

72
2y.



UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO



FACULTAD DE INGENIERÍA

DISEÑO DE UN SISTEMA PARA EL CONTROL Y
LOGÍSTICA DE EVENTOS ACADÉMICOS PARA LA DEPTI

T E S I S

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

MERCADO RODRÍGUEZ HÉCTOR NELSON
PADILLA LEGUÍZAMO LUIS ALBERTO
SILVESTRE MARTÍNEZ JAVIER



DIRECTOR DE TESIS: ING. MARTÍN PÉREZ MONDRAGÓN

MÉXICO, DF

1998

TESIS CON
FALLA DE ORIGEN

268329



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.

Agradezco a:

Dios:

Por su infinita grandeza y el haberme dado la vida y el tiempo suficiente para lograr este objetivo el cual lo comparto con toda mi familia.

Mi familia: Alma, Isidro y Raúl

Sabiendo que no existirá una forma de agradecer una vida de sacrificio y esfuerzo, quiero que sientan que el objetivo logrado también es de ustedes y que la fuerza que me ayudó a conseguirlo fue su apoyo y comprensión...

Al LSI:

Ya que en este laboratorio encontré el apoyo y experiencia suficiente para seguir adelante en mis objetivos, en especial al Ing. Martín Pérez Mondragón por haber dirigido esta Tesis.

La Universidad Nacional Autónoma de México en la que está incluida la Facultad de Ingeniería con Maestros y Amigos

*Con Cariño y Admiración a Todos
Gracias...*

JAVIER

A mi familia:

Por todo el apoyo otorgado para mis estudios durante este tiempo. Por haberme permitido llegar a tener este momento tan anhelado.

Mamá: Por tu cariño, apoyo, orientación, cuidados y más.

Jorge: Por tu paciencia e invaluable apoyo en las buenas y en las malas.

José: Por tu comprensión y el apoyo condicional e incondicional que me ofreciste.

Edgardo: Por tus consejos y visión sobre las cosas, por tu modo tan enérgico de hacerme entender. Por tus enseñanzas.

A todos gracias.

Jazmin, Ixchel:

Por toda la confianza, paciencia, cariño, apoyo y más, que me brindaron. Por aguantarme durante este largo tiempo.

De todo corazón.

Gracias

A la Universidad Nacional Autónoma de México:

A ti, Gracias.

Por permitir alimentarme de tu sapiencia.

Con orgullo pondré tu nombre en alto.

Por todo lo que representas y eres.

A mis compañeros Tesistas Nelson y Javier:

Gracias por todo, esperando que este sea el principio del un nuevo comienzo.

A mis amigos:

Gracias. Ya que con ustedes aprendí a valorar más la amistad.

LUIS ALBERTO

Agradezco a:

A mis padres:

Gracias porque he logrado concluir una de tantas metas que tengo que realizar en esta vida y no lo hubiera logrado sin su apoyo, confianza, comprensión y lo más importante su cariño, sobre todo por haberme mostrado el camino correcto para llegar hasta donde estoy en estos momentos.

A mis hermanos:

Por sus consejos, apoyo y cariño que me dieron cuando más lo necesite porque gracias a eso pude superar uno de los tantos obstáculos que nos depara esta vida.

A la Universidad Nacional Autónoma de México:

Por haber permitido desarrollarme como todo un profesionalista y sobre todo como hombre de bien para el bienestar de nuestro país.

A mis Profesores:

Porque gracias a ustedes pude alimentarme de sus sabios conocimientos y experiencias.

Al Laboratorio de Sistemas de Información

Porque ellos formaron una parte fundamental para concluir esta tesis y sobre todo al Ing. Martín Pérez Mondragón.

A mis amigos y Seres Queridos

Gracias...

Nelson

ÍNDICE

Página

Introducción

CAPÍTULO I ANTECEDENTES GENERALES

Introducción	2
I.1 Diferencia entre Datos e Información	3
I.2 Sistema y Sistema de Información	3
I.3 Ciclo de Vida del Software	4
I.4 Tipos de Ciclo de Vida	5
I.4.1 Aproximación convencional	5
I.4.2 Aproximación prototipo	6
I.4.3 Aproximación evolutiva	7
I.4.4 Aproximación incremental	8
I.4.5 Aproximación espiral	9
I.5 Ciclo de Muerte del Software	11
I.6 Las leyes del Software	13

CAPÍTULO II TÉCNICAS Y HERRAMIENTAS PARA EL DESARROLLO DE SISTEMAS DE INFORMACIÓN

Introducción	17
II.1 Diagramas de Flujo de Datos (DFD)	18
II.1.1 Proceso de análisis en el DFD	20
II.1.2 Propiedades de los DFD	20
II.2 Modelo Entidad-Relación	22
II.2.1 Estándares para Entidades	24
II.2.2 Estándares de Atributos	25
II.2.3 Estándares para Relaciones	25
II.2.4 Estándares del diagrama E-R	26
II.2.5 Tipos de Relaciones	27
II.3 Técnica Arriba-Abajo (Top-Down)	30
II.3.1 Diagrama Jerárquico Funcional	30
II.4 Diagramas de Estructura	31
II.4.1 Elementos principales del Diagrama de Estructura	31

II.5 Matrices	33
II.5.1 Matriz Entidad-Entidad	33
II.5.2 Matriz Evento-Entidad	34
II.5.3 Matriz Papel del Usuario-Función	35
II.6 Herramientas CASE	36
II.6.1 Ejemplos de herramientas CASE	38

CAPÍTULO III ANÁLISIS DEL SICLO-EA

Introducción	41
III.1 Antecedentes	42
III.2 Necesidades de la DEPMI	42
III.3 Definición del problema	42
III.4 Justificación del Sistema	43
III.5 Objetivos	43
III.6 Alcances	44
III.7 Requerimientos del Usuario	44
III.8 Beneficios que se proporcionarán	47
III.9 Riesgos	47
III.10 Herramientas y Técnicas que se emplearán	48
III.10.1 Modelo E-R	48
III.10.2 Técnica Top-Down	50
III.11 Planificación del Desarrollo mediante gráficas de Gantt	50

CAPÍTULO IV DISEÑO DEL SICLO-EA

Introducción	54
IV.1 Concepto del Diseño de Software	55
IV.2 Diseño Estructural	55
IV.2.1 Diagrama Jerárquico Funcional General	56
IV.2.2 Parámetros	57
IV.2.3 Expediente	58
IV.2.4 Accesos	58
IV.2.5 Catálogos	59
IV.2.6 Informes	60
IV.2.7 Búsqueda	60
IV.2.8 Diagramas de Flujo de Datos	61
IV.2.9 Diagrama Entidad-Relación	75
IV.3 Diseño Detallado	77
IV.3.1 Diccionario de Datos	77
IV.4 Especificaciones de Diseño	109
IV.4.1 Diseño de la salida para que sirva al propósito deseado	109
IV.4.2 Diseño de la salida para que se ajuste al usuario	109
IV.4.3 Convenciones para el diseño del reporte	111
IV.4.4 Consideraciones del diseño de reportes	113
IV.4.5 Lineamientos para el diseño de pantallas	114

IV.5 Problemas de los sistemas	115
IV.5.1 Presupuesto limitado	116
IV.6 Cohesión y Acoplamiento	116
IV.6.1 Cohesión	116
IV.6.2 Acoplamiento	117
CAPÍTULO V DESARROLLO DEL SICLO-EA	
Introducción	119
V.1 La fase de desarrollo	120
V.2 Desarrollo del SICLO-EA	120
V.2.1 Desarrollo arriba hacia abajo	120
V.2.2 Desarrollo modular	121
V.2.3 Modularidad en el ambiente Windows	123
V.3 Diagramas de flujo de datos Lógicos y Físicos	123
V.3.1 Desarrollo de diagramas de flujo de datos lógicos	124
V.3.2 Desarrollo de diagramas de flujo de datos físicos	125
V.4 Archivo de base de datos	126
V.4.1 Normalización	126
V.4.2 Uso del diagrama E-R para determinar las llaves de registro	127
V.5 Tipos de interfaz de usuario	128
V.5.1 Interfaces de llenado de forma	128
V.5.2 Interfaces Gráficas de Usuario (GUI)	129
V.6 La administración de la calidad total en el desarrollo del SICLO-EA	131
V.6.1 Responsabilidad de la administración de la calidad total	132
V.7 Elección del manejador de base de datos y lenguaje de programación	134
V.8 Documentación del código	135
CAPÍTULO VI PRUEBAS, LIBERACIÓN Y MANTENIMIENTO DEL SICLO-EA	
Introducción	137
VI.1 Definición y tipos de pruebas	138
VI.1.1 Objetivo de las pruebas	138
VI.1.2 Etapas para el proceso de pruebas	139
VI.1.3 Diseño de pruebas	141
VI.1.4 Pruebas para el SICLO-EA	143
VI.2 Mantenimiento del Software	146
VI.2.1 Costos del mantenimiento	147
VI.2.2 Facilidad de mantenimiento	148
VI.2.3 Factores que afectan la facilidad de mantenimiento	148
VI.2.4 Mantenimiento para el SICLO-EA	149
CONCLUSIONES	150

INTRODUCCIÓN

En la División de Estudios de Posgrado de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México cada año se organizan eventos académicos, como seminarios, congresos, conferencias y otros, dichos eventos requieren manejar grandes cantidades de información, para ello surge la necesidad de crear un sistema que ayude en el manejo de la información. La información debe ser lo más confiable y oportuna posible para que el usuario no tenga que verificar la información o no tenga confianza en ella.

Una vez que se plantearon los objetivos del sistema se comenzó a investigar cual sería la mejor opción para realizar el Software, de la cual se tomaron en cuenta varias posibilidades para hacer el sistema y se optó por ACCESS 97 como herramienta para desarrollar el Software, de esa manera se creó el Sistema para el Control y Logística de Eventos Académicos (SICLO-EA).

Finalmente el objetivo de este trabajo es tener una alternativa en Software para controlar eventos académicos y la logística que un evento debe tener.

Capítulo I

Antecedentes

Generales

ANTECEDENTES GENERALES

Introducción

Podríamos tratar de definir que es software, ya que software significa distintas cosas para personas diferentes y, por ello, se sostiene una definición simple; por ejemplo, el usuario de un sistema puede pensar que el software no es más que una colección de programas y datos que hacen de una computadora de propósito general una máquina de propósito especial diseñada para desempeñar una aplicación particular.

De acuerdo con esta definición, el principal interés sería asegurar que se utiliza un grupo de programas y datos completos y compatibles. Una definición alternativa, desde el punto de vista de un desarrollador de software, puede ser que el software es toda la información (*documentación*) producida durante el desarrollo de un sistema de software intensivo. Las connotaciones de esta definición difieren mucho del punto de vista del usuario.

1.1 Diferencia entre Datos e Información

Los **datos** son la materia prima de la información. Los datos son hechos o entes aislados que se pueden representar a través de símbolos, que para considerarlos como información debe existir un **proceso** para convertir los datos en información, el proceso convierte los datos en información, toda la información consta de datos pero no todos los datos producen información útil para una persona específica. En la forma en que el término se utiliza en el procesamiento de datos, la información consta de datos organizados en una forma que es útil para las personas que la reciben. En otras palabras, la **información** consiste en conocimientos importantes producidos como resultado de las operaciones de procesamiento de datos. Así mismo las personas los adquieren para mejorar su comprensión y para lograr objetivos específicos.

El procesamiento de datos consta de tres actividades básicas:

1. Capturar los datos de entrada. Esto puede hacerse en documentos o se puede capturar a través de un dispositivo de entrada de una computadora.
2. Organizar los datos. Se realiza una serie de operaciones con los datos capturados como son: agruparlos, manipularlos aritméticamente, clasificarlos y hacer una síntesis.
3. Manejar los resultados de salida. Implica realizar alguna de las operaciones siguientes; almacenamiento y recuperación o comunicación y reproducción.

1.2 Sistema y Sistema de Información

Para que exista un proceso se deben tener entradas las cuales van a producir salidas y a esto en conjunto se le denomina Sistema. Un **Sistema** es un conjunto de partes que están integradas con el propósito de lograr un objetivo común, decimos que es un conjunto de partes ya que un Sistema tiene más de un elemento, y que además están integradas ya que debe existir una relación lógica entre las partes del Sistema y por último el lograr un objetivo común ya que el Sistema se diseña para alcanzar uno o más objetivos. Dado que

una computadora es un grupo de partes integradas que tiene el objetivo común de llevar a cabo las operaciones que indica el programa que se está ejecutando, entra en la definición de Sistema.

La información oportuna, completa y exacta son el objetivo primordial de un Sistema de Información. Un Sistema de información es un proceso que contiene varios pasos para identificar y más tarde desarrollar un programa que se va a utilizar para controlar el procesamiento de alguna aplicación específica.

Pasos a seguir en la construcción de un Sistema de información:

1. Definición del problema.
2. Análisis del Sistema.
3. Diseño del Sistema.
4. Desarrollo del Sistema.
5. Pruebas del Sistema.
6. Implantación.
7. Mantenimiento.

I.3 Ciclo de vida del Software

El ciclo de vida proporciona un modelo conveniente que sirve para dos propósitos. En primer lugar, permite representar los procesos de concepción y producción en una forma gráfica y lógica, y segundo, proporciona un marco de trabajo alrededor del cual las actividades de aseguramiento de calidad pueden ser construidas en una manera decidida y disciplinada.

El desarrollo de Software desde el concepto inicial a través de la operación es un proceso involuntario. Es decir, se produce mediante etapas sucesivas de especificación, diseño y modificación. Idealmente, el proceso de desarrollo debe involucrar gradas sucesivas de especificación y diseño donde cada paso es verificado contra los requerimientos de la etapa precedente. Así un producto de Software viable evoluciona con errores que se encuentran y corrigen conforme ocurren.

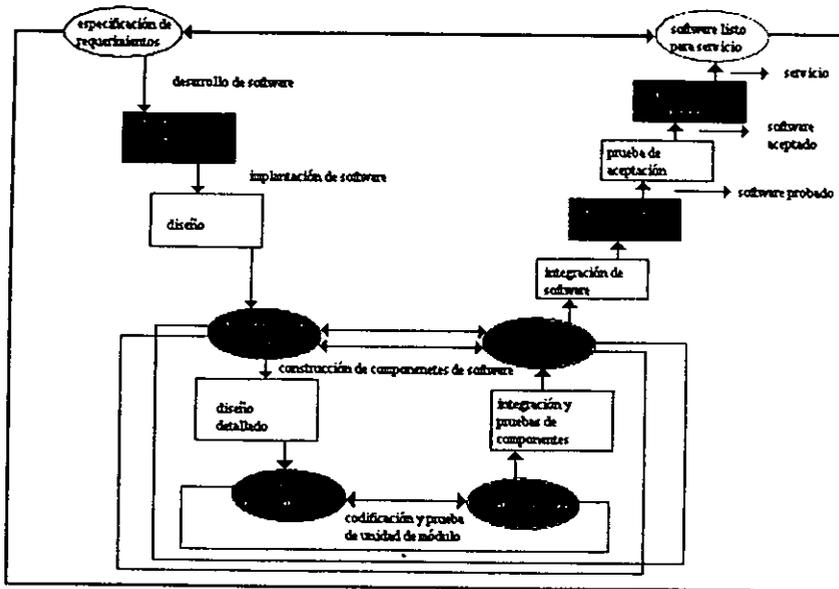


Figura I.1 Ciclo de vida cascada

I.4 Tipos de ciclo de vida

I.4.1 Aproximación convencional

Se introdujo como una técnica rígida para mejorar la calidad y reducir los costos del Software. Se denomina tradicionalmente “modelo en cascada”, porque su filosofía es completar una paso, antes de empezar el siguiente, finalizando cada paso con un alto grado de exactitud.

Los principales problemas que se han detectado con esta aproximación son debidos a que comienza estableciendo todos los requisitos del problema. Sin embargo, el ciclo de desarrollo Software en cascada suministra un marco de referencia para la asignación de todas las actividades de desarrollo de Software, además sigue siendo la aproximación más empleada por los ingenieros de Software.

1.4.2 Aproximación prototipo

Esta aproximación consiste en realizar la fase de definición de requisitos del Sistema caracterizada por estos tres factores:

- *Un alto grado de iteración.* La premisa básica para realizar el análisis mediante prototipos es que los usuarios no pueden indicar qué requisitos desean para el Sistema sin tener una primera experiencia de él. El uso de un prototipo posibilita un mayor conocimiento del problema. En el proceso, el prototipo es modificado de una manera iterativa hasta que los requisitos deseados por el usuario queden claros.
- *Un muy alto grado de participación del usuario.* Aunque hay pocos datos en este tema, se asume que la participación del usuario con esta aproximación será mucho mayor que con la aproximación tradicional. En el análisis mediante prototipos, el usuario es un participante activo, evaluando prototipos, proponiendo mejoras y a la vez detallando los requisitos deseados del Sistema de información.
- *Un uso extensivo de prototipo.*

En la fase de análisis realizada mediante la aproximación prototipo se distinguen cinco pasos tal como se muestra en la siguiente figura.

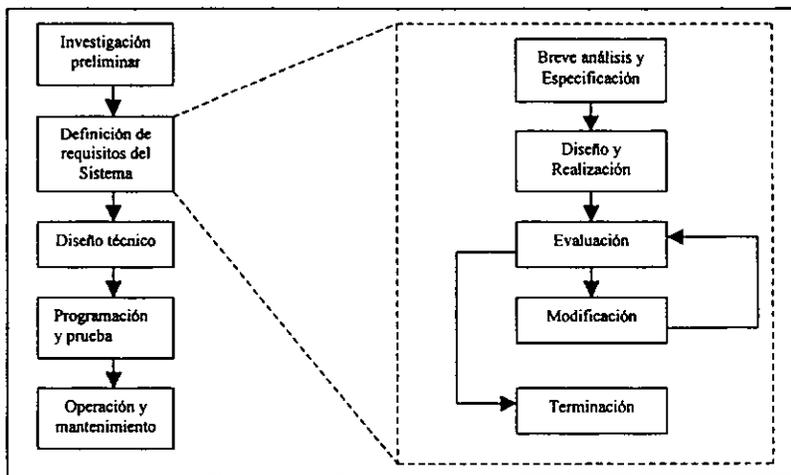


Figura 1.2 Aproximación Prototipo

En el primer paso, **breve análisis y especificación**, se analiza el problema en términos globales, con el objetivo de proveer una base sólida para los siguientes pasos.

En el segundo paso, **diseño y realización**, se construye un prototipo.

En el tercer paso, **evaluación**, los usuarios finales manejan y evalúan el prototipo. Si es necesario cambiar algún requisito, esto se realiza en el cuarto paso **modificación**.

En el último paso, **definición de requisitos del Sistema**, se completa añadiendo los requisitos de funcionamiento y calidad.

Se ha de tener en cuenta siempre que la aproximación del análisis mediante prototipos está enfocada a mejorar la efectividad (*el aspecto de efectividad se refiere a cómo el Sistema de información satisface las necesidades reales de información de los usuarios*) del proceso de desarrollo y no a mejorar la eficiencia (*se refiere al costo en tiempo y dinero que supone la realización de un Sistema de información*) de ese proceso.

1.4.3 Aproximación evolutiva

En esta aproximación, el énfasis está en lograr un Sistema flexible y que se pueda expandir de tal manera que se pueda realizar muy rápidamente una versión modificada del Sistema cuando los requisitos cambien. Sucede una situación parecida cuando se realiza un nuevo prototipo para ajustarse a los cambios de los requisitos de los usuarios.

La diferencia esencial entre la aproximación prototipo y el desarrollo evolutivo es que la primera asume que existe una serie de requisitos reales y que para establecer lo que realmente el usuario quiere, puede ser necesario realizar un número dado de iteraciones, pero que al final, la definición de requisitos se estabilizará; mientras que en el segundo la asunción es que los requisitos cambian continuamente. Tal situación no se considera en la aproximación de prototipo, por que si no el proceso iterativo nunca acabaría.

I.4.4 Aproximación incremental

El desarrollo incremental es un concepto muy parecido y frecuentemente comprendido en la aproximación del desarrollo evolutivo.

En la aproximación de desarrollo incremental, se comienza el desarrollo del Sistema para satisfacer un subconjunto de requisitos especificados. Las últimas versiones proveen los requisitos que faltan. Con esta aproximación se logra una pronta disponibilidad del Sistema que, aunque todavía incompleto, es por lo menos utilizable y satisface alguna de las necesidades básicas de información.

Las versiones sucesivas de un Sistema desarrollado con la aproximación evolutiva y la incremental difieren claramente. En el primer caso, cada vez se desarrolla una nueva versión de todo el Sistema; mientras que en el segundo, cada versión parte de una versión previa sin cambios, más un número de nuevas funciones. Por supuesto, es posible desarrollar un Sistema incrementalmente y al mismo tiempo seguir una aproximación evolutiva para desarrollar componentes particulares del Sistema.

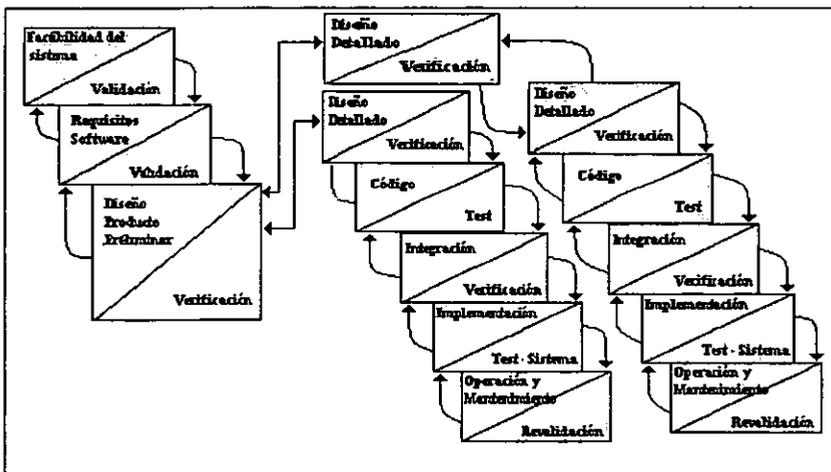


Figura I.3 Aproximación incremental

I.4.5 Aproximación espiral

Esta aproximación nace con el propósito de recoger lo mejor de la aproximación convencional y de la aproximación prototipo, añadiéndole un nuevo componente: *el análisis de riesgo*.

La primera vuelta de la espiral, es decir, en la vuelta más interna, se definen los objetivos, las alternativas y las restricciones, se analizan y se identifican los riesgos, Si como consecuencia del análisis de riesgo se observa que hay incertidumbre (*entendiendo por incertidumbre la diferencia entre el conocimiento ya obtenido de un problema y aquél que se necesita para llegar a una solución aceptable*) sobre el problema de información entonces en la actividad correspondiente a la ingeniería se aplicará la aproximación prototipo cuyo beneficio principal es el de reducir la incertidumbre de la naturaleza del problema de información y los requerimientos que los usuarios establecen para la solución a ese problema.

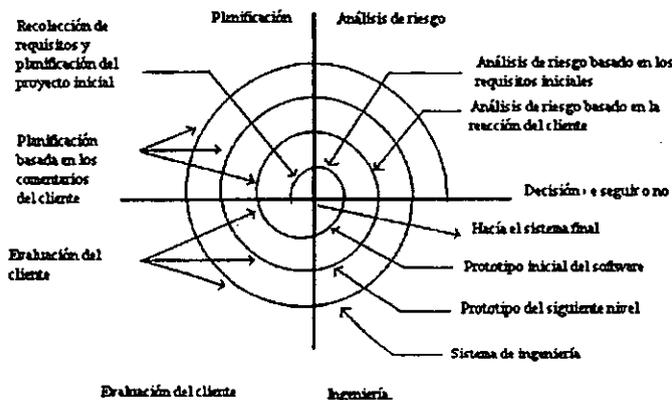


Figura I.4 Aproximación Espiral

ACTIVIDADES	ACCIONES
• Planificación	Determinación de alternativas, identificación y resolución de riesgos.
• Análisis de Riesgo	Evaluación de alternativas, identificación y resolución de riesgos
• Ingeniería	Desarrollo y verificación del producto del siguiente nivel.
• Evaluación del Cliente	Valoración de los resultados del proceso de desarrollo.

La incertidumbre sobre el nivel conceptual puede tener consecuencias muy serias ya que implicará una falta de claridad en las funciones del Sistema y sobre la estructura de la base de datos. En cambio, la incertidumbre en la forma del Sistema normalmente tendrá consecuencias financieras menos serias, aunque sí puede afectar bastante en la satisfacción del usuario.

El mayor riesgo de incertidumbre a nivel conceptual se da cuando después de la fase de análisis no se han obtenido los requisitos del Sistema, ni los usuarios ni los analistas tienen conocimiento o experiencia en el área de aplicación y hay un alto riesgo de problemas de comunicación.

En tales circunstancias, la aproximación mediante prototipos conducirá casi con total seguridad ahorros considerables.

Además de reducir los costos de mantenimiento por reducir la incertidumbre, también se reducirá el tiempo empleado en la fase de prueba funcional y por consiguiente esta fase será más económica. Del mismo modo hay ahorros en las fases de diseño, realización y mantenimiento.

Al final de esta primera vuelta alrededor de la espiral, el cliente evalúa los productos obtenidos y puede sugerir modificaciones.

Se comenzaría avanzando alrededor del camino de la espiral realizando las cuatro actividades ya enunciadas. En cada vuelta de la espiral, la actividad de ingeniería será desarrollada mediante la aproximación convencional o ciclo de desarrollo en cascada o mediante la aproximación de prototipos. Debemos esperar todavía algún tiempo para comprobar la eficacia de este ciclo de desarrollo de Software.

I.5 Ciclo de Muerte del Software

La premisa básica que sostiene el modelo de ciclo de muerte es que el mantenimiento de un Sistema entregado (*para un estándar dado de calidad*) cuesta dinero y tiene que ser compensado con los beneficios que se obtienen del Software. En algún punto, el costo de posesión comienza a sobrepasar el beneficio y este es el aspecto en el cual el neto se vuelve negativo. Este es el momento en que el Software debe ser retirado de uso.

Una observación inmediata es que el ingreso durante la fase de desarrollo es nulo. Esta fase sólo cuesta dinero. Señalar con exactitud cuánto dinero se gasta en ésta, se encuentra sujeto a un gran número de factores.

Una vez entregado o puesto en operación, el Software comienza a producir ganancias. Éstas deben dar cuenta de un número de cosas, la más obvia de ellas es el creciente costo de mantenimiento. Es claro que no tendría sentido desarrollar un Sistema al que darle servicio costara más de lo que es capaz de producir. El costo actual del mantenimiento de un Software varía de acuerdo con, entre otras cosas, el nivel requerido de calidad. Dado esto, el costo probable de mantener un número conocido de variantes de un producto de Software que ha sido instalado en un número conocido de locaciones podría, en una primera aproximación razonable, estimarse.

Existen otros factores después y antes del costo de mantenimiento que chocan con una variable económica; el costo inicial del desarrollo necesita recuperarse tras cierto período y viendo hacia delante, hay necesidad de ganar suficiente dinero para fundamentar un replazo del Software cuando eventualmente muera.

De manera inevitable llega el momento en que el ingreso empieza a declinar y/o la cantidad de esfuerzo de mantenimiento requerido para poder sostener un aceptable nivel de calidad se vuelve prohibitivamente caro. Si el gasto empleado en el Software es mayor que el ingreso generado por él, entonces es el momento de retirarlo.

La gráfica de ciclo de muerte del Software puede ser utilizada como modelo del efecto de muchas de las decisiones planeadas a menudo de forma aislada durante un proyecto de Software. Una vez que el modelo instalado con pronósticos apropiados, hechos y cifras, se permite que la muerte del Software sea conocida y planeada de manera explícita.

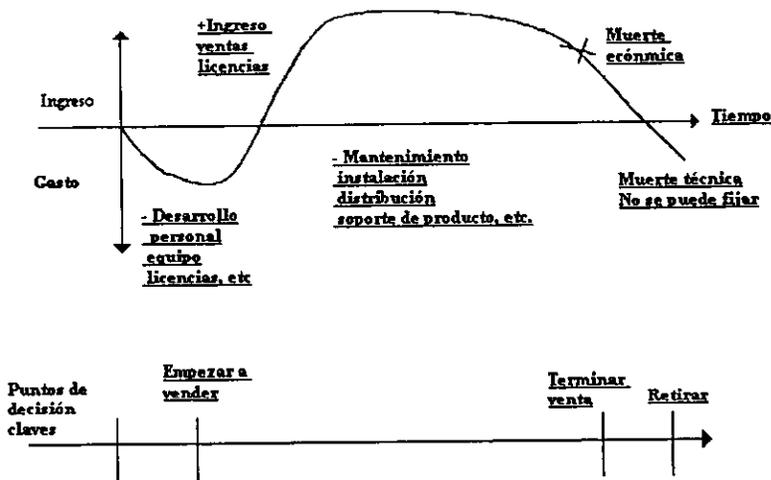


Figura I.5 Ciclo de Muerte del Software

Una consecuencia rápida y muy importante de la modelación del costo de posesión del Software a lo largo de su tiempo de vida es que se vuelve imposible depreciar el Software en la misma forma que otros bienes. Este es un significado efectivo de poner dinero aparte para pagar un replazo; algo que rara vez es planeado en la mayoría de los esquemas contables.

I.6 Las leyes del Software

Por desgracia, no hay leyes de Software en el mismo sentido como las hay para el football. Aun así, hay cuerpos de sabiduría construido en los últimos 30 años que se aplica a todos los grandes proyectos de Software.

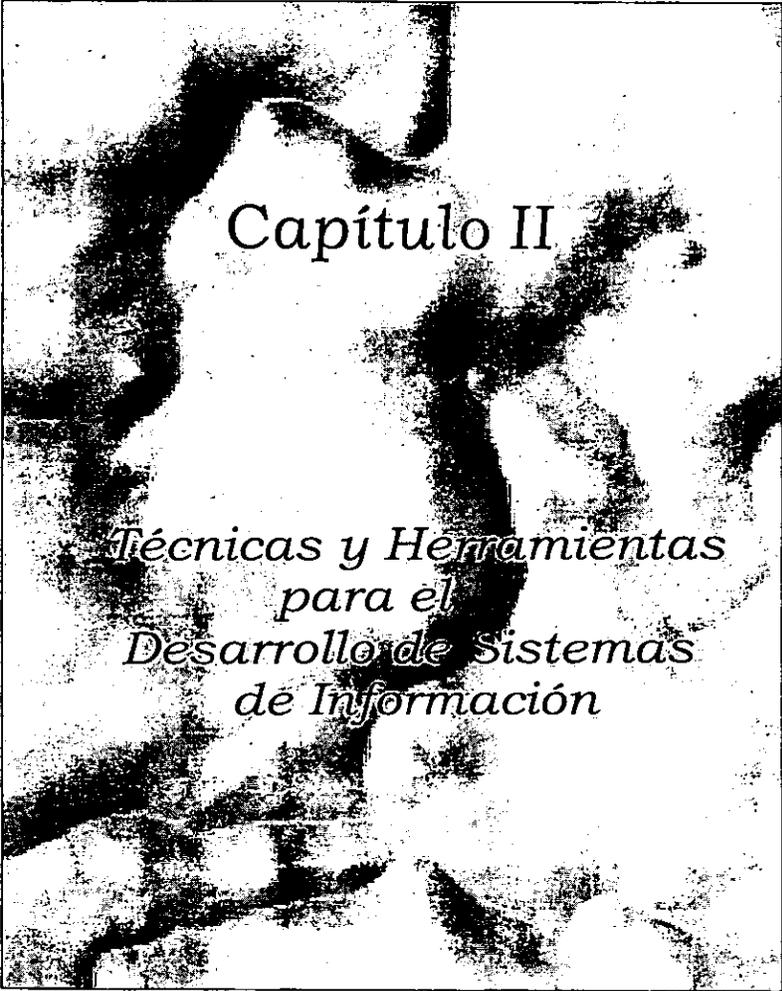
- *El Software no se gasta.* Aunque puede ser cierto en principio, de hecho no concuerda con la realidad. Muy pocos de los sistemas de Software creados hace veinte o treinta años continúan en operación. Las líneas de código son muy fáciles de cambiar, pero los sistemas pueden ser increíblemente difíciles de alterar con éxito. Esto es por el hecho fundamental de que el Software es un Sistema con miles de dependencias entremezcladas e interconectadas. Los requerimientos de los clientes cambian a cada momento y, como ocurre con los carros viejos, mantener el Software antiguo puede ser caro y, si hablamos con honestidad, un pasatiempo caro.
- *El cambio es inevitable.* El Software debe ser diseñado para darle mantenimiento. A fin de sobrevivir debe ser construido pensando que continuará más allá del retiro de sus creadores, reparado para cuidar esos malditos casos especiales y, por lo general, durar más que aquellos que lo crearon. Hay ejemplos contrarios obvios, situaciones en verdad de un tiro, aplicaciones para equipo que pronto será obsoleto, funciones que pasan de moda, pero es mejor equivocarse en el lado de mucha preparación para el futuro en lugar de muy poca.

Mucha información confiable acerca de los aspectos estructurales de un producto de Software se pierde durante el desarrollo. Al inicio, esto no debe suponer un problema, pero en algunas ocasiones es usual que sea precisamente esa información la que se necesita para conservar la integridad del Software durante la evolución. Esto es similar a una organización que se divide en departamentos y después pierde toda la información sobre los empleados pertenecientes a cada departamento; la reorganización será muy difícil.

- *El mantenimiento mejora el Software.* Cada vez que se hace un cambio en un producto de Software, existe una posibilidad finita de que se introduzcan nuevos errores. El efecto neto del mantenimiento debe ser mejorar el Software, pero puede empeorarlo. Es más probable que lo segundo sea cierto si, por ejemplo, se mantiene una documentación inadecuada o si no se ha implantado una administración de modificaciones. Se ha establecido que la información estructural es esencial para el mantenimiento técnico y el control administrativo de un Sistema de Software conforme cambia a través del tiempo. A pesar de una oleada de técnicas que incorporan estructura en sus nombres (*programación, diseño, análisis, prueba*), sigue siendo común encontrar Software producido con poca atención prestada a su estructura física y aún más encontrar que, luego del desarrollo inicial, no se mantiene un registro coherente de la estructura interna del Sistema.
- *El costo real de la posesión de Software es bajo.* Si el costo de mantenimiento de un Software no es explícito, entonces el costo percibido por la posesión es bajo. Tomando el mantenimiento como todo el trabajo colectivo después de la entrega, el costo actual puede ser alto. En primer lugar, crear Software de cualquier magnitud cuesta bastante. Si se ha usado mucho, entonces es muy probable que se haya gastado una considerable suma de dinero a lo largo de los años para repararlo, adaptarlo y agregarle nuevas funciones; no es extraño encontrar que se ha gastado tres o cuatro veces el costo original de esta forma.

- *El Software nunca muere.* El Software vive para siempre por algún número de razones, buenas y malas. La razón más obvia y persuasiva es que una vez construido y en uso es productivo, es por inercia que debe ser vencido si va a ser modificado o reemplazado. A los usuarios les agrada. Al jefe le agrada. A los clientes les agrada. No importa si es lento, propenso a errores, si necesita ser expandido o encargarse de nuevas funciones, y así en lo sucesivo. El Software se ha convertido en parte de sus vidas y lo único que ellos quieren es que sea arreglado un poco, pero no reemplazado o modificado en cualquier forma mayor.
- *El tamaño no importa.* Así como se incrementa la complejidad de un proyecto de Software, crece también la dificultad para lograr un producto satisfactorio. Existen límites más allá de los cuales las técnicas convencionales de desarrollo y administración tienen que llegar. Detrás de esos límites, hay también distintos puntos de ruptura en el tamaño del proyecto de Software; a menudo, en la práctica se da el caso de que duplicar el tamaño de un equipo de proyecto tiene poco efecto en las escalas de tiempo, etc.

La relación entre el Software y las personas que lo rodean (*usuarios, clientes, administradores, operadores, y demás*) no es diferente a la relación entre la gente que se encuentra a sí misma asociada por razones históricas (*el matrimonio, relaciones de trabajo*) que tiene poca orientación con su estado presente. Lleva una tremenda cantidad de esfuerzo cambiar la situación y romper la relación con el Software. El resultado es que el Software vive y sigue viviendo.



Capítulo II

*Técnicas y Herramientas
para el
Desarrollo de Sistemas
de Información*

TÉCNICAS Y HERRAMIENTAS PARA EL DESARROLLO DE SISTEMAS DE INFORMACIÓN

Introducción

En este capítulo se describen las principales técnicas y/o herramientas empleadas en el desarrollo de Sistemas de Información (SI), cuyo objetivo es conocer y seleccionar las mejores para adoptarlas en la fase de diseño del SICLO-EA.

Las técnicas nos indican cómo debe ser realizada una actividad técnica determinada identificada en una metodología. Combinan el empleo de unos modelos o representaciones gráficas junto con el empleo de unos procedimientos detallados.

Debemos tener en cuenta que una técnica determinada puede ser utilizada en una o más actividades de la metodología de desarrollo de Software.

II.1 Diagramas de Flujo de Datos (DFD)

Los Diagramas de Flujo de Datos (*DFD*) son una representación gráfica del Sistema y modeliza la visión que tiene el usuario del Sistema, este es el método más extendido en modelado conceptual de procesos. Esta técnica es también conocida por diagramas de burbujas (*Bubble Diagrams*), fue propuesta por Yourdon a finales de los años setenta.

El objetivo principal de esta técnica es la descomposición de un problema complejo en otros más sencillos y manejables, facilitando la modularidad del Sistema. Puede separar la estructura física del Sistema de la lógica, aumentando la facilidad de mantenimiento de los análisis. Los DFD constan de muy pocos elementos, y además son fáciles de aprender.

El DFD es una excelente técnica de representación gráfica para comunicarse y entenderse entre el analista y el usuario.

El DFD consta de los siguientes elementos:

- **Procesos**

Son los lugares donde se transforma o descompone la información de un flujo de datos. Se representan con un círculo indicando su nombre dentro. Significa que se realizan algunas acciones o grupo de acciones.

- **Entidad externa**

Son aquellas organizaciones, departamentos, personas o entidades en general que no pertenecen al Sistema de información, pero que tienen relación con él. Es de donde proceden los datos que necesita el Sistema o hacia donde van los generados por él. Solo aparecen en el diagrama de contexto inicial. Se representan mediante un rectángulo con su nombre en el interior.

- **Almacenes de datos**

Son aquellos lugares donde se almacenan de forma estática la información dentro del Sistema. Su nombre no indica cómo esta organizado. Se representan gráficamente con dos líneas paralelas.

- **Flujos de datos**

Son las tuberías o caminos por donde fluye la información entre los diferentes elementos del Sistema. Se representan a través de un arco orientado.

Los elementos de un DFD aparecen en la figura II.1.

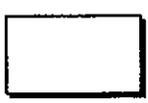
Elemento	Representación
Procesos	
Entidades Externas	
Almacenes de Datos	
Flujo de Datos	

Figura II.1 Elementos de un DFD

Un DFD no debe ocupar más de una hoja, por motivos de claridad. Si queremos representar un Sistema que tenga una complejidad tal que no sea posible realizar esto, se requiere aplicar el principio de descomposición por niveles, que consiste en definir subsistemas de menor complejidad. Si los subsistemas resultantes son todavía demasiado grandes, los seguiremos subdividiendo. Si representamos cada una de estas particiones en un DFD, tendremos al final un *conjunto de DFD jerárquicos*. De esta manera se representa un Sistema bastante más legible que si no lo descompusiéramos y que muestre todos los detalles del Sistema que sean necesarios.

II.1.1 Proceso de análisis en el DFD

El proceso de análisis consiste en un refinamiento progresivo a través de las siguientes fases:

1. Construcción del diagrama de contexto. Es aquel DFD compuesto por un solo proceso que representa todo el Sistema, las entidades externas que participan en él y los flujos de entrada y salida.
2. Descomposición de cada proceso en los subsiguientes niveles de DFD, hasta conseguir todos los procesos primitivos (*aquellos que ya no se pueden seguir descomponiendo*).
3. Descripción de todos los elementos obtenidos en el diccionario de datos.

II.1.2 Propiedades de los DFD

Existen características a tomar en cuenta a la hora de diseñar un diagrama de flujo de datos óptimo:

a. Evitar la utilización de estructuras ilegales. Como son:

- Flujos de datos que se subdividen en el diagrama. Es decir, para que un flujo de datos se pueda desdoblar, es necesario un proceso que realice esta operación.
- Señales de control, y en concreto señales de activación. No pueden aparecer flujos de datos cuyo único contenido sean "flags" (*datos booleanos, señales de final de fichero, pasos de control por llamadas a procedimientos, etc.*)
- Bucles. Si un flujo entra y sale del mismo proceso, significa que es algo que sólo le interesa a dicho proceso, por lo que debe quedar escondido.
- Flujos de datos entre dos almacenes de datos. Para poder hacer una lectura en un almacén y una escritura en otro, es necesaria la existencia de un proceso que realice dicha función.
- Procesos aislados. No tiene sentido la existencia de un proceso que no emita o reciba flujos de datos sin conexión alguna con el resto del Sistema.

- Flujos entre entidades externas. El paso de información que pueda existir entre dos entidades externas es algo que no interesa al Sistema que se está diseñando. Si nos damos cuenta de que sí interesa, es porque esta información debe pasar a través del Sistema.
- Procesos sumidero y procesos fuente. Son aquellos que sólo reciben o emiten información, respectivamente. No son válidos en un DFD.

b. Procurar mantener la propiedad de conservación de los datos

- Los datos no se crean, sólo se transforman. No pueden existir datos en el Sistema que surjan de la nada. Deben aparecer en el Sistema viniendo del exterior, o como resultado de la transformación de otros flujos.
- Balanceo entre niveles (*level balancing*). Esta propiedad se rige por reglas que nos dicen que los flujos de entrada y salida de un proceso de nivel n tienen que ser los mismos que en el diagrama de nivel $n+1$.

c. Propiedades prácticas

Existen una serie de normas que parten de la experiencia:

1. Dividir el Sistema de manera natural. Debemos descomponerlo solo si es necesario no por empeñarnos en descomponerlo el Sistema estará mejor analizado.
2. Establecer conexiones simples. Cuantos menos flujos de datos haya entre dos procesos, mejor.
3. Limitar el número de procesos dentro de un nivel de descomposición determinado.
4. Explosionar cuanto sea necesario. El Sistema no va a ser más complicado porque tenga más niveles, todo lo contrario.
5. Descomponer procesos si tienen varios flujos de entrada y/o salida.

6. Nominar correctamente todos los objetos que surjan en los diagramas. Los elementos lingüísticos que se recomiendan para cada tipo de objeto son los siguientes:
- Flujos de datos: un sustantivo más un adjetivo.
 - Procesos: un verbo más un sustantivo.
 - Almacenes de datos: un sustantivo.

II.2 Modelo Entidad-Relación

El modelo Entidad-Relación (E-R) es una de las técnicas más populares y utilizadas dentro del mundo del diseño de aplicaciones, concretamente, en las bases de datos. El diagrama Entidad-Relación se atribuye básicamente a Peter Chen en los comienzos de los años setenta el cual sirvió como base para el desarrollo de modelos de datos. Una manera en que un analista de sistemas puede definir las fronteras adecuadas del Sistema es usando un modelo E-R.

En un modelo E-R se identifican los siguientes conceptos: entidad, relación (*asociación*), cardinalidad, atributos, ocurrencia, identificador y tipos de datos.

Entidad. Significa todo objeto del que se desea guardar información, es decir, se pretende almacenar una serie de datos para su posterior tratamiento y recuperación. Los elementos que conforman un Sistema organizacional pueden ser llamados entidades. Una entidad puede ser una persona, un lugar o una cosa, tal como un pasajero en una línea aérea, un destino o un avión. En forma alterna una entidad puede ser un evento, tal como el fin del mes, un periodo de ventas o la falla de una máquina.

Relación. Una relación significa básicamente cualquier asociación o interdependencia entre varias entidades es decir que describe la interacción entre las entidades. En la figura II.2 vemos un ejemplo donde las entidades son Cliente, Pedido, Línea, Producto, Tipo de cliente, etc. Ejemplo de relaciones son solicita, contiene, hace referencia a, etc.

Las **cardinalidades** significan las reglas de negocio por las cuales las entidades existen en el modelo; es decir, definen la manera de asociar entidades.

Los **atributos** son características o propiedades que describen a las entidades. Los atributos toman distintos valores. Estos atributos se definen durante el desarrollo de los diagramas y siempre se realizan de forma iterativa; es decir, es un proceso cíclico donde todos los atributos se definen después de varios estudios. A veces, algunos atributos se construyen por propia necesidad en el modelo para poder identificar valores de formas ordenadas. Ejemplo de este tipo de atributos son Código cliente, Dirección cliente, Número cliente, etc.

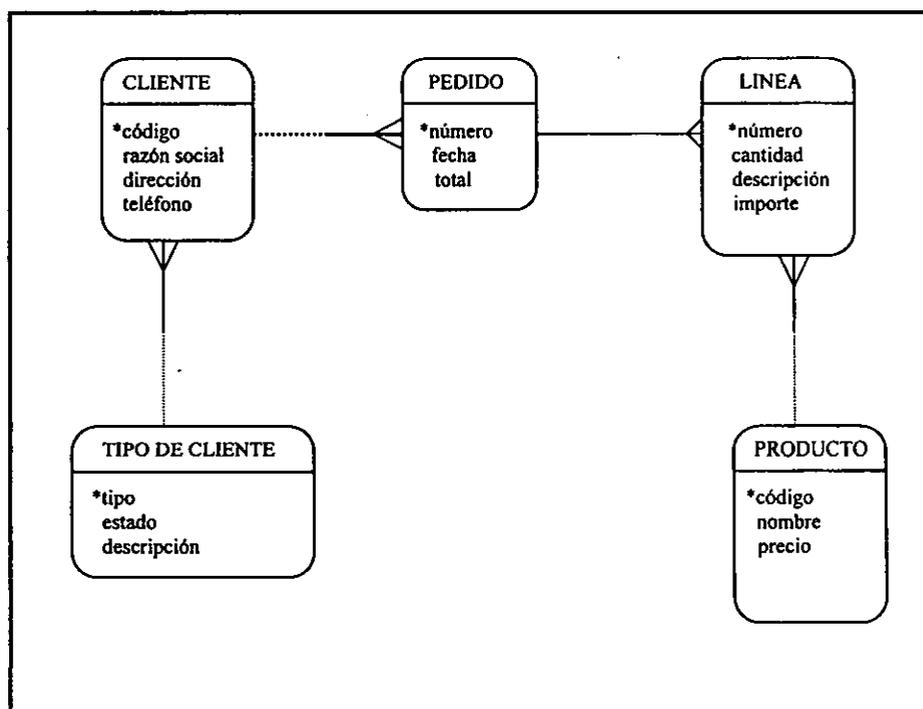


Figura II.2 Ejemplo de diagrama Entidad-Relación.

Una **ocurrencia** de una entidad es un dato definido siguiendo el modelo diseñado. Es decir, para el ejemplo especificado en el diagrama adjunto, las ocurrencias son los distintos valores que puede tomar, por ejemplo, la entidad cliente, como Banco Popular, etc.

Un **identificador** (*clave*) es un atributo que reúne una serie de condiciones y que tiene la utilidad de poder definir una ocurrencia de manera única. Un identificador debe tener un solo valor, que no se puede repetir y no puede ser nulo. En el caso de Cliente, existe el atributo de Código de cliente, el cual sería el candidato ideal para poder ser definido como identificador, y tomaría un valor distinto para cada cliente, y por ello sería único.

Los **tipos de datos** son la manera en la que los atributos se representarán de forma física en el diseño de las bases de datos. Estas formas físicas son propias del entorno sobre el cual se están desarrollando, y por lo tanto son una característica física. Ejemplos pueden ser los tipos numéricos, real, alfanumérico, etc.

La **normalización** es frecuentemente aplicada sobre los modelos E-R para poder diseñarlos de una forma correcta, eliminando toda redundancia que pudiese existir en el diseño de las bases de datos. Normalmente el proceso comienza con el desarrollo de los modelos E-R y se termina aplicando la teoría de la normalización.

II.2.1 Estándares para Entidades

1. Cajas de cualquier dimensión con las esquinas redondeadas.
2. Nombre único para cada entidad.
3. Nombre de la entidad en mayúsculas.
4. Nombre de sinónimo (*nombre alternativo para una entidad*) opcional (*entre paréntesis*).
5. Nombre de los atributos en minúsculas.
6. Si una entidad no puede tener un identificador único (*UID*) ésta no puede ser una entidad.

II.2.2 Estándares de Atributos

Los atributos describen una entidad para calificar, identificar, clasificar, cuantificar o expresar el estado de una entidad.

1. Los nombres de los atributos deben ser claros para un usuario, más no codificado para el desarrollador
2. Los nombres de los atributos deben ser específicos
3. Clarificar siempre la fecha de atributos con una descripción o una frase
4. Los atributos deben de estar en singular y minúsculas
5. Todos los atributos se deben descomponer hasta su mínimo componente con significado
6. Verificar que cada atributo tenga un sólo valor para cada entidad. Un atributo multivalor no es un atributo válido.
7. Verificar que un atributo no sea derivado o calculado de los valores existentes de otros atributos.
8. No incluir atributos derivados en un modelo E-R
 - los atributos derivados son redundantes
 - la redundancia de datos puede ocasionar una inconsistencia de valores de datos. El dato derivado debe ser actualizado cada vez que se modifican los atributos en los que se basa.

II.2.3 Estándares para relaciones

- Una línea entre dos entidades
- Nombre de relaciones en minúsculas
- Opcionalidad
 - Opcional (*Puede ser*)
 - Obligatoria (*Debe ser*)

- Grado

➤ ——— Una o más

————— Una y solamente una

- Una relación es bidireccional y representa la asociación entre dos entidades, o entre una entidad consigo misma

- Sintaxis de una relación:

Cada entidad1 (*debe ser / puede ser*) **Nombre de la relación** (*Una o más / Una y solamente una*) entidad2

II.2.4 Estándares del Diagrama E-R

1. Alinear las cajas de las entidades
2. Dibujar las líneas de relación como rectas horizontales o verticales
3. Hacer todo el texto claro
4. Evitar abreviaciones y modismos (modo de hablar propio de una lengua)
5. Poner los nombres de la relación próximos a su grado y en lados opuestos de la línea
6. Hacer el diagrama E-R fácil de recordar
7. Compactar en la medida de lo posible las cajas de las entidades para ayudar a la visualización del diagrama
8. Dibujar patas de gallo apuntando hacia arriba o a la izquierda
Tratar de posicionar la pata de gallo en la línea en la parte izquierda para las líneas horizontales y en la parte superior para las líneas verticales
9. Posicionar las entidades más volátiles arriba y a la izquierda del diagrama y posicionar las entidades menos volátiles abajo y a la derecha del diagrama

II.2.5 Tipos de Relaciones

En el modelo E-R la conectividad o también llamada relaciones entre entidades tiene gran importancia porque especifica el tipo de asociación entre ocurrencias de las entidades que conforman la relación. Las relaciones entre entidades pueden ser :

- **Relación uno a muchos.**

Una relación es uno a muchos cuando la ocurrencia de la entidad A está relacionada con varias ocurrencias de la entidad B pero la ocurrencia de la entidad B solo puede estar relacionada a lo mucho con una ocurrencia de la entidad A. La figura II.3 muestra una relación uno a muchos.

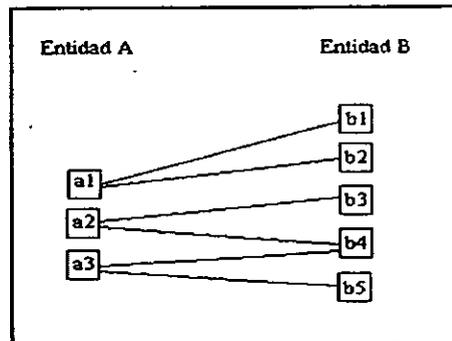
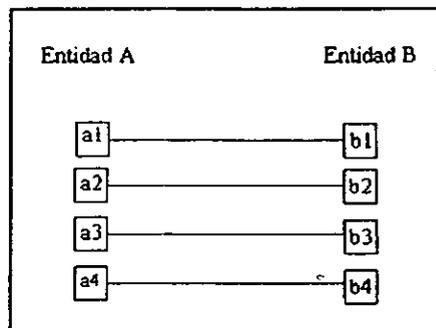


Figura II.3 Representación gráfica de una relación uno a muchos.

- **Relación una a una.**

Este tipo de relación se deriva de la anterior que es un caso particular de uno a muchos donde la ocurrencia de la entidad A se puede asociar solo con una ocurrencia de la entidad B y viceversa. (Figura II.4)

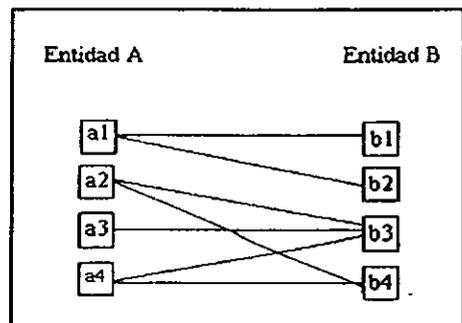
Figura II.4 Representación gráfica de una relación uno a uno.



- **Relación muchos a muchos.**

Esta relación se da cuando se asocian muchas ocurrencias de la entidad A con muchas ocurrencias de la entidad B y viceversa. (Figura II.5)

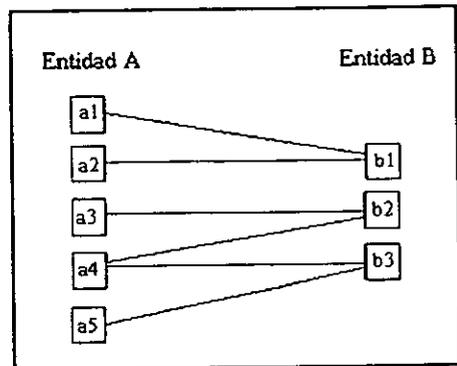
Figura II.5 Representación gráfica de una relación muchos a muchos.



- **Relación muchos a uno**

Esta relación es un caso particular de la anterior ya que las ocurrencias de una entidad A se asocian a lo mas con una sola ocurrencia de la entidad B. (Figura II.6)

Figura II.6 Representación gráfica de una relación muchos a uno.



De estos tipos de relaciones las mas comunes son las relaciones uno a muchos y muchos a uno y por el contrario las relaciones muchos a muchos casi no se manejan.

II.3 Técnica Arriba-Abajo (*Top-Down*)

Esta técnica consiste en un conjunto de diagramas que gráficamente describen las funciones del Sistema de un nivel general a un nivel detallado.

Al principio cada función principal se identifica y luego se subdivide en funciones menores y la suma de estas funciones menores es igual a la función principal.

Los diagramas Top-Down se pueden usar desde el inicio de un proyecto hasta su implantación y son útiles para el mantenimiento de programas.

El principal objetivo de la técnica Top-Down es:

Proveer una estructura por la cual se pueden entender las funciones del Sistema. Los diagramas están organizados en una estructura jerárquica como en un organigrama, donde cada diagrama en cualquier nivel es un subconjunto del nivel de arriba.

II.3.1 Diagrama jerárquico funcional

Este diagrama proporciona una representación del Sistema y la relación jerárquica de sus elementos (*subsistemas, submódulos, procesos, etc.*). Su construcción se realiza con base en el principio de diseño de Arriba-Abajo (*Top-Down*). Cada bloque en el diagrama jerárquico tiene un número para los diagramas de flujo de datos.

II.4 Diagramas de Estructura (Structure Charts)

Los diagramas de estructura son la principal herramienta usada en diseño estructurado para representar la estructura de un Sistema.

La programación estructurada fue la primera técnica que contempló la complejidad de los sistemas modernos. La técnica sugerida por la programación estructurada para hacer una progresión desde la visión general hasta el detalle se conoce como "Top-Down".

Todos los conceptos del diseño "Top-Down" permanecen en el diseño estructurado, pero se añadieron otros principios para poderse obtener una mayor visión de los sistemas.

II.4.1 Elementos principales del diagrama de estructura

- **Módulo:** un módulo es un número de sentencias que hacen una actividad y aquella parte del código que se puede llamar. Un modulo se representa por una caja rectangular con su nombre dentro . El nombre del módulo es una sentencia de su función; es decir, lo que hace cada vez que es llamado. Un modulo predefinido es un módulo que no se tiene que escribir porque ya existe en una librería del Sistema o de la aplicación. Un módulo predefinido se representa gráficamente añadiendo dos líneas paralelas e internas a ambos lados del rectángulo.
- **Conexión Intermodular :** un Sistema está compuesto por módulos organizados jerárquicamente, cooperando y comunicándose entre sí para realizar una tarea.
- **Comunicación intermodular:** los signos de la comunicación son los datos y los flags. Los datos se procesan y están relacionados con el problema: es decir, son importantes para el mundo exterior, mientras que las flags no se procesan, y sólo son importantes para la comunicación de información.
- **Símbolo de llamada iterativa:** ver figura II.6

- **Símbolo de centro de transacción:** ver figura II.6
- **Conectores y continuidad de página:** Nos sirven para evitar el cruce de líneas en el diagrama de estructuras, se utilizan los conectores . Para asegurar la continuidad en el caso de que el diagrama de estructuras ocupe varias páginas se muestra un módulo al pie de una página y repetirlo en la otra, junto con sus subordinados, con una referencia de página.

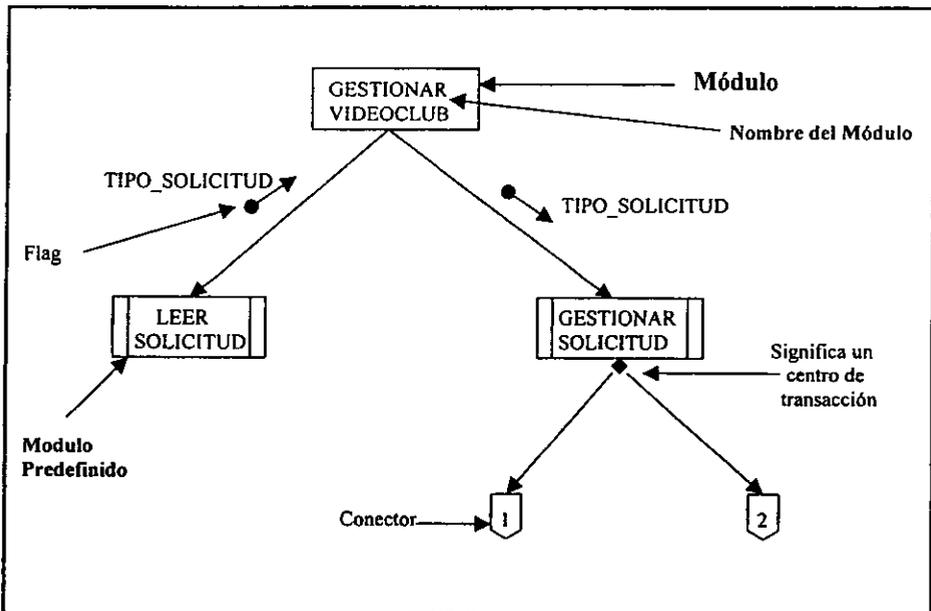


Figura II.6 Ejemplo de un diagrama de estructura

II.5 Matrices

Esta técnica de matrices nos ayuda principalmente a realizar la verificación y validación entre el mundo de los datos y el de los procesos (*matriz evento/entidad*); asimismo, estas técnicas sirven para que se realicen con éxito otras (*matriz entidad/entidad en la realización del modelo E-R*), y para comprender mucho mejor el Sistema en estudio (*matriz papel del usuario/función*).

Este tipo de técnicas puede realizarse y aplicarse sobre cualquier clase de actividad que se realice a lo largo del ciclo de vida del Sistema.

II.5.1 Matriz entidad / entidad

En sistemas pequeños (*unas 12 entidades*), no presenta problemas para identificar las relaciones. Sin embargo, los sistemas grandes pueden incluir varios cientos de entidades con complejas telarañas de relaciones. Una vez identificadas las entidades, una forma de identificar las relaciones entre dichas entidades es usando la matriz entidad/entidad. Cuando existe una relación entre dos entidades, se marca con una "X" la intersección correspondiente.

Entidad	Cliente	Pedido	Producto
Cliente		X	
Pedido			X
Producto			

Figura II.7 Ejemplo de matriz entidad/relación

Cuando una relación entre dos entidades se duplica por lazos que van vía una tercera entidad, entonces no se debe incluir esta relación. Otras veces se indica en la intersección correspondiente el nombre de la relación existente entre las dos entidades.

II.5.2 Matriz Evento / Entidad

Las columnas de la tabla corresponden a todas las entidades identificadas en el modelo lógico de datos. Las filas de la tabla corresponden a todos los eventos que se hayan identificado a partir de los DFD y de las definiciones de funciones (*el nivel mas bajo del DFD muestra los eventos*). Una vez identificadas las entidades y los eventos, se deben relacionar mediante la matriz: se ha de conocer qué efecto (*creación C, modificado M o borrado D*) tiene cada evento sobre las entidades y se ha de marcar en la casilla correspondiente de la matriz. Una vez hecha la primera pasada de la matriz, hay que asegurarse de que cada entidad es creada, de algún modo borrada y es preferiblemente modificada alguna vez. Después hay que asegurarse de que cada evento afecta a la vida de al menos una entidad. Si alguna fila o columna no tiene marca, se ha cometido un error serio en alguna parte. Véase ejemplo en la figura II.8.

Evento \ Entidad	Cliente	Pedido	Producto
1.1 Mantener Cliente	C/M/D		
1.2 Crear producto		C	
1.3 Entrada de pedidos		M	
1.4 Envíos/Facturas		M	
1.5 Archivo		D	
1.6 Mantener producto			C/M/D

Figura II.8 Ejemplo de matriz evento/entidad

II.5.3 Matriz papel del usuario / función

Esta matriz nos muestra qué papel de usuario puede iniciar qué conjunto de funciones. Cada intersección en la matriz representa un diálogo, y por tanto cada fila identifica todos los diálogos que necesita un usuario dado. Esta matriz se debe de verificar con los usuarios para asegurarse de que se han identificado todos los diálogos que se quieren y que ninguno es vano. Ver ejemplo siguiente.

Papel del usuario \ Función	Añadir reserva	Reservar sala	Reemplazar delegados	Reestructurar reserva	Asignar tutor	Crear curso nuevo
Responsable del curso						X
Encargado de reservas	X		X	X		
Encargado de planificación		X			X	

Figura II.9 *Ejemplo de matriz usuario / función*

II.6 Herramientas CASE

Las herramientas para el desarrollo de Software ha sufrido varios cambios y hoy en día existen herramientas de diseño que nos permiten automatizar varias de las etapas del ciclo de vida y que se pueden utilizar en una PC. Aunque muchas de estas herramientas se ocupan en el análisis y diseño del Software, los profesionales del desarrollo pueden crear sistemas de Software con la ayuda de dichas herramientas.

Estas herramientas han venido a cambiar la manera de resolver problemas que desde hace varios años aquejaban a la productividad. Esta nueva tecnología es la llamada Ingeniería de Software Asistida por Computadora o comúnmente conocida como tecnología CASE (*Computer Aided Software Engineering*). Definida de una sencilla manera, la Ingeniería del Software asistida por computadora es la automatización del Software.

La CASE nos propone una nueva formulación del concepto de ciclo de vida del Software basada en la automatización. La idea principal de esta tecnología es proporcionar un conjunto de herramientas bien integradas y que ahorren trabajo, uniendo y automatizando todas o la mayor parte de las fases del ciclo de vida del Software.

Las tecnologías tradicionales del Software son de dos tipos: herramientas y metodologías. La mayoría de estas herramientas son autónomas y basadas en un mainframe (*Ordenador principal*) y dirigidas a la implantación del ciclo de vida del Software.

En lo referido a las metodologías de construcción de sistemas de Software, están incluidas las metodologías de desarrollo manual, como es el análisis estructurado, el diseño estructurado y la programación estructurada. Estas metodologías definen un disciplinado proceso para el desarrollo del Software paso a paso.

La tecnología CASE es una combinación de herramientas de Software y de metodologías. La Case aborda los problemas de la productividad del Software durante todo el ciclo de vida, automatizando muchas de las tareas del análisis y del diseño como también las tareas de la implantación y el mantenimiento de los programas.

La CASE hace prácticas las metodologías estructuradas manuales al automatizar el dibujo de diagramas estructurados y la generación de la documentación del Sistema.

El verdadero soporte de la tecnología CASE es una generación de herramientas de Software. Algunas de estas herramientas están hechas para trabajar en microcomputadoras o PC's, cuentan con grandes entornos gráficos que hacen más amigable la interfaz con el usuario y se incorporan fácilmente a bases de trabajo de donde pueden ser llamadas.

Las herramientas CASE (*CASE tools*) se enfocan hacia la productividad individual del profesional del desarrollo de Software. Con el uso de las estaciones de trabajo, de las redes de área local y de las herramientas basadas en las PC's, el profesional de desarrollo de Software puede trabajar en un entorno de desarrollo de Software fiable y dedicado.

Las herramientas CASE son utilizadas principalmente por los desarrolladores de Software y no por los usuarios finales, son además una tecnología de Software de propósito general para utilizarse en el desarrollo de todo tipo de sistemas (*grandes y pequeños, comerciales y científicos, en línea y de tiempo real, etc.*).

En general, las herramientas CASE se diferencian de otras herramientas de Software por muchas cosas importantes. Algunas de estas cosas son que las herramientas CASE fueron diseñadas para:

- Soportar un entorno personal dedicado.
- Utilizar gráficos para especificar y documentar los sistemas.

- Unir todas las fases del ciclo de vida del Software.
- Capturar y unir en la computadora personal toda la información sobre el entorno del Software de un Sistema, desde los requerimientos iniciales hasta las actividades para una mantenimiento constante.
- Utilizar la inteligencia artificial para realizar automáticamente muchas de las rutinarias tareas del desarrollo y mantenimiento del Software.

II.6.1 Ejemplos de Herramientas CASE

Las herramientas CASE realizan muchos más tipos de tareas que las herramientas de la tercera y cuarta generación, soportan la automatización de una gran variedad de tareas de desarrollo y mantenimiento del Software, incluyendo la gestión del proyecto, como se muestra en la figura II.10. Los ejemplos de herramientas CASE incluyen:

- Herramientas de diagramación para el dibujo de diagramas estructurados.
- Pantallas de informes gráficos para la creación de las especificaciones del Sistema y de formularios de prototipos simples.
- Diccionarios, sistemas de gestión de bases de datos y utilidades para almacenamiento de información y consulta de la información técnica.
- Herramientas de validación de especificaciones para la detección automática de las que son incompletas, incorrectas sintácticamente o inconsistentes.
- Generadores de código que nos proporciona el código ejecutable a partir de las especificaciones obtenidas gráficamente.

- Generadores de documentación para producir la documentación técnica y del usuario requeridas por las técnicas estructuradas.

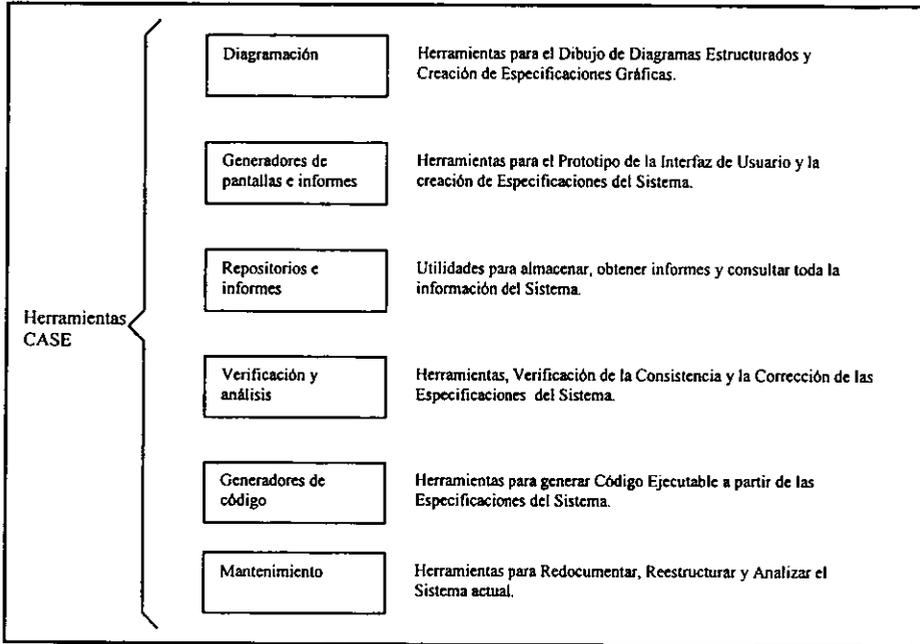
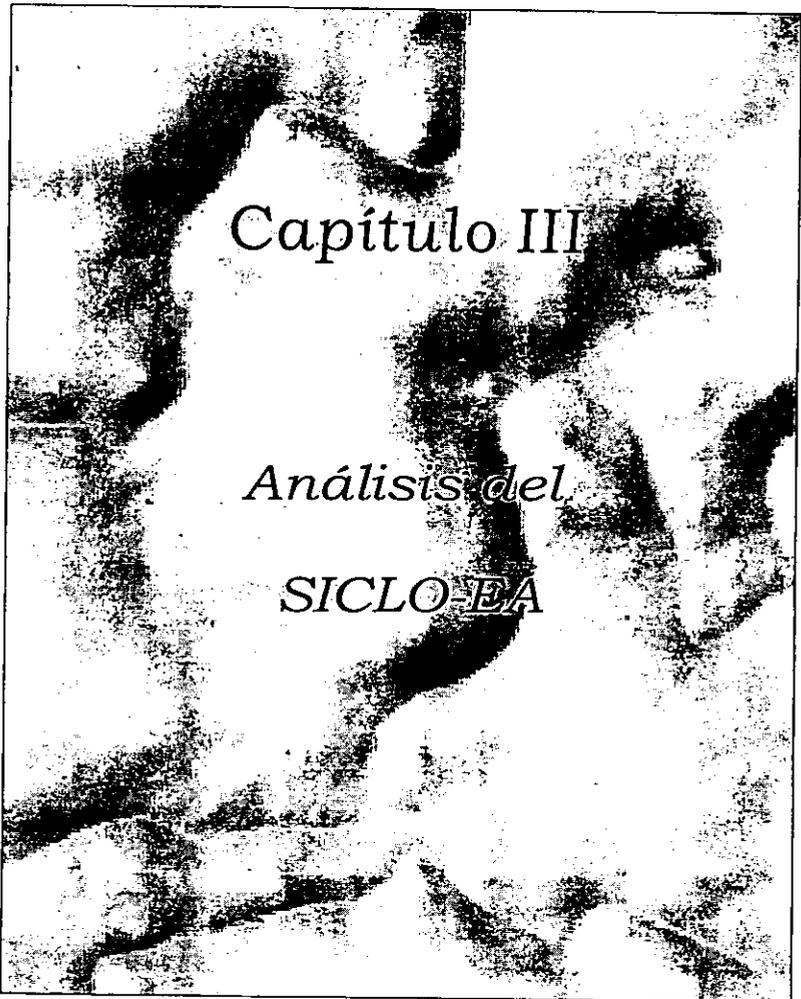


Figura II.10 *Herramientas CASE que cubren muchos de los tipos de las tareas de desarrollo.*

Aunque las herramientas CASE son una parte muy importante de CASE, la tecnología CASE consta de mucho más. La CASE es una redefinición del entorno completo del Software.



Capítulo III

Análisis del

SICLO-EA

ANÁLISIS DEL SISTEMA

Introducción

Durante esta fase del proyecto se llevará a cabo un análisis de las necesidades de la División de Estudios de Posgrado de la Facultad de Ingeniería (DEPFI) y la planificación, en la cual una de las actividades principales es la definición del problema.

El producto de esta etapa es la definición del sistema. En la cual se plantea el problema así como las diversas alternativas de solución, eligiendo y justificando la más viable.

También se incluye la justificación del Sistema, los objetivos de dicho Sistema, los alcances, los beneficios que se proporcionarán, los riesgos, requerimientos del usuario y las herramientas y técnicas que se emplearán.

III.1 Antecedentes

La División de Estudios de Posgrado de la Facultad de Ingeniería (*DEPFI*) queda formalmente establecida en 1957, con el fin de satisfacer las necesidades de educación superior a la Licenciatura, en diversas ramas de la Ingeniería, para formar profesionales especializados, profesores e investigadores.

La actividad de la **DEPFI** orientada a su proceso de enseñanza - aprendizaje hacia la docencia e investigación, de acuerdo a las necesidades científicas y tecnológicas que demanda la sociedad. De esta forma es posible integrar grupos de profesores e investigadores con alumnos que realicen trabajos originales y de alta calidad.

III.2 Necesidades de la DEPFI

La División de Estudios de Posgrado de la Facultad de Ingeniería, organiza por lo menos una vez al año un seminario o congreso, por lo cual se tiene la necesidad de crear un sistema, el cual le permita tener la información en un solo sistema, ya que la información que se tiene esta en bases de datos las cuales son utilizadas en Microsoft Word para hacer cartas invitación o los remitentes para los sobres, la información que se tiene es de diferentes eventos que ha tenido la DEPFI por lo cual el sistema deberá contemplar un almacenamiento cronológico de la información, y que toda esa información la utilice el sistema para que este saque cartas invitación, etiquetas para el remitente para un nuevo evento pero que contemple todas las personas que han asistido a eventos anteriores.

III.3 Definición del problema

Debido a que en la División de Estudios de Posgrado de la Facultad de Ingeniería no existe un área que se encargue de la administración de eventos académicos surge la necesidad de desarrollar un sistema de cómputo que sirva como una herramienta para la organización de dichos eventos.

III.4 Justificación del sistema

La División de Estudios de Posgrado de la Facultad de Ingeniería, a través del Laboratorio de Sistemas de Información que pertenece a dicha división, con el fin de hacer frente eficiente a la organización de los eventos académicos y por la información que se maneja, tomaron la decisión de utilizar un sistema automatizado, que permitiera apoyar la administración de dichos eventos en forma rápida y eficiente, aprovechando para ello los recursos existentes.

III.5 Objetivos

- Desarrollar un sistema para la administración y logística de eventos académicos.
- Simplificar y agilizar la administración y logística en la organización de eventos académicos.
- Generar informes confiables y en forma oportuna para que puedan ser analizados por el personal del comité organizador.
- Desarrollar un sistema que nos permita tener un almacenamiento cronológico en la información.
- Desarrollar un sistema con los atributos de calidad requeridos: confiabilidad, portabilidad, eficiencia, solidez y corrección.

III.6 Alcances

El sistema, desde su diseño, estará enfocado a facilitar la administración y logística de eventos académicos, este Sistema para el Control y Logística de Eventos Académicos llamado (*SICLO-EA*) tendrá el alcance de generar información confiable y completa de los datos almacenados en dicho sistema, además de solucionar el mínimo problema de forma fácil y rápida.

III.7 Requerimientos del Usuario

Un requerimiento es una característica que debe incluirse en un nuevo sistema. Esta puede ser la inclusión de determinada forma para capturar o procesar datos, producir información o controlar una actividad de la empresa. Es así como la determinación de requerimientos vincula el estudio de un sistema existente con la recopilación de detalles relacionados con él.

Dado que los analistas de sistemas no trabajan en los departamentos de usuarios, no tienen los mismos conocimientos, hechos y detalles que los usuarios y gerentes de esas áreas. Por lo tanto el primer paso de analista es comprender la situación.

Dar respuesta a un grupo específico de preguntas, será de gran ayuda para comprender los requerimientos básicos. También existe otra clase de requerimientos, si el sistema está orientado hacia transacciones, toma de decisiones o si se extiende a varios departamentos.

Hace muchos años, el uso de los sistemas era muy completo ya que requería que el usuario fuese un experto en el área para poder seguir los pasos para la ejecución de ciertos programas, esta ejecución exigía aprender varias instrucciones muy complejas las cuales no permitían entender exactamente lo que se hacía. Con el tiempo los desarrolladores de sistemas han ido cambiando la filosofía de uso de las aplicaciones creando pantallas y menús más sencillos sin tener que memorizar instrucciones.

Con la introducción de las computadoras personales, el desarrollo de sistemas tomó un giro importante ya que ahora el usuario no necesita ser un experto en sistemas, sino todo lo contrario, el usuario es una persona común que no tiene idea de lo que es un sistema de cómputo y sólo se limita a hacer uso de la computadora para resolver un problema específico y que se podría pensar que no está muy interesado en conocer lo que sucede a nivel software.

La competencia en el desarrollo de software se fue incrementando cada vez más, esto significa tener que captar mayor número de usuarios. Esta competencia fue benéfica, ya que exigía mayor facilidad de uso y sencillez para el usuario. Todos los cambios que han surgido dentro del diseño de los sistemas han proporcionado una mejor interactividad entre el usuario y el sistema de cómputo.

Antes de iniciar con el desarrollo del sistema SICLO-EA, se recurrió directamente con el personal encargado de la información y estos fueron sus requerimientos:

- Para las personas que se inscribieran al seminario es necesario conocer datos de carácter personal como domicilio, teléfono, email, edad, nacionalidad, profesión, empresa donde trabaja, para poder tener una relación para posteriores eventos.
- Es necesario además, hacer una distinción entre las personas esto es, invitado especial, comité organizador, ponente, asistente, acompañante, becario, ponente invitado, ponente extranjero.
- Poder tener una relación de las personas que estaban invitadas al seminario y cuantas se registraron en el evento para poder hacer una estimación de ingresos.
- De la información capturada tener como salida del sistema datos estadísticos, que le permitiera a los organizadores hacer un análisis del evento.
- El sistema debe permitir a su salida cartas de invitación para nuevos eventos.

- Además de todo lo mencionado se requiere que:
- El sistema sea amigable y de fácil manejo, significando con esto, que no deberá causar problemas para el uso del mismo.
- El sistema debe tener un acceso sencillo sin hacerle demasiadas preguntas al usuario, salvo las necesarias para poder operarlo en forma concisa, ya que algunos programas tienen la particularidad de tener menús demasiado complejos y aunque se disponga de ayuda en pantalla, ésta no debe utilizarse para cada paso a ejecutar.
- El sistema deberá tener la capacidad de un mantenimiento sencillo para futuras actualizaciones o mejoras. Esto proporcionará al usuario información actualizada y veraz.
- La ayuda proporcionada por el sistema en forma inmediata, deberá ofrecer una gran seguridad al usuario y evitar así que se distraiga en busca de manuales demasiado grandes y complejos.
- Las pantallas deben ser legibles y sencillas de comprender.
- El sistema debe ser seguro, es decir, que se pueda controlar el acceso a la información por personal no autorizado.

III.8 Beneficios que se proporcionarán

- Control de la información generada en los módulos considerados en el diseño del sistema.
- Generación oportuna de reportes que apoyen la toma de decisiones.
- Contar con un nuevo sistema que no requiera de recursos de cómputo adicionales para su desarrollo.
- Simplificación de los procesos administrativos.
- Manejo de información confiable y oportuna que fortalezca la toma de decisiones y apoye la operatividad interna de DEPFI.
- Desarrollo de un sistema que permita satisfacer las necesidades específicas de la DEPFI.
- Implantación del sistema en otras Facultades de la U.N.A.M.

III.9 Riesgos

Con base en la experiencia obtenida en el desarrollo de sistemas, se ha observado que los riesgos que se corren son:

- Coordinación insuficiente de los usuarios para darle continuidad a la comunicación que se establecerá mediante el sistema.
- Inadecuada operación por resistencia al cambio.

- Falta de interés y colaboración por parte de los usuarios para su implementación.
- Prolongación en el tiempo estimado para su desarrollo debido a cambios imprevistos tanto internos como externos.

III.10 Herramientas y técnicas que se emplearán

Para el desarrollo del sistema nos auxiliaremos de diversas técnicas dentro de las cuales cabe destacar el modelo *entidad – relación*, el cual nos permitirá obtener un modelo conceptual de los datos mediante esquemas de entidades y relaciones del sistema, la técnica de diseño arriba - *hacia – abajo* (*Top – Down*).

El sistema se desarrollará en Microsoft ACCESS 97 ya que es una herramienta que nos ayuda a tener un ambiente gráfico como el que se tiene en WINDOWS, con este ambiente gráfico Microsoft ACCESS nos permite darle a la base de datos una mejor presentación para que el usuario final no tenga conflicto con el sistema ya que este ambiente hace que el sistema sea amigable y el usuario no tenga que estar sufriendo cuando esta navegando en el sistema, además de que ACCESS también permite tener seguridad en el sistema ya que el sistema se creo con la finalidad de que trabaje en red, por lo tanto el sistema debe tener un código de seguridad confiable.

III.10.1 Modelo Entidad - Relación

EL modelo entidad – relación se ha usado en todo el mundo, como una herramienta de computación entre los analistas, diseñadores de sistemas y los usuarios finales, durante las fases de análisis de requerimientos y de diseño conceptual, debido a que es simple y fácil de entender.

Los enfoques convencionales para el diseño lógico de base de datos, normalmente tienen solo una fase: convertir la información de los objetos del mundo real, directamente al esquema del usuario.

Esta metodología para el diseño lógico de base de datos consiste de dos fases principales:

- Definir el esquema conceptual usando diagramas entidad – relación.
- Traducir el esquema conceptual al esquema del usuario.

Las ventajas de esta técnica son:

- La división de las funciones y el trabajo hace que el diseño de la base de datos sea más simple y más organizado.
- El esquema conceptual es fácil de diseñar ya que no está restringido por las capacidades del DBMS (Data Base Management System) y es independiente del esquema físico y de consideraciones de eficiencia.
- El esquema conceptual es más estable que el esquema del usuario. Si se desea cambiar un DBMS a otro, el esquema conceptual será el mismo, ya que éste es independiente del DBMS que se use.
- El esquema conceptual expresado por un diagrama entidad – relación es más fácil de entender por gente, sin conocimientos en computación.

III.10.2 Técnica Top – Down

Esta técnica consiste en un conjunto de diagramas que gráficamente describen las funciones del sistema de un nivel general a un nivel detallado.

Al principio cada función principal se identifica y luego se subdivide en funciones menores. La suma de esas funciones menores es igual a la función principal.

Los diagramas *Top – Down* se pueden usar desde el inicio de un proyecto hasta su implantación y son útiles para el mantenimiento de programas.

El objetivo de la técnica *Top – Down* es:

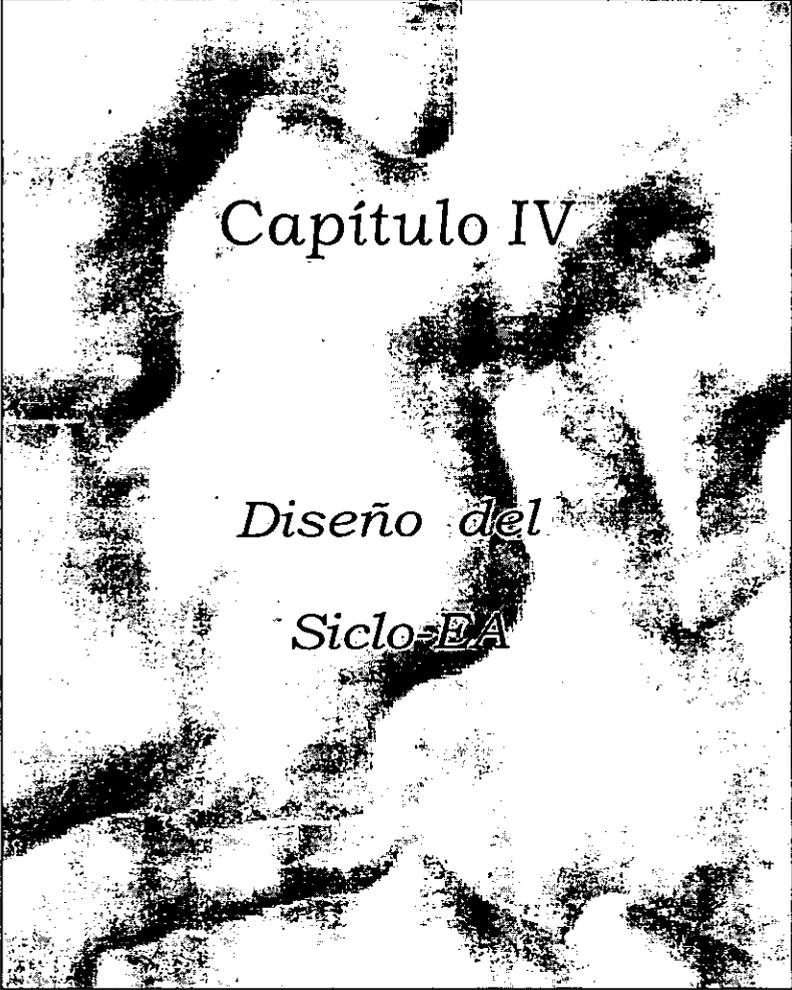
Proveer una estructura por la cual se puedan entender las funciones del sistema. Los diagramas están organizados en una estructura jerárquica como en un organigrama, donde cada diagrama en cualquier nivel es un subconjunto de nivel de arriba.

III.11 Planificación del desarrollo mediante Gráficas de Gantt

Para llevar a cabo un buen proyecto de desarrollo de Software, debemos comprender el ámbito de trabajo a realizar, las tareas a ejecutar y la agenda a seguir. Esta última define los tiempos en los cuales recaerá la duración del proyecto.

El apoyo que proporcionan las técnicas gráficas para la planificación de cualquier desarrollo es indudablemente de gran utilidad. Actualmente existen diversas técnicas gráficas que permiten facilitar éstas tareas, como por ejemplo, las Gráficas de Gantt, Redes Pert, etc.

Con el fin de planificar y controlar adecuadamente las actividades que se llevarán a cabo durante el desarrollo del SICLO-EA, nos apoyaremos en la técnica de Gráficas de Gantt.



Capítulo IV

Diseño del

Siclo-~~EA~~

DISEÑO DEL SICLO-EA

Introducción

Esta fase del proyecto la hemos dividido en dos partes o subfases: el diseño estructural y el detallado. En el primero se identifican los componentes de la programación, su descomposición en módulos de procesamiento y estructuras de datos conceptuales, y la especificación de interconexión entre componentes. El segundo se refiere a detalles de cómo definir tabla, códigos y parámetros.

El diseño es la forma en la cual podemos traducir con precisión los requisitos del cliente en un producto o Sistema acabado.

El diseño de Software sirve como base de todas las posteriores etapas de desarrollo y de la fase de mantenimiento. Sin diseño nos arriesgamos a construir un Sistema inestable, un Sistema que falle cuando se realicen pequeños cambios, un Sistema difícil de probar, un Sistema cuya calidad no pueda ser evaluada hasta más adelante, cuando quede poco tiempo y se haya gastado ya mucho dinero.

IV.1 Concepto del Diseño de Software

El diseño es el primer paso de la fase de desarrollo de cualquier producto o Sistema de ingeniería. Una definición acertada del diseño de Software es la siguiente: *“... el proceso de aplicar distintas técnicas y principios con el propósito de definir un dispositivo, proceso o Sistema con los suficientes detalles como para permitir su realización física”*

El proceso de diseño traduce los requerimientos a una representación donde el Software puede evaluarse para conocer el desempeño del mismo antes que la codificación se realice. Desde un punto de vista de la gestión del proyecto, el diseño del Software puede realizarse en dos pasos el diseño *preliminar* y el diseño *detallado*.

IV.2 Diseño estructural

En el diseño estructural, el Sistema se divide en subsistemas, módulos y submódulos, mediante la identificación de todas sus funciones bajo una estructura jerarquizada. Esta es una filosofía básica aplicable a cualquier clase de problemas, dividir el problema total en subdivisiones que pueden ser manejados en forma separada. Las ventajas del diseño estructural y modular son las siguientes:

- El trabajo tiende a ser clasificado de lo más complejo a lo menos complejo, en forma jerarquizada y controlada.
- La documentación es fácil de crear y mantener.
- El mantenimiento de los sistemas se facilita. Este es un factor importante para reducir los tiempos de mantenimiento y sus implicaciones en el resto de las fases del ciclo de vida.
- Las pruebas del Sistema son fáciles y normalmente más comprensivas.

- Se facilita la integración con otros sistemas.
- Se identifican claramente los módulos de propósito general desarrollados y utilizados por dos o más sistemas diferentes.

El diseño estructurado se centra en la transformación de los requisitos en los datos y la arquitectura del Software. Comprende la identificación de los componentes de la programación, su desacoplamiento y descomposición en módulos de procesamiento y estructuras de datos conceptuales.

IV.2.1 Diagrama Jerárquico Funcional General

La figura IV.1 muestra el diagrama jerárquico general del SICLO-EA. En este se observan los diferentes módulos que se contemplan en el diseño del Sistema:

- Parámetros
- Expediente
- Accesos
- Catálogos
- Informes
- Búsqueda

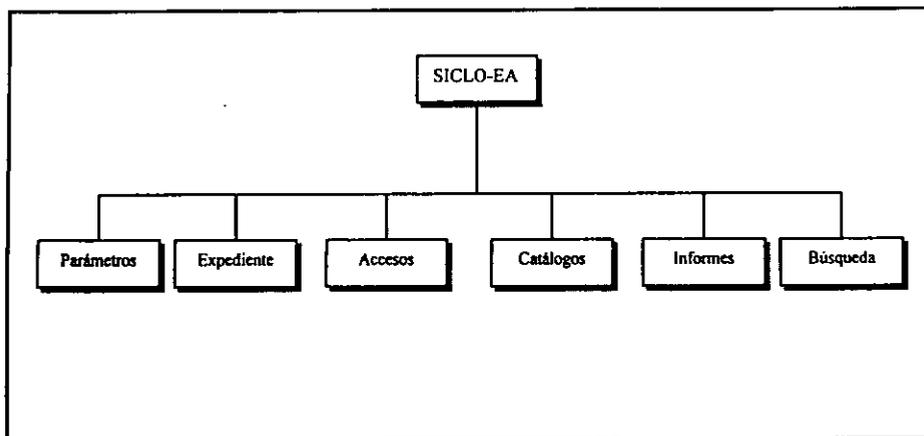


Figura IV.1 Diagrama jerárquico funcional general.

A continuación se presentan los diagramas jerárquicos funcionales para cada módulo del SICLO-EA.

IV.2.2 Parámetros

En la figura IV.2 se muestra el diagrama jerárquico funcional de Parámetros con sus principales submódulos: Nuevo Sistema, Sistema Actual, Sistemas Anteriores.

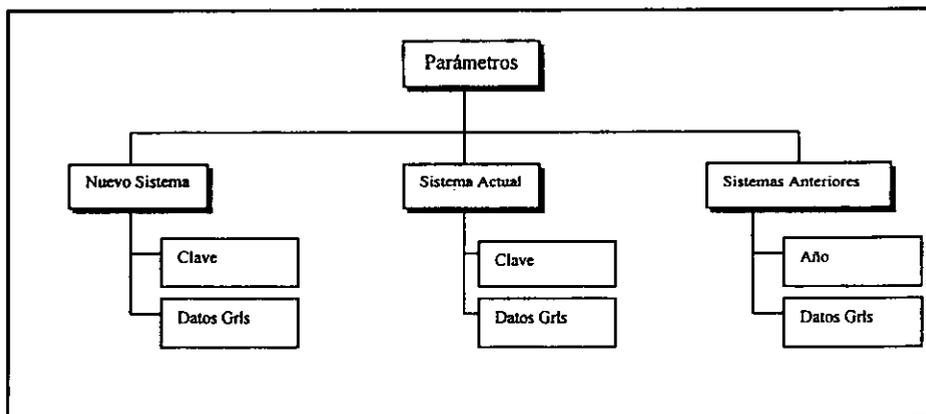


Figura IV.2 Diagrama jerárquico funcional de Parámetros

IV.2.3 Expediente

En la figura IV.3 se muestra el diagrama jerárquico funcional de Expediente con sus principales submódulos: Datos personales y Datos del Sistema.

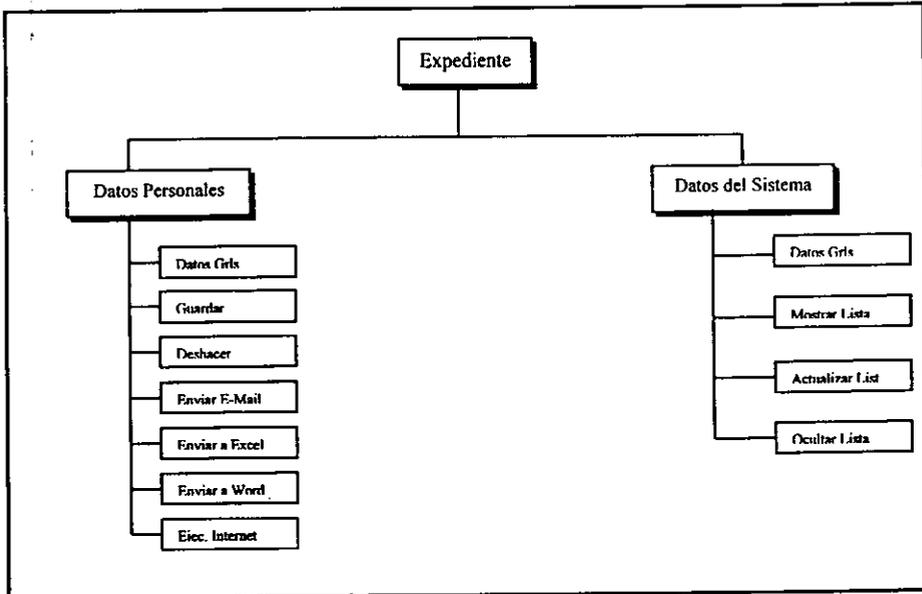


Figura IV.3 Diagrama jerárquico funcional de Expediente

IV.2.4 Accesos

En la figura IV.4 se muestra el diagrama jerárquico funcional de Accesos con sus principales submódulos: Acceso a Word, Acceso a Excel y Acceso a Internet.

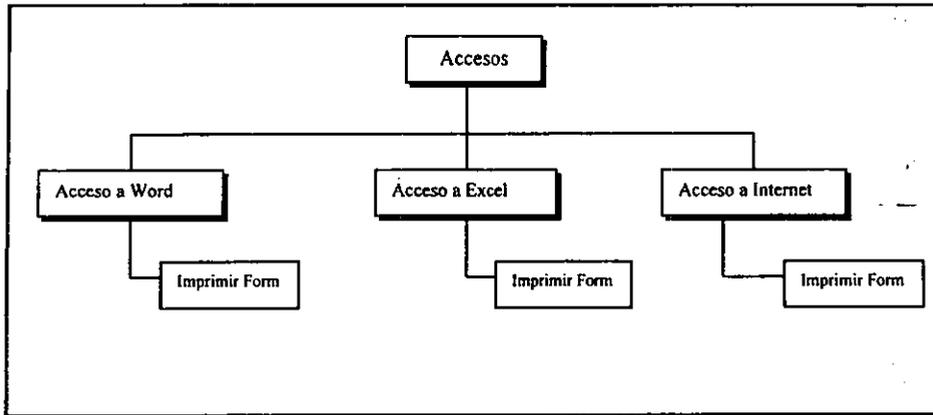


Figura IV.4 Diagrama jerárquico funcional de Accesos.

IV.2.5 Catálogos

En la figura IV.5 se muestra el diagrama jerárquico funcional de Catálogos con sus principales submódulos: Países, Estados, Bancos, Hoteles, Passwords, Universidades, Actividades y Tipos de pagos.

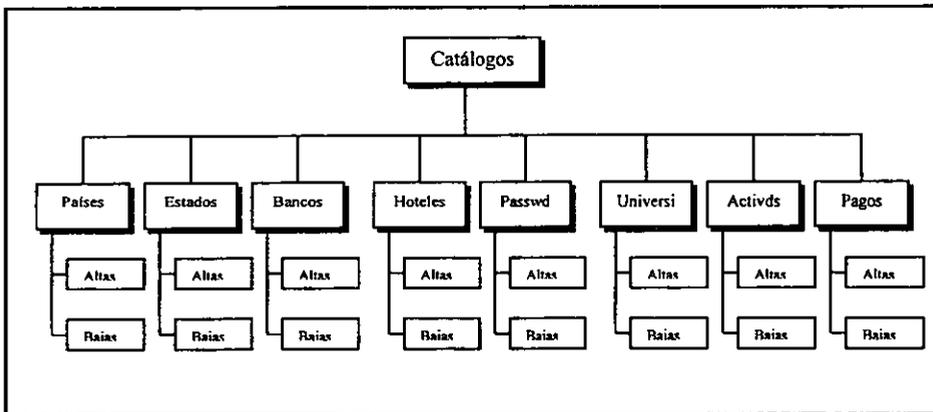


Figura IV.5 Diagrama jerárquico funcional de Catálogos.

IV.2.6 Informes

En la figura IV.6 se muestra el diagrama jerárquico funcional de Informes con sus principales submódulos: Asistencia, Ponentes, Cartas, Empresas, Etiquetas, Gráficas y Varios.

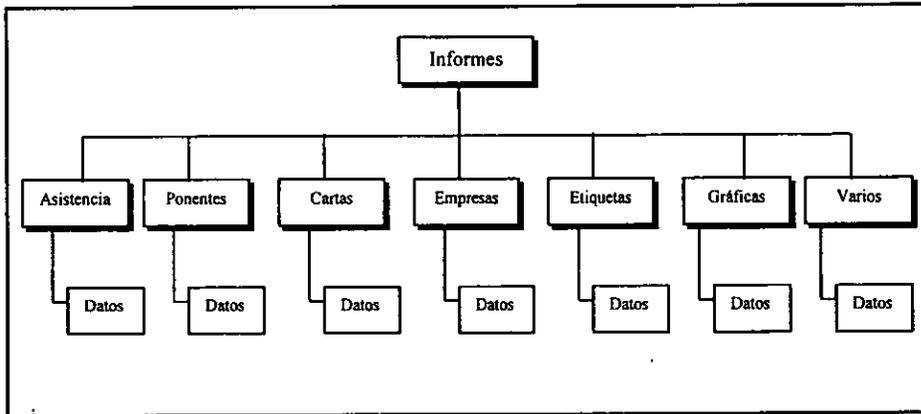


Figura IV.6 Diagrama jerárquico funcional de Informes

IV.2.7 Búsqueda

En la figura IV.7 se muestra el diagrama jerárquico funcional de Búsqueda con sus principales submódulos: Apellido Paterno y Todos.

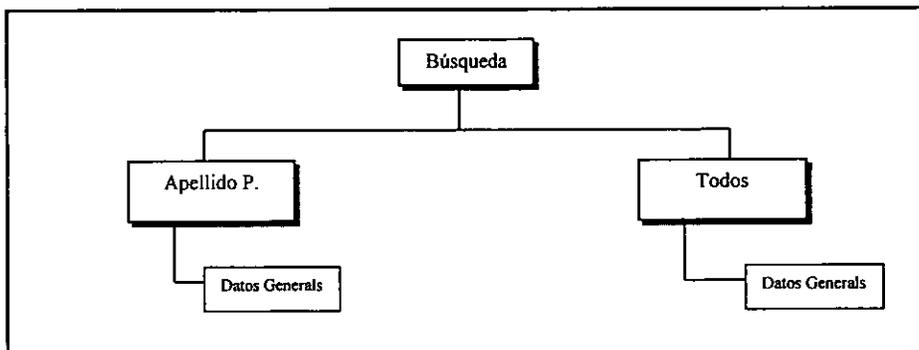


Figura IV.7 Diagrama jerárquico funcional de Búsqueda

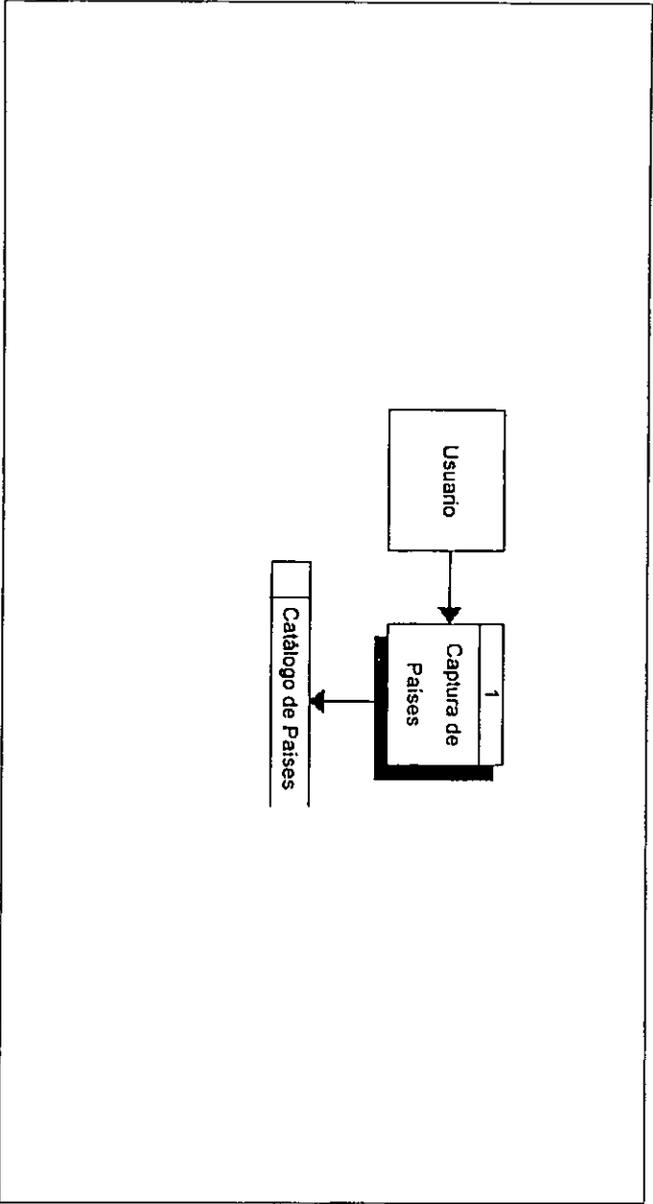
IV.2.8 Diagramas de flujo de Datos

Como ya se dijo el diagrama de flujo de datos es una técnica que tiene como finalidad interpretar gráficamente el flujo de los datos identificando las entradas, archivos y salidas de un Sistema, pasando por todos sus procesos de transformación.

El diagrama de flujo de datos representa un Sistema o Software a cualquier nivel de abstracción. Los DFDs pueden ser refinados a niveles que representan un mayor flujo de información y un mayor detalle funcional. Un DFD de nivel 0 también es denominado modelo fundamental del Sistema o modelo de contexto, y representa al modelo de Software completo como una sola burbuja con datos de entrada y de salida representados por flechas respectivamente. Al partir el DFD de nivel 0 para observar mas detalles, aparecen representados procesos y caminos de flujo de información adicionales.

Los diagramas de flujo de datos generados del SICLO-EA se muestran en las figuras siguientes:

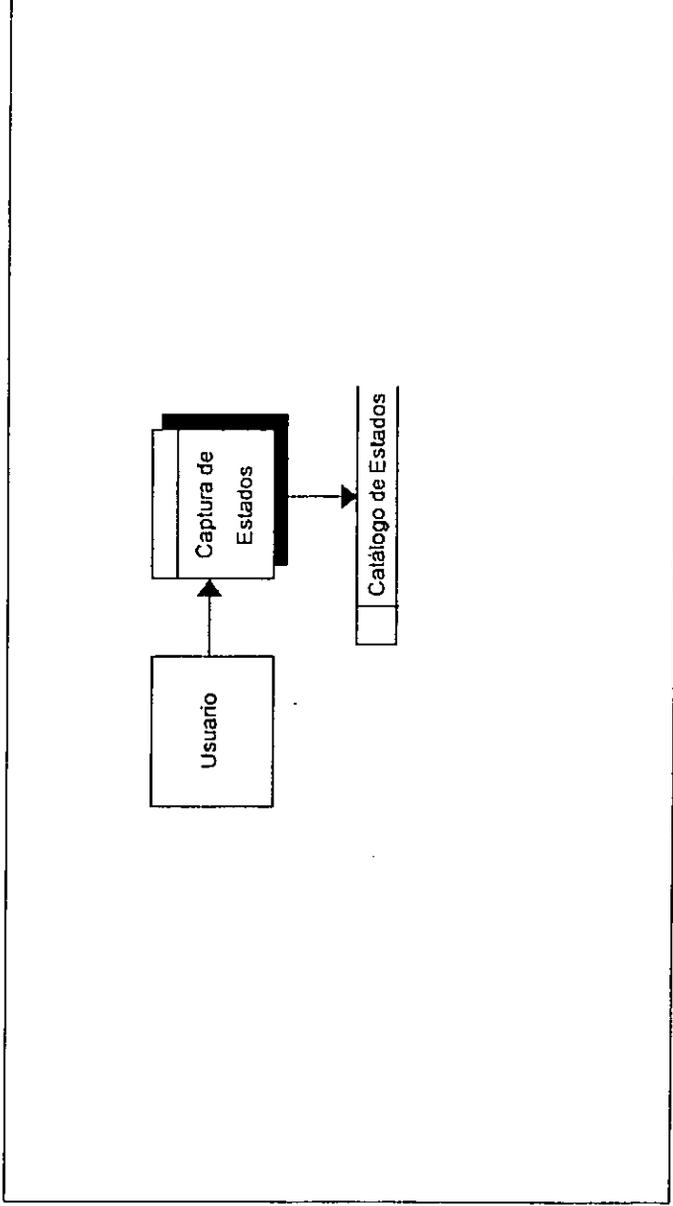
Sistema: Sistema para el Control Logístico de Eventos Académicos
Modulo: Catálogo de Países
Proceso: Captura de Datos



Sistema: Sistema para el Control Logístico de Eventos Académicos

Modulo: Catálogo de Estados

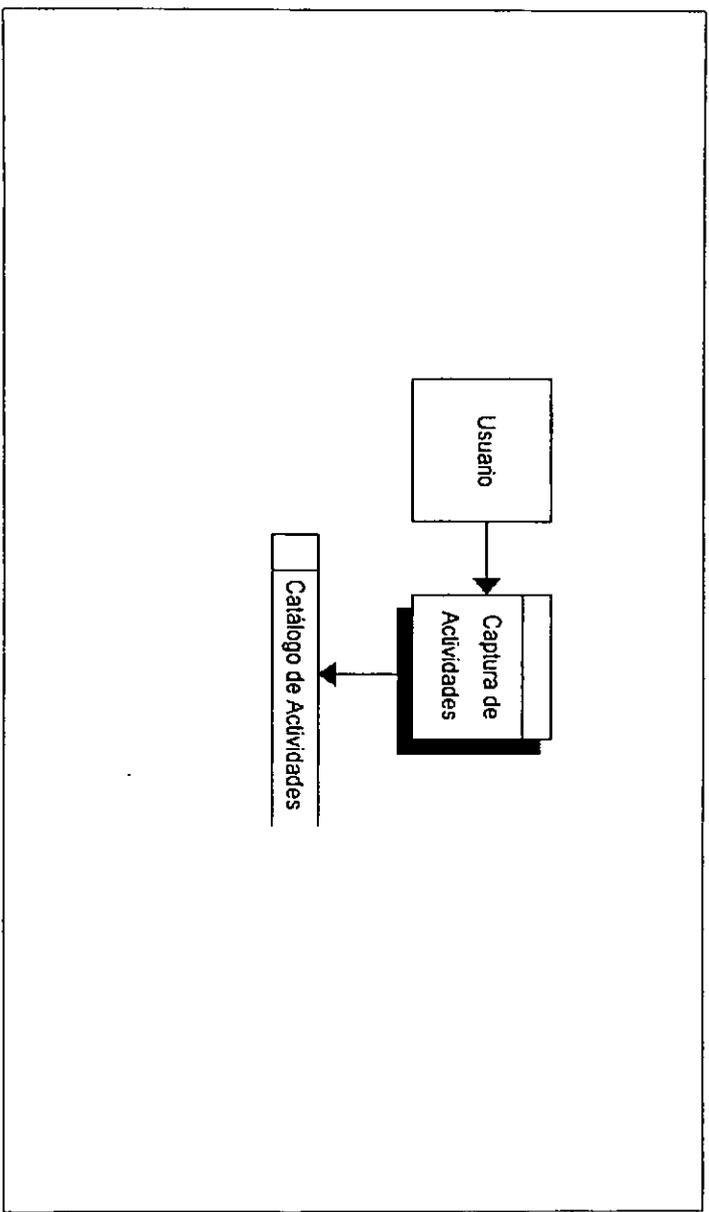
Proceso: Captura de Datos



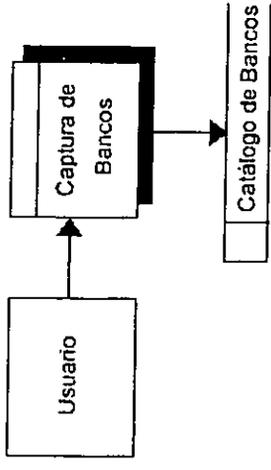
Sistema: Sistema para el Control Logístico de Eventos Académicos

Modulo: Catálogo de Actividades

Proceso: Captura de Datos



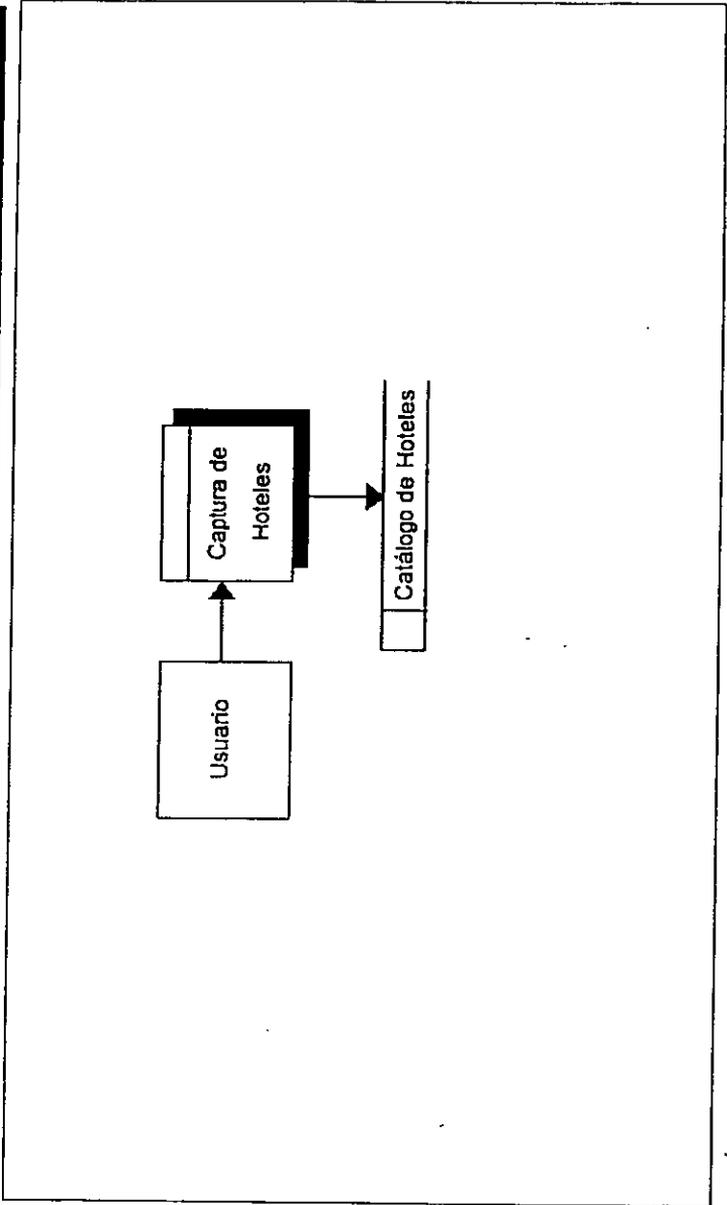
Sistema: Sistema para el Control Logístico de Eventos Académicos
Modulo: Catálogo de Bancos
Proceso: Captura de Datos



Sistema: Sistema para el Control Logístico de Eventos Académicos

Modulo: Catálogo de Hoteles

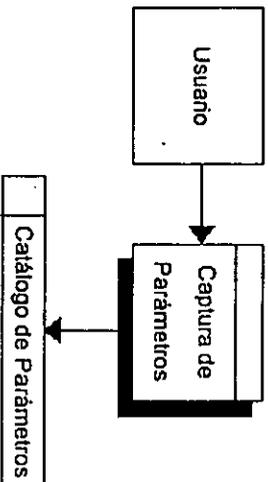
Proceso: Captura de Datos



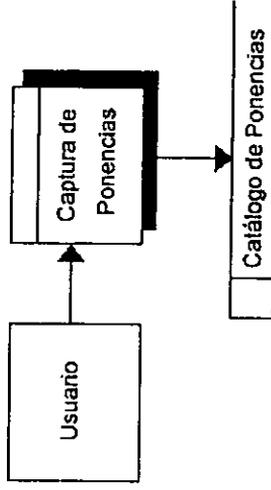
Sistema: Sistema para el Control Logístico de Eventos Académicos

Modulo: Catálogo de Parámetros

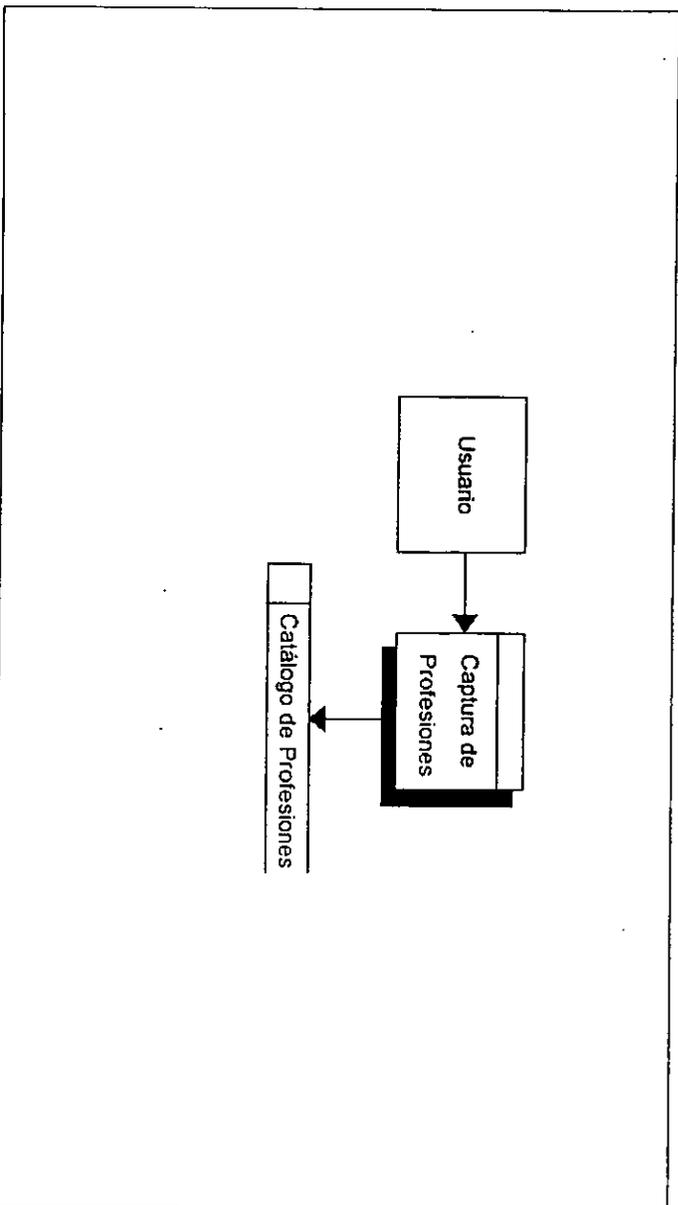
Proceso: Captura de Datos



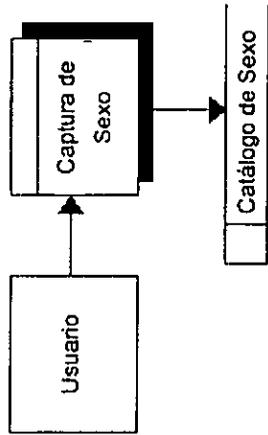
Sistema: Sistema para el Control Logístico de Eventos Académicos
Modulo: Catálogo de Ponencias
Proceso: Captura de Datos



Sistema: Sistema para el Control Logístico de Eventos Académicos
Modulo: Catálogo de Profesiones
Proceso: Captura de Datos



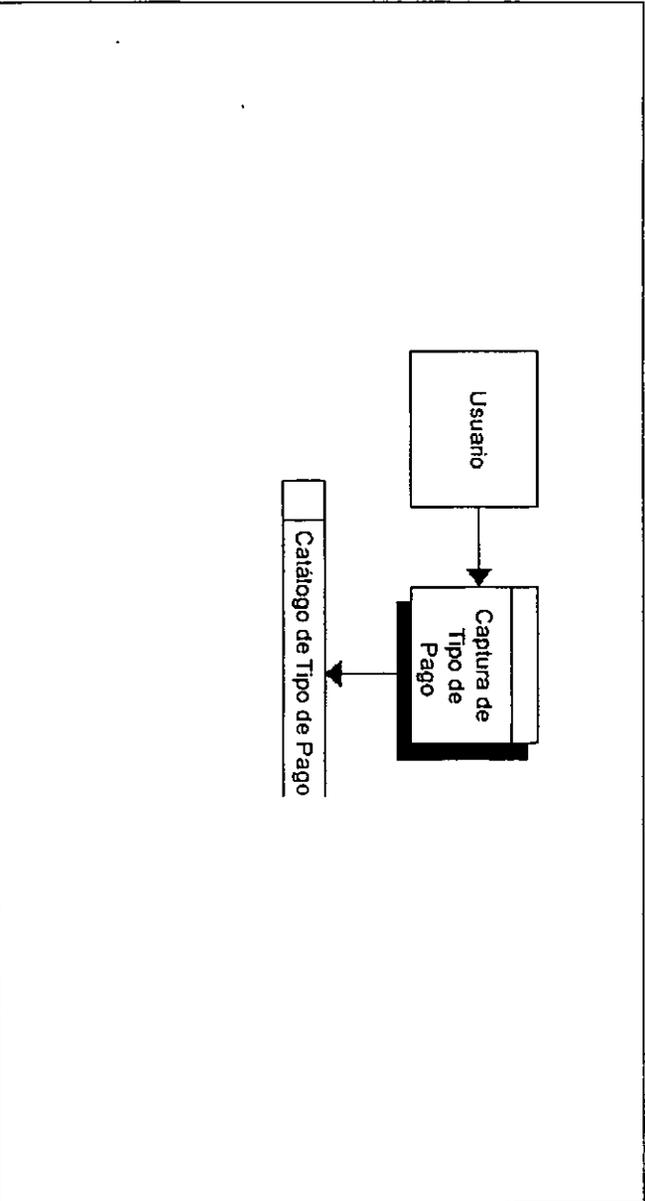
Sistema: Sistema para el Control Logístico de Eventos Académicos
Módulo: Catálogo de Sexo (Sexualidad)
Proceso: Captura de Datos



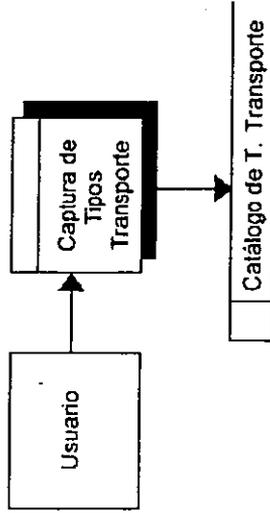
Sistema: Sistema para el Control Logístico de Eventos Académicos

Modulo: Catálogo de Tipo de Pago

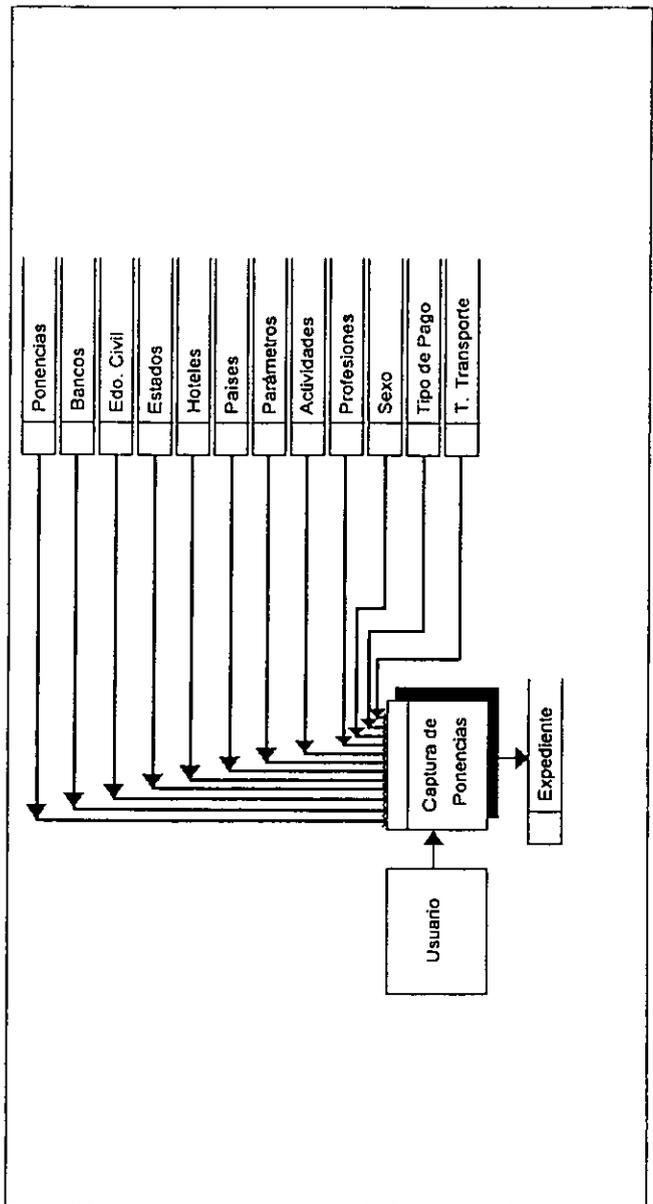
Proceso: Captura de Datos



Sistema: Sistema para el Control Logístico de Eventos Académicos
Modulo: Catálogo de Tipos de Transporte
Proceso: Captura de Datos



Sistema: Sistema para el Control Logístico de Eventos Académicos
Modulo: Expediente
Proceso: Captura de Asistentes



IV.2.9 Diagrama Entidad Relación

Antes de diseñar formalmente los archivos del nuevo Sistema, es importante conocer la relación que tienen sus entidades entre sí, mediante la definición del modelo Entidad-Relación de los archivos, para lo cual se requiere hacer lo siguiente:

1. Relacionar cada uno de los archivos que previamente se identificaron en el diagrama de flujo de datos (*Entidades*).
2. Definir los elementos de datos principales que describen a los registros de cada archivo (*Campos*).
3. Identificar al elemento de datos que por su valor diferencia a un registro de otro (*Clave de registro*).
4. Graficar la relación de los Campos de todas las Entidades o archivos del nuevo Sistema.
5. Una vez que se han descubierto las entidades, definido sus alcances y definido sus llaves primarias se debe de documentar e incluirlas en el diagrama E-R.
6. A continuación, las relaciones se obtienen a partir de dos entidades donde se debe descubrir la relación concentrándose en los verbos, definir el alcance de la relación así como su tipo y documentarlo en el diagrama y en tablas.
7. Para los atributos es necesario primero descubrirlos concentrándose en adjetivos o adverbios, se define el alcance del atributo y se documenta el atributo en la entidad. Para colocar los atributos de manera correcta en una entidad se utiliza el método de normalización que permite verificar si el modelo planteado es correcto y si funcionará al ser implementado.

IV.3 Diseño detallado

El diseño detallado se ocupa del refinamiento de la representación que lleva a una estructura de datos detallada y a las representaciones algorítmicas del Software. Este diseño se relaciona con la adaptación de código existente, modificación de algoritmos estándar, creación de nuevos algoritmos, diseño de representaciones de datos e integración del producto final.

IV.3.1 Diccionario de datos

En el diccionario de datos se encuentra una lista de todos los elementos que forman parte del flujo de datos de todo el Sistema. El diccionario de datos es un medio para que los usuarios compartan datos comunes y proporcione datos correctos, podemos decir que es un catálogo, un depósito de los elementos de un Sistema.

Una ventaja adicional del diccionario de datos es que los sistemas establecidos que usan un diccionario de datos, tienden a ser más efectivos y menos costosos de desarrollar.

El diccionario de datos es una herramienta importante ya que nos da la posibilidad de controlar y manejar la información sobre los datos en las fases de diseño, implantación, operación y expansión de una base de datos.

Los diccionarios de datos son importantes para:

1. Manejar los detalles es sistemas grandes.
2. Comunicar un significado común para todos los elementos del Sistema.
3. Documentar las características del Sistema.
4. Facilitar el análisis de los detalles con la finalidad de evaluar las características y determinar dónde efectuar cambios en el Sistema.
5. Localizar errores y omisiones en el Sistema.

A continuación las paginas siguientes muestran el diccionario de datos del SICLO-EA en el cual se encuentra información acerca de los datos como por ejemplo; cuantos caracteres abarca un determinado dato o qué otros nombres recibe en distintas partes del Sistema. Se encuentran también las definiciones de tablas del SICLO-EA así como sus relaciones:

TÁBLA: ACTIVIDADES

Propiedades

Activar Orden:	Falso
Definición. actualizable:	Verdadero
Fecha de creación:	10/21/97 12:58:42 PM
Última actualización:	5/22/98 4:48:29 PM

Columnas

Nombre	Tipo	Tamaño
Cve_Actividad	Texto	2
Descripcion	Texto	70
Capacidad	Texto	3
Día	Fecha/Hora	8
Hora	Fecha/Hora	8
NoRegistrados	Texto	3

Relaciones

ActividadesAsistencia



Atributos: Uno a varios

Índices de tabla

Nombre	Número de
Llave Primaria	1
Campos:	Cve_Actividad, Ascendente

TABLA: ADEUDO

Propiedades

Activar Orden:	Falso
Definición. actualizable:	Verdadero
Fecha de creación:	10/21/97 1:04:41 PM
Última actualización:	5/22/98 4:48:27 PM

Columnas

Nombre	Tipo	Tamaño
Cve_Adeudo	Texto	1
Total	Número (largo)	4
Pagado	Número (largo)	4
Adeudo	Número (largo)	4

Relaciones

AdeudoExpediente



Atributos: Uno a varios

Índices de tabla

Nombre	Número de
Llave Primaria	1
Campos:	Cve_Adeudo, Ascendente

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

TABLA: ASISTENCIA

Propiedades

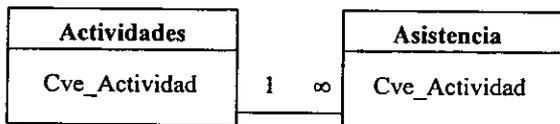
Activar orden: Falso
 Definición. actualizable: Verdadero
 Fecha de creación: 10/21/97 2:48:29 PM
 Última actualización: 5/22/98 4:48:29 PM

Columnas

Nombre	Tipo	Tamaño
Expediente	Texto	5
Cve_Actividad	Texto	2

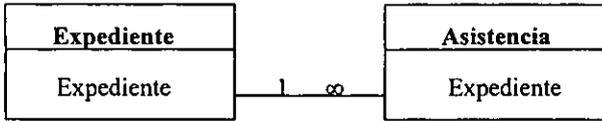
Relaciones

ActividadesAsistencia



Atributos: Uno a varios

ExpedienteAsistencia



Atributos: Uno a varios

Índices de tabla

Nombre	Número de
ActividadesAsistencia	1
Campos:	Cve_Actividad, Ascendente
AsistenciaCve_Actividad	1
Campos:	Cve_Actividad, Ascendente
ExpedienteAsistencia	1
Campos:	Expediente, Ascendente
Llave Primaria	2
Campos:	Expediente, Ascendente Cve_Actividad, Ascendente

TABLA: AVANCE_TESIS

Propiedades

Activar Orden:	Falso
Definición. actualizable:	Verdadero
Fecha de creación:	5/28/98 3:56:08 PM
Última actualización:	5/28/98 3:56:55 PM

Columnas

Nombre	Tipo	Tamaño
Cve_avance_tesis	Texto	2
Descripción	Texto	10

Índices de tabla

Nombre	Número de
Llave Primaria	1
Campos:	Cve_avance_tesis, Ascendente

TABLA: BANCOS**Propiedades**

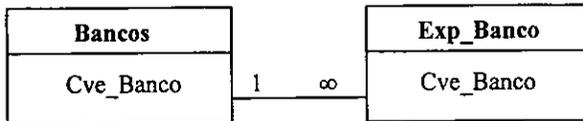
Activar Orden:	Falso
Definición. actualizable:	Verdadero
Fecha de creación:	11/16/97 2:16:14 PM
Última actualización:	5/22/98 4:48:27 PM

Columnas

Nombre	Tipo	Tamaño
Cve_Banco	Texto	3
Descripción	Texto	40

Relaciones

BancosExp_Banco



Atributos: Uno a varios

Índices de tabla

Nombre	Número de
Llave Primaria	1
Campos:	Cve_Banco, Ascendente

TABLA: COSTO_INSCRIPCION

Propiedades

Activar Orden:	Falso
Definición. actualizable:	Verdadero
Fecha de creación:	5/28/98 3:24:25 PM
Última actualización:	5/28/98 3:39:10 PM

Columnas

Nombre	Tipo	Tamaño
Cve_costo	Texto	2
Descripcion	Moneda	8

Índices de tabla

Nombre	Número de
Llave Primaria	1
Campos:	Cve_costo, Ascendente

TABLA: EDO_CIVIL

Propiedades

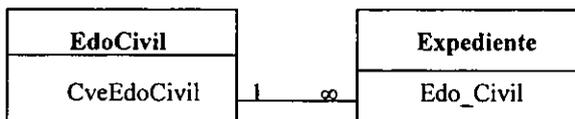
Activar Orden:	Verdadero
Definición actualizable:	Verdadero
Fecha de creación:	11/13/97 4:04:21 PM
Última actualización:	5/22/98 4:48:27 PM
OrdenarPor:	EdoCivil.CveEdoCivil

Columnas

Nombre	Tipo	Tamaño
CveEdoCivil	Texto	1
Descripción	Texto	15

Relaciones

EdoCivilExpediente



Atributos: Uno a varios

Índices de tabla

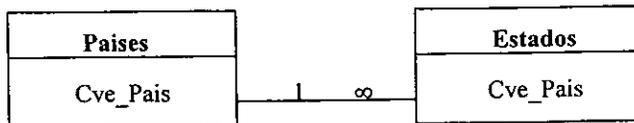
Nombre	Número de
Llave Primaria	1
Campos:	CveEdoCivil, Ascendente

TABLA: ESTADOS**Propiedades**

Activar Orden:	Falso
Definición actualizable:	Verdadero
Fecha de creación:	11/13/97 4:04:24 PM
Última actualización:	5/22/98 4:48:27 PM

Columnas

Nombre	Tipo	Tamaño
Cve_Pais	Texto	3
Cve_Estado	Texto	2
Descripcion	Texto	25

Relaciones**PaísesEstados**

Atributos: Uno a varios

Índices de tabla

Nombre	Número de
EstadosCve_Estado	1
Campos:	Cve_Estado, Ascendente
EstadosCve_Pais	1
Campos:	Cve_Pais, Ascendente
EstadosEstado	1
Campos:	Descripcion, Ascendente
PaísesEstados	1
Campos:	Cve_Pais, Ascendente
Llave Primaria	2
Campos:	Cve_Estado, Ascendente Cve_Pais, Ascendente

TABLA: EXP_BANCO

Propiedades

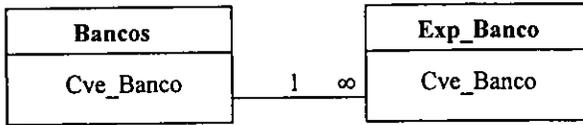
Activar Orden:	Falso
Definición. actualizable:	Verdadero
Fecha de creación:	10/21/97 3:10:39 PM
Última actualización:	5/22/98 4:48:29 PM

Columnas

Nombre	Tipo	Tamaño
Expediente	Texto	5
Cve_Banco	Texto	2

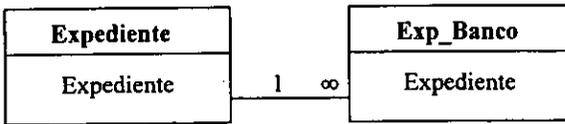
Relaciones

BancosExp_Banco



Atributos: Uno a varios

ExpedienteExp_Banco



Atributos: Uno a varios

Índices de tabla

Nombre	Número de
BancosExp_Banco	1
Campos:	Cve_Banco, Ascendente
ExpedienteExp_Banco	1
Campos:	Expediente, Ascendente
Llave Primaria	2
Campos:	Expediente, Ascendente Cve_Banco, Ascendente

TABLA: EXPEDIENTE**Propiedades**

Activar Orden:	Falso
Definición actualizable:	Verdadero
Fecha de creación:	10/21/97 12:43:44 PM
Última actualización:	9/26/98 12:02:37 PM

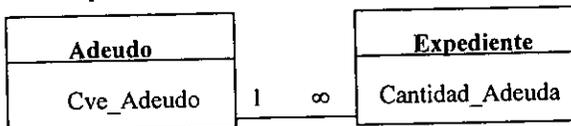
Columnas

Nombre	Tipo	Tamaño
Expediente	Texto	5
Tipo_Persona	Texto	2
Nombre	Texto	30
Ape_paterno	Texto	30
Ape_Adicional	Texto	30
Ape_Materno	Texto	40
Calle_No	Texto	150
Colonia	Texto	60
Del/Mun	Texto	40
ZP	Texto	15
Cve_Pais	Texto	3
Cve_Estado	Texto	2
Telefono1	Texto	20
Telefono2	Texto	20
Celular	Texto	10
Radiolocalizador	Texto	10
Fax	Texto	20

E_mail	Texto	60
Cve_Profesion	Texto	2
Empresa	Texto	80
Puesto	Texto	90
Edo_Civil	Texto	50
Edad	Texto	2
Sexo	Texto	1
Cve_Hotel	Texto	3
Cve_transporte	Texto	1
Cve_Actividad	Texto	2
Cantidad_Adeuda	Texto	1
Cve_adeuda	Texto	2
Cve_preponencia	Texto	2
Cve_costo	Texto	2
Cve_avance_tesis	Texto	2
Nombre_Ponencia	Texto	100
Comentarios	Texto	255
Asistencia	Texto	50
Memorias	Texto	2
foto	Objeto OLE	-

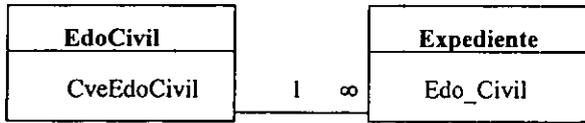
Relaciones

AdeudoExpediente



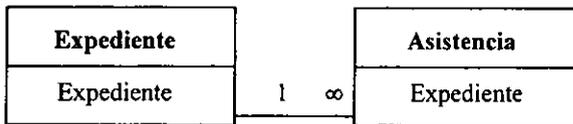
Atributos: Uno a varios

EdoCivilExpediente



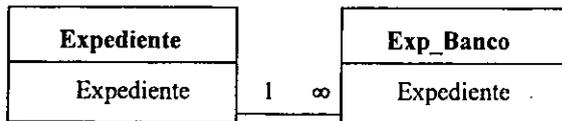
Atributos: Uno a varios

ExpedienteAsistencia



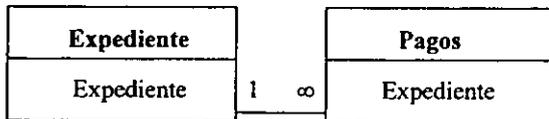
Atributos: Uno a varios

ExpedienteExp_Banco



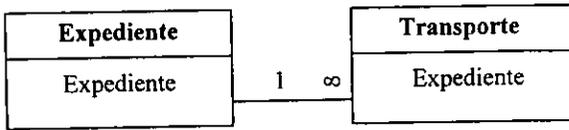
Atributos: Uno a varios

ExpedientePagos



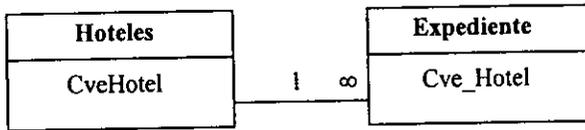
Atributos: Uno a varios

ExpedienteTransporte



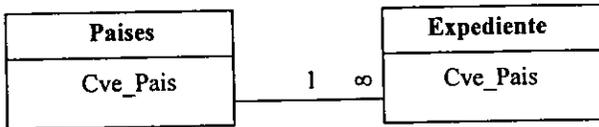
Atributos: Uno a varios

HotelesExpediente



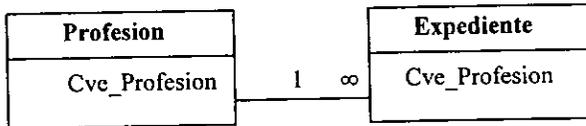
Atributos: Uno a varios

PaisesExpediente



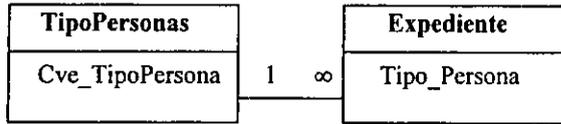
Atributos: Uno a varios

ProfesionExpediente



Atributos: Uno a varios

TipoPersonasExpediente



Atributos: Uno a varios

Índices de tabla

Nombre	Número de
AdeudoExpediente	1
Campos:	Cantidad_Adeuda, Ascendente
EdoCivilExpediente	1
Campos:	Edo_Civil, Ascendente
ExpedienteCve_Pais	1
Campos:	Cve_Pais, Ascendente
HotelesExpediente	1
Campos:	Cve_Hotel, Ascendente
PaisesExpediente	1
Campos:	Cve_Pais, Ascendente
Llave Primaria	1
Campos:	Expediente, Ascendente
ProfesionExpediente	1
Campos:	Cve_Profesion, Ascendente
TipoPersonasExpediente	1
Campos:	Tipo_Persona, Ascendente

TABLA: HOTELES

Propiedades

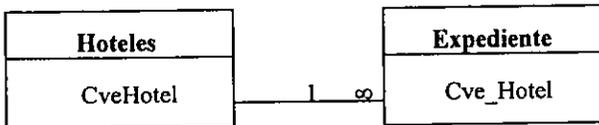
Activar Orden:	Falso
Definición. actualizable:	Verdadero
Fecha de creación:	11/16/97 2:28:38 PM
Última actualización:	9/26/98 11:24:18 AM

Columnas

Nombre	Tipo	Tamaño
CveHotel	Texto	3
Descripcion	Texto	50

Relaciones

HotelesExpediente



Atributos: Uno a varios

Índices de tabla

Nombre	Número de
Llave Primaria	1
Campos:	CveHotel, Ascendente

TABLA: PAGOS

Propiedades

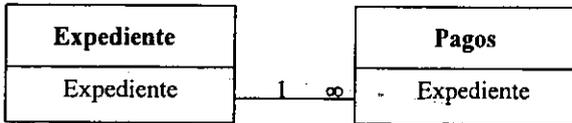
Activar Orden: Falso
 Definición actualizable: Verdadero
 Fecha de creación: 10/21/97 3:01:45 PM
 Última actualización: 5/22/98 4:48:29 PM

Columnas

Nombre	Tipo	Tamaño
Expediente	Texto	5
Cve_TpoPago	Texto	2

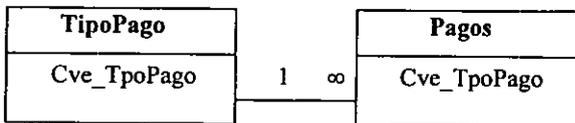
Relaciones

ExpedientePagos



Atributos: Uno a varios

TipoPagoPagos



Atributos: Uno a varios

Índices de tabla

Nombre	Número de
ExpedientePagos	1
Campos:	Expediente, Ascendente
Llave Primaria	2
Campos:	Expediente, Ascendente Cve_TpoPago, Ascendente
TipoPagoPagos	1
Campos:	Cve_TpoPago, Ascendente

TABLA: PAISES**Propiedades**

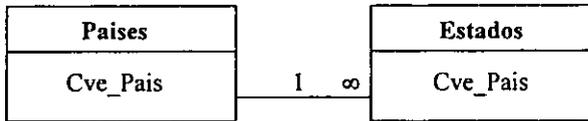
Activar Orden:	Verdadero
Definición actualizable:	Verdadero
Fecha de creación:	11/13/97 4:04:26 PM
Última actualización:	5/22/98 4:48:28 PM

Columnas

Nombre	Tipo	Tamaño
Cve_Pais	Texto	3
Descripción	Texto	30

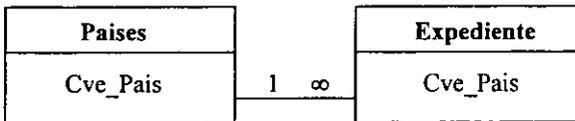
Relaciones

PaisesEstados



Atributos: Uno a varios

PaisesExpediente



Atributos: Uno a varios

Índices de tabla

Nombre	Número de
Llave Primaria	1
Campos:	Cve_Pais, Ascendente

TABLA: PARAMETROS

Propiedades

Activar Orden:	Falso
Definición actualizable:	Verdadero
Fecha de creación:	8/22/98 1:21:17 PM
Última actualización:	8/22/98 1:42:56 PM

Columnas

Nombre	Tipo	Tamaño
id_evento	Número (largo)	4
Activo	Sí/No	1
Nombre_evento	Texto	100
Lugar	Texto	100
Fecha	Texto	50
E-Mail	Texto	100
Organizadores	Texto	200
Telefonos	Texto	100
Patrocinadores	Texto	200
Tipo_de_letra	Texto	50
Color_de_letra	Texto	20
Color_de_fondo_formas	Texto	20
Imagen_tapiz	Texto	100

Índices de tabla

Nombre	Número de
Activo	1
Campos:	Activo, Ascendente
id_evento	1
Campos:	id_evento, Ascendente
Llave Primaria	1
Campos:	id_evento, Ascendente

TABLA: PRESENTA_PONENCIA**Propiedades**

Activar Orden: Falso
Definición actualizable: Verdadero
Fecha de creación: 5/22/98 3:45:05 PM
Última actualización: 5/22/98 4:48:29 PM

Columnas

Nombre	Tipo	Tamaño
Cve_presponencia	Texto	2
Descripcion	Texto	2

Índices de tabla

Nombre: Número de
Llave Primaria: 1
Campos: Cve_presponencia, Ascendente

TABLA: PROFESION**Propiedades**

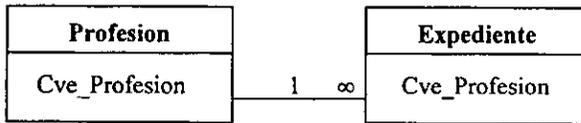
Activar Orden: Falso
Definición actualizable: Verdadero
Fecha de creación: 10/21/97 12:55:25 PM
Última actualización: 07/01/98 6:47:32 PM

Columnas

Nombre	Tipo	Tamaño
Cve_Profesion	Texto	2
Descripcion	Texto	50

Relaciones

ProfesionExpediente



Atributos: Uno a varios

Índices de tabla

Nombre	Número de
Llave Primaria	1
Campos:	Cve_Profesion, Ascendente

TABLA: SEGURIDAD

Propiedades

Activar Orden:	Falso
Definición. actualizable:	Verdadero
Fecha de creación:	11/16/97 2:27:54 PM
Última actualización:	5/22/98 4:48:28 PM

Columnas

Nombre	Tipo	Tamaño
Login	Texto	11
Password	Texto	11
Permisos	Texto	50
Observaciones	Texto	60

Índices de tabla

Nombre	Número de
Llave Primaria	1
Campos:	Login, Ascendente

TABLA: SEXO**Propiedades**

Activar Orden:	Falso
Definición actualizable:	Verdadero
Fecha de creación:	4/20/98 5:01:26 PM
Última actualización:	9/26/98 11:33:56 AM

Columnas

Nombre	Tipo	Tamaño
Cve_sex	Número (largo)	4
Descripcion	Texto	10

Índices de tabla

Nombre	Número de
Llave Primaria	1
Campos:	Cvesexo, Ascendente

TABLA: TABLA_ADEUDO**Propiedades**

Activar Orden:	Falso
Definición actualizable:	Verdadero
Fecha de creación:	5/18/98 1:47:17 PM
Última actualización:	5/22/98 4:48:34 PM

Columnas

Nombre	Tipo	Tamaño
cve_adeudo	Texto	2
Adeuda	Texto	2

Índices de tabla

Nombre	Número de
Llave Primaria	1
Campos:	cve_adeudo, Ascendente

TABLA: TIPO_PAGO

Propiedades

Activar Orden: Falso
 Definición. actualizable: Verdadero
 Fecha de creación: 11/16/97 2:19:13 PM
 Última actualización: 5/22/98 4:48:28 PM

Columnas

Nombre	Tipo	Tamaño
Cve_TpoPago	Texto	3
Descripción	Texto	50

Relaciones

TipoPagoPagos



Atributos: Uno a varios

Índices de tabla

Nombre Número de
 Llave Primaria 1
 Campos: Cve_TpoPago, Ascendente

TABLA: TIPO_PERSONA_INCLUYE

Propiedades

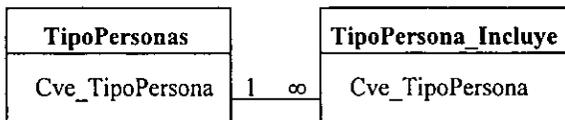
Activar Orden: Falso
 Definición actualizable: Verdadero
 Fecha de creación: 1/21/98 12:29:32 PM
 Última actualización: 5/22/98 4:48:29 PM

Columnas

Nombre	Tipo	Tamaño
Cve_TipoPersona	Texto	5
Cve_Incluye	Texto	2
Descripcion	Texto	70

Relaciones

TipoPersonasTipoPersona_Incluye



Atributos: Uno a varios

Índices de tabla

Nombre Número de

Llave Primaria 2

Campos: Cve_TipoPersona, Ascendente
Cve_Incluye, Ascendente

TipoPersonasTipoPersona_Incluye 1

Campos: Cve_TipoPersona, Ascendente

TABLA: TIPO_PERSONAS

Propiedades

Activar Orden: Falso

Definición actualizable: Verdadero

Fecha de creación: 11/16/97 2:22:02 PM

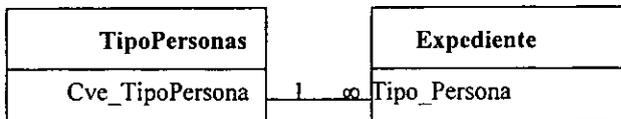
Última actualización: 5/22/98 4:48:28 PM

Columnas

Nombre	Tipo	Tamaño
Cve_TipoPersona	Texto	2
Descripcion	Texto	60

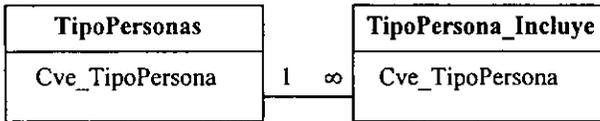
Relaciones

TipoPersonasExpediente



Atributos: Uno a varios

TipoPersonasTipoPersona_Incluye



Atributos: Uno a varios

Índices de tabla

Nombre: Número de
 Llave Primaria: 1
 Campos: Cve_TipoPersona, Ascendente

TABLA: TIPO_TRANSPORTE

Propiedades

Activar Orden: Falso
 Definición actualizable: Verdadero
 Fecha de creación: 11/16/97 2:20:21 PM
 Última actualización: 9/26/98 12:02:44 PM

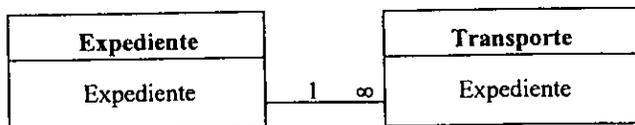
Columnas

Nombre	Tipo	Tamaño
CveTipoTransporte	Texto	2
Descripción	Texto	50

FechaLlegada	Fecha/Hora	8
HoraLlegada	Fecha/Hora	8
FechaSalida	Fecha/Hora	8
HoraSalida	Fecha/Hora	8

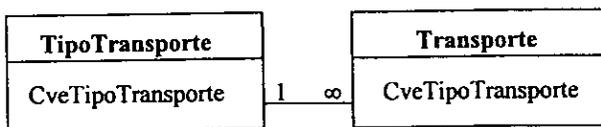
Relaciones

ExpedienteTransporte



Atributos: Uno a varios

TipoTransporteTransporte



Atributos: Uno a varios

Índices de tabla

Nombre	Número de
ExpedienteTransporte	1
Campos:	Expediente, Ascendente
Llave Primaria	2
Campos:	Expediente, Ascendente CveTipoTransporte, Ascendente
TipoTransporteTransporte	1
Campos:	CveTipoTransporte, Ascendente

IV.4 Especificaciones de diseño

Al completar el estudio de viabilidad se escoge una alternativa de Sistema para que entre en servicio. Los resultados de este estudio deben ser unas especificaciones completas para el nuevo Sistema. Y comenzamos por especificar el tipo de salida de información.

IV.4.1 Diseño de la salida para que sirva al propósito deseado

Toda la salida debe tener un propósito. No es suficiente que se ponga a disposición de los usuarios un reporte o pantalla debido a que sea tecnológicamente posible hacerlo. Durante la fase de análisis de determinación de los requerimientos de información, el analista de sistemas encuentra cuales propósitos deben ser atendidos. La salida es diseñada luego con base en esos propósitos.

Usted vera que se tiene diversas oportunidades para proporcionar salida, debido simplemente a que la aplicación permite que se haga. Sin embargo, recuerde la regla de que hay que tener un propósito. Si la salida no es funcional, no debe ser creada, debido a que hay costos de tiempo y materiales asociados con toda la salida del sistema.

IV.4.2 Diseño de la salida para que se ajuste al usuario

Con un gran sistema de información sirviendo a muchos usuarios para muchos propósitos diferentes, es difícil personalizar la salida. Con base a entrevistas, observaciones, consideraciones de costos y tal ves prototipos, será posible diseñar la salida que atienda lo que muchos usuarios, no todos, necesitan y prefieran.

Hablando de términos generales es más practico crear una salida específica cuando se le diseña para un sistema de soporte de decisiones u otras aplicaciones altamente interactivas. Sin embargo, todavía es posible diseñar salidas que se ajuste alas funciones de un usuario en la organización y esto nos lleva al siguiente objetivo:

- Asegurándose que la salida se encuentre donde se necesite

La salida es impresa en papel, desplegada en pantallas difundida en bocinas y guardada en microformas, La salida a veces se produce en un lugar (*por ejemplo el departamento de procesamiento de datos*) y luego se distribuye a los usuarios.

El incremento de salida desplegada en pantallas en línea que es accesible personalmente ha reducido en cierta forma el problema de la distribución pero la distribución adecuada todavía es el objetivo importante para el analista de sistemas. Para ser usada y útil la salida, debe ser presentada al usuario adecuado. Sin importar que tan bien estén diseñados los reportes, si no son vistos por los tomadores de decisiones pertinentes no tienen valor.

- Entrega de la salida a tiempo

Una de las quejas mas comunes de los usuarios es que no reciben la información a tiempo para tomar decisiones necesarias. Por lo tanto los objetivos del analista de sistemas con respecto a la salida son compuestos. No solo se tiene que ser consciente acerca de quien está recibiendo cual salida, sino también preocuparse de la distribución en el tiempo de la salida.

Aunque la temporización no lo es todo, juega un gran papel sobre qué tan útil será para los tomadores de decisiones. Mediante esta fase del ciclo de vida del desarrollo de sistemas usted aprende que salida es necesaria y en que momento para dirigir cada etapa de los procesos de organización. Muchos reportes se requieren diariamente otros solo mensualmente y otros anualmente y otros solamente por excepción. La temporización adecuada de la salida puede ser crítica para las operaciones del negocio.

- Selección del método de salida adecuado

Tal como se dijo anteriormente, la salida puede tomar muchas formas, incluyendo reportes impresos en papel, información en pantallas VDT, audio con sonidos digitalizados

que simulan voz humana y microformas. La selección del método adecuado de salida para cada usuario es otro objetivo para el diseño de la salida.

Para muchos el término de salida todavía los hace pensar en montones de impresiones en papel de computadora pero esto está cambiando rápidamente. Con motivo en sistemas en líneas, mucha salida aparece hoy en pantallas de desplegado. El analista necesita reconocer los compromisos involucrados en la selección de un método de salida. Los costos difieren, así como la flexibilidad, tiempo de vida, distribución, almacenamiento y posibilidades de recuperación, transportabilidad e impacto general en los datos para el usuario. La selección de los métodos de salida no es trivial ni es generalmente una conclusión predecible con certeza.

- Cómo afecta a los usuarios el ascenso de la salida

Cualquiera que sea la forma que tome, la salida no es simplemente un producto neutral que es utilizado subsecuentemente y sobre el que actúan los tomadores de decisiones. La salida afecta a los usuarios en muchas formas diferentes. El significado de este hecho para el analista de sistemas, es que se debe poner mucha imaginación y cuidado al diseñar la salida para evitar que se tome tendencia.

IV.4.3 Convenciones para el diseño del reporte.

Las convenciones a seguir cuando se llena la hoja para el diseño del reporte incluye como indicar el tipo del dato (alfabético, especial o numérico) que aparecerá en cada posición, como mostrar el tamaño exacto de la forma que está siendo preparada las formas de diseño consecutivas.

La *información* constante es información que permanece igual cada vez que es impreso el reporte. Para indicar la información constante, el analista la escribe en la forma del diseño un carácter por espacio. El título del reporte y todos los encabezados de columnas son escritos como información constante.

La *información variable* es información que puede variar cada vez que el reporte es impreso. En nuestro ejemplo, las cifras de ventas en miles de dólares cambiarán y, por lo tanto, están indicadas como información variable.

Para indicar un espacio que llenará la computadora con un carácter alfabético (A-Z y a-z) o un carácter especial, tal como un asterisco(*), signo de pesos (\$), o ampersand (&), se escribe X en el espacio (*tal como se hizo en el diccionario de datos*) Por ejemplo para mostrar la longitud de campo para el nombre de la tienda, que puede ser de hasta 21 caracteres alfabéticos de largo, se podría poner X en cada de los 21 espacios de la cuadrícula. Una convección alterna (*e igualmente aceptada*) podría ser poner una X en el espacio inicial con una línea en los siguientes espacios y una X para anclar el último espacio ocupado por el concepto y, en esta forma , el nombre de la tienda sería indicado por X_____X.

Para mostrar que la computadora llenará el espacio con un carácter numérico (0-9) se escribe un 9 en el espacio correcto de la hoja de diseño. Por ejemplo para indicar la cantidad de espacios necesarios para el número de tienda se podría escribir en la cuadrícula 9999. Esto simboliza que será impreso un número de cuatro dígitos en esa posición del reporte. Los espacios que se dejan en blanco para separar los datos deben ser dejados en blanco en la hoja de diseño de reporte de impresora. E n nuestro ejemplo se dejás tres espacios en blanco entre el nombre de está y el número de la tienda para hacer legible el reporte

Otra convención indica que cuando los datos van a ser repetidos en una misma posición de una columna, no es necesario continuar poniendo detalles en cada espacio sucesivo. Simplemente se traza una línea ondulada (llamada la *línea de detalle*) continuando hacia abajo desde el principio de la columna tal como lo muestra la línea de detalle bajo el encabezado Store No., Store Name y todas las otras columnas que contienen información variable.

Para estimar la anchura del reporte, determine para cada campo el máximo de (1) los requerimientos de longitud de campo de los elementos de datos que se van a usar, tal como

aparece en diccionario de datos, y (2) la línea más larga de los encabezados de columna propuestos. Luego añada dos espacios para espacio para ambos lados que permita una mejor legibilidad. Por último, sume estas estimaciones de campos para obtener las estimaciones de anchura.

1V.4.4 Consideraciones del diseño de reportes

Al diseñar el reporte impreso el analista de sistema incorpora consideraciones, tanto funcionales como estilísticas o estéticas, para que el reporte proporcione al usuario la información necesaria en un formato legible. Dado que la función y la forma se refuerzan entre ellas, una no debe ser enfatizada a expensas de la otra.

- **Atributos funcionales**

Los atributos funcionales de un reporte impreso incluyen el encabezado o título del reporte, el número de página, la fecha de preparación, los encabezados de columna, el agrupamiento de conceptos de datos relacionados y el uso de cortes de control. Cada uno de ellos sirve a un propósito distinto para el usuario.

El encabezado o el título del reporte orienta inmediatamente al lector conciso. A veces es redundante incluir la palabra "reporte" en el título.

Cada página debe estar numerada para que el usuario tenga un punto de referencia fácil cuando trata la salida con otros o para localizar cifras importantes. También; si las páginas de la salida llegan a separarse, los números de páginas son invaluable para la reconstrucción del documento. Incluya la fecha de preparación del reporte en cada impresión. Los encabezados de columnas sirven para orientar todavía más al usuario sobre el contenido del reporte. Cada concepto de datos debe tener un encabezado. Los encabezados deben ser cortos y descriptivos se permite usar abreviaturas si son comunes y significativas para los usuarios. Revise que las abreviaturas no sean usadas en otros lugares de la organización con un significado diferente con el que se pretende para el reporte Use cortes de control (*que son cortes de los datos donde suceden resúmenes*) para ayudar a la legibilidad.

- **Atributos estilísticos / estéticos**

Hay varias consideraciones estilísticas o estéticas que debe observar el analista de sistemas cuando diseñe un reporte impreso. Si la salida impresa es muy difícil de leer y no es atractiva, no será usada efectivamente y puede ser que hasta no se use. El resultado son tomadores de decisiones desinformados y desperdicio de tiempo de computadora así como de otros recursos organizacionales.

El uso de cortes de control fue presentado anteriormente como una consideración funcional, pero la manera en que están puestos en una página de reporte también tienen también que ver con la estética. Hay que llamar la atención hacia los cortes de control, totales es otra información importante encuadrándolos dentro de caracteres especiales tales como asteriscos u espacios adicionales. Esto facilita encontrar información crítica evite la impresión de columnas en formaciones largas y sin cortes. Los espacios en blanco adicionales entre columnas también contribuyen a la legibilidad de un reporte

IV.4.5 Lineamientos para el diseño de pantallas

Hay cuatro lineamientos para facilitar el diseño de pantallas. Resumiéndolos son:

- 1.- Mantenga la pantalla simple
- 2.- Mantenga consistente la presentación de la pantalla
- 3.- Facilite al usuario el movimiento entre pantallas
- 4.- Cree una pantalla atractiva

Así como se usa la hoja de diseño de reporte impreso para planear el espaciado de la salida impresa se usa una forma similar para planear las pantallas y comunicar esos detalles a los programadores. Básicamente se siguen las mismas convenciones de notación para la disposición de la pantalla que para la salida impresa. Las diferencias en el diseño de pantalla incluyen la necesidad de poner instrucción en la pantalla para el cambio de pantallas, el movimiento de pantallas y la terminación del despliegado de la salida.

1V.5 Problemas de los sistemas

Hay problemas clásicos cuando se desea desarrollar un sistema, problemas que en mayor o menor medida el analista, diseñador o programador se encontrarán, en cualquiera de las etapas para la obtención del sistema:

- Cuestan mucho. Toman mucho tiempo para completarse, claro, esto depende de la magnitud del sistema a desarrollar.
- No son entendibles, este punto básicamente es causa de la poca comunicación del usuario y los responsables de desarrollo.
- No son modificables
- No son confiables (*Fallan*), es realmente difícil encontrar un sistema que tenga en control de todos los errores que pudieran presentarse.
- No cumplen con los requisitos de los usuarios, en este punto es clásico que por estética o funcionalidad el usuario este en desacuerdo en lo que se tiene desarrollado.
- Su complejidad excede la capacidad humana. Existen procesos, cálculos, etc., que por el poco potencial de la herramienta de desarrollo seleccionada no se puedan crear, también se da el caso en que por cuestiones de Hardware no es posible tener la fluidez que se desea en la funcionalidad del sistema.
- Depuración no es suficiente, aunque se haya llegado al término del sistema, pero si se tuvo mucha falta de cuidado esto no es suficiente.
- Son costosos en su mantenimiento, cuando un sistema fue tardado, además de complejo en su desarrollo esta es la consecuencia por la poca calidad en todo lo que a este se refiera.
- Son costosos en su operación, esto ocurre cuando el sistema desarrollado necesita equipo de computo de muy alto costo, además de gente que sea capaz de poderle dar uso tanto al sistema como a la plataforma en que trabaja.

IV.5.1 Presupuesto limitado

Cuando se tiene interés de construir un sistema hay factores muy importantes que se deben considerar, el factor que a decir verdad es el que mayor peso tiene, es el de contar con un presupuesto limitado, esto muestra que el tiempo también sea limitado, y haya necesidad de planear todas las tareas a realizar, dándoles tiempo a cada una de ellas para que se puedan cumplir las tareas planeadas. Además del factor tiempo, se tienen recursos humanos limitados, los cuales son los que se encargaran del desarrollo del sistema.

Un factor más que influye de muchas maneras en el diseño, son las prioridades y urgencias, ya sea porque en un periodo ya próximo se necesita una versión o parte del sistema ya funcionando. Un cuarto factor que siempre en todo sistema, pese al excelente análisis que se haya realizado, aunque se interprete correctamente cada parte obtenida por la etapa anterior nunca faltan las situaciones cambiantes.

IV.6 Cohesión y Acoplamiento

Un diseño adecuado es aquel que nos permita una codificación eficiente o puede ser un diseño mínimo o el de más fácil mantenimiento y para tal efecto el diseño del Software debe tener *mucha cohesión y poco acoplamiento*.

IV.6.1 Cohesión

La cohesión es una extensión del concepto de ocultamiento de información, un módulo cohesivo ejecuta una tarea sencilla de un procedimiento de Software y requiere poca interacción con procedimientos que ejecutan otras partes de un programa. Dicho de otra forma, un módulo cohesivo sólo hace (*idealmente*) una cosa.

No es necesario determinar el nivel preciso de cohesión, en su lugar, lo importante es intentar conseguir una cohesión alta y saber reconocer la cohesión baja, de forma que se

pueda modificar el diseño del Software para que disponga de una mayor independencia funcional.

Mucha cohesión quiere decir que todas las partes de un módulo o unidad estén altamente relacionadas funcionalmente de modo que todas sean parte esencial para el cumplimiento del objetivo del módulo o unidad.

IV.6.2 Acoplamiento

El acoplamiento es una medida de la interconexión entre los módulos de una estructura de programa.

En el diseño de Software buscamos el más bajo acoplamiento posible. La conectividad sencilla entre módulos da como resultado un Software que es más fácil de comprender y menos propenso al "efecto onda" producido cuando los errores aparecen en una posición y se propagan a lo largo del Sistema.

Poco acoplamiento significa que entre módulos o unidades no existan conexiones tan fuertes que provoquen dependencia, para que de tal modo las posibles modificaciones a un módulo o unidad afecten lo menos posible a otros.

Las ventajas de un diseño con mucha cohesión y poco acoplamiento, son que cualquier módulo o unidad de programa se puede reemplazar por un equivalente con poco o ningún cambio en las otras unidades.



Capítulo V

Desarrollo del

SICLO-~~EA~~

DESARROLLO DEL SICLO-EA

Introducción

En la etapa de desarrollo encontramos todo lo que se refiere a la programación, aquí es donde todas las especificaciones de diseño se tienen que transformar en código fuente.

El proceso de desarrollo de un programa o Software a partir de un diseño de Software puede enfocarse principalmente a dos formas: el desarrollo ascendente y el descendente, debido a las ventajas del desarrollo descendente se da una descripción detallada.

El desarrollo descendente se aplica cuando la estructura del programa es jerárquica. El programador aplica los niveles más altos del diseño y representa los niveles inferiores con cabos que simulan su función de manera simplificada. Al acabar la aplicación de un nivel, el programador pasa al siguiente nivel inferior y lo aplica a partir de sus subniveles. El desarrollo descendente es conocido también como técnica Top-Down (Arriba-Abajo) como vimos previamente en el capítulo II.

V.1 La fase de desarrollo

En la fase de desarrollo transformaremos todas las especificaciones del diseño a código fuente donde este debe de ser claro y sencillo en la mayor medida posible, ya que al final esto nos permitirá que el Sistema sea fácil de depurar, probar y modificar.

Para lograr lo anterior, existen técnicas de codificación adecuadas para el desarrollo del Sistema, contar con un buen estilo de codificación, documentar el código y haber elegido un lenguaje de programación adecuado.

V.2 Desarrollo del SICLO-EA

En esta sección definimos el diseño de sistemas de abajo hacia arriba y de arriba hacia abajo, así como el enfoque modular a la programación. Trataremos las ventajas de cada uno, así como las precauciones que deben ser observadas cuando se emplea un enfoque de arriba hacia abajo o modular. También trataremos la adecuabilidad de los enfoques de arriba hacia abajo o modular para que ayuden en el aseguramiento de la calidad de los proyectos de sistemas.

V.2.1 Desarrollo arriba hacia abajo.

Es fácil visualizar a lo que se refiere el enfoque de arriba hacia abajo, ya que se refiere haber una gran imagen del sistema y luego explorarla a partes o subsistemas más pequeños. El diseño de arriba hacia abajo permite que el analista de sistemas logre primero los objetivos organizacionales generales. Luego el analista se mueve para dividir el sistema de subsistemas y sus requerimientos.

Las ventajas de usar el enfoque de arriba hacia abajo para el diseño de sistemas incluye, el evitar el caos de diseñar un sistema "todo a la vez". Tal como hemos visto planeación e implementación de un sistema de administración de información es increíblemente compleja. El uso de equipos para el diseño de subsistemas en su lugar y funcionando a la vez es aceptar que se va a fallar. Una segunda ventaja de tomar un enfoque de arriba hacia abajo para el diseño de la habilidad de tener equipos de análisis de

sistemas separados trabajando en paralelo en diferentes, pero necesarios, en subsistemas. Esto puede ahorrar una gran cantidad de tiempo. El uso de los equipos para el diseño de subsistemas está particularmente bien adecuado para un enfoque de aseguramiento de calidad total. Una tercera ventaja se encuentra en evitar el problema principal asociado con el enfoque de abajo hacia arriba. Esto es usando un enfoque de arriba hacia abajo se previene que los analistas de sistemas se enfoquen demasiado a los detalles que pierdan de vista lo cual se supone que debe hacer el sistema.

Hay algunas fallas del sistema de arriba hacia abajo de las que necesita estar consiente el analista de sistemas. La primera es el peligro de que el sistema sea dividido en los subsistemas "equivocados". Se debe poner atención a las necesidades que se traspalan y la compartición de recursos para que el particionado de subsistemas tenga sentido dentro de la imagen del sistema total.

Un segundo peligro es una vez que se han hecho las divisiones de subsistemas sus interfaces pueden ser descuidada o ignoradas. Necesita ser detallada la responsabilidad de la interfaz. Una tercera precaución que acompaña el uso del diseño de arriba hacia abajo es que eventualmente los subsistemas deben ser reintegrados.

V.2.2 Desarrollo modular.

Una vez que ha sido tomado el enfoque de diseño de arriba hacia abajo, el enfoque modular es útil en la programación. Este enfoque involucra la división de la programación en partes o módulos lógicos y manejables. Este tipo de programación se ajusta bien con el diseño de arriba hacia abajo, debido a que enfatiza las interfaces entre módulos, en vez ignorarlas hasta después en el desarrollo de sistemas. Idealmente cada módulo debe ser funcionalmente coherente, por lo que está encargado del logro de una sola función.

El diseño de programación modular tiene tres ventajas principales. Primero, los módulos son más fáciles de escribir y de depurar, debido a que son virtualmente autocontenidos. El rastrear un error en un módulo es menos complicado, debido a que un problema en un módulo no debe causar problemas en otros.

Una segunda ventaja del diseño modular es que los módulos son más fáciles de mantener. Por lo general las modificaciones estarán limitadas a unos módulos y no estarán dispersas por todo el programa completo.

Una tercera ventaja del diseño modular es que los módulos son más fáciles de entender, debido a que son subsistemas autocontenidos. Lo que significa que un programador puede tomar un listado de código de un módulo y comprender su función.

Algo que es bueno comentar es que Access 97 herramienta donde se desarrollo SICLO-EA, hace uso de la filosofía de Visual Basic, donde se ejecutará una rutina (*módulo de código*) que tendrá la tarea de realizar una actividad, todo esto dependiendo del tipo de evento que ocurra en la ventana de donde se consulte, actualice, se borre, etcétera.

Algunos lineamientos para la programación modular incluyen:

- Mantenga cada módulo de un tamaño manejable (*idealmente incluyendo una sola función*).
- Ponga particular atención a las interfaces críticas (*los datos y las variables de control que son pasados a otros módulos*).
- Minimice la cantidad de módulos que necesita el usuario modificar cuando hace cambios.
- Mantenga las relaciones jerárquicas puestas en las fases arriba hacia abajo.

V.2.3 Modularidad en el ambiente Windows.

La modularidad está llegando a ser cada vez más importante. Microsoft desarrolla dos sistemas para enlazar programas en su ambiente Windows. El primero es llamado Intercambio Dinámico de Datos (*DDE*) el cual comparte código usando archivos de bibliotecas de enlace dinámico (*DLL*), mediante el uso del DLL un usuario puede guardar datos en un programa tal vez una hoja de cálculo, tal como Excel, y luego usar esos datos en otro programa, tal como un procesador de palabras como el Word para Windows. El programa que contiene los datos originales es llamado el servidor, y el programa que usa los datos es llamado el cliente (*otros términos para cliente son recipiente y anfitrión*). Uno de los archivos DLL más comúnmente usados es *COMM.DLG.DLL*, que contiene los diálogos del cuadro de apertura de archivo, guardado de archivo, búsqueda e impresión de Windows. Una ventaja del uso de este archivo es que los programas tendrán la misma apariencia que los demás programas de Windows. También agiliza el desarrollo, debido a que los programadores no tienen el código contenido en archivos DLL comunes.

Sin embargo, hay una gran desventaja en el uso de un archivo tal como *COMM.DLG.DLL*: sus características limitadas. Tal vez un diseñador de sistemas piensa que es importante que el usuario tenga la habilidad que crea un subdirectorio a mover o renombrar archivos cuando guarda un archivo.

V.3 Diagramas de Flujos de Datos Lógicos y Físicos.

Los diagramas de flujos de datos son categorizados como lógicos o físicos. Un diagrama de flujo de datos lógico se enfoca en el negocio y la manera en que se opera el negocio. No le importa la manera en que el sistema será construido. En vez de ello, describe los sucesos del negocio que suceden y los datos requeridos y producidos por cada evento. En forma inversa, un diagrama de flujo de datos físico muestra como será implementado el sistema, incluyendo el hardware, software, archivos y personas involucradas en el sistema. Observe que el modelo lógico refleja el negocio y, en cambio, el modelo físico representa el sistema.

Idealmente, los sistemas son desarrollados analizando el sistema actual (el DFD lógico actual) y añadiendo después características que deben incluir el nuevo sistema (*el DFD lógico propuesto*). Por último deben ser desarrollados los mejores métodos para implementar el nuevo sistema (*el DFD físico*).

El desarrollo de un diagrama de flujo de datos lógico para SICLO-EA nos dará los medios para que se tenga una comprensión clara sobre la manera en que opera el sistema, por lo tanto, un buen punto de inicio para el desarrollo del modelo lógico. Este paso consumidor e tiempo es a veces omitido para ir directamente al DFD lógico propuesto.

Un argumento a favor de tomarse el tiempo para construir el diagrama de flujo de datos lógico de SICLO-EA es que puede ser usado para crear el diagrama de flujo de datos para el nuevo sistema. Los procesos que serán innecesarios en el nuevo sistema pueden ser eliminados, y pueden ser añadidas nuevas características, actividades, salidas, entradas y datos guardados. Este enfoque proporciona un medio de asegurarse de que sean guardadas las características esenciales del sistema antiguo en el nuevo sistema.

V.3.1 Desarrollo de diagramas de Flujo de Datos Lógicos.

Primero construya un diagrama de flujo de datos lógico para el sistema actual. Hay varias ventajas de usar un modelo lógico, incluyendo:

- Mejor comunicación con los usuarios
- Sistemas más estables
- Mejor comprensión del negocio por analistas
- Flexibilidad del mantenimiento
- Eliminación de redundancias y creación más fácil del modelo físico

Es más fácil usar el modelo lógico cuando se comunica con los usuarios del sistema, debido a que está centrado en las actividades del negocio. Por lo tanto, los usuarios estarán familiarizados con las actividades esenciales y muchos de los requerimientos de información de cada actividad.

Los sistemas formados usando un diagrama de flujo de datos son frecuentemente más estables que aquellos que no lo son, debido a que están basados en eventos del negocio y no en una tecnología o método de implementación particular. Los diagramas de flujos de datos lógicos representan características de SICLO-EA que podrían existir sin importar cuáles fueran los medio físicos para realizar negocios. Un diagrama de flujos de datos lógico tiene un énfasis en el negocio, y ayuda a que el analista comprenda el negocio que está siendo estudiado, que entienda el porqué son ejecutados los procedimientos y que determine los resultados esperados de la ejecución de una tarea.

SICLO-EA será más flexible y fácil de manejar si su diseño esta basado en un modelo lógico. Las funciones del negocio no están sujetas a cambios frecuentes. Los aspectos físicos del sistema cambian más frecuentemente que las funciones del negocio.

El examen de un modelo lógico puede ayudar a que SICLO-EA sea un mejor sistema, eliminando redundancias y métodos ineficientes. Adicionalmente, el modelo lógico es fácil de crear y simple de usar, debido a que frecuentemente no contiene almacenes de datos aparte de los archivos o base de datos maestros.

V.3.2 Desarrollo de diagramas de Flujo de Datos Físicos.

Cuando el método lógico del nuevo sistema esta terminado puede ser usado para crear un diagrama de flujo de flujo de datos físicos. El diagrama de flujo de datos físicos muestra como será construido el sistema. Así como los diagramas de flujos de datos lógicos tienen ciertas ventajas, los diagramas de flujo de datos físicos tienen otras, incluyendo:

- Clarifican cuales procesos son manuales
- Describen procesos a mayor detalle que como lo hacen los DFD lógicos
- Secuencia los procesos que deben ser hechos en un orden particular
- Identifican almacenes de datos temporales
- Especifican los nombres actuales de archivos e impresiones

- Añaden controles para asegurar que los procesos son realizados adecuadamente

Los diagramas de flujos de datos físicos también tienen almacenes de datos intermedios, frecuentemente en un archivo de transacciones, se requieren archivos de transacciones o maestro o base de datos para enlazar dos procesos cualquiera que operen en momentos diferentes. Por ejemplo se puede usar un programa para procesar la búsqueda de una persona que asistió al último evento para mostrar un reporte de toda la información relacionada a este.

La información que genera ese programa puede tener que ser guardada, luego en un archivo, para el envío a impresión, para reproducir un reporte de en pantalla. También los almacenes de datos físicos son parte de los diagramas de flujo de datos físicos. Estos son indicados por los nombres actuales de los archivos o base de datos (*por ejemplo, EXPEDIENTE, CATALOGO DE ACTIVIDADES, en vez de la etiqueta de ASISTENTE usada en un diagrama de flujo de datos lógico*), y pueden ser descritos adicionalmente incluyendo el nombre de juego de datos, cantidad de registros y otros atributos principales.

V.4 Archivo de Base de Datos

V.4.1 Normalización.

La normalización es la transformación de vistas de uso complejas y almacenes de datos a un conjunto de estructuras de datos estables más pequeñas. Además de ser más simples y más estables, las estructuras de datos normalizadas son más fáciles de mantener.

- **Los tres pasos de la normalización.**

Comenzando ya sea con una vista de usuario o un almacén de datos desarrollado para un diccionario de datos, el analista normaliza una estructura de datos en tres pasos. Cada paso involucra un procedimiento importante para simplificar la estructura de datos.

Es muy probable que la relación derivada de la vista del usuario o del almacén de datos esté no normalizada. La **primera** etapa de proceso incluye la eliminación de todos los grupos repetidos y la identificación de la llave primaria. Para hacer esto, la relación necesita ser dividida en dos o más relaciones. En este momento las relaciones ya pueden evitar en la **tercera** forma normal, pero es muy probable que se necesite más pasos para transformar las relaciones a la tercera forma normal.

El **segundo** paso asegura que todos los atributos que no son llave sean completamente dependientes de la llave primaria. Todas las dependencias parciales son eliminadas y puestas en otra relación.

V.4.2 Uso del diagrama Entidad-Relación para determinar las llaves de registro.

Se puede usar el diagrama de E-R para determinar las llaves de un registro una relación de base de datos. El primer paso es construir el diagrama entidad relación e indicar una llave (*primaria*) única para cada entidad de dato. Hay tres entidades de datos: EXPEDIENTE, con una llave primaria de ID_EXPEDIENTE; ASISTENCIAS, con una llave de ID_ASISTENCIAS; y ACTIVIDADES, con un ID_ACTIVIDADES como llave primaria. Un EXPEDIENTE puede tener muchas asistencias, pero cada ASISTENCIA sólo puede ser colocado por un EXPEDIENTE, por lo que la relación es de uno a muchos. Cada ASISTENCIA puede contener muchas ACTIVIDADES y cada ACTIVIDAD puede estar contenido en muchas ASISTENCIAS, por lo que la relación ASISTENCIA-ACTIVIDAD es de muchos a muchos.

Sin embargo, una llave foránea es un campo de datos de un archivo dado a que es una llave primaria de un archivo maestro diferente. Por ejemplo, un ID_PAIS que indica el país de donde es originario una persona, estudiante, doctor, etc. puede existir en el archivo la tabla de EXPEDIENTE. El ID_ADEUDO también puede ser la llave única del archivo ADEUDOS

V.5 Tipos de interfaz de usuario.

En esta sección se describen varios tipos diferentes de interfaz de usuario, incluyendo interfaces de lenguaje natural, preguntas y respuestas, menús, interfaces de llenado de forma, lenguaje de comandos e interfaces gráficas del usuario (GUI). La interfaz del usuario tiene dos componentes principales: el lenguaje de presentación, que es la parte de la computadora al usuario de la transacción, y el lenguaje de acción, que caracteriza la parte usuario a computadora. Ambos conceptos juntos cubren la forma y el contenido del término interfaz de usuario.

V.5.1 Interfaces de llenado de forma.

Las interfaces de llenado de forma consisten de formas en pantallas las cuales despliegan campos que contienen conceptos de datos o parámetros que necesitan ser comunicados al usuario. La forma es frecuentemente un facsímil de la forma en papel que ya es familiar al usuario. Esta técnica de interfaz también es conocida como método basado en formas y formas de entrada/salida.

Las formas en pantalla son puestas para que muestren qué información debe ser dada y dónde. Los campos en blanco que requieren información pueden ser resaltados con video caracteres parpadeantes. El cursor es movido por el usuario de campo a campo mediante una sola opresión de la tecla de flecha, por ejemplo. Este método el movimiento de un campo hacia delante o hacia atrás mediante la opresión de la tecla de flecha.

La entrada de formas para pantallas puede ser simplificada proporcionando a los campos valores por omisión y permitiendo después que los modifiquen la información por omisión si es necesario. Por ejemplo si un sistema de administración de base de datos diseñado para mostrar una forma para captura de cheques puede proporcionar el siguiente número en cheque en secuencia por omisión cuando se presenta una nueva forma de cheque. Si faltan cheques el usuario cambia el número de cheque para que refleje el cheque actual que está siendo capturado.

La computadora tomará los datos proporcionados y busca en la base de datos adecuada para encontrar información que complete los registros ya comenzados. La computadora llena la forma con dos nombres. La entrada de campos en pantalla puede ser alfanuméricamente restringida, en forma tal que los usuarios puedan teclear solamente números en un campo que solicita un número de expediente, o teclear solamente teclas cuando se requiere el nombre de una sola persona. Si se teclean números cuando solo se permiten letras, la computadora puede indicar al usuario que el campo fue llenado en forma incorrecta. Sin importar cual tipo de coacción se use para asegurar que la forma le llena adecuadamente, se debe tomar en cuenta a los usuarios.

La ventaja principal de la interfaz de formas de entrada/salida es que la versión impresa de la forma impresa proporciona excelente documentación. Muestra etiquetas de campos así como el contexto para la captura.

Hay algunas cuantas desventajas de la forma de entrada/salida. La principal desventaja es similar al problema de "usuario experimentado", tratado antes para las interfaces de pregunta y respuesta de menús. Los usuarios experimentados pueden impacientarse con las formas de entrada/salida y pueden querer formas para capturar datos que sean más eficientes.

V.5.2 Interfaces Gráficas de Usuario (GUI)

Las interfaces gráficas de usuario (*GUI*) permiten el manejo directo de la representación gráfica en la pantalla, por lo que puede lograrse con entrada de teclado, palanqueta o ratón. El manejo directo requiere más sofisticación del sistema que las interfaces tratadas anteriormente, todo este tipo de puntos mencionados anteriormente se implementaran en SICLO-EA, así buscando sea un ambiente semejante al estándar de Windows.

Un ejemplo de la característica de mover y fijar una interfaz GUI (*llamada a veces simplemente interfaz de mover y fijar*). Para tratar con esta interfaz el usuario fija y mueve una forma maestra en su dibujo. Este tipo de interfaz hace mucho más fácil el trabajo del analista para la creación de diagramas de entrada relación u otros diagramas especializados. Visio viene con muchas más pantallas para trazar diagramas de flujo de datos, gráfica de flujo de programas, diagramas de red, diagramas orientados a objetos, y hasta tienen plantillas para el diseño de espacios de oficina.

La clave para el GUI es la retroalimentación constante para el logro de la tarea que proporciona. La retroalimentación continua del objeto manipulado significa que los cambios o reversiones en las operaciones pueden hacerse rápidamente sin caer en el mensaje de error. El concepto de retroalimentación para los usuarios se trata ampliamente en una sección posterior, "Retroalimentación para usuarios".

Los iconos se usan ampliamente cuando se ejecutan los sistemas operativos Windows de Microsoft, OS/2 de IBM o Macintosh de Apple. Presionando rápidamente con el botón del ratón, el usuario selecciona un icono para ejecutar el programa correspondiente. Los iconos están acomodados frecuentemente dentro de ventanas más grandes llamadas grupos de programas o carpetas. Cuando se selecciona un grupo de programas la ventana se abre desplegando los iconos del grupo. El analista debe diseñar pantallas completas o menús desplegables. Un ejemplo de conceptos de grupo y su ventana de grupo de programa correspondiente.

La creación de interfaces GUI plantea un reto, debido a que debe ser inventado un modelo adecuado de la realidad o un modelo conceptual aceptable de la representación. Esto requiere la combinación de varias habilidades en una forma que exige mucho de las capacidades de la mayoría de los analistas de sistemas y programadores.

V.6 La administración de la Calidad Total en el desarrollo del SICLO-EA

La administración de calidad total (TQM) es esencial a lo largo de todos los pasos de desarrollo de sistema. De acuerdo con Dean y Evans (1994) los elementos primarios de la TQM son significativos solamente cuando suceden en un contexto organizacional que de soporte a un esfuerzo de calidad completo. Es dentro de ese contexto en que se enfocan los elementos del cliente, la planeación estratégica y el liderazgo, la mejora continua, y el mando y el trabajo de equipo llegan a unirse para cambiar el comportamiento de los empleados y, a final de cuentas, el curso de la organización. Observe que el concepto de calidad se ha ampliado a lo largo de los años para reflejar un enfoque organizacional, en vez de ser exclusivo de producción. En vez de concebir la calidad como el control de la calidad de productos defectuosos producidos, ahora se piensa sobre la calidad como un proceso evolucionario hacia la perfección, a la que se le menciona como administración de calidad total.

El énfasis sobre la calidad ha pasado del piso de ventas, por los rangos medios y hasta lo más alto de la administración. Una combinación de factores ha contribuido a elevar la preocupación corporativa por la calidad en todos los niveles. Estos factores incluyen una disponibilidad aumentada de datos de desempeño, emulación de las técnicas de administración exitosas de los japoneses y una mayor atención a la productividad de los supervisores.

Los datos de desempeño dentro del negocio están llegando cada vez más fácilmente disponibles para los empleados de todos los niveles. La aceptación de la "administración por objetivos" y el maridaje de las técnicas de administración participativa han hecho que los empleados se preocupen acerca de cómo está siendo medido su desempeño. Esta conciencia del empleado es importante con relación a la administración de calidad total, debido a que el concepto solo puede funcionar si cada individuo está comprometido al trabajo de calidad.

La competencia comercial aguda con el Japón a hecho que los gerentes de Estados Unidos tengan curiosidad acerca de las técnicas de administración que los japoneses están usando para tener éxito, aunque se han discutido muchos factores, el compromiso internacional hacia la calidad parece tener una gran influencia.

Por último, consideramos el enfoque cada vez mayor sobre la productividad de los de los supervisores para que ayuden a alentar la producción de la TQM en muchos negocios. Cuando la productividad de las líneas de producción mejora significativamente, también se le eleva la demanda de una mayor productividad de los supervisores. La presión para incrementar la productividad de la administración fue sustentada por el desarrollo de nuevas técnicas para medir lo que realmente hacen los gerentes. Un movimiento para una mayor contabilización de los niveles superiores, cuyas campañas están siendo reducidas en tamaño también dio impulso al deseo de una productividad administrativa mejorada.

Los analistas de sistemas deben estar conscientes de los factores que están manejando el interés sobre la calidad. Es importante darse cuenta que el compromiso cada vez mayor de los negocios en la TQM cae extraordinariamente bien en los objetivos generales del análisis y diseño de sistemas.

V.6.1 Responsabilidad de la administración de la Calidad Total.

Las lecciones algo sorprendentes recolectadas de la administración de calidad total en situación de producción, son que el cliente es el único factor más importante para establecer y evaluar la calidad de los productos, este desarrollo tiene su equivalencia en lo que se ha encontrado en las investigaciones de MIS y DSS, en que resalta a importancia crítica del usuario para asegurar implementaciones de sistemas satisfactorias.

En términos prácticos una gran parte de la responsabilidad de la calidad de los sistemas de información cae con los usuarios del sistema y la administración. Deben de pasar dos cosas para que la TQM llegue a ser una realidad de los proyectos de sistemas. Primero, debe existir un completo soporte organizacional de la administración, apartándose de simplemente alabar las nuevas banalidades de administración. Esto significa el

establecimiento de un contexto para que la administración considere seriamente como la calidad de los sistemas de información y la información misma afecta a su trabajo.

Se necesita un compromiso temprano del analista y de la empresa hacia la calidad para lograr el objetivo de calidad. Esto da como resultado el ejercicio de un esfuerzo sostenido hacia la calidad a lo largo del ciclo de vida del desarrollo de sistemas, en vez de tener que verter enormes cantidades de esfuerzo para solucionar problemas al final del proyecto.

El soporte organizacional para la calidad de los sistemas de manejo de información se puede lograr por medio de proporcionar tiempo en el trabajo para círculos de calidad, que consisten de seis a ocho compañeros organizacionales encargados específicamente de considerar como mejorar los sistemas de manejo de información y cómo completar las mejoras.

En forma alterna, la administración de calidad total de MIS y DSS puede ser caracterizada a través del centro de información de la organización. El centro puede servir con él depósito del lineamiento de calidad, establecidos por círculos de calidad internos del MIS o propuestos de organizaciones competitivas dentro de la industria particular.

Mediante el trabajo de los círculos de calidad MIS, o por medio de otros mecanismos que ya existan, la administración y los usuarios deben desarrollar lineamientos para estándares de calidad del MIS. Preferiblemente, los estándares serán establecidos cada vez que sea formalmente propuesto por el equipo de análisis de sistemas un nuevo sistema o una modificación principal. No hay problema de que los usuarios que planean los lineamientos de calidad posean poca o ninguna experiencia de sistemas de computo. Se espera que los usuarios contribuyan con su conocimiento sobre la manera en que opera su departamento y lo que consideran calidad aceptable para la entrada, procesamiento y salida del sistema.

Lograr los estándares de calidad no es fácil pero es posible y se ha realizado. Parte del trabajo del analista de sistemas es motivar a los usuarios para cristalizar sus expectativas acerca de los sistemas de información y su interacción con ellos. Luego los

estándares de calidad departamentales deben ser comunicados por medio de retroalimentación del equipo de análisis de sistemas.

A veces el equipo se sorprende de lo que se desarrolla. Las expectativas son, por lo general, menos complejas de lo que los analistas experimentados saben que pudiera realizarse con un sistema. Adicionalmente, temas que han sido desechados o poco valorados por el equipo análisis pueden ser indicados como extremadamente importantes en los estándares de calidad de los usuarios. El hacer que los usuarios estén involucrados en definir los estándares de calidad del MIS, ayudará al analista a evitar errores costosos en desarrollos de sistemas no deseados o innecesarios.

Concluyendo lo referente a la administración de la calidad total (TQM) en SICLO-EA se busca estar dentro de los estándares establecidos por Windows, tanto de diseño como de desarrollo, en la Base de Datos (Back End) y en la interfaz gráfica del usuario (Front End).

V.7 Elección del manejador de base de datos y lenguaje de programación

La elección del lenguaje de programación o de la base de datos dejó de ser un problema ya que el cliente nos dio la única opción para el desarrollo del Sistema, al desear un estándar para trabajar en Microsoft Windows y tomando en cuenta los requerimientos del Sistema se decidió por Microsoft Access 97.

Microsoft Access 97 es una herramienta que ya se contaba con ella instalada y en la que uno de los programadores tenía cierta experiencia con la programación orientada a eventos (*Access Basic*) que es similar a programar en Visual Basic Versión 4.0, cabe mencionar que hubo una etapa de capacitación para los integrantes restantes que poco conocían de esta forma de programación.

En el Apéndice A se presenta una breve descripción de Access 97

V.8 Documentación del código

La documentación del código puede resultar una clara guía de comprensión durante la fase de mantenimiento por lo que se decidió incluir como Apéndice C del presente trabajo.



Capítulo VI

*Pruebas, Liberación
y Mantenimiento del
SICLO-FA*

PRUEBAS, LIBERACION Y MANTENIMIENTO DEL SICLO-EA

Introducción

Durante esta última fase del proyecto se realizarán al SICLO-EA diversas pruebas entre ellas están las pruebas de Unidad, pruebas de Integración, pruebas de Validación y pruebas del Sistema.

Las pruebas de cualquier Sistema implica poner en marcha el Sistema para asegurar que produzca las salidas esperadas y apropiadas para una determinada entrada. Las pruebas son un factor importante para asegurar la garantía de calidad del Sistema.

Uno de los principales objetivos para el desarrollo de las pruebas es someter el trabajo del programador a una serie completa de pruebas que no han sido diseñadas ni ejecutadas por los programadores y que se desarrollan en un medio lo más cercano posible a la realidad con un mínimo de simulación.

Otro de los puntos que se tocan en este capítulo es el de mantenimiento donde el objetivo es hacer la mejor planeación posible del Sistema para que el mantenimiento, sea nada más para perfeccionamiento o para agregarle nuevos módulos. El mantenimiento muchas veces se subestima pero este siempre debe tomarse en cuenta ya que es uno de los costos del Software que mayor peso tiene en el costo del proyecto.

VI.1 Definición y tipos de pruebas

En particular una **prueba** es un proceso de ejecución de un programa con la finalidad o intención de descubrir un error. Un buen ejemplo de prueba es aquel que tiene gran probabilidad de encontrar un error no descubierto hasta el momento.

Si una prueba no encuentra un error, el éxito será de la parte del programa en cuestión y no de la prueba misma, ya que el éxito de una prueba consiste en encontrar errores.

VI.1.1 Objetivo de las pruebas

Como se mencionó al inicio de este capítulo el principal objetivo de la etapa de pruebas del Sistema es someter el trabajo del programador a una serie completa de pruebas que no han sido diseñadas ni ejecutadas por los programadores y que se desarrollan en un medio lo más cercano posible a la realidad.

Si se encuentran una gran cantidad de errores que requieran modificaciones en el diseño, la calidad y la fiabilidad del Software no son muy confiables y quedan en entredicho, de tal forma que es necesario seguir realizando más pruebas para la detección de más errores.

Si por el contrario, el funcionamiento del Software parece ser el correcto y los errores que se encuentran son menores, la calidad y la fiabilidad del Software son aceptables. Por otro lado si durante la prueba no se descubriesen errores, queda también la sospecha de que las pruebas no fueron adecuadas o suficientes y que el Software puede estar defectuoso lo cual deberá corregirse en la fase de mantenimiento del Sistema.

Determinar una lista de pruebas no es nada fácil ya que existen o pueden haber cientos de pruebas a realizar. Los casos de pruebas que se necesitan deberán ser decididos por los integrantes del equipo de trabajo y la idea es que un tipo de prueba cubra una zona

funcional específica, por ejemplo “*iniciación*”, “*procesamiento de errores con respecto a las entradas*”, “*detección de un solo objetivo*”, “*rastreo del objetivo*”, etc.

VI.1.2 Etapas para el Proceso de Pruebas

Durante el proceso de prueba se pueden identificar cinco etapas:

1. *Prueba de Funciones*. La prueba de funciones o de unidades es el nivel básico en donde se prueban las funciones que componen un módulo para garantizar que operan de manera correcta. Aquí cada función debe tener una sola especificación definida con claridad. Las funciones no deben depender de otras funciones de su mismo nivel para posibilitar la prueba de cada función como una entidad aislada, sin la presencia de otras funciones.
2. *Prueba de Módulos*. Un módulo se compone de varias funciones que pueden cooperar entre sí. Después de haber probado cada función individual, es necesario probar la cooperación de estas funciones cuando componen un módulo. Debe ser posible probar un módulo como una entidad aislada, sin la presencia de otros módulos del Sistema.
3. *Prueba de Subsistemas*. Esta prueba es el siguiente paso del proceso en el cual los módulos se agrupan para formar subsistemas. Puesto que los módulos cooperan y se comunican, la prueba de subsistemas se debe centrar en la prueba de las interfaces de aquéllos, dando por hecho que los módulos son correctos.
4. *Prueba del Sistema*. La prueba del Sistema (*llamada también prueba de Integración*) se lleva a cabo cuando se integran los subsistemas para conformar el Sistema completo. En esta etapa, el proceso de prueba tiene que ver con el hallazgo de errores en el diseño y desarrollo. También se relaciona con la confirmación de que el Sistema total proporciona las funciones especificadas en

los requisitos y que sus características dinámicas cumplan con las planteadas en la definición de requisitos.

5. *Prueba de Aceptación.* Hasta esta etapa, todas las pruebas se realizan mediante el empleo de datos generados por la organización encargada de construir el Sistema. La prueba de Aceptación del Sistema se efectúa con datos reales, la información con la que el Sistema deberá operar. El proceso de la prueba de Aceptación a menudo descubre errores en la definición de requisitos del Sistema. Los requisitos pueden no reflejar las características y el rendimiento reales requeridos por el usuario, y la prueba de Aceptación puede revelar que el Sistema no muestra el rendimiento y la funcionalidad previstos.

La fase de prueba se debe especificar y establecer de manera formal en el proyecto. Es deseable empezar la planeación de las pruebas en una etapa relativamente temprana del proceso del desarrollo del Software. Las pruebas de Módulos y Subsistemas se deben planear a medida que se formula el diseño del subsistema, las Especificaciones de las pruebas del Sistema y las de Aceptación se deben preparar en la etapa de diseño o durante la aplicación.

Para las etapas identificadas anteriormente, se debe seguir una estrategia para la aplicación de la prueba, independientemente del diseño de esta. El siguiente esquema (*figura VI.1*) muestra el flujo de información en la prueba que es necesaria para llevar a cabo la prueba.

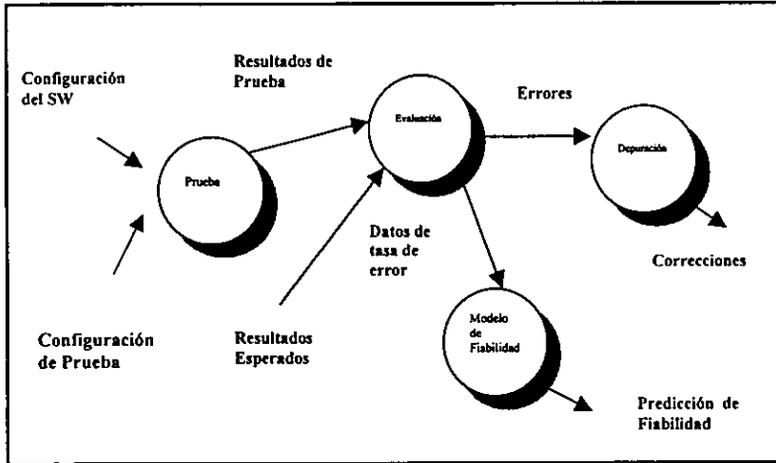


Figura VI.1 *Flujo de Información para las pruebas*

VI.1.3 Diseño de Pruebas

Cada caso de prueba consiste de un guión, datos de entrada/salida y resultado de la misma.

- Un **guión** es una serie de instrucciones paso por paso destinadas a guiar al personal que conducirá la prueba en una forma casi mecánica a través de las pruebas. El guión contiene una lista de todas las acciones requeridas de los operadores humanos en cada pieza del equipo que participa en la prueba. No solo dice a los que aplican la prueba qué deben hacer y cuándo hacerlo, sino también

qué de debe buscar y de que se debe tomar nota para realizar posteriormente un análisis.

- La sección de datos de un caso de prueba incluye datos de entrada simulados, datos de entrada reales y datos de salida previstos.
 - a) Los datos de entrada simulados son datos preparados con anticipación, el fin es ejercitar el Sistema durante una prueba determinada.
 - b) Las entradas reales son las que no pueden ser convenientemente preparadas con anticipación. Esta clase de entradas proporciona ventajas que difícilmente pueden obtenerse de los datos de entrada preparados, simulados, incluyendo aleatoriedad y la probabilidad de cierto volumen de “basura” que nuestro Sistema tendrá que procesar correctamente.
 - c) Las salidas previstas son pronósticos escritos de los datos exactos que deben resultar de un caso de prueba determinado, en donde es posible tal pronóstico. Si los resultados se predicen con anticipación sólo será necesario comparar los resultados reales de prueba con los previstos para determinar la efectividad de la prueba.
- El resultado de la prueba reflejará el éxito de esta en caso de encontrar errores o el éxito de la parte del sistema que se esta probando. De estos resultados dependen las acciones para corregir el o los errores encontrados o la aceptación de la parte en cuestión.

VI.1.4 Pruebas para el SICLO-EA

Para iniciar el proceso de pruebas el equipo seleccionado para llevar a cabo dicho proceso debe ser muy similar en sus características al del equipo para el cual se diseñó y construyó este Sistema. En caso contrario no es posible garantizar la efectividad de las pruebas.

Tomando en cuenta las etapas del proceso de pruebas mencionadas en el punto VI.1.2 de este trabajo, se definió realizar los siguientes tipos de prueba:

- a) Prueba con datos previamente validados para comprobar la efectividad de los resultados.
- b) Prueba con datos no válidos para comprobar mensajes de error.

Para realizar cada prueba ya sea para módulos, procedimientos o controles de Microsoft Access 97 se estableció el siguiente guión de prueba:

- 1.- Ejecutar el código hasta la parte que se está probando usando las herramientas de depuración de Access Basic.
- 2.- Introducir datos validados previamente, teniendo los resultados correctos a la mano.
- 3.- Confrontar los resultados del SICLO-EA con los resultados correctos.
- 4.- Si los resultados coinciden pasar al punto 6.
- 5.- Si no coinciden los resultados o hay errores de programación, corregir y aplicar pruebas hasta obtener la aceptación con datos válidos.

6.- Introducir datos no válidos esperando mensaje de error correspondiente.

7.- Si no hay problema para obtener mensaje de error correspondiente y no se generan problemas de programación aceptar la parte del SICLO-EA que se esta probando.

8.- Si no se obtiene el mensaje de error correcto o hay errores de programación se deberá corregir y aplicar pruebas hasta obtener aceptación con datos válidos o no válidos.

Además de las pruebas anteriores se realizaron las pruebas siguientes:

- **Prueba de Integración**

La prueba de Integración para el SICLO-EA se realizó de forma pausada conforme se desarrollaba cada uno de los módulos, es decir, se utilizó la integración ascendente.

En algunos casos se detectó duplicidad de información y se tuvo que corregir este problema el cual hizo mas eficiente el Sistema.

- **Prueba de Seguridad**

Debido a que el SICLO-EA es un Sistema basado en computadora que maneja información sensible ya que se manejan datos referentes a pagos de inscripción al evento en curso y el manejo inadecuado de estos datos pueden afectar los resultados de informes y en consecuencia perdida de dinero fue necesario realizar las pruebas de Seguridad la cual intenta verificar que los mecanismos de protección incorporados en el Sistema lo protegerán, de hecho de la penetración impropia.

Durante esta prueba de Seguridad , el encargado de la prueba desempeñó el papel de un individuo que desea entrar al Sistema en la cual todo fue valido, se intentó entrar con diferentes cadenas en el *login* y *password*. Esta prueba falló en la seguridad a nivel de usuario como también en la de acceso al Sistema pero al llegar a la parte final de seguridad que se encuentra dentro del Sistema SICLO-EA la prueba detecto algunos errores que se tenían, estos fueron oportunamente corregidos a tiempo, aunque sabemos que con el suficiente tiempo así como los recursos, una buena prueba de seguridad terminará por penetrar en el Sistema. En este caso el papel del diseñador del Sistema es hacer que el coste de penetración sea mayor que el valor de la información obtenida mediante la penetración.

Cabe señalar que se aplicó también una prueba de *aceptación final del SICLO-EA*, que consistió en ponerlo a disposición para el evento del Segundo seminario Internacional de Ingeniería de Sistemas que se efectuó en Huatulco Oaxaca y de donde tal prueba se generaron algunas sugerencias no contempladas en los requerimientos ni objetivos iniciales del Sistema, por lo cual se tomaron en cuenta para el mantenimiento perfecto del mismo.

De esta prueba se obtuvieron resultados satisfactorios de tal modo que el Sistema puede ser usado para el próximo evento académico que tenga la DEPFI.

VI.2 Mantenimiento del Software

El mantenimiento del Software se refiere al proceso de modificar un programa cuando éste ya se ha entregado y se encuentra en operación. Esas modificaciones pueden implicar cambios sencillos en la codificación, cambios mayores de diseño o de especificación de requisitos.

El mantenimiento del Software puede tener como objetivo corregir errores, proporcionar nuevas posibilidades o ambas cosas; de lo cuál se derivan tres actividades de mantenimiento.

1. La primera actividad de mantenimiento es debida a que no es razonable asumir que la prueba del Software haya descubierto todos los errores latentes de un gran sistema de Software. Durante el uso de cualquier gran programa, se encontrarán errores, siendo informado el equipo de desarrollo. El proceso que incluye el diagnóstico y la corrección de uno o más errores se denominan *mantenimiento correctivo*.
2. La segunda actividad que contribuye a la definición de mantenimiento se produce por el rápido cambio inherente a cualquier aspecto de la informática. Se anuncian nuevas generaciones de hardware en ciclos de unos 24 meses; regularmente aparecen nuevos sistemas operativos o nuevas versiones de los antiguos; frecuentemente, se mejoran o modifican los equipos periféricos y otros elementos de los sistemas. Por otro lado la vida útil del Software de aplicación puede fácilmente superar los diez años, sobreviviendo al entorno del Sistema para el que fue originalmente desarrollado. Por tanto, el *mantenimiento adaptativo* una actividad que modifica el Software para que interaccione adecuadamente con su entorno cambiante es tan necesario como usual.

3. La tercera actividad que se puede aplicar a la definición e mantenimiento se produce cuando un paquete de Software tiene éxito. A medida que se usa el Software, se recibe de los usuarios recomendaciones sobre nuevas posibilidades, sobre modificaciones de funciones ya existentes y sobre mejoras en general. Para satisfacer estas peticiones, se lleva a cabo el *mantenimiento perfecto*. Esta actividad contabiliza la mayor cantidad de esfuerzo empleado en el mantenimiento de Software.

Un estudio realizado por Leintz y Swanson (1980) descubrió que alrededor del 65% del mantenimiento era de perfeccionamiento, el 18% adaptativo y el 17% correctivo.

Para adaptar o perfeccionar, debemos determinar nuevos requisitos, rediseñar, generar código y probar el Software existente. Tradicionalmente, tales tareas, cuando se aplican a un programa existente, han sido denominadas mantenimiento.

VI.2.1 Costos del mantenimiento

La etapa de mantenimiento, durante el ciclo de vida del Sistema, constituye la mayor parte del tiempo y costos empleados.

Los costos de mantenimiento son muy difíciles de estimar con anticipación debido a que éstos generalmente se subestiman al diseñar y aplicar los sistemas.

Por lo tanto es importante tomar en cuenta que los programas se deben diseñar, desarrollar y probar para que el mantenimiento de estos sea al menor costo y esfuerzo.

El mantenimiento de Software, también implica una reducción de la productividad (*medida en líneas de código por persona al mes*) ya que el esfuerzo empleado en el

mantenimiento se puede dividir en actividades productivas (p. Ej. : *análisis y evaluación, modificación del diseño, codificación*) y actividades “menos productivas” (p. Ej. : *tratar de comprender lo que es el código; tratar de interpretar las estructuras de datos, las características de la interfaz, los límites de rendimiento*).

VI.2.2 Facilidad de mantenimiento

La facilidad de mantenimiento se puede definir cualitativamente como la facilidad de comprender, corregir, adaptar y/o mejorar el Software. Como ya se ha insistido, la facilidad de mantenimiento es un fin clave que guía los pasos de cualquier metodología de ingeniería de Software.

VI.2.3 Factores que afectan la facilidad de mantenimiento

La facilidad de mantenimiento que se consigna para el Software se ve afectada por muchos factores. Aparte de lo que ya se ha mencionado respecto a una falta de cuidado en el diseño, en la codificación o en las pruebas, también se tiene impacto negativo durante el mantenimiento, si no se cuenta con:

- Disponibilidad de una plantilla de Software calificada.
- Estructura del Sistema comprensible.
- Facilidad de manejo del Sistema.
- Uso de lenguajes de programación estandarizados.
- Uso de sistemas operativos estandarizados.
- Disponibilidad de los casos de prueba.

- Facilidades de depuración incorporadas.
- Disponibilidad de una computadora apropiada para llevar a cabo el mantenimiento.

Además de estos factores, se podría añadir la disponibilidad de la persona o grupo que haya desarrollado originalmente el Software.

Probablemente, el factor más importante que afecte a la facilidad de mantenimiento es la planeación condicionada al mantenimiento. Si se ve el Software como un elemento del Sistema que inevitablemente estará sujeto a cambios serán substancialmente mayores las posibilidades de producir un Software que se pueda mantener fácilmente.

La revisión de mantenimiento más formal se da al terminar la prueba y se denomina revisión de la configuración. Dicha revisión asegura que todos los elementos de la configuración del Software son completos, comprensibles y dispuestos para un control de modificación.

VI.2.4 Mantenimiento para SICLO-EA

El mantenimiento que se prevé para el SICLO-EA, consiste en agregar a medida que sean requeridas más operaciones que correspondan a otros tipos de análisis económicos no contemplados hasta el momento.

El impacto de este mantenimiento perfectivo no generará mayor problema que simplemente agregar módulos y hacer una nueva compilación el Sistema, debido a la modularización que se diseñó para el Sistema, además las operaciones ya programadas pueden ser usadas por cualquiera de estos posibles nuevos análisis.

CONCLUSIONES

Es importante el saber si se cumplieron los objetivos que inicialmente se plantearon así como los alcances y los requerimientos del usuario, por todo esto es importante el desarrollo de conclusiones. Un aspecto que también debemos conocer son los beneficios que se proporcionaron o seguirán proporcionando a la empresa o cliente al que se desarrollo este proyecto de tesis.

Con respecto a los objetivos planteados en el análisis del sistema (SICLO-EA) podemos decir que fueron cubiertos en su totalidad satisfactoriamente puesto que se diseño y desarrollo un sistema con los atributos de calidad requeridos: confiabilidad, portabilidad, eficiencia, solidez y corrección.

Se desarrollo un sistema que ya contribuye en el control y logística de eventos académicos el cual simplifica y agiliza la administración en dichos eventos, el SICLO-EA genera informes que son confiables y que pueden ser analizados y utilizados por el personal del comité organizador de los eventos. El sistema también nos permite tener un almacenamiento cronológico de la información es decir de eventos anteriores donde el sistema ha sido utilizado.

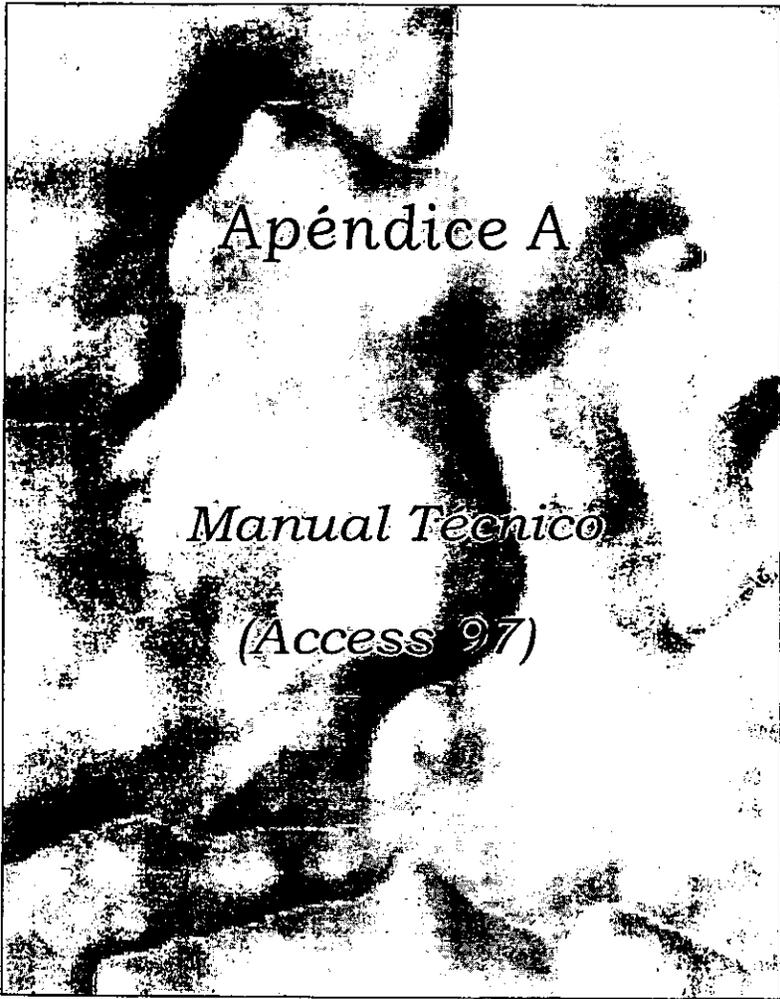
En cuanto a los alcances del sistema enfocados a facilitar la administración y logística de eventos académicos y la generación de información confiable y completa han sido cubiertos.

Los beneficios que seguirá proporcionando SICLO-EA en la DEPFI son que este sistema podrá ser utilizado en eventos posteriores y no solo para el Segundo Seminario Internacional de Ingeniería de Sistemas que se lleva a cabo cada año y que en esta ocasión se realizó en las bahías de Huatulco, en el estado de Oaxaca, México.

Consideramos las fases de análisis y diseño como las más importantes para el desarrollo de SICLO-EA ya que en ellas se determina el éxito del proyecto, así como las más difíciles de evaluar ya que no se pueden determinar parámetros antes de continuar con la siguiente etapa.

Al concluir el presente trabajo nos damos cuenta de la importancia de la optimización de recursos y tiempo así como de contar con diversas alternativas para la solución de un problema no solo de computación sino cualquier problema que puede ser de la vida diaria.

Con todo lo anterior, el Sistema para el Control y Logística de Eventos Académicos (SICLO-EA) ha sido diseñado e implantado en el Departamento de Estudios de Posgrado de la Facultad de Ingeniería (DEPFI) particularmente en el área de Sistemas.



Apéndice A

Manual Técnico

(Access 97)

MICROSOFT ACCESS 97

El contenido de este apéndice es un apoyo técnico que le servirá para conocer las principales características de Microsoft Access 97. Aquí podrá encontrar las ventajas y desventajas que tiene Access 97, las características generales de cada bloque que compone a Access, los mínimos requerimientos de Hardware y Software para su instalación y buen funcionamiento

Microsoft Access 97 es un sistema de administración de bases de datos relacionales estándar para Microsoft Windows en este apartado usted podrá encontrar la información suficiente para adentrarse al entorno de Access 97 desde como abrir una base de datos, insertar y modificar datos, pero lo principal que de una forma sencilla podrá manipular el Sistema que fue desarrollado para la DEPFI el SICLO-EA si se desea dar mantenimiento en un futuro.

ACCESS 97 UNA BASES DE DATOS RELACIONAL

¿Porqué Access 97 ?

Access 97 ha llegado a ser un Sistema de administración de bases de datos relacionales que se ha incorporado a los ordenadores personales o PC debido a la simplicidad de su estructura y manejo.

Además Access 97 es una base de datos relacional estándar para Microsoft y por lo tanto podemos ejecutarlo desde Windows bajo un entorno gráfico bastante amigable. Bajo este entorno puede crear y modificar tablas, formatos, informes, consultas y macros, además le permite pasar objetos de un área de trabajo a otra, unir dos o más tablas con solo dibujar una línea entre los campos comunes, trabajar con todos los datos, ordenarlos según desee y elaborar todo tipo de informes.

Otra de sus ventajas es la posibilidad de incluir objetos OLE ("*Objet Linking and Embedding*", "*Vinculación e incrustación de objetos*") en registros, lo que le permite incluir sonidos, animación, documentos de un procesador de textos, una hoja electrónica o una imagen gráfica.

Access 97 permite importar datos de ficheros de otros gestores de bases de datos, como dBASE, FoxPro, Excel, Lotus y Paradox, y combinarlos hasta conseguir una base de datos con todo tipo de datos.

En Access 97 existe la posibilidad de usar Vínculos con Office (*Office Links*) de Microsoft. Estos son asistentes integrados que le ayudarán a utilizar la información de una aplicación de Office como puede ser Word, Excel o Internet. Además Access 97 le proporcionará dos tipos generales de seguridad de bases de datos. Uno que es muy seguro es el habilitar la base de datos con una contraseña que aparecerá al intentar abrir la base de datos, si no escribe la contraseña correcta la base de datos no se abrirá. El segundo tipo de seguridad se conoce como seguridad de nivel de usuario, es más complejo y mas amplio que solo definir una contraseña. Esta forma de seguridad es similar a la seguridad de sistemas de redes. Cuando abra estas bases de datos, usted como usuario deberá introducir una identificación de usuario y una contraseña.

En definitiva, Access 97 aprovecha todas las ventajas de Microsoft Windows 95 y de una forma fácil y rápida le permite trabajar con cualquier base de datos.

Requisitos para trabajar en Access 97

Para poder usar Access 97 usted debe tener como mínimos requerimientos de Software y Hardware lo siguiente:

☐ Un procesador 486 o superior.

☐ 8 Mb en RAM para correr Windows 95 con una aplicación y 12 Mb para correr Access 97, pero requiere mas memoria para el uso de mas aplicaciones como puede ser Microsoft Explorer, Outlook, etc.

☐ 16 Mb en RAM para el uso de Windows NT pero es necesaria mas memoria para otras aplicaciones.

☐ Una unidad de disquetes de 3 ½" o una unidad de CD-ROM si la instalación lo necesitara.

☐ Monitor VGA o SVGA de 256 colores.

☐ Windows 95

☐ Windows NT versión 4.0 o versión 3.51 con Service Pack 5.

Para poder ejecutar Access cómoda y exitosamente, necesita una computadora con procesador Pentium y 12 megabytes de RAM mientras mas RAM tenga, más aplicaciones pueden ejecutarse de manera simultánea, y su rendimiento general será mejor. También necesita entre 14 y 45 MB de espacio en su disco duro, dependiendo lo que elija instalar además de espacio adicional para las bases de datos que creará. Access 97 requiere un monitor VGA, aunque mucho mejor sería un SVGA, y se recomienda una tarjeta de video rápida. Para cualquier trabajo con Access 97 es necesario un ratón o algún otro dispositivo para apuntar, y para imprimir cualquier cosa a partir de Access, requerirá de una impresora o acceso a una impresora dentro de su red.

Access 97 requiere que esté instalado Windows 95 o Windows NT 4.0 en su computadora.

Capacidades y limitaciones de Access 97

Access 97 es una herramienta muy eficaz para manejar bases de datos sin embargo existen algunas limitaciones o inconvenientes que vale la pena aclarar .

Por ejemplo, Access 97 no tiene integrado un proceso de respaldo así como Microsoft SQL Server tiene capacidades de respaldo integradas es decir que cuando se lleve a cabo el respaldo no se requiere que todos los usuarios tengan que salirse de la base de datos. Access sí requiere que todos los usuarios que se encuentren conectados a la base de datos cierren el archivo de la base antes que pueda hacerse manualmente un respaldo. Existen procedimientos de respaldo automáticos que podrían hacer el respaldo pero siempre y cuando todos los usuarios hayan cerrado su conexión con la base de datos.

Por otra parte Access tiene un límite de 1.2 Gigabytes de datos por base de datos para una Sistema de bases de datos pequeño es suficiente pero para una gran empresa, probablemente no tendrá espacio suficiente de almacenamiento de datos.

En la seguridad de nivel de usuario Access 97 tiene un límite de 255 usuarios totales. De nueva cuenta si esta en un ambiente donde existe una red grande, con cientos y quizá miles de usuarios que deben tener acceso a los datos, probablemente necesitará escoger un Sistema de base de datos distinto.

Instalando Access 97

Actualizaciones a Access 97 a partir de Access 2 o Access 95

Si usted cuenta con una copia de Access 2.0 o de Access 95, entonces puede adquirir e instalar una versión de actualización de Access 97 (Versión 8.0). Si no es propietario de ninguna de las versiones anteriores de Access, necesitará para instalarlo una copia normal de Access 97. Una vez actualizada la nueva versión de Access usted podrá seguir utilizando sus bases de datos de versiones anteriores pero no podrá hacer ningún cambio a dichas bases de datos si no las ha convertido a Access 97. Una vez convertida una base de datos, no podrá reconvertirse a alguna versión anterior de Access, por lo anterior si existen personas que seguirán utilizando la versión anterior, quizá deba instalar la versión anterior en un sitio distinto y de esta forma podrán trabajar con ambas versiones de su base de datos.

El proceso de instalación de Access es similar al de otros productos de Windows 95.

Si está utilizando discos flexibles se le pedirá que inserte los discos requeridos. Si está utilizando un CD-ROM, la instalación continuará de manera automática. Dependiendo de las opciones seleccionadas, la instalación puede tomar de 20 a 40 minutos. Cuando la instalación esté completa, regresará al escritorio de Windows 95. En su menú inicio aparecerá un programa nuevo, llamado Microsoft Access.

Debe tomar en cuenta que existen tres tipos de instalaciones y puede elegir a partir de las siguientes tres opciones:

Típica: Que instala Access 97 con las opciones más comunes, en el directorio especificado por usted.

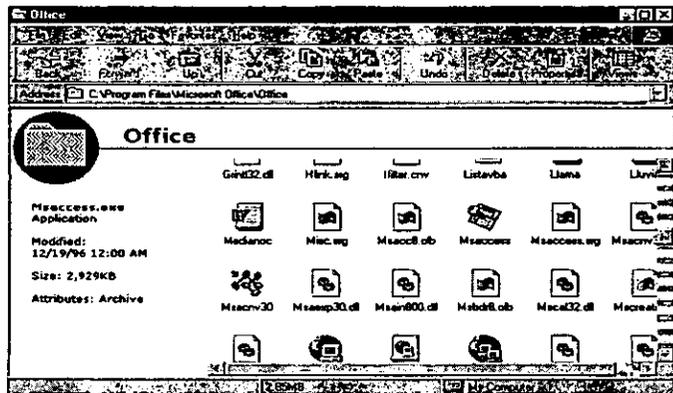
Compacta: Instala el número mínimo de archivos para ejecutar Access 97

Personalizada: Le permite escoger qué archivos desea instalar. Si usted desea tener las bases de datos de muestra deberá escoger esta opción personalizada.

Iniciando Access 97

Existen varias formas de iniciar Access 97. Las cuatro más sencillas son desde el menú de inicio de Windows 95, un icono de acceso directo, un icono Access y desde el Explorador de Windows. Si cuenta con Microsoft Office, también puede iniciar la base de datos Access desde la barra de herramientas de Office. La figura A.1 muestra el icono de Access en el cual puede hacer un doble clic para iniciar.

Figura A.1
Icono de Access que le permite cargar rápida y fácilmente Access 97.

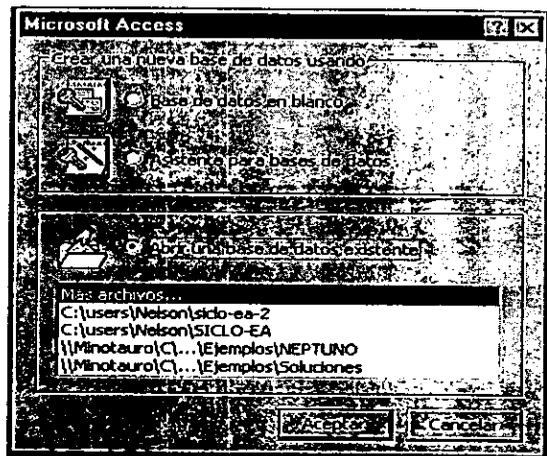


Cómo abrir una base de datos desde Access 97

Una vez cargado Access 97 lo primero que se encontrará es un cuadro de diálogo que le pregunta si desea abrir una base de datos o crear una nueva, según muestra la figura A.2.

Si selecciona abrir una base de datos existente, Access le preguntará qué base de datos desea abrir. Si lo que desea no aparece en la lista, seleccione Más archivos... y Access le permitirá escoger qué base de datos desea abrir.

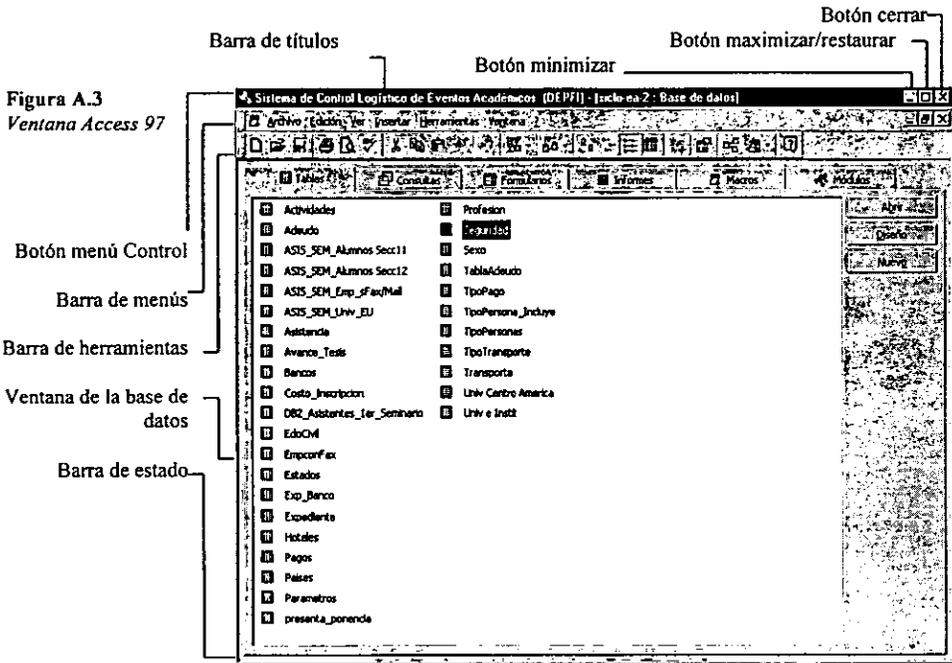
Figura A.2
Menú de apertura de Access 97.



Si selecciona cancelar, Access le mostrará la pantalla principal de Access 97 y podrá abrir una nueva base de datos desde la barra de menús de Access en la opción Archivos.

Ventana de Access 97

La pantalla que se ve en la figura A.3 es un ejemplo de la ventana de Access 97. Esta abierta en la base de datos SICLO-EA, con la ventana Base de datos maximizada. La ventana de Access es el centro de toda la actividad relacionada con su base de datos y está formada de varios elementos que vemos en la figura A.3.



Componentes de una base de datos de Access

Los objetos que en su conjunto forman lo que es la base de datos de Access son los que a continuación se mencionan:

Tablas. El objeto tabla es el componente más obvio de cualquier base de datos. Ahí es donde se almacenan los verdaderos datos. Una tabla es un conjunto de registros que se

pueden dividir en campos. Cada campo conserva una sola parte de la información en relación con el registro en el cual reside.

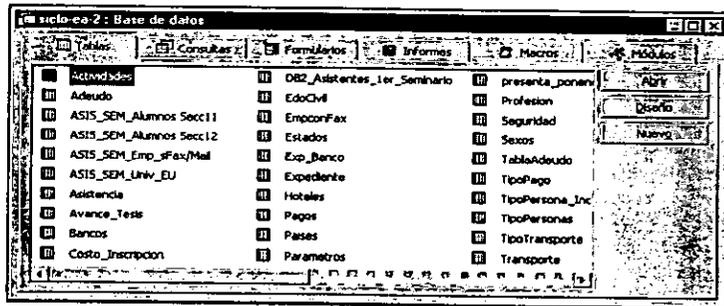
Una tabla es un conjunto de datos sobre un tema específico, como pueden ser productos o proveedores, si usted usa una tabla independiente para cada tema evita la duplicación de datos lo que hace a su base de datos más eficiente y reduce los errores de entrada de datos. Los datos en las tablas son presentados y organizados en columnas (*denominados campos*) y filas o renglones (*denominados registros*).

Su base de datos puede contener más de una tabla y estas tablas en su base de datos pueden estar relacionadas una con la otra. Cada tabla puede contener datos de diferentes tipos, como son texto, números, fechas, imágenes, animaciones, sonidos, etc.

Antes de crear una tabla es decir en su diseño usted debe especificar un identificador exclusivo de registros que en bases de datos se le denomina llave primaria que consiste en uno o más campos que identifican a cada registro almacenado en la tabla.

Con Access 97 puede usar la ventana bases de datos para crear una tabla nueva o ver las tablas que ya fueron elaboradas. Véase figura A.4.

Figura A.4
Ventana Base de Datos



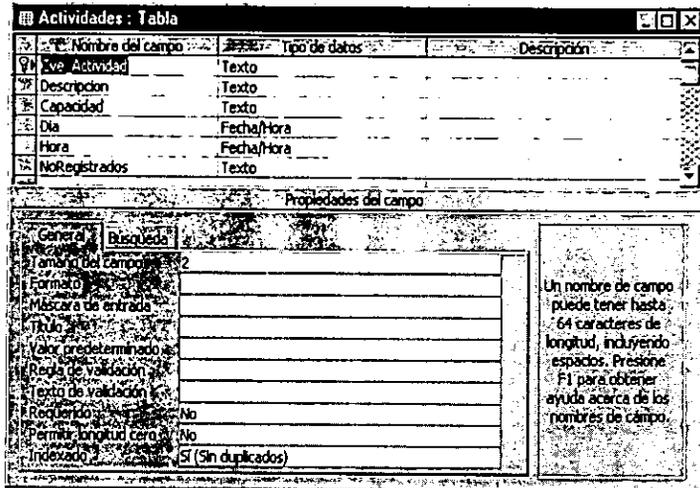
En la ventana Base de Datos puede abrir una tabla y ver su diseño así como crear una nueva tabla.

Una tabla se puede abrir en cualquiera de las dos presentaciones existentes: Diseño y Hoja de datos.

Vista Diseño

Una vez seleccionada una tabla en la ventana Base de datos haga clic con el botón derecho del ratón sobre ésta y elija la opción Diseño para esa tabla, o simplemente haga clic en la opción Diseño de la ventana Base de datos. Esto abrirá la vista Diseño para la tabla seleccionada, según muestra el ejemplo de la figura A.5

Figura A.5
Ventana Vista
Diseño

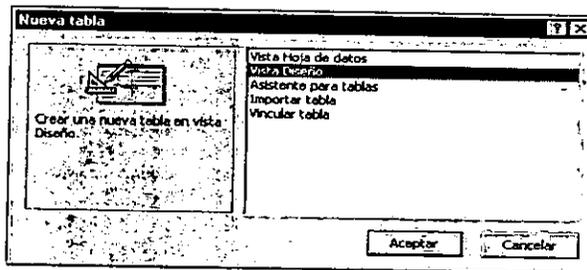


En la figura A.5 observamos la llave primaria de la tabla Actividades que es Cve_Actividad y es el campo que identifica a cada registro almacenado en la tabla.

La vista Diseño está organizada como un arreglo, la cuadrícula en la parte superior de la ventana vista Diseño nos dice todos los campos de la tabla, sus tipos de datos y una descripción opcional del campo. Las fichas en la parte inferior del formulario despliegan otras propiedades que se refieren al campo seleccionado en la cuadrícula.

Usted puede crear una nueva tabla dando un clic en la opción Nuevo de la Ventana Base de Datos se abrirá una nueva ventana la cual tiene la opción de un Asistente para tablas de donde se puede elegir de una gran variedad de ejemplos dejando que el Asistente cree automáticamente las tablas pero si usted está listo para crearla solo tiene que dar la opción Diseño dentro de esta nueva ventana (figura A.6).

Figura A.6
Ventana Nueva
Tabla



Los tipos de datos dentro de la columna Nombre de campo que vimos en la figura A.5 que especifica el tipo de información que se almacena en el campo pueden ser de cualquiera de los siguientes tipos:

- ✍ Texto: Que puede ser una cadena de 3 caracteres utilizados para almacenar datos alfanuméricos.
- ✍ Memo: Almacena campos de texto muy largos no se debe especificar un número máximo pero el límite es de 64,000 caracteres.
- ✍ Número: Almacena datos numéricos.
- ✍ Fecha/hora: Almacena fechas y horas existiendo diferentes formatos.
- ✍ Moneda: Este es un tipo de dato numérico que se utiliza para valores monetarios.
- ✍ Autonúmero: Es un tipo de datos especial para campos clave primarios estos son de solo lectura y se van insertando de forma automática para cada registro.
- ✍ Si/No: Almacena datos booleanos, que pueden contener solo dos valores.
- ✍ Objeto OLE: Un tipo especial de objeto como puede ser una fotografía.
- ✍ Hipervínculo: Almacena el texto para una dirección de hipervínculo como pueden ser direcciones de documentos html, archivos de red y archivos locales o también direcciones electrónicas *e-mails*.

Como puede ver los tipos de datos son muy variados prácticamente para cualquier ocasión.

Vista Hoja de datos

Ahora veamos la vista Hoja de datos que puede activarse desde la ventana Base de datos seleccionando la opción tablas y haciendo clic sobre la tabla a ver y a continuación en el botón Abrir, también puede hacerlo con un doble clic sobre el nombre de la tabla y esta se abre en la vista Hoja de datos como lo muestra la figura A.7

Figura A.7
Como se ve la tabla
Actividades en Vista
Hoja de datos.

Cve	Actividad	Descripción	Capacidad	Día	Hora	No Registrados
01		Inauguración de 20		6/27/98	9:00:00 AM	20
02		Sistemas I	25	6/26/98	3:00:00 PM	24
*						

La vista Hoja de datos despliega los datos utilizando una cuadrícula con formato de hoja de cálculo muy parecida a la de Excel donde las columnas representan los campos de la tabla y las filas son los registros de datos almacenados en la tabla.

En la ventana vista Hoja de datos usted puede colocar un registro nuevo, eliminar un registro, buscar y filtrar registros, copiar y mover registros así como editar los datos.

☒ Consulta de tablas de referencias cruzadas: Son un tipo especial de consulta de selección y resultan similares a las tablas dinámicas que se utilizan en Microsoft Excel y se pueden ver los datos resumidos sobre distintas categorías.

☒ Consulta de los n superiores: Otro tipo especial de consulta de selección son diseñadas para que regresen únicamente cierto número de registros.

☒ Consultas de unión: Consultas que combinan campos correspondientes, provenientes de varias tablas en un solo campo que regresa la consulta. En este tipo deberá devolverse el mismo número de campos de cada una de las tablas que intervienen en la unión.

Vista Hoja de datos para consultas

Cuando usted abra una consulta de la ventana base de datos ésta produce un despliegue muy similar, al producido cuando abre una tabla en vista Hoja de datos. Para abrir la consulta en vista Hoja de datos, seleccione la consulta deseada de la ficha *Consultas* y a continuación haga clic en *Abrir*. La figura A.10 muestra la hoja de datos correspondiente a la consulta seleccionada.

Figura A.10
 Vista hoja de datos de la consulta *Tabla_Expediente*

Tipo_Persona	Nombre	Ape_paterno	Ape_Materno	Puesto	Cve_audada	Cve_presp
01	Fernando	Favela	Lozoya	Presidente de	02	01
01	Soledad	Tomas	Zaragoza		02	02
01	Leopoldo	Cabrera			02	02
01	Luis Enrique	Bracamontes	Galvez		02	02
01	Rodolfo	Felix	Valdez	Presidente	02	01
01	Carlos	Valdés	Maniscal		02	02
01	Jorge	Gil	Mendieta	Jefe del Depart	02	02
01	Alejandro	Bracamontes	Camarena		02	02
02	Javier	Silvestre	Martínez	Coordinación de	02	02
02	José Manuel	Covarrubias	Solis		02	02
02	Abel	Herrera	Camacho		02	02
02	José Jesús	Acosta	Flores		02	02
02	Jaime	Gómez	Vega		02	02

Esta hoja le mostrará los datos existentes en la consulta y la hoja de datos tiene columnas que representan los campos de la consulta y filas que representan los registros devueltos por ésta.

Por el origen de los datos devueltos en esta consulta, no se pueden llevar a cabo ninguna actualización, inserción o eliminación de las filas en la hoja de datos.

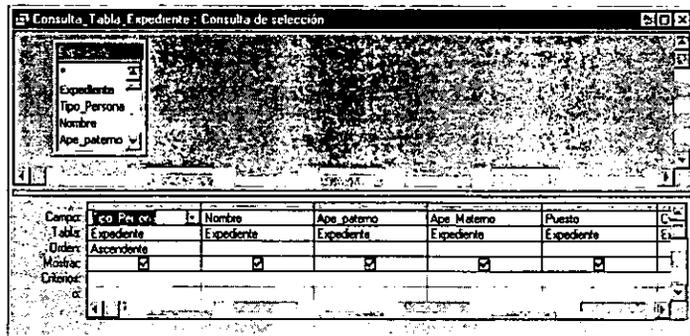
Vista Diseño en consultas

Para crear o modificar el diseño de una consulta ya existente deberá abrir el modo de presentación Diseño ahí es donde usted puede ver cómo se define la consulta, así como las propiedades de las columnas devueltas por la consulta. Las consultas tienen dos vistas de diseño: la Cuadrícula de Diseño en el menú y la barra de herramientas y la vista SQL.

La parte superior de la ventana Cuadrícula de Diseño se conoce como panel de tablas, porque despliega la tabla de Lista de campos correspondiente a cada una de las tablas que intervienen en la consulta.

La cuadrícula en la parte inferior de la ventana es la verdadera cuadrícula de Diseño QBE (*Consulta por ejemplo*). Aquí es donde definirá los campos devueltos por la consulta, establecerá el orden de clasificación especificará sus criterios. Ver figura A.11.

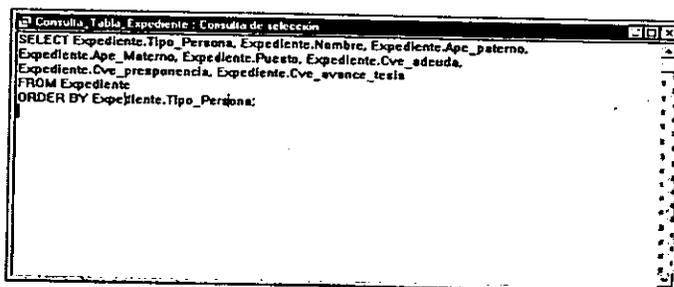
Figura A.11
Cuadrícula de
Diseño
correspondiente
a la consulta
Tabla *Expediente*.



Vista SQL

El uso de la vista SQL es una forma avanzada de examen de la consulta. Esta vista contiene el código SQL real que se ejecutará al operar la consulta. SQL son las siglas correspondientes a Lenguaje Estructurado de Consultas (*Structured Query Language*), que se utiliza para definir una consulta mediante palabras. Como otros lenguajes de computación, SQL tiene formato y palabras clave específicos. La figura A.12 muestra la vista SQL correspondiente a la consulta Tabla_Expediente de la base de datos SICLO-EA.

Figura A.12
Vista SQL
correspondiente a la
consulta
Tabla_Expediente.



```
Consulta: Tabla_Expediente: Consulta de selección
SELECT Expediente.Tipo_Persona, Expediente.Nombre, Expediente.Ape_paterno,
Expediente.Ape_materno, Expediente.Puesto, Expediente.Cve_adeuda,
Expediente.Cve_prespencia, Expediente.Cve_avance_testa
FROM Expediente
ORDER BY Expediente.Tipo_Persona;
```

La información que proporcionada por la vista SQL indica exactamente lo que está representando en vista Cuadrícula de Diseño QBE, pero en el lenguaje utilizado por Access 97 para que ejecute la consulta.

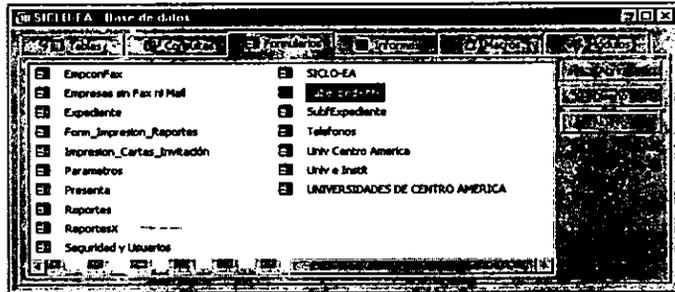
Para crear una consulta nueva puede utilizar el asistente o desde el principio en la vista Diseño de la consulta. En la vista Diseño, especifique los datos con los que desea trabajar agregando las tablas que contienen los datos y llenándolas a continuación en la cuadrícula de diseño.

Formularios

Un formulario es una de las características más eficaces de Access 97 en estos usted podrá escribir, editar y buscar nuevos datos así como imprimirlos, un formulario identifica los datos que se desean recopilar pero también existe la opción de introducir datos en la

base de datos utilizada específicamente esto lo hace en las tablas y el formulario es una forma más sencilla y amigable de ver los datos almacenados en las tablas. Para abrir un formulario siga los mismos pasos vistos en tablas y consultas ya que estos se encuentran junto a sus fichas en la ventana base de datos. La figura A.13 muestra como deberá tener su pantalla para abrir un formulario.

Figura A.13
Ventana base de datos con la ficha Formularios seleccionada que nos muestra todos los formularios de la base de datos SICLO-EA



Al hacer doble clic dentro de la ficha formularios en el formulario seleccionado como en el caso del ejemplo es Subexpediente este formulario se abrirá en su vista hoja de datos. En la figura A.14 aparece un formulario que muestra sus principales componentes. Como puede ver, el formulario tiene un cuadro de texto para introducir los nuevos datos, etiquetas para identificar los diversos elementos en el formulario y botones para llevar a cabo otras acciones.

La información que podemos ver en un formulario es la misma que se encuentra almacenada en las tablas, un formulario se crea a partir de los campos que se crearon en las tablas.

Los gráficos y otras herramientas de efectos especiales le ayudarán a crear formularios de buena apariencia y fáciles de usar.

Figura A.14
Formulario
Subpediente en
la base de datos
SICLO-EA.

The screenshot shows a Microsoft Access 97 form window titled 'Formulario'. The form is for the 'División de Estudios de Posgrado de la Facultad de Ingeniería' and is part of the 'Sistema de Control Logístico de Eventos Académicos'. The user is identified as 'Javier Sánchez Martínez'. The form includes the following fields and controls:

- Nombre:** Javier Sánchez Martínez
- Calle:** Calles López 23
- Celular:** Sin número
- Dirección:** Caracas
- País:** México
- Teléfono 1:** 5555555
- Celular:** [Empty field]
- Fax:** [Empty field]
- E-mail:** [Empty field]
- Estatus Civil:** [Dropdown menu]
- Edad:** 25 años
- Sexo:** Masculino
- Profesión:** [Empty field]
- Empresa:** DIFEM UNAM
- País:** Estados Unidos de México
- Teléfono 2:** 5-22-33-N

On the right side of the form, there is a portrait photo of a man with dark hair, wearing a suit and tie. The status bar at the bottom indicates 'Registro: 24 de 1' and '11:13:21 AM de 1998'.

En Access 97 están disponibles seis tipos básicos de formularios:

- Una sola columna
- Hojas de datos
- Tabular
- Subformularios
- Tabla dinámica
- Gráficos

Formularios de una sola columna

Los formularios de una sola columna muestran los campos de la tabla en la cual se basaron, en columnas. Estos formularios pueden ser más o menos de una pantalla completa, dependiendo de la resolución de su pantalla.

Los formularios de una sola columna son un buen ejemplo de una pantalla de introducción de datos, aunque los datos pueden introducirse directamente en las tablas el hacerlo desde un formulario puede ser más fácil, mucho más confiable y más productiva.

Formulario Hoja de datos

El formulario hoja de datos muestra los datos en formato de filas y columnas. En este modo de presentación puede ver, agregar y editar los datos.

Formularios tabulares

Este tipo de formularios se utiliza para ver varios registros a la vez, junto con todos los campos de estos registros. Este tipo de formulario resulta útil para ver o imprimir todos los registros o campos de una tabla.

En un formulario además de poder utilizar la información almacenada en las tablas se pueden desplegar datos referentes al nombre o logotipo de su empresa.

Subformularios

Un subformulario es un formulario dentro de otro formulario. Un subformulario lo puede utilizar para desplegar e introducir datos de varias tablas. En un formulario normal usted puede editar varias tablas pero si desea desplegar datos de varias consultas o tablas de manera simultánea necesita utilizar un subformulario.

Tabla dinámica

Una tabla dinámica es una tabla interactiva que realiza los cálculos elegidos, como sumas y cuentas, que esta basada en cómo están dispuestos los datos en la tabla dinámica.

Reciben este nombre porque en ellas es posible cambiar dinámicamente sus diseños para analizar los datos de diferentes formas, en ellas es posible reorganizar los encabezados de filas, encabezados de columnas y campos de página hasta que se consiga el diseño deseado. Cada vez que se modifica el diseño, la tabla dinámica inmediatamente vuelve a calcular los datos en función de la nueva organización. Además la tabla dinámica se actualiza cada vez que cambian los datos de origen.

Gráficos

Los gráficos y otras herramientas de efectos especiales le ayudarán a diseñar formularios de muy buena apariencia y de fácil manejo.

En la mayoría de los casos utilizará el Asistente para gráficos a la hora de crear un formulario gráfico el Asistente determina a partir de los datos especificados como mostrar los datos.

Para utilizar el Asistente para gráficos, debe estar instalado el Microsoft Graph 97, este no viene instalado si tiene un tipo de instalación típica de Access 97.

También puede crear un gráfico mediante la existencia de un gráfico existente de algún archivo externo.

Vista Diseño de Formularios

En el modo de presentación vista Diseño usted puede crear o modificar la estructura de un formulario existente. Es decir aquí usted puede efectuar todo el trabajo para indicar a Access qué datos desplegar y en qué formato. Para abrir la vista Diseño haciendo clic en él, desde la ventana Base de datos principal y, luego, en el botón Diseño del lado derecho de la pantalla. (*Ver figura A.13*)

La vista Diseño del formulario consta de tres secciones:

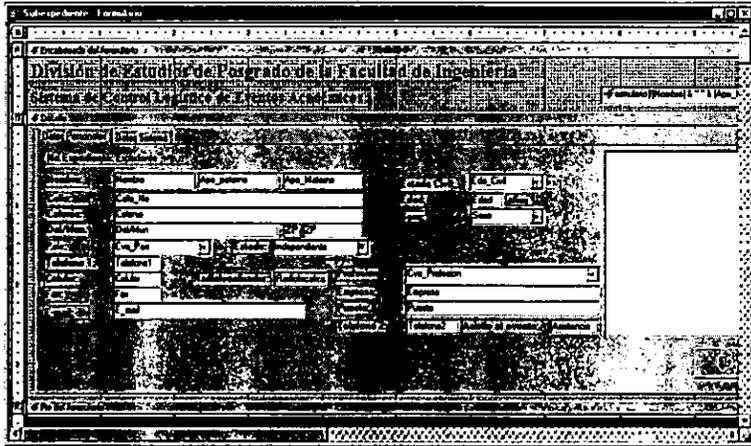
Sección Detalles: Que contiene el cuerpo principal del formulario, sus controles aparecen desplegados en esta sección y no se puede eliminar.

Sección Encabezado o pie de formulario: Aquí se coloca información como título, fecha u otra información deseable que se desplegará solo al inicio y/o final del formulario. Esta sección se puede eliminar y agregar, se despliegan los datos al imprimir o en la pantalla presentación preliminar.

Secciones Encabezado o pie de página: Contiene información como fecha, nombre del formulario, número de página u otra información deseable que se despliegue en la parte superior o inferior de cada pagina pero únicamente en el formulario impreso y no aparecerán en pantalla. Estas secciones pueden eliminarse o agregarse.

La utilización de la vista Diseño del formulario se observa en la figura A.15.

Figura A.15
 Vista Diseño del
 formulario
 Subexpediente de
 la base de datos
 SICLO-EA.



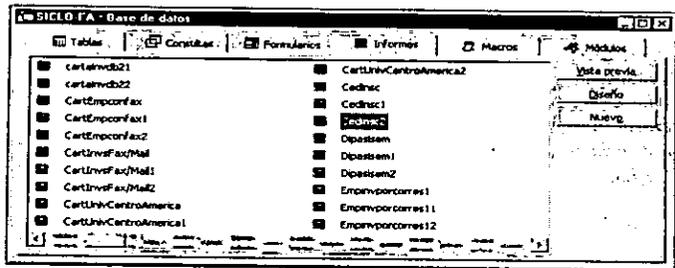
En un formulario usted podrá navegar, agregar y editar registros, eliminar registros, copiar los registros, localizar registros mediante un cuadro de diálogo Buscar, utilizar filtros y ordenar e imprimir datos; todo esto a partir de un formulario abierto.

Informes

Un informe es una forma sencilla y efectiva de presentar los datos en formato impreso. Estos le permiten enviar estos datos a cualquier tipo de sitio en un formato muy fácil de leer. Estos se pueden utilizar para imprimirlos en una hoja normal vía impresora o puede exportar el informe en diversos tipos de formato y por mencionar alguno el formato html o en un formato para enviarlos por correo electrónico.

Para abrir un informe usted debe seguir los mismos pasos antes mencionados para abrir tablas, consultas y formularios que se encuentran en la ventana Base de datos. Ver figura A.16.

Figura A.16
 Ventana Base de datos en la ficha Informes que presenta todos los informes de la base de datos SICLO-EA.



Los informes, al igual que los formularios utilizan como fuente de datos una tabla o una consulta que nos proporcionarán datos reales. En el diseño del informe usted creará la presentación de los datos al imprimirlo.

Al igual que en los formularios existen cuatro tipos de informes básicos:

- Informes Tabulares: Imprime los datos en filas y columnas.
- Informes de una columna: Imprime los datos como un formulario.
- Informes de combinación de correspondencia: Imprime cartas preimpresas o modelo.
- Informe de etiquetas de correo: Imprime etiquetas de varias columnas.

Los informes tabulares despliegan datos en filas y columnas, de manera parecida a una tabla sin embargo estos agrupan sus datos en uno o más valores de campo. Pueden incluir otros elementos como fechas y subtotales de campos numéricos. La figura A.17 muestra un ejemplo de un informe tabular.

Figura A.17
Ejemplo de un
informe
tabular.

Productos por categoría

Categoría: Bebidas		Categoría: Carnes		Categoría: Condimentos	
Nombre de producto:	En existencia:	Nombre de producto:	En existencia:	Nombre de producto:	En existencia:
Queso de Mozzarella	17	Empanada de carne	21	Ajón negro M. blanca	27
Cerveza Heineken Alambicada	125	Pañ chino	115	Especias Cajun del chef Anton	53
Cerveza Laughlin Lighthouse	92			Especias picantes de Louisiana	4
Cerveza negra Bluebird	26	Número de productos: 2		Mermelada de grosellas de la abuela	126
Cerveza Outback	15			Salsa de arándanos Northwoods	6
Cerveza Bassendish	111			Salsa de arándanos picante de Louisiana	76
Cerveza Heineken Beer	17			Salsa de sésamo en medio	36
Licor Chambéry	57			Salsa verde orgánica Frankfurter	32
Licor verde Chartreuse	96			Sambalich de vegetales	24
Té Chamomila	26			Sirope de arce	113
Vino Cabé de Sauv	17			Sirope de agave	13
Número de productos: 11				Número de productos: 11	

Los informes de una columna despliegan verticalmente los datos, con uno o más registros por página. Despliegan sus datos de una manera parecida a los formularios de entrada de datos pero sólo sirven para verlos e imprimirlos y no para la escritura de datos. La figura A.18 muestra un ejemplo de un informe de una columna para imprimir facturas.

Figura A.18
Ejemplo de un
informe de una
columna.

Factura

NEPTUNO

7000 Avenida 90, Mision 3801
Teléfono: (512) 994-4451 Fax: (512) 994-1234

Fecha: 24-Sep-98

Emisor: Rattlesnake Canyon Grocery
2817 Milton Dr.
Abuquerque NM 87110
Estados Unidos

Facturado a: Rattlesnake Canyon Grocery
2817 Milton Dr.
Abuquerque NM 87110
Estados Unidos

Id. de producto	Id. de cliente	Vendedor	Fecha de pedido	Fecha de entrega	Título de envío	Título de envío
11077	RATTC	Nancy Davila	05-Jun-98	03-Jul-98	United Package	

Id. de producto	Nombre de producto	Cantidad	Precio unitario	Descuento	Precio con descuento
2	Cerveza tibetana barley	24	\$ 18.00	20%	\$ 364.80
3	Sirope de agave	4	\$ 10.00	0%	\$ 40.00
4	Especias Cajun del chef Anton	1	\$ 22.00	0%	\$ 22.00
6	Mermelada de grosellas de la abuela	1	\$ 25.00	2%	\$ 24.50
7	Papas secas orgánicas del tío Bob	1	\$ 30.00	5%	\$ 28.50
8	Salsa de arándanos Northwoods	2	\$ 40.00	10%	\$ 72.00
10	Pañ chino	1	\$ 31.00	0%	\$ 31.00
12	Queso Manchego La Pastora	2	\$ 38.00	5%	\$ 72.20
13	Algas Kombu	4	\$ 6.00	0%	\$ 24.00
14	Cuajada de judías	1	\$ 23.25	3%	\$ 22.55

Se pueden utilizar informes de combinación de correspondencia para la impresión de cartas preimpresas o modelo, utilizando datos provenientes de su base de datos que quedan vinculados con Microsoft Word para ordenar e imprimir los documentos de combinación de correspondencia. Para que encuentre mas detalles acerca de estos informes puede consultar la Ayuda que le proporciona Access 97. Busque en Ayuda bajo Combinación de correspondencia para que le muestre un recorrido completo de los procedimientos.

El Informe etiquetas de correo lo puede utilizar para la impresión de etiquetas con la posibilidad de elegir diversos tamaños para las etiquetas disponibles en un correo. La figura A.19 muestra la pantalla vista preliminar del Informe Etiquetas de clientes, que es un informe para etiquetas de correo. La calidad y la facilidad de la impresión de estas etiquetas de correo dependerán de su impresora y un mejor resultado lo encontrará en una buena impresora láser o de chorro de tinta.

Figura A.19
Ejemplo de un Informe para imprimir Etiquetas de correo.

Etiquetas de cliente : Informe		
QUICK-Stop Taucherstraße 10 Cunewalde 01307 Alemania	Morgenstern Gesundkost Heerstr. 22 Leipzig 04179 Alemania	Alfreds Futurkiste Obere Str. 57 Berlin 12209 Alemania
Königlich Essen Maubelstr. 90 Brandenburgo 14776 Alemania	Toms Spezialitäten Luisenstr. 48 Münster 44087 Alemania	Ottlies Käseladen Mehrermerstr. 369 Köln 50739 Alemania
Drachentul Delikatessen Walsenweg 21 Aachen 52066 Alemania	Lehmanns Marktstand Magazinweg 7 Frankfurt 60528 Alemania	Blauer See Delikatessen Forsterstr. 57 Mannheim 68306 Alemania
Die Wandermühle Adenauerallee 900 Stuttgart 70563	Frankenversand Berliner Platz 43 München 80805	Cactus Comidas para llevar Cerrito 333 Buenos Aires 1010

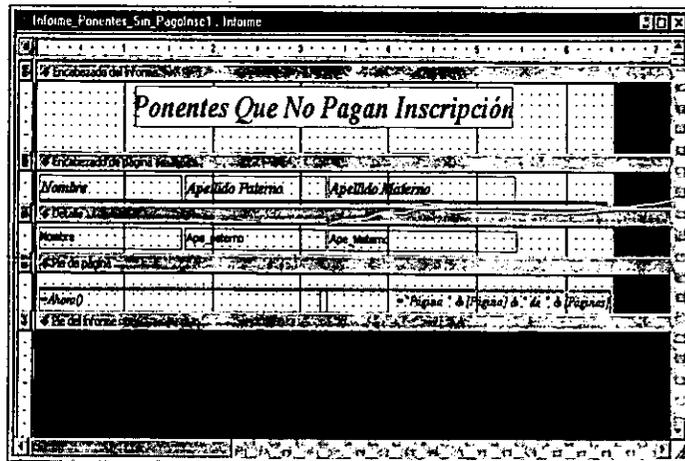
Los formularios y los informes son similares en muchos aspectos, aunque le servirán para propósitos muy distintos. La diferencia principal es la razón de los resultados.

Generalmente los formularios se utilizan para la entrada de datos y los informes se utilizan para ver los datos en pantalla o en papel es decir la salida de la información.

Vista Diseño de un Informe

Una vista Diseño en un informe consta de varias áreas. Los componentes principales en una vista Diseño del informe son la lista Campo y la hoja de Propiedades. La lista Campo contiene los campos de la tabla o consulta en la cual se basa el informe. La hoja de Propiedades es donde se pueden modificar o establecer las diversas propiedades del informe. El cuadro de herramientas es otro componente importante y se utiliza como una forma fácil de agregar controles y elementos de diseño al informe al igual que en un diseño de formulario. La figura A.20 muestra la vista Diseño de un informe.

Figura A.20
Informe de la base de datos SICLO-EA en su vista Diseño.



Al igual que en un formulario en un informe tenemos diferentes componentes o secciones distintas a la hora de diseñarlo estas secciones son: El encabezado del informe que es el encabezado principal de este, el encabezado de página que contiene información que se imprimen en la parte superior de cada una de las páginas, los encabezados de grupo

que no utilizan todos los informes, la sección detalle que es la sección principal del informe, los pies de grupo, el pie de página que contiene información a imprimirse al final de cada página y el pie de informe que es el pie principal del informe.

Vista preliminar

Cuando selecciona la vista preliminar de un informe se ejecuta la consulta con la cual se vincula el informe si es que existe y se desplegará la información actualizada. Si el informe se basa solo en una tabla se utiliza la misma información existente en esta. El informe se desplegará únicamente como aparece impreso y haya sido diseñado. En la figura A.21 se muestra una vista preliminar de un informe de la base de datos SICLO-EA.

Figura A.21
Vista preliminar del informe constancia de la base de datos SICLO-EA.

U.N.A.M.
 Facultad de Ingeniería

2º Seminario Internacional
 de
Ingeniería de Sistemas
 COMITE ORGANIZADOR

25 al 27 de Junio de 1999, Bahías de Huatulco, Oaxaca, México.

FACULTAD DE INGENIERIA
 DIVISION DE ESTUDIOS DE POSGRADO
 Ciudad Universitaria, México, D.F. C.p. 04510, Ap. 70256,
 Circuito Exterior, Edificio "B" 3er. Piso, Departamento de Ingeniería de Sistemas,

La cuota de inscripción al Seminario es por inscripción de \$1200.00.

CEDEULA DE INSCRIPCION

NOMBRE: _____ CARGO: _____
 INSTITUCION O EMPRESA: _____ PROFESION: _____
 RFC: _____ ESTADO CIVIL: _____ EDAD: _____
 CALLE: _____ COLONIA: _____ CIUDAD: _____
 DELEGACION O MUNICIPIO: _____
 ESTADO: _____ C.P.: _____ PAIS: _____
 TELEFONO PARTICULAR: _____ E-MAIL: _____
 TELEFONO OFICINA: _____ FAX: _____

Anexo a la presente envierlos cheque No. _____ del Banco _____ a favor de la Facultad de Ingeniería,
 División de Estudios de Posgrado, por la cantidad de _____

Existe la opción de poder enviar un informe mediante correo electrónico mediante una acción de Access llamada EnviarObjeto, pueden enviarse correos electrónicos mediante

el uso de Microsoft Exchange, de Microsoft Mail, del correo de Microsoft Windows para Trabajo en grupo y de otras aplicaciones de correo electrónico que admitan la interfaz estándar Microsoft MAPI,

Si tiene una aplicación de correo electrónico que usa el protocolo Correo independiente del proveedor (*VIM*) y tiene instalada y configurada la biblioteca de vínculo dinámico (*Mapivim.dll*) que convierte los mensajes de correo MAPI al protocolo VIM, puede enviar objetos de Microsoft Access a la aplicación de correo VIM.

La acción EnviarObjeto sólo estará disponible si hay una aplicación de correo electrónico compatible con MAPI instalada en su Sistema o si tiene una aplicación de correo electrónico compatible con VIM y ha instalado y configurado Mapivim.dll. Para obtener información acerca de cómo instalar y configurar el soporte Microsoft Access para las aplicaciones VIM, consulte el Kit de recursos de Microsoft Office 97.

El objeto enviado en el mensaje se encuentra en el formato de salida especificado. Al hacer doble clic en el objeto, la aplicación correspondiente se iniciará cargándolo. Las aplicaciones asociadas a cada formato de salida son las siguientes: Microsoft Excel (*para los archivos .xls*), Microsoft Windows Notepad (*para los archivos .txt*) y Microsoft Word (*para los archivos .rtf*). Para los archivos .html, se inicia el explorador Internet predeterminado (*tal como Microsoft Internet Explorer*).

Macros y módulos

Las bases de datos de Access también tienen macros y módulos que ayudan a producir el lado de acción de una base de datos, proporcionan un procedimiento de actuar y de utilizar tablas, consultas, formularios e informes existentes dentro de su base de datos.

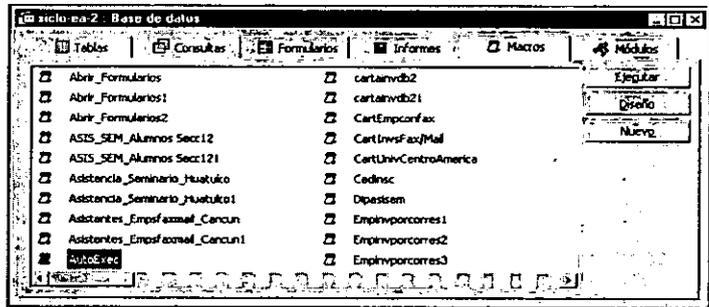
Las macros son simplemente un conjunto de acciones en el cual cada acción ejecuta una tarea específica. Como ejemplo usted puede definir una macro para abrir un formulario o informe específico en la ventana Vista preliminar, las macros le resultarán útiles para automatizar el uso de Access 97.

Cada una de las tareas individuales que puede ejecutar una macro se le conoce como acción, y Access 97 le permite escoger y llevar a cabo docenas de acciones distintas.

Cuando ejecute una macro, Access 97 utiliza los objetos y datos especificados en las acciones de la macro, para llevarlas a cabo en la secuencia como aparecen enlistadas.

La figura A.22 muestra la lista de macros de la base de datos SICLO-EA. Como puede ver a diferencia de las fichas vistas anteriormente en esta opción en lugar de existir el botón Abrir, para una macro existe el botón Ejecutar el cual correrá la macro y ejecutará las acciones para la macro seleccionada.

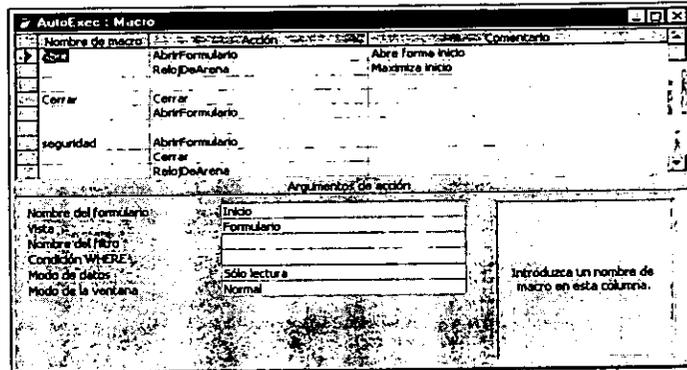
Figura A.22
Ventana Base de datos que muestra la ficha Macros de la base de datos SICLO-EA.



Ventana Macro

La ventana Macro es un lugar de trabajo de diseño gráfico que se utiliza para la creación de las macros, muy similar a la ventana Diseño de otros objetos de Access. Para abrir la ventana Diseño de macros seleccione una macro de la lista en la ventana Base de datos y haga clic en el botón Diseño. La figura A.23 muestra el ejemplo de una macro en su vista Diseño.

Figura A.23
Ejemplo de la vista
Diseño de una
Macro.



Note que la ventana Diseño está dividida en cuatro partes. El menú y la barra de herramientas se localizan en la parte superior de la pantalla, de manera semejante a los demás objetos de Access. La sección superior de la ventana Macro es donde se diseña esta. La sección tiene dos columnas principales, Acción y Comentarios. Usted utilizará la columna Acción para agregar las acciones de cada paso de la macro y utilizará la columna de Comentarios para agregar una descripción de cada una de las acciones, estos datos de la columna comentarios Access los ignorará aparecen ahí solo para su propia referencia.

La sección inferior de la ventana Macro define los argumentos de cada acción. La mayor parte de las acciones necesitan más información acerca de los objetos que especifique antes de que puedan ejecutarse. Esta sección enlista los argumentos correspondientes a la acción.

Existen varias formas de agregar una acción. Puede escribir el nombre de la acción directamente en la columna Acción. También puede arrastrar y soltar una acción de la ventana Base de datos en la columna Acción. Puede agregar más de una acción a una macro y Access ejecutará las acciones en el orden en el cual aparezcan enlistadas, agregar una descripción no es obligatorio.

Puede utilizar macros para efectuar una amplia variedad de tareas, como:

Ⓞ Abrir una tabla, formulario, consulta o informe en vista Diseño o en vista hoja de datos.

Ⓞ Abrir varios formularios e informes juntos.

- Ⓞ Cerrar tablas, formularios e informes abiertos.
- Ⓞ Utilizar formularios de validación de datos para asegurar la precisión de estos.
- Ⓞ Abrir y dar una vista previa o imprimir informes.
- Ⓞ Ejecutar consultas de acción.
- Ⓞ Mover datos entre diferentes tablas.
- Ⓞ Ejecutar cualquier comando de la barra de menús de Access.
- Ⓞ Definir valores de controles en formularios e informes.
- Ⓞ Llevar a cabo ciertas acciones cuando se haga clic en determinado botón de comando o se oprima una tecla.
- Ⓞ Mover, minimizar, maximizar, cambiar de tamaño o restaurar cualquier ventana.
- Ⓞ Desplegar información, como por ejemplo, una advertencia para el usuario.
- Ⓞ Cambiar de nombre a objetos de base de datos.
- Ⓞ Iniciar otras aplicaciones.

Como ejecutar sus macros

Una vez creada una macro, algunos lugares distintos en Access le permiten ejecutar la macro. Puede ejecutarla desde la ventana Diseño de macro haciendo clic en el botón Ejecutar de la barra de herramientas, el cual tiene un icono de signo de exclamación. También puede escoger Ejecutar del menú Diseño. Otra forma es desde la ventana Base de datos. Seleccione la macro de la lista de macros bajo la ficha Macro de la ventana Base de datos, y haga clic en el botón Ejecutar. También puede hacerlo con un doble clic en la macro que desea ejecutar.

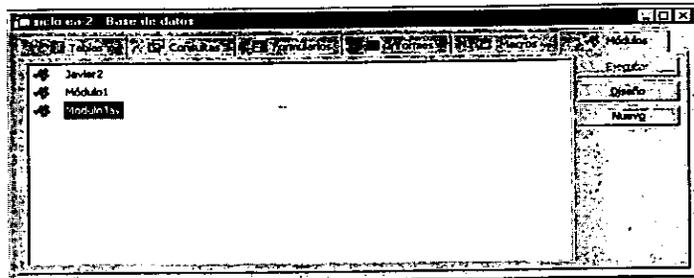
Existe en Access una macro autoejecutable muy común esta es AutoExec, que se ejecuta automáticamente cada vez que usted abra su base de datos. Para crear esta macro simplemente cree la macro que desea se ejecute cada vez que se abra la base de datos. Guarde la macro y llámela AutoExec y siempre que abra su base de datos se ejecutará la macro AutoExec, lo que resulta útil, si desea desplegar una pantalla de apertura para sus usuarios, o desea abrir un menú principal de Panel de control.

Módulos

Los módulos son conjuntos de procedimientos de Visual Basic for Applications (VBA). Access 97 incluye un lenguaje de programación incorporado, esencialmente idéntico a la herramienta de elaboración de aplicaciones tan popular de Microsoft, Visual Basic. VBA en Access 97 le permite crear sus propias funciones y procedimientos personalizados. También puede controlar de manera programática el mecanismo inferior de la base de datos de Access 97.

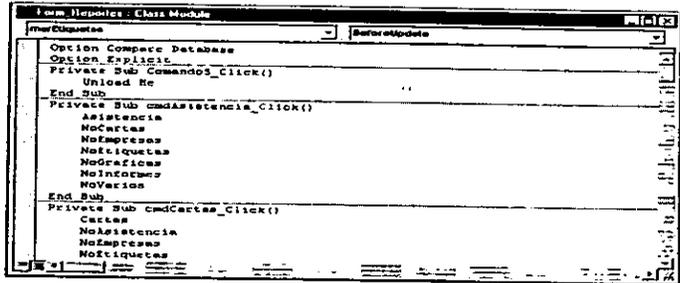
Para diseñar un nuevo módulo usted debe abrir primeramente la ventana Base de datos y moverse hacia la ficha Módulos que se encuentra al final después de la ficha Macros vista anteriormente, al igual que en la ficha anterior aparece en la derecha tres opciones que son Ejecutar, Diseño y Nuevo, véase la figura A.24.

Figura A.24
Ficha Módulos en la ventana Base de datos.



Una vez abierta la ventana Base de datos situada en la ficha módulos, usted podrá seleccionar la opción Diseño si es que existe algún módulo creado o la opción de Nuevo si quiere crear un nuevo módulo la pantalla o vista que aparecerá al seleccionar alguna de las dos opciones es como la mostrada en la figura A.25

Figura A.25
Ejemplo de un
Módulo en su vista
Diseño de la base de
datos SICLO-EA.



Para comprender mejor acerca de cómo crear los módulos veamos dos conceptos que nos ayudarán a crear un módulo, estos son procedimiento y evento.

Procedimiento

Un procedimiento es una unidad de código de Visual Basic para aplicaciones. Un procedimiento contiene una serie de instrucciones y métodos que realizan una operación o calculan un valor. Por ejemplo, el siguiente procedimiento de evento usa el método OpenForm para abrir el formulario Pedidos:

```
Private Sub AbrirPedidos_Click()
```

```
DoCmd.OpenForm "Pedidos"
```

```
End Sub
```

Hay dos tipos de procedimientos:

- Los procedimientos Sub realizan una operación o una serie de operaciones pero no devuelven un valor. Puede crear sus propios procedimientos Sub o usar las plantillas de procedimientos de evento que Microsoft Access crea por usted.

Cada formulario o informe de la base de datos tiene un módulo de formulario o un módulo de informe incorporado que contiene plantillas de procedimientos de evento. Puede agregar código que se ejecute en respuesta a los eventos que se produzcan en el formulario, en el informe o en los controles del formulario o informe. Cuando Microsoft Access reconoce que se ha producido un evento en un formulario, informe o control, ejecuta de forma automática el procedimiento de evento nombrado para el objeto o evento. Por ejemplo, puede escribir un procedimiento de evento que establece el enfoque a un cuadro de texto Nombre esposo cuando el usuario selecciona la casilla de verificación Casado.

Muchos de los asistentes (*por ejemplo, el Asistente para botones de comando*) que crean objetos crean también procedimientos de evento para el objeto.

Los procedimientos Function (*llamados a menudo simplemente funciones*) devuelven un valor, tal como el resultado de un cálculo. Visual Basic incluye muchas funciones incorporadas; por ejemplo, la función Now devuelve la fecha y hora actual. Además de estas funciones incorporadas, puede crear sus propias funciones personalizadas.

Puesto que las funciones devuelven valores, puede usarlas en expresiones. Puede usar funciones en expresiones en muchos lugares de Microsoft Access, incluyendo en una instrucción o método de Visual Basic, en muchos valores de propiedades o en una expresión de criterios en un filtro o consulta.

Éste es un ejemplo de un procedimiento Function, PrimerDíaDelMesSiguiente, que devuelve la fecha del primer día del mes siguiente a la fecha actual:

```
Function PrimerDíaDelMesSiguiente()
```

```
PrimerDíaDelMesSiguiente = _
```

```
    DateSerial(Year(Now), Month(Now) + 1, 1)
```

```
End Function
```

Esta función personalizada consiste en una única instrucción de asignación que asigna el resultado de una expresión (*en el lado derecho del signo igual [=]*) al nombre de la

función, *PrimerDíaDelMesSiguiete* (en el lado izquierdo del signo igual). La función calcula el resultado usando las funciones incorporadas de Visual Basic DateSerial, Year, Now y Month.

Una vez que haya creado esta función, puede usarla en una expresión prácticamente en cualquier parte de Microsoft Access. Por ejemplo, podría especificar que un cuadro de texto muestre el primer día del mes que sigue a la fecha actual como el valor predeterminado estableciendo la propiedad ValorPredeterminado. (*DefaultValue*) del control del cuadro de texto a la siguiente expresión en la hoja de propiedades:

=PrimerDíaDelMesSiguiete()

Para usar una función como valor de una propiedad, la función debe estar en el módulo del formulario o del informe, o en un módulo estándar. No se puede usar una función en un módulo de clase que no esté asociada con un formulario o informe como valor de propiedad de formulario o informe.

Eventos

Un evento es una acción determinada que se produce en, o con, un objeto particular. Microsoft Access puede responder a una variedad de eventos: clics del mouse, cambios en los datos, formularios que se abren o se cierran, y muchos otros. Los eventos son normalmente el resultado de una acción del usuario.

Usando un procedimiento de evento o una macro, se pueden agregar respuestas personalizadas a un evento que se produzca en un formulario, informe, o control.

Suponga que desea que un formulario Detalles de producto se abra cuando haga clic en un botón de comando Detalles de producto en un formulario Pedidos.

Responder al evento Al hacer clic usando un procedimiento de evento

Cuando se crea un procedimiento de evento para un objeto, Microsoft Access agrega una plantilla de procedimiento de evento nombrada para el evento y el objeto al módulo de

formulario o módulo de informe. Todo lo que necesita hacer es agregar código que responda de la forma que desee cuando se produzca el evento en el formulario o informe.

Tipos de módulos

Hay dos tipos básicos de módulos: los módulos de clases y los módulos estándar. Cada procedimiento en un módulo puede ser un procedimiento Function o un procedimiento Sub.

Módulos de clases

Los módulos de formularios y los módulos de informes son módulos de clases que están asociados con un formulario o informe determinado. Los módulos de formularios y de informes contienen a menudo procedimientos de evento que se ejecutan en respuesta a un evento en un formulario o informe. Puede usar los procedimientos de evento para controlar el comportamiento de los formularios e informes, y sus respuestas a las acciones de los usuarios, tal como hacer clic con el mouse en un botón de comando.

Cuando se crea el primer procedimiento de evento para un formulario o informe, Microsoft Access crea automáticamente un módulo de formulario o un módulo de informe. Para ver el módulo para un formulario o informe, haga clic en Código en la barra de herramientas en la vista Diseño de informe o de formulario.

Los procedimientos en los módulos de formularios y de informes pueden llamar a procedimientos que haya agregado a los módulos estándar.

En Microsoft Access 95, los módulos de clase sólo existen en asociación con un formulario o informe. En Microsoft Access 97, los módulos de clase pueden existir también con independencia de un formulario o informe, y este tipo de módulo de clase se enumera en la ficha Módulos de la ventana Base de datos. Puede usar un módulo de clase en la ficha Módulos para crear una definición para un objeto personalizado.

Módulos estándar

Los módulos estándar contienen procedimientos generales que no están asociados con ningún otro objeto y procedimientos usados con frecuencia que pueden ser ejecutados desde cualquier parte de la base de datos.

La lista de módulos estándar de la base de datos se puede ver haciendo clic en la ficha Módulos de la ventana Base de datos. Los formularios, informes y módulos estándar se enumeran también en el Examinador de objetos.

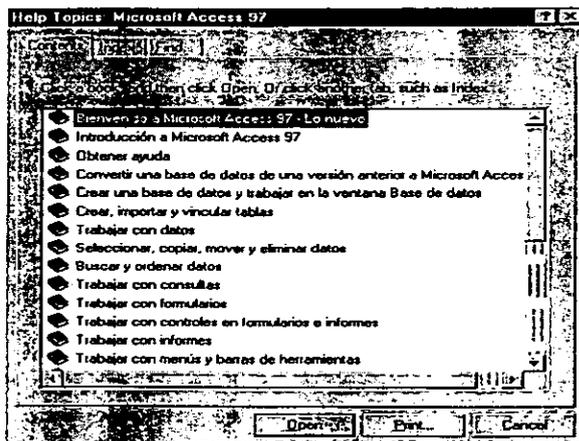
Ayuda de Access 97

Access 97 cuenta con un archivo de ayuda en pantalla que nos proporciona diferentes temas de ayuda con fichas Contenido, Índice y Buscar. Con estos temas de ayuda usted podrá ampliar mas sus conocimientos acerca de los diferentes tópicos que aquí se trataron.

Para abrir el archivo de ayuda de Microsoft Access 97 haga clic sobre el signo que se encuentra a un lado de la opción Ventana en la barra principal de Access, la Ayuda esta distinguida como un signo de interrogación (?).

La figura A.26 muestra la opción Temas de ayuda de Microsoft Access 97.

Figura A.26
Ventana de Ayuda de
Microsoft Access 97.



Lo nuevo de Access 97

A diferencia de las versiones anteriores de Access, Access 97 cuenta con diversas características que lo hacen más interesante y poderoso.

Microsoft Access proporciona muchas características nuevas diseñadas para ayudar a utilizar fácilmente Internet y programar aplicaciones para el World Wide Web.

Necesita un explorador de Web, tal como Microsoft Internet Explorer y un módem, una conexión de intranet u otra conexión de red para tener acceso a Internet y aprovechar algunas de estas nuevas características:

Importar, Exportar Vincular

Características	Se utilizan para
Importar o vincular archivos HTML	Importar o vincular tablas (<i>de sólo lectura</i>) o listas desde un archivo HTML utilizando el comando Obtener datos externos del menú Archivo. El Asistente para importación de HTML y el Asistente para vinculación de HTML también convierten los vínculos HTML a un campo con un tipo de datos Hipervínculo.
Exportar objetos a formato HTML	Exportar informes a formato HTML estático, y hojas de datos y formularios a formato HTML estático o dinámico utilizando el comando Guardar como o exportar del menú Archivo. Mejore la apariencia, la coherencia y el desplazamiento por sus páginas Web mediante un archivo de plantilla de HTML.
Asistente para publicar en el Web	Crear una aplicación de Web con el Asistente para publicar en el Web mediante el comando Guardar como HTML del menú Archivo. Puede enviar hojas de datos, formularios o informes a formato HTML estático o dinámico utilizando uno o varios archivos de plantilla de HTML; crear una página principal; almacenar todos los archivos en una carpeta especificada como una publicación de Web, copiar los archivos a un servidor Web utilizando el Asistente para publicación en el Web y guarde un perfil de publicación de Web para utilizar posteriormente.
Acceso a servidores FTP y HTTP	Importar o vincular datos (<i>de sólo lectura</i>) en servidores FTP o HTTP, exportar un objeto de base de datos a un servidor FTP y agregar o modificar una lista de ubicaciones de servidores FTP mediante el cuadro Buscar en el cuadro de diálogo Abrir.

Hipervínculos

Características	Se utilizan para
Tipo de datos Hipervínculo	Almacenar un hipervínculo en una tabla. Agregar la dirección del hipervínculo en una hoja de datos o formulario utilizando el botón Insertar hipervínculo. Un hipervínculo puede ser una ruta de acceso a un archivo en la unidad de disco duro, una ruta de acceso UNC o una dirección URL. Cuando hace clic en un campo de hipervínculo, Microsoft Access salta a un objeto, un documento, una página Web u otro destino.
Destinos de hipervínculos	Saltar a una base de datos u objeto de Microsoft Access desde un hipervínculo en una hoja de datos, formulario o informe, o desde dentro de otros documentos de Microsoft Office, tales como un documento de Microsoft Word o una hoja de cálculo de Microsoft Excel. Saltar a un documento de Office desde un hipervínculo dentro de una hoja de datos, formulario o informe.
Hipervínculos en formularios e informes.	Crear una etiqueta, control de imagen o botón de comando en un formulario o informe que salta a un destino de hipervínculo.

Controles ActiveX

Características	Se utilizan para
Control Explorador de Web de Microsoft	Explorar World Wide Web, ver páginas Web, tener acceso a otros documentos en su PC, en la red o en Internet y transferir datos desde Internet a través de un formulario de la aplicación.
Controles ActiveX adicionales	Programar aplicaciones Web brillantes y dinámicas. Disponible en Microsoft Office 97, edición Developer.

Herramientas de Office

Características	Se utilizan para
Barra de herramientas Web.	Examinar fácilmente bases de datos y objetos de Microsoft Access o documentos de Office utilizando una funcionalidad similar a la de la barra de herramientas de Microsoft Internet Explorer.
Búsqueda rápida en el Web.	Ubicar bases de datos de Microsoft Access y otros documentos de Microsoft Office que están indexados mediante las propiedades de base de datos y archivo.

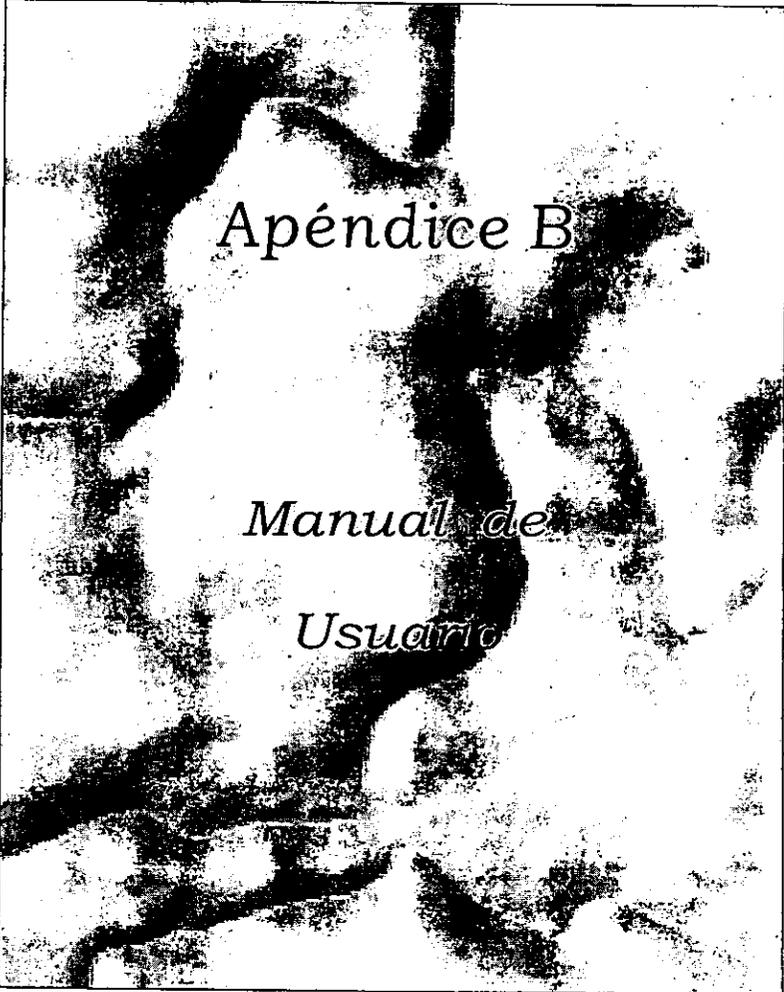
Otras características

Características	Se utilizan para
Réplica	Sincronizar la réplica de una base de datos con una réplica o Diseño principal en un servidor de Internet.
Aplicaciones de ejemplo	Ver ejemplos de características de Internet en la base de datos de ejemplo Neptuno y en la aplicación de ejemplo Soluciones para el programador.

Lo nuevo en seguridad y administración de la base de datos.

Características	Se utilizan para
Nuevo en 97 Archivo MDE	Eliminar el código de Visual Basic que se puede modificar y proteger el diseño de los formularios, informes y módulos al guardar la base de datos como un archivo MDE.
Nuevo en 95 Propiedad Hidden	Ocultar los objetos de la base de datos que no desea que los otros usuarios vean o abran en la ventana Base de datos.
Nuevo en 95 Contraseña para proteger la base de datos	Asignar una contraseña única para controlar quién puede abrir una base de datos en lugar de, o además de, implementar seguridad a nivel de usuario.

<p>Nuevo en 95 Asistente para seguridad a nivel de usuario</p>	<p>Puede proteger la base de datos más fácilmente usando este asistente.</p>
<p>Nuevo en 95 Opción de la línea de comandos <code>/wrkgrp</code> del archivo de información de grupo de trabajo</p>	<p>Cambiar a un archivo de información de grupo de trabajo distinto cuando inicia Microsoft Access.</p>



Apéndice B

Manual de

Usuario

MANUAL DE USUARIO

INSTALACION DE SICLO-EA

¿Necesita saber algo especial para trabajar con SICLO-EA?

No realmente. Las únicas cosas con las que debería familiarizarse son los conceptos básicos de Windows, como por ejemplo:

Como elegir el menú, lanzar un programa, guardar un archivo y así sucesivamente.

Si no ha trabajado con Windows 95 o posterior antes, por favor, consulte su manual de Windows antes de comenzar con SICLO-EA.

Lo que necesita para ejecutar SICLO-EA para Windows.

HARDWARE

SICLO-EA puede trabajar en cualquier computadora que ejecute Windows 95 o posterior. La computadora debe tener al menos 640 Kb de RAM interna y 16Mb de memoria RAM extendida, si se cuenta con mas memoria SICLO-EA trabajara mejor. Es recomendable contar con un Mouse para un mejor manejo del sistema.

SOFTWARE

Para que pueda trabajar SICLO-EA es indispensable que se tenga instalado OFFICE 97, en particular ACCESS 97.

Espacio en disco duro: SICLO-EA usa al menos necesario 20 Mb de espacio libre en disco cuando se instale en su sistema.

Presentación de SICLO-EA

Características de SICLO-EA

Función	Descripción
Administrador de Seguridad	SICLO-EA cuenta con un control restringido para proteger la información. Solicitando clave de usuario y password como primer nivel de seguridad, posteriormente solicita la clave que da acceso al ambiente de trabajo de SICLO-EA.
Botones	
Expediente	Este nos permite desplegar la pantalla que en la cual se captura la información referente a las personas que tienen interés a asistir al evento a realizarse.
Accesos	Mediante esta opción se muestra la ventana que permite el acceso a herramientas de Office 97 como son Word, Excel, Power Point, además del Bloc de Notas y en caso dado a Internet.
Catálogos	A través de este botón se muestra la serie de catálogos con que cuenta el sistema. Catálogos de los cuales se alimenta SICLO-EA por eso es importante el llenado de estos antes de hacer uso directo de la funcionalidad real de SICLO-EA.
Informes	Esta opción de la pantalla principal proporciona informes

	<p>detallados de la información que se ha capturado en SICLO-EA. Por ejemplo:</p> <ul style="list-style-type: none"> ✓ Cartas invitación ✓ Cédula de Inscripción ✓ Estadísticas ✓ Asistencia ✓ Etc.
Búsqueda	Finalmente este botón permite hacer la búsqueda de un asistente por letra inicial del apellido paterno o todos a la vez, mostrando los datos personales mínimos.

Uso de SICLO-EA

Para poder tener acceso a SICLO-EA es necesario que tenga abierto Access, una vez que ya este abierto, este le pregunta que si quiere abrir una base de datos que ya existe, como se muestra en la **Figura 1**. Si esta la base seleccione la base y de aceptar.

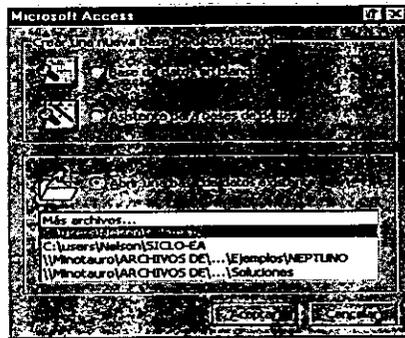


Figura 1. Ventana para seleccionar base de datos

Si la base de datos no esta en esa pantalla, lo que tiene que hacer, es una búsqueda para encontrar la ruta que permite acceder a SICLO-EA, esto lo puede hacer mediante el icono estándar para abrir de Windows el cual le mostrará la pantalla como se muestra en la

Figura 2. en la cual usted puede buscar la ruta y de esa forma abrir SICLO-EA. Dicho sistema ya debe estar instalado en su computadora para que usted lo pueda abrir.

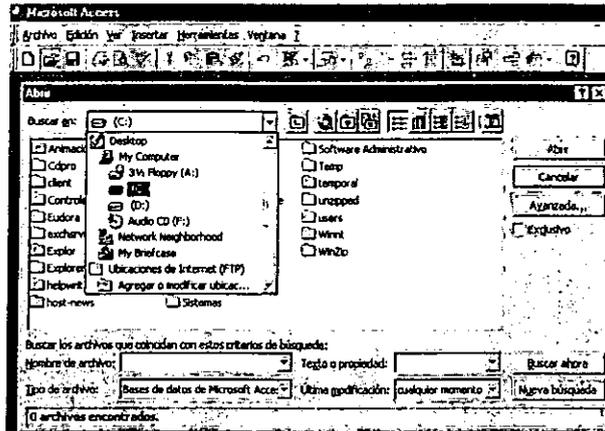


Figura 2. Ventana para abrir base de datos

Una vez que ya encontró la ruta para abrir SICLO-EA se encontrará con el primer nivel de seguridad como se muestra en la Figura 3. el cual le pide la contraseña del sistema.

La contraseña se la proporcionara la persona que instale SICLO-EA en su computadora si la persona no conoce la contraseña esta se la puede proporcionar el administrador del sistema. Dicha contraseña solo deben saber las personas que tengan que ver con SICLO-EA, cuando teclee la contraseña esta se convierte en asteriscos para evitar que personas ajenas al sistema puedan hacer mal uso de la información con la que cuenta SICLO-EA.

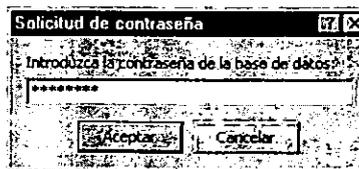


Figura 3. Ventana primer nivel de seguridad

Una vez que ya tecleó la contraseña correcta, SICLO-EA le muestra la pantalla de presentación, dicha ventana se muestra en la Figura 4. Esta sirve para que el sistema sea amigable, en esta pantalla usted no tiene que hacer nada, solo esperar unos segundos para que esta pantalla se cierre y entonces si puede empezar a utilizar SICLO-EA.

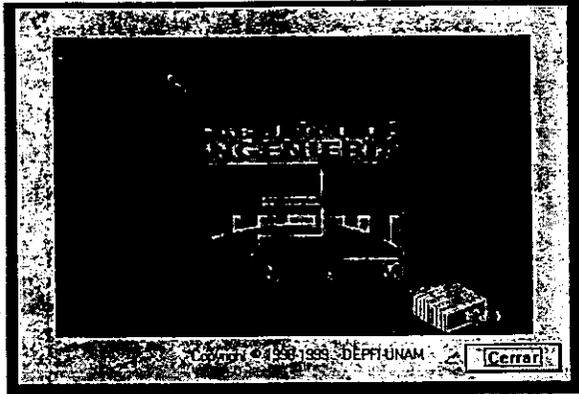


Figura 4. *Ventana de presentación*

Cuando la pantalla de la Figura 4. Ya se cerro aparece la pantalla del segundo nivel de seguridad la cual se muestra en la Figura 5. En esta pantalla lo que se hace es dar una segunda contraseña pero ahora como usuario, dicha contraseña permite al sistema saber que tipo de usuario es el que esta manejando el sistema, o lo que es lo mismo el sistema sabe por medio del password y el login SICLO-EA sabe si un usuario común o es un super usuario, todo esto es con el fin de que la información del sistema no sea modificada por cualquier persona, es decir que no haya un mal uso o manejo de la información.

El Password y el Login le serán proporcionados por el administrador del sistema, el login usted lo puede sugerir al administrador para que sea algo que usted recuerde fácilmente este puede contener números y letras que es lo más conveniente para que no sea tan fácil de decodificar o que alguna persona lo pudiera saber. El password le será dado por el

administrador y también usted lo puede sugerir dicho campo puede combinar números y letras que es lo mejor para que no pueda decodificar tan fácil.

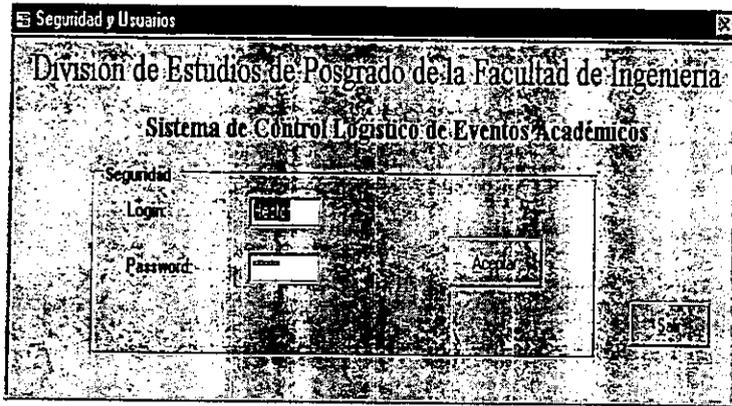


Figura 5. Ventana segundo nivel de seguridad

Después de haber pasado el segundo nivel de seguridad ahora el sistema nos muestra la pantalla inicio de SICLO-EA (Figura 6.).

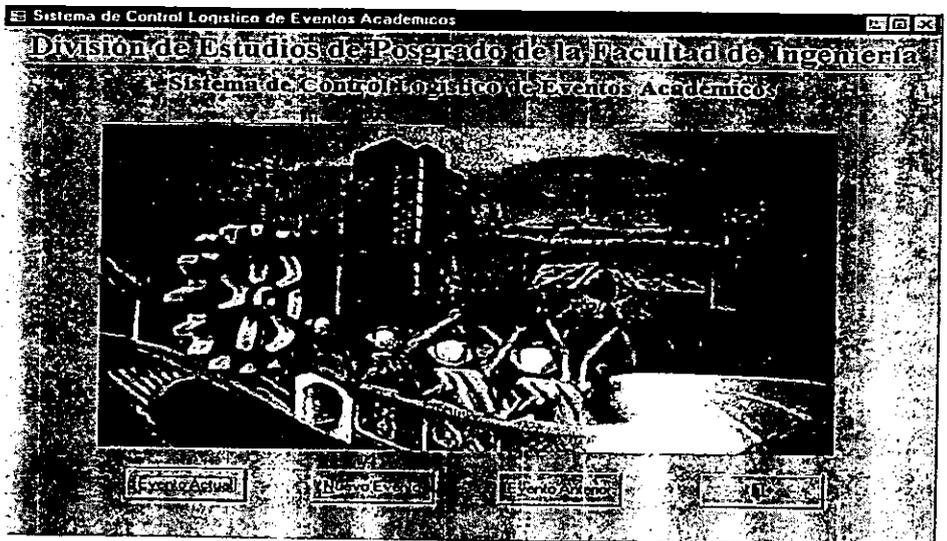


Figura 6. Ventana Inicio del Sistema

Cuando abre la ventana de inicio del sistema, dicha pantalla tiene tres botones, evento actual, evento nuevo, evento anterior y el botón estándar de salir.

Nombre Botón	Descripción
Evento Actual	Al presionar este botón, se abre la pantalla de principal de SICLO-EA, con el evento que esta activo, ese es el evento que se vera reflejado en el sistema.
Evento Nuevo	Al presionar este botón, se abre la pantalla de parámetros, en dicha ventana usted puede dar de alta un nuevo evento, tiene que seleccionar el evento o ponerlo activo para que este se refleje en el sistema.
Evento Anterior	Al presionar este botón se abre la pantalla de parámetros y usted puede seleccionar el evento anterior, para seleccionar el evento hay que seleccionar el evento, al seleccionar un evento hay que quitar la paloma del evento que estaba seleccionado, por que si no hace eso puede tener problemas.
Salir	Este es el botón estándar de salir en todo el sistema.

Al presionar el botón de Evento Actual o Evento Nuevo SICLO-EA, le muestra la pantalla de parámetros (*Figura 7.*) en dicha ventana usted puede dar de alta un nuevo evento o moverse en los eventos que usted ya dio de alta.

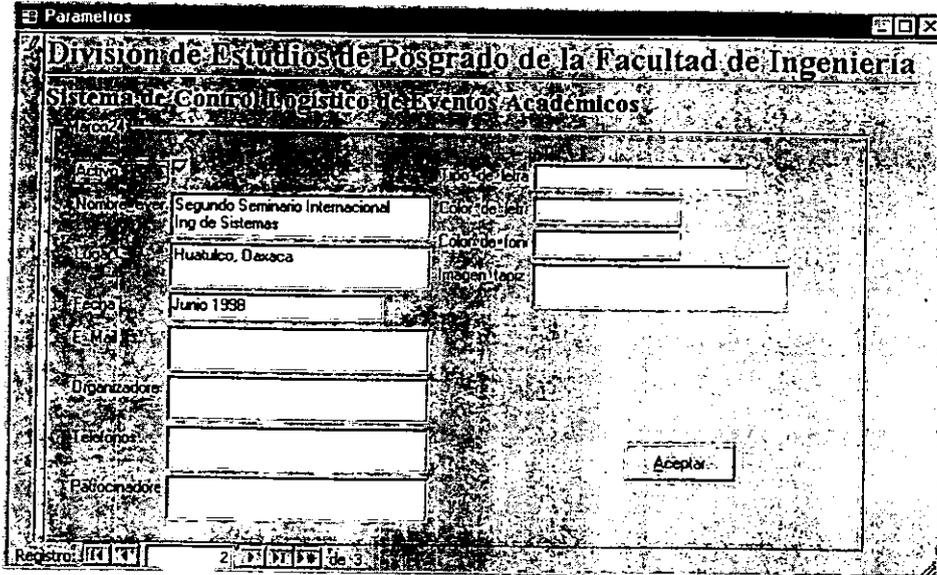


Figura 7. Ventana de Parámetros

En la ventana de parámetros, el evento que quiere que se ve en el sistema, debe seleccionarlo en la casilla de activo y no debe haber mas de un evento seleccionado por que ello le podría causar algún conflicto, después de seleccionar o dar de alta el evento usted debe presionar el botón de aceptar y con ello usted puede ahora sí empezar a trabajar con SICLO-EA.

Después de presionar el botón de Aceptar el sistema nos muestra la pantalla principal de SICLO-EA (Figura 8.) en dicha pantalla se encuentran los módulos principales del sistema, estos son:

- EXPEDIENTES
- ACCESOS
- CATALOGOS

- INFORMES
- BUSQUEDA



Figura 8. Ventana principal de SICLO-EA

Para ejecutar el módulo de expedientes bastará con dar un clic con el mouse sobre el botón con la leyenda Expedientes. Al hacer esto se despliega la pantalla de la Figura 9.

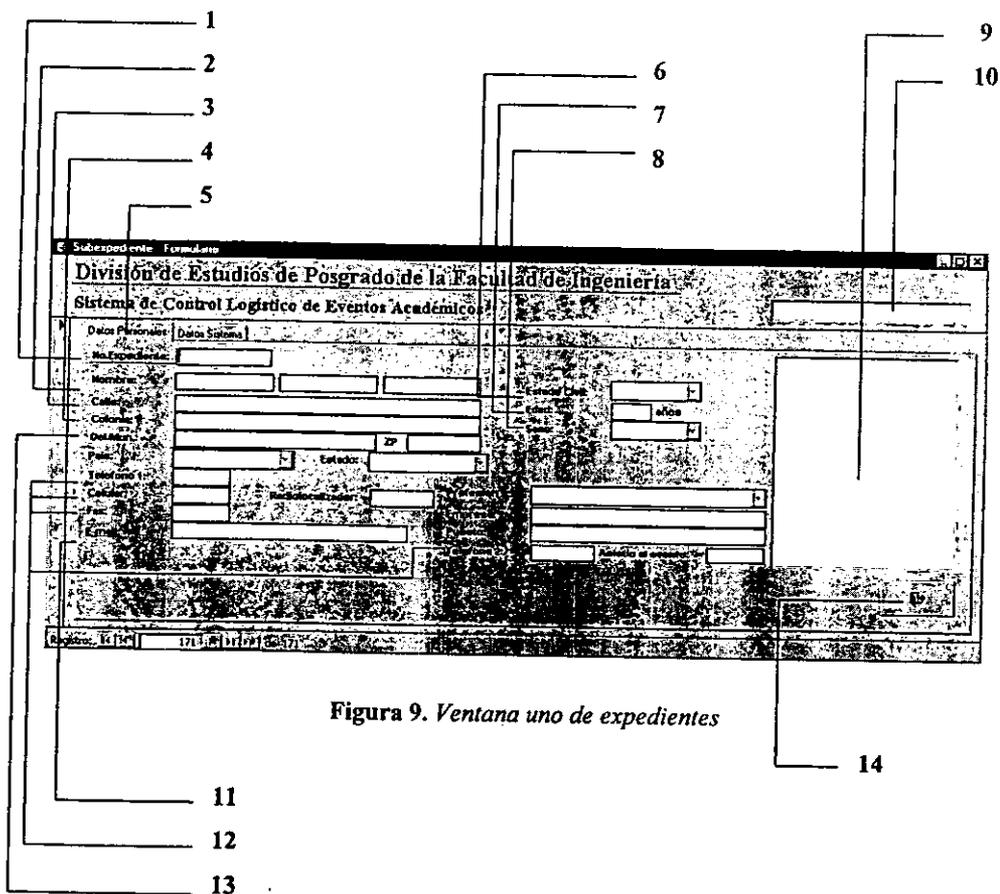


Figura 9. Ventana uno de expedientes

Opción	Nombre del Campo	Descripción
1	No. Expediente	Este campo se genera automáticamente al dar de alta un nuevo registro, la magnitud de este campo es de tres caracteres alfanuméricos
2	Nombre	En este campo nombre se llena el nombre con 30

		caracteres alfanuméricos como máximo, en el segundo campo se llena con el apellido paterno con un máximo de 30 caracteres alfanuméricos y el tercer campo que es el apellido materno que al igual que los anteriores tiene un máximo de 30 caracteres alfanuméricos.
3	Calle	El campo de calle acepta como máximo 150 caracteres alfanuméricos, este se debe llenar con el nombre de la calle y el número de la casa, departamento etc.
4	Colonia	Este campo se debe llenar con la colonia en donde vive la persona la cual se esta registrando, este campo acepta un máximo de 60 caracteres alfanuméricos.
5	Datos Personales	Esta pestaña nos permite ver la ventana de datos personales ya que este formulario tiene dos ventanas la de datos personales y la de datos del sistema.
6	Estado Civil	Este campo se debe llenar, seleccionando alguna de las opciones que nos muestra el sistema, si no esta la que necesita, entonces, tiene que darla de alta en el formulario de catálogos, en el catálogo de estado civil, este campo acepta un máximo de 50 caracteres alfanuméricos.
7	Edad	Este campo se debe llenar con la edad de la persona y acepta como máximo dos caracteres.
8	Sexo	Este campo se llena con las opciones que nos da el sistema, si no esta definido el campo que necesita, tiene que definirlo en el catálogo de sexo este campo acepta un máximo de 20 caracteres alfanuméricos.
9	Nombre Persona	Este campo despliega el nombre completo de la persona que este consultando, este campo nos sirve

		<p>en la pantalla de Datos del Sistema ya que en esta ventana no tenemos los datos de la persona, por eso dicho campo es útil para usted ya que despliega el nombre del registro que este utilizando, usted no tiene que hacer nada para que lo pueda ver este siempre esta únicamente cambia el nombre cuando usted cambia de registro.</p>
10	Foto	<p>Para capturar la foto de la persona, usted debe pedirle a las personas que están inscritas a su evento que le manden una fotografía reciente, pero esta tiene que estar escaneada en un formato BMP entonces usted con una cuenta de super usuario tiene que dar de alta en la tabla de Expediente, lo que tiene que hacer es insertar la fotografía en el campo de foto en la tabla expediente y con esto la fotografía de la persona se despliega cada vez que usted selecciona el registro de dicha persona.</p>
11	E-mail	<p>Con esta opción usted tiene la posibilidad de mandar correo sin salir de SICLO-EA lo único que tiene que hacer es ponerse con el mouse encima del letrero azul que dice E-mail, al hacer clic el sistema toma la dirección del registro que usted este viendo y con ello le puede mandar un mail a dicha persona, las posibilidades para que funcione bien son que tenga algún software de correo pero este tiene que ser de Microsoft, si usted tiene algún otro software que no sea de Microsoft el sistema le marcara un error, que no puede mandar el correo o que no reconoce o no tiene el software para mandar correo por lo que se recomienda tener un software para mandar correo pero de Microsoft, si no lo tiene puede tener</p>

		problemas de incompatibilidad y su correo no llegue o definitivamente no lo pueda mandar.
12	Teléfono 1 Teléfono 2 Celular Fax	Todos estos campos son para tener algún teléfono en donde poder localizar a la persona, estos campos aceptan como máximo 20 caracteres alfanuméricos se definió así para poder tener la clave del estado o el país o tal vez la extensión si es que la persona esta en una empresa y tiene una extensión.
13	Del/Mun.	En este campo se debe capturar la Delegación o el Municipio según sea el caso, este campo acepta como máximo 40 caracteres de tipo alfanumérico.
14	Botón de Salir	Este es el botón estándar de salir de SICLO-EA que por lo general se encuentra en la parte inferior derecha de todas las pantallas.
	País	Este campo se llena con las opciones que nos proporciona el sistema con 207 países capturados, si el país no esta definido, lo puede dar de alta en el catálogo de países.
	Estado	En este campo se dieron de alta los 31 estados de México y el Distrito Federal, que nos sirve para las personas que son de México, para los extranjeros, hay que dar de alta el estado en el catálogo de estados.
	ZP	En este campo hay que llenarlo con el código postal, tiene como máximo 15 caracteres.
	Profesión	Este campo se debe llenar con las opciones que nos da el sistema, si la Profesión que necesitamos no esta en dichas opciones, tenemos que darla de alta en el catálogo de Profesiones.
	Empresa	Se llena con la empresa para la cual trabaja la persona que asiste a su evento o la empresa que

		representa, se debe llenar con un máximo de 80 caracteres alfanuméricos.
	Puesto	En este campo se pone el puesto que actualmente ocupa la persona en su empresa o institución con un máximo de 90 caracteres alfanuméricos.

En esta pantalla de expedientes tiene dos ventanas, en la primera se deben capturar todos los datos personales de las personas que asisten al evento, los datos son: Nombre, Apellido Paterno, Apellido Materno, Dirección, Colonia, Delegación, Municipio, País, Estado, Profesión, Calle, Teléfono particular, Teléfono oficina, Fax, Profesión, Puesto, Empresa, Estado civil, Edad, Sexo, e-mail, todos los campos anteriores pueden llenarse o no pero mientras más campos se llenen los informes que arroja el sistema serán más completos. En el campo de e-mail si la persona la cuál se esta consultando tiene alguna dirección electrónica, el sistema puede mandarle un correo a esa persona tomando su dirección electrónica automáticamente, lo único que tiene que hacer es dar un clic con el mouse en el campo que dice e-mail en ese campo le aparecerá un mano la cual le indica que en ese campo puede hacer alguna tarea que en este caso es abrir la aplicación que permite mandar correos electrónicos, sin tener que salirse del sistema o tener que anotar la dirección para que más tarde usted pueda mandarle un correo a esa persona, todos estos campos se llenan en la primera ventana del formulario Expedientes y en la segunda ventana se llenan campos como: Tipo persona, si la persona Presenta Ponencia, Nombre de la Ponencia, si su Ponencia es avance de tesis, Hotel donde se hospeda, la forma en que llego al evento, costo de la inscripción, si la persona tiene algún adeudo con respecto a su inscripción, la asistencia de la persona al evento y los comentarios que se tengan con respecto a la persona que se esta capturando. El botón de salir es el que tiene una puerta y una flecha y generalmente es el que esta en la parte inferior derecha de todas las ventanas, si no quiere utilizar ese botón puede utilizar el botón de cerrar estándar de Windows que es la x que esta en la parte superior derecha de todas las ventanas y con cualquiera de estas dos opciones usted regresara al menú anterior.

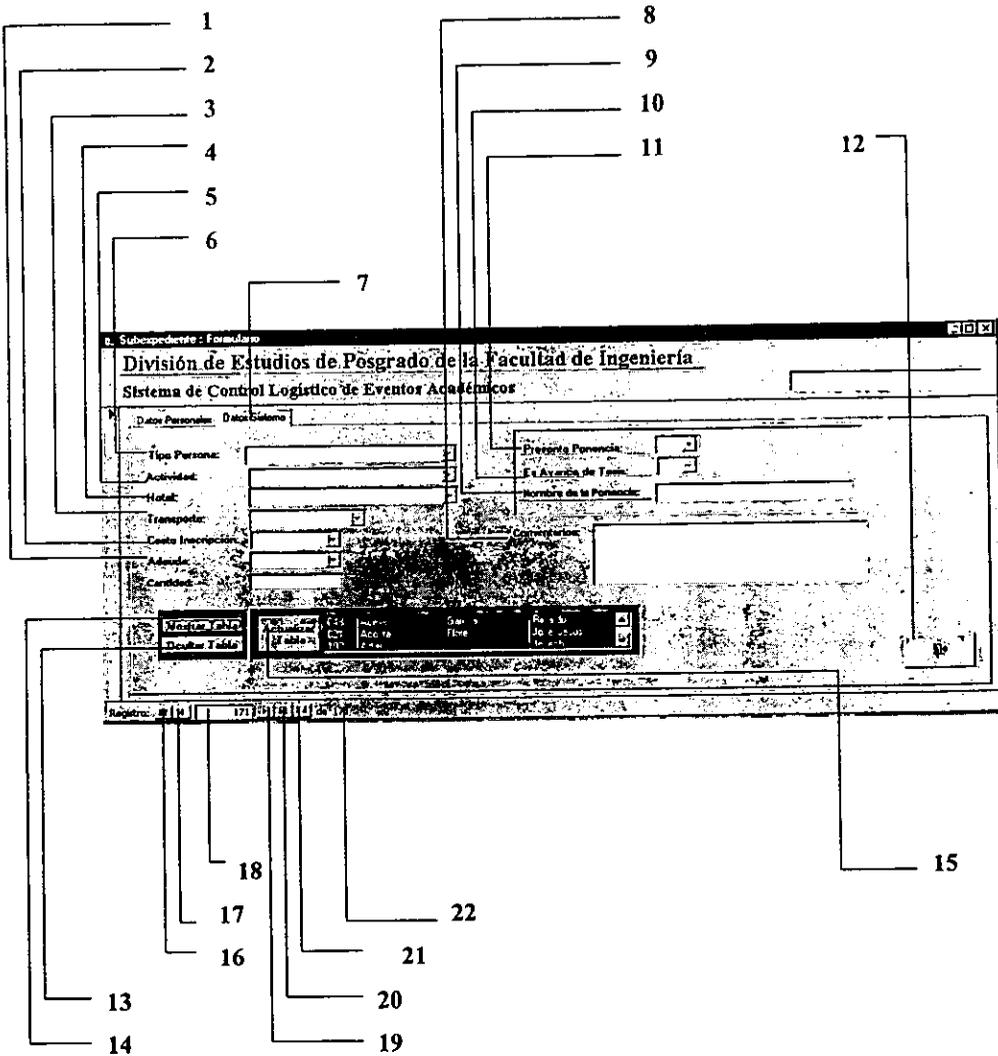


Figura 10. Ventana dos de Expedientes

Número	Nombre del Campo	Descripción
6	Tipo de Persona	Este campo nos indica que tipo de persona es la que se esta capturando, lo que nos indica es si la persona es ponente, acompañante, etc. Lo que nos va a ayudar a tener una mejor organización y control de las persona, este campo se llena con un máximo de 30 caracteres alfanuméricos.
5	Actividad	Este campo se llena con las opciones que usted lleno en el catálogo de actividades por lo que si la actividad que dicha persona va a realizar en el evento no esta, la tiene que dar de alta en el catálogo de actividades.
4	Hoteles	Este campo se debe llenar con las opciones que le da el sistema, estas opciones se llenaron previamente en el catálogo de hoteles, este campo le muestra los hoteles en donde se esta llevando su evento ya que si alguna persona que no esta hospedada en el lugar de la cede, usted pueda registrar el hotel de dicha persona.
3	Transporte	Este campo se tiene que llenar con las opciones que se dieron de alta, en el catálogo de transporte.
2	Costo de Inscripción	Este campo se llena con alguna de las opciones que se dieron de alta en el catálogo de Costo de Inscripción y estas varían, dependiendo del tipo de persona o en el periodo en el cual hizo el pago de su inscripción.
1	Adeuda	Este campo lo único que nos indica es si adeuda la persona o no adeuda nada. Este campo se llena con un SI o con un NO.
	Cantidad	Este campo nos indica el monto, que la persona adeuda, con un máximo de 20 caracteres

		alfanuméricos.
8	Comentarios	Este campo es para poner alguna observación o comentario con respecto a la persona para después considerarlo o tomarlo en cuenta, se llena con un máximo de 255 caracteres.
9	Nombre Ponencia	En este campo tiene un máximo de 100 caracteres alfanuméricos, este campo se utiliza para capturar el nombre de la ponencia, si es que la persona presentara alguna ponencia.
10	Avance de Tesis	Algunas de las ponencias pueden ser avance de tesis y este campo le puede servir para sacar alguna gráfica o estadística.
11	Presenta Ponencia	Este campo nos sirve para saber si la persona presenta o no alguna ponencia.
12	Botón Salir	Botón estándar de SICLO-EA para salir del formulario.
14	Mostrar Tabla	Con este botón nos muestra la tabla para buscar alguna persona que ya se dio de alta en el sistema, este botón existe por que no podemos ver la tabla cuando abrimos el formulario, por lo que para poder ver la tabla tiene que presionar dicho botón.
13	Ocultar Tabla	Con este botón la tabla se oculta, por si no quiere tener la tabla en el formulario.
15	Actualizar Tabla	Este botón nos sirve para actualizar la información de la tabla ya que si usted abre el formulario de expediente y luego abre la tabla en ese momento carga la información y si hace algún cambio o da de alta a una persona la tabla no se actualiza automáticamente, para eso tiene el botón de actualizar para que la tabla cargue los datos más recientes, esto también le sirve cuando el sistema

		esta en red y lo están ocupando varias personas, usted no sabe de los cambios que las otras personas le hagan a la información, con este botón usted puede ver los últimos cambios que la información haya sufrido.
16	Primer Registro	Con este botón usted se puede mover desde cualquier registro hasta el primer registro.
17	Registro Anterior	Con este botón se mueve un registro hacia atrás.
18	Registro Actual	Es el número de registro que esta leyendo o es el registro en el que la base esta.
19	Registro Siguiete	Con este botón se puede mover un registro hacia delante no importando en donde se encuentre.
20	Ultimo Registro	Con este botón se puede ir hasta el último registro no importando en que posición este.
21	Registro Nuevo	Con este botón puede dar de alta un registro nuevo no importa en que registro se encuentre, con este botón genera un nuevo registro, para poder grabar el registro lo único que tiene que hacer es moverse un registro, hacia delante o hacia atrás y con eso se graba automáticamente el registro nuevo.
22	Actualizar Tabla	Este botón sirve para actualizar los datos en la tabla ya que cuando se abre el formulario se cargan los datos de la tabla, y si se hace algún cambio o se dan de alta nuevos registros no se ven reflejados automáticamente por lo que hay que dar una actualización de datos para que se carguen los cambios y con esto se actualiza la información en el formulario.

La segunda ventana del formulario expedientes se muestra en la Figura 9. En esta ventana se puede hacer una consulta rápida de las personas que se tienen registradas en la base de datos, esta consulta le puede ser útil para verificar si una persona ya esta registrada en el sistema para que no se den de alta usuarios que ya están registrados y así evitar la redundancia de la información que le puede traer problemas a la hora que usted imprima los reportes de su evento y pensar que hay mas gente de la que realmente hay en el evento.

Para ejecutar el módulo de Accesos bastará con dar un clic con el mouse sobre el botón con la leyenda Accesos. Al hacer esto se despliega la pantalla de la Figura 11.

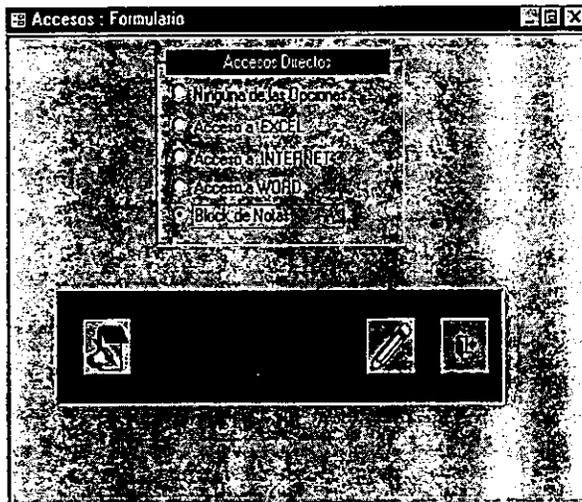


Figura 11. Ventana de Accesos

Lo que usted puede hacer en este Formulario de Accesos es que sin tener que cerrar SICLO-EA usted puede abrir cualquier aplicación de Office 97 como son: PowerPoint, Word y Excel, también puede abrir Internet, el Block de notas de Windows e incluso otra sesión de Access. Al cerrar este Formulario de Accesos usted podrá regresar al Formulario principal de SICLO-EA.

Para ejecutar el módulo de Catálogos bastará con dar un clic con el mouse sobre el botón con la leyenda de Catálogos. Al hacer esto se despliega la pantalla de la Figura 12.



Figura 12. Ventana de Catálogos

Lo que usted puede hacer en la pantalla de catálogos es hacer clic en cualquiera de los botones para abrir las siguientes pantallas que son submódulos de la pantalla de catálogos, toda la información que usted puede meter en estos submódulos, es información que le servirá cuando este llenando el módulo de Expedientes ya que la información que usted meta en cualquier submódulo de catálogos la vera en la pantalla de Expedientes, cuando este llenando ciertos campos como el banco, país, estado, y otros, en dichos campos le aparece un combo en el cual usted puede seleccionar alguno que anteriormente usted ya ha capturado antes, o usted puede dar de alta uno nuevo.

Los submódulos de la pantalla Catálogos son:

- Actividades
- Bancos

- Estado Civil
- Estados
- Hoteles
- Países
- Profesiones
- Tipo de pago
- Tipos de transporte
- Tipo de persona
- Universidades de Centro América
- Universidades e Instituciones de todo el Mundo

Para que pueda abrir el submodulo de Actividades lo único que tiene que hacer es dar un clic sobre el botón con el nombre de actividades. Al hacer esto el sistema lo que hace es abrir el catálogo de Actividades dicha pantalla se muestra en la Figura 13.

División de Estudios del Posgrado de la Facultad de Ingeniería	
Sistema de Control Logístico de Eventos Académicos	
Evento:	Inauguración del Evento
Capacidad:	20
Día:	6/27/98
Hora:	9:00:00 AM
Número de personas:	20
Salir	

Figura 13. Ventana Catálogo de Actividades

En este submodulo de actividades usted debe llenar este formulario, en esta pantalla usted puede definir las actividades que va a tener su evento por ejemplo: Conferencia Magistral, que día a que hora, el número de personas que pueden asistir a la actividad y el número de personas que están registradas para dicho evento o actividad. Al presionar el botón de salir

el sistema regresa a la pantalla anterior que es la del módulo de Catálogos y con esto usted puede escoger cualquier otro de los submodulos.

Si ahora usted quiere abrir el submodulo de Bancos lo único que tiene que hacer es dar un clic en el botón con el nombre de Bancos. Al hacer esto se abre la pantalla del submodulo de Bancos la cual se muestra en la Figura 14.

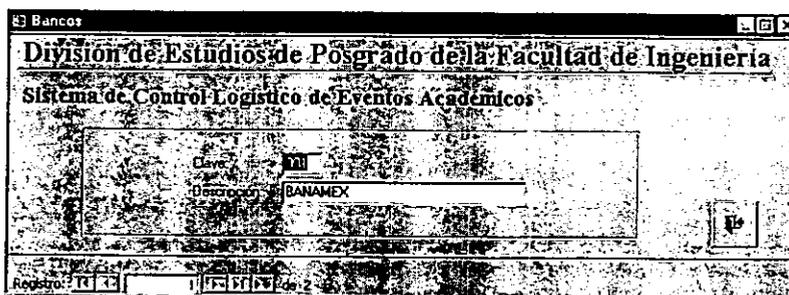


Figura 14. Ventana Catálogo de Bancos

Este módulo se creo para que usted pueda tener un control y poder saber donde se hizo el pago de la inscripción al evento, con esto podemos saber en que banco pago y en que cuenta se hizo el pago. ya que podemos tener cuenta en varios bancos, de esta forma usted puede llevar un control del dinero que esta ingresando a sus cuentas. Cuando presiona el botón de salir el sistema cierra la pantalla de bancos y regresa al módulo de catálogos.

Para abrir el catálogo de Estado Civil lo único que tiene que hacer es hacer clic en el botón con dicho nombre. Al hacer esto el sistema abre la pantalla de Estado Civil la cual se muestra en la Figura 15.



Figura 15. Ventana Catálogo de Estado Civil

Este catálogo se creó para poder saber el estado civil de las personas que asisten a su evento esto quiere decir que usted puede hacer un reporte del estado civil de las personas que asisten a su evento esto para tener una estadística y que usted pueda tomar decisiones para futuros eventos o en el mismo evento, tomar decisiones con relación al Estado Civil lo cual le puede ayudar a proponer un nuevo concepto de su evento u orientarlo más a personas solteras si es su caso o cualquiera que sea su caso esto le puede ayudar bastante a tomar decisiones.

Si lo que usted quiere es abrir el catálogo de Universidades de Centro América lo único que tiene que hacer es dar un clic en el botón Univ de Centro América. Al hacer esto el sistema lo que hace es abrir la dicho catálogo este se muestra en la Figura 16.

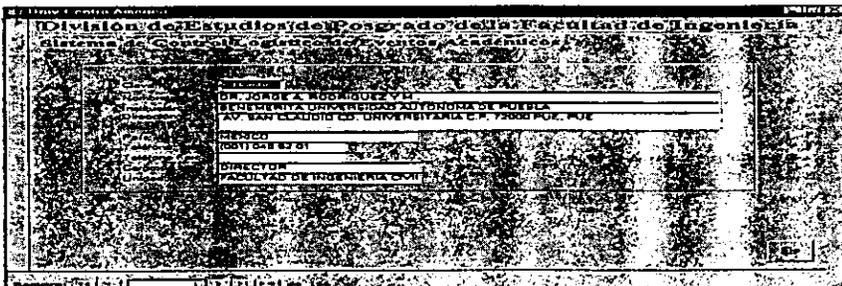


Figura 16. Ventana Catálogo de Universidades de Centro América

Este formulario se diseño para capturar las Universidades de Centro América ya que en la organización de eventos académicos existe relación con muchas Universidades y hay que mandarles cartas de invitación al evento el proceso de llenar los campos es el mismo que en los anteriores catálogos lo que hay que hacer es llenar todos los campos o la mayoría para que los informes que se saquen de eta información sean lo más completos que se pueda. Al hacer clic en el botón de salir el sistema lo regresa a la pantalla de Catálogos para que pueda seleccionar otro catálogo o salir hasta la pantalla principal.

Las siguientes figuras muestran los catálogos de los cuales no se dio una explicación detallada de ellos:

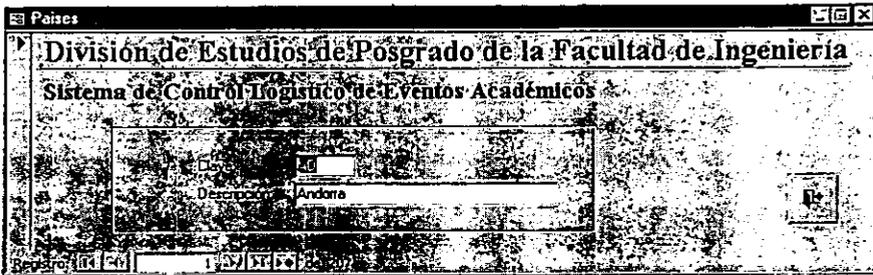


Figura 17. Ventana Catálogo de Países

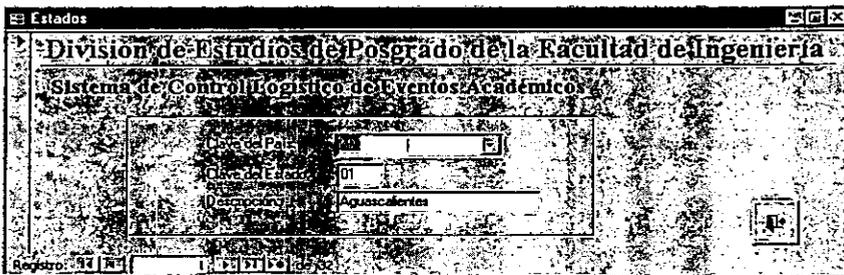


Figura 18. Ventana Catálogo de Estados

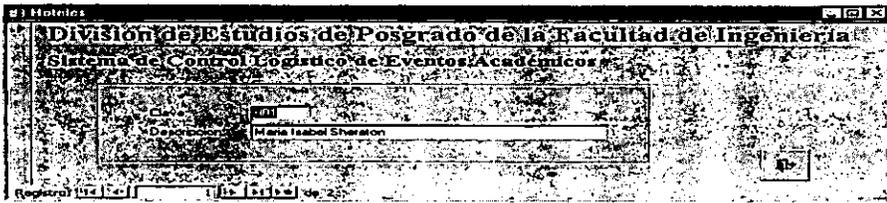


Figura 19. Ventana Catálogo de Hoteles

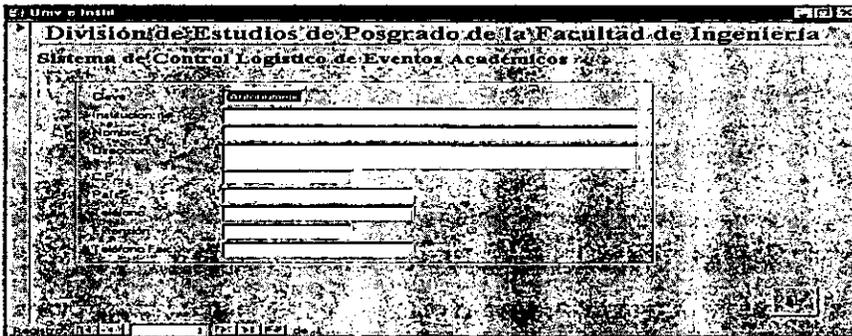


Figura 20. Ventana Catálogo de Universidades e Instituciones de todo el Mundo



Figura 21. Ventana Catálogo de Profesiones

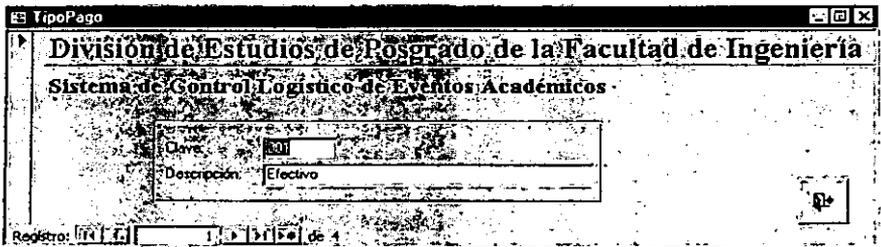


Figura 22. Ventana Catálogo Tipo de Pago

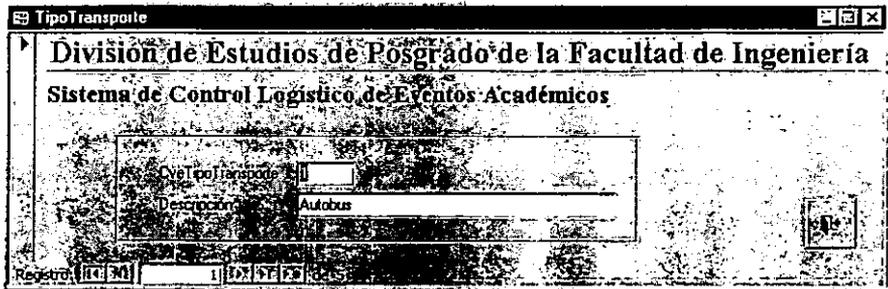


Figura 23. Catálogo Tipo de Transporte

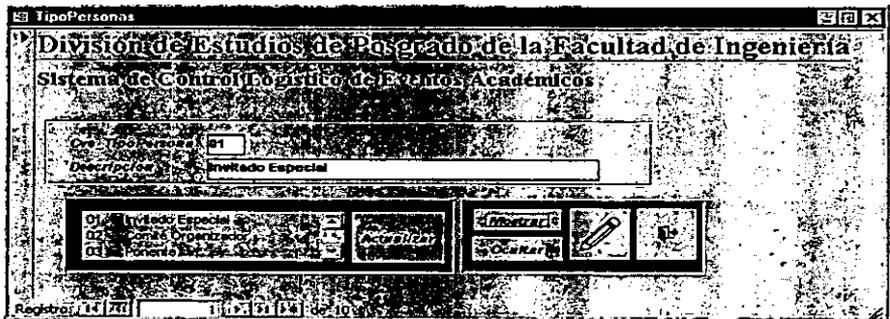


Figura 24. Catálogo Tipo de Personas

Todos los catálogos de SICLO-EA se llenan de la misma forma todos son en cierta forma parecidos y el proceso de llenado es el mismo, lo único que tiene que hacer es llenar el campo que usted desea con los datos adecuados a sus necesidades para que pueda tener la información en la pantalla de Expedientes que usted necesita, todos los campos son alfanuméricos.

Nota: Una vez que se le ha instalado SICLO-EA en su computadora, seguramente la base de datos esta vacía por lo que se recomienda, que lo primero debe hacer es llenar los Catálogos del sistema, ya que estos son los que alimentan de información a la tabla principal de la base de datos, si no lo hace así entonces podría tener problemas a la hora que este llenando el formulario de expedientes y le podría marcar un error el sistema, por lo que lo más recomendable es llenar los catálogos con la información que usted requiere antes de empezar a navegar en SICLO-EA.

Cuando ya cerro la pantalla de catálogos el sistema lo regresa a la pantalla anterior que en este caso es la pantalla principal en donde usted puede escoger el módulo de Búsqueda, para abrirlo lo único que debe hacer es un clic sobre el botón que esta rotulado como Búsqueda al hacer esto el sistema abre la pantalla de Búsqueda dicho formulario se muestra en la Figura 25.

Nombre de compañía	Apellido Paterno	Apellido Materno	Nombre	Teléfono	Asistencia
IBM	Narro	Ramirez	Ana Elena	7-73-55-95	si
UNAM	Ursul	Solanes	Juan	6-65-38-10	si
DEPI - UNAM	Ojeda	Toche	Lila	(77) 14-08-55	si
Instituto Nacional de Investigaciones	Conzález		Hector Eduardo	5748141	no
	Hidalgo	Ordana	Sanzuel de Jesús		si

Registro: 10 de 170

Figura 25. Ventana de Búsqueda rápida

En este formulario lo que usted puede hacer es una búsqueda rápida de alguna persona dicha búsqueda se hace por el Apellido Paterno, este formulario tiene botones con todas las letras del alfabeto entonces cuando presiona, cualquier letra por ejemplo la "A", lo que hace el formulario es jalar todos los registros con Apellidos que empiezan con "A", esta es una lista más pequeña que todos los registros de la base de datos lo cual hace más sencillo encontrar a la persona que esta buscando, si no se ha dado de alta no podrá encontrar el registro. Si presiona algún botón en el cual no hay ningún registro el sistema le indica que no hay Apellidos que empiecen con esa letra lo cual le puede ser útil si es que busca alguna persona con esa letra entonces usted ya sabe que la persona no se ha dado de alta en el sistema. Cuando usted oprima el botón de salir el sistema lo regresa a la pantalla anterior la cual es la pantalla principal.

Si esta en la pantalla principal y usted quiere ir al formulario de informes lo único que debe hacer es dar un clic sobre el botón con el nombre de Informes. Al hacer esto el sistema abre la pantalla de Informes este formulario se muestra en la Figura 26. En el formulario de Informes en la cual usted tiene cierto número de reportes los cuales le van a ser de gran ayuda para organizar y controlar su evento.

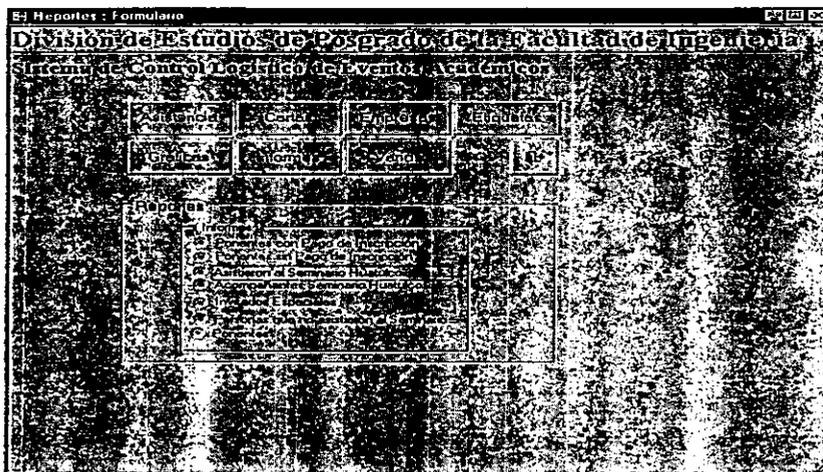


Figura 26. Ventana de Informes

Para que pueda abrir cualquiera de los reportes, lo único que tiene que hacer es seleccionar alguno de los botones, en los cuales están organizados en:

1. Asistencia
2. Cartas
3. Empresas
4. Etiquetas
5. Gráficas
6. Informes
7. Varios

Si selecciona el botón de Reportes el formulario le muestra todas las opciones que este botón tiene, los informes que el sistema en esta opción tiene son:

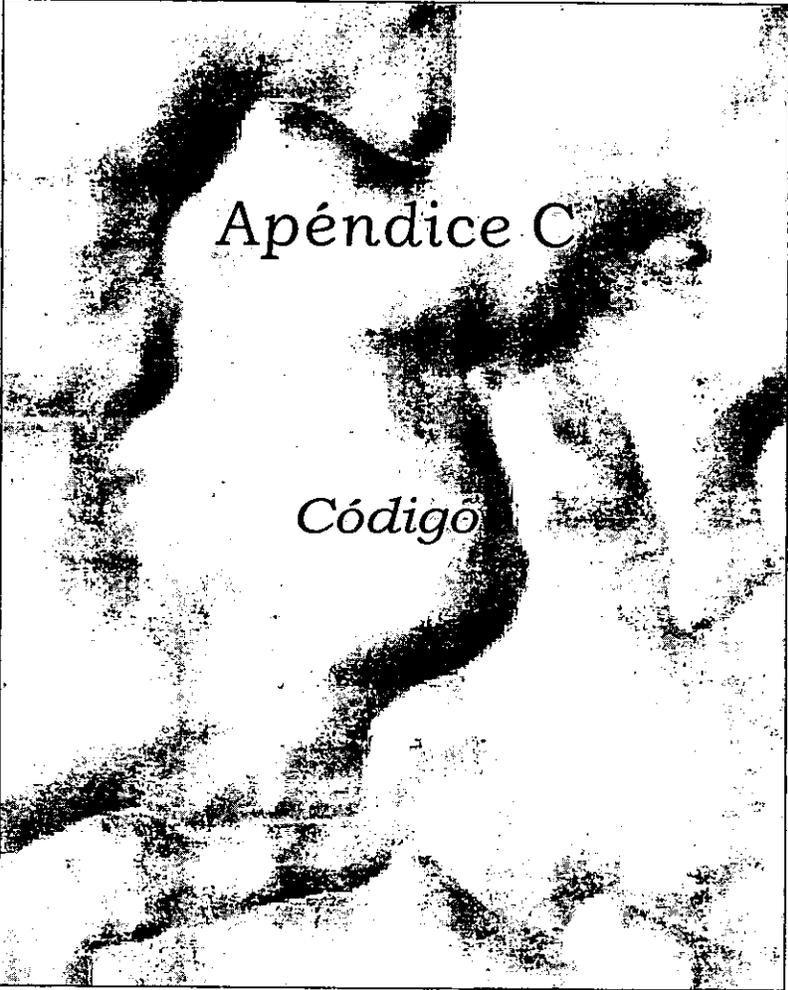
- Ponentes con pago de inscripción
- Ponentes sin pago de inscripción
- Invitados especiales
- Número de personas que asistieron al evento
- Personas que son acompañantes y que asistieron al evento
- Personas que están registradas y que no asistieron al evento
- Numero de ponentes inscritos al evento

Para poder ver alguno de estos reportes lo único que hay que hacer es seleccionar alguno, con el mouse y con esto el reporte se abre, debemos decir que estos reportes ya están diseñados y que cuando abre alguno el sistema carga los datos en el reporte y usted puede ver dichos informes, una vez que el reporte ya esta abierto, lo puede mandar a Word, Excel, o lo puede imprimir esto es para que usted pueda tomar decisiones de dichos reportes o ver como se esta comportando su evento.

Es preciso mencionarle que todos los reportes son similares en la parte de operación por lo que solo explicamos uno y los demás operan exactamente de la misma forma.

Si presiona el botón de salir el sistema nos regresa a la pantalla anterior que en este caso es la pantalla principal, si ya no quiere seguir trabajando con SICLO-EA lo único que tiene que hacer, no importando donde se encuentre presionar los botones de salir, hasta llegar a la pantalla principal de SICLO-EA, si ya esta en la pantalla principal lo que tiene que hacer es presionar de nuevo el botón de salir y en este formulario dicho botón lo que hace es cerrar todo y se sale completamente de SICLO-EA y de ACCESS, con esto usted acaba su sesión con SICLO-EA.

Este sistema le permite personalizar su evento, es decir que SICLO-EA le permite tener el nuevo nombre del evento, así como el tamaño de la letra, el color y otros parámetros, que le son de utilidad para personalizar su evento, esto quiere decir que si usted quiere organizar y controlar dos eventos con cambiarle en nombre al evento usted puede hacer referencia a cualquier evento que organice sin tener que tener cargado en su computadora dos versiones de SICLO-EA para poder organizar sus eventos.



Apêndice C

Código

Código SICLO-EA



Figura 1. Pantalla SICLO-EA

Código de la Figura 1. Pantalla SICLO-EA

Option Compare Database

Option Explicit

Private Sub Comando0_Click()

On Error GoTo Err_Comando0_Click

```
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Expediente"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Comando0_Click:
Exit Sub
Err_Comando0_Click:
MsgBox Err.Description
Resume Exit_Comando0_Click
End Sub
Private Sub Comando1_Click()
On Error GoTo Err_Comando1_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Catálogos"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Comando1_Click:
Exit Sub
Err_Comando1_Click:
MsgBox Err.Description
Resume Exit_Comando1_Click
End Sub
Private Sub Comando2_Click()
On Error GoTo Err_Comando2_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Seguridad y Usuarios"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Comando2_Click:
Exit Sub
Err_Comando2_Click:
```

```
MsgBox Err.Description
Resume Exit_Comando2_Click
End Sub
Private Sub Comando10_Click()
On Error GoTo Err_Comando10_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Catálogos"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Comando10_Click:
Exit Sub
Err_Comando10_Click:
MsgBox Err.Description
Resume Exit_Comando10_Click
End Sub
Private Sub Comando13_Click()
On Error GoTo Err_cmdInforme_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Reportes"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_cmdInforme_Click:
Exit Sub
Err_cmdInforme_Click:
MsgBox Err.Description
Resume Exit_cmdInforme_Click
End Sub
Private Sub Comando3_Click()
On Error GoTo Err_Comando3_Click
DoCmd.Close
Exit_Comando3_Click:
```

```
Exit Sub
Err_Comando3_Click:
    MsgBox Err.Description
    Resume Exit_Comando3_Click
End Sub
Private Sub cmdInforme_Click()
On Error GoTo Err_cmdInforme_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Reportes"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_cmdInforme_Click:
    Exit Sub
Err_cmdInforme_Click:
    MsgBox Err.Description
    Resume Exit_cmdInforme_Click
End Sub
Private Sub Comando9_Click()
On Error GoTo Err_Comando9_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Subexpediente"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Comando9_Click:
    Exit Sub
Err_Comando9_Click:
    MsgBox Err.Description
    Resume Exit_Comando9_Click
End Sub
Private Sub cmdBusqueda_Click()
On Error GoTo Err_cmdBusqueda_Click
```

```

Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Telefonos"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_cmdBusqueda_Click:
Exit Sub
Err_cmdBusqueda_Click:
MsgBox Err.Description
Resume Exit_cmdBusqueda_Click
End Sub

```

Subexpediente : Formulario

División de Estudios de Posgrado de la Facultad de Ingeniería

Sistema de Control Logístico de Eventos Académicos

Hector Nelson Mercado Rodriguez

Datos Personales | Datos Sistema

No. Expediente: 001

Nombre: Hector Nelson Mercado Rodriguez Estado Civil: Soltero

Calle: San boya # 1028 Edad: 25 años

Colonia: Del valle Sexo: Masculino

Del/Mun.: Benito Juarez ZP: 03100

País: México Estado:

Telefono 1: 55533002

Celular: Radiocalificador: Profesión: Ingeniero en Computación

Fac: Empresa: UNAM

E-mail: hectornelamocent.com Puesto:

Telefono 2: 5223278 (Asistir al evento:)

Registro: 13 33 16 18 20

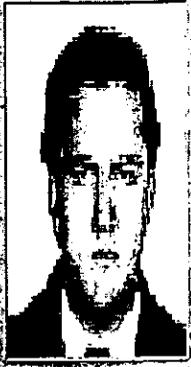


Figura 2. Pantalla Expediente

Código de la Figura 2. Pantalla Expediente

```
Option Compare Database
Option Explicit
Private Sub Comando13_Click()
    Lista16.Visible = True
    Comando17.Visible = True
    Cuadro15.Visible = True
End Sub
Private Sub Comando14_Click()
    Lista16.Visible = False
    Comando17.Visible = False
    Cuadro15.Visible = False
End Sub
Private Sub Comando17_Click()
On Error GoTo Err_Comando17_Click
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, 5, , acMenuVer70
Exit_Comando17_Click:
    Exit Sub
Err_Comando17_Click:
    MsgBox Err.Description
    Resume Exit_Comando17_Click
End Sub
Private Sub Comando20_Click()
On Error GoTo Err_Comando20_Click
    DoCmd.Close
Exit_Comando20_Click:
    Exit Sub
Err_Comando20_Click:
    MsgBox Err.Description
    Resume Exit_Comando20_Click
```

```
End Sub
```

```
Private Sub EncabezadoDelFormulario_Click()
```

```
End Sub
```

```
Private Sub lblCorreo_Click()
```

```
    E_mail.SetFocus
```

```
    If E_mail.Text <> "" Then
```

```
        lblCorreo.HyperlinkAddress = "mailto:" & E_mail.Text
```

```
    End If
```

```
End Sub
```

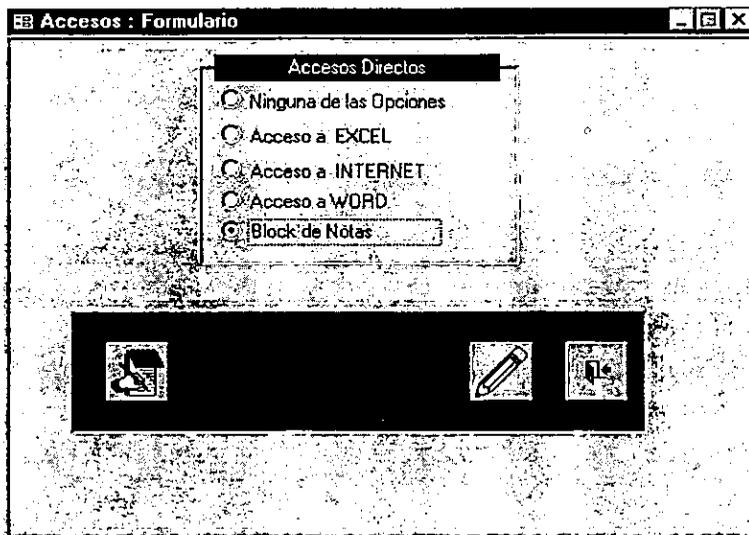


Figura 3. Pantalla Accesos

Código de la Figura 3. Pantalla Accesos

Option Compare Database

Option Explicit

```
Private Sub Cmdword_Click()
On Error GoTo Err_Cmdword_Click
    Dim oApp As Object
    Set oApp = CreateObject("Word.Application")
    oApp.Visible = True
Exit_Cmdword_Click:
    Exit Sub
Err_Cmdword_Click:
    MsgBox Err.Description
    Resume Exit_Cmdword_Click
End Sub

Private Sub cmdinternet_Click()
On Error GoTo Err_cmdinternet_Click
    Dim stAppName As String
    stAppName = "C:\Netscape\cp32e404.exe"
    Call Shell(stAppName, 1)
Exit_cmdinternet_Click:
    Exit Sub
Err_cmdinternet_Click:
    MsgBox Err.Description
    Resume Exit_cmdinternet_Click
End Sub

Private Sub Cmdintenet_Click()
On Error GoTo Err_Cmdintenet_Click
    Dim stAppName As String
    stAppName = "C:\Netscape\Communicator\Program\netscape.exe"
    Call Shell(stAppName, 1)
Exit_Cmdintenet_Click:
    Exit Sub
Err_Cmdintenet_Click:
    MsgBox Err.Description
```

```
Resume Exit_Cmdintenet_Click
End Sub
Private Sub Comando13_Click()
On Error GoTo Err_Comando13_Click
Dim oApp As Object
Set oApp = CreateObject("Word.Application")
oApp.Visible = True
Exit_Comando13_Click:
Exit Sub
Err_Comando13_Click:
MsgBox Err.Description
Resume Exit_Comando13_Click
End Sub
Private Sub Comando26_Click()
On Error GoTo Err_Comando26_Click
DoCmd.PrintOut
Exit_Comando26_Click:
Exit Sub
Err_Comando26_Click:
MsgBox Err.Description
Resume Exit_Comando26_Click
End Sub
Private Sub Comando28_Click()
On Error GoTo Err_Comando28_Click
Dim oApp As Object
Set oApp = CreateObject("Excel.Application")
oApp.Visible = True
'Only XL 97 supports UserControl Property
On Error Resume Next
oApp.UserControl = True
Exit_Comando28_Click:
```

```
Exit Sub
Err_Comando28_Click:
    MsgBox Err.Description
    Resume Exit_Comando28_Click
End Sub
Private Sub Comando30_Click()
On Error GoTo Err_Comando30_Click
    Dim stAppName As String
    stAppName = "c:\netscape\communicator\program\netscape.exe"
    Call Shell(stAppName, 1)
Exit_Comando30_Click:
    Exit Sub
Err_Comando30_Click:
    MsgBox Err.Description
    Resume Exit_Comando30_Click
End Sub
Private Sub Comando32_Click()
On Error GoTo Err_Comando32_Click
    Call Shell("NOTEPAD.EXE", 1)
Exit_Comando32_Click:
    Exit Sub
Err_Comando32_Click:
    MsgBox Err.Description
    Resume Exit_Comando32_Click
End Sub
Private Sub Comando36_Click()
On Error GoTo Err_Comando36_Click
    Dim oApp As Object
    Set oApp = CreateObject("Word.Application")
    oApp.Visible = True
Exit_Comando36_Click:
```

```
Exit Sub
Err_Comando36_Click:
    MsgBox Err.Description
    Resume Exit_Comando36_Click
End Sub
Private Sub Marco14_Click()
Select Case Marco14
    Case "1"
        Cuadro11.Visible = True
        Comando26.Visible = True
        cmdSalir.Visible = True
        Comando32.Visible = False
        Comando30.Visible = False
        Comando36.Visible = False
        Comando28.Visible = False
    Case "2"
        Cuadro11.Visible = True
        Comando26.Visible = True
        cmdSalir.Visible = True
        Comando32.Visible = False
        Comando30.Visible = False
        Comando36.Visible = False
        Comando28.Visible = True
    Case "3"
        Cuadro11.Visible = True
        Comando26.Visible = True
        cmdSalir.Visible = True
        Comando32.Visible = False
        Comando30.Visible = True
        Comando36.Visible = False
        Comando28.Visible = False
```

Case "4"

Cuadro11.Visible = True

Comando26.Visible = True

cmdSalir.Visible = True

Comando32.Visible = False

Comando30.Visible = False

Comando36.Visible = True

Comando28.Visible = False

Case "5"

Cuadro11.Visible = True

Comando26.Visible = True

cmdSalir.Visible = True

Comando32.Visible = True

Comando30.Visible = False

Comando36.Visible = False

Comando28.Visible = False

End Select

End Sub

Private Sub Comando38_Click()

On Error GoTo Err_Comando38_Click

Dim oApp As Object

Set oApp = CreateObject("Word.Application")

oApp.Visible = True

Exit_Comando38_Click:

Exit Sub

Err_Comando38_Click:

MsgBox Err.Description

Resume Exit_Comando38_Click

End Sub



Figura 4. Pantalla Catálogos

Código de la Figura 4. Pantalla Catálogos

Option Compare Database

Option Explicit

Private Sub Comando0_Click()

On Error GoTo Err_Comando0_Click

Dim stDocName As String

```
    Dim stLinkCriteria As String
    stDocName = "Cat_Actividades"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Comando0_Click:
    Exit Sub
Err_Comando0_Click:
    MsgBox Err.Description
    Resume Exit_Comando0_Click
End Sub
Private Sub Comando1_Click()
On Error GoTo Err_Comando1_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Cat_Bancos"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Comando1_Click:
    Exit Sub
Err_Comando1_Click:
    MsgBox Err.Description
    Resume Exit_Comando1_Click
End Sub
Private Sub Comando2_Click()
On Error GoTo Err_Comando2_Click
    Screen.PreviousControl.SetFocus
    DoCmd.DoMenuItem acFormBar, acEditMenu, 10, , acMenuVer70
Exit_Comando2_Click:
    Exit Sub
Err_Comando2_Click:
    MsgBox Err.Description
    Resume Exit_Comando2_Click
End Sub
```

```
Private Sub Comando3_Click()
On Error GoTo Err_Comando3_Click
    Screen.PreviousControl.SetFocus
    DoCmd.DoMenuItem acFormBar, acEditMenu, 10, , acMenuVer70
Exit_Comando3_Click:
    Exit Sub
Err_Comando3_Click:
    MsgBox Err.Description
    Resume Exit_Comando3_Click
End Sub
Private Sub Comando19_Click()
End Sub
Private Sub Comando4_Click()
On Error GoTo Err_Comando4_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Cat_EdoCivil"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Comando4_Click:
    Exit Sub
Err_Comando4_Click:
    MsgBox Err.Description
    Resume Exit_Comando4_Click
End Sub
Private Sub Comando5_Click()
On Error GoTo Err_Comando5_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Cat_Estados"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Comando5_Click:
```

Exit Sub

Err_Comando5_Click:

MsgBox Err.Description

Resume Exit_Comando5_Click

End Sub

Private Sub Comando6_Click()

On Error GoTo Err_Comando6_Click

Dim stDocName As String

Dim stLinkCriteria As String

stDocName = "Cat_Hoteles"

DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Comando6_Click:

Exit Sub

Err_Comando6_Click:

MsgBox Err.Description

Resume Exit_Comando6_Click

End Sub

Private Sub Comando7_Click()

On Error GoTo Err_Comando7_Click

Dim stDocName As String

Dim stLinkCriteria As String

stDocName = "Cat_Paises"

DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Comando7_Click:

Exit Sub

Err_Comando7_Click:

MsgBox Err.Description

Resume Exit_Comando7_Click

End Sub

Private Sub Comando8_Click()

```
On Error GoTo Err_Comando8_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Cat_Profesion"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Comando8_Click:
    Exit Sub
Err_Comando8_Click:
    MsgBox Err.Description
    Resume Exit_Comando8_Click
End Sub
Private Sub Comando9_Click()
On Error GoTo Err_Comando9_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Cat_TipoPago"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Comando9_Click:
    Exit Sub
Err_Comando9_Click:
    MsgBox Err.Description
    Resume Exit_Comando9_Click
End Sub
Private Sub Comando10_Click()
On Error GoTo Err_Comando10_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Cat_TipoTransporte"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Comando10_Click:
    Exit Sub
```

```
Err_Comando10_Click:
    MsgBox Err.Description
    Resume Exit_Comando10_Click
End Sub
Private Sub Comando12_Click()
On Error GoTo Err_Comando12_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Cat_Actividades"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Comando12_Click:
    Exit Sub
Err_Comando12_Click:
    MsgBox Err.Description
    Resume Exit_Comando12_Click
End Sub
Private Sub Comando13_Click()
On Error GoTo Err_Comando13_Click
    DoCmd.Close
Exit_Comando13_Click:
    Exit Sub
Err_Comando13_Click:
    MsgBox Err.Description
    Resume Exit_Comando13_Click
End Sub
Private Sub Comando15_Click()
On Error GoTo Err_Comando15_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Univ Centro America"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
```

```
Exit_Comando15_Click:
    Exit Sub
Err_Comando15_Click:
    MsgBox Err.Description
    Resume Exit_Comando15_Click
End Sub
Private Sub Comando16_Click()
On Error GoTo Err_Comando16_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Univ e Instit"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Comando16_Click:
    Exit Sub
Err_Comando16_Click:
    MsgBox Err.Description
    Resume Exit_Comando16_Click
End Sub
Private Sub Comando18_Click()
On Error GoTo Err_Comando18_Click
    Screen.PreviousControl.SetFocus
    DoCmd.DoMenuItem acFormBar, acEditMenu, 10, , acMenuVer70
Exit_Comando18_Click:
    Exit Sub
Err_Comando18_Click:
    MsgBox Err.Description
    Resume Exit_Comando18_Click
End Sub
```

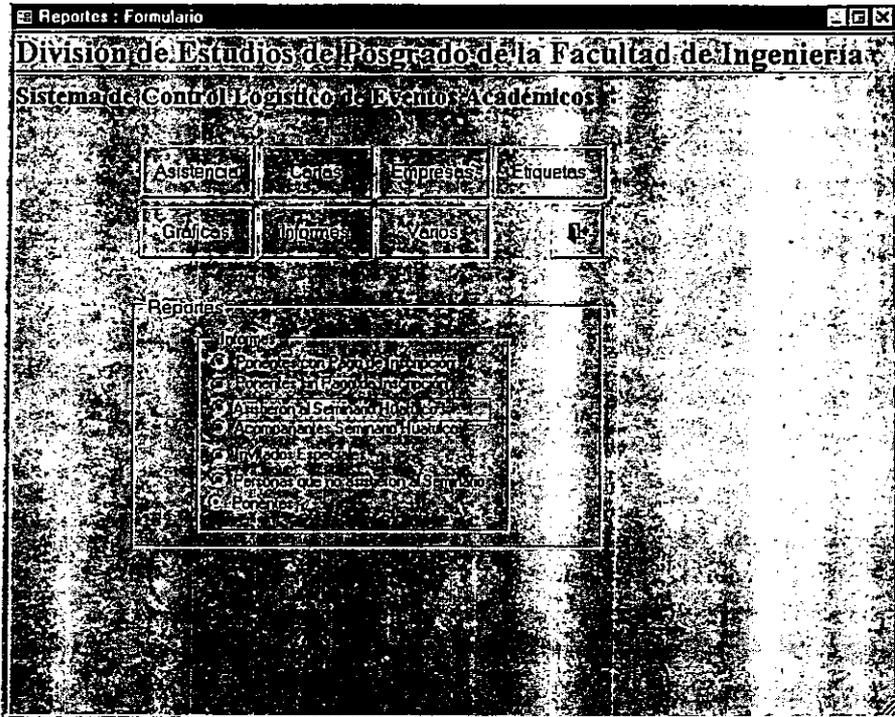


Figura 5. Pantalla Informes

Código de la Figura 5. Pantalla Informes

Option Compare Database

Option Explicit

Private Sub Comando5_Click()

 Unload Me

End Sub

Private Sub cmdAsistencia_Click()

 Asistencia

 NoCartas

 NoEmpresas

```
NoEtiquetas
NoGraficas
NoInformes
NoVarios
End Sub
Private Sub cmdCartas_Click()
    Cartas
    NoAsistencia
    NoEmpresas
    NoEtiquetas
    NoGraficas
    NoInformes
    NoVarios
End Sub
Private Sub cmdEmpresas_Click()
    NoAsistencia
    NoCartas
    Empresas
    NoEtiquetas
    NoGraficas
    NoInformes
    NoVarios
End Sub
Private Sub cmdEtiquetas_Click()
    NoAsistencia
    NoCartas
    NoEmpresas
    Etiquetas
    NoGraficas
    NoInformes
    NoVarios
```

End Sub

Private Sub cmdGraficas_Click()

NoAsistencia

NoCartas

NoEmpresas

NoEtiquetas

Graficas

NoInformes

NoVarios

End Sub

Private Sub cmdInformes_Click()

NoAsistencia

NoCartas

NoEmpresas

NoEtiquetas

NoGraficas

Informes

NoVarios

End Sub

Private Sub cmdSalir_Click()

On Error GoTo Err_cmdSalir_Click

DoCmd.Close

Exit_cmdSalir_Click:

Exit Sub

Err_cmdSalir_Click:

MsgBox Err.Description

Resume Exit_cmdSalir_Click

End Sub

Public Sub Asistencia()

pnlAsistencia.Visible = True

marAsistencia.Visible = True

```
optNum0.Visible = True
```

```
optNum1.Visible = True
```

```
optNum2.Visible = True
```

```
End Sub
```

```
Public Sub NoAsistencia()
```

```
    pnlAsistencia.Visible = False
```

```
    marAsistencia.Visible = False
```

```
    optNum0.Visible = False
```

```
    optNum1.Visible = False
```

```
    optNum2.Visible = False
```

```
End Sub
```

```
Public Sub Cartas()
```

```
    pnlCartas.Visible = True
```

```
    marCartas.Visible = True
```

```
    optNum3.Visible = True
```

```
    optNum4.Visible = True
```

```
    optNum5.Visible = True
```

```
End Sub
```

```
Public Sub NoCartas()
```

```
    pnlCartas.Visible = False
```

```
    marCartas.Visible = False
```

```
    optNum3.Visible = False
```

```
    optNum4.Visible = False
```

```
    optNum5.Visible = False
```

```
End Sub
```

```
Public Sub NoEmpresas()
```

```
    pnlEmpresas.Visible = False
```

```
    marEmpresas.Visible = False
```

```
    optNum8.Visible = False
```

```
    optNum9.Visible = False
```

```
    optNum10.Visible = False
```

optNum11.Visible = False

optNum12.Visible = False

End Sub

Public Sub Empresas()

pnlEmpresas.Visible = True

marEmpresas.Visible = True

optNum8.Visible = True

optNum9.Visible = True

optNum10.Visible = True

optNum11.Visible = True

optNum12.Visible = True

End Sub

Public Sub NoEtiquetas()

lblEtiquetas.Visible = False

marEtiquetas.Visible = False

optNum13.Visible = False

optNum14.Visible = False

optNum15.Visible = False

optNum16.Visible = False

optNum17.Visible = False

End Sub

Public Sub Etiquetas()

lblEtiquetas.Visible = True

marEtiquetas.Visible = True

optNum13.Visible = True

optNum14.Visible = True

optNum15.Visible = True

optNum16.Visible = True

optNum17.Visible = True

End Sub

Public Sub NoGraficas()

```
lblGraficas.Visible = False
marGraficas.Visible = False
optNum18.Visible = False
optNum19.Visible = False
optNum20.Visible = False
optNum21.Visible = False
```

End Sub

Public Sub Graficas()

```
lblGraficas.Visible = True
marGraficas.Visible = True
optNum18.Visible = True
optNum19.Visible = True
optNum20.Visible = True
optNum21.Visible = True
```

End Sub

Public Sub NoInformes()

```
lblInformes.Visible = False
marInformes.Visible = False
optNum82.Visible = False
optNum84.Visible = False
optNum86.Visible = False
optNum22.Visible = False
optNum23.Visible = False
optNum24.Visible = False
optNum25.Visible = False
```

End Sub

Public Sub Informes()

```
lblInformes.Visible = True
marInformes.Visible = True
optNum82.Visible = True
optNum84.Visible = True
```

```
optNum86.Visible = True  
optNum22.Visible = True  
optNum23.Visible = True  
optNum24.Visible = True  
optNum25.Visible = True
```

End Sub

Public Sub NoVarios()

```
lblVarios.Visible = False  
marVarios.Visible = False  
optNum30.Visible = False  
optNum31.Visible = False  
optNum32.Visible = False  
optNum33.Visible = False  
optNum34.Visible = False
```

End Sub

Public Sub Varios()

```
lblVarios.Visible = True  
marVarios.Visible = True  
optNum30.Visible = True  
optNum31.Visible = True  
optNum32.Visible = True  
optNum33.Visible = True  
optNum34.Visible = True
```

End Sub

Private Sub cmdVarios_Click()

```
NoAsistencia  
NoCartas  
NoEmpresas  
NoEtiquetas  
NoGraficas  
NoInformes
```

Varios

End Sub

Private Sub Form_Load()

NoAsistencia

NoCartas

NoEmpresas

NoEtiquetas

NoGraficas

NoInformes

NoVarios

End Sub

Nombre de compañía	Apellido Paterno	Apellido Materno	Nombre	Telefonos	Asistencia
IBM	Narro	Ramirez	Ana Elena	7-23-55-95	si
UNAM	Ursul	Solones	Juan	6-65-38-10	si
DEPT - UNAM	Ojeda	Toche	Lilia	(72) 14-08-55	si
Instituto Nacional de Investigaciones	González		Hector Eduardo	5748141	no
	Hidalgo	Orellana	Samuel de Jesús		si

Figura 6. Pantalla de Búsqueda

Código de la Figura 6. Pantalla de Búsqueda

Option Compare Database

Option Explicit

APÉNDICE C


```

On Error GoTo Err_Comando12_Click
    DoCmd.Close
Exit_Comando12_Click:
    Exit Sub
Err_Comando12_Click:
    MsgBox Err.Description
    Resume Exit_Comando12_Click
End Sub

```

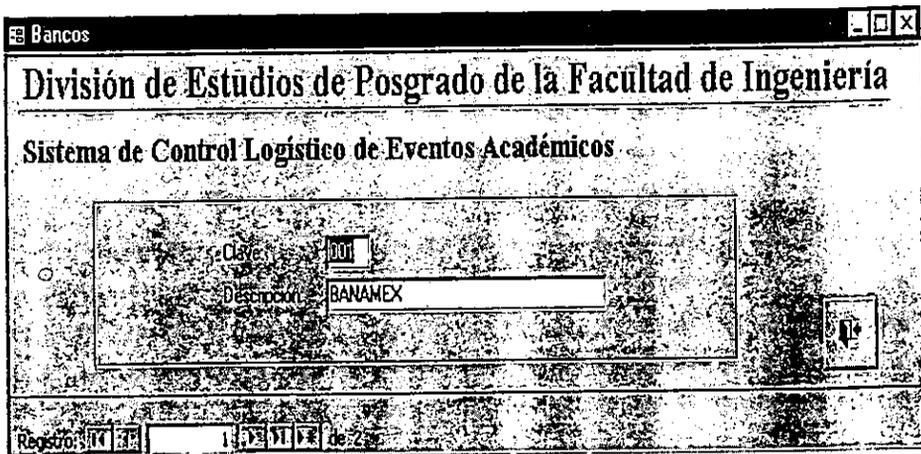


Figura 8. Pantalla Catálogo de Bancos

Código de la Figura 8. Pantalla Catálogo de Bancos

```

Option Compare Database
Option Explicit
Private Sub Comando7_Click()
On Error GoTo Err_Comando7_Click
    DoCmd.Close
Exit_Comando7_Click:

```

```
Exit Sub  
Err_Comando7_Click:  
    MsgBox Err.Description  
    Resume Exit_Comando7_Click  
End Sub
```

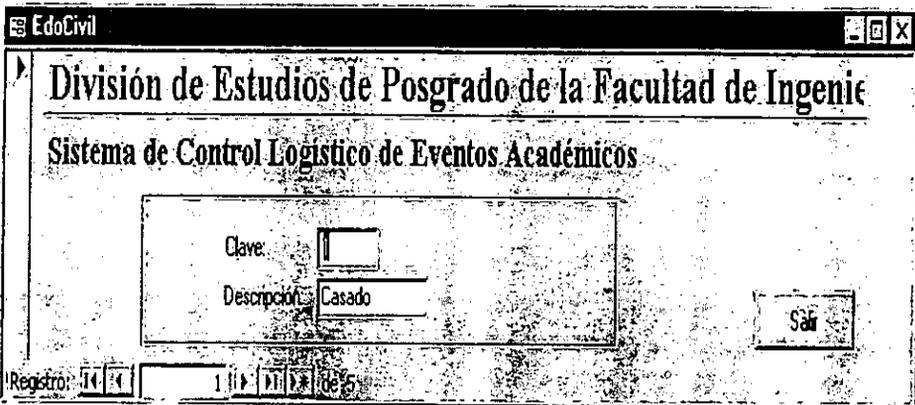


Figura 9. Pantalla Catálogo de Estado Civil

Código de la Figura 9. Pantalla Catálogo de Estado Civil

```
Option Compare Database  
Option Explicit  
Private Sub Comando6_Click()  
On Error GoTo Err_Comando6_Click  
    DoCmd.Close  
Exit_Comando6_Click:  
    Exit Sub  
Err_Comando6_Click:  
    MsgBox Err.Description  
    Resume Exit_Comando6_Click  
End Sub
```

Univ Centro America

División de Estudios de Posgrado de la Facultad de Ingeniería

Sistema de Control Logístico de Eventos Académicos

Clave:	<input type="text" value="Nombre"/>
Nombre:	DR. JORGE A. RODRIGUEZ Y M.
Institución:	BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA
Dirección:	AV. SAN CLAUDIO CD. UNIVERSITARIA C.P. 72000 PUE. PUE
País:	MEXICO
Teléfono:	(001) 045 82 01
Teléfono Fax:	
Cargo:	DIRECTOR
Unidad:	FACULTAD DE INGENIERIA CIVIL

Requisito: 14 de 16

Figura 10. Pantalla Catálogo Universidades de Centro América

Código de la Figura 10. Pantalla Universidades de Centro América

```

Option Compare Database
Option Explicit
Private Sub cmdSalir_Click()
On Error GoTo Err_cmdSalir_Click
    DoCmd.Close
Exit_cmdSalir_Click:
    Exit Sub
Err_cmdSalir_Click:
    MsgBox Err.Description
    Resume Exit_cmdSalir_Click
End Sub

```

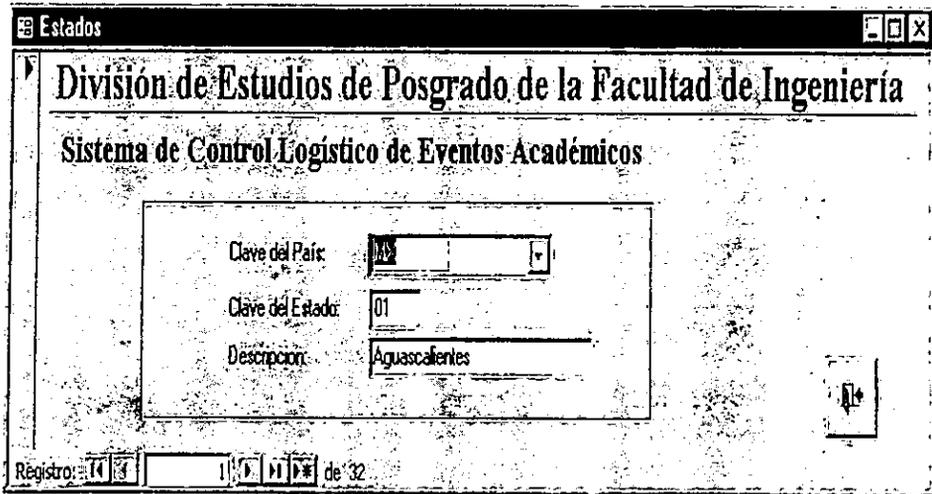


Figura 11. Pantalla Catálogo de Estados

Código de la Figura 11. Pantalla Catálogo de Estados

```

Option Compare Database
Option Explicit
Private Sub Comando8_Click()
On Error GoTo Err_Comando8_Click
    DoCmd.Close
Exit_Comando8_Click:
    Exit Sub
Err_Comando8_Click:
    MsgBox Err.Description
    Resume Exit_Comando8_Click
End Sub

```

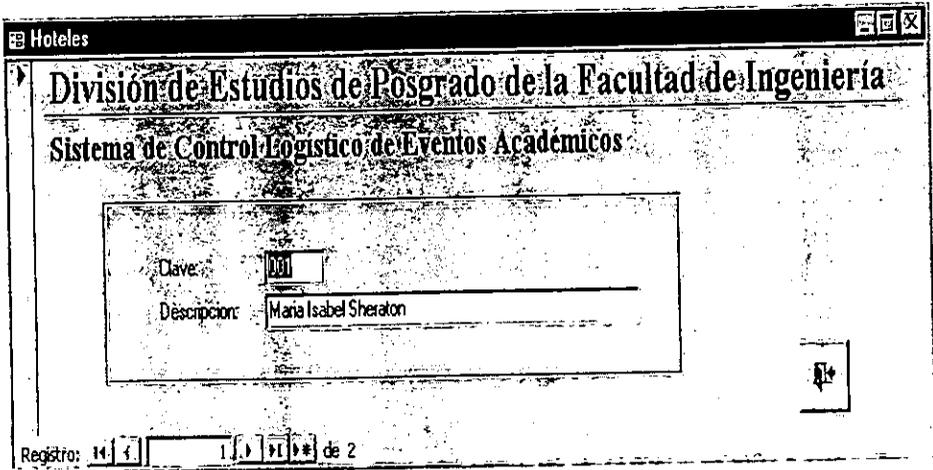


Figura 12. Pantalla Catálogo de Hoteles

Código de la Figura 12. Pantalla Catálogo de Hoteles

Option Compare Database

Option Explicit

Private Sub Comando4_Click()

On Error GoTo Err_Comando4_Click

DoCmd.Close

Exit_Comando4_Click:

Exit Sub

Err_Comando4_Click:

MsgBox Err.Description

Resume Exit_Comando4_Click

End Sub

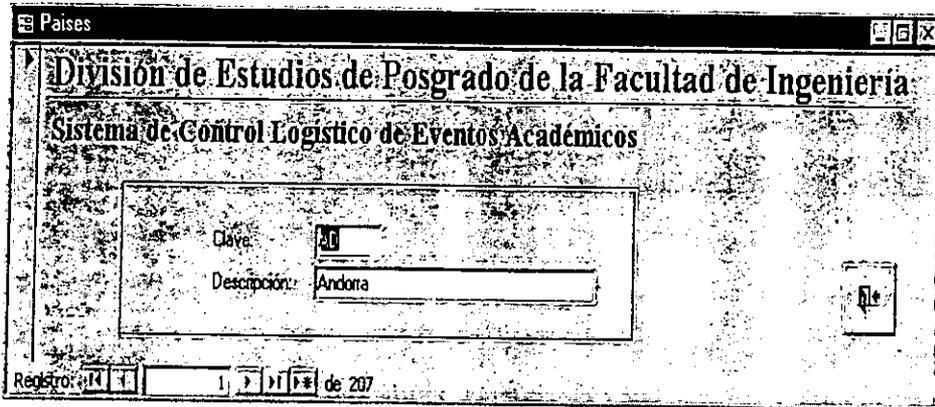


Figura 13. Pantalla Catálogo de Países

Código de la Figura 13. Pantalla Catálogo de Países

Option Compare Database

Option Explicit

Private Sub Comando6_Click()

On Error GoTo Err_Comando6_Click

DoCmd.Close

Exit_Comando6_Click:

Exit Sub

Err_Comando6_Click:

MsgBox Err.Description

Resume Exit_Comando6_Click

End Sub

Univ e Instit

División de Estudios de Posgrado de la Facultad de Ingeniería

Sistema de Control Logístico de Eventos Académicos

Clave:

Institución:

Nombre:

Dirección:

C.P.:

País:

Teléfono:

Extensión:

Teléfono Fax:

Registro: 14 de 1

Figura 14. Pantalla Catálogo de Universidades e Instituciones

Código de la Figura 14. Pantalla Catálogo de Universidades e Instituciones

```

Option Compare Database
Option Explicit
Private Sub cmdSalir_Click()
On Error GoTo Err_cmdSalir_Click
DoCmd.Close
Exit_cmdSalir_Click:
Exit Sub
Err_cmdSalir_Click:
MsgBox Err.Description

```

Resume Exit_cmdSalir_Click
 End Sub

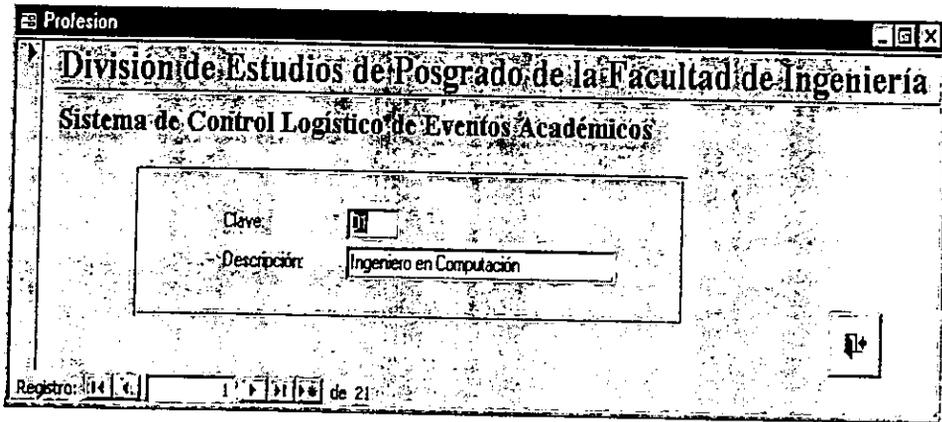


Figura 15. Pantalla Catálogo de Profesiones

Código de la Figura 15. Pantalla Catálogo de Profesiones

```
Option Compare Database
Option Explicit
Private Sub Comando6_Click()
On Error GoTo Err_Comando6_Click
    DoCmd.Close
Exit_Comando6_Click:
    Exit Sub
Err_Comando6_Click:
    MsgBox Err.Description
    Resume Exit_Comando6_Click
End Sub
```

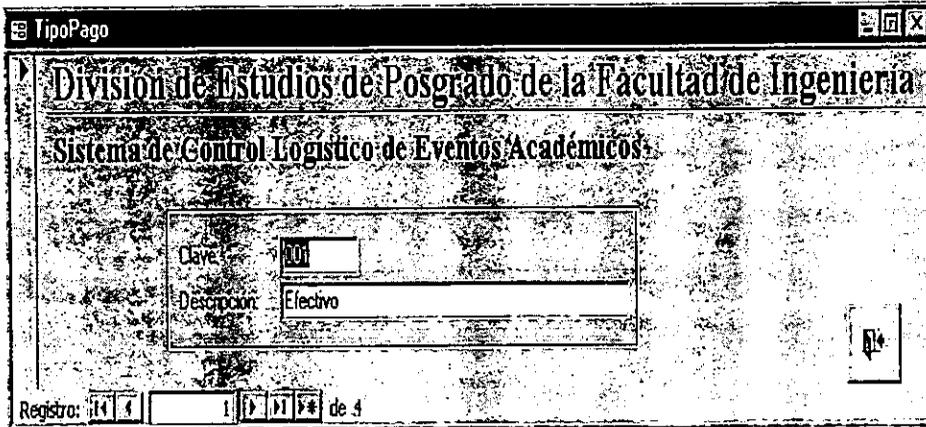


Figura 16. Pantalla Catálogo de Tipo de Pago

Código de la Figura 16. Pantalla Catálogo de Tipo de Pago

Option Compare Database

Option Explicit

Private Sub Comando6_Click()

On Error GoTo Err_Comando6_Click

DoCmd.Close

Exit_Comando6_Click:

Exit Sub

Err_Comando6_Click:

MsgBox Err.Description

Resume Exit_Comando6_Click

End Sub

APÉNDICE C

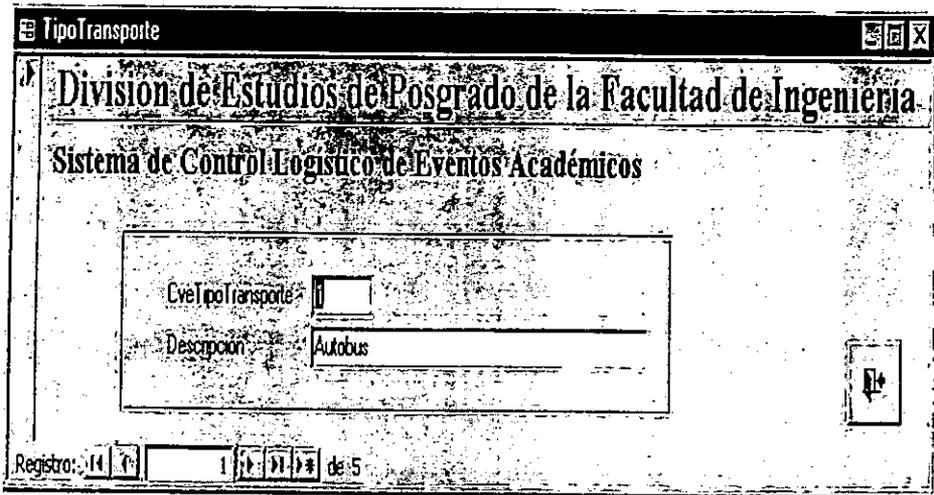


Figura 17. Pantalla Catálogo Tipo de Transporte

Código de la Figura 17. Pantalla Catálogo Tipo de Transporte

Option Compare Database

Option Explicit

Private Sub Comando6_Click()

On Error GoTo Err_Comando6_Click

 DoCmd.Close

Exit_Comando6_Click:

 Exit Sub

Err_Comando6_Click:

 MsgBox Err.Description

 Resume Exit_Comando6_Click

End Sub

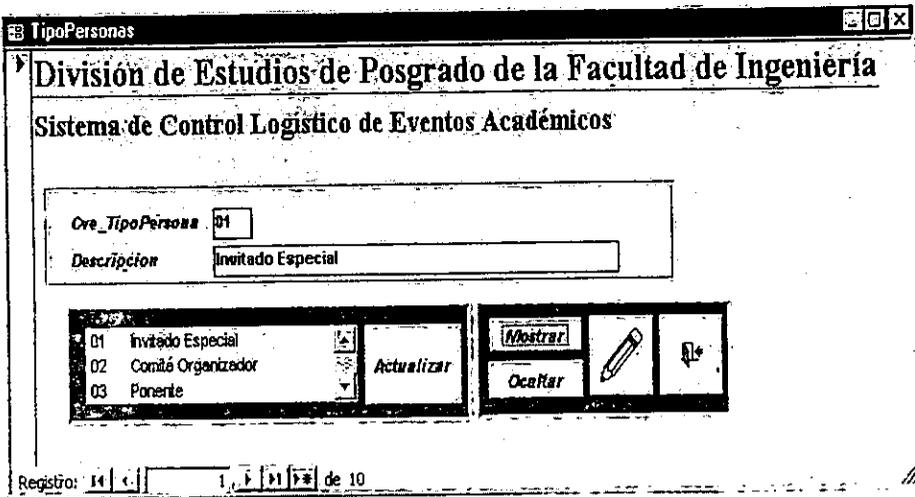


Figura 18. Pantalla Catálogo Tipo de Personas

Código de la Figura 18. Pantalla Catálogo Tipo e Personas

Option Compare Database

Option Explicit

Private Sub Comando5_Click()

On Error GoTo Err_Comando5_Click

DoCmd.Close

Exit_Comando5_Click:

Exit Sub

Err_Comando5_Click:

MsgBox Err.Description

Resume Exit_Comando5_Click

End Sub

Private Sub Comando6_Click()

APÉNDICE C

On Error GoTo Err_Comando6_Click

DoCmd.PrintOut

Exit_Comando6_Click:

Exit Sub

Err_Comando6_Click:

MsgBox Err.Description

Resume Exit_Comando6_Click

End Sub

Private Sub Comando12_Click()

On Error GoTo Err_Comando12_Click

DoCmd.DoMenuItem acFormBar, acRecordsMenu, 5, , acMenuVer70

Exit_Comando12_Click:

Exit Sub

Err_Comando12_Click:

MsgBox Err.Description

Resume Exit_Comando12_Click

End Sub

Private Sub Comando7_Click()

Cuadro9.Visible = True

Comando12.Visible = True

Lista10.Visible = True

End Sub

Private Sub Comando8_Click()

Cuadro9.Visible = False

Comando12.Visible = False

Lista10.Visible = False

End Sub

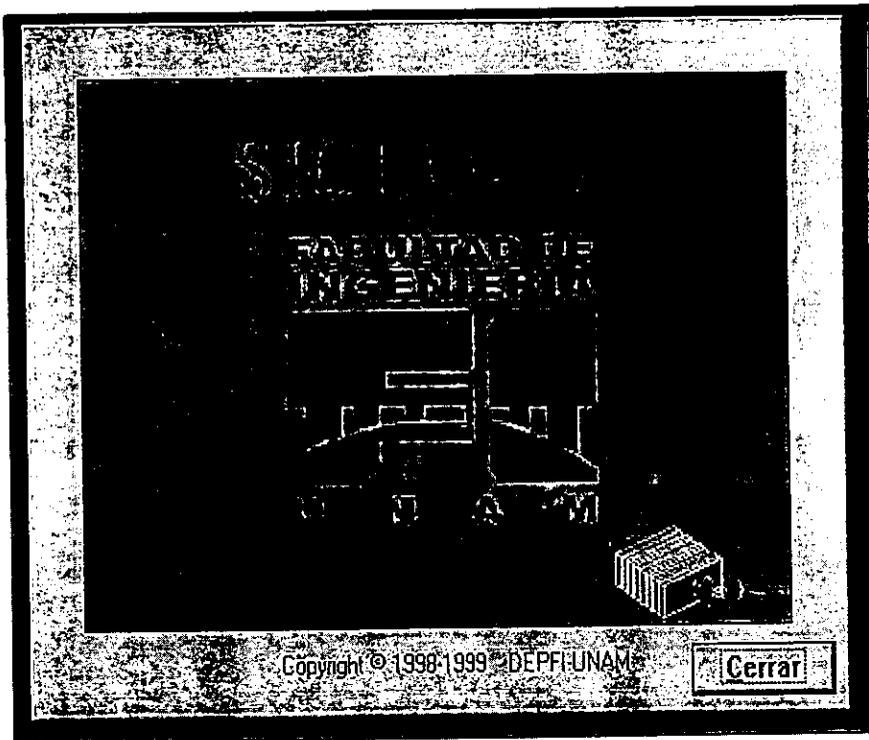


Figura 19. Pantalla Presentación

Código de la Figura 19. Pantalla Presentación

Option Compare Database

Option Explicit



Figura 20. Pantalla Principal

Código de la Figura 20. Pantalla Principal

```
Option Compare Database
Option Explicit
```

```
Private Sub Comando3_Click()
On Error GoTo Err_Comando3_Click
```

```
DoCmd.Close
```

```
Exit_Comando3_Click:
Exit Sub
```

```
Err_Comando3_Click:
MsgBox Err.Description
```

```
Resume Exit_Comando3_Click
```

```
End Sub
```

```
Private Sub Comando21_Click()
```

```
On Error GoTo Err_Comando21_Click
```

```
Dim stDocName As String  
Dim stLinkCriteria As String
```

```
stDocName = "Parametros"  
DoCmd.OpenForm stDocName, , , stLinkCriteria
```

```
Exit_Comando21_Click:
```

```
Exit Sub
```

```
Err_Comando21_Click:
```

```
MsgBox Err.Description  
Resume Exit_Comando21_Click
```

```
End Sub
```

```
Private Sub Comando22_Click()
```

```
On Error GoTo Err_Comando22_Click
```

```
Dim stDocName As String  
Dim stLinkCriteria As String
```

```
stDocName = "siclo-ea"  
DoCmd.OpenForm stDocName, , , stLinkCriteria
```

```
Exit_Comando22_Click:
```

```
Exit Sub
```

```
Err_Comando22_Click:
```

```
MsgBox Err.Description  
Resume Exit_Comando22_Click
```

```
DoCmd.Close
```

```
End Sub
```

```
Private Sub Detalle_Click()
```

End Sub

Private Sub Form_Deactivate()

 DoCmd.ShowToolbar "Vinculos", acToolbarWhereApprop

End Sub

Private Sub Form_Load()

 DoCmd.RunMacro "CerrarSeguridad"

 Comando22.SetFocus

End Sub

Private Sub Comando20_Click()

On Error GoTo Err_Comando20_Click

 Dim stDocName As String

 Dim stLinkCriteria As String

 stDocName = "Parametros"

 DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Comando20_Click:

Exit Sub

Err_Comando20_Click:

 MsgBox Err.Description

 Resume Exit_Comando20_Click

End Sub

GLOSARIO

Access Basic. Lenguaje de programación orientado a eventos parecido a Visual Basic para el desarrollo de aplicaciones diseñadas para ejecutarse en Access de Microsoft Windows.

Algoritmo. Procedimiento matemático o lógico para resolver un problema.

Base de Datos. Colección de datos interrelacionados de tal forma que pueden representarse como varios archivos pero no como uno sólo. En una base de datos las definiciones de los datos y las relaciones entre ellos están separadas de las declaraciones de procedimientos de un programa.

Base de Datos Relacional. Forma de especificación de base de datos en la que se consigue la estructuración por la utilización de operadores relacionales.

Bit. Unidad básica de información en un sistema de numeración binario; proviene del término Binary digit (*dígito binario*). La circuitería electrónica de las computadoras detecta la diferencia entre dos estados (*corriente alta o baja*) y representa estos estados como uno de los dos números de un sistema binario: el 1 o el 0. Estas unidades básicas de información se conocen como bits.

Buffer. Memoria provisional de datos, se utiliza frecuentemente para adecuar la diferencia de velocidades de dos dispositivos de tratamiento de datos durante la transferencia. La memoria puede estar dentro de un dispositivo periférico, tal como una impresora o una unidad de disco duro o también puede formar parte del sistema de la memoria central.

Byte. Ocho bits contiguos, que es la unidad de información fundamental. Como guarda el equivalente de un carácter, el byte también es una unidad de medida básica para el almacenamiento en computadoras.

Código. Un conjunto de reglas no ambiguas que especifican la forma en que los datos son representados.

Compresión. Técnica usada para aumentar la cantidad de bits por segundo enviados sobre un enlace de datos mediante el reemplazo por código electrónico de los caracteres, hileras y secuencias de órdenes que se repiten con frecuencia. Cuando estos datos llegan al otro extremo remoto del enlace de transmisión, los datos codificados son reemplazados con los datos reales.

Configuración. Arreglo lógico o físico de estaciones en una red de relación una con otra.

Contraseña. Una palabra o grupo de caracteres que un usuario debe ingresar para tener acceso a una computadora o a archivos.

Correo electrónico. Mensajes enviados electrónicamente entre suscriptores mediante un sistema público o privado de comunicación de datos.

Cursor. Un caracter parpadeante en pantalla que muestra dónde aparecerá el siguiente caracter.

FTP. Protocolo de transmisión de archivos que permite que el usuario transfiera archivos de una ubicación a otra por medio de Internet y HTTP (*Protocolo de transmisión de hipertexto*) el cual distribuye información por medio de World Wide Web.

Gigabyte. Unidad de medición que equivale a unos mil millones de bytes (1 073,741,824). Se usa al establecer una cantidad de memoria o la capacidad de un disco.

Hardware. Parte física de una computadora, incluyendo los componentes eléctricos, electrónicos, electromecánicos y mecánicos.

HTML. Lenguaje de Marcado de Hipertexto (*Hipertext Markup Language*). Lenguaje para generar páginas dentro del Web basado en la teoría del hipertexto (*conjunto de documentos cada uno de los cuales tienen ligas visibles a cada uno de los cuales tienen ligas visibles a cada uno de los demás documentos*).

Icono. Símbolo en pantalla que representa un programa, un archivo de datos u otra entidad o función de la computadora.

Interfase. En hardware, se aplica al límite entre dos unidades a través del cual todas las señales que pasan son cuidadosamente definidas. En cuanto al software, este término hace referencia a las características de la forma empleada para comunicar dos módulos que actúan dentro de un entorno relacionado.

Internet. Llamada la "red de redes" de cobertura mundial que están interconectadas entre sí usando el protocolo IP (*protocolo que permite que un "paquete" viaje a través de múltiples redes hasta alcanzar su destino*).

Kilobyte. Unidad básica de medida para la memoria de las computadoras y la capacidad de los discos; equivale a 1,024 bytes.

Mainframe. Computadora multiusuario concebida para cubrir los requerimientos de computación de grandes empresas.

Megabyte. Medida de capacidad de almacenamiento equivalente a aproximadamente un millón de bytes (1 048,576 bytes)

Menú. Despliegue en pantalla que enlista las opciones de comandos disponibles.

Microprocesador. Un circuito integrado electrónico, generalmente un paquete de un solo chip, capaz de recibir y ejecutar instrucciones codificadas.

Multimedia. Tecnología que utiliza varios medios de comunicación (*audio, video, texto, etc.*) en una presentación o programa.

OLE. Vinculación e incrustación de objetos (*Objet linking and embedding*), conjunto de estándares, desarrollados por Microsoft e incorporados a Microsoft Windows que permite crear vinculaciones dinámicas y de actualización automática entre documentos, así como incrustar un documento creado en cierta aplicación en otro documento creado en otra aplicación. La vinculación es útil cuando se desea la versión original de un archivo que pueda incluir en muchos otros documentos y aplicaciones como objetos vinculados. OLE también soporta la incrustación. Al crear un objeto incrustado, lo que en realidad se hace es colocar una copia independiente y completamente editable del documento fuente en el archivo destino, lo que da por resultado un archivo compuesto.

Protocolo. Conjunto de reglas utilizadas para el intercambio de información entre entidades que colaboran. Por regla general, este acuerdo comprende la cantidad de información que se ha de enviar, la frecuencia con la que se envía, la firma de recuperación de los errores de transmisión y quien va a recibir la información.

Ratón (Mouse). Dispositivo de entrada que cuenta con uno o más botones de control; a medida que se mueve el ratón, sus circuitos transmiten señales que correspondientemente mueven un puntero en la pantalla.

Red Local. Red de comunicación que enlaza varias estaciones en la misma área local. Las redes locales tienen tasas muy bajas de errores y pueden emplear protocolos simplificados de comunicación de datos.

Reingeniería. Rediseño de la forma en que se realiza el trabajo, para después seleccionar las herramientas de cómputo que mejoren el proceso del trabajo rediseñado.

Sistema Operativo. Conjunto de programas que controlan los recursos del sistema y los procesos que los utilizan.

Software. Término genérico que se aplica a los componentes de un sistema informático que no son tangibles o físicos, también llamados programas.

SQL. Lenguaje de Consulta Estructurado (*Structured Query Language*) desarrollado por IBM y de uso extendido en sistemas de macrocomputadoras y minicomputadoras.

Visual Basic. Lenguaje de programación de alto nivel para el desarrollo de aplicaciones diseñadas para ejecutarse bajo Microsoft Windows 95.

Word Wide Web. Sistema para explorar Internet mediante hipervínculos. Los campos hipervínculos contienen texto o combinaciones de texto y número almacenados como texto que se utilizan para las direcciones de los hipervínculos.

BIBLIOGRAFIA

- **Diseño de Sistemas de Información: Teoría y Practica**
Burch, John G. Y Grunitzki, Gray
Limusa, 1992
- **Ingeniería de Software de Gestión, Análisis y diseño de aplicaciones.**
Antonio de Amescua, García Sánchez, Martínez Fernández, Díaz Pérez
Paraninfo
- **Elementos y Herramientas en el Desarrollo de Sistemas de Información. Una visión actual de la tecnología CASE.**
Mario G. Diattini, Sunil N. Daryanani.
Ra-Ma.
- **Ingeniería de Software Aplicada.**
Norris Rigby.
Megabyte Noriega Editores.
- **Análisis y Diseño de Sistemas de Información.**
James A. Senn
1992.
- **Ingeniería del Software. Un Enfoque Practico.**
Roger S. Pressman.
Mcgraw-Hill, 1993.

- **Sistemas de Información: Análisis, Diseño, Puesta a Punto.**
Henry C. Lucas Jr.
Paraninfo, 1987.
 - **Técnicas de Base de Datos. Estructuración en Diseño y Administración.**
Shakuntala Atre.
Trillas, 1988.
 - **Introducción a los Sistemas de Base de Datos.**
C. J. Date.
Addison Wesley Iberoamericana
 - **Object – Oriented Databases a Semantic Data Model Approach.**
Gray, P. M. D. & Krishmarao, G. K. & Paton, N.W..
Prentice Hall International Serier and Computer Science, Herfordshire UK, 1992.
 - **Microsoft ACCESS para Windows a su Alcance.**
Sanchez Navarro, José Daniel.
Mcgraw-Hill, 1994.
 - **Essentials of Management Information Systems.**
Laudon, Kenneth C. Y Laudon, Jane P.
Prentice- Hall, 1995.
 - **Sistemas de Información Administrativa.**
Murdick, Robert G.
Prentice-Hall, 1996.
-