



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA DE INFORMACIÓN INTEGRAL DEL CENTRO COAPA

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
P R E S E N T A:
SANDRA OLIVARES CEDILLO



DIRECTOR: ING GERMAN SANTOS JALMES

TESIS CON FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A Dios por caminar siempre junto a mí, porque me ha amado tanto que me dio la familia que tengo. Por velar por mí siempre y por que sin él esto no hubiese sido posible.

Gracias Dios mío, gracias.

A mis padres, José Ramón y María del Carmen porque son el mejor ejemplo de lucha, entrega y compromiso.

Porque con mi Padre, aprendo cada día el verdadero significado de esta palabra papá, gracias. A mi madre porque con su amor me impulsa a ir siempre adelante, a su compañía incondicional, gracias madre esto es más tuyo que mío te pertenece.

Gracias a los dos.

A mis hermanos José Ramón, Nancy Areli, María Judith, gracias por su apoyo, por su amor, por ser el mejor ejemplo para seguirme superando día con día. Cada uno de ustedes me ha dado lecciones de vida. Pepe su espíritu de lucha, Nancy su tenacidad, dulzura y templanza, Judith siempre sonriendo, la chispa de la casa. Siempre caminaremos juntos, si en algo puedo apoyarlos nunca duden en pedírmelo.

A la Universidad Nacional Autónoma de México, por ser mi alma mater, y haberme brindado la oportunidad de ser parte de ella.

A la gloriosa Facultad de Ingeniería, por la formación que me dio, por los conocimientos que adquirir y que siempre llevare en alto el buen nombre de esta institución.

A todos los amigos, aquellos que caminaron junto a mí en mi desarrollo profesional, gracias por su amistad. A tantas personas especiales que Dios a puesto en mi camino, mil gracias.

Pido a Dios los cuide y proteja siempre.

Índice	
ÍNDICE	1
ÍNDICE DE FIGURAS	4
CAPÍTULO 1. INTRODUCCIÓN	7
1.1 OBJETIVO.....	7
1.2 MARCO TEÓRICO.....	7
1.2.1 <i>Metodologías en el Desarrollo de Sistemas</i>	7
1.3 ANÁLISIS DEL SISTEMA.....	15
1.3.1 ANÁLISIS.....	15
1.4 DISEÑO.....	21
1.4.1 <i>Diseño de la base de datos</i>	21
1.4.2 <i>Diseño del sistema</i>	21
1.5 MANEJADORES DE BASES DE DATOS.....	24
1.5.1 <i>Características de un Manejador de Base de Datos Relacional</i>	25
1.6 HERRAMIENTAS DE DESARROLLO.....	25
1.7 PROTOCOLOS DE COMUNICACIÓN.....	26
1.8 DESARROLLO DE APLICACIONES CLIENTE/SERVIDOR.....	28
1.8.1 <i>Arquitectura CLIENTE/SERVIDOR de dos capas</i>	30
1.8.2 <i>Arquitectura CLIENTE/SERVIDOR de tres capas</i>	31
1.9 SEGURIDAD.....	32
CAPÍTULO 2. DEFINICIÓN DEL PROBLEMA	35
2.1 PROBLEMÁTICA ACTUAL.....	35
2.2 DEPARTAMENTOS Y SUS REQUERIMIENTOS DE INFORMACIÓN.....	35
2.2.1 <i>Descripción de las Areas Involucradas</i>	35
2.3 RECOLECCIÓN DE REQUERIMIENTOS.....	37
2.4 ALCANCE.....	38
2.5 SOLUCIONES PROPUESTAS.....	39
CAPÍTULO 3. DESARROLLO DE LA SOLUCIÓN PROPUESTA	45
3.1 INTRODUCCIÓN.....	45
3.2 ANÁLISIS.....	45
3.2.1 <i>Análisis de los Formatos</i>	45
3.2.1 <i>Resultado de las entrevistas</i>	55
3.3 DISEÑO RELACIONAL.....	63
3.5 DISEÑO DEL SISTEMA.....	65
3.5.2 <i>Diagrama de Contexto</i>	68
3.5.3 <i>Operaciones sobre Catálogos</i>	72
3.5.4 <i>Diseño de reportes</i>	73
3.6 MÓDULOS QUE COMPONEN EL SISTEMA.....	75
3.7 BENEFICIOS DEL SISTEMA.....	76
CAPÍTULO 4 LIBERACIÓN	77
4.1 PARAMETROS GENERALES.....	77
4.2 PRUEBAS.....	78
4.3 MANUAL DE USUARIO.....	79
4.3.1 <i>Navegando en la aplicación</i>	79
4.3.2 <i>Navegando en los módulos</i>	81

4.3.3	<i>Navegando por los Catálogos</i>	88
4.3.4	<i>¿Cómo se genera un reporte?</i>	89
CONCLUSIONES Y TRABAJO FUTURO.....		92
BIBLIOGRAFÍA		93
APÉNDICE A		94
SECCION I.....		95
SECCIÓN II		109
APÉNDICE B		114
APÉNDICE C		124

Índice de Figuras

Fig. 1. 1	Proceso de la Ingeniería de Software	8
Fig. 1. 2	Metodología clásica o de cascada	9
Fig. 1. 3	Metodología de ciclo de vida evolutivo	10
Fig. 1. 4	Pasos en el diseño de base de datos	12
Fig. 1. 5	Las vistas en UML	14
Fig. 1. 6	Algunos elementos de modelado comunes	14
Fig. 1. 7	Ejemplos de algunas relaciones	15
Fig. 1. 8	Representación de una Entidad	17
Fig. 1. 9	Arquitectura Cliente/Servidor de dos capas	30
Fig. 1. 10	Arquitectura CLIENTE/SERVIDOR Tres Capas	31
Fig. 2. 1	Organigrama del Centro de Computo	36
Fig. 2. 2	Tabla comparativa de DBMS	41
Fig. 2. 3	Tabla comparativa de Herramientas de Desarrollo	44
Fig. 3. 1	Forma de Inscripción	48
Fig. 3. 2	Solicitud para Examen de Evaluación de Conocimientos Previos en Cómputo	49
Fig. 3. 3	Formato de recepción documentos para candidatos a Diplomados y Líneas de Especialización	50
Fig. 3. 4	Tabla Definición de Entidades	54
Fig. 3. 5	Ejemplo de un registro de la tabla CAT_CURSO	57
Fig. 3. 6	Registro resultante al aplicar la 1FN	57
Fig. 3. 7	Definición de la tabla CURSO_MODULO	58
Fig. 3. 8	Definición de la tabla CAT_CURSO	58
Fig. 3. 9	Definición de la tabla CURSO_MODULO	58
Fig. 3. 10	Definición de la tabla Modulo	59
Fig. 3. 11	Relación recursiva en CAT_CURSO	59
Fig. 3. 12	Ejemplo de un registro	59
Fig. 3. 13	Diagrama Entidad -Relación	64
Fig. 3. 14	Diagrama jerárquico funcional de Nivel 1	65
Fig. 3. 15	Diagrama jerárquico funcional de Nivel 2	67
Fig. 3. 16	Diagrama de contexto	68
Fig. 3. 17	Funciones pncipales del Sistema	71
Fig. 4. 1	Pantalla de los Datos del Personal	77
Fig. 4. 2	Pantalla de Identificación del usuario	79
Fig. 4. 3	Menú	80
Fig. 4. 4	Pantalla de selección de impresora	80
Fig. 4. 5	Barra de Herramientas	81
Fig. 4. 6	Datos del Alumno	83
Fig. 4. 7	Ventana de la Solicitud	83
Fig. 4. 8	Ventana Documentos anexos	84
Fig. 4. 9	Ventana de Datos del Grupo	85
Fig. 4. 10	Ventana de captura del recibo	86
Fig. 4. 11	Opción Catálogos	88
Fig. 4. 12	Ventana General de Captura de Catálogos	89
Fig. 4. 13	Opción Reportes	90
Fig. 4. 14	Ventana de Entrada de Parámetros	90
Fig. 4. 15	Contenedor de Reportes	91
Fig. 1	Relación ALUMNO - P_ALTO_REMTO	95
Fig. 2	Relación ALUMNO - DEPENDENCIA	95
Fig. 3	Relación ALUMNO - NIVEL_ESCOLAR	95
Fig. 4	Relación PROPUESTA - ALUMNO	96
Fig. 5	Relación ALUMNO - EDO_PROFESION	96
Fig. 6	Relación ALUMNO - ESC_FAC	96
Fig. 7	Relación CALIF_PREVIO - ALUMNO_CURSO	97

Fig. 8	Relación	ALUMNO - TIPO_PERSONAL.....	97
Fig. 9	Relación	TIPO_PERSONAL - PERSONAL.....	97
Fig. 10	Relación	GRUPO - ALUMNO_CURSO - ALUMNO.....	98
Fig. 11	Relación	ALUMNO_CURSO - BECA.....	98
Fig. 12	Relación	GRUPO - ESTADO_GRUPO.....	98
Fig. 13	Relación	GRUPO - SEDE.....	99
Fig. 14	Relación	GRUPO - AULA.....	99
Fig. 15	Relación	GRUPO - TIPO_GRUPO.....	99
Fig. 16	Relación	GRUPO - CAT_CURSO.....	100
Fig. 17	Relación	INSTITUCION - INST_CURSO - GRUPO.....	100
Fig. 18	Relación	SOLICITUD - MEDIO_DIFUSION.....	100
Fig. 19	Relación	SOLICITUD - TIPO_SOLICITUD.....	101
Fig. 20	Relación	SOLICITUD - ALUMNO_CURSO.....	101
Fig. 21	Relación	SOLICITUD - RECIBO.....	101
Fig. 22	Relación	RECIBO - ESTADO_RECIBO.....	102
Fig. 23	Relación	RECIBO - TIPO_PAGO.....	102
Fig. 24	Relación	RECIBO - BANCO.....	102
Fig. 25	Relación	CAT_CURSO - AREA.....	103
Fig. 26	Relación	CAT_CURSO - CAT_SOFTWARE - SOFTWARE.....	103
Fig. 27	Relación	CAT_CURSO - CAT_MODULO - MODULO.....	104
Fig. 28	Relación	CAT_CURSO - PROG_CAT_CURSO - PROGRAMA.....	104
Fig. 29	Relación	CONCEPTO_RECIBO - ARTICULO.....	105
Fig. 30	Relación	CONCEPTO_RECIBO - RECIBO.....	105
Fig. 31	Relación	EGRESO - TIPO_EGRESO.....	105
Fig. 32	Relación	PERSONAL - DEPARTAMENTO.....	106
Fig. 33	Relación	EXAMEN - SOLICITUD.....	106
Fig. 34	Relación	DOCUMENTO - SOLICITUD_DOC - SOLICITUD.....	106
Fig. 35	Relación	GRUPO - PROFESOR.....	107
Fig. 36	Relación	MODULO - MODULO_ESP - CURSO_ESP.....	107
Fig. 37	Relación	P_ALT_REND - NIVEL.....	107
Fig. 38	Relación	CAT_CURSO - TIPO_CAT_CURSO.....	108
Fig. 39	Relación	CAT_CURSO - CLASIFICA_CURSO.....	108
Fig. 40	Diagrama	de Flujo de Mantenimiento de Catálogos.....	109
Fig. 41	Diagrama	de Flujo de Mantenimiento de Catálogos.....	110
Fig. 42	Diagrama	de Flujo de Mantenimiento de Catálogos.....	111
Fig. 43	Diagrama	de Flujo de Mantenimiento de Catálogos.....	112
Fig. 44	Diagrama	de Flujo de Mantenimiento de Catálogos.....	113

Capítulo 1. Introducción

En este primer capítulo, se da a conocer el objetivo del sistema de información, se hace un repaso de los elementos teóricos relacionados con el desarrollo de sistemas.

1.1 Objetivo

Realizar un sistema de información, que automatice ciertas tareas que se llevan a cabo en forma manual en el Centro de Computo DGSCA Coapa, se hará uso de una metodología para el análisis y diseño, aplicaremos algunas de las técnicas y elementos teóricos adquiridos durante la carrera.

El sistema será capaz de administrar los cursos que se dan en esta institución y los procesos que intervienen en él, conoceremos una visión particular de cada departamento y una visión general de lo que ocurre en el centro.

1.2 Marco teórico

1.2.1 Metodologías en el Desarrollo de Sistemas

Metodologías de Desarrollo del Sistema

Una metodología es una serie de pasos a seguir para desarrollar un sistema información a continuación, se listan los pasos generales del proceso de Ingeniería de Software, más adelante abordamos a detalle cada una de las metodologías que se utilizan en el desarrollo de software.

Proceso de Ingeniería de Software.

- Especificación
- Diseño
- Implementación
- Prueba
- Instalación
- Mantenimiento

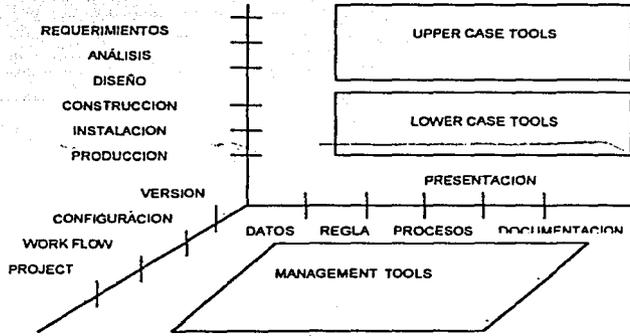


Fig. 1. 1 Proceso de la Ingeniería de Software

Metodología clásica o de cascada

La metodología clásica o de cascada se basa en un enfoque sistemático y una serie de etapas que se suceden secuencialmente: análisis, diseño, desarrollo, pruebas y mantenimiento, no se continúa con la siguiente etapa hasta que no se hayan terminado las especificaciones de la primera. Esta metodología abarca las siguientes actividades:

Ciclo de vida en cascada.

- Captura y análisis de requisitos.
- Diseño del sistema
- Implementación
- Integración y Prueba.
- Operación y Mantenimiento.

Definición de requisitos.

- Con la participación del cliente, se define la funcionalidad, restricciones y objetivos del sistema.

Diseño del sistema.

- Se establece la funcionalidad del software. El resultado es la arquitectura del sistema.

Limitaciones

- Es el modelo más simple, más conocido y usado.
- Propone una organización sistemática y secuencial del proceso de software.
- El único objeto formalizado es el código y aparece al final del proceso.
- Sólo se prueba el código.
- Sólo hay herramientas de apoyo en la fase de implementación y prueba.

CICLO DE VIDA EN CASCADA

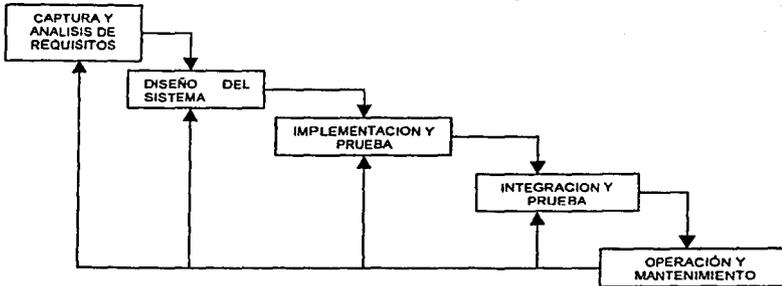


Fig. 1. 2 Metodología clásica o de cascada

Modelo evolutivo

Ciclo de vida evolutivo.

El objetivo es vencer las limitaciones del modelo de ciclo de vida en cascada motivadas por la deficiente realimentación entre sus fases.

Prototipo evolutivo:

- El objetivo es trabajar con los clientes y construir un sistema final desde una especificación informal.
- El desarrollo debe comenzar con una completa comprensión de los requisitos.
- El sistema evoluciona añadiendo nuevas facilidades propuestas por el cliente.

TESIS CON FALLA DE ORIGEN

CICLO DE VIDA EVOLUTIVO

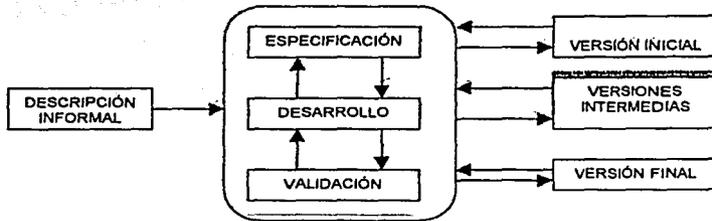


Fig. 1.3 Metodología de ciclo de vida evolutivo.

Modelo en espiral

- Desarrollado por Bohem con el objetivo de aunar las ventajas de los modelos en cascada y evolutivo.
- Hace un tratamiento explícito del riesgo del proceso.
- El modelo tiene la forma de una espiral, se puede establecer una relación entre cada giro y una fase del proceso de desarrollo:
 - 1° Giro: Objetivos, alternativas, restricciones, etc. del sistema.
 - 2° Giro: Especificación de requisitos.
 - 3° Giro: Diseño.

Fases del modelo en espiral

- No hay un número fijo de fases. Las fases deben decidirlas el equipo de gestión del proyecto.
- Cada vuelta en la espiral se divide en 4 sectores:
 - Definición de objetivos
 - Identificación de los objetivos específicos de cada fase
 - Evaluación y reducción del riesgo
 - Desarrollo y validación
 - Elección de un modelo adecuado para la siguiente fase del desarrollo.
 - Planificación.
 - Se revisa el estado del proyecto y se planifica el siguiente giro.

Metodología Gane & Sarson

Esta metodología esta basada en la construcción de un modelo lógico del sistema partiendo de lo general a lo particular, hace uso de las técnicas gráficas como son los diagramas de flujo de datos y entidad relación.

Las etapas que la forman son: estudio inicial donde se determinan las posibles mejoras y se definen las áreas de la organización que están involucradas en el desarrollo del sistema, la primera etapa se compone de una serie de entrevistas con los responsables de cada área para conocer sus requerimientos; El estudio detallado es la siguiente etapa en donde se determina las

limitaciones y los alcances, en esta etapa se hacen uso de los diagramas de flujos de datos. Una vez realizado el DFD se define el diccionario de datos, que es una recopilación detallada de cada uno de los datos que se encuentran involucrados en el DFD; se definen la lógica de los procesos como una serie de pasos ordenados que son llevados a cabo para realizar un proceso en particular; en la refinación del diseño lógico, se determinan los aspectos físicos de contenido y organización tanto del almacenamiento de los datos como de los flujos de datos y procesos.

Metodología CASE

La metodología CASE se basa en la utilización de herramientas de software que facilitan el desarrollo de sistemas. Estas herramientas incluyen lenguajes de cuarta generación para la consulta y manipulación de los datos, generación automática del código fuente basándose en las especificaciones del analista, ayudan en la interacción y definición de pantallas, y ofrecen facilidades para la documentación del sistema.

Esta metodología inicia con la recolección de requisitos que son traducidos directamente en un prototipo operacional. Para sistemas pequeños, es fácil traducir los requerimientos del cliente de manera que una herramienta CASE los interprete. Sin embargo, para sistemas más grandes se requiere un mayor tiempo para el análisis y el diseño del sistema, de tal forma que la herramienta CASE pueda interpretarla.

La ventaja en la utilización de la metodología CASE es que una vez realizado el análisis y el diseño permite efectuarse con rapidez la documentación del sistema, la generación del código fuente y la generación de pantallas.

Herramientas Case -

Es el conjunto de una serie de productos orientados a la automatización de la producción de software, específicamente al desarrollo asistido de sistemas.

A continuación vamos a ver diversos aspectos técnicos sobre ella.

En primer lugar, por las fases del ciclo de vida que abordan se distinguen en:

Upper-Case.- Son las que automatizan esencialmente las etapas de planificación, análisis y diseño es decir, las primeras del desarrollo de sistemas.

Lower-Case.- Son la que automatizan la fase de producción o generación de código y su mantenimiento.

Una herramienta CASE moderna debe contar con las siguientes características:

- Dibujo de diagramas
- Soporte de navegación
- Soporte multiusuario
- Generación de código
- Ingeniería de reversa
- Integración con otras herramientas
- Intercambio de modelos

Algunas de las desventajas que presenta son:
El cliente debe estar seguro de lo que necesita
El costo de estas herramientas es elevado.

Metodología CASE para bases de datos relacionales.

Esta metodología fue introducida por E. F. Codd en 1979 se desarrollo con base en el concepto matemático de la relación, donde una relación es un subconjunto de un Producto Cartesiano. El dominio es el conjunto de valores. En esta se representan los datos como una colección de relaciones. El álgebra relacional fue desarrollada en 1970 y el cálculo relacional en 1971 por Codd. El álgebra relacional es un lenguaje "procedural" para la manipulación de relaciones.

En esta se especifica paso por paso la respuesta a una consulta de los datos contenidos en una relación. El cálculo relacional es un "non procedural language". En el cálculo relacional una consulta se resuelve en un solo paso. El álgebra y el cálculo relacional proveen una forma teórica de manipular una base de datos relacional. Codd demostró que el álgebra relacional y el cálculo relacional son lógicamente equivalentes.

En el análisis se determinan primero las entidades, después las relaciones y finalmente los atributos. Se hace caso omiso de la implantación y el procedimiento en cuestión, y nos enfocamos en los nombres y las descripciones.

Las entidades, relaciones y atributos emergen de una manera confusa durante las entrevistas, a esta fase se le conoce como la recolección de requerimientos

En el diseño relacional, las entidades se convierten en tablas, los atributos en columnas, los identificadores únicos en llaves primarias y las relaciones en llaves foráneas.

En esta etapa también se aplican las formas normales en la base de datos según corresponda a la base de datos. El propósito de las formas normales es reducir la redundancia en la base de datos, ya que trae consigo problemas como la inconsistencia de datos, anomalías en la ejecución de updates.

En el diseño físico cubre todo lo que es invisible para la aplicación y los usuarios. Específicamente incluye la creación de índices, particiones, el tamaño físico de renglones, tablas etc. La meta del diseño físico es optimizar el rendimiento de la base de datos, reducir el acceso a disco para leer o escribir, reducir tiempo de CPU y el almacenamiento físico.

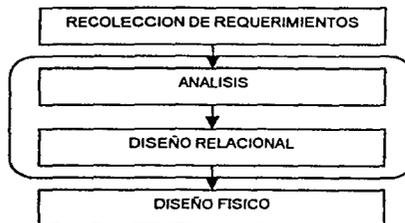


Fig. 1. 4 Pasos en el diseño de base de datos

La técnica utilizada para representar las relaciones entre cada una de las entidades de nuestra base de datos esta dada por BECH y es una de las utilizadas por algunas herramientas CASE.

Enfoque orientado a objetos UML

El Lenguaje Unificado de Modelo UML (Unified Modelling Language) "es un lenguaje visual de modelado de propósito general que es utilizado para especificar, visualizar, construir y documentar sistemas de software".

Elementos del lenguaje

A continuación se describen las diferentes partes que conforman a UML.

- **Vistas (Views):** Las vistas muestran diferentes aspectos del sistema que esta siendo modelado. Una vista no es gráfica, es más bien, una abstracción consistente en un número de diagramas. Solo definiendo un número de vistas, las cuales muestran cada una un aspecto particular del sistema, se puede tener una representación del sistema que se esta construyendo. Las vistas también ligan el lenguaje de modelado con el método de desarrollo escogido.
- **Diagramas:** Los diagramas son las gráficas que describen el contenido en una vista. UML tiene nueve diferentes tipos de diagramas que son usados en combinación para proveer todas las vistas del sistema
- **Elementos de modelado:** Los conceptos usados en los diagramas son elementos de modelado que representan conceptos comunes de orientación a objetos como clases, objetos y mensajes, y la relación entre estos conceptos, incluyendo asociaciones, dependencia y generalización. Un elemento de modelado es usado en diferentes diagramas, pero siempre tienen el mismo significado y símbolo.
- **Mecanismos generales:** Los mecanismos generales proveen comentarios extra, información o semántica acerca de un elemento de modelado, también proveen un mecanismo de extensión para adaptar o extender el UML a un método específico, organización o usuario.

Vistas

Un sistema es descrito mediante diferentes aspectos: el funcional (su estructura estática e interacción dinámica), el no funcional (requerimientos en cuanto al tiempo, disponibilidad, confiabilidad, de distribución, etc.) y aspectos de organización. Es por esto que un sistema es descrito mediante un conjunto de vistas, donde cada vista representa una proyección de la descripción completa del sistema, mostrando un aspecto particular del sistema.
"Una vista es simplemente un subconjunto de construcciones de modelado de UML que representan un aspecto del sistema".

Tipos de vistas

- **Vista de casos de uso (Use – case view):** Una vista que muestra la funcionalidad del sistema tal y como es percibida por los actores externos. Un actor puede ser un usuario del sistema u otro sistema
- **Vista Lógica (Logical View):** Una vista que muestra como es diseñada la funcionalidad dentro del sistema, en términos de la estructura dinámica del sistema y su comportamiento dinámico.

- Vista de Componentes (Componet view): Una vista que muestra la organización de los componentes de software.
- Vista de Concurrencia (Currency view): Una vista que muestra la concurrencia en el sistema mostrado los problemas de comunicación y sincronización que están presentes en un sistema concurrente.
- Vista de Distribución (Deployment view): Una vista que muestra la distribución del sistema en la arquitectura física, con computadoras y dispositivos llamados nodos.



Fig. 1. 5 Las vistas en UML.

Elementos de modelado

Un elemento de modelado es definido por ciertas reglas semánticas de modelado, una definición formal del elemento o el significado exacto de lo que representa en oraciones no ambiguas. Un elemento de modelado tiene también un elemento visual, que es la representación gráfica del elemento usado para representarlo en los diagramas.

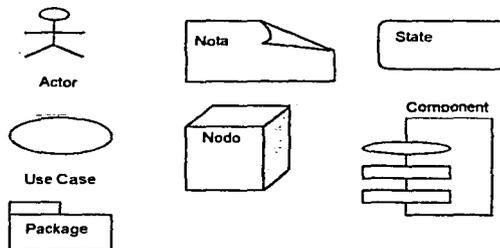


Fig. 1. 6 Algunos elementos de modelado comunes

Además de estos elementos, también las relaciones son otros ejemplos de elementos de modelado, y son usadas para conectar otros elementos de modelado. Algunos ejemplos son:

- **Asociación:** Conecta elementos y liga instancias.
- **Generalización:** También llamada herencia, significa que un elemento puede ser una especialización de algún otro.
- **Dependencia:** Indica que un elemento depende de alguna manera de algún otro

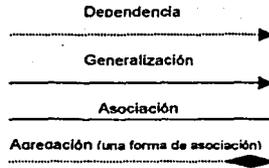


Fig. 1. 7 Ejemplos de algunas relaciones

1.3 Análisis del Sistema

1.3.1 Análisis

En esta etapa se determinan las necesidades y requerimientos del usuario, la tarea de análisis de requerimientos y especificaciones es un proceso de descubrir, refinar, modelar y especificar. El alcance del software es establecido inicialmente por el ingeniero de sistemas y refinado durante la planeación del proyecto. Tanto el desarrollador como el usuario participan activamente. Es la tarea del usuario intentar concretar, detallar y formular un concepto (a veces no muy claro) de funciones y desarrollo del software.

El análisis de requerimientos de software puede ser dividido en cinco áreas:

- 1) Reconocimiento de problemas;
- 2) Evaluación y síntesis;
- 3) Modelación,
- 4) especificación y
- 5) Revisión.

Al inicio, el analista estudia las especificaciones del sistema (si existen) y el plan de proyecto de software. Es importante entender el software dentro del contexto de sistemas y revisar el alcance de este. La meta del analista es reconocer los elementos problemáticos básicos, como son percibidos por el cliente - usuario. En esta primera fase se debe realizar la observación honesta de lo que sucede en el negocio. Luego se analiza la información recolectada y se identifican los problemas, las oportunidades y objetivos.

Las primeras personas involucradas en esta primera fase son los analistas, usuarios y administradores del sistema y las actividades consisten de entrevistas a los administradores de los usuarios, sumariazión del conocimiento obtenido, estimación del alcance del proyecto y documentación de los resultados. La salida de esta fase es un estudio de factibilidad que contiene una definición del problema y la sumariazión de los objetivos.

La siguiente fase es la de la determinación de los requerimientos de información para los usuarios particulares involucrados. Entre las herramientas utilizadas se encuentran muestreo e investigación de los datos relevantes, entrevistas, cuestionarios, el comportamiento de los tomadores de decisiones y su ambiente de oficina y hasta la elaboración de prototipos.

En esta fase, el analista está esforzándose por comprender qué información necesitan los usuarios para realizar su trabajo. Varios de los métodos para determinar los requerimientos de información involucran la interacción con los usuarios. Esta fase sirve para formar la imagen que el analista tiene de la organización y sus objetivos.

La evaluación del problema y síntesis de solución es la siguiente área en que se enfocan los esfuerzos de los analistas. Estos deben definir todos los elementos externos de datos observables, evaluar el flujo y contenido de la información, definir y elaborar todas las funciones del software, entender el comportamiento del software dentro del contexto de los eventos que afectan al sistema, establecer las características de la interfaz y descubrir restricciones adicionales del diseño. Cada una de estas tareas servirá para describir el problema a fin de que una solución o enfoque pueda ser sintetizado.

El análisis de las decisiones de criterios múltiples (decisiones en las que deben ser balanceados muchos factores) también parte de esta fase. Se dispone de muchas técnicas para el análisis de costo/beneficio de las alternativas y hace recomendaciones sobre lo que debe ser hecho (en caso de haberlo). Si alguna de las recomendaciones es aceptable para la administración, el analista continúa sobre ese curso. Cada problema de sistema es único y nunca hay una solución correcta. La manera en que se formula una solución o recomendación depende de la capacidad y entrenamiento profesional individual de cada analista.

Entrevistas

Las entrevistas son una herramienta importante, a través de estas se definen cuales son las necesidades del usuario.

Estas pueden realizarse con la aplicación de un cuestionario, o preguntas dirigidas en una conversación normal, en la cual se toman notas y se detectan los requerimientos, estas se realizan antes de tener un diseño de sus necesidades y después de este para afinar detalles en posibles enfoques diferentes de sus necesidades.

Generalmente se pregunta:

- ¿Existe un sistema de información propiamente dicho para sus requerimientos?
- ¿Cuáles son los principales problemas que ha detectado con el actual sistema?
- ¿Cuántos sistemas se tienen hasta este momento?
- ¿Qué desearía que haga el sistema?

Diagrama Entidad - Relación (DER)

El diagrama entidad relación tiene como objetivo definir todas las cosas que el sistema requiere que son objeto de estudio para el proyecto y que van a servir para almacenar información y establecer la relación entre ellas.

En el diagrama entidad relación plasmamos los requerimientos del cliente para poder obtener una aplicación adecuada.

El modelo Entidad - Relación está formado por:

Entidades
Relaciones
Atributos

Entidades

La entidad es el elemento básico del diagrama Entidad - Relación. Es un elemento importante sobre el cual requerimos cierta información.

Modelado de Entidades

Para obtener el modelado de las entidades necesitamos seguir los siguientes pasos:

Examinar los sustantivos, importantes de las entrevistas con el cliente.

Para poder incluirla como entidad en nuestro diagrama Entidad - Relación es importante hacernos la siguiente pregunta:

¿Existe información de interés para la compañía acerca de la entidad?

Esto nos permite evitar incluir entidades adicionales.

Al seleccionarla, nos tenemos que cuestionar acerca de lo siguiente:

¿La entidad es identificable de manera única?

¿Cuáles atributos sirven como UID (identificador único)?

Se requiere describir la descripción de la entidad, en el diccionario de datos.

Una entidad se representa por un cuadro de esquinas redondeadas, con un nombre en singular y con letras mayúsculas

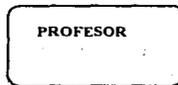


Fig. 1.8 Representación de una Entidad

Relaciones

Una relación es lo que asocia una entidad con otra.

Convenciones para Nombres de una Relación

Una relación se representa por una línea entre una entidad y otra
Cada dirección de una relación tienen:

1. Un nombre
2. Una opción
3. Un grado

Tipos de Relaciones

El tipo de relaciones se distingue por su cardinalidad, es decir el grado, entre las entidades.

Las relaciones que existen son:

Uno y solamente uno ó uno o más

Modelado de Relaciones

Para definir las relaciones debemos seguir los siguientes pasos:

Concentrarse en los verbos.

Definir si la relación es de interés para el sistema.

Definir el tipo de relación.

Dibujar la relación en el diagrama Entidad - Relación.

Atributos

Un atributo es el que califica, identifica, clasifica, cuantifica o expresa el estado de la entidad. Una entidad contiene uno o más atributos. Si un atributo tiene atributos propios, entonces es una entidad.

Un atributo debe tener un nombre concreto a lo que define, este debe ir en singular, el cual no debe ser el nombre de la entidad. Debe existir un valor por atributo para cada ocurrencia de un tipo de entidad. Para cada atributo es necesario definir su opcionalidad.

Convenciones para Mostrar los Atributos en un Diagrama

Este debe ser esquematizado, siempre y cuando sea necesario para el entendimiento del diagrama Entidad - Relación, Este debe tener un símbolo que indique su opcionalidad o obligatoriedad.

SIMBOLO	SIGNIFICADO
*	Obligatorio (El atributo debe siempre contener un valor)
o	Opcional (Puede no tener valor)

Identificadores Unicos

Los identificadores únicos son los que identifican, a una entidad como única. Con estos es posible, identificar un alumno entre otro, aunque ambos tengan el mismo nombre. Se le conoce como identificador único o llave primaria.

Un identificador único puede estar formado por uno o más atributos y es representado por el símbolo "#".

Modelado de Atributos

Para descubrir los atributos debemos seguir los siguientes pasos:

Concentrarse en adjetivos o adverbios

Definir el alcance del atributo, esto es, si este es de interés para el sistema.

Incluir el atributo en el diagrama Entidad - Relación.

Una vez encontrados todos los atributos de una entidad debemos determinar cual será su

identificador único.

Normalización

En la normalización se debe evaluar y validar los atributos y las relaciones para identificar cada entidad. Normalizar el Modelo de Datos nos sirve para validar cada atributo usando las siguientes reglas de normalización.

Las tres formas normales más importantes son:

- 1º. Cada uno de los campos que componen la tabla debe depender de su llave primaria
- 2º. Cada uno de los campos que componen la tabla debe depender de su llave primaria compuesta.
- 3º. Debe de haber independencia entre campos.

Diagrama Jerárquico Funcional (DJF)

El modelo de datos jerárquico es muy similar al diagrama de tipo de red. Una base de datos jerárquica se compone de un conjunto de registros conectados entre sí por medio de ligas. Un registro, igual que en el modelo de red, es un conjunto de datos o atributos que almacenan cada uno un valor. Una liga es una asociación entre dos registros. Los registros en el modelo de datos jerárquico se organizan formando conjuntos de árboles a diferencia del modelo de red en donde se forman gráficas arbitrarias.

Función del Sistema

Una función de un sistema de información es una actividad que será ejecutada para ayudar a cumplir la misión del negocio y de esta forma lograr sus objetivos.

Una función puede ser dividida, es decir, descompuesta en una serie de funciones, cada una de las cuales es dividida de la misma manera. El resultado es una jerarquía de funciones. El principio de la descomposición de funciones consiste en definir una función general que muestre una visión global del sistema

Se pueden detectar funciones en las notas de entrevistas y otras fuentes, escribiendo una función general que describa al sistema como un todo y dividiéndolo en varias partes.

El producto es un modelo jerárquico de funciones el cual consiste de un diagrama con las funciones y la información necesaria para especificar cada función. Un diagrama jerárquico no muestra la secuencia en que las funciones son ejecutadas.

Definición de Funciones

- Empezar con un verbo
- Verbos entendibles para el usuario.
- Evitar verbos ambiguos como procesar, manejar.
- Referirse primeramente a lo definido en el modelo Entidad - Relación como entidades, atributos, relaciones.
- Ser conciso pero entendible
- Definir las condiciones bajo las cuales la función operará.

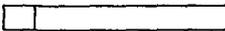
Convenciones Para Dibujar el Diagrama Jerárquico Funcional

- Dibujar cada función en una caja.
- Usar texto normal para la descripción de la función.
- Empezar la descripción de la función con un verbo y tratar de usar una frase explicativa.
- Si se desea descomponer la función posteriormente, dibujar tres asteriscos en la parte superior derecha de la caja.
- Mostrar la referencia de la función sobre o al lado izquierdo de la caja.
- Dibujar las funciones comunes con una línea vertical a un lado de la descripción de la función dentro de la caja.
- Mostrar la etiqueta de identificación de la función maestra en la parte derecha superior de la caja en cada una de las funciones copia.

Diagramas de Flujo de Datos (DFD)

En el caso del Análisis una técnica muy utilizada son los diagramas de flujos de datos (DFD's). Los DFD's especifican un proceso indicando las entradas y las salidas de datos. Los DFD's representan proceso automáticos o manuales. Metodología que utiliza un esquema top-down

Símbolos:



Rectángulo abierto: almacenamiento de datos (Base de datos)



Cuadrado: Fuente o destino de datos



Círculo: Proceso que transforma datos



Flecha: Flujo de los datos

Es la representación gráfica de los orígenes y destinos de un sistema de información mediante símbolos plenamente definidos. El objetivo de estos diagramas es mostrar como fluye la información en el sistema, suministrando a las funciones los datos requeridos.

Los DFD son de dos tipos o niveles:

Nivel 0 / Conceptual (general)

Nivel 1 / Detalle

Diccionario de Datos

El diccionario de datos forma parte de la documentación y contiene todas las definiciones de los elementos en el sistema en forma detallada –flujo de datos, procesos y almacenes de datos–. Si algún miembro del equipo encargado del proyecto desea saber alguna definición del nombre de un dato o el contenido particular de un flujo de datos, esta información debe encontrarse disponible en el diccionario de datos.

1.4 Diseño

Durante la etapa de diseño se tomarán los requerimientos detallados de análisis para realizar el diseño lógico del sistema, el cual está formado por el diseño de la base de datos, que muestra la forma como se almacenará la información, el diseño de la interfaz de usuario, que muestra el diseño de las pantallas, y el diagrama de estructuras, que se apoya en el diseño descendente para representar gráficamente los módulos que conforman el sistema.

1.4.1 Diseño de la base de datos

Durante el diseño de la base de datos se planea como las entidades y los atributos del sistema llegarán a ser tablas y columnas. Empezando con un diseño conceptual normalizado, representado por un modelo de entidad, se puede producir un diseño lógico inicial de la base de datos.

Producir un diseño involucra varios pasos interrelacionados:

- Escoger las llaves primarias
- Convertir entidades en tablas
- Convertir atributos en columnas
- Convertir relaciones en llaves foráneas
- Escoger candidatos a índices

Otros puntos que se deben considerar en el diseño son:

Definición de campos

Desnormalización

Datos derivados

El objetivo de aplicar las reglas de normalización, es obtener un modelo de la base de datos sin redundancia de información. Con esto se evita desperdiciar recursos de almacenamiento y tener tiempos de respuesta grande.

El tiempo de respuesta, es el tiempo en el cual el sistema nos devuelve la información. Si se requiere mejorar este tiempo aplicaremos la desnormalización y tendremos información redundante sólo de aquello que sirva para agilizar las consultas, es decir, tendremos una "redundancia controlada de información".

1.4.2 Diseño del sistema

Diseño de la Interfaz del Usuario

La interfaz es el medio por el cual el usuario se comunicará con la computadora y está formado por pantallas. Las pantallas muestran las formas de captura y los reportes, es decir, son el medio por el cual se desplegará la información.

El objetivo principal en el diseño de una interfaz que ayude a los usuarios y a sus empresas a obtener o introducir información al sistema y que satisfagan los siguientes objetivos.

Eficacia, el usuario debe tener acceso al sistema, de tal forma que sea congruente con sus necesidades particulares.

Eficiencia, al desarrollar interfaces que mejoren la velocidad de captura de los datos y reduzcan los errores.

Consideración del usuario, es decir, debe existir una retroalimentación para saber la manera en que progresa su trabajo.

El diseño de la interfaz del usuario es formado por el diseño de formas, pantallas y reportes. Al cumplir con los lineamientos de cada uno se estará cumpliendo con los objetivos del diseño de la interfaz del usuario.

Un buen diseño de las formas y las pantallas de entrada debe tomarse en cuenta lo que se requiere como entradas del sistema y cuando se llega a una comprensión de la manera en que el usuario responderá a los diferentes elementos de las formas y las pantallas.

Diseño de las Formas

El analista debe contar con la habilidad para diseñar de manera integral una forma que resulte útil, para lograr tal objetivo debe seguir los cuatro lineamientos para el diseño de formas:

Diseñe formas fáciles de llenar.

Asegúrese que las formas cumplan con el propósito para el cual fueron diseñadas.

Diseñe formas que aseguren un llenado preciso.

Mantenga las formas atractivas

El diseño de la forma con un flujo adecuado minimiza el tiempo invertido y el esfuerzo realizado por los empleados en el llenado. Las formas deben seguir un flujo de izquierda a derecha, y de arriba hacia abajo.

Una segunda técnica que facilita el llenado correcto de las formas consiste en la agrupación lógica de la información. Las siete secciones que le confieren solidez a una forma son:

Secciones presentes en una forma bien diseñada

El uso de un rotulado claro es una técnica que facilita llenar la forma. Los rótulos le indican a las personas qué anotar en un espacio en blanco, en un renglón o en un recuadro. Se dispone de las siguientes alternativas para rotular.

Los rótulos de línea pueden encontrarse a la izquierda de áreas en blanco y en el mismo renglón, o bien pueden imprimirse debajo de la línea donde se registrará el dato. La ventaja de ubicar rótulos debajo de las líneas es que se dispone de más espacio en tal línea para el dato. Otra forma de rotular es proporcionar un recuadro para los datos, en lugar de la línea. Los rótulos pueden ubicarse dentro, fuera o debajo del recuadro. Los recuadros auxilian a la gente a introducir los datos en el sitio correcto y también facilitan la lectura del receptor de la forma. Los cuadros de selección son más convenientes cuando el número de alternativas de respuesta se encuentra necesariamente restringido y las tablas son cuando se requieren detalles.

Las formas no deben aparecer amontonadas, sino más bien, deben dar una apariencia de organización y lógica, aún después de llenarse.

Diseño de Pantallas

Para el buen diseño de pantallas existen cuatro lineamientos:

- Mantenga la pantalla sencilla
- Mantenga una presentación consistente
- Facilite el movimiento del usuario entre pantallas
- Cree pantallas atractivas

La pantalla debe mostrar sólo lo que es necesario para la acción particular que se lleva a cabo. Ésta se divide en tres secciones: la primera de ellas se llama encabezado y se localiza en la parte superior, sirve para indicar al usuario donde se encuentra dentro de la aplicación; en la sección del cuerpo, es donde el usuario realizará la captura de los datos y finalmente la sección de comentarios, donde puede ir un menú de palabras clave acerca de las funciones principales del sistema.

Secciones de una pantalla

Las pantallas deben atraer al usuario y mantener su atención. Esto se logra con el uso de espacios abiertos que rodeen los campos de captura de datos, de tal forma que la pantalla no se vea sobrecargada. Nunca se debe saturar una forma, así como nunca se debe saturar una pantalla. Siempre será mejor utilizar pantallas múltiples, que amontonar todo en una sola.

Al plantear sus pantallas utilice un flujo lógico. Organice el material de la pantalla para tomar ventaja de la forma en que la gente trabaja y así facilitarles su ubicación en la pantalla.

Para crear pantallas atractivas podemos hacer uso de diferentes tipos y tamaños de letras, usar imágenes y colores. Existen ciertos lineamientos para el uso de imágenes, una de ellas es utilizar las imágenes típicas que los usuarios puedan interpretar fácilmente.

Diseño de Reportes

Los reportes son la información de salida que requieran los usuarios, por tal razón, juzgan al sistema dependiendo de los resultados que otrezca.

Cuando se diseña un reporte se deben cumplir los siguientes objetivos:

- Diseñar un reporte para satisfacer el objetivo planteado
- Diseñar un reporte que se adapte al usuario
- Provee la cantidad adecuada de información
- Asegura que el reporte este disponible donde se necesite
- Proporciona oportunamente el reporte

Durante la fase de análisis de determinación de los requerimientos, el analista de sistemas identifica los propósitos a satisfacer y con base en tales propósitos se diseña el reporte. Éstos deben diseñarse con base a las necesidades de los usuarios y sólo deben de proporcionar la

cantidad de información adecuada.

Al diseñar el reporte el analista de sistemas incorpora elementos tanto funcionales como estéticos, de tal forma que el informe proporcione al usuario la información que requiere en un formato legible.

Los atributos funcionales de un reporte incluyen el encabezado o título del reporte, el número de la página, la fecha del reporte, las etiquetas en las columnas, el agrupamiento de los datos relacionados y el uso de cortes de control, los cuales son separaciones entre los datos cuando aparece una sumarización.

Etapa de Desarrollo

Una vez terminada la etapa de diseño tendremos todos los elementos para poder desarrollar el sistema de información. El diseño de la base de datos nos ayudará para crear físicamente las tablas que forman el sistema y la relación que existe entre ellas; el diseño de la interfaz nos dará una guía para saber que pantallas componen nuestro sistema y empleando alguna herramienta de desarrollo procederemos a construir las. Además tendremos que programar las líneas de código necesarias para hacer que los eventos que se encuentran en el sistema respondan de manera correcta.

1.5 Manejadores de Bases de Datos

Un manejador de base de datos (DBMS, Database Management System) o back-end, es un conjunto de programas que se encargan de la administración de una o más bases de datos. El objetivo primordial de un DBMS es crear un ambiente en que sea posible guardar y recuperar información de la base de datos en forma conveniente y eficiente.

Los manejadores de base de datos son utilizados para manejar grandes cantidades de información. El manejo de los datos incluye tanto la definición de las estructuras para el almacenamiento como los mecanismos para el manejo de la información. Además, el manejador de base de datos debe cuidar la seguridad de la información, contra las caídas del sistema así como contra los intentos de acceso no autorizado.

Se encarga de las siguientes actividades:

- Interacción con el manejador de archivos
- Implantación de la integridad
- Almacena, obtiene y modifica los datos
- Puesta en práctica de la seguridad
- Respaldo y recuperación
- Control de consistencia

1.5.1 Características de un Manejador de Base de Datos Relacional

Un manejador de base de datos relacional (RDBMS, Relational Database Management System) tiene las siguientes características:

- Representación de los datos por medio de tablas relacionadas
- Utiliza un lenguaje de cuarta generación (4GL)
- Flexibilidad
- Modificación de los datos y
- los cambios a la estructura de la base de datos resultan muy sencillos
- Contiene un diccionario de datos

¿Qué manejador de base de datos usaremos?

En ocasiones el usuario dirá cual, en otras tendremos que decidir cuál es el más conveniente.

Los parámetros que influyen generalmente en la elección del manejador de base de datos son: la plataforma, el equipo que se tiene, la cantidad de información que se almacenará y el costo, en caso de adquirirse éste último.

1.6 Herramientas de Desarrollo

Existe en el mercado un gran número de herramientas para las distintas fases del proceso. A la hora de analizar diferentes herramientas, es necesario tener en cuenta, entre otros, parámetros como los siguientes:

- Entorno físico sobre el que funciona
- Sistema operativo sobre el que opera
- Gestores de bases de datos o fuentes a los que es aplicable
- Grado de especialización
- Profundidad del análisis que realiza
- Información adicional que aporta
- Necesidades de formación que exige a sus usuarios
- Costo económico
- En la fase de estudio habrá de solicitarse a los proveedores las oportunas demostraciones
- Técnicas que permitan evaluar la herramienta adecuadamente
- Por último, siempre existe la posibilidad de fabricar una propia que contemple las particularidades del entorno sobre el que va a utilizarse.

Visual Basic, Microsoft Access, Power Builder, Delphi

¿Que es una Herramienta de Desarrollo?

Las herramientas de desarrollo o front-ends, son programas que sirven principalmente para el desarrollo de interfaces gráficas.

Se pueden clasificar en cuatro categorías: add-ons a productos ya existentes, los cuales son módulos que permiten que una aplicación de PC consulte al servidor de la base de datos, ejemplos de add-ons son los existentes para Dbase, Paradox, Access, Clarion, y hojas de cálculo como Lotus o Excel; herramientas de desarrollo de aplicaciones, son usadas principalmente por los programadores y están diseñadas para facilitar el proceso de creación de aplicaciones front-end; reportadores, permiten hacer consultas no planeadas a la base de datos, facilitan la creación de consultas y reportes al back-end a los no programadores; y las herramientas de análisis e integración de datos, que están diseñadas para que los tomadores de decisiones examinen los datos a partir de diferentes fuentes para así, construir cuadros de decisiones complejas.

1.7 Protocolos de Comunicación

TCP/IP (Transmission Control Protocol/Internet Protocol)

Hay 4 grandes grupos de protocolos:

- a. Protocolos de gestión del sistema: todas las tareas relacionadas con el sistema de interconexión de los equipos en la red, por ejemplo, el protocolo TCP/IP
- b. Protocolos de gestión de la aplicación: procesos que controlan la asignación de recursos del equipo, bloqueos de la pc, accesos no autorizados,...
- c. Protocolos de sistema: tareas del sistema operativo, tales como los accesos a discos y ficheros, la gestión de la memoria, la ejecución de tareas,...
- d. Protocolos específicos de aplicaciones: dependen del tipo de aplicación con la que vayamos a trabajar.

TCP/IP

El TCP/IP (Transmission Control Protocol/Internet Protocol) es un conjunto de protocolos, que proporcionan una serie de reglas para la comunicación entre computadoras.

Es una forma de resolver la cuestión de comunicación entre nodos que pertenecen a redes con distintos niveles lógicos y de enlace, por ejemplo, entre un nodo de una red Ethernet y otro nodo de una red de tipo token ring) Son los protocolos de interconexión de redes más utilizados en la actualidad y son la base de la red de Internet.

La principal característica del TCP/IP es que establece la comunicación por medio de paquetes de información.

Internet Protocol (IP):

IP actúa en el nivel 3 del modelo OSI y permite el intercambio de mensajes entre redes que no comparten los mismos niveles físico y de enlace de datos (niveles 1 y 2 del modelo OSI). Al conjunto de datos transmitidos por IP se le denomina " datagrama IP ". Cada datagrama IP posee información del destino, de forma que puede llegar automáticamente al nodo de destino. El mecanismo de conseguirlo es dotar a cada nodo de una dirección IP única.

En el esquema de IP existen dos tipos de dispositivos (nodos):

- **Host:** nodos pertenecientes a una subred
- **Róuter:** nodos que enlazan dos subredes distintas

Dirección IP:

Consiste en 32 bits que normalmente se expresan en forma decimal, en cuatro grupos de tres dígitos separados por puntos. Esos dígitos irán de 0 a 255.

Máscara de subred.

La máscara de subred es un conjunto de 32 bits que permiten separar de una dirección IP la parte del identificador de la subred de la parte del nodo.

Asignación dinámica de direcciones IP (Servidor DHCP). El servidor DHCP tiene un juego de direcciones IP disponibles que va asignando a cada nodo dinámicamente según se las van pidiendo.

Servidores Proxy. Un único nodo (el servidor proxy) tiene una dirección IP "real" y los demás utilizan direcciones falsas.

- Espacio de direcciones ampliado. Se utilizarán direcciones de 126 bits en lugar de 32 bits.
- Capacidades de seguridad. Incluye características que permiten la autenticación de los mensajes y la privacidad.

Asignación dinámica de direcciones IP. Ipv6 incluye dicha asignación.

DNS

Para comunicarse con un nodo mediante IP es necesario conocer su dirección IP, que al ser un número no es fácil de recordar. Por ello surgió un proyecto denominado DNS (Domain Name System o Sistema de Nombres de Dominio), que establece una relación entre una dirección IP y un nombre de host, haciéndolo más fácil de recordar.

La relación entre IP y nombre de host está almacenada en unos nodos denominados servidores DNS. De esta forma, si se quiere utilizar el sistema de nombres de dominio es necesario conocer el nombre del host y la dirección IP de algún servidor de DNS que nos pueda proporcionar la relación entre direcciones IP y nombres de dominio.

Otra ventaja de la DNS es que es independiente de la subred, es decir, si cambiamos un nodo de una subred a otra, en general deberemos cambiar su dirección IP, mientras que no será necesario cambiar su nombre de dominio.

El protocolo TCP está formado por dos protocolos, el TCP y el UDP. El TCP permite establecer un circuito virtual entre dos nodos (protocolo orientado a conexión), mientras que el UDP permite el envío de un segmento entre nodos sin comprobar que el envío ha sido satisfactorio (protocolo no orientado a conexión).

Las funciones del protocolo TCP son:

- Segmentación de los mensajes: TCP decide cómo trocear el mensaje que le llega de las capas superiores y cómo formar el segmento.

- **Etiquetado de procedencia:** TCP etiqueta cada segmento indicando el lugar que ocupa en un mensaje completo.
- **Control de flujo:** TCP permite la eliminación de duplicados y la petición de reenvío de los segmentos que falten.
- **Indicación de los datos urgentes:** es posible indicar si los datos que porta el segmento son urgentes o no.
- **Confirmación de la llegada de datos:** permite confirmar la recepción de los segmentos en el destino.
- **Direccionamiento de los servicios de red:** permite encaminar mensajes a un servicio de nivel superior u otro mediante el empleo de puertos. De esta manera, es posible distinguir servicios distintos recibiendo y enviando mensajes dentro de un mismo nodo.

1.8 Desarrollo de Aplicaciones CLIENTE/SERVIDOR

La tecnología Cliente/Servidor puede definirse como un conjunto de elementos de software como de hardware, entre los cuales se destacan tres tecnologías: El Cliente, El Servidor y La Red. El Servidor central quien acepta y procesa los requerimientos de otro elemento llamado Cliente, quien es el encargado de recibir el resultado del proceso; estos dos elementos son unidos por medio de una Red de comunicaciones.

El Cliente

Es el elemento encargado de interactuar directamente con el usuario final. Mediante éste el usuario realiza el acceso a la información sin importar el lugar en donde se encuentre. El Cliente maneja la presentación de los datos, realiza la captura y la validación de los mismos, genera consultas ejecuta operaciones y recibe información procedente del Servidor o de otro Cliente. Por lo tanto el Cliente debe contar con una gran capacidad de procesamiento, y debe poseer una interface amigable para el usuario final. Una interface Gráfica de Usuario (GUI) es la ideal para un Cliente, ya que le permite realizar operaciones complejas mediante labores sencillas como oprimir botones, los cuales están ubicados en la pantalla gráfica; teniendo esto como consecuencia, que los usuarios finales no necesiten conocimientos profundos sobre computación.

El Servidor

El Servidor es el encargado de satisfacer los requerimientos del Cliente. Procesa las consultas del Cliente, envía, recibe y almacena información, provee seguridad y control de acceso. Existen varias clases de servidores: de datos, de correo electrónico, de imágenes, de impresión, entre otros. Los servidores deben contar con elementos que gestionen los datos, esto se lleva a cabo mediante un (DBMS) Sistema Manejador de Bases de Datos, que permita una transparencia de acceso, de distribución y de integridad a todas las transacciones de la base de datos.

Entre los Servidores se puede contar con máquinas como AlphaServer, AS400, RS/6000, Servidores PC's entre otros. Y entre los sistemas operativos de los Servidores: Windows NT, OpenVMS, Gateways, AIX, SunOS/Solaris, UNIX, Netware, etc.

La Red

La Red es elemento encargado de realizar la transmisión de los requerimientos del Cliente al Servidor y del Servidor al Cliente. También controla la transmisión de datos entre los diferentes servidores que conformen el ambiente. La red debe proveer mucha seguridad, de tal forma como si estuviera trabajando de una manera local. La Red necesita de un propio Sistema Operativo para así ofrecer varios niveles de seguridad, tolerancia a fallas, alto rendimiento, herramientas de administración y sobre todo transparencia al Usuario final. Para administrar la Red existen varios sistemas y protocolos como: AppleTalk, LAN Manager, Netware, VINES, TCP/IP, Ethernet, etc.

Principales Ventajas

- **Control Centralizado:** el usuario final tiene el control sobre todos los clientes de la red, de otra parte el administrador del sistema ejerce el control sobre el servidor y sobre la red, permitiendo mantener la seguridad en la base de datos. Un Cliente podrá hacer las veces de Servidor en el momento que se requiera.
- **Sistemas Abiertos:** soporta múltiples ambientes, múltiples plataformas, múltiples manejadores de bases de datos. Permitiendo la adquisición de hardware y software sin pensar en su compatibilidad.
- **Flexibilidad y Escalabilidad:** permite reemplazar, ampliar o agregar componentes sin necesidad de realizar grandes cambios a la aplicación, facultando a las empresas de una posición ventajosa en el mercado frente a sus competidores.
- **Incremento de la Productividad:** con las plataformas amigables, los usuarios podrán emplear menos tiempo en la realización de las tareas que antes eran dispendiosas. La disponibilidad de nuevas tecnologías de programación (P.O.O, SQL, Herramientas Case, etc.) permite la implantación de Aplicaciones de Desarrollo Rápido (RAD), contribuyendo esto a una pronta satisfacción de las nuevas necesidades de la empresa. También facilita la toma de decisiones, ya que la información es oportuna y se encuentra disponible. Entre las RAD's se encuentran: Power Builder, Delphi, Gupta, Visual C++, Sapiens, Uniface, Powersoft, entre otros.
- **Reducción de Tráfico:** la Red se descongiona por que la manipulación de los datos ocurre en el Cliente y en el Servidor, dependiendo de cuál sea la forma más efectiva para cada tarea. La base de datos se encuentra distribuida entre el Cliente y el Servidor. La red dedica mayor tiempo a transportar los resultados y no las consultas.

1.8.1 Arquitectura CLIENTE/SERVIDOR de dos capas

Esta arquitectura propone centralizar los procesos, como control de acceso, conexión a la base de datos, ejecución de funciones, etc.

La arquitectura se puede representar de la siguiente manera:

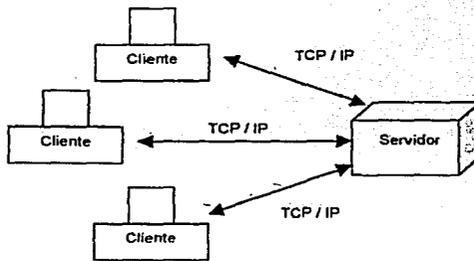


Fig. 1.9 Arquitectura Cliente/Servidor de dos capas

En la arquitectura Cliente/Servidor el servidor recibe las peticiones de los clientes y se encarga de establecer las conexiones necesarias con la base de datos (tantas como clientes existan), siempre y cuando el cliente este autorizado (autenticación).

Ya establecida la conexión los clientes a través del Front End (interfaces que permiten interactuar al usuario con el sistema), los procesos solicitan y reciben y despliegan los datos extraídos de la base de datos en el servidor.

Por lo tanto, el servidor debe estar constituido por los siguientes módulos:

- **Listener** (receptor de peticiones) Se encarga de escuchar a través de un puerto a los clientes que requieren conectarse, éstos envían su solicitud y el servidor establecen una conexión dedicada con ellos.
- **Método de Autenticación.** El cliente envía información de usuario y contraseña de tal manera que se pueda validar el usuario, de ser así se inicia la sesión.
- **Administración de Recursos.** Ya iniciada la sesión, el servidor es el responsable de proveer los datos que requieren los procesos que el cliente está ejecutando, hasta que finalmente el cliente termina la sesión.

Ventajas

- Esta arquitectura puede ser implantada en ambientes locales, Intranet e Internet
- Permite centralizar el almacenamiento de datos
- Maneja auditorías de usuarios.

Desventajas

- El proceso de autenticación depende exclusivamente del servidor de bases de datos.
- El procesamiento de datos se realiza directamente en el servidor.
- Resuelve conexiones con los clientes y a la base de datos.

1.8.2 Arquitectura CLIENTE/SERVIDOR de tres capas

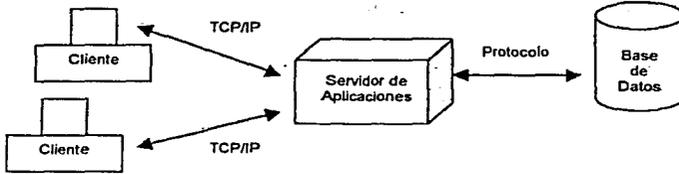


Fig. 1. 10 Arquitectura CLIENTE/SERVIDOR Tres Capas

Esta arquitectura conocida como de tres capas: cliente servidor de aplicaciones y servidor de bases de datos, permite que las capas sean especializadas en funciones; de tal manera que provee a la arquitectura los siguientes módulos:

Cliente

- Es más ligero, ya que solo se encarga de construir y enviar peticiones al servidor a través de métodos con parámetros; es decir no hay procesamiento en el cliente.
- Recibe los resultados de los procesos solicitados y los despliega.

Servidor de Aplicaciones

- Se encarga de la autenticación de los usuarios y de la identificación de sus perfiles.
- Ejecuta los procesos solicitados a través de los métodos enviados por los clientes.
- Resuelve las conexiones con la base de datos, a través de un spool de conexiones, identificando si es necesario abrir o cerrar sesiones, dependiendo de la demanda a la base de datos por los procesos que se están ejecutando en el servidor.

Servidor de Bases de Datos

- Atiende peticiones de inserción, actualización, eliminación y consulta de datos para procesos específicos

Ventajas

- A diferencia de la arquitectura de dos capas, en este caso no resulta tan sencillo violar las reglas del sistema ya que tenemos dos procesos de autenticación: uno en el servidor de aplicaciones y otro en el servidor de bases de datos. Dada la separación explícita de funciones, se reduce la posibilidad de que algún cliente pudiera causar daño al sistema.

- La ejecución de todos los procesos es del lado del servidor, dando acceso al sistema a clientes cuyos recursos sean mínimos. Los requerimientos de hardware y software se concentran en el servidor.
- El servidor de aplicaciones y el de base de datos pueden instalarse en distintas máquinas.

Desventajas

- El monitoreo y la administración del sistema es complejo, debido al incremento de puntos de falla derivado del número de componentes.
- El desarrollo de la infraestructura del sistema (arquitectura) es más sofisticada.

1.9 Seguridad

Para mantener la seguridad de la información se establecen políticas generales y controles que previenen o detectan cualquier intento de acceso no autorizado a un sistema de computo.

Para evitar los riesgos en la seguridad de la información, se debe considerar lo siguiente:

- Proteger los datos de cualquier uso indebido
- Definir y asignar niveles de control de acceso
- Monitorear los sistemas para detectar cualquier uso indebido

Las normas y lineamientos que se deben seguir para mantener la seguridad en los Sistemas de Información y evitar la mala manipulación de los datos por las personas no autorizadas en el manejo de estos, se lista a continuación.

Privacía de la Información

La información puede ser recopilada de varias fuentes pero ésta deberá usarse únicamente para los fines que la Dependencia estipule y no dejar que usuarios ajenos tengan acceso a estos.

Integridad de la Información

El término integridad se usa para referirse al problema de asegurar la corrección de la información (cobra una relevancia especial en el mundo de las bases de datos). Los valores en un banco de datos se pueden volver inválidos de varias maneras. Probablemente la causa más común es una entrada de datos o una actualización en la cual algunos valores son especificados incorrectamente. Por ello cuando se lleve a cabo la actualización en un dato, está actualización deberá llevarse a cabo en todos los datos relacionados con él.

Confiability del personal

El mayor daño que puede sufrir un centro de computo es el que se hace dentro; ni siquiera las fortalezas o los equipos, más sofisticados y costosos pueden contra la deslealtad, la deshonestidad o la negligencia. Los empleados descontentos, o los que recientemente han sido despedidos de la

Dependencia representan un riesgo mayor. Por lo tanto, la selección de personal es parte importante del esquema integral de seguridad.

Los errores en los datos son otra fuente de riesgos, ya que los sistemas de Información no pueden ser buenos si la información que manejan no es correcta, por esta razón se debe llevar a cabo:

- Una capacitación especificando los tipos de datos que se manejan en los Sistemas.
- De donde y como obtener los datos para capturarlos en el Sistema
- Como introducirlos al Sistema y que hacer en caso de encontrar un dato desconocido

Esto se desarrolla con el fin de disminuir el margen de errores que posiblemente el usuario pueda cometer.

Capítulo 2. Definición del Problema

En este capítulo se define la problemática que enfrenta el Centro; se dan a conocer las áreas involucradas y se reconocen sus requerimientos.

2.1 Problemática Actual

A través de un profundo análisis del proceso actual y del sistema en operación se detectaron las siguientes carencias:

El centro de computo no tiene su información centralizada y existen varios cálculos que se realizan manualmente, por lo cual pueden tenerse variaciones de datos.

El problema fundamental radica en la falta de control en su información, en el Departamento de Informes y Relaciones se maneja la información de manera inconsistente, los datos que utilizan están contenidos en tablas aisladas de Microsoft Access 97, que no tienen relación alguna entre sí, y mucho menos un análisis ni diseño que constituyan una base de datos relacional.

El control de las aulas se da en forma manual, la asignación de estas, esta dada por las características del curso.

Uno de los problemas que enfrenta, es el control del número de solicitudes que fueron aceptadas en el departamento de informes y relaciones, y cuantas fueron pagadas en caja.

Cabe destacar, que el Centro de Computo depende directamente de las políticas manejadas por DGSCA CU, con respecto a las claves, temarios y nombres de los cursos, formatos de entrega de reportes, constancias de los alumnos, archivos en disco, periodos de entrega, etc., por lo que el Sistema deberá cumplir con dichos requerimientos.

2.2 Departamentos y sus Requerimientos de Información

2.2.1 Descripción de las Áreas Involucradas

Para conocer más acerca del centro describiremos a continuación los departamentos que forman parte de este. La siguiente figura nos muestra como está el organigrama de esta institución:

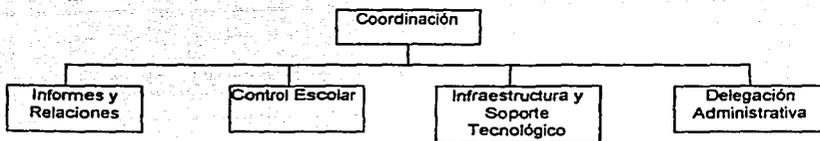


Fig.-2. 1 Organigrama del Centro de Computo

Departamento de Control Escolar

Es el organismo encargado de organizar las actividades de control escolar para los diferentes programas de capacitación, formación y actualización en cómputo, ofrecidos a la comunidad universitaria y público en general.

El departamento de control escolar también es el encargado de realizar las siguientes actividades:

- Elaboración de calendario de cursos.
- Contratación de profesores.
- Evaluación y seguimiento de cursos y profesores.
- Elaboración de constancias para alumnos y diplomas para los profesores.
- Elaboración de trípticos informativos.
- Capacitación a profesores en materia didáctica.
- Apoya administrativamente a la Dirección de Cómputo para la Administración Académica, al Departamento de Cómputo Infantil y al Departamento de Educación a Distancia.

Departamento de Informes y Relaciones

En este departamento se brindan los siguientes servicios:

- Información sobre los cursos que se imparten en la DGSCA.
- Información sobre la realización de exámenes en caso de tener conocimientos previos en cómputo.
- Inscripción de los diferentes cursos que se imparten en la DGSCA.
- Entrega de calificaciones y/o constancias de los cursos tomados.
- Orientación sobre otros servicios que ofrece la DGSCA.
- Orientación sobre los cursos que debe tomar de acuerdo con sus conocimientos o intereses.

Departamento de Infraestructura.

El Departamento de Infraestructura y Soporte Tecnológico se encarga de brindar, supervisar y controlar los diversos servicios (sala de microcomputadoras y terminales, aulas equipadas, visitas guiadas y revisión de discos infectados) a estudiantes y profesores.

Estos servicios los brinda a través de:

1-Cubículo de supervisores

Esta área se encarga de proporcionar los servicios requeridos para el buen funcionamiento del equipo existente en las aulas donde se imparten los cursos de cómputo. Entre las labores realizadas destacan las siguientes:

1. Vacunar los disquetes de los alumnos o público en general con objeto de prevenir la existencia de virus.
2. Proporciona copia de las vacunas disponibles.
3. Establecer un contacto frecuente con los profesores que imparten los cursos correspondientes en cada periodo, con la finalidad de detectar y/o en su caso, solucionar todos aquellos percances que se hayan suscitado durante la impartición del curso.
4. Efectuar la instalación oportuna de la paquetería necesaria para la realización de los cursos programados en cada aula siguiendo los lineamientos establecidos.
5. Optimizar los recursos del equipo de cómputo instalado en las aulas.
6. Llevar a cabo la supervisión del mantenimiento del equipo de cómputo (CPU, monitor, teclado y mouse).
7. Supervisar la instalación de software para los cursos impartidos en esta dependencia.
8. Coordinar la supervisión y control de la asignación de los recursos materiales para impartir cursos.

Delegación Administrativa.

Administra los recursos que capta el centro por concepto de cursos, así como los gastos de todo el centro y el material que requiere para su funcionamiento. La Delegación Administrativa, es el departamento encargado de la caja, aquí se generan los recibos por concepto de cursos, artículos y otros servicios. Son los encargados del control de pagos de servicios (luz, agua, renta), pago a becarios y profesores.

2.3 Recolección de requerimientos

Un requerimiento es una característica que debe incluirse en el nuevo sistema. Para hacer la recolección de requerimientos visite cada uno de los departamentos y me entreviste con los jefes de departamentos y algunas personas que laboran en este, lo que se obtuvo mediante este proceso fue lo siguiente:

El Departamento de Informes y Relaciones fue el primero que se visitó. La jefa de dicho Departamento nos mostró la manera en que se trabaja actualmente: se maneja la información de manera inconsistente en el sentido de que los datos que utilizan están contenidos en tablas, asiladas de Microsoft Access 97, que no tienen relación alguna entre sí, y mucho menos un análisis ni diseño que constituyan una base de datos relacional. En este sentido es indispensable reestructurar lo que se tiene hecho, debido a que el manejo de datos es complejo e ineficaz.

En el Departamento de Control Escolar se llevan a cabo muchas actividades que pueden ser automatizadas, tales como la impresión de listas de cursos, calendarios, constancias, informes, evaluaciones para los profesores, estadísticas. Cabe destacar que en este Departamento la información se maneja únicamente en hojas de cálculo de Excel 97, además de tener formatos preestablecidos en Word 97, por lo que la información se encuentra aislada en diferentes fuentes (Word, Excel), y no tiene relación directa alguna. La reestructuración del manejo de información resultaría muy conveniente, en cuanto al ahorro de tiempo, integridad de la información, y simplicidad en el manejo de la misma.

El Departamento de Infraestructura y Soporte Tecnológico asigna las aulas manualmente dependiendo las necesidades de Cursos, en cuanto a capacidad, y requerimientos tanto de software como de hardware. En este sentido, sería muy conveniente que dicha asignación fuera automatizada, debido a que al tener la información debidamente almacenada, se evitarían conflictos en el control de aulas, en cuanto a cupo, necesidades, y disponibilidad de las mismas.

En la Delegación Administrativa, como en los demás Departamentos, la información se maneja en tablas de Excel 97, y en formatos definidos de Word 97, por lo que se presenta el mismo problema de inconsistencia de datos; cabe destacar que existe un conflicto entre éste Departamento y el Departamento de Informes y Relaciones, ya que cada uno maneja su información por separado, y cuando se requiere conjuntar y comparar, la inconsistencia de la misma es muy notoria. Aquí se entregan reportes a la Coordinación, los cuales contienen gráficos de Microsoft Excel; estos reportes requieren ser automatizados para un manejo eficiente.

2.4 Alcance

El sistema propuesto deberá conjuntar toda la información en una sola base de datos, de manera que se eviten los problemas actuales de redundancia, inconsistencia, integridad y seguridad que actualmente existen. Los datos serán transmitidos a través de la red interna del centro, por lo que será necesario definir privilegios (permisos) de seguridad para que la información este al alcance únicamente del departamento que la deba tener.

La información deberá estar contenida en una misma Base de Datos, utilizando el modelo Relacional; debido a esto, el análisis y diseño de la Base de Datos, deberá ser muy bien estructurado, para que toda la información que se maneje en el Centro, este contenida ahí. El Sistema deberá asignar privilegios a los diferentes usuarios de cada Departamento, así cada usuario solo tendrá acceso a la información correcta, y podrá desempeñar operaciones de actualización, consulta, y eliminación de datos.

El Sistema deberá ser capaz de automatizar todas las tareas que se llevan a cabo manualmente en los Departamentos, verificar que la información sea consistente y segura, elaborar los reportes que cada departamento requiere, vigilar el acceso y uso al Sistema por cada uno de los usuarios, restringir el uso a usuarios no autorizados, manejar adecuadamente los errores cometidos por los usuarios al utilizarlo, permitir el manejo de información utilizando la Red interna del Centro.

Entre las principales funciones que deberá cumplir, destacan las siguientes: Proporcionar al usuario la información detallada acerca de los cursos impartidos dentro y fuera del Centro; Automatizar el proceso de inscripción a los cursos, diplomados y líneas de especialización impartidos dentro y fuera del Centro, Manejar estadísticas, y controles de ingresos y egresos por conceptos de cursos, y venta de material en el Centro; En base a la demanda de cursos, formar grupos, administrar cursos, aulas, profesores y calendarios; Generar los reportes correspondientes a cada Departamento.

2.5 Soluciones Propuestas

Al identificar los requerimientos del sistema, a continuación daremos unas soluciones que bien pueden resolver la situación actual. Los aspectos a considerar a la hora de proponer una solución son los siguientes:

La decisión en cuanto a la "arquitectura," determinara el tipo de hardware y de software, tanto para el desarrollo como de operación.

¿Qué manejador de base de datos usaremos?

En ocasiones el usuario dirá cual, en otras tendremos que decidir cuál es él más conveniente. Los parámetros que influyen generalmente en la elección del manejador de base de datos son: la plataforma, el equipo que se tiene, la cantidad de información que se almacenará y el costo, en caso de adquirirse éste último.

A continuación se tiene una tabla comparativa en cuanto a los manejadores de bases de datos que existen en el mercado:

Nombre	Descripción	Características
ORACLE	Es un manejador de base de datos relacional utilizado para misiones críticas de negocios. Actualmente por las características que posee lo hace ser él más utilizado por las empresas. Soporta las siguientes plataformas. UNIX, VMS, Windows NT, Windows 95, OS/2, Netware, y actualmente con su nueva versión, Linux.	Seguridad, utiliza sofisticados mecanismos de seguridad para controlar el acceso a los datos a través del manejo de privilegios. Realización de copias de seguridad y recuperación, la estrategia de copias de seguridad y recuperación de Oracle permite minimizar la pérdida de datos y el tiempo de caída del sistema cuando se produce un problema. Conectividad de cliente abierto, permite acceder hacia y desde paquetes de software de otros fabricantes mediante ODBC's y SQL*Net, éste último, es su protocolo de comunicación. Herramientas de desarrollo, su principal herramienta de desarrollo es Developer/2000, el cual incluye un conjunto de herramientas para la creación de aplicaciones: Oracle Forms, Oracle Reports, Oracle Graphics y Procedure Builder; cuenta también con Designer/2000, el cual incluye herramientas para la reingeniería de procesos de negocios, modeladores y generadores. Incluye en su nueva versión la tecnología OLAP, Data Warehouse y el manejo de base de datos objeto - relacional.
SYBASE	Sybase es un manejador de bases de datos relacional utilizado para aplicaciones.	Soporta conectividad con otros manejadores mediante ODBC's. Provee una alta seguridad.

	<p>empresariales. Por su facilidad y la integración entre sus productos, lo ha hecho disputar junto con Oracle el mercado internacional. Puede correr bajo las siguientes plataformas: UNIX, Windows NT, Windows 95, etc.</p>	<p>mediante contraseñas de entrada y el uso de privilegios Proporciona mecanismos de respaldo y recuperación, aún cuando la base de datos este levantada. Cuenta con productos de desarrollo para la creación de aplicaciones cliente/servidor (Power Builder) y aplicaciones en el Web mediante Jaguar de PowerSoft.</p>
<p>INFORMIX</p>	<p>Informix es otro manejador de base de datos relacional, pero no tiene tanto mercado en el ámbito mundial como Oracle y Sybase, sin embargo, es uno de los manejadores de base de datos más utilizados en organismos gubernamentales en México. Soporta las siguientes plataformas: UNIX, Windows 95, Windows NT, OS/2 y Macintosh.</p>	<p>Multiprocesamiento simétrico, el servidor dinámico INFORMIX-OnLine es un poderoso servidor de base de datos de configuración multi-threaded, diseñado para aprovechar al máximo la capacidad del multiprocesamiento simétrico (SMP) y arquitecturas uniprocador. Procesamiento transnacional excelente Gran disponibilidad Capacidad multimedia Alta seguridad Mecanismos de respaldo de la información en cintas y mecanismos de seguridad de la información como discos en espejo Herramientas para el monitoreo de las actividades en la de la base de datos y de los recursos del servidor. Ofrece una gran gama de productos para el desarrollo de sistemas cliente/servidor, tales como: Informix-NewEra, producto para el desarrollo de interfaces de usuario, Informix-NewEra ViewPoint Pro, suite de herramientas de desarrollo gráficas que incluyen productos para la creación de formas, reportes, menús, etc. Conectividad mediante Informix-Net, el cual permite la comunicación entre sus aplicaciones cliente y el servidor de base de datos, y con otros manejadores de bases de datos mediante ODBC's.</p>
<p>MICROSOFT SQL SERVER</p>	<p>Microsoft SQL Server es un manejador de bases de datos relacional multiusuario diseñado para soportar una gran cantidad de transacciones sobre computadoras</p>	<p>Administración de la base de datos en forma gráfica, a través de la herramienta Microsoft SQL Server Enterprise Manager, la cual facilita mucho las tareas administrativas.</p>

	<p>personales, corre solo bajo la plataforma Windows NT. En cuanto a su arquitectura es muy parecido a SYBASE, solo que este, ocupa menos recursos y esta enfocado para sistemas de mediano tamaño.</p>	<p>Ofrece un alto rendimiento sobre el sistema operativo Windows NT.</p> <p>Soporta la arquitectura cliente/servidor, es decir, divide una aplicación en dos partes: el front-end (cliente), el cual provee una interactividad alta y una interfaz fácil de usar, ya que es gráfico -típicamente encontrada en computadoras personales -, y un back-end (servidor), el cual proporciona la administración de datos, información compartida, administración sofisticada, y seguridad.</p> <p>Soporta aplicaciones distribuidas.</p> <p>Alta seguridad, que se ve reforzada con la seguridad que ofrece el sistema operativo Windows NT.</p> <p>Replicación heterogénea, permite replicar la información a otros manejadores de bases de datos, tales como, Oracle y Sybase para mantener la consistencia en los datos.</p> <p>Soporta aplicaciones Internet e Intranet, el Microsoft SQL Server Web Assistant permite a SQL Server generar automáticamente páginas HTML o rellenar plantillas HTML con datos del servidor.</p> <p>Soporta las aplicaciones Data-Warehouse</p> <p>Permite la recuperación y tolerancia de fallos.</p>
--	---	---

Fig. 2.2 Tabla comparativa de DBMS

Herramientas de Desarrollo

En cuanto a las herramientas de desarrollo se tiene que tomar en cuenta lo siguiente: Existe en el mercado un gran número de herramientas para las distintas fases del proceso. A la hora de analizar diferentes herramientas, es necesario tener en cuenta, entre otros, parámetros como los siguientes:

- Entorno físico sobre el que funciona
- Sistema operativo sobre el que opera
- Gestores de bases de datos o fuentes a los que es aplicable
- Grado de especialización
- Profundidad del análisis que realiza
- Información adicional que aporta
- Necesidades de formación que exige a sus usuarios
- Coste económico
- En la fase de estudio habrá de solicitarse a los proveedores las oportunas demostraciones
- Técnicas que permitan evaluar la herramienta adecuadamente.
- Por último, siempre existe la posibilidad de fabricar una propia que contemple las particularidades del entorno sobre el que va a utilizarse.
Visual Basic, Microsoft Access, Power Builder, Delphi

A continuación se tiene una tabla comparativa en cuanto a las herramientas de desarrollo que existen en el mercado:

Nombre	Descripción	Características
Visual Basic	Es una herramienta de desarrollo que corre sobre plataformas Windows y es una de las más utilizadas en el mercado.	Permite desarrollar sistemas amigables y fáciles de utilizar Soporta la creación de aplicaciones cliente/servidor mediante el uso de ODBC's (Open DataBase Connectivity). Utiliza la programación orientada a eventos, es decir, cuando suceden ciertos eventos (mover el mouse o dar click en algún control) se realizan las acciones que se encuentran programados en éstos Permite crear aplicaciones ejecutables que pueden ser distribuidos a todos los usuarios Permite crear aplicaciones para el Web.
Power Builder	Es la herramienta de desarrollo nativa de Sybase utilizada para	Soporta el desarrollo de aplicaciones cliente/servidor.

	<p>la creación de aplicaciones cliente/servidor. Permite desarrollar aplicaciones en diversas plataformas</p>	<p>Permite crear aplicaciones ejecutables y generar los discos de instalación. Utiliza la programación orientada a eventos y hace uso de las características de la programación orientada a objetos: encapsulación, polimorfismo y herencia.</p>
<p>Delphi</p>	<p>Es una herramienta de desarrollo creado por la empresa Borland para las plataformas Windows (Windows 95, Windows 98 y Windows NT).</p>	<p>Permite el desarrollo de aplicaciones en Internet/Web mediante asistentes (wizards) que crean aplicaciones en forma rápida. Además dispone de una biblioteca de 25 componentes nativos para Internet para crear aplicaciones como lectores de noticias y servidores de correo. Soporte completo y alto desempeño a manejadores de bases de datos Oracle 8. Soporta múltiples manejadores de base de datos de alto desempeño en 32 bits, como SQL Link, Sybase, Informix, DB2, SQL e InterBase. Soporta los estándares creados por Microsoft como ActiveX. Optimización para el desarrollo cliente/servidor mediante Builder SQL, que construye directamente consultas complejas, además de un monitor y un explorador SQL. Incluye componentes para la creación de aplicaciones de información gerencial. Analiza y visualiza los datos corporativos, para la toma de mejores decisiones. El lenguaje de programación que utiliza se llama Object Pascal. En su nueva versión añade Docking Support, que permite construir fácilmente barras de herramientas con el comportamiento Windows Docking, que puede verse en Windows 98, o el ControlBar, que facilita incluir Office 97 en las aplicaciones generadas.</p>

Fig. 2. 3 Tabla comparativa de Herramientas de Desarrollo

Debido a las características del sistema de información se decidió utilizar las siguientes herramientas, para el análisis y diseño de la base de datos, se utilizaron herramientas CASE, para el análisis del sistema utilizaremos la metodología de Chris Gane y Trish Sarson. Para el lenguaje de desarrollo utilizaremos Power Builder, el DBMS será SQL Server y la plataforma de desarrollo será Windows NT. La arquitectura propuesta es Cliente/Servidor de tres capas, a continuación, se justifica él porque de estas herramientas.

PowerBuilder 7.0

Power Builder es una herramienta de desarrollo rápido (RAD) muy poderosa, la cual permite la creación y manipulación de bases de datos para sistemas mono o multiusuario. Presenta una interfaz de usuario y desarrollo muy amigable y favorece el desarrollo de aplicaciones en forma rápida y eficiente pues contiene herramientas destinadas para este fin. Power Builder es un desarrollador visual, que permite manejar bases de datos y crear aplicaciones.

Selección del DBMS

La elección del DBMS para nuestro desarrollo, me llevó indagar acerca de los diferentes productos disponibles en el mercado. El DBMS debía ser el más adecuado para las necesidades del sistema, por lo que un Oracle, o un Sybase Adaptive Server eran demasiado grandes para propósitos de la aplicación que se está desarrollando: consumen grandes recursos de hardware, los costos de las licencias superan los 8000 dls, tienen problemas de configuración bajo Windows NT, y se disponía de muy poca documentación al respecto. Por otra parte, programas tan pequeños, poco robustos y sin seguridad como Microsoft Access quedaban cortos a nuestras necesidades. Así después de un profundo análisis, se tuvieron dos alternativas: Microsoft SQL Server 7.0, y Adaptive Server Anywhere de Sybase. Se realizaron pruebas de rendimiento con Power Builder en ambos casos, y el desempeño fue similar, sin embargo, en cuanto a documentación, y adaptabilidad al sistema operativo Windows NT, el producto SQL Server de Microsoft fue superior al programa de Sybase, por lo que finalmente fue el producto acorde a nuestras necesidades.

Windows NT

La plataforma sobre la cual va a trabajar es Windows NT, ya que actualmente es la que soporta la red del Centro.

Capítulo 3. Desarrollo de la Solución Propuesta

3.1 Introducción

Los Sistemas de Información están constituidos por factores tales como: costos, seguridad y desempeño (performance).

Un Sistema de Información pobremente diseñado, implementado, y mantenido es el mayor problema para muchas compañías que hacen uso de ellos. Estos aspectos han hecho centrar la atención en los procesos por medio de los cuales los sistemas de Información se desarrollan, se mantienen y soportan la tecnología basada en computadoras que los soporta. Esto incluye todos los aspectos: gerenciales, administrativos o técnicos de cualquier parte del proyecto de desarrollo de los Sistemas de Información.

Los Sistemas de Información de calidad se logran con procesos de desarrollo claramente definidos, utilización de una metodología adecuada al tipo de Sistema de Información que se esté desarrollando y soportar la aplicación de ésta, con tecnología de punta.

En el desarrollo de un sistema, se requiere de una serie de etapas sistemáticas llamada metodología o lo que también se conoce como ciclo de vida de un sistema de información (CVSI).

3.2 Análisis

La información que genera y recibe el centro de computo se realiza básicamente a través de documentos llenados manualmente, estos formatos varían de un departamento a otro. Haciendo uso de la información obtenida en las entrevistas a cada uno de los jefes departamento y personal que en ellos labora se llego a definir el papel que juegan dentro del proceso de las inscripciones a los cursos. En este apartado se presentan los formatos más importantes que se utilizan dando una breve explicación sobre su uso y contenido.

3.2.1 Análisis de Formatos

Forma de Inscripción

Este formato es utilizado por el departamento de Informes y Relaciones para realizar las inscripciones a un curso, en este se obtiene la siguiente información.

- Los datos del alumno
- Nivel escolar del alumno
- Medio de difusión por el cual se entero del curso
- Procedencia
 - La procedencia define el tipo de pago que alumno deberá hacer en caja:
 - Particular paga el 100%, en caso de contar con un curso anterior y haber obtenido constancia al siguiente curso se le hace un descuento de 10%.
 - Estudiante de la UNAM se le hace un descuento del 50%, en caso de contar con un curso anterior y haber obtenido constancia al siguiente curso se le hace un descuento del 50% y el 10% adicional.
 - Estudiante de otra institución 25%, en caso de contar con un curso anterior y haber obtenido constancia al siguiente curso se le hace un descuento del 25% y el 10% adicional.

Para hacer efectivo el descuento del 10% adicional, es necesario mostrar la constancia obtenida, el recibo de pago al curso anterior, ya que no se cuenta con el control automático de esta información, de lo contrario no se otorga el descuento.

- Documentos que se anexan a la solicitud de inscripción.
- Datos del curso al que ingresa.
- Se especifica el programa de alto rendimiento al que ingresa, en caso de existir.

Solicitud para el examen de evaluación de conocimientos previos en cómputo:

El alumno deberá reunir los requisitos establecidos para poder tomar un curso, en caso de no contar con ellos, realizará un examen de conocimientos previos. En este formato se solicita la siguiente información.

- Los datos del alumno
- Nivel escolar del alumno
- Medio de difusión por el cual se entero del curso
- Curso al que desea ingresar
- Examen a realizar

Documentos para Candidatos a Diplomados y Líneas de Especialización

Entre los cursos que ofrece el centro existen los diplomados y las líneas de especialización, los cuales requieren de documentos adicionales dependiendo de las características de estos.



**Dirección General de Servicios de Cómputo Académico
Dirección de Cómputo para la Docencia**

FORMA DE INSCRIPCIÓN

Clave: _____
Cambio: _____

Para uso exclusivo del Departamento de Información y Relaciones

Documentos que presenta: _____ Calle: _____
 Observaciones: _____ Vo.Bo. _____
 Procedencia: UNAM OTRA INST. EDUCATIVA PARTICULAR
 Num. Recibo: _____ Monto: _____ Fecha: _____

PROGRAMA AL QUE SE INSCRIBE:

- Actualización en cómputo Cómputo para niños Supercómputo Videoconferencia
 Capacitación para el trabajo Diplomado Otro ¿Cuál? _____
PROGRAMA DE ALTO RENDIMIENTO ACADÉMICO Licenciatura Bachillerato

Escriba claramente con letra de molde. Marque con una X en el lugar que corresponde.

Nombre		Apellido		RFC	
Institución		Código		C.P.	
Tel. Principal		Tel. Celular		Correo electrónico	

SE INSCRIBE A LOS CURSOS COMO:

- Estudiante UNAM Programa Alto Rendimiento Académico Escuela o Facultad
 (o en alguno de los Centros de Investigación Académica)

Estado de cuenta: _____ Tipo de línea: _____ Fecha de ingreso al programa: _____

- Empleada UNAM Investigadora UNAM Académico UNAM

Dependencia:

- Estudiante de otra inst. Educ. Profesor de otra inst. Educ. Incorporado a la UNAM SI No

Nombre de la inst. _____

- Particular

CURSO AL QUE INGRESA:

Nombre del curso _____

Período _____

Horario _____

Año _____

Curso _____

Instituto _____

- ¿Ha tomado otros cursos en esta sede? SI No ¿Cuál(es)? _____

continúa al reverso

Fig. 3. 1 Forma de Inscripción



Dirección General de Servicios de Cómputo Académico
 Dirección de Cómputo para la Docencia

SOLICITUD PARA EXAMEN DE EVALUACIÓN DE CONOCIMIENTOS PREVIOS EN CÓMPUTO

Para uso exclusivo del Departamento de Información y Relaciones

Núm. de recibos: _____ Clave: _____ Fecha: _____

Escribe claramente con letra de molde. Marqua con una X en el lugar que corresponde.

Nombre: _____ P. _____ A. _____ M. _____ RFC: _____

Dirección: _____ C. y N. _____ C. P. _____ Tel. y Fax _____

Tel: _____ Correo electrónico: _____

NIVEL ACTUAL DE ESCOLARIDAD:

Primaria Secundaria Escuela Grado

Bachillerato Escuela Semestre

Licenciatura Carrera Semestre

Maestría Titulado Otro ¿Cuál? _____

Trabaja Si No Emplea _____

Parece _____

CURSO AL QUE DESEA INGRESAR

EXAMEN A REALIZAR

¿POR QUE MEDIO SE ENTERO DE LOS CURSOS?

Gaceta UNAM Triptico Internet Audiopospecta (vía telefónica) Radio

Periódico ¿Cuál? _____ Recomendación

Centro Nuevo León Centro Mascarones DGSCA CU

Otro _____

NOTA: La acreditación del examen solamente es válida para la inscripción a cursos en las dependencias dentro de la DGSCA.

Firma _____

Fig. 3. 2 Solicitud para Examen de Evaluación de Conocimientos Previos en Cómputo

TESIS CON FALLA DE ORIGEN



Dirección General de Servicios de Cómputo Académico
Dirección de Cómputo para la Docencia

Documentos para Candidatos a Diplomados y Líneas de Especialización

Diplomado ó Línea de especialización: _____

Nombre: _____

Profesión: _____

Teléfono: _____



- Fotos
- Resumen de Curriculum Vitae Actualizado (3 páginas)
- Copia de certificado de estudios completos, Título ó Cédula Profesional
- Carta de Exposición de Motivos (una cuartilla)
- Carta de Presentación de la empresa donde trabaja (una cuartilla)

Observaciones: _____

TIPO DE PAGO: (P) (O) (U) _____

Fecha _____ Nombre de quien Recibe _____ Firma del Alumno _____

Dictamen del Comité Evaluador: _____

Vo Bo Comité Evaluador _____

Fecha _____

Fig. 3.3 Formato de recepción documentos para candidatos a Diplomados y Líneas de Especialización

Las entrevistas que realice me permitieron conocer y analizar los reportes que utilizan, con esto fue posible definir entidades y los atributos de las mismas.

Los departamentos tienen actividades bien definidas y repetitivas, mas adelante se dan a conocer cuales son las actividades y como quedaron sin que los resultados finales dejen de ser los mismos.

Se analizo la importancia de cada documento y que información era importante en cada una de las áreas.

Las principales actividades por Departamento son:

DEPARTAMENTO	FUNCIONES
Informes y Relaciones	<ul style="list-style-type: none"> • Información a los usuarios acerca de los cursos impartidos en el Centro. • Recepción de la solicitud para los cursos, diplomados, y líneas de especialización. • Recepción del recibo por concepto de cursos. • Control de ingresos que por concepto de cursos se perciben en el centro • La generación de listas oficiales, lista para personal de vigilancia • Reportes
Control Escolar	<ul style="list-style-type: none"> • En base a la demanda de cursos, se forman grupos. • Administran los horarios para dichos grupos • Administración de calendarios para cursos • Cancelan los grupos en caso de que sea necesario • Selección de profesores • Manejo de estadísticas relacionadas con el desempeño de los alumnos • Evaluaciones de profesores • Reportes
Infraestructura y Soporte Tecnológico	<ul style="list-style-type: none"> • Administra la programación de las aulas, dependiendo del tipo de cursos • Proveer a los profesores de equipo complementario para impartir los cursos (TV, VCR, DataShow, Proyector de acetatos) • Dar mantenimiento preventivo y correctivo al equipo de computo empleado en los eventos de capacitación • Reportes

Delegación Administrativa	<ul style="list-style-type: none">• Administra los recursos que capta el centro por concepto de cursos• Capta los gastos de todo el centro y el material que se requiere para su funcionamiento• Es el encargado de la caja, donde se pueden adquirir discos, manuales y otros artículos.• Generación de recibos• Reportes de ingreso diario
----------------------------------	--

Después de analizar la información de los formatos que utilizan para la captura de información de alumnos, cursos, profesores, y demás catálogos de importancia, se detectaron las siguientes entidades que formarán parte de la base de datos:

ENTIDADES	DESCRIPCION
ALUMNO	Es el alumno que toma ó tomo clases en esta institución
ALUMNO_GRUPO	Guarda la relación de alumno y grupo
ÁREA	Sirve para clasificar el área a la que pertenece cada curso.
ARTICULO	Catálogo de materiales, bonos y varios.
AULA	Es el lugar donde se imparte un curso
BANCO	Catálogo de institución bancaria
BECA	Son las becas que puede tener un alumno al tomar un curso.
CALIF_PREVIO	Calificación del examen previo
CAT_CURSO	Catálogo de cursos que se imparten en esta dependencia.
CAT_MODULO	Guarda la relación entre un curso y un modulo
CONCEPTO_RECIBO	Es el concepto de un recibo
CLASIFICA_CURSO	Catálogo de estados de un curso de acuerdo al nivel de especialización
CURSO_ESP	Catálogo de cursos especiales
GRUPO	Son los grupos que se formaron en esta institución
CURSO_SOFTWARE	Guarda la relación del software requiendo por un curso
DEPARTAMENTO	Es el departamento que forma parte de la institución
DEPENDENCIA	Catálogo de dependencias de la UNAM
DOCUMENTO	Catálogo de documentos que se pueden anexar a una solicitud.
ESTADO_CURSO	Catálogo de estados de un curso
EDO PROFESION	Catálogo de estados de profesión
ESTADO_RECIBO	Catálogo de estados de recibo
ESC_FAC	Catálogo de Escuelas y Facultades en la UNAM
EGRESO	Son los egresos del centro a una fecha dada
EXAMEN	Catálogo de exámenes previos
INSTITUCION	Son las instituciones a las que se le imparte cursos
INST_CURSO	Guarda la relación de un curso con una institución
MEDIO_DIFUSION	Catálogo de los medios de difusión existentes
MODULO	Es una serie de cursos
MODULO_ESP	Catálogo de cursos pertenecientes a un modulo
NIVEL_ESCOLAR	Son los tipos de nivel de escolaridad
P_ALTO_REMD	Catálogo de los programas de alto rendimiento.
PERSONAL	Es el personal que labora en el centro
PROFESOR	Es la persona que imparte cursos en el centro.
PROG_CURSO	Relación entre programas y cursos.

PROGRAMA	Los cursos están agrupados por programas.
PROPUESTA	Es la forma para tomar un diplomado ó línea de especialización.
RECIBO	Es el documento que comprueba un pago.
SEDE	Lugar donde se lleva a cabo un curso
SOFTWARE	Guarda el software que se utiliza en cada curso
SOLICITUD	Es la forma para solicitar un curso, diplomado ó una línea de especialización
SOLICITUD_DOC	Es la relación entre solicitud y documentos
TIPO_CAT_CURSO	Es el tipo al que pertenece un curso.
TIPO_EGRESO	Es el catálogo de los tipos de gastos que tiene el centro
TIPO_PAGO	Es un catálogo que contiene los tipos de pago
TIPO_PERSONAL	Catálogo de los tipos de personal.
TIPO_SOLICITUD	Catálogo de tipos de solicitud.

Fig. 3. 4 Tabla Definición de Entidades

3.2.1 Resultado de las entrevistas

Para establecer las relaciones entre cada una de las entidades se tomaron en cuenta entre otras cosas los siguientes puntos que fueron detectados en las entrevistas hechas al personal del centro. Cabe mencionar que alguna vez fui parte de los alumnos del centro, esto me permitió conocer desde fuera el procedimiento de inscripción, y observe algunos puntos que me ayudaron a formular preguntas en el momento de las entrevistas.

- Para que una alumno pueda inscribirse a un curso debe cumplir con los requisitos establecidos.
- El número de becas para un grupo es restringido
- Existen medias becas, becas completas, becas a personal del centro, becas sindicales y cada una se maneja de forma diferente.
- Un grupo tiene un mínimo de ingreso para poder abrirse
- La asignación de aulas esta determinada por los requerimientos del curso
- Un curso no esta restringido a ser dado en las instalaciones del centro, este puede darse en las instalaciones de quien lo solicite
- El pago para una línea de especialización y diplomado, se puede hacer en pagos parciales
- Un curso puede ser a su vez requisito de otro curso
- Dentro de la información de un profesor se debe conocer que cursos a dado y que cursos puede dar.
- El personal del centro esta clasificado
- La clasificación en cuanto a cursos se refiere, esta dada por DGSCA, de CU
- El horario y el periodo son dados desde la programación de calendarios, el profesor y el aula son determinados después.
- Un curso es programado dependiendo de la demanda y esto lo determinan las encuestas y estadísticas que se generan después de cada periodo.
- El diplomado y la línea de especialización esta formados por una serie de cursos, agrupados en módulos.
- Se puede hacer referencia a un mismo módulo en mas de un diplomado o línea de especialización.
- El alumno que desee ser candidato a un diplomado debera tener una propuesta escrita, una vez aceptados, entrega la documentación solicitada.
- El alumno que no cumpla con los requisitos puede solicitar que le apliquen un examen de conocimientos previos

- Cada examen de conocimientos previos tiene un costo, y el número de exámenes que se tiene que aplicar al aspirante depende del tipo de curso al que vaya ingresar.
- Un recibo es generado inmediatamente después, de llenar el formato de inscripción
- En un formato de inscripción sólo se puede solicitar un curso
- Se otorgan gafetes a los alumnos inscritos como identificación durante su estancia en el Centro de Computo.
- Se entrega un archivo con la relación de los alumnos que obtienen constancia al final de cada periodo a departamento de Control Escolar de DGSCA CU.
- Cada departamento tiene que rendir reportes con características especiales a la coordinación
- Existe un reporte de ingresos diario
- Un curso pertenece a un programa, este determina el grado de especialización del curso.
- Los cursos están agrupados por áreas, ejemplo: Sistemas Operativos, los cursos que pertenece a esta área son UNIX, MS-DOS, Windows NT, etc.
- El medio de difusión en una solicitud es requerido, para estadísticas posteriores.
- Los cursos pueden ser de lunes a viernes o sábados.
- Existen Cursos Institucionales que se crean a petición de una institución particular, la cual solicita cursos en un periodo dado (un año, seis meses, etc).
- Un alumno puede ser parte del personal, el control de estos es importante en el momento de rendir reportes de contraloría interna, para efecto de becas otorgadas.

Las relaciones que existen entre cada una de estas entidades se describen a detalle en el Apéndice A, Sección I.

Normalización

En el proceso de la normalización se minimiza la redundancia de los datos. La redundancia de datos causa problemas de integridad. Las transacciones de actualización y borrado pueden no ser consistentes en todas las copias de los datos causando inconsistencia en la base de datos. La normalización ayuda a identificar tablas y llaves foráneas.

Como ejemplo en nuestro proceso de normalización se tomó la tabla de CAT_CURSO, donde existen grupos de repetición para un renglón.

CAT_CURSO

Cve_catcurso	Nom_oficial	Nom_comercial	No_horas	Objetivo	Costo_unam	Costo_institucion
AM02	Windows 98	Windows 98	20		474	758.4

Costo_particular	Ing_minimo	Cve_area	Id_tipo_curso	Id_clas_curso	modulo
948	17500	3	3	2	Módulo básico en cómputo Módulo básico en cómputo con Internet Módulo II de actualización en cómputo para profesores del bachillerato

Fig. 3. 5 Ejemplo de un registro de la tabla CAT_CURSO

Como podemos observar existe más de un módulo al que pertenece este curso, para una sola columna. De esta forma no es posible hacer la siguiente clasificación:

Un diplomado y una línea de especialización, son programas diseñados para apoyar a los profesionales de las diversas disciplinas en diferentes áreas del conocimiento. Estos cursos especiales están formados por una combinación de módulos que a su vez están constituidos por una serie de cursos, los cursos pueden estar repetidos para diferentes módulos.

Conversión a la primera forma normal

Primera Forma Normal (1FN) - La tabla no puede contener grupos de repetición.

Eliminar el grupo de repetición.

Crear una nueva tabla con la PK (llave primaria) de la tabla base y el grupo de repetición.

CAT_CURSO

Cve_catcurso	Nom_oficial	Nom_comercial	No_horas	Objetivo	Costo_unam	Costo_institucion
AM02	Windows 98	Windows 98	20		474	758.4

Costo_particular	Ing_minimo	Cve_area	Id_tipo_curso	Id_clas_curso
948	17500	3	3	2

Fig. 3. 6 Registro resultante al aplicar la 1FN

CURSO_MODULO

Cve_modulo	Nom_modulo	Cve_catcurso
MC01	Módulo básico en computo	AM02 PT01
MC25	Módulo II de actualización en computo para profesores del bachillerato	AM02 HC03

Fig. 3. 7 Definición de la tabla CURSO_MODULO

Conversión a la Segunda Forma Normal

Segunda Forma Normal(2FN). - La tabla debe de estar en 1FN. Cada columna que no es llave debe de ser dependiente de la llave primaria como un todo.

Por lo que se observa se tiene que crear otra tabla para eliminar los grupos de repetición en la columna de la tabla CURSO_MODULO.

CAT_CURSO

Cve_catcurso	Nom_oficial	Nom_comercial	No_horas	Objetivo	Costo_unam	Costo_institucion
AM02	Windows 98	Windows 98	20		474	758.4

Costo_particular	Ing_minimo	Cve_area	Id_tipo_curso	Id_clas_curso
948	17500	3	3	2

Fig. 3. 8 Definición de la tabla CAT_CURSO

CURSO_MODULO

Cve_modulo	Cve_catcurso
MC01	AM02
MC01	PP01
MC25	HC01
MC25	AM02

Fig. 3. 9 Definición de la tabla CURSO_MODULO

MODULO

Cve modulo	Nom modulo
MC01	Módulo básico en computo
MC25	Módulo II de actualización en computo para profesores del bachillerato

Fig. 3. 10 Definición de la tabla Modulo

El proceso de normalización en el modelo entidad relación, se realizo como lo hicimos para la tabla de CAT_CURSO, sin caer en la sobrenormalización. La Desnormalización puede ser definido como cualquier tipo de redundancia, intencionalmente es introducida para incrementar el rendimiento de la base de datos, en algunos casos.

Es importante señalar que en el diseño relacional de la tabla CAT_CURSO nos encontramos con una relación recursiva de UNO A MUCHOS, ya que un curso puede ser requisito para otros cursos.

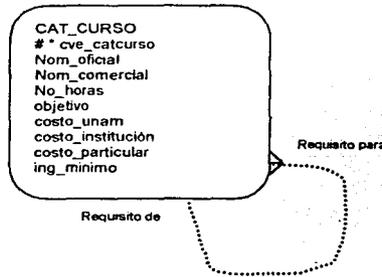


Fig. 3. 11 Relación recursiva en CAT_CURSO

En el diseño físico para una relación recursiva 1:M, se agrega una columna FK (llave foránea) a la tabla. Esta columna FK (llave foránea) debe referenciar valores de la columna PK (llave primaria).

CAT_CURSO

Cve_catcurso	Nom_oficial	Nom_comercial	No_horas	Objetivo	Costo_unam	Costo_institucion
AM03	Windows 98	Windows 98	20		474	758.4
DC01						

Costo_particular	Ing_minimo	Cve_area	Id_tipo_curso	Id_clas_curso	Id_req_curso
948	17500	3	3	2	LP15
		2	1	4	LP15

Fig. 3. 12 Ejemplo de un registro

Los atributos para cada una de las tablas, así como, las llaves primarias y foráneas se muestran a continuación:

ALUMNO	ALUMNO GRUPO	AREA
Id Alumno (PK)	Id Alumno (PK,FK)	Id Area (PK)
Rfc	Id Grupo (PK,FK)	Nom_area
Nom_alumno	Calificación	
A_paterno	Asistencia	
A_materno	Id_beca (FK)	
Calle	Constancia	
Colonia		
Delegación		
Estado		
CP		
Tel_oficina		
Tel_casa		
Tel_recado		
E_mail		
No_cuenta		
Trabajo		
Ocupacion		
Puesto		
Id_escolar (FK)		
Id_tipo_personal (FK)		
Id_prog_alto_ren (FK)		
Id_edo_profesion (FK)		
Id_esc_fac (FK)		
Id_proc(FK)		
Id_dep (FK)		
ARTICULO	AULA	BANCO
Id_art (PK)	Id Aula (PK)	Id banco (PK)
Nom_art	Num_Maq	Nom_banco
Costo		observaciones
BECA	CALIF PREVIO	CAT CURSO
Id_Beca (PK)	id_alumno(PK,FK)	dgsc_a_clave (PK)
Nom_beca	calificacion	Nom_oficial
Por_desc	fecha	Nom_comercial
		No_horas
		Costo_unam
		Costo_institucion
		Costo_particular
		Objetivo
		Req_curso
		Ingreso_minimo
		Id_area (FK)
		Edo_temario

CAT_MODULO	CONCEPTO_RECIBO	Id_tipo curso (FK)
Dgsca_clave (PK,FK)	id_articulo (PK,FK)	Id_Curso (PK)
Id_modulo (PK,FK)	Id_recibo (PK,FK)	Hora_inicio
	Observacion	Hora_fin
		Fecha_inicio
		Fecha_fin
		Dias
		módulos
		Dgsca_clave (FK)
		Id_cdo curso (FK)
		Id_aula (FK)
		Id_sede (FK)
		Id_tipo curso (FK)
		Id_profesor (FK)
CURSO_ESP	CURSO_SOFT	DEPARTAMENTO
Id_curso_esp (PK)	Id_Soft (PK, FK)	Id_depto (PK)
Nom_curso_esp	Dgsca_clave (PK, FK)	Nom_depto
Monto		función
DEPENDENCIA	DOCUMENTO	EGRESO
Id_Dependencia (PK)	Id_doc (PK)	Id_egreso (PK)
Nom_dependencia	Nom_doc	Costo
Siglas		Fecha
Calle		Id_tipo_egreso(FK)
Colonia		
Delegacion		
Cp		
Tel1_dep		
Tel2_dep		
E_mail		
Contacto		
ESTADO_GRUPO	EDO_PROFESION	ESTADO_RECIBO
Id_estado (PK)	Id_cdo_profesion (PK)	Id_recibo (PK)
Nom_edo_grupo	Nom_profesion	Nom_edo_recibo
Observaciones		observaciones
ESC_FAC	EXAMEN	INSTITUCION
Id_esc_fac (PK)	Id_Examen (PK)	Id_institucion(PK)
Nom_esc_fac	Nom_examen	Nom_institucion
	costo	Siglas
		Responsable
		Calle
		Colonia
		Delegación
		estado
		C.P.
		Tel_1institucion
		Tel_2institucion
		E_mail
INST_CURSO	MEDIO_DIFUSION	MODULO
Id_Inst (PK)	Id_Medio difusion (PK)	Id_Modulo (PK)
Id_Curso (PK)	Nom_medio difusion	Nom_modulo
MODULO_ESP	NIVEL	NIVEL_ESCOLAR
Id_modulo (PK,FK)	Id_nivel (PK)	Id_nivel escolar(PK)
Id_curso_esp (PK,FK)	Nom_nivel	Nom_nivel_escolar
P_ALT_REMD	PERSONAL	PROFESOR
Id_p_alt_rend (PK)	Id_personal (PK)	Id_Profesor (PK)
P_alt_rend	Nom_personal	Nom_profesor
Id_nivel (FK)	A_paterno	A_paterno

	A materno	A materno
	Calle	Edad
	Colonia	Sexo
	Delegacion	Nacionalidad
	Cp	Calle
	Tel casa	Colonia
	Tel recado	Delegación
	Horas semana	estado
	Ocupacion	CP
	Contraseña	Tel oficina
	Lectura	Tel casa
	Actualización	E_mail
	Borrado	Rfc
	Id tipo personal (FK)	Profesion
	E_mail	
	Id_depto (FK)	
PROG_CAT_CURSO	PROGRAMA	PROG_CURSO
Id_prog_curso (PK,FK)	Id_Programa (PK)	Dgsca_clave (PK,FK)
Dgsca_clave (PK, FK)	Nom_programa	Id_programa (PK,FK)
PROPUESTA	RECIBO	SEDE
Id_propuesta (PK)	Id_recibo (PK)	Id_Sede (PK)
Id_alumno (PK,FK)	Id_edo_recibo (PK,FK)	Nom_sede
Descripcion	Id_tipo_pago (PK,FK)	Calle
Fecha	Id_banco (PK,FK)	Colonia
	Fecha	Delegación
	Observaciones	estado
	Hora	CP
	monto	Tel 1
		Tel 2
		contacto
SOFTWARE	SOLICITUD	SOLICITUD_DOC
Id_Software (PK)	Id_solicitud (PK)	Id_solicitud (PK,FK)
Nom_software	Id_tipo_solicitud (PK,FK)	Id_documento (PK,FK)
Licencia	Id_recibo (PK,FK)	
	Dysca clave (PK,FK)	
	Id_medio_difusion (PK,FK)	
	Id_grupo	
	Id_alumno (PK,FK)	
	fecha	
TIPO GRUPO	TIPO EGRESO	TIPO PAGO
Id_Tipo_Curso (PK)	Id_tipo_egreso (PK)	Id_tipo_pago (PK)
Nom_tipo_curso	Nom_tipo_egreso	Nom_tipo_pago
TIPO_PERSONAL	TIPO_SOLICITUD	
Id_tipo_personal (PK)	Id_tipo_solicitud (PK)	
Nom_tipo_personal	Nom_tipo_solicitud	
descripcion		

PK - Llave primaria
 FK - Llave foránea

3.3 Diseño Relacional De la Base de Datos

A partir del diagrama *Entidad – Relación* convertiremos las entidades en tablas y los atributos en columnas.

De esta forma, las tablas por las que estará compuesto se describen en el Apéndice B Diccionario de la Base de Datos.

3.4 Diseño Físico

En el diseño físico de la base de datos, se detalla las estructuras de almacenamiento, índices primarios y secundarios, se asignan las tablas a los dispositivos de almacenamiento.

Un índice es una lista de valores de una columna, con apuntadores a los renglones que contienen estos valores. Un índice no es lo mismo que una llave. Un índice es una construcción física, que mantiene el rendimiento en disco, mientras que una llave es una construcción relacional usada para identificar renglones. Los índices se generan para mejorar el rendimiento de las búsquedas en disco(Ver Apéndice C).

3.5 Diseño del Sistema

3.5.1 Diagrama Jerárquico Funcional

El diagrama jerárquico funcional tiene dos objetivos, la primera es el propio entendimiento del sistema y la segunda es definir la estructura de las actividades del sistema tan simple como sea posible, de manera que los usuarios puedan revisarlas fácilmente.

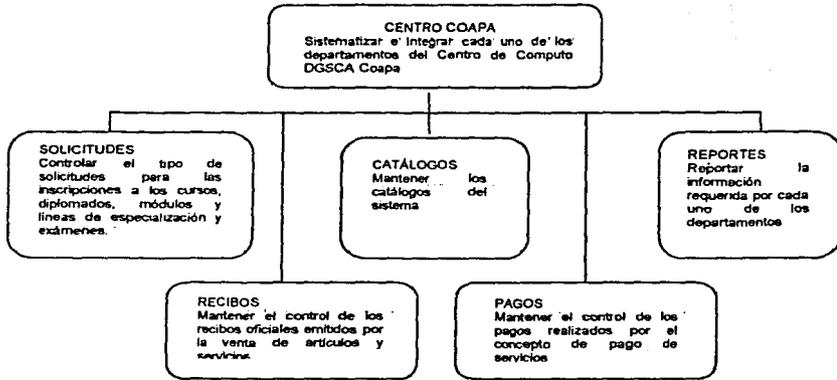
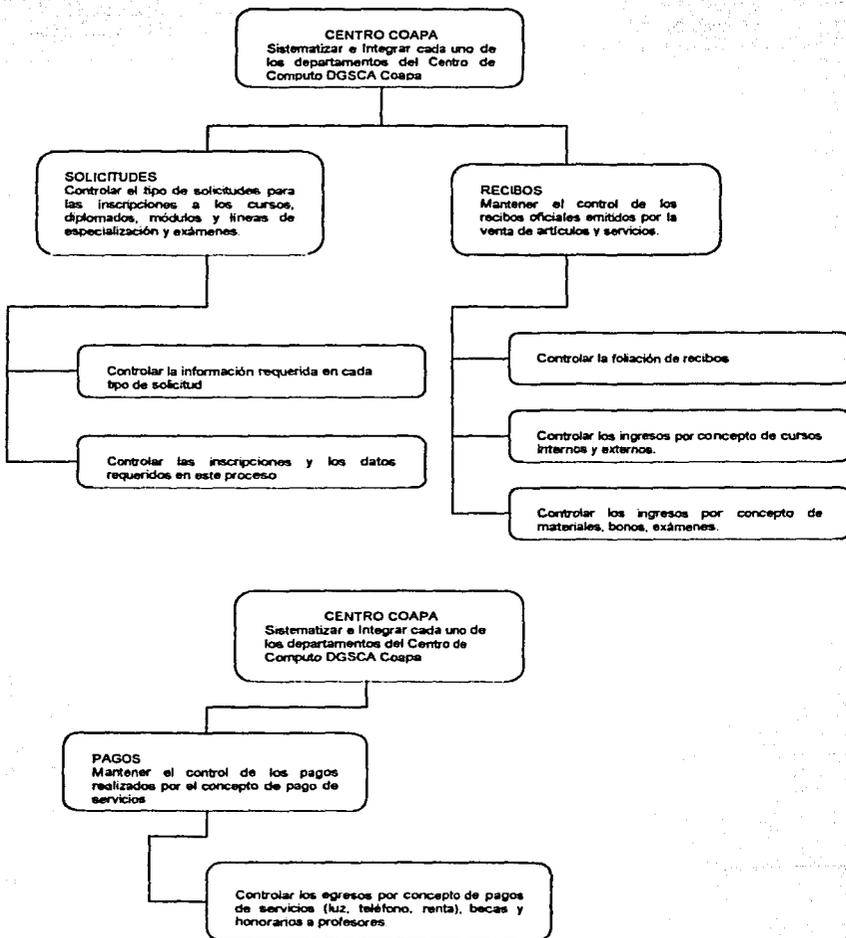


Fig. 3. 14 Diagrama jerárquico funcional de Nivel 1

A continuación se muestran las subfunciones de las funciones principales del sistema:



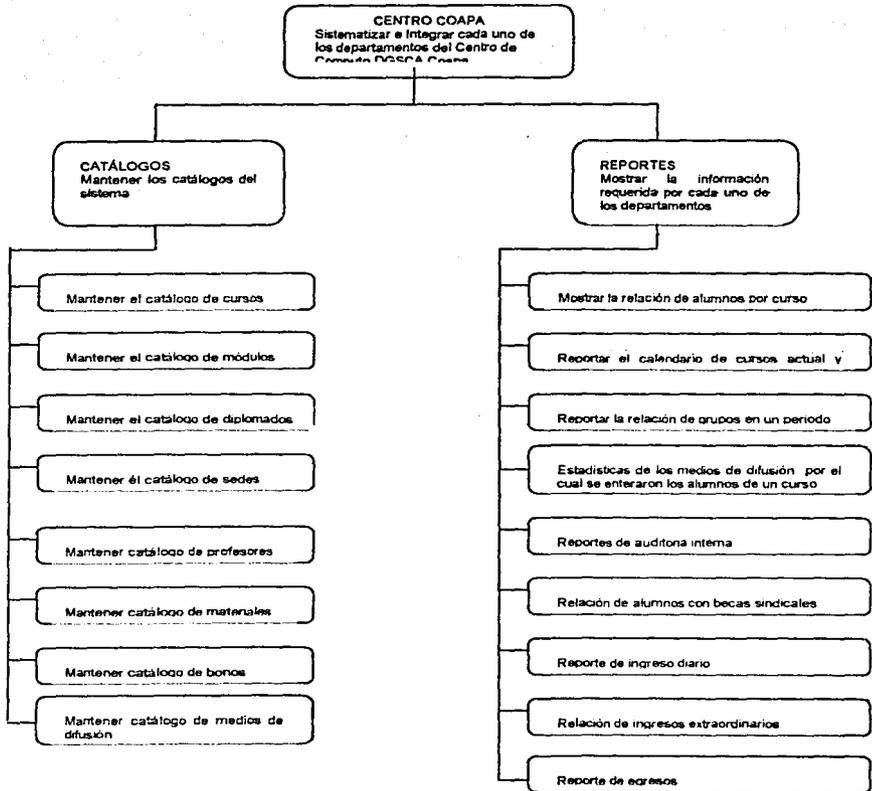


Fig. 3. 15 Diagrama jerárquico funcional de Nivel 2

Diagrama de Flujo de Datos

El objetivo de estos diagramas es mostrar como fluye la información alrededor de las áreas del sistema, suministrando a las funciones los datos requeridos.

3.5.2 Diagrama de Contexto

El diagrama de contexto se refiere al diagrama de flujo de datos de nivel 0. Este muestra de manera general el flujo de información del sistema.

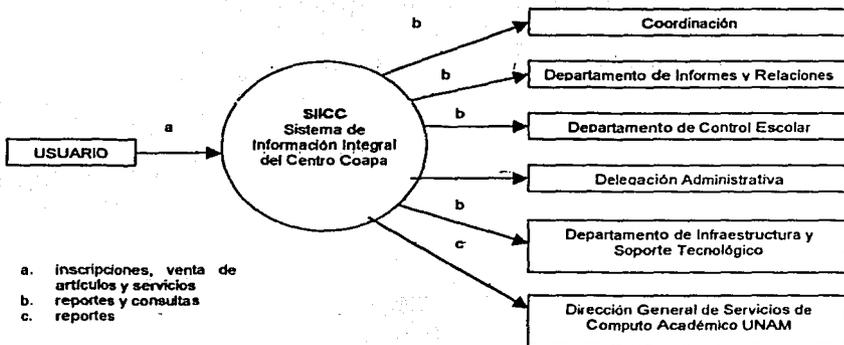


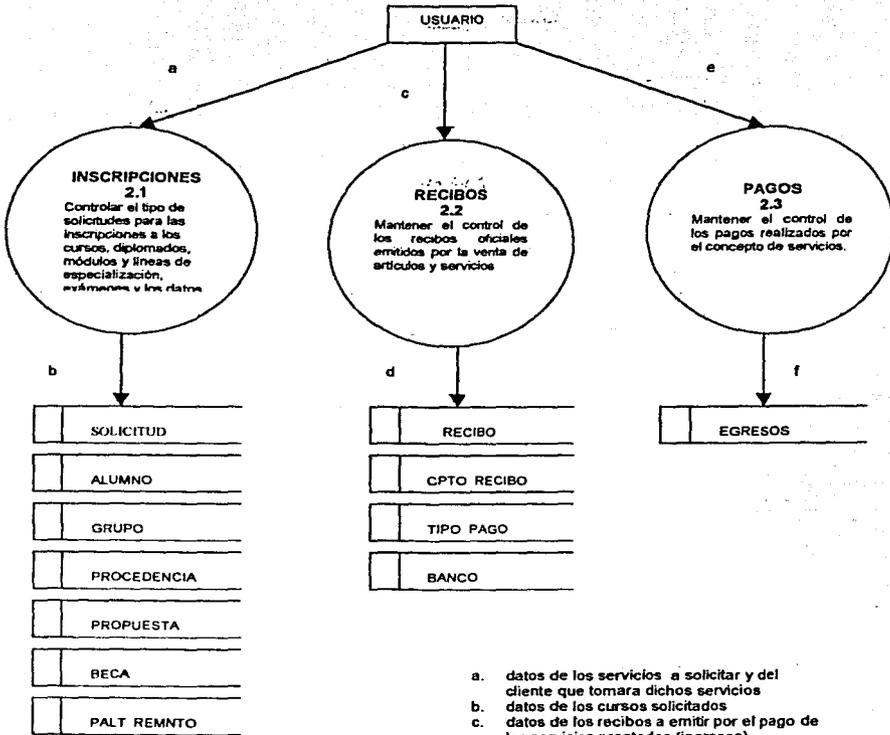
Fig. 3. 16 Diagrama de contexto

A continuación se define a detalle las funciones del Menú Principal:

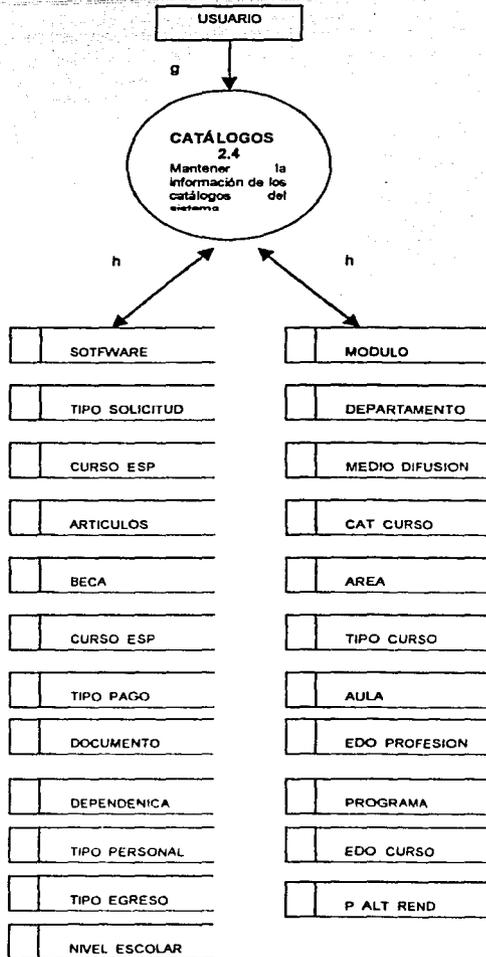
Funciones del Menú Principal

Al entrar al sistema éste mostrará el menú principal, el cual contiene las opciones para poder acceder a cada uno de los módulos del sistema.

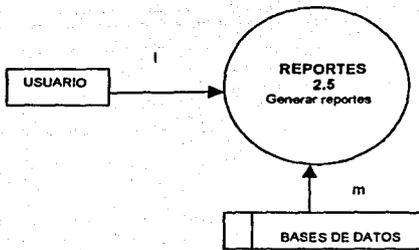
A continuación se muestra el diagrama de flujo de datos del sistema.



- a. datos de los servicios a solicitar y del cliente que tomara dichos servicios
- b. datos de los cursos solicitados
- c. datos de los recibos a emitir por el pago de los servicios prestados (ingresos)
- d. registrar los datos del recibo
- e. datos por el concepto de pago de servicios (egresos)
- f. registrar los datos



- g. Consultas y modificaciones
- h. nueva información
- i. operaciones de respaldo
- j. información a respaldar



- I. opciones para elegir un reporte
- m. información para generar el reporte

Fig. 3. 17 Funciones principales del Sistema

3.5.3 Operaciones sobre Catálogos

A continuación vemos una tabla que muestra las tablas que se afectan en cada operación a un catálogo determinado

Catálogos	Operaciones	Tablas afectadas
ARTICULOS	Agregar Consultar Modificar Borrar	ARTICULOS ARTICULOS ARTICULOS ARTICULOS, CONCEPTO_RECIBO
AULAS	Agregar Consultar Modificar Borrar	AULAS AULAS AULAS AULAS, GRUPOS
AREAS	Agregar Consultar Modificar Borrar	AREAS AREAS AREAS AREAS, CAT_CURSO
BANCOS	Agregar Consultar Modificar Borrar	BANCOS BANCOS BANCOS BANCOS, RECIBOS
BECAS	Agregar Consultar Modificar Borrar	BECAS BECAS BECAS BECAS, ALUMNO_CURSO
CLASIFICA_CURSO	Agregar Consultar Modificar Borrar	CLASIFICA_CURSO CLASIFICA_CURSO CLASIFICA_CURSO CLASIFICA_CURSO, CTA_CURSO
EDO_GRUPO	Agregar Consultar Modificar Borrar	EDO_GRUPO EDO_GRUPO EDO_GRUPO EDO_GRUPO, GRUPO
EDO_RECIBO	Agregar Consultar Modificar Borrar	EDO_RECIBO EDO_RECIBO EDO_RECIBO EDO_RECIBO, RECIBO
EDO_PROFESION	Agregar Consultar Modificar Borrar	EDO_PROFESION EDO_PROFESION EDO_PROFESION EDO_PROFESION, ALUMNO
EXAMENES	Agregar Consultar Modificar Borrar	EXAMENES EXAMENES EXAMENES EXAMENES, SOLICITUD
CURSO_ESP	Agregar Consultar Modificar Borrar	CURSO_ESP CURSO_ESP CURSO_ESP CURSO_ESP, MODULO_ESP
CAT_CURSO	Agregar Consultar Modificar Borrar	CAT_CURSO CAT_CURSO CAT_CURSO CAT_CURSO, GRUPO, PROG_CURSO, CURSO_SOFT
DEPENDENCIAS	Agregar	DEPENDENCIAS

	Consultar Modificar Borrar	DEPENDENCIAS DEPENDENCIAS DEPENDENCIAS, ALUMNOS
DEPARTAMENTOS	Agregar Consultar Modificar Borrar	DEPARTAMENTOS DEPARTAMENTOS DEPARTAMENTOS DEPARTAMENTOS, PERSONAL
DOCUMENTOS	Agregar Consultar Modificar Borrar	DOCUMENTOS DOCUMENTOS DOCUMENTOS DOCUMENTOS, SOLICITUD_DOC
ESC_FAC	Agregar Consultar Modificar Borrar	ESC_FAC ESC_FAC ESC_FAC ESC_FAC, ALUMNO
NIVELES	Agregar Consultar Modificar Borrar	NIVELES NIVELES NIVELES NIVELES, P_ALT_REND
NIVEL_ESCOLAR	Agregar Consultar Modificar Borrar	NIVEL_ESCOLAR NIVEL_ESCOLAR NIVEL_ESCOLAR NIVEL_ESCOLAR, ALUMNO
SEDES	Agregar Consultar Modificar Borrar	SEDES SEDES SEDES SEDES, GRUPO
P_ALTO_REND	Agregar Consultar Modificar Borrar	P_ALTO_REND P_ALTO_REND P_ALTO_REND P_ALTO_REND, ALUMNO
INSTITUCIONES	Agregar Consultar Modificar Borrar	INSTITUCIONES INSTITUCIONES INSTITUCIONES INSTITUCIONES, INST_CURSO
TIPO_CAT_CURSO	Agregar Consultar Modificar Borrar	TIPO_CAT_CURSO TIPO_CAT_CURSO TIPO_CAT_CURSO TIPO_CAT_CURSO, CAT_CURSO
TIPO_PERSONAL	Agregar Consultar Modificar Borrar	TIPO_PERSONAL TIPO_PERSONAL TIPO_PERSONAL TIPO_PERSONAL, ALUMNO, PERSONAL
TIPO_PERSONAL	Agregar Consultar Modificar Borrar	TIPO_PERSONAL TIPO_PERSONAL TIPO_PERSONAL TIPO_PERSONAL, SOLICITUD
MODULOS	Agregar Consultar Modificar Borrar	MODULOS MODULOS MODULOS MODULOS, CAT_MODULO, MODULO_ESP
SOFTWARE	Agregar Consultar Modificar Borrar	SOFTWARE SOFTWARE SOFTWARE SOFTWARE, CURSO_SOFT
TIPO_EGRESO	Agregar Consultar Modificar	TIPO_EGRESO TIPO_EGRESO TIPO_EGRESO

	Borrar	TIPO EGRESO, EGRESO
TIPO_GRUPO	Agregar Consultar Modificar Borrar	TIPO_GRUPO TIPO_GRUPO TIPO_GRUPO TIPO_GRUPO, GRUPOS
MEDIO_DIFUSION	Agregar Consultar Modificar Borrar	MEDIO_DIFUSION MEDIO_DIFUSION MEDIO_DIFUSION MEDIO_DIFUSION, SOLICITUD
TIPO_PAGO	Agregar Consultar Modificar Borrar	TIPO_PAGO TIPO_PAGO TIPO_PAGO TIPO_PAGO, RECIBO
PROGRAMAS	Agregar Consultar Modificar Borrar	PROGRAMAS PROGRAMAS PROGRAMAS PROGRAMAS, PROG_CURSO

El diagrama de Flujo del mantenimiento a los catálogos se detalla, en el Apéndice A, Sección II.

3.5.4 Diseño de reportes

Para el diseño de reportes se tomaran en cuenta las normas y lineamientos establecidos por centro, cabe mencionar que para generar alguno se tomaran en cuenta los parámetros iniciales que podrán ser dados por el usuario, como fechas, estados, grupos etc.

Para realizar este tipo de funcionalidad, se hace uso de procedimientos almacenados que maneja SQL SERVER junto con la funcionalidad que aporta Power Builder y el uso de sus datawindows.

3.6 Módulos que Componen el Sistema

El Sistema de Información Integral del Centro Coapa esta formado de los siguientes módulos:

1. Control Escolar
2. Informes y Relaciones
3. Infraestructura y Soporte
4. Delegación Administrativa

El SIICC del Centro Coapa cuenta con un subsistema de seguridad a nivel cliente, el cual permitirá otorgar permisos de acceso/restricción a los usuarios en los subsistemas que lo conforman. La seguridad cuenta con las siguientes consideraciones:

- Se tiene seguridad a nivel: subsistema, menú del subsistema controles de las ventanas del subsistema (botones, cajas de edición, etc.).
- El acceso/restricción consiste en habilitar o deshabilitar controles, opciones de menú, etc., así como visualizar o no los subsistemas respectivos.
- Los usuarios que requieran acceder al sistema y sus subsistemas, tendrán que ser dados de alta como parte del personal del centro.
- A cada usuario se le identificará por RFC y una palabra clave. A través del RFC y su palabra clave, el usuario podrá ingresar al sistema, de acuerdo a los privilegios que se le hayan otorgado.

Para cada modulo se despliegan una serie de catálogos que son administrados por cada departamento.

3.7 Beneficios del Sistema.

A continuación se describen los beneficios que se obtienen a partir de las características del sistema.

Características Funcionales	Beneficios
Diseño integral del sistema	<ul style="list-style-type: none"> • Esta constituido por varios sistemas de información y mediante el cual se puede apreciar en cualquier momento tanto el estado actual del centro como el estado de cada una de sus áreas. • Cubre necesidades actuales
Consultas	<ul style="list-style-type: none"> • Información oportuna para la toma de decisiones.
Eliminación de procesos manuales	<ul style="list-style-type: none"> • Agilización en los procesos
Diseño de entradas y consultas al sistema mediante interfaz gráfica	<ul style="list-style-type: none"> • Permite uso intuitivo y por tanto una rápida capacitación del usuario • Permite la disminución/eliminación de uso de papel
Diseño paramétrico /modular	<ul style="list-style-type: none"> • Facilidad de mantenimiento
Utilización de una base de datos relacional	<ul style="list-style-type: none"> • Integridad de la información • Facilidad de mantenimiento y crecimiento • Facilidad de la explotación de la información
Arquitectura Cliente /Servidor	<ul style="list-style-type: none"> • Actualización tecnológica • Flexibilidad y facilidad de mantenimiento y crecimiento
Diseño Distribuido	<ul style="list-style-type: none"> • Soporta la descentralización • Información en el lugar que se requiere (in-situ) • Autonomía de operación en los departamentos
Actualización tecnológica de acuerdo a estándares del mercado	<ul style="list-style-type: none"> • Menores costos y diversidad de infraestructura.

Capítulo 4 Liberación

En esta etapa se hace una presentación de la aplicación al usuario y se realizan pruebas operativas de forma individual y un ciclo completo, se hace entrega del manual de usuario, y se definen los parámetros generales.

4.1 Parámetros Generales

Para comenzar a utilizar el sistema SIIC es necesario que se definan los parámetros generales del centro. Nos referimos a los permisos de acceso a cada uno de los módulos, y las operaciones que podrán realizar. Los permisos iniciales serán dados en la instalación del sistema y estos serán asignados a los encargados de cada uno de los departamentos del Centro. Posteriormente estos permisos serán otorgados por cada uno de los encargados del Centro, al personal que tengan a su cargo. Los permisos podrán darse de alta en cada uno de los módulos en la opción de <Módulos> <Personal>, en esta ventana se van a dar de alta los datos personales, permisos, el login y el password.

En caso que alguien del personal olvide su password, se debe notificar al administrador del sistema o al encargado del departamento para que de la alta su nuevo password, y así pueda tener acceso al sistema.

DATOS DEL PERSONAL			
Rfc	NLSJ40509		
Nombre	Laura	Apellido	George
Calle	Nueva #12	Delegación	Alvaro Obregón
Colonia	Santa Fe	Cp	09090
Estado	P.F.	Tel Casa	53878899
Tel Casa	53878899	Tel Recado	567890345
E Mail	laura@holmail.com		
Categoría			
Ocupación	Administrado		
Horas Semana	1		
Departamento	Control Escolar	Tipo Personal	Jefe de Departamento
Contraseña	XXXXXXXXXX		
Permisos			
Lectura	<input checked="" type="checkbox"/>	Actualización	<input type="checkbox"/>
Insertion	<input checked="" type="checkbox"/>	Borrado	<input type="checkbox"/>
Nuevo	Buscar	Guardar	Salir

Fig 4.1 Pantalla de los Datos del Personal

4.2 Pruebas

Una vez instalado el sistema de información en el Centro Coapa, se realizaron las pruebas operativas en cada uno de los departamentos del centro y también se realizó un proceso completo. Para llevar a cabo estas pruebas se dieron de alta algunos registros en los catálogos del sistema.

Para dichas pruebas se realizaron las siguientes acciones:

- Se dieron de alta 10 grupos con estados diferentes (programado, abierto, cancelado).
- Se generaron grupos de cursos normales, diplomados y líneas de especialización para realizar las inscripciones.
- Solicitud de inscripción a cursos normales.
- Recibos por concepto de cursos, artículos de la tienda y otros servicios.
- Se dieron de alta pagos de servicios (luz, agua, teléfono, renta etc.).
- Se genero el archivo de la relación de alumnos que se les dará constancia
- Se simulo la inscripción de alumnos a un grupo que no tenía cupo, para verificar el tipo de mensajes enviados.

Se generaron los siguientes reportes, que se utilizan para rendir los informes trimestrales:

- Reporte de grupos internos: abiertos, cancelados, programados.
- Reporte de ingresos por grupos abiertos.
- Reporte de grupos internos: atendidos alumnos e ingresos, en una fecha dada
- Reporte de becas internas
- Reporte del medio por el cual se enteraron los alumnos
- Reporte de auditoría interna
- Lista definitiva
- Reporte de ingreso diario
- Reporte de total de ingresos a una fecha dada

En las pruebas realizadas se obtuvo la entera satisfacción del usuario, se cumplió con el objetivo del sistema, existen algunas cosas que falta por hacer, que podrían ser parte en desarrollos futuros dentro del centro.

4.3 Manual de Usuario

El propósito del manual es instruir al usuario sobre el uso del sistema, dirigiéndolo desde la entrada del sistema, hasta la operación de sus opciones. Se describe como puede realizar la operación deseada, que opciones y parámetros tienen disponibles, que valores tienen que tomar para elegir la opción seleccionada, así como condiciones de error y puntos importantes dentro de cada subsistema.

Después de seleccionado el icono de la aplicación aparece una ventana que te pide la clave y el password, estos parámetros son validados por el sistema después de ser introducidos.

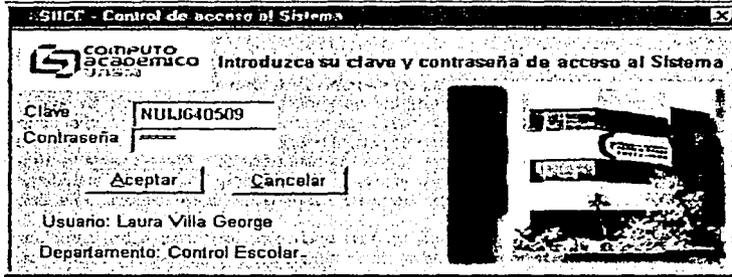


Fig. 4. 2 Pantalla de identificación del usuario

El usuario debe teclear su clave y password (que por seguridad no se muestra, sólo se despliegan asteriscos). Al oprimir el botón <Aceptar>, el sistema comprueba que la clave y el password sean válidos para permitir el acceso a las funciones que le corresponden, el botón de <Cancelar> permite abortar esta operación.

Siempre y cuando su clave esté registrada en el sistema, se desplegará la ventana principal y en la parte superior se mostrara un menú.

Al entrar al SIICC, solo podrás ver y elegir el módulo al que perteneces. Cada modulo tiene catálogos propios y reportes específicos.

4.3.1 Navegando en la aplicación.

Como se observa en la figura 4.3 Menú, al referirnos a un menú, estamos hablando de un conjunto de opciones que nos permiten abrir las ventanas pertenecientes a cada una de las aplicaciones, o desplegar nuevas opciones (submenús). A la vez, las ventanas al igual que los menús, tienen controles - botones, cajas de edición - que permiten llevar a cabo acciones tales como: aceptar datos, eliminarlos, modificarlos o consultarlos.

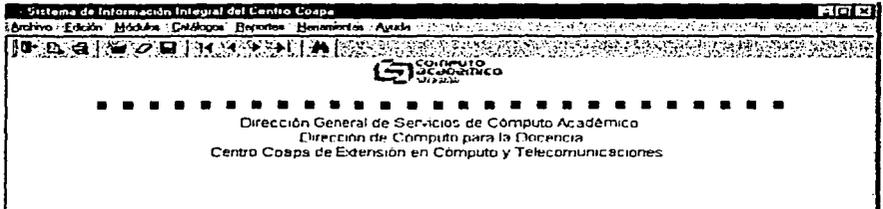


Fig. 4.3 Menú

El menú de cabecera tiene las siguiente opciones en <Archivo> Imprimir, Seleccionar impresora y Salir de la aplicación, el menú de <Edición> tiene las operaciones usuales de edición como son nuevo, eliminar guardar, en la opción <Módulos> despliega los módulos que componen el sistema, Control Escolar, Infraestructura y Soporte Tecnológico, Delegación Administrativa, Informes y Relacione, en la opción de <Catálogos> se tienen todos los catálogos que son parte de la base de datos estos pueden ser seleccionados de forma directa y realizar operaciones de actualización, borrado e inserción siempre y cuando se tenga permiso sobre ellos, para realizar este tipo de operaciones.

De esta forma solo el personal autorizado administra este tipo de operaciones sobre cada uno de los catálogos, guardan información útil para sus departamentos y solo ellos podrán realizar cambios sobre la información les interesa. También se tiene las opciones de Herramientas y Ayuda.

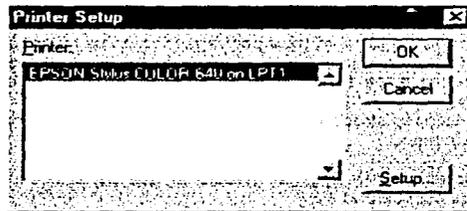


Fig. 4.4 Pantalla de selección de impresora

Barra de Herramientas.

La barra de herramientas le permite al usuario acceder ciertas funciones sin necesidad de tener que seleccionarlas a través del menú. Estas herramientas proveen de funcionalidad general para el funcionamiento de la aplicación. Los botones de la barra de herramientas son:



Fig. 4.5 Barra de Herramientas

-  Salir
-  Imprimir
-  Nuevo
-  Borrar
-  Guardar
-  Primero
-  Anterior
-  Siguiente
-  Ultimo

4.3.2 Navegando en los módulos.

Cuando se pretenda acceder a cualquiera de las aplicaciones en el cliente, se debe de considerar el siguiente procedimiento:

Identificar cual es el modulo deseado. Para hacerlo se elige en el menú principal de la opción <Módulos>:

PROCESO	MODULO	OPCION	OBSERVACIÓN
---------	--------	--------	-------------

Acceso a los módulos

	<Módulos><Delegación Administrativa>		Cada módulo cuenta con opciones especiales, que les permiten llevar a cabo sus tareas.
	<Módulos><Infraestructura y soporte Tecnológico>	y	
	<Módulos><Informes Relaciones>	y	
	<Módulos><Control Escolar>		

Principales pantallas de captura.

Datos del Alumno

Para acceder a esta opción, apóyese en la siguiente tabla:

PROCESO	MODULO	OPCION	OBSERVACIÓN
---------	--------	--------	-------------

Inscripción

Informes y Relaciones	<Modulo><Informes Relaciones><Forma Inscripción>	y de	La siguiente figura, nos muestra la ventana de captura de los datos del alumno, en el proceso de inscripciones, se cuenta con un registro histórico del alumno, a partir de este puede generarse una solicitud de inscripción a un curso normal o especial ¹ y a un examen. Si el registro del alumno no existe se pueden dar de alta por primera vez.
-----------------------	--	------	---

¹ Un curso especial es considerado un diplomado, línea de especialización y los módulos de capacitación para el trabajo.

Datos del Alumno			
Rfc: <input type="text" value="R760313"/> <input type="text" value="sollicitud"/>			
Nombre: <input type="text" value="JUVARES"/> <input type="text" value="PEDILLO"/> <input type="text" value="JOSE RAMON"/>		A. Paterno <input type="text" value="A. Materno"/> <input type="text" value="Nombre"/>	
Calle: <input type="text" value="EL PANTEON #27"/>		Colonia: <input type="text" value="SANTIAGO TEPALCATLAPAN"/>	
Delegación: <input type="text" value="XICMILCO"/>		Estado: <input type="text" value=""/>	
CP: <input type="text" value="16701"/>		Email: <input type="text" value="jramonr@telmex.com.mx"/>	
Teléfono: <input type="text" value="5354330"/> <input type="text" value="5775541"/> <input type="text" value="5768800"/>		Particular <input type="text" value="Oficina"/> <input type="text" value="Recibo"/>	
SE INSCRIBE A LOS CURSOS COMO:			
Precedencia: <input type="text" value="Particular"/>		Nivel Escolar: <input type="text" value="Licenciatura"/>	
Escuela o Facultad: <input type="text" value="Facultad de Estudios Superiores (F.E.S.) Zaragoza"/>		NO Cuenta <input type="text" value=""/>	
Edo Profesión: <input type="text" value="Titulado"/>			
¿ES ALUMNO DE ALTO RENDIMIENTO DELEGACIONAR EL TIPO			
Programa de Alto Rendimiento: <input type="text" value=""/>			
SI ES PERSONAL DEL CENTRO, INDICAR EL TIPO			
Tipo Personal: <input type="text" value=""/>			
Trabajo: <input type="text" value=""/>			
Puesto: <input type="text" value=""/>			
<input type="button" value="NUEVO"/> <input type="button" value="BUSCAR"/> <input type="button" value="GUARDAR"/> <input type="button" value="SALIR"/>			

Fig. 4.6 Datos del Alumno

Ventana de captura de la Solicitud

En esta ventana se captura la solicitud de inscripción, se anexan los documentos expuestos por el alumno.

Solicitud			
Solicitud	<input type="text" value="10.00"/>	DOCUMENTOS ANEXOS	Grupo: <input type="text" value="AM12-01"/>
Digito Clave	Trámite de inscripción preventivo y correctivo para microcomputadoras personales y periféricos		
Tipo Solicitud	<input type="text" value="Curso"/>		
Num. Recibo	<input type="text" value="145"/>		
Medio Distorsión	<input type="text" value="Solicitud"/>		
Examen	<input type="text" value=""/>		
<input type="button" value="NUEVO"/> <input type="button" value="GUARDAR"/> <input type="button" value="CERRAR"/>			

Fig. 4.7 Ventana de la Solicitud

TESIS CON FALLA DE ORIGEN

Documentos Anexos

Para acceder a esta opción, apóyese en la siguiente tabla:

PROCESO	MODULO	OPCION	OBSERVACIÓN
---------	--------	--------	-------------

Alta
de documentos
anexos a una solicitud

Informes Relaciones	y	<Modulo><Informes Relaciones><Forma Inscripción>	y de	Cuando oprimimos el botón de <DOCUMENTOS ANEXOS> en la solicitud se despliega la ventana, en la que pueden darse de alta los documentos.
Informes Relaciones	y	<Modulo><Informes Relaciones><Documentos Anexos>	y	

Documentos Anexos a la Solicitud

No de Solicitud: 1020

Seleccionar los Documentos anexados a la solicitud

- Titulo
- Cédula Profesional
- Carta de Exposición de Motivos
- Resumen de Curriculum Vitae Actualizado (3 paginas)
- Copia de Certificado de Estudios Completos
- Carta de Presentación de la Empresa donde trabaja (una cuartilla)

ACEPTAR CERRAR

Fig. 4.8 Ventana Documentos anexos

Programación de Grupos

Para acceder a esta opción, apóyese en la siguiente tabla:

PROCESO	MODULO	OPCION	OBSERVACIÓN
---------	--------	--------	-------------

Programación de grupos

Control Escolar	<Modulo><Control Escolar><Grupo>	Por medio de esta ventana se podrán dar de alta nuevo grupos o modificaciones. En esta ventana, se puede dar de alta un nuevo grupo, aun si no se cuenta con los datos para el aula y el profesor.
-----------------	----------------------------------	---

DATOS DEL GRUPO

Grupo	AM12-01		
Fecha Inicio	01/11/2001	Fecha Fin	10/12/2001
H Inicio	07:00	H Fin	19:00
Edo Grupo	Abierto		
Dgsca Clave	Taller de mantenimiento preventivo y correctivo menor para microcomputadoras personales		
Aula	Aula 2 cursos	Sede	Centro Compa
Tipo Grupo	Calendario - Lunes a Viernes		
Profesor	PEDRO RUIZ NAPOLES		
Maximo cupo	6		
<input type="button" value="NUEVO"/> <input type="button" value="BUSCAR"/> <input type="button" value="GUARDAR"/> <input type="button" value="CERRAR"/>			

Fig. 4. 9 Ventana de Datos del Grupo

Recibo

Para acceder a esta opción, apóyese en la siguiente tabla:

PROCESO	MODULO	OPCION	OBSERVACIÓN
---------	--------	--------	-------------

Generación de recibo

Delegación Administrativa	<Modulo><Delegación Administrativa><Caja>	A partir de esta ventana se genera recibo donde se registra el pago de un curso u otros conceptos, que generan ingresos para el Centro.
---------------------------	---	---

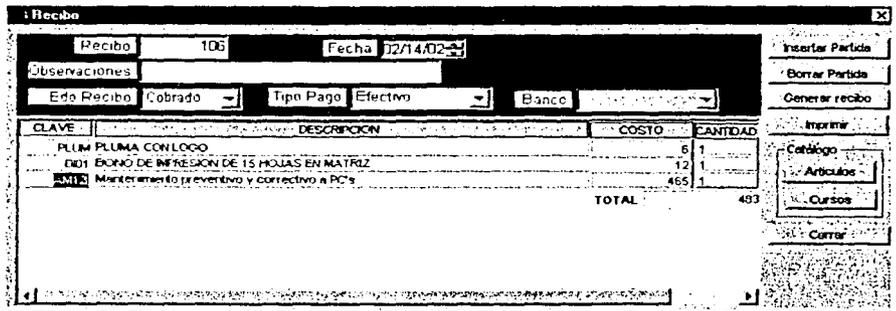
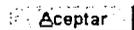


Fig. 4. 10 Ventana de captura del recibo

Funcionamiento de los botones

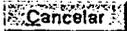
En general, la mayoría de las ventanas del sistema cuentan con los botones que se muestran a continuación. Esto permite que la interface sea consistente en toda la aplicación y su funcionalidad es la misma en cualquier ventana. Los botones generales son los siguientes:

Botón ACEPTAR



Este botón se utiliza para que los cambios hechos en la ventana se reflejen físicamente. Si se realizaron cambios en la ventana y no se oprime este botón, los cambios no se verán reflejados.

Botón CANCELAR



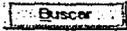
Cierra la ventana activa.

Botón NUEVO



Inserta un nuevo registro de información en la ventana en la que se esté trabajando. Este registro se actualizará en la base de datos solamente cuando se oprima el botón GUARDAR.

Botón BUSCAR



La función que realiza este botón es la de llevar a cabo la selección de un registro según las condiciones que se ingresaron.

Botón GUARDAR



Este botón se utiliza para que los cambios hechos en la ventana se reflejen físicamente en la base de datos que se este utilizando en ese momento. Si se realizaron cambios en la ventana y no se oprime este botón, los cambios no se verán reflejados.

Botón CERRAR



Cierra la ventana en la que se está trabajando. Si se realizaron algunos cambios en la ventana y no se oprimió el botón de GUARDAR, estos cambios no se verán reflejados en la base de datos.

Botón IMPRIMIR



Realiza la impresión del documento que se este generando en ese momento.

Botón INSERTAR PARTIDA



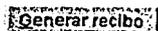
Agrega una nueva línea de concepto en la pantalla de recibo.

Botón BORRAR PARTIDA



Borra una línea de concepto seleccionada.

Botón GENERAR RECIBO



Genera el recibo que se esta visualizando y manda imprimir automáticamente.

4.3.3 Navegando por los Catálogos

Para realizar operaciones sobre los Catálogos, apóyese en la siguiente tabla:

PROCESO	MÓDULO	OPCIÓN	OBSERVACIÓN
---------	--------	--------	-------------

Alta de los registros en los Catálogos

Para cualquier módulo del sistema	<Catálogos> Selección de catálogo deseado	Haciendo uso de las opciones de la barra de herramientas o la opción del menú <Edición> se puede agregar, modificar, borrar, guardar y exportar los datos del catálogo que tenemos en pantalla a un archivo. La opción para exportar un archivo al formato que se desee se localiza en la opción <Archivo><Exportar> del menú principal.
-----------------------------------	---	---

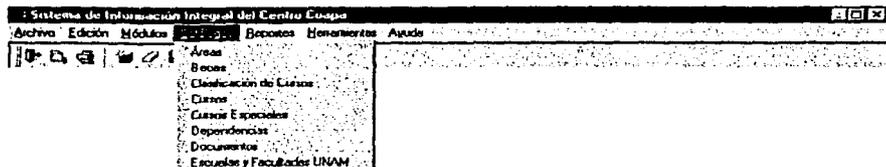


Fig. 4. 11 Opción Catálogos

En la figura siguiente vemos cual es la ventana general de captura, para los catálogos.

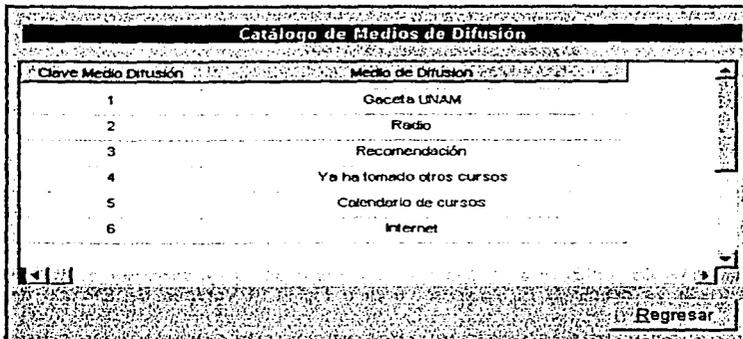


Fig. 4. 12 Ventana General de Captura de Catálogos

4.3.4 ¿Cómo se genera un reporte?

Existen reportes que para ser generados requieren parámetros especiales de entrada como pueden ser fechas, estados, grupo, etc., y reportes generales que no requieren de ninguna entrada de datos para poder ser generados.

Para acceder a esta opción, apóyese en la siguiente tabla:

PROCESO	MÓDULO	OPCIÓN	OBSERVACIÓN
Generar reportes		<Reportes>	Para generar un reporte seleccionamos la opción <Reportes> de menú principal, se elige el reporte deseado. Como se puede observar en la Fig. 4.4 Opción Reportes

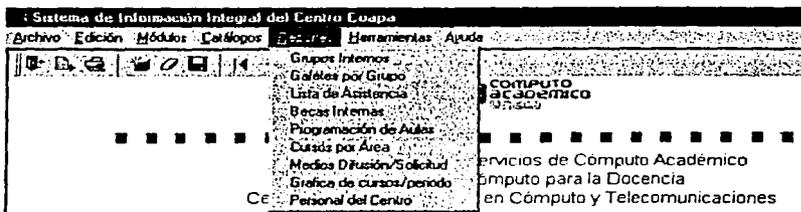


Fig. 4. 13 Opción Reportes

Inmediatamente después de realizada esta operación para el caso de reportes que requieren parámetros de entrada se despliega la siguiente ventana Fig. 4.14 Ventana de Entrada de Parámetros.



Fig. 4. 14 Ventana de Entrada de Parámetros

Generar reportes desde el contenedor

PROCESO	MODULO	OPCION	OBSERVACIÓN
---------	--------	--------	-------------

Generar reportes

Desde cualquier módulo	<Reportes>	En el contenedor de reportes contamos con los siguientes botones, el de <PARAMETROS> nos permite acceder a la ventana de entrada de parámetros, desde esta se pueden modificar los datos de entrada como fechas, grupos, etc. El botón de <REGRESAR> me permite salirme del contenedor.
------------------------	------------	---

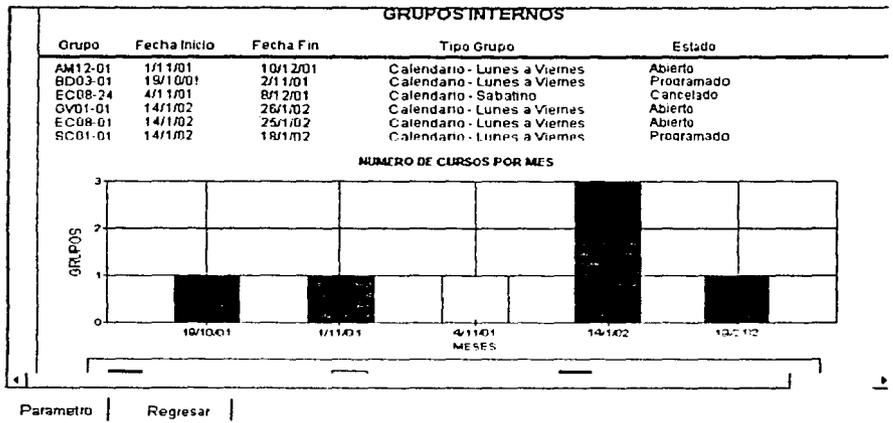


Fig. 4.15 Contenedor de Reportes.

Conclusiones y Trabajo Futuro

- El Centro de Computo DGSCA Coapa, originalmente manejaba su información de forma aislada, algunos departamentos utilizaban programas de computadora que los auxiliaban en sus tareas, existen formatos que actualmente se siguen utilizando ya que son el comprobante de que un alumno se registro en un curso. El proceso de inscripción se ha sistematizado, ahora contamos con un registro histórico de los datos de un alumno, para la generación automática de estadísticas, becas y consulta de calificaciones en cursos anteriores de una forma rápida y eficiente.
- Esto nos permite otorgar descuentos adicionales del 10% a los alumnos que lo acrediten, sin necesidad de presentar la constancia y el recibo del curso anterior, ya que contamos con el historial del alumno en el centro,.
- Se automatizo la generación de una solicitud a partir de la generación de un recibo, con esto se elimina el problema de cuadrar el número de solicitudes, que fueron aceptadas en el mostrador de Informes y Relaciones con el número de recibos que fueron generados por la Delegación Administrativa
- A partir de la fase del análisis se detecto que las políticas de trabajo en el Centro DGSCA Coapa están dadas por DGSCA CU, esto permite que dicho sistema se adecue no solo para las necesidades del centro, sino de igual forma puede ser utilizado para el centro Nuevo León, Mascarones y dentro de Ciudad Universitana.
- El diseño del sistema nos permite agregar funciones futuras en cada uno de los módulos.
- Un área que sería candidato a desarrollo es la Delegación Administrativa, en su manejo de inventarios. En este momento no fue parte del desarrollo ya que se pretendía solucionar la administración de los cursos y las tareas que intervienen en este proceso.
- En lo que se refiere a los reportes, estos pueden ser diseñados a partir del modelo de la base de datos sin ningún problema
- En cuanto al archivo que se genera después de un periodo de cursos, para la entrega de constancias, puede enviarse via correo electrónico automáticamente, al departamento de Control Escolar de DGSCA CU.
- Con el uso de una metodología en el desarrollo del sistema, se aplicaron los conocimientos teóricos adquiridos en las aulas, esto me permitió aplicarlos en un problema real. La experiencia adquirida ahora es parte de mi formación profesional
- Power Builder era una herramienta que se desconocía, fue seleccionada por el manejo que tiene para las bases de datos y el desarrollo de reportes a partir de los procedimientos almacenados. Es una herramienta que nos permitió trabajar con los recursos con los que cuenta el Centro de Computo.

Bibliografía

Date, C. J. An Introduction to Database Systems Addison-Wesley Reading, MA, 1990

Gane, Chris Computer – Aided Software Engineering – the Methodologies, the Products and the future Prentice Hall USA 1990

Codd, E. F. A relational Model for Large Shared Data Banks Communications of the ACM June 1970

Kendall Kenneth. Análisis y diseño de sistemas. Prentice Hall México 1991

Sommerville Ian. Ingeniería de Software. Un enfoque practico Mc Graw Hill 1998

Database Programming & Design, Case Getting the Big Picture by Paul Winsberg

Communications of the ACM, A simple guide to five normal forms in relational database theory by William Kent Internacional Bussiness Machines Corporation

Apéndice A

Sección I

Las relaciones que existen entre cada una de estas entidades se describen a continuación:

Un ALUMNO "puede estar registrado" a uno y sólo un P_ALT_REMTO y un P_ALT_REMTO "debe estar perteneciendo" a uno o más ALUMNOS



Fig. 1 Relación ALUMNO – P_ALTO_REMTO

Un ALUMNO "puede estar perteneciendo" a una y sólo una DEPENDENCIA y una DEPENDENCIA "puede estar albergando" a uno o más ALUMNOS.

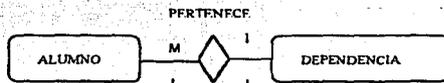


Fig. 2 Relación ALUMNO – DEPENDENCIA.

Un ALUMNO "debe ser clasificado" por uno y sólo un NIVEL_ESCOLAR y un NIVEL_ESCOLAR "puede estar clasificando" a uno o más ALUMNOS.

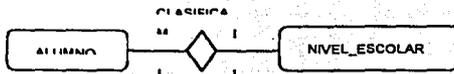


Fig. 3 Relación ALUMNO – NIVEL_ESCOLAR.

Un ALUMNO "puede estar sugiriendo" una y sólo una PROPUESTA y una PROPUESTA "debe ser sugerida" por uno o más ALUMNOS.

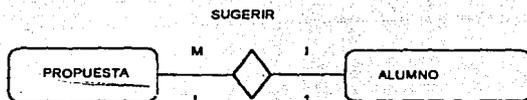


Fig. 4 Relación PROPUESTA - ALUMNO.

Un ALUMNO "puede estar clasificado" en uno y sólo un EDO_PROFESION y un EDO_PROFESION "debe estar clasificando" a uno o más ALUMNOS



Fig. 5 Relación ALUMNO - EDO_PROFESION.

Un ALUMNO "puede estar estudiando" en una y sólo una Esc_FAC y en una Esc_FAC "pueden estar registrados" a uno o más ALUMNOS

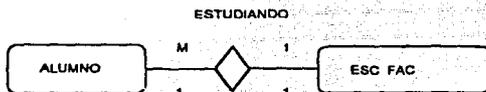


Fig. 6 Relación ALUMNO - ESC_FAC.

Una CALIF_PREVIO "debe estar perteneciendo" uno y un sólo ALUMNO y un ALUMNO "puede ser acreedor" a uno o más CALIF_PREVIO.



Fig. 7 Relación CALIF_PREVIO - ALUMNO_CURSO.

Un ALUMNO "puede estar clasificado" en uno y sólo un TIPO_PERSONAL y un TIPO_PERSONAL "puede estar clasificando" a uno o más ALUMNOS



Fig. 8 Relación ALUMNO - TIPO_PERSONAL.

Un TIPO_PERSONAL "debe estar clasificando" a uno o más PERSONAL y un PERSONAL "puede estar clasificado" en uno y sólo un TIPO_PERSONAL

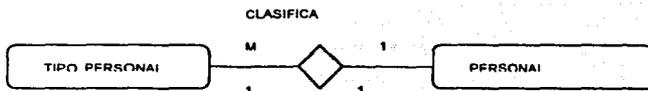


Fig. 9 Relación TIPO_PERSONAL - PERSONAL.

Un ALUMNO_CURSO "debe ser para" uno y sólo un ALUMNO y un ALUMNO "puede estar inscrito" en uno o más ALUMNO_CURSO.

Un ALUMNO_CURSO "debe ser para" uno y sólo un GRUPO y un GRUPO "puede ser tomado" por uno o más ALUMNO_CURSO.

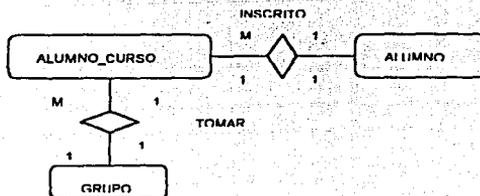


Fig. 10 Relación GRUPO - ALUMNO_CURSO - ALUMNO.

Una BECA "puede ser para" uno y sólo un ALUMNO_CURSO y un ALUMNO_CURSO "puede ser acreedor" a una y sólo una BECA.



Fig. 11 Relación ALUMNO_CURSO - BECA.

Un GRUPO "debe ser clasificado" por uno y sólo un EDO_GRUPO y un EDO_GRUPO "puede estar clasificando" a uno o más GRUPOS.

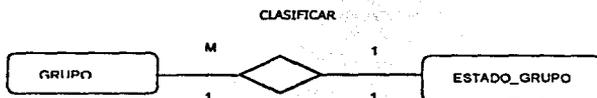


Fig. 12 Relación GRUPO - ESTADO_GRUPO.

Un GRUPO "puede ser impartido" en una y sólo una SEDE y una SEDE "debe estar albergando" uno o más GRUPOS

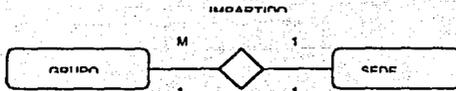


Fig. 13 Relación GRUPO - SEDE.

Un GRUPO "puede ser impartido" en una y sólo una AULA y una AULA "debe estar albergando" a uno y sólo un GRUPO

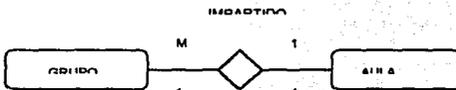


Fig. 14 Relación GRUPO - AULA

Un GRUPO "puede ser clasificado" por un y sólo un TIPO_GRUPO y una TIPO_GRUPO "debe estar clasificando" a uno o más GRUPOS



Fig. 15 Relación GRUPO - TIPO_GRUPO

Un GRUPO "puede ser contenido" en un y sólo un CAT_CURSO y un CAT_CURSO "debe estar conteniendo" a uno o más GRUPOS

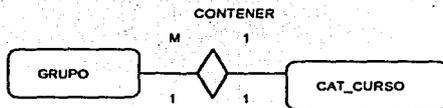


Fig. 16 Relación GRUPO - CAT_CURSO

Un INST_CURSO "debe ser para" uno y sólo un GRUPO y un GRUPO "puede estar contenido" en uno o más INST_CURSO.

Un INST_CURSO "debe ser para" uno y sólo una INSTITUCION y INSTITUCION "puede estar contenido" por uno o más INST_CURSO.

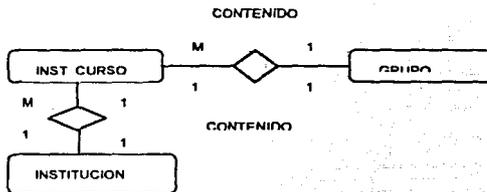


Fig. 17 Relación INSTITUCION - INST_CURSO - GRUPO

Una SOLICITUD "puede estar utilizando" uno y sólo un MEDIO_DIFUSION y un MEDIO_DIFUSION "debe ser utilizado" en una o más SOLICITUDES

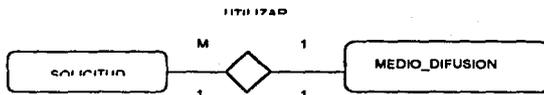


Fig. 18 Relación SOLICITUD - MEDIO_DIFUSION

Una SOLICITUD "puede ser clasificada" por uno y sólo un TIPO_SOLICITUD y un TIPO_SOLICITUD "debe estar clasificando" a una o más SOLICITUDES

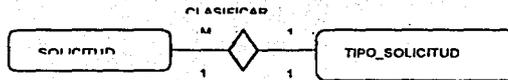


Fig. 19 Relación SOLICITUD - TIPO_SOLICITUD

Una SOLICITUD "puede estar solicitada" por uno y sólo un ALUMNO_CURSO y un ALUMNO_CURSO "debe estar contenida" en una y sólo una SOLICITUD

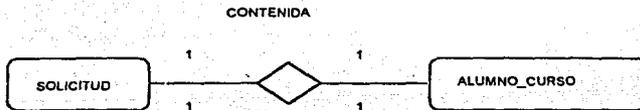


Fig. 20 Relación SOLICITUD - ALUMNO_CURSO

Una SOLICITUD "debe estar generando" a uno y sólo un RECIBO y un RECIBO "puede ser generado" por una o más SOLICITUDES

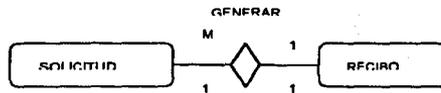


Fig. 21 Relación SOLICITUD - RECIBO

Un RECIBO "debe estar clasificado" por uno y sólo un ESTADO_RECIBO y ESTADO_RECIBO un "puede estar clasificando" a uno o más RECIBOS.



Fig. 22 Relación RECIBO - ESTADO_RECIBO.

Un RECIBO "debe ser pagado" por uno y sólo un TIPO_PAGO y un TIPO_PAGO "puede ser una forma de pago" de uno o más RECIBOS.

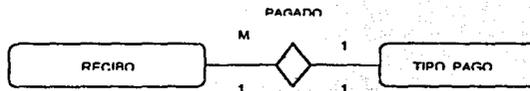


Fig. 23 Relación RECIBO - TIPO_PAGO

Un RECIBO "debe ser pagado" por uno y sólo un BANCO y un BANCO "puede estar pagando" uno o más RECIBOS.

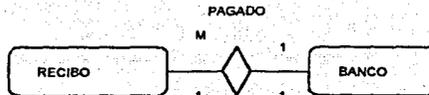


Fig. 24 Relación RECIBO - BANCO

Un CAT_CURSO "debe ser clasificado" por una y sólo una AREA y un AREA "puede estar clasificando" uno o más CAT_CURSOS.

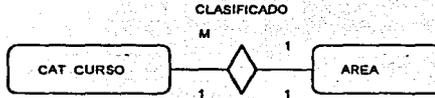


Fig. 25 Relación CAT_CURSO - AREA

Un CAT_SOFTWARE "debe ser para" uno y sólo un CAT_CURSO y un CAT_CURSO "puede estar contenido" en uno o más CAT_SOFTWARE.

Un CAT_SOFTWARE "debe ser para" uno y sólo un SOFTWARE y un SOFTWARE "puede estar contenido" por uno o más CAT_SOFTWARE.

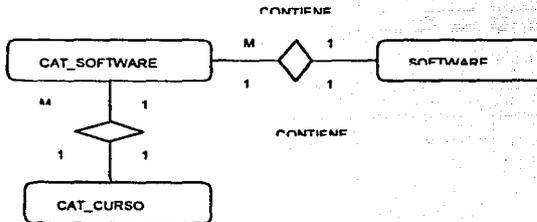


Fig. 26 Relación CAT_CURSO - CAT_SOFTWARE - SOFTWARE

Un CAT_MODULO "debe ser para" uno y sólo un MODULO y un MODULO "puede estar contenido" en uno o más CAT_MODULO.

Un CAT_MODULO "debe ser para" uno y sólo un CAT_CURSO y un SOFTWARE "puede estar contenido" por uno o más CAT_SOFTWARE.

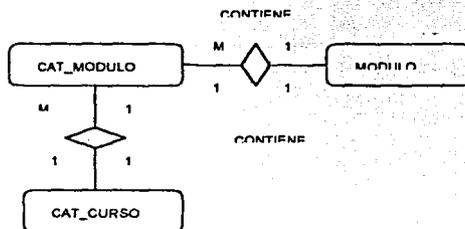


Fig. 27 Relación CAT_CURSO - CAT_MÓDULO - MÓDULO.

Un PROG_CAT_CURSO "debe ser para" uno y sólo un CAT_CURSO y un CAT_CURSO "puede estar contenido" en uno o más PROG_CAT_CURSO.

Un PROG_CAT_CURSO "debe ser para" uno y sólo un PROGRAMA y un PROGRAMA "puede estar contenido" por uno o más PROG_CAT_CURSO.

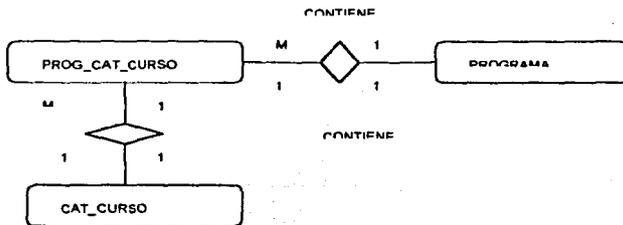


Fig. 28 Relación CAT_CURSO - PROG_CAT_CURSO - PROGRAMA

Un CONCEPTO_RECIBO "puede estar conteniendo" uno y sólo un ARTICULO y un ARTICULO "debe ser contenido" en uno o más CONCEPTO_RECIBOS

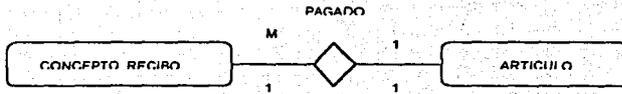


Fig. 29 Relación CONCEPTO_RECIBO - ARTICULO

Un CONCEPTO_RECIBO "debe estar registrado" en uno y sólo un RECIBO y un RECIBO "puede estar conteniendo" uno o más CONCEPTO_RECIBOS

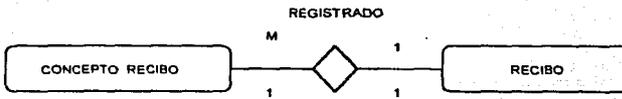


Fig. 30 Relación CONCEPTO_RECIBO - RECIBO

Un EGRESO "debe estar clasificado" por uno y sólo un TIPO_EGRESO y un TIPO_EGRESO "puede estar clasificando" uno o más EGRESOS

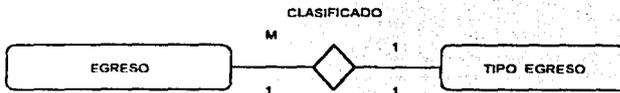


Fig. 31 Relación EGRESO - TIPO_EGRESO

Un PERSONAL "debe estar asignado" a uno y sólo un DEPARTAMENTO y un DEPARTAMENTO "puede estar albergando" a uno o más de PERSONAL.

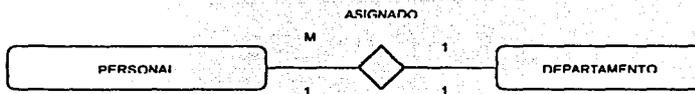


Fig. 32 Relación PERSONAL - DEPARTAMENTO

Un EXAMEN "debe estar asignado" a uno y sólo una SOLICITUD y una SOLICITUD "puede estar albergando" a uno o más de EXAMEN.

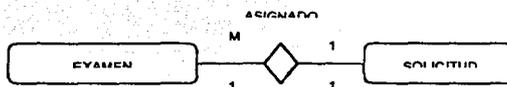


Fig. 33 Relación EXAMEN - SOLICITUD

Una SOLICITUD_Doc "debe ser para" uno y sólo una SOLICITUD y una SOLICITUD "puede estar contenido" en uno o más SOLICITUD_Doc.

Una SOLICITUD_Doc "debe ser para" uno y sólo un DOCUMENTO y un DOCUMENTO "puede estar contenido" por uno o más SOLICITUD_Doc

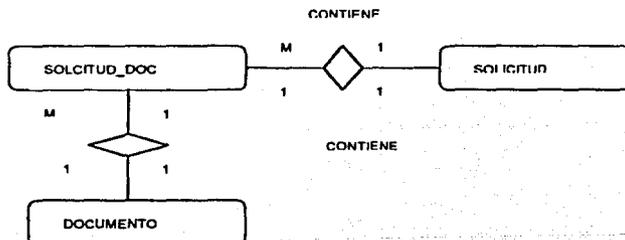


Fig. 34 Relación DOCUMENTO - SOLICITUD_DOC - SOLICITUD

Un GRUPO "debe ser para" uno y sólo un PROFESOR y un PROFESOR "puede estar contenido" en uno o más GRUPOS.

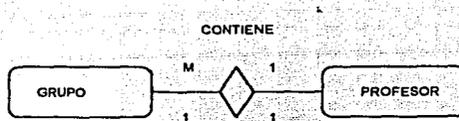


Fig. 35 Relación GRUPO - PROFESOR

Un MODULO_ESP "debe ser para" uno y sólo un MODULO y un MODULO "puede estar contenido" en uno o más MODULO_ESP.

Un MODULO_ESP "debe ser para" uno y sólo un CURSO_ESP y un CURSO_ESP "puede estar contenido" por uno o más MODULO_ESP

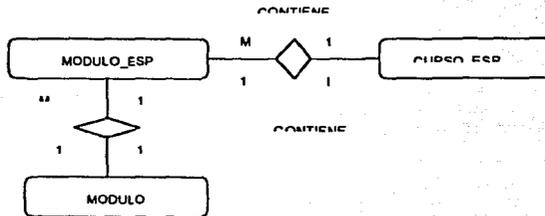


Fig. 36 Relación MODULA - MODULO_ESP - CURSO_ESP

Un P_ALTO_REND "debe estar asignado" a uno y sólo un NIVEL y un NIVEL "puede estar albergando" a uno o más de P_ALTO_REND.

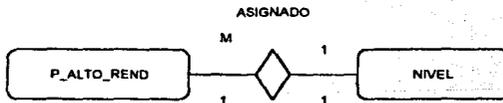


Fig. 37 Relación P_ALT_REND - NIVEL

Un CAT_CURSO "puede ser clasificado" por un y sólo un TIPO_CAT_CURSO y una TIPO_CAT_CURSO "debe estar clasificando" a uno o más CAT_CURSO

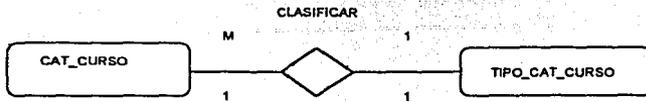


Fig. 38 Relación CAT_CURSO - TIPO_CAT_CURSO

Un CAT_CURSO "puede ser clasificado" por un y sólo un CLASIFICA_CURSO y una CLASIFICA_CURSO "debe estar clasificando" a uno o más CAT_CURSO



Fig. 39 Relación CAT_CURSO - CLASIFICA_CURSO

Sección II

Mantenimiento de los Catálogos

El mantenimiento a catálogos del sistema se refiere a la posibilidad de agregar, modificar, borrar y consultar información de los catálogos.

El siguiente diagrama muestra el flujo de información y los procesos involucrados en este modulo.

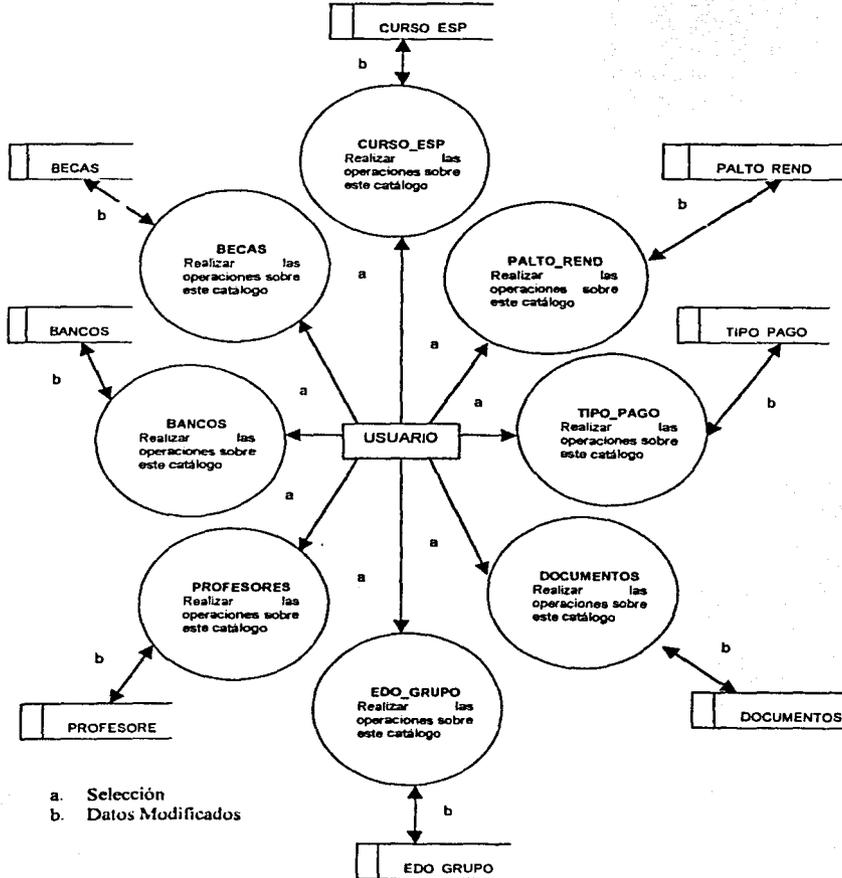


Fig. 1 Diagrama de Flujo de Mantenimiento de Catálogos

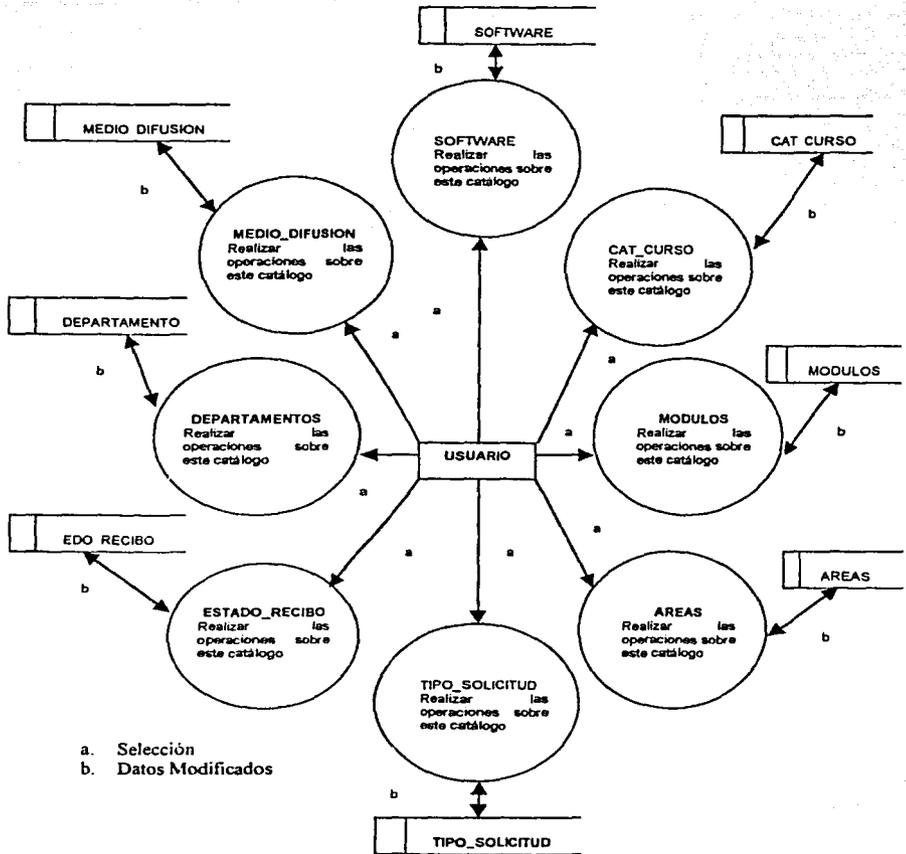


Fig. 2 Diagrama de Flujo de Mantenimiento de Catálogos

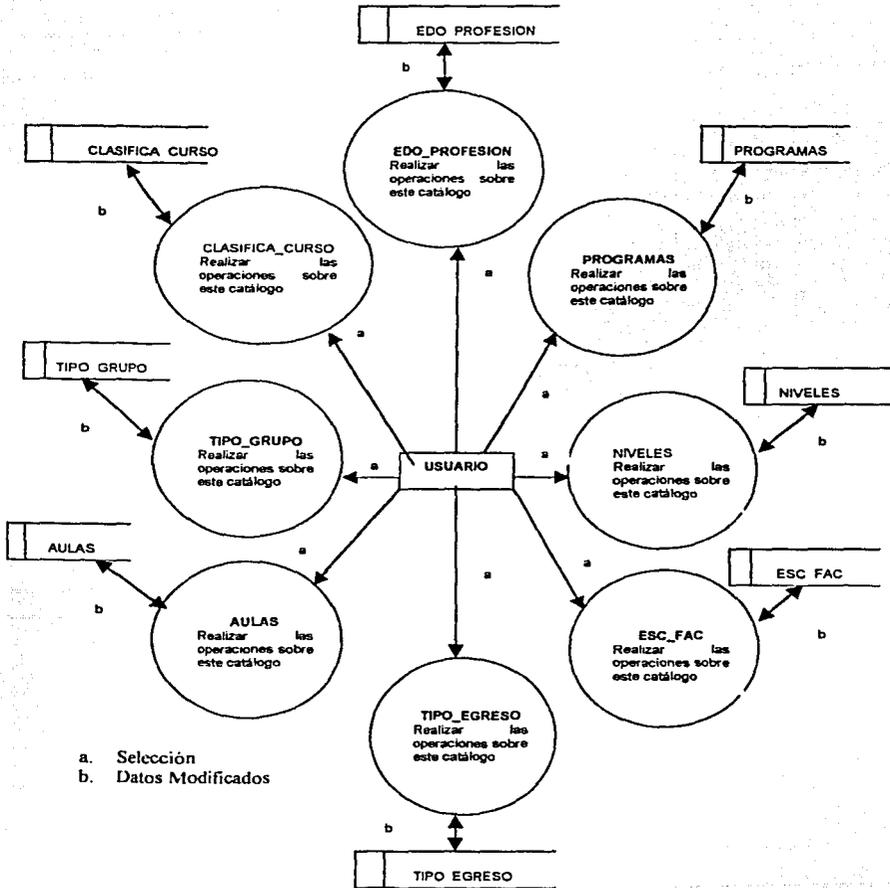


Fig. 3 Diagrama de Flujo de Mantenimiento de Catálogos

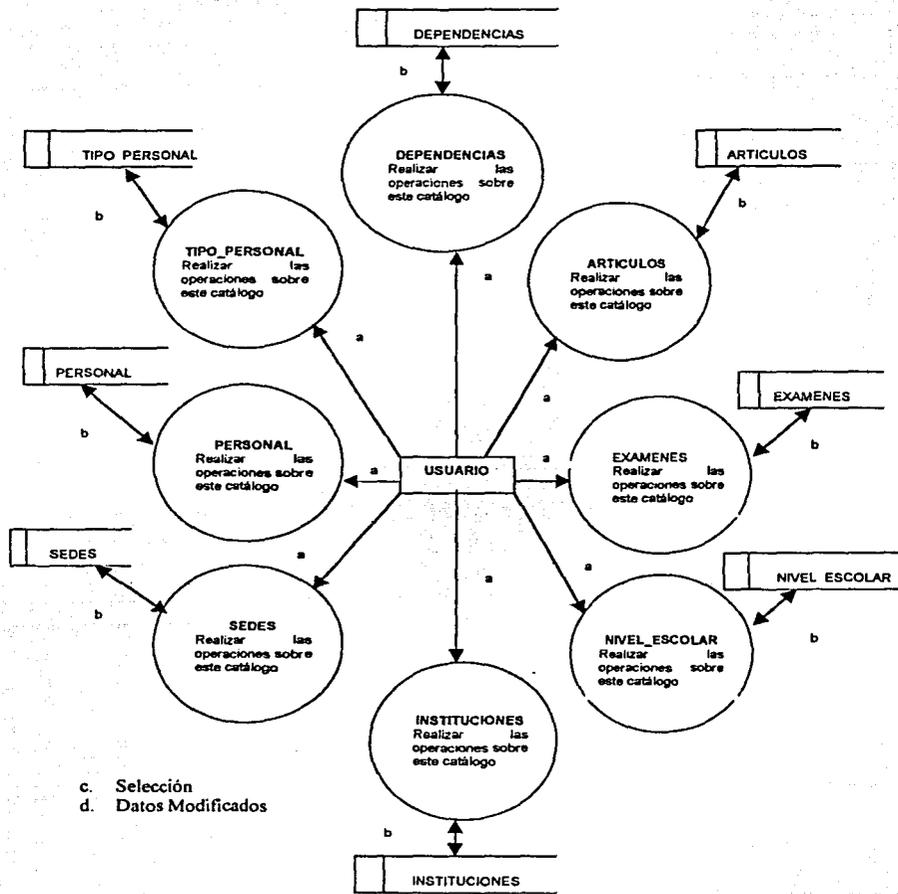


Fig. 4 Diagrama de Flujo de Mantenimiento de Catálogos

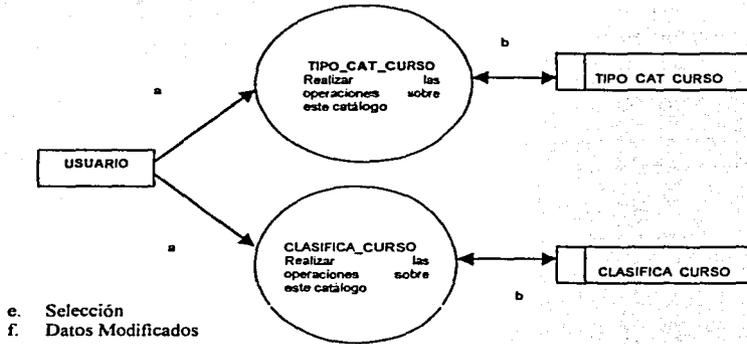


Fig. 5 Diagrama de Flujo de Mantenimiento de Catálogos

Apéndice B

ALUMNOS

CAMPO	TIPO		OBSERVACIONES
id_alumno	INTEGER	NOT NULL	Identificador de la entidad ALUMNO
rfc	VARCHAR(10)	NOT NULL	Es el rfc del alumno
nom_alumno	VARCHAR(25)	NOT NULL	Nombre del alumno
a_materno	VARCHAR(25)	NOT NULL	Apellido materno
a_paterno	VARCHAR(25)	NOT NULL	Apellido paterno
calle	VARCHAR(30)	NOT NULL	Forma parte de la dirección del alumno
Colonia	VARCHAR(50)	NOT NULL	Forma parte de la dirección del alumno
Delegación	VARCHAR(50)	NULL	Forma parte de la dirección del alumno
estado		NULL	Forma parte de la dirección del alumno
CP	INTEGER	NULL	Código de la zona postal
Tel_oficina	INTEGER	NULL	Teléfono de la oficina del alumno
Tel_casa	INTEGER	NULL	Teléfono de su casa
Tel_recado	INTEGER	NULL	Teléfono en donde se le puede dejar recados
E_mail	VARCHAR(30)	NULL	Correo electrónico del alumno
No_cuenta	INTEGER	NULL	En caso de ser estudiante de la UNAM.
Trabajo	CHAR	NULL	En caso de tener un trabajo
ocupacion	CHAR	NULL	Nombre de la profesión del alumno
puesto	CHAR	NULL	Nombre del puesto que ocupa
id_escolar	INTEGER	NOT NULL	Identificador del nivel escolar del alumno
id_tipo_personal	INTEGER	NULL	Identificador del tipo de personal
id_prog_alto_ren	INTEGER	NULL	Identificador de programa de alto rendimiento
id_edo_profesion	INTEGER	NOT NULL	Identificador del estado de la profesión de alumno
id_esc_fac	INTEGER	NOT NULL	Identificador de escuela o facultad de la UNAM.
id_dep	INTEGER	NOT NULL	Identificador de la dependencia

ALUMNO_CURSO

CAMPO	TIPO		OBSERVACIONES
id_alumno	INTEGER	NOT NULL	Identificador alumno
id_curso	INTEGER	NOT NULL	Identificador curso
id_beca	INTEGER	NOT NULL	Identificador de beca
Calificación	FLOAT	NULL	Calificación del alumno en el curso
Asistencia	INTEGER	NOT NULL	Asistencia durante el curso
Constancia	BIT	NOT NULL	Constancia del curso

ÁREAS

CAMPO	TIPO		OBSERVACIONES
id_área	INTEGER	NOT NULL	Identificador de área
Nom_área	VARCHAR(45)	NOT NULL	Nombre del área

ARTICULOS

CAMPO	TIPO		OBSERVACIONES
Id_art	INTEGER	NOT NULL	Identificador de artículo
Nom_art	VARCHAR(50)	NOT NULL	Nombre del artículo
Costo	MONEY	NOT NULL	Costo del artículo

AULAS

CAMPO	TIPO		OBSERVACIONES
Id Aula	INTEGER	NOT NULL	Identificador
Num Maq	INTEGER	NOT NULL	Número de máquinas
Nom aula	CHAR	NOT NULL	Nombre del aula

BANCOS

CAMPO	TIPO		OBSERVACIONES
Id_banco	INTEGER	NOT NULL	Identificador del banco
Nom_banco	CHAR	NOT NULL	Nombre del banco
Observaciones	VARCHAR(50)	NULL	Observaciones del banco

BECAS

CAMPO	TIPO		OBSERVACIONES
Id_beca	INTEGER	NOT NULL	Identificador tipo de beca
Nom_beca	VARCHAR(50)	NOT NULL	Nombre de la beca
Porc_desc	FLOAT	NULL	Porcentaje de descuento

CALIF_PREVIO

CAMPO	TIPO		OBSERVACIONES
Id_alumno	INTEGER	NOT NULL	Identificador alumno
Calificación	FLOAT	NOT NULL	Calificación de examen previo
fecha	DATE	NULL	Fecha de aplicación

CAT_CURSO

CAMPO	TIPO		OBSERVACIONES
Dgsca_clave	CHAR	NOT NULL	Identificador del curso
Nom_oficial	VARCHAR()	NOT NULL	Nombre oficial
Nom_comercial	VARCHAR()	NOT NULL	Nombre comercial
No_horas	VARCHAR()	NULL	Número de horas del curso
Costo_unam	MONEY	NOT NULL	Costo UNAM
Costo_institucion	MONEY	NOT NULL	Costo institución
Costo_particular	MONEY	NOT NULL	Costo particular
Objetivo	VARCHAR()	NOT NULL	Descripción del objetivo
Req_curso		NOT NULL	Identificador del curso
Ingr_min	MONEY	NULL	Ingreso mínimo para abrir el curso
Id_área	INTEGER	NOT NULL	Identificador del área a la que pertenece el curso
Id_tipo curso	INTEGER	NULL	Identificador del tipo del curso

CURSO_ESP

CAMPO	TIPO		OBSERVACIONES
Id_curso_esp	INTEGER	NOT NULL	Identificador del curso especial
Nom_curso_esp	VARCHAR()	NOT NULL	Nombre del curso especial, puede ser diplomado o una

			línea de especialización
--	--	--	--------------------------

CAT_MODULO

CAMPO	TIPO		OBSERVACIONES
Dqscs clave	CHAR	NOT NULL	Identificador de curso
Id modulo	INTEGER	NOT NULL	Identificador de modulo

CONCEPTO_RECIBO

CAMPO	TIPO		OBSERVACIONES
Id articulo	INTEGER	NOT NULL	Identificador del articulo
Id recibo	INTEGER	NOT NULL	Identificador del recibo
observaciones	VARCHAR(50)	NOT NULL	Observaciones

GRUPOS

CAMPO	TIPO		OBSERVACIONES
Id grupo	VARCHAR(10)	NOT NULL	Identificador de grupo
Dqscs clave	CHAR	NOT NULL	Identificador de curso
Fecha Inicio	DATE	NULL	Fecha de inicio del curso
Fecha fin	DATE	NULL	Fecha fin del curso
Num modulo	INTEGER	NULL	Numero de modulo
Id_eoa_curso	INTEGER	NOT NULL	Identificador del estado del curso
Id aula	INTEGER	NOT NULL	Identificador del aula
Id sede	INTEGER	NOT NULL	Identificador de la sede
Id_tipo_curso	INTEGER	NOT NULL	Identificador del tipo de curso
Id profesor	INTEGER	NOT NULL	Identificador del profesor

CURSO_SOFT

CAMPO	TIPO		OBSERVACIONES
Id soft	INTEGER	NOT NULL	Identificador del software
Dqscs clave	CHAR	NOT NULL	Identificador del curso

DEPARTAMENTOS

CAMPO	TIPO		OBSERVACIONES
Id_depto	INTEGER	NOT NULL	Identificador
Nom_depto	VARCHAR(50)	NOT NULL	Nombre del departamento
Funcion	VARCHAR(150)	NOT NULL	Descripción de las funciones

DEPENDENCIAS

CAMPO	TIPO		OBSERVACIONES
Id_Dependencia	INTEGER	NOT NULL	Identificador
nom_dependencia	VARCHAR(80)	NOT NULL	Nombre de la dependencias, únicamente de la UNAM
Siglas	VARCHAR(8)	NOT NULL	Siglas de la Dependencia
direccion	VARCHAR(50)	NOT NULL	Dirección de la dependencia
Teléfono	INTEGER	NULL	Teléfono de la dependencia
E_mail	ALFANUMERICO	NULL	Correo electrónico
responsable	VARCHAR(50)	NULL	Persona con la que nos podemos comunicar

DOCUMENTOS

CAMPO	TIPO		OBSERVACIONES
Id_doc	INTEGER	NOT NULL	Identificador
nom_doc	VARCHAR(80)	NOT NULL	Nombre del documento

ESTADO_GRUPO

CAMPO	TIPO		OBSERVACIONES
Id_edo_grupo	INTEGER	NOT NULL	Identificador
Nom_edo_grupo	VARCHAR(50)	NOT NULL	Nombre del estado

EDO_PROFESION

CAMPO	TIPO		OBSERVACIONES
Id_edo_profesion	INTEGER	NOT NULL	Identificador
Nom_edo_profesion	VARCHAR(60)	NOT NULL	Descripción

ESTADO_RECIBO

CAMPO	TIPO		OBSERVACIONES
Id_recibo	INTEGER	NOT NULL	Identificador
Nom_edo_recibo	VARCHAR(60)	NOT NULL	Nombre de estados de recibo

EGRESOS

CAMPO	TIPO		OBSERVACIONES
Id_egreso	INTEGER	NOT NULL	Identificador
Concepto	MONEY	NOT NULL	Es el concepto del egreso a una fecha
Fecha	INTEGER	NOT NULL	Es la fecha del egreso

ESC_FAC

CAMPO	TIPO		OBSERVACIONES
Id_esc_fac	INTEGER	NOT NULL	Identificador
Nom_esc_fac	VARCHAR()	NOT NULL	Nombre de la escuela o facultad

EXAMENES

CAMPO	TIPO		OBSERVACIONES
Id_Examen	INTEGER	NOT NULL	Identificador
Nombre	VARCHAR(80)	NOT NULL	Nombre del examen
Costo	INTEGER	NULL	Costo del examen

INSTITUCIONES

CAMPO	TIPO		OBSERVACIONES
Id_inst	INTEGER	NOT NULL	Identificador
Nom_institucion	VARCHAR(80)	NOT NULL	Es el nombre de la institución
Siglas	VARCHAR(8)	NOT NULL	Son las siglas de la institución
Responsable	VARCHAR(10)	NULL	Persona a la que le puedes pedir información
Calle	VARCHAR(20)	NOT NULL	Calle de la dirección de la institución
Colonia	VARCHAR(30)	NOT NULL	Es parte de la dirección

Delegación	VARCHAR(30)	NULL	Es parte de la dirección
C.P.	INTEGER	NOT NULL	Código de la zona postal
Teléfono	INTEGER	NOT NULL	Teléfono de la institución

INST_CURSO

CAMPO	TIPO		OBSERVACIONES
Id Inst	INTEGER	NOT NULL	Identificador institución
Id Curso	INTEGER	NOT NULL	Identificador curso

MEDIO_DIFUSION

CAMPO	TIPO		OBSERVACIONES
Id Medio	INTEGER	NOT NULL	Identificador
Nombre	VARCHAR(50)	NOT NULL	Nombre del tipo de medio de difusión

MODULOS

CAMPO	TIPO		OBSERVACIONES
Id Modulo	INTEGER	NOT NULL	Identificador
Nombre	VARCHAR(30)	NOT NULL	Nombre del módulo

MODULO_ESP

CAMPO	TIPO		OBSERVACIONES
Id Modulo	INTEGER	NOT NULL	Identificador de modulo
Id curso esp	VARCHAR(30)	NOT NULL	Identificador de curso especial

NIVELES

CAMPO	TIPO		OBSERVACIONES
Id nivel	INTEGER	NOT NULL	Identificador
Nom nivel	VARCHAR(30)	NOT NULL	Nombre del nivel

NIVEL_ESCOLAR

CAMPO	TIPO		OBSERVACIONES
Id nivel escolar	INTEGER	NOT NULL	Identificador
Nom nivel escolar	VARCHAR(30)	NOT NULL	Nombre del nivel

PALT_REND

CAMPO	TIPO		OBSERVACIONES
Id p a remnto	INTEGER	NOT NULL	Identificador
Nom_p	VARCHAR(30)	NOT NULL	Nombre del programa de alto rendimiento

PERSONAL

CAMPO	TIPO		OBSERVACIONES
Id_personal	INTEGER	NOT NULL	Identificador
Nom_personal	VARCCHAR(80)	NOT NULL	Nombre
A_paterno	VARCHAR(8)	NOT NULL	Apellido Paterno
A_materno	VARCHAR(10)	NOT NULL	Apellido Materno
Calle	VARCHAR(20)	NOT NULL	Es parte de la dirección
Colonia	VARCHAR(30)	NOT NULL	Es parte de la dirección
Delegación	VARCHAR(30)	NOT NULL	Es parte de la dirección
C_P	INTEGER	NOT NULL	Código de la zona postal
Te_casa	INTEGER	NULL	Teléfono de la casa
Tel_recado		NULL	Teléfono de recado
Horas_semana	INTEGER	NULL	Horas que labora a la semana
Ocupacion	VARCAHAR	NOT NULL	Ocupación de la persona
Contraseña	VARCHAR(10)	NOT NULL	Contraseña en el sistema
Lectura	BIT	NOT NULL	Permiso de lectura
Actualizacion	BIT	NOT NULL	Permiso de escritura
Borrado	BIT	NOT NULL	Permiso de borrado
Id_tipo_personal	INTEGER	NOT NULL	Identificador de tipo personal
Id_depto	INTEGER	NOT NULL	Identificador de departamento

PROFESORES

CAMPO	TIPO		OBSERVACIONES
Id_profesor	INTEGER	NOT NULL	Identificador
Nom_profesor	VARCHAR(25)	NOT NULL	Nombre del alumno
a_materno	VARCHAR(30)	NOT NULL	Apellido materno
a_paterno	VARCHAR(30)	NOT NULL	Apellido paterno
Edad	INTEGER	NOT NULL	Edad del profesor
Sexo	VARCHAR(1)	NOT NULL	Sexo del profesor
Nacionalidad	VARCHAR(30)	NOT NULL	Nacionalidad del profesor
Calle	VARCHAR(30)	NOT NULL	Es parte de la dirección del profesor
Colonia	VARCHAR(50)	NOT NULL	Es parte de la dirección del profesor
Delegación	VARCHAR(50)	NOT NULL	Es parte de la dirección del profesor
CP	INTEGER	NOT NULL	Código de la zona postal
Tel_oficina	INTEGER	NULL	Teléfono de la oficina
Tel_casa	INTEGER	NULL	Teléfono de la casa
E_mail	VARCHAR(10)	NULL	Correo electrónico
Rtc	VARCHAR(10)	NOT NULL	Es el rtc del profesor
Profesion	VARCHAR(50)	NULL	Es la profesión del profesor

PROG_CAT_CURSO

CAMPO	TIPO		OBSERVACIONES
Id_Programa	INTEGER	NOT NULL	Identificador de programa
Id_Curso	INTEGER	NOT NULL	Identificador de curso

PROGRAMAS

CAMPO	TIPO		OBSERVACIONES
Id_Programa	INTEGER	NOT NULL	Identificador
Nombre	VARCHAR(50)	NOT NULL	Nombre del programa de cómputo

PROPUESTAS

CAMPO	TIPO		OBSERVACIONES
Id_propuesta	INTEGER	NOT NULL	Identificador de la propuesta
Id_alumno	VARCHAR(50)	NOT NULL	Identificador alumno
fecha	DATE	NULL	Fecha de la propuesta
descripcion	VARCHAR(50)	NULL	

RECIBOS

CAMPO	TIPO		OBSERVACIONES
Id_recibo	INTEGER	NOT NULL	Identificador
Fecha	INTEGER	NOT NULL	Fecha en la que fue emitido el recibo
Id_edo_recibo	INTEGER	NOT NULL	Si se afecto o no
Observaciones	VARCHAR(40)	NULL	Observaciones
Hora	TIME	NULL	Hora en la que fue emitido el recibo

SEDES

CAMPO	TIPO		OBSERVACIONES
Id_Sede	INTEGER	NOT NULL	Identificador
Nom_sede	VARCHAR()	NOT NULL	Es el nombre de la sede
Calle	VARCHAR(30)	NOT NULL	Es parte de la dirección de la sede
Colonia	VARCHAR(30)	NOT NULL	Es parte de la dirección
Delegación	VARCHAR(30)	NULL	Es parte de la dirección
CP	INTEGER	NOT NULL	Código de la zona postal
Tel_sede	INTEGER	NULL	Es el teléfono de la zona
responsable	VARCHAR(80)	NULL	Persona con la que nos podemos comunicar

SOFTWARE

CAMPO	TIPO		OBSERVACIONES
Id_Soft	INTEGER	NOT NULL	Identificador
Nombre	VARCHAR(50)	NOT NULL	Es el nombre del software
Tipo	VARCHAR(50)	NOT NULL	Es el tipo de software
licencia	INTEGER	NULL	Es el número de la licencia del software

SOLICITUDES

CAMPO	TIPO		OBSERVACIONES
Id_solicitud	INTEGER	NOT NULL	Identificador
Nom_solicitud	VARCHAR(30)	NOT NULL	Nombre de la solicitud

SOLICITUD_DOC

CAMPO	TIPO		OBSERVACIONES
Id_solicitud	INTEGER	NOT NULL	Identificador de solicitud
Id_documento	VARCHAR(30)	NOT NULL	Identificador de documento

TIPO_CAT_CURSO

CAMPO	TIPO		OBSERVACIONES
Id_tp_cat_curso	INTEGER	NOT NULL	Identificador
Descripción	VARCHAR(30)	NOT NULL	Es el tipo de curso

TIPO_CURSO

CAMPO	TIPO		OBSERVACIONES
Id_Tip_Curso	INTEGER	NOT NULL	Identificador
Descripción	VARCHAR(30)	NOT NULL	Es el tipo de curso

TIPO_EGRESO

CAMPO	TIPO		OBSERVACIONES
Id_tip_egreso	INTEGER	NOT NULL	Identificador
Norm_tip_egreso	VARCHAR(50)	NOT NULL	Es la definición del tipo de egreso

TIPO_PAGO

CAMPO	TIPO		OBSERVACIONES
Id_pago	INTEGER	NOT NULL	Identificador de pago
Norm_tip_pago	VARCHAR(40)	NOT NULL	Descripción

TIPO_PERSONAL

CAMPO	TIPO		OBSERVACIONES
Id_tipo_personal	INTEGER	NOT NULL	Identificador
Norm_cve_personal	VARCHAR(50)	NOT NULL	Nombre del tipo de personal

TIPO_SOLICITUD

CAMPO	TIPO		OBSERVACIONES
Id_tipo_solicitud	INTEGER	NOT NULL	Identificador de tipo de solicitud
Norm_tipo_solicitud	VARCHAR(30)	NOT NULL	Nombre del tipo de solicitud

Descripción de las llaves foráneas en la base de datos:

ALUMNO_CURSO

IDENTIFICADORES	NOMBRE DE LAS FK	TABLA
Id_alumno	Fk_alcurso_alu	ALUMNOS
Id_curso	Fk_alcurso_cursos	CURSOS
Id_beca	Fk_alcurso_beca	BECAS

ALUMNOS

IDENTIFICADORES	NOMBRE DE LAS FK	TABLA
Id_escolar	Fk_alumnos_nivelescolar	NIVEL ESCOLAR
Id_tipo_personal	Fk_alumnos_tipopersonal	TIPO PERSONAL
Id_prog_alt_ren	Fk_alumnos_progaltrend	P ALT REND
Id_edo_profesion	Fk_alumnos_edoprofesion	EDO PROFESION
Id_esc_fac	Fk_alumnos_escfac	ESC FAC
Id_dep	Fk_alumnos_dependencias	DEPENDENCIAS

CAT_CURSO

IDENTIFICADORES	NOMBRE DE LAS FK	TABLA
Id_area	Fk_catcurso_area	AREAS
Id_tipo_curso	Fk_catcurso_tipocurso	TIPO CURSO

Apéndice B

CALIF_PREVIO

IDENTIFICADORES	NOMBRE DE LAS FK	TABLA
Id_alumno	Fk_califprevio_alumnos	ALUMNOS

CAT_MODULO

IDENTIFICADORES	NOMBRE DE LAS FK	TABLA
Dgsca_clave	Fk_catmodulo_catcurso	CAT_CURSO
Id_modulo	Fk_catmodulo_modulo	MODULOS

CONCEPTO_RECIBO

IDENTIFICADORES	NOMBRE DE LAS FK	TABLA
Id_recibo	Fk_conceptorecibo_recibo	RECIBOS
Id_articulo	Fk_conceptorecibo_articulos	ARTICULOS

CURSOS

IDENTIFICADORES	NOMBRE DE LAS FK	TABLA
Dgsca_clave	Fk_curso_catcurso	CAT_CURSO
Id_edo_grupo	Fk_curso_edocurso	ESTADO_GRUPO
Id_prog_ali_ten	Fk_curso_aula	AULAS
Id_edo_profesion	Fk_curso_sede	SEDES
Id_esc_fac	Fk_curso_tipocurso	TIPO_CURSO
Id_dep	Fk_curso_profesores	PROFESORES

CURSO_SOFTWARE

IDENTIFICADORES	NOMBRE DE LAS FK	TABLA
Id_soft	Fk_cursosoftware_software	SOFTWARE
Dgsca_clave	Fk_cursosoftware_catcurso	CAT_CURSO

MODULO_ESP

IDENTIFICADORES	NOMBRE DE LAS FK	TABLA
Id_modulo	Fk_moduloesp_modulo	MODULOS
Id_curso_esp	Fk_moduloesp_cursoesp	CURSO_ESP

PROG_CURSO

IDENTIFICADORES	NOMBRE DE LAS FK	TABLA
Dgsca_clave	Fk_progcurso_catcurso	CAT_CURSO
Id_programa	Fk_progcurso_programa	PROGRAMAS

SOLICITUD_DOC

IDENTIFICADORES	NOMBRE DE LAS FK	TABLA
Id_solicitud	Fk_solicitudoc_solicitud	SOLICITUD
Id_doc	Fk_solicitudoc_documentos	DOCUMENTOS

Apéndice C

```

CREATE TABLE [ALUMNO_GRUPO] (
    [id_alumno] [numeric](18, 0) NOT NULL ,
    [id_grupo] [char](15) NOT NULL ,
    [calificacion] [smallint] NULL ,
    [asistencia] [smallint] NULL ,
    [id_beca] [int] NULL ,
    [constancia] [bit] NOT NULL ,
    CONSTRAINT [PK_ALUMNO_CURSO] PRIMARY KEY NONCLUSTERED
    (
        [id_alumno],
        [id_grupo]
    ) ON [PRIMARY] ,
    CONSTRAINT [fk_alcurso_beca] FOREIGN KEY
    (
        [id_beca]
    ) REFERENCES [BECAS] (
        [id_beca]
    ),
    CONSTRAINT [fk_alcurso_cursos] FOREIGN KEY
    (
        [id_grupo]
    ) REFERENCES [GRUPOS] (
        [id_grupo]
    ),
    CONSTRAINT [fk_alcurso_alu] FOREIGN KEY
    (
        [id_alumno]
    ) REFERENCES [ALUMNOS] (
        [id_alumno]
    )
) ON [PRIMARY]
GO

```

```

CREATE TABLE [ALUMNOS] (
    [id_alumno] [numeric](18, 0) IDENTITY (1, 1) NOT NULL ,
    [rfc] [char](10) NOT NULL ,
    [no_cuenta] [char](10) NULL ,
    [nom_alumno] [char](20) NOT NULL ,
    [a_paterno] [char](15) NOT NULL ,
    [a_materno] [char](15) NOT NULL ,
    [calle] [char](40) NOT NULL ,
    [colonia] [char](30) NOT NULL ,
    [delegacion] [char](25) NULL ,
    [estado] [char](25) NULL ,
    [cp] [char](5) NOT NULL ,
    [tel_oficina] [char](8) NULL ,
    [tel_casa] [char](8) NULL ,
    [tel_recado] [char](8) NULL ,
    [email] [char](35) NULL ,
    [trabajo] [char](20) NULL ,
    [ocupacion] [char](20) NULL ,
    [puesto] [char](20) NULL ,
    [id_escolar] [smallint] NULL ,
    [id_tipo_personal] [smallint] NULL ,
    [id_prog_alto_rend] [smallint] NULL ,
    [id_edo_profesion] [smallint] NULL ,
    [id_esc_fac] [int] NULL ,
    [id_dep] [int] NULL ,
    [id_proc] [int] NULL ,
    CONSTRAINT [PK_ALUMNOS] PRIMARY KEY NONCLUSTERED
    (
        [id_alumno]
    ) ON [PRIMARY] ,
    CONSTRAINT [fk_alumnos_dependencias] FOREIGN KEY
    (
        [id_dep]
    ) REFERENCES [DEPENDENCIAS] (
        [id_dependencia]
    ),
    CONSTRAINT [fk_alumnos_edoprofesion] FOREIGN KEY

```

```

(
    [id_edo_profesion]
) REFERENCES [EDO_PROFESION] (
    [id_edo_profesion]
),
CONSTRAINT [fk_alumnos_nivelescolar] FOREIGN KEY
(
    [id_escolar]
) REFERENCES [NIVEL_ESCOLAR] (
    [id_nivel_escolar]
),
CONSTRAINT [FK_ALUMNOS_PROCEDENCIAS] FOREIGN KEY
(
    [id_proc]
) REFERENCES [PROCEDENCIAS] (
    [id_proc]
),
CONSTRAINT [fk_alumnos_progaltrend] FOREIGN KEY
(
    [id_prog_alto_rend]
) REFERENCES [P_ALT_REND] (
    [id_p_alt_rend]
),
CONSTRAINT [fk_alumnos_tipopersona] FOREIGN KEY
(
    [id_tipo_personal]
) REFERENCES [TIPO_PERSONAL] (
    [id_tipo_personal]
),
CONSTRAINT [kf_alumnos_escfac] FOREIGN KEY
(
    [id_esc_fac]
) REFERENCES [ESC_FAC] (
    [id_esc_fac]
)
) ON [PRIMARY]
GO

CREATE TABLE [AREAS] (
    [id_area] [smallint] NOT NULL ,
    [nom_area] [char] (50) NOT NULL ,
    CONSTRAINT [PK_AREA] PRIMARY KEY NONCLUSTERED
(
    [id_area]
) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [ARTICULOS] (
    [id_art] [int] NOT NULL ,
    [nom_art] [char] (45) NOT NULL ,
    [costo] [char] (10) NULL ,
    CONSTRAINT [PK_ARTICULOS] PRIMARY KEY NONCLUSTERED
(
    [id_art]
) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [AULAS] (
    [id_aula] [smallint] NOT NULL ,
    [nom_aula] [char] (20) NULL ,
    [num_max] [smallint] NOT NULL ,
    CONSTRAINT [PK_AULAS] PRIMARY KEY NONCLUSTERED
(
    [id_aula]
) ON [PRIMARY]
) ON [PRIMARY]
GO

```

Apéndice C

```
CREATE TABLE [BANCOS] (  
    [id_banco] [smallint] NOT NULL ,  
    [nom_banco] [char] (40) NOT NULL ,  
    CONSTRAINT [PK_BANCOS] PRIMARY KEY NONCLUSTERED  
    (  
        [id_banco]  
    ) ON [PRIMARY]  
) ON [PRIMARY]  
GO
```

```
CREATE TABLE [BECAS] (  
    [id_beca] [int] NOT NULL ,  
    [nom_beca] [varchar] (35) NOT NULL ,  
    [porc_desc] [smallint] NOT NULL ,  
    CONSTRAINT [PK_BECAS] PRIMARY KEY NONCLUSTERED  
    (  
        [id_beca]  
    ) ON [PRIMARY]  
) ON [PRIMARY]  
GO
```

```
CREATE TABLE [CALIF_PREVIO] (  
    [calificacion] [smallint] NOT NULL ,  
    [fecha] [datetime] NOT NULL ,  
    [id_alumno] [numeric](18, 0) NOT NULL ,  
    [descripcion] [varchar] (70) NULL ,  
    CONSTRAINT [fk_califprevio_alumnos] FOREIGN KEY  
    (  
        [id_alumno]  
    ) REFERENCES [ALUMNOS] (  
        [id_alumno]  
    )  
) ON [PRIMARY]  
GO
```

```
CREATE TABLE [CAT_CURSO] (  
    [dgsca_clave] [char] (10) NOT NULL ,  
    [nom_oficial] [varchar] (170) NOT NULL ,  
    [nom_comercial] [varchar] (170) NOT NULL ,  
    [no_horas] [smallint] NOT NULL ,  
    [costo_unam] [money] NOT NULL ,  
    [costo_institucion] [money] NOT NULL ,  
    [costo_particular] [money] NOT NULL ,  
    [objetivo] [varchar] (400) NULL ,  
    [observaciones] [varchar] (400) NULL ,  
    [edo_temario] [bit] NOT NULL ,  
    [ingr_min] [money] NULL ,  
    [id_area] [smallint] NOT NULL ,  
    [id_clasificacion] [smallint] NOT NULL ,  
    CONSTRAINT [PK_CAT_CURSO] PRIMARY KEY NONCLUSTERED  
    (  
        [dgsca_clave]  
    ) ON [PRIMARY] ,  
    CONSTRAINT [fk_cat_curso_clasificacion] FOREIGN KEY  
    (  
        [id_clasificacion]  
    ) REFERENCES [CLASIFICA_CURSO] (  
        [id_clasifica_curso]  
    ) ,  
    CONSTRAINT [fk_catcurso_area] FOREIGN KEY  
    (  
        [id_area]  
    ) REFERENCES [AREAS] (  
        [id_area]  
    )  
) ON [PRIMARY]  
GO
```

```
CREATE TABLE [CAT_MODULO] (  
    [dgsca_clave] [char] (10) NOT NULL ,
```

```

[id_modulo] [float] NOT NULL ,
CONSTRAINT [fk_catmodulo_catcurso] FOREIGN KEY
(
    [dgsca_clave]
) REFERENCES [CAT_CURSO] (
    [dgsca_clave]
),
CONSTRAINT [fk_catmodulo_modulo] FOREIGN KEY
(
    [id_modulo]
) REFERENCES [MODULOS] (
    [id_modulo]
)
) ON [PRIMARY]
GO

CREATE TABLE [CLASIFICA_CURSO] (
    [id_clasifica_curso] [smallint] NOT NULL ,
    [clasifica_curso] [char] (30) NOT NULL ,
    CONSTRAINT [PK_CLASIFICA_CURSO] PRIMARY KEY NONCLUSTERED
(
    [id_clasifica_curso]
) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [CONCEPTO_RECIBO] (
    [id_recibo] [float] NOT NULL ,
    [id_articulo] [int] NOT NULL ,
    [observaciones] [char] (35) NULL ,
    CONSTRAINT [PK_CONCEPTO_RECIBO] PRIMARY KEY NONCLUSTERED
(
    [id_recibo]
) ON [PRIMARY] ,
CONSTRAINT [fk_conceptorecibo_articulos] FOREIGN KEY
(
    [id_articulo]
) REFERENCES [ARTICULOS] (
    [id_art]
),
CONSTRAINT [fk_conceptorecibo_recibo] FOREIGN KEY
(
    [id_recibo]
) REFERENCES [RECIBOS] (
    [id_recibo]
)
) ON [PRIMARY]
GO

CREATE TABLE [CURSO_ESP] (
    [id_curso_esp] [float] NOT NULL ,
    [nom_curso_esp] [varchar] (85) NOT NULL ,
    [monto] [money] NULL ,
    CONSTRAINT [PK_CURSO_ESP] PRIMARY KEY NONCLUSTERED
(
    [id_curso_esp]
) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [CURSO_SOFTWARE] (
    [id_soft] [int] NOT NULL ,
    [dgsca_clave] [char] (10) NOT NULL ,
    CONSTRAINT [fk_cursosoftware_catcurso] FOREIGN KEY
(
    [dgsca_clave]
) REFERENCES [CAT_CURSO] (
    [dgsca_clave]
),
CONSTRAINT [fk_cursosoftware_software] FOREIGN KEY

```

Apéndice C

```
(
    [id_sof]
) REFERENCES [SOFTWARE] (
    [id_software]
)
) ON [PRIMARY]
GO

CREATE TABLE [CURSOS_DISP] (
    [id_profesor] [int] NULL ,
    [dgsea_clave] [char] (10) NULL
) ON [PRIMARY]
GO

CREATE TABLE [DEPARTAMENTOS] (
    [id_depto] [smallint] NOT NULL ,
    [nom_depto] [char] (40) NOT NULL ,
    [funcion] [char] (50) NOT NULL ,
    CONSTRAINT [PK_DEPARTAMENTO] PRIMARY KEY NONCLUSTERED
(
    [id_depto]
) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [DEPENDENCIAS] (
    [id_dependencia] [int] NOT NULL ,
    [nom_dependencia] [varchar] (70) NOT NULL ,
    [siglas] [char] (10) NOT NULL ,
    [calle] [char] (30) NOT NULL ,
    [colonia] [char] (30) NOT NULL ,
    [delegacion] [char] (30) NOT NULL ,
    [cp] [char] (5) NOT NULL ,
    [tel1_dep] [char] (8) NULL ,
    [tel2_dep] [char] (8) NULL ,
    [email] [char] (30) NULL ,
    [contacto] [char] (40) NULL ,
    CONSTRAINT [PK_DEPENDENCIAS] PRIMARY KEY NONCLUSTERED
(
    [id_dependencia]
) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [DOCUMENTOS] (
    [id_doc] [smallint] NOT NULL ,
    [nom_doc] [varchar] (65) NOT NULL ,
    CONSTRAINT [PK_DOCUMENTOS] PRIMARY KEY NONCLUSTERED
(
    [id_doc]
) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dtproperties] (
    [id] [int] IDENTITY (1, 1) NOT NULL ,
    [objectid] [int] NULL ,
    [property] [varchar] (64) NOT NULL ,
    [value] [varchar] (255) NULL ,
    [image] [image] NULL ,
    [version] [int] NOT NULL CONSTRAINT [DF_dtpropert_verst_0EA330E9] DEFAULT (0),
    CONSTRAINT [pk_dtproperties] PRIMARY KEY CLUSTERED
(
    [id],
    [property]
) ON [PRIMARY]
)
GO

CREATE TABLE [EDO_PROFESION] (
```

```

[id_edo_profesion] [smallint] NOT NULL ,
[nom_profesion] [char] (30) NOT NULL ,
CONSTRAINT [PK_EDO_PROFESION] PRIMARY KEY NONCLUSTERED
(
    [id_edo_profesion]
) ON [PRIMARY]
) ON [PRIMARY]
GO

```

```

CREATE TABLE [EGRESOS] (
    [id_egreso] [int] NOT NULL ,
    [costo] [float] NOT NULL ,
    [fecha] [datetime] NOT NULL ,
    [id_tipo_egreso] [smallint] NOT NULL ,
    CONSTRAINT [PK_EGRESOS] PRIMARY KEY NONCLUSTERED
    (
        [id_egreso]
    ) ON [PRIMARY] ,
    CONSTRAINT [fk_egresos_tipo_egresos] FOREIGN KEY
    (
        [id_tipo_egreso]
    ) REFERENCES [TIPO_EGRESO] (
        [id_tipo_egreso]
    )
) ON [PRIMARY]
GO

```

```

CREATE TABLE [ESC_FAC] (
    [id_esc_fac] [int] NOT NULL ,
    [esc_fac] [varchar] (65) NOT NULL ,
    CONSTRAINT [PK_ESC_FAC] PRIMARY KEY NONCLUSTERED
    (
        [id_esc_fac]
    ) ON [PRIMARY]
) ON [PRIMARY]
GO

```

```

CREATE TABLE [ESTADO_GRUPO] (
    [id_edo_grupo] [int] NOT NULL ,
    [nom_edo_grupo] [char] (10) NOT NULL ,
    [observaciones] [char] (30) NULL ,
    CONSTRAINT [PK_ESTADO_CURSO] PRIMARY KEY NONCLUSTERED
    (
        [id_edo_grupo]
    ) ON [PRIMARY]
) ON [PRIMARY]
GO

```

```

CREATE TABLE [ESTADO_RECIBO] (
    [id_edo_recibo] [smallint] NOT NULL ,
    [nom_edo_recibo] [char] (30) NOT NULL ,
    [observaciones] [char] (40) NULL ,
    CONSTRAINT [PK_ESTADO_RECIBO] PRIMARY KEY NONCLUSTERED
    (
        [id_edo_recibo]
    ) ON [PRIMARY]
) ON [PRIMARY]
GO

```

```

CREATE TABLE [EXAMENES] (
    [id_examen] [varchar] (10) NOT NULL ,
    [nom_examen] [varchar] (50) NOT NULL ,
    [costo] [money] NOT NULL ,
    CONSTRAINT [PK_EXAMENES] PRIMARY KEY NONCLUSTERED
    (
        [id_examen]
    ) ON [PRIMARY]
) ON [PRIMARY]
GO

```

```

CREATE TABLE [GRUPOS] (
    [id_grupo] [char] (15) NOT NULL ,
    [fecha_inicio] [datetime] NOT NULL ,
    [fecha_fin] [datetime] NOT NULL ,
    [h_inicio] [char] (10) NULL ,
    [h_fin] [char] (10) NULL ,
    [id_edo_grupo] [int] NOT NULL ,
    [dgsca_clave] [char] (10) NOT NULL ,
    [id_aula] [smallint] NULL ,
    [id_sede] [smallint] NULL ,
    [id_tipo_grupo] [smallint] NOT NULL ,
    [id_profesor] [int] NULL ,
    [max_cupo] [int] NULL ,
    CONSTRAINT [PK_CURSOS] PRIMARY KEY NONCLUSTERED
    (
        [id_grupo]
    ) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [INST_CURSO] (
    [id_institucion] [int] NOT NULL ,
    [id_curso] [char] (15) NOT NULL ,
    CONSTRAINT [PK_INST_CURSO] PRIMARY KEY NONCLUSTERED
    (
        [id_institucion],
        [id_curso]
    ) ON [PRIMARY]
    CONSTRAINT [fk_inst_curso_cursos] FOREIGN KEY
    (
        [id_curso]
    ) REFERENCES [GRUPOS] (
        [id_grupo]
    ),
    CONSTRAINT [fk_inst_curso_institucion] FOREIGN KEY
    (
        [id_institucion]
    ) REFERENCES [INSTITUCIONES] (
        [id_institucion]
    )
) ON [PRIMARY]
GO

CREATE TABLE [INSTITUCIONES] (
    [id_institucion] [int] NOT NULL ,
    [nom_institucion] [varchar] (70) NOT NULL ,
    [siglas] [char] (10) NOT NULL ,
    [calle] [char] (30) NOT NULL ,
    [colonia] [char] (30) NOT NULL ,
    [delegacion] [char] (30) NOT NULL ,
    [cp] [char] (10) NOT NULL ,
    [estado] [char] (30) NOT NULL ,
    [tel_1inst] [char] (10) NULL ,
    [tel_2inst] [char] (10) NULL ,
    [email] [char] (20) NULL ,
    [contacto] [char] (40) NOT NULL ,
    CONSTRAINT [PK_INSTITUCIONES] PRIMARY KEY NONCLUSTERED
    (
        [id_institucion]
    ) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [MEDIO_DIFUSION] (
    [id_medio_difusion] [smallint] NOT NULL ,
    [nom_medio_difusion] [char] (30) NOT NULL ,
    CONSTRAINT [PK_MEDIO_DIFUSION] PRIMARY KEY NONCLUSTERED
    (
        [id_medio_difusion]
    ) ON [PRIMARY]
)

```

```

) ON (PRIMARY)
GO

CREATE TABLE [MODULO_ESP] (
    [id_modulo] [float] NULL,
    [id_curso_esp] [float] NULL,
    CONSTRAINT [fk_moduloesp_cursoesp] FOREIGN KEY
    (
        [id_curso_esp]
    ) REFERENCES [CURSO_ESP] (
        [id_curso_esp]
    ),
    CONSTRAINT [fk_moduloesp_modulo] FOREIGN KEY
    (
        [id_modulo]
    ) REFERENCES [MODULOS] (
        [id_modulo]
    )
) ON (PRIMARY)
GO

CREATE TABLE [MODULOS] (
    [id_modulo] [float] NOT NULL,
    [nom_modulo] [char] (30) NOT NULL,
    CONSTRAINT [PK_MODULOS] PRIMARY KEY NONCLUSTERED
    (
        [id_modulo]
    ) ON (PRIMARY)
) ON (PRIMARY)
GO

CREATE TABLE [NIVEL_ESCOLAR] (
    [id_nivel_escolar] [smallint] NOT NULL,
    [nom_nivel_escolar] [char] (30) NOT NULL,
    CONSTRAINT [PK_NIVEL_ESCOLAR] PRIMARY KEY NONCLUSTERED
    (
        [id_nivel_escolar]
    ) ON (PRIMARY)
) ON (PRIMARY)
GO

CREATE TABLE [NIVELES] (
    [id_nivel] [smallint] NOT NULL,
    [nom_nivel] [char] (20) NOT NULL,
    CONSTRAINT [PK_NIVELES] PRIMARY KEY NONCLUSTERED
    (
        [id_nivel]
    ) ON (PRIMARY)
) ON (PRIMARY)
GO

CREATE TABLE [P_ALT_REND] (
    [id_p_alt_rend] [smallint] NOT NULL,
    [p_alt_rend] [char] (35) NOT NULL,
    [id_nivel] [smallint] NOT NULL,
    CONSTRAINT [PK_P_ALT_REND] PRIMARY KEY NONCLUSTERED
    (
        [id_p_alt_rend]
    ) ON (PRIMARY),
    CONSTRAINT [fk_p_alt_rend_niveles] FOREIGN KEY
    (
        [id_nivel]
    ) REFERENCES [NIVELES] (
        [id_nivel]
    )
) ON (PRIMARY)
GO

CREATE TABLE [pbeatcol] (
    [pbc_tnam] [char] (30) NULL,

```

```

[psc_tid] [int] NULL ,
[psc_owner] [char] (30) NULL ,
[psc_cnam] [char] (30) NULL ,
[psc_cid] [smallint] NULL ,
[psc_labf] [varchar] (254) NULL ,
[psc_lpos] [smallint] NULL ,
[psc_hdr] [varchar] (254) NULL ,
[psc_hpos] [smallint] NULL ,
[psc_jtly] [smallint] NULL ,
[psc_mask] [varchar] (31) NULL ,
[psc_case] [smallint] NULL ,
[psc_hght] [smallint] NULL ,
[psc_wdth] [smallint] NULL ,
[psc_ptrn] [varchar] (31) NULL ,
[psc_bmap] [char] (1) NULL ,
[psc_int] [varchar] (254) NULL ,
[psc_cmnt] [varchar] (254) NULL ,
[psc_edit] [varchar] (31) NULL ,
[psc_tag] [varchar] (254) NULL
) ON (PRIMARY)
GO

CREATE TABLE [pscatedt] (
[psc_name] [varchar] (30) NOT NULL ,
[psc_edit] [varchar] (254) NULL ,
[psc_type] [smallint] NOT NULL ,
[psc_cnt] [int] NULL ,
[psc_seqn] [smallint] NOT NULL ,
[psc_flag] [int] NULL ,
[psc_work] [char] (32) NULL
)
GO

CREATE TABLE [pscatfmt] (
[psbf_name] [varchar] (30) NOT NULL ,
[psbf_fmft] [varchar] (254) NOT NULL ,
[psbf_type] [smallint] NOT NULL ,
[psbf_cnt] [int] NULL
)
GO

CREATE TABLE [pscattbl] (
[psb_tnam] [char] (30) NULL ,
[psb_tid] [int] NULL ,
[psb_owner] [char] (30) NULL ,
[psb_flg] [smallint] NULL ,
[psb_fwgt] [smallint] NULL ,
[psb_fit] [char] (1) NULL ,
[psb_fun] [char] (1) NULL ,
[psb_fchr] [smallint] NULL ,
[psb_ftpc] [smallint] NULL ,
[psb_ffce] [char] (32) NULL ,
[psb_fhgt] [smallint] NULL ,
[psb_fwgt] [smallint] NULL ,
[psb_fit] [char] (1) NULL ,
[psb_fun] [char] (1) NULL ,
[psb_fchr] [smallint] NULL ,
[psb_ftpc] [smallint] NULL ,
[psb_ffce] [char] (32) NULL ,
[psb_fhgt] [smallint] NULL ,
[psb_fwgt] [smallint] NULL ,
[psb_fit] [char] (1) NULL ,
[psb_fun] [char] (1) NULL ,
[psb_fchr] [smallint] NULL ,
[psb_ftpc] [smallint] NULL ,
[psb_ffce] [char] (32) NULL ,
[psb_cmnt] [varchar] (254) NULL
) ON (PRIMARY)
GO

```

```

CREATE TABLE [pbcatvld] (
    [pbv_nombre] [varchar] (30) NOT NULL ,
    [pbv_vald] [varchar] (254) NOT NULL ,
    [pbv_tipo] [smallint] NOT NULL ,
    [pbv_ctr] [int] NULL ,
    [pbv_msg] [varchar] (254) NULL
)
GO

CREATE TABLE [PERSONAL] (
    [id_personal] [int] IDENTITY (1, 1) NOT NULL ,
    [nom_personal] [char] (25) NOT NULL ,
    [a_paterno] [char] (15) NOT NULL ,
    [a_materno] [char] (15) NOT NULL ,
    [calle] [char] (40) NOT NULL ,
    [colonia] [char] (30) NOT NULL ,
    [delegacion] [char] (25) NOT NULL ,
    [estado] [char] (30) NOT NULL ,
    [cp] [char] (5) NOT NULL ,
    [horas_semana] [smallint] NOT NULL ,
    [tel_casa] [char] (10) NULL ,
    [tel_recado] [char] (10) NULL ,
    [e_mail] [char] (30) NULL ,
    [ocupacion] [char] (50) NULL ,
    [contrasena] [char] (8) NULL ,
    [lectura] [bit] NOT NULL ,
    [actualizacion] [bit] NOT NULL ,
    [insercion] [bit] NOT NULL ,
    [borrado] [bit] NOT NULL ,
    [id_depto] [smallint] NOT NULL ,
    [id_tipo_personal] [smallint] NOT NULL ,
    CONSTRAINT [PK_PERSONAL] PRIMARY KEY NONCLUSTERED
    (
        [id_personal]
    ) ON [PRIMARY] ,
    CONSTRAINT [fk_per_dep] FOREIGN KEY
    (
        [id_depto]
    ) REFERENCES [DEPARTAMENTOS] (
        [id_depto]
    ) ,
    CONSTRAINT [fk_per_tipo_per] FOREIGN KEY
    (
        [id_tipo_personal]
    ) REFERENCES [TIPO_PERSONAL] (
        [id_tipo_personal]
    )
) ON [PRIMARY]
GO

CREATE TABLE [PROCEDENCIAS] (
    [id_proc] [int] NOT NULL ,
    [procedencia] [varchar] (50) NOT NULL ,
    CONSTRAINT [PK_PROCEDENCIAS] PRIMARY KEY NONCLUSTERED
    (
        [id_proc]
    ) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [PROFESORES] (
    [id_profesor] [int] NOT NULL ,
    [rfc] [char] (14) NOT NULL ,
    [nom_profesor] [char] (35) NOT NULL ,
    [a_paterno] [char] (30) NOT NULL ,
    [a_materno] [char] (30) NOT NULL ,
    [calle] [char] (40) NOT NULL ,
    [colonia] [char] (30) NOT NULL ,
    [delegacion] [char] (30) NOT NULL ,
    [estado] [char] (30) NOT NULL ,

```

```

[cp] [char] (5) NOT NULL ,
[tel_ofna] [char] (10) NULL ,
[tel_casa] [char] (10) NULL ,
[tel_cel] [char] (15) NULL ,
[localizador] [char] (10) NULL ,
[cve_localizador] [char] (10) NULL ,
[email] [char] (30) NULL ,
[profesion] [char] (40) NULL ,
[edo_profesion] [smallint] NOT NULL ,
[ cursos_impactic] [char] (100) NULL ,
[certificaciones] [char] (150) NULL ,
[actividad_actual] [char] (50) NULL ,
[horario_disponible] [char] (100) NULL ,
[adscricion] [char] (100) NULL ,
[actualizado_a] [datetime] NULL ,
CONSTRAINT [PK_PROFESOR] PRIMARY KEY NONCLUSTERED
(
    [id_profesor]
) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [PROG_CURSO] (
    [id_prog_curso] [int] NOT NULL ,
    [dgscs_clave] [char] (10) NOT NULL ,
    [id_programa] [int] NOT NULL ,
    CONSTRAINT [PK_PROG_CURSO] PRIMARY KEY NONCLUSTERED
    (
        [id_prog_curso]
    ) ON [PRIMARY] ,
    CONSTRAINT [fk_progcurso_catcurso] FOREIGN KEY
    (
        [dgscs_clave]
    ) REFERENCES [CAT_CURSO] (
        [dgscs_clave]
    ) ,
    CONSTRAINT [fk_progcurso_curso] FOREIGN KEY
    (
        [id_programa]
    ) REFERENCES [PROGRAMAS] (
        [id_programa]
    )
) ON [PRIMARY]
GO

CREATE TABLE [PROGRAMAS] (
    [id_programa] [int] NOT NULL ,
    [nom_programa] [char] (40) NOT NULL ,
    CONSTRAINT [PK_PROGRAMAS] PRIMARY KEY NONCLUSTERED
    (
        [id_programa]
    ) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [PROPUESTAS] (
    [id_propuesta] [int] NOT NULL ,
    [id_alumno] [numeric] (18, 0) NOT NULL ,
    [descripcion] [char] (40) NOT NULL ,
    [fecha] [datetime] NOT NULL ,
    CONSTRAINT [PK_PROPUESTA] PRIMARY KEY NONCLUSTERED
    (
        [id_propuesta]
    ) ON [PRIMARY] ,
    CONSTRAINT [fk_propuestas_alumnos] FOREIGN KEY
    (
        [id_alumno]
    ) REFERENCES [ALUMNOS] (
        [id_alumno]
    )
)

```

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [RECIBOS] (
    [id_recibo] [float] NOT NULL ,
    [fecha] [datetime] NOT NULL ,
    [monto] [money] NULL ,
    [observaciones] [char] (40) NOT NULL ,
    [id_edo_recibo] [smallint] NOT NULL ,
    [id_tipo_pago] [smallint] NOT NULL ,
    [id_banco] [smallint] NULL ,
    CONSTRAINT [PK_RECIBOS] PRIMARY KEY NONCLUSTERED
    (
        [id_recibo]
    ) ON [PRIMARY] ,
    CONSTRAINT [fk_recibos_bancos] FOREIGN KEY
    (
        [id_banco]
    ) REFERENCES [BANCOS] (
        [id_banco]
    ),
    CONSTRAINT [fk_recibos_estado_recibos] FOREIGN KEY
    (
        [id_edo_recibo]
    ) REFERENCES [ESTADO_RECIBO] (
        [id_edo_recibo]
    ),
    CONSTRAINT [fk_recibos_tipo_pago] FOREIGN KEY
    (
        [id_tipo_pago]
    ) REFERENCES [TIPO_PAGO] (
        [id_tipo_pago]
    )
)
```

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [SEDES] (
    [id_sede] [smallint] NOT NULL ,
    [nom_sede] [char] (40) NOT NULL ,
    [calle] [char] (30) NOT NULL ,
    [colonia] [char] (30) NOT NULL ,
    [delegacion] [char] (30) NOT NULL ,
    [estado] [char] (30) NOT NULL ,
    [cp] [char] (5) NOT NULL ,
    [tel_1] [char] (10) NULL ,
    [tel_2] [char] (10) NULL ,
    [email] [char] (40) NULL ,
    [contacto] [char] (40) NOT NULL ,
    CONSTRAINT [PK_SEDES] PRIMARY KEY NONCLUSTERED
    (
        [id_sede]
    )
)
```

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [SOFTWARE] (
    [id_software] [int] NOT NULL ,
    [nom_software] [char] (40) NOT NULL ,
    [licencia] [char] (40) NOT NULL ,
    CONSTRAINT [PK_SOFTWARE] PRIMARY KEY NONCLUSTERED
    (
        [id_software]
    )
)
```

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [SOLICITUD] (
    [id_solicitud] [float] NOT NULL ,
    [fecha] [datetime] NOT NULL ,
    [dgsca_clave] [char] (10) NOT NULL ,
)
```

```

[id_tipo_solicitud] [smallint] NOT NULL ,
[id_recibo] [float] NOT NULL ,
[id_medio_difusion] [smallint] NOT NULL ,
[id_alumno] [numeric](18, 0) NOT NULL ,
[id_grupo] [char] (15) NULL ,
[id_examen] [varchar] (10) NULL
CONSTRAINT [PK_SOLICITUD] PRIMARY KEY NONCLUSTERED
(
    [id_solicitud]
) ON [PRIMARY] .
CONSTRAINT [fk_solicitud_alumo_curso] FOREIGN KEY
(
    [id_alumno],
    [id_grupo]
) REFERENCES [ALUMNO_GRUPO] (
    [id_alumno],
    [id_grupo]
),
CONSTRAINT [fk_solicitud_medio_difusion] FOREIGN KEY
(
    [id_medio_difusion]
) REFERENCES [MEDIO_DIFUSION] (
    [id_medio_difusion]
),
CONSTRAINT [fk_solicitud_recibo] FOREIGN KEY
(
    [id_recibo]
) REFERENCES [RECIBOS] (
    [id_recibo]
),
CONSTRAINT [fk_solicitud_tipo_solicitud] FOREIGN KEY
(
    [id_tipo_solicitud]
) REFERENCES [TIPO_SOLICITUD] (
    [id_tipo_solicitud]
),
CONSTRAINT [fki_solicitud_examenes] FOREIGN KEY
(
    [id_examen]
) REFERENCES [EXAMENES] (
    [id_examen]
)
) ON [PRIMARY]
GO

CREATE TABLE [SOLICITUD_DOC] (
    [id_solicitud] [float] NOT NULL ,
    [id_doc] [smallint] NOT NULL ,
    CONSTRAINT [fk_solicitudoc_documentos] FOREIGN KEY
    (
        [id_doc]
    ) REFERENCES [DOCUMENTOS] (
        [id_doc]
    ),
    CONSTRAINT [fk_solicitudoc_solicitud] FOREIGN KEY
    (
        [id_solicitud]
    ) REFERENCES [SOLICITUD] (
        [id_solicitud]
    )
) ON [PRIMARY]
GO

CREATE TABLE [sysallocations] (
    [objid] [int] NOT NULL ,
    [idid] [smallint] NOT NULL ,
    [fileid] [smallint] NOT NULL ,
    [gaminterval] [int] NOT NULL ,
    [iampage] [binary] (6) NOT NULL
)

```

GO

```
CREATE TABLE [syscolumns] (
    [name] [sysname] NOT NULL ,
    [id] [int] NOT NULL ,
    [xtype] [tinyint] NOT NULL ,
    [typestat] [tinyint] NOT NULL ,
    [xusertype] [smallint] NOT NULL ,
    [length] [smallint] NOT NULL ,
    [xprec] [tinyint] NOT NULL ,
    [xscale] [tinyint] NOT NULL ,
    [colid] [smallint] NOT NULL ,
    [xoffset] [smallint] NOT NULL ,
    [bitpos] [tinyint] NOT NULL ,
    [reserved] [tinyint] NOT NULL ,
    [colstat] [smallint] NOT NULL ,
    [cdefault] [int] NOT NULL ,
    [domain] [int] NOT NULL ,
    [number] [smallint] NOT NULL ,
    [colorder] [smallint] NOT NULL ,
    [autovail] [varbinary] (8000) NULL ,
    [offset] [smallint] NOT NULL ,
    [status] AS (convert(tinyint,(((bitpos) & 7) + case when ((typestat) & 1 = 0) then 8 else 0 end + case when
    (((typestat) & 2 <> 0 or (type_name(xtype) = 'image' or type_name(xtype) = 'text') and (colstat) & 0x1000 <> 0)) then 16
    else 0 end + case when (((type_name(xtype) = 'image' or type_name(xtype) = 'text') and (colstat) & 0x2000 <> 0 or
    (type_name(xtype) = 'timestamp' or type_name(xtype) = 'char' or type_name(xtype) = 'binary') and (typestat) & 1 = 0))
    then 32 else 0 end + case when ((colstat) & 4 <> 0) then 64 else 0 end + case when ((colstat) & 1 <> 0) then 128 else 0
    end))) ,
    [type] AS (convert(tinyint,typeotids(xtype),(1 - ((typestat) & 1)))) ,
    [usertype] AS (convert(smallint,columnproperty([id],[name],oldusertype))) ,
    [printfmt] AS (convert(varchar(255),autovail)) ,
    [prec] AS (convert(smallint,case when ((type_name(xtype)) = 'ntext' or type_name(xtype) = 'image' or
    type_name(xtype) = 'text') then null when (type_name(xtype) = 'uniqueidentifier') then [xprec] else
    (odbcprec(xtype),[length],[xprec]) end)) ,
    [scale] AS (odbcscale(xtype,[xscale])) ,
    [iscomputed] AS (convert(int,sign(((colstat) & 4)))) ,
    [isoutparam] AS (convert(int,sign(((colstat) & 4)))) ,
    [isnullable] AS (convert(int,(1 - ((typestat) & 1))))
)
GO
```

```
CREATE TABLE [syscomments] (
    [id] [int] NOT NULL ,
    [number] [smallint] NOT NULL ,
    [colid] [smallint] NOT NULL ,
    [status] [smallint] NOT NULL ,
    [cxtxt] [varbinary] (8000) NOT NULL ,
    [language] AS (convert(smallint,(2 + 4 * ((status) & 1)))) ,
    [language] AS (convert(smallint,0)) ,
    [encrypted] AS (convert(bit,(status) & 1)) ,
    [compressed] AS (convert(bit,(status) & 2)) ,
    [text] AS (convert(nvarchar(4000),case when ((status) & 2 = 2) then (uncompress(cxtxt)) else (cxtxt) end))
)
GO
```

```
CREATE TABLE [sysdepends] (
    [id] [int] NOT NULL ,
    [depid] [int] NOT NULL ,
    [number] [smallint] NOT NULL ,
    [depnumber] [smallint] NOT NULL ,
    [status] [smallint] NOT NULL ,
    [depdbid] AS (convert(smallint,0)) ,
    [depsiteid] AS (convert(smallint,0)) ,
    [selali] AS (convert(bit,(status) & 2)) ,
    [resultobj] AS (convert(bit,(status) & 4)) ,
    [readobj] AS (convert(bit,(status) & 8))
)
GO
```

```
CREATE TABLE [sysfilegroups] (
```

Apéndice C

```
[groupid] [smallint] NOT NULL ,
[allocpolicy] [smallint] NOT NULL ,
[status] [int] NOT NULL ,
[groupname] [sysname] NOT NULL
)
GO
```

```
CREATE TABLE [sysfiles] (
    [fileid] [smallint] NOT NULL ,
    [groupid] [smallint] NOT NULL ,
    [size] [int] NOT NULL ,
    [maxsize] [int] NOT NULL ,
    [growth] [int] NOT NULL ,
    [status] [int] NOT NULL ,
    [perf] [int] NOT NULL ,
    [name] [nchar] (128) NOT NULL ,
    [filename] [nchar] (260) NOT NULL
)
GO
```

```
CREATE TABLE [sysfiles1] (
    [status] [int] NOT NULL ,
    [fileid] [smallint] NOT NULL ,
    [name] [nchar] (128) NOT NULL ,
    [filename] [nchar] (260) NOT NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [sysforeignkeys] (
    [constid] [int] NOT NULL ,
    [fkeyid] [int] NOT NULL ,
    [rkeyid] [int] NOT NULL ,
    [fkey] [smallint] NOT NULL ,
    [rkey] [smallint] NOT NULL ,
    [keyno] [smallint] NOT NULL
)
GO
```

```
CREATE TABLE [sysfulltextcatalogs] (
    [ftcatid] [smallint] NOT NULL ,
    [name] [sysname] NOT NULL ,
    [status] [smallint] NOT NULL ,
    [path] [nvarchar] (260) NULL
)
GO
```

```
CREATE TABLE [sysindexes] (
    [id] [int] NOT NULL ,
    [status] [int] NOT NULL ,
    [first] [binary] (6) NOT NULL ,
    [indid] [smallint] NOT NULL ,
    [root] [binary] (6) NOT NULL ,
    [minlen] [smallint] NOT NULL ,
    [keycnt] [smallint] NOT NULL ,
    [groupid] [smallint] NOT NULL ,
    [dpages] [int] NOT NULL ,
    [reserved] [int] NOT NULL ,
    [used] [int] NOT NULL ,
    [rowcnt] [binary] (8) NOT NULL ,
    [rowmodctr] [int] NOT NULL ,
    [soid] [tinyint] NOT NULL ,
    [csid] [tinyint] NOT NULL ,
    [xmaxlen] [smallint] NOT NULL ,
    [maxirow] [smallint] NOT NULL ,
    [OrigFillFactor] [tinyint] NOT NULL ,
    [StatVersion] [tinyint] NOT NULL ,
    [reserved2] [int] NOT NULL ,
    [FirstAM] [binary] (6) NOT NULL ,
    [impid] [smallint] NOT NULL ,
    [lockflags] [smallint] NOT NULL ,

```

```

[pgmodctr] [int] NOT NULL ,
[keys] [varbinary] (816) NULL ,
[name] [sysname] NOT NULL ,
[statblob] [image] NULL ,
[maxlen] AS (8000) ,
[rows] AS (case when (convert(int,[rowcnt]) <> 0) then 2147483647 else
(convert(int,convert(binary(8),reverse(convert(char(8),[rowcnt]))))) end)
)
GO

CREATE TABLE [sysindexkeys] (
[id] [int] NOT NULL ,
[indid] [smallint] NOT NULL ,
[coleid] [smallint] NOT NULL ,
[keyno] [smallint] NOT NULL
)
GO

CREATE TABLE [sysmembers] (
[memberuid] [smallint] NOT NULL ,
[groupuid] [smallint] NOT NULL
)
GO

CREATE TABLE [sysobjects] (
[name] [sysname] NOT NULL ,
[id] [int] NOT NULL ,
[xtype] [char] (2) NOT NULL ,
[uid] [smallint] NOT NULL ,
[info] [smallint] NOT NULL ,
[status] [int] NOT NULL ,
[base_schema_ver] [int] NOT NULL ,
[replinfo] [int] NOT NULL ,
[parent_obj] [int] NOT NULL ,
[crdate] [datetime] NOT NULL ,
[ftcatid] [smallint] NOT NULL ,
[schema_ver] AS ((base_schema_ver) & ((~15))),
[stats_schema_ver] AS ((base_schema_ver) & 15) ,
[type] AS (convert(char(2),case when (([xtype] = 'UQ' or [xtype] = 'PK') then 'K' else [xtype] end) ,
[userstat] AS (convert(smallint,case when ([xtype] = 'S' or [xtype] = 'U') then 1 else 0 end) ,
[sysstat] AS (convert(smallint,case [xtype] when 'S' then 1 when 'V' then 2 when 'U' then 3 when 'P' then 4 when
'RF' then 4 when 'X' then 4 when 'L' then 5 when 'D' then 6 when 'R' then 7 when 'TR' then 8 when 'PK' then 9 when 'UQ'
then 9 when 'C' then 10 when 'F' then 11 when 'AP' then 13 else 0 end + case when ([xtype] = 'S' or [xtype] = 'U') then
(case when ([status] & 1 <> 0) then 16 else 0 end + case when ([status] & 2 <> 0) then 32 else 0 end + 64 + case when
(substring([name],1,1) = 'P') then 256 else 0 end + case when ([status] & 0x01000000 <> 0) then 512 else 0 end + case
when ([status] & 0x200000 <> 0) then 1024 else 0 end + case when ([status] & 0xc4000000 <> 0) then 2048 else 0 end +
case when ([status] & 4 <> 0) then 8192 else 0 end + case when ([status] & 0x020c0000 <> 0) then 16384 else 0 end + case
when (substring([name],1,2) = '##') then (-32768) else 0 end) else 0 end)) ,
[indexdef] AS (convert(smallint,(((base_schema_ver) & ((~15))) / 65535)) ,
[refdate] AS (convert(datetime,[crdate])) ,
[version] AS (convert(int,0)) ,
[deltrig] AS (convert(int,case when ([xtype] = 'S' or [xtype] = 'U') then (objectproperty([id], 'TableDeleteTrigger'))
when ([xtype] = 'TR') then [parent_obj] else 0 end) ,
[instrig] AS (convert(int,case when ([xtype] = 'S' or [xtype] = 'U') then (objectproperty([id], 'TableInsertTrigger'))
else 0 end) ,
[updtrig] AS (convert(int,case when ([xtype] = 'S' or [xtype] = 'U') then (objectproperty([id], 'TableUpdateTrigger'))
else 0 end) ,
[seltrig] AS (convert(int,0)) ,
[category] AS (convert(int,(case when ([status] & 0x80000000 <> 0) then 2 else 0 end + case when ([replinfo] & 1
<> 0) then 32 else 0 end + case when ([replinfo] & 2 <> 0) then 64 else 0 end + case when ([replinfo] & 4 <> 0) then 256 else
0 end + case when ([xtype] = 'P' and [status] & 2 <> 0) then 16 else 0 end + case when ([xtype] = 'D' and [parent_obj] <> 0)
then 2048 else 0 end + case when ([xtype] = 'S' or [xtype] = 'U') then (case when ([status] & 0x20 <> 0) then 1 else 0 end +
case when ([status] & 0x0400 <> 0) then 4 else 0 end + case when ([status] & 0x0800 <> 0) then 8 else 0 end + case when
([status] & 0x1000 <> 0) then 128 else 0 end + case when ([status] & 0x0100 <> 0) then 512 else 0 end + case when
([status] & 0x0200 <> 0) then 1024 else 0 end + case when ([status] & 0x2000 <> 0) then 2048 else 0 end + case when
([status] & 0x4000 <> 0) then 4096 else 0 end + case when ([status] & 0x10 <> 0) then 16384 else 0 end)) ,
[cache] AS (convert(smallint,0)
)
GO

```

```
CREATE TABLE [syspermissions] (
    [id] [int] NOT NULL ,
    [grantee] [smallint] NOT NULL ,
    [granted] [smallint] NOT NULL ,
    [actadd] [smallint] NOT NULL ,
    [actmod] [smallint] NOT NULL ,
    [seladd] [varbinary] (4000) NULL ,
    [selmod] [varbinary] (4000) NULL ,
    [updadd] [varbinary] (4000) NULL ,
    [updmod] [varbinary] (4000) NULL ,
    [refadd] [varbinary] (4000) NULL ,
    [refmod] [varbinary] (4000) NULL
)
GO
```

```
CREATE TABLE [sysprotects] (
    [id] [int] NOT NULL ,
    [uid] [smallint] NOT NULL ,
    [action] [tinyint] NOT NULL ,
    [protecttype] [tinyint] NOT NULL ,
    [columns] [varbinary] (4000) NULL ,
    [grantor] [smallint] NOT NULL
)
GO
```

```
CREATE TABLE [sysreferences] (
    [constid] [int] NOT NULL ,
    [rkeyid] [int] NOT NULL ,
    [rkeyid] [int] NOT NULL ,
    [rkeyindid] [smallint] NOT NULL ,
    [rkeycnt] [smallint] NOT NULL ,
    [forkeys] [varbinary] (32) NOT NULL ,
    [refkeys] [varbinary] (32) NOT NULL ,
    [rkeybid] AS (convert(smaliint,0)) ,
    [rkeybid] AS (convert(smaliint,0)) ,
    [rkey1] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([forkeys],1,2)),0))) ,
    [rkey2] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([forkeys],3,2)),0))) ,
    [rkey3] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([forkeys],5,2)),0))) ,
    [rkey4] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([forkeys],7,2)),0))) ,
    [rkey5] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([forkeys],9,2)),0))) ,
    [rkey6] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([forkeys],11,2)),0))) ,
    [rkey7] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([forkeys],13,2)),0))) ,
    [rkey8] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([forkeys],15,2)),0))) ,
    [rkey9] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([forkeys],17,2)),0))) ,
    [rkey10] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([forkeys],19,2)),0))) ,
    [rkey11] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([forkeys],21,2)),0))) ,
    [rkey12] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([forkeys],23,2)),0))) ,
    [rkey13] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([forkeys],25,2)),0))) ,
    [rkey14] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([forkeys],27,2)),0))) ,
    [rkey15] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([forkeys],29,2)),0))) ,
    [rkey16] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([forkeys],31,2)),0))) ,
    [rkey1] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([refkeys],1,2)),0))) ,
    [rkey2] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([refkeys],3,2)),0))) ,
    [rkey3] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([refkeys],5,2)),0))) ,
    [rkey4] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([refkeys],7,2)),0))) ,
    [rkey5] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([refkeys],9,2)),0))) ,
    [rkey6] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([refkeys],11,2)),0))) ,
    [rkey7] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([refkeys],13,2)),0))) ,
    [rkey8] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([refkeys],15,2)),0))) ,
    [rkey9] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([refkeys],17,2)),0))) ,
    [rkey10] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([refkeys],19,2)),0))) ,
    [rkey11] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([refkeys],21,2)),0))) ,
    [rkey12] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([refkeys],23,2)),0))) ,
    [rkey13] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([refkeys],25,2)),0))) ,
    [rkey14] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([refkeys],27,2)),0))) ,
    [rkey15] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([refkeys],29,2)),0))) ,
    [rkey16] AS (convert(smaliint,isnull(convert(binary(2),reverse(substring([refkeys],31,2)),0)))
)
GO
```

```

CREATE TABLE [sysotypes] (
    [name] [sysname] NOT NULL ,
    [xtype] [tinyint] NOT NULL ,
    [status] [tinyint] NOT NULL ,
    [xuserype] [smallint] NOT NULL ,
    [length] [smallint] NOT NULL ,
    [xprec] [tinyint] NOT NULL ,
    [xscale] [tinyint] NOT NULL ,
    [tdefault] [int] NOT NULL ,
    [domain] [int] NOT NULL ,
    [uid] [smallint] NOT NULL ,
    [reserved] [smallint] NOT NULL ,
    [userype] AS (convert(smallint,typeproperty([name], 'olduserype'))) ,
    [variable] AS (convert(bit,case when ((type_name([xtype]) = 'nvarchar' or type_name([xtype]) = 'varchar' or
type_name([xtype]) = 'varbinary')) then 1 else 0 end)) ,
    [allownulls] AS (convert(bit,(1 - ([status] & 1)))) ,
    [type] AS (convert(tinyint,xtpetolds([xtype],0))) ,
    [printfmt] AS (convert(varchar(255),null)) ,
    [prec] AS (convert(smallint,case when ((type_name([xtype]) = 'ntext' or type_name([xtype]) = 'image' or
type_name([xtype]) = 'text') then null else (typeproperty([name], 'precision')) end)) ,
    [scale] AS (convert(tinyint,typeproperty([name], 'scale')))
)
GO

CREATE TABLE [sysusers] (
    [uid] [smallint] NOT NULL ,
    [status] [smallint] NOT NULL ,
    [name] [sysname] NOT NULL ,
    [sid] [varbinary] (85) NULL ,
    [roles] [varbinary] (2048) NOT NULL ,
    [createdate] [datetime] NOT NULL ,
    [updatedate] [datetime] NOT NULL ,
    [alluid] [smallint] NOT NULL ,
    [password] [varbinary] (256) NULL ,
    [suid] AS (convert(smallint, user_id([user_sname]([sid]))) ,
    [gid] AS (convert(smallint,case when ([uid] >= 16400) then [uid] when ([datalength([roles]) is null or
datalength([roles]) <= 2) then 0 else (16384 - 8 * datalength([roles]) * 8 + case when (convert(tinyint,[roles]) & 1 <> 0) then
0 when (convert(tinyint,[roles]) & 2 <> 0) then 1 when (convert(tinyint,[roles]) & 4 <> 0) then 2 when (convert(tinyint,[roles]) &
8 <> 0) then 3 when (convert(tinyint,[roles]) & 16 <> 0) then 4 when (convert(tinyint,[roles]) & 32 <> 0) then 5 when
(convert(tinyint,[roles]) & 64 <> 0) then 6 when (convert(tinyint,[roles]) & 128 <> 0) then 7 end end)) ,
    [environ] AS (convert(varchar(255),null)) ,
    [hasdbaccess] AS (convert(int,case when ([status] & 2 = 2) then 1 else 0 end)) ,
    [islogin] AS (convert(int,case when ([status] & 32 = 0 and [uid] > 0 and [uid] < 16384) then 1 else 0 end)) ,
    [isntname] AS (convert(int,case when ([status] & 4 = 4) then 1 else 0 end)) ,
    [isntgroup] AS (convert(int,case when ([status] & 12 = 4) then 1 else 0 end)) ,
    [isqluser] AS (convert(int,case when ([status] & 12 = 12) then 1 else 0 end)) ,
    [issqluser] AS (convert(int,case when ([status] & 60 = 0 and [uid] > 0 and [uid] < 16384) then 1 else 0 end)) ,
    [isaliases] AS (convert(int,case when ([status] & 16 = 16) then 1 else 0 end)) ,
    [issqrole] AS (convert(int,case when ([uid] >= 16384 or [uid] = 0) then 1 else 0 end)) ,
    [isapprole] AS (convert(int,case when ([status] & 32 = 32) then 1 else 0 end))
)
GO

CREATE TABLE [TIPO_EGRESO] (
    [id_tipo_egreso] [smallint] NOT NULL ,
    [nom_tipo_egreso] [char] (40) NOT NULL ,
    CONSTRAINT [PK_TIPO_EGRESO] PRIMARY KEY NONCLUSTERED
(
    [id_tipo_egreso]
) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [TIPO_GRUPO] (
    [id_tipo_grupo] [smallint] NOT NULL ,
    [nom_tipo_grupo] [char] (30) NOT NULL ,
    CONSTRAINT [PK_TIPO_CURSO] PRIMARY KEY NONCLUSTERED
(
    [id_tipo_grupo]
) ON [PRIMARY]

```

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [TIPO_PAGO] (
    [id_tipo_pago] [smallint] NOT NULL ,
    [nom_tipo_pago] [char] (40) NOT NULL ,
    CONSTRAINT [PK_TIPO_PAGO] PRIMARY KEY NONCLUSTERED
    (
        [id_tipo_pago]
    ) ON [PRIMARY]
)
```

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [TIPO_PERSONAL] (
    [id_tipo_personal] [smallint] NOT NULL ,
    [nom_tipo_personal] [char] (25) NOT NULL ,
    [descripcion] [char] (50) NULL ,
    CONSTRAINT [PK_TIPO_PERSONAL] PRIMARY KEY NONCLUSTERED
    (
        [id_tipo_personal]
    ) ON [PRIMARY]
)
```

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [TIPO_SOLICITUD] (
    [id_tipo_solicitud] [smallint] IDENTITY (1, 1) NOT NULL ,
    [nom_tipo_solicitud] [char] (30) NOT NULL ,
    CONSTRAINT [PK_TIPO_SOLICITUD] PRIMARY KEY NONCLUSTERED
    (
        [id_tipo_solicitud]
    ) ON [PRIMARY]
)
```

```
) ON [PRIMARY]
GO
```

TESIS CON
FALLA DE ORIGEN