



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

SISTEMA INTEGRAL DE LA FACULTAD  
DE DERECHO, MODULO III: SISTEMA DE  
INFORMACION ACADEMICA PARA LOS  
ALUMNOS.

T E S I S

QUE PARA OBTENER EL TITULO DE:  
INGENIERO EN COMPUTACION

P R E S E N T A N:

MEDRANO PEINADO BEATRIZ ADRIANA  
SERRANO SOLORZANO VERONICA

DIRECTOR DE TESIS:

ING. GABRIEL CASTILLO HERNANDEZ

282498



MEXICO, D. F.

SEPTIEMBRE 2000



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.

*Beatriz Adriana Medrano Peinado.*

A DIOS:

*Doy gracias por darme la oportunidad de vivir grandes momentos, uno de los cuales es poder compartir este trabajo con mis seres queridos y por aquellos que te ayudan a ser mejor cada vez.*

A MIS PADRES JOSE E IRENE:

*Por el apoyo que me han dado en mi etapa de estudiante, por la educación y valores que me han enseñado, los cuales me han permitido llegar a este momento de mi vida.*

*Gracias.*

A MI HERMANA LAURA:

*Por todos aquellos momentos que hemos compartido juntas, alegrías y por que no tristezas, los cuales me dan animos para seguir adelante.*

*Gracias por ser una gran hermana.*

A MI AMIGA VERONICA:

*Por permitirme realizar este trabajo juntas y por ser una gran amiga.*

Y a todas aquellas personas:

*Que siempre me han demostrado su apoyo incondicionalmente.*

*Veronica Serrano Sobrzano*

*Primeramente agradezco a Dios el don de la vida, y por haberme permitido alcanzar esta meta.*

*A mis padres, gracias por la confianza y el apoyo incondicional que siempre me han demostrado.*

*A mis hermanos, por el apoyo y respaldo que siempre me han brindaron*

*A mi abuelo por la confianza y la presión que ejerció sobre mi para no desfallecer en el último momento.*

*Y a todos mis familiares y amigos que han estado conmigo y me han brindado apoyo, tanto en las buenas como en las malas.*

*A todos Gracias.*

**A LA FACULTAD DE INGENIERÍA:**

*Por todo lo que nos ha dado: conocimientos, una formación como profesionistas, triunfos, fracasos, anécdotas.*

**AL CENTRO DE COMPUTO DE LA FACULTAD DE DERECHO:**

*Al personal que en su momento nos brindo apoyo para el desarrollo del presente trabajo. Gracias por compartir sus conocimientos con nosotras.*

**AL ING. GABRIEL CASTILLO HERNANDEZ:**

*Por la confianza y el apoyo nos ha dado en el desarrollo de este trabajo.*

*Gracias.*

## Tabla de contenido.

INTRODUCCION .....	1
<b>CAPITULO I</b> .....	<b>1</b>
ANALISIS .....	1
PLANTEAMIENTO DEL PROBLEMA .....	1
AREAS DE LA INSTITUCION .....	3
RECURSOS DISPONIBLES .....	8
OBJETIVO DEL PROYECTO .....	9
<b>CAPITULO II</b> .....	<b>13</b>
DISEÑO .....	13
PROPUESTA DE SOLUCION .....	13
DISEÑO .....	14
DISEÑO DEL MODULO DE INFORMACION ACADEMICA PARA LOS ALUMNOS .....	39
<b>CAPITULO III</b> .....	<b>45</b>
IMPLEMENTACION .....	45
MODELADO DE DATOS EN HERRAMIENTA CASE .....	45
GENERACION DE LA BASE DE DATOS EN SYBASE .....	48
MIGRACION DE DATOS DEL SISTEMA ANTERIOR AL NUEVO .....	50
DESARROLLO DE LA APLICACION EN DELPHI .....	51
<b>CAPITULO IV</b> .....	<b>59</b>
CONCLUSIONES .....	59
<b>APÉNDICE A</b> .....	<b>61</b>
METODOLOGÍA DE DESARROLLO DE SISTEMAS DE INFORMACIÓN .....	61
Ciclo de vida .....	62
Herramientas y Métodos .....	66
Análisis de sistemas .....	71
Análisis estructurado .....	72
Definiciones .....	76
<b>APENDICE B</b> .....	<b>79</b>
DISEÑO ORIENTADO A OBJETOS .....	79
Determinación de Clases .....	81
Determinación de Responsabilidades .....	82
Determinación de la Colaboración .....	83
<b>APENDICE C</b> .....	<b>84</b>
BASES DE DATOS .....	84
Arquitectura de una Base de Datos .....	85
Sistema de Bases de Datos .....	86
Diccionario de Datos .....	87
Conceptos .....	88
Modelos de bases de datos .....	89
Normalización .....	90

<b>APENDICE D</b> .....	<b>92</b>
<b>ARQUITECTURA CLIENTE/SERVIDOR</b> .....	<b>92</b>
<i>Estructura de los sistemas cliente/servidor</i> .....	92
<i>Componentes de software para sistemas C/S</i> .....	93
<i>Distribución de componentes del software</i> .....	94
<i>Líneas generales para distribuir los componentes</i> .....	95
<i>Diseño de bases de datos</i> .....	97
<b>APENDICE E</b> .....	<b>99</b>
<b>REDES</b> .....	<b>99</b>
<i>LAN</i> .....	100
<b>BIBLIOGRAFIA</b> .....	<b>113</b>

---

## INTRODUCCION

Durante las primeras décadas de la informática, se le daba mayor importancia al hardware, ya que el propósito era el procesamiento y almacenamiento de la información, con el tiempo esto fue cambiando dando mayor auge al desarrollo de software.

Dentro de las primeras etapas, el software se desarrollaba a la medida para cada aplicación. Su distribución era muy pequeña, ya que se llevaba a cabo dentro de la misma organización.

Otro punto importante, es la falta de documentación, esto con el paso del tiempo ha resultado ser imprescindible, puesto que al presentar un error o al querer realizar modificaciones a la aplicación, se tiene que recurrir al desarrollador del mismo. Pero ¿qué ocurre cuando el desarrollador se retira de la empresa?, resulta aún más difícil reparar el sistema.

En la actualidad el desarrollo de software permite crear a la par del análisis y el diseño la documentación del mismo.

La Facultad de Derecho carece de documentación de los sistemas con los que cuenta, encontrándose en el punto de querer realizar modificaciones sin contar con el personal que llevó a cabo los sistemas.

En el presente trabajo, el objetivo es analizar, diseñar, construir e implementar un sistema integral de información para la Facultad de Derecho de la UNAM.

Actualmente, la Facultad se apoya para diferentes procesos en sistemas desarrollados en Paradox, los cuales a su vez presentan problemas de diversa índole, complejidad en el mantenimiento y adecuación de cambios de los reglamentos de la UNAM, implicando todo esto un esfuerzo y costo de mantenimiento.

En el primer capítulo se presenta:

- El planteamiento del problema a resolver, considerando la forma en la que está estructurada la institución que demanda la solución.
- La recopilación de las necesidades de cada área de acuerdo a los procedimientos que se efectúan en cada una de ellas.
- Una descripción de los recursos de hardware y de herramientas de desarrollo disponibles.

En el segundo capítulo, se expone:

- El diseño de la base de datos.
- La división de los diferentes módulos que accedan como clientes al servidor.
- Se presenta, a manera de ejemplo el diseño del módulo: Sistema de Información Académica para los Alumnos.

En el tercer capítulo se presenta:

- La creación de la base de datos en el manejador Sybase.
- La implementación del módulo diseñado en el capítulo anterior.

Cabe mencionar que el diseño se apoya en una herramienta de desarrollo CASE ErWin (Computer Aided Software Engineering – Ingeniería de Software Asistida por Computadora). Permitiendo la documentación del sistema de manera paralela al diseño.

Por último en los apéndices se presentan los fundamentos teóricos más importantes sobre los cuales se sustenta la presente tesis. Se estudia la metodología de desarrollo de software, respecto a la ingeniería de software. Se presenta el ciclo de vida del software, el proceso de análisis y diseño estructurado, el enfoque orientado a objetos. Tecnología cliente/servidor. Se plantean los conceptos sobre bases de datos: arquitectura, modelos, el proceso de diseño y los manejadores de bases de datos. Conceptos principales de redes de computadoras, como arquitecturas de comunicación, el modelo OSI, y el protocolo TCP/IP.

## CAPITULO I

---

# ANALISIS

## PLANTEAMIENTO DEL PROBLEMA.

### **Problemática Actual.**

La Facultad de Derecho cuenta con alrededor de 8,000 alumnos, quienes están incorporados al plan de estudios de 38 materias (viejo plan) y de 76 materias (nuevo plan que entro en vigor a partir del semestre 91-1). Además de contar con 800 profesores, de los cuales 700 son de licenciatura, 40 de posgrado y cerca de 60 del sistema abierto (SUA).

Para llevar el control sobre alumnos, maestros y planes de estudio, se cuenta con varias secretarías que llevan a cabo el control de estos recursos, para ello se auxilian de pequeños sistemas aislados, los cuales, medianamente cumplen con las necesidades de estas secretarías. Los sistemas que actualmente se utilizan son los siguientes:

- Sistema de Inscripciones.
- Exámenes Profesionales.
- Horarios.

Dichos sistemas presentan las siguientes características:

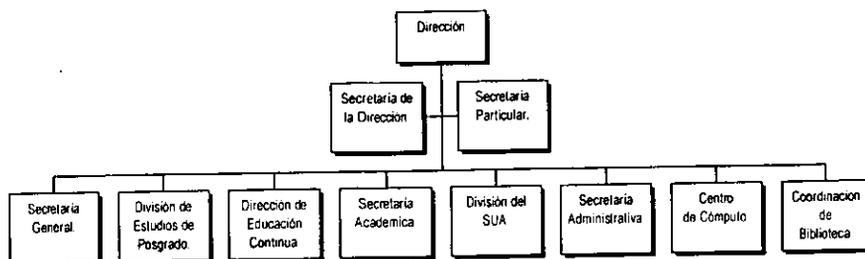
- Desarrollados en Paradox 3.5.
- Trabajan en un servidor Netware 3.x, bajo una plataforma de MS-DOS.
- No contemplan mecanismos de respaldo.
- No contemplan mecanismos de recuperación.
- Los programas se almacenan en Scripts interpretados por el run-time de Paradox.
- Cada uno localizado en una red local aislada, por lo que el manejo de la información es de forma independiente.
- No están diseñados para un ambiente en red.

Dentro de los problemas que se presentan referentes al manejo de información podemos citar los siguientes:

- Redundancia innecesaria. Cada aplicación trabaja con su propia base de datos. Debido a esto cada dato se encuentra almacenado varias veces provocando que los procesos de inserción, modificación o borrado tengan que realizarse en repetidas ocasiones, además de arrojar resultados erróneos en informes o estadísticas.
- Falta de Integridad. Surge como consecuencia de la redundancia, el realizar alguna modificación, inserción o borrado, se debe elaborar de forma repetida por cada base de datos con que se cuente.
- Datos compartidos de forma limitada. No se puede compartir información entre diferentes datos, ya que están instalados como unidades separadas.
- Inconsistencia de datos. Existe un desorden de datos, ya que existen varias versiones de los mismos.
- Documentación deficiente e inexistente. No existe documentación de los sistemas que se tienen actualmente. Por lo tanto resulta difícil entender la estructura del sistema.
- No tienen formalidad en la programación. Debido a modificaciones realizadas en el código, resulta complicado interpretar el mismo.

## AREAS DE LA INSTITUCION.

El organigrama de la Facultad permite entender cómo está organizada la institución donde residirá la solución. En este caso, el desarrollo se realiza en el centro de cómputo, quien es parte de la institución.



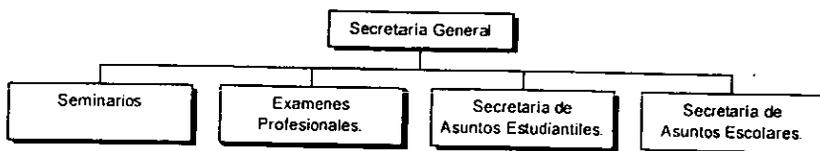
La Facultad de Derecho se compone de diferentes áreas para el desempeño de sus actividades, a continuación se describen las funciones y procedimientos de algunas de ellas, tomando el criterio que serán las que hagan uso del sistema.

### Secretaría General.

Tiene como objetivos principales:

- Que el personal académico del sistema escolarizado, cumpla con la labor docente asignada: ajustándose al plan de estudios, al reglamento actual, dentro del período escolar correspondiente.
- Que los profesores entreguen las evaluaciones de exámenes ordinarios, extraordinarios apegándose al período escolar correspondiente.
- Que el área de cómputo cuente oportunamente con las actas de calificación del personal académico, para que a su vez las transmita a la D.G.A.E., en el período escolar correspondiente.
- Que los alumnos realicen oportunamente sus trámites escolares, apegándose a los procedimientos y reglamentación vigentes dentro del período escolar correspondiente.
- Que los alumnos cuenten con los servicios de revisión, certificación de estudios y seguro facultativo; apegándose al procedimiento y reglamentación vigentes.
- Que los alumnos egresados de la facultad, realicen su examen profesional, una vez hayan cubierto los requisitos para ello y reciban el título correspondiente.

La Secretaria General coordina las siguientes áreas :



A continuación se presentan los objetivos y procedimientos de las áreas de Secretaria General.

### **Seminarios.**

Para el caso de los seminarios su objetivo en particular es que los alumnos reciban la asesoría para el desarrollo de su tesis. Además establece los siguientes procedimientos:

- Inscripción a un seminario.
- Asesoría de tesis.
- Tramite para examen profesional.

### **Exámenes profesionales.**

Tiene los siguientes objetivos:

- Que los alumnos egresados de la facultad, realicen su examen profesional una vez hayan hecho su solicitud y haber cubierto los requisitos requeridos para ello.
- Que los profesores entreguen las evaluaciones de exámenes ordinarios y extraordinarios de los alumnos de conformidad al calendario escolar.

### **Secretaría de Asuntos Estudiantiles.**

La secretaria de asuntos estudiantiles se encarga de:

- Asignación de aulas para examen profesional .
- Préstamo de material didáctico.
- Que el personal académico y estudiantil autorizado cuente con material didáctico y audiovisual de conformidad al procedimiento de préstamo.

### **Secretaría de Asuntos Escolares.**

La Secretaria de Asuntos Escolares se encarga de:

- Que los alumnos cuenten con la documentación certificada de sus estudios.
- Que los alumnos realicen su inscripción, ingreso y reinscripción de conformidad a procedimiento.
- Que los alumnos tengan su expediente escolar completo y actualizado.

- Que los alumnos cuenten con seguro facultativo.
- Control de actas.
- Calendario escolar.

Los procedimientos que corresponden a esta área son:

- Tramite de constancia de estudios. Para informar avance en créditos, promedio, horario, si se encuentra inscrito o último movimientos escolar, porcentaje de avance escolar.
- Tramite de carta a la dirección general de profesiones-SEP obtención cédula profesional. Se elaborara constancia de acreditación de estudios académicos a los alumnos de la facultad: para la autorización del ejercicio profesional temporal sin estar titulado por la dirección general de profesiones de la SEP.
- Revisión de estudios.
- Reinscripción viejo y nuevo plan.
- Cambios de grupo viejo y nuevo plan.
- Inscripción primer ingreso.
- Tramite de servicio social.
- Registro de exámenes extraordinarios.
- Asuntos estudiantiles.

### **Secretaría Académica.**

Los objetivos de la Secretaría Académica son:

- Que el personal académico de la facultad cuente con estímulos y servicios requeridos, para respaldar y mejorar su actividad docente.
- Que el personal académico reciba cursos para su formación y actualización docente.
- Que la facultad cuente con el personal académico, con base en los concursos de oposición y el dictamen de las comisiones dictaminadoras.
- Que las comisiones dictaminadoras decidan la definitividad del personal académico, con base en los concursos de oposición (de conformidad a la normatividad vigente).
- Que el personal académico reciba cursos para su formación y actualización docente.
- Que el personal académico reciba los estímulos a su ejercicio docente de conformidad a los programas previamente acordados.

Sus procedimientos son:

- Renovación y control de las comisiones dictaminadoras del personal académico.
- Verificar y mantener la vigencia de las comisiones dictaminadoras del personal académico, a efecto de que los profesores que los integran, ejerzan sus funciones; dentro del plazo que fueron designados, con apego a la normatividad universitaria.
- Concurso de oposición abierto.
- Concurso de oposición cerrado.
- Seguimiento de profesores de carrera.
- Obtener información actualizada de las actividades académicas de los profesores de carrera a fin de aplicar sus conocimientos y experiencias en los programas escolares de la facultad.
- Programa de primas al desempeño del personal académico de tiempo completo. (PRIDE).
- Curso de formación de profesores en nuevas asignaturas.
- Tramite de becas Nacionales e Internacionales.
- Programa de Estímulos académicos.
- Cursos de actualización de personal docente.

### **Secretaría Administrativa.**

Tiene como objetivos principales:

- Que todas las áreas de la facultad, cuenten con los recursos económicos, materiales, instalaciones y servicios de intendencia; para apoyar sus operaciones académico/administrativas, dentro de las restricciones presupuestales asignadas para el ejercicio vigente
- Que el personal académico y administrativo de la facultad de derecho tengan tramitados sus movimientos, en las fechas convenidas, de conformidad a procedimiento y normatividad vigentes.
- Que la facultad de derecho cuente oportunamente con un presupuesto anual para respaldar sus operaciones académico administrativas y de servicios.
- Que las diferentes áreas de la facultad cuenten con recursos económicos; para respaldar las operaciones que están dentro del reglamento.
- Que todas las áreas sustantivas de la facultad cuenten oportunamente con los insumos que soliciten de conformidad a requisición.
- Que el personal académico y administrativo de la facultad tenga tramitado su alto o su baja de conformidad a procedimientos en las fechas convenidas.
- Que el personal de la facultad, cuente con los servicios de intendencia, de mantenimiento y construcción de instalaciones.

Procedimientos a seguir:

- Registro y control de movimientos de personal académico de asignaturas.
- Nombramientos personal académico de asignatura nuevo ingreso.
- Nombramientos de asignatura por reanudación, prórroga, aumento o disminución de horas.
- Nombramiento de asignatura por reingreso.
- Baja de nombramientos.

### **Centro de cómputo.**

Tiene los siguientes objetivos.

- Que las áreas administrativas y académicas de la facultad tengan automatizados sus sistemas de trabajo; y están comunicadas en la red interna de cómputo, en la red de la universidad y con otras instituciones nacionales y extranjeras.
- Que la comunidad académica, administrativa, y estudiantil de la facultad, cuenten con cursos de capacitación para el conocimiento de los sistemas de cómputo, de instalaciones y técnicas
- Que los sistemas de información de la facultad cuenten oportunamente con el soporte técnico y de mantenimiento de su equipo de cómputo.
- Que las áreas administrativas y académicas de la facultad, cuenten con los servicios de cómputo para apoyar sus operaciones y servicios que dan a la comunidad.

Procedimientos.

- Desarrollo de los sistemas que sean requeridos.
- Asesoría a las diferentes áreas de la facultad.
- Área de atención y capacitación a usuarios.
- Cursos de cómputo.

## RECURSOS DISPONIBLES.

Para el sistema que se desarrollará, se cuenta con hardware y software, cuyas características se presentan a continuación:

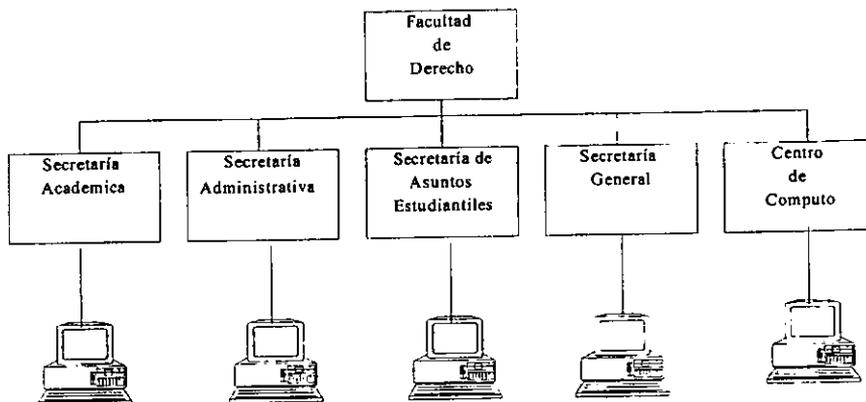
Cantidad Aprox.	Hardware / Software	Características
400	PC	Compaq con 1.2GB en Disco duro y 16 Mb en RAM.
1	Ultra Sparc	A 1476 Mhz, 64 MB en RAM , 7 Gb en Disco duro. Tlatoani almacena la información más importante de la facultad.
1	Super Sparc	Con 2 procesadores a 50 Mhz 128 MB en RAM, 1 Gb en disco Duro. Themis brinda servicios de Internet (correo, web, FTP, telnet, etc.)
	Red	Protocolo de transmisión TCP/IP. Protocolo de acceso al medio CSMA/CD Ethernet. Topología estrella. Medios de transmisión fibra óptica y par trenzado.
	Sistema Operativo: DOS v6.22, Solaris 2.5, Windows '95, '98	
	Sybase System	Manejador de Base de datos, con capacidad de 32 conexiones.
	Erwin	Herramienta Case.
	Delphi 3, Power Builder, Paradox,	Herramientas de desarrollo.
	Office 6, '95, '97	

Cabe mencionar que de las 400 computadoras, aproximadamente el 80% se encuentran conectadas en red y el 95% cuentan windows '95 o '98.

## OBJETIVO DEL PROYECTO.

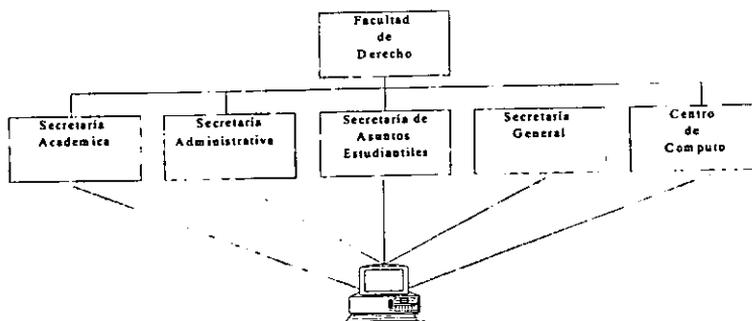
Analizando los recursos y necesidades de la Facultad, se propone a través de esta tesis, integrar todas las áreas, mediante la creación de una única base de datos. Permitiendo que se incorpore toda la información y que a su vez las diversas áreas tengan acceso a la base de datos, dando lugar a un manejo de información consistente y confiable

En la siguiente figura se observa a manera de ejemplo la estructura de los sistemas actuales y la propuesta de integrar las diferentes áreas de la facultad.



*Sistemas por área.*

Es decir cada secretaria cuenta con una aplicación y tantas bases de datos como aplicaciones por secretaria se tengan, esto es debido a que cada usuario captura la información que es utilizada dentro de la secretaria



*Integración de todas las áreas.*

Esto es, cada secretaria contará con una aplicación para poder acceder a la única base de datos, localizada en una máquina que funcione como servidor.

Lo anterior será posible debido a que existe una amplia relación entre las diversas áreas que conforman la facultad, es decir, la información concerniente a los profesores de la facultad, es manejada por: la Secretaría Administrativa, la Secretaría General y la Secretaría Académica, de la misma manera, la información relacionada con los alumnos involucra a la Secretaría de Asuntos Estudiantiles, Seminarios, Exámenes profesionales, Secretaría de Asuntos Escolares, Biblioteca servicio al público y Secretaría General.

Una vez unificada la información, se requieren aplicaciones mediante las cuales, las secretarías puedan tener acceso a los datos.

El realizar un solo sistema para acceder a la información es complicado y bastante extenso, además de que cada secretaria solo debe tener acceso a cierta parte de la información, por ello se propone, dividir la aplicación en pequeños módulos los cuales se listan a continuación:

- Reinscripciones a cursos ordinarios (Inscripciones)
- Exámenes profesionales
- Asistencia de profesores
- Directorio de profesores
- Horarios
- Exámenes de oposición
- Información académica para los alumnos

El sistema al dividirse en módulos permite:

- Integrar la información que se maneja en las diversas áreas de la facultad.
- Reestructurar la base de datos para que sea usada por las diversas áreas de la institución.
- Actualizar los sistemas existentes, utilizando herramientas visuales.
- Aumentar la seguridad, funcionalidad e integridad de los datos.
- Documentación de cada módulo desarrollado, para facilitar el mantenimiento.
- Implementar cada uno de los módulos que se desarrollen.

El proyecto requiere de herramientas que 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 mismos. Así como el sincronizar el acceso a múltiples aplicaciones.

A manera de ejemplo la presente tesis describirá el análisis, diseño e implementación del módulo "Sistema de Información académica para los alumnos", cuyo objetivo es que los alumnos tengan acceso a su información académica de manera sencilla, ya que hasta ahora debido al crecimiento de

la población estudiantil el proceso resulta lento. Es necesario que los alumnos puedan consultar su historia académica y horarios de inscripción de forma sencilla. Además de emitir constancias de forma automática como beneficio de los alumnos y de la secretaría de asuntos estudiantiles.

Por ello, se requiere de un sistema capaz y eficiente para poder llevar a cabo las tareas del proceso de administración de alumnos.

---

## CAPITULO II

# DISEÑO

### PROPUESTA DE SOLUCION.

De acuerdo a los requerimientos del sistema, la solución que se plantea a continuación es la que cubre las demandas de la Facultad de Derecho, para los problemas que presenta actualmente.

Una vez analizados los requerimientos y tomando en cuenta los recursos con los que cuenta la facultad se decidió centralizar la información para eliminar la redundancia así como la inconsistencia de los datos, creando una sola base de datos, que contenga toda la información relacionada con los alumnos y profesores.

En cuanto a los alumnos se refiere, no solo los datos generales son necesarios, sino también se requiere información sobre su situación académica (seguimiento de los 10 semestres de la carrera), así como el seguimiento de trámites de titulación.

Por parte de los profesores, los datos personales son indispensables, además del seguimiento sobre el trabajo que realizan en el plantel (tipo de profesor, cargos, etc.).

El diseñar la base de datos es fundamental dentro de la solución, ya que actualmente la facultad no cuenta con una base de datos sólida, que garantice el intercambio de información sin redundancia de la misma, solo cuenta con pequeñas bases de datos sobre las cuales trabajan algunas dependencias de la facultad, por tanto la información es monopolizada por estas, y el intercambio de información representa grandes conflictos.

Con el diseño de la base de datos se busca un tener un mejor control y manejo de la información en toda la facultad, además de contribuir con el centro de computo del plantel quien a partir de esta base, generará diversos modulo o aplicaciones, los cuales son requeridos dentro de la Institución.

Una vez analizada qué información es necesaria, la base será diseñada y creada para que sea administrada por un manejador de base de datos que sea robusto y sobre todo que garantice la seguridad e integridad de la información.

Por lo antes mencionado y partiendo de los recursos con los que cuenta la Facultad, se hará uso del DBMS Sybase que presenta las características de robustez, además de garantizar seguridad e integridad.

Como parte complementaria de la presente tesis, se diseñará e implementará la aplicación denominada Sistema de Información Académica para los alumnos, la cual pretende satisfacer la necesidad que tienen los alumnos de mantenerse informados sobre su situación académica, mediante el cual tengan acceso a su horario, día y hora de inscripción, historial académico, así como la impresión de constancias, además del envío y recepción de mensajes cortos. Este sistema se realizará utilizando una arquitectura cliente-servidor, cuyo cliente sea amigable utilizando un lenguaje orientado a objetos y el servidor estará en SYBASE.

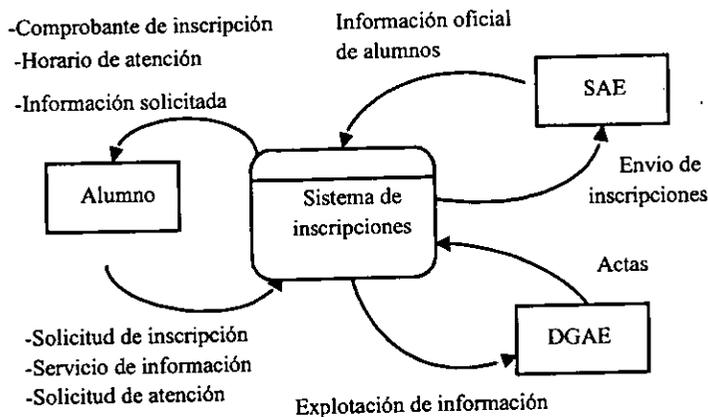
## DISEÑO.

Una vez establecido los requerimientos de la facultad, con el análisis previo, se debe identificar el ámbito en el cual trabajará la base de datos, para ello se hace uso de los diagramas de contexto que permiten percibir la relación entre el sistema y agentes externos.

A continuación se presentan los diagramas de contexto de las aplicaciones que se realizarán a partir de la base de datos.

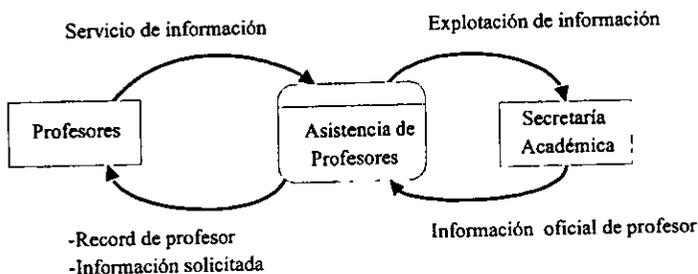
### Diagramas de Contexto.

El diagrama muestra los principales eventos que acontecen al momento de realizar una inscripción, las entidades y dependencias que participan.



*Diagrama de Contexto del Sistema de Inscripciones*

La principal beneficiaria al realizar los siguientes módulos será la Secretaría Académica debido a que es la encargada de llevar el control y darle seguimiento al trabajo desarrollado por los profesores.



*Diagrama de Contexto del Sistema de Asistencia de profesores*

En el modulo de asistencia de profesores se llevará un control sobre el rendimiento de los profesores, permitiendo obtener estadísticas.

En el siguiente modulo llevará el control del proceso de selección del personal académico.

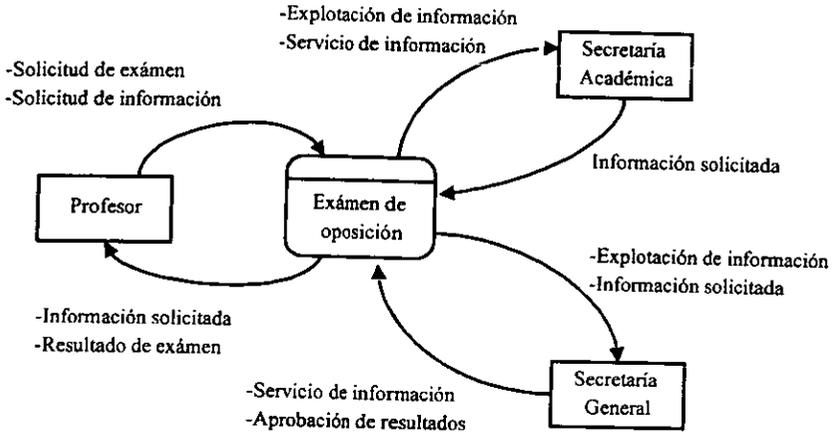


Diagrama de Contexto del Sistema de Exámenes de oposición

El modulo del directorio de profesores, permitirá que las Secretarías Académica y General tengan un fácil acceso a este tipo de información.

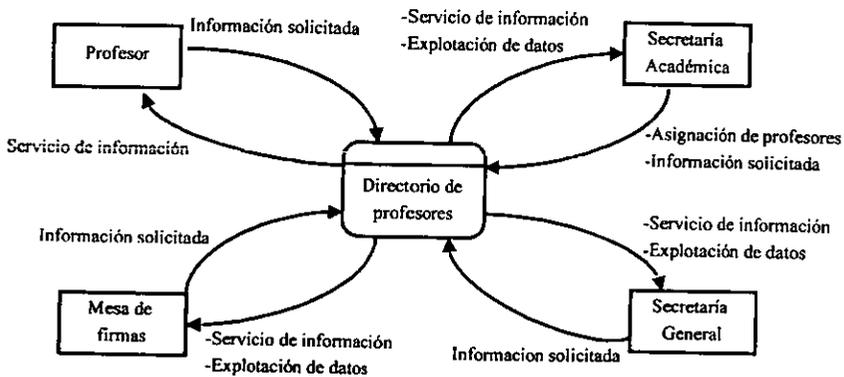
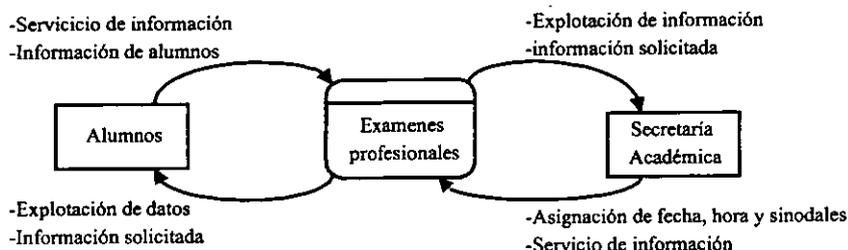


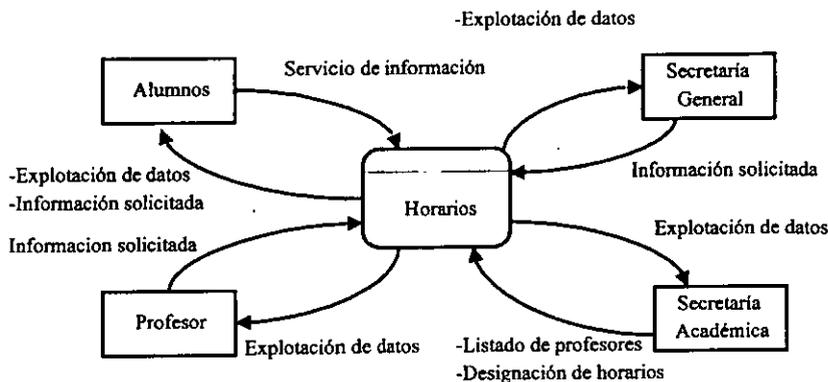
Diagrama de Contexto del Sistema de Directorio de profesores

El modulo de exámenes profesionales permitirá que cada uno de los seminarios registre al alumno con su respectivo tema de tesis, de esta forma la Secretaría Académica compartirá la información



*Diagrama de Contexto del Sistema de Exámenes Profesionales*

Es importante automatizar la generación de los horarios de inscripciones, es por ello que el modulo de horarios, se encargará de llevar a cabo esta tarea, apegándose a los lineamientos establecidos.

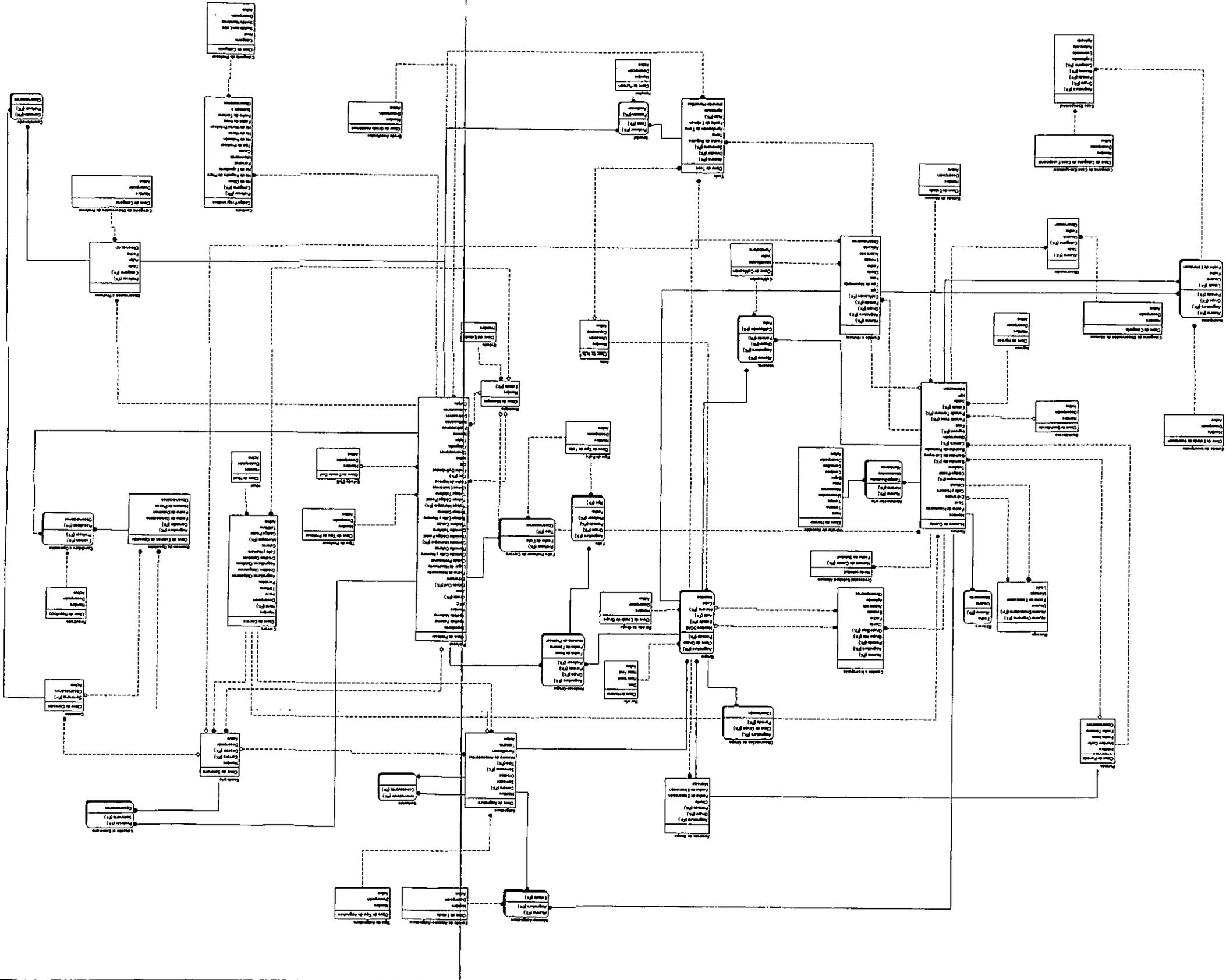


*Diagrama de Contexto del Sistema de Horarios*

Una vez establecido el ámbito en el que trabajaran los sistemas se deben identificar las entidades, así como la relación entre ellas, cardinalidad, reglas de validación, atributos, tipo de datos, índices y llaves primarias y foráneas, dando como resultado el diagrama de entidad relación.

Una vez obtenido el diagrama de entidad relación se aplicaron la 1FN, 2FN y 3FN, es decir el proceso de normalización, dando como resultado, el diseño de la base de datos normalizada.

A continuación se presenta el diagrama de Entidad-Relación de la base de datos, diseñada en la herramienta CASE llamada Erwin, esta herramienta permite el modelado de los datos tanto físicos como lógicos de la base de datos, además permite establecer los campos llaves e índices así como la integridad referencial entre entidades de la base.



**Diagramas Entidad – Relación.**

A continuación se presenta la división de la base de datos en módulos, los cuales contribuirán para la realización de las aplicaciones requeridas.

Con la realización de este modulo de inscripciones se pretende tener las siguientes ventajas:

- Actualizar el sistema de inscripciones para mejorar la velocidad de respuesta agilizando así el periodo de inscripciones.
- Además se requiere aumentar la integridad y seguridad de los datos.
- El llevar un estricto control sobre la relación alumno-profesor y asignatura-seriación, es algo fundamental.
- Además de mejorar la interfaz gráfica

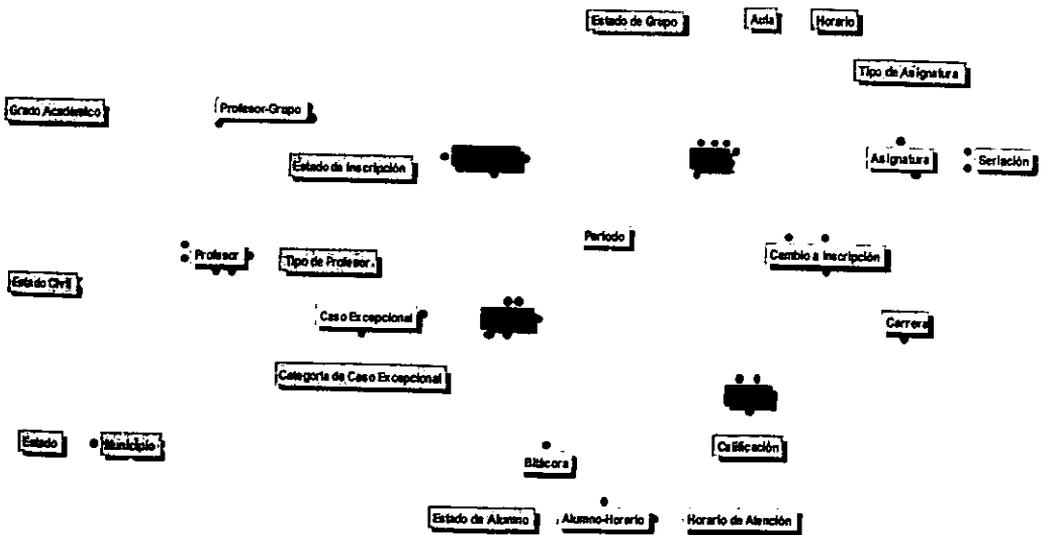
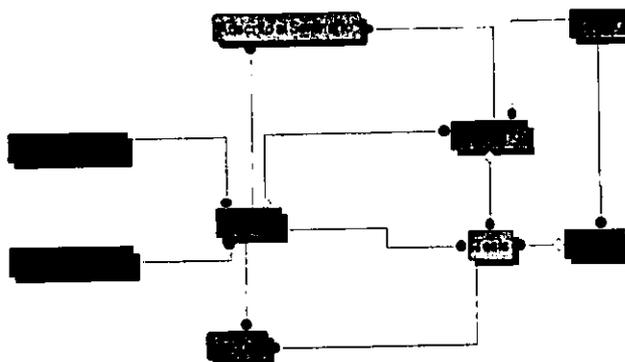


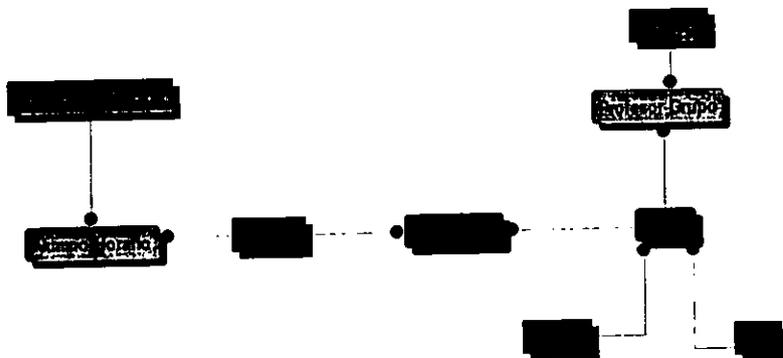
Diagrama Entidad - Relación de Inscripciones.

Con el modulo de Asistencia de Profesores, se llevará el control sobre el rendimiento del mismo, así como mantener un estricto seguimiento sobre el trabajo que desarrolla cada académico.



*Diagrama Entidad - Relación de Asistencia de Profesores.*

El contar con una fuente de datos para los posibles candidatos a ocupar las plazas de profesores, es algo importante, es por ello que se realizará el modulo que tendrá un amplio control sobre esta información.



*Diagrama Entidad - Relación de Exámenes de Oposición.*

Uno de los propósitos de unificar la información es el intercambio de datos entre las diversas dependencias, así como automatizar algunos de los procesos que cada una de estas realiza, es por ello que el modulo de Exámenes Profesionales será de gran ayuda para la Secretaría Escolar, debido a que podrá llevar un buen control sobre los alumnos en vías de titulación así como el generar información sobre la tendencia (seminarios) de las ramas del Derecho.

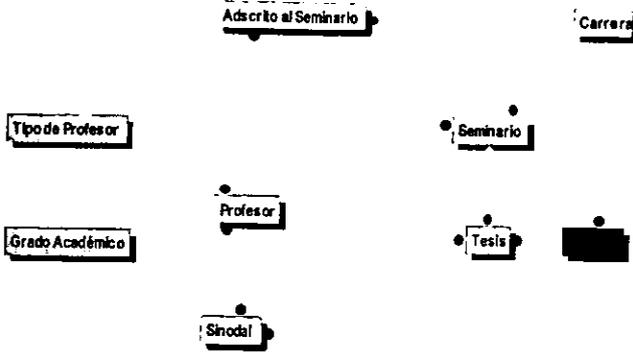


Diagrama Entidad - Relación de Exámenes Profesionales.

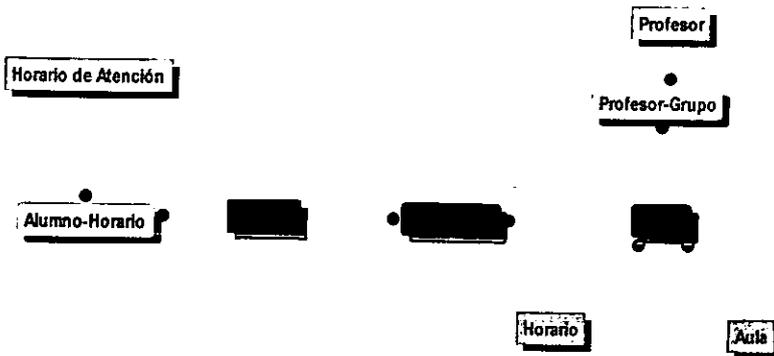


Diagrama Entidad - Relación de Horarios.

Un punto importante a la par del diseño, es la creación del diccionario de datos, el cual permite la documentar la base, definiendo nombre de las entidades, nombre de atributos, indicación de llaves, nulidad, tipo de dato, descripciones, etcétera.

Con la ayuda de Erwin se realizó el diseño de la base de datos, además esta misma herramienta permitió la creación del diccionario de datos, el cual se presenta a continuación:

### Diccionario de datos.

Entidad	Atributo	Llave	Nulidad	Tipo de dato	Descripción del campo
	Bachillerato Extranjero		NOT NULL	Bit	Indicador de que el alumno estudio el bachillerato en el extranjero.
	Bachillerato Promedio		NULL	Numeric(4,2)	Promedio que el alumno obtuvo en el bachillerato.
	Código Postal		NULL	Numeric(5)	Código postal del domicilio del alumno.
	Calle y Número		NULL	Varchar(255)	Calle y número del domicilio del alumno.
	Clave de Bachillerato	(FK)	NULL	Tinyint	Identificador de donde el alumno curso el bachillerato.
	Clave de Carrera	(FK)	NOT NULL	Tinyint	Identificador de la carrera que cursa el alumno.
	Clave de Estado	(FK)	NOT NULL	Tinyint	Identificado del estado donde se encuentra el domicilio del alumno.
	Clave de Ingreso	(FK)	NOT NULL	Tinyint	Identificador del tipo de ingreso del alumno a la licenciatura.
	Clave de Municipio	(FK)	NULL	Tinyint	Identificador del municipio donde se encuentra el domicilio del alumno
	Clave de Periodo	(FK)	NOT NULL	Smallint	Identificador del periodo que esta cursando actualmente el alumno
	Colonia		NULL	Varchar(255)	Colonia del domicilio del alumno.
	Extranjero		NOT NULL	Bit	Indicador de que el alumno es extranjero.
	Fecha de nacimiento		NULL	Smalldatetime	Fecha de nacimiento del alumno.
	Folio		NULL	Int	Folio que solicita DGAE.
	Generación		NOT NULL	Tinyint	Generación a la que pertenece el alumno.

Entidad	Atributo	Llave	Nulidad	Tipo de dato	Descripción del campo
	Información		NULL	Text	Información sobre el alumno en particular.
	Número de cuenta	(PK)	NOT NULL	Int	Número de cuenta del alumno.
	NIP		NULL	Int	Número de Identificación Personal del alumno.
	Nombre	(IE1)	NOT NULL	Varchar(64)	Nombre completo del alumno. Apellido paterno - Apellido materno - Nombre (s)
	Saldo		NOT NULL	Tinyint	Saldo disponible del alumno.
	Sexo		NOT NULL	Bit	Indicador del sexo del alumno.
	Teléfono		NULL	Int	Teléfono del domicilio del alumno.
Asignatura	Acreditación		NOT NULL	Bit	
	Activo		NOT NULL	Bit	Indicador de que la asignatura esta activa.
	Clave de Asignatura	(PK)	NOT NULL	Smallint	Identificador de la asignatura.
	Clave de Carrera	(IE2) (FK)	NOT NULL	Tinyint	Identificador de la carrera que cursa el alumno.
	Clave de Seminario	(FK)	NULL	Tinyint	Identificador del seminario al que pertenece la asignatura.
	Clave Tipo Asignatura	(FK)	NOT NULL	Tinyint	Indicador de tipo de asignatura.
	Créditos		NULL	Tinyint	Créditos que se obtienen al cursar la asignatura.
	Antecedentes		NULL	Tinyint	Asignaturas antecedentes.
	Nombre	(IE1)	NOT NULL	Varchar(C64)	Nombre de la asignatura
	Semestre	(IE2)	NOT NULL	Tinyint	Semestre al que pertenece.
Carrera	Temario		NULL	Text	Temas que se estudian en la asignatura.
	Activo		NOT NULL	Bit	Indicador de que la carrera se encuentra activa.
	Asignaturas obligatorias		NOT NULL	Tinyint	Asignaturas que son obligatorias para la carrera.
	Asignaturas optativas		NOT NULL	Tinyint	Asignaturas que son optativas para la carrera.
	Calle y Número		NULL	Varchar(255)	Calle y número del domicilio donde se imparte la carrera.

Entidad	Atributo	Llave	Nulidad	Tipo de dato	Descripción del campo
	Clave de Carrera	(PK)	NOT NULL	Tinyint	Identificador de la carrera.
	Clave de Municipio		NULL	Tinyint	Identificador del municipio donde se encuentra el domicilio donde se imparte la carrera.
	Clave de Nivel		NOT NULL	Tinyint	Identificador
	Código postal		NULL	Numeric(5)	Código postal del domicilio donde se imparte la carrera.
	Colonia		NULL	Varchar(255)	Colonia del domicilio donde se imparte la carrera.
	Créditos Obligatorios		NOT NULL	Smallint	Créditos obligatorios que se deben cubrir.
	Créditos Optativos		NOT NULL	Smallint	Créditos optativos que se deben cubrir.
	Descripción		NULL	Text	Información sobre la carrera.
	Inicio		NOT NULL	Smalldatetime	
	Nombre		NOT NULL	Varchar(64)	Nombre de la carrera.
	Periodos		NOT NULL	Tinyint	Periodos de los que consta la carrera.
	Término		NULL	Smalldatetime	
	Teléfono		NULL	Int	Teléfono del domicilio de donde se imparte.
Adscrito al Seminario	Clave de Profesor	(FK)	NOT NULL	Smallint	Identificador único del profesor.
	Clave de Seminario	(FK)	NOT NULL	Tinyint	Identificador único del seminario al que pertenece el profesor.
	Observaciones		NULL	Text	Información del profesor en particular.
Alumno-Asignatura	Clave de Asignatura	(FK)	NOT NULL	Smallint	Identificador de la asignatura en la que está inscrito el alumno.
	Clave de Estado	(FK)	NOT NULL	Tinyint	Identificador del estado de reinscripción.
	Número de Cuenta	(FK)	NOT NULL	Int	Número de cuenta del alumno.
Alumno-Horario	Atenciones		NULL	Tinyint	Número de atenciones que el alumno ha tenido dentro de su horario de atención.
	Clave de Horario	(FK)	NOT NULL	Tinyint	Identificador único de horario de atención.
	Movimientos		NULL	Tinyint	Movimientos que se han elaborado dentro de el horario de atención.

Entidad	Atributo	Llave	Nulidad	Tipo de dato	Descripción del campo
	Número de Cuenta	(FK)	NOT NULL	Int	Número de cuenta del alumno.
	Tiempo Restante		NULL	Tinyint	Tiempo restante de atención.
Aula	Activo		NOT NULL	Bit	Indicador de que la aula se encuentre disponible.
	Capacidad		NULL	Tinyint	Capacidad del aula.
	Clave de Aula	(PK)	NOT NULL	Tinyint	Identificador del aula.
	Nombre	(AK)	NOT NULL	Varchar(64)	Nombre del aula.
	Ubicación		NULL	Text	Lugar donde se encuentra el aula.
Bachillerato	Activo		NOT NULL	Bit	Indicador de que el bachillerato se encuentre activo.
Cambio de historia	Clave de Bachillerato	(PK)	NOT NULL	Tinyint	Identificador de donde el alumno curso el bachillerato.
	Descripción		NULL	Text	Información acerca del bachillerato.
	Nombre	(AK)	NOT NULL	Varchar(64)	Nombre del bachillerato.
	Aplicado		NOT NULL	Bit	Indicador que muestra si se aplicó el cambio a historia.
	Asignatura	(IE1)(FK)	NOT NULL	Smallint	Identificador de la asignatura a la cual se aplicará el cambio.
	Autorizado		NOT NULL	Bit	Indicador de autorización.
	Clave de calificación	(FK)	NOT NULL	Tinyint	Identificador de calificación.
	Clave de Grupo	(IE1)(FK)	NOT NULL	Tinyint	Identificador de grupo.
	Ciente		NOT NULL	Smallint	Persona que realiza el cambio.
	Enviado		NOT NULL	Bit	Indicador
	Fecha		NOT NULL	Smalldatetime	Fecha en la que se realiza el cambio.
	Folio		NOT NULL	Int	Folio DGAE.
	Número de Cuenta	(IE1)(FK)	NOT NULL	Int	Número de cuenta del alumno.
	Observaciones		NULL	Text	Información acerca del cambio de calificación.
	Periodo	(IE1)(FK)	NOT NULL	Smallint	Periodo en el que se curso la asignatura.
Tipo		NOT NULL	Char(1)	Tipo de cambio de calificación.	
Anuncio Global	Tipo de Movimiento		NOT NULL	Char(1)	
	Ciente		NOT NULL	Smallint	Persona que realiza el anuncio.
	Fecha de Elaboración		NOT NULL	Smalldatetime	Fecha de elaboración del mensaje.

Entidad	Atributo	Llave	Nulidad	Tipo de dato	Descripción del campo
	Fecha de Eliminación	(IE1)	NOT NULL	Smalldatetime	Fecha de eliminación del mensaje.
	Mensaje		NULL	Text	Información del anuncio.
Anuncio de Grupo	Asignatura	(IE1)(FK)	NOT NULL	Smallint	Identificador de la asignatura donde se requiere enviar el mensaje.
	Clave de Grupo	(IE1)(FK)	NOT NULL	Tinyint	Identificador del grupo donde se requiere enviar el mensaje.
	Cliente		NOT NULL	Smallint	Persona que realiza el anuncio.
	Fecha de Elaboración		NOT NULL	Smalldatetime	Fecha de elaboración del anuncio.
	Fecha de Eliminación		NOT NULL	Smalldatetime	Fecha de eliminación del anuncio.
	Mensaje		NULL	Text	Información del anuncio.
	Periodo	(IE1)(FK)	NOT NULL	Smallint	Identificador del periodo.
Bitácora	Fecha	(PK)	NOT NULL	Smalldatetime	Fecha en la que se elaboraron movimientos.
	Movimiento		NOT NULL	Smallint	Movimientos que se realizaron.
	Número de Cuenta	(FK) (PK)	NOT NULL	Int	Número de cuenta del alumno que realiza movimientos.
	Usuario		NOT NULL	Smallint	Persona que realizo los movimientos.
Calificación	Aprobatoria		NOT NULL	Bit	Indicador de que una calificación es aprobatoria.
	Clave de Calificación	(PK)	NOT NULL	Tinyint	Identificador único de calificación.
	Identificador	(AK)	NOT NULL	Char(2)	Representación numérica o literal de la calificación
	Valor		NULL	Numeric(4,2)	Valor numérico que corresponde a la representación literal.
Cambio a Inscripción	Aplicado		NOT NULL	Bit	Indicador de cambio de inscripción.
	Asignatura	(IE1)(FK)	NOT NULL	Smallint	Identificador de la asignatura que se solicita cambiar.
	Autorizado		NOT NULL	Bit	Indicador de la autorización.

Entidad	Atributo	Llave	Nulidad	Tipo de dato	Descripción del campo
	Clave de Grupo	(FK)	NULL	Tinyint	Identificador del grupo de donde se solicita el cambio.
	Clave de Grupo	(FK)	NULL	Tinyint	Identificador del grupo a donde se solicita el cambio.
	Cliente		NOT NULL	Smallint	Persona que realiza el cambio.
	Enviado		NOT NULL	Bit	Indicador de envío.
	Fecha		NOT NULL	Smalldatetime	Fecha en la que se realiza el cambio.
	Número de Cuenta	(IE1)(FK)	NOT NULL	Int	Número de cuenta del alumno.
	Observaciones		NULL	Text	Información acerca del cambio de inscripción.
Candidato a Oposición	Periodo	(IE1)(FK)	NOT NULL	Smallint	Periodo al que corresponde la inscripción.
	Clave de Examen	(FK) (PK)	NOT NULL	Smallint	Identificador del examen que se presenta.
	Clave de Profesor	(FK) (PK)	NOT NULL	Smallint	Identificador del profesor que es candidato.
	Clave de Resultado	(FK)	NOT NULL	Tinyint	Identificador del los posibles resultados de examen.
	Observaciones		NULL	Text	Información del profesor.
Caso Excepcional	Alumno	(IE1)(FK)	NOT NULL	Int	Identificador único del alumno. Número de cuenta.
	Aplicada		NOT NULL	Bit	Indicador de la aplicación del caso.
	Asignatura	(IE1)(FK)	NOT NULL	Smallint	Identificador de la asignatura que se solicita.
	Autorizada		NOT NULL	Bit	Indicador de la autorización.
	Clave de Categoría de Caso Excepcional	(FK)	NOT NULL	Tinyint	Identificador de la categoría del caso excepcional.
	Examinada		NOT NULL	Bit	Indicador
	Explicación		NULL	Text	Información acerca del movimiento que se realizó.
	Grupo	(IE1)(FK)	NOT NULL	Tinyint	Identificador del grupo que se solicita.
	Periodo	(IE1)(FK)	NOT NULL	Smallint	Periodo al que pertenece la inscripción.

Entidad	Atributo	Llave	Nulidad	Tipo de dato	Descripción del campo
Categoría de Caso Excepcional	Activo		NOT NULL	Bit	Indicador de que la categoría se encuentre activa.
	Clave de Categoría	(PK)	NOT NULL	Tinyint	Identificador de la categoría que se trata.
	Descripción		NULL	Text	Información acerca de la categoría.
	Nombre	(AK)	NOT NULL	Varchar(64)	Nombre de la categoría.
Categoría de Observación de Alumno	Activo		NOT NULL	Bit	Indicador de que la categoría se encuentre activa.
	Clave de Categoría	(PK)	NOT NULL	Tinyint	Identificador de la categoría.
	Descripción		NULL	Text	Información acerca de la categoría.
	Nombre	(AK)	NOT NULL	Varchar(64)	Nombre de la categoría.
Categoría de Observación de Profesor	Activo		NOT NULL	Bit	Indicador de que la categoría se encuentre activa.
	Clave de Categoría	(PK)	NOT NULL	Tinyint	Identificador de la categoría.
	Descripción		NULL	Text	Información acerca de la categoría.
	Nombre	(AK)	NOT NULL	Varchar(64)	Nombre de la categoría.
Categoría de Profesor	Activo		NOT NULL	Bit	Indicador de que la categoría se encuentre activa.
	Categoría		NOT NULL	Char(3)	Nombre de la categoría.
	Clave de Categoría	(PK)	NOT NULL	Tinyint	Identificador de la categoría.
	Descripción		NULL	Text	Información acerca de la categoría.
Categoría de Profesor	Nivel		NULL	Char(1)	Nivel en el que se encuentra el profesor.
	Sueldo Numérico		NULL	Money	Representación numérica del sueldo del profesor, según categoría.
Categoría de Profesor	Sueldo con Letra		NULL	Varchar(255)	Sueldo del profesor con letra, según categoría.
	Activo		NOT NULL	Bit	Indicador de que la comisión se encuentre activa.
Comisión	Clave de Comisión	(PK)	NOT NULL	Tinyint	Identificador de la comisión.
	Clave de Seminario	(FK)	NULL	Tinyint	Identificador del seminario al que pertenece la comisión.
	Observaciones		NULL	Text	Información acerca de la comisión.

Entidad	Atributo	Llave	Nulidad	Tipo de dato	Descripción del campo
Comisionado	Clave de Comisión	(FK) (PK)	NOT NULL	Tinyint	Identificador de la comisión a la que pertenece.
	Clave de Profesor	(FK)	NOT NULL	Smallint	Identificador del profesor que pertenece a la categoría
	Observaciones		NULL	Text	Información sobre la comisión.
Contrato	Código Programático	(PK)	NOT NULL	Tinyint	Identificador único del contrato.
	Causa		NULL	Tinyint	
	Clave de Categoría	(FK)	NOT NULL	Tinyint	Identificador de la categoría a la que pertenece el profesor.
	Clave de Profesor	(FK)	NOT NULL	Smallint	Identificador del profesor que se contrata.
	Fecha de Inicio		NULL	Smalldatetime	Fecha de inicio del contrato.
	Fecha de Término		NULL	Smalldatetime	Fecha de término del contrato.
	Movimiento		NULL	Tinyint	
	No de Expediente		NULL	Tinyint	No. de expediente que lleva el contrato.
	No de Horas		NULL	Tinyint	
	No de Horas Prácticas		NULL	Tinyint	
	No de Oficio		NULL	Tinyint	
	No de Registro de Plaza		NULL	Tinyint	
	No de Relación		NULL	Tinyint	
	Observaciones		NULL	Text	Información acerca del contrato.
	Personal		NULL	Tinyint	
	Sustituye a		NULL	Char(13)	Persona a la que sustituye en el cargo.
Tipo de Profesor		NULL	Tinyint		
Credencial Alumno	Fecha de Solicitud		NOT NULL	Smalldatetime	Fecha en que solicita credencial.
	Número de Cuenta	(FK)	NOT NULL	Int	Número de cuenta del alumno.
	No de solicitud	(PK)	NOT NULL	Numeric	Identificador de la solicitud del alumno.
Estado	Clave del Estado	(PK)	NOT NULL	Tinyint	Identificado del estado donde se encuentra el domicilio.
	Nombre	(AK)	NOT NULL	Varchar(64)	Nombre del estado.
Estado Civil	Activo		NOT NULL	Bit	Indicador de que se encuentre activo el estado civil.
	Clave de Estado Civil	(PK)	NOT NULL	Tinyint	Identificador del estado civil.

Entidad	Atributo	Llave	Nulidad	Tipo de dato	Descripción del campo
	Descripción		NULL	Text	Información sobre el estado civil.
	Nombre		NOT NULL	Varchar(64)	Nombre del estado civil.
Estado de Alumno	Activo		NOT NULL	Bit	Indicador de que se encuentre activo el estado del alumno.
	Clave de Estado	(PK)	NOT NULL	Tinyint	Identificador del estado del alumno.
	Descripción		NULL	Text	Información acerca del estado.
	Nombre	(AK)	NOT NULL	Varchar(64)	Nombre del estado del alumno.
Estado de Alumno-Asignatura	Activo		NOT NULL	Bit	Indicador de que se encuentre activo el estado alumno-asignatura.
	Clave de Estado	(PK)	NOT NULL	Tinyint	Identificador del estado.
	Descripción		NULL	Text	Información acerca del estado.
	Nombre	(AK)	NOT NULL	Varchar(255)	Nombre del estado.
Estado de Grupo	Activo		NOT NULL	Bit	Indicador de que se encuentre activo el estado de grupo.
	Clave de Estado	(PK)	NOT NULL		Identificador del estado de grupo.
	Descripción		NULL	Text	Información acerca del estado de grupo.
	Grupo			Tinyint	Grupo al que se está refiriendo.
	Nombre	(AK)	NOT NULL	Varchar(64)	Nombre del estado.
Estado de Inscripción	Activo		NOT NULL	Bit	Indicador de que se encuentre activo el estado de inscripción.
	Clave de Estado	(PK)	NOT NULL	Tinyint	Identificador del estado de inscripción del alumno.
	Descripción		NULL	Text	Información acerca del estado.
	Nombre	(AK)	NOT NULL	Varchar(64)	Nombre del estado.
Examen de Oposición	Clave de Asignatura	(FK)	NULL	Smallint	Identificador de la asignatura que requiere de un profesor.
	Clave de Comisión	(FK)	NULL	Tinyint	Identificador de la comisión que lo elegirá.
	Clave de Examen	(PK)	NOT NULL	Smallint	Identificador del examen que se realizará para obtener la plaza.
	Fecha de Convocatoria		NOT NULL	Smalldatetime	Fecha en que se lanza la convocatoria.
	Fecha de Dictamen		NULL	Smalldatetime	Fecha en que se dan resultados.

Entidad	Atributo	Llave	Nulidad	Tipo de dato	Descripción del campo
	Número de Plazas		NOT NULL	Tinyint	Número de plazas a cubrir.
	Observaciones		NULL	Text	Información acerca del examen.
Falta	Asignatura	(FK) (PK)	NOT NULL	Smallint	Identificador de la asignatura en la que se registro falta.
	Clave de Tipo de Falta	(FK)	NOT NULL	Tinyint	Identificador del tipo de falta.
	Fecha	(PK)	NOT NULL	Smalldatetime	Fecha en que se registro la falta.
	Grupo	(FK) (PK)	NOT NULL	Tinyint	Identificador del grupo donde se registro la falta.
	Periodo	(FK) (PK)	NOT NULL	Smallint	Periodo en el que se cursa la asignatura.
	Profesor	(FK) (PK)	NOT NULL	Smallint	Identificador del profesor que registro la falta.
Falta Profesor de Carrera	Clave de Profesor	(FK) (PK)	NOT NULL	Smallint	Identificador del profesor de carrera que registro la falta.
	Clave de Tipo de Falta	(FK)	NOT NULL	Tinyint	Identificador del tipo de falta.
	Fecha de Falta	(PK)	NOT NULL	Smalldatetime	Fecha en que se registro la falta.
	Observaciones		NULL	Text	Observaciones acerca de la falta.
Función	Activo		NOT NULL	Bit	Indicador de que se encuentre activa la función.
	Clave de Función	(PK)	NOT NULL	Tinyint	Identificador de la función que se desempeña.
	Descripción		NULL	Text	Información acerca de la función que se desempeña.
	Nombre	(AK)	NOT NULL	Varchar(64)	Nombre de la función.
Grado Académico	Activo		NOT NULL	Bit	Indicador de que se encuentre activo el grado académico del profesor.
	Clave de Grado	(PK)	NOT NULL	Tinyint	Identificador del grado académico.
	Descripción		NULL	Text	Información del grado académico.
	Nombre		NOT NULL	Varchar(64)	Nombre del grado académico.
Grupo	Clave de Asignatura	(FK) (PK)	NOT NULL	Smallint	Identificador de la asignatura que pertenece al grupo.

Entidad	Atributo	Llave	Nulidad	Tipo de dato	Descripción del campo
	Clave de Aula	(FK)	NOT NULL	Tinyint	Identificador de la aula donde se imparte la asignatura.
	Clave de Estado de Grupo	(FK)	NOT NULL	Tinyint	Identificador del estado en el que se encuentra el grupo.
	Clave de Grupo	(PK)	NOT NULL	Tinyint	Identificador del grupo.
	Clave de Horario	(FK)	NOT NULL	Tinyint	Identificador del horario en que se imparte cada asignatura.
	Clave de Periodo	(FK) (PK)	NOT NULL	Smallint	Identificador del periodo que se está cursando actualmente.
	Cupo		NOT NULL	Tinyint	Cupo disponible por grupo.
	Inscritos		NOT NULL	Tinyint	Número de alumnos inscritos.
	Nombre DGAE		NOT NULL	Char(4)	Nombre del grupo que da DGAE.
Historia	Asignatura	(IE1)(FK) (PK)	NOT NULL	Smallint	Identificador de la asignatura en la que asentará calificación.
	Clave de Calificación	(FK)	NOT NULL	Tinyint	Identificador de la calificación que se obtiene.
	Clave de Grupo	(FK) (PK)	NOT NULL	Tinyint	Identificador del grupo al que pertenece el alumno.
	Folio		NOT NULL	Int	Folio del acta donde se asienta calificación.
	Número de Cuenta	(IE1)(FK) (PK)	NOT NULL	Int	Número de cuenta del alumno.
	Periodo	(FK) (PK)	NOT NULL	Smallint	Periodo en el que se cursa la asignatura.
Horario	Activo		NOT NULL	Bit	Indicador de que se encuentra activo el horario.
	Clave de Horario	(PK)	NOT NULL	Tinyint	Identificador del horario.
	Días	(AK)	NOT NULL	Tinyint	Días en lo que se divide el horario.
	Hora Final	(AK)	NOT NULL	Smalldatetime	Hora final de clase.
	Hora Inicial	(AK)	NOT NULL	Smalldatetime	Hora inicial de clase.
Nivel	Activo		NOT NULL	Bit	Indicador de que se encuentre activo el nivel.
	Clave de Nivel	(PK)	NOT NULL	Tinyint	Identificador del nivel.
	Descripción		NULL	Text	Información acerca del nivel.
	Nombre	(AK)	NOT NULL	Varchar(255)	Nombre del nivel.

Entidad	Atributo	Llave	Nulidad	Tipo de dato	Descripción del campo
Observación	Clave de Categoría	(FK)	NOT NULL	Tinyint	Identificador de la categoría de observación.
	Fecha		NOT NULL	Smalldatetime	Fecha en la que se realiza la observación.
	Número de Cuenta	(FK)	NOT NULL	Int	Número de cuenta del alumno.
	Observación		NULL	Text	Texto.
	Título		NULL	Varchar(255)	Nombre de la observación.
	Usuario		NOT NULL	Smallint	Persona que realiza la observación.
Observación a Profesor	Autor		NOT NULL	Tinyint	Persona que realiza la observación.
	Clave de Categoría	(FK)	NOT NULL	Tinyint	Identificador de la categoría de observación.
	Clave de Profesor	(FK)	NOT NULL	Smallint	Identificador del profesor al que se hace la observación.
	Fecha		NOT NULL	Smalldatetime	Fecha en la que se realiza la observación.
	Observación		NULL	Text	Texto.
	Título		NOT NULL	Tinyint	Nombre de la observación.
Observación de Grupo	Asignatura	(FK) (PK)	NOT NULL	Smallint	Identificador de la asignatura a la que se hace la observación.
	Clave de Grupo	(FK) (PK)	NOT NULL	Tinyint	Identificador del grupo al que pertenece la asignatura.
	Observación		NULL	Text	Texto.
	Periodo	(FK) (PK)	NOT NULL	Smallint	Periodo en el que se cursa la asignatura.
Parámetro	Clave de Parámetro	(PK)	NOT NULL	Tinyint	Identificador del parámetro.
	Valor Cadena		NULL	Varchar(255)	
	Valor Lógico		NULL	Bit	
	Valor Numérico		NULL	Int	
Periodo	Clave de Periodo	(PK)	NOT NULL	Smallint	Identificador del periodo en el que se cursan la asignaturas.
	Fecha Inicio		NOT NULL	Smalldatetime	Fecha de inicio del periodo.
	Fecha Término		NOT NULL	Smalldatetime	Fecha de término del periodo.
	Nombre	(AK)	NOT NULL	Varchar(64)	Nombre del periodo.
	Nombre Corto		NOT NULL	Char(5)	Nombre corto del periodo.
	Observaciones		NULL	Text	Información acerca del periodo.

Entidad	Atributo	Llave	Nulidad	Tipo de dato	Descripción del campo
Profesor	Activo		NOT NULL	Bit	Indicador que muestra si el profesor se encuentra activo.
	Apellido Materno	(IE1)	NOT NULL	Varchar(32)	Apellido paterno del profesor.
	Apellido Paterno	(IE1)	NOT NULL	Varchar(32)	Apellido materno del profesor.
	Asociaciones		NULL	Text	Asociaciones a las que pertenece.
	Cargos		NULL	Text	Cargos que ha tenido el profesor.
	Cédula Profesional		NULL	Int	Cédula profesional del profesor.
	Clave de Estado Civil	(FK)	NULL	Tinyint	Identificador del estado civil del profesor.
	Clave de Grado Académico	(FK)	NOT NULL	Tinyint	Identificador del grado académico del profesor.
	Clave de Municipio	(FK)	NULL	Tinyint	Identificador del municipio donde se encuentra el domicilio del profesor.
	Clave de Municipio	(FK)	NULL	Tinyint	Identificador del municipio donde se encuentra el domicilio del trabajo del profesor.
	Clave de Profesor	(PK)	NOT NULL	Smallint	Identificador único del profesor.
	Clave de Tipo de Profesor	(FK)	NOT NULL	Tinyint	Identificador del tipo de profesor.
	Correo Electrónico		NULL	Varchar(255)	Correo electrónico del profesor.
	Distinciones		NULL	Text	Distinciones que ha tenido el profesor.
	Domicilio Código Postal		NULL	Numeric(5)	Código postal del domicilio del profesor.
	Domicilio Calle y Número		NULL	Varchar(255)	Calle y número del domicilio del profesor.
	Domicilio Colonia		NULL	Varchar(255)	Colonia del domicilio del profesor.
	Domicilio Teléfono		NULL	Int	Teléfono del domicilio del profesor.
	Expediente		NULL	Numeric(12)	Número del expediente.
	Extranjero		NOT NULL	Bit	Indicador de que el profesor es extranjero.
Fecha Definitividad		NULL	Smalldatetime		
Fecha de Ingreso		NULL	Smalldatetime	Fecha de ingreso del profesor.	
Idiomas		NULL	Text	Idiomas que domina el profesor.	
Instituciones		NULL	Text	Instituciones donde ha laborado.	

Entidad	Atributo	Llave	Nulidad	Tipo de dato	Descripción del campo
	Lugar de Nacimiento		NULL	Varchar(255)	Lugar de nacimiento del profesor.
	Nombre	(IE1)	NOT NULL	Varchar(32)	Nombre(s) del profesor.
	Observaciones		NULL	Text	Observaciones acerca del profesor.
	Publicaciones		NULL	Text	Publicaciones que elaborado.
	RFC		NULL	Char(13)	Registro Federal de Causantes.
	SNI		NOT NULL	Bit	
	Sexo		NOT NULL	Bit	Indicador del sexo del profesor.
	Teléfono Celular		NULL	Int	Teléfono celular del profesor.
	Títulos		NULL	Text	
	Trabajo Código Postal		NULL	Numeric(5)	Código postal del domicilio del trabajo del profesor.
	Trabajo Calle y Número			Varchar(255)	Calle y número del domicilio del trabajo del profesor.
	Trabajo Colonia		NULL	Varchar(255)	Colonia del domicilio del trabajo del profesor.
	Trabajo Teléfono		NULL	Int	Teléfono del trabajo del profesor.
Profesor-Grupo	Asignatura	(FK) (PK)	NOT NULL	Smallint	Identificador de la asignatura que imparte el profesor.
	Clave de Grupo	(FK) (PK)	NOT NULL	Tinyint	Identificador del grupo donde imparte clase.
	Clave de Profesor	(FK) (PK)	NOT NULL	Smallint	Identificador del profesor.
	Fecha de Inicio		NOT NULL	Smalldatetime	Fecha de inicio
	Fecha de Término		NOT NULL	Smalldatetime	Fecha de término
	Número de Profesor		NULL	Tinyint	
	Periodo	(FK) (PK)	NOT NULL	Smallint	Periodo en el que se imparte la asignatura.
Resultado	Activo		NOT NULL	Bit	Indicador de que el resultado se encuentre activo.
	Clave de Resultado	(PK)	NOT NULL	Tinyint	Identificador del tipo de resultado.
	Descripción		NULL	Text	Información del resultado.
	Nombre	(AK)	NOT NULL	Varchar(64)	Nombre del resultado.
Seminario	Activo		NOT NULL	Bit	Indicador de que se encuentre activo el seminario.
	Clave de Carrera	(FK)	NOT NULL	Tinyint	Identificador de la carrera que cursa el alumno.

Entidad	Atributo	Llave	Nulidad	Tipo de dato	Descripción del campo
	Clave de Profesor	(FK)	NULL	Smallint	Identificador del profesor que pertenece al seminario.
	Clave de Seminario	(PK)	NOT NULL	Tinyint	Identificador del seminario.
	Descripción		NULL	Text	Información acerca del seminario.
	Nombre	(AK)	NOT NULL	Varchar(64)	Nombre del seminario.
	Clave de Asignatura	(FK) (PK)	NOT NULL	Smallint	Identificador de la asignatura relacionada con el seminario.
	Clave de Asignatura	(FK)(PK)	NOT NULL	Smallint	
Sinodal	Asistencia		NOT NULL	Bit	Indicador de la asistencia del profesor al examen profesional.
	Clave de Función	(FK)	NOT NULL	Tinyint	Identificador de la función del profesor en el examen profesional.
	Clave de Profesor	(FK) (PK)	NOT NULL	Smallint	Identificador del profesor.
	Clave de Tesis	(FK) (PK)	NOT NULL	Smallint	Identificador de la tesis que se presenta.
Tesis	Aprobación de Tema		NULL	Smalldatetime	Fecha de aprobación del tema de tesis.
	Aprobado		NOT NULL	Bit	Indicador de aprobado.
	Clave de Aula	(FK)	NULL	Tinyint	Identificador de la aula donde se presenta.
	Clave de Profesor	(FK)	NOT NULL	Smallint	Identificador del profesor a cargo de la tesis.
	Clave de Seminario	(FK)	NULL	Tinyint	Identificado del seminario relacionado con el tema de tesis.
	Clave de Tesis	(PK)	NOT NULL	Smallint	Identificador del tema de tesis.
	Fecha de Examen		NULL	Smalldatetime	Fecha de realización de examen profesional.
	Fecha de Registro		NOT NULL	Smalldatetime	Fecha de registro.
	Mención Honorífica		NOT NULL	Bit	Indicador de mención honorífica.
	Número de Cuenta	(FK)	NOT NULL	Int	Número de cuenta del alumno.
Tema		NOT NULL	Varchar(255)	Nombre del tema de tesis.	
Tipo de Asignatura	Activo		NOT NULL	Bit	Indicador de que se encuentre activo el tipo de asignatura.
	Clave de Tipo	(PK)	NOT NULL	Tinyint	Identificador del tipo de asignatura.

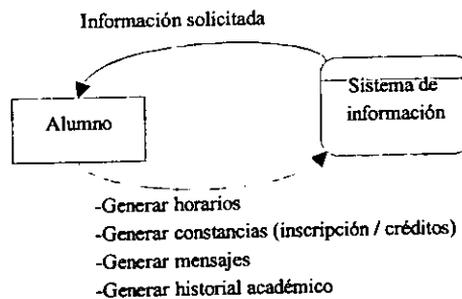
Entidad	Atributo	Llave	Nulidad	Tipo de dato	Descripción del campo
	Fecha de Registro		NOT NULL	smalldatetime	Fecha de registro.
	Mención Honorífica		NOT NULL	bit	Indicador de mención honorífica.
	Número de Cuenta	(FK)	NOT NULL	int	Número de cuenta del alumno.
	Tema		NOT NULL	varchar(255)	Nombre del tema de tesis.
Tipo de Asignatura	Activo		NOT NULL	bit	Indicador de que se encuentre activo el tipo de asignatura.
	Clave de Tipo	(PK)	NOT NULL	tinyint	Identificador del tipo de asignatura.
	Descripción		NULL	text	Información relacionada con el tipo de asignatura.
	Nombre	(AK)	NOT NULL	varchar(64)	Nombre del tipo de asignatura.
Tipo de Falta	Activo		NOT NULL	bit	Indicador de que se encuentre activo el tipo de falta.
	Clave de Tipo de Falta	(PK)	NOT NULL	tinyint	Identificador del tipo de falta.
	Descripción		NULL	text	Información relacionada con el tipo de falta.
	Nombre	(AK)	NOT NULL	varchar(64)	Nombre del tipo del falta.
Tipo de Profesor	Activo		NOT NULL	bit	Indicador de que se encuentre activo el tipo de profesor.
	Clave de Tipo	(PK)	NOT NULL	tinyint	Identificador del tipo de profesor.
	Descipción		NULL	text	Información relacionada con el tipo de profesor.
	Nombre	(AK)	NOT NULL	varchar(64)	Nombre el tipo de profesor.

## DISEÑO DEL MÓDULO DE INFORMACION ACADÉMICA PARA LOS ALUMNOS.

Una vez diseñada la base y proporcionando la división de la misma, de acuerdo a los módulos que serán realizados, es necesario que a partir de este momento solo se preste atención al caso práctico que complementa esta tesis, refiriéndose así al Sistema de información Académica para los alumnos.

El propósito del Sistema de información Académica para los alumnos es mantener informado a los alumnos sobre su situación académica, además de proporcionar el servicio de impresión de constancias tales como: constancia de inscripción y constancia de créditos, adicional a esto, proporcionará el servicio de recepción y envío de mensajes.

A continuación se presenta el diagrama de contexto del módulo III : Sistema de Información para los alumnos.



*Diagrama de Contexto del Sistema de información Académica para los alumnos.*

### Lista de Acontecimientos.

La siguiente tabla muestra la lista de acontecimientos del Sistema de información Académica para los alumnos.

No.	Acontecimiento	Entidad Externa
1	Generar horarios	Alumno
2	Genera constancia de inscripción	Alumno
3	Genera constancia de créditos	Alumno
4	Genera historial académico	Alumno
5	Genera mensajes	Alumno
6	Solicitud de información	Alumno

A continuación se presenta la porción de la base de datos que contiene las entidades que serán requeridas para realizar dicho módulo.

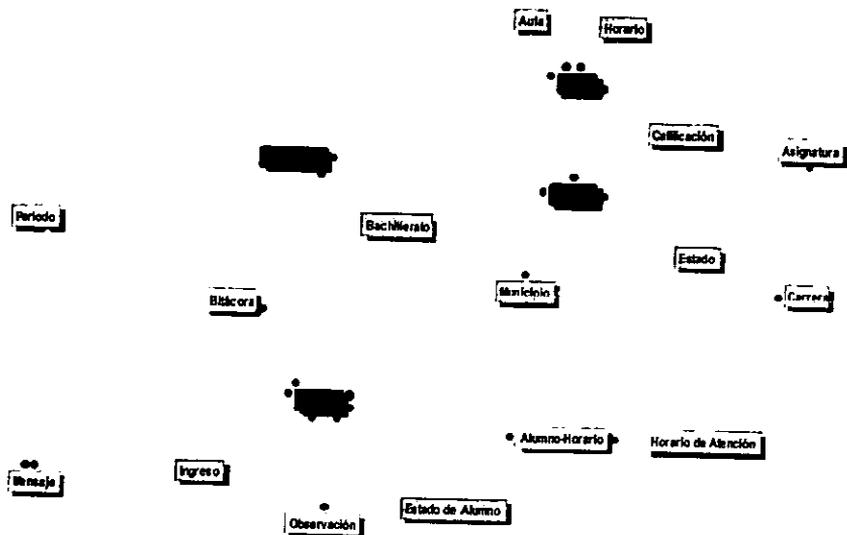
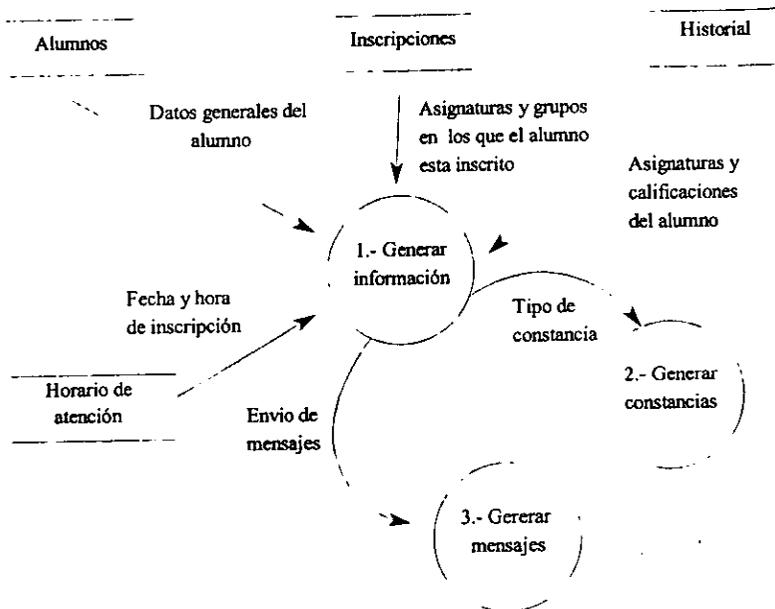
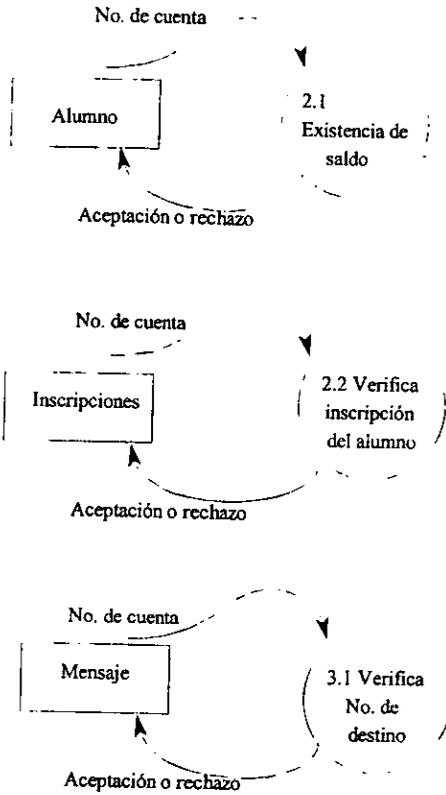


Diagrama Entidad Relación del Sistema información Académica para los alumnos.

### Diagrama de Flujo de datos.

A continuación se presenta el diagrama de flujo de datos de la aplicación, el cual permitirá ubicar el contexto en el cual funcionará el sistema, así como el diagramas de flujo de datos, el cual mostrara la dirección en que fluirán los datos.

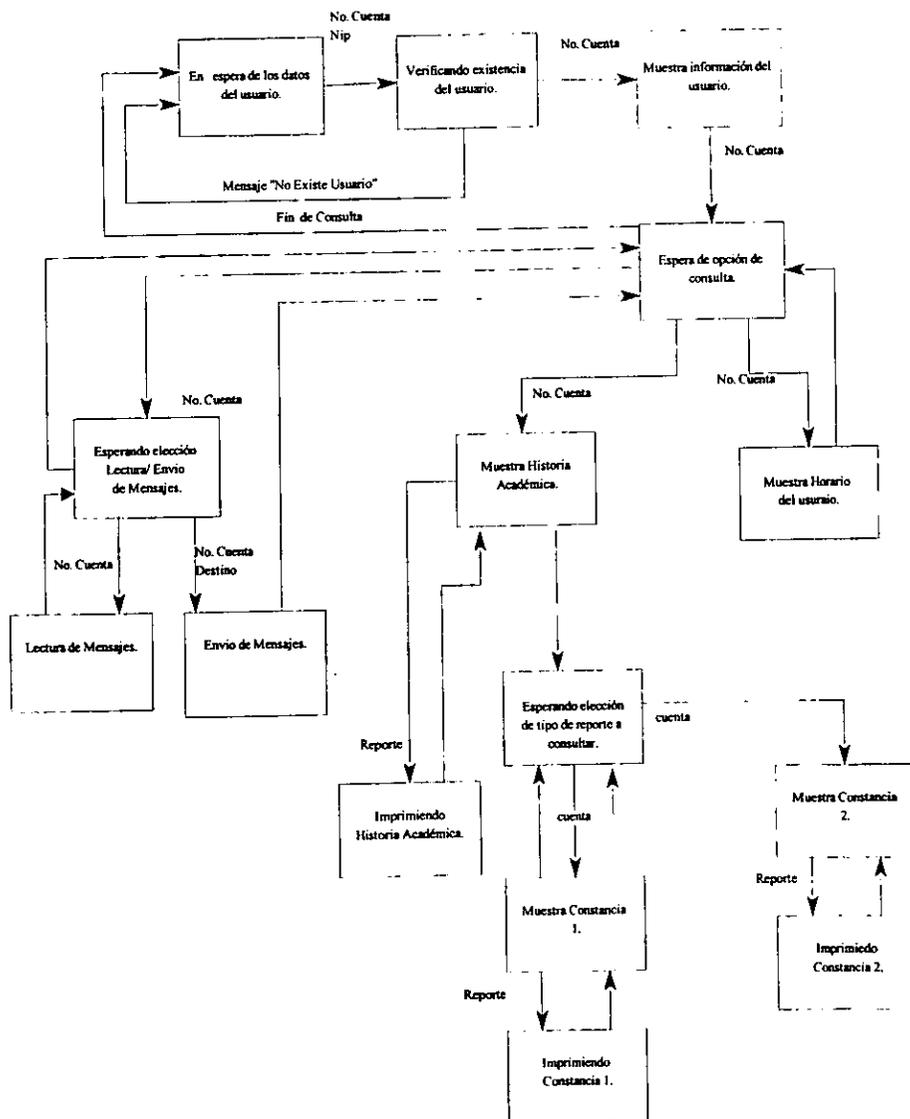




*Diagrama de Flujo de datos*

A continuación se presenta el diagrama de transición de estados, el cual enfatiza el comportamiento del sistema dependiente del tiempo, proporcionando una visión de los estados o eventos que ocurrirán al ir avanzando sobre la aplicación, así como los posibles caminos o alternativas que podremos encontrar.

DTE ( Diagrama de Transición de Estados )



Hasta el momento se ha realizado el diseño de la base de datos, así como del caso práctico, que es el complemento de la presente tesis.

En el siguiente capítulo se presenta la implementación tanto de la base de datos como del módulo de Información Académica para los alumnos.

---

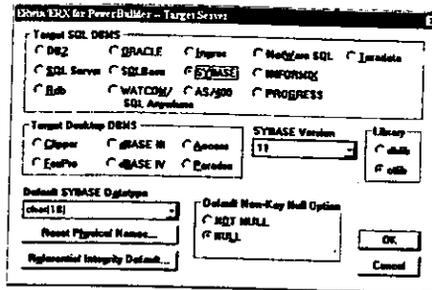
## CAPITULO III

# IMPLEMENTACION

### MODELADO DE DATOS EN HERRAMIENTA CASE.

ERwin es la herramienta CASE mediante la cual se realizó el modelado de la base de datos, además de generar el código para el manejador de la base de datos SYBASE.

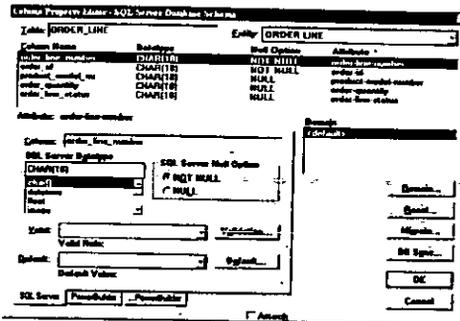
Primeramente, para utilizar esta herramienta se debe de seleccionar el manejador de bases de datos a utilizar (DBMS), que en este caso es el Sybase versión 11.



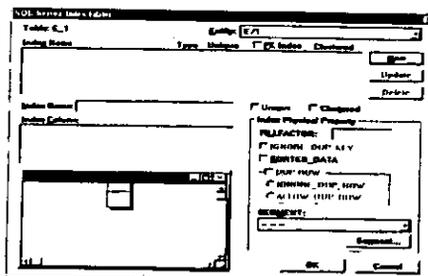
Este modelado se realiza a través de entidades, atributos y relaciones. Para crear estos objetos se hace uso de Erwin Toolbox.



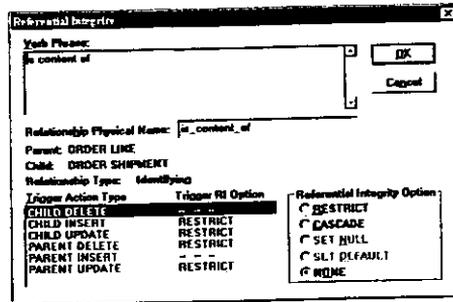
Una vez creadas las entidades con sus atributos y enlazadas por medio de relaciones, se debe de establecer por entidad el tipo de dato, dominio, nulidad, nombre físico y lógico del atributo.



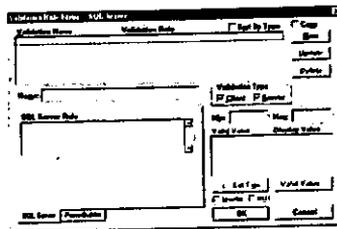
Además de establecer el ó los atributos que fungirán como índices de cada entidad.



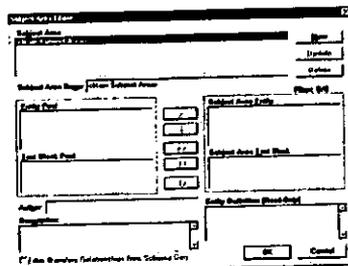
Además se debe establecer la Integridad Referencial, así como la frase que será asociada con cada relación



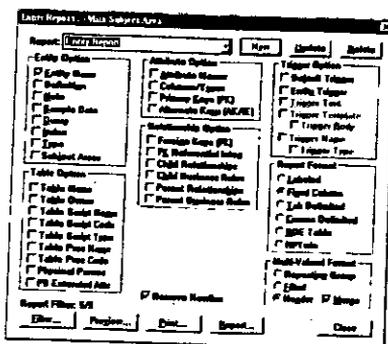
Otro punto importante a ser modelado, son las reglas de validación, según requieran los atributos.



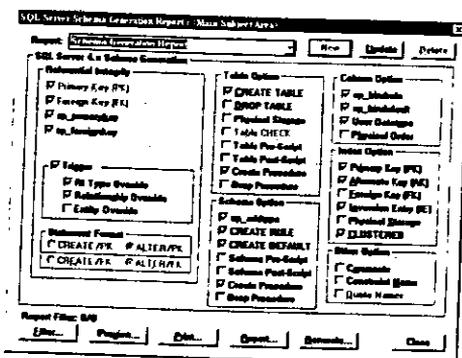
Cuando el modelado llega a ser extenso, éste se puede dividir en áreas para hacer menos engorroso el trabajo, decidiendo cuales de las entidades formaran parte de que área.



Una vez terminado el proceso de modelado se generan reportes por atributos, entidades etc.



Con ayuda del generador de esquemas, se puede obtener el código para el DBMS seleccionado.



El uso de la herramienta CASE denominada Erwin, dio como resultado la creación de la base de datos mostrada en el capítulo anterior, además de haber generado el código para el DBMS Sybase, el cual posteriormente permitió crear la base de datos.

## GENERACION DE LA BASE DE DATOS EN SYBASE.

Para la generación de la base de datos se utilizó Sybase SQL Server Manager, administrando servidores SQL y recursos físicos del SQL, bases de datos, acceso al servidor.

Para dar de alta la base de datos, es necesario establecer conexión con el servidor (Tlatoani), por lo tanto es necesario tener privilegios en Sybase SQL Server Manager, una vez establecida la conexión, se crea la base de datos, estableciendo nombre, dueño, espacio en los dispositivos para almacenar la base de datos.

Las estructuras creadas para la administración y funcionamiento de la base de datos son:

- Users. Usuarios de la base de datos.
- Groups. Categorías de usuarios que existen.
- Systems tables. Tablas que usa Sybase para operar.
- User tablas. Tablas definidas por el usuario.
- Views. Vistas que el usuario define.
- Indexes. Índices definidos por el usuario..
- Triggers. Auxilia al usuario para la integridad de la información.
- Procedures. Procedimientos almacenados definidos en el sistema.
- Rules. Reglas con respecto a los valores de los datos.
- Defaults. Valores defaults definidos para los datos.
- User Datatypes. Tipos de datos que el usuario define.

La base de datos Sistema Integral contiene los elementos anteriormente descritos. El término usuario se refiere a la persona que diseña e implementa la base de datos, así como aquellas personas que utilizaran información de la base,. A dichos usuarios se les otorga ciertos privilegios con el fin de establecer seguridad de la base de datos. A continuación se presentan los tipos de usuarios así como sus privilegios.

- |  |   |
|--|---|
| ➤ Administrador del sistema              | Es el usuario con mayor capacidad de acción, puede operar sobre todo el sistemas , además de administrar el resto de los usuarios |
| ➤ Personal técnico del centro de cómputo | Tiene acceso a todos los módulos, su limitante es que no tiene capacidad para crear nuevos usuarios.                              |
| ➤ Funcionarios de las Secretarías        | Operaran sobre cada módulos que les corresponda.  |

Una vez diseñada la base de datos se genera la base de datos. Al utilizar en el diseño una herramienta CASE, se genera la base de datos desde Erwin, permitiendo la conexión al servidor y ejecutando el código que la misma herramienta crea. Definiendo la plataforma destino del modelo creado en el caso del Sistema Integral se utiliza Sybase, se realiza la conexión al servidor y a la base de datos, y por último se ejecuta el código SQL generando: llaves primarias y foráneas, Triggers, opciones para las tablas, reglas, defaults, índices y usuarios.

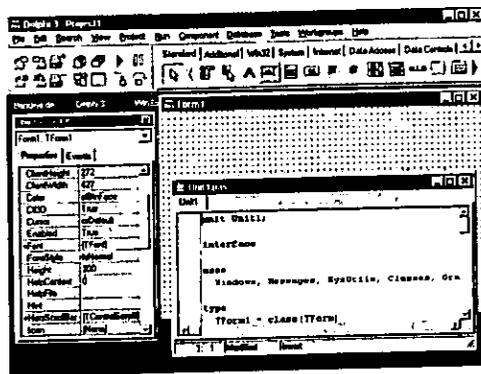


inconsistencia, en cuanto a la información de alumnos se refiere. Así mismo se realizaron pruebas con la base de datos de Secretaría Académica y Administrativa para obtener información veraz del personal académico.

## DESARROLLO DE LA APLICACION EN DELPHI.

Como se menciona en el capítulo anterior, esta tesis presenta como parte complementaria, el diseño e implementación del módulo denominado Sistema de Información Académica para los alumnos.

Delphi es una herramienta RAD (Rapid Application Development), desarrollo rápido de aplicaciones, la cual permite el desarrollo de software de forma visual. Los objetos que utiliza esta herramienta se desarrollan en ambiente Windows, es decir: ventanas, menús, botones, campos de edición, memos, etcétera. Estos objetos reaccionan a eventos activados por el usuario.



Delphi cuenta con un ambiente Integrado de Desarrollo, proporcionando todas las herramientas necesarias para diseñar, ejecutar y probar una aplicación. En Delphi, el IDE consta de un editor de código, un depurador, una barra de herramientas, un editor de imágenes y herramientas de bases de datos trabajando de manera integrada.

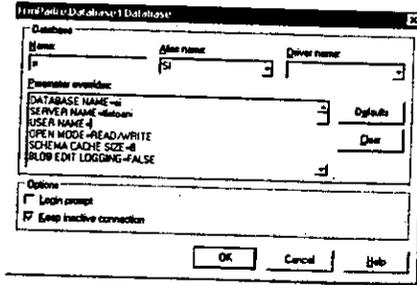
Una vez generada la base de datos, es decir el back end, se inicia el desarrollo del front end, es decir, la implementación del módulo III, Sistema de Información Académica para los alumnos, cuya descripción se mostrará a continuación.

Dentro del ambiente Delphi la primera ventana de inicio de sesión es la siguiente, como se mencionó anteriormente un ambiente IDE.

Para crear una aplicación se selecciona File/New Application, una vez que se elige esta opción el proyecto muestra el formulario y la ventana de código correspondiente al formulario como elementos más importantes para el desarrollo del sistema.

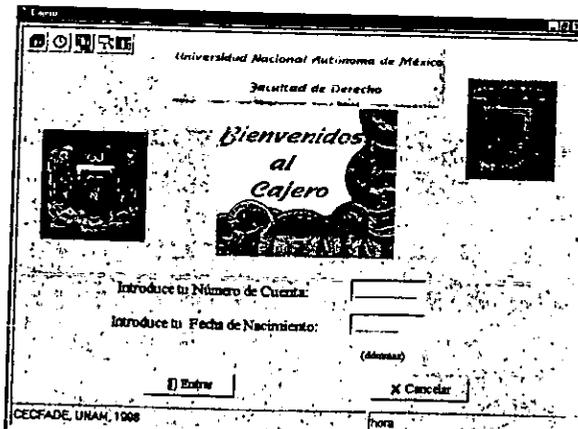
La primera pantalla de la aplicación permite el acceso del alumno a su información académica, para ello es necesario introducir el número de cuenta y la clave de acceso, para realizar esta tarea es necesario utilizar los siguientes objetos para la conexión a la base de datos en sybase: Datasource, Database, Table, Query.

El database permite la conexión a la base datos, una vez hecha la conexión el elemento Table permite seleccionar la tabla que contiene los datos que se necesitan, para este caso se utiliza la tabla alumnos, debido a que es necesario hacer una consulta y la comparación de campos se utiliza el elemento Query.



Para el elemento Query el código que realiza la consulta es:

```
select * from alumno  
where cuenta=:cuenta and nip=:nip
```



Al oprimir el botón entrar, se dispara el evento que consiste en realizar la comparación de los datos introducidos y los almacenados en la tabla de alumno.

Una vez validada la entrada de los datos del alumno se tiene la siguiente pantalla, en ella se seleccionan los campos que se han de utilizar para mostrar la información personal del alumno, para este caso se utilizan las tablas atención y aluaten. Además los botones muestran la información que puede ser consultada por el alumno.

Se tienen las siguientes opciones a consultar:

- Horario del semestre actual.

- Edición y consulta de mensajes.
- Historia Académica, constancia de créditos y constancia de materias.

La pantalla de datos personales utiliza los objetos Query, Table y Datasource.

En la pantalla del historial académico se accesa a las tablas alumnos, materias, calificación, historia, inscripción, asignatura, carrera, por medio de un Query:

```

SELECT historia.asignatura , asignatura.nombre , alumno.estado,
historia.grupo , calificacion.valor , periodo.corto , historia.folio,
asignatura.tipo, calificacion.id, asignatura.creditos, asignatura.semestre
FROM asignatura , historia , alumno,
calificacion , periodo
WHERE      (alumno.cuenta=:cuenta)
          AND
          (historia.cuenta=alumno.cuenta)
          AND
          ( historia.calificacion = calificacion.clave )
          AND
          ( historia.asignatura = asignatura.clave )
          AND
          ( historia.periodo = periodo.clave )

```

Además permite realizar la impresión de la historia académica, con un reporte que contiene con los campos utilizados en el formulario. El objeto utilizado para la impresión es PrintDialog.

Cabe mencionar que el promedio de la calificación es un dato que es calculado por medio de código.

El formulario del historial académico presenta las opciones de consulta de constancias de créditos y de materias.

Carrera	Asignatura	Grupo	Periodo	Calificación	Fecha
---------	------------	-------	---------	--------------	-------

Para la generación del reporte es necesario utilizar el objeto Quick Report de Delphi.

Los formularios correspondientes a las constancias son los siguientes.

SECRETARIA DE SERVICIOS ACADÉMICOS  
 DIRECCION GENERAL DE ADMINISTRACION ESCOLAR  
 SUBDIRECCION DE CERTIFICACION Y NORMATIVIDAD  
 SECRETARIA DE ASUNTOS ESCOLARES

**A QUIEN CORRESPONDA**

Se hace constar que el alumno  
 con Número de Cuenta  
 de la Carrera de  
 le corresponde la siguiente situación escolar

que se imparte en esta Facultad

está inscrito actualmente      Inicia Sem.      Fin Sem.

Pausado  
 Créditos Completos  
 Porcentaje de Créditos Completos

Las constancias se obtuvieron por medio de consultas realizadas con el objeto Query.

SECRETARIA DE SERVICIOS ACADÉMICOS  
 DIRECCION GENERAL DE ADMINISTRACION ESCOLAR  
 SUBDIRECCION DE CERTIFICACION Y NORMATIVIDAD  
 SECRETARIA DE ASUNTOS ESCOLARES

**A QUIEN CORRESPONDA**

Se hace constar que el alumno  
 con Número de Cuenta  
 de la Carrera de  
 está inscrito actualizante en las siguientes asignaturas

que se imparte en esta Facultad

Clave	Asignatura	Grupo	Presencia	Créditos

Para la parte de impresión de las constancias, se generó un reporte para cada una de ellas.

The screenshot shows a window titled 'constancia'. At the top, it displays the logo of the 'SECRETARIA DE SERVICIOS ACADÉMICOS' and the 'UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO'. Below this, it lists the 'DIRECCIÓN DE CERTIFICACIÓN Y HOMOLOGACIÓN' and 'SECRETARÍA DE ASISTENTES EDUCATIVOS'. The main section is titled 'A QUIÉN CORRESPONDA' and contains several fields for student information: 'Inscripción en el curso', 'Inscripción en la materia', 'Inscripción en esta Paralela', 'Inscripción en la asignatura', 'Inscripción en el semestre', 'Inscripción en el grupo', 'Inscripción en el aula', 'Inscripción en el turno', 'Inscripción en el día', 'Inscripción en el horario', 'Inscripción en el profesor', 'Inscripción en el docente', 'Inscripción en el alumno', 'Inscripción en el profesor', 'Inscripción en el docente', 'Inscripción en el alumno'. Below these fields, there is a section for 'Inscripción en el curso' and 'Inscripción en la materia'. At the bottom, there is a section for 'Inscripción en el curso' and 'Inscripción en la materia'.

El formulario horario realiza una consulta de la tablas alumno, asignatura, horario utilizando los campos alumno , no. de cuenta, asignatura, inicio, fin, aula.

The screenshot shows a window titled 'Horario'. It has two input fields: 'Alumno:' with the value '00000' and 'No. Cuenta:' with the value '00000'. Below these fields, there is a table with columns 'Clase', 'Asignatura', 'Grupo', 'Inicio', and 'Fin'. At the bottom of the window, there is a button labeled 'A. Salir'.

El formulario mensaje permite la edición y consulta mensajes.

Cabe mencionar que la aplicación esta pensada para ser utilizada, solo con mouse o solo con teclado, por lo cual se diseño un formulario con teclado numérico, en el caso de ser ejecutada con el uso del mouse.

The image shows a graphical user interface window titled "Numero de Cuenta". It contains two input fields: "No. de Cuenta:" and "Clave de Acceso:". Below these fields is a numeric keypad with buttons for digits 1 through 9, 0, a "Limpiar" (Clear) button, and an "OK" button with a checkmark icon.

Numero de Cuenta		
No. de Cuenta: <input type="text"/>		
Clave de Acceso: <input type="text"/>		
1	2	3
4	5	6
7	8	9
0	Limpiar	✓ OK

---

## CAPITULO IV

# CONCLUSIONES

El desarrollo del sistema integral permitió comprender la metodología de desarrollo de aplicaciones cliente/servidor y la metodología de Yourdon en su parte de análisis y diseño del software.

A su vez se utilizó una herramienta de diseño de software asistido por computadora (CASE), permitiendo diseñar una base de datos, además de generar el código para su creación, de este modo se le da un mayor peso a la parte de diseño, ya que de ello depende el buen desarrollo de un sistema. Además de crear paralelamente la documentación del diseño.

Además de permitir aplicar los conocimientos adquiridos en un problema real, para posteriormente poder desarrollar aplicaciones con la experiencia de haber desarrollado la presente tesis.

Habría que resaltar el haber empleado las herramientas de desarrollo: como el sistema operativo UNIX, el administrador de bases de datos Sybase, el lenguaje de programación Delphi, todo esto aplicado sobre una amplia red de cómputo, permitirá que más adelante se utilicen otras herramientas afines a las utilizadas en la presente tesis.

En general el sistema realizado cubre las diferentes expectativas de desempeño, en cuanto a la integración de la información.

Con respecto al sistema se lograron los siguientes objetivos:

- El ambiente amigable para los usuarios del sistema en sus diferentes módulos, en beneficio de los alumnos, profesores y el resto del personal administrativo.
- La integridad y la seguridad de la información es alta debido a la integración de ella por medio de la base de datos.
- El servidor es el encargado de realizar el proceso de la información, reduciendo los tiempos de espera, una vez que se trabaja con diversos módulos.
- El costo de operación y mantenimiento del sistema disminuirá notablemente debido a que tanto el código del cliente como la base de datos estarán documentados.
- Las principales perspectivas del producto a mediano y largo plazo son: que sea escalable, una base de datos que permita compartir la información entre las diversas áreas con las que cuenta la facultad.
- La tecnología cada vez es más cambiante, existiendo la posibilidad de operar otros medios de comunicación como lo es Internet. Por su parte el servidor está preparado para ello y sólo se afectaría la interfaz del usuario y el medio de acceso.
- La metodología de desarrollo planteada en la presente tesis, se puede aplicar a otro tipo de proyectos.

---

## APÉNDICE A

---

# METODOLOGÍA DE DESARROLLO DE SISTEMAS DE INFORMACIÓN

Por metodología se entiende "un conjunto integrado de procedimientos, técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo de un sistema de información"

*Sistema de Información:* Es un conjunto de componentes (computadoras, periféricos, software y usuarios) que trabajan juntos para conseguir un objetivo, transformando elementos de entrada (datos) al sistema en otros elementos de salida (información).

*Método:* Precisa en etapas o pasos exactos, tanto el proceso de desarrollo en general (ciclo de vida) como cada una de las fases generales en las que se divide éste.

*Técnicas:* Mecanismos, procedimientos y recursos de que se sirve la metodología.

La utilización de técnicas de diagramas favorece la comunicación entre el personal de desarrollo y los usuarios para los que se realiza el sistema. La documentación esquemática que se establece facilita el mantenimiento del sistema, lo que hace que se consiga mayor calidad del mismo.

La labor primaria como analista, es analizar o estudiar el sistema para determinar su esencia: comportamiento requerido, independientemente de la tecnología utilizada para implantar el sistema.

La incapacidad de estimar el tiempo, coste y esfuerzo necesario para el desarrollo de un producto software y, la falta de calidad del software producido son la causa de la aparición de la Ingeniería del Software como una disciplina científica.

Para clasificar el software se parte de la definición de Pressman: "El software se compone de los siguientes elementos:

- Instrucciones o programas que cuando se ejecutan tienen el comportamiento esperado.
- Estructuras de datos.
- Documentación que explica el uso y manipulación del programa."
- La Calidad del software se mide en base a los conceptos de:
- **Fiabilidad:** Capaz de dar los mismos resultados bajo las mismas condiciones.
- **Eficiencia:** Utilización óptima de los recursos del sistema.
- **Robustez o Tolerancia a Fallos:** No poseer comportamiento catastrófico ante situaciones excepcionales.
- **Corrección:** Debe ajustarse a las especificaciones dadas por el usuario y sin errores de diseño y codificación.
- **Adaptabilidad:** Fácilmente modificables algunas de sus funciones sin afectar al resto.
- **Portabilidad:** Capacidad de poder integrarse en entornos distintos con un mínimo esfuerzo.
- **Inteligible:** Que tenga un diseño claro y fácil, documentado y estructurado.
- **No erróneo:** No debe haber diferencia entre valores reales y calculados.

## **Ciclo de vida.**

Es el conjunto de fases por las que debe pasar un proyecto desde su concepción inicial, hasta que el sistema deja de utilizarse o se transforma en otro.

Existen diferentes modelos de ciclo de vida, que pueden aplicarse en función del tipo de sistema a desarrollar.

### **Ciclo de vida clásico o en cascada.**

Este ciclo establece una serie de fases, al finalizar las cuales se obtiene una serie de productos (documentos, diagramas, programas) que permite evaluar lo realizado hasta ese momento y continuar con la fase siguiente o modificar algunos aspectos de las fases anteriores.

**Fase de Planificación estratégica:** Su existencia es opcional. Si existe, su objetivo es adecuar los objetivos estratégicos de la organización (usuario) y la información necesaria para soportar dichos objetivos. Se debe determinar si el sistema es factible de informatizar, incluyendo algunas especificaciones básicas acerca de coste y tiempo necesarios para construir el sistema, así como los beneficios que se obtengan del nuevo sistema.

**Fase de Análisis:** El objetivo de esta fase es el estudio de las necesidades de información que debe satisfacer el sistema a desarrollar, elaborando una serie de especificaciones formales que describan la funcionalidad del mismo y que permitan abordar con garantías la siguiente fase. Se puede estructurar en dos subfases:

**Análisis de Requerimientos del Sistema:** Se trata de establecer el alcance, los objetivos y requisitos del sistema, examinando las posibles alternativas que podrían solucionar las necesidades del usuario y recomendar una de ellas. Al final de esta subfase se obtiene un documento llamado "Documento de Requisitos del Sistema"

**Especificación Funcional del Sistema:** Conocida como Análisis Funcional. Una vez aceptado formalmente el documento anterior por ambas partes (equipo de desarrollo y usuario), se elabora un conjunto de especificaciones formales que describan la funcionalidad del sistema, estableciendo los subsistemas en que se descompondrá, definiendo los datos que utilizará y las interfaces de usuario. También se planificarán las pruebas que deberá superar el sistema, se estimará la relación coste/beneficio para comprobar si interesa su construcción y se establecerán los plazos de entrega del sistema. Todo ello se recoge en dos documentos, denominados "Documento de Especificación Funcional del Sistema" y "Documento de Pruebas del Sistema"

**Fase de Diseño:** Conocida como Análisis Orgánico. El objetivo de esta fase es obtener un conjunto de especificaciones que contemplarán los aspectos físicos del sistema, considerando las características tecnológicas del entorno específico en el que se implantará, que constituirá el punto de partida para la construcción del sistema. Equivale a la creación de una jerarquía apropiada de módulos de programas y de interfaces entre ellos para implantar la especificación creada en la fase anterior. Además, se encarga de la transformación de modelos de datos de Entidad-Relación en un diseño de base de datos. Al final de esta fase se obtienen el "Documento de Diseño Técnico" y el anterior "Documento de Pruebas del Sistema" con las ampliaciones relativas a la definición del entorno de pruebas.

**Fase de Construcción:** El propósito de esta fase es, a partir de las especificaciones de diseño, la obtención del sistema completamente construido y probado, listo para ser implantado en la organización del usuario. También durante esta fase se desarrollará el conjunto de procedimientos y se llevará a cabo la formación necesaria que permitirá, tanto al personal del área de usuario final, como al personal del área de explotación o proceso de datos (si existe), la utilización óptima del sistema. Al final de esta fase se obtiene el Software correspondiente y, los siguientes documentos: "Documentación Técnica de Programación", "Manual de usuario", "Manual de Explotación", "Documento de Pruebas del Sistema" ampliado con los informes de las pruebas unitarias, de integración y globales.

**Generación de Pruebas Unitarias:** Comprobar la validación de cada uno de los módulos.

**Generación de Pruebas de Integración:** unión de todos los módulos y prueba de la unión.

**Fase de Implantación:** El objetivo de esta fase es la puesta en servicio del sistema construido y conseguir su adaptación final por parte de los usuarios del mismo, para lo cual tratará de hacerse ver a éstos, mediante demostraciones formales, es decir, pruebas de aceptación que el sistema cumple todos los objetivos y requisitos para los que fue desarrollado. En esta fase se incluye la ejecución y el mantenimiento del sistema, con lo que su duración se prolongará hasta que el sistema deje de utilizarse o sea sustituido por otro.

**Generación de Pruebas de Aceptación:** Detección de errores de que el programa no se ajusta a las especificaciones.

**Inconvenientes:**

Desarrollo manual.

Las herramientas utilizadas no están integradas ni relacionadas entre sí.

Los errores de análisis y diseño son muy caros de eliminar, ya que se encuentran muy tarde.

Se produce efecto bola de nieve: los errores se arrastran a las fases siguientes.

**Ciclo de vida de prototipos.**

*Prototipo:* Primera versión de un producto, construido con poca funcionalidad y poca fiabilidad. La diferencia entre prototipo y producto final es que el prototipo es eficiente y el producto final debe serlo y, que en el prototipo no están todos los detalles y, el producto final debe contenerlos.

**Clases de prototipos:**

**Vertical:** No recoge todas las funciones del sistema final.

**Horizontal:** Recoge todas las funciones, pero no las desarrolla por completo.

Este ciclo casi siempre supone que el modelo será operante, es decir, una colección de programas que simularán alguna o todas las funciones que el usuario desea. Pero dado que se pretende que dichos programas sean solo de modelo, también se supone que al concluirse el modelado, los programas se descartarán y se reemplazarán con programas reales. Normalmente se utilizan las siguientes herramientas:

Un diccionario de datos integrado.

Un generador de pantallas.

Un generador de informes no guiado por procedimientos.

Un lenguaje de programación de cuarta generación.

Un lenguaje de consultas no guiado por procedimientos.

Medios poderosos de administración de bases de datos.

**Ventajas:**

Se incrementa la productividad del equipo de desarrollo. Se incrementa la calidad del producto final, ya que el prototipo permite trabajar, ensayar.

Disminuyen los costes de mantenimiento del producto final. Los tiempos de desarrollo son inferiores.

El tamaño del sistema es menor.

La especificación actúa como interface entre cliente y equipo de desarrollo.

El propio prototipo sirve de contrato con el cliente y cualquier cambio en el prototipo debe estar consolidado por ambas partes.

El prototipo es un documento vivo de buen funcionamiento del producto final.

Ayuda para determinar requerimientos expresados en el prototipo. Experimenta sobre los aspectos del sistema que representan mayor complejidad. Demuestran la viabilidad del sistema.

El cliente reacciona mucho mejor ante el prototipo, sobre el que puede experimentar, que no sobre una especificación escrita.

#### **Inconvenientes:**

Fuerte inversión en un producto que es desechable: Los prototipos se descartan.

Tendencia a tratar de convertir el prototipo mismo en el sistema de producción.

Aumento del coste.

Se arrastran decisiones del diseño de prototipos al producto final.

#### **Ciclo de vida en espiral.**

Este ciclo recoge lo mejor de la combinación del ciclo en cascada y el ciclo de prototipo, solo añade una etapa denominada análisis de riesgo. En el ciclo de vida se define el desarrollo del software en cuatro etapas.

La vuelta interna del espiral, corresponde a la planificación, es donde se definen los objetivos, las alternativas y las restricciones, además se analizan e identifican los riesgos.

El segundo cuadrante corresponde a la etapa de análisis de riesgo, aquí se analizan los riesgos basados en los requisitos iniciales, los riesgos basados en la reacción del cliente y se decide la continuación.

La tercera etapa es la Ingeniería, se basa en el desarrollo del prototipo inicial del software, del prototipo del siguiente nivel, es decir, se encarga del desarrollo y verificación del producto del siguiente nivel.

La última etapa es la Evaluación del cliente, se encarga de la valoración de los resultados del proceso de desarrollo.

### **Desventajas.**

Incertidumbre relacionada con aspectos conceptuales y externos del sistema.

Falta de claridad en las funciones del sistema y sobre la base de datos.

### **Ventajas.**

Reducción de costos de mantenimiento.

Reducción de tiempos en las fases de diseño, realización y mantenimiento.

## **Herramientas y Métodos.**

### **Herramientas.**

Podemos definir las herramientas como el útil que se utiliza para desarrollar un análisis. Se usan para:

- Concentrarse en las propiedades importantes del sistema,
- Discutir cambios y correcciones de los requerimientos del usuario, a bajo coste y con el riesgo mínimo.
- Verificar que el analista comprende correctamente el entorno del usuario.
- Dan soporte automático o semiautomático a los métodos. Como las herramientas de modelado de sistemas importantes son:
- Diagramas Flujo de Datos. Ilustra las funciones que el sistema debe realizar.
- Diagrama Entidad-Relación. Hacen énfasis en las relaciones entre los datos.
- Diagrama de Transición de Estados. Se enfoca al comportamiento dependiente del tiempo del sistema.

### **Métodos o Metodología.**

Indican los pasos a seguir para desarrollar una aplicación (cómo desarrollar todas las tareas del desarrollo del software). Hay tres tendencias:

- Americana. Yourdon, Martin, Gane & Sarson
- Europea. Inglesa - SSADM
- Francesa. Merise

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software.

Las fases comunes a todas las metodologías son:

## **Análisis.**

Durante esta fase obtienen datos suficientes para establecer los servicios, restricciones y objetivos del sistema, a través de información proporcionada por los usuarios, además permite establecer una justificación económica y la viabilidad técnica del desarrollo e implementación del proyecto propuesto, junto con la definición funcional, modelado ambiental, modelado conceptual y modelado de datos.

El proceso de análisis se divide en tres etapas.

Investigación inicial.

Estudio de viabilidad.

Definición de requerimientos.

### **Investigación inicial.**

El proceso de análisis da como resultado la identificación de las necesidades a través de entrevistas realizadas por el analista, quien tendrá contacto con el cliente y usuarios.

El analista tendrá que distinguir entre lo que se "necesita" lo que se "quiere", con la finalidad de establecer las limitaciones del sistema propuesto en cuanto a costo y beneficio se refiere.

Esta investigación proporciona una amplia visión de los problemas existentes, costos de proyectos y esquemas para desarrollo y operación del sistema propuesto, además de producir un plan de trabajo con un estudio de viabilidad. Esta etapa se divide en las siguientes actividades:

- Establecer los requerimientos del proyecto. Determina los problemas, las necesidades y valida que los costos y beneficios indicados en los requerimientos del proyecto sean realizables.
- Preparar la conducción de la investigación inicial. Preparación de material de apoyo para realizar las entrevistas.
- Análisis de los datos coleccionados. Analizar y ordenar los datos coleccionados para hacer determinaciones preliminares relativas a costos del proyecto y tiempos esperados.

### **Estudio de Viabilidad.**

Implica hacer estimaciones aproximadas del costo y tiempo necesarios para construir un sistema nuevo y los beneficios que se derivarán de ello.

El estudio de viabilidad proporciona suficientes datos para permitir tomar decisiones basadas en la justificación económica y viabilidad técnica del proyecto propuesto. Esta etapa implica las siguientes actividades.

- Conducir entrevistas a los usuarios. Recopila datos para hacer posible el procesamiento y el control de usuarios para evaluar los costos y beneficios del sistema propuesto.
- Revisión del sistema actual. Obtener conocimientos sobre las funciones del sistema actual, en caso de existir.

- Definición de los objetivos del sistema. Resumir los requerimientos preliminares del sistema propuesto.
- Identificar alternativas. Identificar posibles soluciones alternativas que satisfagan los requerimiento del usuario.
- Evaluación de costos y beneficios. Comparar costos y beneficios para determinar la viabilidad económica de cada alternativa.
- Preparación del reporte de estudio de factibilidad. Realizar el informe que contenga los resultados del estudio de factibilidad.

### **Definición de requerimientos.**

En esta etapa se definen los requerimientos tanto físicos como funcionales del sistema propuesto. La definición de requerimientos se divide en:

- Conducir entrevistas para definir los requerimientos. Ampliar los datos obtenidos en la etapa anterior.
- Definir los requerimientos del sistema. Define las funciones que realizará el sistema, apoyado en el modelo ambiental, diagramas de flujo de datos, diccionario de datos e interfaces.
- Definir los requerimientos del cliente y del servidor. Define los requerimientos para establecer el ambiente cliente/servidor que soportarán las demandas funcionales y de arquitectura del sistema. Definiendo elementos como:
  - Hardware del cliente.
  - Software del cliente.
  - Hardware del servidor.
  - Software del servidor.
  - Administrador de datos del cliente.
- Definir los requerimientos de servicio de la red. Define los servicios que se le demandarán a la red en las siguientes direcciones:
  - Tiempos de respuesta
  - Capacidad de salida.
  - Confiabilidad y disponibilidad requeridas.
  - Topología y componentes de la red.
- Definir los requerimientos de implementación. Define los requerimientos físicos que deben ser satisfechos en la implementación del sistema propuesto.
- Identificar alternativas. Proponer alternativas practicas para diseñar el sistema, lo que implica:
- Identificar software, redes y hardware que puedan satisfacer los requerimientos.
- Investigar y contactar proveedores.

- Analizar el potencial de cada alternativa.
- Construcción el modelo de requerimientos. Construir un modelo que plasme los requerimientos obtenidos durante la definición de requerimientos.

### **Diseño.**

La fase de diseño se divide en:

#### **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. La etapa de diseño externo se divide en las siguientes actividades:

- Revisión de los requerimientos de definición de datos.
- Definición de subsistemas. Diseñar subsistemas y asignar actividades a cada subsistema.
- Definir las funciones del cliente y del servidor, además de asignar las tareas a los procesadores. Asignación de porciones de la especificación a procesadores adecuados, y a tareas apropiadas dentro de cada procesador.
- Definir las interfaces de los subsistemas. Determinar limitaciones entre los sistemas independientes y los módulos que requerirán de servicios de comunicación.
- Definir el sistema de control de seguridad. Establecer métodos de seguridad física así como del control procedural.
- Evaluación de alternativas de diseño. Comprobar si se satisfizo los requerimientos de: desempeño, datos, ambiente y organizacionales o reglas del negocio.
- Terminación preliminar del modelo de diseño. Construcción de un modelo de diseño preliminar.

#### **Diseño interno.**

El diseño interno establece la estructura para la implementación y pruebas del sistema. Esta fase produce documentos con una especificación precisa de como los elementos funcionales definidos durante el diseño externo pueden ser transformados en una arquitectura física para el sistema. Además contiene especificaciones necesarias para código de programas de aplicación. Esta fase de diseño interno se divide en las siguientes actividades:

- Diseño de la arquitectura del sistema. Diseñar los elementos técnicos que deben ser integrados en el diseño del sistema:
  - Diseñar arquitectura del sistema.
  - Diseñar la arquitectura cliente/servidor.
  - Diseñar arquitectura de recursos compartidos.
  - Diseñar base de datos relacionales.

Diseñar arquitectura de procesamiento distribuido.

- Especificar estructura de datos. Terminar las especificaciones de datos para proporcionar entradas y salidas al sistema, tomando en cuenta las transacciones, presentaciones, reportes y salidas indexadas.
- Diseñar la base de datos. Diseñar los requerimientos de la base de datos para implementar el diseño del sistema.
- Consideraciones específicas del diseño. Definir las consideraciones especiales pertinentes para el diseño del sistema.
- Definir planes de implementación y pruebas. Desarrollar planes de trabajo para la implementación y pruebas del sistema, tomando en cuenta:
  - Componentes reutilizables.
  - Creación de nuevos componentes.
  - Implementar sistema cliente/servidor.
  - Instalar manejador de bases de datos.
  - Crear base de datos.
  - Instalar aplicación.
  - Probar red, aplicación y DBMS.

## Implementación.

El diseño interno establece la estructura para la implementación y pruebas del sistema. Durante esta fase los elementos especificados en el modelo del diseño son transformados en una arquitectura física. El proceso de implementación se divide en las siguientes actividades:

- Revisión del diseño interno de datos.
- Extraer componentes reutilizables. Reducir costos de implementación por utilizar código escrito que sea reutilizable.
- Conducir las pruebas de unidad e integridad. Conducir pruebas de unidad para asegurar que las operaciones individuales y métodos se comparten como se espera y conforme a las especificaciones de diseño, y conducir integración de pruebas.
- Preparación de manual de usuario. Realizar el manual de usuario que contemple: información general, resumen del sistema, además de procedimientos de consulta y actualización.
- Preparación de documentación técnica. Recolectar y organizar la documentación para las operaciones personales para el soporte del sistema.
- Instalar la arquitectura del sistema.
- Instalar el DBMS.
- Creación de la base de datos.
- Instalación de la aplicación.
- Verificar y validar aplicaciones.

## Análisis de sistemas.

Se realiza teniendo en cuenta los siguientes objetivos:

- Identificar las necesidades del cliente.
- Evaluar la viabilidad del sistema.
- Realizar un análisis técnico y económico.
- Asignar funciones al software, al hardware, a la gente, a la base de datos y a otros elementos del sistema.
- Establecer restricciones de coste y tiempo.
- Crear una definición del sistema que sea la base para todo el trabajo posterior de ingeniería.

Gran parte de la labor que desempeñará el analista involucra el modelado del sistema que desea el usuario. Se construyen los modelos para enfatizar ciertas propiedades críticas del sistema, mientras que simultáneamente se desacentúan otros aspectos. Esto permite comunicarnos con el usuario de una manera enfocada, sin distraernos en asuntos y características ajenas al sistema, se pueden hacer cambios en el modelo o desecharlo y hacer uno nuevo en caso de ser necesario. Por esta razón, se hace uso de herramientas de modelado para:

- Concentrarse en las propiedades importantes del sistema y al mismo tiempo restar importancia a otras menores.
- Discutir cambios y correcciones de los requerimientos de usuario, a bajo coste y riesgo mínimo.
- Verificar que el analista comprende correctamente el ambiente del usuario y que lo haya respaldado con información documentada para que los diseñadores de sistemas y los programadores puedan construir el sistema.

## Análisis estructurado.

Es una actividad de construcción de modelos. A medida que fluye la información por un sistema basado en computadora, se transforma. El sistema acepta entradas en una gran variedad de formas, aplica los elementos de hardware, software y humanos para transformar la entrada en salida y produce salidas en una gran variedad de formas. El análisis estructurado es una técnica de modelado del flujo y del contenido de la información. Las herramientas en las que se apoya el análisis estructurado son:

### Herramientas del análisis estructurado.

El Diagrama de Flujo de Datos (DFD) es una técnica gráfica que representa el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida. Se puede usar el DFD para representar un sistema o un software a cualquier nivel de abstracción.

Los elementos de un DFD son:



Una entidad externa, es decir, un elemento del sistema u otro sistema, es quien produce la información a ser transformada por el sistema, es decir, recibe información del sistema y reside fuera de los límites del sistema a ser modelado.



Un círculo representa un proceso o transformación que se aplica a los datos y los cambia de alguna forma y reside dentro de los límites del sistema a modelar.



Las flechas en un DFD deben estar debidamente etiquetadas y representan un elemento de datos o una colección de elementos de datos, indicando la dirección de los datos.



Una línea doble representa un almacén de información que es almacenada y manejada por el sistema, por uno o más procesos.

Un DFD representa el flujo de la información sin representación explícita de la lógica de procesamiento.

La notación básica que se utiliza para desarrollar un DFD no es suficiente en sí misma para describir los requerimientos del sistema, ya que no se dice nada acerca del contenido de los datos implicados en las flechas o en los almacenes de datos. Para ello, es necesario otro componente del análisis estructurado: el Diccionario de Datos (DD), que representa qué información se transforma. Se necesita, además, una narrativa para describir cada proceso o burbuja del DFD. Esta Especificación de Proceso describe la entrada del proceso, el algoritmo que se aplica a la entrada y,

la salida que produce, además de indicar las restricciones y limitaciones impuestas al proceso, y describe cómo se transforma la información.

La modelización del comportamiento es uno de los principios fundamentales de todos los métodos de análisis de requisitos. El Diagrama de Transición de Estados (DTE) representa el comportamiento de un sistema que muestra los estados y los sucesos que hacen que el sistema cambie de estado. Además, el DTE indica qué acciones se llevan a cabo como consecuencia de un determinado cambio de estado.

La notación básica para el análisis estructurado va bien cuando la información que fluye a través de una serie de procesos es relativamente sencilla. Sin embargo, para representar una serie de relaciones complejas entre colecciones de datos, es necesario incluir un componente de modelización de datos.

La modelización de datos se puede realizar mediante el Modelo de Entidad - Relación (E-R), que permite identificar objetos de datos y sus relaciones, usando una notación gráfica. En el contexto del análisis estructurado, el Diagrama Entidad-Relación (DER) proporciona un entendimiento adicional sobre los detalles de los almacenes de datos y complementa la información del DD.

Cada modelo gráfico describe un aspecto distinto del sistema: el DFD resalta las funciones del sistema; el DER resalta las relaciones entre datos; el DTE resalta el comportamiento dependiente del tiempo del sistema. Estos tres aspectos deben ser consistentes y compatibles entre sí.

### Herramientas CASE.

Debido a las características específicas del ciclo de vida y, sobre todo, debido a la gran cantidad de información que se debe generar en cada etapa, comienzan a introducirse herramientas y entornos automatizados de producción, organización, edición y gestión de dicha información, que facilitan, o en muchos casos posibilitan el uso de metodologías formales de forma cómoda y eficiente. Por tanto, la tecnología CASE sustituye el papel y el lápiz por el ordenador para hacer del desarrollo del software un proceso más productivo. El objetivo fundamental de CASE es proporcionar un conjunto de herramientas que automaticen los trabajos de desarrollo y mantenimiento del software. Algunas de las facilidades ofrecidas por este tipo de herramientas son:

- Soporte a la gestión para aumentar el control y la coordinación por parte de la dirección.
- Diseño de diagramas estructurados y creación de especificaciones gráficas.
- Comprobación de errores en etapas iniciales de desarrollo.
- Mantenimiento a nivel especificación.
- Prototipado de las pantallas de usuario.
- Centralización de la información del sistema en un diccionario en el que se pueden almacenar, obtener informes y consultar.
- Verificación de la consistencia y la corrección de las especificaciones del sistema.
- Mantenimiento: re-documentación, reestructuración y análisis del sistema diseñado.
- Soporte de reusabilidad.

- Eliminar errores fácilmente.

Dado que el modelado es una actividad gráfica, se pueden conseguir DFD, DER y DTE más eficientemente y conseguir resultados más estéticos con herramientas CASE. A medida que se refina cada nivel del modelo de flujo, la herramienta CASE construye una jerarquía interna, de forma que cada burbuja "padre" está asociada con sus burbujas "hijas" y, éstas con él.

Aunque la gestión del DD está presente en todas las herramientas CASE, la mayoría también ofrecen otra serie de utilidades como son la normalización de las estructuras de datos, la generación de las bases de datos a partir de las especificaciones de diseño, la generación de código fuente a partir de las especificaciones funcionales, la elaboración de prototipos que permitirán al usuario observar cómo funcionará el sistema una vez implantado, etc. Según la capacidad de la herramienta CASE, podemos hacer la siguiente clasificación:

- CASE Superior: Denominada también Planificación Asistida por Ordenador, se basa en la utilización de software que permita automatizar las tareas de planificación empresarial, por ejemplo, lo que concierne a la gestión de proyectos de desarrollo de Sistemas de Información.
- CASE Intermedia: Se refiere a la automatización del análisis y diseño de Sistemas de Información.
- CASE Inferior: Pretende la asistencia de la computadora en la construcción (programación o generación de código) de los Sistemas de Información.

### Diagramas de contexto.

Un Diagrama de Contexto (DC) es un caso especial de DFD, en donde un proceso único representa el sistema externo. Define la frontera o marco del análisis o lo que es lo mismo, las interfaces del sistema y el resto del universo. Destaca varias características importantes del sistema:

- Aquellos agentes externos que se comunican con el sistema, que a veces se conocen como terminadores.
- Los datos que el sistema recibe del exterior y que se deben procesar de alguna forma.
- Los datos producidos por el sistema y que se mandan al mundo exterior.
- La frontera entre el sistema y el resto del mundo.
- Los almacenes de datos que el sistema comparte con los agentes externos. Estos almacenes se crean fuera del sistema para su uso, o bien son creados en él y usados fuera.

Los elementos de un DC son los mismos que los de un DFD. Por tanto:

Sistema:



Representado por una burbuja. Indica una parte del sistema que transforma entradas en salidas. En este caso, representa el sistema entero. Trabaja sobre los datos. Debe tener un nombre claro, compuesto por un verbo de acción más un sustantivo.

Entidad / Agente Externo



Define la frontera del sistema. Son los emisores y/o receptores de los flujos de información. Deben tener un nombre claro. Aparecen exclusivamente en el nivel 0 o DC. No se puede cambiar el contenido de una entidad externa. Cualquier relación entre entidades externas no debe aparecer en el diagrama.

Son la conexión con el mundo exterior.

Flujo de Datos



Representa los datos del sistema. Es una flecha que indica el movimiento de los datos. Define origen, dirección y destino de los datos. Une al emisor y al receptor de los datos. Conecta al resto de los componentes del DFD. Se puede bifurcar o converger. Deben ser claros, concisos y fácil de definir. Se debe evitar flujos diferentes con el mismo nombre.

### Modelo de Datos.

El modelado de datos es el proceso de ordenar los datos y sus relaciones con el fin de desarrollar el modelo lógico de la base de datos. Los objetivos que se pretenden son: conseguir estructuras de datos flexibles, estables y normalizadas y, separar procesos de los datos.

La modelización de los datos procesados por un Sistema de información se realiza en diferentes niveles consecutivos de abstracción:

**Nivel Conceptual:** a este nivel se realiza una formalización de los datos almacenados en el sistema (los de los almacenes del DFD) mediante una descripción de las entidades (objetos materiales o inmateriales del sistema), los atributos (propiedades) de estas entidades y las posibles relaciones entre ellas. Este modelo se realiza durante la fase de análisis del sistema.

**Nivel Lógico:** mientras que el modelo conceptual es independiente del tipo de software de gestión de información, en el nivel lógico se realiza la adaptación de aquel modelo (ya validado) al tipo de Sistema de Gestión de Base de Datos (relacional, jerárquico o en red) que se vaya a utilizar. Al final se obtiene un modelo lógico de registros que representa la estructura de los datos (a nivel de registros lógicos) en dicho sistema. Este modelo se realiza durante la fase de diseño del sistema, se suele completar con información adicional sobre el volumen de los datos y la forma de acceso a los mismos.

**Nivel Físico:** a este nivel se debe determinar cómo se organiza físicamente el almacenamiento de los datos en ficheros. Todos estos detalles se pueden ignorar, ya que son competencia del Sistema de Gestión de Base de Datos que se utilice.

Con la modelización de los datos, además de especificar las características de la información, se pretende conseguir la simplificación de las estructuras definidas en el DD del proyecto, buscando y eliminando los elementos de datos (campos de registro) redundantes, para lo cual se produce una reorganización de estas estructuras para eliminar repeticiones, proceso que se conoce con el nombre de normalización.

### Modelo conceptual de datos.

Es una representación abstracta de los datos utilizados por un sistema. Este modelo tiene un alto nivel de independencia, ya que:

- No presupone ninguna hipótesis sobre la utilización que se hace de ellos.
- No tiene en cuenta la localización de los datos en los distintos soportes.
- Traduce las elecciones u opciones de gestión fundamentales, en términos de objetos y relaciones.
- No hay razón para que se modifique su diseño, a menos que se produzca un cambio radical de los requisitos del sistema.

### Definiciones.

Entidad o Individuo: es un objeto abstracto o concreto del sistema, con existencia propia y fácilmente identificable. Es una clase de objetos sobre los que se quiere guardar información. Puede ser de tres tipos:

- Fundamental: Cuando es de interés por sí misma, independiente de cualquier relación.
- Atributiva: Cuando sirve para completar la descripción de otro tipo de entidad.
- Asociativa: Su razón de existir es relacionar otras entidades.

Cada miembro de la entidad puede identificarse de manera única por algún medio. El sistema no puede operar sin tener acceso a estos miembros. Cada uno de los miembros puede describirse con uno o más datos.

Atributo o Propiedad: es el mínimo elemento lógico de información que se puede encontrar en una entidad o asociación de entidades. El Atributo Clave de una entidad es aquel que puede ser utilizado para identificar cada ocurrencia de la entidad.

Relación: es una asociación entre entidades, cuya existencia está condicionada por las entidades que relaciona. Tiene asociada una dimensión o número de entidades que relaciona, pudiendo ser 1, 2,.... Para una relación binaria entre A y B, pueden darse de tres casos:

- uno a uno (1-1): en la cual una ocurrencia de A no está en relación más que con una ocurrencia de B y, cada ocurrencia de B no está en relación más que con una ocurrencia de A.
- uno a muchos (1-n): en la cual una ocurrencia de A está en relación con una o muchas ocurrencias de B y, cada ocurrencia de B no está en relación más que con una ocurrencia de A.
- muchos a muchos (n-n): en la cual una ocurrencia de A está en relación con una o muchas ocurrencias de B y, cada ocurrencia de B está en relación con una o muchas ocurrencias de A.

La relación representa la memoria del sistema, ya que representa algo que debe ser recordado por el sistema. No se muestran las relaciones que se pueden calcular o derivar.

La cardinalidad es el número de objetos que participan en la relación. Es el número de ocurrencias de cada tipo de entidad que intervienen o pueden intervenir en una relación.

La cardinalidad mínima es el número mínimo de veces que una ocurrencia de una entidad está implicada en una ocurrencia de la relación. El valor 0 indica que no participa en la relación o lo que es lo mismo, que una ocurrencia de una entidad puede existir sin estar implicada en ninguna ocurrencia de la relación. El valor 1 que participa una sola ocurrencia o lo que es lo mismo, que no puede existir sin estar implicada en una o más ocurrencias de la relación.

La cardinalidad máxima es el número máximo de veces que una ocurrencia de una entidad está implicada en una ocurrencia de la relación. El valor 1 indica que participa en la relación con una sola ocurrencia, o lo que es lo mismo, que no puede estar implicada más que en un máximo de una ocurrencia de la relación. El valor n que participa con muchas ocurrencias o lo que es lo mismo, que puede estar implicada en n ocurrencias de la relación.

Una vez realizado el primer modelo de datos, se comprueba que cada entidad es determinada sin ambigüedad por su clave y que todas las entidades asociadas a cada relación son necesarias para definir cada atributo de la relación. También se procede a eliminar redundancias mediante la normalización.

### **Modelo lógico de datos.**

El Modelo Conceptual de Datos es independiente del software de gestión de bases de datos utilizado. Para proceder al diseño de los ficheros y de las bases de datos del sistema, se debe convertir previamente el modelo conceptual que incluía tipos de entidades y relaciones con atributos asociados, en un modelo lógico que únicamente considere tipos de registros compuestos por campos de datos. Al Modelo Lógico de Datos normalmente se le suele llamar Diagrama de Estructura de Datos (DED) o Diagrama de Bachman.

Existen tres tipos de SGBD. Basados en una base de datos relacional, en el que se almacenan los datos como un conjunto de tablas o relaciones. Basados en una base de datos en red, en el que la estructura de los registros establece una relación subordinada "padre-hijo" entre ellos. Basados en una base de datos jerárquica, similar a la anterior, con la diferencia que cada registro sólo puede tener un padre. Nos limitaremos a las basadas en bases de datos relacionales, ya que son las más utilizadas y poseen la ventaja de que la conversión del modelo conceptual es directa, sin más que considerar las entidades como registros y los atributos como campos de éstos.

### **Diagrama de transición de estados**

El diagrama de transición de estados, enfatiza el comportamiento del sistema dependiente del tiempo. Se utiliza fundamentalmente para sistemas de tiempo real, en los que se manejan fuentes externas de datos de alta velocidad y a los que se debe proporcionar alguna respuesta lo suficientemente rápida para manejar el ambiente externo. En este tipo de modelado es importante qué sucede cuando ocurre tal evento.

Los principales componentes de este tipo de diagrama son los estados y, las flechas o cambios de estado.

Los estados se representan con un rectángulo al que se le ha añadido el nombre del estado que simbolizan. Se define como un conjunto de circunstancias o atributos que caracterizan a un objeto

en un tiempo dado: forma de ser, condición,... representa algún comportamiento del sistema que es observable y que perdura durante algún tiempo finito.

El DTE se utiliza para desarrollar un modelo de cómo se comportaría el sistema si hubiera tecnología perfecta. Un aspecto de esta tecnología perfecta sería que el sistema trabajara de manera infinitamente rápida. Así cualquier estado observable en el que el sistema se puede encontrar corresponde a periodos en los que el sistema está esperando que algo ocurra en el ambiente exterior o está esperando a que alguna actividad que se está desarrollando en ese momento en el ambiente cambie a otra.

**ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA**

---

## **APENDICE B**

### **Diseño Orientado a Objetos.**

Un modelo abstracto ó mental es una vista simplificada de como trabajan las cosas y como interactuan, este proceso de construir modelos es similar al diseño de software.

Este proceso de abstracción es psicológicamente natural y necesario para poder entender el mundo. Además es indispensable para diseñar un buen software y es de gran ayuda para que los analistas diseñen software de propósito general.

La abstracción, por tanto se define como la extracción de las propiedades esenciales de un concepto, esta abstracción de datos permite el despreocuparse por los detalles no esenciales.

La Programación Orientada a Objetos es una técnica de programación que surgió a principios de la década de los setenta, esta técnico o estilo de programación utiliza a los objetos como bloque esencial de construcción.

Por tanto dentro del Diseño Orientado a Objetos (DOO) se descomponen a los sistemas en objetos que son los componentes básicos del diseño.

Antes de entrar de lleno al DOO es necesario conocer algunos términos como los siguientes:

- Objeto: es una unidad formada por dos partes: datos y funciones, las cuales se encuentran unidas (encapsuladas) formando una sola entidad.
- Todo objeto tiene una Interfaz pública con la cual se presenta ante otros objetos y una Representación privada que consiste en lo que puede realizar y a quién pueden llamar.
- Mensaje: Es la forma mediante la cual un objeto accesa a otro. Este mensaje esta formado por el nombre, operación y argumentos requeridos.
- Método: Algoritmo paso a paso de cómo será ejecutada una operación. El método forma parte de la representación privada de un objeto.
- Forma: Es el nombre del método, tipo de parámetros y tipo de objetos que regresa el método, es decir, es una especificación formal de las entradas y salidas de un método.
- Clases: Forma de agrupar a objetos de la misma especie. Es una forma de representación genérica de objetos similares.
- Instancia: Objeto que pertenece a una clase
- Herencia: Es la propiedad de que una clase sea definida a partir de otra u otras clases.
- Poliformismo: Es la habilidad de que dos o más clases respondan a un mismo mensaje de una manera propia.
- Subclase: Clase que nace a partir de la herencia de otra clase.

El DOO es el proceso mediante el cual los requerimientos del software, son convertidos en una detallada especificación de objetos, en la cual incluye una descripción completa de los roles y responsabilidades de los mismos, además de especificar la forma mediante la cual se comunican con otros objetos.

Cabe mencionar que del diseño no saldrá un producto final, pues el diseño solo es la primera fase del Ciclo de Vida del Software.

Tradicionalmente la mayor porción del tiempo empleado para el desarrollo de un sistema lo ocupan tanto el diseño como la implementación. No así en el DOO en donde el software es:

- Reutilizado: es decir, parte del código es utilizado por otro sistema para el cual no fue diseñado.
- Redefinido: cuando el código es usado como base para otro software.
- Probado: cuando el comportamiento del software es determinado por otro, conforme las especificaciones del software .
- Mantenido: cuando se encuentran errores y son corregidos.
- Extendido: cuando se le adiciona una nueva función al software.

Debido a esto el ciclo de vida del software orientado a objetos, es una espiral donde la fracción más grande es ocupada por la fase de diseño.

Para obtener un buen diseño se debe de realizar un análisis preliminar detallado, el cual consiste de los dos siguientes pasos:

Analizar los factores comunes de las responsabilidades para crear una jerarquía de clases, y  
 Analizar cuidadosamente la colaboración entre los objetos.

El proceso de Diseño Orientado a Objetos comienza con una exploración inicial, en donde se buscan las clases para encontrar un esquema donde se muestre de manera natural y razonable la forma de observar al sistema de manera abstracta.

El DOO consiste de los siguientes pasos:

Encontrar las clases del sistema

Determinar operaciones y responsabilidades de cada clase.

Determinar la manera en cómo los objetos colaboran entre sí.

Estos pasos producen la siguiente información:

Una lista de las clases de la aplicación.

Una descripción de las operaciones de cada clase y la responsabilidad que tienen

Una descripción de la colaboración entre los objetos.

Con la información antes mencionada es posible comenzar la fase de diseño. Primero se debe de examinar la relación de herencia entre las clases, tomando en cuenta el comportamiento de cada una de ellas. Después se debe analizar la colaboración entre las clases.

Esta colaboración entre objetos es vista como una manera de interacción, en donde un objeto requiere un servicio de otro objeto. El objeto que requiere el servicio es conocido como *Cliente* y el objeto que recibe el requerimiento y provee el servicio es el *Servidor*. La manera de como el cliente interactúa con el servidor es mediante un *contrato*, por tanto, el contrato es un requerimiento que el cliente hace al servidor.

## Determinación de Clases.

Dentro de la primer paso del DOO se deben de tomar en cuenta aspectos como los siguientes:

Búsqueda de los objetos que participaran dentro del sistema.

Determinar que metas a cumplir por cada objeto.

Establecer lo que tendrán que cumplir los objetos.

Comportamiento detallada de cada objeto.

Estructura abstracta del sistema.

Para ello es necesario tener una especificación de requerimientos, este documento proporcionara las metas a lograr con el diseño, además de proporcionar bases para saber que entradas se requerirán y decir que respuestas se obtendrían.

Una vez adquirido este documento, es necesario cambiar todo lo plural en singular, además de realizar una lista de los temas tratando de establecer 2 posibles categorías: clases obvias y clases candidato. Cabe mencionar que no todas las clases de esta lista llegan a aparecer en el diseño final, ya que a través del proceso pueden ser desechadas.

También es necesario a partir de este momento el comenzar a hablar de las cosas con el lenguaje de DOO, y de ser posible formular el nombre de las clases de acuerdo al propósito para que el cual serán empleadas, enfatizando en lo elemental y eliminando lo irrelevante.

Para encontrar las frases y las clases candidato hay que tomar en cuenta criterios como:

Modelo físico de los objetos.

Modelo conceptual de las entidades.

Construcción de un vocabulario común, es decir, si se usa más de una palabra para nombrar un objeto, es necesario solo dejar aquellos términos que sean principales.

Tener cuidado al usar adjetivos, debido a que un adjetivo puede significar varias cosas.

Modelo de categoría de clases, esto para clasificar a cada una de ellas de acuerdo a los objetos que contienen.

Modelo de interfaces.

Modelo de los valores de los atributos de los objetos.

A partir de los anterior de obtiene una lista tentativa de las clases para la aplicación, las cuales pueden existir hasta el final o ser eliminadas en el transcurso del proceso.

Una vez teniendo esta lista de clases, se deben de crear categorías de clases con características similares, para comenzar a enumerar los atributos de estas categorías.

Esta clasificación por categorías es importante ya que a partir de ella, se puede observar si faltan clases por determinar.

## Determinación de Responsabilidades.

Dentro de este punto, se debe de establecer la lista de requerimientos de los objetos, para ello es necesario saber detalladamente que metas deben de cumplir los objetos dentro del sistema.

Las responsabilidades son una manera de repartir el trabajo, dentro del sistema de acuerdo a las metas a cumplir. Estas responsabilidades son los servicios que puede realizar sobre un objeto. Estos servicios o contratos pueden realizar ya sea una varias acciones o regresar algún tipo de información.

Para poder identificar las responsabilidades, es necesario recurrir al documento de especificación de requerimientos y hacer una lista de todos los verbos que al fin de cuentas serán las acciones a realizar, además de ocupar la lista de las clases candidato que ya se obtuvo en el punto anterior, de aquí se podrán observar algunas de las necesidades que tendrán las clases, dependiendo del rol que jueguen y que en este momento se convierten en responsabilidades.

Las responsabilidades de los objetos se otorgan de acuerdo a dos criterios:

El conocimiento que tiene cada uno de ellos, y

Las acciones que pueden realizar

Esto es cada objeto debe de tener conocimiento de cierta parte del sistema, debe de realizar ciertas acciones y debe de saber como relacionarse con otras instancias.

## **Determinación de la Colaboración.**

Para determinar la colaboración entre objetos es necesario establecer que rol jugara cada uno de ellos, es decir , establecer si serán clientes o servidores. Estos roles no son un atributo para las instancias. Estos conceptos de cliente y servidor solo son requeridos para saber en que momento un objeto es cliente y cuando se convierte en servidor, además es usado para establecer los contratos los cuales posteriormente se convierten en responsabilidades y además para determinar la colaboración entre cliente y servidor de acuerdo a la lista de contratos.

El analizar la interacción entre clases es la forma de determinar la colaboración entre clases.

Es necesario recordar que una colaboración esta asociada con una responsabilidad pero no funciona de forma inversa, además esta colaboración parte del rol que juegan las clases.

---

## APENDICE C

### Bases de datos.

Una base de datos es una colección integrada de datos, interrelacionados entre si, almacenados sin redundancia perjudicial para servir a una o más aplicaciones. El propósito de usar una base de datos, es el de mantener un control centralizado de los datos, además de evitar duplicados o pérdida de los datos.

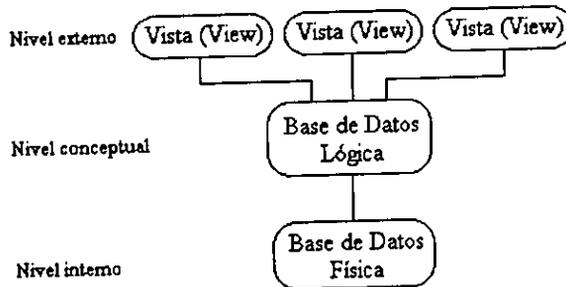
Entre las principales ventajas de utilizar una base de datos podemos citar:

- Reducción de la redundancia. Evitar desperdicio del espacio de almacenamiento al eliminar la existencia de datos duplicados.
- Evitar la inconsistencia. Garantizar una misma salida para peticiones similares en un mismo momento.

- Compartir datos. Que las diferentes aplicaciones operen con los datos almacenados en la base.
- Cumplir con las normas establecidas. Garantizar que se cumplan las reglas del negocio.
- Seguridad. Define controles de autorización para ser aplicados al momento de tener acceso a la base, realizar modificaciones o pérdida de datos.
- Integridad. Verificar que las reglas de validación se cumplan .
- Recuperación. Capacidad de restaurar la integridad y consistencia de la base, después de una falla.
- Concurrencia. Control de múltiples accesos por diferentes usuarios a la misma información.
- Eficiencia (performance). Tiempo de respuesta de una aplicación en el momento de acceder a la base.

## Arquitectura de una Base de Datos

La estructura o arquitectura de una base de datos esta formada por tres niveles generales: interno, conceptual y externo.



### Arquitectura de una Base de Datos.

- Nivel interno: Es la estructura básica, la cual es utilizada por el administrador de la base de datos (DBA), es lo más cercano al almacenamiento físico, es decir, la manera de como los datos son almacenados en realidad.
- Nivel externo: Esta formado por la interfaz de la aplicación, es lo que los usuarios tanto finales como ocasionales observan a través de los objetos virtuales llamadas vistas (view).
- Nivel conceptual: Es el mediador entre los otros dos niveles, es la forma en como los programadores observan la base de datos de manera lógica.
- Cada usuario de la base de datos se comunica con esta de diversas formas.
- El DBA es la persona encargada de la base de datos, conoce a fondo el funcionamiento de esta.

- El Programador es la persona que crea programas para explotar la base utilizando SQL (Lenguaje Estructurado de Consultas), además de lenguajes de 3a y 4a generación (3GL o 4GL).
- Usuario final u ocasional, utiliza SQL para realizar consultas a la base y utiliza los programas creados por los programadores.
- Usuarios ingenuos son aquellos que solo saben usar los programas de aplicación.

Además de los lenguajes mencionados, los usuarios hacen uso de un sublenguaje de datos, el cual es combinación de dos lenguajes:

- Lenguaje de Definición de Datos (DDL). Es el lenguaje que permite la definición o descripción de los objetos de la base de datos.
- Lenguaje de Manipulación de Datos (DML). Lenguaje que apoya al manejo o procesamiento de los objetos de la base.

## Sistema de Bases de Datos.

Un sistema de base de datos puede definirse como un sistema de mantenimiento de registros en computadora, es decir, un sistema cuyo propósito general es registrar y mantener la información.



Diagrama de un Sistema de Bases de Datos.

Un sistema de base de datos incluye cuatro componentes principales: datos, hardware software y usuarios.

El hardware se compone de los volúmenes de almacenamiento secundario donde reside la base de datos, junto con los dispositivos asociados como las unidades de control, los canales, computadora central, además del procesador de entrada/salida.

Entre la base de datos física y los usuarios del sistema existe un nivel de software, que recibe el nombre de sistema de administración de base de datos (DBMS). El DBMS es la parte más importante del software de un sistema de base de datos.

Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

Las principales funciones de un DBMS son:

- Crear y organizar la base de datos.
- Mantiene la integridad.
- Centraliza la información.

- Facilita la explotación de la información.
- Registra el uso de la base.
- Consolida la información.

El DBMS interpreta y procesa las peticiones del usuario para recobrar información de la base. El sistema administrador interactúa con el sistema operativo mediante rutinas de bajo nivel, para extraer datos y depositarlos en el área de trabajo del DBMS, no se extrae registro por registro, sino bloques completos.

La petición del programa de aplicación es atendida filtrando la información necesaria y depositada en el área de trabajo del programa de aplicación.

## Diccionario de Datos.

El Diccionario de Datos es una parte importante dentro del sistema de base de datos. El Diccionario contiene información de la información, es decir, contiene una descripción detallada de toda la base.

“El Diccionario de Datos es una base de datos por propio derecho, es una base de datos que contiene datos acerca de los datos, es decir, descripciones de otros objetos del sistema y no tan sólo datos en bruto” [DATE]

Para la descripción de los diferentes datos se utilizan los siguiente símbolos:

=	está compuesto de
+	para unir campos elementales
()	campos opcionales
[ ]	selección de una, entre varias opciones
{ }	Iteraciones
**	Comentarios
@	Identificación o clave de un almacenamiento
	Separación de opciones alternativas

La documentación proporcionada por el Diccionario de Datos es muy valiosa para todos los usuarios de la base. Este se puede considerar como un mapa de carreteras que guía a los usuarios para acceder información dentro de una base grande de datos.

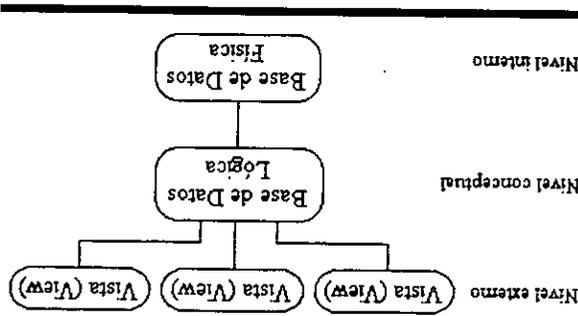
El Diccionario de Datos debe incluir información acerca de la base como:

- Descripción externa, conceptual e interna de la base.
- Descripción de entidades y atributos.
- Sinónimos, homónimos y código de autorización y seguridad.

Hay dos tipos de Diccionario de Datos: integrado y stand-alone.

- Nivel interno: Es la estructura básica, la cual es utilizada por el administrador de la base de datos (DBA), es lo más cercano al almacenamiento físico, es decir, la manera de como los datos son almacenados en realidad.
- Nivel externo: Esta formado por la interfaz de la aplicación, es lo que los usuarios tanto finales como ocasionales observan a través de los objetos virtuales llamadas vistas (view).
- Nivel conceptual: Es el mediador entre los otros dos niveles, es la forma en como los programadores observan la base de datos de manera lógica.
- Cada usuario de la base de datos se comunica con esta de diversas formas.
- El DBA es la persona encargada de la base de datos, conoce a fondo el funcionamiento de esta.

Arquitectura de una Base de Datos.



La estructura o arquitectura de una base de datos esta formada por tres niveles generales: interno, conceptual y externo.

### Arquitectura de una Base de Datos

- Compartir datos. Que las diferentes aplicaciones operen con los datos almacenados en la base.
- Cumplir con las normas establecidas. Garantizar que se cumplan las reglas del negocio.
- Seguridad. Define controles de autorización para ser aplicados al momento de tener acceso a la base, realizar modificaciones o perdida de datos.
- Integridad. Verificar que las reglas de validación se cumplan.
- Recuperación. Capacidad de restaurar la integridad y consistencia de la base, después de una falla.
- Concurrencia. Control de múltiples accesos por diferentes usuarios a la misma información.
- Eficiencia (performance). Tiempo de respuesta de una aplicación en el momento de acceder a la base.

## Modelos de bases de datos.

Las bases de datos se clasifican de acuerdo al modelo. Existen básicamente tres modelos: red, jerárquico y relacional.

### Modelo Jerárquico.

Este modelo consiste en un conjunto de registros que se conectan entre sí por medio de ligas, usa tres estructuras de árbol para representar arreglos lógicos.

Los datos y las relaciones entre ellos se representan por medio de registros y ligas respectivamente.

Registros: Conjunto de campos que contiene un solo valor. El contenido de un registro específico puede repetirse en varios lugares del árbol.

Liga: Asociación entre dos registros.

Raíz: Registro padre, y solo debe existir uno.

Característica:

- Los datos se representan en una estructura de árbol.
- Utiliza el procesamiento top-down.
- El único acceso se realiza por la raíz.
- Un registro padre puede relacionarse con 0 ó más hijos.

Desventajas:

- Dificultad al modelar relaciones de muchos a muchos.
- Difícil reorganización de borrado e inserción.

### Modelo de red o reticular.

Este modelo consiste de un conjunto de registros conectados entre sí por medio de ligas (links). Su estructura es básicamente una estructura de árbol de dos niveles que conecta las ocurrencias de dos registros.

Los datos se expresan por medio de registros y las relaciones por medio de ligas.

La base de datos es un conjunto de registros ligados formando datos interceptados.

Características:

- Utiliza el procesamiento multidireccional.
- No tiene dificultad al modelar cualquier tipo de relación.
- Propietario y miembro. El registro padre de un conjunto se llama propietario del conjunto, mientras que el registro hijo se llama miembro.
- Parentesco múltiple. Un registro de tipo miembro puede asociarse con más de un tipo propietario.

Desventajas:

- Difícil de mantener.
- Difícil de realizar nuevas relaciones.
- Desperdicio de recursos.

### **Modelo relacional.**

Una base de datos relacional esta constituida por un conjunto de archivos bidimensionales (tablas), las cuales se interconectan por medio de relaciones.

Características:

- La idea de tablas es intuitiva y fácil de manejar.
- Una tabla puede relacionarse con otra mediante un atributo en común.
- El orden de las columnas es arbitrario.
- Los proceso de modificación, borrado e inserción son fáciles de realizar.

Desventajas:

- Una base de datos mal diseñada provoca redundancia.
- Acceso lento basado en el tipo de índice y tamaño.

### **Normalización.**

La primera fase para el diseño de una base de datos es llegar a agrupar los datos relacionados dentro de relaciones, mediante distintas interpretaciones de las reglas semántica, el procedimiento de normalización se repite y se requiere de diversas pruebas con distintas agrupaciones antes de llegar al diseño satisfactorio.

El proceso de normalización descompone una relación inicial universal conteniendo todos los datos del diseño de la base en varias relaciones más pequeñas, estas relaciones derivadas pueden reconstruir cualquier información de las relaciones originales o intermedias. Este proceso es un sistema seguro para el diseño de la base de datos.

Con el procedimiento de normalización, un archivo se representa como una tabla de dos dimensiones llamada relación, la forma más simple para representar datos es mediante una tabla. El enfoque relacional generalmente se usa en el proceso de normalización para llegar a modelos de base de datos, sin embargo, un modelo así se puede modificar para su implantación con un enfoque jerárquico, de red o relacional.

Las relaciones normalizadas se agrupan en categorías llamadas formas normales, siendo cada nivel una descomposición más completa de una relación que la del nivel anterior. La meta final de proceso de normalización es la agrupación de todos los atributos de una base de datos en relaciones adecuadas para que la base se pueda almacenar con el mínimo de datos redundantes.

El propósito de este proceso es quitar las cualidades indeseables de una relación que puedan causar anomalías en el almacenamiento cuando se efectúen operaciones de actualización en la base de datos. El proceso empieza con la combinación de todos los datos de la base en una relación la que a su vez se descompone en dos o más relaciones más pequeñas. Este proceso es sucesivo hasta haber aplicado las formas normales.

**Primera Forma Normal (1FN).**

Una estructura está en la primera forma normal si todos los campos en cada registro contienen un solo valor tomado de sus dominios respectivos.

El dominio de un campo es el rango de valores continuos o discretos permitidos para el campo.

**Segunda Forma Normal (2FN).**

Una estructura está en 2FN, si está en 1FN y cada atributo no llave es funcionalmente dependiente de su llave primaria.

Un atributo A es funcionalmente dependiente del atributo B si el valor de A está determinado por el valor de B.

**Tercera Forma Normal (3FN).**

Una estructura está en 3FN si está en 2FN y no existe dependencia transitiva en ninguno de sus atributos no primos.

Los atributos no primos son aquellos que no forman parte de la llave primaria.

Dado un atributo A que depende de B ( $B \rightarrow A$ ) y un atributo B depende de C ( $C \rightarrow B$ ). Se dice que hay dependencia transitiva si el atributo A depende de C ( $C \rightarrow A$ ).

En realidad existe hasta la quinta forma normal (5FN) pero con aplicar hasta la 3FN, se considera que se ha realizado un buen diseño de la base de datos.

## APENDICE D

---

### Arquitectura Cliente/Servidor.

Los sistemas cliente/servidor (C/S) que evolucionado en conjunción con los avances de la computación, con unas nuevas tecnologías de almacenamiento, con una mejora de las comunicaciones por red y con una mejora de la tecnología de bases de datos.

El objetivo de este tema es presentar una breve visión general de los sistemas cliente/servidor haciendo especial hincapié en los problemas especiales de ingeniería del software que es preciso abordar cuando se presta apoyo a estos sistemas, y también cuando estos se analizan, diseñan y comprueban.

#### Estructura de los sistemas cliente/servidor.

La integración de las tecnologías de hardware, de software, de bases de datos y de redes contribuyen conjuntamente a las arquitecturas distribuidas y cooperativas. Estas constan de un sistema raíz, que típicamente será una gran computadora, que actúa como depósito de los datos corporativos. El sistema raíz está conectado con servidores (que típicamente son estaciones de

trabajo potentes, o PC) y que poseen un doble papel. Los servidores actúan para actualizar y solicitar los datos corporativos mantenidos por el sistema raíz. Además mantienen sistemas departamentales locales y desempeñan un papel clave al poner en red los PC de nivel de usuario a través de una red de área local (LAN).

En una estructura C/S, la computadora que reside por encima de otra se denomina servidor, y las computadoras de nivel inferior se denominan clientes. Los clientes solicitan servicios, y el servidor los proporciona. En el contexto de esta arquitectura se pueden llevar a cabo un cierto número de implementaciones distintas.



#### Arquitectura Cliente/Servidor.

**Servidores de archivos.** El cliente solicita registros específicos de un archivo. El servidor transmite estos registros al cliente a través de la red.

**Servidores de bases de datos.** El cliente envía solicitudes en lenguaje de consulta estructurado (SQL) al servidor. Éstas se transmiten como mensajes a través de la red. El servidor procesa la solicitud SQL y halla la información solicitada, pasando únicamente los resultados al cliente.

**Servidores de transacciones.** El cliente envía una solicitud que invoca procedimientos remotos en el centro servidor. Los procedimientos remotos pueden ser un conjunto de sentencias SQL. Se produce una transacción cuando una solicitud da lugar a la ejecución de procedimientos remotos y a la transmisión del resultado devuelto al cliente.

**Servidores de grupos de trabajo.** Cuando el servidor proporciona un conjunto de aplicaciones que hacen posible la comunicación entre clientes (y entre las personas que los usan) mediante el uso de texto, imágenes, boletines electrónicos, vídeo y otras representaciones, existe una arquitectura de grupos de trabajo.

### Componentes de software para sistemas C/S.

En lugar de visualizar el software como una aplicación que deberá implementarse en una máquina, el software que es adecuado para una arquitectura cliente/servidor posee varios componentes distintos que se pueden asociar al cliente o al servidor, o se pueden distribuir entre ambas máquinas:

**Componente de interacción con el usuario y presentación.** Este componente implementa todas las funciones que típicamente se asocian a una interfaz gráfica de usuario (GUI).

**Componente de aplicación.** Este componente implementa los requisitos definidos por la aplicación en el contexto del dominio en el cual funciona la aplicación. Por ejemplo, una aplicación de gestión podría producir toda una gama de informes impresos basados en entradas numéricas, cálculos, información de una base de datos, y otros aspectos. Una aplicación para trabajo en grupo podría proporcionar las capacidades adecuadas para hacer posible la comunicación mediante un boletín electrónico o correo electrónico. En ambos casos, el software de aplicación se puede descomponer de tal modo que alguno de los componentes residan en el cliente y otros residan en el servidor.

Gestión de bases de datos. Este componente lleva a cabo la manipulación y gestión de datos requerida por una aplicación. La manipulación y gestión de datos puede ser tan sencilla como la transferencia de un registro, o tan compleja como el procesamiento de sofisticadas transacciones SQL.

Además de estos componentes, existe otro bloque de construcción del software, que suele denominarse software intermedio en todos los sistemas C/S. El software intermedio consta de elementos de software que existen tanto en el cliente como en el servidor, e incluye elementos de sistemas operativos en red así como un software de aplicación especializado que presta su apoyo a las aplicaciones específicas de bases de datos, a estándares de distribución de solicitudes de objetos, a tecnologías de trabajo en grupo, a gestión de comunicaciones, y a otras características que facilitan la conexión cliente/servidor.

## Distribución de componentes del software.

Una vez que se han determinado los requisitos básicos de una aplicación cliente/servidor, el ingeniero del software debe decidir la forma en que distribuirá los componentes de software, entre el cliente y el servidor. Cuando la mayor parte de la funcionalidad asociada a cada uno de los tres componentes se le asocia al servidor, se ha creado un diseño de servidor principal. A la inversa, cuando el cliente implementa la mayor parte de los componentes de interacción/presentación con el usuario, de aplicación y de bases de datos, se tiene un diseño de cliente principal.

Los clientes principales suelen encontrarse cuando se implementan arquitecturas de servidor de archivo y de servidor de base de datos. En este caso, el servidor proporciona apoyo para la gestión de datos, pero todo el software de aplicación y de GUI reside en el cliente. Los servidores principales son los que suelen diseñarse cuando se implementan sistemas de transacciones y de trabajo en grupo. El servidor proporciona el apoyo de la aplicación necesario para responder a transacciones y comunicaciones que provengan de los clientes. El software de cliente se centra en la gestión de GUI y de comunicaciones.

Para la asignación de componentes de software se definen cinco configuraciones diferentes (enumeradas de menor a mayor sofisticación tecnológica):

**Presentación distribuida.** En este enfoque cliente/servidor, la lógica de la base de datos y la lógica de la aplicación permanecen en el servidor, típicamente en un computador central. El servidor contiene también la lógica para preparar información de pantalla, empleando un software especial basado en PC para transformar la información de pantalla basada en caracteres que se transmite desde el servidor en una presentación GUI en un PC.

**Presentación remota.** En esta extensión del enfoque de presentación distribuida, la lógica primaria de la base de datos y de la aplicación permanecen en el servidor, y los datos enviados por el servidor serán utilizados por el cliente para preparar la presentación del mismo.

**Lógica distribuida.** Se asignan al cliente todas las tareas de presentación del usuario y también los procesos asociados a la introducción de datos tales como la validación de nivel de campo, la formulación de consultas de servidor, y las solicitudes de informaciones de actualizaciones del servidor. Se asignan al servidor las tareas de gestión de las bases de datos, y los procesos para las consultas del cliente, para actualizaciones de archivos del servidor, para control de versión de clientes, y para aplicaciones de ámbito general de la empresa.

**Gestión de datos remota.** Las aplicaciones del servidor crean una nueva fuente de datos dando formato a los datos que se han extraído de algún otro lugar. Las aplicaciones asignadas al cliente se

utilizan para explotar los nuevos datos a los que se ha dado formato mediante el servidor. En esta categoría se incluyen los sistemas de apoyo de decisiones.

Bases de datos distribuidas. Los datos de que consta la base de datos se distribuyen entre múltiples clientes y servidores. Consiguientemente, el cliente debe admitir componentes de software de gestión de datos así como componentes de aplicación y de GUI.

### Líneas generales para distribuir los componentes.

Aun cuando no existen reglas absolutas que describan la distribución de componentes de aplicaciones entre el cliente y el servidor, suelen seguirse las siguientes líneas generales:

- El componente de presentación/interacción suele ubicarse en el cliente. La disponibilidad de entornos basados en ventanas y de la potencia de cómputo necesaria para una interfaz gráfica de usuario hace que este enfoque sea eficiente en términos de coste.
- Si es necesario compartir la base de datos entre múltiples usuarios conectados a través de la LAN, entonces la base de datos suele ubicarse en el servidor. El sistema de gestión de la base de datos y la capacidad de acceso a la base de datos también se asignan al servidor, junto con la base de datos física.
- Los datos estáticos que se utilicen como referencia deberían asignarse al cliente. Esto sitúa los datos más próximos al usuario que tiene necesidad de ellos, y minimiza un tráfico de red innecesario y la carga del servidor.

El resto de los componentes de aplicación se distribuye entre cliente y servidor basándose en la distribución que optimice las configuraciones de cliente y servidor y de la red que los conecta.

La decisión final acerca de la distribución de componentes debe basarse no solo en la aplicación individual sino en la mezcla de aplicaciones que estén funcionando en el sistema. Por ejemplo, una instalación podría contener aplicaciones que requieran un extenso procesamiento de GUI y poco procesamiento central de la BD. Esto daría lugar a la utilización de potentes estaciones de trabajo en el lado cliente, y a un servidor muy sencillo. Una vez implantada esta configuración, otras aplicaciones favorecerían el enfoque de cliente principal, para que las capacidades del servidor pudieran verse aumentadas. A medida que madura el uso de la arquitectura C/S, la tendencia es a ubicar la lógica de la aplicación volátil en el servidor. Esto simplifica la implantación de actualizaciones de software cuando se hacen cambios en la lógica de la aplicación.

Para el enlace de componentes del software C/S se utiliza toda una gama de mecanismos distintos dentro de esta. Estos mecanismos están incluidos en la estructura de la red y del sistema operativo, y resultan transparentes para el usuario final situado en el centro cliente. Los tipos más comunes de enlazado son:

- Tubos (pipes): permiten la mensajería entre distintas máquinas que funcionen con distintos sistemas operativos.
- Llamadas a procedimientos remotos: permiten que un proceso invoque la ejecución de otro proceso o módulo que resida en una máquina distinta.
- Interacción cliente/servidor SQL: se utiliza para pasar solicitudes SQL y datos asociados de un componente (típicamente situado en el cliente) a otro componente (típicamente el SGBD del servidor). Este mecanismo está limitado únicamente a las aplicaciones SGBDR.

### **Software intermedio (middleware) y arquitecturas de redistribución de objetos.**

Los componentes de software C/S descritos en las secciones anteriores están implementados mediante objetos que deben ser capaces de interactuar entre sí en una sola máquina (bien sea cliente o servidor) o a través de la red. Un distribuidor de solicitudes de objetos (ORB= Object Request Broker) es un componente de software intermedio (middleware) que capacita a un objeto que reside en un cliente para enviar un mensaje a un método que esté encapsulado en otro objeto que reside en un servidor. En esencia, el ORB intercepta el mensaje y maneja todas las actividades de comunicación y de coordinación necesarias para hallar el objeto al cual se había destinado el mensaje, para invocar su método, para pasar al objeto los datos adecuados, y para transferir los datos resultantes de vuelta al objeto que generase el mensaje inicialmente.

El desarrollo del software para sistemas C/S modernos está orientado a objetos.

### **Análisis de sistemas C/S.**

Aun cuando muchos modelos de proceso de software se podrían adaptar para su utilización en el desarrollo de software para sistemas C/S, una ingeniería del software basada en sucesos y/o orientada a objetos es la que parece funcionar mejor.

Los sistemas cliente/servidor se desarrollan empleando las actividades de ingeniería del software clásicas - el análisis, diseño, construcción y depuración- a medida que evoluciona el sistema a partir de un conjunto de requisitos generales para llegar a ser una colección de componentes de software ya validados que han sido implementados en máquinas cliente y servidor.

La actividad de requisitos para los sistemas C/S difiere poco de los métodos de modelado de análisis que se aplicaban para la arquitectura de computadoras más convencionales. Consiguientemente, los principios y métodos de análisis ya descritos son igualmente aplicables al software C/S.

Dado que el modelado de análisis evita la especificación de detalles de implementación, sólo cuando se haga la transición al diseño se considerarán los problemas asociados a la asignación de componentes de software al cliente y al servidor.

Sin embargo, dado que se aplica un enfoque evolutivo de la ingeniería del software para los sistemas C/S, las decisiones de implementación acerca del enfoque C/S global (p. ej.: cliente principal frente a servidor principal) se podrán hacer durante las primeras iteraciones de análisis y diseño.

### **Diseño para sistemas C/S.**

Quando se está desarrollando un software para su implementación empleando una arquitectura de computadores concreta, el enfoque de diseño debe de considerar el entorno específico de construcción. En esencia, el diseño debería de personalizarse para adecuarlo a la arquitectura del hardware. Cuando se diseña software para su implementación empleando una arquitectura C/S, el enfoque de diseño debe de ser « personalizado » para adecuarlo a los problemas siguientes:

El diseño de datos domina el proceso de diseño. Para utilizar efectivamente las capacidades de un sistema de gestión de bases de datos relacional (SGBDR) o un sistema de bases de datos orientado a objetos (SGBDOO)- los sistemas de gestión de bases de datos SGBDR y orientados a objetos se utilizan mucho en las arquitecturas C/S- el diseño de los datos para a ser todavía más significativo que en las aplicaciones convencionales.

Cuando se selecciona el paradigma controlado por sucesos, el modelado del comportamiento (una actividad de análisis), deberá realizarse y será preciso traducir los aspectos orientados al control implícitos en el modelo de comportamiento al modelo de diseño.

El componente de interacción/presentación del usuario de un sistema C/S implementa todas aquellas funciones que se asocian típicamente con una interfaz gráfica de usuario (GUI). Consiguientemente, se verá incrementada la importancia del diseño de interfaces.

Suele seleccionarse un punto de vista orientado a objetos para el diseño. En lugar de la estructura secuencial que proporciona un lenguaje de procedimientos se proporciona una estructura de objetos mediante la vinculación entre los sucesos iniciados en la GUI y una función de gestión de sucesos que reside en el software basado en el cliente.

Aun cuando prosigue todavía el debate acerca del mejor enfoque de análisis y diseño para los sistemas C/S, los métodos orientados a objetos parecen ofrecer la mejor combinación de características. La notación convencional para análisis y diseño incluye los diagramas de flujo de datos (DFD), los diagramas de entidades y relaciones (DER), y los diagramas de estructuras.

### **Enfoques de diseño convencional.**

En los sistemas C/S, los DFD se pueden utilizar para establecer el ámbito del sistema, para identificar las funciones de nivel superior y las áreas de datos temáticas (almacenes de datos), y para permitir la descomposición de funciones de nivel superior.

El DER adopta también un papel más importante. Sigue utilizándose para descomponer las áreas de datos temáticas (de almacenes de datos) de los DFD con objeto de establecer una visión de alto nivel de la base de datos que haya que implementar empleando un SGBDR.

### **Diseño de bases de datos.**

El diseño de bases de datos se utiliza para definir y después especificar la estructura de la información para los usuarios que se emplean en el sistema cliente/servidor. El análisis necesario para identificar la información para los usuarios se lleva a cabo empleando los métodos de ingeniería de la información habituales. La notación de modelado de análisis convencional, tal como DER, se podrá utilizar para definir objetos de negocios, pero es preciso establecer un depósito de base de datos para capturar la información adicional que no se puede documentar por completo empleando una notación gráfica tal como un DER.

Un sistema de base de datos relacional (SGBDR) hace fácil el acceso a datos distribuidos mediante uso del lenguaje de consulta estructurado (SQL).

### **Estrategia general de pruebas C/S.**

La aplicación se comprueba en solitario en un intento de descubrir errores de su funcionamiento.

Comprobaciones de servidor. Se comprueban la coordinación y las funciones de gestión de datos del servidor. También se considera el rendimiento del servidor (tiempo de respuesta y trasvase de datos en general).

Comprobaciones de bases de datos. Se comprueba la precisión e integridad de los datos almacenados en el servidor. Se examinan las transacciones enviadas por las aplicaciones cliente

para asegurar que los datos se almacenen, actualicen y recuperen adecuadamente. También se comprueba el archivado.

Comprobación de transacciones. Se crea una serie de comprobaciones adecuada para comprobar que todas las clases de transacciones se procesen de acuerdo con los requisitos. Las comprobaciones hacen especial hincapié en la corrección de procesamiento, y también en los temas de rendimiento (p. ej.: tiempo de procesamiento de transacciones y comprobación de volúmenes de transacciones).

Comprobaciones de comunicaciones a través de la red. Estas comprobaciones verifican que la comunicación entre los nudos de la red se produzca correctamente, y que el paso del mensaje, las transacciones y el tráfico de red relacionado tenga lugar sin errores. También se pueden efectuar comprobaciones de seguridad de la red como parte de esta actividad de comprobación.

La comprobación tradicional visualiza la integración de módulos para subsistemas/sistemas y su comprobación como un proceso descendente, ascendente, o alguna variación de los dos anteriores. La integración de módulos en el desarrollo C/S puede tener algunos aspectos ascendentes o descendentes, pero la integración en proyectos C/S tiende más hacia el desarrollo paralelo y hacia la integración de módulos en todos los niveles de diseño. El hecho de que el sistema no se esté construyendo para utilizar un hardware y un software especificado anteriormente tiene su impacto sobre la comprobación del sistema. La naturaleza multiplataforma en red de los sistemas C/S requiere que se preste bastante más atención a la comprobación de configuraciones y a la comprobación de compatibilidades.

---

## APENDICE E

# REDES

Una red es un sistema de interconexión entre computadoras que permite compartir recursos e información :

Según su cobertura, se pueden distinguir tres tipos de redes :

- LAN (Local Area Network). Red de Area Local - Cubre una distancia limitada y facilita al compartir recursos e información
- WAN (Wide Area Network). Red de Area Amplia - Enlaza estaciones de trabajo ubicadas en sitios que cubren un área geográfica extensa.
- MAN(Metropolitan Area Network). Red de Area Metropolitana – Se encuentra distribuida en distancias no superiores al ámbito urbano.

## LAN.

Una LAN, tiene impuesta una restricción de alcance, limitando el área de cobertura al entorno definido por un usuario o tipo de usuario. Siendo un sistema de comunicación intra-oficina, intra-edificio, intra-servicios, que apoya algún tipo de procesamiento de comunicaciones y transferencia de información transparente entre usuarios y/o dispositivos electrónicos.

### Características de las redes locales.

Las redes locales se caracterizan por :

- La Topología,
- Método de acceso,
- Medio de Transmisión.

### Topología.

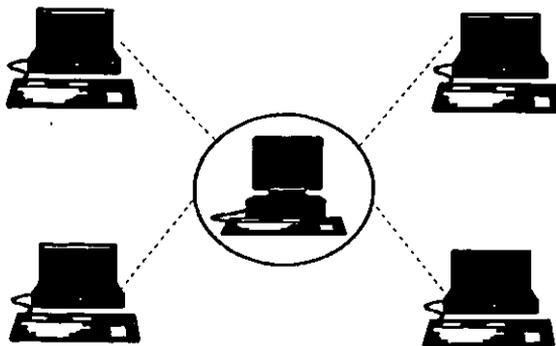
Topología o arquitectura de redes. Se denomina topología a la forma geométrica en que están distribuidas las estaciones de trabajo y los cables que las conectan.

Las estaciones de trabajo de una red se comunican entre sí mediante una conexión física, y el objeto de la topología es buscar la forma más económica y eficaz de conectarlas para, al mismo tiempo , facilitar la fiabilidad del sistema, evitar los tiempos de espera en la transmisión de los datos, permitir un mejor control de la red y permitir de forma eficiente el aumento de las estaciones de trabajo. Tomemos en cuenta que la forma de la red LAN no limita los medios de transmisión.

Normalmente una red LAN cuenta con 5 topologías principales, en su configuración:

- Estrella
- Anillo
- Canal ó Bus
- Malla
- Soporte

### Topología Estrella.

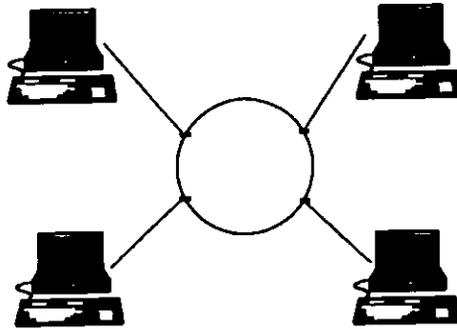


Esta topología depende de un servidor o una Unidad Central de Proceso, la cual controla el flujo de información a través de la red hasta todos los nodos. Todas las comunicaciones se han de hacer necesariamente a través de él.

Tiene un tiempo de respuesta rápido en las comunicaciones de los nodos con el servidor o computadora Central, pero el tiempo de respuesta es lento en la comunicación entre los distintos nodos.

Si se produce un fallo en uno de los nodos no repercutirá en el funcionamiento general de la red, si fallo fuera en la unidad Central de Proceso, todo el sistema deja de funcionar.

#### **Topología Anillo.**



Esta topología consta de varios nodos unidos formando un círculo lógico. Los mensajes se mueven de nodo a nodo en una sola dirección. El mensaje que entra en una red anillo debe contener un grupo de "bits" indicando la dirección donde se debe de entregar el mensaje en el anillo.

Algo muy importante dentro de esta topología es la necesidad de asegurar que todas las estaciones de trabajo tengan el mismo acceso a red. Además este tipo de red permite aumentar el número de nodos sin dificultad, pero a medida que aumenta el flujo de información, será menor la velocidad de respuesta. Si un nodo o un elemento de la red se detiene, toda la red podría dejar de funcionar.

#### **Topología Bus o Canal.**



Servidor de la Red

En este tipo de topología no existe una unidad central de proceso, cada nodo esta conectado a un medio único de comunicación, es decir todos los nodos comparten el mismo canal de comunicación; toda la información circula por ese canal, y cada nodo recoge la información que le corresponde.

El cableado de bus es sencillo, requiere de la menor cantidad de cableado que cualquier topología importante, tiene una gran flexibilidad a la hora de aumentar o disminuir el número de nodos.

Una falla en un solo nodo no necesariamente obstruye el funcionamiento de toda la red; esto depende del tipo de falla. Una falla en el cableado de un nodo sí afectará a la red.

Por lo general, debe haber un mínimo de distancia entre las derivaciones, de forma que los nodos puedan evadir la interferencia de señales.

#### **Topología Malla.**

Este tipo de topología se puede llevar a cabo en una red local, pero digamos que se utiliza con más frecuencia en redes de cobertura amplia o metropolitana (WAN, MAN), 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.

Esta topología se trabaja en tiempo real.

Su mantenimiento es difícil y costoso.

#### **Topología Soporte.**

Se utilizan ruteadores para administrar el flujo de tráfico entre las subredes y el soporte. Se encuentran en grandes consorcios de oficinas o campus (Ciudad Universitaria cuenta con este tipo de topología).

#### **Modelo OSI (Open System Interconnection).**

Para poder establecer una comunicación entre computadoras, es necesario contar con una serie de normas que regulen dicho proceso. Para lograr cierto nivel de uniformidad entre los fabricantes de redes, la organización Internacional de Normas (ISO, International Standards Organization), ha propuesto las normas (estándares) de interconexión para los sistemas abiertos (OSI). Al interconectar computadoras es necesario saber en que forma recibirán la información.

Este modelo consiste de 7 niveles, para realizar las tareas que se llevan a cabo en una comunicación entre computadoras. Cada capa proporciona un servicio para la capa inmediatamente superior. Las 4 primeras capas tendrán funciones de comunicación, las 3 capas siguientes función de proceso. La jerarquía de las capas se realiza desde lo general, en la capa más alta, hasta lo particular en la capa de menor nivel.

Nivel 1	Físico
Nivel 2	Enlace
Nivel 3	Red
Nivel 4	Transporte
Nivel 5	Sesión
Nivel 6	Presentación
Nivel 7	Aplicación.

*Capas del Modelo OSI.*

### **Nivel Físico.**

Es un conjunto de reglas respecto al hardware que se emplea para transmitir datos. Se definen las características eléctricas y mecánicas de la red necesarias para mantener conexión física. Esta capa comprende los conectores e interfaces aceptables para el medio.

La capa física establece si los bits se enviarán en semidúplex o dúplex integral.

### **Nivel de Enlace.**

Se ocupa de empacar los datos en bloques para transmitirlos y asegurar que se reciban en forma correcta.

Provee la conexión lógica a través de la línea, el direccionamiento, secuencia y control de errores. En este nivel de enlace se añaden señalizadores para indicar el inicio y fin de los mensajes.

Incluye formato de los bloques de datos, códigos de dirección, orden de los datos transmitidos, detección y la recuperación de errores.

### **Nivel de Red.**

Establece y mantiene las conexiones.

Se encarga de decidir por donde se han de transmitir los datos dentro de la red (se incluyen la administración y gestión de los datos, la emisión de mensajes y la regulación de tráfico en la red). Asegurando que los bloques de datos lleguen al lugar correcto.

### **Nivel de Transporte.**

Proporciona el control entre nodos de usuarios a través de la red. Asegura la transferencia de la información a pesar de los fallos que pudieran ocurrir en los niveles anteriores.

### **Nivel de sesión.**

La capa de sesión se ocupa de la administración de la red. Maneja procedimientos de reconocimiento de contraseña, de registros de entrada y salida y el monitoreo e informe de la red.

Organiza las funciones que permiten que dos usuarios se comuniquen a través de la red.

Una sesión se identifica por "Identificadores de destino final".

### **Nivel de presentación.**

La capa de presentación se ocupa de la seguridad de la red, de la transferencia de archivos y de las funciones de formato.

Traduce la información del formato de la máquina a un formato entendible por el usuario.

### **Nivel de aplicación.**

El nivel de aplicación maneja mensajes, solicitud de accesos remotos, y es responsable de las características de la administración de la red.

Los programas de red que se encuentran en este nivel incluyen el correo electrónico, los controladores de bases de datos, el software de servidor de archivos y el de impresoras.

Se encarga del intercambio de información entre los usuarios y el sistema operativo.

### **Medios de Acceso.**

Como hemos visto anteriormente, el proceso de transmisión de datos va desde el nivel físico hasta la presentación de la información en cierto formato. El nivel de Enlace es el encargado del control de la comunicación y del acceso al medio

La forma de controlar la transferencia de la información depende exclusivamente del protocolo que se utilice, y deberá realizar las siguientes funciones:

- Sincronización de la comunicación.
- Control de los errores de transmisión.
- Coordinación de la comunicación.
- Recuperación ante los fallos que produzcan.

Protocolo es un conjunto de normas que indican cómo deben actuar las computadoras para comunicarse unas con otras.

En una red local todas las estaciones de trabajo comparten un mismo medio de transmisión, por lo que es necesario que exista un protocolo que determine cuál de las estaciones puede "hablar" en un momento dado. Estos protocolos de acceso al medio, conocidos como protocolos nivel MAC, pertenecientes al nivel 2 del modelo de referencia OSI, son:

Cada uno de estos protocolos tiene asociada una topología de funcionamiento lógico y maneja un formato de trama, que aunque es particular para cada protocolo, posee características comunes con las de los demás protocolos.

El protocolo más utilizado a nivel mundial es el CSMA/CD, conocido también con el nombre de Ethernet. Este es también el que utiliza la red corporativa de TELECOM en cada una de sus dependencias.

### ***Método CSMA/CD (Acceso múltiple por detección de portadora con detección de colisiones).***

Este método es apropiado en una topología BUS.

Es un protocolo para definir las formas en que las redes evitan las colisiones.

Este protocolo es recomendable para cargas de tipo bajo medio y para una longitud media de la red.

La manera en que funciona el protocolo es la siguiente:

Cuando una estación desea transmitir, primero escucha el medio de transmisión. (Esto es el CSMA)

Si el medio está ocupado, espera un tiempo aleatorio y vuelve a intentar la transmisión.

Si el medio de transmisión está desocupado, la estación se lo toma y empieza a transmitir.

La estación escucha el medio de transmisión durante un tiempo para detectar si su transmisión colisionó con la de alguna otra estación que haya empezado a transmitir a mas o menos el mismo instante.

Si hubo colisión, la estación deja de transmitir su mensaje y en cambio emite una señal de interferencia que le permite a todas las estaciones saber que hubo una colisión. La estación espera un tiempo aleatorio antes de volver a intentar la transmisión.

La probabilidad de que haya una colisión depende de varios factores:

**Las características eléctricas del cable:** La velocidad a la que viajan las señales por el medio de transmisión no es ni siquiera la velocidad de la luz. Depende de las características del cable.

**La longitud del cable:** Si el cable es demasiado largo, puede suceder que una estación lejana comience a transmitir mientras que el mensaje de nuestra estación va apenas viajando por el medio de transmisión. La estación lejana no tiene forma de saber que un mensaje viene en camino.

**El tiempo que dura la estación transmisora escuchando el medio:** Este tiempo debe ser lo suficientemente largo como para permitirle a la estación darse cuenta si hubo o no una colisión.

**Aleatoriedad:** Es importante que cuando hallan colisiones, las estaciones esperen diferentes tiempos antes de volver a intentar la transmisión, pues de lo contrario serían inevitables nuevas colisiones.

#### ***Método CSMA/CA (Acceso múltiple por detección de portadora evitando colisiones).***

En este tipo de protocolo, cuando una estación va a enviar un bloque de datos comprueba que la línea está libre, y cuando verifica que lo está, indica que tiene intención de transmitir.

Si hay varias que se encuentran esperando, la transmisión se realiza por turno. En este turno se tienen en cuenta la prioridad de la estación y el orden en que ha indicado que desea transmitir; por tanto, primero transmitirá la que lo haya solicitado primero entre las que tienen la máxima prioridad y no la que lo haya solicitado primero si tiene una prioridad baja.

Este protocolo es recomendable para cargas de tipo medio alto y para una longitud media de la red.

#### ***Token Bus.***

Este método es más aplicado en topologías BUS, aunque no es exclusivo de una forma de red en particular.

Por su característica determinística permite que el usuario asegure el acceso desde cualquier nodo en un tiempo razonable cualquiera sea la aplicación.

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

Aunque la red físicamente es un bus, lógicamente se configura como un anillo para el paso del token de una computadora a otra.

### **Token Ring.**

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 limitado 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 a busy token (testigo ocupado) e incluye el propio busy token con la unidad de datos que transmitía. La unidad de datos viaja de computadora en computadora alrededor del anillo. En caso de que no vaya dirigida a la computadora que en instancia tenga 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.

### **Medios de Transmisión.**

El medio de transmisión es cualquier medio físico que permiten transportar información de una estación de trabajo al servidor o a otra estación de trabajo.

La selección del medio físico a utilizar depende de :

- Tipo de ambiente donde se va a trabajar.
- Tipo de equipo a usar.
- Tipo de aplicación a usar.
- Capacidad económica.

### **Tipos de cables.**

#### **Cable par trenzado.**

Es un cable formado por un par de hilos de cobre trenzados entre sí y recubierto de plástico. El grosor de los hilos y el número de vueltas del trenzado puede variar. Normalmente no tiene blindaje o es muy reducido.

#### **Cable coaxial.**

Conductor de cobre rodeado por un material aislante, una malla trenzada como un segundo conductor y una cubierta de plástico. : conexiones del tipo terminal/host, redes de datos y sistemas de seguridad.

La diferencia de construcción de los dos conductores son las características eléctricas, que da como resultado un desbalance en el circuito, el cual ayuda a prevenir la interferencia electromagnética.

Aplicaciones del cable coaxial.

La clasificación general de las aplicaciones del cable coaxial incluyen redes LAN, sistemas de servidores, sistemas de datos de alta velocidad. En algunas aplicaciones el cable coaxial es utilizado únicamente para la transmisión de datos entre 2 sistemas (punto a punto) mientras que en otro

normalmente los canales de transmisión llevan tipos de bits altos conocidos como configuración multipunto.

#### **Fibra Óptica.**

Es un tipo de cable que transmite pulsos luminosos representando información digital, a través de una fibra de vidrio óptico. Usualmente se le asocia con alta velocidad de voz y aplicaciones de datos. La característica principal de los sistemas basados en fibras es la representación a ciertas duraciones de onda, a 1310 y 1550 nanómetros. Un cable de fibra óptica está formado por : Capa interna o centro, capa media y cubierta. Capa exterior. La capa externa está formada de múltiples capas, combinadas de Kevlar , plástico, PVC, teflón y acero.

#### **Características del cable de fibra óptica.**

Transmisión Half-Duplex.

Diámetro pequeño del cable y alta capacidad de procesamiento.

Falta de flexibilidad.

Resistencia a la interferencia electromagnética.

Seguridad

#### **Aplicaciones comunes de fibra óptica.**

Redes de Area Local.

Redes de Area Metropolitana.

Redes de Area amplia.

CAD/CAM

#### **Ventajas de los sistemas de fibra óptica.**

Amplio ancho de banda.

Distancia de transmisión.

Resistencia a interferencia electromagnética.

Seguridad.

#### **Desventajas de los sistemas de fibra óptica.**

Alto costo.

Problemas de terminación y empalme.

Limitaciones físicas.

#### **Medios Inalámbricos.**

Este medio de transmisión se puede dividir en dos ramas :

LAN inalámbrica.

Móvil inalámbrica.

La diferencia radica en las facilidades de transmisión mientras que la comunicación LAN inalámbrica utiliza transmisores y receptores que se encuentran dentro de un área limitada, la móvil inalámbrica involucra a terceros, como compañías de telecomunicaciones.

Se cuentan con tres diferentes formas de transmisión inalámbrica:

Luz infraroja.

Radio de frecuencia simple.

Radio de espectro expandido.

**Microondas.**

Este medio de transmisión es inalámbrico, 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 :

Satélite.

Enlaces de edificio a edificio en área metropolitana.

Enlaces a través de suelos difíciles para tendido de cable.

## MODELOS DE INTERCONEXION.

### **Netware.**

Las capas del modelo Netware son :

Física. Transmisión del equipo físico de la red a otro.

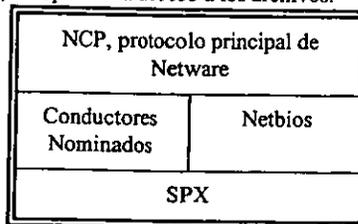
Interfaz abierta de enlace de datos y especificaciones de la interfaz del controlador de red. Esta capa se divide en ODI y NDIS que corresponden a la capa de enlace del modelo OSI. Su función es definir que envíen y reciban información entre unidades conectadas.

IPX (Internetwork Packet Exchange). Define los protocolos de sin conexión que encaminan los datos en forma dinámica.

SPX (Sequence Packet Exchange). Se encarga de que la información transmitida de un punto a otro punto llegue en el mismo orden en que fue enviada.

NetBIOS. Se encarga de la gestión de la sesión de comunicaciones, la denominación de nodos, difusión de los nombres y emplazamientos de los nombres de los servidores y la transmisión de los datagramas sin conexión.

NCP (Netware Core Protocol). Proporciona acceso a los archivos.



IPX.	
Controladores LAN	
ODI	NDIS
Física	

*Capas Del Modelo Netware***LAN MANAGER**

Es utilizado por productos de Microsoft para conexión en red, es Windows NT su principal impulsor.

Bloques de mensajes del servidor	
Conductores Nominados	Netbios
NetBEUI	
Controladores LAN.	
NDSI	
Física	

*Modelo LAN Manager.*

Aplicación
Presentación
Sesión
Transporte
Red
Enlace
Física

**TCP/IP.**

TCP/IP es un conjunto de protocolos de comunicación de datos. Estos protocolos permiten encaminar la información de una máquina a otra, la entrega de correo electrónico y noticias, e incluso el uso de capacidades de registro remotas.

El nombre de TCP/IP se refiere a los dos protocolos principales: Protocolo de Control de Transmisión y Protocolo Internet. Aunque hay muchos otros protocolos que ofrecen servicios que operan sobre TCP/IP.

TCP/IP solo define 3 capas:

Aplicación.

Transporte.

Red.

**CAPA DE RED.**

La capa de red llamada IP por "Internet Protocol" (que es el protocolo más importante de esta capa) realiza la función de enviar los paquetes provenientes de la capa de transporte. Esto lo logra definiendo una estructura muy simple de direcciones de 32 bits que se subdividen en 2 campos: uno para la dirección de la red y otro para la dirección de la máquina y utilizando un mecanismo sin conexión para el envío de datagramas.

En pocas palabras, el protocolo IP es el que se encarga de enviar los datos a través de la red; sólo se encarga de enviar y no asegura que lleguen los datos a su destino. Para la realización y funciones de la capa IP requiere de una o más capas inferiores (las llamadas capas de enlace y físicas del modelo ISO/OSI). Sin embargo, en TCP/IP estas capas no están definidas, sólo IP deberá de usar los múltiples y cada día más abundante protocolos y medios disponibles para la transmisión de datos.

**CAPA DE TRANSPORTE.**

Mientras que la capa de red envía los mensajes de una máquina a otra, la capa de transporte realiza la función de proporcionar la comunicación entre los programas de aplicación. Además, podría ofrecer la confiabilidad en la transmisión, tanto en la integridad de los datos como el orden en que estos se envían.

El mecanismo que usa la capa de transporte utiliza conceptos conocidos como buzón (o "mail-box") para establecer la comunicación entre los programas de aplicación.

Bajo TCP/IP existen dos protocolos para esta capa, el primero TCP "transmission Control Protocol" que tiene 4 características principales:

1. -Confiabilidad en la transmisión: los datos llegan en el mismo orden en que se enviaron y sin errores.
2. -Conexión de circuito virtual: antes de poder transmitir información, se debe establecer una "conexión" y al finalizar la transmisión se debe efectuar una desconexión.
3. -La transmisión se realiza bajo la idea de un flujo de datos. El programa que lee de la conexión no sabe dónde empieza o termina un paquete de información.

4.-La conexión en "full-duplex". Ambos extremos de la conexión pueden leer y/o escribir en la conexión.

**CAPA DE APLICACION.**

Por último, la capa de aplicación debe realizar sus funciones basadas en la capa de transporte. Ejemplos de los protocolos de esta capa: TELNET que es un protocolo que se utiliza para simular una terminal a través de la red, otro es el protocolo FTP "File Transfer Protocol" o protocolo de transferencia de archivo, que permite transferir archivos de una máquina a otra.

## BIBLIOGRAFIA

Booch Grady.  
Object Oriented Analisis and Design With Applications.  
1994

Douglas Comer  
Internetworking with TCP/IP  
Prentice Hall, 1991

Hazlehurst Peter  
Using Sybase System XII  
QUE, 1996

José Luis Raya Cabrera,  
Cristina Raya Pérez,  
Redes Locales y TCP/IP;  
Ed. Alfa Omega, 1995

Luis Joyanes Aguilar  
C++ Un enfoque Orientado a Objetos.  
McGraw-Hill, 1994

Nestor González Sainz.  
Comunicaciones y redes de procesamiento de datos,  
Ed. McGraw Hill, 1987

Neil Jenkins,  
Stan Schatt,  
Redes de Area Local;  
Ed. Prentice Hall, 1996

Pressman Roger.  
Ingeniería de Software.  
McGraw- Hill, 1993

Rebecca Wirfs-Brock, Brian Wilkerson, Lauren Wiener

Designing Object-Oriented Software

Prentice-Hall, 1990

Sommerville Ian.

Ingeniería de Software

Addison-Wesley Iberoamericana, 1988

Tanenbaum Andrew S.

Redes de computadoras.

Prentice Hall, 1991

Tanenbaum Andrew S

Sistemas Operativos Modernos,

Prentice hall, 1992

Vaughn Larry T.

Client/Server System Design and Implementation

McGraw-Hill, 1994

Yourdon Edward

Modern Structured Analysis,

Prentice Hall, 1989