullent = 0
select @nvalident = count(*)
from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotnullFK(inserted," is null","select @nnullent =
count(*) from inserted where"," and") */
if @nvalident + @nnullent != @numrows
begin
select @erromo = 30002,

```

```

@errmsg = 'Cannot INSERT "Adscritos_a_seminario"
because "Profesores" does not exist.'
goto error
end
end
/* ERwin Built-in Sun Feb 02 18:41:12 1997 */
/* Seminario Contiene Adscritos_a_seminario ON CHILD INSERT
RESTRICT */
if
/* %ChildFK(" or" update) */
update(Clave_de_seminario)
begin
select @nnullent = 0
select @nvalident = count(*)
from inserted,Seminario
where
/* %JoinFKPK(inserted,Seminario) */
inserted.Clave_de_seminario = Seminario.Clave_de_seminario
/* %NotnullFK(inserted," is null","select @nnullent = count(*) from
inserted where"," and") */
if @nvalident + @nnullent != @numrows
begin
select @erromo = 30002,
@errmsg = 'Cannot INSERT "Adscritos_a_seminario"
because "Seminario" does not exist.'
goto error
end
end
/* ERwin Built-in Sun Feb 02 18:41:12 1997 */
return
error:
raiserror @erromo @errmsg
rollback transaction
end
go

create trigger tU_Adscritos_a_seminario on Adscritos_a_seminario
for UPDATE as
/* ERwin Built-in Sun Feb 02 18:41:12 1997 */
/* UPDATE trigger on Adscritos_a_seminario */
begin
declare @numrows int,
@nnullent int,
@nvalident int,
@insRfc char(13),
@insClave_de_seminario tinyint,
@erromo int,
@errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin Built-in Sun Feb 02 18:41:12 1997 */
/* Profesores Se encuentra Adscritos_a_seminario ON CHILD
UPDATE RESTRICT */
if
/* %ChildFK(" or" update) */
update(Rfc)
begin
select @nnullent = 0
select @nvalident = count(*)

```

```

from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotNullFK(inserted," is null", "select (@null =
count(*) from inserted where," and") */

if (@valid + @null != @numrows
begin
select @errno = 30007,
@errmsg = 'Cannot UPDATE
"Adscritos_a_seminario" because "Profesores" does not
exist.'
goto error
end
error
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Seminario Contiene Adscritos_a_seminario ON CHILD
UPDATE RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave_de_seminario)
begin
select @null = 0
select @valid = count(*)
from inserted,Seminario
where
/* %JoinFKPK(inserted,Seminario) */
inserted.Clave_de_seminario =
Seminario.Clave_de_seminario
/* %NotNullFK(inserted," is null", "select (@null =
count(*) from inserted where," and") */

if (@valid + @null != @numrows
begin
select @errno = 30007,
@errmsg = 'Cannot UPDATE
"Adscritos_a_seminario" because "Seminario" does not exist.'
goto error
end
error
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
return
error:
raiserror @errno @errmsg
rollback transaction
end
go

create trigger tD_Examenes_de_oseposicion on
Examenes_de_oseposicion for DELETE as
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* DELETE trigger on Examenes_de_oseposicion */
begin
declare @errno int,
@errmsg varchar(255)
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Examenes_de_oseposicion Esta Inscrito en
Candidatos_a_oseposicion ON PARENT DELETE RESTRICT
*/

```

```

if exists t
select * from deleted,Candidatos_a_oseposicion
where
/* %JoinFKPK(Candidatos_a_oseposicion,deleted," "," and") */
Candidatos_a_oseposicion.Periodo = deleted.Periodo and
Candidatos_a_oseposicion.Clave_de_asignatura =
deleted.Clave_de_asignatura
)
begin
select @errno = 30001,
@errmsg = 'Cannot DELETE "Examenes_de_oseposicion"
because "Candidatos_a_oseposicion" exists.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
return
error:
raiserror @errno @errmsg
rollback transaction
end
go

create trigger tI_Examenes_de_oseposicion on Examenes_de_oseposicion
for INSERT as
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* INSERT trigger on Examenes_de_oseposicion */
begin
declare @numrows int,
@null int,
@valid int,
@errno int,
@errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Asignaturas Esta en un Examenes_de_oseposicion ON CHILD
INSERT RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave_de_asignatura)
begin
select @null = 0
select @valid = count(*)
from inserted,Asignaturas
where
/* %JoinFKPK(inserted,Asignaturas) */
inserted.Clave_de_asignatura =
Asignaturas.Clave_de_asignatura
/* %NotNullFK(inserted," is null", "select (@null = count(*) from
inserted where," and") */

if (@valid + @null != @numrows
begin
select @errno = 30002,
@errmsg = 'Cannot INSERT "Examenes_de_oseposicion"
because "Asignaturas" does not exist.'
goto error
end
error
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */

```

## Apéndice A

```

/* Comision Es asignado a una comision
Exámenes_de_oposicion ON CHILD INSERT RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Comision
where
/* %JoinFKPK(inserted,Comision) *
inserted.Clave = Comision.Clave
/* %NotNullFK(inserted," is null","select @nullent =
count(*) from inserted where"," and") */

if @@valident + @@nullent != @@numrows
begin
select @errno = 30002,
@errmsg = "Cannot INSERT
"Exámenes_de_oposicion" because "Comision" does not
exist."
goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18 41 12 1997 */
return
error:
raiserror (@.errno,@errmsg
rollback transaction
end
go

create trigger tU_Exámenes_de_oposicion on
Exámenes_de_oposicion for UPDATE as
/* ERwin BuiltIn Sun Feb 02 18 41 12 1997 */
/* UPDATE trigger on Exámenes_de_oposicion */
begin
declare @numrows int,
@nullent int,
@valident int,
@insPeriodo char(5),
@insClave_de_asignatura char(4),
@errno int,
@errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin BuiltIn Sun Feb 02 18 41 12 1997 */
/* Exámenes_de_oposicion Esta Inserto en
Candidatos_a_oposicion ON PARENT UPDATE CASCADE
*/
if
/* %ParentPK(" or",update) */
update(Periodo) or
update(Clave_de_asignatura)
begin
if @numrows = 1
begin
select @insPeriodo = inserted.Periodo,
@insClave_de_asignatura =
inserted.Clave_de_asignatura

```

```

from inserted
update Candidatos_a_oposicion
set
/* %JoinFKPK(Candidatos_a_oposicion,@ins," = ",".") */
Candidatos_a_oposicion.Periodo = @insPeriodo,
Candidatos_a_oposicion.Clave_de_asignatura =
@insClave_de_asignatura
from Candidatos_a_oposicion,inserted,deleted
where
/* %JoinFKPK(Candidatos_a_oposicion,deleted," = "," and") */
Candidatos_a_oposicion.Periodo = deleted.Periodo and
Candidatos_a_oposicion.Clave_de_asignatura =
deleted.Clave_de_asignatura
end
else
begin
select @errno = 30006,
@errmsg = "Cannot cascade "Exámenes_de_oposicion"
UPDATE because more than one row has been affected."
raiserror (@.errno,@errmsg
end
end
/* ERwin BuiltIn Sun Feb 02 18 41 12 1997 */
/* Asignaturas Esta en un Exámenes_de_oposicion ON CHILD
UPDATE RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave_de_asignatura)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Asignaturas
where
/* %JoinFKPK(inserted,Asignaturas) */
inserted.Clave_de_asignatura =
Asignaturas.Clave_de_asignatura
/* %NotNullFK(inserted," is null","select @nullent = count(*) from
inserted where"," and") */

if @valident + @nullent != @numrows
begin
select @errno = 30007,
@errmsg = "Cannot UPDATE "Exámenes_de_oposicion"
because "Asignaturas" does not exist."
goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18 41 12 1997 */
/* Comision Es asignado a una comision Exámenes_de_oposicion
ON CHILD UPDATE RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Comision
where
/* %JoinFKPK(inserted,Comision) */
inserted.Clave = Comision.Clave

```

```

/* %NotNullFK(inserted, " is null", "select @null = count(*)
from inserted where", " and") */
if @@valid + @@null != @@numrows
begin
select @errno = 30007,
@errmsg = 'Cannot UPDATE
"Exámenes de oposición" because "Comision" does not
exist.'
goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
return
error:
raiserror @errno @errmsg
rollback transaction
end
go

create trigger tD_Tesis on Tesis for DELETE as
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* DELETE trigger on Tesis */
begin
declare @errno int,
@errmsg varchar(255)
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Tesis Contiene a Sinodales ON PARENT DELETE
RESTRICT */
if exists (
select * from deleted, Sinodales
where
/* %JoinFKPK(Sinodales,deleted," = ", " and") */
Sinodales No_registro = deleted No_registro
)
begin
select @errno = 30001,
@errmsg = 'Cannot DELETE "Tesis" because
"Sinodales" exists.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
return
error:
raiserror @errno @errmsg
rollback transaction
end
go

create trigger tI_Tesis on Tesis for INSERT as
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* INSERT trigger on Tesis */
begin
declare @numrows int,
@nullcnt int,
@errno int,
@errmsg varchar(255)

select @numrows = @@rowcount

```

```

/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Alumnos Inscribe Tesis ON CHILD INSERT RESTRICT */
if
/* %ChildFK(" or",update) */
update(No_de_cuenta)
begin
select @nullcnt = 0
select @valid = count(*)
from inserted, Alumnos
where
/* %JoinFKPK(inserted,Alumnos) */
inserted No_de_cuenta = Alumnos No_de_cuenta
/* %NotNullFK(inserted, " is null", "select @null = count(*) from
inserted where", " and") */

if @@valid + @@null != @@numrows
begin
select @errno = 30002,
@errmsg = 'Cannot INSERT "Tesis" because "Alumnos"
does not exist.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Profesores Contiene Tesis ON CHILD INSERT RESTRICT */
if
/* %ChildFK(" or",update) */
update(Rfc)
begin
select @nullcnt = 0
select @valid = count(*)
from inserted, Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted Rfc = Profesores Rfc
/* %NotNullFK(inserted, " is null", "select @null = count(*) from
inserted where", " and") */

if @@valid + @@null != @@numrows
begin
select @errno = 30002,
@errmsg = 'Cannot INSERT "Tesis" because "Profesores"
does not exist.'
goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
return
error:
raiserror @errno @errmsg
rollback transaction
end
go

create trigger tU_Tesis on Tesis for UPDATE as
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* UPDATE trigger on Tesis */
begin
declare @numrows int,
@nullcnt int,

```

## Apéndice A

```

@valident int,
@insNo_registro int,
@errno int,
@errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Tesis Contiene a Sinodales ON PARENT UPDATE
CASCADE */
if
/* %ParentFK(" or",update) */
update(No_registro)
begin
if @numrows = 1
begin
select @insNo_registro = inserted.No_registro
from inserted
update Sinodales
set
/* %JoinFKPK(Sinodales,@ins, " - ",") */
Sinodales.No_registro = @insNo_registro
from Sinodales,inserted,deleted
where
/* %JoinFKPK(Sinodales,deleted," - ",and) */
Sinodales.No_registro = deleted.No_registro
end
else
begin
select @errno = 30006,
@errmsg = "Cannot cascade "Tesis" UPDATE
because more than one row has been affected"
raiserror @errno @errmsg
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Alumnos Inscribe Tesis ON CHILD UPDATE
RESTRICT */
if
/* %ChildFK(" or",update) */
update(No_de_cuenta)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Alumnos
where
/* %JoinFKPK(inserted,Alumnos) */
inserted.No_de_cuenta = Alumnos.No_de_cuenta
/* %NotNullFK(inserted," is null","select @nullent =
count(*) from inserted where"," and") */

if @valident = @nullent != @numrows
begin
select @errno = 30007,
@errmsg = "Cannot UPDATE "Tesis" because
"Alumnos" does not exist."
goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */

```

```

/* Profesores Contiene Tesis ON CHILD UPDATE RESTRICT */
if
/* %ChildFK(" or",update) */
update(Rfc)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotNullFK(inserted," is null","select @nullent = count(*) from
inserted where"," and") */

if @valident = @nullent != @numrows
begin
select @errno = 30007,
@errmsg = "Cannot UPDATE "Tesis" because "Profesores"
does not exist."
goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
return
error:
raiserror @errno @errmsg
rollback transaction
end
go

create trigger t1_Sinodales on Sinodales for INSERT as
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* INSERT trigger on Sinodales */
begin
declare @numrows int,
@nullent int,
@valident int,
@errno int,
@errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Profesores Es Sinodales ON CHILD INSERT RESTRICT */
if
/* %ChildFK(" or",update) */
update(Rfc)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotNullFK(inserted," is null","select @nullent = count(*) from
inserted where"," and") */

if @valident = @nullent != @numrows
begin
select @errno = 30002,

```

```

        @errmsg = 'Cannot INSERT "Sinodales" because
"Profesores" does not exist.'
        goto error
    end
end
/* ERwin Builtin Sun Feb 02 18:41:12 1997 */
/* Tesis Contiene a Sinodales ON CHILD INSERT
RESTRICT */
if
    /* %ChildFK(" or",update) */
    update(No_registro)
begin
    select @nnullent = 0
    select @valident = count(*)
    from inserted.Tesis
    where
        /* %JoinFKPK(inserted,Tesis) */
        inserted.No_registro = Tesis.No_registro
    /* %NotNullFK(inserted," is null","select @nnullent =
count(*) from inserted where"," and") */

    if @valident + @nnullent != @numrows
    begin
        select @errno = 30002,
        @errmsg = "Cannot INSERT "Sinodales" because
"Profesores" does not exist."
        goto error
    end
end
/* ERwin Builtin Sun Feb 02 18:41:12 1997 */
return
error:
    raiserror @errno @errmsg
    rollback transaction
end
go

```

```

create trigger tU_Sinodales on Sinodales for UPDATE as
/* ERwin Builtin Sun Feb 02 18:41:12 1997 */
/* UPDATE trigger on Sinodales */
begin
    declare @numrows int,
            @nnullent int,
            @valident int,
            @insNo_registro int,
            @insRfc char(13),
            @errno int,
            @errmsg varchar(255)

    select @numrows = @@rowcount
    /* ERwin Builtin Sun Feb 02 18:41:12 1997 */
    /* Profesores Es Sinodales ON CHILD UPDATE
RESTRICT */
    if
        /* %ChildFK(" or",update) */
        update(Rfc)
    begin
        select @nnullent = 0
        select @valident = count(*)
        from inserted.Profesores

```

```

    where
        /* %JoinFKPK(inserted,Profesores) */
        inserted.Rfc = Profesores.Rfc
    /* %NotNullFK(inserted," is null","select @nnullent = count(*) from
inserted where"," and") */

    if (@valident + @nnullent != @numrows
    begin
        select @errno = 30007,
        @errmsg = "Cannot UPDATE "Sinodales" because
"Profesores" does not exist."
        goto error
    end
end
/* ERwin Builtin Sun Feb 02 18:41:12 1997 */
/* Tesis Contiene a Sinodales ON CHILD UPDATE RESTRICT */
if
    /* %ChildFK(" or",update) */
    update(No_registro)
begin
    select @nnullent = 0
    select @valident = count(*)
    from inserted.Tesis
    where
        /* %JoinFKPK(inserted,Tesis) */
        inserted.No_registro = Tesis.No_registro
    /* %NotNullFK(inserted," is null","select @nnullent = count(*) from
inserted where"," and") */

    if (@valident + @nnullent != @numrows
    begin
        select @errno = 30007,
        @errmsg = "Cannot UPDATE "Sinodales" because "Tesis"
does not exist."
        goto error
    end
end
/* ERwin Builtin Sun Feb 02 18:41:12 1997 */
return
error:
    raiserror @errno @errmsg
    rollback transaction
end
go

create trigger tD_Asignaturas on Asignaturas for DELETE as
/* ERwin Builtin Sun Feb 02 18:41:12 1997 */
/* DELETE trigger on Asignaturas */
begin
    declare @errno int,
            @errmsg varchar(255)
    /* ERwin Builtin Sun Feb 02 18:41:12 1997 */
    /* Asignaturas Tiene varios Grupos ON PARENT DELETE
RESTRICT */
    if exists (
        select * from deleted,Grupos
        where
            /* %JoinFKPK(Grupos,deleted," = "," and") */
            Grupos.Clave_de_asignatura = deleted.Clave_de_asignatura
    )

```

```

begin
  select @errno = 30001,
         @errmsg = "Cannot DELETE "Asignaturas" because
"Grupos" exists."
  goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Asignaturas Esta en un Exámenes_de_oposicion ON
PARENT DELETE RESTRICT */
if exists (
  select * from deleted,Exámenes_de_oposicion
  where
  /* "%JoinFKPK(Exámenes_de_oposicion,deleted," = ","
and") */
  Exámenes_de_oposicion.Clave_de_asignatura =
deleted.Clave_de_asignatura
)
begin
  select @errno = 30001,
         @errmsg = "Cannot DELETE "Asignaturas" because
"Exámenes_de_oposicion" exists."
  goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
return
error:
  raiserror @errno @errmsg
  rollback transaction
end
go

create trigger tl_Asignaturas on Asignaturas for INSERT as
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* INSERT trigger on Asignaturas */
begin
  declare @numrows int,
         @nullent int,
         @valident int,
         @errno int,
         @errmsg varchar(255)

  select @numrows = @@rowcount
  /* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
  /* Seminario Esta formado por Asignaturas ON CHILD
INSERT RESTRICT */
  if
  /* "%ChildFK(" or",.update) */
  update(Clave_de_seminario)
  begin
    select @nullent = 0
    select @valident = count(*)
    from inserted,Seminario
    where
    /* "%JoinFKPK(inserted,Seminario) */
    inserted.Clave_de_seminario =
Seminario.Clave_de_seminario
    /* "%NoNullFK(inserted," is null",select @nullent =
count(*) from inserted where," and") */
    select @nullent = count(*) from inserted where
    inserted.Clave_de_seminario is null

```

```

if @valident + @nullent != @numrows
begin
  select @errno = 30002,
         @errmsg = "Cannot INSERT "Asignaturas" because
"Seminarrio" does not exist."
  goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
return
error:
  raiserror @errno @errmsg
  rollback transaction
end
go

create trigger tl_Asignaturas on Asignaturas for UPDATE as
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* UPDATE trigger on Asignaturas */
begin
  declare @numrows int,
         @nullent int,
         @valident int,
         @insClave_de_asignatura char(4),
         @errno int,
         @errmsg varchar(255)

  select @numrows = @@rowcount
  /* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
  /* Asignaturas Tiene varios Grupos ON PARENT UPDATE
CASCADE */
  if
  /* "%ParentPK(" or",.update) */
  update(Clave_de_asignatura)
  begin
    if @numrows = 1
    begin
      select @insClave_de_asignatura = inserted.Clave_de_asignatura
      from inserted
      update Grupos
      set
      /* "%JoinFKPK(Grupos,@ins," = ","") */
      Grupos.Clave_de_asignatura = @insClave_de_asignatura
      from Grupos,inserted,deleted
      where
      /* "%JoinFKPK(Grupos,deleted," = "," and") */
      Grupos.Clave_de_asignatura = deleted.Clave_de_asignatura
    end
  else
  begin
    select @errno = 30006,
           @errmsg = "Cannot cascade "Asignaturas" UPDATE because
more than one row has been affected."
    raiserror @errno @errmsg
    end
  /* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
  /* Asignaturas Esta en un Exámenes_de_oposicion ON PARENT
UPDATE CASCADE */
  if

```

```

/* %ParentPK(" or",update) */
update(Clave_de_asignatura)
begin
  if @numrows = 1
  begin
    select @insClave_de_asignatura =
inserted.Clave_de_asignatura
    from inserted
    update Examenes_de_oposicion
    set
    /* %JoinFKPK(Examenes_de_oposicion, @ins, " = ", " ")
*/
    Examenes_de_oposicion.Clave_de_asignatura =
@insClave_de_asignatura
    from Examenes_de_oposicion, inserted, deleted
    where
    /* %JoinFKPK(Examenes_de_oposicion, deleted, " = ", " ")
and") */
    Examenes_de_oposicion.Clave_de_asignatura =
deleted.Clave_de_asignatura
  end
  else
  begin
    select @errno = 30006,
    @errmsg = 'Cannot cascade "Asignaturas" UPDATE
because more than one row has been affected.'
    raiserror @errno @errmsg
  end
end
/* ERwin BuiltIn Sun Feb 02 18:41:12 1997 */
/* Seminario Esta formado por Asignaturas ON CHILD
UPDATE RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave_de_seminario)
begin
  select @nullcnt = 0
  from inserted.Seminario
  where
  /* %JoinFKPK(inserted.Seminario) */
  inserted.Clave_de_seminario =
Seminario.Clave_de_seminario
  /* %NotNullFK(inserted, " is null", "select @nullcnt =
count(*) from inserted where", " and") */
  select @nullcnt = count(*) from inserted where
inserted.Clave_de_seminario is null
  if @validcnt + @nullcnt != @numrows
  begin
    select @errno = 30007,
    @errmsg = 'Cannot UPDATE "Asignaturas" because
"Seminario" does not exist.'
    goto error
  end
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
return
error:
raiserror @errno @errmsg
rollback transaction

```

```

end
go

create trigger tl_Res_Hist_Acad on Res_Hist_Acad for INSERT as
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* INSERT trigger on Res_Hist_Acad */
begin
  declare @numrows int,
    @nullcnt int,
    @validcnt int,
    @errno int,
    @errmsg varchar(255)

  select @numrows = @@rowcount
  /* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
  /* Alumnos Tiene un Res_Hist_Acad ON CHILD INSERT
RESTRICT */
  if
  /* %ChildFK(" or",update) */
  update(No_de_cuenta)
  begin
    select @nullcnt = 0
    select @validcnt = count(*)
    from inserted.Alumnos
    where
    /* %JoinFKPK(inserted, Alumnos) */
    inserted.No_de_cuenta = Alumnos.No_de_cuenta
    /* %NotNullFK(inserted, " is null", "select @nullcnt = count(*) from
inserted where", " and") */
    if @validcnt + @nullcnt != @numrows
    begin
      select @errno = 30002,
      @errmsg = 'Cannot INSERT "Res_Hist_Acad" because
"Alumnos" does not exist.'
      goto error
    end
  end
  /* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
  return
error:
raiserror @errno @errmsg
rollback transaction
end
go

create trigger tU_Res_Hist_Acad on Res_Hist_Acad for UPDATE as
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* UPDATE trigger on Res_Hist_Acad */
begin
  declare @numrows int,
    @nullcnt int,
    @validcnt int,
    @insNo_de_cuenta char(9),
    @errno int,
    @errmsg varchar(255)

  select @numrows = @@rowcount
  /* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */

```

```

/* Alumnos Tiene un Res_Hist_Acad ON CHILD UPDATE
RESTRICT */
if
/* %JoinFK(" or",update) */
update(No_de_cuenta)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Alumnos
where
/* %JoinFKPK(inserted,Alumnos) */
inserted.No_de_cuenta = Alumnos.No_de_cuenta
/* %NotnullFK(inserted," is null"," and" @nullent =
count(*) from inserted where," and") */
if @valident + @nullent != @numrows
begin
select @errno = 30007,
@errmsg = 'Cannot UPDATE "Res_Hist_Acad"
because "Alumnos" does not exist.'
goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
return
error:
raiserror @errno @errmsg
rollback transaction
end
go

create trigger tD_Grupos on Grupos for DELETE as
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* DELETE trigger on Grupos */
begin
declare @errno int,
@errmsg varchar(255)
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Grupos tiene Rectificacion ON PARENT DELETE
RESTRICT */
if exists (
select * from deleted,Rectificacion
where
/* %JoinFKPK(Rectificacion,deleted," = "," and") */
Rectificacion.Clave_de_grupo =
deleted.Clave_de_grupo
)
begin
select @errno = 30001,
@errmsg = 'Cannot DELETE "Grupos" because
"Rectificacion" exists.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Grupos Tiene Clase ON PARENT DELETE RESTRICT
*/
if exists (
select * from deleted,Clase
where
/* %JoinFKPK(Clase,deleted," = "," and") */

```

```

Clase.Clave_de_grupo = deleted.Clave_de_grupo
)
begin
select @errno = 30001,
@errmsg = 'Cannot DELETE "Grupos" because "Clase"
exists.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Grupos Forma un Contratos ON PARENT DELETE RESTRICT
*/
if exists (
select * from deleted,Contratos
where
/* %JoinFKPK(Contratos,deleted," = "," and") */
Contratos.Clave_de_grupo = deleted.Clave_de_grupo
)
begin
select @errno = 30001,
@errmsg = 'Cannot DELETE "Grupos" because "Contratos"
exists.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Grupos Es parte de una Asistencias ON PARENT DELETE
RESTRICT */
if exists (
select * from deleted,Asistencias
where
/* %JoinFKPK(Asistencias,deleted," = "," and") */
Asistencias.Clave_de_grupo = deleted.Clave_de_grupo
)
begin
select @errno = 30001,
@errmsg = 'Cannot DELETE "Grupos" because "Asistencias"
exists.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
return
error:
raiserror @errno @errmsg
rollback transaction
end
go

create trigger tI_Grupos on Grupos for INSERT as
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* INSERT trigger on Grupos */
begin
declare @numrows int,
@nullent int,
@valident int,
@errno int,
@errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Aula Alberga a Grupos ON CHILD INSERT RESTRICT */
if

```

```

/* %ChildFK(" or",update) */
update(Clave_aula)
begin
select @nullcnt = 0
select @validcnt = count(*)
from inserted,Aulas
where
/* %JoinFKPK(inserted,Aulas) */
inserted.Clave_aula = Aulas.Clave_aula
/* %NotnullFK(inserted," is null",select @nullcnt =
count(*) from inserted where"," and") */
select (@nullcnt = count(*) from inserted where
inserted.Clave_aula is null
if @validcnt + @nullcnt != @numrows
begin
select @errno = 30002,
@errmsg = 'Cannot INSERT "Grupos" because
"Aulas" does not exist.'
goto error
end
end
/* ERwin Built-in Sun Feb 02 18:41:13 1997 */
/* Asignaturas Tiene varios Grupos ON CHILD INSERT
RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave_de_asignatura)
begin
select @nullcnt = 0
select @validcnt = count(*)
from inserted,Asignaturas
where
/* %JoinFKPK(inserted,Asignaturas) */
inserted.Clave_de_asignatura =
Asignaturas.Clave_de_asignatura
/* %NotnullFK(inserted," is null",select @nullcnt =
count(*) from inserted where"," and") */
if @validcnt + @nullcnt != @numrows
begin
select @errno = 30002,
@errmsg = 'Cannot INSERT "Grupos" because
"Asignaturas" does not exist.'
goto error
end
end
/* ERwin Built-in Sun Feb 02 18:41:13 1997 */
return
error:
raiserror @errno @errmsg
rollback transaction
end
go

create trigger IU_Grupos on Grupos for UPDATE as
/* ERwin Built-in Sun Feb 02 18:41:13 1997 */
/* UPDATE trigger on Grupos */
begin
declare @numrows int,
@nullcnt int,

```

```

@validcnt int,
@insClave_de_grupo char(13),
@errno int,
@errmsg varchar(255)

select @numrows = @@@rowcount
/* ERwin Built-in Sun Feb 02 18:41:13 1997 */
/* Grupos tiene Rectificacion ON PARENT UPDATE CASCADE */
if
/* %ParentPK(" or",update) */
update(Clave_de_grupo)
begin
if @numrows = 1
begin
select @insClave_de_grupo = inserted.Clave_de_grupo
from inserted
update Rectificacion
set
/* %JoinFKPK(Rectificacion,@ins," ","") */
Rectificacion.Clave_de_grupo = @insClave_de_grupo
from Rectificacion,inserted,deleted
where
/* %JoinFKPK(Rectificacion,deleted," "," and") */
Rectificacion.Clave_de_grupo = deleted.Clave_de_grupo
end
else
begin
select @errno = 30006,
@errmsg = 'Cannot cascade "Grupos" UPDATE because
more than one row has been affected.'
raiserror @errno @errmsg
end
end
/* ERwin Built-in Sun Feb 02 18:41:13 1997 */
/* Grupos Tiene Clase ON PARENT UPDATE CASCADE */
if
/* %ParentPK(" or",update) */
update(Clave_de_grupo)
begin
if @numrows = 1
begin
select @insClave_de_grupo = inserted.Clave_de_grupo
from inserted
update Clase
set
/* %JoinFKPK(Clase,@ins," ","") */
Clase.Clave_de_grupo = @insClave_de_grupo
from Clase,inserted,deleted
where
/* %JoinFKPK(Clase,deleted," "," and") */
Clase.Clave_de_grupo = deleted.Clave_de_grupo
end
else
begin
select @errno = 30006,
@errmsg = 'Cannot cascade "Grupos" UPDATE because
more than one row has been affected.'
raiserror @errno @errmsg
end
end

```

```

/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Grupos Forma un Contratos ON PARENT UPDATE
CASCADE */
if
/* %ParentPK(" or",update) */
update(Clave_de_grupo)
begin
if @numrows = 1
begin
select @insClave_de_grupo = inserted.Clave_de_grupo
from inserted
update Contratos
set
/* %JoinFKPK(Contratos,@ins," ","","") */
Contratos.Clave_de_grupo = @insClave_de_grupo
from Contratos,inserted,deleted
where
/* %JoinFKPK(Contratos,deleted," ","" and") */
Contratos.Clave_de_grupo = deleted.Clave_de_grupo
end
else
begin
select @errno = 30006,
@errmsg = 'Cannot cascade "Grupos" UPDATE
because more than one row has been affected.'
raiserror @errno @errmsg
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Grupos Es parte de una Asistencias ON PARENT
UPDATE CASCADE */
if
/* %ParentPK(" or",update) */
update(Clave_de_grupo)
begin
if @numrows = 1
begin
select @insClave_de_grupo = inserted.Clave_de_grupo
from inserted
update Asistencias
set
/* %JoinFKPK(Asistencias,@ins," ","","") */
Asistencias.Clave_de_grupo = @insClave_de_grupo
from Asistencias,inserted,deleted
where
/* %JoinFKPK(Asistencias,deleted," ","" and") */
Asistencias.Clave_de_grupo = deleted.Clave_de_grupo
end
else
begin
select @errno = 30006,
@errmsg = 'Cannot cascade "Grupos" UPDATE
because more than one row has been affected.'
raiserror @errno @errmsg
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Aulas Alberga a Grupos ON CHILD UPDATE
RESTRICT */
if

```

```

/* %ChildFK(" or",update) */
update(Clave_aula)
begin
select @valident = 0
select @valident = count(*)
from inserted,Aulas
where
/* %JoinFKPK(inserted,Aulas) */
inserted.Clave_aula = Aulas.Clave_aula
/* %NotNullFK(inserted," is null","select @nullent = count(*) from
inserted where"," and") */
select @nullent = count(*) from inserted where
inserted.Clave_aula is null
if @valident + @nullent != @numrows
begin
select @errno = 30007,
@errmsg = 'Cannot UPDATE "Grupos" because
" Aulas" does
not exist.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Asignaturas Tiene varios Grupos ON CHILD UPDATE
RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave_de_asignatura)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Asignaturas
where
/* %JoinFKPK(inserted,Asignaturas) */
inserted.Clave_de_asignatura =
Asignaturas.Clave_de_asignatura
/* %NotNullFK(inserted," is null","select @nullent = count(*) from
inserted where"," and") */
if @valident + @nullent != @numrows
begin
select @errno = 30007,
@errmsg = 'Cannot UPDATE "Grupos" because
" Asignaturas" does not exist.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
return
error:
raiserror @errno @errmsg
rollback transaction
end
go

create trigger tl_Comisionado on Comisionado for INSERT as
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* INSERT trigger on Comisionado */
begin
declare @numrows int,
@nullent int,

```

```

@valident int,
@errno int,
@errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Comision Esta compuesta por Comisionado ON CHILD
INSERT RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Comision
where
/* %JoinFKPK(inserted,Comision) */
inserted.Clave = Comision.Clave
/* %NotNullFK(inserted," is null","select @nullent =
count(*) from inserted where"," and") */

if @valident + @nullent != @numrows
begin
select @errno = 30002,
@errmsg = 'Cannot INSERT "Comisionado" because
"Comision" does not exist.'
goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Profesores Se encuentra en Comisionado ON CHILD
INSERT RESTRICT */
if
/* %ChildFK(" or",update) */
update(Rfc)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotNullFK(inserted," is null","select @nullent =
count(*) from inserted where"," and") */

if @valident + @nullent != @numrows
begin
select @errno = 30002,
@errmsg = 'Cannot INSERT "Comisionado" because
"Profesores" does not exist.'
goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
return
error:
raiserror @errno @errmsg
rollback transaction
end
go

```

```

create trigger tU_Comisionado on Comisionado for UPDATE as
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* UPDATE trigger on Comisionado */
begin
declare @numrows int,
@nullent int,
@valident int,
@insClave tinyint,
@insRfc char(13),
@errno int,
@errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Comision Esta compuesta por Comisionado ON CHILD
UPDATE RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Comision
where
/* %JoinFKPK(inserted,Comision) */
inserted.Clave = Comision.Clave
/* %NotNullFK(inserted," is null","select @nullent = count(*) from
inserted where"," and") */

if @valident + @nullent != @numrows
begin
select @errno = 30007,
@errmsg = 'Cannot UPDATE "Comisionado" because
"Comision" does not exist.'
goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Profesores Se encuentra en Comisionado ON CHILD UPDATE
RESTRIC */
if
/* %ChildFK(" or",update) */
update(Rfc)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotNullFK(inserted," is null","select @nullent = count(*) from
inserted where"," and") */

if @valident + @nullent != @numrows
begin
select @errno = 30007,
@errmsg = 'Cannot UPDATE "Comisionado" because
"Profesores" does not exist.'
goto error

```

## Apéndice A

```

end
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
return
error:
raiserror @:errno @:errmsg
rollback transaction
end
go

create trigger tl_Candidatos_a_oposicion on
Candidatos_a_oposicion for INSERT as
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* INSERT trigger on Candidatos_a_oposicion */
begin
declare @numrows int,
        @nullent int,
        @valident int,
        @:errno int,
        @:errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Exámenes_de_oposicion Esta Inscrito en
Candidatos_a_oposicion ON CHIL INSERT RESTRICT */
if
/* %ChildFK(" or",update) */
update(Periodo) or
update(Clave_de_asignatura)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Exámenes_de_oposicion
where
/* %JoinFKPK(inserted,Exámenes_de_oposicion) */
inserted.Periodo = Exámenes_de_oposicion.Periodo
and
inserted.Clave_de_asignatura =
Exámenes_de_oposicion.Clave_de_asignatura
/* %NotNullFK(inserted," is null","select @nullent =
count(*) from inserted where"," and") */
if @valident + @nullent != @numrows
begin
select @:errno = 30002,
        @:errmsg = 'Cannot INSERT
"Candidatos_a_oposicion" because
"Exámenes_de_oposicion" does not exist.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Profesores Debe ser Candidatos_a_oposicion ON CHIL
INSERT RESTRICT */
if
/* %ChildFK(" or",update) */
update(Rfc)
begin
select @nullent = 0
select @valident = count(*)

```

```

from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotNullFK(inserted," is null","select @nullent = count(*) from
inserted where"," and") */

if @valident + @nullent != @numrows
begin
select @:errno = 30002,
        @:errmsg = 'Cannot INSERT "Candidatos_a_oposicion"
because "Profesores" does not exist.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
return
error:
raiserror @:errno @:errmsg
rollback transaction
end
go

create trigger tu_Candidatos_a_oposicion on
Candidatos_a_oposicion for UPDATE as
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* UPDATE trigger on Candidatos_a_oposicion */
begin
declare @numrows int,
        @nullent int,
        @valident int,
        @:insClave_de_asignatura char(4),
        @:insRfc char(13),
        @:insPeriodo char(5),
        @:errno int,
        @:errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Exámenes_de_oposicion Esta Inscrito en Candidatos_a_oposicion
ON CHIL UPDATE RESTRICT */
if
/* %ChildFK(" or",update) */
update(Periodo) or
update(Clave_de_asignatura)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Exámenes_de_oposicion
where
/* %JoinFKPK(inserted,Exámenes_de_oposicion) */
inserted.Periodo = Exámenes_de_oposicion.Periodo and
inserted.Clave_de_asignatura =
Exámenes_de_oposicion.Clave_de_asignatura
/* %NotNullFK(inserted," is null","select @nullent = count(*) from
inserted where"," and") */

if @valident + @nullent != @numrows
begin
select @:errno = 30007,

```

```

@errmsg = 'Cannot UPDATE
"Candidatos_a_oposicion" because
"Exámenes_de_oposicion" does not exist.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Profesores Debe ser Candidatos_a_oposicion ON CHILD
UPDATE RESTRICT */
if
/* %ChildFK(" or",update) */
update(Rfc)
begin
select @nnull = 0
select @valid = count(*)
from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotNullFK(inserted," is null","select @nnull =
count(*) from inserted where"," and") */
if @valid + @nnull != @numrows
begin
select @errno = 30007,
@errmsg = 'Cannot UPDATE
"Candidatos_a_oposicion" because "Profesores" does not
exist.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
return
error:
raiserror @errno @errmsg
rollback transaction
end
go

create trigger tl_Asiistencia_profesores_de_ca on
Asistencia_profesores_de_carre for INSERT as
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* INSERT trigger on Asistencia_profesores_de_carre */
begin
declare @numrows int,
@nnull int,
@valid int,
@errno int,
@errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Profesores Tiene Asistencia_profesores_de_carre ON
CHILD INSERT RESTRICT */
if
/* %ChildFK(" or",update) */
update(Rfc)
begin
select @nnull = 0
select @valid = count(*)
from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotNullFK(inserted," is null","select @nnull = count(*) from
inserted where"," and") */
if @valid + @nnull != @numrows
begin
select @errno = 30007,
@errmsg = 'Cannot UPDATE
"Asistencia_profesores_de_carre" because "Profesores" does not
exist.'

```

```

from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotNullFK(inserted," is null","select @nnull + count(*) from
inserted where"," and") */
if @valid + @nnull != @numrows
begin
select @errno = 30002,
@errmsg = 'Cannot INSERT
"Asistencia_profesores_de_carre" because "Profesores" does not
exist.'
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
return
error:
raiserror @errno @errmsg
rollback transaction
end
go

create trigger tl_Asiistencia_profesores_de_ca on
Asistencia_profesores_de_carre for UPDATE as
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* UPDATE trigger on Asistencia_profesores_de_carre */
begin
declare @numrows int,
@nnull int,
@valid int,
@insRfc char(13),
@insFecha_Falta smalldatetime,
@errno int,
@errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Profesores Tiene Asistencia_profesores_de_carre ON CHILD
UPDATE RESTRICT */
if
/* %ChildFK(" or",update) */
update(Rfc)
begin
select @nnull = 0
select @valid = count(*)
from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotNullFK(inserted," is null","select @nnull = count(*) from
inserted where"," and") */
if @valid + @nnull != @numrows
begin
select @errno = 30007,
@errmsg = 'Cannot UPDATE
"Asistencia_profesores_de_carre" because "Profesores" does not
exist.'

```

```

goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
return
error:
  raiserror @errno @errmsg
  rollback transaction
end
go

create trigger tD_Comision on Comision for DELETE as
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* DELETE trigger on Comision */
begin
  declare @errno int,
           @errmsg varchar(255)
  /* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
  /* Comision Es asignado a una comision
  Exámenes_de_oposicion ON PARENT DELETE RESTRICT */
  if exists (
    select * from deleted,Exámenes_de_oposicion
    where
    /* %JoinFKPK(Exámenes_de_oposicion,deleted," = "," and") */
    and") */
    Exámenes_de_oposicion.Clave = deleted.Clave
  )
  begin
    select @errno = 30001,
           @errmsg = 'Cannot DELETE "Comision" because
  "Exámenes_de_oposicion" exists.'
    goto error
  end
  /* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
  /* Comision Esta compuesta por Comisionado ON
  PARENT DELETE RESTRICT */
  if exists (
    select * from deleted,Comisionado
    where
    /* %JoinFKPK(Comisionado,deleted," = "," and") */
    Comisionado.Clave = deleted.Clave
  )
  begin
    select @errno = 30001,
           @errmsg = 'Cannot DELETE "Comision" because
  "Comisionado" exists.'
    goto error
  end
  /* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
  return
error:
  raiserror @errno @errmsg
  rollback transaction
end
go

create trigger tU_Comision on Comision for UPDATE as
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* UPDATE trigger on Comision */

```

```

begin
declare @numrows int,
        @nullent int,
        @valident int,
        @insClave tinyint,
        @errno int,
        @errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Comision Es asignado a una comision Exámenes_de_oposicion
ON PARENT UPDATE RESTRICT */
if
  /* %ParentFK(" or",update) */
  update(Clave)
begin
  if exists (
    select * from deleted,Exámenes_de_oposicion
    where
    /* %JoinFKPK(Exámenes_de_oposicion,deleted," = "," and") */
    )
  )
  begin
    select @errno = 30005,
           @errmsg = 'Cannot UPDATE "Comision" because
  "Exámenes_de_oposicion" exists.'
    goto error
  end
  /* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
  /* Comision Esta compuesta por Comisionado ON PARENT
  UPDATE CASCADE */
  if
    /* %ParentFK(" or",update) */
    update(Clave)
  begin
    if @numrows = 1
    begin
      select @insClave = inserted.Clave
        from inserted
        update Comisionado
        set
        /* %JoinFKPK(Comisionado,@ins," = ","") */
        Comisionado.Clave = @insClave
        from Comisionado,inserted,deleted
        where
        /* %JoinFKPK(Comisionado,deleted," = "," and") */
        Comisionado.Clave = deleted.Clave
    end
    else
    begin
      select @errno = 30006,
             @errmsg = 'Cannot cascade "Comision" UPDATE because
  more than one row has been affected.'
        raiserror @errno @errmsg
    end
  end
  /* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
  return
error:

```

```

raiserror (@@erno @errmsg
rollback transaction
end
go

create trigger tI_Rectificacion on Rectificacion for INSERT
as
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* INSERT trigger on Rectificacion */
begin
declare @numrows int,
        @nullent int,
        @valident int,
        @errno int,
        @errmsg varchar(255)

select @numrows = @@@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Alumnos Solicita una Rectificacion ON CHILD INSERT
RESTRICT */
if
/* %ChildFK(" or",update) */
update(No_de_cuenta)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Alumnos
where
/* %JoinFKPK(inserted,Alumnos) */
inserted.No_de_cuenta = Alumnos.No_de_cuenta
/* %NotNullFK(inserted," is null","select @nullent =
count(*) from inserted where"," and") */

if @valident + @nullent != @numrows
begin
select @errno = 30002,
        @errmsg = "Cannot INSERT "Rectificacion" because
"Alumnos" does not exist."
goto error
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Profesores Realiza una Rectificacion ON CHILD
INSERT RESTRICT */
if
/* %ChildFK(" or",update) */
update(Rfc)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotNullFK(inserted," is null","select @nullent =
count(*) from inserted where"," and") */

if @valident + @nullent != @numrows
begin
select @errno = 30002,

```

```

        @errmsg = "Cannot INSERT "Rectificacion" because
"Profesores" does not exist."
goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Grupos tiene Rectificacion ON CHILD INSERT RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave_de_grupo)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Grupos
where
/* %JoinFKPK(inserted,Grupos) */
inserted.Clave_de_grupo = Grupos.Clave_de_grupo
/* %NotNullFK(inserted," is null","select @nullent = count(*) from
inserted where"," and") */

if @valident + @nullent != @numrows
begin
select @errno = 30002,
        @errmsg = "Cannot INSERT "Rectificacion" because
"Grupos" does not exist."
goto error
end
end
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
return
error:
raiserror (@@erno @errmsg
rollback transaction
end
go

create trigger tU_Rectificacion on Rectificacion for UPDATE as
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* UPDATE trigger on Rectificacion */
begin
declare @numrows int,
        @nullent int,
        @valident int,
        @insClave_de_grupo char(13),
        @insNo_de_cuenta char(9),
        @errno int,
        @errmsg varchar(255)

select @numrows = @@@rowcount
/* ERwin BuiltIn Sun Feb 02 18:41:13 1997 */
/* Alumnos Solicita una Rectificacion ON CHILD UPDATE
RESTRICT */
if
/* %ChildFK(" or",update) */
update(No_de_cuenta)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Alumnos
where

```

## Apéndice A

```

/* %JoinFKPK(inserted,Alumnos) */
inserted.No_de_cuenta = Alumnos.No_de_cuenta
/* %NotNullFK(inserted," is null","select @null =
count(*) from inserted where"," and") */

if @valid = @null != @numrows
begin
select @erno = 30007,
@errmsg = 'Cannot UPDATE "Rectificacion"
because "Alumnos" does not exist.'
goto error
end
end
/* ERwin Built-in Sun Feb 02 18:41:13 1997 */
/* Profesores Realiza una Rectificacion ON CHILD
UPDATE RESTRICT */
if
/* %ChildFK(" or",update) */
update(Rfc)
begin
select @null = 0
select @valid = count(*)
from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotNullFK(inserted," is null","select @null =
count(*) from inserted where"," and") */

if @valid + @null != @numrows
begin
select @erno = 30007,
@errmsg = 'Cannot UPDATE "Rectificacion"
because "Profesores" does not exist.'
goto error
end
end
/* ERwin Built-in Sun Feb 02 18:41:13 1997 */
/* Grupos tiene Rectificacion ON CHILD UPDATE
RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave_de_grupo)
begin
select @null = 0
select @valid = count(*)
from inserted,Grupos
where
/* %JoinFKPK(inserted,Grupos) */
inserted.Clave_de_grupo = Grupos.Clave_de_grupo
/* %NotNullFK(inserted," is null","select @null =
count(*) from inserted where"," and") */

if @valid + @null != @numrows
begin
select @erno = 30007,
@errmsg = 'Cannot UPDATE "Rectificacion"
because "Grupos" does not exist.'
goto error
end

```

```

end
/* ERwin Built-in Sun Feb 02 18:41:13 1997 */
return
error
raiserror @erno @errmsg
rollback transaction
end
go

create trigger tl_Clase on Clase for INSERT as
/* ERwin Built-in Sun Feb 02 18:41:13 1997 */
/* INSERT trigger on Clase */
begin
declare @numrows int,
@null int,
@valid int,
@erno int,
@errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin Built-in Sun Feb 02 18:41:13 1997 */
/* Profesores Imparte una Clase ON CHILD INSERT RESTRICT */
if
/* %ChildFK(" or",update) */
update(Rfc)
begin
select @null = 0
select @valid = count(*)
from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotNullFK(inserted," is null","select @null = count(*) from
inserted where"," and") */

if @valid + @null != @numrows
begin
select @erno = 30002,
@errmsg = 'Cannot INSERT "Clase" because "Profesores"
does not exist.'
goto error
end
end
/* ERwin Built-in Sun Feb 02 18:41:13 1997 */
/* Grupos Tiene Clase ON CHILD INSERT RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave_de_grupo)
begin
select @null = 0
select @valid = count(*)
from inserted,Grupos
where
/* %JoinFKPK(inserted,Grupos) */
inserted.Clave_de_grupo = Grupos.Clave_de_grupo
/* %NotNullFK(inserted," is null","select @null = count(*) from
inserted where"," and") */

if @valid + @null != @numrows
begin

```

```

select @errno = 30002,
       @errmsg = 'Cannot INSERT "Clase" because
"Grupos" does not exist.'
goto error
end
/* ERwin Builtin Sun Feb 02 18:41:13 1997 */
return
error:
raiserror @:errno @:errmsg
rollback transaction
end
go

create trigger tU!_Clase on Clase for UPDATE as
/* ERwin Builtin Sun Feb 02 18:41:13 1997 */
/* UPDATE trigger on Clase */
begin
declare @numrows int,
        @nullent int,
        @valident int,
        @insRfc char(13),
        @insClave_de_grupo char(13),
        @insFecha_inicio smalldatetime,
        @errno int,
        @errmsg varchar(255)

select @numrows = @@rowcount
/* ERwin Builtin Sun Feb 02 18:41:13 1997 */
/* Profesores Imparte una Clase ON CHILD UPDATE
RESTRICT */
if
/* %ChildFK(" or",update) */
update(Rfc)
begin
select @nullent = 0
select @valident = count(*)
from inserted,Profesores
where
/* %JoinFKPK(inserted,Profesores) */
inserted.Rfc = Profesores.Rfc
/* %NotnullFK(inserted," is null","select @nullent =
count(*) from inserted where"," and") */

if @valident + @nullent != @numrows
begin
select @:errno = 30007,
       @:errmsg = 'Cannot UPDATE "Clase" because
"Profesores" does not exist.'
goto error
end
/* ERwin Builtin Sun Feb 02 18:41:13 1997 */
/* Grupos Tiene Clase ON CHILD UPDATE RESTRICT */
if
/* %ChildFK(" or",update) */
update(Clave_de_grupo)
begin
select @nullent = 0
select @valident = count(*)

```

```

from inserted,Grupos
where
/* %JoinFKPK(inserted,Grupos) */
inserted.Clave_de_grupo = Grupos.Clave_de_grupo
/* %NotnullFK(inserted," is null","select @nullent = count(*) from
inserted where"," and") */

if @valident + @nullent != @numrows
begin
select @:errno = 30007,
       @:errmsg = 'Cannot UPDATE "Clase" because "Grupos" does
not exist.'
goto error
end
/* ERwin Builtin Sun Feb 02 18:41:13 1997 */
return
error:
raiserror @:errno @:errmsg
rollback transaction
end
go

```

**APENDICE B**  
**Manual de Usuario**

**EJECUCION DEL SISTEMA.**

1. Entre al ambiente Windows 95 (si es Windows bajo dos, teclee win desde el sistema operativo).
2. Del administrador de programas. Seleccione el subdirectorio Derecho.
3. Seleccione oprimiendo dos veces el botón izquierdo del Mouse el icono correspondiente a Derecho.

Al realizar la tercera acción el sistema mostrará la pantalla de la **Ilustración B 1**:



**Ilustración B 1**

Seleccione Archivo de la barra de menú principal, ubicada en la parte superior izquierda de la pantalla. Al efectuar esta acción el sistema mostrará un menú desplegable.

Seleccione la opción Conectarse y el sistema desplegará la pantalla de captura. Que aparece en la **Ilustración B 2**.

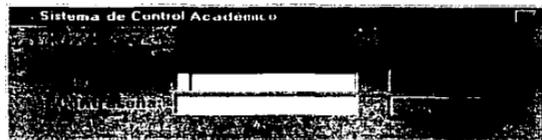


Ilustración B 2

Esta pantalla permitirá introducir una clave de usuario y una contraseña para poder acceder al módulo de control de profesores, para ello realice los siguientes pasos:

1. Seleccione la casilla correspondiente a usuario, teclee la clave de usuario.
2. Seleccione ahora la casilla correspondiente a la contraseña, escriba la contraseña.
3. Para continuar seleccione el botón "Aceptar"; en caso contrario seleccione el botón "Cancelar"

Si su contraseña y clave de usuario está correcta el sistema desplegará la pantalla de la **Ilustración B 3**, en caso contrario el sistema desplegará una pantalla de error y usted deberá volver a introducir los datos.



Ilustración B 3

Si introdujo bien la clave de usuario el sistema mostrará la pantalla de la **Ilustración B 4**.

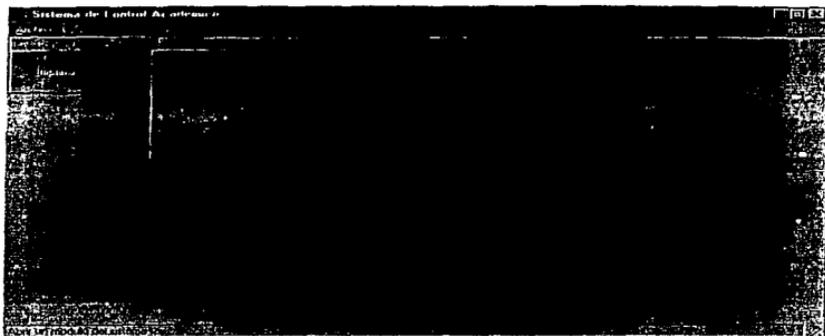


Ilustración B 4

Seleccione con el Mouse el módulo que desea utilizar.

## ACTUALIZACION DE PROFESORES.

Si seleccionó profesores en seguida aparecerá la pantalla que aparecerá en la **Ilustración B 5**.



Ilustración B 5

Esta pantalla permitirá actualizar la información correspondiente a los profesores, para ello siga los siguientes pasos:

### Alta de Profesores.

1. Seleccione el botón "Nuevo". La pantalla mostrará todas las casillas en blanco (**Ilustración B 5**) en las cuales podrá capturar la información del nuevo profesor.
2. Seleccione la casilla correspondiente al RFC del profesor , escriba el registro federal de contribuyentes del profesor, en caso de no llenar esta casilla, el sistema mostrará una ventana que le especifica que la casilla no puede quedarse vacía (**Ilustración 6**), y no lo dejará capturar otra información hasta que esta información sea proporcionada

correctamente. En caso de existir ya el RFC del profesor, el sistema mostrará la información que se tiene de ese profesor (**Ilustración B 7**) y le permitirá hacer cambios o capturar la información restante, si así lo desea.

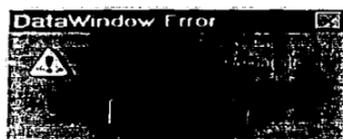


Ilustración B 6

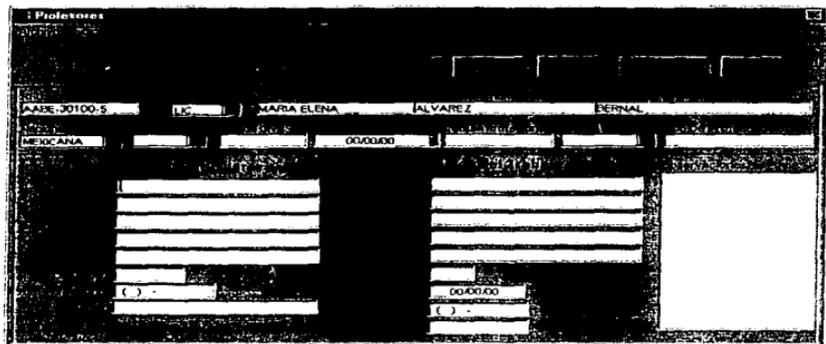


Ilustración B 7

3. Seleccione la casilla correspondiente al grado del profesor, puede hacerlo seleccionando la casilla con el mouse u oprimiendo la tecla TAB (tabulador) del teclado, señale la flecha que se muestra en la casilla, oprima el botón izquierdo del mouse y obtendrá una lista desplegable de la cual deberá seleccionar la opción que desee (**Ilustración B 8**), esta acción la deberá repetir en los campos de Nacionalidad, sexo y profesor activo.

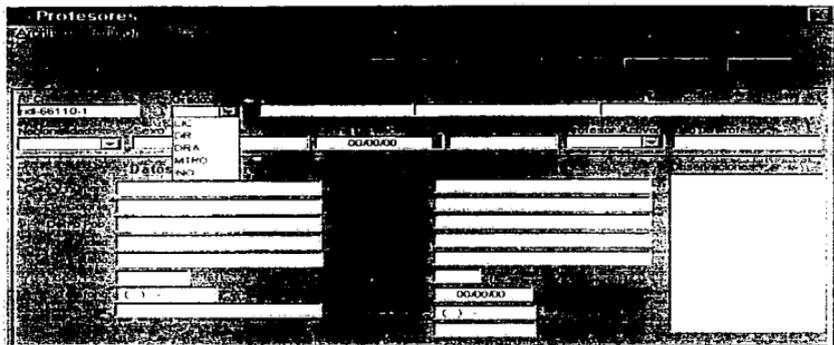


Ilustración B 8

4. Seleccione con el mouse la casilla nombre del profesor y capture (el nombre del profesor no debe exceder de 20 caracteres), de igual forma debe de llenar todos los demás campos.

Nota: las casillas correspondientes a fechas deben ser llenadas con el formato **día/mes/año**.

5. Al terminar de capturar la información, si desea guardar los cambios en la base de datos deberá seleccionar el botón "Guardar" de la pantalla que proporciona el sistema.
6. Si desea dar de alta a varios profesores deberá repetir todos los pasos anteriores dependiendo de la cantidad de profesores a dar de alta. Si desea salir del sistema seleccione el botón "Salir".

#### **Baja de Profesores.**

1. Seleccione la casilla correspondiente al RFC del profesor. Capture el registro federal de contribuyentes. En caso de existir información de este el sistema mostrara la información que se tiene de ese profesor.

2. Seleccione el botón "Eliminar".
3. Seleccione el botón "Guardar".
4. Selecciones el botón "Salir" si desea terminar la acción.

#### Búsqueda de Profesores.

En caso de no conocer el registro federal de contribuyentes de un profesor, se puede hacer la búsqueda del profesor por su nombre para realizar bajas o modificaciones del mismo, para poder realizar este procedimiento siga los siguientes pasos:

1. Seleccione el botón "Buscar". Al realizar esta acción el sistema mostrará la **Ilustración B 9**.

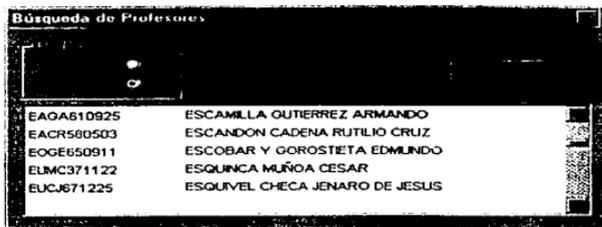
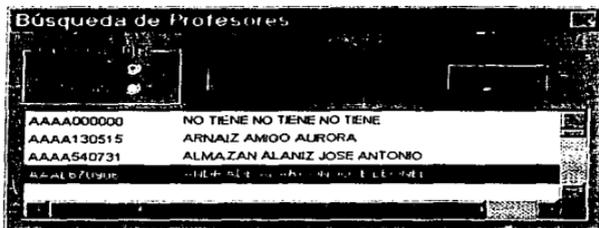


Ilustración B 9

2. Seleccione el botón "Cancelar" en caso de no querer realizar la búsqueda.
3. Seleccione con el Mouse la forma de ordenamiento. Si selecciona la opción "Nombre" el sistema muestra la información ordenada por nombre del profesor (**Ilustración B 9**).
4. Seleccione el nombre del profesor a buscar señalando con el puntero del Mouse en la opción deseada y después oprima el botón izquierdo del Mouse (**Ilustración B 10**). Si la lista no contiene la información que desea, puede visualizar la demás información, haciendo clic a la flecha superior o inferior según sea el caso, con el puntero del Mouse



6. En caso de seleccionar la forma de ordenamiento "R.F.C." el sistema despliega la información ordenada por el registro federal de contribuyentes (**Ilustración B 11**).



**Ilustración B 12**

7. En este caso el procedimiento para seleccionar el profesor a buscar es el mismo descrito en los puntos anteriores.
8. Seleccione el botón "Guardar" al terminar de capturar su información.
9. Seleccione el botón "Salir" para finalizar.

## ACTUALIZACION DE CONTRATOS.

Al seleccionar la opción de contratos el sistema muestra la pantalla de la **Ilustración B 13** que permite la generación y actualización de contratos de los profesores. Esta pantalla al igual que la de profesores permite realizar altas, bajas, cambios y búsquedas de los profesores que son contratados y darán clases en el semestre que inicia

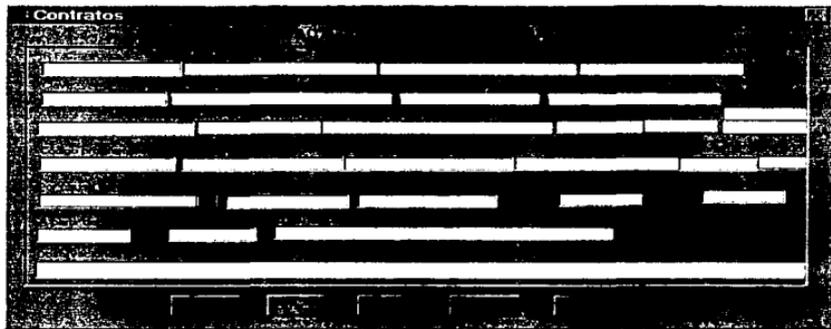


Ilustración B 13

### Alta de Contratos.

1. Seleccione con el Mouse la casilla correspondiente al RFC del profesor, en caso de existir información de este profesor en la base de datos el sistema desplegará la información; en caso de no existir información el sistema dejará las casillas en blanco para comenzar a capturar la información de un nuevo profesor, para esto realice las siguiente acciones.
2. Seleccione con el Mouse la casilla correspondiente a "Clave". Al estar ubicado en esta casilla deberá necesariamente capturar la información correspondiente ya que de no ser

así el sistema le mostrará la ventana mostrada en la ilustración 14 y no le permitirá ubicarse en otra casilla hasta que introduzca la información correcta.



Ilustración B 14

- Después de haber capturado la clave seleccione una a una las siguientes casillas y capture la información correspondiente a cada una de ellas.
- Para el caso de las casillas correspondientes a "Por causa de", "Personal", "Solicitud" y "Categoría", señale la flecha que aparece en la casilla con el puntero del Mouse, oprima una vez el botón izquierdo del Mouse, al realizar esta acción el sistema desplegará una lista en la que se deberá seleccionar con el Mouse la opción requerida.

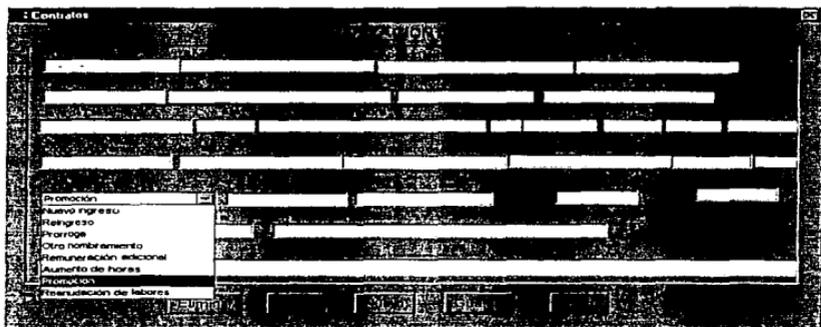
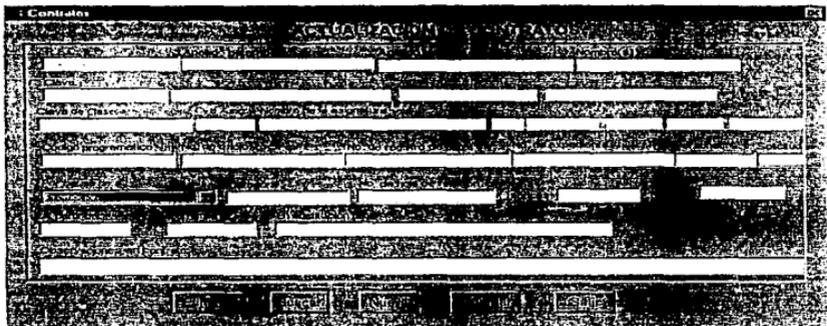


Ilustración B 15

5. Después de realizar el punto 4 el sistema actualiza la casilla como se observa en la **Ilustración B 16.**



**Ilustración B 16**

6. Para salvar los cambios efectuados en la pantalla seleccione el botón "Guardar" y posteriormente el botón "Salir".
7. Tantas veces como sea necesario si desea capturar o actualizar la información de varios profesores deberá realizar el mismo procedimiento.

#### **Baja de Contratos.**

1. Seleccione la casilla correspondiente al R.F.C. capture con el teclado la información correspondiente al registro federal de contribuyentes del profesor. Al efectuar esta acción el sistema desplegará la información correspondiente al R.F.C. que se capturó.
2. Seleccione el botón "Eliminar".
3. Seleccione el botón "Guardar" para guardar los cambios en la base de datos.

4. Seleccione el botón "Salir" si desea terminar.

#### **Búsqueda de Contratos.**

En caso de no conocer el R.F.C. del profesor se puede realizar una búsqueda de contratos, para ello realice las siguientes actividades:

1. Seleccione el botón "Buscar" con el Mouse. Al efectuar esta acción el sistema desplegará la pantalla de la **Ilustración B 17**.



**Ilustración B 17**

2. Seleccione la forma que desea ordenar la información que aparece en la pantalla. Esto es seleccione "R.F.C." o "Paterno" o "Nombre".
3. Seleccione con el Mouse la casilla del contrato a buscar y capture el contrato a buscar
4. Seleccione el botón de "Aceptar" para realizar la búsqueda.
5. Seleccione el botón "Salir" si desea terminar.

## ACTUALIZACION DE CATEGORIAS.

1. Al seleccionar el botón de Categorías el sistema desplegará la pantalla de captura de la **Ilustración B 18.**

The screenshot shows a window titled 'Categorías' with a sub-header 'ACTUALIZACION DE CATEGORIAS'. The main area contains five empty text input fields. At the bottom, there are five buttons: 'Eliminar', 'Buscar', 'Nuevo', 'Modificar', and 'Salir'.

Ilustración B 18

The screenshot shows the same window as in the previous illustration, but with data entered in the input fields. The first field contains 'D-16593', the second 'PROF. E INV. ORDINARIOS DE CARRERA', the third 'TITULAR B T C', and the fourth '5061 6'. The buttons at the bottom are 'Eliminar', 'Buscar', 'Nuevo', 'Modificar', and 'Salir'.

Ilustración B 19

2. Para dar de alta una nueva categoría, seleccione el botón "Nuevo" con el Mouse, al realizar esta acción el sistema mostrará la pantalla anterior con las casillas en blanco (Ilustración B 18).

#### **Alta de Categorías.**

1. Seleccione con el Mouse la casilla correspondiente a "Clave de categoría". Si la información correspondiente a esta casilla ya existe en la base de datos el sistema desplegará la información contenida en la base de datos, la cual podrá ser modificada en caso de así requerirlo (Ilustración B 19). En caso de no existir información el sistema dejará las casillas en blanco para comenzar a capturar la información de una nueva categoría, para esto realice las siguientes acciones.
2. Seleccione con el Mouse cada casilla, capturando con el teclado la información que se pide en la pantalla.
3. Para salvar los cambios realizados o la alta del profesor seleccione el botón "Guardar".
4. Seleccione el botón "Salir" si desea terminar.

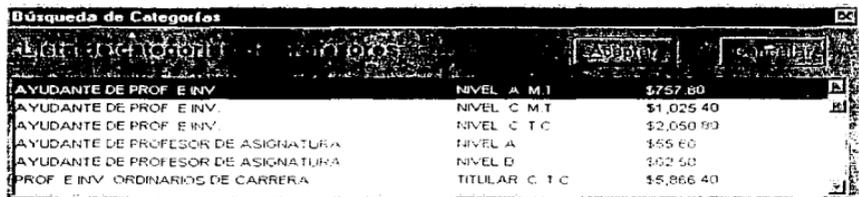
#### **Baja de Categorías.**

1. Para dar de baja la información correspondiente a una categoría, seleccione con el Mouse la casilla correspondiente a la "Clave de categoría", como ya se mencionó anteriormente el sistema desplegará la información que se tiene de esa clave de categoría.
2. Seleccione el botón "Eliminar".
3. Seleccione el botón "Guardar" para actualización de la base de datos.
4. Seleccione el botón "Salir" si desea salir del sistema.

#### **Búsqueda de Categorías.**

Si desea eliminar o realizar cambios en alguna categoría y desconoce la clave de categoría, el sistema le proporciona una opción de búsqueda de categoría, para realizar esta acción siga los siguientes pasos:

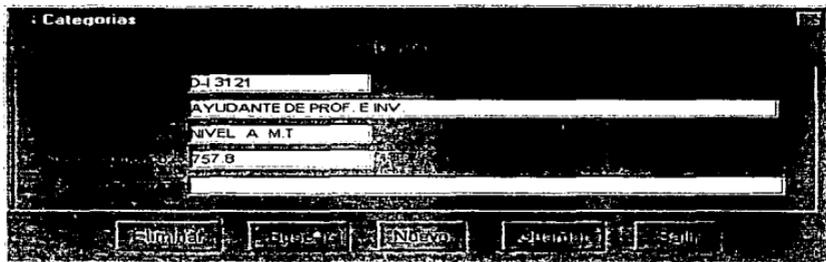
1. Seleccione con el Mouse el botón "Buscar". El sistema desplegará en seguida la pantalla de la **Ilustración B 20**.



Búsqueda de Categorías		
AYUDANTE DE PROF. E INV.	NIVEL A M.T	\$757.80
AYUDANTE DE PROF. E INV.	NIVEL C M.T	\$1,025.40
AYUDANTE DE PROF. E INV.	NIVEL C T.C	\$2,050.80
AYUDANTE DE PROFESOR DE ASIGNATURA	NIVEL A	165.00
AYUDANTE DE PROFESOR DE ASIGNATURA	NIVEL D	302.50
PROF. E INV. ORDINARIOS DE CARRERA	TITULAR C. T.C	\$5,866.40

Ilustración B 20

2. Seleccione con el Mouse el nombre de la categoría a buscar.
3. Seleccione con el Mouse el botón "Aceptar". Al efectuar esta acción el sistema desplegará la pantalla de la **Ilustración B 21**.



**Categorías**

D-13121

AYUDANTE DE PROF. E INV.

NIVEL A M.T

757.8

Aceptar    Buscar    Nuevo    Eliminar    Salir

Ilustración B 21

4. Realice los cambios necesarios, seleccionando con el Mouse cada casilla a modificar u oprima el botón eliminar si desea dar de baja la información de toda la categoría.
5. Seleccione el botón "Guardar" para actualizar la base de datos.
6. Seleccione el botón "Salir" si desea concluir.

## ACTUALIZACION DE ASISTENCIAS.

Al seleccionar la opción de asistencias el sistema mostrará la pantalla de captura de la **Ilustración B 22**, con la cual se podrá llevar a cabo la actualización de la información de la inasistencias de los profesores en la base de datos.

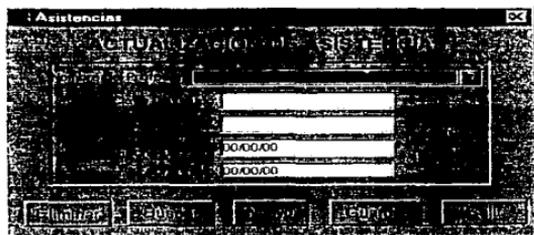


Ilustración B 22

### Alta de Inasistencias.

1. Señale la flecha que aparece en la casilla correspondiente al nombre del profesor, oprima el botón izquierdo del Mouse. Al realizar esta acción el sistema desplegará una lista de profesores como se muestra en la **Ilustración B 23**.

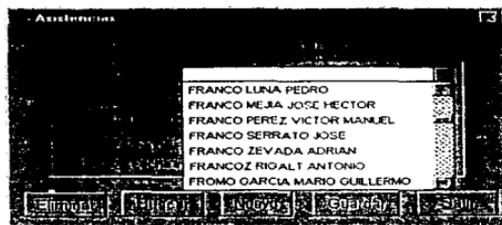


Ilustración B 23

2. Seleccione con el Mouse el nombre del profesor, en caso de no encontrarse en la lista, puede deslizarse sobre la lista de la siguiente manera: a) Señale con el puntero del Mouse una de las flechas (según se desee deslizarse hacia arriba o abajo), oprima el botón izquierdo del Mouse tantas veces como sea necesario hasta localizar el nombre del profesor a seleccionar ó b) Señale con el puntero del Mouse el botón que aparece en medio de las dos flechas, oprima el botón izquierdo del Mouse, sin dejar de oprimir el botón del Mouse, mueva el Mouse de tal forma que el botón que se encuentra en medio de las flechas se deslice hacia arriba o abajo hasta localizar el nombre del profesor a seleccionar.
3. Señale con el puntero del Mouse el nombre del profesor, oprima el botón izquierdo del Mouse, al realizar esta acción el sistema mostrará la información de la **Ilustración B 24**.

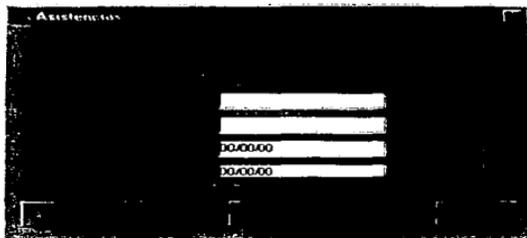


Ilustración B 24

4. Seleccione la casilla correspondiente a la Clave de Grupo, seleccione el botón "Buscar". Al realizar esta acción el sistema desplegará una pantalla de búsqueda en la cual deberá capturar el periodo y seleccionar el botón "Buscar", al seleccionar el botón "Buscar" el sistema desplegará una lista de la cual deberá seleccionar la información que desea.

#### **Baja de Inasistencias.**

1. Seleccione con el Mouse el botón "Buscar". Al realizar esta acción el sistema desplegará una pantalla de búsqueda en la cual deberá capturar el periodo y seleccionar el botón

“Aceptar”, al seleccionar el este botón el sistema desplegará una lista de la cual deberá seleccionar la información que desea.

2. El sistema mostrará la información que la base de datos contiene de ese profesor.
3. Seleccione con el Mouse el botón “Eliminar”.
4. Seleccione con el Mouse el botón “Guardar” para salvar los cambios.
5. Seleccione el botón “Salir” para finalizar.

## ASISTENCIA DE PROFESORES DE CARRERA.

Al seleccionar la opción Asistencia de profesores de carrera el sistema muestra la pantalla de la **Ilustración B 25** la cual permite actualizar en la base de datos las inasistencias de los profesores de carrera. Para efectuar la actualización de esta información se deben de seguir los siguientes pasos:

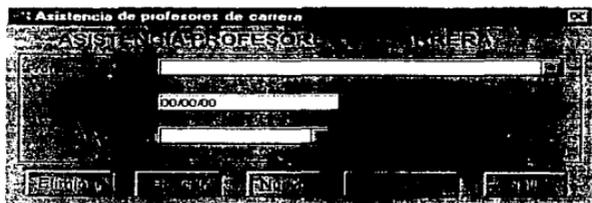


Ilustración B 25

### Alta de Inasistencia.

1. Para capturar la información correspondiente a las casillas de Profesor de carrera y Tipo de falta realice lo siguiente para ambos casos. Seleccione la flecha, al realizar esta acción el sistema desplegará una lista de Nombres de Profesores o Tipo de Falta según sea el caso **Ilustración B 26**.

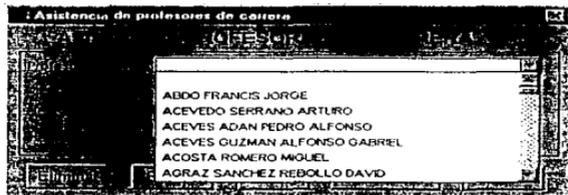


Ilustración B 26

2. Seleccione con el Mouse el nombre del profesor o el tipo de falta según sea el caso, en caso de que en la lista no aparezca la información deseada, puede visualizarla, seleccionando una de las flechas o el botón que se encuentra en medio de las flechas para deslizarse en la lista.
3. Para capturar la información correspondiente a la casilla "Fecha falta" se debe utilizar el formato día/mes/año.
4. Después de haber capturado toda la información Seleccione el botón "Guardar" para actualizar la base de datos.
5. Seleccione el botón "Salir" si desea finalizar.

**Baja de Inasistencia.**

1. Seleccione el botón "Buscar". Al efectuar esta acción el sistema mostrará una pantalla de búsqueda que le permitirá visualizar la información correspondiente al profesor seleccionado.
2. Al visualizar esta información seleccione el botón "Eliminar".
3. El sistema dará de baja la inasistencia del profesor, para actualizar la información en la base de datos, seleccione el botón "Guardar".
4. Seleccione el botón "Salir" en caso de querer finalizar.

## ACTUALIZACION DE GRUPOS.

Al seleccionar la opción de Grupos el sistema permite la actualización (Altas, Bajas, Cambios y Búsquedas de información de los grupos) que se manejarán en el semestre, mostrando la pantalla de captura y actualización de información de la **Ilustración B 27**.

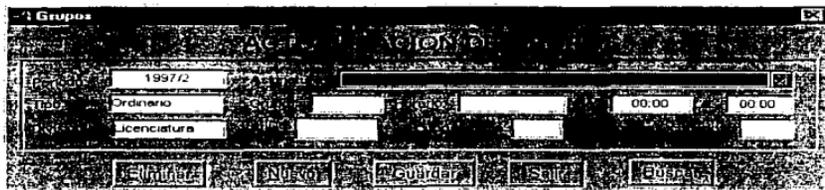


Ilustración B 27

### Alta de Grupos.

1. Seleccione la casilla correspondiente a "Asignatura". Señale con el puntero del Mouse la flecha que aparece en la casilla prestone el botón izquierdo del Mouse, al realizar esta acción el sistema muestra una lista de las asignaturas (**Ilustración B 28**), de la cual, seleccione con el Mouse la asignatura deseada. Repita el mismo procedimiento para llenar la casilla correspondiente a Horario (**Ilustración B 29**).

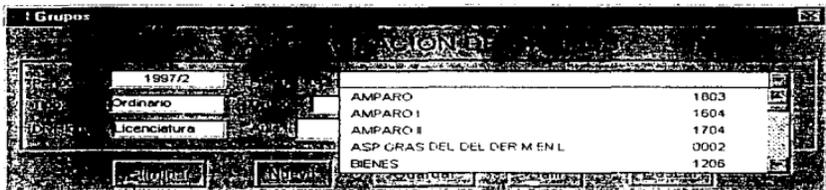


Ilustración B 28

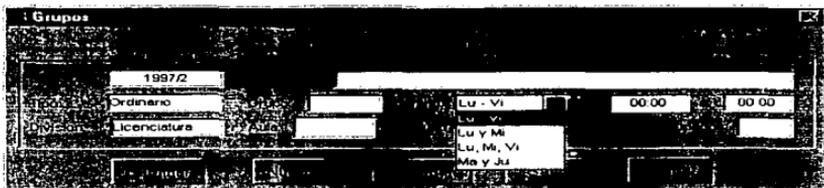


Ilustración B 29

2. Seleccione la casilla correspondiente a Grupo, teclee el no. de grupo a dar de alta, esta información es requerida por el sistema y si no es proporcionada, el sistema desplegará una pantalla pidiendo requerimiento de información como las mencionadas anteriormente.
3. Seleccione y capture con el teclado los demás campos restantes.
4. Si la información correspondiente a las casillas de "Asignatura", "Grupo", "Horario", "de", "a" es igual a la contenida en la base de datos, es decir el grupo ya está dado de alta, el sistema mostrará en la pantalla la información que se tiene de ese grupo y permitirá modificarla si así se requiere.
5. Seleccione el botón "Guardar" para actualizar la base de datos.
6. Si desea capturar otro grupo, seleccione el botón "nuevo". El sistema le mostrará las casillas en blanco para capturar la nueva información.
7. Seleccione el botón "Salir" si desea concluir.

#### **Búsqueda, Cambio y Baja de Grupos.**

1. Seleccione el botón "Buscar". Al seleccionar este botón el sistema desplegará una pantalla que le permitirá realizar una búsqueda (Ilustración B 30) en la base de datos de la información del grupo que desea dar de baja o modificar.

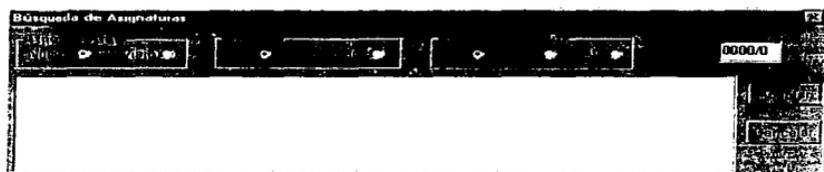


Ilustración B 30

2. Seleccione la casilla correspondiente a "Periodo". Capture con el teclado el periodo en el que se encuentra el grupo a buscar.
3. Para los casos de "Tipo de Plan", "División" y "Tipo de periodo", seleccione para cada caso con el Mouse la opción deseada, al efectuar esta acción el sistema desplegará la información contenida en la base de datos como se muestra en la **Ilustración B 31**. Seleccione de la lista la información deseada.



Ilustración B 31

4. Seleccione el botón "Aceptar" y el sistema mostrará en la pantalla la información que se estaba buscando (**Ilustración B 32**).

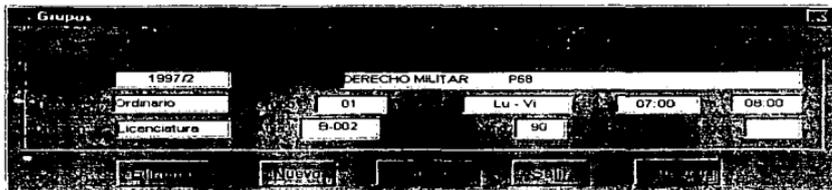


Ilustración B 32

5. Seleccione una a una con el Mouse la casillas a modificar, realice las modificaciones necesarias. Si desea eliminar la información seleccione el botón "eliminar".
6. Después de haber realizado una de las acciones descritas en el punto anterior, seleccione el botón "Guardar" para actualizar la base de datos.
7. Seleccione el botón "Salir" si desea terminar la captura de Grupos.

## ASIGNACION DE CLASES A PROFESORES.

Al seleccionar la opción de Clase el sistema despliega la pantalla de captura de la **Ilustración B 33**, la cual permitirá llevar a cabo la actualización de la información correspondiente a una clase (que profesor la imparte, horario, nombre de la asignatura, etc.) Para llevar a cabo la actualización de la base siga los siguientes pasos:

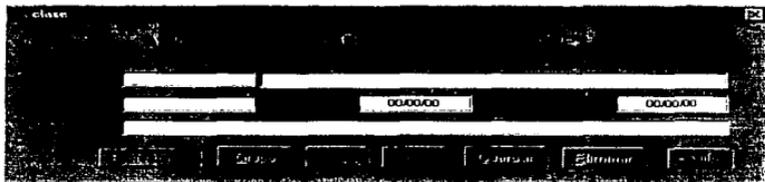


Ilustración B 33

### Altas de Clase.

1. Seleccione la casilla correspondiente a "Profesor (RFC)".
2. Seleccione el botón "Profesores". Al realizar esta acción el sistema desplegará la pantalla que se muestra en la **Ilustración B 34**.

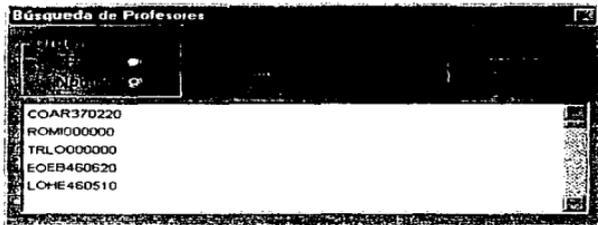


Ilustración B 34

3. Seleccione la forma en que el sistema debe ordenar la información a buscar, R.F.C. o Nombre. Al seleccionar nombre, el sistema ordenará la información por nombre del profesor como se observa en la **Ilustración B 34**, en caso de seleccionar R.F.C., el sistema desplegará la información ordenada por el R.F.C. del profesor como se muestra en la **Ilustración B 35**.
4. Seleccione para ambos casos de la lista la información que desea, si no alcanza a visualizar la información, puede deslizarse en la lista por medio de la barra de desplazamiento que se encuentra a la derecha de la celda.

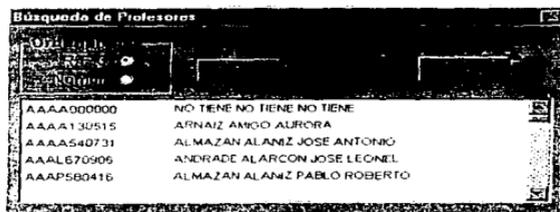


Ilustración B 35

5. Seleccione el botón "Aceptar". Al realizar esta operación el sistema mostrará la información que la base de datos contiene del profesor (nombre del profesor y R.F.C) como se muestra en la **Ilustración B 36**.

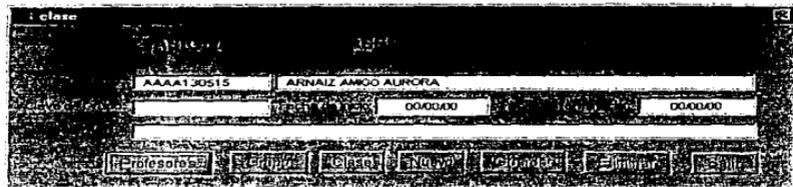


Ilustración B 36

6. Seleccione la casilla correspondiente a la "Clave de Grupo". Capture la información correspondiente a esta casilla en caso de no conocer la clave de grupo el sistema le proporciona el botón de "Grupo" para ello.
7. Seleccione el botón "Grupo", el sistema desplegará una pantalla que le permitirá buscar la clave del grupo correspondiente **Ilustración B 37**.

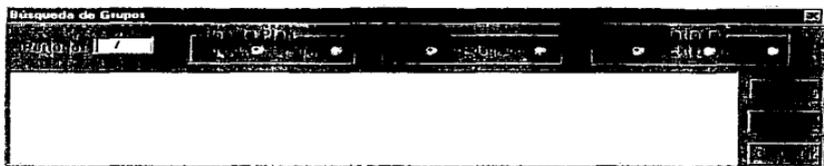


Ilustración B 37

8. Seleccione la casilla correspondiente a "Periodo". Capture la información correspondiente al periodo a buscar con el siguiente formato: 1997/2
9. Seleccione con el Mouse "Tipo de Plan" ( Nuevo o Viejo), "División" (Licenciatura, Posgrado y SUA) y "Tipo de periodo (Ordinario, Extraordinario y Acuerdo de pasantes).
10. Seleccione "Localizar". Al realizar esta acción el sistema mostrará la información en una lista desplegable, como se muestra en la **Ilustración B 38**.

ID	Tipo	División	Fecha	Clave	Plan	Periodo	Día	Horario
19972	ORD	LC	01	1604	AMPARO I	VEJO	09.00	10.00
19972	ORD	LC	02	1604	AMPARO I	VEJO	08.00	09.00
19972	ORD	LC	03	1604	AMPARO I	VEJO	07.00	08.00
19972	ORD	LC	04	1604	AMPARO I	VEJO	10.00	11.00
19972	ORD	LC	05	1604	AMPARO I	VEJO	08.00	09.00
19972	ORD	LC	06	1604	AMPARO I	VEJO	08.30	10.00
19972	ORD	LC	07	1604	AMPARO I	VEJO	08.30	10.00

Ilustración B 38

11. Seleccione con el Mouse como se indica en la Ilustración la información deseada. En caso de no alcanzar a visualizar la información, se puede desplazar en la lista por medio de la barra de desplazamiento que se observa al lado derecho de la lista.
12. Seleccione el botón "Aceptar". Al realizar esta acción el sistema mostrará en la pantalla principal de Clase la información que corresponde a esta clase como se muestra en la **Ilustración B 39**.

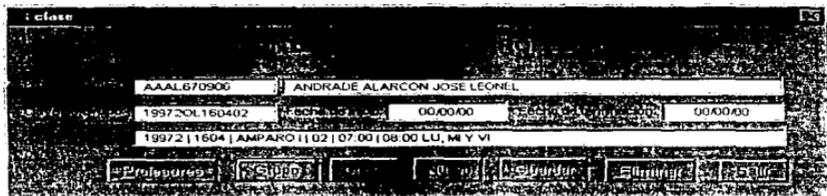


Ilustración B 39

13. Seleccione la casilla correspondiente a "Fecha de Inicio" y capture la información correspondiente con el siguiente formato: día/mes/año, haga lo mismo para la casilla "Fecha de Terminación".
14. Seleccione el botón "Guardar" para actualizar la base de datos.
15. Seleccione el botón "Nuevo" para capturar la información de una nueva clase.
16. Seleccione el botón "Eliminar" en caso de querer dar de baja la información visualizada.
17. Seleccione el botón "Salir" para finalizar.

#### **Búsqueda de Clase.**

Cuando de alguna manera se tiene la información capturada en la base de datos y se requiere realizar bajas o cambios en la misma, el sistema permite realizar una búsqueda de clase de la siguiente forma:

1. Seleccione el botón "Clase". El sistema mostrará la pantalla de la **Ilustración B 40**.

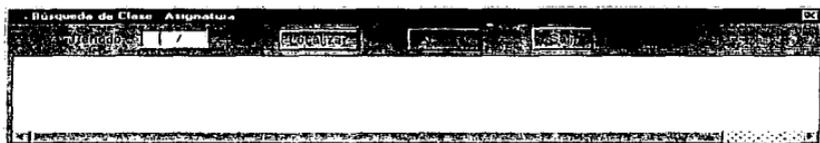


Ilustración B 40

2. Seleccione la casilla correspondiente a "Periodo". Capture la información correspondiente con el siguiente formato: día/mes/año.
3. Seleccione el botón "Localizar". Al realizar esta acción el sistema mostrará una lista con la información que la base de datos contiene en ese momento (**Ilustración B 40**).

ID	NOMBRE	FECHA	HORA	ASIGNATURA
04	AGUIAYO TERAN JESUS ALEJANDRO	10/02/97	06:06/97	16:00 17:00 L.M.V SISTEMA AFRESCOS C
02	ALMANZA PEDRAZA ENRIQUE	10/02/97	06:06/97	06:30 10:00 Mar-Jue DERECHO FAMILIAR
03	ALMAZAN ALANZ JOSE ANTONIO	10/02/97	06:06/97	17:00 18:00 L.VI DERECHO MERCANTIL
04	ALMAZAN ALANZ JOSE ANTONIO	10/02/97	06:06/97	18:00 19:00 L.VI DERECHO MERCANTIL

Ilustración B 41

4. Seleccione con el Mouse de la lista la información que le interesa. En caso de no visualizar la información, puede deslizarse en la lista con la barra de desplazamiento que se encuentra a la derecha y en la parte inferior de la lista.
5. Seleccione el botón "Aceptar". El sistema retornará a la pantalla principal y le mostrará la información que se tiene en ese momento en la base de datos (**Ilustración B 42**).

AESAB61228	ACEVEDO SERRANO ARTURO
1997.ZOL160615	10/02/97
(1606) DERECHO FAMILIAR GPO 15 LU, MI Y VI DE: 19.00 A: 20.00	

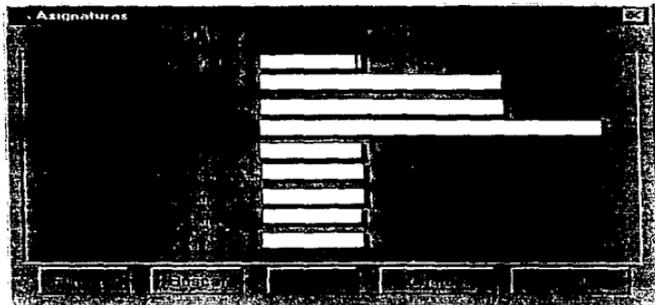
Eliminar    Guardar    Salir

Ilustración B 42

6. Seleccione el botón "Eliminar" si desea dar de baja la información o realice los cambios necesarios en las casillas que desee.
7. Seleccione el botón "Guardar" para actualizar la base de datos.
8. Seleccione el botón "Salir" para terminar la asignación de clases.

**ACTUALIZACION DE ASIGNATURAS.**

Al seleccionar la opción de asignaturas el sistema muestra la pantalla de captura de la **Ilustración B 43**, por medio de la cual se puede llevar a cabo la actualización de la base de datos.



**Ilustración B 43**

**Alta de Asignaturas.**

1. Seleccione con el Mouse la casilla correspondiente a clave de la asignatura y capture la clave correspondiente a la materia que desea dar de alta. Si la clave que capturó ya existe en la base de datos el sistema desplegará la información que contiene la base de datos correspondiente a esa clave.
2. Seleccione con el Mouse la casilla correspondiente a "Tipo de Plan". Al realizar esta acción el sistema mostrará una lista desplegable como se muestra en la **Ilustración B 44**.



Ilustración B 44

3. Seleccione con el Mouse de la lista la opción que desea capturar, de igual forma realice la misma operación para la casilla correspondiente a "División". Al realizar estas acciones el sistema captura la información.
4. Seleccione con el Mouse una a una y capture la información de cada una de las casillas restantes.
5. Al terminar de capturar la información seleccione el botón "Guardar", para actualizar la base de datos.
6. Seleccione el botón "Nuevo" en caso de querer capturar más de una asignatura.
7. Seleccione el botón "Salir" si desea terminar.

#### **Búsqueda, Cambio y Baja de Asignaturas.**

En caso de querer realizar una baja o cambio de información con respecto a las asignaturas el sistema cuenta con un botón "Búsqueda", que permite realizar una búsqueda de asignatura para realizar después con ella los cambios necesarios.

1. Seleccione el botón "Buscar". Al realizar esta acción el sistema desplegará la pantalla de la **Ilustración B 45**.

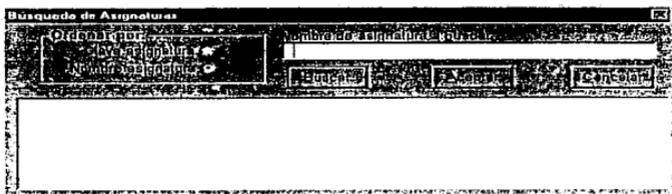


Ilustración B 45

2. Seleccione con el Mouse una de las dos opciones que tiene para ordenar la información. Al realizar esta selección el sistema mostrará la información a buscar ordenada según la elección que se hizo, en este caso por "Nombre de la asignatura" como se muestra en la **Ilustración B 46**.
3. Seleccione la casilla de "Nombre de Asignatura a buscar" y capture el nombre de la asignatura a buscar.
4. Seleccione el botón buscar. Al realizar esta acción el sistema muestra una lista que contiene la información correspondiente a la asignatura como se muestra en la **Ilustración B 46**.

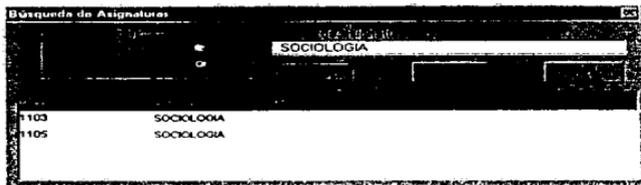


Ilustración B 46

5. Seleccione con el Mouse la opción deseada de la lista.
6. Seleccione el botón "Aceptar". Al realizar esta acción el sistema retornará a la pantalla principal de asignaturas y mostrará la información que la base de datos contiene de la asignatura seleccionada (Ilustración B 47).
7. Seleccione el botón "Eliminar", en caso de querer suprimir la información.
8. Seleccione el botón "Guardar" para actualizar la base de datos.
9. Seleccione el botón "Salir" para finalizar.

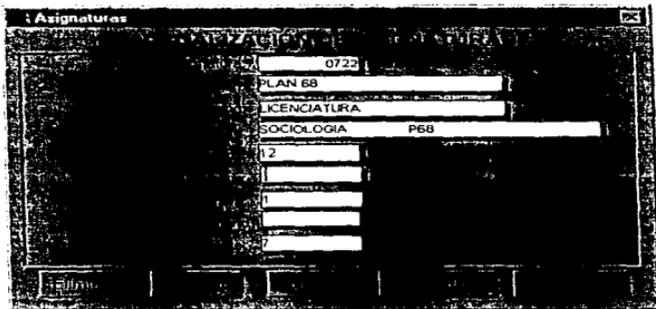


Ilustración B 47

**ACTUALIZACION DE ALUMNOS.**

Al seleccionar la opción de Alumnos el sistema despliega la pantalla de captura que se muestra en la **Ilustración B 48** con la cual se podrá actualizar en la base de datos la información correspondiente a los alumnos.

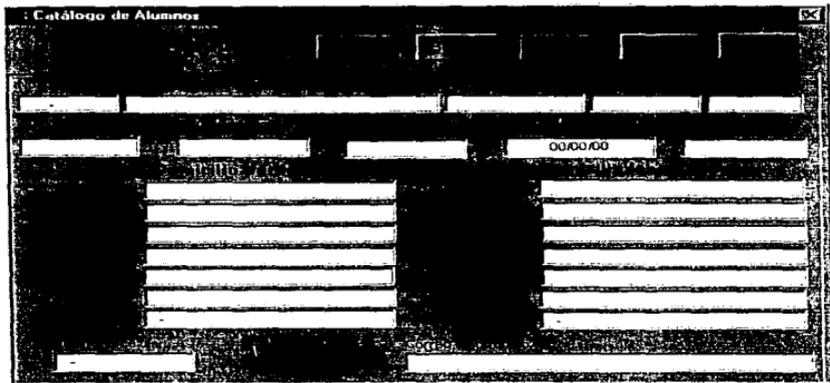


Ilustración B 48

**Alta de Alumnos.**

1. Seleccione la casilla correspondiente al número de cuenta. Capture la información correspondiente. En caso de que la base de datos contenga ya la información correspondiente al número de cuenta que tecléó, el sistema desplegará la información que la base de datos contiene.
2. Al seleccionar la casilla correspondiente a Nacionalidad, el sistema mostrará una lista desplegable, de la cual deberá de seleccionar con el Mouse la opción deseada. Lo mismo



**Catálogo de Alumnos**

BRASILEÑA

FEMENINO

000000

MASCULINO

000000

Dom(cil)ib(ati)on

Dom(cil)ib(ati)on

Ilustración B 50

**Catálogo de Alumnos**

BRASILEÑA

FEMENINO

000000

Dom(cil)ib(ati)on

Dom(cil)ib(ati)on

BAJA CALIFORNIA NORTE

AGUASCALIENTES

BAJA CALIFORNIA NORTE

BAJA CALIFORNIA SUR

CAMPECHE

COAHUILA

COLIMA

CHIAPAS

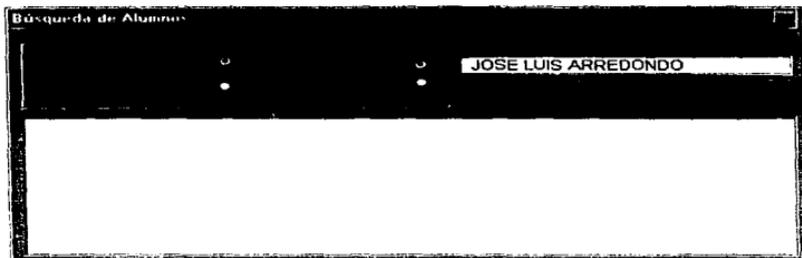
Ilustración B 51

3. Seleccione con el Mouse una a una y capture la información de cada una de las casillas restantes.
4. Al terminar de capturar la información, seleccione el botón "Guardar", para actualizar la base de datos.
5. Seleccione el botón "Nuevo", en caso de querer capturar más de una asignatura.
6. Seleccione el botón "Salir" para terminar.

#### **Búsqueda, Cambio y Baja de Alumnos.**

En caso de querer realizar una baja o cambio de información con respecto a los alumnos, el sistema cuenta con un botón "Búsqueda", con el cual encontramos al alumno que se le desean realizar los cambios necesarios.

1. Seleccione el botón "Buscar". Al realizar esta acción el sistema desplegará la pantalla que se muestra en la **Ilustración B 52**.



**Ilustración B 52**

2. Seleccione con el Mouse una de las dos opciones que tiene para ordenar la información.  
Al realizar esta selección el sistema mostrará la información a buscar ordenada según la

elección que se hizo, en este caso, por Número de cuenta del alumno como se muestra en la **Ilustración B 52**.

3. Seleccione con el Mouse una de los dos opciones que aparecen en "Buscar por".
4. Seleccione la casilla de "Buscar a" y capture el "Nombre" o el "No. de cuenta", según sea el caso que usted haya seleccionado en "Buscar por". En este caso se escribió el nombre del alumno, porque se seleccionó en "Buscar por" nombre. Si selecciona "No. de cuenta", deberá teclear el número de cuenta del alumno.
5. Seleccione el botón "Buscar". Al realizar esta acción el sistema muestra una lista que contiene la información correspondiente al alumno, en caso de no encontrarse el sistema mostrará una pantalla en la que se informa que no se encontró en la base de datos ninguna información correspondiente a ese alumno, como se muestra en la **Ilustración B 53**.



**Ilustración B 53**

6. Seleccione de la lista la opción deseada. Al realizar esta acción el sistema mostrará en la pantalla principal de alumnos la información que la base de datos contiene de ese alumno.
7. Seleccione el botón "Eliminar" en caso de querer dar de baja la información del alumno o realice los cambios necesarios si así lo desea.
8. Seleccione el botón "Guardar" para actualizar la base de datos.
9. Seleccione el botón "Salir" si desea finalizar.

## ACTUALIZACION DE AULAS.

Al seleccionar la opción correspondiente a aulas, el sistema desplegará la pantalla de captura que se muestra en la **Ilustración B 54**. Para llevar a cabo la actualización de esta información en la base de datos realice:

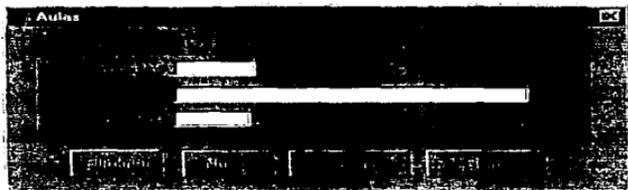


Ilustración B 54

### Alta, Baja y Cambio de Aulas.

1. Seleccione la casilla correspondiente a "Clave aula". Capture la información correspondiente. En caso de existir la clave del aula en la base de datos el sistema mostrará en la pantalla principal de aulas la información que se tiene en ese momento.
2. Seleccione el botón "Eliminar" si desea dar de baja la información del aula.
3. Seleccione el botón "Nuevo" en caso de capturar varias aulas.
4. Después de capturar la información de una aula, seleccione el botón "Guardar" para actualizar los cambios en la base de datos.
5. Seleccione el botón "Salir" para finalizar.

**ACTUALIZACION DE SEMINARIOS.**

Al seleccionar la opción de seminarios el sistema desplegará la pantalla que se muestra en la **Ilustración B 55**, la cual sirve para llevar a cabo la actualización de la información de los mismos en la base de datos.

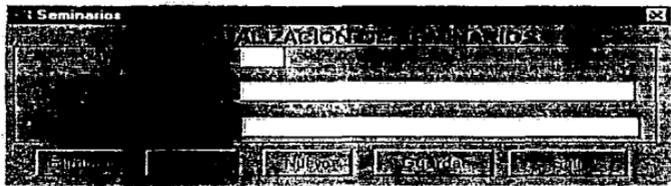


Ilustración B 55

1. Seleccione la casilla correspondiente a clave del seminario, capture la información correspondiente a ésta, en caso de existir la clave que se capturo el sistema mostrará en la pantalla la información correspondiente a esa clave como se muestra en la **Ilustración B 56**.



Ilustración B 56

2. Seleccione el botón "Eliminar" en caso de querer dar de baja la información.
3. Seleccione el botón "Guardar" para actualizar la base de datos.

4. Seleccione la casilla de "Director del Seminario" y capture la información correspondiente al mismo.
5. Seleccione el botón "Guardar" para actualizar la base de datos.
6. Seleccione el botón "Nuevo" si desea capturar otro seminario.
7. Capture la información correspondiente a cada casilla.
8. Seleccione el botón "Guardar" para actualizar la base de datos.
9. Seleccione el botón "Salir" para finalizar.

#### Búsqueda de Seminarios.

La pantalla principal de seminarios consta de un botón "Búsqueda" el cual le permitirá realizar una búsqueda como se indica a continuación:

1. Seleccione el botón "Buscar", al realizar esta acción el sistema desplegará la pantalla que se muestra en la **Ilustración B 57**.

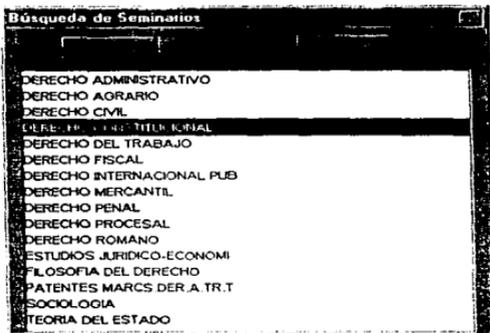


Ilustración B 57

2. Seleccione con el Mouse la opción deseada. Al realizar esta acción el sistema mostrará en la pantalla principal la información correspondiente a la opción seleccionada (**Ilustración B 58**).

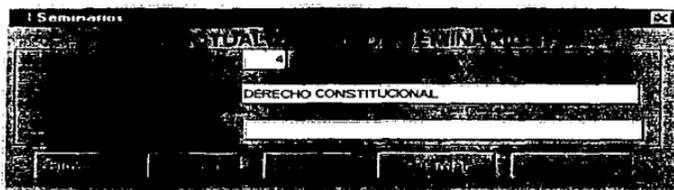


Ilustración B 58

3. Seleccione el botón "Eliminar" en caso de querer dar de baja la información.
4. Seleccione el botón "Guardar" para actualizar la base de datos.
5. Seleccione la casilla de "Director del Seminario" y capture la información correspondiente al mismo.
6. Seleccione el botón "Guardar" para actualizar la base de datos.
7. Seleccione el botón "Nuevo" si desea capturar otro seminario.
8. Capture la información correspondiente a cada casilla.
9. Seleccione el botón "Guardar" para actualizar la base de datos.
10. Seleccione el botón " Salir" para finalizar.

## ACTUALIZACION DE EXAMENES PROFESIONALES.

Al seleccionar la opción de exámenes profesionales el sistema presenta la pantalla de captura de la Ilustración B 59. Para llevar a cabo la actualización de la base de datos realice lo siguiente:

Ilustración B 59

### Alta de Exámenes Profesionales.

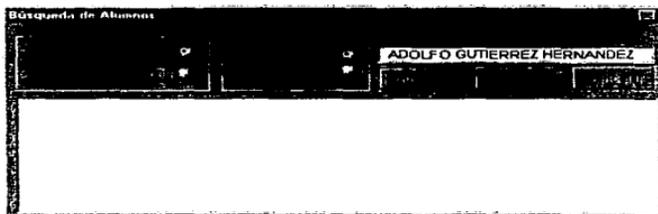
1. Seleccione la casilla a "No. de Registro". Capture la información correspondiente a esta casilla. En caso de existir información en la base de datos correspondiente a este no. de registro el sistema desplegará en la pantalla principal de exámenes profesionales la información contenida en la base de datos.
2. Para cambiar a la siguiente casilla oprima una vez la tecla <Tab> en el teclado. Esta operación la deberá de repetir para poder capturar las casillas siguientes.

3. Al seleccionar la casilla correspondiente a "Profesor", el sistema mostrará una lista desplegable de la cual deberá seleccionar con el Mouse el nombre del profesor que desea.
4. Seleccione una a una las siguientes casillas y capture la información correspondiente a las mismas.
5. Seleccione el botón "Sortea" para asignar sinodales aleatoriamente.
6. Seleccione el botón "Registrar" para actualizar en la base de datos la alta realizada.
7. Seleccione el botón "Nuevo" si desea dar de alta otro examen.
8. Seleccione el botón "Cancelar" si desea salir de Exámenes Profesionales.

#### **Búsqueda, Baja y Cambio de Exámenes Profesionales.**

Si desea dar de baja o realizar cambios en la información que contiene la base de datos en relación con exámenes profesionales y no conoce el "No. de registro del alumno", el sistema le permite realizar una búsqueda de la información como se explica a continuación.

1. Seleccione el botón "Buscar Alumnos" de la pantalla principal de exámenes profesionales. Al realizar esta acción el sistema desplegará la pantalla de. **Ilustración B 60**



**Ilustración B 60**

2. Seleccione con el Mouse una de las dos opciones que tiene para ordenar la información. Al realizar esta selección el sistema mostrará la información a buscar ordenada según la elección que se hizo, en este caso por Número de cuenta del alumno como se muestra en la **Ilustración B 60**.
3. Seleccione con el Mouse una de las dos opciones que aparecen en "Buscar por".
4. Seleccione la casilla de "Buscar a" y capture el nombre del alumno o el No. de cuenta, según sea el caso que usted haya seleccionado en "Buscar por". En este caso se escribió el nombre del alumno, porque se seleccionó en "buscar por" nombre. Si selecciona No. de cuenta, deberá teclear el número de cuenta del alumno.
5. Seleccione el botón "Buscar". Al realizar esta acción el sistema muestra una lista que contiene la información correspondiente al alumno, en caso de no encontrarse el sistema mostrará una pantalla en la que se informa que no se encontró en la base de datos ninguna información correspondiente a ese alumno, como se muestra en la **Ilustración B 61**.



**Ilustración B 61**

6. Seleccione de la lista la opción deseada. Al realizar esta acción el sistema mostrará en la pantalla principal de alumnos la información que la base de datos contiene de ese alumno.
7. Realice los cambios necesarios a la información que desee.
8. Seleccione el botón "Registrar" para actualizar la base de datos
9. Seleccione el botón "Cancelar" si desea salir del sistema.

Al seleccionar la opción de Comisiones el sistema permite llevar a cabo la actualización de la información en la base de datos de las comisiones a las que se asigna a los profesores, presentando la pantalla de captura de la **Ilustración B 62**.

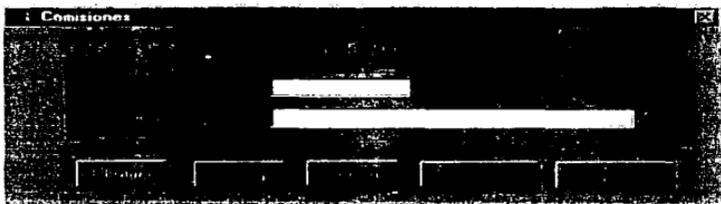


Ilustración B 62

#### Alta, Baja y Cambio de Comisiones.

1. Seleccione la casilla correspondiente a "Clave de Comisión". Capture la información correspondiente. En caso de existir información en la base de datos con respecto a la clave tecleada, el sistema presentará la información como se muestra en la **Ilustración B 63**.

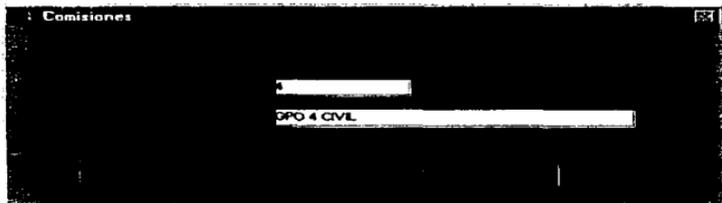


Ilustración B 63

2. Seleccione el botón "Eliminar" si desea dar de baja la información o haga los cambios necesarios en las casillas correspondientes.

3. Seleccione el Botón "Guardar" para llevar a cabo la actualización de la base de datos.
4. En caso de no existir información en la base de datos, capture la información necesaria en cada casilla.
5. Seleccione el botón "Guardar" Para efectuar la actualización de la base de datos.
6. Seleccione el botón "Nuevo" para capturar la información de una alta de comisión.
7. Seleccione el botón "Salir" si desea terminar.

La pantalla principal de comisiones cuenta con una opción para realizar una búsqueda de alguna comisión, con la cual se desea realizar algún cambio o baja, para poder hacer uso de esta opción realice lo siguiente:

1. Seleccione con el Mouse el botón "Buscar". Al realizar esta acción el sistema despliega la pantalla de la **Ilustración B 64**.

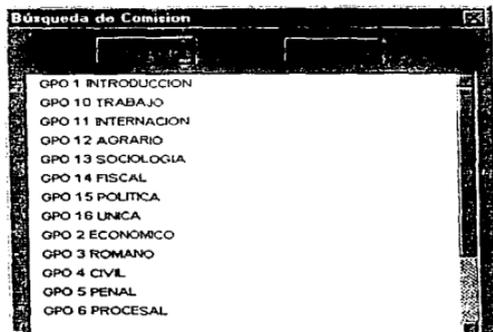
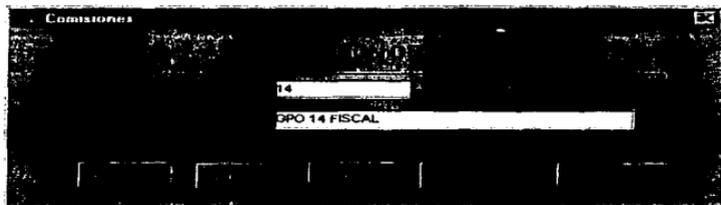


Ilustración B 64

2. Seleccione con el Mouse el nombre de la comisión deseada de la lista.

3. Seleccione el botón "Aceptar". Al realizar esta acción el sistema muestra la información contenida en la base de datos como se muestra en la pantalla de la **Ilustración B 65**.



**Ilustración B 65**

4. Seleccione el botón "Eliminar" si desea dar de baja la información o haga los cambios necesarios en las casillas correspondientes
5. Seleccione el botón "Guardar" para llevar a cabo la actualización de la base de datos.
6. Seleccione el botón "Salir" si desea terminar.

## ACTUALIZACION DE COMISIONADOS.

Al seleccionar la opción de comisionados el sistema muestra la pantalla de la **Ilustración B 66** que permite llevar a cabo la actualización en la base de datos de los profesores que fueron comisionados y el nombre de la comisión a la cual fueron asignados.

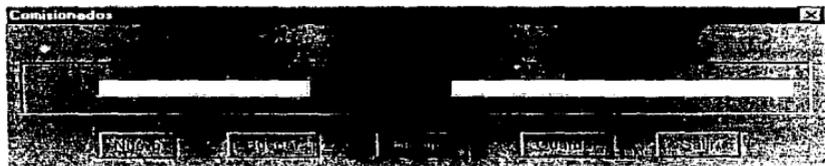


Ilustración B 66

### Alta de Profesores Comisionados.

1. Seleccione con el Mouse la casilla correspondiente a "Comisión". Al realizar esta acción el sistema presentará una lista desplegable como se muestra en la Ilustración B 67.



Ilustración B 67

2. Seleccione con el Mouse la flecha que aparece en la casilla correspondiente a "Comisión". Al realizar esta acción el sistema desplegará una lista de la cual deberá seleccionar con el Mouse el nombre de la comisión que desea. No podrá capturar la información correspondiente a la siguiente casilla sino captura la información de esta, ya que el sistema desplegará un mensaje de que esta información es requerida.

3. Seleccione la casilla correspondiente a "Nombre del Profesor". Al realizar esta acción el sistema muestra una lista desplegable de la cual deberá seleccionar con el Mouse el nombre del profesor deseado.



Ilustración B 68

4. Seleccione el botón "Guardar" para efectuar la actualización de la base de datos.
5. Seleccione el botón "Salir" para terminar.

La pantalla de comisionados contiene un botón "Buscar" que al igual que los demás casos permitirá realizar la búsqueda de información como se muestra en la **Ilustración B 69**.

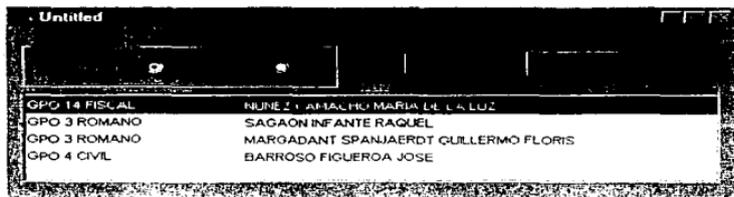


Ilustración B 69

1. Seleccione en la parte de "Ordenar por" la forma en que desea ordenar la lista que aparece en la pantalla, si selecciona "Profesor" la lista aparecerá ordenada por nombre del profesor, por el contrario si selecciona "Comision" la información aparecerá

ordenada por nombre de la comisión. En este caso está ordenada por nombre del profesor.

2. Seleccione con el Mouse la opción deseada.
3. Seleccione el botón "Aceptar". Al realizar esta acción el sistema mostrará la información en la pantalla principal de comisionados. En la cual podrá realizar los cambios deseados.

**ACTUALIZACION DE CANDIDATOS A OPOSICION.**

Al seleccionar la opción de candidatos a oposición el sistema despliega la pantalla de captura de la **Ilustración B 70** que al igual que las anteriores permite la actualización de la base de datos en relación con la información que se maneja.

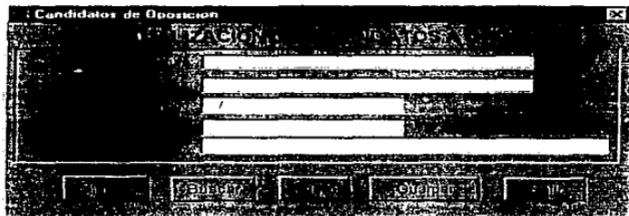


Ilustración B 70

**Alta, Baja y Cambio de Candidatos a Oposición.**

1. Seleccione la casilla correspondiente a "Asignatura". Al efectuar esta operación el sistema muestra una lista desplegable de la cual deberá seleccionar con el Mouse el nombre de la asignatura deseada, como se muestra en la **Ilustración B 71**. Realice la misma operación para el caso de "Nombre de Profesor".

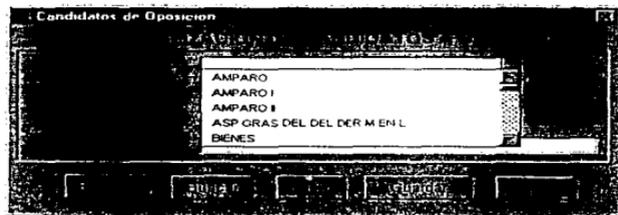


Ilustración B 71

## **Apéndice B**

---

2. Seleccione y capture la información correspondiente a las demás casillas.
3. Seleccione el botón "Guardar" para actualizar la base de datos.
4. Seleccione el botón "Nuevo" para capturar la información de un nuevo profesor.
5. Seleccione el botón "Buscar" para realizar una búsqueda de información. Posteriormente, seleccione el botón "Eliminar" si desea dar de baja la información que se buscó o realice los cambios necesarios en la casilla deseada.
6. Seleccione el botón "Guardar" para efectuar la actualización de la base de datos.
7. Seleccione el botón "Salir" para finalizar.

**ACTUALIZACION DE EXAMENES DE OPOSICION.**

Al seleccionar la opción de Exámenes de Oposición el sistema despliega la pantalla de captura de la **Ilustración B 72**, para llevar a cabo la actualización de la base de datos.

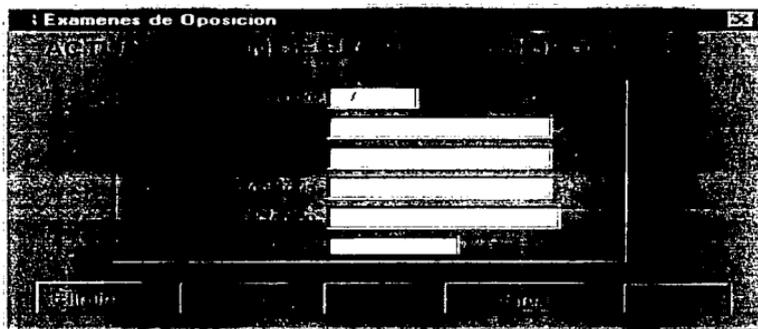


Ilustración B 72

**Alta de Exámenes de Oposición.**

1. Seleccione la casilla correspondiente a "Período". Capture la información correspondiente a éste, el formato de captura es el siguiente: 1997/2.
2. Seleccione la casilla correspondiente a "Asignatura", en este caso el sistema presentará una lista desplegable con el nombre de la asignatura, de la cual deberá seleccionar la deseada. De igual forma deberá actuar para capturar la casilla correspondiente a "Comisión", como se muestra en las **Ilustraciones B 73 y B 74**.

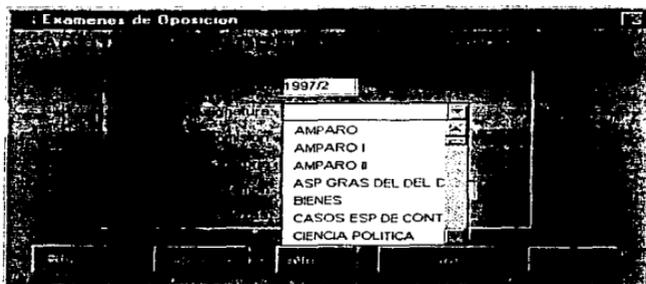


Ilustración B 73

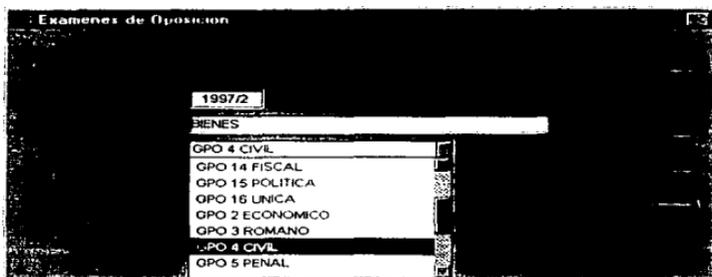


Ilustración B 74

3. Señale una a una las demás casillas y capture la información correspondiente a cada una de ellas. El formato que se debe tomar en cuenta para la captura de las fechas es: día/mes/año.
4. Seleccione el botón "Guardar" para actualizar la base de datos.
5. Seleccione el botón "Nuevo" para capturar la información de un nuevo examen de oposición.

**6. Seleccione el botón "Salir" para finalizar la sesión de Exámenes de Oposición.**

**Baja y Cambio de Exámenes de Oposición.**

Si desea realizar cambios, bajas o consultas de información en la base de datos el sistema proporciona en la pantalla principal de Exámenes de Oposición un botón que permite realizar una búsqueda de información para poder efectuar estos movimientos. Para ello siga los siguientes pasos:

1. Seleccione el botón "Buscar" de la pantalla principal de Exámenes de Oposición. Al realizar esta acción el sistema muestra la pantalla de búsqueda de la cual deberá seleccionar con el Mouse el nombre de la comisión.
2. Seleccione el botón "Aceptar". Al efectuar esta operación el sistema mostrará en la pantalla principal de Exámenes de Oposición la información correspondiente.
3. Seleccione el botón "Eliminar" si desea dar de baja la información o realice los cambios necesarios.
4. Seleccione el botón "Guardar" para actualizar la base de datos.
5. Seleccione el botón "Salir" para terminar la sesión de Exámenes Profesionales.

**APENDICE C**  
**Sybase SQL Server 11**

## **SYBASE SQL SERVER 11.**

La información confiable se ha vuelto el alma para sobrevivir en el mundo cambiante de hoy. Cada día, más compañías corren más de sus operaciones con un Sistema Manejador de Bases de Datos Relacional (SMBDR). En la decisión de dar soporte de orden de entrada a tiempo real, la demanda por aplicaciones de procesamiento de transacción en línea (PTEL) ha crecido a un paso récord. Cuando se opera un negocio con software SMBDR, se necesita alta ejecución, alta eficiencia y confiabilidad. Este es el porque miles de negocios cambian a un servidor Sybase SQL para sus aplicaciones PTEL. Sybase fue el primer SMBDR cliente/servidor en cubrir las demandas de fiabilidad y escalabilidad de las más desafiantes aplicaciones de negocios. Y Sybase es el primer software diseñado en Norteamérica en recibir la certificación de calidad ISO9001, asegurándose que cada componente de un servidor SQL haya sido probado cuidadosamente antes de ser transportado.

El servidor Sybase SQL 11 construido durante años de experiencia, agregando nuevas características avanzadas para ayudar a las organizaciones a conseguir lo mejor de sus recursos humanos y de información. El servidor SQL 11 permite a los negocios correr más eficientemente, mejora sus servicios al cliente, y controla sus operaciones en tiempo real - entregando una substancial ventaja competitiva a través de un amplio campo de aplicaciones de base de datos-.

### **REQUERIMIENTOS FUNDAMENTALES QUE PROVEE EL SERVIDOR SQL A LOS SISTEMAS LLAMADOS DE "MISION CRITICA":**

**Alta ejecución escalable y alta eficiencia con tecnología comprobada.**

- Corre en una variedad de plataformas, desde PC's hasta superservidores de multi CPU's, así que se puede seleccionar el hardware apropiado para cada trabajo o cambiar el hardware si las necesidades cambian.

- Proporciona excelente ejecución a cada computadora como resultado de la cooperación cercana con los proveedores de hardware y optimización de las características de sus productos.
- Logra velocidades de transacción extremadamente grandes y soporta a muchos usuarios debido a su eficiencia y simetría completa.

**Confiabilidad e integridad: componentes clave para una solución completa.**

- Sybase almacena procedimientos y triggers que mantienen la integridad.
- Si la integridad de los datos es violada, el trigger de Sybase deshace la transacción, preservando la integridad de los datos.
- Almacena complejos procedimientos encapsulados lógicos de negocios dentro de unidades preempaquetadas de código que pueden reutilizarse para múltiples aplicaciones para el correcto manejo de los datos.
- El diseño satisface los niveles de eficiencia que define la National Computer Security Council.

**La carga de trabajo se adapta al máximo uso de los recursos.**

- Se puede configurar cualquier número de memorias cache por el servidor para que se pueda lograr una optimización completa de los recursos de base de datos.
- PTEL y aplicaciones para decisión de soporte pueden correr en un mismo servidor, maximizando la inversión de los sistemas de información.
- El manejador de memoria lógica, compatible con el monitoreo del servidor SQL, permite mejorar la optimización de los recursos de memoria y completar infinidad de aplicaciones.

**Disponibilidad de datos que estimulan la productividad.**

- **Reflejo de disco y alta velocidad de almacenaje y recuperación que minimizan el impacto de las fallas de hardware en la corrida de aplicaciones.**
- **Soporte completo del servidor SQL de almacenaje y recuperación en línea, haciendo los datos mucho más accesibles a los usuarios.**
- **El respaldo del servidor se toma de la tarea de almacenaje y recuperación del servidor SQL, permitiendo al servidor poner en marcha aplicaciones casi sin afectarse.**

**Interoperabilidad y estándares de conformidad.**

- **Conformidad con el estándar ANSI/ISO SQL-89 y entra al nivel ANSI/ISO SQL-92.**
- **Soporta aplicaciones escritas para ODBC y estándares X/Open XA.**
- **Soporta varios protocolos de red, habilita virtualmente a cualquier cliente a conectar su máquina a un servidor SQL corriendo en cualquier plataforma.**

**Facilidad de manejo con control y soporte.**

- **Sofisticada arquitectura multihilo, lo que significa que cada máquina tiene solamente un proceso en servidor para manejar.**
- **Para ambientes de multiprocesamiento simétrico, la arquitectura del servidor virtual de Sybase permite el control de la cantidad de recursos de CPU designados para el SMBDR.**
- **Una completa línea de productos de manejo de sistema es facilitado para asistir al cliente en el manejo de almacenaje, usuarios, seguridad y ejecución -desde una a muchas bases de datos en múltiples redes en todas partes de una empresa.**

## **SERVIDOR DE SYBASE SQL 11: LA FUNDACION DEL MANEJO DE INFORMACION DE LAS EMPRESAS.**

La información es convertida rápidamente en una organización de más recursos. Este es el porque el servidor Sybase SQL descubrió que la información en cualquier lugar en la actualidad puede literalmente hacer o deshacer un negocio. Servicios financieros, numerosos segmentos de industrias de telecomunicaciones para aerolíneas, y compañías de manufactura, escogen un servidor Sybase SQL por ejecución, confiabilidad, conectividad universal, y manejo efectivo de sistemas distribuidos.

**Ejecución, escalabilidad, confiabilidad, y cómputo distribuido, son lo que más satisface en el mundo real.**

Aunque el servidor de SMBDR Sybase SQL tiene liderazgo en los resultados de las pruebas de comparación en las industrias, las pruebas de comparación no dan la historia completa. Sybase da satisfacción en ambientes de aplicación del mundo real, donde la ejecución, escalabilidad, confiabilidad, y cómputo distribuido son igualmente importantes. Sybase tiene consistencia que se prueba por sí mismo en miles de sitios donde existen sus clientes, bajo las condiciones actuales de trabajo, donde la ejecución es medida a diferentes escalas en líneas de resultados.

Transact SQL, un poderoso conjunto de estándares ANSI/ISO SQL, permite a una aplicación ejecutar procedimientos almacenados o SQL dinámico para controlar transacciones. Además, Transact SQL da a los diseñadores un programa de organización amplia para sus negocios, transacciones, y parametriza queries dentro de procedimientos de compilación almacenados para incrementar la eficiencia de programación y la ejecución de bases de datos.

El servidor SQL también tiene madurez, implementación funcional de procedimientos almacenados, probada en ambientes de producción por más de siete años. Porque los procedimientos almacenados son precompilados, ellos son procesados más rápido que una secuencia de declaraciones SQL dinámicas. Estos procedimientos hacen la estructura de

datos transparente para las aplicaciones del cliente, decrementando los costos de mantenimiento de software; ellos también reducen el tráfico de red y pruebas de seguridad. Y los procedimientos almacenados pueden retornar múltiples líneas de datos, para una respuesta más rápida y una arquitectura más flexible.

#### **Características de sistema que dan resultados.**

- El servidor SQL puede crear procedimientos de bases de datos remotas que son llamados por otras fuentes de datos y servicios, haciendo posible el integrar un sistema complejo, ocultando los detalles de las aplicaciones a los programadores.
- Madurez basada en el costo de optimización que da el mejor plan de queries por costo de efectividad del uso de los recursos del sistema.
- Respaldo del servidor que da mucha velocidad al almacenaje y recuperación con mínimo impacto en la corrida de aplicaciones.

#### **Alta ejecución escalable.**

En el ambiente del mundo real, las aplicaciones originalmente diseñadas para un número limitado de usuarios en un solo lugar pueden rápidamente crecer para incluir a miles de usuarios en lugares geográficos dispersos. Para ayudar a manejar este crecimiento y dispersión, el servidor Sybase SQL escala de manera uniforme de un grupo de trabajo de pocos usuarios accediendo una base de datos de varios megabytes a miles de usuarios en corporaciones conectadas a sistemas integrales de muchos gigabytes. Porque Sybase maneja más usuarios en una configuración de máquina dada que otras compañías de productos SMBDR, una organización puede correr su más demandada aplicación al más bajo costo de hardware.

#### **Escalabilidad de arquitectura.**

- El servidor SQL requiere 48 KB para conectar a un usuario, reduciendo los requerimientos de hardware y dejando más memoria habilitada.

- El servidor SQL mejora la eficiencia del manejo de funciones de multiusuarios, tales como programar y seleccionar tareas con el servidor en proceso.
- El servidor SQL provee a las organizaciones de gran procesamiento con cientos de miles de usuarios sin tener que hacer uso de monitoreo de procesos de transacciones complejas.
- La arquitectura del servidor virtual de Sybase utiliza solo un proceso para cada CPU asignado al servidor, asegurando el óptimo uso de sistemas de multiprocesamiento simétrico (MPS) mientras automáticamente se balancea la carga a través de los CPU's.
- El servidor Sybase SQL ofrece una mejor respuesta en inversión de hardware, número de usuarios y costo de transacción.

**Carga adaptable a la arquitectura para ajustar aplicaciones.**

- El servidor Sybase SQL hace más eficiente el manejo de PTEL y la decisión de las aplicaciones de soporte.
- Múltiples memorias cache permiten entonar y optimizar múltiples aplicaciones en el mismo servidor al mismo tiempo.
- Tres niveles de tonos de configuración que permiten un alto grado de categoría de sistema y optimización de aplicaciones.

**SOPORTE DE RESPALDO PARA BASES DE DATOS MUY GRANDES: MANTENIENDO NIVELES DE EJECUCIÓN.**

En el ambiente del mundo real, archivar una gran base de datos puede impactar la ejecución global del sistema, tomando recursos lejanos de otras aplicaciones y alentando los tiempos de respuesta. El servidor de respaldo de Sybase, un componente del servidor SQL estándar, brinda una ejecución excelente para bases de datos tan grandes como pueden ser cientos de gigabytes. Con su propia arquitectura escalable, el servidor de respaldo puede ser

configurado de manera independiente para soportar velocidades de respaldo superiores a los 50 GB por hora -sin provocar impacto en la corrida de aplicaciones-.

#### **Características de bases de datos grandes.**

- El servidor de respaldo descarga los desechos del servidor mientras es realizado el respaldo, resultando una ejecución con un despreciable impacto en las aplicaciones que están corriendo.
- La partición de datos permite cargas paralelas y múltiples inserciones concurrentes.
- Particiones indexadas que ayudan a modificar y recuperar registros casi tan rápido en una gran base de datos como en una pequeña.
- Las bases de datos de más de 2 GB son direccionadas en memoria, habilitadas para una alta ejecución.
- Múltiples memorias cache permiten la partición y optimización de una memoria grande.

#### **ESTANDARES PARA UNA GRAN FLEXIBILIDAD.**

A través de una completa compatibilidad con estándares industriales e interoperabilidad con estándares de compatibilidad de componentes, Sybase protege la inversión tecnológica de la organización de la información y garantiza la gran flexibilidad al paso del tiempo. Los usuarios de Sybase pueden adquirir cualquier aplicación de seguridad para ampliar la funcionalidad de un sistema o construir y ampliar sus aplicaciones en casa. Los diseñadores en Sybase pueden escribir nuevamente el estándar de aplicación de interface de programación de su elección -tal como el ODBC de Microsoft- y entonces correr las aplicaciones de nuevo en el servidor SQL. Tal flexibilidad permite a las organizaciones hacer el mejor uso de sus recursos financieros y humanos.

**Completa compatibilidad ANSI/ISO.**

- Completo y eficiente soporte que permite escribir programas de cliente, triggers, y procedimientos almacenados que hacen eficiente el manejo de registros en un tiempo de procesamiento.
- La declaración de la integridad referencial reduce la necesidad de código de triggers para mejorar relaciones básicas entre tablas.
- La semántica de la transacción flexible provee la opción del uso del modelo de transacción en cadena ANSI/ISO.
- Cien por ciento compatible con ANSI/ISO SQL-89 y a un gran nivel con ANSI/ISO SQL-92.

**CONTROL DE LA BASE DE DATOS DISTRIBUIDA.**

Desde su primera versión, el servidor Sybase SQL SMBDR ha soportado programación de 2 fases de commit (2FC) para asegurar la integridad de los datos de aplicaciones distribuidas. Con programación 2FC los diseñadores manejan errores individuales para cada transacción, en vez de dejar que los controles del error sean manejados por el sistema interno. La programación 2FC permite un mayor detalle del nivel de control sobre la transacción. Cambiando programación 2FC con software de Sybase, los usuarios pueden coordinar las transacciones heterogéneas y el mismo nivel de las fuentes de datos no relacionales.

**Características de bases de datos relacionales.**

- Los procedimientos almacenados aíslan a los diseñadores de detalles de implementación, donde el Administrador de Bases de Datos (ABD) local tiene completa autonomía en la configuración de los objetos de la base de datos.
- El servidor SQL provee control de integridad, ejecutando transacciones constantes cuando los datos están distribuidos entre los diferentes servidores.

- El acceso distribuido permite a un servidor SQL central soportar cientos de aplicaciones en diferentes máquinas.
- El servidor SQL puede usar múltiples lenguajes internacionales en una sola base de datos; después el usuario selecciona un lenguaje al momento del acceso, la aplicación y los mensajes del sistema aparecen en el lenguaje seleccionado.

#### **ASEGURAR LA INTEGRIDAD DE LOS DATOS.**

Sybase es el pionero en el esfuerzo de mantener la integridad de los datos del servidor. Usando procedimientos almacenados de Sybase y triggers, se programa el diseño de aplicaciones y almacena todas las tareas de una organización de negocios y controla la integridad en el servidor, el cual fuerza la transacción lógica para todas las aplicaciones de clientes en la red. Otros triggers en venta pueden solo reportar errores, requiriendo de la aplicación para preservar la integridad de los datos. Los triggers de Sybase forman una pared inviolable alrededor de los datos, así que los diseñadores de aplicaciones no necesitan programas que revisen la integridad con cada aplicación. Sybase también permite las características de la declaración de integridad referencial estándar ANSI/ISO para ser combinados con mecanismos de integridad procedural. Y el servidor SQL otorga facilidades para la seguridad de la base de datos.

Los triggers son procedimientos almacenados programables que están unidos a una tabla y son automáticamente activados por intentos de inserción, borrado, o actualización de registros. Ellos ayudan a mantener la consistencia de los datos; las aplicaciones del cliente no pueden evitarlos. Un trigger en un valor primario puede actualizar todas las demás copias cuando los cambios están hechos. Agregando, los triggers pueden estar en cascada, ser recursivos, y pueden retomar la transacción que causaron al encenderse.

#### **Otras características de integridad.**

- La integridad referencial incluye mecanismos para hacer en cascada las acciones de borrado/actualización, bloqueo, y nulificación.

- Los diseñadores pueden usar reglas para especificar valores válidos para declarar un campo para ayudar a asegurar la integridad completa del sistema.
- Los tipos de datos restringen la clase de información almacenada en las columnas de las tablas de bases de datos para una integridad de datos básica.
- La definición de tipos de datos por el usuario agrega flexibilidad.
- Un módulo de revisión configurable puede grabar una variedad de diferentes acciones en la base de datos.

#### **MANEJO SIMPLIFICADO DEL SISTEMA.**

Como los sistemas de información crecen más fuertes y más complejos, manejarlos se hace cada vez más difícil. El servidor Sybase SQL SMBDR ofrece facilidades poderosas para simplificar el manejo del sistema.

#### **Características de administración.**

- Tres niveles de configuración, de básico a detallado, permiten un alto grado de optimización del sistema.
- El servidor de respaldo hace el respaldo y recupera procedimientos de manera rápida y automática.

#### **MEJORANDO LA PRODUCTIVIDAD DEL USUARIO.**

La facilidad de uso es una importante consideración en la selección de una base de datos. El servidor Sybase SQL SMBDR tiene muchas características que lo hacen fácil de usar.

#### **Características de productividad.**

- Arreglo obligatorio para alto diseño de productividad y tiempo de corrida eficiente.
- Múltiples acciones en la misma conexión facilitan la aplicación del diseño.

## **MEJORANDO LA CONFIABILIDAD DEL SISTEMA.**

Cuando las organizaciones dependen de sus sistemas de información, una caída puede ser desastrosa. El servidor Sybase SQL está diseñado para soportar aplicaciones en línea para poder ser habilitadas 24 horas al día, 7 días a la semana.

### **Características de disponibilidad y recuperación.**

- Los administradores pueden ejecutar mantenimiento en línea mientras las aplicaciones continúan accediendo y actualizando la base de datos.
- Un mecanismo configurable de revisión escribe todos los cambios en páginas de datos de la cache al disco, incrementando la velocidad de recuperación del sistema.

## **SYBASE: INTEGRANDO LA DISTRIBUCIÓN DE LA EMPRESA.**

Las organizaciones alrededor del mundo están enfrentando cambios extraordinarios experimentando una explosión de tecnología y datos. Como las fuerzas de trabajo se han venido incrementando en forma descentralizada, manejar información en forma efectiva es tal vez el único gran cambio para los negocios hoy. Sybase, el pionero en las bases de datos cliente/servidor, ofrece una completa solución de bases de datos, herramientas, y servicios que ayudan a los usuarios a acceder información donde y cuando lo necesiten. Con Sybase, se tendrá el poder para simplificar e integrar la distribución de la empresa, así que será factible responder a los rápidos cambios en los negocios.

## **ESPECIFICACIONES TECNICAS.**

### **Requerimientos de hardware.**

- 15 MB de RAM para el servidor SQL.
- 48 KB de RAM por usuario adicional.
- 6 MB de espacio de disco para programas del sistema.

## **ESTADÍSTICAS DEL PRODUCTO.**

### **Bases de datos.**

- 32767 bases de datos por servidor SQL.
- Tamaño de base de datos mayor a 32TB.
- Más de 8 bases de datos habilitadas para una actualización.
- Más de 16 bases de datos abiertas para una transacción.
- Más de 16 tablas en un query.

### **Tablas.**

- 2 GB de objetos para base de datos.
- 250 columnas por tabla.
- 251 índices por tabla.
- Registros por tabla solo limitados por el espacio de disco.
- 16 columnas por índice compuesto.
- 30 caracteres por nombre del objeto de base de datos.

### **Procedimientos almacenados y triggers.**

- Ilimitada longitud KB.
- Número limitado solo por el espacio de disco.
- Pueden configurarse más de 16 anidamientos.
- Pueden configurarse triggers con autorreferencia.

**TIPOS DE DATOS.****Numéricos.****int**

entre -2,147,483,648 y +2,147,483,647 inclusive.

**smallint**

entre -32,768 y +32,767 inclusive.

**tinyint**

entre 0 y 255 inclusive.

**float**

números de 8 bytes de punto flotante.

**shortfloat**

números de 4 bytes de punto flotante.

**money**

las columnas de moneda almacenan valores exactos entre +/- 922,337,203,685,447.5807 dólares con 4 lugares para decimales.

**double precision**

números de 8 bytes de punto flotante .

**Caracter.****char(n)**

## **Apéndice C**

---

columnas de caracteres (letras, números, símbolos), más de 255 caracteres de longitud.

### **varchar(n)**

columnas de caracteres de longitud variable (letras, números, símbolos), más de 255 caracteres de longitud.

### **Gran objeto binario.**

#### **text**

columnas de caracteres de longitud variable de más de 2 GB de longitud.

#### **image**

columna binaria de longitud variable de más de 2 GB de longitud.

#### **binary**

columna binaria de más de 255 bytes de longitud.

#### **varbinary(n)**

columna binaria de longitud variable, más de 255 bytes de datos binarios .

### **Miscelánea.**

#### **bit**

columna de bits que soporta únicamente 0's y 1's.

#### **datetime**

fecha y hora del día con una precisión de 1/30 de milisegundo.

#### **shortdatetime**

fecha y hora del día con una precisión de 1 minuto.

**timestamp**

una columna de registro de tiempo se actualiza automáticamente cuando un registro es alterado.

**identity**

número de secuencia de mantenimiento de sistema.

# APENDICE D

## PowerBuilder 5.0

Desde hace tiempo se han descubierto varias alternativas de aplicaciones cliente/servidor para cubrir las necesidades de los negocios modernos. Ahora, no sólo se requiere cubrir la demanda de aplicaciones, hay altas expectativas acerca de la ejecución y flexibilidad de las aplicaciones.

Muchos de los negocios de hoy requieren operaciones multiplataforma y procesos distribuidos.

Debido a que la revolución en los negocios de computación continúa, los desarrolladores de aplicaciones requieren herramientas que los ayuden a actualizarse, necesitan de un ambiente de desarrollo estratégico diseñado para ayudar al rápido crecimiento de la tecnología.

#### **OBJETOS DISTRIBUIDOS.**

La arquitectura orientada a objetos de PowerBuilder nos proporciona flexibilidad para distribuir objetos a través de un ambiente de red. Estos objetos particionados pueden ser compilados para alta ejecución. El desarrollo orientado a objetos incluye encapsulación de objetos, polimorfismo, herencia, etc.

PowerBuilder 5.0 incluye la nueva librería "PowerBuilder Foundation Class" (PFC), el cual acelera el desarrollo de ejecuciones con un set de objetos y servicios reutilizables.

#### **APLICACIONES RAPIDAS.**

La capacidad de compilación de PowerBuilder mejora la ejecución de las aplicaciones, incluyendo la ejecución rápida de scripts, expresiones matemáticas, ciclos (tales como FOR ... NEXT), y procesamiento de arreglos. Además, PowerBuilder optimiza los manejadores nativos de bases de datos para conectar inmediatamente las aplicaciones a un amplio rango de fuentes de datos.

### **APLICACIONES EN CUALQUIER LUGAR.**

En PowerBuilder 5.0 se pueden crear aplicaciones cliente/servidor que corran en Windows, Macintosh, Solaris, particiones de estas aplicaciones para utilizar Windows NT y servidores Solaris, en LAN's, WAN's o nuevas redes Internet o Intranet.

### **INTERNET.**

Se pueden construir aplicaciones del servidor Web que genera contenidos dinámicos para visualizadores Web y permite que las aplicaciones trabajen con cualquier visualizador y en cualquier servidor Web que haya en el mercado.

### **EDICIONES DE POWERBUILDER.**

- Empresa : Aplicaciones distribuidas.
- Profesional: Aplicaciones departamentales.
- Escritorio : Aplicaciones cliente/servidor para grupos de trabajo pequeños.

Existen diversas formas mediante las cuales nos podemos actualizar continuamente en información acerca de productos, servicios, programas y de la comunidad virtual de desarrolladores de cliente/servidor e Internet.

### **REQUERIMIENTOS MINIMOS DE SISTEMA.**

#### **Macintosh.**

- 68030, 68040 o procesador PowerPC.
- Macintosh System Software 7.5.3.
- 10 MB en RAM (18 MB para procesadores de 64K).
- 45 MB de espacio en disco duro (configuración mínima).

**Windows 3.x, Windows 95, o Windows NT.**

- Procesador 486SX.
- 12 MB en RAM (se recomiendan 16 MB).
- 32 MB de espacio en disco duro.

**UNIX.**

- Sun Solaris 2.5.
- CDE 1.01.
- 32 MB en RAM.
- 96 MB swap.
- 100 MB de espacio en disco duro.

**DISTRIBUCION Y DESPLIEGUE DE APLICACIONES EN POWERBUILDER.**

PowerBuilder 5.0 permite distribuir lógica de aplicaciones dentro de servidores para implementaciones múltiples. Las aplicaciones en PowerBuilder diseñadas para una arquitectura distribuida ofrece una solución ideal para un despliegue dinámico en Internet.

- Ahora se pueden acceder objetos distribuidos en PowerBuilder, no solamente aplicaciones de PowerBuilder, sino también de otros paquetes populares como el Notes o el Excel.
- Creación de "clientes ligeros" moviendo lógica de negocios, incluyendo objetos compilados para aplicaciones servidor, dando como resultado un ambiente gráfico rápido y el mantenimiento de aplicaciones simplificadas.

## Apéndice D

---

- Con PowerBuilder 5.0 se pueden generar archivos estándar HTML de una DataWindow (incluyendo una DataWindow no visual conocida como DataStore), y crear archivos disponibles HTML a través de un visualizador Web.
- PowerBuilder permite crear aplicaciones basadas en Web que genera contenido dinámico para visualizadores Web.
- Las aplicaciones clientes en PowerBuilder corren en plataformas Windows 95, Windows 3.x, Windows NT, UNIX Sun Solaris, y Macintosh.
- Las aplicaciones servidores en PowerBuilder corren en plataformas Windows 95, Windows NT y UNIX Sun Solaris.

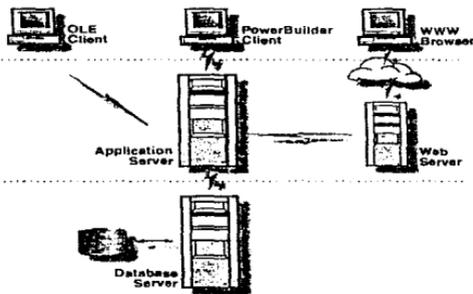


Ilustración D 1

**ANTECEDENTES.****Two-Tier.**

Hoy en día, PowerBuilder ha tenido mucho éxito al proveer a las corporaciones con un ambiente desarrollado en two-tier para crear sistemas cliente/servidor. Una arquitectura two-tier divide las aplicaciones en dos partes:

1. Un cliente para interface de usuario y lógica de aplicaciones.
2. Un servidor para la administración de las bases de datos.



Ilustración D 2

**Modelo tradicional de cómputo distribuido Two-tier.****Multi-Tier.**

Una arquitectura Multi-Tier tiene la mitad de un Two-Tier que consiste de una o más aplicaciones servidor donde reside la lógica de procesamiento de negocios. Particiona la aplicación dentro de capas separadas para la interface de usuario gráfica, lógica de negocio y procesamiento de bases de datos. El modelo Multi-Tier hace fácil la expansión y mantenimiento de aplicaciones además de que nos proporciona un alto nivel de ejecución.

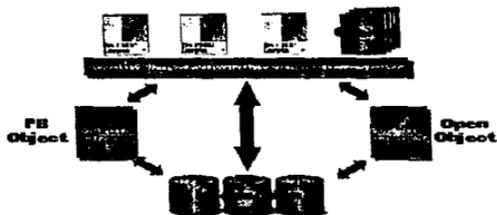


Ilustración D 3

Modelo de cómputo distribuido Multi-tier.

## **BENEFICIOS DEL COMPUTO DISTRIBUIDO.**

### **Encapsulamiento Lógico.**

Se trata de la habilidad de separar una porción de la lógica de negocio de una aplicación sin tener que desplegar nuevas interfaces de usuario. La habilidad de cambiar la programación back-end del servidor sin interrumpir a los usuarios, distribución de interfaces de usuario nuevas o reinstrucciones es un enorme beneficio para las organizaciones IS.

### **Performance.**

Las arquitecturas Multi-Tier mejoran la ejecución de las aplicaciones transfiriendo el procesamiento de lógica de aplicación/negocio para un servidor independiente.

**Consistencia/Control.**

En ambientes de cómputo distribuido, la consistencia y el control van de mano en mano. Como la lógica de negocio no reside en el cliente, IS puede controlar y monitorear muy de cerca cómo el servidor está siendo accedido.

La parte del cliente de más organizaciones IS varía ampliamente y tiene múltiples tipos de hardware, software y locaciones físicas del cliente. A estas organizaciones les es difícil controlar y garantizar que todos los clientes corran las mismas rutinas. Con el modelo de cómputo distribuido, muchos de estos problemas operacionales son minimizados.

**Flexibilidad.**

Con mas presión sobre las organizaciones IS para reaccionar a las condiciones de negocio, los desarrolladores requieren de una metodología de desarrollo flexible que permita al IS cubrir la demanda de aplicaciones, mientras que los usuarios finales permanecen coniendo a alta velocidad. La lógica de negocios distribuida para servidores permite a las organizaciones IS realizar cambios sin intervenir en la productividad del cliente.

**ESTRATEGIA DE COMPUTO DISTRIBUIDO DE POWERSOFT.**

Las aplicaciones distribuidas que se crean con PowerBuilder pueden tomar dos parámetros:

- Soporte abierto para herramientas de cómputo distribuido de terceros.
- Soporte de desarrollo para aplicaciones PowerBuilder particionadas o distribuidas.

**SOPORTE ABIERTO PARA HERRAMIENTAS DE COMPUTO DISTRIBUIDO DE TERCEROS.**

Los pasos que un desarrollador toma para crear aplicaciones three-tier con PowerBuilder 4.0 y 5.0 son los que se muestran a continuación:

- Desarrollar la interface para el procedimiento remoto usando herramientas de cómputo distribuido de nuestra elección. En caso de más monitores deberá usarse un lenguaje de definición de interface. Un lenguaje que defina los parámetros para la función lógica de la aplicación que es invocada.
- Compilar el lenguaje de definición de interface dentro de un cliente y un servidor para permitir la interface que será invocada por un cliente PC. Esto probablemente incluye algún código "C" o "C++" para convertir los tipos de datos de PowerBuilder a tipos de datos soportados por la interface.
- Crear una definición externa dentro del ambiente de PowerBuilder para llamar a la función de interface y crear la DataWindow para desplegarla.

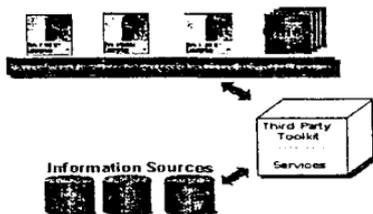


Ilustración D 4

**Herramientas de cómputo distribuido de terceros así como PowerBuilder.**

**DISPONIBILIDAD DE HERRAMIENTAS DE COMPUTO DISTRIBUIDO.**

Las herramientas están actualmente disponibles por los siguientes vendedores:

Compañía	Producto
AT&T	TopEnd
Gradient Technologies Inc.	Visual DCE
Greenbrier Russell	RPC Painter
Magna Software	MAGNA-X
NobleNet	EZ-RPC Product Family
Novell	Tuxedo
Open Environment Corp.	Entera
Tangen International	Distributed Computing Integrator
Test Systems International	SYSQL y ADAQL
Transarc Corporation	Encina y DCE

**POWER BUILDER DISTRIBUIDO.**

PowerBuilder 5.0 es un ambiente de desarrollo de cómputo distribuido, donde los desarrolladores pueden crear objetos que contienen reglas de negocios y las distribuyen dentro de uno o más servidores de aplicaciones. El ambiente "Multi-Tier" permite un número ilimitado de servidores de aplicaciones.

PowerBuilder 5.0 ofrece alternativas rápidas para desarrollar aplicaciones distribuidas sofisticadas. Los desarrolladores pueden crear aplicaciones distribuidas rápida y fácilmente usando objetos no visuales PowerBuilder para definir la lógica de procesamiento de negocio. Además, ahora pueden distribuir objetos PowerBuilder a través de la red para incrementar la escalabilidad y administración centralizada.

#### CONCEPTOS Y TERMINOLOGIA.

PowerBuilder distribuido (DPB) introduce algunos nuevos conceptos y terminología y construye un lenguaje nuevo PowerScript para soportar DPB. Esto incluye:

- Nuevas clases de objetos PB heredadas de objetos no visuales y estructuras en la jerarquía de clase super objeto. Estas nuevas clases son:

**nonvisualobject**

**connectionobject (new)**

**connect (new)**

**transport (new)**

**remoteobject (new)**

**structure**

**connectioninfo (new)**

- PowerBuilder 5.0 introduce el concepto de aplicación de servidor PowerBuilder donde una aplicación contiene objetos de usuario no visuales (NVUOs) para ser invocados por una o más aplicaciones cliente.
- Una aplicación cliente PowerBuilder invoca objetos y servicios remotos. La aplicación cliente DPB contiene un nuevo objeto Powerscript llamado "Conexión" que sirve para conectarse a una aplicación servidor PB.

- Un objeto remoto contiene lógica de negocios escrita como métodos PowerScript.
- El método de transporte de objeto "Listen" es ejecutado para aceptar peticiones del cliente basados en los atributos del método de comunicación en el transporte de objeto.
- Al crear una conexión PowerScript, el método del objeto de la conexión "ConnectToServer" es ejecutado para conectar la aplicación cliente al servidor basado en la lista de atributos del método de comunicación en el objeto "Conexión".
- Save Proxy guarda un nombre remoto para el objeto y genera código interno para habilitar el objeto usuario a ser controlado a distancia.
- Object Proxy se refiere al código interno generado por remoteobject para habilitar la ejecución remota del método objeto.

### **CREACION DE UNA APLICACION SERVIDOR.**

Crear una aplicación de servidor DPB es muy similar a crear una aplicación tradicional en PowerBuilder.

Una aplicación puede funcionar sin DPB intercalando con las peticiones de los clientes a través de un conjunto de controles de interface de usuario. Se puede desplegar una interface simple del panel de control para empezar, parar y desplegar un diagnóstico o información estadística.

Los objetos son clases de objetos reales PB, los desarrolladores pueden construir nuevos objetos con los existentes pero con nueva funcionalidad. Esto no nos proporciona una solución para las clases de problemas direccionables por aplicaciones y almacenamientos de objetos, pero el grupo centralizado de la funcionalidad de negocio que proveen los objetos DPB es consistente con una migración eventual para tales efectos.

### **CREACION DE UNA APLICACION CLIENTE.**

Una aplicación cliente DPB es idéntica a una PB tradicional, la única diferencia es que un objeto "Conexión" ha sido agregado para permitir que clientes DPB se conecten a aplicaciones servidores.

El creador de la aplicación cliente debe implementar la interface de usuario y los scripts de la aplicación cliente para ejecutar la aplicación y evaluar su código.

El encapsulado funcional de la aplicación en NVUO puede ser incrementado y evaluado con la aplicación del cliente residente con el parámetro de la librería de la aplicación cliente y es movido o copiado al servidor PBL y ejecutado remotamente, o bien, si ya existen las librerías en un servidor PBL o PBD siendo posteriormente copiado al cliente PBL.

**PLATAFORMAS QUE SOPORTA POWER BUILDER DISTRIBUIDO.**

<b>Plataforma</b>	<b>Manejador de Comunicaciones</b>	<b>Servidor</b>	<b>Cliente</b>
Windows 3.x	NamedPipes	No	No
	Sockets	No	Si
	OpenClientServer	No	Si
Windows NT 3.51	NamedPipes	Si	Si
	Sockets	Si	Si
	OpenClientServer	Si	Si
Windows95	NamedPipes	No	Si
	Sockets	Si	Si
	OpenclientServer	No	Si
PowerBuilder UNIX	N/A		
Powerbuilder Mac	N/A		

**REQUERIMIENTOS DE PLATAFORMAS DEL MANEJADOR DE COMUNICACIÓN.**

Manejador de Comunicaciones	Plataforma	Requerimientos
Named Pipes Server PENPS050.DLL	Windows NT	N/A
Named Pipes Client PBNC050.DLL	Windows NT	N/A
Winsock Server PBWSS050.DLL	Windows NT Windows 95	Requiere TCP/IP instalado y configurado Requiere TCP/IP instalado y configurado
Winsock Client PBWSC050.DLL	Windows NT Windows 95 Windows WFW 3.11	Requiere TCP/IP instalado y configurado Requiere TCP/IP instalado y configurado Requiere TCP/IP 32 para WFW
Open Client/Server Server PBOSS050.DLL PBOSRV050.EXE	Windows NT	Requiere Sybase Open Server instalado y configurado.  Limitaciones:  - El servidor no puede funcionar como un cliente por sí mismo, el servidor puede ser un cliente para otros servidores, no solo

<b>Manejador de Comunicaciones</b>	<b>Plataforma</b>	<b>Requerimientos</b>
		<p>para si mismo.</p> <p>-Únicamente un objeto de transporte "Open Client Server" puede "Listen" (escuchar) al mismo tiempo en un servidor app..</p> <p>Notas: Estas son limitaciones de CSLIB/CTLIB.</p> <p>Un objeto de transporte puede ser escuchado en múltiples protocolos simultáneamente (Names Pipes, Winsock, IPX).</p>
<p>Open Client/Server Client</p> <p>PBOSC050.DLL</p>	<p>Windows NT</p> <p>Windows 95</p> <p>Windows 3.x</p>	<p>Requiere Sybase Open Client para Windows NT instalado y configurado.</p> <p>Requiere Sybase Open Client instalado y configurado. Los planes actuales nos dicen que Open Client para NT debería trabajar. Sybase puede tener una versión separada de Windows 95.</p> <p>Requiere Sybase Open client para Windows instalado y configurado (este manejador no forma parte de Power Builder)</p> <p>Limitaciones: No se pueden enviar &gt; 64 K de datos al servidor, CAN recibe &gt; 64K.</p>

Apéndice D

Manejador de Comunicaciones	Plataforma	Requerimientos
Local Driver PBDPB050.DLL	Windows NT	No se requiere software adicional.
	Windows 95	No se requiere software adicional.
	Windows 3.x	No se requiere software adicional

**APENDICE E**  
**CASE ERWin**

## INTRODUCCION.

ERwin es una herramienta para modelar el diseño de bases de datos cliente/servidor, Web/Intranet y aplicaciones warehousing de alto desempeño. Con la combinación de herramientas visuales, una base madura en la definición de datos y soportando a los líderes de bases de datos SQL y a las de escritorio, ERwin facilita su rápida creación o bien ayuda en procesos de reingeniería para los sistemas de bases de datos relacionales del mundo actual.

En términos simples ERwin ayuda a crear el modelo de datos, en un ambiente de desarrollo visual permitiendo manejar datos complejos tal y como lo requieren los sistemas actuales.

Con ERwin se pueden dibujar diagramas de entidad-relación como los mostrados en la **Ilustración E 1**, pero ERwin es más que una herramienta de dibujo. ERwin no solo ayuda en el diseño lógico del modelo de datos, si no que automáticamente construye la estructura física del modelo de datos con la información contenida en el diagrama.

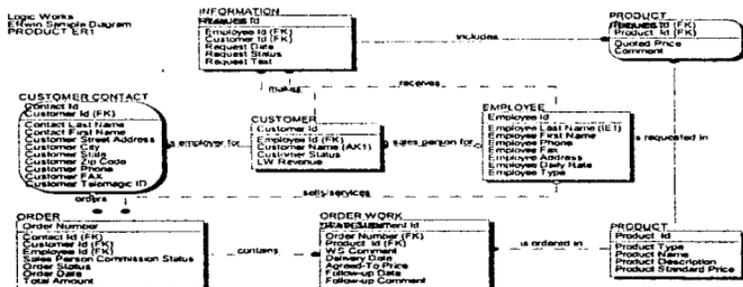


Ilustración E 1

Cuando el modelo de datos lógico esta listo solo basta con seleccionar la plataforma de bases de datos con la cual se desea trabajar y se elige la generación del esquema. En

minutos ERwin desarrolla el modelo de datos físico de la base de datos, incluyendo todas las tablas, índices, procedimientos almacenados, integridad referencial (triggers) y demás componentes que se requieren para un mejor manejo de los datos.

## REPRESENTACIÓN Y MANEJO DEL MODELO LÓGICO.

En ERwin el diagrama de modelo de datos es utilizado para generar una base de datos relacional; un sistema de almacenamiento en el que cada dato se encuentra en las tablas. Cuando se crea un diagrama en ERwin , el modelo de información está representado por entidades (gente, lugares, cosas), atributos (características de las entidades como nombre de personas, dirección, edad, etc.) y relaciones entre entidades. Cada entidad corresponde a una tabla en la base de datos, cada instancia corresponde a un renglón de la tabla y los atributos de la entidad corresponden a las columnas.

CUSTOMER Database Table			
ID	Name	Address	Status
1	B. Jones	7 Birch St.	1
2	T. Wall	18 Main St.	2
3.	U. Calc	2 Lark Blvd.	1

Customer Entity
Customer ID
Customer Name
Customer Address
Customer Status

Ilustración E 2

Cada instancia o renglón de la entidad está identificada como única por la llave primaria de la entidad, uno o más atributos pueden formar parte de la llave primaria, los atributos que forman parte de la llave primaria están ubicados arriba de la línea, los demás atributos se encuentran por debajo de esta línea. Una entidad cuya llave primaria esta formada por uno o más atributos de otra entidad se denomina entidad dependiente y se indica como un cuadrado con los vértices redondeados.

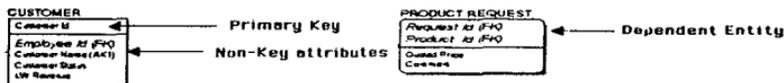


Ilustración E 3

Los DBMS utilizan relaciones para ligar los renglones de datos de diferentes tablas, se representan con una frase y una línea que une a ambas entidades. Por ejemplo las tablas *Customer* y *Order* establecen una relación como se muestra en la **Ilustración E 4** en la cual un comprador realiza una orden de compra.

La cardinalidad de la relación se representa como un punto al final de la línea. La relación expresa una cardinalidad de uno a muchos; esto es, un consumidor (entidad padre) genera muchas órdenes (entidad hija). Si la entidad hija hereda un atributo de la entidad padre, el

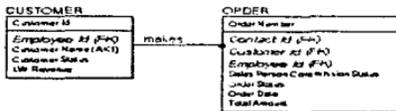


Ilustración E 4

atributo heredado se denomina llave foránea y se etiqueta en ERwin como (FK).

Cuando se realiza la generación de la base de datos física, ERwin genera automáticamente un script con sentencias SQL como *create table* y *create index* para crear las tablas, incluirles índices, restricciones y triggers que refuerzan la integridad referencial entre las tablas.

## INTERFAZ GRÁFICA COMPLETA.

El espacio de trabajo de ERwin esta contenido en una ventana en donde se crean los diagramas de modelo lógico de datos y desde donde se tiene acceso a todas las herramientas usadas durante el proceso de modelado, **Ilustración E 5.**

- Barra de Menú de ERwin para obtener información acerca del uso del menú y editor de ERwin.
- Una barra de herramientas completa que muestra la descripción y uso de los botones de una manera fácil y sencilla.
- Listas desplegables en fuentes y color para mayor facilidad en el uso de texto para los diagramas.
- Lista desplegable contextual en el lado derecho del objeto para acceder a las opciones más importantes para el objeto señalado.

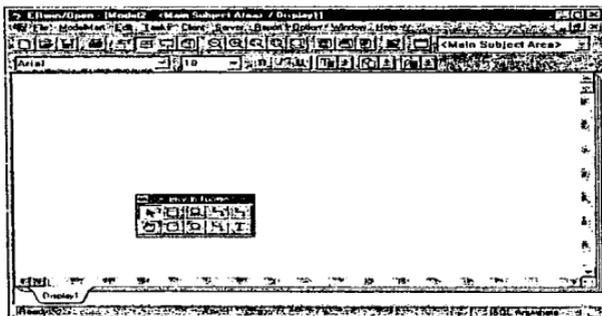


Ilustración E 5

## INGENIERÍA DE REVERSA Y HACIA ADELANTE.

ERwin utiliza la información especificada en el modelo de datos lógico para poder crear el modelo físico de una base de datos relacional, incluyendo diversas etapas: mantenimiento, documentación y exportación. El proceso de generación de la base de datos física a partir del modelo de datos lógico es llamado Ingeniería hacia adelante; y al proceso de construir el modelo de datos lógico a partir de una base de datos física es llamado Ingeniería de reversa.

La Ingeniería hacia adelante de Erwin genera automáticamente una base de datos física directamente del modelo lógico de datos desarrollado, sin la necesidad de escribir una sola línea de código SQL tanto para sentencias como para triggers.

La ingeniería de reversa de Erwin puede inferir exactamente las relaciones entre las tablas de datos, de igual forma facilita la modificación de las estructuras de la base de datos existente y convertirla a otro DBMS- independiente. Puedes usar el modelo para un análisis detallado, redefinirlo después y distribuirlo como parte de la documentación requerida durante el ciclo de desarrollo del sistema.

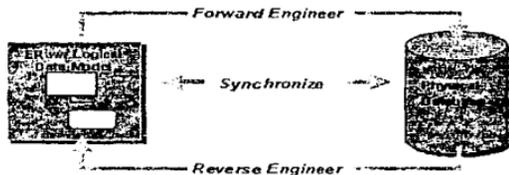


Ilustración E 6

Si se quiere migrar una base de datos existente de una plataforma a otra, Erwin puede realizar una Ingeniería de reversa con esta base de datos, crear el modelo de datos lógico, permitir la modificación y agregar nuevos elementos y finalmente poder construir la base de datos física en alguno de los siguientes servidores de bases de datos:

**Bases de datos SQL**

- As/400
- DB2/MVS
- DB2/2
- Informix
- Ingres/Openingres
- InterBase
- Netware SQL
- Oracle

- Progress
- Rdb
- SqlBase
- SQL Server
- SYBSE
- Teradata
- Watcom/SQL Anywhere

**Bases de datos Desktop**

- Microsoft Access
- Paradox
- dBASE III
- dBASE IV
- Microsoft FoxPro
- Clipper

**ERWIN/DESKTOP PARA POWERBUILDER.**

ERwin/Desktop para PowerBuilder es una herramienta de modelado que permite trabajar con más de 40 entidades. ERwin/Desktop permite realizar ingeniería de reversa y hacia adelante a aplicaciones y genera de manera rápida los triggers y bases de datos para Watcom SQL. Con el diccionario extendido de ERwin se puede tener acceso a los atributos de las aplicaciones desarrolladas en PowerBuilder.

El diccionario de ERwin almacena atributos extendidos de PowerBuilder para reutilizarlos.

La liga bidireccional entre ERwin y PowerBuilder soporta las especificaciones de los atributos extendidos de PowerBuilder contenidos en el modelo lógico de datos. Los atributos extendidos son definidos y manejados en el diccionario de ERwin garantizando la integridad de los datos, consistencia y reforzando las transacciones de operaciones.

**Ventajas**

- Diseño lógico y físico del modelo de datos con más de 40 entidades.
- Generación de las tablas a partir del modelo lógico hacia Watcom SQL o ingeniería de reversa para la generación del modelo lógico a partir de bases de datos existentes.

- Lectura y escritura de SQL scripts.
- Liga bidireccional de ERwin a PowerBuilder permitiendo especificar atributos extendidos de PowerBuilder contenidos en el modelo de datos.
- Sincronización de los valores de atributos extendidos entre PowerBuilder y ERwin con despliegue de diferencias y recomendaciones para su rápida conciliación.
- Permite la selección de los parámetros de la base de datos y los atributos extendidos de PowerBuilder para ser utilizados en múltiples columnas.
- Facilidad de reportes para la vinculación a través de DDE hacia Word, Excel y Word Perfect.
- Escalable a ERwin/ERX para PowerBuilder.

**Requerimientos de hardware y software.**

- Procesador 386,486, o Pentium corriendo con Windows 3X, NT o 95.

**ERwin/ERX PARA SOLARIS.**

**Características:**

- ERwin/ERX para Solaris es una herramienta nativa de 32-bit para Sun SPARC.
- Interface GUI intuitiva con soporte para cualquier X-terminal que cumpla con la interface.
- Desarrollo visual de diagramas de entidad relación (E-R) que incluye una notación de tipo estándar.

- El mismo modelo de bases de datos puede ser generado para diferentes plataformas o convertir un modelo de bases de datos de una plataforma a otra.
- Server FRE (Forward- and Reverse-Engineering) crea una conexión directa entre el modelo lógico de datos y las tablas del sistema de Oracle y Sybase.
- ERwin genera archivos script de SQL para una amplia variedad de bases de datos incluyendo Informix, CA-Ingres y DB2.
- Editor de plantillas para generar de manera automática procedimientos almacenados y triggers.

Simplemente basta realizar algunos clicks para diseñar un modelo gráfico E-R. Los diagramas E-R desplegarán todas las entidades, atributos, índices, llaves primarias y foráneas. El modelo de datos puede ser editado visualizado e impreso en muchas formas desde los modelos lógicos de alto nivel. Los diagramas de ERwin de presentación y clase imprimen diversos tipos de letra (fonts) y colores.

Server FRE proporciona una conexión entre ERwin y su sistema de tablas de Sybase u Oracle. ERwin también captura parámetros de almacenamiento físico, maneja tus mensajes y controla las propiedades físicas de la base de datos tales como el espacio, el tamaño y la ubicación de las tablas.

Los cambios del modelo de datos pueden ser transmitidos para actualizar la base de datos actual o crear una nueva. Esto le permite rediseñar aplicaciones con nuevos datos y fracciones requeridas por el ambiente cliente/servidor.

### **Independencia de DBMS.**

El mismo modelo puede ser utilizado para generar múltiples bases de datos o para convertir una aplicación de una plataforma a otra.

**Características importantes en la elaboración de modelos E-R:**

1. Migración de llaves foráneas.
2. Utilización de llaves alternas para el manejo de índices.
3. Dominios.
4. Particionando y unificando.
5. Supertipo/subtipo.
6. Integridad referencial.
7. Editor de características avanzadas.
  - Definición de Atributos y Notas.
  - Utilización de Default.
  - Reglas de Validación.
  - Validación/valores.
  - Entidad-Atributo.
  - Definición de Entidad.
  - Entidad nota, consulta y ejemplo de datos.
  - Índice.
  - Cardinalidad.
  - Integridad Referencial.
  - Definición de tipos de usuarios.
  - Definición de frases verbales.

**Productividad de desarrollo en Cliente/Servidor.**

1. Servidor basado en Ingeniería de reversa y hacia adelante (Server FRE) para Oracle y Sybase. Otros DBMS son soportados a través de la generación del archivo script SQL.
2. Sincronización bidireccional del servidor de Oracle y Sybase.
3. Diccionario con chequeo interno y externo.
4. Modelado particionado para el desarrollo de varias áreas.
5. Procedimientos almacenados y editores de triggers.

**Calidad de presentación de los Reportes.**

1. Calidad de presentación gráfica
2. Reportes Flexibles.
  - Reportes de entidad.
  - Reportes de atributos.
  - Reportes relacionales.
  - Dominios y default.
  - Reporte por filtros.

**Requerimientos de hardware y software.**

- Sun Solaris 2.3 o Windows.
- 40 MB de espacio libre en disco para los archivos del producto
- A lo menos 100 MB de espacio para compartición.

- Unidad de CD\_ROM.

## MODELADO EN EL SERVIDOR.

De la misma manera en que se usa el editor de ERwin para definir los nombres lógicos de las entidades cuando se está diseñando un modelo de bases de datos lógico, ERwin provee de un editor de características del DBMS utilizado, donde se pueden definir los nombres físicos de las entidades, las propiedades de las tablas, columnas y las relaciones que se crean durante la generación del modelo físico.

Por ejemplo cuando se define algún tipo de propiedad para las columnas, como se muestra en la figura siguiente; se puede asignar los tipos de datos utilizados por el servidor SQL y utilizar la validación de datos para conocer los tipos de datos generados durante la creación física.

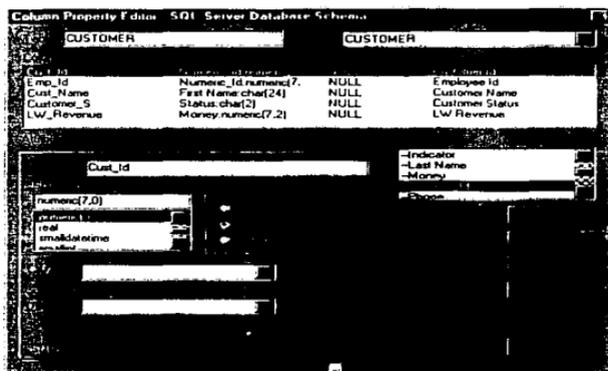


Ilustración E 7

Cuando ERwin crea el esquema de bases de datos, éste genera un archivo DLL con la sintaxis acorde a la plataforma de base de datos especificada.

Si el servidor de base de datos seleccionado soporta características avanzadas como procedimientos almacenados e integridad referencial por triggers, ERwin provee de un editor de plantillas para la generación de este tipo de estructuras.

## **MODELADO EN EL CLIENTE.**

ERwin/OPEN es una versión cliente/servidor de Logic Works que combina el diseño en el back-end y el desarrollo de aplicaciones en el front-end para un ambiente unificado.

Las características de desarrollo de multi-cliente abarca los líderes de desarrollo de aplicaciones front-end: Visual Basic, PowerBuilder y PROGRESS desde el mismo modelo lógico de datos, haciendo el desarrollo de aplicaciones a nivel corporativo con un alto grado de eficiencia y facilidad.

Por ejemplo dando click en la carpeta de Visual Basic en la parte inferior del editor de propiedades, se le pueden asignar diferentes estilos a las columnas de datos (ej. cuadros de texto, cuadros de listas, botones radiales, etc), valores por default, reglas de validación y otras características propias de cada cliente y las cuales serán creadas en una interfaz para ser ejecutadas de manera inmediata como una forma de Visual Basic.

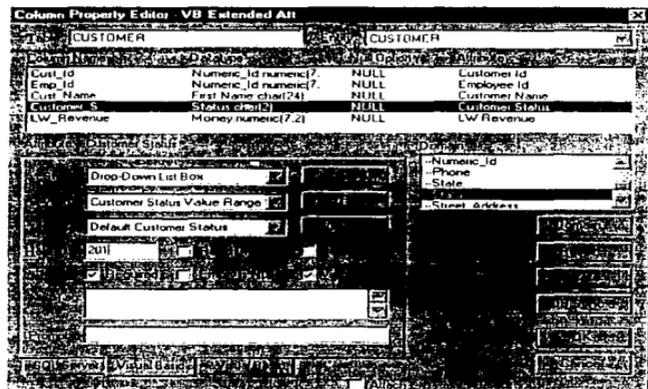


Ilustración E 8

De esta manera se simplifica el desarrollo de aplicaciones cliente. Logic Works también proporciona DataBOT, un software de generación dinámica de código de acceso SQL para crear rápidamente y de manera sofisticada aplicaciones robustas para bases de datos. Este producto está disponible para los más importantes líderes de desarrollo de aplicaciones visuales.

# GLOSARIO

---

<b>Administrador de Archivos</b>	Software que administra y controla el acceso a los archivos de datos.
<b>Administrador de Red</b>	Individuo responsable de administrar una red.
<b>Ambiente</b>	Son las circunstancias o elementos que rodean a una aplicación.
<b>Ambiente Heterogéneo</b>	Una configuración de computadoras y programas de una gran variedad de fabricantes.
<b>Análisis Estructura</b>	Técnicas desarrolladas a finales de los 70s por Yourdon, DeMarco, Gane y Sarson para aplicar un método sistemático al análisis de sistemas. Incluía la utilización de diagramas de flujo de datos y fomentaba el uso de notas gráficas independientes para la documentación.
<b>Ancho de Banda</b>	Capacidad de transmisión de una línea de comunicación o canal.
<b>ANSI (Instituto Nacional Americano de Estándares)</b>	Es una organización que agrupa a las industrias para implementar estándares que faciliten el desarrollo de su trabajo.
<b>API (Interface del Programa de Aplicación)</b>	Es la interface que ocurre en el nivel de programación permitiendo que dos piezas de software interactúen.

<b>Aplicación</b>	Tradicionalmente este término se asocia al software que realiza una función específica.
<b>Archivo</b>	Es un respaldo de información que se guarda en diferentes almacenamientos como discos, cintas, etc..
<b>ARPANET (Agencia Avanzada de Investigación de Proyectos de Redes)</b>	Es una red que consistió de 10000 computadoras en los años 60's permitiendo a las universidades y centros de investigación el intercambio de información.
<b>Arquitectura</b>	La estructura de una computadora, sus funciones a detalle y componentes. Una arquitectura puede incluir tanto hardware como software.
<b>Arquitectura Cerrada</b>	Software o hardware con especificaciones técnicas propias que no son hechas para aplicaciones públicas.
<b>Arquitectura Cliente/Servidor</b>	Es una arquitectura que consiste de un módulo cliente que hace requerimientos a un módulo servidor, el cual responde a dichos requerimientos. El módulo cliente comúnmente esta en una PC o workstation y es conectada a un

---

	servidor vía red.
<b>Arquitectura de Red</b>	La estructura de un sistema de comunicaciones que incluye hardware, software, protocolos y métodos de acceso.
<b>ASCII (American Standard Code for Information Interchange)</b>	Código americano estándar para intercambio de información. Código binario de datos que se usa en comunicaciones, en la mayor parte de minicomputadoras y en todas las computadoras personales.
<b>ATM (Asynchronous Transfer Mode)</b>	Modo de transferencia asíncrona. Red estándar para transmitir información a alta velocidad por medio de fibras ópticas. Utiliza un paquete de 53 bytes de longitud fija para datos.
<b>Backbone</b>	Un término de red usado para describir la porción principal de una red a la cual se conectan subredes y consecuentemente por donde pasa la mayoría de tráfico.
<b>Banda</b>	Un rango de frecuencias usado para transmitir una señal.
<b>Banda Base</b>	Una técnica de comunicación usualmente limitada, en la cual las señales digitales están situadas sobre la

	línea de transmisión sin cambios de modulación.
<b>Base de Datos</b>	Conjunto de archivos interrelacionados.
<b>Base de Datos Distribuida</b>	Una colección de múltiples bases de datos interrelacionadas distribuidas sobre una red.
<b>Base de Datos Orientada a Objetos</b>	Una base de datos que incorpora el paradigma orientado a objetos en DBMS.
<b>Bath (Procesamiento en Lote)</b>	Procesamiento de transacciones agrupadas.
<b>Benchmark</b>	Evaluación comparativa, prueba de referencia. Una prueba de rendimiento de una computadora, dispositivo o software.
<b>Bit</b>	Digito binario que es la unidad de almacenamiento más pequeña utilizada en un sistema de cómputo, cada uno posee dos estados, 0 o 1.
<b>Bus</b>	Canal o ruta común entre dispositivos de hardware.
<b>Byte</b>	Unidad de almacenamiento de información en una computadora 8 bits.

---

<b>Canal de Comunicaciones</b>	Una ruta de transmisión de datos entre dos dispositivos remotos. Este puede ser físico como un cable o una frecuencia de comunicación.
<b>CASE (Computer Aided Software Engineering)</b>	Herramienta de software que automatiza cualquier fase de desarrollo de un sistema de información.
<b>Cliente</b>	Un módulo de software que hace un requerimiento de servicio. O una computadora que actúa como un cliente en una arquitectura cliente/servidor.
<b>Computación Paralela</b>	Una computadora individual que utiliza diversos procesadores trabajando al mismo tiempo o múltiples computadoras trabajando juntas.
<b>Comunicación Punto a Punto (Peer-to-peer)</b>	Comunicación en la cual cualquier computadora puede iniciar una sesión.
<b>Comunicación Remota</b>	Interconexión de computadoras a grandes distancias.
<b>Concentrador</b>	Dispositivo que reúne diversos canales de comunicación.
<b>Configuración</b>	Es la suma de componentes de hardware y software que hacen que funcione un sistema.

<b>CSMA/CD (Carrier Sense Multiple Access/ Collision Detection)</b>	Método de acceso de comunicación que usa una técnica de detección de colisión.
<b>DBMS (Sistema Manejador de Base de Datos)</b>	Software que controla la organización, almacenamiento, recuperación, seguridad e integridad de datos en una base de datos.
<b>DDE (Dynamic Data Exchange)</b>	Protocolo de liga de Microsoft Windows que permite la comunicación entre programas de aplicación usando un modelo cliente/servidor.
<b>Desarrollador de Aplicaciones</b>	Se refiere a la persona que crea una aplicación.
<b>DLL (Dynamically Linked Library)</b>	Librerías que permiten ligar aplicaciones. El mismo código puede ser compartido por varias tareas en lugar de que cada tarea contenga copias de las librerías en uso.
<b>Downsizing</b>	Conversión de un sistema que corre en un mainframe a una computadora de menor tamaño mas no capacidad.
<b>Engine</b>	Software que realiza rutinas o funciones específicas.
<b>Escalabilidad</b>	Capacidad de extender un programa en tamaño y/o configuración con un

---

	cambio mínimo.
<b>Ethernet</b>	Estándar de red IEEE 802.3 que utiliza el método de acceso CSMA/CD.
<b>FDDI (Fiber optic Data Distribution Interface)</b>	Es un estándar de ANSI para el paso del token a 100 Mbps en una red local que usa fibra óptica.
<b>Fibra Óptica</b>	Sistema de comunicación a través del cual se transmiten señales luminosas.
<b>Frame (Trama)</b>	En comunicaciones, es un grupo de bits que conforman un bloque elemental de datos para su transmisión por ciertos protocolos.
<b>Frame Relay</b>	Protocolo de alta velocidad que proporciona transmisiones de datos, imágenes y voz.
<b>FTP (Protocolo de Transferencia de Archivos)</b>	Convención para transmitir archivos sobre una red entre programas o entre computadoras sin pérdida de datos.
<b>Full-duplex</b>	Transmisión y recepción simultánea de datos.
<b>Gateway</b>	Un nodo sobre una red que conecta dos redes incompatibles vía un protocolo de conversión.
<b>Groupware</b>	Software que está diseñado para ser usado en grupo típicamente conectado

**GUI (Interface Gráfica de Usuario)**

por una red.

Un despliegue gráfico que típicamente reside en un cliente de una arquitectura cliente/servidor y facilita el uso de la aplicación para el usuario final.

**Half-duplex**

La capacidad de enviar y recibir datos, pero solo en una dirección a la vez.

**Hardware**

Parte física de los equipos de cómputo.

**Herramientas de Desarrollo**

Un conjunto de rutinas de software usadas para una aplicación, sistema operativo o interface.

**Host-based**

Un sistema de comunicaciones que está controlado por una computadora central comúnmente un mainframe.

**Hub**

Un punto céntrico en una red donde diversos circuitos se conectan.

**Icono**

Pequeña representación pictórica de un objeto.

**IEEE (Instituto de Ingeniería Eléctrica y Electrónica)**

Esta asociación patrocina una gran cantidad de publicaciones simposios y esfuerzos en el desarrollo de estándares eléctricos.

**Integridad de Datos**

Datos incorruptibles.

**Interface Orientad a Objetos**

Una interface gráfica de usuarios que

---

	usa iconos para representar elementos.
<b>Internet</b>	La red más grande del mundo, con más de ocho mil redes conectadas.
<b>ISDN (Integrated Services Digital Network)</b>	Estándar de telecomunicaciones para transmitir video, voz y datos sobre una línea digital de comunicaciones.
<b>ISO (International Standards Organizatio)</b>	Organización que conjunta estándares internacionales, excepto eléctrico y electrónicos.
<b>Kernel</b>	Núcleo de un sistema operativo o programa.
<b>LAN (Local Area Network)</b>	Red de área local. Es un sistema de comunicación de datos que permite a un número independiente de dispositivos conectarse directamente con un área geográfica moderada sobre un canal de comunicación físico.
<b>Lenguaje de Cuarta Generación (4GL)</b>	Un lenguaje de computadora con comandos parecidos al inglés.
<b>MAN (Metropolitan Area Network)</b>	Red de área metropolitana. Es un red de comunicaciones para un área geográfica específica tal como una ciudad o un país.
<b>Mensaje</b>	Datos y bits de control que son transmitidos sobre una línea de

<b>Menú</b>	comunicaciones.
<b>Método</b>	Lista de opciones en una aplicación.  En programación orientada a objetos es una operación que manipula las instancias de una clase. Un método es invocado para enviar un mensaje a la instancia.
<b>Mouse</b>	Un dispositivo usado para ejecutar acciones sobre iconos y menús en un ambiente gráfico.
<b>Multiprocesamiento Asimétrico</b>	Dispositivos que están físicamente conectados a una sola unidad de procesamiento, cada CPU está generalmente dedicado a una función específica.
<b>Multithreading</b>	Sistema multihilos que corre diversos procesos en un programa.
<b>Nodo</b>	Un punto de conexión en una red.
<b>NOS (Sistema Operativo de Red)</b>	Sistema operativo en una red que responde a requerimiento de múltiples estaciones de trabajo.
<b>ODBC (Open Data Base Connectivity)</b>	Es una interfaz de conexión de bases de datos.
<b>OLE (Object Linking and Embedding)</b>	Protocolo que permite a un editor insertar parte de un documento a otro

---

<b>Operación Asíncrona</b>	editor. Eventos no sincronizados. Eventos que no actúan al mismo tiempo.
<b>OSI (Open System Interconnection)</b>	Modelo de sistema de interconexión abierto que contiene siete niveles estándares de la organización internacional de estándares (ISO)
<b>Outsourcing</b>	Término que se refiere a cuando los sistemas de datos son encargados a un consultor externo a la empresa.
<b>Plataforma</b>	Arquitectura de hardware y software de una computadora.
<b>Procedimiento</b>	Subrutina o función de una aplicación.
<b>Procesamiento Concurrente</b>	Es el procesamiento de dos o más programas a la vez, sin embargo no es simultáneo.
<b>Procesamiento Distribuido</b>	Un número de elementos de procesamiento autónomos interconectados por una red, que cooperan en la realización de sus tareas independientes.
<b>Procesamiento Paralelo</b>	División de una tarea entre diversos procesadores trabajando simultáneamente en una computadora.

<b>Programación Orientada a Objetos (OOP)</b>	Un método de programación que utiliza una colección de objetos que interactúan con otros objetos.
<b>Protocolo</b>	Es un acuerdo en comunicaciones por medio del cual se rige la transmisión y recepción de datos.
<b>Protocolo Asíncrono</b>	Protocolo de comunicación que controla la transmisión de forma asíncrona.
<b>Prototyping</b>	Un modelo de construcción de aplicaciones de software que puede ser probado y refinado durante el desarrollo.
<b>Puerto</b>	Es una conexión a la cual se enlazan otros dispositivos.
<b>Query</b>	Consulta o requerimiento de información a una base de datos.
<b>Red de anillo</b>	Topología de red que conecta terminales y computadoras en un anillo cerrado.
<b>Red Punto a Punto (Peer-to-peer)</b>	Red de área local que permite acceso igual a datos o dispositivos sobre todas las computadoras.
<b>Repetidor</b>	Un dispositivo que amplía la distancia de transmisión amplificando y

---

<b>Reutilización</b>	regenerando la señal.  La capacidad de usar todo o parte del mismo código de programación en otra aplicación.
<b>RISC (Reduced Instruction Set Computing)</b>	Es una arquitectura de procesador diseñada para procesar instrucciones rápidamente.
<b>Rollback</b>	Un procedimiento de un DBMS que retorna a la base de datos a un estado previo si una transacción es interrumpida o falla completamente.
<b>RPC (Llamadas de Procedimiento Remotas)</b>	Una forma de comunicación que permite a un procedimiento hacer una llamada a otro procedimiento que reside y puede ser ejecutado en un lugar remoto.
<b>Runtime</b>	Tiempo de ejecución de un programa.
<b>Ruteador</b>	Un dispositivo en una red que selecciona la mejor ruta entre dos puntos.
<b>Script</b>	Conjunto de instrucciones y/o comandos cuyo fin es automatizar una acción.
<b>Servidor</b>	Una computadora en una red que controla los recursos, múltiples

	<p>usuarios comparten un servidor .En una arquitectura cliente/servidor, el servidor realiza el proceso de los requerimientos del cliente y le transmite los resultados.</p>
<b>Servidor de Archivos</b>	<p>Computadora servidor en una red local que proporciona a los usuarios el servicio de compartir archivos.</p>
<b>Servidor de Base de Datos</b>	<p>Una computadora especialmente diseñada para base de datos que frecuentemente usa múltiples procesadores.</p>
<b>Sesión</b>	<p>Una comunicación activa entre un usuario y una computadora o entre dos computadoras.</p>
<b>Shell</b>	<p>Un software que tiene la capacidad de facilitar la comunicación entre el usuario y el sistema operativo.</p>
<b>Sistema de Multiprocesamiento</b>	<p>Un sistema diseñado para permitir la ejecución de múltiples programas simultáneamente.</p>
<b>Sistema Multitareas</b>	<p>Sistema diseñado para permitir la ejecución de múltiples programas casi simultáneamente.</p>
<b>Sistema Operativo</b>	<p>Programa que administra y controla los</p>

---

	requerimientos en una computadora.
<b>Sistemas Abiertos</b>	Estándares de hardware o software que pueden trabajar con productos de múltiples fabricantes.
<b>SMP (Multiprocesamiento Simétrico)</b>	Multiprocesamiento en el cual un CPU realiza una gran variedad de tareas. En sistema SMP los CPUs comparten memoria.
<b>SMTP (Simple Mail Transfer Protocol)</b>	Protocolo principal de correo electrónico usado en redes TCP/IP
<b>SNA (Systems Network Architecture)</b>	Protocolo de transmisión de datos de IBM.
<b>SNMP (Simple Network Management Protocol)</b>	Principal herramienta de administración de red con TCP/IP.
<b>Sockets</b>	Puerto de comunicación.
<b>Software</b>	Una serie de instrucciones que realizan una tarea en particular en un sistema de cómputo.
<b>SQL (Strutired Query Language)</b>	Lenguaje de consulta estructurado. Sirve para consultar, actualizar y administrar los datos en una base de datos relacional.
<b>SQL Dinámico</b>	Instrucciones de SQL que son interpretadas por la base de datos SQL

	en tiempo de ejecución.
<b>T1</b>	Multicanal que trabaja a 1544 Mbps usado primordialmente en comunicaciones amplias.
<b>T2</b>	Equivalente a 4 T1s
<b>T3</b>	Equivalente a 28 T1s
<b>TCP/IP</b> (Transmission Protocol/Internet Protocol)	<b>Control</b> Protocolo desarrollado para la red ARPANET.
<b>Tecnología Orientada a Objetos</b>	Paradigma que ve al mundo como objetos más que como procediniento.
<b>Transmisión Asíncrona</b>	Transmisión de datos donde la información es enviada de un carácter a la vez.
<b>Transmisión de Datos</b>	El envío de datos sobre una red.
<b>Transmisión Paralela</b>	Transmisión de bits simultáneamente.
<b>Transmisión Sincrona</b>	Transmisión de datos en el cual ambas computadoras se sincronizan
<b>UNIX</b>	Sistema operativo multiusuario y multitareas creado por los laboratorios Bell (AT&T).
<b>Usuario Final</b>	Comúnmente llamada la persona que utiliza los sistemas de computadora en una organización.

**Ventana Activa**

La ventana que puede ser afectada por acciones del usuario, en un ambiente donde múltiples ventanas pueden ser desplegadas.

**WAN (Wide Area Network)**

Una red que cubre una área geográfica grande que atraviesa estados y países.