

201

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO



FACULTAD DE INGENIERIA

SISTEMA DE CONTROL ACADEMICO PARA EL DEPARTAMENTO DE SISTEMAS DE LA D.E.P.F.I.

T E S I S
QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A :
BONIFAZ GOMEZ, MARTHA EUGENIA

DIRECTOR DE TESIS: ING. MARTIN PEREZ MONDRAGON



MEXICO, D. F.

1998

TESIS CON FALLA DE ORIGEN

259292



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está 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.

AGRADECIMIENTOS

A Dios:

Por darme la fuerza y el valor necesarios para ir cumpliendo todas mis metas.

A mis padres:

Adrián y Martha con todo mi amor, porque gracias a ellos y por ellos, he logrado llegar a este gran momento.

A mis hermanos:

Mercedes, Francisco y Adrián, por estar siempre al pendiente de mí.

A mis amigos:

Que realmente sería difícil mencionar a todos y cada uno de ellos, gracias por su apoyo y sus palabras de aliento.

Al Ing. Martín Pérez Mondragón:

Por su confianza y sus consejos.

Al Laboratorio de Sistemas de Información:

Por haber sido como una segunda casa y brindarme el apoyo para la realización de este trabajo.

Martha.

ÍNDICE

	Página
Introducción	i
 CAPÍTULO I SISTEMAS DE INFORMACIÓN	
I.1 Introducción	1
I.2 Definiciones y objetivos	2
I.3 Sistemas de Información	4
I.4 Componentes estructurales de los Sistemas de Información	5
I.5 Categorías de los sistemas de información	7
I.5.1 Sistemas para el procesamiento de transacciones (TPS)	7
I.5.2 Sistemas administradores de información (MIS)	7
I.5.3 Sistemas para el soporte de decisiones (DSS)	10
I.5.4 Sistemas Expertos	10
I.6 Estrategias para el desarrollo de sistemas	11
I.6.1 Método del ciclo de vida para el desarrollo de sistemas	12
I.6.2 Método del desarrollo del análisis estructurado	15
I.6.3 Método del prototipo	18
I.6.4 ¿Qué método de desarrollo es el más apropiado?	19
 CAPÍTULO II BASES DE DATOS	
II.1 Conceptos básicos de Bases de Datos	20
II.2 Sistemas Manejadores de Bases de Datos (DBMS)	21
II.3 Modelos de Manejadores de Bases de Datos	23
II.3.1 Modelo Manejador de Archivos	24
II.3.2 Modelo Jerárquico	24
II.3.3 Modelo de Red	26
II.3.4 Modelo Relacional	27
II.3.5 Bases de Datos Orientadas a Objetos	33
II.4 Características de las bases de datos relacionales.	43
II.4.1 El proceso de normalización	48
II.4.2 Formas Normales	50
II.5 ¿Porqué utilizar Bases de Datos?	53

CAPÍTULO III ANÁLISIS Y DETERMINACIÓN DE REQUERIMIENTOS

III.1	Objetivo general	56
III.2	Antecedentes	56
III.3	Definición del problema	60
III.4	Requerimientos del usuario	62
III.5	Propuesta de solución	65
III.5.1	Elección de la base de datos	65
III.6	Especificaciones de diseño	66
III.7	Diagrama de flujo de datos	70
III.8	Diseño estructurado	74

CAPÍTULO IV DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

IV.1	Modelo Entidad-Relación	80
IV.2	Diccionario de datos	89
IV.3	Estructura del sistema	92

CAPÍTULO V INTEGRACIÓN Y PRUEBAS DEL SISTEMA

V.1	Pruebas	97
-----	---------	----

CONCLUSIONES	104
---------------------	-----

APÉNDICES

Apéndice A	
Bases de datos Access	A1
Apéndice B	
Manual de usuario	B1
Apéndice C	
Catálogos	C1
Apéndice D	
Código Access Basic	D1

GLOSARIO

BIBLIOGRAFÍA

REFERENCIAS

INTRODUCCIÓN

La característica dominante de nuestra sociedad es la explosión de la información basada en la tecnología. La computadora y las telecomunicaciones están cambiando totalmente la forma en la que trabajamos y vivimos por lo que resulta sumamente difícil predecir aún a corto plazo la naturaleza de la sociedad caracterizada por la información, pues los cambios son muy rápidos. Esta explosión de información ha llegado a todos lugares y la División de Estudios de Posgrado de la Facultad de Ingeniería (DEPFI), particularmente el Departamento de Ingeniería de Sistemas no fue la excepción. Por lo que esta dependencia decide sistematizar la administración académica, ya que el procedimiento que venía operando, no satisfacía *todas las necesidades*. Este problema fue más evidente al verse incrementado considerablemente el alumnado de las áreas o campos que comprenden el Departamento de Ingeniería de Sistemas. Por este motivo se requiere de un Sistema de Información que permita llevar un control de los procesos que se realizan durante el ciclo escolar, desde las inscripciones hasta el término de un ciclo escolar a nivel maestría. Es importante mencionar que este Departamento requiere de un sistema eficiente, fácil de operar, un bajo costo de mantenimiento, con amplias posibilidades de crecimiento para satisfacer necesidades futuras.

La presente tesis: "Sistema de Control Académico para el Departamento de Sistemas de la DEPFI" (SICAP) es el resultado de un esfuerzo conjunto para resolver la problemática existente dentro del Departamento, en cuanto al manejo, control y distribución de la información del alumnado y profesorado.

El contenido de este trabajo se encuentra dividido en cinco capítulos:

Capítulo I SISTEMAS DE INFORMACIÓN

En este capítulo se tratan los conceptos fundamentales de los Sistemas de Información, son sus componentes estructurales, categorías y sus principales características. También se hace referencia a las estrategias para el desarrollo de sistemas como lo son el Método del ciclo de vida para el desarrollo de sistemas, Método del desarrollo del análisis estructurado y el Método del prototipo.

Capítulo II BASES DE DATOS

En esta sección se presentan conceptos teóricos sobre las bases de datos, así como los diferentes Modelos de Manejadores de Bases de Datos y sobre todo las características relacionales. Como se observa, estos primeros capítulos proporcionan los antecedentes necesarios para el desarrollo de este trabajo de tesis.

Capítulo III ANÁLISIS Y DETERMINACIÓN DE REQUERIMIENTOS

Aquí se señalan los objetivos que el sistema debe cumplir a fin de satisfacer las necesidades propias del Departamento.

Capítulo IV DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

Este capítulo constituye la parte medular de este trabajo, ya que aquí se plantea el diseño del sistema, así como los módulos que lo integrarán.

Capítulo V DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

Este capítulo describe una serie de pruebas realizadas al sistema, con el fin de demostrar que se cubrieron los objetivos planteados al inicio de esta tesis.

Apéndice A Bases de Datos Access

Este apéndice contiene información esencial acerca del manejar de bases de datos Microsoft Access.

Apéndice B Bases de Datos Access

Aquí se proporciona el Manual de Usuario del sistema que será de gran utilidad para el buen manejo del mismo.

Apéndice C Catálogos y formatos

Se presentan los catálogos que se utilizaron a fin de optimizar el funcionamiento del sistema y además, se muestran los formatos que los alumnos de la DEPEI deben llenar a fin de poder ingresar a una maestría del área de Sistemas.

Apéndice D Código Access Basic

Se muestra el código generado para la aplicación.

**C
a
p
í
t
u
l
o**

Sistemas de Información

I

1.1 Introducción

El incremento de los sistemas de información día con día es mayor, los directivos de las empresas u organizaciones se han percatado de la necesidad de controlar la información que arrojan sus procesos, tanto administrativos, contables y de producción, los cuales les permitirán planear y gestionar. De lo antes mencionado, se deriva la importancia de conocer más a fondo los conceptos y técnicas empleadas para el análisis y diseño de los mismos.

Dentro de una organización, el análisis y diseño de sistemas se refiere al proceso de examinar la situación de una empresa con el propósito de mejorarla con métodos y procedimientos más adecuados.

El desarrollo de sistemas puede considerarse, en general, formado por dos grandes componentes: *el análisis de sistemas* y *el diseño de sistemas*.¹ El análisis de sistemas, es el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de la información para recomendar mejoras al sistema. El diseño de sistemas es el proceso de planificar, reemplazar o complementar un sistema organizacional existente. Es necesario comprender cómo trabaja el sistema actual y, de manera más específica, cuál es el flujo de información en todo el sistema. Sólo después de haber reunido todos los hechos, nos encontraremos en la posición de determinar cómo y dónde un sistema de información basado en computadora será benéfico para los usuarios del sistema.

El análisis especifica *qué* es lo que el sistema debe hacer. El diseño establece *cómo* alcanzar el objetivo.

¹ Análisis y Diseño de Sistemas de Información

1.2 Definiciones y objetivos.

Antes de continuar, es necesario responder a ciertas interrogantes y definir conceptos que más adelante serán de gran utilidad.

Datos e Información.

No es posible contar con sistemas grandes sin sistemas de comunicación que unan sus componentes. En los extensos sistemas de las empresas, los gerentes han de recibir información para poder dirigirlos y controlarlos. En muchas ocasiones se confunden las masas de datos con la información. Es importante distinguir entre ambas cosas para desarrollar un sistema de información.

La información es un signo o conjunto de signos que impulsan a la acción. Se distingue de los datos porque éstos no son estímulos de la acción, si no simplemente cadenas de caracteres o patrones sin interpretar.

¿Qué es un Sistema?

En el sentido más amplio, un *sistema* es un conjunto de componentes que interactúan entre sí para lograr un objetivo común². Una organización es un sistema, sus componentes trabajan juntos para crear utilidades que beneficien tanto a los empleados como a los accionistas de la compañía. Cada uno de estos componentes es a su vez un sistema.

Para que un sistema funcione de manera apropiada, debe involucrar los siguientes principios fundamentales:

- **Confiabilidad.** Se refiere al grado de seguridad con que un sistema realiza su función, produciendo los mismos resultados en procesos sucesivos.
- **Disponibilidad.** Significa que el sistema es accesible a los usuarios. Un sistema puede ser confiable, pero no estar disponible cuando se está probando o cuando se le agrega un componente.
- **Flexibilidad.** El requerimiento de flexibilidad se refiere a la habilidad del sistema para cambiar o adaptarse para satisfacer los requerimientos cambiantes del ambiente.
- **Expectativa de vida y potencial de crecimiento.** Algunos sistemas no cuentan con una expectativa de vida, debido a que ya son obsoletos en el momento en que se implantan. Los sistemas deben diseñarse para satisfacer requerimientos durante un tiempo predeterminado.
- **Capacidad para recibir mantenimiento.** Una vez que el sistema se implanta, éste debe tener la capacidad de permitir mantenimiento en sus principales facetas como pueden ser la prevención o corrección; y se tenga por lo tanto continuidad en el funcionamiento del mismo.

² Análisis y Diseño de Sistemas de Información

Todo sistema debe cumplir con una estructura básica tal y como se ilustra en la figura I.1.

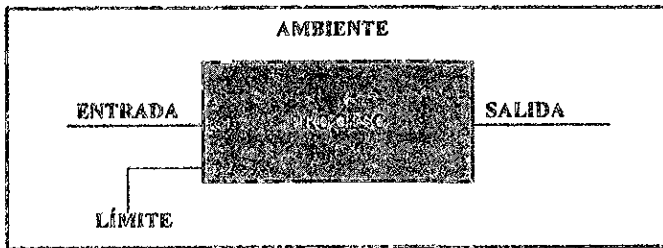


Figura I.1

En la figura anterior se observa que un sistema está relacionado con una serie de actividades que permiten transformar una entrada en una salida a través de un proceso determinado.

Existen diversos elementos que conforman un sistema, como son:

- **Ambiente.** Es el conjunto de elementos a través de los cuales un sistema se comunica con el universo en que está inmerso. Es todo lo que se encuentra fuera del sistema.
- **Entradas.** Son los recursos (datos, materias, etc.) que son consumidos y manipulados dentro del sistema.
- **Salida.** Es el producto que se obtiene como resultado del proceso realizado por el sistema.
- **Límite.** Es la delimitación en la cual los elementos del sistema están incluidos. Se asume que los elementos que están dentro de este límite, son cambiados y controlados con mayor facilidad que los que están fuera.

La finalidad de un sistema es la razón de su existencia. Para alcanzar sus objetivos, los sistemas interactúan con su medio ambiente, el cual está formado por todos los objetos que se encuentran fuera de las fronteras de los sistemas. Los sistemas que interactúan con su medio ambiente (reciben entradas y producen salidas) se denominan *sistemas abiertos*. En contraste, aquellos que no interactúan con su medio ambiente se conocen como *sistemas cerrados*. Todos los sistemas actuales son abiertos. Es así como los sistemas cerrados existen sólo como un concepto. El elemento de *control* está relacionado con la naturaleza de los sistemas, sean cerrados o abiertos. Por otra parte, los sistemas trabajan mejor cuando operan dentro de niveles de desempeño tolerables. Todos los sistemas tienen niveles aceptables de desempeño, denominados *estándares* y contra los que se comparan los niveles de desempeño actuales. La información proporcionada al comparar los resultados con los estándares recibe el nombre de *retroalimentación*.

Es decir, los sistemas emplean un modelo de control básico consistente en:

- 1.- Un estándar para lograr un desempeño aceptable
- 2.- Un método para medir el desempeño actual
- 3.- Un método para comparar el desempeño actual contra el estándar
- 4.- Un método de retroalimentación

El concepto de interacción con el medio ambiente, que es lo que caracteriza a los sistemas abiertos, es esencial para el control. Recibir y evaluar la retroalimentación, permite al sistema determinar qué tan bien está operando.

Los componentes que forman un sistema pueden ser a su vez sistemas más pequeños; es decir, los sistemas pueden estar formados por varios niveles de sistemas o subsistemas, es común tener varios niveles de sistemas interactuando entre sí.

1.3 Sistemas de información

Todo sistema organizacional depende, en mayor o menor medida, de una entidad abstracta denominada *sistema de información*. Este sistema es el medio por el cual los datos fluyen de una persona o departamento hacia otros.

Un sistema de información se puede definir técnicamente como un conjunto de componentes interrelacionados que recupera, procesa, almacena y distribuye información para tomar decisiones y tener el control de una organización³. Los sistemas de información proporcionan servicio a todos los demás sistemas de una organización y enlazan todos sus componentes en forma tal que éstos trabajen con eficiencia para alcanzar el mismo objetivo.

Los sistemas de información deben contener información representativa acerca de los usuarios, lugares y cosas que están dentro de la organización o del medio ambiente que lo rodea.

La finalidad de los sistemas de información, como las de cualquier otro sistema dentro de una organización, es procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas. Los sistemas de información están formados por subsistemas que incluyen hardware, software, medios de almacenamiento de datos y bases de datos. El conjunto particular de subsistemas utilizados -equipo específico, programas, archivos y procedimientos- es lo que se denomina una *aplicación* de sistemas de información.

Las actividades básicas de un sistema de información con las que resulta posible producir la información necesaria para una organización, son: entradas, procesamiento y salidas. Figura 1.2.

³ Sistemas de Información Administrativa

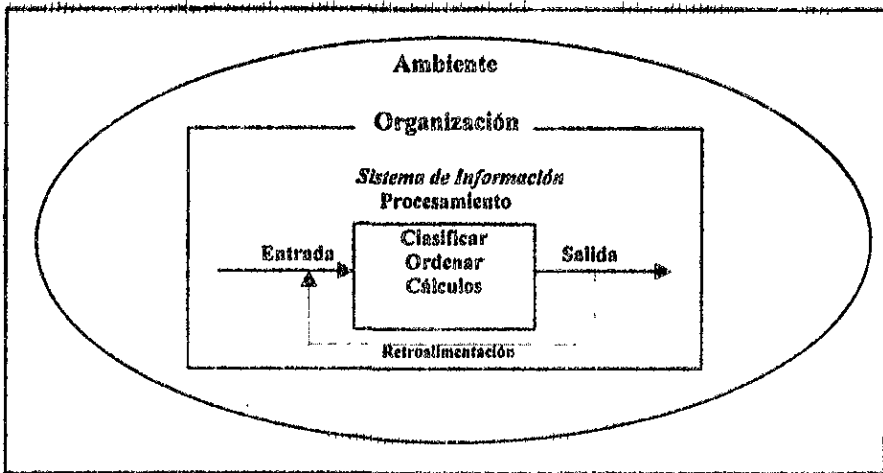


Figura 1.2

1.4 Componentes estructurales de los sistemas de información

Sin importar las organizaciones a las que sirven o la forma en que se desarrollan y diseñan, todos los sistemas de información están compuestos de los siguientes componentes estructurales: entrada, modelos, salida, tecnología, base de datos y controles. Estos componentes estructurales pueden tomar diferentes formas, valores y contenido; pueden parecer diferentes y trabajar en forma diferente; algunos pueden soportar sistemas bien diseñados; algunos pueden soportar sistemas diseñados con deficiencia; algunos pueden ser imperfectos, algunos pueden ser altamente sofisticados, todo esto es irrelevante. Estos son los seis componentes estructurales básicos de todos los sistemas de información. Estos componentes pueden conjuntarse para obtener sistemas de información funcionales que satisfagan las necesidades de las organizaciones y de sus usuarios.

- **Bloque de entrada**

La entrada representa a todos los datos, texto, voz e imágenes que entran al sistema de información y los métodos y los medios por los cuales se capturan e introducen. La entrada está compuesta de transacciones, solicitudes, consultas, instrucciones y mensajes. Por lo general, la entrada sigue un protocolo y un formato para que el contenido, la identificación, la autorización, el arreglo y el procesamiento sean adecuados. La introducción puede hacerse mediante escritura manual, formas en papel, reconocimiento de características físicas como geometría manual o huellas digitales, teclados, ratones, voz, sensores táctiles, y caracteres y códigos ópticos y magnéticos.

En la actualidad, los medios más comunes para la introducción de transacciones y texto son las lectoras de códigos de barras y láser y el teclado, respectivamente. También

existe una diversidad de dispositivos con entrada eficiente, como las pantallas sensibles al tacto que responden directamente a la presión de los dedos.

- *Bloque de modelos*

Este componente consta de modelos lógico-matemáticos que manipulan de diversas formas la entrada y los datos almacenados, para producir los resultados deseados o salida.

- *Bloque de salida*

El producto del sistema de información es la salida de información de calidad y documentos para todos los niveles de la gerencia y para todos los usuarios dentro y fuera de la organización. La salida es, en gran medida, el componente que guía e influye en los otros componentes. Si el diseño de este componente no satisface las necesidades del usuario, entonces los otros componentes tienen poca importancia. La salida representa en otro extremo de la entrada y obviamente no puede ser mejor que la entrada y los modelos empleados para producirla.

Con frecuencia, la entrada y la salida son interactivas. La entrada se convierte en salida; la salida se convierte en entrada. De manera lógica, la salida está compuesta de elementos tales como estados financieros, facturas, órdenes de compra, reportes de presupuestos, respuestas a consultas, mensajes, órdenes, etc. La calidad de esta salida se basa en su exactitud, oportunidad y relevancia. Además, esta salida debe tratarse en función de su destino, frecuencia de uso y seguridad.

- *Bloque de tecnología*

La tecnología es la "caja de herramientas" del trabajo en sistemas de información. Captura la entrada, activa los modelos, almacena e ingresa datos, produce y transmite salida, y ayuda a controlar todo el sistema, realiza todo el trabajo pesado y une a todos los componentes estructurales. La tecnología consta de tres componentes principales; la computadora y el almacenamiento auxiliar, las telecomunicaciones y el software. Las telecomunicaciones comprenden el empleo de medios electrónicos y de transmisión de luz para la comunicación entre nodos a lo largo de una distancia. El software corresponde a los programas que hace que funcione el hardware de la computadora y le dan instrucciones sobre la forma de procesar los modelos. El hardware está compuesto de una variedad de dispositivos que proporcionan el soporte físico para los componentes estructurales. En su esencia misma, la tecnología es un sustituto del esfuerzo humano. De los seis componentes estructurales, la tecnología es el más evidente. La mayoría de los sistemas de información actuales y del futuro estarán basados en la tecnología.

- *Bloque de base de datos*

La base de datos es el lugar en donde se almacenan todos los datos necesarios para atender a las necesidades de todos los usuarios. Nuevamente, los datos pueden ser una combinación de voz, imágenes, texto y números. La base de datos se considera desde dos puntos de vista, el físico y el lógico. La base de datos física está compuesta de los medios

de almacenamiento, como los disquetes, cintas, discos, etc. Ésta es la forma en que los datos se almacenan realmente. Sin embargo, otro problema probablemente más importante es cómo buscar, asociar y recuperar los datos almacenados para satisfacer necesidades específicas de información. Esto, por supuesto, es el lado lógico de la base de datos y, si está estructurada correctamente, asegura la recuperación oportuna, relevante y exacta de la información.

- **Bloque de controles**

Todos los sistemas de información están sujetos a una diversidad de peligros y amenazas, como incendios, fallas en los sistemas, errores y omisiones, deficiencias y mutilaciones maliciosas. En muchos casos, sin embargo, los peores abusos del sistema provienen de procedimientos operacionales inadecuados. Algunos de los controles que necesitan diseñarse en el sistema para asegurar su protección, integridad y operación uniforme son la instalación de un sistema de administración de registros, la creación de un plan de contingencias, la preparación de una documentación completa y actualizada, el establecimiento de sistemas de respaldo, la aplicación de una diversidad de procedimientos de seguridad, dispositivos y controles de acceso.

1.5 Categorías de los sistemas de información

Los sistemas de información se desarrollan con diferentes propósitos, los cuales dependen de las diversas necesidades de una empresa. Los diferentes tipos de sistemas de información computarizados se analizan y diseñan bajo los mismos conceptos y técnicas del diseño y análisis de sistemas.

1.5.1 Sistemas de procesamiento de datos. Los sistemas de procesamiento de datos son aquellos sistemas de información computarizados que se desarrollan para procesar grandes volúmenes de información generada de las funciones administrativas, tales como la nómina o el control de inventarios. Los sistemas de procesamiento de datos liberan del tedio y la rutina a las tareas que se realizan manualmente; sin embargo, el elemento humano sigue participando, al llevar a cabo la captura de la información requerida.

Tales sistemas ejecutan periódicamente los programas de manera automática. Una vez preparados, escasamente se requiere el tomar decisiones. En términos generales, los sistemas de procesamiento de datos ejecutan las actividades de carácter rutinario de las empresas.

1.5.2 Sistemas administradores de Información. Los sistemas administradores de información, conocidos como MIS ("*Management Information Systems*") no sustituyen a los sistemas de procesamiento de datos, más bien todos toman en cuenta a las funciones de procesamiento de datos. Los MIS son sistemas que se sustentan en la relación que surge entre las personas y las computadoras. Los MIS requieren para su operación de: las personas, del software (programas de cómputo) y del hardware (computadoras, impresoras, etc.) Estos sistemas soportan un amplio espectro de tareas de las organizaciones, más aún

que los sistemas de procesamiento de datos, incluyendo el análisis, decisiones y la toma de decisiones.

En la figura I.3 se ilustran los elementos básicos de un MIS que son: La administración, la información y los sistemas.

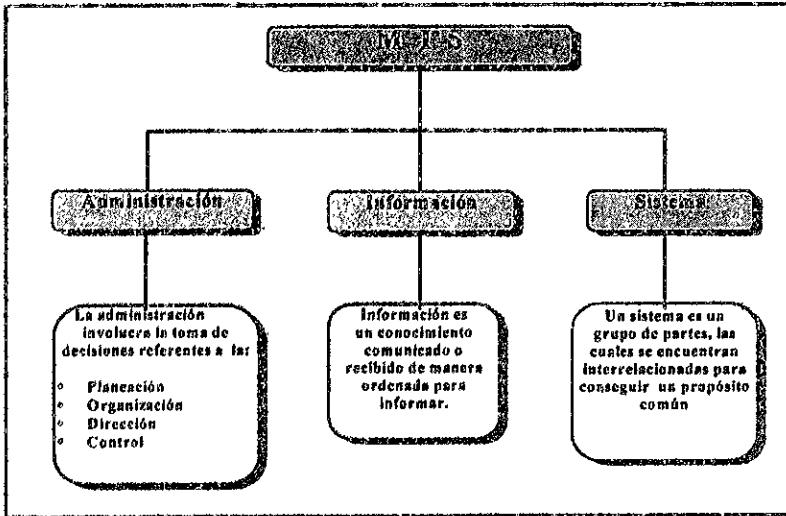


Figura I.3

El elemento central es la información ya que el objetivo de los MIS es el manejo organizado de la misma. La administración también está presente, ya que mediante el proceso administrativo (planeación, organización, dirección y control) se logrará la eficiencia en el manejo de la información y por último se menciona a los sistemas que son los que interrelacionan los elementos para el manejo adecuado de la información.

Por otra parte, se puede observar que las actividades determinan los requerimientos para los datos y su subsecuente organización, transformación y análisis (Figura I.4). La demanda de flujo de información ocurre cuando los administradores requieren información para llevar a cabo la toma de decisiones. Si dicha información se encuentra disponible en los archivos de la empresa o en las computadoras, entonces ellos únicamente necesitan accederla o recuperarla para hacerla utilizable. En caso contrario, es decir, que no se encuentre disponible, se tiene que recurrir a la recolección de la información a través de las diferentes fuentes de datos, las cuales pueden ser internas o externas a la empresa.

Un MIS está formado por:

• Componentes 

• Actividades 

Flujos de Información

 Flujo de suministros

 Flujo de demandas

 Límite del sistema

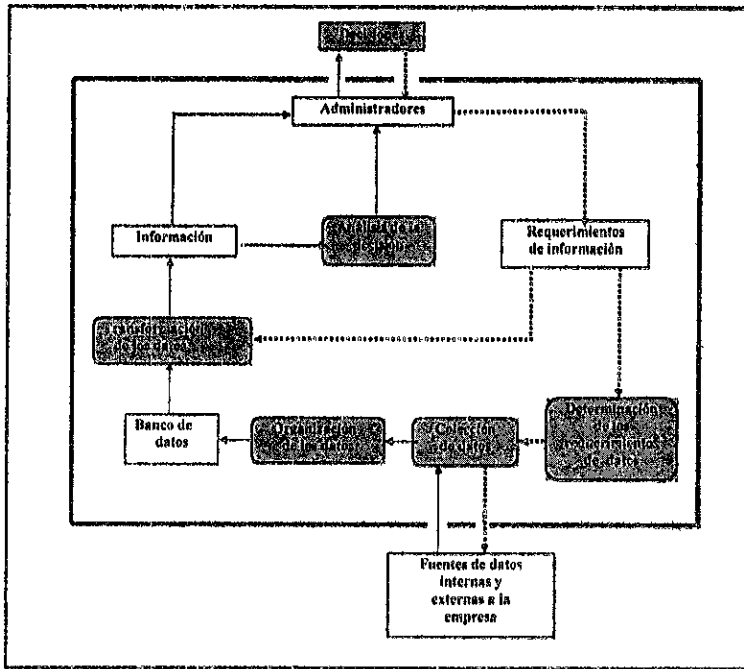


Figura 1.4

Los Sistemas Administradores de Información tienen una serie de objetivos cuya finalidad es satisfacer las necesidades de la organización y algunos de estos objetivos son:

- Ofrecer información que contribuya al logro de los objetivos generales de la organización.

- Agilizar el proceso de toma de decisiones a través de la obtención o entrega de información confiable.
- Automatización de las operaciones dentro de la organización.
- Permitir al administrador la pronta detección de soluciones asociadas a su organización.

1.5.3 Sistemas para el soporte de decisiones. Los sistemas para el soporte de decisiones es un tercer tipo de sistema de información computarizada (DSS). Este sistema es similar a los sistemas de información administrativa, en el sentido de que ambos dependen de una base de datos como una fuente de información; pero se distingue de los MIS, al hacer énfasis en el soporte en cada una de las etapas de la toma de decisiones. Sin embargo, la decisión en sí, depende de la persona responsable de la misma. Los sistemas para el soporte de decisiones se diseñan con una orientación hacia la persona o el grupo que los utilizará.

1.5.4 Sistemas expertos. Los sistemas expertos son en sí, un tipo especial de sistemas de información, que tienen un uso práctico en los negocios debido a la reciente y amplia disponibilidad de hardware y de software. Un sistema experto captura; y en efecto utiliza, el conocimiento de un experto, para la solución de un problema particular de la organización. Hay que hacer notar que, a diferencia del sistema para el soporte de decisiones, que finalmente deja al responsable tomar las decisiones, un sistema experto selecciona la mejor solución al problema.

En la figura 1.5 se presentan las diferencias entre los sistemas de procesamiento de datos, los MIS, los DSS y los sistemas expertos.

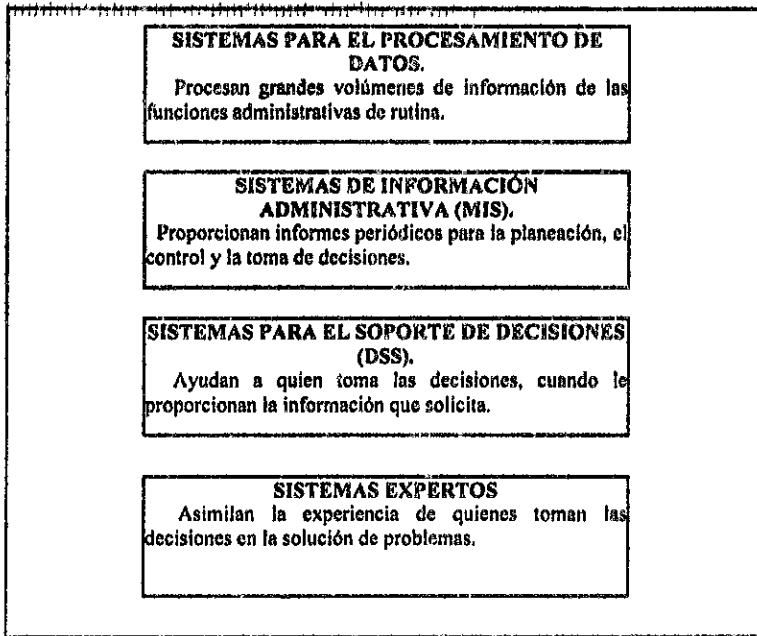


Figura 1.5

1.6 Estrategias para el desarrollo de los sistemas

Los sistemas de información basados en computadora sirven para diversas finalidades que van desde el procesamiento de las transacciones de una empresa, hasta proveer de la información necesaria para decidir sobre asuntos que se presentan con frecuencia. En algunos casos los factores que deben considerarse en un proyecto de sistemas de información, tales como el aspecto más apropiado de la computadora o la tecnología de comunicaciones que se va a utilizar, el impacto del nuevo sistema sobre los empleados de la empresa y las características específicas que el sistema debe tener, se pueden determinar de una manera secuencial. Existen entonces tres diferentes enfoques al desarrollo de sistemas de información basados en computadora ⁴:

- ◆ Método del ciclo de vida para el desarrollo de sistemas
- ◆ Método del desarrollo del análisis estructurado
- ◆ Método del prototipo de sistemas

⁴ Análisis y Diseño de Sistemas de Información

1.6.1 Ciclo de vida clásico del desarrollo de sistemas.

El desarrollo de sistemas, un proceso formado por las etapas de análisis y diseño, comienza cuando la administración o algunos miembros del personal encargado de desarrollar sistemas, detectan un sistema de la empresa que necesita mejoras. *El método del ciclo de vida para desarrollo de sistemas (SDLC)* es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información. En la mayor parte de las situaciones dentro de una empresa todas las actividades están muy relacionadas, en general son inseparables, y quizá sea difícil determinar el orden de los pasos que se siguen para efectuarlas. Las diversas partes del proyecto pueden encontrarse al mismo tiempo en distintas fases de desarrollo; algunos componentes en la fase de análisis mientras que otros en etapas avanzadas de diseño.

El método del ciclo de vida para desarrollo de sistemas consta de las siguientes actividades:

- 1.- Investigación preliminar
- 2.- Determinación de los requerimientos del sistema
- 3.- Diseño del sistema
- 4.- Desarrollo de software
- 5.- Prueba de los sistemas
- 6.- Implantación y evaluación

1.- Investigación Preliminar

El proceso se inicia con la petición de una persona -administrador, empleado o especialista en sistemas-. Cuando se formula la solicitud comienza la primera actividad de sistemas, la investigación preliminar. Esta actividad tiene tres partes: aclaración de la solicitud, estudio de factibilidad y aprobación de la solicitud.

- **Aclaración de la solicitud.** Muchas solicitudes que provienen de empleados y usuarios no están formuladas de manera clara. Por consiguiente, antes de considerar cualquier investigación de sistemas, la solicitud de proyecto debe examinarse para determinar con precisión lo que el solicitante desea.
- **Estudio de factibilidad.** Un resultado importante de la investigación preliminar es la determinación de que el sistema solicitado sea factible. En la investigación preliminar existen tres aspectos relacionados con el estudio de factibilidad:

1.- Factibilidad técnica. El trabajo para el proyecto, ¿puede realizarse con el equipo actual, la tecnología existente de software y el personal disponible? Si se necesita nueva tecnología, ¿cuál es la posibilidad de desarrollarla?.

2.- Factibilidad económica. Al crear el sistema, ¿los beneficios que se obtienen serán suficientes para aceptar los costos?, ¿Los costos asociados con la

decisión de no crear el sistema son tan grandes que se debe aceptar el proyecto?

3.- Factibilidad operacional. Si se desarrolla e implanta ¿será utilizado el sistema?, ¿Existirá cierta resistencia al cambio por parte de los usuarios que dé como resultado una disminución de los posibles beneficios de la aplicación?

- **Aprobación de la solicitud.** No todos los proyectos solicitados son deseables o factibles. Algunas organizaciones reciben tantas solicitudes de sus empleados que sólo es posible atender unas cuantas. En algunos casos el desarrollo puede comenzar inmediatamente, cuando esto ocurre, la administración decide qué proyectos son los más importantes y decide el orden en que se llevarán a cabo.

2.-Determinación de los requerimientos del sistema.

El aspecto fundamental del análisis de sistemas es comprender todas las facetas importantes de la parte de la empresa que se encuentra bajo estudio. De esta forma podemos responder a las siguientes preguntas:

- 1.- ¿Qué es lo que se hace?
- 2.- ¿Cómo se hace?
- 3.- ¿Con qué frecuencia se presenta?
- 4.- ¿Qué tan grande es el volumen de transacciones o de decisiones?
- 5.- ¿Cuál es el grado de eficiencia con que se efectúan las tareas?
- 6.- ¿Existe algún problema?
- 7.- Si existe un problema, ¿qué tan serio es?
- 8.- Si existe un problema, ¿cuál es la causa que lo origina?

Para responder estas preguntas, es necesario conversar con varias personas para reunir detalles relacionados con los procesos de la empresa, sus opiniones sobre porqué ocurren las cosas, las soluciones que proponen y sus ideas para cambiar el proceso. Se emplean cuestionarios para obtener esta información cuando no es posible entrevistarse, en forma personal. Conforme se reúnen los detalles, se procede a estudiar los datos sobre requerimientos con la finalidad de identificar las características que debe tener el nuevo sistema, incluyendo la información que deben producir los sistemas junto con características operacionales tales como controles de procesamiento, tiempos de respuesta y métodos de entrada y salida.

3.- Diseño del sistema

El diseño de un sistema de información produce los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase de análisis. Se comienzan el proceso de diseño identificando los reportes y demás salidas que debe producir el sistema. Hecho lo anterior se determinan con toda precisión los datos específicos para cada reporte y salida. Es común realizar un bosquejo del formato o pantalla que esperan que aparezca cuando el sistema esté terminado. El diseño de un sistema también indica los datos de entrada, aquellos que serán calculados y los que deben ser almacenados.

4.- Desarrollo de software

Los encargados de desarrollar software pueden instalar (o modificar y después instalar) software comprado a terceros o escribir programas diseñados a la medida del solicitante. La elección depende del costo de cada alternativa, del tiempo disponible para escribir el software y de la disponibilidad de los programadores. Los programadores también son responsables de la documentación de los programas y de proporcionar una explicación de cómo y porqué ciertos procedimientos se codifican en determinada forma. La documentación es esencial para probar el programa y llevar a cabo el mantenimiento una vez que la aplicación se encuentra instalada.

5.- Prueba del sistema

Durante la fase de prueba de sistemas, el sistema se emplea de manera experimental para asegurarse de que el software no tenga fallas, es decir, que funciona de acuerdo a las especificaciones y en la forma en que los usuarios esperan que lo haga. Se alimentan como entradas conjuntos de datos de prueba para su procesamiento y después se examinan los resultados. En ocasiones se permite que varios usuarios utilicen el sistema para observar si tratan de emplearlo en formas no previstas. En ocasiones, las pruebas son conducidas por personas ajenas al grupo que escribió los programas originales; con esto se persigue asegurar, por una parte, que las pruebas sean completas e imparciales y, por otra, que el software sea más confiable.

6.- Implantación y evaluación

La implantación es el proceso de verificar e instalar nuevo equipo, entrenar a los usuarios, instalar la aplicación y construir todos los archivos de datos necesarios para utilizarla. Una vez instaladas, las aplicaciones se emplean durante muchos años. Sin embargo, las organizaciones y los usuarios cambian con el paso del tiempo, incluso el ambiente es diferente con el paso de las semanas y los meses. Por consiguiente, es indudable que debe darse mantenimiento a las aplicaciones; realizar cambios y modificaciones en el software, archivos o procedimientos para satisfacer las nuevas necesidades de los usuarios. La evaluación de un sistema se lleva a cabo para identificar puntos débiles y fuertes. La evaluación ocurre a lo largo de cualquiera de las siguientes dimensiones:

Evaluación operacional. Valoración de la forma en que funciona el sistema, incluyendo su facilidad de uso, tiempo de respuesta, lo adecuado de los formatos de información, confiabilidad global y nivel de utilización.

Impacto organizacional. Identificación y medición de los beneficios para la organización en áreas tales como finanzas, eficiencia operacional e impacto competitivo.

Opinión de los administradores. Evaluación de las actitudes de directivos y administradores dentro de la organización así como de los usuarios finales.

Desempeño del desarrollo. La evaluación del proceso de desarrollo de acuerdo con los criterios tales como tiempo y esfuerzo de desarrollo, concuerdan con presupuestos y estándares, y otros criterios de administración de proyectos.

1.6.2 Método de desarrollo por análisis estructurado.

Muchos especialistas en sistemas de información reconocen la dificultad de comprender de manera completa sistemas grandes y complejos. El método de desarrollo del análisis estructurado tiene como finalidad superar esta dificultad por medio de 1) la división del sistema en componentes y 2) la construcción de un modelo del sistema. El método incorpora elementos tanto de análisis como de diseño.

¿Qué es el análisis estructurado?

El análisis estructurado se concentra en especificar lo que se requiere que haga el sistema o la aplicación. No se establece cómo se cumplirán los requerimientos o la forma en que implantará la aplicación. Más bien permite que las personas observen los elementos lógicos (*lo que hará el sistema*) separados de los componentes físicos (*computadoras, terminales, sistemas de almacenamiento, etc.*)

Elementos del análisis estructurado.

Los elementos esenciales del análisis estructurado son símbolos gráficos, diagramas de flujo de datos y el diccionario centralizado de datos.

- **Descripción gráfica.**

Una de las formas de describir un sistema es preparar un bosquejo que señale sus características, identifique la función para la que sirve e indique cómo éste interactúa con otros elementos, entre otras cosas. Sin embargo, describir de esta manera un sistema grande es un proceso tedioso y propenso a errores ya que es fácil omitir algún detalle. En lugar de las palabras el análisis estructurado utiliza símbolos, o iconos, para crear un modelo gráfico del sistema. Los modelos de este tipo muestran los detalles del sistema pero sin introducir procesos manuales computarizados, archivos en cinta o discos magnéticos.

Tal como lo indica la figura 1.6, los iconos identifican los elementos básicos de los procesos, el flujo de datos, el sitio donde se almacenan los datos y las fuentes y destinos de éstos. Se dibuja una línea alrededor del sistema para señalar qué elementos se encuentran dentro del sistema y cuáles fuera de su frontera.

El diagrama lógico de flujo de datos muestra las fuentes y destinos de los datos, identifica y da nombre a los grupos de datos que relacionan una función con otra y señala los almacenes de datos a los que se tiene acceso.

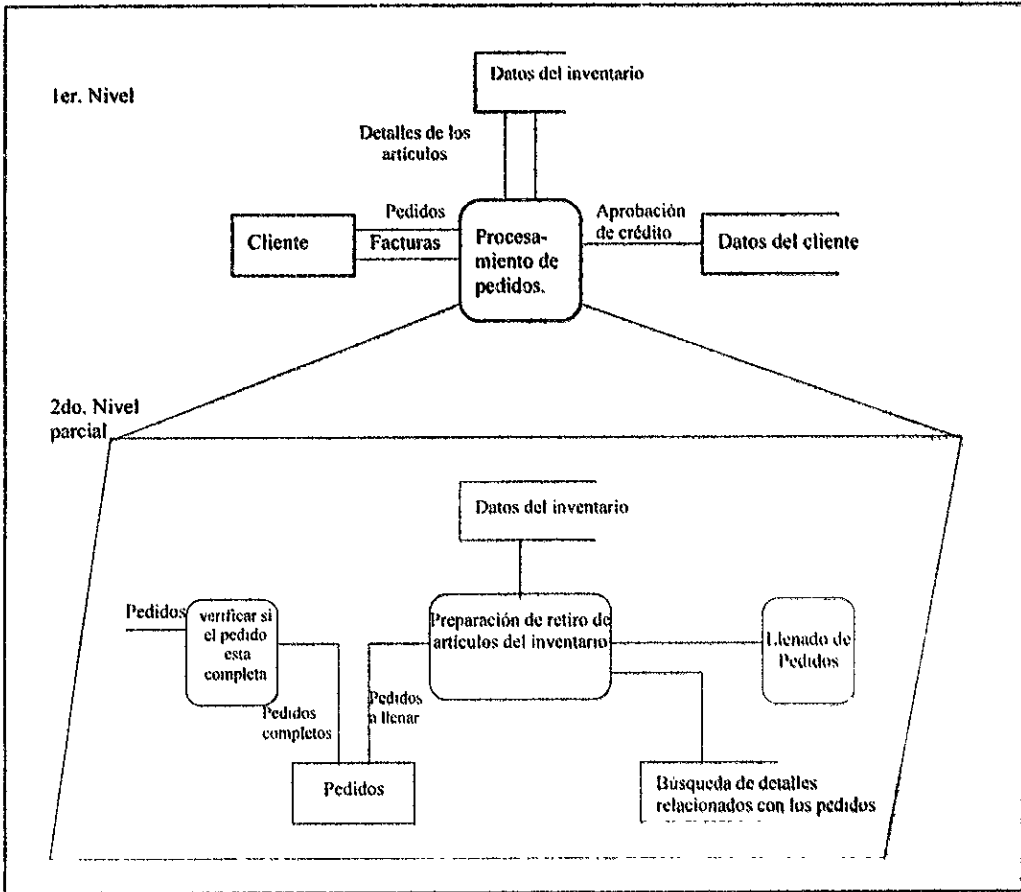


Figura 1.6 Diagrama de flujo de datos utilizados con el método de análisis estructurado

• **Diagrama de flujo de datos.**

El modelo del sistema recibe el nombre de diagrama de flujo de datos (**DFD**). La descripción completa de un sistema está formado por un conjunto de diagramas de flujos de datos. Para desarrollar una descripción del sistema por el método de análisis estructurado se sigue un proceso descendente (**top-down**). El modelo original se detalla en diagramas de bajo nivel que muestran características adicionales del sistema. Cada proceso puede desglosarse en diagramas de flujo de datos cada vez más detallados. Esta secuencia se repite hasta que se obtienen suficientes detalles que permiten comprender en su totalidad la parte del sistema que se encuentra bajo investigación.

• **Diccionario de datos.**

Todas las definiciones de los elementos en el sistema -flujo de datos, procesos y almacenes de datos- están descritos en forma detallada en el diccionario de datos. Si alguien desea saber alguna definición del nombre de un dato o el contenido particular de un flujo de datos, esta información debe encontrarse disponible en el diccionario de datos.

¿Qué es el diseño estructurado?

El diseño estructurado, otro elemento del análisis estructurado que emplea la descripción gráfica, se enfoca en el desarrollo de especificaciones del software. La meta del diseño estructurado es crear programas formados por módulos independientes unos de otros desde el punto de vista funcional. El diseño estructurado es una técnica específica para el diseño de programas y no un método de diseño de comprensión. Es decir, no indica nada relacionado con el diseño de archivos o bases de datos, la presentación de entradas o salidas, la secuencia de procesamiento o el hardware que dará soporte a la aplicación. Esta técnica conduce a la especificación de módulos de programa que son funcionalmente independientes. La herramienta fundamental del diseño estructurado es el diagrama estructurado (figura 1.7). Al igual que los diagramas de flujo de datos, los diagramas estructurados son de naturaleza gráfica y evitando cualquier referencia relacionada con el hardware o detalles físicos. Su finalidad no es mostrar la lógica de los programas (que es la tarea de los diagramas de flujo). Los diagramas estructurados describen la interacción entre módulos independientes junto con los datos que un módulo pasa a otro cuando interactúa con él.

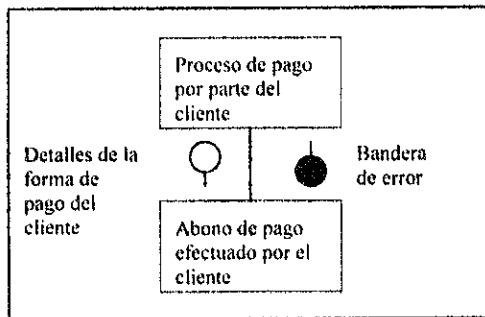


Figura 1.7

La gráfica de estructura muestra la relación entre dos módulos, el que invoca y el que es invocado, junto con los datos que pasan entre ellos. Las flechas con la circunferencia indican datos mientras que las que tienen un círculo representan información de control del programa tales como notas o condiciones de error.

1.6.3 Método del prototipo de sistemas

Este método hace que el usuario participe de manera más directa en la experiencia de análisis y diseño que cualquiera de los ya presentados. La construcción de prototipos es muy eficaz en las circunstancias correctas.

¿Qué es un prototipo?

El prototipo es un sistema que funciona - no solo una idea en el papel-, desarrollado con la finalidad de probar ideas y suposiciones relacionadas con el nuevo sistema. Está constituido por software que acepta entradas, realiza cálculos, produce información ya sea impresa o presentada en pantalla. Es la primera versión, o iteración, de un sistema de información; es el modelo original.

Los usuarios evalúan el diseño y la información generada por el sistema. Lo anterior sólo puede hacerse con efectividad si los datos utilizados, al igual que las situaciones, son reales. Por otra parte, deben esperarse cambios a medida que el sistema es utilizado.

Razones para desarrollar prototipos de sistemas

Los requerimientos de información no siempre están bien definidos. Es probable que los usuarios conozcan sólo ciertas áreas de la empresa donde se necesitan mejoras o cambios en los procedimientos actuales. Los requerimientos del usuario pueden ser demasiado vagos al formular el diseño. El prototipo es, en realidad, un modelo piloto o de prueba; el diseño evoluciona con el uso. Aunque el prototipo es un sistema que funciona, está diseñado para ser modificado con facilidad. La información obtenida con su uso se aplica en un nuevo diseño que se emplea, otra vez, como prototipo y que revela más información valiosa sobre el diseño. El proceso se repite las veces que sea necesario para revelar los requerimientos esenciales de diseño.

El proceso fundamental del desarrollo de prototipos es el siguiente:

Los usuarios pueden señalar las características que les agradaría o no tener, junto con los problemas que presenta un sistema que existe y funciona, con mayor facilidad que si se les pidiese que las describieran en forma teórica o por escrito. El uso y la experiencia producen comentarios más significativos que el análisis de diagramas y las propuestas por escrito.

En general, los pasos a seguir en el proceso de desarrollo de prototipos es el siguiente:

- ✓ Identificar los requerimientos de información que el usuario conoce junto con las características necesarias del sistema.
- ✓ Desarrollar un prototipo que funcione.
- ✓ Utilizar el prototipo anotando las necesidades de cambios y mejoras. Esto expande la

lista de los requerimientos de los sistemas conocidos.

- ✓ Revisar el prototipo con base en la información obtenida a través de la experiencia del usuario.
- ✓ Repetir los pasos anteriores las veces que sea necesario, hasta obtener un sistema satisfactorio.

Tal como lo sugieren los pasos anteriores, la construcción de prototipos no es un proceso de desarrollo por prueba y error, ya que antes que dé inicio cualquier actividad de diseño o programación, es necesario identificar todos los requerimientos.

Métodos para el desarrollo de prototipos.

Con los prototipos la velocidad de desarrollo es más importante que la eficiencia en el procesamiento. Un sistema prototipo se construye con rapidez, frecuentemente en días o semanas. Por otro lado, el costo asociado a esta tarea es mucho menor comparado con el de un sistema convencional, aún a pesar de no ser tan eficiente como los sistemas desarrollados sobre periodos de meses. Los sistemas prototipo pueden desarrollarse con métodos y lenguajes de programación convencionales, aunque no contengan todas las características y toques finales que normalmente se incluyen en un sistema terminado.

En algunos casos, aquellos donde el sistema será utilizado con poca frecuencia, el prototipo puede, de hecho, convertirse en el sistema terminado.

1.6.4 ¿Qué método de desarrollo es el más apropiado?

No existe ningún método correcto para desarrollar un sistema de información, pero sí existen diferentes formas para producir el sistema correcto para una aplicación. En ciertas ocasiones el único método adecuado será un enfoque paso por paso, comparable con el ciclo de vida de desarrollo de un sistema. En otros casos, el desarrollo de prototipos es el único método que tiene sentido. En otras situaciones se combinan los métodos y, además, los usuarios desarrollan parte de la aplicación. El indicador definitivo del éxito de un método de desarrollo en particular es aquel que se refiere a los resultados obtenidos y no a la "precisión" teórica del método.

C

a

p

i

t

u

l

o

Bases de Datos

II

II Generalidades

II.1 Conceptos básicos de Bases de Datos

Una base de datos es un conjunto de datos relacionados entre sí formalmente definidos, los cuales son compartidos por numerosos usuarios para diversas aplicaciones. La esencia de una base de datos es *el manejador de la base de datos (DBMS)* que es el programa que organiza y mantiene los datos, y la aplicación de la base de datos, la cual nos permite consultar, insertar, actualizar la información almacenada, etc.

Un sistema de base de datos es en esencia un sistema cuyo propósito general es registrar y mantener información. Tal información puede estar relacionada con cualquier cosa que sea significativa para la organización donde el sistema opera, en otras palabras, cualquier dato necesario para los procesos de toma de decisiones relacionados a la organización que se pretende automatizar.

Un sistema de bases de datos incluye cuatro componentes principales:

- Datos
- Hardware
- Software
- Usuarios

Los objetivos que debe cubrir un sistema de base de datos son:

1.- Asegurar que los datos puedan ser compartidos por los usuarios y por lo tanto, tener variedad de aplicaciones para lo cual, se necesita integridad en los datos y eliminación o minimización de redundancia.

2.- Permitir un mantenimiento de datos preciso y consistente

3.- Asegurar que los datos siempre estén disponibles tanto para aplicaciones actuales como futuras, es decir, que la base de datos pueda evolucionar conforme se modifiquen las necesidades de los usuarios y de sus aplicaciones.

4.- Permitir que el usuario desarrolle su propia visión de los datos sin preocuparse por la manera en que los datos se encuentran almacenados físicamente.

5.- Asegurar una protección de la base contra accesos o modificaciones no autorizadas.

¹ Database System Concepts

11.2 Sistemas Manejadores de Bases de Datos

Un sistema manejador de bases de datos es una colección de archivos interrelacionados y un conjunto de programas que son utilizados por diferentes usuarios para acceder o modificar dichos archivos.²El principal propósito de un manejador de bases de datos es proporcionar a los usuarios el acceso a los datos de manera eficiente y transparente, es decir, completamente independiente a la forma de almacenamiento.

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

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

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

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

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

Un manejador de bases de datos proporciona a los usuarios una visión abstracta de la información. Es decir, el sistema oculta ciertos detalles relativos a la forma como los datos se almacenan y mantienen.

Tradicionalmente se definen tres niveles de abstracción:

- ◆ Nivel físico o esquema interno en donde se describe cómo se almacenan los datos, sus interrelaciones y la mecánica del acceso. Por ejemplo en el nivel interno, los empleados

² Sistemas de Bases de Datos

de una empresa se representan por un tipo de registro almacenado llamado EMP_ALMCDO, de 18 bytes de longitud. EMP_ALMCDO contiene cuatro tipos de campo almacenado: un prefijo de 6 bytes y tres campos de datos que corresponden a las tres propiedades de EMPLEADO. Además, los registros de EMP_ALMCDO están indicados en el campo EMP# por medio de un índice llamado EMPX.

- ◆ Nivel o esquema conceptual que describe cuáles son los datos que están almacenados y sus relaciones existentes entre ellos. Este nivel es usado por los administradores de la base de datos quienes deciden qué información es guardada. También se conoce como modelo lógico de la base de datos. A nivel conceptual, la base de datos contiene información acerca de un tipo de entidad llamado EMPLEADO donde cada EMPLEADO tiene un NUMERO_EMPLEADO de seis caracteres, un NUMERO_DEPARTAMENTO de cuatro caracteres y un SALARIO de cinco dígitos.
- ◆ Nivel de vista o esquema del usuario en el cual se describe solamente una parte de los datos y donde el sistema puede proporcionar muchas vistas diferentes de la misma base de datos. Un ejemplo de este nivel puede ser que un usuario tenga una vista externa de la base de datos donde cada empleado se representa por un registro que contiene dos campos omitiendo los números de departamento y un usuario que de igual manera tiene una vista externa donde cada empleado se representa por un registro que también contiene dos campos pero que en lugar de omitir el número de departamento omita el salario.

Como ya se mencionó una parte importante del sistema de base de datos son los usuarios que pueden ser programadores de aplicaciones o usuarios terminales en línea cuyo nivel de destreza varía.

Cada usuario tiene un lenguaje a su disposición. Para el programador de aplicaciones se trata de un lenguaje convencional de programación; para el usuario de una terminal se trata de un lenguaje de consulta o un lenguaje de propósito especial, de sus necesidades y apoyado por un programa de aplicación en línea. Para nuestros propósitos, lo importante del lenguaje del usuario es que incluye un sublenguaje de datos (DSL), es decir, un subconjunto del lenguaje total que concierne a los objetos y a las operaciones de la base de datos.

Se dice del sublenguaje de datos que está inmerso en un lenguaje anfitrión. Un sistema específico puede incluir múltiples lenguajes de este tipo y numerosos sublenguajes de datos. En principio, cualquier sublenguaje de datos es una combinación de dos lenguajes: un lenguaje de definición de datos (DDL), que permita la definición o descripción de los objetos de la base de datos tal como lo percibe el usuario y un lenguaje de manipulación de datos (DML) que apoya el manejo o procesamiento de esos objetos. En conjunto lo descrito hasta aquí forma la arquitectura de un sistema de bases de datos. (Figura II.1).

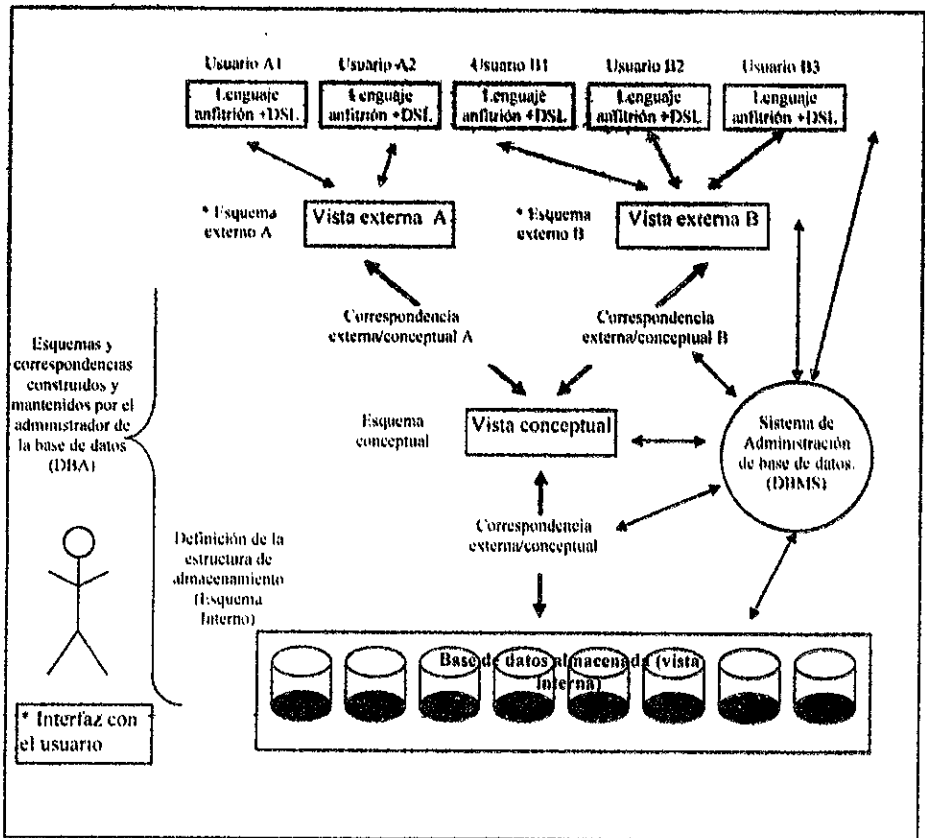


Figura II.1 Arquitectura del sistema de base de datos

II.3 Modelos de Manejadores de Bases de Datos

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

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

II.3.1 Modelo Manejador de Archivos

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

Ventajas

- La ventaja que tiene este modelo es su sencillez.

Desventajas

- No existe ninguna indicación de las relaciones entre varios elementos dentro del almacenamiento secuencial.
- Crea ciertos problemas a la integridad de los datos; los valores de los datos tienen que ser verificados por la aplicación antes de ser almacenados en el disco. La misma base de datos puede ser accesada por diferentes aplicaciones, y cada una puede tener diferentes valores para un mismo campo. Las diferentes aplicaciones tienen que ser controladas manualmente para asegurar que la definición de cada campo sea la adecuada respecto al dominio.
- La única manera de ordenar los datos es leyendo todo el archivo y reescribiéndolo en el nuevo orden. Esto puede ser resuelto con el uso de un archivo de índices, el cual contiene apuntadores a cada registro de la base de datos.
- El sistema manejador de archivos no permite que se modifique fácilmente la estructura de la base de datos.

II.3.2 Modelo Jerárquico

El modelo se parece a un árbol invertido que se compone de nodos. Cada nodo contiene elementos de datos que constituyen un tipo de registro o una entidad. Los nodos sin ningún subordinado se llaman hojas (Figura II.2)

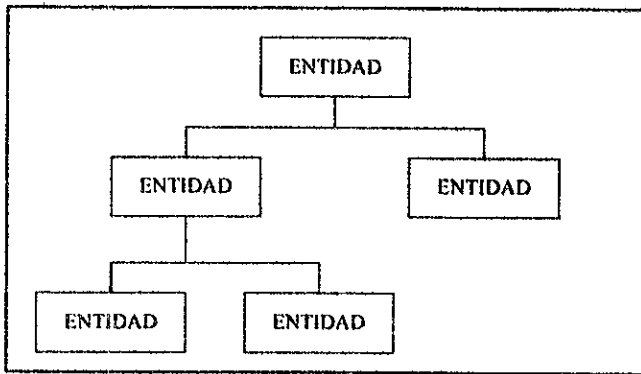


Figura II.2 Estructura Jerárquica.

Al nivel más alto de la jerarquía se llama raíz. Con excepción de la raíz todo nodo está vinculado con un solo nodo en el nivel superior, es decir, ninguno de los nodos puede tener más de un padre, en cambio todo padre puede tener uno o más nodos relacionados en un nivel más abajo denominados hijos.

Ventajas

- La relativa simplicidad y facilidad de uso del modelo jerárquico y la familiaridad de los usuarios del procesamiento de datos con la jerarquía.
- Existe una reducción de la dependencia de los datos.
- La predicción del funcionamiento se simplifica a través de relaciones predefinidas.

Desventajas

Las relaciones varios a varios pueden implementarse sólo de una manera deficiente. Esto puede traer como consecuencia redundancia en los datos almacenados lo cual es indeseable a nivel físico.

- Como resultado del estricto ordenamiento jerárquico, las operaciones conocidas como de inserción y de supresión se vuelven extremadamente complejas.
- La eliminación de los padres trae como consecuencia la eliminación de los hijos. Por tal motivo, los usuarios deben tener cuidado cuando tengan la necesidad de poner en funcionamiento la operación de la supresión.

La raíz es el tipo de nodo dominante. Cualquier nodo hijo es accesible solamente a través de su nodo padre.

II.3.3 Modelo de Red

En un modelo de datos de red, llamada algunas veces estructura plex, cualquier nodo puede vincularse con otro; es decir, un hijo puede tener más de un padre a diferencia del modelo anterior. Para que haya relación entre nodos no es necesario iniciar por la raíz, si no que se pueden relacionar los nodos a partir de un elemento que tengan en común (Figura II.3)

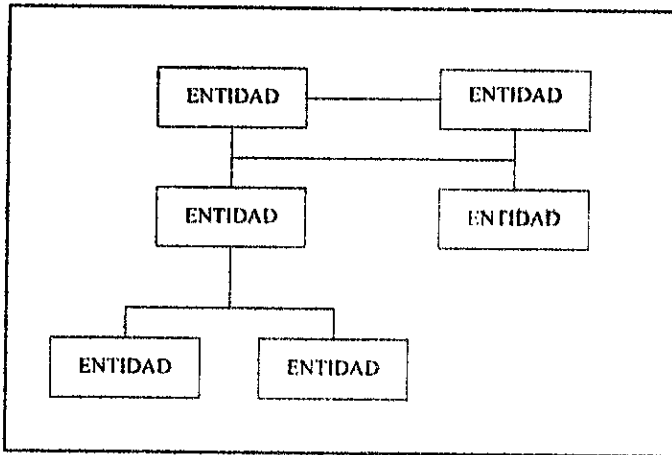


Figura II.3 Estructura de red.

Ventajas

- La principal ventaja consiste en que existen sistemas de manejo de base de datos que utilizan con éxito, el modelo de red como estructura básica.
- La relación varios a varios que ocurre con frecuencia en la vida real, se puede implementar fácilmente.

Desventajas

- La principal desventaja del modelo de red es su complejidad ya que el programador debe navegar a través de diferentes ocurrencias de conjunto, con la ayuda de las ocurrencias del tipo de registro conector.
- Cuando la base de datos se reorganiza, es posible, a menos que se tenga mucho cuidado, que se pierda la independencia de los datos.

II.3.4 Modelo Relacional

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

En este modelo los datos son organizados en conjuntos lógicos y matemáticos en una estructura tabular, donde cada campo representa una columna de la tabla y cada registro un renglón. (Figura II.4)

# eta	nombre	facultad
89877407	Sánchez Cruz	Psicología
90127839	Torres Gil	Medicina
91542818	Castro López	Medicina

Figura II.4 Tabla

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

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

Ventajas

- Simplicidad. El usuario final se representa como un modelo de datos. Un modelo relacional es lo que el usuario ve y no necesariamente lo que se implementará físicamente.
- Debido a que no hay dependencia de posición entre las relaciones, las consultas no tienen por qué reflejar ninguna estructura preferida y por lo tanto pueden ser procedimientos no estándares de consulta.
- El modelo relacional elimina los detalles relativos a la estructura del almacenamiento y la estrategia del acceso de la interfase con el usuario.
- Proporciona un grado de independencia de los datos relativamente más alto que los modelos anteriores.

Desventaja

- El diseño de las relaciones debe ser completo y exacto.

Debido a que este modelo es el más utilizado, se profundizará más sobre el mismo.

Modelo entidad-relación

El propósito del modelo entidad-relación es permitir la descripción del esquema conceptual sin prestar atención al diseño físico de la base de datos. Fue desarrollado para facilitar el diseño de bases de datos. Este modelo se basa en una percepción de la realidad, que consta de una colección de objetos básicos denominados entidades y de las relaciones entre ellos.

Una entidad es cualquier objeto o evento, acerca del cual, se recolectan datos. Una entidad puede ser una persona, un lugar o un objeto y las características o adjetivos que permiten diferenciar una entidad de otra se denominan atributos. Las relaciones son asociadas entre entidades.

En los modelos entidad-relación la conectividad de una relación es un punto de gran importancia, ya que especifica el tipo de asociación entre las ocurrencias de las entidades que conforman la relación. Los valores de la conectividad son "uno" o "muchos". El número asociado con el término muchos es llamado la cardinalidad de la conectividad.

Los tipos básicos de conectividad se describen a continuación:

Relación uno a uno.

Se dice que una relación entre entidades es uno a muchos si la ocurrencia de una entidad A está relacionada con ocurrencias múltiples de la entidad B y una ocurrencia de la entidad B está relacionada a lo más con una ocurrencia de la entidad A. En la figura II.5 se muestra una relación uno a muchos.

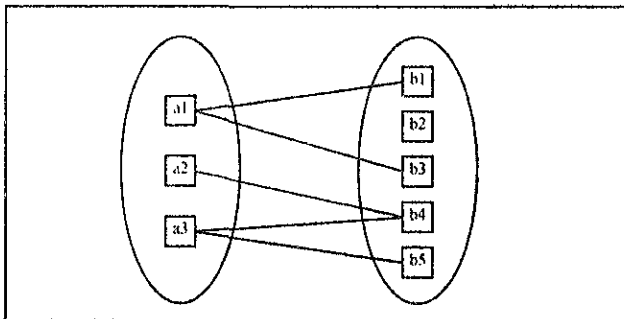


Figura II 5 Vista conceptual de una relación uno a muchos.

Relación uno a uno.

Este es un caso particular de la relación uno a muchos donde la ocurrencia de una entidad A se puede asociar a lo más a una sola ocurrencia de la entidad B y viceversa. (Figura II.6)

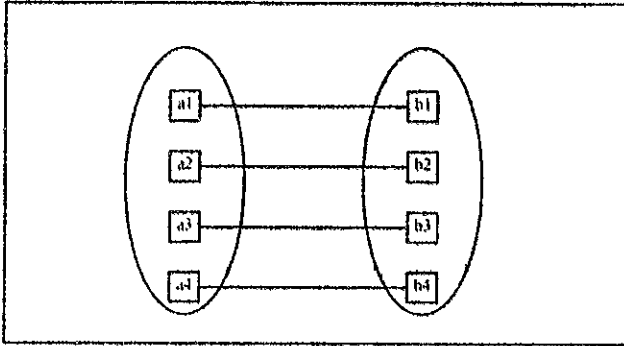


Figura II.6 Vista conceptual de una relación uno a uno.

Relación muchos a muchos.

Sucede cuando se pueden asociar muchas ocurrencias en una entidad A con muchas ocurrencias en la entidad B y viceversa. (Figura II.7)

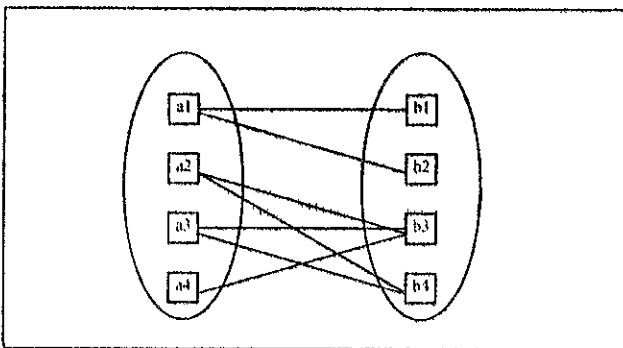


Figura II.7 Vista conceptual de una relación muchos a muchos.

Casos particulares.

Relación ISA.

Las relaciones ISA son un caso particular que ocurre cuando una entidad se parte por diferentes valores de su atributo en común. Por ejemplo, STAFF (equipo) es una generalización de empleados ASALARIADOS y por HONORARIOS. Ambos comparten atributos comunes, como número de empleado o fecha de contratación, pero ambos tienen atributos distintos asociados con la generalización/especialización. Por ejemplo ASALARIADO puede tener el atributo SALARIO, mientras que HONORARIOS tendría al atributo PAGO-POR-HORA.

Por lo tanto, un miembro del STAFF puede ser ASALARIADO u HONORARIOS. Si fuese ASALARIADO tendría los atributos de STAFF más los atributos de ASALARIADO. De la misma manera si fuera HONORARIOS tendría los atributos de STAFF más los de HONORARIOS. (Figura II.8)

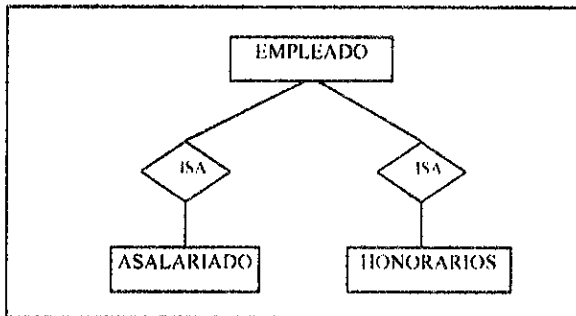


Figura II.8 Ejemplo de una relación ISA

Las relaciones ISA se utilizan para resaltar los parecidos entre tipo de entidades y ocultar sus diferencias. La distinción se hace a través de un proceso llamado de herencia de atributos donde los atributos de los conjuntos de entidades del nivel superior son heredados por las entidades de nivel inferior.

Entidad débil.

Se tiene una entidad débil cuando la existencia de una entidad depende únicamente de la existencia de otra. Por ejemplo, la existencia de la entidad HIJO en la base de datos depende de la existencia de los empleados asociados. En otras palabras, si un empleado deja la compañía, ya no se registrará a sus hijos. (Figura II.9).

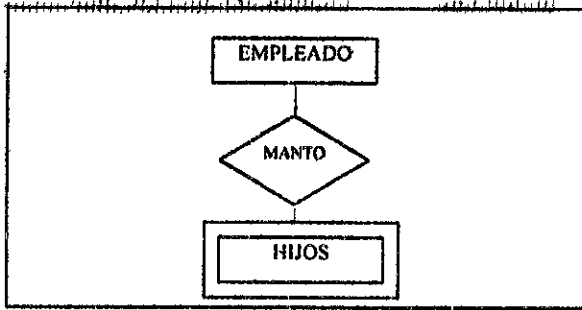


Figura II.9 Ejemplo de una entidad débil

Una entidad débil se representa con un rectángulo de línea doble y por lo general la relación entre una entidad débil y su entidad asociada es de muchos a muchos. Por ejemplo si un padre de la compañía, la entidad HIJO podría seguir existiendo si la madre también es empleada de la compañía.

Gerundio.

Un gerundio es una relación convertida en entidad y se representa con un rombo dentro de un rectángulo. (Figura II.10)

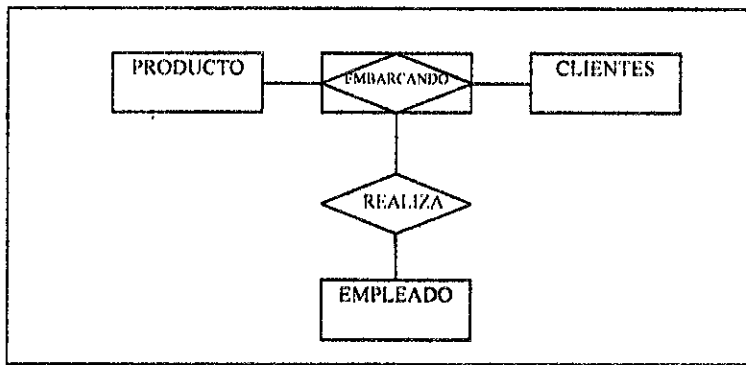


Figura II.10 Ejemplo de un gerundio

Conceptualmente las entidades y las relaciones son distintas pero desde una perspectiva de base de datos la diferencia debe expresarse en términos de atributos. Para hacer tal distinción se asigna una llave primaria a cada entidad.

El concepto de llave primaria surge a partir del de llave candidata. Una llave candidata es un atributo elemental o conjunto de ellos en un registro que se pueden utilizar como criterios de identificación para éste.

Cuando una llave candidata es elegida para identificar de manera exclusiva a un registro se le denomina llave primaria (PK).

La estructura lógica de la base de datos puede ser mostrada gráficamente por un diagrama entidad-relación que consta de los siguientes elementos que se muestran en la figura II.11

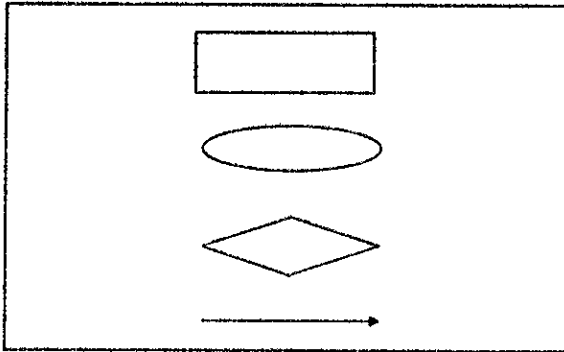


Figura II.11 Elementos de un diagrama entidad-relación

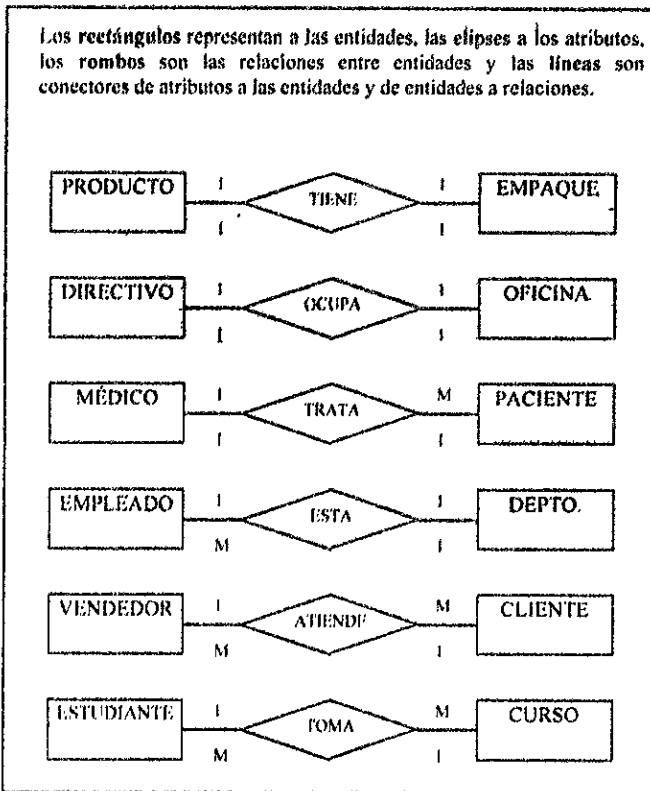


Figura 11.12 Ejemplos de diferentes tipos de relación utilizando diagramas entidad-relación.

Para desarrollar el diagrama entidad-relación se deben definir las entidades de interés para el sistema así como sus atributos, las relaciones existentes entre las entidades y la cardinalidad de las asociaciones.

11.3.5 Bases de Datos Orientadas a Objetos.

La evolución de las Bases de Datos no se detiene, por lo tanto, es necesario contribuir e investigar para lograr insertarlas en paradigmas de programación eficientes y actuales, para que formen parte de los nuevos métodos de programación y que sea posible aplicarles todas las ventajas de programación que se han logrado. En base a lo anterior, se han integrado las Bases de Datos al paradigma de Programación Orientada a Objetos, con el fin de tomar toda la riqueza de este paradigma y aplicarla directamente al desarrollo de

software. Antes de continuar, es necesario dejar en claro, lo que es la Programación Orientada a Objetos.³

- ***El Paradigma Orientado a Objetos***

La programación estructurada ha dominado la computación durante los últimos años. Ahora a llegado la programación orientada a objetos cuyas siglas en inglés son OOP ("*Object Oriented Programing*")

En términos sencillos, la programación orientada a objetos (OOP) es la programación con objetos. En términos simplistas, un objeto es la forma de construir un lenguaje que ate los datos con las funciones que operan en los datos. Los objetos pueden ser extendidos para que incorporen nuevos elementos y funciones utilizando la propiedad de herencia.

- ***Diferencias entre la Programación Orientada a Objetos y la Programación Tradicional.***

Para aprovecharse de la flexibilidad y de la potencia que ofrece la programación orientada a objetos, es necesario cambiar el enfoque que se le da a la programación.

La figura II.13 ilustra la diferencia clave entre el enfoque tradicional de programación orientada a funciones y el enfoque orientado a objetos. Se observa que la diferencia principal tiene que ver con la forma como el código y los datos se "empaquetan" y "organizan". Otras importantes diferencias son:

1. En la vista tradicional, las funciones son lo más importante. Todo el código en un programa está diseñado alrededor de esas funciones.
2. En la vista de programación orientada a objetos, los objetos son lo más importante. Los programas están diseñados en función de objetos; las funciones son secundarias. Este cambio de rol (función) se pone de manifiesto en la manera como se usan los objetos. En lugar de simplemente pasar los objetos a las funciones, los objetos se usan para llamar a las funciones.
3. Los programas pueden evitar el uso de grandes funciones lógicas para casos múltiples. En su lugar se crean objetos para representar los diferentes componentes lógicos de un programa.

³ Object-Oriented Databases A semantic Data Model Approach

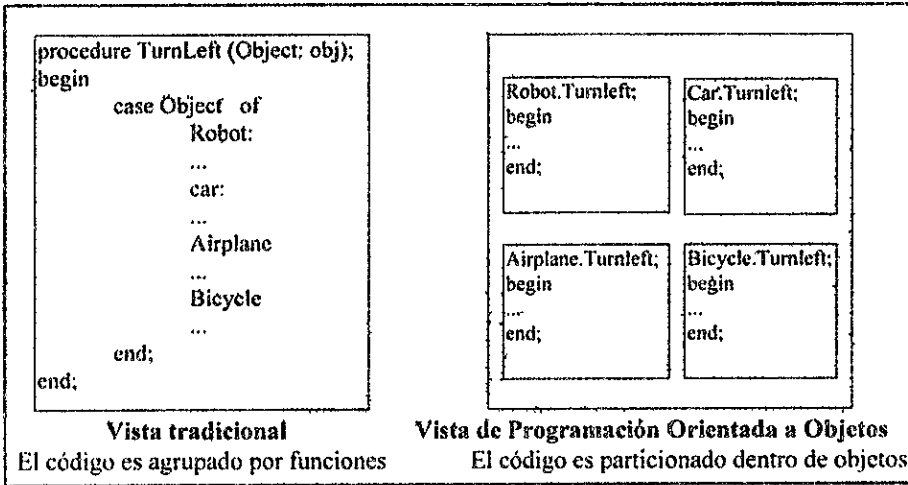


Figura II.13 Programación Tradicional contra Programación Orientada a Objetos

• *Conceptos fundamentales de la Programación Orientada a Objetos*

➤ *¿Qué es un objeto?*

Las personas nos formamos conceptos desde temprana edad. Cada concepto es una idea particular o una comprensión de nuestro mundo. Los conceptos adquiridos nos permiten sentir y razonar acerca de las cosas en el mundo. A estas cosas a las que se aplican nuestros conceptos se llaman *objetos*. Un objeto puede ser real o abstracto, como los ejemplos siguientes:

- Una factura
- Una empresa
- Un satélite
- Una figura
- Todo un plano de ingeniería
- Un teclado

En la programación orientada a objetos, nos interesa el comportamiento del objeto. El software que implanta el objeto contiene estructuras de datos y operaciones que expresan dicho comportamiento. Las operaciones se codifican como *métodos*. La representación en software orientado a objetos del objeto es entonces una colección de tipos de datos y métodos. En software orientado a objetos:

“Un objeto es cualquier cosa real, o abstracta, acerca de la cual almacenamos datos y los métodos que controlan dichos objetos.”

Los conceptos que poseemos se aplican a tipos determinados de objetos. Por ejemplo, *empleado* se aplica a los objetos que son personas empleadas por alguna organización. Algunas instancias de *empleado* podrían ser Juan Pérez, María Sánchez, etc. En el análisis orientado a objetos, estos conceptos se llaman *tipos de objetos*; las instancias se llaman *objetos*. En la figura II.14 se ilustra este ejemplo.

Un tipo de objeto es una categoría de objeto
Un objeto es una instancia de un tipo de objeto

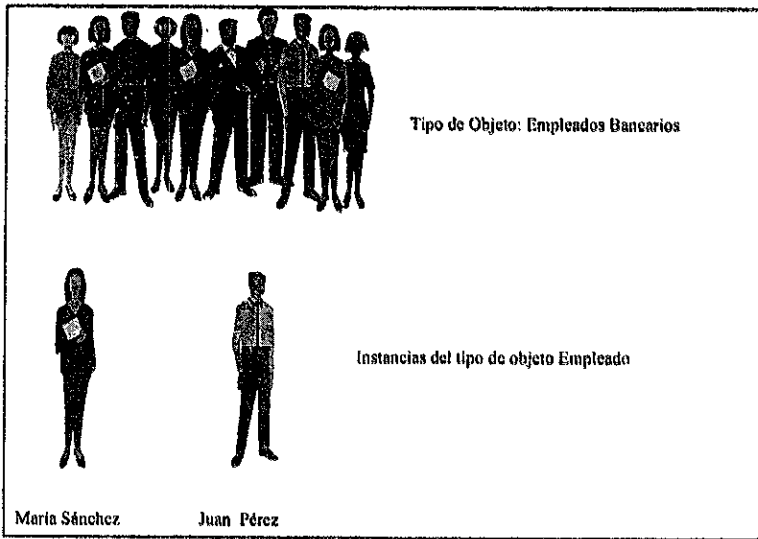


Figura II.14 Ejemplo de tipo de objeto

► **Métodos.**

Los métodos especifican la forma en que se controlan los datos de un objeto. Un método asociado con el tipo de objeto *factura* podría ser aquél que calcule el total de una factura, otro podría transmitir la factura a un cliente, otro podrá verificar de manera periódica si la factura ha sido pagada, etc.

Los métodos en un tipo de objeto sólo hacen referencia a las estructuras de datos de ese tipo de objeto. No deben tener acceso directo a las estructuras de datos de otros objetos. Para utilizar la estructura de datos de otro objeto, deben enviar un mensaje a éste. El tipo de objeto empaqueta juntos los tipos de datos y los métodos.

Un objeto es entonces un elemento cuyas propiedades están representadas por tipos de datos y su comportamiento por métodos.

► **Encapsulado**

Al empaque conjunto de datos y métodos se le llama *encapsulado*. El objeto esconde sus datos de los demás objetos y permite el acceso a los datos mediante sus propios métodos. Esto recibe el nombre de ocultamiento de la información. Por otra parte, el encapsulado evita la corrupción de los datos de un objeto.

El encapsulado es importante porque separa el comportamiento del objeto de su implantación. Esto permite la modificación de la implantación de un objeto sin que se tengan que modificar las aplicaciones que lo utilizan.

La figura II.15 ilustra un objeto. La estructura de datos en el centro sólo puede ser utilizada por los métodos del anillo exterior.

“Encapsulado es el resultado (o acto) de ocultar los detalles de implantación de un objeto respecto de su usuario.”

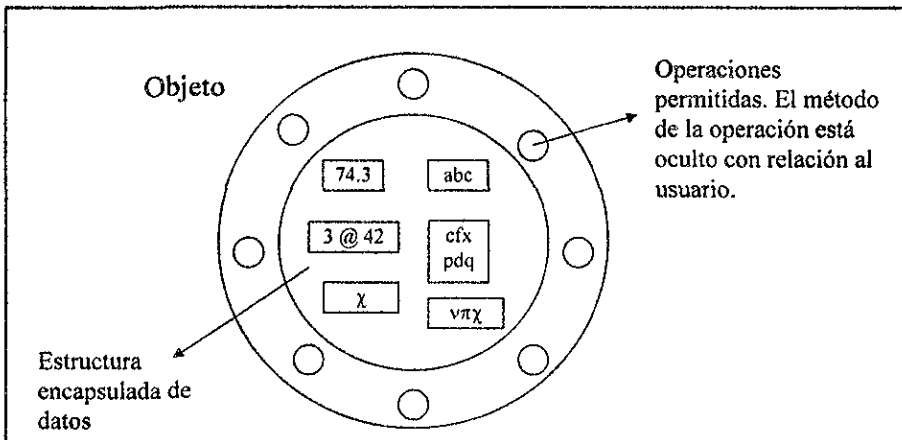


Figura II.15 Cada objeto encapsula una estructura de datos y métodos. Una estructura de datos está en la parte central del objeto. El objeto es controlado por métodos que implantan las operaciones permitidas. La estructura de datos sólo puede ser utilizada por dichos métodos. A esta restricción de acceso se le llama encapsulado.

► **Mensajes**

Para que un objeto haga algo, le enviamos una solicitud. Eso hace que se produzca una operación. La operación ejecuta el método apropiado y, de manera opcional, produce

una respuesta. El mensaje que constituye la solicitud contiene el nombre del objeto, el nombre de una operación y, a veces, un grupo de parámetros, como se muestra en la figura II.16

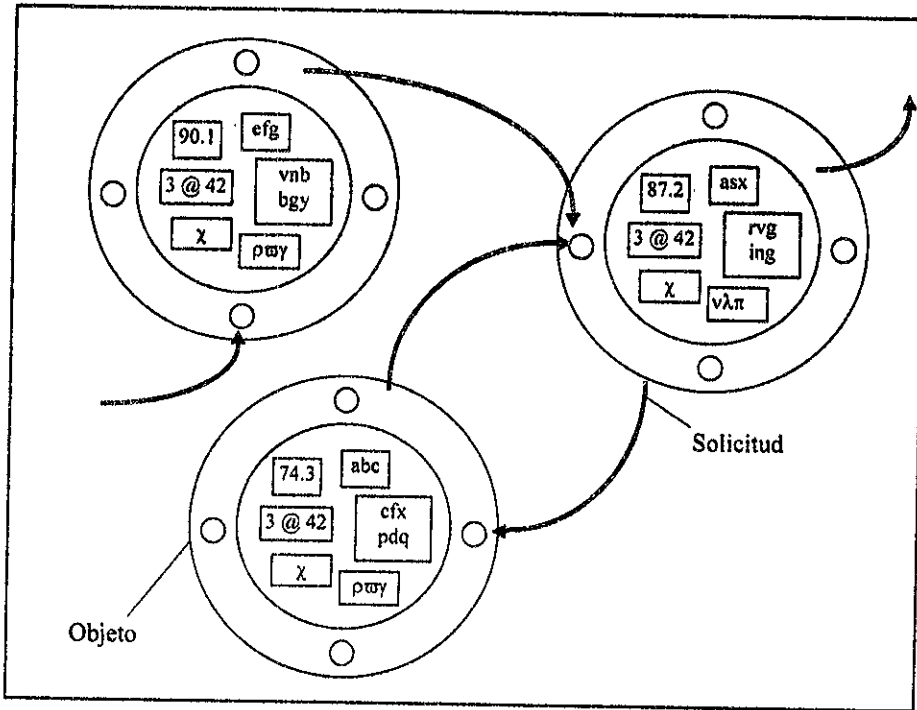


Figura II.16 Los objetos se comunican mediante solicitudes. Una solicitud es un mensaje que especifica la realización de la operación con uno o más objetos y que, de manera opcional, genera un resultado.

► **BLOB** ("Binary Large Object", objeto binario de gran tamaño).

Las computadoras actuales almacenan grandes cadenas de bits que representan imágenes, diagramas, enunciados, inclusive música o video. Cada uno de esos objetos es conocido como BLOB. Los BLOBs tienen métodos que permiten mostrarlos o utilizarlos.

La creciente aceptación del procesamiento de imágenes aumentará la importancia de las bases de datos orientadas a objetos que puedan administrar los BLOBs de manera eficiente. Enviaremos solicitudes a los BLOBs para indicarles que se exhiban a sí mismos, se cifren a sí mismos, se ligen para su edición, etc.

En una base de datos orientada a objetos, una factura se podría mantener en una forma gráfica hecha a mano junto con las instrucciones orales del cliente, así como un registro alfanumérico. La imagen gráfica es un objeto; las instrucciones orales son otros objetos, el registro alfanumérico es otro. Juntos forman un objeto único *factura*.

» *¿Qué es una clase?*

El término clase se refiere a la implantación en software de un tipo de objeto. Especifica una familia de objetos sin estipular la forma en que se implanten.

“Una clase es una implantación de un tipo de objeto. Especifica una estructura de datos y los métodos operativos permisibles que se aplican a cada uno de sus objetos.”

Por ejemplo, una clase *empleado* incluirá datos de seguro social, puesto, salario, extensión telefónica, etc. En la figura II.17 se ilustra este ejemplo. Además, cada clase define un conjunto de operaciones permisibles que permiten el acceso y modificación de los datos del objeto. Una clase *empleado* podría incluir operaciones como contratar, promover y cambiar número de extensión para una clase. Los detalles del método operativo se especifican en la clase.

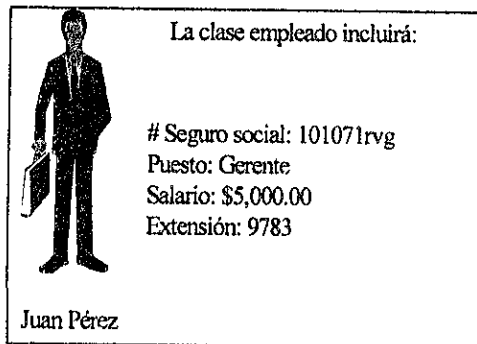


Figura II.17 Ejemplo de clase

» *Herencia.*

Un tipo de objeto de alto nivel puede especializarse en tipos de objetos de bajo nivel. Un tipo de objeto puede tener subtipos.

Por ejemplo, el tipo de objeto persona puede tener subtipos Civil y Militar. Militar puede tener subtipos Oficial y Enrolado. Oficial puede tener subtipos teniente, capitán y mayor, y también subtipos como marino o zapador, oficial en servicio activo u oficial retirado. Existe una jerarquía de tipos, subtipos, subsubtipos, etc. En la figura II.18 se ilustra este ejemplo.

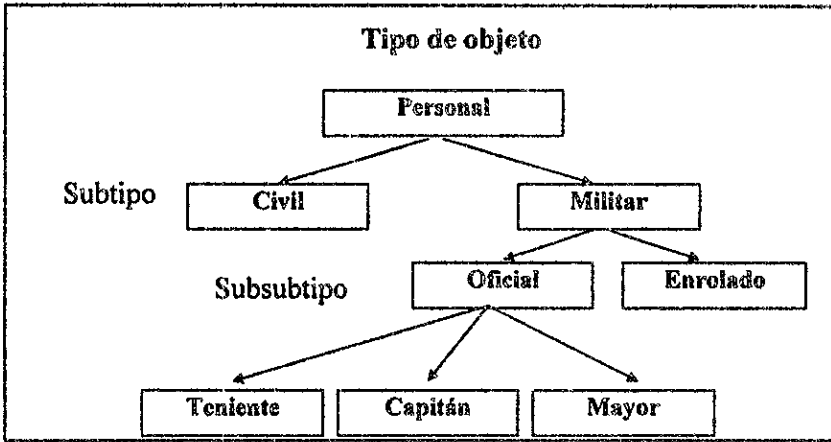


Figura II.18 Ejemplo de Herencia

La figura II.19 muestra una clase y una subclase. La subclase tiene los mismos métodos que su superclase, pero tiene también el método G. A veces, una clase hereda propiedades de más de una superclase. Esto recibe el nombre de herencia múltiple.

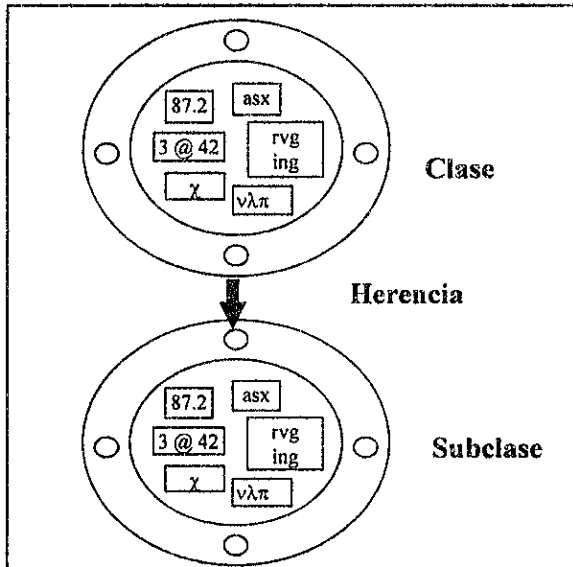


Figura II.19 Una clase puede tener sus propios métodos y estructura de datos, así como también heredarlos de su superclase.

► **Abstracción.**

La abstracción consiste en enfocarse a lo esencial, principalmente sobre los aspectos inherentes de una entidad e ignorando sus propiedades accidentales. En desarrollo de sistemas la abstracción está enfocada, a lo que hace y dónde está un objeto, antes de decidir cómo debería ser implementado. El uso de la abstracción guarda la libertad de tomar decisiones tan grandes como sea posible para evitar detalles o compromisos prematuros. El uso apropiado de la abstracción deja que el mismo modelo se use para el análisis, diseño de alto nivel, programación estructurada, estructuras de bases de datos, y documentación.

► **Polimorfismo.**

Polimorfismo es la habilidad de dos o más clases de objetos de responder al mismo mensaje, cada una en su propia manera. Esto significa que un objeto no necesita saber a quién envía un mensaje. Sólo necesita saber cuántas clases diferentes de objetos han sido definidas para responder a un mensaje en particular.

El polimorfismo nos permite reconocer y explotar similitudes entre diferentes clases de objetos. Cuando reconocemos que muchas clases diferentes de objetos pueden responder al mismo mensaje, nosotros reconocemos la diferencia entre el nombre del mensaje y un método.

• **Diferencia entre la programación orientada a objetos y eventos.**

Para hacer una comparación entre ambos tipos de programación, tomaremos en cuenta el sistema de programación de Microsoft Access 2.0. Microsoft Access 2.0 nos permite crear objetos, (ventanas, iconos, etc.) establecer y cambiar sus propiedades y, posteriormente, asignarles un código. Esto es diferente del método tradicional de elaboración de un programa en el cual existen estructuras para controlar el flujo del programa de un procedimiento a otro de manera lógica hasta que el programa termine.

Al realizar una interfase gráfica para el usuario, se generan formularios, dibujando controles y escribiendo módulos de código. Los botones de comando, menús, cajas de diálogo, son ejemplos de controles. Cuando un usuario activa un control al seleccionarlo, el código Basic que está asignado al control es invocado y realiza una tarea específica.

Microsoft Access 2.0 utiliza la metáfora del "evento" para describir sus paradigmas de programación. Siempre se utiliza este enfoque orientado a los eventos cuando se crea una aplicación. Orientado a eventos significa que todos los controles que se dibujen en un formulario, especificarán cómo se comportará la interfase. En otras palabras, los controles de Microsoft Access 2.0 esperan que sucedan eventos particulares antes de que responda. Un evento es una acción que es reconocida por un control.

• **Panorama de las Bases de Datos Orientadas a Objetos**

Una base de datos orientada a objetos (BDOO) almacena y manipula información que puede ser digitalizada (representada) como objetos, proporciona una estructura flexible con acceso ágil, rápido, con gran capacidad de modificación. Además, combina las mejores cualidades de los archivos planos, las bases jerárquicas y relacionales.

Actualmente, el creciente uso de las metodologías de programación orientadas a objetos está promoviendo la aparición de manejadores de BDOO en el mercado. Esto tiene sentido, puesto que la tecnología de objetos proviene del desarrollo de metodologías avanzadas de programación. Más aún, la comunidad internacional está convencida de que los manejadores de BDOO tienen la flexibilidad tanto en la definición del modelo de datos como en el desempeño tan anhelado por muchos desarrolladores de aplicaciones, lo que es imposible encontrar en los modelos jerárquicos de red o relacionales.

Los objetos pueden estar compuestos por cualquier tipo de información que, eventualmente, puede almacenarse en forma digital; por ejemplo, imágenes barridas, voz, sonido, dibujos, planos arquitectónicos complejos, esquemas electrónicos y diagramas desarrollados por ingenieros, así como los tradicionales tipos de datos alfanuméricos. Comúnmente, las aplicaciones que producen este tipo de objetos complejos, al terminar, los guardan en archivos de datos en distintos formatos. Cuando el programa es reactivado, los objetos, se cargan nuevamente. En estos ambientes, los objetos son accesibles sólo a un usuario en cada momento, no existen mecanismos de seguridad, no hay manera de protegerse ante la eliminación accidental de un objeto complejo. Las BDOO superan todas estas dificultades porque permiten que múltiples usuarios compartan objetos complejos para manipularlos en ambiente seguro y estructurado.

Las bases de datos convencionales fueron diseñadas para manejar tipos de datos alfanuméricos, por ello, difícilmente pueden usar objetos y métodos. Una base de datos de red o jerárquica puede almacenar objetos complejos, pero esta arquitectura no es flexible, lo cual motiva, el uso del modelo relacional. El problema principal con los modelos en red o jerárquicos es que la estructura es definida rigidamente, cuando la base de datos se crea. Estos sistemas casi no permiten flexibilidad para modificaciones, el sistema debe desactivarse cuando se requiere modificar estructuras de objetos o métodos.

Una base de datos relacional tiene una estructura más flexible, pero no puede manejar tipos de datos complejos. Para sobreponerse a estas limitaciones, algunos proveedores han desarrollado las bases de datos orientadas a objetos, las cuales son diseñadas para manipular los objetos con los conceptos de la programación orientada a objetos, proporcionando un concepto persistente en un ambiente multiusuario seguro.

► **Ventajas**

Entre las ventajas más ilustrativas de las BDOO está su flexibilidad, soporte para el manejo de tipos de datos complejos. Por ejemplo, en una base de datos convencional, si una

empresa adquiere varios clientes por referencia de clientes servicio, pero la base de datos existente, que mantiene la información de clientes y sus compras, no tiene un campo para registrar quién proporcionó la referencia, de qué manera fue dicho contacto, o si debe compensarse con una comisión, sería necesario reestructurar la base de datos para añadir este tipo de modificaciones. Por el contrario, en una BDOO, el usuario puede añadir una "subclase" de la clase de clientes para manejar las modificaciones que representan los clientes por referencia.

La subclase heredará todos los atributos, características de la definición original, además, se especializará en especificar los nuevos campos que se requieren así como los métodos para manipular solamente estos campos. Naturalmente se generan los espacios para almacenar la información adicional de los nuevos campos. Esto presenta la ventaja adicional que una BDOO puede ajustarse a usar siempre el espacio de los campos que son necesarios, eliminando espacio desperdiciado en registros con campos que nunca usan.

La segunda ventaja de una BDOO, es que manipula datos complejos en forma rápida y ágilmente. La estructura de la base de datos está dada por referencias (o apuntadores lógicos) entre objetos. No se requieren búsquedas en tablas o uniones para crear relaciones. Esta capacidad resulta atractiva en aplicaciones de la ingeniería, donde las relaciones entre componentes dependen de factores diversos.

II.4 Características de las bases de datos relacionales.

Un sistema de base de datos debe ser capaz de representar y manipular entidades o registros y sus relaciones de manera fácil y conveniente. En el enfoque jerárquico se representa la relación entre los segmentos por la posición relativa de arriba hacia abajo y de derecha a izquierda de los tipos de segmentos involucrados. En el enfoque de red, las relaciones se representan mediante mecanismos de "set" con uso de apuntadores, los cuales enlazan a un tipo de registro propietario con un tipo de registro miembro. Para el caso de una base de datos grande y complicada, el modelo lógico y la manera como pueden tener acceso a él los usuarios, por medio de un lenguaje de manipulación de datos puede volverse muy complejo. Aún más, el acceso a los datos se inclina demasiado a las rutas de acceso, en término de los enlaces o posiciones jerárquicas que estableció el diseñador. De esta manera puede ocurrir que muchos cambios a la base de datos violen la independencia de los datos o afecten los programas de aplicación.

El modelo relacional de bases de datos concebido en 1970 por E. F. Cood ⁴, constituye un enfoque muy diferente para la descripción y manipulación lógica de los datos.

En forma concisa, visualiza la base de datos lógica como una simple colección de tablas bidimensionales llamadas "relaciones". Estas tablas evitan la repetición de grupos.

⁴ Sistemas de Bases de Datos

Una base de datos relacional consiste en una colección de tablas, a cada una de las cuales se les asigna un nombre único. Un registro o tupla de una tabla representa una relación entre un conjunto de valores. Puesto que una tabla es una colección de dichas relaciones, hay una estrecha correspondencia entre el concepto tabla y el concepto matemático de relación, y por esto el modelo se llama modelo relacional.

El modelo relacional cumple con los principales objetivos de la etapa de análisis de bases de datos:

- Permite la conversión directa de la especificación relacional a una implementación física. Esto no era factible cuando Codd propuso el modelo relacional, pero la aceptación del modelo condujo al desarrollo de sistemas manejadores de bases de datos que soportan dicho modelo. Se debe hacer notar que una especificación relacional también se puede convertir en otras estructuras físicas, por ejemplo a un sistema de archivos donde una relación se puede convertir en un archivo, cada columna en un campo y cada renglón en un registro.
- El modelo relacional basa su desarrollo en el cumplimiento de los siguientes criterios, con la finalidad de generar buenas estructuras lógicas de datos:
 - 1.- Cada evento debe almacenarse sólo una vez en la base de datos.
 - 2.- La base de datos debe ser consistente.
 - 3.- La base de datos debe ser fácil de modificar.

El modelo relacional también proporciona lenguajes de consulta para poder acceder relaciones. Codd definió las bases para los lenguajes, el álgebra y el cálculo relacionales.

Un lenguaje de consulta es un lenguaje en el que el usuario solicita información de la base de datos. Los lenguajes de consulta pueden clasificarse en lenguajes procedurales o no procedurales. En los primeros el usuario da instrucciones al sistema para que realice una secuencia de operaciones en la base de datos para calcular el resultado deseado. En el segundo el usuario describe la información deseada sin dar un procedimiento específico para obtener esa información.

El álgebra relacional es un lenguaje de consulta procedural. Consta de un conjunto de operaciones que toman una o dos relaciones como entrada y producen una nueva relación como resultado. Las operaciones fundamentales del álgebra relacional son seleccionar, proyectar, producto cartesiano, renombrar, unión y diferencia de conjuntos.

Por otro lado, el cálculo relacional es un lenguaje de consulta no procedural. Describe la información deseada sin dar un procedimiento específico para obtener esa información. Es decir, busca el conjunto de registros que hagan verdadero a un predicado.

Con los puntos hasta aquí descritos se llegó a tener un modelo bien estructurado con las siguientes características:

- Simplicidad en su representación en la estructura lógica de la base de datos.
- Flexibilidad para establecer relaciones de datos por medio de campos de conexión.
- Todas las entidades en una base relacional están representadas como tablas separadas o relaciones y no están colocadas en una jerarquía fija.
- Cualquier tabla se puede acceder sin necesidad de acceder otras relaciones basadas en una estructura de datos fija.

Las tablas de las bases de datos relacionales deben cumplir con lo siguiente:

- Cada columna debe contener valores relativos al mismo atributo y cada valor de una columna debe ser simple o atómico, es decir, con un solo valor indivisible.
- Cada columna tiene un nombre distinto y el orden de las columnas no es importante.
- Cada renglón es distinto; esto es, un renglón no puede ser igual a otro para lo cual nos auxiliamos de las llaves.
- La secuencia de los renglones no es importante.

Todos los atributos no llave deben ser totalmente dependientes del o los atributos que funcionan como llave.

- Cada atributo no llave debe depender sólo de la llave primaria, no de otro atributo no llave.

El modelo relacional puede ser obtenido, partiendo del modelo entidad-relación (Figura II.20) y esto se logra teniendo en cuenta algunas convenciones que a continuación se mencionan:

1. Las entidades del modelo entidad-relación corresponden a las tablas del modelo relacional.
2. Los atributos de las entidades son las columnas o campos de las tablas del modelo relacional.

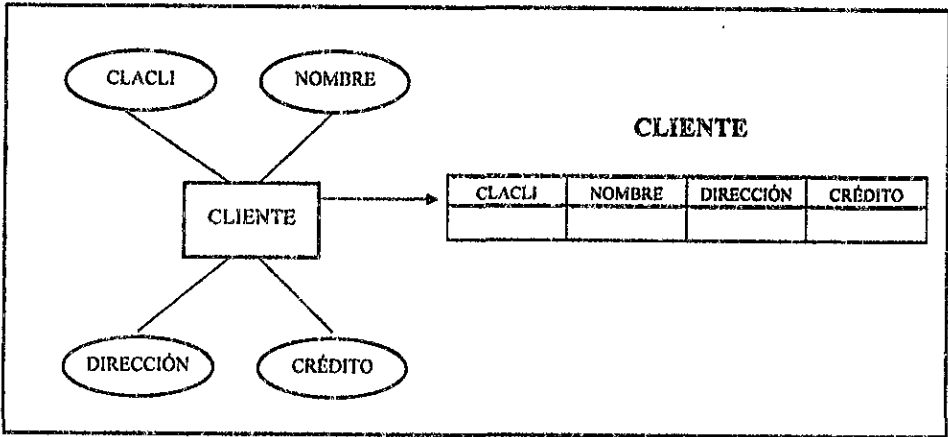


Figura II.20 Conversión de la entidad CLIENTE a la tabla CLIENTE.

De las relaciones uno a muchos (Figura II.21), el identificador de la llave entidad (formado por él o los atributos que conforman la llave primaria) correspondiente a la cardinalidad “uno” pasa a ser un campo en la tabla que representa a la entidad con cardinalidad “muchos” y tendrá el nombre de llave foránea (FK). Una llave foránea es entonces un atributo que es la llave primaria de otra tabla.

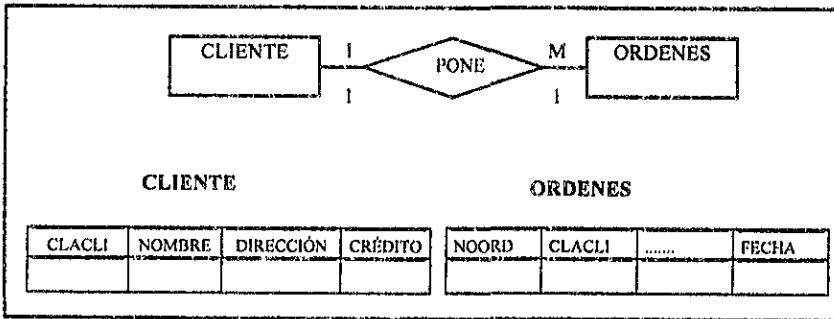


Figura II.21 Conversión de una relación uno a muchos

En las relaciones muchos a muchos (Figura II.22) se debe incluir una tabla intermedia que corresponde a la relación. Esta tabla contendrá los campos marcados como llaves primarias de las dos entidades asociadas y los campos propios de la relación.

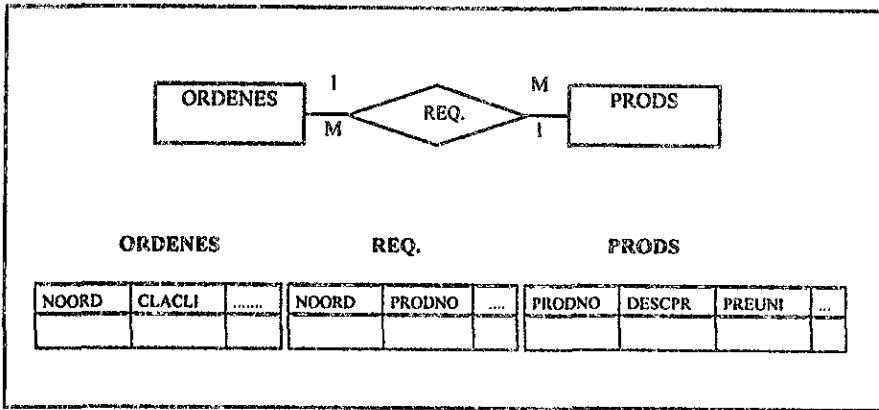


Figura II.22 Conversión de las relaciones muchos a muchos

En el caso de contar con entidades débiles (Figura II.23), la tabla correspondiente a la entidad débil debe contener la llave primaria de la tabla de la cual depende.

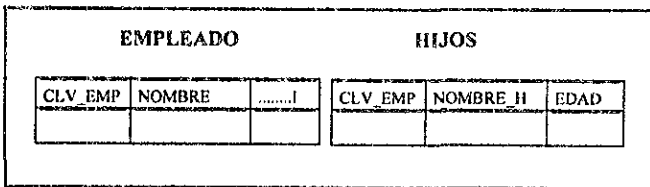


Figura II.23 Conversión de la entidad débil de la figura II.9

Para las relaciones ISA (Figura II.24) se pueden emplear dos procedimientos:

1.- Crear las tablas correspondientes a las entidades superiores e inferiores y agregar a las últimas la llave primaria de las entidades superiores.

2.- No crear las tablas correspondientes a las entidades superiores y crear las tablas para las entidades inferiores con sus correspondientes campos más aquellos que resulten de los atributos de las entidades superiores.

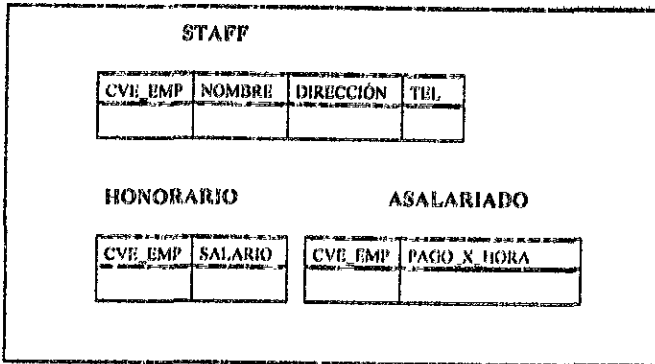


Figura II.24 Conversión de la relación ISA

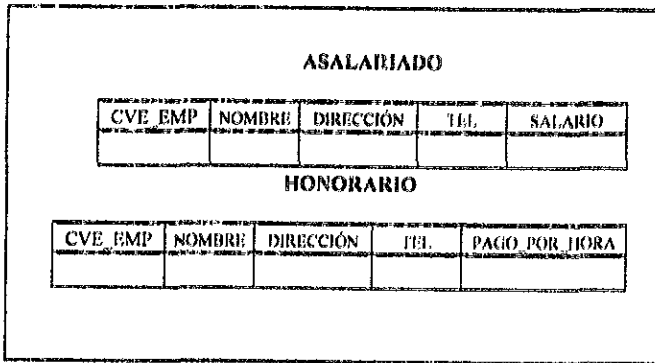


Figura II.25 Conversión de una relación ISA

II.4.1 Proceso de Normalización

La normalización es una técnica para analizar paso a paso las asociaciones entre los datos. Es una forma de verificar que al pasar del modelo entidad-relación al modelo relacional se obtenga un esquema en tercera forma normal. Esto es importante ya que una relación no normalizada puede contener grupos repetidos o bien puede suceder que un sólo atributo no sirva como llave candidata lo que genera datos redundantes.

Un ejemplo de una relación no normalizada puede ser la que se muestra en la figura II.26. En este ejemplo hay varios valores en la intersección de ciertos renglones y columnas. Dado que un estudiante toma más de un curso, los datos de los cursos constituyen un grupo repetitivo dentro de los datos del estudiante. Hay dos entradas para

cada columna del estudiante 38214 con el atributo CURSO. La relación entre MATRÍCULA y los demás atributos se muestran en la figura II.27.

MATRÍCULA	NOMBRE	CARRERA	CURSO	TÍTULO	PROF	CUB	CALIF
19483689	TREJO	IS	IS350	DIBUJO	ORTIZ	B104	10
			IS465	CONTROL	FLORES	B213	9
58974466	SOLÍS	I.C	IS465	ÁLGEBRA	FLORES	B213	8
			LC300	FÍSICA	GÓMEZ	D317	8
			LC440	QUÍMICA	FLORES	B213	10

Figura II.26 Boleta de calificaciones BOL_CALIF no normalizada

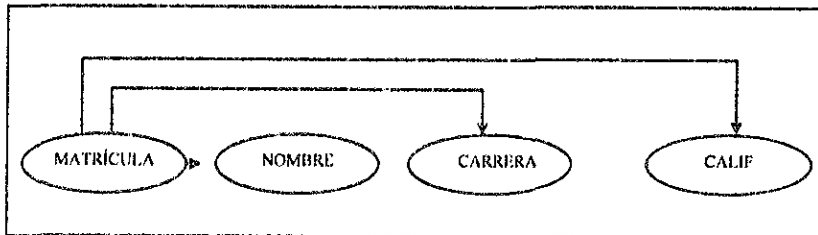


Figura II.27 Relación de matrícula con los demás campos

En la figura II.27 se puede observar que hay una relación uno a uno de MATRÍCULA a NOMBRE y a CARRERA. Sin embargo, la relación es de uno a muchos de MATRÍCULA a CURSO y los demás atributos. Por lo tanto, MATRÍCULA no es una llave candidata ya que no identifica de manera única todos los atributos de la relación.

Se le llama dependencia funcional a una relación entre atributos. Un atributo "b" es funcionalmente dependiente de un atributo "a", si cada valor de "a" tiene un solo valor de "b" asociado a él.

La teoría de la normalización está basada en el concepto de formas normales. Se dice que una relación está en forma normal particular si satisface cierto conjunto de restricciones.

II.4.2 Formas Normales.

Hay varias formas normales, de las cuales las tres primeras son las más conocidas y las que a continuación se detallan.

Primera forma normal

La primera forma normal debe cumplir con la identificación de la llave primaria y la eliminación de grupos repetidos. Para normalizar una relación que contiene un sólo grupo repetitivo, se elimina el grupo repetitivo y se forman dos nuevas relaciones.

Este proceso se ilustra a continuación. (Figura II.28)

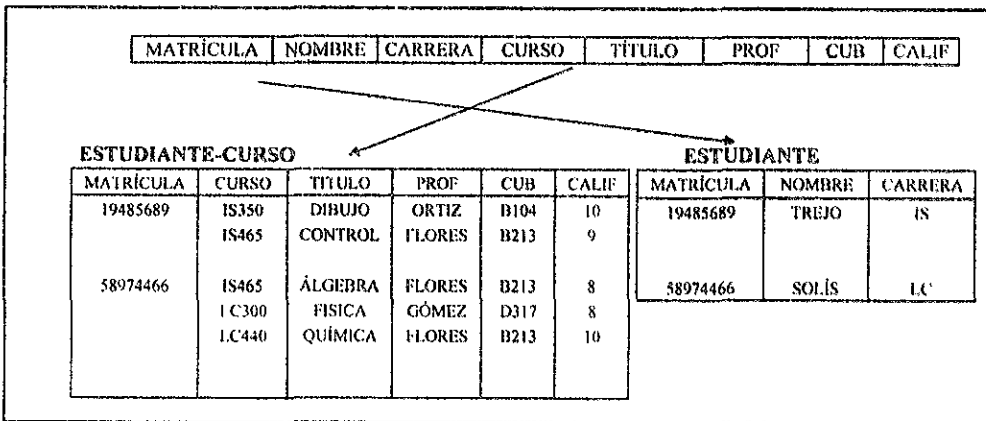


Figura II.28 Conversión de BOL_CALIF a primera forma normal

Las dos nuevas relaciones son las siguientes:

1.- La relación ESTUDIANTE, que contiene los atributos que no son parte del grupo repetitivo: MATRÍCULA, NOMBRE y CARRERA. La llave primaria de esta relación es MATRÍCULA.

2.- La relación ESTUDIANTE-CURSO, que tiene los atributos del grupo repetitivo. La llave primaria de esta relación es una llave compuesta: MATRÍCULA más CURSO. MATRÍCULA es la llave primaria de la primera relación, mientras que CURSO es un atributo que identifica a cada curso de un grupo repetitivo para un estudiante dado. Es necesario una llave compuesta ya que ambas identifican de manera única la CALIFICACIÓN del estudiante.

La relación ESTUDIANTE-CURSO está en primera forma normal.

Aunque la relación está en primera forma normal, aún no es representación ideal de los datos. Revisando se puede encontrar una redundancia considerable. Si se deja así, se tendrían anomalías, inconsistencia al insertar, borrar o actualizar los datos. La razón de estas anomalías es que varios atributos no llave dependen sólo de CURSO y no de la llave primaria completa MATRÍCULA+CURSO.

La CALIFICACIÓN es el único atributo que depende de ambas llaves. Esto es porque se deben saber el estudiante y el curso que toma para determinar la calificación. Un atributo que depende de una llave compuesta se llama totalmente dependiente de esa llave.

Los demás atributos no llave NOMBRE-CURSO, PROG y CUB dependen sólo de CURSO y no de la combinación MATRÍCULA+CURSO. A estos atributos se les llama parcialmente dependientes de la llave primaria.

Segunda forma normal.

Una relación está en segunda forma normal si se asegura que todos los atributos no llave sean completamente dependientes de la llave primaria, es decir, todos los atributos serán funcionalmente dependientes de la llave primaria.

Para convertir una relación con dependencias parciales a la segunda forma normal, se crean dos nuevas relaciones, una con los atributos totalmente dependientes de la llave primaria y otra con los atributos dependientes de una de las partes de esa llave. (Figura II.29)

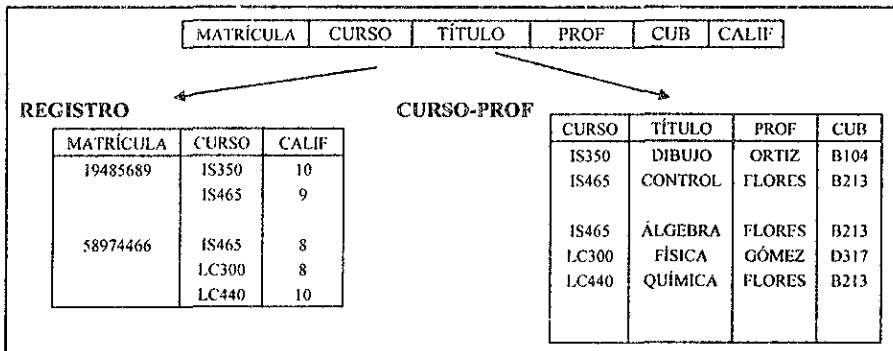


Figura II.29 Convirtiendo a segunda forma normal

Para nuestro ejemplo se crean las siguientes relaciones:

1.- Una relación REGISTRO, con la llave compuesta MATRÍCULA+CURSO. El atributo no llave CALIF es totalmente dependiente de la llave primaria.

2.- Una relación CURSO-PROF, con la llave primaria CURSO. Los atributos no llave NOMBRE-CURSO, PROF y CUB son los que sólo dependen de CURSO.

Se eliminan las anomalías descritas para la primera forma normal. Ahora cada curso está una sola vez en la relación CURSO-PROF. Como resultado, cualquier actualización a los datos del curso es un solo registro. También, ya que los datos de curso están aparte de los estudiantes, se pueden insertar o borrar sin referirse a los datos del estudiante.

Aunque la segunda forma normal representa una mejora, se requiere algunas adicionales, ya que hay datos del problema "escondidos" en la relación CURSO-PROF.

Cada atributo no llave depende de curso. Sin embargo, CUB también depende de PROF. Esto es, sólo hay un lugar (número de cubículo) para cada profesor. Esto es un ejemplo de dependencia transitiva.

Tercera forma normal

Aquí se deben eliminar todas las dependencias transitorias, es decir, aquella en la que los atributos no llave son dependientes de otros atributos no llave.

El proceso de eliminación de una dependencia transitiva se muestra a continuación: (Figura II.30)

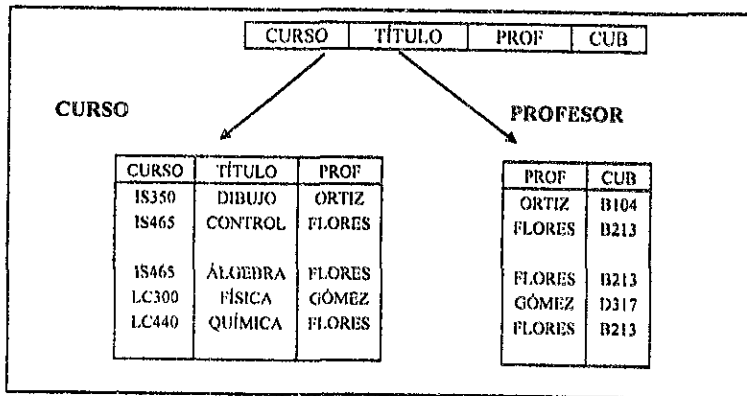


Figura II.30 Eliminación de dependencias transitivas

Los atributos que participan en la dependencia transitiva PROF y CUB se eliminan para formar una nueva relación PROFESOR. La llave primaria de esta relación es PROF ya que, se supone que este atributo identifica de manera única a su oficina CUB.

Aunque el atributo PROF se convierte en la llave primaria de la relación PROFESOR, también es un atributo no llave de la nueva relación CURSO. Se dice que PROF es una llave foránea de la relación CURSO. Esto permite asociar un curso en particular con el instructor que lo imparte. Las relaciones resultantes son: (Figura II.31)

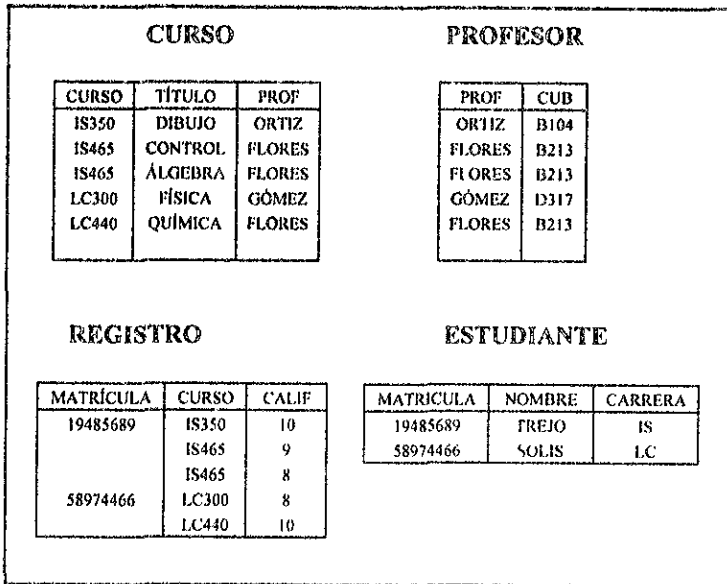


Figura II.31 Ejemplo que cumple con la primera, segunda y tercera formas normales.

Finalmente, se debe tener en cuenta que una vez que se tiene un modelo en tercera forma normal, éste puede ser desnormalizado, y representa un gran beneficio para la inserción, supresión y actualización de la información de la base de datos.

II.5 ¿Porqué utilizar Bases de Datos?

¿Porqué una empresa debe optar por almacenar sus datos de operación en una base de datos integrada?. La respuesta general a esta pregunta es que un sistema de base de datos proporciona a la empresa un *control centralizado* de sus datos de operación que constituye una de sus activos más valiosos. Esto contrasta de manera aguda con la situación que prevalece actualmente en muchas empresas, donde a menudo cada aplicación tiene sus propios archivos de modo que los datos de operación se hallan muy dispersos y, por tanto, es probable que sean difíciles de controlar. Lo anterior implica que en una empresa que utilice un sistema de bases de datos debe existir una persona específica cuya

responsabilidad central sea controlar los datos de operación. Esta persona es el administrador de bases de datos (DBA). Considerando ahora algunas de las ventajas de tener un control centralizado de los datos:

- ✓ Puede reducirse la redundancia.

En sistemas de no usan bases de datos, cada aplicación tiene sus propios archivos privados. Esto a menudo origina enorme redundancia en los datos almacenados, así como desperdicio resultante del espacio de almacenamiento. No se pretende que toda la redundancia por fuerza habrá de eliminarse. A veces hay sólidas razones comerciales o técnicas para mantener múltiples copias de los mismos datos. En un sistema de base de datos, sin embargo, debe controlarse, es decir, el sistema debe estar al tanto de la redundancia y asumir la responsabilidad de propagar las actualizaciones.

- ✓ Puede evitarse la inconsistencia.

Supongamos un hecho real –por ejemplo el hecho de que el empleado E3 trabaja en el departamento D8 –, se representa por dos entradas distintas en la base de datos, y que el sistema no está al tanto de esta duplicación. Habrá entonces en algunas ocasiones en que las dos entradas no concuerden (es decir, cuando solo una de ellas se haya actualizado). En tales circunstancias se dice que la base de datos es inconsistente. Desde luego, una base de datos que se halle en estado de inconsistencia puede suministrar información incorrecta o contradictoria.

- ✓ Los datos pueden compartirse.

Esto no sólo significa que las aplicaciones existentes pueden compartir los datos de la base de datos, sino también que es factible desarrollar nuevas aplicaciones que operen con los mismos datos almacenados. En otras palabras, las necesidades de datos de las nuevas aplicaciones pueden atenderse sin tener que crear nuevos archivos almacenados.

- ✓ Pueden cumplirse las normas establecidas.

Con un control central de la base de datos, el DBA puede garantizar que se cumplan todas las formas aplicables a la representación de los datos. Las normas aplicables pueden comprender la totalidad o parte de lo siguiente: normas de la compañía, de instalación, departamentales, industriales, nacionales o internacionales. Es muy deseable unificar los formatos de los datos almacenados como ayuda para el intercambio o migración de datos entre sistemas.

- ✓ Pueden aplicarse restricciones de seguridad.

Al tener jurisdicción completa sobre los datos de operación, el DBA puede: a) asegurar que el único medio de acceder la base de datos sea a través de los canales establecidos y, por tanto, b) definir controles de autorización para que se apliquen cada vez que se intente el acceso a datos sensibles. Diferentes controles pueden establecerse para cada tipo de acceso (recuperación, modificación, eliminación, etc.) a cada parte de la información de la base de datos.

- ✓ Puede conservarse la integridad.

El problema de la integridad es garantizar que los datos de la base de datos sean exactos. La inconsistencia entre dos entradas que representan al mismo hecho es un ejemplo de falta de integridad (que por supuesto sólo ocurre si existe redundancia en los datos almacenados).

- ✓ Pueden equilibrarse los requerimientos contradictorios.

Cuando los requerimientos globales de la empresa -en contraste con los requerimientos de cualquier usuario individual - el DBA puede estructurar el sistema de base de datos para brindar un servicio que sea el mejor para la empresa en términos globales. Por ejemplo, puede elegirse una representación de los datos almacenados que ofrezca rápido acceso a las aplicaciones más importantes a costa de un desempeño de menor calidad en algunas otras aplicaciones.

Análisis y determinación de requerimientos

III.1 Objetivo General

Automatizar el control docente y académico con el fin de agilizar y optimizar el manejo de la información en el Departamento de Sistemas de la División de Estudios de Posgrado de la Facultad de Ingeniería.

III.2 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 orienta su proceso de enseñanza-aprendizaje hacia la docencia e investigación, de acuerdo a las necesidades científicas y tecnológicas que demanda la sociedad. De estas formas es posible integrar grupos de profesores e investigadores con alumnos que realicen trabajos originales y de alta calidad.

> Estudios de Maestría

Los propósitos de la maestría son brindar a los alumnos una cultura científica y tecnológica, además de una formación metodológica que los capacite para resolver problemas dentro de su área de especialidad, tanto en la investigación como en la docencia. El nivel de maestría se considera como un programa que profundiza los conocimientos del nivel licenciatura, abre perspectivas y acentúa los aspectos formativos, a la vez que desarrolla en el profesional una amplia y alta capacidad innovadora.

> Estudios de Doctorado

Las funciones de los estudios de doctorado de la DEPFI son preparar al alumno en la investigación original, a la vez que se le proporciona una sólida formación disciplinaria, para el ejercicio profesional o académico del más alto nivel.

Maestría Flexible

Por acuerdo del Consejo Universitario, el 6 de diciembre de 1994, se aprobó la Maestría Flexible, mediante la cual la Universidad Nacional Autónoma de México otorga el grado de **Maestro en Ingeniería**, en el que se especifica un **campo mayor**, elegido por el alumno bajo las siguientes opciones:

- Estructuras
- Mecánica de Suelos
- Construcción
- Hidráulica
- Ingeniería Ambiental
- Ingeniería Eléctrica
- Ingeniería Mecánica
- Ingeniería Petrolera
- Exploración de recursos energéticos del subsuelo
- Ingeniería Energética
- Planeación
- Investigación de Operaciones
- Transporte

Es una maestría flexible porque proporciona:

1. un programa de estudios personalizado para cada alumno, seleccionando entre una amplia variedad de posibilidades;
2. una autoridad académica, el tutor, para diseñar el plan de estudios personalizado y obligatorio para el alumno, y avalar su desempeño académico;
3. una gran flexibilidad para adaptarse y dar respuesta a los acelerados cambios científicos y tecnológicos;
4. los elementos del conocimiento acordes con los intereses del alumno y con su proyección profesional futura.

› *Plan de Estudios*

Está organizado por áreas y por diversos campos del quehacer profesional de la ingeniería, lo que permite una estructura académica individualizada en la que, a partir de los conceptos adquiridos en el área básica y de las asignaturas definidas como obligatorias en el programa particular, aunadas al programa específico y dirigido de investigación, el alumno integrará los conocimientos teóricos y prácticos que comprenden su plan de maestría.

Áreas y campos de la ingeniería.

Áreas	Campos
Ingeniería Civil	<ul style="list-style-type: none"> > Construcción > Estructuras > Hidráulica > Ingeniería Ambiental > Mecánica de Suelos
Ingeniería Electromecánica	<ul style="list-style-type: none"> > Ingeniería Eléctrica > Ingeniería Mecánica
Ingeniería de Recursos Energéticos y Minerales	<ul style="list-style-type: none"> > Energéticos del Subsuelo > Exploración de Recursos > Ingeniería Energética > Ingeniería Petrolera
Ingeniería de Sistemas	<ul style="list-style-type: none"> > Investigación de Operaciones > Planeación > Transporte

Para nuestros propósitos, únicamente nos enfocaremos al Área de Sistemas, de la cual el *Departamento de Ingeniería de Sistemas* es responsable.

- Campo de Planeación. Analiza y diseña métodos, procesos y organizaciones para la solución anticipada de problemas en la gestión pública y privada.
- Campo de Investigación de Operaciones. Procura aplicar el método científico para la solución óptima de problemas de asignación, transporte, almacenamiento y utilización de recursos de toda índole.
- Campo de Transporte. Tiene como finalidad la planeación y operación del sistema de transporte, planteando alternativas de solución en las que se concilian aspectos técnicos de ingeniería, con factores económicos, sociales y de gestión.

Los créditos del Plan de Estudios a cubrir serán los siguientes:

1. Asignaturas del campo básico.

Estas actividades tienen como función, proporcionar al alumno los conocimientos y habilidades fundamentales, principalmente en matemáticas, indispensables para abordar el estudio de los diversos problemas de la ingeniería en cualquiera de sus áreas. El alumno debe elegir cursar al menos una de estas asignaturas.

2. Asignaturas del campo mayor.

Tienen como objetivo proporcionar elementos teóricos y técnicos, tanto generales como específicos, adecuados y útiles para el estudio y solución problemas del campo de la ingeniería seleccionado por el alumno. El alumno debe inscribirse en un campo mayor y elegir de acuerdo con su tutor, cuando menos seis asignaturas de éste.

3. Actividades de Investigación.

a) Seminario de Investigación. El tutor orienta al alumno a realizar investigación documental y tareas prácticas sobre el tema seleccionado, así como en la definición de cuáles deberán ser las actividades de apoyo.

b) Trabajo de Investigación. El alumno integrará los conocimientos y habilidades adquiridos tanto en el campo básico como en el campo mayor mediante el diseño y realización de un proyecto de investigación para el estudio de un problema particular de la ingeniería.

4. Asignaturas optativas o del campo menor. Estas asignaturas complementan la formación del alumno e incorporan a los estudios los contenidos de actualidad, en los diferentes campos de la ingeniería. El alumno deberá elegir al menos tres asignaturas optativas, de acuerdo con su tutor, pudiendo ser éstas del campo mayor elegido o de otros campos.

De acuerdo con las actividades académicas que el alumno deberá desarrollar, los créditos por cubrir son los siguientes:

Estructura	Créditos	# aprox. de asignaturas
Asignaturas de campo básico	6	1
Asignaturas de campo mayor	36	6
Asignaturas optativas o de campo menor	18	3
Asignaturas de actividades de investigación (desarrollo de tesis)	16	3
	60	13

III.3 Definición del Problema

En los últimos años el Departamento de Ingeniería de Sistemas de la División de Estudios de Posgrado de la Facultad de Ingeniería (DEPFI) ha sufrido un incremento considerable de su población estudiantil, personal docente y administrativo; teniendo como consecuencia un aumento en cuanto a volumen de información se refiere.

Debido a esto, el Departamento se ve en la imperiosa necesidad de diseñar e implantar un Sistema de Cómputo que controle de manera eficiente y rápida la información, por lo cual se decide la realización del Sistema de Control Académico para el Departamento de Sistemas de la DEPFI. (*SICAP*)

> **Registro de aspirantes.**

El alumno deberá cubrir una serie de requisitos para poder cursar una de las maestrías del área de sistemas de la DEPFI. Estos requisitos son, entre otros:

- ✓ Poseer título profesional
- ✓ Tener promedio mayor o igual a 8.0
- ✓ Aprobar los exámenes de admisión: Matemáticas, Probabilidad y Estadística.
- ✓ Aprobar en los dos primeros semestres los requisitos sin crédito académico y antecedentes de los cursos regulares de los programas de estudio:
 - Redacción Técnica
 - Un idioma (inglés)
 - Economía
 - Programación de computadoras

Después de haber cubierto los requisitos antes mencionados y una vez que el aspirante es aceptado se procede a registrarse e inscribirse al posgrado. El registro de los alumnos al posgrado se realiza mediante una forma o solicitud de registro la cual es llenada manualmente por el alumno.

La información de interés que contiene la forma es la siguiente:

- Datos Generales
- Antecedentes Académicos de Licenciatura
- Antecedentes Académicos de Posgrado
- Datos de Inscripción
- Datos de beca

Para esclarecer los puntos anteriores se requiere de información más detallada:

a) Datos Generales

- N.- de Cuenta
- N.- de Expediente
- Nombre
- Dirección
- Teléfono domicilio
- Teléfono oficina y extensión
- Sexo
- Nacionalidad
- Fecha de nacimiento

b) Antecedentes Académicos de Licenciatura

- Carrera cursada
- Escuela o facultad donde realizó sus estudios
- Institución donde realizó sus estudios
- País donde se encuentra la institución
- Estado donde se encuentra la institución
- Promedio
- Fecha de titulación

c) Antecedentes Académicos de Posgrado

- Plan en el que estuvo inscrito
- Escuela o facultad donde realizó sus estudios
- Institución donde realizó sus estudios
- País donde se encuentra la institución
- Estado donde se encuentra la institución
- Indicador de culminación de créditos
- Indicador de obtención del grado
- Fecha de titulación

d) Datos de Inscripción

- Año y semestre de ingreso
- Clave de la escuela o facultad
- Clave del plan al que se inscribe
- Indicador de entrega de constancia de idiomas
- Nombre del Asesor o Tutor Académico

e) Datos de Beca

Aquí se indica qué institución otorga la beca al alumno, las opciones son:

- Sin beca
- UNAM
- CONACYT
- Instituciones Gubernamentales
- Fundación o Asociación
- Gobierno extranjero
- Exento de pago UNAM
- Becado pero no específica

➤ **Inscripción o reinscripción de alumnos a la DEPFI.**

Además de llenar la forma de registro al Posgrado, el alumno debe inscribirse (por primera vez) o reinscribirse (dos veces al año) al inicio de cada semestre escolar, para lo cual el alumno debe llenar en el caso de inscripción una segunda forma conocida como Solicitud de Inscripción que contiene lo siguiente:

- Datos generales del alumno
- Datos generales del plan o área a la que se inscribe
- Asignaturas a las que se inscribe

III.4 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 ó 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 consiguiente, el primer paso del 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 que depende si el sistema está orientado hacia transacciones, toma de decisiones ó se extiende a varios departamentos.

En la antigüedad, el uso de los sistemas era muy completo ya que requería que el usuario fuese un experto en el área para poder los pasos a seguir 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 la necesidad de memorizar instrucciones.

Con la introducción de computadoras personales, el desarrollo de sistemas tomó un giro importante ya que ahora el usuario no necesita ser un experto en sistemas, si no todo lo contrario, el usuario es una persona común que no tiene idea de lo que es un sistema de cómputo y que sólo se limita a hacer uso de la computación 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 *SICAP*, se recurrió directamente al personal encargado del manejo de información y estos fueron sus requerimientos:

- Para los alumnos de la DEPEFI que cursan los estudios de maestría en el área de ingeniería de sistemas en sus tres diferentes campos: Investigación de Operaciones, Planeación y Transporte, es necesario conocer datos de carácter personal como Domicilio, Nacionalidad, Fecha de Nacimiento, Antecedentes académicos de licenciatura, Antecedentes académicos de posgrado, si los tiene, si cuenta o no con una beca, el profesor que será su tutor, etc. También es necesario llevar un control de la propuesta académica que el alumno, de acuerdo con su tutor, pretenda cubrir durante su maestría.
- Es necesario, además, hacer una distinción entre los alumnos que provengan de la UNAM ó alumnos que provengan de alguna otra dependencia, ya que para este caso, se les deberá asignar un número de cuenta provisional. También es indispensable asegurarse de la validez del número de cuenta del alumno de la UNAM.
- Para el personal académico del Departamento, se requiere conocer sus datos personales.
- Saber las clases que un profesor imparte en un semestre determinado.
- Proporcionar a los profesores, una lista de alumnos inscritos en cada una de las clases que imparten.
- Determinar el número de grupos que se abren en cada semestre de una materia específica.
- Saber todos los grupos que se abren en un semestre determinado.
- Proporcionar al alumno un comprobante de inscripción actual, así como un Kárdex que registre su historia académica.
- Datos estadísticos sobre alumnos por sección, por tipo de beca, etc.

- Generar informes que den a conocer a los alumnos, el profesor que será su tutor durante su estancia en la maestría, además de proporcionar a cada tutor la lista correspondiente de los alumnos bajo su tutela.

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 capacidad de 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 voluminosos y complejos quizá.
- las pantallas sean legibles y sencillas de comprender.
- las especificaciones de hardware y software del sistema a desarrollar, estén acordes con el equipo de cómputo con el que el Departamento cuenta.
- la probabilidad de operación libre de fallos en un entorno determinado, y durante un tiempo específico del sistema sea mínima. (Fiabilidad)
- el sistema sea seguro, es decir, que se pueda controlar el acceso a la información por personal no autorizado. (Integridad)
- el sistema presente facilidad de comprender, corregir, adaptar y/o mejorar el software. (Facilidad de mantenimiento)
- el sistema sea portable (esfuerzo requerido para transferir el programa desde un hardware y/o un entorno de sistemas de software a otro)
- el sistema presente eficiencia en la ejecución.
- el sistema tenga facilidad de expansión

III.5 Propuesta de Solución.

Los puntos a considerar para la propuesta de solución, son los siguientes:

- El equipo de cómputo con que cuenta el Departamento está realmente limitado.
- El sistema será utilizado por personas con poca experiencia en cuanto al uso de herramientas computacionales.
- Por la interfase gráfica tan sencilla y amigable que presenta al usuario, el sistema a desarrollar trabajará sobre ambiente Windows.
- El Manejador de Bases de Datos que se seleccione, deberá estar disponible comercialmente y ser capaz de manipular fácil y rápidamente una gran cantidad de información

III.5.1 Elección de la base de datos

En el apéndice A de este trabajo se mencionan las características del Manejador de Bases de Datos Microsoft Access. Este manejador nos proporciona grandes ventajas como son:

- Facilidad de uso
- Microsoft Access permite desarrollar una interfase gráfica muy similar a la que nos ofrece el ambiente de trabajo Windows y con esto brindamos al usuario un método consistente de interacción con la computadora.
- Un usuario sin mucha experiencia en el desarrollo de sistemas, puede en determinado momento entender el diseño de la base de datos.
- La facilidad de compartir información entre los demás productos de Microsoft, como son Word y Excel.
- Al igual que la gran mayoría de los programas manejadores de bases de datos, Microsoft Access nos proporciona un lenguaje de programación denominado Access Basic que tiene muchas características del lenguaje Visual Basic. La utilización principal de este lenguaje es crear aplicaciones y funciones adaptables a nuestras necesidades. La metodología de este lenguaje nos permite trabajar con la "Programación Orientada a Eventos".

Microsoft Access 2.0 tiene una gran disponibilidad comercial y sus requerimientos de hardware y software son fáciles de cubrir.

Por todo lo anterior, se tomó la decisión de elaborar este sistema de información en Microsoft Access 2.0

III.6 Especificaciones del diseño.

El diseño de sistemas presenta dos etapas: *el diseño lógico y la construcción física*. Cuando se formula el *diseño lógico*, se describen características tales como: salidas, entradas, archivos, bases de datos y procedimientos. El conjunto formado por todas estas características recibe el nombre de *especificaciones de diseño*.

La construcción física, produce el software, los archivos y un sistema que funciona. Durante el diseño, los requerimientos del usuario se trasladan en características del sistema.

Se dice que un sistema de información satisface las necesidades del usuario si cumple con lo siguiente:

- Realiza en forma apropiada los procedimientos correctos.
- Presenta información e instrucciones en una forma aceptable y efectiva.
- Produce resultados exactos.
- Proporciona una interfase y métodos de interacción aceptables.
- Es percibido por los usuarios como un sistema confiable.

Elementos del diseño.

Los componentes de un sistema de información descritos durante el análisis de requerimientos, son el punto focal del diseño de sistemas. Los elementos a diseñar son los siguientes:

Flujos de datos: Los movimientos de datos hacia, o alrededor del sistema.

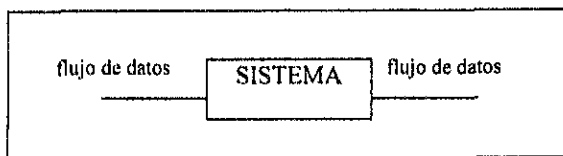


Figura III.1

Almacenes de datos: Conjuntos temporales permanentes de datos

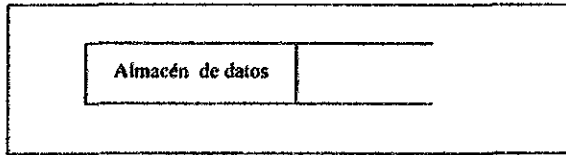


Figura III.2

Procesos: Métodos y rutinas para utilizar el sistema de información y lograr con ello los resultados esperados.

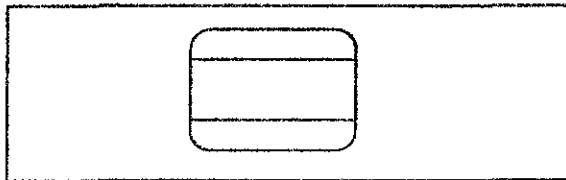


Figura III.3

Controles: Estándares o lineamientos para determinar si las actividades están ocurriendo en la forma anticipada o aceptada, es decir, si se encuentran "bajo control". Así mismo, debe especificar las acciones que deben emprenderse cuando ocurren problemas o se presentan circunstancias inesperadas.

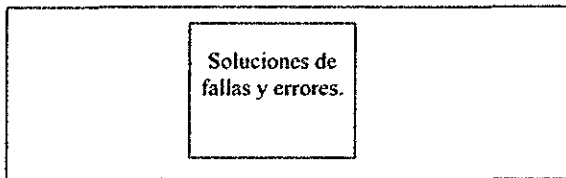


Figura III.4

Funciones del personal: Las responsabilidades de todas las personas tienen que ver con el nuevo sistema, incluyendo usuarios y personal de apoyo. A menudo las funciones del personal se establecen en forma de procedimiento.

Diseño de la salida.

El término *salida*, se refiere a los resultados e información generados por el sistema. Para muchos usuarios finales, la salida es la única razón para el desarrollo del sistema y la base sobre la que ellos evaluarán la utilidad de la aplicación. En realidad, muchos usuarios no operan el sistema y tampoco ingresan datos en él, pero sí utilizan las salidas generadas.

Para el diseño de la salida, se debe realizar lo siguiente:

- > Determinar qué información presentar.
- > Decidir si la información será presentada en forma visual, verbal o impresa y seleccionar el medio de salida.
- > Disponer la presentación de la información en un formato aceptable.
- > Decidir cómo distribuir la salida entre los posibles destinatarios.

Para llevar a cabo las actividades antes mencionadas, se requieren decisiones específicas tales como el empleo de formatos ya impresos cuando se preparan reportes, cuántas líneas planear sobre una página impresa o si es conveniente emplear gráficas y colores.

Diseño de la Entrada.

Los puntos a considerar para diseño de entradas son:

1. Qué datos ingresan al sistema
2. Qué medios utilizar.
3. La forma en que se deben disponer o codificar los datos.
4. El diálogo que servirá de guía para dar entrada a los datos.
5. Validación necesaria de datos y transacciones para detectar errores.
6. Métodos para llevar a cabo la validación de las entradas y los pasos a seguir cuando se presentan errores.

Las decisiones para en manejo de entradas, especifican la forma en que serán aceptados los datos para su procesamiento por computadora. Se decide si los datos serán proporcionados directamente, quizá a través de una estación de trabajo, ó por el uso de documentos.

Diseño de Archivos

El diseño de archivos incluye decisiones con respecto a la naturaleza y contenido del propio archivo. Entre las decisiones que se toman durante el diseño de archivos, se encuentran las siguientes:

- > Los datos que deben incluirse en el formato de los registros contenidos en el archivo.

- La longitud de cada registro, con base en las características de los datos que contiene.
- La secuencia o disposición de los registros dentro del archivo, la estructura de almacenamiento que puede ser secuencial, indexada o relativa.

Diseño de interacciones con la base de datos.

Muchos sistemas de información, ya sean implantados en sistemas de cómputo grandes o pequeños, interactúan con las bases de datos. Dada la importancia de éstas, su diseño es establecido y vigilado por un administrador de bases de datos, que es una persona (o un grupo de personas) que tienen la responsabilidad de desarrollar y mantener la base de datos.

Diseño de controles.

Es necesario anticiparse a los errores que se cometan al ingresar datos al sistema o al solicitar la ejecución de ciertas funciones. Algunos errores no tienen importancia ni consecuencias, pero otros pueden ser tan serios que ocasionarían el borrado de datos o el uso inapropiado del sistema. Aunque exista sólo la más mínima probabilidad de cometer un error serio, un buen diseño del sistema de información ofrecerá los medios para detectar y manejar el error. Los controles de entrada proporcionan medios para:

1. Asegurar que sólo los usuarios autorizados tengan acceso al sistema.
2. Garantizar que las transacciones sean aceptables.
3. Validar los datos para comprobar su exactitud.
4. Determinar si se han omitido datos que son necesarios.

Diseño de procedimientos.

Los procedimientos especifican qué tareas deben efectuarse al utilizar el sistema y quiénes son los responsables de llevarlas a cabo, entre los procedimientos importantes se encuentran:

- Procedimientos para entrada de datos: Métodos para la captura de datos y su ingreso al sistema de información. (Por ejemplo: secuencia para dar entrada a los datos).
- Procedimientos durante la ejecución: Pasos y acciones emprendidas por los operadores del sistema y en ciertos casos, por los usuarios finales que interactúan con el sistema para alcanzar los resultados deseados.

- Procedimiento para manejo de errores: Acciones a seguir cuando se presentan resultados inesperados.
- Procedimiento de seguridad y respaldo: Acciones para proteger al sistema y sus recursos contra posibles daños.

III.7 Diagrama de flujo de datos

El *diagrama de flujo de datos (DFD)* es una técnica gráfica que presenta el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida. El DFD también es conocido como *grafo de flujo de datos* o como *diagrama de burbujas*.

Se puede usar el diagrama de flujo de datos para representar un sistema o un software a cualquier nivel de abstracción. De hecho, los DFDs pueden ser refinados a niveles que representen 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 de entrada y de salida, respectivamente. Al partir el DFD de nivel 0 para mostrar más detalles, aparecen representados procesos (burbujas) y caminos de flujo de información adicionales. Por ejemplo, un DFD de nivel 1 puede contener cinco o seis burbujas con flechas interconectadas. La notación ¹ que se utilizará para desarrollar el DFD del sistema propuesto, será la que se presenta en la figura III.5, y los diagramas generados se muestran en las figuras III.6, III.7, III.8, III.9 y III.10.

¹ Ingeniería del software, un enfoque práctico.

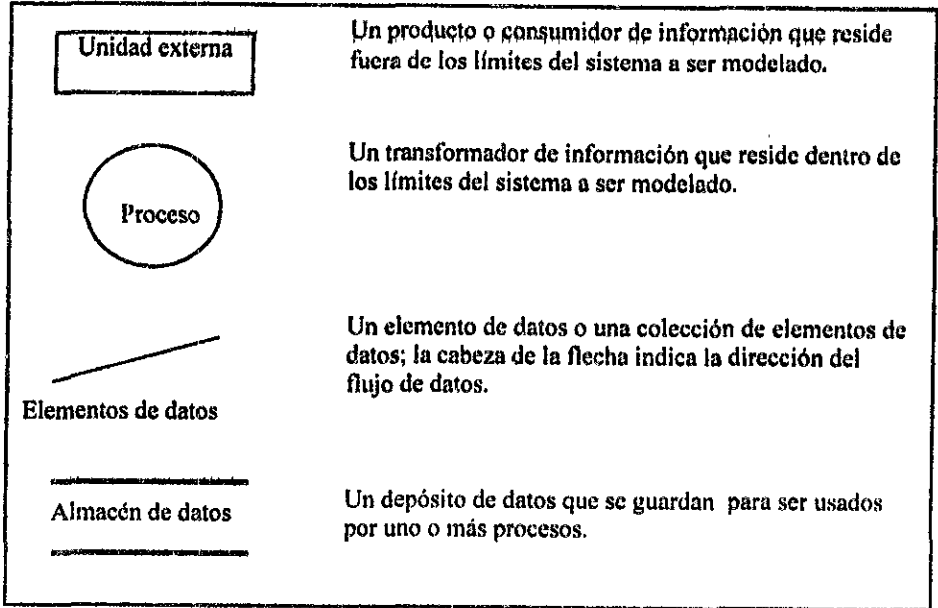


Figura III.5

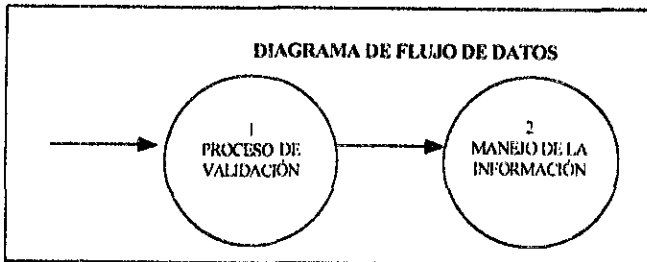


Figura III.6

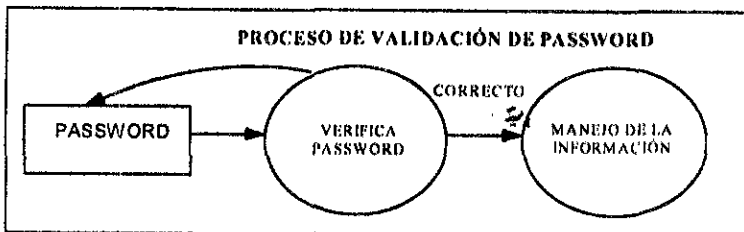


Figura III.7

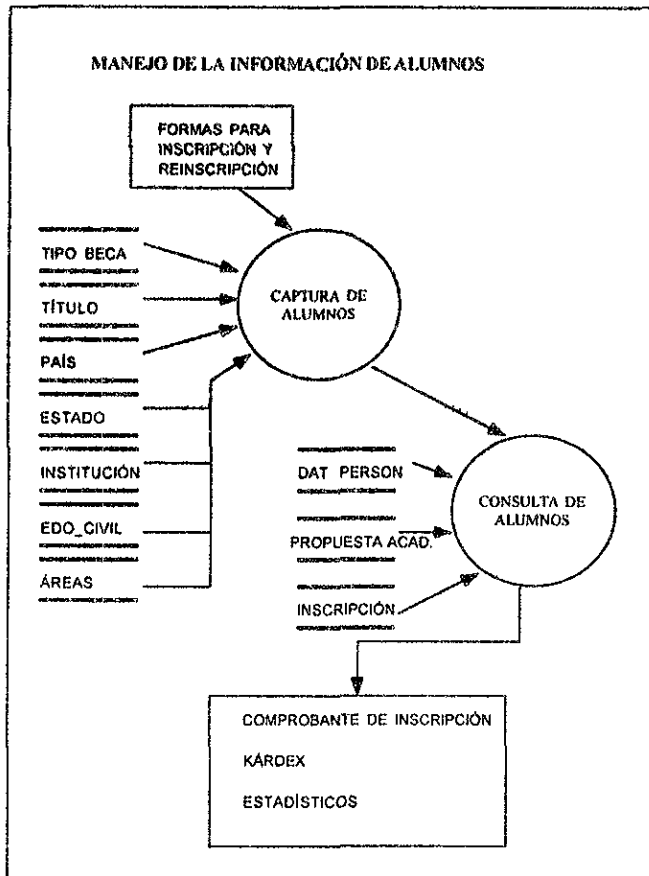
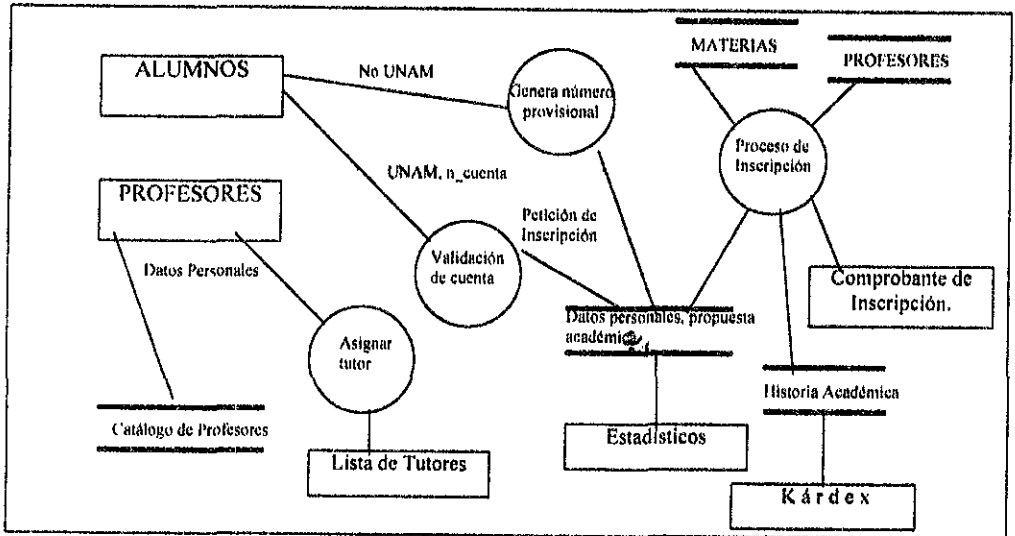
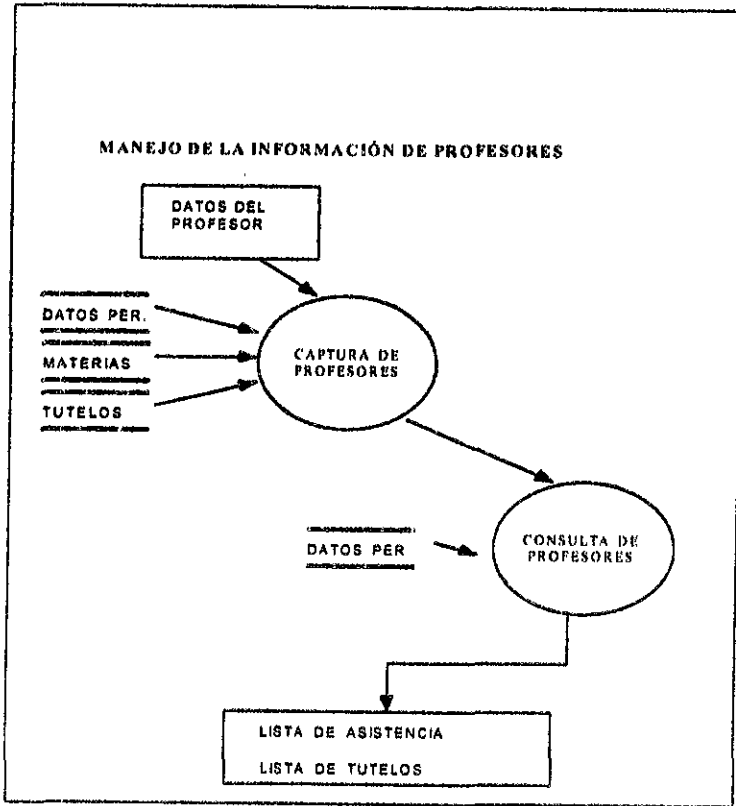


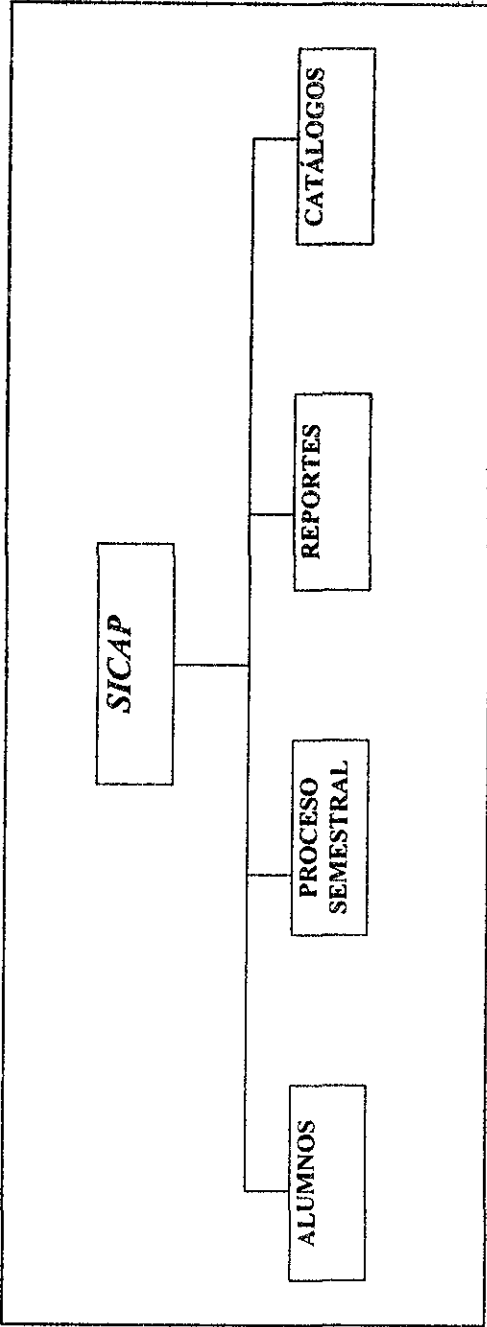
Figura III.8



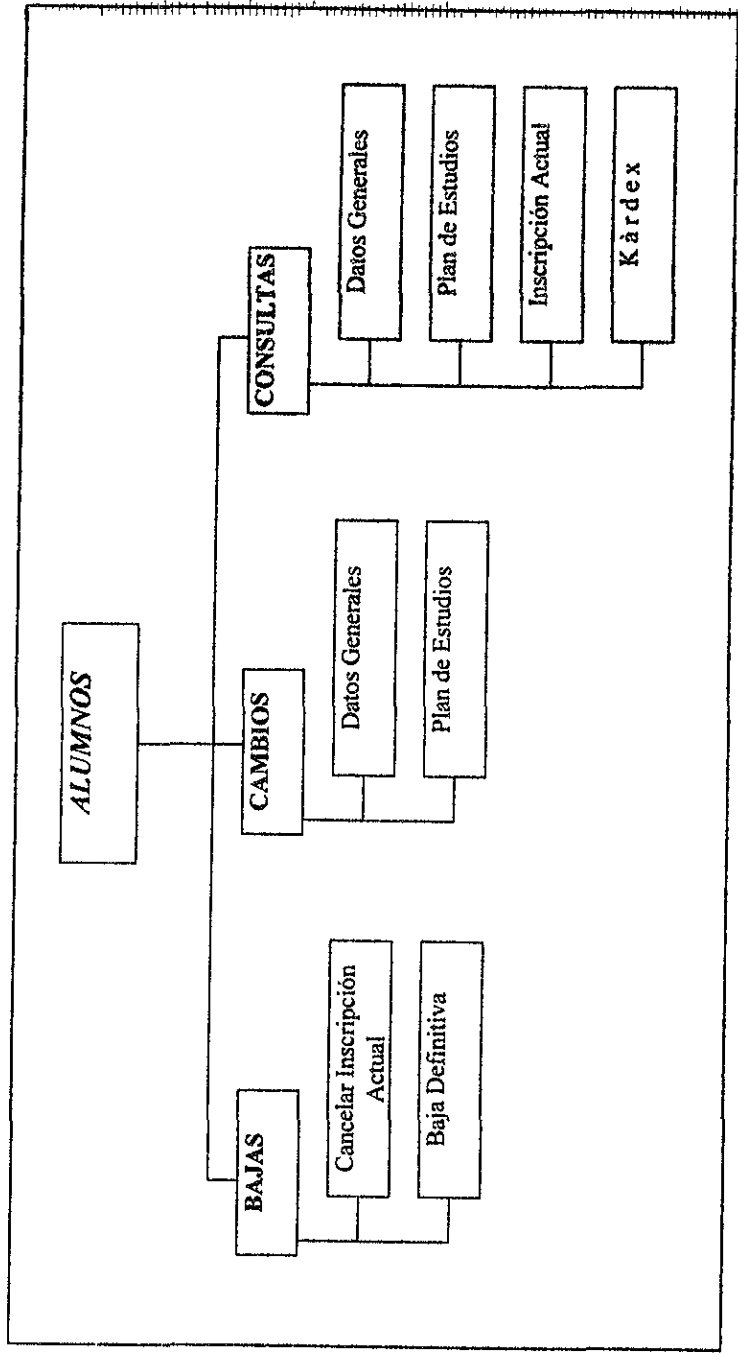
Figuras III.9 y III.10

III.8 Diseño estructurado

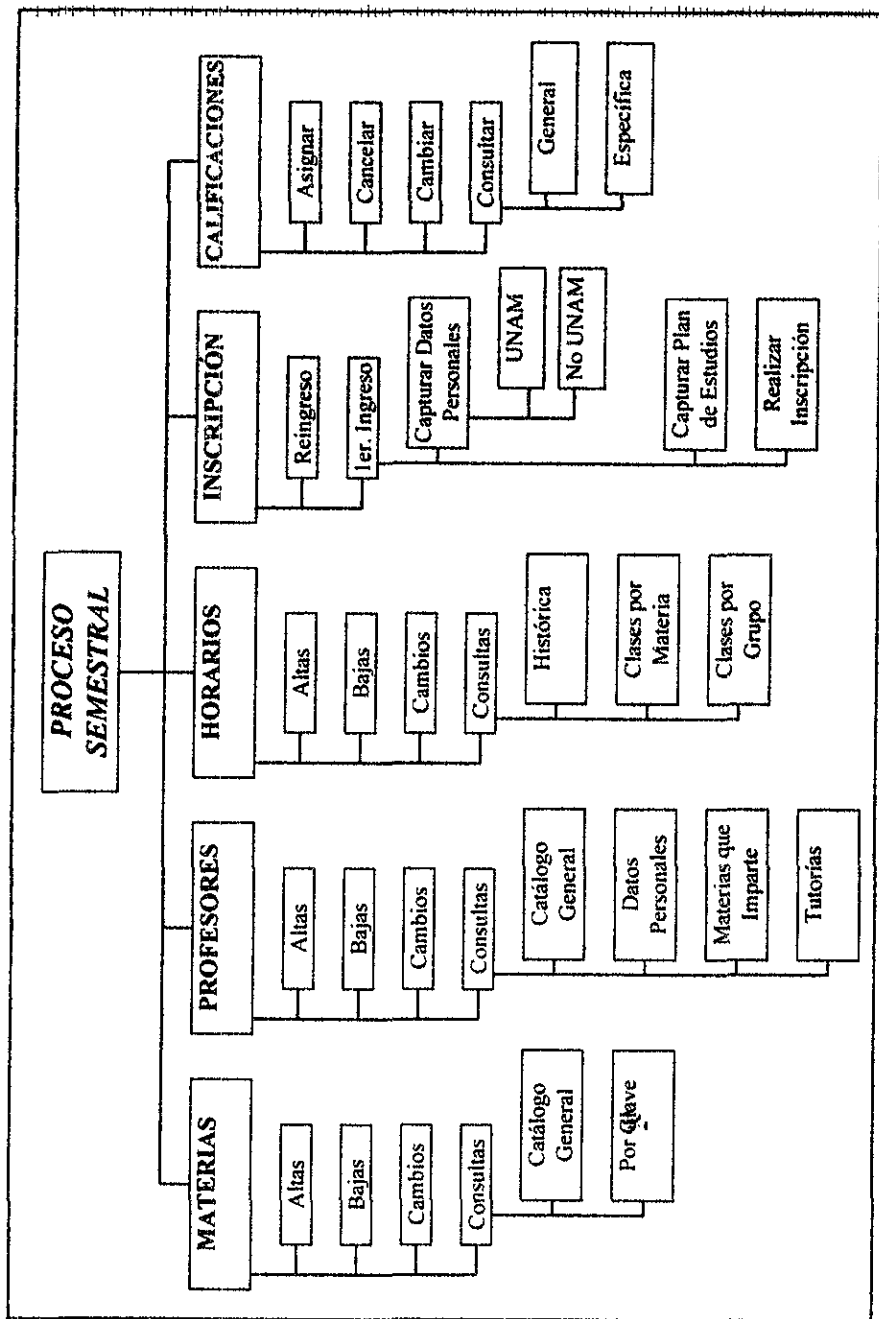
Siguiendo la metodología señalada por el diseño estructurado, a continuación se mostrará la descripción gráfica de cada uno de los módulos que conformarán el **SICAP**.



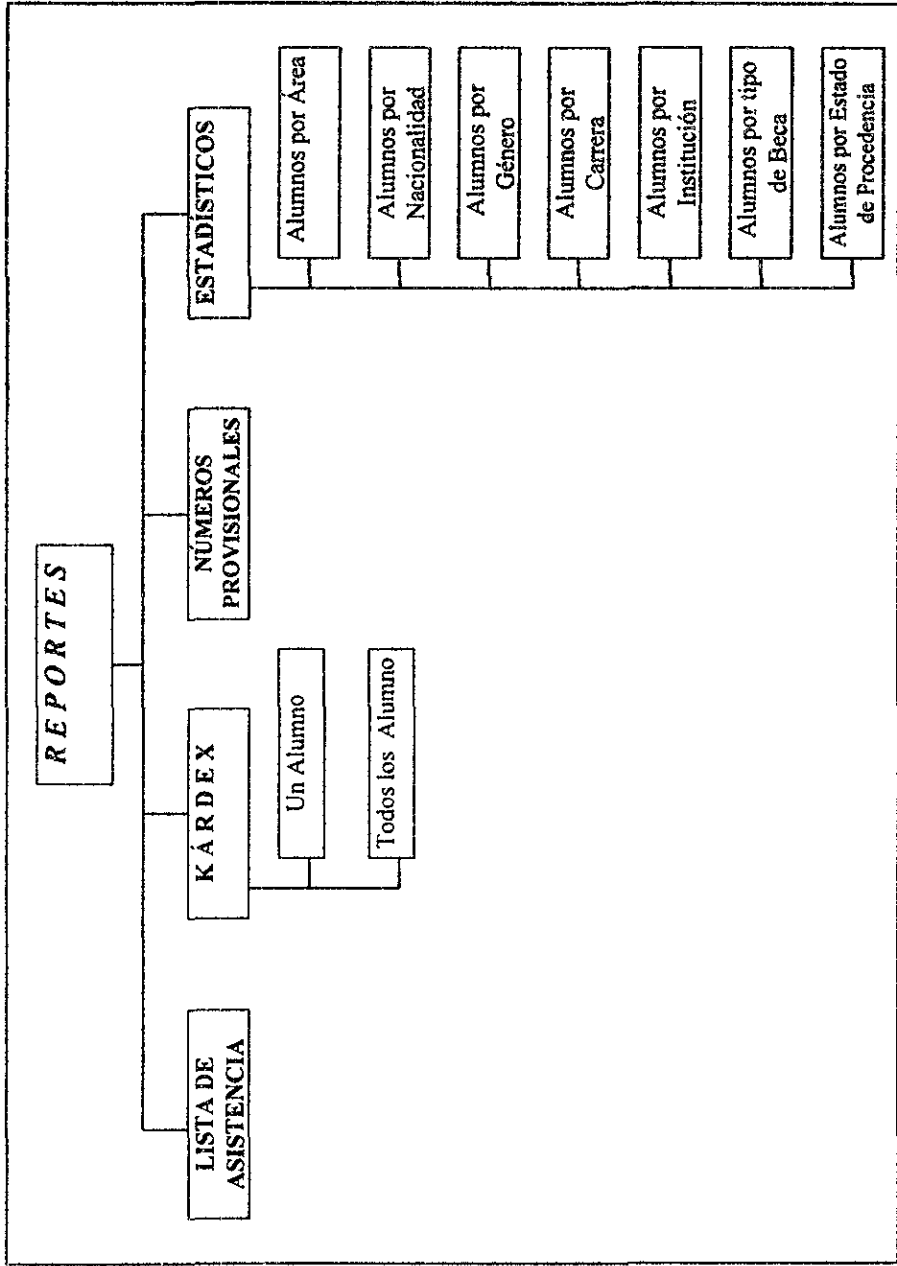
Módulos principales del SICAP

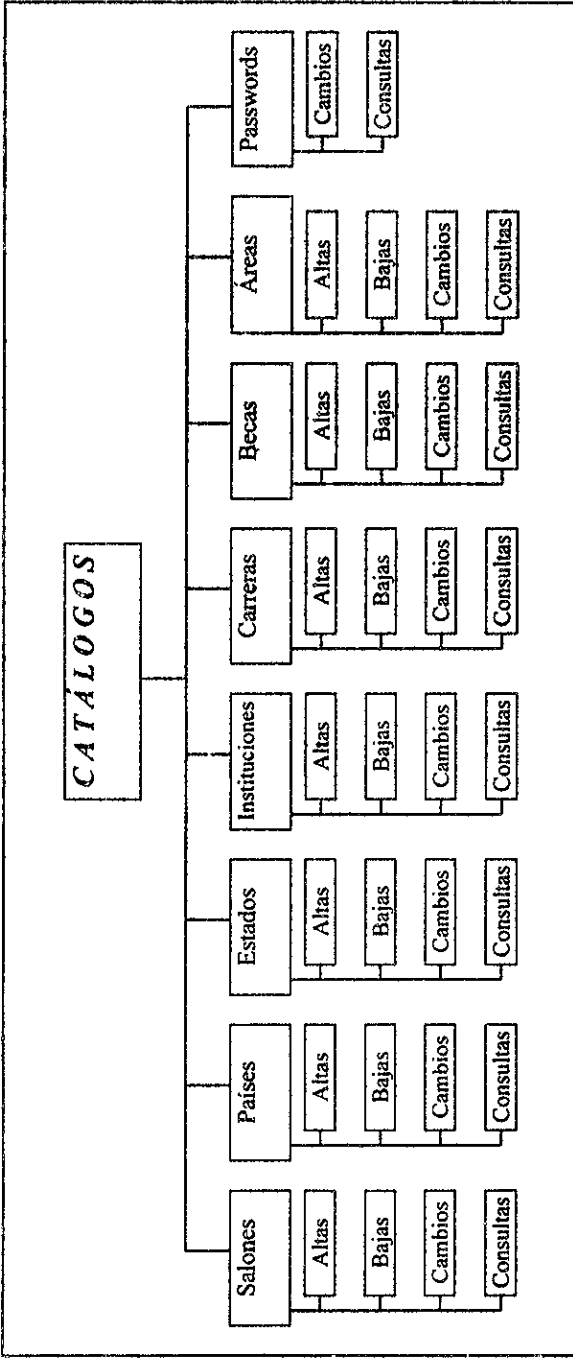


Detalles del Proceso ALUMNOS



Detalles del Proceso SEMESTRAL





ESTA COPIA NO DEBE
SALIR DE LA BIBLIOTECA

C

a

p

i

t

u

l

o

**Diseño e implementación
del sistema**

IV

IV.1 Modelo Entidad-Relación

El modelo Entidad-Relación nos indica que podemos dividir el análisis en tres fases:

- a) Modelo de Entidades
- b) Modelo de Relaciones
- c) Modelo de Atributos

Modelo de entidades

El modelo de entidades se basa en los siguientes pasos:

1. Descubrir entidades: Concentrarse en sujetos, las entidades existen por sí mismas. Un sujeto es el nombre de una persona, de un tipo de persona, de un animal, una planta, un lugar, una cosa, una sustancia o una idea.
2. Definir el alcance de la entidad: Definir si la entidad es de interés para el sistema.
3. Definir una llave primaria (PK). Una llave primaria es un atributo de la entidad que permitirá identificar cada renglón dentro de la tabla que formará, de manera única; sus características son:

ND: No duplicada, garantiza la existencia única de ese elemento.

NN: No nula, garantiza que este dato siempre exista.

NC: No cambia, no permite cambiar este dato.

4. Documentar.
5. Incluirla en el diagrama E-R.

Las entidades identificadas pasarán a ser las tablas de la base de datos.

Por lo tanto, podemos distinguir las siguientes entidades:

- ALUMNO. Alumno de la División de Estudios de Posgrado de la Facultad de Ingeniería del Departamento de Ingeniería de Sistemas, inscrito en el campo de Planeación, Transporte o Investigación de Operaciones
- PROFESOR. Personal de la DEPFI, encargado de la impartición de clases y tutorías en el semestre en curso.
- MATERIA. Asignatura que está contemplada en el plan de estudios de los campos de Planeación, Transporte o Investigación de Operaciones de la DEPFI.
- CLASE. Para cada grupo en el que se imparte una materia, esta relación indica, por cada día de la semana, en que salón se tiene clase, el profesor encargado, el horario correspondiente y el semestre que se está cursando.

En el diagrama E-R, las entidades se representan como rectángulos. (Figura IV.1)

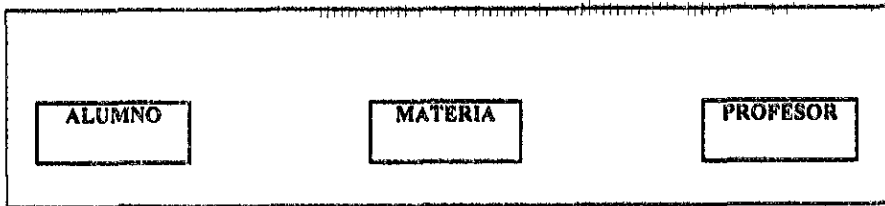


Figura IV.1

Para cada uno de las entidades, definimos ahora un atributo que será la llave primaria:

ALUMNO	_____	N_cuenta
MATERIA	_____	Clv_mat
PROFESOR	_____	RFC

Modelo de relaciones

Una vez identificadas las entidades, las relaciones se obtienen a partir de los siguientes pasos:

1. Descubrir relaciones, concentrarse en los verbos. Un verbo es una palabra que describe un modo de ser, una asociación o un evento. Los verbos describen el estado de los sujetos y relacionan a los sujetos dentro de las situaciones.
2. Definir el alcance de la relación, determinando si la relación es importante para el sistema.
3. Definir el tipo de relación
4. Documentar en el diagrama E-R.
5. Documentar en tablas

Para hacer referencia a las relaciones, plantearemos el siguiente diagrama: (Figura IV.2)

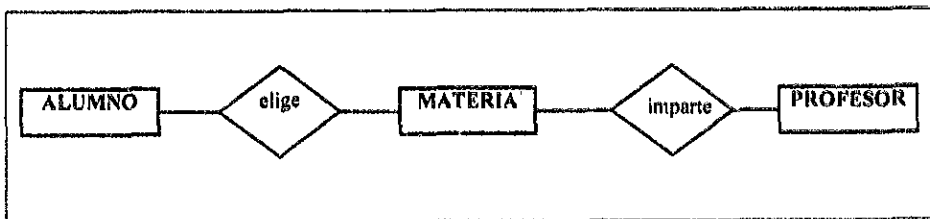


Figura IV.2

Definimos ahora el tipo de relación entre las entidades (cardinalidad): (Figura IV.3)

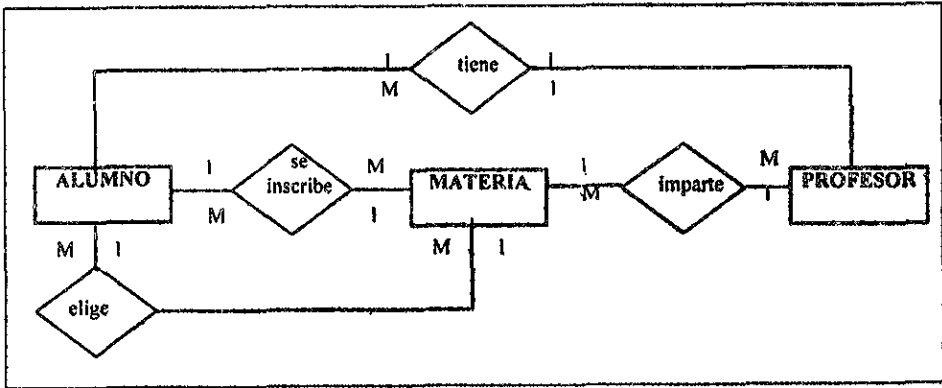


Figura IV.3

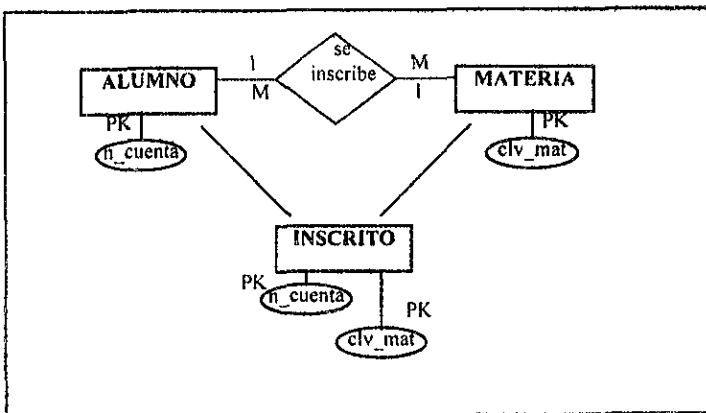
Recordando que:

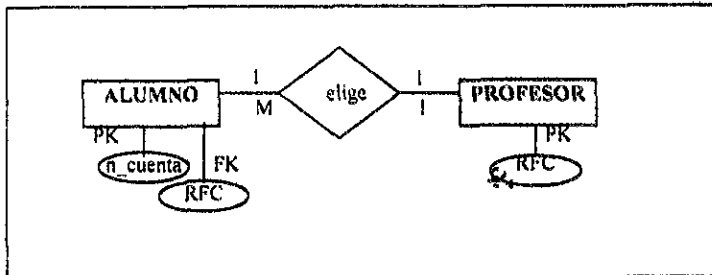
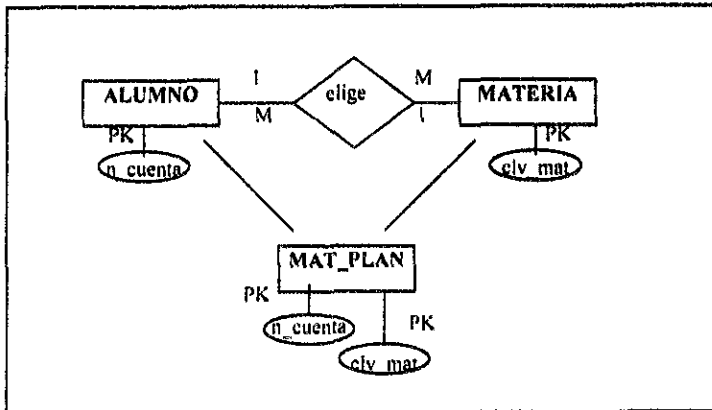
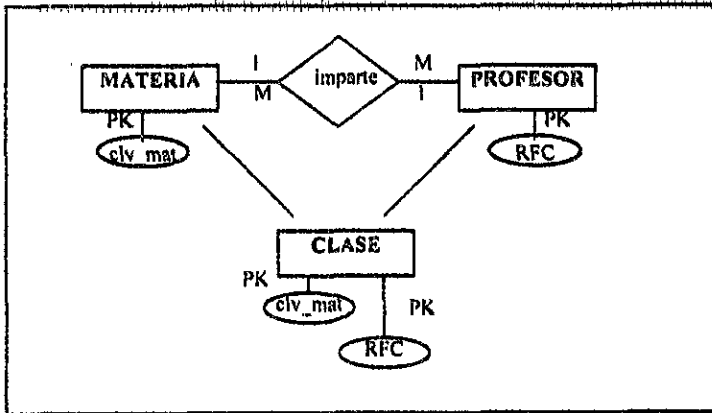
Relación 1 a 1 : La llave primaria de cualquiera de las dos entidades involucradas debe migrar hacia la otra entidad, es indistinto cuál de las dos entidades migra su llave primaria, sin embargo de acuerdo a la utilización de los datos esto puede cambiar.

Relación 1 a M : La llave primaria de la entidad donde sólo se involucra 1 elemento se migra hacia la entidad donde se involucran muchos elementos (M).

Relación M a M : Se debe generar una nueva entidad cuya llave primaria es la combinación de las llaves primarias de las entidades que la originaron.

De acuerdo a lo anterior, hacemos los siguientes cambios:





Modelado de atributos.

1. Descubrir atributos, concentrarse en adjetivos o adverbios (modificadores).
2. Definir el alcance del atributo
3. Documentar 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.

Después de haber aplicado el proceso de normalización, los atributos para cada una de las entidades resultantes, se muestran a continuación:

ALUMNOS

N_CUENTA (PK) El número de cuenta será el atributo que nos ayudará a distinguir de manera única a los alumnos.

N_EXP. Número de expediente

NOM_ALUM. Nombre del Alumno

APE_PAT. Apellido paterno del alumno

APE_MAT. Apellido materno del alumno

EDO_CIV. Estado civil del alumno

DIREC. Calle y número del alumno

COLONIA Colonia donde reside el alumno

ESTADO. Estado donde reside el alumno

CODIGO_P. Código postal del alumno

TEL_DOM. Teléfono del domicilio del alumno

TEL_OFNA. Teléfono de oficina del alumno

CLV_NAC. (FK) Clave de nacionalidad del alumno

SEXO. Sexo del alumno

FECHA_NAC. Fecha de nacimiento del alumno

CLV_TIT (FK). Clave de título o carrera de licenciatura del alumno

CLV_ESC. Clave de la escuela donde el alumno cursó su licenciatura

CLV_INS. (FK). Clave de la institución donde el alumno cursó su licenciatura

CLV_PAIS. (FK). Clave del país donde el alumno cursó su licenciatura

CLV_ESTADO (FK). Clave del estado donde el alumno cursó su licenciatura

PROMEDIO. Promedio de licenciatura del alumno

FECHA_TIT. Fecha en la que obtuvo el alumno el título de licenciatura

P_PLAN. Plan de posgrado anterior

P_ESC. Escuela o facultad donde cursó el posgrado

P_INS. (FK) . Clave de la institución donde el alumno cursó un posgrado.

P_PAIS (FK) . Clave del país donde el alumno cursó un posgrado.

P_ESTADO (FK) . Clave del estado donde el alumno cursó un posgrado.

P_CREDITOS. Porcentaje de créditos obtenidos durante el posgrado

P_GRADO. Grado alcanzado durante el posgrado

P_FECHATIT. Fecha de obtención del grado

I_INGRESO. Semestre de ingreso a la DEPFI

I_CLV_MAES. (FK) Clave del campo elegido para cursar una maestría.

I_NOMTUT. (FK). RFC del tutor asignado al alumno

BECA. (FK). Clave del tipo de beca del alumno.

PROFESOR

RFC. (PK). RFC del profesor

NOM_PROF. Nombre del profesor

APE_PAT. Apellido paterno del profesor

APE_MAT. Apellido materno del profesor

HISTORIA. Campo de desarrollo del profesor

MATERIA

CLV_MAT. (PK) Clave de la materia

NOM_MAT. Nombre de la materia

CREDITOS. Créditos de la materia

CLAVE CAMPO (FK) Clave del campo al cual pertenece la materia

INSCRITO

N_CUENTA. (PK). Número cuenta del alumno

SEMESTRE (PK). Semestre en el que el alumno realiza su inscripción

CLV_MAT (PK). Clave de la materia

NUMERO (PK) Número de grupo

SALON (PK). Clave del salón

PROFESOR (PK) RFC del profesor

ACTA. Número de folio donde se asienta la calificación de la materia

CALIF Calificación obtenida por el alumno

CLASE

RFC, (PK) RFC del profesor que imparte la clase

SALON. (PK) Clave del salón donde se da la clase

SEMESTRE. (PK). Semestre en el que se imparte la clase

CLV_MAT. (PK). Clave de la materia

NUMERO. (PK). Grupo de la materia

DIA1, DIA2, DIA3, DIA4, DIA5, DIA6. Días en los que se imparte la clase

HORA1, HORA2, HORA3, HORA4, HORA5, HORA6. Horario de la clase

OBS. Observaciones

MAT_PLAN

N_CUENTA. (PK). Número de cuenta del alumno

MATERIA (PK). Clave de la materia

SEMESTRE. Semestre actual

NUMERO. Semestre en el que cursará las materias

BECAS

CLAVE_BECA (PK). Clave del tipo de beca que el alumno tiene

NOMBRE_BEC. Nombre de la beca

ECIVIL

CLAVE. (PK). Clave del estado civil

ESTADO. Nombre del estado civil del alumno

ESTADOS

CLAVE_EDO. (PK) Clave del estado

NOMBRE_EDO. Nombre del estado

PAISES

CLAVE_PAIS. (PK). Clave del país

NOMBRE_PAI. Nombre del país

INSTITUC

CLAVE_INS. (PK). Clave de la institución

NOMBRE_INS. Nombre de la institución

TITULOS

CLAVE_TIT. (PK). Clave de título

TITULO. Nombre de título

SALON

N_SALON. (PK) Clave de salón

UBICACION. Ubicación del salón

SECCION

CLV_SEC. (PK) Clave de sección o área

NOM_SEC. Nombre de sección o área

CREDITOS Créditos de la sección o área

CAMPO

CLAVE CAMPO. (PK) Clave de campo

NOMBRE CAMPO. Nombre de campo

IV.2 Diccionario de datos

Un *diccionario de datos* es un catálogo, un depósito, de los elementos de un sistema. Es aquí donde se encuentra una lista de todos los elementos que forman parte del flujo de datos de todo el sistema. Si se desea conocer cuántos caracteres abarca un determinado dato o qué otros nombres recibe en distintas partes del sistema, o dónde se utiliza; un diccionario de datos desarrollado apropiadamente, nos será de gran utilidad.

Los diccionarios de datos son importantes, por cinco razones principalmente:

1. Para manejar los detalles en sistemas grandes.
2. Para comunicar un significado común para todos los elementos del sistema.
3. Para documentar las características del sistema.
4. Para 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.

Según el tipo de información que cada campo almacene, se le asociará un tipo de dato:

ALUMNO (Datos personales del alumno)

Campo	Tipo de dato	Longitud
N CUENTA	Texto	8
N EXP	Texto	8
NOM ALUM	Texto	20
APE PAT	Texto	15
APE MAT	Texto	15
EDO CIV	Texto	1
DIREC	Texto	40
COLONIA	Texto	20
ESTADO	Texto	2
CODIGO P	Texto	5
TEL DOM	Texto	9
TEL OFNA	Texto	15
CLV NAC	Texto	1
SEXO	Texto	1
FECHA NAC	Fecha/Hora	8
CLV TIT	Texto	2
CLV ESC	Texto	20
CLV INS	Texto	4
CLV PAIS	Texto	2
CLV ESTADO	Texto	2
PROMEDIO	Texto	5
FECHA TIT	Fecha/Hora	8
P PLAN	Texto	10
P ESC	Texto	20
P INS	Texto	4
P PAIS	Texto	2
P ESTADO	Texto	2
P CREDITOS	Número (Doble)	8
P GRADO	Texto	10
P FECHATIT	Fecha/Hora	8
I INGRESO	Texto	6
I CLV MAES	Texto	6
I NOMTUT	Texto	13
BECA	Texto	2

PROFESOR (Datos personales del profesor)

Campo	Tipo de dato	Longitud
RFC	Texto	13
NOM PROF	Texto	20
APE PAT	Texto	15
APE MAT	Texto	15
HISTORIA	Memo	-

MATERIA (Información de materias)

Campo	Tipo de dato	Longitud
CLV MAT	Texto	6
NOM MAT	Texto	35
CREDITOS	Número (Entero)	2
CLAVE CAMPO	Texto	1

MAT_PLAN (Información sobre la propuesta académica del alumno)

Campo	Tipo de dato	Longitud
N CUENTA	Texto	8
MATERIA	Texto	6
SEMESTRE	Texto	6
NUMERO	Texto	1

INSCRITO (Materias a las que el alumno está inscrito)

Campo	Tipo de dato	Longitud
N CUENTA	Texto	8
SEMESTRE	Texto	6
CLV MAT	Texto	6
NUMERO	Texto	4
SALON	Texto	4
PROFESOR	Texto	13
ACTA	Texto	8
CALIF	Número (Doble)	1

CLASE (Información de las clases que se imparten)

Campo	Tipo de dato	Longitud
RFC	Texto	13
SALON	Texto	4
SEMESTRE	Texto	6
CLV MAT	Texto	6
NUMERO	Texto	4
DIA1	Texto	2
DIA2	Texto	2
DIA3	Texto	2
DIA4	Texto	2
DIA5	Texto	2
DIA6	Texto	2
HORA1	Texto	11
HORA2	Texto	11
HORA3	Texto	11
HORA4	Texto	11
HORA5	Texto	11
HORA6	Texto	11
OBS	Texto	110

BECA (Catálogo de becas)

Campo	Tipo de dato	Longitud
CLAVE BECA	Texto	2
NOMBRE BECA	Texto	30

ECIVIL (Catálogo de estado civil para un alumno)

Campo	Tipo de dato	Longitud
CLAVE	Texto	1
ESTADO	Texto	10

PAISES (Catálogo de países)

Campo	Tipo de dato	Longitud
CLAVE PAIS	Texto	2
NOMBRE PAI	Texto	25

INSTITUC (Catálogo de Instituciones)

Campo	Tipo de dato	Longitud
CLAVE INS	Texto	4
NOMBRE INS	Texto	50

TITULOS (Catálogo de carreras de licenciatura)

Campo	Tipo de dato	Longitud
CLAVE TIT	Texto	2
TITULO	Texto	42

SALON (Información sobre los salones)

Campo	Tipo de dato	Longitud
N SALON	Texto	4
UBICACION	Texto	60

SECCION (Catálogo de áreas)

Campo	Tipo de dato	Longitud
CLV SEC	Texto	6
NOM SEC	Texto	30
CREDITOS	Texto	3

ESTADOS (Catálogo de países)

CAMPO (Campo al que pertenece cada materia)

Campo	Tipo de dato	Longitud
CLAVE EDO	Texto	2
NOMBRE EDO	Texto	25

Campo	Tipo de dato	Longitud
CLAVE CAMPO	Texto	3
NOMBRE CAMPO	Texto	30

IV.3 ESTRUCTURA DEL SISTEMA

El sistema de control académico para el Departamento de Sistemas de la DEPFI (**SICAP**) está conformado por cuatro módulos, estos módulos son:

1. ALUMNOS
2. PROCESO SEMESTRAL
3. REPORTES
4. CATÁLOGOS

1. ALUMNOS

Este módulo contiene las siguientes opciones:

1.1 Bajas. Nos permite cancelar la inscripción al semestre actual de un alumno y/o dar de baja definitivamente a un alumno del sistema.

1.2 Cambios. Permite hacer modificaciones en los datos personales del alumno y/o en la propuesta académica o plan de estudios del mismo.

1.3 Consultas. Permite consultar datos personales, plan de estudios, inscripción al semestre actual y/o kárdex.

2. PROCESO SEMESTRAL

Este módulo contiene las siguientes opciones:

2.1 Materias

2.2 Profesores

2.3 Horarios

2.4 Inscripción

2.5 Calificaciones

A su vez, se presentan los siguientes submódulos:

2.1 Materias

2.1.1 Altas

2.1.2 Bajas

2.1.3 Cambios

2.1.4 Consultas

2.1.4.1 Catálogo General

2.1.4.2 Por clave

2.2 Profesores

2.2.1 Altas

2.2.2 Bajas

2.2.3 Cambios

2.2.4 Consultas

2.2.4.1 Catálogo General

2.2.4.2 Datos personales

2.2.4.3 Materias que imparte

2.2.4.4 Tutorías

2.3 Horarios

2.3.1 Altas

2.3.2 Bajas

2.3.3 Cambios

2.3.4 Consultas

2.3.4.1 Histórica

2.3.4.2 Clase por materia

2.3.4.3 Clase por grupo

2.4 Inscripción

2.4.1 Primer Ingreso

2.4.1.1 Capturar datos personales

2.4.1.1.1 UNAM

2.4.1.1.2 No UNAM

2.4.1.2 Capturar plan de estudios

2.4.1.3 Realizar Inscripción

2.4.2 Reingreso

2.5 Calificaciones

2.5.1 Asignar

2.5.2 Cancelar

2.5.3 Cambiar

2.5.4 Consultar

2.5.4.1 General

2.5.4.2 Específica

El módulo *Proceso Semestral* permite realizar todas las actividades que dentro de un semestre escolar se llevan a cabo, estas actividades son:

- *Submódulo Materias.* Nos permite realizar todas las tareas referentes a las materias que se imparten en el Departamento de Ingeniería de Sistemas. Estas tareas son: Dar de alta, eliminar, cambiar ó consultar.
- *Submódulo Profesores.* Se llevan a cabo tareas como la captura de datos personales de un profesor, dar de baja, cambiar algún dato ó consultas al catálogo general, datos personales, materias que imparte ó tutorías a cargo de los profesores.
- *Submódulo Horarios.* Nos permite generar, eliminar y cambiar los horarios de clase, así

como hacer una consulta histórica de clases impartidas en un determinado semestre, información de una clase específica ó grupos abiertos de una materia.

- **Submódulo Inscripción.** Si el alumno es de primer ingreso, es decir, es la primera vez que se inscribe a una maestría del Departamento de Sistemas, se debe proceder a la captura de sus datos personales y plan de estudios. Si el alumno proviene de alguna dependencia de la UNAM la entrada al sistema es mediante el número de cuenta, en caso contrario, si el alumno es totalmente ajeno a la UNAM, el sistema le genera un número de cuenta provisional. Si el alumno es de Reingreso -el alumno ya a cursado alguna materia en semestres anteriores- se procede a realizar su inscripción al semestre actual. Cabe señalar que para alumnos procedentes de la UNAM, se hizo uso de un algoritmo que nos permite determinar la autenticidad del número de cuenta. Este algoritmo se presenta en el Apéndice D.
- **Submódulo Calificaciones.** Este submódulo nos permite asignar calificaciones a los alumnos. El sistema genera un Número de folio que asocia a una materia y grupo, se listan todos los alumnos que estén inscritos en dicha materia y grupo, y se les asigna una calificación numérica. Este submódulo también nos permite cancelar, cambiar o consultar un número de folio.

3. REPORTES

Este módulo contiene las siguientes opciones:

- 3.1 Lista de Asistencia.** Nos permite ver e imprimir listas de asistencia para cada profesor.
- 3.2 Kárdex.** Nos permite ver e imprimir el kárdex para un alumno o para todos.
- 3.3 Números provisionales.** Presenta una lista de alumnos a los cuales el sistema les generó un número de cuenta provisional.
- 3.4 Estadísticos.** Nos permite ver e imprimir, así como enviar a un documento de Word 6.0 ó Excel 5.0, información de carácter estadístico sobre la población de alumnos del Departamento.

El módulo *Estadísticos*, tiene las siguientes opciones:

- 3.4.1 Alumnos por área
- 3.4.2 Alumnos por nacionalidad
- 3.4.3 Alumnos por género
- 3.4.4 Alumnos por carrera de licenciatura
- 3.4.5 Alumnos por institución
- 3.4.6 Alumnos por beca

3.4.7 Alumnos por estado

4. CATÁLOGOS

Este módulo contiene las siguientes opciones:

4.1 Salones

4.2 Países

4.3 Estados

4.4 Instituciones

4.5 Carreras

4.6 Becas

4.7 Áreas

4.8 Passwords

Para todas las opciones anteriores, se tiene:

- Altas
- Bajas
- Cambios
- Consultas.

Este módulo nos permite dar mantenimiento a los catálogos generales del sistema, a través de altas, bajas, cambios y consultas.

C

a

p

i

t

u

l

o

v

Integración y pruebas del sistema

V.1 Pruebas

La prueba de cualquier sistema implica ejercitar éste para asegurar que produzca las salidas apropiadas y exhiba el comportamiento adecuado para una amplia gama de entradas.

La prueba del software es un elemento importante ya que constituye la garantía de calidad del sistema, representando una revisión final de las especificaciones proporcionadas por el usuario, del diseño y de la programación, de tal forma que esta fase tiene como objetivos:

- La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.

Es probable que el proceso de prueba del sistema tome tanto tiempo como la mitad del tiempo programado para su desarrollo, dependiendo esto de qué tan cuidadosamente se hayan hecho las actividades iniciales de análisis, diseño y programación.

El flujo de información para las pruebas sigue el esquema de la figura V.1.

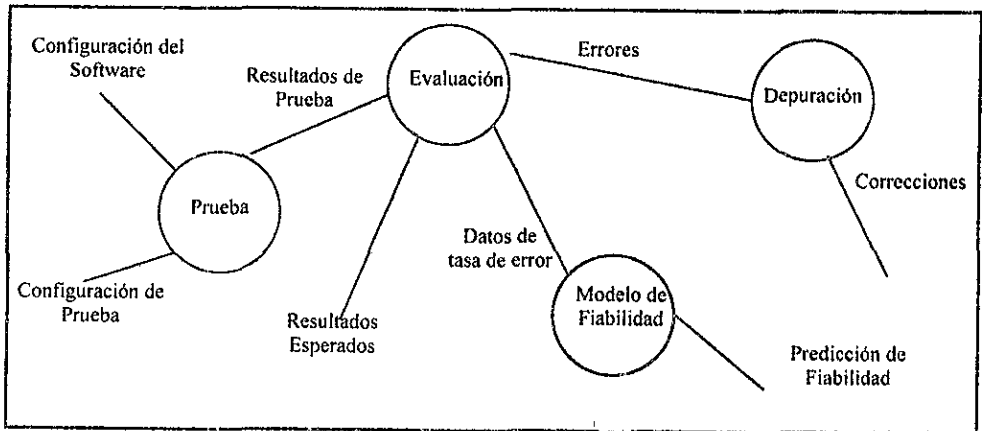


Figura V.1 Flujo de información para las pruebas.

En la figura anterior se tienen dos entradas: a) La configuración del software que incluye la especificación de requisitos del software, la especificación del diseño y el código fuente; b) una configuración de prueba que incluye un plan y un procedimiento de prueba, así como los casos de prueba a utilizar y los resultados esperados.

Se realizan las pruebas y se evalúan los resultados esperados contra los generados,

Se realizan las pruebas y se evalúan los resultados esperados contra los generados, cuando se descubren errores se inicia un proceso de depuración que tiene como finalidad realizar las correcciones pertinentes.

A medida que se recopilan y se evalúan los resultados de la prueba se puede determinar medida cualitativa de la calidad y fiabilidad del software¹. Si se encuentran frecuentemente errores que requieren modificaciones en el diseño, la calidad y la fiabilidad del software quedan en entre dicho, de tal forma que es necesario seguir realizando pruebas. Si por el contrario, el funcionamiento del software parece ser el correcto y los errores que se encuentran son menores, se pudiese pensar que la calidad y la fiabilidad del software son aceptables, o que las pruebas fueron inadecuadas, ya que no permitieron descubrir errores importantes. Así mismo, si durante la prueba no se descubren errores, quedará la sospecha de que las pruebas no fueron adecuadas y que el software puede estar defectuoso, para lo cual se considera que estos defectos pudiesen ser descubiertos por el usuario y deberán ser corregidos en la fase de mantenimiento del sistema.

Para realizar las pruebas del sistema se tienen *estrategias* y *técnicas*. Las estrategias de prueba de los sistemas constituyen la planificación o pasos a seguir para su elaboración, en tanto que las técnicas representan los métodos para realizar dichas pruebas.

* *Técnicas.*

En cuanto a las técnicas para las pruebas del software existen dos enfoques denominados pruebas de caja negra y pruebas de caja blanca.

a) Prueba de caja negra. Esta prueba pretende demostrar que la entrada se acepta en forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. La prueba de caja negra examina algunos aspectos del modelo fundamental del sistema sin tomar mucho en cuenta la estructura interna lógica del sistema.

Esta prueba intenta encontrar errores de las siguientes categorías:

1. Funciones incorrectas o ausentes
2. Errores de interfase
3. Errores en estructura de datos o en accesos a base de datos externas
4. Errores de rendimiento
5. Errores de inicialización y de terminación

b) Prueba de caja blanca. Esta prueba se basa en un minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software, proponiendo casos de prueba que provoquen que se cumplan conjuntos específicos de condiciones examinando el comportamiento del programa en varios puntos para determinar si el resultado real coincide con el esperado. El inconveniente de la prueba de la caja blanca es que para algunos casos es complicado e imposible probar todos los caminos lógicos.

¹ Ingeniería de Software, un enfoque práctico.

• Estrategias.

El proceso de ingeniería de software y de las estrategias se puede representarse como una espiral en donde se tienen pruebas unidad, pruebas de integración, pruebas de validación y pruebas del sistema. (Figura V.2)

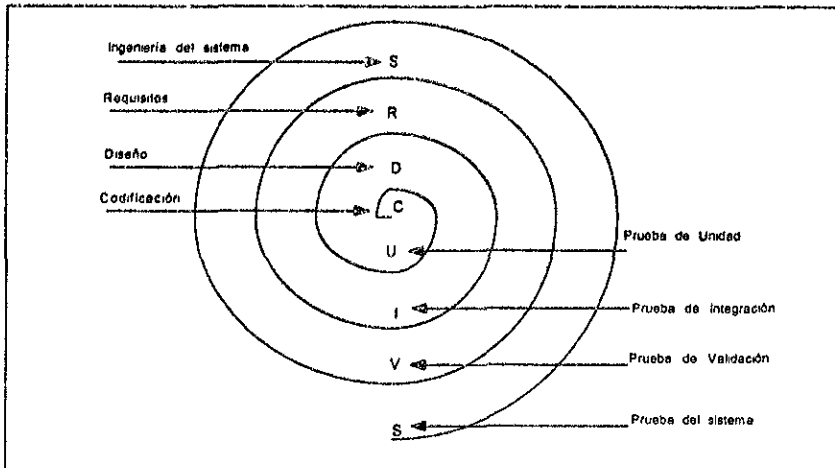


Figura V.2 Estrategia de Prueba

La prueba de unidad comienza en el vértice de la espiral y se centra en cada unidad del software tal como se encuentre implementado el código fuente. La prueba avanza, al moverse fuera de la espiral, hasta llegar a la prueba de integración, que corresponde al diseño y a la construcción del software, posteriormente se encuentra la prueba de validación, en donde se validan los requisitos establecidos por el usuario, comparándolos con el sistema que ha sido construido, finalmente se tiene la prueba del sistema en la que se prueba todo el software.

a) Pruebas de unidad.

Las pruebas que conforman la prueba de unidad son las siguientes:

- › *Interfase.* Se comprueba que la información fluye de forma adecuada hacia y desde la unidad o programa que está siendo probado. Se examinan las estructuras de los datos locales para asegurar que los datos que se mantienen temporalmente conservan su integridad durante todos los pasos de ejecución. Se prueban las condiciones límite para asegurar que el programa funciona correctamente en los límites establecidos como restricciones de procesamiento. De esta manera se prueban todos los caminos de la estructura de control con el fin de asegurar que todas las instrucciones del programa se

ejecuten al menos una vez.

- **Manejo de errores.** Entre los errores potenciales que se comprueban tenemos los siguientes:
 1. Descripción ininteligible del error.
 2. El error señalado no corresponde con el error encontrado.
 3. La condición de error hace que intervenga el sistema antes que el mecanismo de manejo de errores.
 4. El procesamiento de la condición excepcional es incorrecto.
 5. La condición de error no proporciona suficiente información para ayudar a la localización del error.
- **La prueba de límites.** Verifica los valores máximos y mínimos permitidos, los valores de los datos por debajo y por encima de los máximos y mínimos son apropiados para descubrir estos errores.

b) Pruebas de integración.

Una vez que se han realizado las pruebas de unidad o de programas por separado, el paso siguiente es ponerlos ahora interaccionando, es decir, realizar la prueba de integración. En esta caso se revisa, entre otros:

- Que un programa pueda tener un efecto adverso o inadvertido sobre otro.
- Las funciones cuando se combinan, pueden no producir la función principal deseada.
- Las estructuras de datos globales pueden presentar problemas.
- Los datos se pueden perder en alguna interfase.

Para este tipo de pruebas, existen dos estrategias: *prueba ascendente* y *prueba descendente*. El enfoque *ascendente* empieza por probar módulos individuales pequeños separadamente, luego los módulos individuales se combinan para formar unidades cada vez más grandes que se probarán en masa; esto se conoce como prueba de subsistemas. Finalmente, todos los componentes del sistema se combinan para probarse; esto se conoce como prueba del sistema y suele estar seguido por las pruebas de aceptación, donde se permite al usuario usar sus propios casos de prueba para verificar que el sistema esté trabajando de forma correcta. El enfoque de prueba *descendente* empieza con un esqueleto del sistema, es decir, la estrategia de prueba supone que se han desarrollado los modelos ejecutivos de alto nivel del sistema, pero que los de bajo nivel existen sólo como módulos que no procesan nada.

La selección de la estrategia de integración depende de las características del software y, a veces, del plan del proyecto. En términos generales se puede utilizar un planteamiento combinado.

c) Prueba de validación.

Después de la prueba de integración, el paso siguiente es la prueba de validación. La prueba de validación es la que se obtiene cuando el software desarrollado funciona de acuerdo a las expectativas razonables del cliente. La validación del software se obtiene aplicando una serie de pruebas que demuestran la conformidad con los requerimientos.

d) Prueba del sistema.

- Prueba de recuperación. El propósito de este tipo de prueba es asegurar que el sistema pueda recuperarse adecuadamente de diversos tipos de fallas. Las pruebas de recuperación pueden requerir que el equipo que realiza el proyecto simule o provoque fallas de hardware, fallas de corriente, fallas en el sistema operativo, etc. Este tipo de prueba puede ser manual o automática. Si la recuperación es automática se deberá evaluar que el sistema se reinicialice correctamente, si la recuperación requiere de la intervención humana, hay que evaluar los tiempos medios de recuperación para determinar si están dentro de los límites aceptables.

- Prueba de seguridad. Consiste en verificar los mecanismos de protección incorporados al sistema. Estos mecanismos tienen la finalidad de proteger al sistema contra acciones impropias de gente que traten de perjudicar u obtener información en forma ilícita.

- Prueba de resistencia. Ejecuta un sistema de forma que demande recursos en cantidades anormales, es decir, diseñar pruebas especiales que generen 10 interrupciones por segundo, pero sólo una o dos son normales, incrementar las frecuencias de datos de entrada en un orden de magnitud con el fin de comprobar cómo responden las funciones de entrada, ejecutar casos de prueba que puedan dar problemas con el esquema de gestión de memoria, diseñar casos de prueba que produzcan excesivas búsquedas de datos en disco.

- Prueba de rendimiento. Está diseñada para probar el rendimiento del sistema en tiempo de ejecución dentro del contexto de un sistema integrado. Las pruebas de rendimiento a menudo van emparejadas a las pruebas de resistencia. La instrumentación consiste en monitorear los intervalos de ejecución, los sucesos ocurridos y muestras de los estados de la máquina en funcionamiento normal. De esta forma se pueden descubrir situaciones que lleven a degradaciones y posibles fallos del sistema.

- *Pruebas realizadas al sistema SICAP.*

El equipo seleccionado para llevar a cabo las pruebas del sistema, 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.

Para la realización de pruebas del SICAP, existieron dos fases, la primera consistió en utilizar datos de prueba en una PC estándar, y la segunda consistió en la verificación del sistema utilizando datos reales y en el equipo destinado para el uso exclusivo del sistema, estos datos reales se tomaron del periodo 97-2. Las pruebas fueron realizadas por el personal que se encargará del uso de dicho sistema.

- *Técnicas de prueba aplicadas al SICAP*

Se realizaron pruebas de “caja negra” determinándose de antemano sus salidas para después compararlas con datos reales. Así mismo, se realizaron pruebas de “caja blanca” en donde se revisó la secuencia lógica de módulo, para luego integrarlos.

- *Estrategias de prueba aplicadas al SICAP.*

- ✓ *Pruebas de unidad.*

En todo sistema la idea de funcionalidad va de acorde a los procesos de creación, actualización, protección y revisión de archivos denominados archivos maestros o base de datos. Estos archivos forman la base del sistema y por ello se debe de contar con datos de prueba que permitan la verificación de los movimientos a dichos archivos.

A continuación se señalan los movimientos que se consideraron para verificar el correcto funcionamiento de cada módulo del sistema:

- Recuperar. Permite recuperar un registro en base a la clave especificada
- Agregar. Permite añadir los datos para un nuevo registro.
- Modificar. Permite actualizar o cambiar los datos del registro que se esté visualizando.
- Eliminar. Permite eliminar el registro que se esté visualizando.
- Total. Permite determinar el total de registros capturados.
- Filtro. Permite registrar en algunos campos los datos por los que se desea filtrar los registros. Si existen registros para el filtro especificado, se pueden aplicar los movimientos mencionados anteriormente.
- Salida. Permite retornar a la pantalla previa.
- Generar. Permite la emisión de los reportes escritos para visualizarlos en pantalla o en papel.

Las opciones o movimientos permitidos en cada módulo, dependen del objetivo de cada uno de ellos, por ejemplo, para un reporte que tiene el objetivo de emitir información en forma escrita, se utilizan los movimientos “Genera” e “Imprime”, mientras que para una consulta de información que tiene como objetivo primordial visualizar información específica, contiene casi todos los movimientos excepto los de “Agregar”, “Modificar” y “Eliminar”.

En base a los movimientos definidos, se procedió a probar cada módulo. Las pruebas de unidad comenzaron por:

- Dar de alta al menos 4 alumnos procedentes de la UNAM y 4 alumnos ajenos a la UNAM, esto es, capturar sus datos personales, el nombre de su tutor, así como su propuesta académica o plan de estudios.
- Dar de alta al menos 6 profesores capturando también sus datos personales.

- › Dar de alta materias de todos los campos para las áreas de Investigación de operaciones, Planeación y Transporte.
- › Formar clases
- › Dar de alta los catálogos.
- › Utilizar todas y cada una de las opciones de reporte con las que el sistema cuenta.
- › Utilizar el programa para el mantenimiento del sistema.

Para el registro de toda esta información se probaron cada una de las opciones de captura de información disponibles en cada pantalla del sistema. (Altas, Bajas, Cambios, etc.).

✓ *Pruebas de integración.*

La prueba de integración para el caso del SICAP se realizó de manera paulatina conforme se desarrollaba cada uno de los módulos, es decir, se utilizó la integración ascendente. En algunos casos se detectó duplicidad de funciones y al corregir este problema hizo mucho más eficiente el sistema.

✓ *Pruebas de validación.*

La prueba de validación para el SICAP se basó en los siguientes aspectos:

- La confiabilidad de los datos
- El adecuado comportamiento de la interfase con el usuario.

Para la confiabilidad de los datos se verificó primero que los datos presentados en las pantallas de captura del sistema, fueran los mismos que los anotados en los formatos que la DEPMI entrega a cada aspirante al posgrado.

CONCLUSIONES

Las conclusiones tienen una importancia significativa en el desarrollo de cualquier tipo de sistema, ya que nos permite saber si se cumplieron los objetivos inicialmente planteados, así como determinar si éstos satisfacen los requerimientos del usuario.

Los objetivos que se perseguían con la elaboración de este proyecto de tesis fueron cubiertos en su totalidad puesto que se logró crear una herramienta de calidad que contribuirá enormemente en el control y administración de la información con lo cual los usuarios realizarán su trabajo de una manera más sencilla, flexible y ágil.

En el desarrollo del sistema se considera que el aspecto del análisis constituye la etapa más importante, pero por lo mismo la más difícil de evaluar; importante porque se encarga de determinar la mayor o menor probabilidad de éxito de todo el proyecto, difícil puesto que no existen parámetros reales que determinen, antes de continuar con la siguiente etapa, si el análisis ha sido lo suficientemente exhaustivo.

Finalmente, al concluir este trabajo de tesis, se ha comprendido que una de las actividades fundamentales del ingeniero es la optimización de recursos y el desarrollo de nuevas alternativas para la solución de problemas computacionales. Bajo esto, se ha diseñado e implementado el Sistema de Control Académico para el Departamento de Sistemas de la DEPFI, (SICAP) en el que se procuró desde el primer momento utilizar los recursos disponibles en el Departamento con el fin de producir un sistema que será clave para resolver la problemática existente en cuanto al procesamiento y administración de la información.

A

P

é

n

d

i

e

e

A

Bases de datos Access

¿Qué es Microsoft Access?

Microsoft Access 2.0 resulta ser el sistema de administración de bases de datos relacionales estándar para Microsoft Windows¹.

Todas las aplicaciones bajo Windows disponen de una interfase gráfica de usuario similar, que permite realizar todas las tareas mediante las opciones incluidas en un conjunto de menús en la ventana o ventanas de trabajo. Lo que distingue al sistema de administración de bases de datos relacionales Microsoft Access de las bases de datos típicas es su sencilla filosofía de diseño. Una base de datos debe ser fácil de usar, independientemente quién la utilice.

Las ventanas proporcionadas por Microsoft Access contienen una serie de menús que permiten realizar muchas tareas de la administración de bases de datos, tales como: crear tablas para almacenar datos, visualizar y editar los datos, realizar consultas, generar informes, diseñar pantallas atractivas para introducir o visualizar los datos, automatizar el trabajo utilizando macros, etc.

Una base de datos Access está formada por un conjunto de objetos: las tablas, las consultas, los formularios, los informes, las macros y los módulos, que agrupan de forma estructurada toda la información disponible que se va a administrar.

En la figura A.1, se ilustra la forma cómo se encuentra estructurado Microsoft Access 2.0, ilustrando los bloques principales que lo componen y que más adelante se mencionarán las características de cada uno.

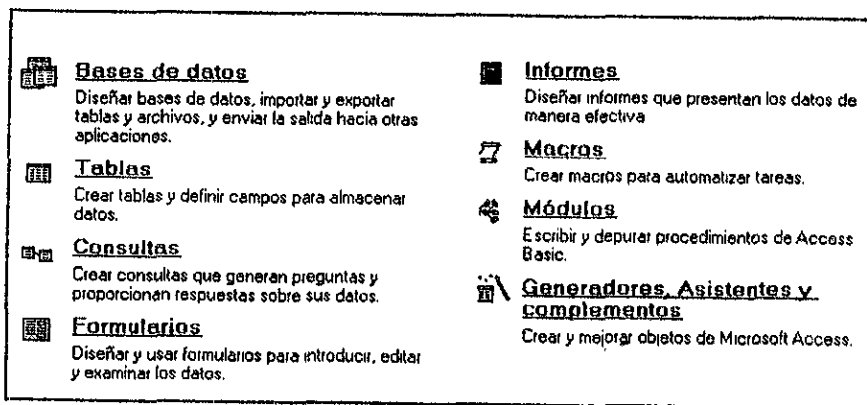


Figura A.1 Estructura de Microsoft Access 2.0

¹ a) Microsoft Windows es un entorno operativo que funciona con MS-DOS. Es un entorno de ventanas multitarea basado en gráficos. b) Windows es una shell (intérprete de comandos) que da un ambiente de trabajo mucho más intuitivo de operar. Ambas definiciones se aplican hasta la versión de Windows 3.11.

Los objetos que a continuación se mencionan son los que almacenan la información y que en conjunto forman lo que es la base de datos.

<i>Objeto</i>	<i>Descripción</i>
Tablas	Almacena datos que contienen un determinado tipo de información en registros (filas) y campos (columnas).
Consultas	Nos permiten localizar, organizar o editar la información contenida en las tablas.
Formularios	Forma que se utiliza para visualizar, y editar información de la base de datos, registro por registro.
Reportes	Forma de presentar los datos contenidos en una o varias tablas, impresos en una hoja de papel.
Macros	Acción o conjunto de acciones que nos permiten automatizar tareas.
Módulo de Programación	Colección de declaraciones, instrucciones y procedimientos de Access Basic.

Características Generales.

Access es el primer administrador de bases de datos de Microsoft. Trabaja con Windows, lo que proporciona las facilidades y el potencial de Microsoft Windows. Incluye un entorno gráfico para crear y modificar tablas, formularios, informes, consultas y macros. Así mismo nos permite pasar objetos de un área de trabajo a otra, fusionar dos o más tablas dibujando una línea entre los campos comunes, trabajar con todos los datos, ordenarlos según se desee y elaborar todo tipo de informes.

Otra de las ventajas de Access es la posibilidad de incluir objetos OLE ("*Object Linking and Embedding*", "*Vinculación e incrustación de objetos*") en los registros.²

En Microsoft Access, se puede importar, exportar o vincular datos de diferentes formatos de base de datos, hoja de datos y archivo de texto. Para ello, Microsoft Access utiliza un controlador incorporado o un controlador ODBC ("*Open Database Connectivity*", "*Conectividad abierta de bases de datos*"); que es una biblioteca de vínculos dinámicos (DLL) utilizado por Microsoft Access para conectarse a una base de datos SQL como, por ejemplo, Microsoft SQL Server, Sybase SQL Server u ORACLE Server. Cada tipo de base de datos SQL requiere un controlador ODBC diferente.

Microsoft Access contiene controladores incorporados que permiten importar, exportar o vincular estos tipos de datos:

² Posibilidad de incluir sonidos, animación, documentos de un procesador de textos, una hoja de cálculo o una imagen gráfica.

- Otras bases de datos de Microsoft Access
- Archivos de base de datos de Microsoft FoxPro versiones 2.x y 3.0
- Tablas de Paradox versiones 3.x, 4.x y 5.0
- Archivos de dBASE III, III+, IV y 5
- Hojas de cálculo de Microsoft Excel y Lotus 1-2-3
- Archivos de texto (de ancho fijo y delimitados)

Características y Diseño de una Base de Datos en Microsoft Access 2.0

Como miembro de la familia de programas Microsoft Office[1], Microsoft Access incorpora la tecnología IntelliSense³. Esto automatiza las tareas rutinarias y simplifica las tareas complejas. Por ejemplo, docenas de Asistentes⁴ ayudan a automatizar todos los procesos, desde la creación de formularios hasta la creación de código.

Resulta muy sencillo compartir los datos gracias a la tecnología OfficeLink⁵, de forma que se puedan enviar informes a Microsoft Excel, por ejemplo.

Para poder obtener el máximo provecho de las herramientas con que cuenta Access, es importante saber diseñar correctamente una base de datos. La clave para comprender el funcionamiento del manejador de Bases de datos estriba en tener bases sólidas acerca de la organización de la base de datos.

Los pasos para el diseño de una base de datos son:

- Determinar el propósito de la base de datos
- Determinar las tablas
- Definir los campos
- Determinar las relaciones entre tablas
- Depurar el diseño

En la figura A.2 se muestra la ventana de Base de Datos con los respectivos elementos que lo forman.

³ Inteligencia incorporada que detecta lo que se desea hacer y produce el resultado deseado.

⁴ Una herramienta de Microsoft Access que hace preguntas y crea un objeto de acuerdo con las respuestas.

⁵ Tecnología que nos permite conectar todos los miembros de la familia Office.

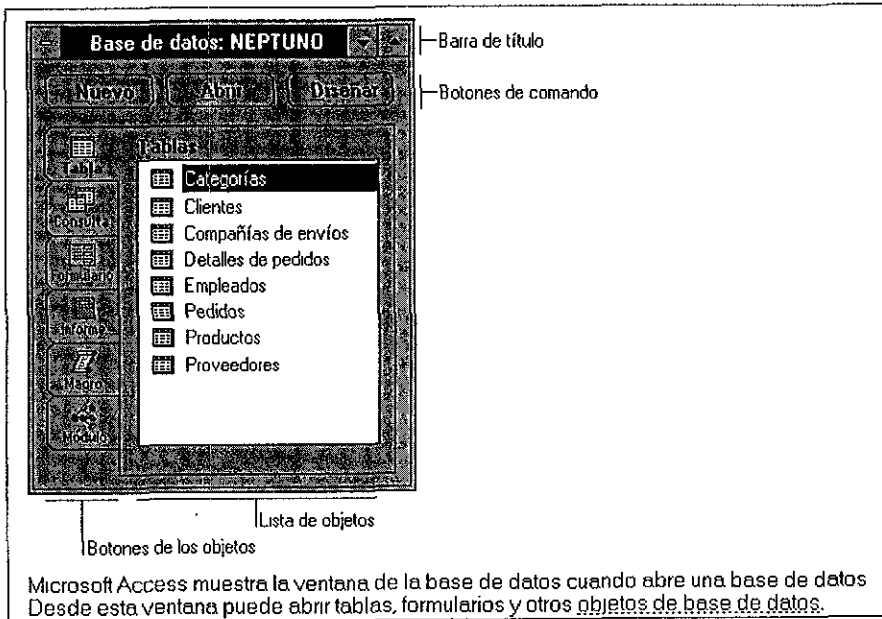


Figura A.2 Ventana de la Base de Datos

El poder de manejo de una base de datos relacional como Access incide en su rápida búsqueda y extracción de información almacenada en tablas separadas. Para que Access trabaje eficientemente, cada tabla debe incluir un campo o conjunto de campos que identifiquen únicamente cada renglón o registro almacenado en la tabla.

En la figura A.3 se ilustran los elementos que forman la ventana de base de Datos, con sus características.

La ventana de la base de datos contiene cuatro elementos	
Elemento	Propósito
Barra de título	Muestra el nombre de la base de datos abierta
Botones de comando	Abren el objeto seleccionado en una ventana o modo de presentación determinados. Por ejemplo, haga clic en el botón "Diseñar" para abrir el objeto seleccionado en el modo de presentación Diseño.
Botones de objetos	Muestran una lista de los objetos que pueden ser abiertos o modificados. Por ejemplo, haga clic en el botón "Formulario" para mostrar una lista de los formularios de la base de datos abierta.
Lista de objetos	En la lista, seleccione el objeto que desee abrir.

Figura A.3 Descripción de los elementos que forman la ventana de Base de Datos.

Características de las tablas.

Antes de crear cualquier otro objeto de una base de datos como lo son las consultas, formas o reportes, se deben crear primero las tablas.

Una tabla es una colección de datos acerca de un tema en particular. Los datos en una tabla son presentados en un formato matricial con columnas llamadas **campos** y renglones llamados **registros**.

En una tabla, un campo es una categoría de información, esto puede ser por ejemplo, los nombres de categorías de productos, descripción de la categoría, etc., en tanto que un registro es una colección de información acerca de una persona, producto, etc.

Cada registro en una tabla contiene el mismo conjunto de campos y cada campo contiene el mismo tipo de información de cada registro.

En una Base de Datos pueden haber muchas tablas, cada una de las cuales almacena información sobre un tema distinto. Cada tabla puede contener datos de muchos tipos, como son texto, números, fechas e imágenes.

Con Microsoft Access se puede usar la ventana para crear una nueva tabla ó ver las tablas existentes. La tabla se puede abrir en cualquiera de las dos presentaciones existentes: Diseño y Hoja de datos. En la figura A.4 se muestra la ventana tabla en el modo de presentación Diseño, y se señalan los elementos que la forman.

Al diseñar las tablas es necesario especificar un identificador exclusivo de registros, mismo que se le denomina **llave primaria**, y consiste en uno o más campos que identifican a cada registro almacenado en la tabla. Por ejemplo, en la figura A.4 se observa que la llave primaria es: **id de categoría**.

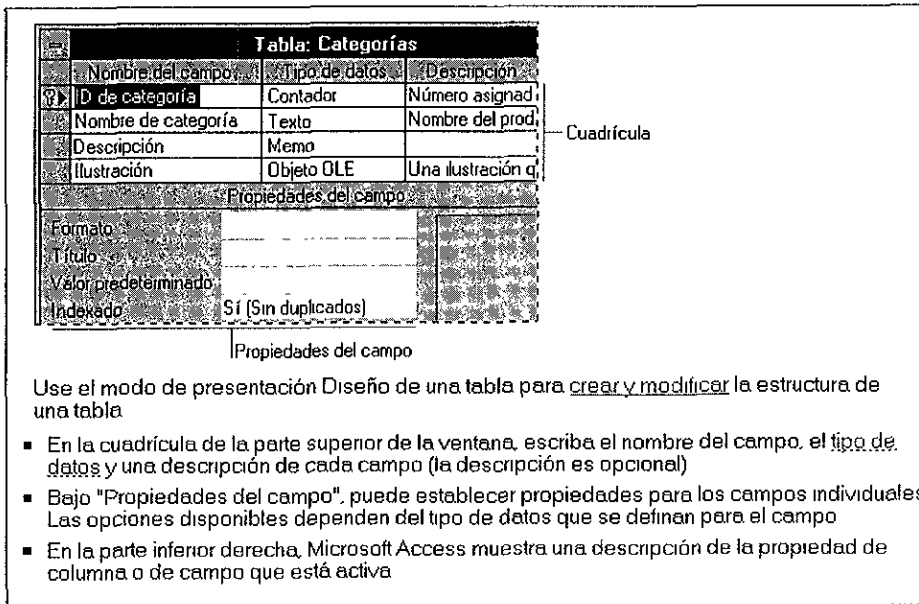


Figura A.4 Ventana tabla, modo de presentación Diseño.

Las reglas de validación⁶ que se establezcan para las tablas y sus campos siempre estarán en vigor tanto si se cambian los datos desde un formulario, como si se importan nuevos datos, o se utilizan consultas de acciones, una macro o código de Access Basic para cambiar datos. Estableciendo las propiedades de los campos en el modo de presentación Diseño de una tabla, es posible controlar la apariencia de los datos, evitar su introducción incorrecta, especificar valores predeterminados y acelerar la búsqueda y la ordenación de la tabla.

Por ejemplo se le puede dar formato a los números para facilitar su lectura o establecer reglas que los datos deban cumplir para que sea posible introducirlos en un campo.

Microsoft Access cuenta con un Asistente para tablas, en donde se puede elegir entre docenas de tablas de ejemplo y cientos de campos prácticos de ejemplo, y dejar así, que el Asistente cree automáticamente las tablas. También se pueden filtrar los registros de

⁶ Una regla que establece límites o condiciones acerca de lo que se puede introducir en uno o más campos. Las reglas de validación pueden establecerse para campos, registros o controles de un formulario. Las reglas de validación para campos o controles se comprueban al pasar a otro campo o control. Las reglas de validación para registros se comprueban al pasar a otro registro distinto.

una tabla para ver únicamente los registros deseados. Además se pueden presentar y editar simultáneamente todos los índices de una tabla, en la ventada Índices.

En la figura A.5 se muestra la ventana tabla en el modo de presentación Hoja de Datos, con los todos elementos que la forman.

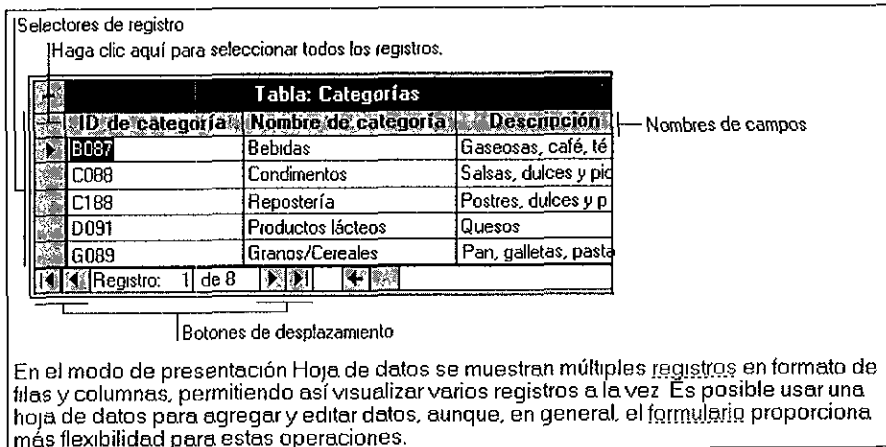


Figura A.5 Ventana tabla, Modo Presentación de Hoja de Datos.

Características de las relaciones entre tablas.

Cuando se crea una tabla, ésta no está relacionada⁷ con las demás tablas de la Base de Datos. Para crear una relación, se debe abrir la ventana de relaciones y añadir las tablas o consultas que se deseen relacionar. Microsoft Access cuenta con una ventana gráfica llamada Relaciones, que permite realizar de forma visual un boceto del diseño de la Base de Datos y observar simultáneamente todas las relaciones. (Figura A.6)

Gracias a las actualizaciones y eliminaciones en cascada, ahora se puede actualizar un campo de una tabla y hacer que Microsoft Access actualice automáticamente los campos relacionados de las tablas relacionadas. También se puede eliminar un registro y hacer que Microsoft Access elimine automáticamente los registros relacionados de las tablas relacionadas.

⁷ Una asociación establecida entre campos comunes (columnas) en dos tablas. Una relación puede ser de tipo uno a uno o uno a varios

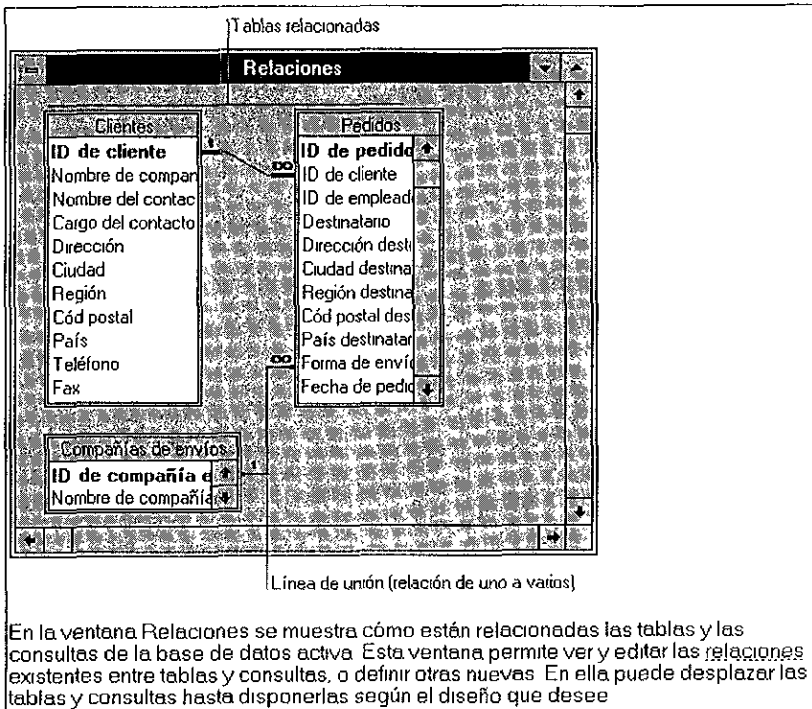


Figura A.6 Ventana Relaciones

Características de las Consultas.

Una consulta es la respuesta a una serie de preguntas acerca de datos almacenados en tablas o incluso de consultas. La forma en que se diseñe la consulta, indicará con exactitud a Microsoft Access qué datos debe recuperar.

Microsoft Access cuenta con el Asistente para consultas, que nos podrán ayudar a crear consultas complejas para tareas frecuentes de administración de datos, tales como ver una presentación en tabla de referencias cruzadas de los datos, buscar todos los registros duplicados de una tabla o consulta ó buscar todos los registros que sean distintos de los registros de otra tabla.

Aún cuando no se hayan establecido relaciones entre tablas, Microsoft Access crea automáticamente combinaciones en una consulta entre tablas que tengan campos coincidentes.

La figura A 7 muestra la ventana Consulta, con sus diferentes modos de presentación.

Use la ventana Consulta para crear y ejecutar una consulta y ver sus resultados. Puede abrir una consulta en cualquiera de dos presentaciones:

Modo de presentación	Propósito
Diseño	Crear y modificar la estructura de una consulta.
Hoja de datos	Mostrar y – en algunos casos (bmc emdash.bmp) agregar y editar datos
Presentación preliminar	Mostrar la apariencia que tendrá el objeto una vez impreso
SQL	Mostrar y editar la instrucción SQL de la consulta

Figura A.7 Ventana Consulta, con sus Modos de Presentación.

Para crear una nueva consulta ó modificar el diseño de una ya existente, es necesario utilizar el modo de presentación Diseño, en esta presentación se podrán emplear herramientas gráficas para la creación de las consultas. En la figura A.8 se muestra la ventana Consulta en el modo de presentación Diseño, con los elementos que la conforman.

Use el modo de presentación Diseño de una consulta para crear y modificar la estructura de una consulta

- La barra de título indica el tipo y el título de la consulta mostrada
- En la parte superior de la ventana, Microsoft Access sitúa las listas de campos de las tablas o consultas que puede agregar a la consulta. Las tablas están conectadas por líneas de combinación, y los símbolos que aparecen sobre estas líneas indican el tipo de relación entre las tablas
- En la cuadrícula QBE, los campos se agregan a la consulta arrastrándolos desde la lista de campos hasta la fila Campo, o bien, puede seleccionarlos en el cuadro de lista desplegable en cada celda de la fila. En las filas debajo de la fila Campo, puede establecer el orden de clasificación, ocultar un campo y establecer criterios
- En una consulta de selección, también puede agregar a la cuadrícula QBE una fila Totales para introducir cálculos de resumen y una fila Nombres de tablas para mostrar el nombre de la tabla de origen de cada campo. En el menú Ver, elija Totales o Nombres de tabla para mostrar esta filas.
- Si cambia a una consulta de tabla de referencias cruzadas o a una consulta de acciones, Microsoft Access mostrará otras filas en la cuadrícula QBE que son necesarias para crear esas consultas

Figura A.8 Ventana Consulta en el modo de presentación Diseño

Para mostrar los datos recuperados por la consulta, se utiliza el modo de presentación Hoja de Datos. (Figura A.9)

Selectores de registro
Haga clic aquí para seleccionar todos los registros.

Tabla: Categorías		
ID de categoría	Nombre de categoría	Descripción
B087	Bebidas	Gaseosas, café, té
C088	Condimentos	Salsas, dulces y pic
C188	Repostería	Postres, dulces y p
D091	Productos lácteos	Quesos
G089	Granos/Cereales	Pan, galletas, pasta

Nombres de campos

Registro. 1 de 8

Botones de desplazamiento

En el modo de presentación Hoja de datos se muestran múltiples registros en formato de filas y columnas, permitiendo así visualizar varios registros a la vez. Es posible usar una hoja de datos para agregar y editar datos, aunque, en general, el formulario proporciona más flexibilidad para estas operaciones.

Figura A.9 Ventana Consulta, en el modo de Presentación Hoja de Datos.

El modo de Presentación Preliminar nos sirve para ver la apariencia que tendrán los datos de la consulta una vez impresos. (Figura A.10)

Gastos de envíos
28. Sep. 93

Nombre de compañía	ID de pedido	Gastos de envío
Federal Shipping:	10516	\$62.78
	10519	\$91.76

Página 1

Botones de desplazamiento

Use la ventana Presentación preliminar para ver el informe, hoja de datos o módulo activo como aparecerá cuando se imprima.

Figura A.10 Ventana consulta, en el modo de Presentación Preliminar

Cuando se desee crear o modificar una consulta mediante inscripciones SQL ⁸, habrá que utilizar el modo de Presentación SQL. (Figura A.11)

⁸ SQL (Structured Query Language, Lenguaje de Consulta Estructurado) El SQL inicialmente fue desarrollado como un lenguaje para realizar accesos a una base de datos basada en el modelo relacional. La versión inicial de este lenguaje se conoció con SEQUEL y fue desarrollado por IBM como un lenguaje estándar que corría en mainframes de la misma compañía. Posteriormente su nombre fue abreviado a SQL.

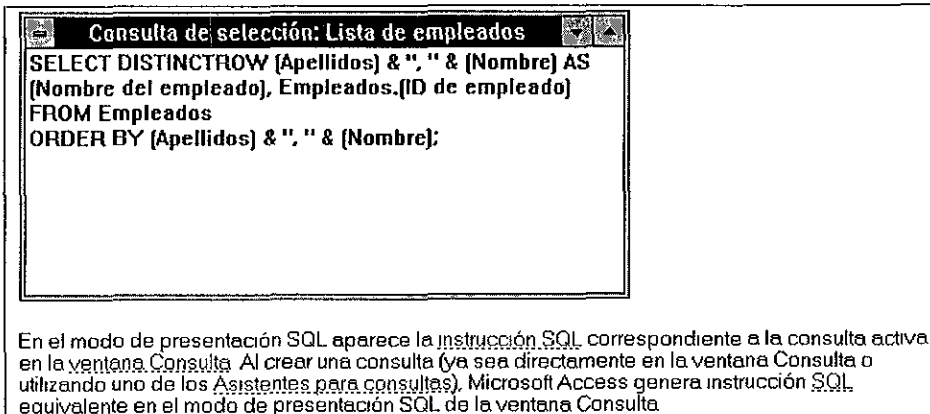


Figura A.11 Ventana consulta, en el modo de Presentación SQL

Dentro de la ventana Presentación SQL, se puede introducir cualquier instrucción SQL, y mientras se edita la instrucción SQL, también se tiene acceso a los comandos del menú y a otras ventanas de Microsoft Access.

Tipos de Consultas.

Microsoft Access 2.0, cuenta con los siguientes tipos de consultas.

- **Consultas de Selección.** Son las más comunes, aquí se plantean preguntas sobre datos almacenados en las tablas y la respuesta es una hoja de datos, en donde se puede ver y cambiar los datos de las tablas subyacentes.
- **Consulta de Tabla de Referencias Cruzadas.** Presenta los datos con títulos en las filas y columnas, como en una hoja de cálculo. Con una consulta de tabla de referencias cruzadas se puede resumir gran cantidad de información en un formato de fácil lectura.
- **Consultas de Unión.** Combina campos coincidentes de dos o más tablas.
- **Consultas de Definición de Datos.** Crea, modifica, o elimina tablas de una base de datos de Microsoft Access utilizando instrucciones de SQL.
- **Consultas de Acción.** Se pueden modificar muchos registros en una sola operación. Se utiliza para crear una nueva tabla, eliminar registros de una tabla, añadir nuevos registros o modificarlos. A partir de la selección, se actualizan, agregan y eliminan datos. Microsoft Access provee cuatro tipos de consultas de este tipo que son: consultas de creación de tablas, consultas de eliminación, consultas de datos añadidos y consultas de actualización.

Por medio de las consultas, es posible:

- Elegir campos: No es necesario incluir todos los campos de una tabla.
- Elegir registros: Se pueden especificar los criterios que deben cumplir los registros para que se incluyan en la hora de respuesta dinámica de la consulta.

- **Ordenar registros:** Se pueden ver los registros en un orden determinado. Para mostrar los datos ordenados en un formulario, se puede crear una consulta que ordene los registros y a continuación utilizarla como origen de los datos para el formulario.
- **Formular preguntas sobre datos de varias tablas:** Se puede utilizar una consulta para responder a una pregunta sobre datos procedentes de más de una tabla y ver los resultados en una sola hoja de datos.
- **Realizar cálculos:** Se pueden crear nuevos campos, llamados campos calculados, que contengan el resultado de un cálculo. Para mostrar campos calculados en formularios o informes, se puede crear una consulta que contenga el campo calculado y basar en ella el formulario o informe.
- **Usar una consulta como origen de los datos para crear formularios, informes y otras consultas.** Para seleccionar sólo los datos que deben aparecer en un formulario o informe, también se puede crear una consulta de selección y utilizarla como origen de los datos del formulario o informe. Utilizando una consulta se pueden incluir datos de varias tablas y establecer criterios para mostrar sólo un conjunto limitado de datos. Cada vez que se abra el formulario o imprima el informe, la consulta recuperará información actualizada de las tablas. También se pueden introducir nuevos datos o modificar los existentes directamente en la hoja de datos de la consulta, o bien en un formulario que esté basado en ésta consulta.
- **Modificar los datos de las tablas:** Con las consultas de acciones se puede actualizar, añadir o eliminar un grupo de registros simultáneamente. También se pueden utilizarlas para crear una nueva tabla que contenga registros de otra tabla o de un grupo de tablas existentes.

Características de los Formularios

Los formularios de Microsoft Access son muy versátiles: pueden ser utilizados para introducir, modificar, ver e imprimir datos.

Al igual que un impreso en papel, que se complementa con un bolígrafo o un lápiz, un formulario de Microsoft Access identifica los datos que se desean recopilar. No obstante también proporciona una forma de introducir los datos en la base de datos.

Con Microsoft Access se pueden diseñar formularios fáciles de utilizar y que presenten la información en forma deseada. Se pueden utilizar diversos elementos de diseño (texto, datos, imágenes, líneas y color) para crear exactamente el formulario deseado.

La figura A.12 muestra la ventana Formulario, con sus diferentes modos de presentación.

Use la ventana Formulario para crear y usar formularios en Microsoft Access. Puede mostrar la ventana Formulario en cualquiera de las tres presentaciones

Modo de presentación	Propósito
<u>Diseño</u>	Crea y modifica la estructura de un formulario
<u>Formulario</u>	Muestra los datos de registro en registro. En este modo de presentación puede ver, agregar y editar los datos.
<u>Hoja de datos</u>	Muestra los datos en formato de filas y columnas. En este modo de presentación puede ver, agregar y editar los datos
<u>Presentación preliminar</u>	Muestra la apariencia que tendrá un formulario al ser impreso

Figura A.12 Ventana Formulario, con sus modos de presentación.

El modo de presentación Diseño se utiliza para crear un formulario o cambiar la estructura de un formulario existente. (Figura A 13).

Use el modo de presentación Diseño de la ventana Formulario para crear y modificar un formulario. Puede ver cómo se ve su formulario en la pantalla haciendo clic en el botón "Presentación Formulario" o cómo aparecerá impreso haciendo clic en el botón "Presentación preliminar"

Elemento	Descripción
Sección Detalle	Contiene el cuerpo principal del formulario. Ubique los cuadros de texto y otros controles relacionados con los campos de sus tablas en esta sección. La sección Detalle no se puede quitar.
Secciones de encabezado/pie del formulario	Contienen información tal como un título, una fecha o los encabezados de columnas que desee que aparezcan sólo en la parte superior o inferior de un formulario, ya sea en pantalla o al imprimir. Para agregar o eliminar estas secciones (como una unidad encabezado-pie), en el menú Formato , elija <u>Encabezado/pie del formulario</u>
Secciones de encabezado/pie de página	Contienen información como por ejemplo los encabezados de columnas o un número de página que se desea que aparezca en la parte superior o inferior de cada página, pero sólo en el formulario impreso. Para agregar o eliminar estas secciones (como una unidad encabezado-pie), en el menú Formato , elija <u>Encabezado/pie de página</u>
Reglas	Le ayudan a colocar los <u>controles</u> y a determinar el tamaño de las <u>secciones</u> en su formulario. Para mostrar u ocultar las reglas, en el menú Ver , elija <u>Regla</u> .
Cuadrículas	Le ayuda a colocar los controles en su formulario. Si la cuadrícula es demasiado <u>linda</u> , no se podrá ver. Para mostrar u ocultar la cuadrícula, en el menú Ver , elija <u>Cuadrícula</u>

Figura A.13 Ventana Formulario, modo de Presentación Diseño.

El modo de presentación Formulario se utiliza para introducir, modificar y ver datos utilizando para ello el formulario propiamente dicho. En este modo de presentación, normalmente se puede ver al mismo tiempo todos los campos correspondientes a un registro.

En la figura A.14 se muestra la ventana Formulario, en el modo de presentación Formulario, con los elementos que la componen.

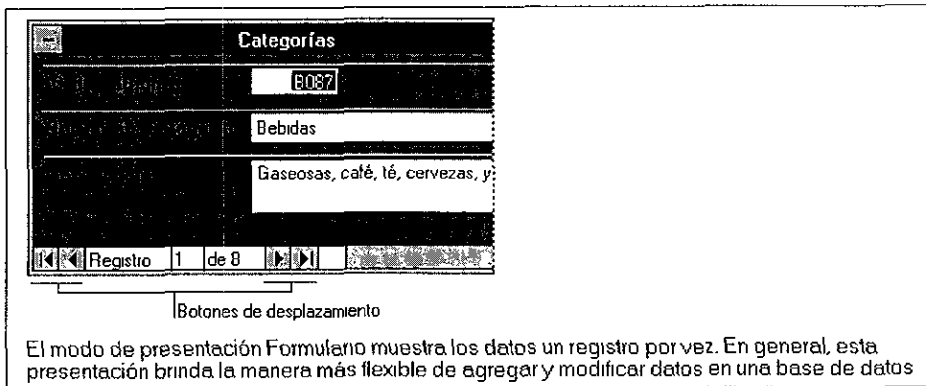


Figura A.14 Ventana Formulario, modo de presentación Formulario.

El modo de presentación Hoja de Datos se utiliza para introducir, modificar y ver datos utilizando para ello una hoja de datos. En este modo de presentación se pueden ver muchos registros al mismo tiempo, pero puede que la hoja de datos sea mucho más ancha

que la pantalla, en cuyo caso no se podrán ver todos los campos correspondientes a cada registro sin desplazarse por el formulario. (Figura A.15)

Selector de registro
Haga clic aquí para seleccionar todos los registros

Tabla: Categorías		
id categoría	Nombre de categoría	Descripción
B087	Bebidas	Gaseosas, café, té
C088	Condimentos	Salsas, dulces y pic
C188	Repostería	Postres, dulces y p
D091	Productos lácteos	Quesos
G089	Granos/Cereales	Pan, galletas, pasta

Nombres de campos

Registro: 1 de 8

Botones de desplazamiento

En el modo de presentación Hoja de datos se muestran múltiples registros en formato de filas y columnas, permitiendo así visualizar varios registros a la vez. Es posible usar una hoja de datos para agregar y editar datos, aunque, en general, el formulario proporciona más flexibilidad para estas operaciones

Figura A.15 Ventana Formulario, modo de presentación Hoja de Datos.

El modo de Presentación Preliminar se utiliza para ver la apariencia que tendrá un formulario cuando se imprima. (Figura A.16)

Gastos de envíos		
28-Sep-99		
Nombre de compañía:	ID de pedido:	Gastos de envío:
Federal Shipping:	10516	\$62.78
	10519	\$91.76

Página 1

Botones de desplazamiento

Use la ventana Presentación preliminar para ver el informe, hoja de datos o módulo activo como aparecerá cuando se imprima.

Figura A.16 Ventana Formulario, modo de Presentación Preliminar.

Los formularios proporcionan una forma sencilla de ver los datos. Se pueden ver todos los valores de un registro en el modo de presentación Formulario o bien se puede cambiar al modo de presentación Hoja de Datos para ver todos los registros del formulario.

El uso de formularios también es un modo eficiente de introducir los datos, pues ahorra tiempo y se evitan errores de mecanografía. Por ejemplo, en lugar de escribir los valores de todos los campos, se pueden crear listas en el formulario a partir de las cuales se

El formulario ofrece el diseño más cómodo para introducir, modificar y ver los registros de la base de datos.

Subformularios.

Los subformularios son un método para incluir en un formulario información que proviene de más de una tabla. Un subformulario es un formulario dentro de otro formulario.

En Microsoft Access, el formulario principal se denomina formulario principal y el formulario incluido en él recibe el nombre de subformulario. Al utilizar un subformulario es sencillo ver la relación existente entre los registros de dos o más tablas.

Por ejemplo, en la figura A.17, el formulario principal Categorías de Neptuno muestra toda la información sobre una categoría de productos y el subformulario que contiene, muestra los productos pertenecientes a esa categoría.

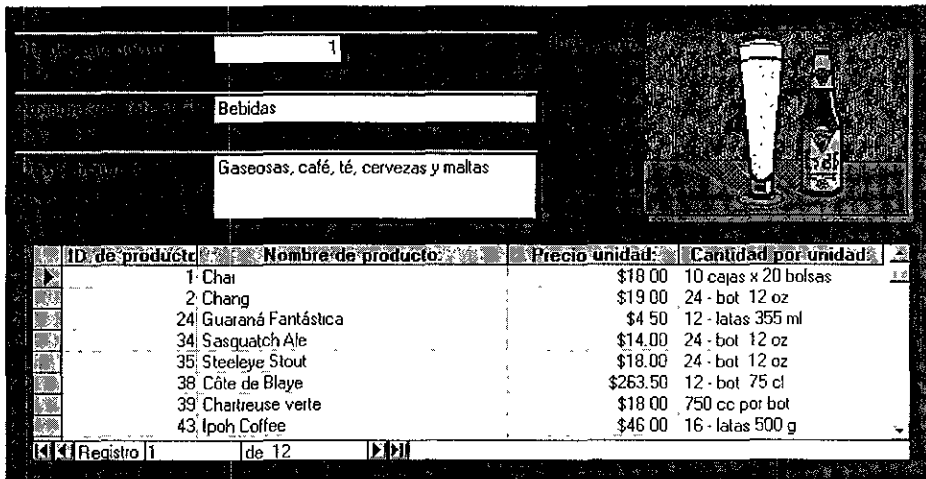


Figura A.17 Formulario Categorías de la base de datos Neptuno. (Base de datos ejemplo que se incluye en Microsoft Access 2.0)

Además, utilizando un Asistente se puede crear rápidamente un formulario básico que muestra todos los campos y los registros de una tabla o consulta, o también se puede crear un formulario personalizado que muestre los datos de registros seleccionados con un formato determinado, por ejemplo como un gráfico.

Se tienen cinco tipos de Asistentes:

- Asistente para formularios/informes de columna simple.

Muestra los valores de cada registro en una sola columna. Cada valor aparece en una línea distinta, con una etiqueta a su izquierda, y los registros aparecen a lo largo de la página.

- Asistente para formularios/informes tabulares

Muestra los valores de registros en columnas dispuestas de izquierda derecha en un formulario o informe. Encima de cada columna aparece una etiqueta, y cada fila representa un nuevo registro.

- Asistente para gráficos

Una herramienta de Microsoft Access que crea un formulario que contiene un gráfico.

- Asistente para formulario/subformulario

Creación de un formulario con un subformulario planteando ciertas preguntas y luego crea el formulario y el subformulario en base a las respuestas.

- Asistente Formulario Automático

Creación rápida y automáticamente un formulario que muestra los valores de todos los campos y registros de la tabla o la consulta seleccionada.

Características de los Informes

Un informe es un conjunto de información que se organiza de acuerdo a determinados criterios y al que se le aplica un formato según especificaciones. Un informe constituye una forma de recuperar y presentar los datos como información significativa que se puede usar y distribuir.

Con Microsoft Access, se pueden diseñar informes que presenten la información de la forma deseada. Se pueden emplear numerosos elementos de diseño (texto, datos, imágenes, líneas, cuadros y gráficos) para crear exactamente el informe que se necesite. Crear un informe es un método muy eficaz de presentar los datos en forma de documento impreso. Si bien es posible imprimir los formularios y las hojas de datos, los informes brindan mayor flexibilidad a la hora de presentar información de resumen. En la figura A.18 de muestra la Ventana Informe con sus diferentes modos de presentación.

Use la ventana Informe para crear y ver una presentación preliminar de los informes. La ventana Informe se puede abrir en cualquiera de las tres presentaciones

Modo de presentación	Propósito
Diseño	Crea y modifica la estructura de un informe
Presentación preliminar	Muestra un informe tal como aparecerá cuando se imprima
Muestra preliminar	Muestra el diseño básico de un informe (La ventana Muestra preliminar sólo está disponible para los informes abiertos en el modo de presentación Diseño)

Figura A.18 Ventana Informe, con sus modos de presentación.

El modo de presentación Diseño se utiliza para cambiar el diseño de un informe existente o para crear uno nuevo. (Figura A.19)

Informe: Lista de Productos por Cate				
0	1	2	3	
Encabezado del informe				Sección Encabezado del informe
0	Lista de productos por categoría			
1	Fecha			
Encabezado de página				Sección Encabezado de página
0	Categoría:	Producto:	ID de produ	
Encabezado: Nombre de categoría				Sección Encabezado del grupo
0	Nombre de categoría			
Detalle				Sección de detalle
0	Nombre de producto	ID de produ		
Pie: Nombre de categoría				Sección Pie del grupo
0	Total: =Cuenta([ID de producto])			
Pie de página				Sección del pie de página
0		=Página		
Pie del informe				Sección del pie del informe
0				

Elemento	Propósito
Sección Detalle	Contiene el cuerpo principal del informe. Ubicar cuadros de texto y otros controles que muestran registros de la tabla o consulta base en esta sección. No es posible eliminar la sección Detalle, pero puede dejarla en blanco u ocultarla usando la propiedad <u>Visible</u> .
Secciones de encabezado/pie del informe	Contienen información, como por ejemplo, un título, fecha o resumen de informe que se desea que aparezca sólo una vez en la parte superior o inferior de un informe. Para agregar o eliminar estas secciones (como una unidad encabezado-pie), en el menú Formato , elija <u>Encabezado/Pie del informe</u> .
Secciones de encabezado/pie de página	Contienen información, como por ejemplo, encabezados de columnas o un número de página que se desea que aparezca en la parte superior o inferior de cada página de un informe. Para agregar o eliminar estas secciones (como una unidad encabezado-pie), en el menú Formato , elija <u>Encabezado/Pie de página</u> .
Secciones de encabezado y pie de grupo	Contienen información, como por ejemplo, un nombre de grupo o totales de grupo que se desea que aparezcan al comienzo o final de un grupo de registros. Microsoft Access agrega automáticamente la sección apropiada cuando define un encabezado o pie de grupo en el cuadro <u>Ordenar y agrupar</u> .
Reglas	Le ayudan a colocar los <u>controles</u> y determinar el tamaño de las <u>secciones</u> de su informe. Para mostrar u ocultar las reglas, en el menú Ver , elija <u>Regla</u> .
Cuadrícula	Le ayudan a colocar los controles en su informe. Si la cuadrícula es demasiado <u>linda</u> , no se podrá ver. Para mostrar u ocultar la cuadrícula, en el menú Ver , elija <u>Cuadrícula</u> .

Figura A.19 Ventana Informe, modo de presentación Diseño.

En el modo Presentación Preliminar se utiliza para comprobar los datos del y informe. (Figura A.20)

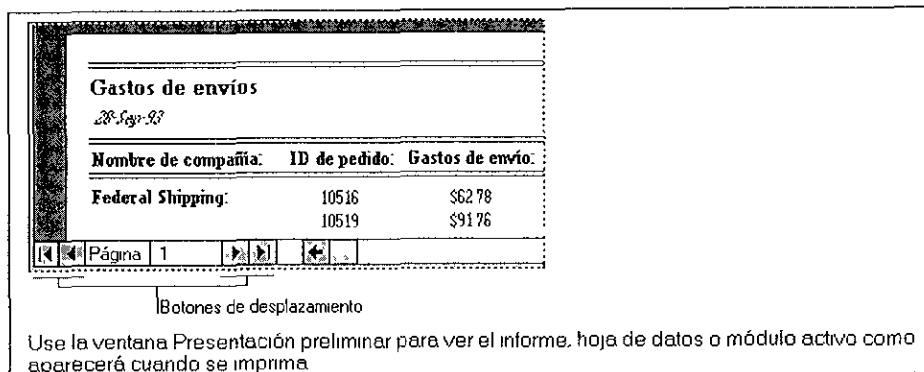


Figura A.20 Ventana Informe, modo de Presentación Preliminar.

El modo Muestra Preliminar se utiliza para comprobar la fuente y el diseño general del informe. (Figura A.21)

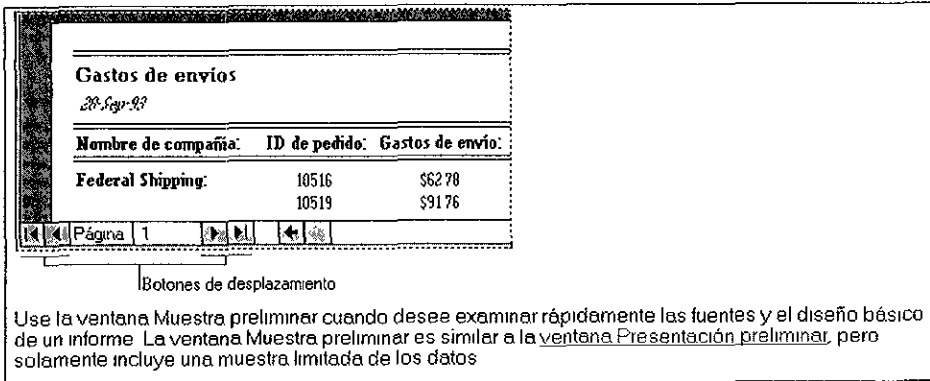


Figura A.21 Ventana Informe, modo de Muestra Preliminar.

Con Microsoft Access se pueden crear informes para:

- Organizar y presentar los datos en grupos
- Calcular totales parciales, totales de grupo, sumas de totales y porcentajes de totales.
- Presentar los datos en un formato atractivo, con imágenes, líneas y fuentes especiales.

También se cuenta con un Asistente para informes que nos ayuda a crear informes planteando preguntas sobre el informe que se desea crear y creándolo de acuerdo a las respuestas.

Microsoft Access cuenta con tipos de Asistentes para Informes:

- Asistente para formularios/informes de columna simple

Muestra los valores de cada registro en una sola columna. Cada valor aparece en una línea distinta, con una etiqueta a su izquierda, y los registros aparecen a lo largo de la página.

- Asistente para informes de grupos/totales

Organiza los datos en grupos que se presentan en un formato tabular. Se puede calcular un subtotal para cada grupo y un total general para todos los grupos. Las etiquetas aparecen al principio de cada columna de datos.

- Asistente para informes de etiquetas postales

Una herramienta de Microsoft Access que crea etiquetas postales.

- Asistente para informes de resumen

Un informe de resumen organiza los datos en grupos y puede incluir un subtotal para cada grupo y una suma total para todos los grupos. Los datos agrupados pueden presentarse en formato tabular. Un informe de resumen es similar a un Informe de grupos/totales, salvo que un Informe de resumen excluye los registros de detalle del informe impreso.

- Asistente para formularios/informes tabulares

Muestra los valores de registros en columnas dispuestas de izquierda derecha en un formulario o informe. Encima de cada columna aparece una etiqueta, y cada fila representa un nuevo registro.

- Asistente para combinar correspondencia con Microsoft Word

Una herramienta de Microsoft Access que nos ayuda a combinar datos de una tabla o consulta de Microsoft Access en un documento de Word para Windows versión 6.0 para crear cartas modelo y otros documentos combinados.

Características de las macros.

Una macro realiza automáticamente una tarea o usa serie de tareas. Cada tarea que se desea realizar con Microsoft Access se denomina **acción**. En la figura A.22 se muestra la Ventana Macro, con los elementos que la componen.

Columna de nombres de macros	Columna de de acciones	Columna de comentarios
Macro: Botones del Panel de control de formularios		
Nombre de macro:	Acción	Comentario
Categorías	AbrirFormulario	Abrir el formulario Categorías
Cientes	AbrirFormulario	Abrir el formulario Cientes.
Empleados	AbrirFormulario	Abrir el formulario Empleados
Productos	AbrirFormulario	Abrir el formulario Productos
Argumentos de acción		
Nombre del formulario:	Categorías	
Presentación:	Formulario	
Nombre de fila:		
Condición donde:		
Modo de datos:	Editar	
Modo de ventana:	Normal	
Argumentos de acción		

Use la ventana Macro para **crear y modificar macros**. éstas se usan para automatizar tareas en Microsoft Access

Elemento	Propósito
Columna Nombres de macros	Mostrar esta columna si se define más de una macro dentro de una macro (grupo de macros). En la barra de herramientas, haga clic en el botón "Nombres de macros" para mostrar u ocultar la columna.
Columna Condiciones	Muestre esta columna si desea escribir una expresión que establezca condiciones para una acción. En la barra de herramientas, haga clic en el botón "Condiciones" para mostrar u ocultar la columna.
Columna de acciones	Elegir acciones en el cuadro de lista desplegable en el orden en que se desea que ocurran.
Columna de comentarios	Agregar comentarios (opcional).
Argumentos de acciones	Seleccione o escriba información adicional sobre la acción. Los argumentos disponibles dependen de la acción seleccionada.
En la parte inferior derecha de la ventana, Microsoft Access muestra una breve descripción de la columna, acción o argumento seleccionado.	

Figura A.22 Ventana Macro

Microsoft Access ofrece una lista de acciones, entre las que se puede seleccionar una o más para crear una macro. Cuando se ejecute la macro, Microsoft Access realizará las acciones en el mismo orden en que aparecen dentro de la macro, utilizando los objetos o los datos especificados.

Mediante el uso de macros se puede conseguir que los formularios. Los informes y los demás objetos de la base de datos funcionan conjuntamente de forma aún más inteligente. También son útiles para automatizar tareas rutinarias o repetitivas, como la impresión de informes semanales. Además, las macros ofrecen versátiles posibilidades de personalización tales como realizar complejas validaciones de los datos introducidos en formularios.

Las macros nos permiten:

- Hacer que los formularios e informes funcionen conjuntamente: Se pueden utilizar dos formularios juntos para ver datos relacionados entre sí.
- Buscar y filtrar registros automáticamente: Las macros pueden acelerar el proceso de búsqueda de los registros deseados.
- Establecer valores en controles: Las macros permiten asignar a un control de formulario el valor resultante de un cálculo o un valor procedente de otra tabla.
- Asegurar la exactitud de los datos: Las macros también son idóneas para el manejo de datos y para su visualización en formularios. Por ejemplo, se puede utilizar una macro para responder con distintos mensajes a los diferentes tipos de datos almacenados en un campo.
- Establecer propiedades de formularios, informes y controles: Se pueden utilizar macros para establecer la mayor parte de las propiedades de formularios, informe y control. Por ejemplo, se puede usar una macro para ocultar un formulario cuando no sea necesario verlo en la pantalla pero si utilizar los datos incluidos en él.
- Automatizar transferencias de datos: Se puede usar una macro para importar automáticamente los datos entre distintos formatos de archivos.
- Crear un entorno de trabajo personalizado: Se puede utilizar una macro para abrir automáticamente un grupo de consultas, formularios e informes cuando se abra la base

de datos. Además, las macros nos permiten personalizar las barras de menús de los formularios. En la figura A.23 se muestra una macro llamada Autoexec, que se ejecuta automáticamente al abrir la Base de Datos. Esta macro utilizará tres acciones para indicar a Microsoft Access que: active la acción Eco, posteriormente que active el reloj de arena y por último que abra el formulario.

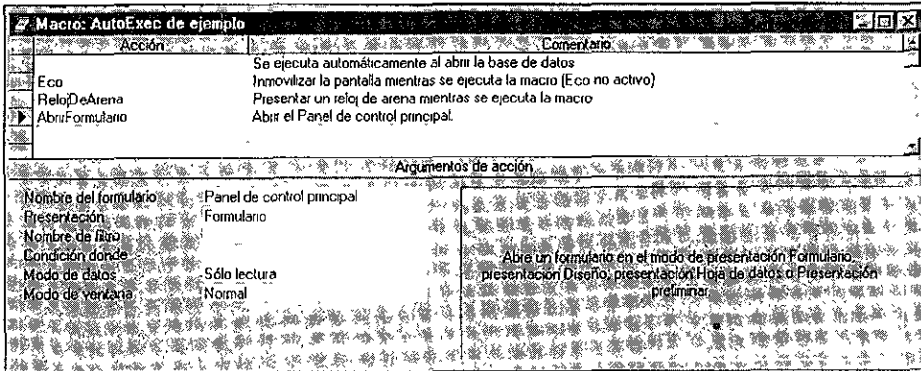


Figura A.23 Macro Autoexec

Características de los módulos.

Los módulos almacenan código de Access Basic⁹ que puede escribirse para personalizar, mejorar y extender las posibilidades de la base de datos.

La figura A.24 muestra la Ventana Módulo, con los elementos que la conforman.

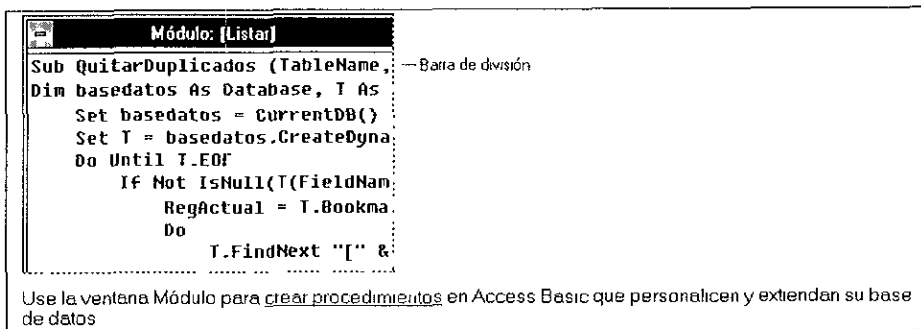


Figura A.24 Ventana Módulo

⁹ Lenguaje de programación que Microsoft Access incorpora para poder desarrollar aplicaciones, este lenguaje ofrece muchas características del lenguaje Visual Basic de Microsoft

En la siguiente figura (A.25), se muestra el módulo Funciones de Utilidad, que pertenece a la base de datos de ejemplo Neptuno.

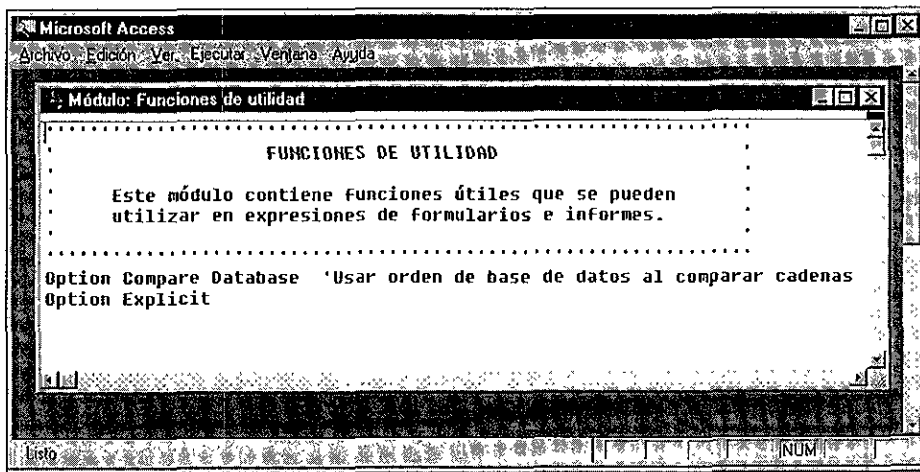


Figura A.25 Módulo Funciones de utilidad (Neptuno)

Microsoft Access también cuenta con una ayuda en pantalla que nos proporciona información de referencia e instrucciones prácticas para todas las tareas, también contiene información de referencia completa en orden alfabético sobre objetos, propiedades, acciones, funciones, instrucciones y métodos

Se tiene además, Fichas-Guía que son asistentes en pantalla que nos dan indicaciones sobre la forma de realizar una determinada tarea mientras se trabaja con Microsoft Access. Por ejemplo, si necesitamos ayuda para diseñar una tabla, podemos consultar la ficha-guía correspondiente.(Figura A.26).

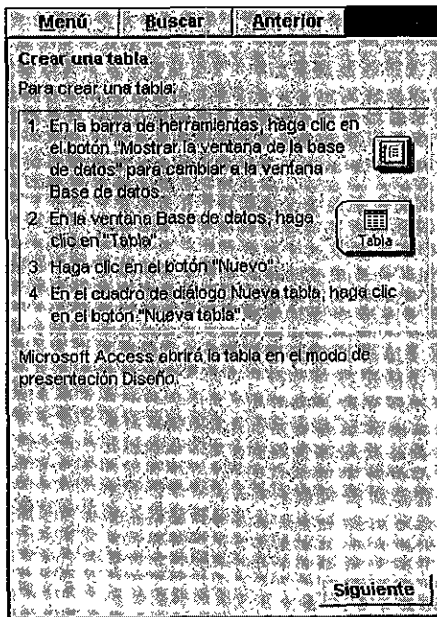


Figura A.26 Ficha-Guía para crear una tabla.

Requisitos de Hardware y Software de Microsoft Access 2.0

Para poder instalar y utilizar Microsoft Access 2.0, es necesario disponer de los siguientes elementos.

- MS-DOS versión 3.1 o posterior (Se recomienda 6.2 en adelante)
- Windows 3.1, Windows para Trabajo en Grupo 3.11
- Mouse
- Procesador 80386SX o posterior
- 4 MB en memoria RAM
- 6 MB de espacio en disco duro para la instalación mínima ó 23 MB para la instalación completa.
- Unidad de disquetes de alta densidad de 3 1/2"
- Monitor de color (se recomienda SVGA)
- Impresora

Capacidad del Sistema

Base de Datos

Tamaño máximo de la Base de Datos	1 gigabyte. (Sin embargo, debido a que la base de datos puede incluir tablas de otros archivos, su tamaño total sólo está limitado por la capacidad de almacenamiento disponible.)
Número de objetos en la base de datos	32,768
Número de caracteres para el nombre de un objeto	64
Número de caracteres en un password	14
Número de caracteres para un nombre de usuario o un grupo de trabajo	20
Número de usuarios simultáneos	255

Tablas

Número de caracteres en el nombre de una tabla	64
Número de caracteres para el nombre de un campo	64
Número de campos en un registro	255
Tamaño de la tabla	1 Gigabyte
Número de tablas abiertas	1.024. El número real puede ser menor debido a tablas que abre Microsoft Access internamente
Número de caracteres para un campo tipo texto	255
Número de caracteres para un campo Memo	64,000
Tamaño para un campo de tipo OLE	1 Gigabyte
Número de índices para una tabla	32
Número de caracteres en una regla de validación	2,048
Número de campos para un índice	10
Número de caracteres para la descripción de un campo	255

Consultas

Número de tablas en una consulta	32
Número de campos para un dynaset	255
Número de campos ordenados	10
Número de niveles de consultas anidadas	50

Número de caracteres para un parámetro en una consulta de parámetros	255
Número de caracteres en una sentencia SQL	64,000

Formularios y Reportes

Número de caracteres para una etiqueta	2048
Número de caracteres en una caja de texto	64,000
Anchura para un formulario o reporte	55.87 cm
Número de niveles de formularios o informes anidados	3
Altura del formulario o informe en presentación diseño	508 cm
Número de campos o expresiones que se pueden agrupar (solamente informes)	10
Número de encabezados o pies de página en un reporte	1 encabezado/pie de informe; 1 encabezado/pie de página; 10 encabezados/pies de grupo
Número de páginas impresas en un reporte	65,536

Macros

Número de acciones en una macro	999
Número de caracteres en una condición	255
Número de caracteres para un comentario	255
Número de caracteres para argumentos de acción	255

Microsoft Access para Windows 95

Como ocurre con la mayoría de los programas y paquetes con miles de usuarios, por muy buenos que sean, van apareciendo nuevas versiones con mejoras y actualizaciones, así como versiones que se acoplan a nuevos sistemas operativos, y que ofrecen ventajas y funciones mejoradas con respecto a las versiones anteriores. La versión 2.0 incluyó mejoras y novedades que complementaron la primera versión y permitió la aplicación a todo tipo de usuarios y el desarrollo de todo tipo de tareas empresariales y particulares.

Al aparecer Windows 95¹⁰[1], Microsoft no podía dejar atrás la actualización y adaptación de las versiones de los propios paquetes y creó una nueva versión actualizada de

¹⁰ Un sistema operativo para los microprocesadores basados en Intel, como 80386DX, 486 y Pentium. Windows 95 está diseñado para ejecutar aplicaciones Windows, incluyendo aplicaciones multimedia. Una interfaz gráfica de usuario (GUI) rediseñada por completo facilita su aprendizaje y uso cotidiano.

Access, con las mejoras y nuevas funciones, aprovechando la potencia del nuevo sistema operativo.

Si se dispone de la versión anterior de Access, se podrá trabajar con ella desde Windows 95, pero no se dispondrá de las nuevas funciones ni mejoras de la versión para Windows 95.

Requisitos de Hardware y Software de Microsoft Access para Windows 95

Access para Windows 95 ha sido diseñado para ejecutarse en una computadora IBM o compatible, con un procesador 80386 SX o superior, aunque se recomienda un procesador 80486 con las características siguientes:

- ☐ 12 MB en RAM para uso de Windows 95
- ☐ 16 MB en RAM para uso de Windows NT ¹¹ Workstation.
- ☐ El disco duro deberá disponer de 16 MB de espacio libre si se realiza la instalación compacta, 39 MB si se va a realizar la instalación típica y 53 MB si se realiza la instalación personalizada con todos los elementos posibles.
- ☐ Una unidad de disquetes de alta densidad de 3 1/2" o una unidad de CD-ROM, dependiendo del tipo de discos que se utilizarán en la instalación.
- ☐ Un ratón
- ☐ Un adaptador de video VGA o con una resolución superior. Se recomienda SVGA de 256 colores.
- ☐ Microsoft Windows 95 o Microsoft Windows NT Workstation versión 3.51 o posterior.

Mejoras de Microsoft Access para Windows 95

• *Nuevo sistema de ayuda*

Esta versión incluye la posibilidad de obtener ayuda mediante el Asistente para Ayuda, que permite acceder al tema o función escribiendo directamente la palabra o cuestión de la que se precisa ayuda.

El Asistente para Ayuda dispone de un índice por temas, por palabras y ejemplos que facilitan paso a paso toda la información que se precise.

• *Mejoras en la creación de bases de datos*

¹¹ Sistema operativo avanzado de 32 bits para computación cliente-servidor (en esta arquitectura el procesamiento de la base de datos se encuentra repartido en el cliente que generalmente son PCs y en el servidor. Los clientes se encargan de ejecutar la aplicación, y el servidor del manejador de base de datos y de compartir recursos como el espacio en disco, impresoras, etc.) diseñado para aprovechar al máximo las poderosas máquinas de hoy en día.

Ofrece la posibilidad de crear bases de datos mediante la incorporación del Asistente para la creación de bases de datos.

Para crear una base de datos, bastará con seleccionar un modelo entre los existentes y el Asistente creará las tablas, formularios y los informes que precise. Sólo se necesitará rellenar las tablas con los datos. De igual forma, el Asistente permitirá incluir en las nuevas tablas los datos procedentes de otras aplicaciones mediante la importación de texto, con lo cual se podrá introducir sólo una vez los datos a utilizar en varias aplicaciones.

Con el nuevo Asistente para tablas podrá elegir entre docenas de tablas y cientos de campos de ejemplo, siendo ACCESS quien creará las tablas automáticamente.

- *Novedades en formularios e informes.*

Las novedades de la nueva versión permitirá crear y manipular formularios e informes de una forma más rápida y sencilla:

- Menos trabajo para crear formatos de formularios e informes. Bastará con utilizar **Autoformato** para aplicar formatos de una forma rápida y automática.

- Copiar características para controles. La nueva versión dispone de un botón **Copiar Formato**, que permitirá copiar las características de formato de un control a otro de forma rápida y automática.

III.5.5 Microsoft Access 97

Al igual que las versiones anteriores de Access, Microsoft Access 97 [2] incluye un entorno gráfico para crear o modificar tablas, formularios, informes, consultas y macros. Así mismo, permite pasar objetos de un área de trabajo a otra, fusionar dos o más tablas dibujando una línea entre los campos comunes, trabajar con todos los datos, ordenarlos según se desee y elaborar todo tipo de informes.

Requisitos de Hardware y Software de Microsoft Access 97

- ☐ Procesador 486 ó superior
- ☐ 8 MB en RAM para Windows 95 para correr una aplicación, 12 MB para correr Access, y se requiere más memoria para correr otras aplicaciones.
- ☐ 16 MB en RAM para el uso de Windows NT sólo para correr una aplicación, es necesaria más memoria para correr otras aplicaciones.
- ☐ Una unidad de disquetes de alta densidad de 3 1/2" o una unidad de CD-ROM, dependiendo del tipo de discos que se utilizarán en la instalación.
- ☐ Un ratón

- ☑ Monitor VGA ó SVGA de 256 colores
- ☑ Windows 95
- ☑ Windows NT versión 4.0
- ☑ Windows NT versión 3.51 con Service Pack 5

A continuación se mencionan las características principales de esta nueva versión.

- Se pueden abrir muchos formularios e informes de manera más rápida, porque los formularios e informes que no tienen procedimientos de evento ya no incluyen un módulo de formulario o informe.
- Microsoft Access 97 cuenta ahora con diversas ayudas (en particular el Table Analyzer Wizard) para facilitar el diseño de bases de datos y pasar fácilmente de una base de datos a un documento en Microsoft Word, y a la inversa.
- Trabajando en Internet [3]. Microsoft Access proporciona muchas características nuevas diseñadas para ayudar a utilizar fácilmente Internet y programar aplicaciones para el World Wide Web[4].

Importar o vincular archivos HTML[5]	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
Asistente para publicar en el Web	Se 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.
Acceso a servidores FTP.	Importar o vincular datos (de sólo lectura) 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.

- Ejecutar utilidades de base de datos en una base de datos abierta. Ahora se pueden usar los subcomandos Compactar base de datos y Reparar base de datos en la base de datos actual.

A

P

e

n

d

í

e

e

B

Manual de Usuario

» **Requerimientos Básicos.**

El sistema de control académico para el Departamento de Sistemas de la DEFFI (SICAP), tiene como requerimientos básicos de hardware y software los siguientes:

- ☑ Una computadora PC 486 o superior con al menos 8 MB en RAM.
- ☑ 5 MB de espacio en disco duro para los archivos del sistema.
- ☑ Mouse IBM PS/2 o compatible.
- ☑ Unidad de disquetes de alta densidad de 3 1/2"
- ☑ Monitor de color (se recomienda SVGA)
- ☑ Impresora de Inyección de tinta o Láser
- ☑ MS-DOS versión 3.1 o posterior (Se recomienda 6.2 en adelante)
- ☑ Windows 3.1 o posteriores.
- ☑ Microsoft Access 2.0 (Instalación básica)

Para la instalación del sistema, es necesario crear el directorio SICAP en la raíz del disco duro, esto es:

```
C:\MD SICAP
```

Una vez que se ha creado el directorio se deberán copiar en él los siguientes archivos:

```
SICAP.MDB
UTIL.MDB
RENOM.BAT
AYUDA.HLP
SICAP.ICO
UTIL.ICO
```

Estos archivos se encuentran compactados en uno solo llamado SICAP.ZIP, para descompactarlos se utiliza un programa llamado PKUNZIP que también se proporciona en los discos y se utiliza de la siguiente manera:

```
C:\SICAP\A(o B):\PKUNZIP SICAP.ZIP -$ C:\
```

Es importante mencionar que el proceso de instalación que aquí se describe, será utilizando el ambiente de Windows 95.

A continuación será necesario crear un Acceso Directo en el escritorio de Windows, para esto utilizamos el explorador de Windows, seleccionamos con el Mouse el archivo llamado SICAP.MDB y:

1. En el menú Archivo, hacer clic en Crear acceso directo.
2. Arrastrar el icono de acceso directo hasta el escritorio.

Se sigue el mismo procedimiento ahora con el archivo UTIL.MDB. Para una mejor identificación de estos programas, se sugiere cambiar los iconos seleccionando los que se proporcionan con el sistema.

El escritorio se ve de la siguiente manera:

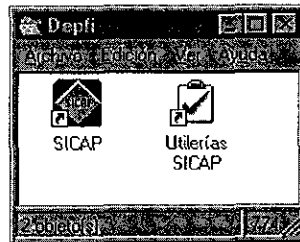


Figura B.1

► Inicio del sistema.

Para iniciar el programa, bastará con hacer doble clic con el botón izquierdo del mouse sobre el icono SICAP. Una vez activado el sistema, se presenta la siguiente pantalla:

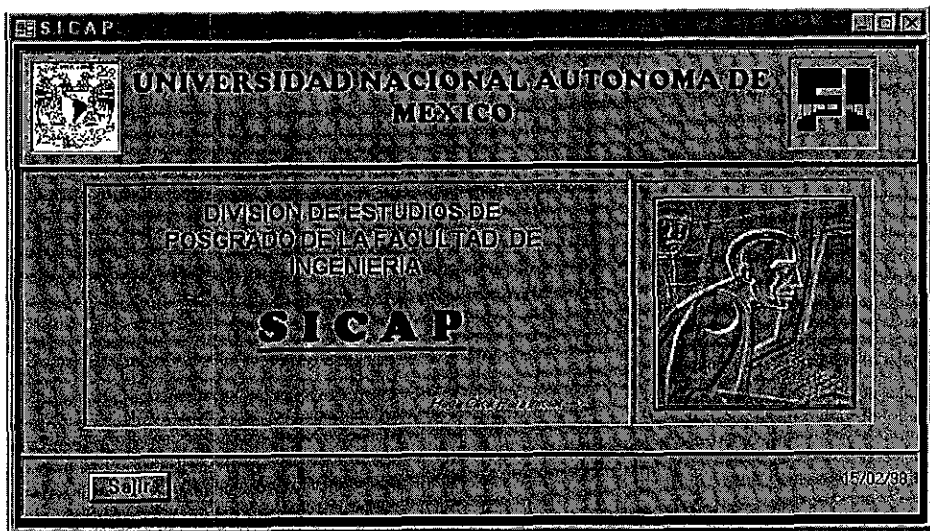


Figura B.2 Inicio del Sistema

► *Acceso al sistema*

En esta ventana hacer clic con el mouse sobre el botón (Figura B.3) para entrar al sistema. Se debe teclear entonces un password o clave para poder tener acceso. (Figura B.4). Si la clave de acceso se digitó incorrectamente el sistema mandará un mensaje de advertencia, en caso de que no se desee continuar con el sistema, hacer clic en el botón *Salir*.



Figura B.3

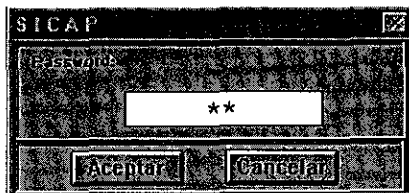


Figura B.4

Una vez que se haya proporcionado la clave correcta aparece la siguiente pantalla:

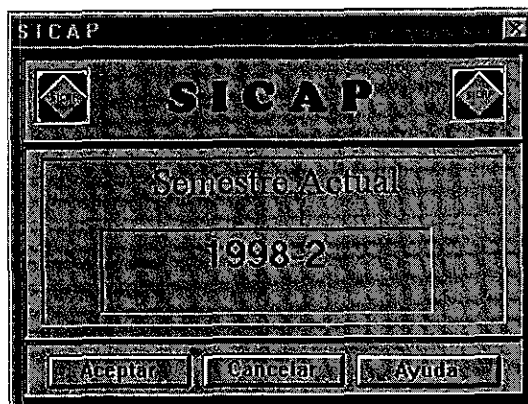


Figura B.5

El sistema está configurado para trabajar en el periodo 1998-2, si se desea trabajar en este semestre hacer clic en el botón *Aceptar*. Para cambiar el semestre, posicionar el puntero del mouse en este campo y modificarlo. Si no se quiere continuar con el sistema, hacer clic en el botón *Cancelar*. Para obtener más ayuda, sobre este tema, hacer clic en el botón *Ayuda*. A continuación aparecerá el Menú Principal. (Figura B.6)

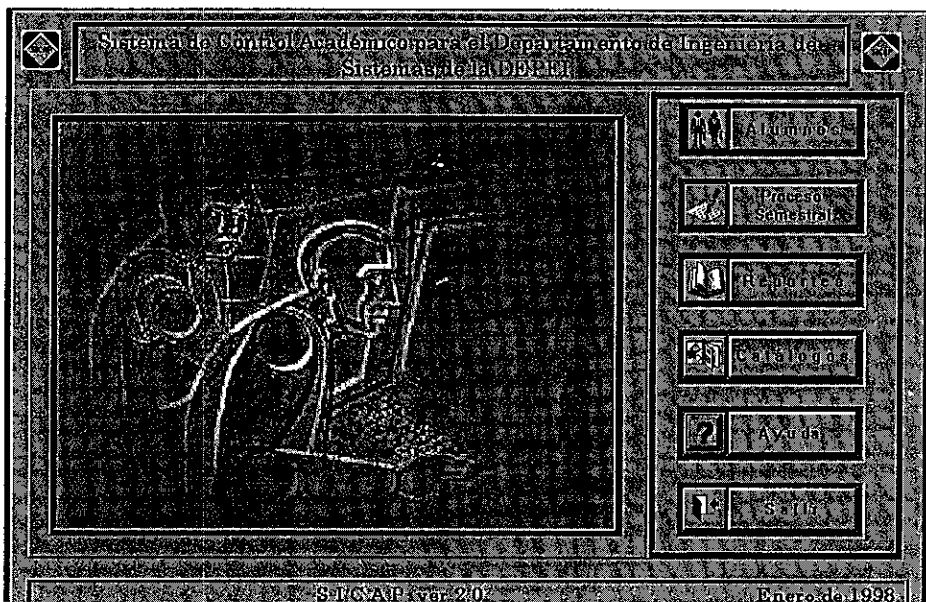


Figura B.7

La pantalla muestra los módulos principales que conforman al Sistema de Control Académico para el Departamento de Sistemas de la DEPFI, SICAP. Estos módulos son:

- ALUMNOS
- PROCESO SEMESTRAL
- REPORTES
- CATÁLOGOS
- AYUDA
- SALIR

MÓDULO 1: ALUMNOS

Para ejecutar este módulo bastará con dar un clic con el mouse sobre el botón con la leyenda ALUMNOS o bien, sobre el icono. Al hacer esto se despliega la pantalla de la figura B.8.

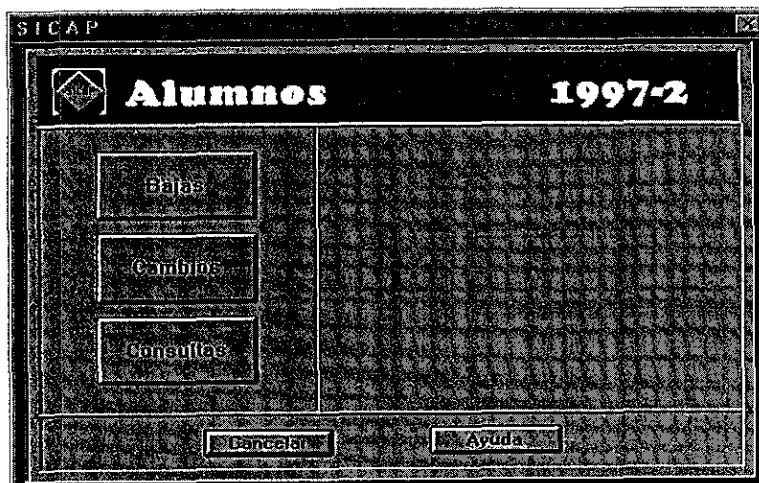


Figura B.8.

Tenemos entonces las siguientes opciones:

- Bajas
- Cambios
- Consultas

1.1 Bajas Alumnos.

Al seleccionar esta opción, se presenta la pantalla de la figura B.9, donde podemos *Cancelar la Inscripción Actual* de un Alumno o *Baja Definitiva* de mismo. En el caso de *Cancelar la Inscripción Actual*, se solicita el número de cuenta del alumno mediante la pantalla que aparece en la figura B.10. Si el número de cuenta no existe, el sistema manda un mensaje de error y lo mismo sucede si el número de cuenta no es válido. Si el alumno existe y además está cursando al menos una materia en el semestre actual, aparece una pantalla con las materias a las cuales está el alumno inscrito y habrá que escoger cuál o cuáles se cancelarán. (Figura B.11). Si lo que se desea es una *Baja Definitiva* del alumno, aparecerá también una pantalla similar a la de la figura B.10 solicitando el número de cuenta, si el alumno existe, entonces se desplegará información referente al alumno y que será se suma importancia considerar si es que se desea darlo de baja definitivamente Figura (B.12).

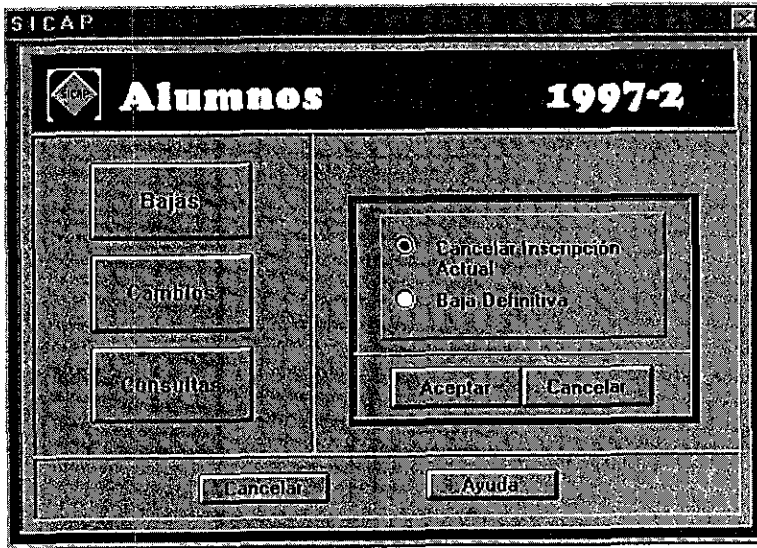


Figura B.9

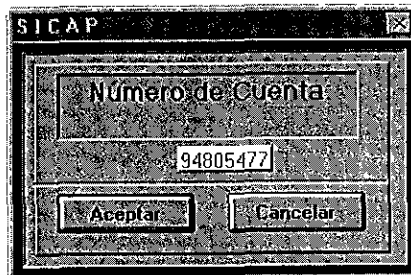


Figura B.10

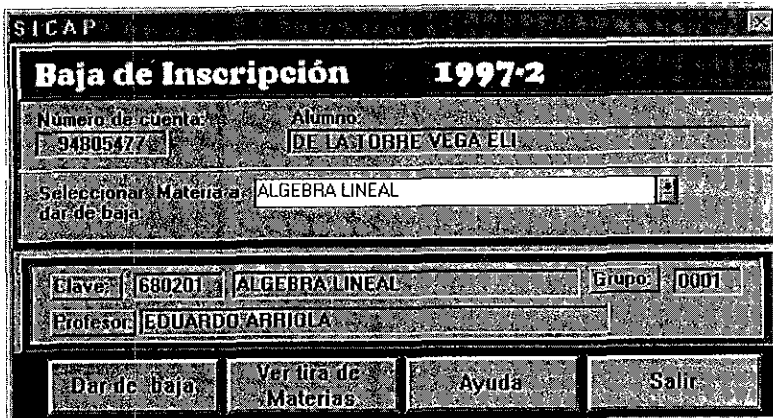


Figura B.11

SICAP

DATOS DEL ALUMNO

N. Cuenta: **94805477** N. Expediente: **1010123456**

Apellido P: **DE LA TORRE** Apellido M: **VEGA** Nombre: **ELI**

Dirección: **CDA ZAPOTLAN L-13,M-2, 15**

Colegio: **COYOACAN** Nacionalidad: Mexicana Extranjera

Ciudad: **DISTRITO FEDERAL** C.P.: **14130**

Año Semestre de Ingreso: **1997-1** Sexo: M F Nombre: **ACOSTA FLORES JOSE JESUS**

Asignatura: **INVESTIGACION DE OPERACIONES**

Institución: **INSTITUCION GUBERNAMENTAL**

Inscripción **Eliminar** **Cancelar** **Ayuda**

Figura B.12

Cuando se realiza la baja definitiva de un alumno, es importante saber si cursa o no materias en el semestre actual; esto se obtiene al presionar el botón *Inscrito en:* que se muestra en la figura B.12. Para proceder con la baja definitiva hacer clic en el botón *Eliminar*, de esta manera toda la información relacionada con ese alumno es eliminada definitivamente del sistema.

1.2 Cambios Alumnos.

Al elegir esta opción (Figura B.13), se presentan dos opciones: *Cambiar Datos Generales* y *Cambiar Plan de Estudios*. Si se elige *Cambiar Datos Generales* se habilitará entonces la ventana que solicitará el número de cuenta del alumno, al hacer clic sobre el botón *Aceptar* el sistema comenzará a realizar la búsqueda de este registro. Si el sistema lo encuentra, se irán desplegando cuatro ventanas con Datos Generales del Alumno (Nombre completo, N.- de Expediente Dirección, Teléfono, Nacionalidad, etc.), Antecedentes Académicos de Licenciatura (Nombre de la licenciatura de cursó, Escuela o Facultad donde la cursó, Promedio Obtenido, Fecha de Titulación, etc.), Antecedentes Académicos de Posgrado (Lo mismo para licenciatura) y Datos de Inscripción del alumno. (Periodo de Ingreso, Area o Campo al que pertenece, Tipo de Beca y Nombre del Tutor). En la figura B.14 únicamente se muestra la primera ventana, ya que las restantes son muy similares. De igual manera, si se elige *Cambiar Plan de Estudios*, se solicitará el número de cuenta y si es que el alumno tiene ya una plan de estudios o propuesta académica dada de alta, ésta inmediatamente aparecerá como se muestra en la figura B.15. Aquí es posible cambiar alguna(s) materia(s), así como también modificar el semestre en el que se pretenda(n) cursar dicha(s) materia(s).

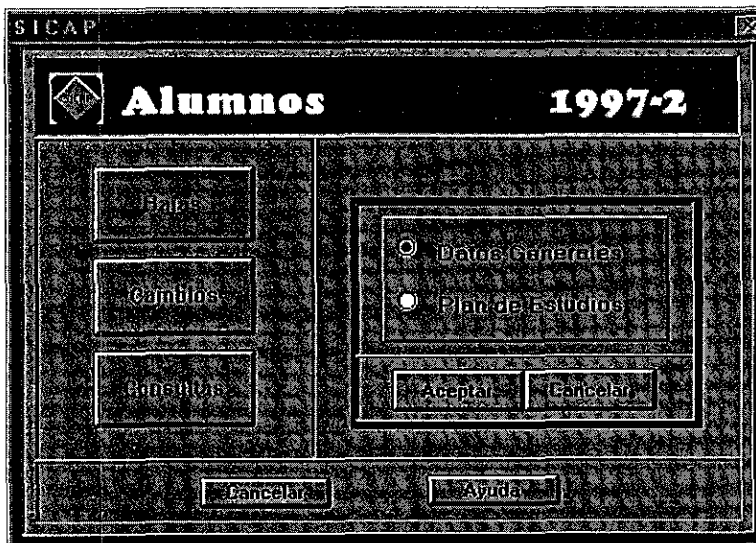


Figura B.13

 Masculino, Femenino; Nacionalidad: Mexicana, Extranjera; Fecha de Nacimiento: 26/03/66; Edo. Civil: SOLTERO. At the bottom, there are buttons for 'Cancelar', 'Cambiar', and 'Continuar'."/>

Figura B.14

1.3 Consultar Alumnos.

Esta opción nos permite consultar *Datos Generales, Plan de Estudios, Inscripción Actual* y el *kárdex*. (Figura B.16)

Datos Generales.- Se solicitará el número de cuenta del alumno para el cual se desee consultar sus datos personales. (Pantalla similar a la de la figura B.14)

Clave	Materia	Créditos	Semestre
613441	GASTOS Y PRESUPUESTOS	6	1
601500	CONFIABILIDAD	6	1
613197	COMPUTACION GRAFICA EN PLANEAC	6	1
680201	ALGEBRA LINEAL	6	1
604100	ECONOMETRIA	6	2

Figura B.15

Figura B 16

Plan de Estudios.- Se pide el número de cuenta del alumno e inmediatamente se despliega la pantalla de la figura B.17. Se tiene también la opción de mandar a impresión esta consulta

SICAP

Consultar Plan de Estudios

ALUMNO: **SANCHEZ COMPOY JORGE LUIS** Número de Cuenta: **94805587**

Clave	Materia	Créditos	Semestre	Campo
613441	COSTOS Y PRSUPUESTOS	6	1	Campo Menor
601500	CONFIABILIDAD	6	1	Campo Básico
613197	COMPUTACION GRAFICA EN PLANEACII	6	1	Campo Menor
680201	ALGEBRA LINEAL	6	1	Investigación y Tesis
604100	ECONOMETRIA	6	2	Campo Mayor
613403	FINANZAS CORPORATIVAS	6	3	Campo Básico
617000	PROBABILIDAD Y ESTADISTICA	6	4	Investigación y Tesis

Registro 1 de 7

Figura B.17

Inscripción Actual.- Se pide el número de cuenta del alumno y se despliega entonces su l comprobante de inscripción correspondiente al semestre actual. En caso de que el alumno no esté cursando ninguna materia en este semestre, el sistema indica lo anterior con un mensaje de advertencia. Esta consulta también se puede enviar a la impresora.

SICAP

División de Estudios de Posgrado de la Facultad de Ingeniería
Comprobante de Inscripción

Sección: **INVESTIGACION DE OPERACIONES** Semestre de Inscripción: **1997-2**

Número de Cuenta: **94005477** Nombre del Alumno: **DE LA TORRE VEGA ELI**

Clave	Asignatura	Grupo	Cred.	Profesor
680201	ALGEBRA LINEAL	0001	6	EDUARDO ARRIOLA
613197	COMPUTACION GRAFICA EN PLANE.	1111	6	JOSE LUIS ABURTO AVILA
601500	CONFIABILIDAD	6666	6	SERGIO AMBRIZ MAGUEY
Total de Materias: 3				

Imprimir Cerrar

Figura B 17

Kárdex.- Se pide el número de cuenta del alumno y se despliega entonces el kárdex, actualizado al semestre actual. (Figura B.18)

SICAP

División de Estudios de Posgrado de la Facultad de Ingeniería
kárdex

Sección: **INVESTIGACION DE OPERACIONES** Semestre de Inscripción: **1997-2**

Número de Cuenta: **94005477** Nombre del Alumno: **DE LA TORRE VEGA ELI**

Actualizado al: **1997-2**

ASIGNATURAS:

ACTIV. LEY	<input type="text" value="3"/>	Campo Básico	<input type="text" value="1"/>
NO. ASIGNATURAS	<input type="text" value="1"/>	Campo Mayor	<input type="text" value="1"/>
TOTAL	<input type="text" value="4"/>	Campo Menor (Obligatorias)	<input type="text" value="1"/>
PROMEDIO	<input type="text" value="7.85"/>	Actividad de Investigación (Desarrollo de Tesis)	<input type="text" value="1"/>

Clave	Materia	Grupo	Cred.	Calif.	Corresponde a	Periodo	Acta
680201	ALGEBRA LINEAL	0001	6	6	Investigación y Tesis	1997-2	1000
613197	COMPUTACION GRAFICA EN PLANEA	1111	6	5.5	Campo Menor	1997-2	1002
601500	CONFIABILIDAD	6666	6	10	Campo Básico	1997-2	1003
604100	ECONOMETRIA	1010	6	9.9	Campo Mayor	1997-2	1004

Cerrar

Registro 1 de 4

Figura B 18

MÓDULO 2: PROCESO SEMESTRAL.

En este módulo se llevan a cabo todas las actividades que se presentan durante un periodo de clases (Figura B 19). Estas actividades son:

- » Materias
- » Profesores
- » Horarios
- » Inscripción
- » Calificaciones

Y a su vez, se tienen los siguientes submódulos:

SUBMÓDULO MATERIAS.- Para este submódulo, se solicita un password o clave de acceso en el caso de Altas, Bajas y Cambios de materias. Esta ventana es similar a la que se presenta a la entrada del sistema. (Figura B.4). Después de introducir la clave correcta, se despliega la ventana que aparece en la figura B.20.

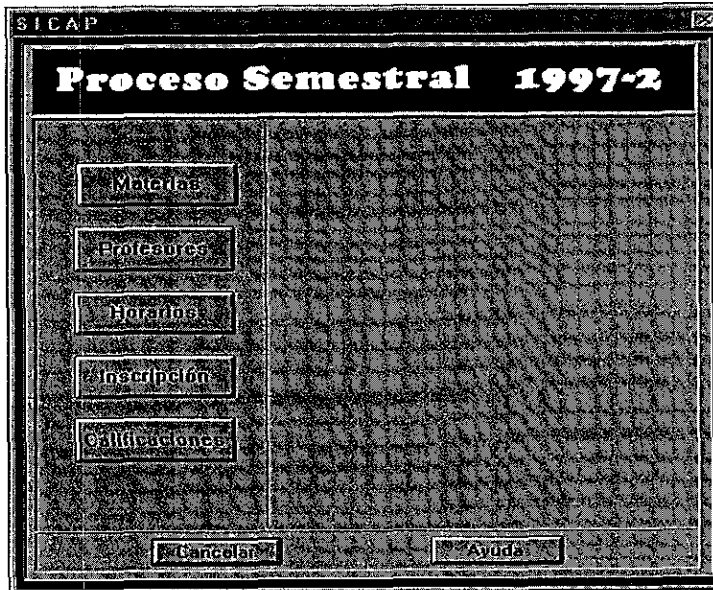


Figura B.19

Altas.- Se solicita la clave de la materia, que debe ser de 6 dígitos, el nombre de la materia, los créditos y el campo al que pertenece esta materia. Para dar de alta la materia hacer clic en el botón *Dar de Alta*. Si se quiere consultar el catálogo de materias, hacer clic en el botón *Consultar Claves*. (Figura B.21).

Bajas.- Se despliega una lista de materias para seleccionar la que se desea dar de baja. Para proceder con la baja, hacer clic en el botón *Dar de baja*

Cambios.- Al igual que en el caso de Bajas, se despliega una lista con las materias y se elige la que se quiere modificar. Para realizar el cambio, hacer clic en el botón *Hacer Cambio*.

Consultas.- Se presentan 2 tipos de consultas: *Catálogo General* y *Por clave*. El *Catálogo General* muestra todas las materias que se han dado de alta sin importar el periodo. La consulta *Por Clave*, solicita que se indique la clave de una materia específica y si ésta existe, se despliega toda su información.

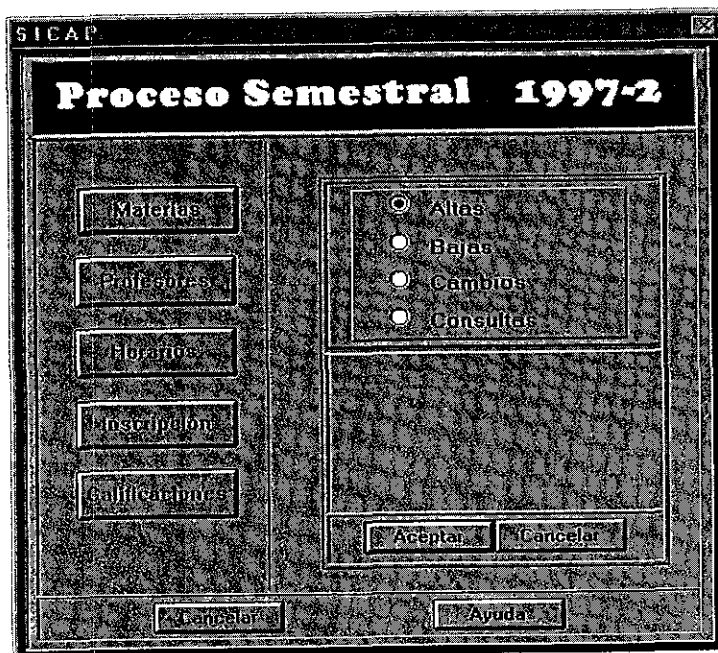


Figura B.20

The image shows a graphical user interface window for a system named 'SICAP'. The window title is 'SICAP' and the main heading is 'Alta de Materias 1997-2'. The interface includes four input fields: 'Clave', 'Rango al que pertenece', 'Materia', and 'Creditos'. Below these fields are four buttons: 'Guardar Alta', 'Salir', 'Ayuda', and 'Consultar Claves'. At the bottom of the window, a status bar displays 'Registro: 1 de 1'.

Figura B.21

SUBMÓDULO PROFESORES.- Para este submódulo, se solicita también una clave de acceso en el caso de Altas, Bajas y Cambios de profesores. Esta ventana es similar a la que se presenta a la entrada del sistema. (Figura B.4).

Altas.- Se solicita el RFC del profesor, nombre completo y una breve descripción sobre el área de desarrollo del mismo (Figura B.22). Para dar de alta al profesor hacer clic en el botón *Dar de Alta*. Si se quiere consultar el catálogo de profesores, hacer clic en el botón *Consultar Profesores*

Bajas.- De manera similar que la opción Baja de Materias, se despliega una lista de profesores para seleccionar al que se desea dar de baja. Para proceder con la baja, hacer clic en el botón *Dar de baja*.

Cambios.- Al igual que en el caso de Bajas de profesor, se despliega una lista de los profesores y se elige al que se quiere modificar. Para realizar el cambio, hacer clic en el botón *Hacer Cambio*.

The image shows a graphical user interface window titled 'SICAP' and 'Profesores 1997-2'. The window contains several input fields: 'RFC', 'Apellido Paterno', 'Apellido Materno', 'Nombre(s)', and a larger 'Campo de Investigación'. Below the input fields are four buttons: 'Dar de Alta', 'Salir', 'Ayuda', and 'Consultar Profesores'. At the bottom of the window, there is a status bar showing 'Registe 1 de 1' and a cursor icon.

Figura B.22

Consultas.- Se presentan 4 tipos de consultas: *Catálogo General*, *Datos Personales*, *Materias que Imparten* y *Tutorías a Cargo*. El *Catálogo General* muestra a todos los profesores que se han dado de alta. (Figura B.23). Se puede realizar la búsqueda mediante filtros, por ejemplo, para consultar a todos los profesores cuyo apellido paterno inicie con la letra A, presionar entonces el botón con la etiqueta A. Al hacer clic en el botón *Tutorías*, se despliega una lista de los alumnos que se encuentran bajo la tutela del profesor seleccionado. El botón *Materias* presenta una lista de materias que el profesor imparte en el semestre actual. Mediante el botón *Imprimir Lista de Profesores*, se obtiene un reporte impreso de los mismos. Al hacer clic en el botón *Consultar Profesores*, se presenta una lista sencilla de todos los profesores que se encuentran dados de alta.

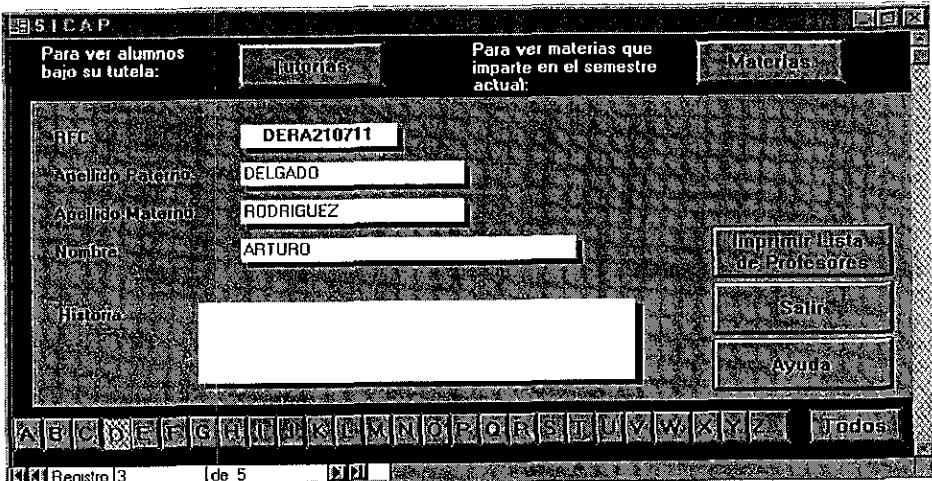


Figura B.23

Con la opción *Datos Personales*, se elige de la lista el nombre del profesor a consultar y a continuación se despliega el registro completo. Para saber las materias que un determinado profesor está impartiendo en el semestre actual, seleccionamos la opción *Materias que Imparte* y se obtiene la pantalla que se muestra en la figura B.24. El botón *Imprimir* proporciona una lista de las materias que imparte el profesor seleccionado.

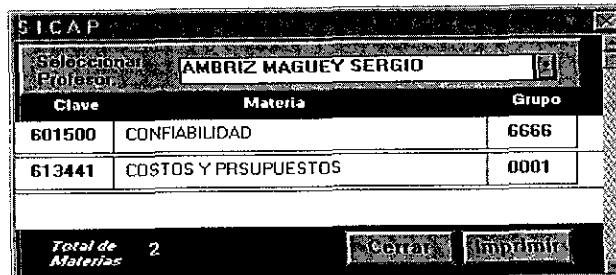


Figura B.24

El botón *Tutorías a Cargo* presenta una ventana con los alumnos que están bajo la tutela del profesor seleccionado. Se puede mandar a imprimir la lista de tutelos de un profesor determinado haciendo clic en el botón *Imprimir Selección Actual*, o de todos los profesores con el botón *Imprimir Lista General*.

SICAP

Tutorías a cargo 1997-2

Seleccionar Profesor: ACEVES GARCIA RICARDO

Número de cuenta	Nombre del Alumno	Sección	Semestre de Ingreso
94807392	ALVAREZ BRAVO JULIO FERNANDO	PLANEACION	1997-1
94805958	GARCIA MORALES NANTYELI	INVESTIGACION DE OPERACIONES	1997-1
94806151	MARCE VAZQUEZ MARIA DEL CARMEN	INVESTIGACION DE OPERACIONES	1997-1
94806704	MELO JIMENEZ AGUSTIN	TRANSPORTE	1997-1
94806285	REYES GRAJAJES JUAN GABRIEL	TRANSPORTE	1997-1
94805642	SPINOSO FRANCO LUIS GERARDO	TRANSPORTE	1997-1

Figura B 25

SUBMÓDULO HORARIOS.- Para este submódulo, se solicita una clave de acceso en el caso de Altas, Bajas y Cambios de horarios. La ventana es similar a la que se presenta a la entrada del sistema. (Figura B.4). Para los Horarios al igual que para Materias y Profesores, se tienen las opciones de Altas, Bajas, Cambios y Consultas.

Altas.- Después de introducir la clave correcta, se despliega la pantalla de la figura B.26, donde habrá que formar la "clase" seleccionando un profesor, una materia, indicando un grupo, así como los días y las horas. Si no se indica alguno de estos campos el registro no se dará de alta.

SICAP

Horarios 1997-2

Profesor:

Materia:

Indicar Día(s) y Hora (de inicio y fin) de la Clase:

Observaciones:

Figura B 26

Para consultar las clases que hayan sido dadas de alta, hacer clic en el botón *Consultar Clases*.

Bajas.- Para dar de baja una clase, es necesario indicar seleccionar el nombre de la materia y el grupo, con lo que a continuación aparecerá toda la información de la clase. Para proceder con la baja, bastará con dar un clic en el botón *Dar de Baja*. (Figura B.27).

SICAP

Baja de Clases 1997-2

Seleccionar Materia: 680201 0001

Asignatura: ALGEBRA LINEAL Grupo: 0001

Profesor: ARIOLA EDUARDO

Horario

DIA	HORA
Lunes	08:00-12:00
Martes	14:00-18:00
Miércoles	08:00-12:00
Jueves	
Viernes	08:00-12:00
Sábado	14:00-18:00

Observaciones:

Salón: PRIMER PISO

Dar de baja Salir Cancelar Ayuda

Registro 11 de 1

Figura B.27

Cambios.- Esta opción permite modificar algún campo de una clase, para esto habrá que seleccionar el nombre de la materia y el grupo. Para realizar el cambio es necesario hacer clic en el botón *Hacer el Cambio*. (La ventana es similar a la de la figura B.27).

Consultas.- Se presentan 3 opciones de consulta: *Histórica*, *Clase por Materia* y *Clase por Grupo*. La consulta *Histórica* proporciona todos los horarios (clases) que se impartieron en un semestre determinado, esta consulta puede ser enviada a impresión. (Figura B.28). *Clase por Materia* solicita que se teclee la clave de una materia específica y si existen clases con dicha materia, se despliega toda la información de la misma. (Figura B.29). *Clase por grupo* proporciona un concentrado de los grupos abiertos para una materia en particular y también esta consulta puede enviarse a impresora (Figura B.30.)

SICAP

Consulta Histórica de Horarios

1997-2

Clave:	Materia:	Grupo:	Profesor
680201	ALGEBRA LINEAL	0003	ACEVES GARCIA RICARDO
680201	ALGEBRA LINEAL	0002	ABURTO AVILA JOSE LUIS
680201	ALGEBRA LINEAL	0001	ARRIOLA EDUARDO
613197	COMPUTACION GRAFICA EN PLANE	1111	ABURTO AVILA JOSE LUIS
601500	CONFIABILIDAD	6666	AMBRIZ MAGUEY SERGIO
613441	COSTOS Y PRSUPUESTOS	0001	AMBRIZ MAGUEY SERGIO

Grupos Abiertos durante el semestre seleccionado: 8

Otro:Semestre Imprimir Cerrar

Figura B 28

SICAP

Materia: 680201 ALGEBRA LINEAL Grupo: 0001
 Profesor: ARRIOLA EDUARDO Créditos: 6
 Salón: 205 Ubicación: PRIMER PISO

Horario Observaciones:

DIA	HORA
Lunes	11:00-12:00
Martes	
Miércoles	11:00-12:00
Jueves	
Viernes	10:00-11:00
Sábado	

Grupos Abiertos para esta Materia: 3

Registro anterior Registro Siguiente Salir

Registro 3 de 3

Figura B.29

Grupos por Materia	
680201 ALGEBRA LINEAL	
Grupo	Profesor
0002	ABURTO AVILA JOSE LUIS
0003	ACEVES GARCIA RICARDO
0001	ARRIOLA EDUARDO
Grupos Abiertos para esta Materia: 3	
<input type="button" value="Imprimir"/> <input type="button" value="Salir"/>	

Figura B 30

SUBMÓDULO INSCRIPCIÓN.- Para este submódulo, se presentan dos casos: *Primer Ingreso* y *Reingreso*. *Primer Ingreso* se refiere a los alumnos que por primera vez cursarán una maestría del área de Sistemas de la DEPTI, por lo que antes de proceder con la inscripción al semestre actual es necesario capturar sus *Datos Personales*, *Plan de Estudios* y posteriormente inscribirlo. Cuando se capturan los datos personales de un alumno de primer ingreso será necesario especificar si el alumno proviene de la UNAM o proviene de alguna otra dependencia, ya que en este último caso el sistema le generará un número de cuenta provisional. Si el alumno proviene de la UNAM, entonces se utilizará su número de cuenta. La primera pantalla de captura de los *Datos Personales* se muestra en la figura B.31, el resto de las pantallas de captura son similares. Para capturar la propuesta académica (Plan de Estudios) se utiliza la ventana de la figura B.32. Con el botón *Consultar Plan* podemos ver en pantalla la Propuesta Académica del alumno.

DATOS GENERALES DEL ALUMNO	
N. Cuenta: 91542818	N. Expediente:
Apellido P:	Apellido M: Nombre(s):
Dirección:	Sexo: <input type="radio"/> Masculino <input type="radio"/> Femenino
Colonia:	
Estado:	C.P.:
Tel. Dom.:	Nacionalidad: <input checked="" type="checkbox"/> Mexicana <input type="checkbox"/> Extranjera
Tel. Oficina:	
Edo. Civil:	Fecha de Nacimiento: (dd/mm/aa)
<input type="button" value="Cancelar Alta"/>	<input type="button" value="Continuar"/>

Figura B.31

SICAP

Alta del Plan de Estudios

N.-Cuenta: **91542818** N.-Expediente: _____

Alumno: **BONIFAZ GOMEZ MARTHA EUGENIA**

Primer Semestre

613400	INGENIERIA FINANCIERA	6	Investigación y Tesis
615000	ESTADISTICA	6	Campo Básico

Figura B.31

Una vez que ya se han capturado Datos Personales y Propuesta Académica, se procede entonces a inscribir al alumno a materias del semestre actual. Para esto se solicita el número de cuenta del alumno y se despliega a continuación una pantalla donde habrá que seleccionar la(s) materia(s) a la(s) cual(es) el alumno se inscribirá. La pantalla se muestra en la figura B.32

SICAP


Inscripciones 1997-2

91542818 MARTHA EUGENIA BONIFAZ GOMEZ

604100 Seleccionar Grupo: 1010

Clave: 604100 ECONOMETRIA Grupo: 1010

Profesor: ARTURO DELGADO RODRIGUEZ

Horario:  Observaciones:

DIA	HORA
Lunes	
Martes	
Miércoles	
Jueves	
Viernes	
Sábado	

Salón: TERCER PISO

Figura B.32.

Para realizar la inscripción a la materia y grupo seleccionado, hacer clic en el botón *Aceptar* con lo que se aparecerá una ventana secundaria para confirmar la inscripción. Para consultar si el alumno ya está inscrito en alguna materia del semestre actual, hacer clic en el botón *Ver Inscripción*, para cancelar la inscripción a la materia actual, presionar el botón *Cancelar*.

SUBMÓDULO CALIFICACIONES.- Para este submódulo, se presentan 4 opciones: *Asignar Calificaciones, Cancelar, Cambiar y Consultar.*

Asignar Calificaciones.- Al asignar calificaciones el sistema generará un número de folio que identificará de manera única esta "acta", a continuación aparecerá la ventana que se muestra en la figura B.33. Aquí se muestran todos los alumnos que están inscritos en la materia y grupo seleccionados.

Número de cuenta	Nombre del Alumno	Calificación
91542818	BONIFAZ GOMEZ MARTHA EUGENIA	0 00
94805477	DE LA TORRE VEGA ELI	0 00

Alumnos Inscritos en esta materia: 2

Otro Folio Salir Ayuda

Registro: 1 de 2

Figura B.33

Cancelar.- Esta opción se utiliza para cancelar un número de folio en particular.

Cambiar.- Permite modificar el número del folio o cambiar alguna calificación.

Consultar.- Aquí se presentan 2 opciones: *Consulta General* y *Consulta específica*. En la *Consulta General* se muestran todos los folios que se han generado hasta ese momento, indicando a qué materia y grupo corresponden, esta lista se puede mandar directamente a impresora. (Figura B.34). La *Consulta Específica* solicita un número de folio y muestra toda la información referente al mismo, también se tiene la opción de mandar a imprimir la lista mediante el botón *Imprimir*. (Figura B.35).

S.I.C.A.P.

Folios de Calificaciones generados para el semestre actual 1997-2

N. de Folio	N. de Cuenta	Expediente	Alumnos	Clave	Materia	Grupo	Crédit
1000	94805477	11941016	DE LA TORRE VEGA ELI	60020	ALGEBRA LINEAL	0001	6
1002	94805477	11941016	DE LA TORRE VEGA ELI	613197	COMPUTACION GRAFICA EN PLANEACI	1111	6
1003	94805477	11941016	DE LA TORRE VEGA ELI	601500	CONFIABILIDAD	6666	6
1005	91542818		BONIFAZ GOMEZ MARTHA EUGENIA	604100	ECONOMETRIA	1010	6
1005	94805477	11941016	DE LA TORRE VEGA ELI	604100	ECONOMETRIA	1010	6

Imprimir Cerrar

Registro.1 de 5

Figura B.34

S.I.C.A.P.

Consultar Lista de Calificaciones 1997-2

N. de Folio

Clave	Materia	Grupo	Creditos	Profeso
604100	ECONOMETRIA	1010	6	ARTURO DELGADO RODRIGUEZ

N. de Cuenta	Expediente	Alumnos	Calificación
91542818		BONIFAZ GOMEZ MARTHA EUGENIA	9 00
94805477	11941016	DE LA TORRE VEGA ELI	8 50

N. de Alumnos:

Imprimir Cerrar

Registro.1 de 2

Figura B 35

MÓDULO 3. REPORTES

Para ejecutar este módulo bastará con dar un clic con el mouse sobre el botón con la leyenda REPORTES o bien, sobre el icono. Al hacer esto se despliega la pantalla de la figura B 36 Todos los reportes que se obtienen de este módulo se presentan al final de este manual de usuario.

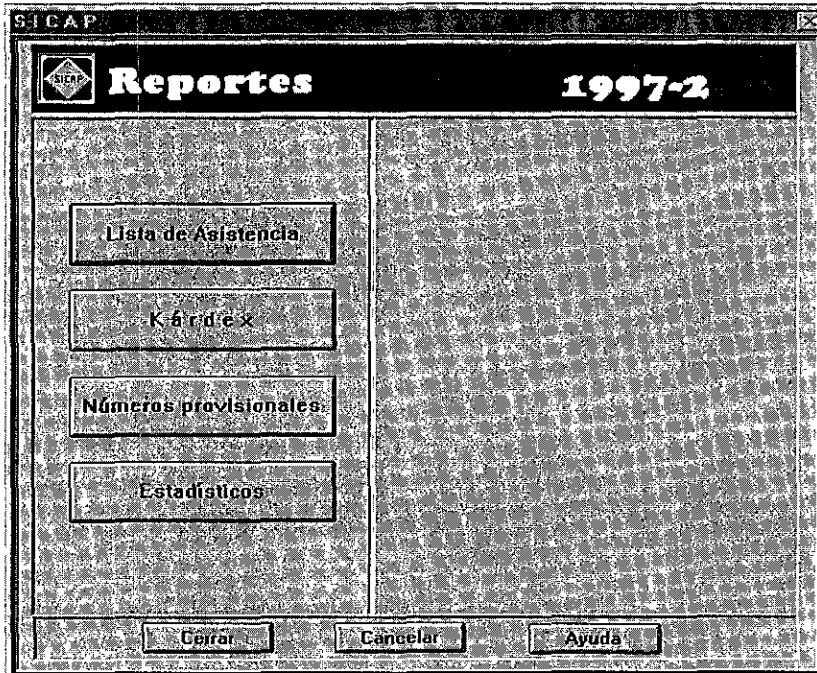


Figura B 36

3.1 Lista de Asistencia.- Proporciona la lista de asistencia para cada profesor, de la materia y grupos seleccionados. El reporte se muestra en pantalla y a la vez se puede mandar directamente a impresora mediante el botón *Imprimir* o bien ver una presentación preliminar con el botón *Vista Preliminar*.

3.2 Kárdex.- El kárdex (Historial Académica) se puede generar para un alumno en particular o para todos. En ambos casos se tiene la opción de enviarlo directamente a impresora o ver una presentación preliminar.

3.3 Números Provisionales.- Despliega una lista de aquellos alumnos que por no provenir de la UNAM el sistema les generó un número de cuenta provisional. La lista se puede mandar a imprimir o ver una presentación preliminar.

3.4 Estadísticos.- Esta opción nos permite obtener información de carácter estadístico sobre los alumnos inscritos en el Departamento de Sistemas. (Figura B 37).

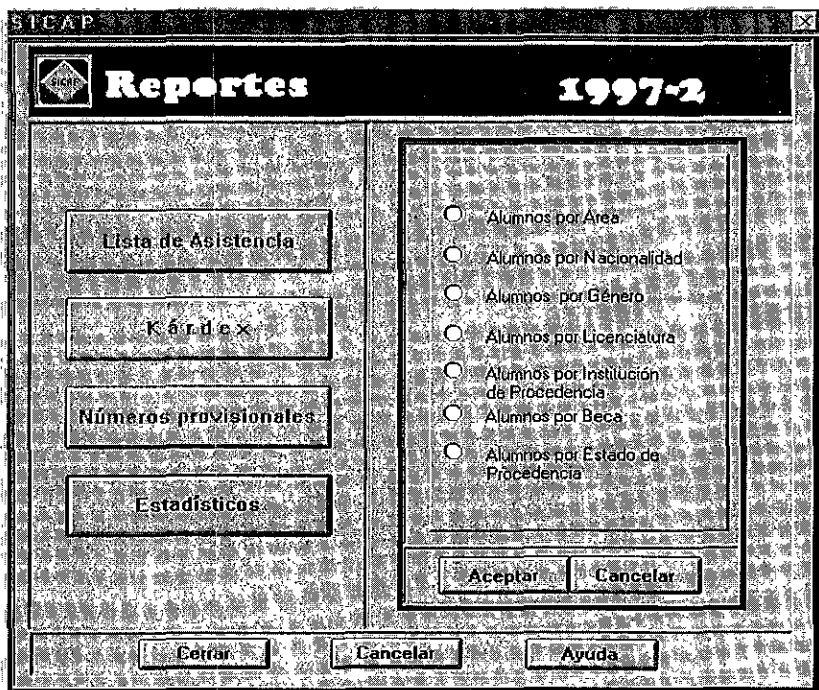


Figura B.37

En todos los casos se tiene la opción de enviar el reporte (General o Especifico) a un documento de Excel 5.0 o de Word 6, como se muestra en la figura B.38

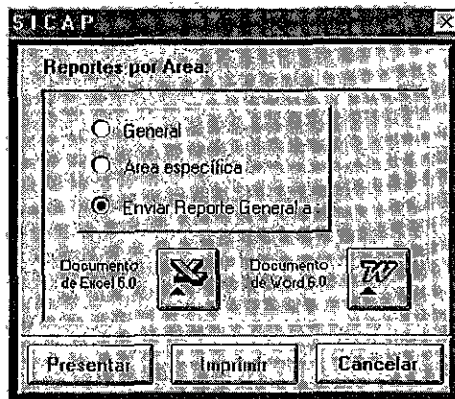


Figura B 38

MÓDULO 4: CATÁLOGOS

Para ejecutar este módulo bastará con dar un clic con el mouse sobre el botón con la leyenda CATÁLOGOS o bien, sobre el icono. Al hacer esto se despliega la pantalla de la figura B.39. Para cada catálogo se tiene las opciones de *Altas*, *Bajas*, *Cambios* y *Consultas* a excepción del catálogo de passwords donde solamente se disponen de las opciones de *Cambios* y *Consultas*. El funcionamiento de este módulo se ejemplificará solamente con el catálogo de Becas, ya que los demás trabajan de manera similar.

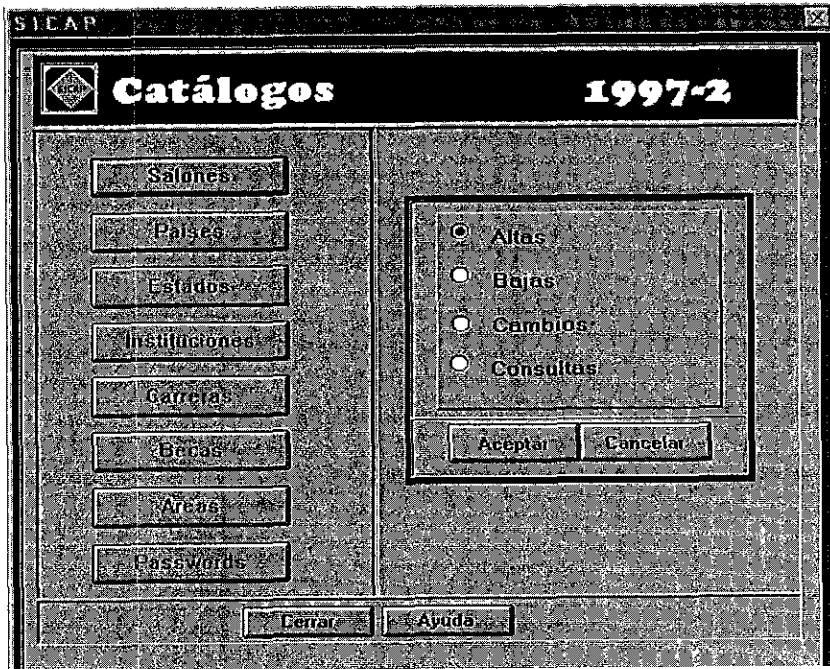


Figura B.39

Altas.- Permite dar de alta un nuevo tipo de beca, para lo cual se utiliza una pantalla como la que se muestra en la figura B.40.

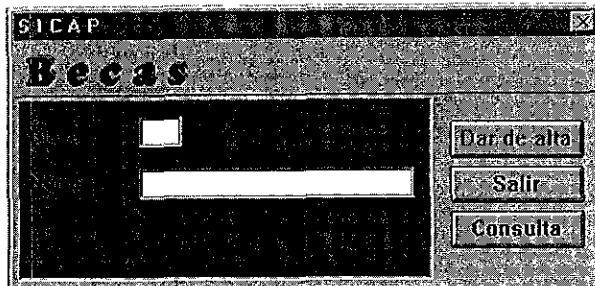


Figura B.40

El botón *Consulta* proporciona una lista de todas las becas que están dadas de alta.

Bajas.- Esta opción permite seleccionar un tipo de beca para eliminarla. La pantalla se ilustra en la figura B.41.

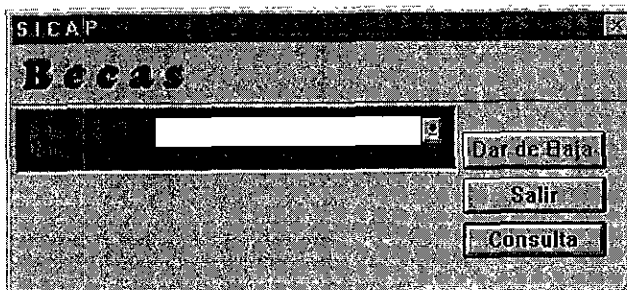


Figura B.41

Cambios.- Esta opción permite seleccionar un tipo de beca para modificarla. La pantalla es similar a la de la figura B.41.

Consultas.- Esta opción permite ver el catálogo general de becas. (Figura B.42.)

Clave	Clave
01	SIN BECA
02	INSTITUCION GUBERNAMENTA
03	EXENTO DE PAGO UNAM
04	UNAM
05	FUNDACION O ASOCIACION
06	BECADO PERO NO ESPECIFICA
07	CONACYT
08	GOBIERNO EXTRANJERO

Figura B.42

MÓDULO 4: AYUDA

La ayuda general del sistema se puede acceder al hacer clic sobre el botón con la etiqueta AYUDA o bien con el icono. (Figura B.43).



Figura B.43

MÓDULO 4: SALIR

Este módulo permite abandonar el sistema. (Figura B.44).



Figura B.44

» *Utilerías.*

El sistema cuenta con un programa adicional que son las utilerías, este programa se accede al hacer doble clic sobre el icono correspondiente. (Figura B.45)



Figura B.45

A continuación se presenta una pantalla que solicita una clave de acceso, esta clave únicamente deberá conocerla la persona encargada del manejo del sistema, ya que un mal uso de este programa puede ocasionar daños al sistema. Al proporcionar la clave correcta se despliega entonces el Menú Principal. (Figura B.46)

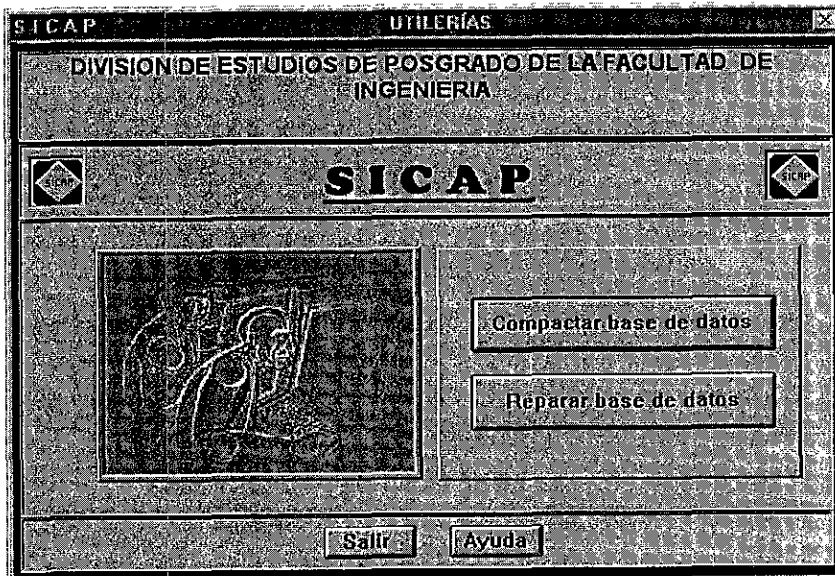


Figura B.46

Las opciones que se presentan son:

Compactar Base de Datos.

Permite crea una copia de la base de datos, reorganizando la manera en que el archivo de base de datos se almacena en el disco, si es que está fragmentado. El almacenamiento de archivos podría fragmentarse cuando se hacen muchos cambios en una base de datos. El archivo de base de datos compactado suele ser más pequeño y normalmente ofrece mejor rendimiento. Al compactar la Base de Datos se solicitará al usuario que indique la ruta donde se encuentra. (Figura B.47).

Reparar Base de Datos .

La base de datos puede resultar dañada si Microsoft Access se detiene inesperadamente: por ejemplo, si se produce un corte del fluido eléctrico o si el equipo tiene un problema de hardware. La base de datos no se dañará si sale de la aplicación de la manera normal. Al reparar la Base de Datos se solicitará al usuario que indique la ruta donde se encuentra. (Figura B.48).

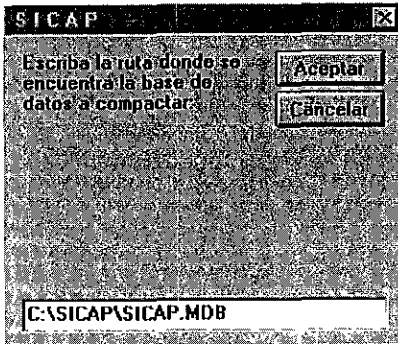


Figura B.47

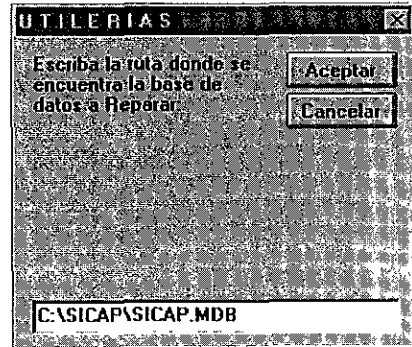


Figura B.48

Universidad Nacional Autónoma de México
División de Estudios de Posgrado de la Facultad de Ingeniería

Datos Personales

12/01/98

N.- de Cuenta:	N.- de Expediente:	Nombre:	Ingreso: 1997-1
94805587	11941032	SANCHEZ COMPOY JORGE LUIS	

Plan:	TRANSPORTE
--------------	------------

DOMICILIO:	Carrera de Procedencia:
Calle: PERIFERICO SUR 4091 F-2-7	INGENIERO EN CIENCIAS DE LA COMPUTACION
Colonia: U.H.PORTES GIL	
C.P. : 14130	UNIVERSIDAD LA SALLE
Estado: DISTRITO FEDERAL	
Teléfono Domicilio: 645 89 25	
Teléfono Oficina: 3791594	Fecha de nacimiento: 29/10/61

Tutor: AMBRIZ MAGUEY SERGIO

Universidad Nacional Autónoma de México
División de Estudios de Posgrado de la Facultad de Ingeniería

Plan de Estudios

12/01/98

N.- de Cuenta:	N.- de Expediente:	Nombre:	Ingreso:
94805587	11941032	SANCHEZ COMPOY JORGE LUIS	1997-1

Plan: TRANSPORTE

Clave	Materia	Créditos	Semestre	Campo
1) 680201	ALGEBRA LINEAL	6	1	Investigación y Tesis
2) 613197	COMPUTACION GRAFICA EN PLANEAC	6	1	Campo Menor
3) 601500	CONFIABILIDAD	6	1	Campo Básico
4) 613441	COSTOS Y PRSUPUESTOS	6	1	Campo Menor
5) 604100	ECONOMETRIA	6	2	Campo Mayor
6) 613403	FINANZAS CORPORATIVAS	6	3	Campo Básico
7) 617000	PROBABILIDAD Y ESTADISTICA	6	4	Investigación y Tesis

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
DIVISIÓN DE ESTUDIOS DE POSGRADO DE LA FACULTAD DE INGENIERÍA

Lista de Profesores

12/01/98

R.F.C.	Nombre del Profesor
1 AACF390123	ALVAREZ CASO FRANCISCO
2 AEGR510923	ACEVES GARCIA RICARDO
3 AIMS631007	AMBRIZ MAGUEY SERGIO
4 AOFJ450714	ACOSTA FLORES JOSE JESUS
5 AUAL440919	ABURTO AVILA JOSE LUIS
6 CUSA680312PRV	CUEVAS SALGADO ALEJANDRO
7 FUZA511008	FUENTES ZENON ARTURO
8 SURJ570816	SUAREZ ROCHA JAVIER
9 TEAM450407	TREJOS T MAYRA
10 TESR450713	TELLEZ SANCHEZ RUBEN
11 CAGX331120	CAMACHO GALVAN ABEL
12 COCP471201	CORTEZ PAPI CLAUDE
13 CUAF640829	CRUZ ARANDA FERNANDO
14 DIHE640413	D'ALBA IBARRA HECTOR
15 DERA210711	DELGADO RODRIGUEZ ARTURO
16 DORF350525	DOVALI RAMOS FEDERICO
17 GACF551031	GARFIAS CAMPOS FRANCISCO
18 GUAM530802	GUTIERREZ ANDRADE MIGUEL A.
19 SAGG530805	SANCHEZ GUERRERO GABRIEL
20 FUMS441211	FUENTES MAYA SERGIO
21 GUBS380327	GUILLEN BURGUETE SERGIO
22 TESJ421204	TELLO SANDOVAL JAVIER
23 FOMI590624	FLORES DE LA MATA IDALIA

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
DIVISIÓN DE ESTUDIOS DE POSGRADO DE LA FACULTAD DE INGENIERÍA
Lista de Profesores

12/01/98

R.F.C.	Nombre del Profesor
24 COSS531025PRV	DE COBOS SERGIO
25 TOCL630212RSA	TORRES C. JONATAN
26 DOOM690518	DOMINGUEZ MARIANA
27 AIVE450225	ARRIOLA EDUARDO
28 GOVA-530810	GOMEZ VEGA JAIME F.

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
DIVISIÓN DE ESTUDIOS DE POSGRADO DE LA FACULTAD DE INGENIERÍA
Lista de Asistencia

6/03/98

Periodo **1997-2**

<i>Clave</i>	<i>Asignatura</i>	<i>Créditos</i>	<i>Grupo</i>	<i>Profesor</i>
604100	ECONOMETRIA	6	1010	DELGADO RODRIGUEZ ARTURO

	Número de Cuenta	Número de Expediente	Programa	Nombre del Alumno
1	91542818		000468	BONIFAZ GOMEZ MARTHA EUGENIA
2	94805477	11941016	000468	DE LA TORRE VEGA ELI

Total de Alumnos. 2

Número de Cuenta	Nombre Alumno	Programa	Año de Ingreso
94805477	DE LA TORRE VEGA ELI	000468 INVESTIGACION DE OPERACIONES	1997-1

	Clave	Asignatura	Grupo	Créditos
1	680201	ALGEBRA LINEAL	0001	6
2	613197	COMPUTACION GRAFICA EN PLANEACI	1111	6
3	601500	CONFIABILIDAD	6666	6
4	604100	ECONOMETRIA	1010	6

RECIBI DE CONFORMIDAD

Firma del Alumno

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
DIVISIÓN DE ESTUDIOS DE POSGRADO DE LA FACULTAD DE INGENIERÍA

K á r d e x

20/02/98

N. de Cuenta:	Expediente:	Alumno:	
94805477	11941016	DE LA TORRE VEGA ELI	
Plan:		Ingreso:	Actualizado al:
INVESTIGACION DE OPERACIONES		1997-1	1997-2
Aprobadas 3			Campo Básico: 1
No Aprobadas 1			Campo Mayor: 1
TOTAL: 4			Campo Menor: 1
PROMEDIO ██████	CRÉDITOS: 24		Investigación y Tesis: 1

Clave	Materia	Grupo	Créditos	Calif.	Período	Folio
601500	CONFIABILIDAD	8866	6	10	1997-2	1003
604100	ECONOMETRIA	1010	6	8.5	1997-2	1005
613197	COMPUTACION GRAFICA EN PLANEA	1111	6	5.5	1997-2	1002
680201	ALGEBRA LINEAL	0001	6	6	1997-2	1000

Universidad Nacional Autónoma de México
División de Estudios de Posgrado de la Facultad de Ingeniería

Calificaciones

Semestre: 1997-2

20/02/98

N.- de Folio	Clave	Materia	Grupo	Créditos	Profesor
	604100	ECONOMETRIA	1010	6	ARTURO DELGADO RODRIGUEZ

	N- de Cuenta	Expediente	Alumnos	Calificación
1)	91542818		BONIFAZ GOMEZ MARTHA EUGENIA	9
2)	94805477	11941016	DE LA TORRE VEGA ELI	8.5

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
DIVISIÓN DE ESTUDIOS DE POSGRADO DE LA FACULTAD DE INGENIERÍA

Lista de Profesores

12/01/98

R.F.C.	Nombre del Profesor
24 COSS531025PRV	DE COBOS SERGIO
25 TOCL630212RSA	TORRES C. JONATAN
26 DOOM690518	DOMINGUEZ MARIANA
27 AIVE450225	ARRIOLA EDUARDO
28 GOVA-530810	GOMEZ VEGA JAIME F.

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
DIVISIÓN DE ESTUDIOS DE POSGRADO DE LA FACULTAD DE INGENIERÍA

6/03/98

Lista de Asistencia

Periodo 1997-2

<i>Clave</i>	<i>Asignatura</i>	<i>Créditos</i>	<i>Grupo</i>	<i>Profesor</i>
604100	ECONOMETRIA	6	1010	DELGADO RODRIGUEZ ARTURO

	Número de Cuenta	Número de Expediente	Programa	Nombre del Alumno
1	91542818		000468	BONIFAZ GOMEZ MARTHA EUGENIA
2	94805477	11941016	000468	DE LA TORRE VEGA ELI

Total de Alumnos: 2

Número de Cuenta	Nombre Alumno	Programa	Año de Ingreso
94805477	DE LA TORRE VEGA ELI	000468 INVESTIGACION DE OPERACIONES	1997-1

	Clave	Asignatura	Grupo	Créditos
1	680201	ALGEBRA LINEAL	0001	6
2	613197	COMPUTACION GRAFICA EN PLANEACI	1111	6
3	601500	CONFIABILIDAD	6666	6
4	604100	ECONOMETRIA	1010	6

RECIBI DE CONFORMIDAD

Firma del Alumno

Alumnos bajo la tutela de:

AMBRIZ MAGUEY SERGIO

	Número de Cuenta	Número de Expediente	Programa	Nombre del Alumno
1	95803672	11951558	ELECTRICA	ALVAREZ ESTRADA ERNESTO ESTEB
2	94805477	11941016	INVESTIGACION DE OPERACIONES	DE LA TORRE VEGA ELI
3	95803452	11951510	INVESTIGACION DE OPERACIONES	JIMENEZ NAVA MONICA
4	95803627	11951552	INVESTIGACION DE OPERACIONES	MIRAMONTES ZAPATA ELOISA
5	95803737	11951564	INVESTIGACION DE OPERACIONES	RIVERA ULLOA NESSI JEZREEL
6	94805587	11941032	TRANSPORTE	SANCHEZ COMPOY JORGE LUIS
7	95803706	11951561	INVESTIGACION DE OPERACIONES	VENEROS SANCHEZ FRANCISCO

Relación de Alumnos por Estado de Procedencia

Número de Cuenta	Número de Expediente	Nombre del Alumno	Periodo de Ingreso	Estado	
1	94807392	11941359	ALVAREZ BRAVO JULIO FERNANDO	1997-1	DISTRITO FEDERAL
2	95803445	11951509	BUENDIA SERVIN DE LA MORA NEIL SPAR	1997-2	DISTRITO FEDERAL
3	95803476	11951513	CORREA GOMEZ ARMANDO	1997-2	DISTRITO FEDERAL
4	94805477	11941016	DE LA TORRE VEGA ELI	1997-1	DISTRITO FEDERAL
5	94805958	11941095	GARCIA MORALES NANTVELI	1997-1	DISTRITO FEDERAL
6	94806151	11941126	MARCE VAZQUEZ MARIA DEL CARMEN	1997-1	DISTRITO FEDERAL
7	95803414	11951495	NIETO ARISTA HECTOR	1997-2	DISTRITO FEDERAL
8	94806285	11941167	REYES GRAJAJES JUAN GABRIEL	1997-1	DISTRITO FEDERAL
9	95803500	11951524	ROMANO HERAS IRMA ALEJANDRA	1997-2	DISTRITO FEDERAL
10	94805587	11941032	SANCHEZ COMPOY JORGE LUIS	1997-1	DISTRITO FEDERAL
11	94807292	11941347	SOLIS BRITO CESAR EMILIO	1997-1	DISTRITO FEDERAL
12	94805642	11941038	SPINOSO FRANCO LUIS GERARDO	1997-1	DISTRITO FEDERAL
13	95804349	11951088	CORONA ARMENTA JOSE RAMON	1997-2	HIDALGO
14	95803452	11951510	JIMENEZ NAVA MONICA	1997-2	HIDALGO
15	94806704	11941249	MELO JIMENEZ AGUSTIN	1997-1	HIDALGO
16	95804356	11951089	GONZALEZ RODRIGUEZ ANTONIO	1997-2	JALISCO
17	95803672	11951558	ALVAREZ ESTRADA ERNESTO ESTEBAN	1997-2	MORELOS
18	95803627	11951552	MIRAMONTES ZAPATA ELOISA	1997-2	MORELOS
19	95803713	11951562	PALACIOS VALDEZ RODOLFO	1997-2	MORELOS
20	95803737	11951564	RIVERA ULLOA NESSI JEZREEL	1997-2	MORELOS
21	95803744	11951565	SEDANO FLORES PEDRO	1997-2	MORELOS
22	95803706	11951561	VENEROS SANCHEZ FRANCISCO	1997-2	MORELOS
23	99900000		PEREZ RAMIREZ JOSE	1997-2	SAN LUIS POTOSI
24	99900001		TORRES GIL SUSANA	1997-2	SAN LUIS POTOSI
25	95803641	11951555	CABANILLAS LOPEZ MIRIAM JUDITH	1997-2	SONORA

Relación de Alumnos por Institución donde se cursó Licenciatura.

Número de Cuenta	Número de Expediente	Nombre del Alumno	Periodo de Ingreso	Institución
1	95803500	11951524 ROMANO HERAS IRMA ALEJANDRA	1997-2	INSTITUTO POLITECNICO NACIONAL
2	95803445	11951509 BUENDIA SERVIN DE LA MORA NEIL SPAR	1997-2	INSTITUTO TECNOLOGICO REGIONAL DE CELAYA
3	95804356	11951089 GONZALEZ RODRIGUEZ ANTONIO	1997-2	INSTITUTO TECNOLOGICO REGIONAL DE CIUDAD GUZMAN
4	99900000	PEREZ RAMIREZ JOSE	1997-2	INSTITUTO TECNOLOGICO REGIONAL DE CIUDAD GUZMAN
5	99900001	TORRES GIL SUSANA	1997-2	INSTITUTO TECNOLOGICO REGIONAL DE CIUDAD GUZMAN
6	95804349	11951088 CORONA ARMENTA JOSE RAMON	1997-2	INSTITUTO TECNOLOGICO REGIONAL DE PACHUCA
7	95803452	11951510 JIMENEZ NAVA MONICA	1997-2	INSTITUTO TECNOLOGICO REGIONAL DE PACHUCA
8	95803414	11951495 NIETO ARISTA HECTOR	1997-2	INSTITUTO TECNOLOGICO Y DE ESTUDIOS DE MONTERREY
9	95803744	11951565 SEDANO FLORES PEDRO	1997-2	INSTITUTO TECNOLOGICO Y DE ESTUDIOS DE MONTERREY
10	94805958	11941095 GARCIA MORALES NANTYELJ	1997-1	OTROS
11	95803737	11951564 RIVERA ULLOA NESSI JEZREEL	1997-2	OTROS
12	94806151	11941126 MARCE VAZQUEZ MARIA DEL CARMEN	1997-1	UNIVERSIDAD ANAHUAC
13	95803672	11951558 ALVAREZ ESTRADA ERNESTO ESTEBAN	1997-2	UNIVERSIDAD AUTONOMA DE MORELOS
14	95803706	11951561 VENEROS SANCHEZ FRANCISCO	1997-2	UNIVERSIDAD AUTONOMA DE MORELOS
15	95803627	11951552 MIRAMONTES ZAPATA ELOISA	1997-2	UNIVERSIDAD AUTONOMA DE ZACATECAS
16	95803713	11951562 PALACIOS VALDEZ RODOLFO	1997-2	UNIVERSIDAD DE NUEVO LEON
17	95803641	11951555 CABANILLAS LOPEZ MIRIAM JUDITH	1997-2	UNIVERSIDAD DE SONORA
18	94807392	11941359 ALVAREZ BRAVO JULIO FERNANDO	1997-1	UNIVERSIDAD LA SALLE

Número de Cuenta	Número de Expediente	Nombre del Alumno	Periodo de Ingreso	Institución
19 95803476	11951513	CORREA GOMEZ ARMANDO	1997-2	UNIVERSIDAD LA SALLE
20 94805587	11941032	SANCHEZ COMPOY JORGE LUIS	1997-1	UNIVERSIDAD LA SALLE
21 94807292	11941347	SOLIS BRITO CESAR EMILIO	1997-1	UNIVERSIDAD LA SALLE
22 94805477	11941016	DE LA TORRE VEGA ELI	1997-1	UNIVERSIDAD MICHOACANA DE SAN NICOLAS HIDALGO
23 94806704	11941249	MELO JIMENEZ AGUSTIN	1997-1	UNIVERSIDAD MICHOACANA DE SAN NICOLAS HIDALGO
24 94806285	11941167	REYES GRAJALES JUAN GABRIEL	1997-1	UNIVERSIDAD VERACRUZANA
25 94805642	11941038	SPINOSO FRANCO LUIS GERARDO	1997-1	UNIVERSIDAD VERACRUZANA

Relación de Alumnos por Beca

Número de Cuenta	Número de Expediente	Nombre del Alumno	Periodo de Ingreso	Tipo de Beca	
1	95803452	11951510	JIMENEZ NAVA MONICA	1997-2	BECADO PERO NO ESPECIFICA
2	95804349	11951088	CORONA ARMENTA JOSE RAMON	1997-2	CONACYT
3	95804356	11951089	GONZALEZ RODRIGUEZ ANTONIO	1997-2	CONACYT
4	94806704	11941249	MELO JIMENEZ AGUSTIN	1997-1	EXENTO DE PAGO UNAM
5	95803672	11951558	ALVAREZ ESTRADA ERNESTO ESTEBAN	1997-2	FUNDACION O ASOCIACION
6	95803445	11951509	BUENDIA SERVIN DE LA MORA NEIL SPAR	1997-2	FUNDACION O ASOCIACION
7	95803476	11951513	CORREA GOMEZ ARMANDO	1997-2	FUNDACION O ASOCIACION
8	95803713	11951562	PALACIOS VALDEZ RODOLFO	1997-2	FUNDACION O ASOCIACION
9	95803737	11951564	RIVERA ULLOA NESSI JEZREEL	1997-2	FUNDACION O ASOCIACION
10	95803500	11951524	ROMANO HIRAS IRMA ALEJANDRA	1997-2	FUNDACION O ASOCIACION
11	95803744	11951565	SEDANO FLORES PEDRO	1997-2	FUNDACION O ASOCIACION
12	94805642	11941038	SPINOSO FRANCO LUIS GERARDO	1997-1	FUNDACION O ASOCIACION
13	95803706	11951561	VENEROS SANCHEZ FRANCISCO	1997-2	FUNDACION O ASOCIACION
14	99900000		PEREZ RAMIREZ JOSE	1997-2	GOBIERNO EXTRANJERO
15	99900001		TORRES GIL SUSANA	1997-2	GOBIERNO EXTRANJERO
16	94807392	11941359	ALVAREZ BRAVO JULIO FERNANDO	1997-1	INSTITUCION GUBERNAMENTAL
17	94805477	11941016	DE LA TORRE VEGA ELI	1997-1	INSTITUCION GUBERNAMENTAL
18	94805958	11941095	GARCIA MORALES NANTYELI	1997-1	INSTITUCION GUBERNAMENTAL

Número de Cuenta	Número de Expediente	Nombre del Alumno	Periodo de Ingreso	Tipo de Beca
19 94806151	11941126	MARCE VAZQUEZ MARIA DEL CARMEN	1997-1	INSTITUCION GUBERNAMENTAL
20 95803414	11951495	NIETO ARISTA HECTOR	1997-2	INSTITUCION GUBERNAMENTAL
21 94806285	11941167	REYES GRAJAJES JUAN GABRIEL	1997-1	INSTITUCION GUBERNAMENTAL
22 94805587	11941032	SANCHEZ COMPOY JORGE LUIS	1997-1	INSTITUCION GUBERNAMENTAL
23 94807292	11941347	SOLIS BRITO CESAR EMILIO	1997-1	INSTITUCION GUBERNAMENTAL
24 95803641	11951555	CABANILLAS LOPEZ MIRIAM JUDITH	1997-2	UNAM
25 95803627	11951552	MIRAMONTES ZAPATA ELOISA	1997-2	UNAM

Relación de Alumnos por Nacionalidad

	Número de Cuenta	Número de Expediente	Nombre del Alumno	Período de Ingreso	Nacionalidad
1	94807392	11941359	ALVAREZ BRAVO JULIO FERNANDO	1997-1	M
2	95803672	11951558	ALVAREZ ESTRADA ERNESTO ESTEBAN	1997-2	M
3	95803641	11951555	CABANILLAS LOPEZ MIRIAM JUDITH	1997-2	M
4	95804349	11951088	CORONA ARMENTA JOSE RAMON	1997-2	M
5	95803476	11951513	CORREA GOMEZ ARMANDO	1997-2	M
6	94805477	11941016	DE LA TORRE VEGA ELI	1997-1	M
7	94805958	11941095	GARCIA MORALES NANTYELI	1997-1	M
8	95804356	11951089	GONZALEZ RODRIGUEZ ANTONIO	1997-2	M
9	95803452	11951510	JIMENEZ NAVA MONICA	1997-2	M
10	94806151	11941126	MARCE VAZQUEZ MARIA DEL CARMEN	1997-1	M
11	94806704	11941249	MELO JIMENEZ AGUSTIN	1997-1	M
12	95803627	11951552	MIRAMONTES ZAPATA ELOISA	1997-2	M
13	95803414	11951495	NIETO ARISTA HECTOR	1997-2	M
14	95803713	11951562	PALACIOS VALDEZ RODOLFO	1997-2	M
15	99900000		PEREZ RAMIREZ JOSE	1997-2	M
16	94806285	11941167	REYES GRAJAJES JUAN GABRIEL	1997-1	M
17	95803737	11951564	RIVERA ULLOA NESSI JEZREEL	1997-2	M
18	95803500	11951524	ROMANO HERAS IRMA ALEJANDRA	1997-2	M
19	94805587	11941032	SANCHEZ COMPOY JORGE LUIS	1997-1	M
20	95803744	11951565	SEDANO FLORES PEDRO	1997-2	M
21	94807292	11941347	SOLIS BRITO CESAR EMILIO	1997-1	M
22	94805642	11941038	SPINOSO FRANCO LUIS GERARDO	1997-1	M
23	99900001		TORRES GIL SUSANA	1997-2	M
24	95803706	11951561	VENEROS SANCHEZ FRANCISCO	1997-2	M
25	95803445	11951509	BUENDIA SERVIN DE LA MORA NEIL SPAR	1997-2	E

(M) Mexicana. (E) Extranjera

Relación de Alumnos por Género

	Número de Cuenta	Número de Expediente	Nombre del Alumno	Periodo de Ingreso	Sexo
1	94807392	11941359	ALVAREZ BRAVO JULIO FERNANDO	1997-1	M
2	95803672	11951558	ALVAREZ ESTRADA ERNESTO ESTEBAN	1997-2	M
3	95803445	11951509	BUENDIA SERVIN DE LA MORA NEIL SPAR	1997-2	M
4	95804349	11951088	CORONA ARMENTA JOSE RAMON	1997-2	M
5	95803476	11951513	CORREA GOMEZ ARMANDO	1997-2	M
6	95804356	11951089	GONZALEZ RODRIGUEZ ANTONIO	1997-2	M
7	94806704	11941249	MELO JIMENEZ AGUSTIN	1997-1	M
8	95803414	11951495	NIETO ARISTA HECTOR	1997-2	M
9	95803713	11951562	PALACIOS VALDEZ RODOLFO	1997-2	M
10	99900000		PEREZ RAMIREZ JOSE	1997-2	M
11	94806285	11941167	REYES GRAJAJES JUAN GABRIEL	1997-1	M
12	95803737	11951564	RIVERA ULLOA NESSI JEZREEL	1997-2	M
13	94805587	11941032	SANCHEZ COMPOY JORGE LUIS	1997-1	M
14	95803744	11951565	SEDANO FLORES PEDRO	1997-2	M
15	94807292	11941347	SOLIS BRITO CESAR EMILIO	1997-1	M
16	94805642	11941038	SPINOSO FRANCO LUIS GERARDO	1997-1	M
17	95803706	11951561	VENEROS SANCHEZ FRANCISCO	1997-2	M
18	95803641	11951555	CABANILLAS LOPEZ MIRIAM JU' DITH	1997-2	F
19	94805477	11941016	DE LA TORRE VEGA ELI	1997-1	F
20	94805958	11941095	GARCIA MORALES NANTYELI	1997-1	F
21	95803452	11951510	JIMENEZ NAVA MONICA	1997-2	F
22	94806151	11941126	MARCE VAZQUEZ MARIA DEL CARMEN	1997-1	F
23	95803627	11951552	MIRAMONTES ZAPATA ELOISA	1997-2	F
24	95803500	11951524	ROMANO HERAS IRMA ALEJANDRA	1997-2	F
25	99900001		TORRES GIL SUSANA	1997-2	F

(M) Masculino, (F) Femenino

Relación de Alumnos por licenciatura de procedencia

Número de Cuenta	Número de Expediente	Nombre del Alumno	Carrera	Promedio	Fecha de Titulación	Universidad
1	94805958	11941095 GARCIA MORALES NANTYELI	ACTUARIO	8.69	5/03/93	OTROS
2	94806151	11941126 MARCE VAZQUEZ MARIA DEL CARMEN	ACTUARIO	9.57	15/02/93	UNIVERSIDAD ANAHUAC
3	94805477	11941016 DE LA TORRE VEGA ELI	FISICO MATEMATICO	9.23	29/06/90	UNIVERSIDAD MICHOACANA DE SAN NICOLAS HIDALGO
4	95803641	11951555 CABANILLAS LOPEZ MIRIAM JUDITH	INGENIERO CIVIL	7.68	27/01/89	UNIVERSIDAD DE SONORA
5	95804349	11951088 CORONA ARMENTA JOSE RAMON	INGENIERO CIVIL	9.3	19/03/93	INSTITUTO TECNOLÓGICO REGIONAL DE PACHUCA
6	95803476	11951513 CORREA GOMEZ ARMANDO	INGENIERO CIVIL	9	27/10/93	UNIVERSIDAD LA SALLE
7	95803627	11951552 MIRAMONTES ZAPATA ELOISA	INGENIERO CIVIL	7.93	22/10/93	UNIVERSIDAD AUTONOMA DE ZACATECAS
8	94806285	11941167 REYES GRAJALES JUAN GABRIEL	INGENIERO CIVIL	7.82	7/05/93	UNIVERSIDAD VERACRUZANA

Número de Cuenta	Número de Expediente	Nombre del Alumno	Carrera	Promedio	Fecha de Titulación	Universidad
9	94805642	11941038 SPINOSO FRANCO LUIS GERARDO	INGENIERO CIVIL	7,64	30/07/93	UNIVERSIDAD VERACRUZANA
10	94805587	11941032 SANCHEZ COMPOY JORGE LUIS	INGENIERO EN CIENCIAS DE LA COMPUTACION	9,15	18/02/91	UNIVERSIDAD LA SALLE
11	95803672	11951558 ALVAREZ ESTRADA ERNESTO ESTEBAN	INGENIERO INDUSTRIAL	0	26/04/93	UNIVERSIDAD AUTONOMA DE MORELOS
12	95803445	11951509 BUENDIA SERVIN DE LA MORA NEIL SPAR	INGENIERO INDUSTRIAL	8,4	25/01/94	INSTITUTO TECNOLOGICO REGIONAL DE CELAYA
13	95803452	11951510 JIMENEZ NAVA MONICA	INGENIERO INDUSTRIAL	8,7	19/09/94	INSTITUTO TECNOLOGICO REGIONAL DE PACHUCA
14	94806704	11941249 MELO JIMENEZ AGUSTIN	INGENIERO INDUSTRIAL	7,94	26/06/93	UNIVERSIDAD MICHOACANA DE SAN NICOLAS HIDALGO
15	95803744	11951565 SEDANO FLORES PEDRO	INGENIERO INDUSTRIAL	8	13/12/94	INSTITUTO TECNOLOGICO Y DE ESTUDIOS DE MONTERREY
16	95803706	11951561 VENEROS SANCHEZ FRANCISCO	INGENIERO INDUSTRIAL	8,2	5/06/93	UNIVERSIDAD AUTONOMA DE MORELOS
17	95803414	11951495 NIETO ARISTA HECTOR	INGENIERO MECANICO	8,2	13/12/93	INSTITUTO TECNOLOGICO Y DE ESTUDIOS DE MONTERREY

Número de Cuenta	Número de Expediente	Nombre del Alumno	Carrera	Promedio	Fecha de Titulación	Universidad
18	94807392	11941359 ALVAREZ BRAVO JULIO FERNANDO	LICENCIADO EN ADMINISTRACION DE EMPRESAS	9.1	6/11/91	UNIVERSIDAD LA SALLE
19	95804356	11951089 GONZALEZ RODRIGUEZ ANTONIO	LICENCIADO EN ADMINISTRACION DE EMPRESAS	9.29	21/04/89	INSTITUTO TECNOLOGICO REGIONAL DE CIUDAD GUZMAN
20	99900000	Perez RAMIREZ JOSE	LICENCIADO EN ADMINISTRACION DE EMPRESAS	9.10	21/04/89	INSTITUTO TECNOLOGICO REGIONAL DE CIUDAD GUZMAN
21	99900001	TORRES GIL SUSANA	LICENCIADO EN ADMINISTRACION DE EMPRESAS	6.41	21/04/89	INSTITUTO TECNOLOGICO REGIONAL DE CIUDAD GUZMAN
22	95803500	11951524 ROMANO HERAS IRMA ALEJANDRA	LICENCIADO EN ECONOMIA	8.9	23/03/94	INSTITUTO POLITECNICO NACIONAL
23	95803713	11951562 PALACIOS VALDEZ RODOLFO	OTROS	8.7	3/07/89	UNIVERSIDAD DE NUEVO LEON
24	95803737	11951564 RIVERA UJILLO NESSI JEZREEL	OTROS	8.2	10/12/93	OTROS
25	94807292	11941347 SOLIS BRITO CESAR EMILIO	OTROS	0	29/09/92	UNIVERSIDAD LA SALLE

Relación de Alumnos con número de cuenta provisional

Número de Cuenta	Número de Expediente	Nombre del Alumno	Periodo de Ingreso	Plan	Institución de Procedencia
1	99900000	PEREZ RAMIREZ JOSE	1997-2	MATEMATICAS	INSTITUTO TECNOLOGICO REGIONAL DE CIUDAD GUZMAN
2	99900001	TORRES GIL SUSANA	1997-2	MATEMATICAS	INSTITUTO TECNOLOGICO REGIONAL DE CIUDAD GUZMAN

Relación de Alumnos por Area

Número de Cuenta	Número de Expediente	Nombre del Alumno	Periodo de Ingreso	Area
1	91542818	BONIFAZ GOMEZ MARTHA EUGENIA	1997-2	INVESTIGACION DE OPERACIONES
2	95803641	11951555 CABANILLAS LOPEZ MIRIAM JUDITH	1997-2	INVESTIGACION DE OPERACIONES
3	94805477	11941016 DE LA TORRE VEGA ELI	1997-1	INVESTIGACION DE OPERACIONES
4	94805958	11941095 GARCIA MORALES NANTYELI	1997-1	INVESTIGACION DE OPERACIONES
5	95804356	11951089 GONZALEZ RODRIGUEZ ANTONIO	1997-2	INVESTIGACION DE OPERACIONES
6	95803452	11951510 JIMENEZ NAVA MONICA	1997-2	INVESTIGACION DE OPERACIONES
7	94806151	11941126 MARCE VAZQUEZ MARIA DEL CARMEN	1997-1	INVESTIGACION DE OPERACIONES
8	95803627	11951552 MIRAMONTES ZAPATA ELOISA	1997-2	INVESTIGACION DE OPERACIONES
9	95803713	11951562 PALACIOS VALDEZ RODOLFO	1997-2	INVESTIGACION DE OPERACIONES
10	95803737	11951564 RIVERA ULLOA NESSI JEZREFI.	1997-2	INVESTIGACION DE OPERACIONES
11	95803706	11951561 VENEROS SANCHEZ FRANCISCO	1997-2	INVESTIGACION DE OPERACIONES
12	94807392	11941359 ALVAREZ BRAVO JULIO FERNANDO	1997-1	PLANEACION
13	95803672	11951558 ALVAREZ ESTRADA ERNESTO ESTEBAN	1997-2	PLANEACION
14	95803445	11951509 BUENDIA SERVIN DE LA MORA NEIL SPAR	1997-2	PLANEACION
15	95804349	11951088 CORONA ARMENTA JOSE RAMON	1997-2	PLANEACION
16	95803476	11951513 CORREA GOMEZ ARMANDO	1997-2	PLANEACION
17	95803744	11951565 SEDANO FLORES PEDRO	1997-2	PLANEACION
18	94807292	11941347 SOLIS BRITO CESAR EMILIO	1997-1	PLANEACION

	Número de Cuenta	Número de Expediente	Nombre del Alumno	Periodo de Ingreso	Area
19	94806704	11941249	MELO JIMENEZ AGUSTIN	1997-1	TRANSPORTE
20	95803414	11951495	NIETO ARISTA HECTOR	1997-2	TRANSPORTE
21	99900000		PEREZ RAMIREZ JOSE	1997-2	TRANSPORTE
22	94806285	11941167	REYES GRAJAJES JUAN GABRIEL	1997-1	TRANSPORTE
23	95803500	11951524	ROMANO HERAS IRMA ALEJANDRA	1997-2	TRANSPORTE
24	94805587	11941032	SANCHEZ COMPOY JORGE LUIS	1997-1	TRANSPORTE
25	94805642	11941038	SPINOSO FRANCO LUIS GERARDO	1997-1	TRANSPORTE
26	99900001		TORRES GIL SUSANA	1997-2	TRANSPORTE

A

p

é

n

d

i

c

e

C

Catálogos

CATALOGOS

BECAS

Clave de la beca	Nombre de la beca
1	SIN BECA
2	INSTITUCIÓN GUBERNAMENTAL
3	EXENTO DE PAGO UNAM
4	UNAM
5	FUNDACIÓN O ASOCIACIÓN
6	BECADO PERO NO ESPECIFICADO
7	CONACYT
8	GOBIERNO EXTRANJERO

INSTITUCIONES

Clave de la institución	Nombre de la institución
1001	CENTRO DE ENSEÑANZA TÉCNICA Y SUPERIOR DE BAJA CALIFORNIA
1002	ESCUELA DE INGENIERÍA CIVIL DE CHIAPAS
1003	ESCUELA DE INGENIERÍA MUNICIPAL
1004	ESCUELA NACIONAL DE AGRICULTURA CHAPINGO
1005	HEROICA ESCUELA NAVAL MILITAR
1006	INSTITUTO POLITÉCNICO NACIONAL
1007	INSTITUTO TECNOLÓGICO REGIONAL DE AGUASCALIENTES
1008	INSTITUTO TECNOLÓGICO REGIONAL DE CELAYA
1009	INSTITUTO TECNOLÓGICO REGIONAL DE CIUDAD GUZMAN
1010	INSTITUTO TECNOLÓGICO REGIONAL DE CIUDAD JUÁREZ
1011	INSTITUTO TECNOLÓGICO REGIONAL DE CIUDAD MADERO
1012	INSTITUTO TECNOLÓGICO REGIONAL DE CULIACÁN
1013	INSTITUTO REGIONAL DE CHIHUAHUA
1014	INSTITUTO TECNOLÓGICO REGIONAL DE DURANGO
1015	INSTITUTO TECNOLÓGICO REGIONAL DEL ISTMO
1016	INSTITUTO TECNOLÓGICO REGIONAL DE LA LAGUNA
1017	INSTITUTO TECNOLÓGICO REGIONAL DE LEÓN
1018	INSTITUTO TECNOLÓGICO REGIONAL DE MATAMOROS
1019	INSTITUTO TECNOLÓGICO REGIONAL DE MERIDA
1020	INSTITUTO TECNOLÓGICO REGIONAL DE MINATITLAN (BENITO JUÁREZ)
1021	INSTITUTO TECNOLÓGICO REGIONAL DE MORELIA
1022	INSTITUTO TECNOLÓGICO REGIONAL DE NUEVO LAREDO
1023	INSTITUTO TECNOLÓGICO REGIONAL DE OAXACA
1024	INSTITUTO TECNOLÓGICO REGIONAL DE ORIZABA
1025	INSTITUTO TECNOLÓGICO REGIONAL DE PACHUCA
1026	INSTITUTO TECNOLÓGICO REGIONAL DE PUEBLA
1027	INSTITUTO TECNOLÓGICO REGIONAL DE QUERÉTARO
1028	INSTITUTO TECNOLÓGICO REGIONAL DE SALTILLO
1029	INSTITUTO TECNOLÓGICO REGIONAL DE SONORA
1030	INSTITUTO TECNOLÓGICO REGIONAL DE TIJUANA
1031	INSTITUTO TECNOLÓGICO REGIONAL DE TLALNEPANTLA
1032	INSTITUTO TECNOLÓGICO REGIONAL DE TUXTLA GUTIÉRREZ
1033	INSTITUTO TECNOLÓGICO REGIONAL DE VERACRUZ
1034	INSTITUTO TECNOLÓGICO Y DE ESTUDIOS DE MONTERREY
1035	INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE
1036	INSTITUTO TECNOLÓGICO REGIONAL DE ZACATEPEC
1037	UNIVERSIDAD ANAHUAC

1038	UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
1039	UNIVERSIDAD AUTÓNOMA DE CIUDAD JUAREZ
1040	UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA
1041	UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MEXICO
1042	UNIVERSIDAD AUTÓNOMA DE GUADALAJARA
1043	UNIVERSIDAD AUTÓNOMA DE GUERRERO
1044	UNIVERSIDAD AUTÓNOMA DE HIDALGO
1045	UNIVERSIDAD AUTÓNOMA DE MORELOS
1046	UNIVERSIDAD AUTÓNOMA DE PUEBLA
1047	UNIVERSIDAD AUTÓNOMA DE QUERETARO
1048	UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSI
1049	UNIVERSIDAD AUTÓNOMA DE SINALOA
1050	UNIVERSIDAD AUTÓNOMA DE TAMAULIPAS
1051	UNIVERSIDAD AUTÓNOMA DE ZACATECAS
1052	UNIVERSIDAD DE COAHUILA
1053	UNIVERSIDAD DE GUADALAJARA
1054	UNIVERSIDAD DE GUANAJUATO
1055	UNIVERSIDAD DE NUEVO LEON
1056	UNIVERSIDAD DE SONORA
1057	UNIVERSIDAD DE YUCATAN
1058	UNIVERSIDAD IBEROAMERICANA
1059	UNIVERSIDAD LA SALLE
1060	UNIVERSIDAD MICHOACANA DE SAN NICOLAS HIDALGO
1061	UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO
1062	UNIVERSIDAD REGIONMONTANA, A. C.
1063	UNIVERSIDAD VERACRUZANA
1064	UNIVERSIDAD JUAREZ AUTÓNOMA TABASCO
1888	OTROS
3001	UNIVERSIDADES DE AFRICA
3002	UNIVERSIDADES DE ARGENTINA
3003	UNIVERSIDADES DE ASIA
3004	UNIVERSIDADES DE AUSTRALIA
3005	UNIVERSIDADES DE BOLIVIA
3006	UNIVERSIDADES DE BRASIL
3007	UNIVERSIDADES DE CANADA
3008	UNIVERSIDADES DE CHILE
3009	UNIVERSIDADES DE COLOMBIA
3010	UNIVERSIDADES DE COSTA RICA
3011	UNIVERSIDADES DE CUBA
3012	UNIVERSIDADES DE ECUADOR
3013	UNIVERSIDADES DEL SALVADOR
3014	UNIVERSIDADES DE ESTADOS UNIDOS
3015	UNIVERSIDADES DE EUROPA
3016	UNIVERSIDADES DE GUATEMALA
3017	UNIVERSIDADES DE HAITI
3018	UNIVERSIDADES DE HONDURAS
3019	UNIVERSIDADES DE NICARAQUA
3020	UNIVERSIDADES DE PANAMA
3021	UNIVERSIDADES DE PARAGUAY
3022	UNIVERSIDADES DE PERU
3023	UNIVERSIDADES DE PUERTO RICO
3024	UNIVERSIDADES DE LA REPUBLICA DOMINICANA
3025	UNIVERSIDADES DE URUGUAY
3026	UNIVERSIDADES DE VENEZUELA
3888	OTROS

ÁREAS

Clave de Dirección	Nombre de Dirección
000468	INVESTIGACIÓN DE OPERACIONES
000013	TRANSPORTE
000467	PLANEACIÓN

ESTADOS

Clave del Estado	Nombre del Estado
01	AGUASCALIENTES
02	BAJA CALIFORNIA NORTE
03	BAJA CALIFORNIA SUR
04	CAMPECHE
05	COAHUILA
06	COLIMA
07	CHIAPAS
08	CHIHUAHUA
09	DISTRITO FEDERAL
10	DURANGO
11	GUANAJUATO
12	GUERRERO
13	HIDALGO
14	JALISCO
15	MEXICO
16	MICHOACAN
17	MORELOS
18	NAYARIT
19	NUEVO LEON
20	OAXACA
21	PUEBLA
22	QUERETARO
23	QUINTANA ROO
24	SAN LUIS POTOSI
25	SINALOA
26	SONORA
27	TABASCO
28	TAMAULIPAS
29	TLAXCALA
30	VERACRUZ
31	YUCATAN
32	ZACATECAS

PÁISES

Número de País	Nombre del País
01	AFRICA
02	ARGENTINA
03	ASIA
04	AUSTRALIA
05	BOLIVIA
06	BRASIL
07	CANADA
08	CHILE
09	COLOMBIA
10	COSTA RICA
11	CUBA
12	ECUADOR
13	SALVADOR
14	ESTADOS UNIDOS
15	EUROPA
16	GUATEMALA
17	HAITI
18	HONDURAS
19	MEXICO
20	NICARAGUA
21	PANAMA
22	PARAGUAY
23	PERU
24	PUERTO RICO
25	REPUBLICA DOMINICANA
26	URUGUAY
27	VENEZUELA
88	OTROS

SALONES

Número de	Ubicación
205	PRIMER PISO
206	PRIMER PISO
305	SEGUNDO
306	SEGUNDO
405	TERCER PISO
406	TERCER PISO

TÍTULOS

Clave de título	Nombre de título
01	ACTUARIO
02	ARQUITECTO
03	BIOLOGO
04	FÍSICO
05	INGENIERO ARQUITECTO
06	INGENIERO CIVIL
07	INGENIERO MECANICO
08	INGENIERO MECANICO ELECTRICISTA
09	INGENIERO ELECTRICO
10	INGENIERO PETROLERO
11	INGENIERO EN AERONAUTICA
12	INGENIERO QUIMICO
13	INGENIERO QUIMICO INDUSTRIAL
14	INGENIERO QUIMICO METALURGICO
15	INGENIERO DE MINAS Y METALURGISTA
16	INGENIERO MECANICO NAVAL
17	LICENCIADO EN ADMINISTRACION DE EMPRESAS
18	LICENCIADO EN ECONOMIA
19	FISICO MATEMATICO
20	MATEMATICO
21	LICENCIADO EN INFORMATICA
22	INGENIERO GEOLOGO
23	INGENIERO EN CIENCIAS DE LA COMPUTACION
24	INGENIERO FISICO
25	INGENIERO TEXTIL
26	QUIMICO
27	QUIMICO FARMACOBIOLOGO
28	CONTADOR PUBLICO
29	INGENIERO MUNICIPAL
30	INGENIERO FORESTAL
31	INGENIRO TOPOGRAFO GEODESTA
32	INGENIERO INDUSTRIAL
33	INGENIERO DE COMUNICACIONES Y ELECTRONICA
88	OTROS

ESTADO CIVIL

Clave de estado civil	Estado civil
1	SOLTERO
2	CASADO
3	DIVORCIADO
4	OTRO

CÓDIGO DE ACCES

```

Sub CLAVE_BECA_AfterUpdate ()
    If Len(Lave_beca) < 2 Then
        m2 = "La clave es incorrecta"
        f2 = 48 + 0 + 0
        r2 = MsgBox(m2, f2, titulo)
        SendKeys "{esc}", True
        DoCmd CancelEvent
    End If
    Me!Bar de alta.Enabled = True
    Dim m1() As Database, m1tabla As Recordset
    Set m1() = dbengine.workspaces(0).databases(0)
    Set m1tabla = m1(0).OpenRecordset("UDCAS", DB_OPEN_TABLE)
    m1tabla.Index = "CLAVE" 'Establece el índice activo.
    m1tabla.Seek "=", &Me![CLAVE_BECA]
    If Not (m1tabla.NoMatch) Then
        m2 = "Existe una Beca con la misma clave"
        f2 = 48 + 0 + 0
        r2 = MsgBox(m2, f2, titulo)
        SendKeys "{esc}", True
    End If
    m1tabla.Close
End Sub

Sub Consulta_Click ()
    Dim frm1 As Database, m1tabla1 As Recordset
    Set frm1 = dbengine.workspaces(0).databases(0)
    Set m1tabla1 = frm1.OpenRecordset("UDCAS", DB_OPEN_TABLE)
    total = m1tabla1.RecordCount
    If total = 0 Then
        m3 = "No hay becas dadas de alta"
        f3 = 48 + 0 + 0
        r3 = MsgBox(m3, f3, titulo)
        m1tabla1.Close
    Else
        DoCmd OpenForm "Catálogo de becas"
    End If
End Sub

Sub Dar_de_Alta_Click ()
    p1 = "Dar de alta"
    d1 = 4 + 32 + 0
    r1 = MsgBox(p1, d1, titulo)
    If r1 = 4 Then
        If IsNull(Lave_beca) Then
            msg1 = "La clave es incorrecta"
            dgde1 = 48 + 0 + 0
            RL1 = MsgBox(msg1, dgde1, titulo)
            DoCmd GoToControl "CLAVE_BECA"
            Exit Sub
        End If
        If IsNull(NOMBRE_BECA) Then
            msg1 = "El nombre de la Beca es incorrecto"
            dgde1 = 48 + 0 + 0
            RL1 = MsgBox(msg1, dgde1, titulo)
            DoCmd GoToControl "nombre_BECA"
            Exit Sub
        End If
    End If
    Dim m1() As Database, m1tabla As Recordset
    Set m1() = dbengine.workspaces(0).databases(0)
    Set m1tabla = m1(0).OpenRecordset("UDCAS", DB_OPEN_TABLE)
    m1tabla.Index = "CLAVE" 'Establece el índice activo.
    m1tabla.Seek "=", Me![CLAVE_BECA]
    If (m1tabla.NoMatch) Then
        msg20 = "Dar de alta otra beca"
        dgde20 = 4 + 32 + 0
        r20 = MsgBox(msg20, dgde20, titulo)
        If r20 = 7 Then
            DoCmd Close
        Else
            DoCmd RunMacro "NTRADA5 otra beca"
            DoCmd GoToControl "Lave_beca"
        End If
    Else
        m2 = "Existe una Beca con la misma clave"
        f2 = 48 + 0 + 0
        r2 = MsgBox(m2, f2, titulo)
        SendKeys "{esc}", True
    End If
    m1tabla.Close
Else
    SendKeys "{esc}", True
End If
End Sub

Sub Form_BeforeUpdate (Cancel As Integer)
    If IsNull(Lave_beca) Then
        msg1 = "La clave es incorrecta"
        dgde1 = 48 + 0 + 0
        RL1 = MsgBox(msg1, dgde1, titulo)
        DoCmd GoToControl "CLAVE_BECA"
    End Sub
End If
If IsNull(NOMBRE_BECA) Then
    msg1 = "El nombre de la Beca es incorrecto"
    dgde1 = 48 + 0 + 0
    RL1 = MsgBox(msg1, dgde1, titulo)
    DoCmd GoToControl "nombre_BECA"
End Sub
End If

Sub Form_Error (DataErr As Integer, Response As Integer)
    Response = DATA_ERRCONTINUE
    Const CLAVERRUTICAADA = 8022
    Const NOMBREERRUTICA = 91
    If DataErr = CLAVERRUTICAADA Then
        m2 = "Existe una Beca con la misma clave"
        f2 = 48 + MB_ICONALTO + MB_DEFECTION2
        r2 = MsgBox(m2, f2, titulo)
    End If

```

```

End Sub
Sub Form_Open (Cancel As Integer)
DoCmd RunMacro "entrada.otra beca"
Me!{the de alta}.Enabled = False
End Sub
Sub Salir_Click ()
SendKeys "{esc}", True
DoCmd Close
End Sub
Sub aceptar_Click ()
m1 = " A l l a  C u e n t a  a  d a  "
If m1 < > 4
r1 = MsgBox(m1, 1, "título")
m2 = "¿Dar de Alta Plan de Estudios para otro alumno?"
f2 = # + 32 + MB_DIALOGBOX
r2 = MsgBox(m2, f2, "título")
If r2 = 6 Then
DoCmd Close
DoCmd Close A_FORM, "Alta del plan de estudios"
DoCmd Close A_FORM, "Lee cuenta para plan"
DoCmd Openform "Lee cuenta para plan"
DoCmd GoToControl "leoplan"
Else
DoCmd Close
DoCmd Close A_FORM, "Alta del plan de estudios"
DoCmd Close A_FORM, "Lee cuenta para plan"
End If
End Sub
Sub ayuda_Click ()
m2 = "¿Dar de Alta Plan de Estudios para otro alumno?"
f2 = # + 32 + MB_DIALOGBOX
r2 = MsgBox(m2, f2, "título")
If r2 = 6 Then
DoCmd Close
DoCmd Close A_FORM, "Alta del plan de estudios"
DoCmd Close A_FORM, "Lee cuenta para plan"
DoCmd Openform "Lee cuenta para plan"
DoCmd GoToControl "leoplan"
Else
DoCmd Close
DoCmd Close A_FORM, "Alta del plan de estudios"
DoCmd Close A_FORM, "Lee cuenta para plan"
End If
End Sub
Sub consultar_plan_Click ()
Dim tes As Database, mibliba As Recordset
Set tes = dbengine.workspaces(0).databases(0)
Set mibliba = tes.OpenRecordset("MAT_PLAN", DB_OPEN_TABLE)
mibliba.index = "n_cuenta"
mibliba.Seek "=", numerodecuenta
If mibliba.NoMatch Then
m2 = "El Alumno No tiene Plan de Estudios"
f2 = # + 0 + 0
r2 = MsgBox(m2, f2, "título")
Else
DoCmd Openform "Consulta de plan en pantalla de altas", , "In_cuenta={formularios}[Lee cuenta para plan][leoplan]"
End If
mibliba.Close
End Sub
Sub cuarto_Click ()
Me!semestre4.Visible = True
Me!primer0.Enabled = False
Me!segundo0.Enabled = False
Me!tercer0.Enabled = False
End Sub
Sub Form_Load ()
Me!semestre1.Visible = False
Me!semestre2.Visible = False
Me!semestre3.Visible = False
Me!semestre4.Visible = False
End Sub
Sub primer0_Click ()
Me!semestre1.Visible = True
Me!segundo0.Enabled = False
Me!tercer0.Enabled = False
Me!cuarto0.Enabled = False
End Sub
Sub quinto_Click ()
Me!semestre5.Visible = True
Me!primer0.Enabled = False
Me!segundo0.Enabled = False
Me!tercer0.Enabled = False
Me!cuarto0.Enabled = False
End Sub
Sub segundo_Click ()
Me!semestre2.Visible = False
Me!semestre3.Visible = True
Me!primer0.Enabled = False
Me!tercer0.Enabled = False
Me!cuarto0.Enabled = False
End Sub
Sub sexto_Click ()
Me!semestre6.Visible = True
Me!primer0.Enabled = False
Me!segundo0.Enabled = False
Me!tercer0.Enabled = False
Me!cuarto0.Enabled = False
End Sub
Sub tercer0_Click ()
Me!semestre3.Visible = True
Me!primer0.Enabled = False
Me!segundo0.Enabled = False
Me!cuarto0.Enabled = False
End Sub
Sub Consultar_profesor_Click ()
Dim test As Database, mibliba1 As Recordset
Set test = dbengine.workspaces(0).databases(0)
Set mibliba1 = test.OpenRecordset("PROFESOR", DB_OPEN_TABLE)

```

```

total = mitabla1.RecordCount
If total = 0 Then
    m3 = "No hay profesores dados de alta"
    f3 = 48 + 0 + 0
    r3 = MsgBox(m3, f3, titulo)
mitabla1.Close
Docmd Close A_form, "ASIST/CRI"
Exit Sub
Else
    Docmd OpenForm "Consulta de Profesores"
End If
End Sub
Sub Dar_de_Alta_Click ()
    p1 = "Datos Correctos?"
    d1 = 4 + 32 + 0
    r1 = MsgBox(p1, d1)
    If r1 = 0 Then
        msg1 = "¿Desea dar de alta al Profesor?"
        dgdef1 = 4 + 32 + 0
        RES1 = MsgBox(msg1, dgdef1, titulo)
        If RES1 = 0 Then
            If IsNull(paterno) Then
                msg1 = "El campo Apellido Paterno del Profesor Vacío!"
                dgdef1 = 48 + 0 + 0
                RES1 = MsgBox(msg1, dgdef1, titulo)
                Docmd GoToControl "paterno"
            End Sub
            If IsNull(NOMA_PROF) Then
                msg1 = "El campo Nombre del Profesor Vacío!"
                dgdef1 = 48 + 0 + 0
                RES1 = MsgBox(msg1, dgdef1, titulo)
                Docmd GoToControl "nombn."
            End Sub
        End If
    End If
    If IsNull(RFC) Then
        msg0 = "El RFC es incorrecto!"
        dgdef0 = 48 + 0 + 0
        RES0 = MsgBox(msg0, dgdef0, titulo)
        Docmd GoToControl "rfc"
    End Sub
End If
Dim mIBD As Database, mitabla As Recordset
Set mIBD = Dberregne.workspaces(0).databases(0)
Set mitabla = mIBD.OpenTable("CRUCLESOR", DB_OPEN_TABLE)
mitabla.Index = "INRFAC" 'Establece el índice activo
mitabla.Seek "=", Form!alta de profesor||RFC
If (mitabla.Recordset) Then 'si no está
    Docmd DoCmdOpenForm A_FORMBAR, A_PTE, A_SAVI RLC_CRI, A_MONI_VFRZO
    msg2 = "¿Dar de alta otro Profesor?"
    dgdef2 = 4 + 32 + 0
    res2 = MsgBox(msg2, dgdef2, titulo)
    If res2 = 0 Then
        Sendkeys "{esc}", True
        Docmd RunMacro "Intradas otro profesor"
        Docmd GoToControl "RFC"
    Else
        Docmd Close
    End If
Else
    'encuentro repetido
    m2 = "¿Existe un Profesor con el mismo RFC?"
    f2 = 48 + MB_ICONCAUTION + MB_YESNOCANCEL
    r2 = MsgBox(m2, f2, titulo)
    Sendkeys "{esc}", True
    Docmd RunMacro "Intradas otro profesor"
    msg2 = "¿Dar de alta otro Profesor?"
    dgdef2 = 4 + 32 + 0
    res2 = MsgBox(msg2, dgdef2, titulo)
    If res2 = 0 Then
        Sendkeys "{esc}", True
        Docmd RunMacro "Intradas otro profesor"
        Docmd GoToControl "RFC"
    Else
        Docmd Close
    End If
End If
If Is "Datos Incorrectos"
    Docmd GoToControl "RFC"
End If
End Sub
Sub Form_ReferetUpdate (Cancel As Integer)
    If IsNull(RFC) Then
        msg1 = "El RFC del Profesor es incorrecto!"
        dgdef1 = 48 + 0 + 0
        RES1 = MsgBox(msg1, dgdef1, titulo)
        Docmd GoToControl "RFC"
    End Sub
    Exit Sub
End If
If IsNull(paterno) Then
    msg1 = "El campo Apellido Paterno del Profesor Vacío!"
    dgdef1 = 48 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, titulo)
    Docmd GoToControl "paterno"
    Exit Sub
End If
If IsNull(NOMA_PROF) Then

```

```

msg1 = "El campo Nombre del Profesor Vacío"
dgdgf1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, título)
DoCmd GoToControl "nombre"
Exit Sub
End If
End Sub

Sub Form_Error (DataErr As Integer, Response As Integer)
Response = DATA_ERRORCONTINUE
Const MASCARAINCORRECTA = 2270
If DataErr = MASCARAINCORRECTA Then
msg1 = "RFC incorrecto"
dgdgf1 = 0 + 48 + 0
RES1 = MsgBox(msg1, dgdef1, título)
DoCmd CancelEvent
End If
If DataErr = CLAVLIDTLCALVA Then
m2 = "Existe un Profesor con el mismo RFC"
f2 = 48 + MB_LCONCALTO + MB_DIBROTONZ
r2 = MsgBox(m2, f2, título)
End If
End Sub

Sub Form_Ops_n_Cancel As Integer)
DoCmd RunMacro "n_rtrados. otro profesor"
End Sub

Sub RFC_beforeUpdate (Cancel As Integer)
If Len(rfc) < 10 Then
msg1 = "El RFC del Profesor es incorrecto"
dgdgf1 = 0 + 48 + 0
RES1 = MsgBox(msg1, dgdef1, título)
DoCmd CancelEvent
End If
Dim mIB() As Database, mitable As Recordset
Set mIB() = dbengine.workspaces(0).databases(0)
Set mitable = mIB().OpenRecordset("TRONTRON", DB_OPEN_TABLE)
mitable.Index = "TRFR" * (mitable.cuenta)
mitable.Seek "=", Form!alrus de profesor||RFC
If (mitable.NoMatch) Then
Else
m2 = "Existe un Profesor con el mismo RFC"
f2 = 48 + MB_LCONCALTO + MB_DIBROTONZ
r2 = MsgBox(m2, f2, título)
DoCmd CancelEvent
End If
End Sub

Sub salir_Click ()
msg2 = "¿Dar de alta otro Profesor?"
dgdgf2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdef2, título)
If res2 = 4 Then
Sendkeys "ewj", True
DoCmd RunMacro "Dar de alta otro profesor"
DoCmd GoToControl "RFC"
Else
DoCmd Close
End If
End Sub

Sub Cancelar_Click ()
DoCmd Close
End Sub

Sub Form_Open (Cancel As Integer)
Dim tex As Database, mitable As Recordset
Set tex = dbengine.workspaces(0).databases(0)
Set mitable = tex.OpenRecordset("alumnos", DB_OPEN_TABLE)
mitable.Index = "cuenta"
mitable.Seek "=", "0000000"
If (mitable.NoMatch) Then
MsgBox "no hay provisionales, crea el primer"
MsgBox "provisional" = "0000000"
Else
Buscamos el último número para generar el siguiente
mitable.MoveLast
numero = (Val(mitable(cuenta)) + 1)
MsgBox "provisional" = numero
End If
mitable.Close
m3 = "El Número de cuenta ya existe"
f3 = MB_RIFICAN + MB_LCONCALTO + MB_DIBROTONZ
r3 = MsgBox(m3, f3, título)
End Sub

Sub Provisional_AfterUpdate ()
Dim tex As Database, mitable As Recordset
Set tex = dbengine.workspaces(0).databases(0)
Set mitable = tex.OpenRecordset("alumnos", DB_OPEN_TABLE)
mitable.Index = "cuenta"
mitable.Seek "=", Form!alrus de Cuenta Provisional||Provisional
If mitable.NoMatch Then
mitable.Close
DoCmd OpenForm "datos generales (No 1 'NAMI'"
Else
m3 = "El Número de cuenta ya existe"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, f3, título)
If r3 = 2 Then
DoCmd Close
End If
End If
End Sub

Sub Isca_w_dar_de_baja_AfterUpdate ()
Me!jetiqueta clave| Visible = True
Me!jetiqueta nombre| Visible = True
Me!Llave_BKCAL.Visible = True
Me!NOMBRE_BK| Visible = True
Me!manej Visible = True
Me!Par de Baja| Enabled = True
DoCmd ApplyFilter "TIPO Baja de Baja"
End Sub

Sub Isca_w_dar_de_baja_NoIsca (NewData As String, Response As Integer)
Response = DATA_ERRORCONTINUE

```

```

msg1 = "La Beca debe contener con un valor de la lista"
dgdelf1 = 48 + 0 + 0
RfA1 = MsgBox(msg1, dgdelf1, titulo)
End Sub
Sub Consulta_Click ()
DoCmd.OpenForm "Catalogo de Becas"
End Sub
Sub Dar_de_baja_Click ()
pf = "¿Desa dar de Baja la Beca?"
d1 = 4 + 32 + 0
r1 = MsgBox(p1, d1, titulo)
If r1 = 6 Then
Dim miID As Database, Matable As Records1
Set miID = DBOEngine.Workspaces(0).Database(0)
Set Matable = miID.OpenRecordset("ALUMINOS", DB_OPEN_TABLE)
Matable.Index = "IDCA" 'Establece el índice a usar
Matable.Seek "=", Form!Baja de Becas!IDBeca a dar de baja
If Not (Matable.NoMatch) Then
msg1 = "El registro no se puede eliminar ya que al menos un Alumno tiene esta Beca!"
dgdelf1 = 48 + 0 + 0
RfA1 = MsgBox(msg1, dgdelf1, titulo)
Exit Sub
Else
msg1 = "La Beca se ra eliminado definitivamente!"
dgdelf1 = 48 + 0 + 0
RfA1 = MsgBox(msg1, dgdelf1, titulo)
DoCmd.ShowWarnings True
On Error GoTo Err_Dar_de_baja_Click
DoCmd.DeleteAllInTable A_JORMBAR, A_IDIYMIANI, A_SILECTRECORDER_V2, A_MIMU_VIRZ0
DoCmd.DeleteAllInTable A_JORMBAR, A_IDITMILNI, A_DELETE_V2, A_MIMU_VIRZ0
DoCmd.DeleteAllInTable A_JORMBAR, A_RECORDSMINI, A_RPTRES1
Exit_Dar_de_baja_Click
Exit Sub
Err_Dar_de_baja_Click
msg1 = "Se canceló la operación!"
dgdelf1 = 48 + 0 + 0
RfA1 = MsgBox(msg1, dgdelf1, titulo)
Resume Exit_Dar_de_baja_Click
DoCmd.DeleteAllInTable A_JORMBAR, A_IDITMILNI, A_SILECTRECORDER_V2, A_MIMU_VIRZ0
DoCmd.DeleteAllInTable A_JORMBAR, A_IDITMILNI, A_DELETE_V2, A_MIMU_VIRZ0
DoCmd.DeleteAllInTable A_JORMBAR, A_RECORDSMINI, A_RPTRES1, A_MIMU_VIRZ0
Me!Dar de Baja.Enabled = False
End If
Matable.Close
End If 'En caso de no eliminar
End Sub
Sub Form_Open (Cancel As Integer)
Me!Etiqueta clave1.Visible = False
Me!Etiqueta nombre1.Visible = False
Me!LAVI_BICAI.Visible = False
Me!INSMIRE_BIC1.Visible = False
Me!marcel.Visible = False
Me!Dar de Baja.Enabled = False
End Sub
Sub salir_Click ()
SendKeys "{esc}", True
DoCmd.Close
End Sub
Sub Aceptar_Click ()
DoCmd.OpenForm "Inscripcion Actual"
End Sub
Sub Boton1_Click ()
DoCmd.ShowWarnings True
End Sub
Sub Cancelar_Click ()
Sub Cancelar_Click ()
Me!Seleccionar Material.Enabled = True
Me!Seleccionar grupo1.Enabled = True
End Sub
Sub Consulta_grupos_Click ()
DoCmd.OpenForm "QUI MATRIA"
End Sub
Sub Dar_de_baja_Click ()
pf = "¿LA CLASE SERÁ EJECUTADA DE NUEVO (VARIABLE) 1) SI SÍ O CONTINUAR?"
d1 = 4 + 32 + 0
r1 = MsgBox(p1, d1)
If r1 = 7 Then
DoCmd.DeleteAllInTable A_JORMBAR, A_IDITMILNI, A_SILECTRECORDER_V2, A_MIMU_VIRZ0
DoCmd.DeleteAllInTable A_JORMBAR, A_IDITMILNI, A_DELETE_V2, A_MIMU_VIRZ0
DoCmd.DeleteAllInTable A_JORMBAR, A_RECORDSMINI, A_RPTRES1
msg2 = "Dar de baja otra clase?"
dgdelf2 = 4 + 32 + 0
r2 = MsgBox(msg2, dgdelf2, titulo)
If r2 = 7 Then
DoCmd.Close
Else
Me!Seleccionar Material.Enabled = True
Me!Seleccionar grupo1.Enabled = True
End If
Else
SendKeys "{esc}", True
msg2 = "Dar de baja otra clase?"
dgdelf2 = 4 + 32 + 0
r2 = MsgBox(msg2, dgdelf2, titulo)
If r2 = 7 Then
DoCmd.Close
Else
Me!Seleccionar Material.Enabled = True
Me!Seleccionar grupo1.Enabled = True
End If
End If
End Sub
Sub Form_AfterUpdate ()
Me!Seleccionar grupo1.Requery
End Sub
Sub Form_Open (Cancel As Integer)
DoCmd.ShowWarnings False

```

```

Dim tipoObjeto As Integer, nombreObjeto As String, estadoObjeto As Integer
tipoObjeto = A_form
nombreObjeto = "Semestre Actual"
If estadoObjeto = SysCmd(SysCmd_CANCEL) Or (Me.F114, tipoObjeto, nombreObjeto)
If estadoObjeto = 0 Then
msg1 = "No hay semestre actual. Elegir semestre y repetir la acción que desea"
dgdef4 = 0 + 0 + 0
res4 = MsgBox(msg4, dgdef4, titulo)
DoCmd.OpenForm "Semestre Actual"
DoCmd.CancelEvent
End If
Me!Bar.de_Baja.Enabled = False
End Sub
Sub grupo_RefreshUpdate (Cancel As Integer)
If Len(grupo) < 4 Then
m2 = "El número de grupo debe ser de 4 dígitos"
f2 = 48 + 0 + 518_DEPTONZ
r2 = MsgBox(m2, f2, titulo)
SendKeys "{esc}", True
DoCmd.CancelEvent
End If
If IsNull(grupo) Then
m2 = "El número de grupo debe ser de 4 dígitos"
f2 = 48 + 0 + 518_DEPTONZ
r2 = MsgBox(m2, f2, titulo)
SendKeys "{esc}", True
DoCmd.CancelEvent
End If
End Sub
Sub hacer_c1_combo_Click ()
End Sub
Sub rk_profesor_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRORCONTINUE
msg1 = "El nombre de l profesor debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RS1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub Nbr_Click ()
msg2 = "¿Bar de Baja otra clase?"
dgdef2 = 4 + 52 + 0
res2 = MsgBox(msg2, dgdef2, titulo)
If res2 = 7 Then
DoCmd.Close
Else
Me!Seleccionar_Materia.Enabled = True
Me!Seleccionar_grupo.Enabled = True
End If
End Sub
Sub Nlon_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRORCONTINUE
msg1 = "El salón debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RS1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub Seleccionar_grupo_AfterUpdate ()
Me!Seleccionar_Materia.Enabled = True
DoCmd.GoToControl "Materia"
Me!Seleccionar_grupo.Enabled = False
Me!Bar.de_Baja.Enabled = True
DoCmd.ApplyFilter "Filtro para eliminar clase"
End Sub
Sub Seleccionar_grupo_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRORCONTINUE
msg1 = "El grupo debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RS1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub Seleccionar_Materia_AfterUpdate ()
Me!Seleccionar_grupo.Enabled = True
Me!Seleccionar_grupo.Require
End Sub
Sub Seleccionar_Materia_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRORCONTINUE
msg1 = "La Materia debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RS1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub Ver_tira_de_materias_Click ()
Dim m1! As Database, M1tabla As Recordset
Set m1! = DEngine.Workspaces(0).Databases(0)
Set M1tabla = m1!.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
M1tabla.Index = "MATERIA" 'Establece el índice activo
M1tabla.Seek "n", M1tabla
If M1tabla.NoMatch() Then
m2 = "El Alumno no está inscrito en ningún materia"
f2 = 48 + 0 + 0
r2 = MsgBox(m2, f2)
Else
DoCmd.OpenForm "Ver tira de Materias", , "In_cuenta" = formularios\Ficha de materias y grupos\CUENTA"
End If
End Sub
Sub Consulta_Click ()
DoCmd.OpenForm "Catálogo de Estudios"
End Sub
Sub Bar.de_Baja_Click ()
p1 = "¿Vaca dar de Baja el Estada?"
d1 = 4 + 52 + 0
r1 = MsgBox(p1, d1, titulo)
If r1 = 0 Then
Dim m1! As Database, M1tabla As Recordset
Set m1! = DEngine.Workspaces(0).Databases(0)
Set M1tabla = m1!.OpenRecordset("ALUMNOS", DB_OPEN_TABLE)
M1tabla.Index = "ESTADO" 'Establece el índice activo
M1tabla.Seek "e", formularios de estados\Estado a dar de baja)
If Not (M1tabla.NoMatch) Then
msg1 = "El registro no se puede eliminar ya que al menos un Alumno pertenece a este Estado"
dgdef1 = 48 + 0 + 0
RS1 = MsgBox(msg1, dgdef1, titulo)

```

```

Exit Sub

11a.
Dim mibd2 As Database, Mitabla2 As Recordset
Set mibd2 = DBEngine.Workspaces(0)(Database2)
Set Mitabla2 = mibd2.OpenRecordset("ALUMNOS", DB_OPEN_TABLE)
Mitabla2.Index = "ESTADO (DECLARACION)" Establece el índice activo
If Not (Mitabla2.NoMatch) Then
    msg1 = "El registro no se puede eliminar ya que al menos un Alumno cursó su Licenciatura en este Estado"
    dgdef1 = 48 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, titulo)
Exit Sub

11b.
Dim mibd3 As Database, Mitabla3 As Recordset
Set mibd3 = DBEngine.Workspaces(0)(Database3)
Set Mitabla3 = mibd3.OpenRecordset("ALUMNOS", DB_OPEN_TABLE)
Mitabla3.Index = "ESTADO (GRADUACION)" Establece el índice activo
If Not (Mitabla3.NoMatch) Then
    msg1 = "El registro no se puede eliminar ya que al menos un Alumno cursó un Posgrado en este Estado"
    dgdef1 = 48 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, titulo)
Exit Sub

11c.
msg1 = "El estado será eliminado definitivamente!"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
DoCmd.ShowWarning True
DoCmd.ShowMenuItem A_FORMBAR, A_EDITMENU, A_SELECTRECORD_VZ, A_MENU_VIRZO
DoCmd.ShowMenuItem A_FORMBAR, A_EDITMENU, A_DELETE_VZ, A_MENU_VIRZO
DoCmd.ShowMenuItem A_FORMBAR, A_EDITMENU, A_REFRESH, A_MENU_VIRZO
Me!Bar de Baja.Enabled = False
End If
Mitabla3.Close
End If
Mitabla2.Close
End If
Mitabla1.Close
End If 'In caso de no eliminar
End Sub
Sub Estado_a_baja_AfectaDatos ()
Me!Etiqueta.clave1.Visible = True
Me!Etiqueta.nombre1.Visible = True
Me!CLAVE_ESTADO.Visible = True
Me!BORNOMBRE_ESTADO.Visible = True
Me!marco1.Visible = True
Me!Dar de Baja.Enabled = True
DoCmd.ApplyFilter "TIPO RAJA DE ESTADO"
End Sub
Sub Estado_a_dar_de_baja_NotFound (NewData As String, Response As Integer)
Response = DATA_ERRORCONTINUI
msg1 = "El estado debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub Form_Opin (Cancel As Integer)
Me!Etiqueta.clave1.Visible = False
Me!Etiqueta.nombre1.Visible = False
Me!CLAVE_ESTADO.Visible = False
Me!BORNOMBRE_ESTADO.Visible = False
Me!marco1.Visible = False
Me!Dar de Baja.Enabled = False
End Sub
Sub Salir_Click ()
SaveKey "Esc", True
DoCmd.Close
End Sub

Sub Acabar_Click ()
DoCmd.OpenForm "Inscripcion Actual"
End Sub
Sub Dar_de_baja_Click ()
p1 = "¿Usted cancela inscripción a esta materia?"
d1 = 4 + '2' + 0
r1 = MsgBox(p1, d1)
If r1 = 6 Then
    msg1 = "La baja será definitiva!"
    dgdef1 = 48 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, titulo)
    DoCmd.ShowWarning True
    DoCmd.ShowMenuItem A_FORMBAR, A_EDITMENU, A_SELECTRECORD_VZ, A_MENU_VIRZO
    DoCmd.ShowMenuItem A_FORMBAR, A_EDITMENU, A_DELETE_VZ, A_MENU_VIRZO
    DoCmd.ShowMenuItem A_FORMBAR, A_EDITMENU, A_REFRESH, A_MENU_VIRZO
    Me!Dar de Baja.Enabled = False
End If
End Sub
Sub Form_Opin (Cancel As Integer)
Me!Etiqueta.clave1.Visible = False
Me!Etiqueta.profesor1.Visible = False
Me!clave1.Visible = False
Me!grupo1.Visible = False
Me!profesor1.Visible = False
Me!Cuadro de datos1.Visible = False
Me!materna1.Visible = False
Me!sombra1.Visible = False
Me!Dar de Baja.Enabled = False
End Sub
Sub Salir_Click ()
msg2 = "¿Cancelar otra Materia?"
dgdef2 = 4 + '2' + 0
res2 = MsgBox(msg2, dgdef2, titulo)
If res2 = 7 Then
    DoCmd.Close
    DoCmd.Close a_form, "Lee cuenta a dar de baja inscripción"
Else
    Formulario1.Baja de inscripción1.Etiqueta.clave1.Visible = False
    Formulario1.Baja de inscripción1.Etiqueta.profesor1.Visible = False
    Formulario1.Baja de inscripción1.clave1.Visible = False
    Formulario1.Baja de inscripción1.grupo1.Visible = False

```



```

Dim miBD As Database, MiTabla As Recordset
Set miBD = DBEngine.Workspace(0).Database(0)
Set MiTabla = miBD.OpenRecordset("ALUMNOS", DB_OPEN_TABLE)
MiTabla.Index = "idmat" 'Establece el índice activo
MiTabla.Seek "=", Form!Baja de Maestrias\Sección a dar de baja
If Not (MiTabla.NoMatch) Then
    msg1 = "El registro no se puede eliminar ya que al menos un Alumno pertenece a esta área"
    dgdef1 = 48 + 0 + 0
    RLS1 = MsgBox(msg1, dgdef1, titulo)
    Exit Sub
Else
    msg1 = "Esta área será eliminada definitivamente!"
    dgdef1 = 48 + 0 + 0
    RLS1 = MsgBox(msg1, dgdef1, titulo)
    DoCmd.ShowWarnings True
    DoCmd DoMenuItem A_FORMBAR, A_FUTIMENU, A_SILETRACORD_V2, A_MENU_V1R20
    DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, A_DELETE_V2, A_MENU_V1R20
    DoCmd DoMenuItem A_FORMBAR, A_RECOPRSMENU, A_RETIRSI, A_MENU_V1R20
    Me!Dar de Baja.Enabled = False
End If
MiTabla.Close
End If 'En caso de no eliminar
End Sub
Sub Form_Open (Cancel As Integer)
Me!Etiqueta clave.Visible = False
Me!Etiqueta nombre.Visible = True
Me!CLV_SXC.Visible = False
Me!NOM_SFC.Visible = False
Me!Etiqueta obligatorio.Visible = False
Me!CRL_OBL.Visible = False
Me!marco.Visible = False
Me!Dar de Baja.Enabled = False
End Sub
Sub Sub_Click ()
SendKeys "f5", True
DoCmd.Close
End Sub
Sub Sección_a_dar_de_baja_AfterUpdate ()
Me!Etiqueta clave.Visible = True
Me!Etiqueta nombre.Visible = True
Me!CLV_SXC.Visible = True
Me!NOM_SFC.Visible = True
Me!marco.Visible = True
Me!Etiqueta obligatorio.Visible = True
Me!CRL_OBL.Visible = True
Me!Dar de Baja.Enabled = True
DoCmd ApplyFilter "FILTRO Baja de Maestria"
End Sub
Sub Sección_a_dar_de_baja_NotInList (NewData As String, Response As Integer)
Response = DATA_EXISTS_CONTINUE
msg1 = "El Área debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RLS1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub Dar_de_Baja_Click ()
p1 = "¿Desea dar de Baja esta Materia?"
d1 = 4 + 32 + 0
r1 = MsgBox(p1, d1, titulo)
If r1 = 0 Then
Dim miBD As Database, MiTabla As Recordset
set miBD = DBEngine.Workspace(0).Database(0)
Set MiTabla = miBD.OpenRecordset("CLAS", DB_OPEN_TABLE)
MiTabla.Index = "MATFRM" 'Establece el índice activo
MiTabla.Seek "=", Form!Baja de materia\materia a dar de baja
If Not (MiTabla.NoMatch) Then
    If (MiTabla![SEMESTRI] = Form!semestre.Actual)![SEMESTRI] Then
        msg1 = "Existen grupos de esta Materia sem actual, deberá dar de baja primero la clase."
        dgdef1 = 48 + 0 + 0
        RLS1 = MsgBox(msg1, dgdef1, titulo)
    Else
        msg1 = "Existen grupos de esta Materia para el semestre " & MiTabla![SEMESTRI] & ", deberá dar de baja primero la clase."
        dgdef1 = 48 + 0 + 0
        RLS1 = MsgBox(msg1, dgdef1, titulo)
    End If
Else
    'no estuvo en clase ahora busca en MAT_PLAN
    'Verificar en tabla de Plan de estudios
Dim miBD1 As Database, MiTabla1 As Recordset
set miBD1 = DBEngine.Workspace(0).Database(0)
Set MiTabla1 = miBD1.OpenRecordset("MAT_PLAN", DB_OPEN_TABLE)
MiTabla1.Index = "MATERIA" 'Establece el índice activo
MiTabla1.Seek "=", Form!Baja de maestria\materia a dar de baja
If Not (MiTabla1.NoMatch) Then
    msg1 = "Existen Planes de Estudio con esta Materia, deberá dar de baja primero dicho Plan"
    dgdef1 = 48 + 0 + 0
    RLS1 = MsgBox(msg1, dgdef1, titulo)
    Exit Sub
End If
msg1 = "La materia será eliminada definitivamente!"
dgdef1 = 48 + 0 + 0
RLS1 = MsgBox(msg1, dgdef1, titulo)
DoCmd.ShowWarnings True
DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, A_MICTRACORD_V2, A_MENU_V1R20
DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, A_DELETE_V2, A_MENU_V1R20
DoCmd DoMenuItem A_FORMBAR, A_RECOPRSMENU, A_RETIRSI, A_MENU_V1R20
Me!Dar de Baja.Enabled = False
MiTabla1.Close
End If
MiTabla.Close
End If 'En caso de no eliminar
End Sub
Sub Form_beforeUpdate (Cancel As Integer)
If IsNull(CLV_MAT) Then
msg1 = "La clave de la Materia es incorrecta"
dgdef1 = 48 + 0 + 0
RLS1 = MsgBox(msg1, dgdef1, titulo)
DoCmd GoToControl "mat_nueva"
DoCmd CancelEvent
Exit Sub

```

```

End If
If Len(CLV_MAT) < 4 Then
    msg1 = "La clave de la Materia es incorrecta!"
    dgdef1 = 48 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, titulo)
    DoCmd GoToControl "MAT_NI"VVA"
    DoCmd CancelEvent
    Exit Sub
End If
If IsNull(NOM_MAT) Then
    msg1 = "El nombre de la materia es incorrecto!"
    dgdef1 = 48 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, titulo)
    DoCmd GoToControl "nombre_materia"
    DoCmd CancelEvent
    Exit Sub
End If
If IsNull(creditos) Then
    msg1 = "Debe especificar Numero de Créditos!"
    dgdef1 = 48 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, titulo)
    DoCmd GoToControl "creditos"
    DoCmd CancelEvent
    Exit Sub
End If
Sub Form_Open (Cancel As Integer)
Dim tipoObjeto As Integer, nombresObjeto As String, estadoObjeto As Integer
tipoObjeto = A_Form
nombreObjeto = "Semestre Actual"
estadoObjeto = SysCmd(SysCmdSETOBJECTSTATE, tipoObjeto, nombreObjeto)
If estadoObjeto = 0 Then
    msg4 = "No hay semestre actual. Elegir semestre y repetir la acción que desea!"
    dgdef4 = 0 + 0 + 0
    res4 = MsgBox(msg4, dgdef4, titulo)
    DoCmd OpenForm "Semestre Actual"
    DoCmd CancelEvent
Else
Me!Dar de Baja!Enabled = False
Me!clave de la materia!visible = False
Me!sombra de clave!visible = False
Me!etiqueta clave!visible = False
Me!etiqueta nombre materia!visible = False
Me!nombre de la materia!visible = False
Me!sombra nombre!visible = False
Me!etiqueta creditos!visible = False
Me!CREDITOS!visible = False
Me!sombra creditos!visible = False
Me!etiqueta campo!visible = False
Me!elpe campo!visible = False
End If
Sub mat_nueva_BeforeUpdate (Cancel As Integer)
If IsNull(mat_nueva) Then
    msg1 = "La clave es incorrecta!"
    dgdef1 = 48 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, titulo)
    SendKeys "{esc}", True
    DoCmd CancelEvent
End If
Dim mIBD As Database, MatTabla As Recordset
Set mIBD = DBEngine.Workspaces(0).Database(1)
Set MatTabla = mIBD.OpenRecordset("MATERIA") Dim _DFIN_LAB11
MatTabla.Index = "CLAVE". Establece el índice activo
MatTabla.Seek "=", forms!jaja de mat_nuevas!mat_nueva!
If Not (MatTabla.NoMatch) Then
    m2 = "Existe una materia con la misma clave!"
    f2 = 48 + 0 + 0 + MB_YESNO
    r2 = MsgBox(m2, f2, titulo)
    SendKeys "{esc}", True
    DoCmd CancelEvent
End If
Sub Materia_n_dar_de_baj_AfterUpdate ()
Me!Dar de Baja!Enabled = True
Me!clave de la materia!visible = True
Me!sombra de clave!visible = True
Me!etiqueta clave!visible = True
Me!etiqueta nombre materia!visible = True
Me!nombre de la materia!visible = True
Me!sombra nombre!visible = True
Me!etiqueta creditos!visible = True
Me!CREDITOS!visible = True
Me!sombra creditos!visible = True
Me!etiqueta campo!visible = True
Me!elpe campo!visible = True
DoCmd ApplyFilter "MATRO LAJA 11. MATERIAS"
End Sub
Sub Materia_n_dar_de_baj_NoMatch (NewData As String, Response As Integer)
Response = DATA_CONTINUE
msg1 = "La Materia debe coincidir con un valor de la lista!"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub nombre_materia_BeforeUpdate (Cancel As Integer)
If IsNull(NOM_MAT) Then
    msg1 = "El campo Nombre de Materia Vacío!"
    dgdef1 = 0 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, titulo)
    DoCmd CancelEvent
    DoCmd GoToControl "nombre_materia"
End If
End Sub
Sub Subr_Click ()
    msg2 = "Dar de baja otra Materia?"
    dgdef2 = 4 + 32 + 0
    res2 = MsgBox(msg2, dgdef2, titulo)
    If res2 = 7 Then
        DoCmd Close
    
```

```

End If

Sub Consulta_Click ()
    DoCmd.OpenForm "Catálogo de Pasos"
End Sub

Sub Dar_de_Baja_Click ()
    p1 = "¿A qué dar de Baja el Paso?"
    d1 = 4 * 32 + 0
    r1 = MsgBox(p1, d1, título)
    If r1 = 0 Then
        Dim miDB As Database, MiTabla As Recordset
        %1 miDB = D:\Program Works\pac\01 Database\sf01
        %1 MiTabla = miDB.OpenRecordset("ALUMINOS", DB_OPEN_TABLE)
        MiTabla.Seek "=", Form!Baja de Pasos!Filtros a dar de Baja!
        If Not (MiTabla.Recordset) Then
            msg1 = "El registro no se pudo eliminar ya que al menos un Alumno curso o la naturaleza en este Paso!"
            dgdx1 = 48 * 0 + 0
            RES1 = MsgBox(msg1, dgdx1, título)
        End Sub
    Else
        Dim miDB2 As Database, MiTabla2 As Recordset
        %1 miDB2 = D:\Program Works\pac\01 Database\sf01
        %1 MiTabla2 = miDB2.OpenRecordset("ALUMINOS", DB_OPEN_TABLE)
        MiTabla2.Seek "=", Form!Baja de Pasos!Filtros a dar de Baja!
        If Not (MiTabla2.Recordset) Then
            msg1 = "El registro no se pudo eliminar ya que al menos un Alumno curso o la naturaleza en este Paso!"
            dgdx1 = 48 * 0 + 0
            RES1 = MsgBox(msg1, dgdx1, título)
        End Sub
    Else
        msg1 = "El Paso será eliminado definitivamente!"
        dgdx1 = 48 * 0 + 0
        RES1 = MsgBox(msg1, dgdx1, título)
        DoCmd.ShowWarnings True
        DoCmd.DeleteInTable A_FORMBAR, A_EDITITEM1, A_DELETE_VZ, A_MENU_VR20
        DoCmd.DeleteInTable A_FORMBAR, A_EDITITEM1, A_DELETE_VZ, A_MENU_VR20
        DoCmd.DeleteInTable A_RECORDSMENU, A_REFRESH1, A_MENU_VR20
        Me!Bar de Baja!Enabled = False
    End If
End Sub

Sub Atable2_Close
    End If
    MiTabla.Close
End Sub

Sub Form_Open (Cancel As Integer)
    Me!Etiqueta clave1.Visible = False
    Me!Etiqueta nombre1.Visible = False
    Me!CLAVE_PAIS.Visible = False
    Me!NOMBRE_PAIS.Visible = False
    Me!Iniciar1.Visible = False
    Me!Bar de Baja!Enabled = False
End Sub

Sub Pas_a_dar_de_Baja_AltosPelado ()
    Me!Etiqueta clave1.Visible = True
    Me!Etiqueta nombre1.Visible = True
    Me!CLAVE_PAIS.Visible = True
    Me!NOMBRE_PAIS.Visible = True
    Me!Iniciar1.Visible = True
    Me!Bar de Baja!Enabled = True
    DoCmd.ApplyFilter "TRFKO Baja de Pasos"
End Sub

Sub Pas_a_dar_de_Baja_Notinist (N, w) As String, Response As Integer)
    %1 Response = DATA_ERROCONTINUA
    msg1 = "El Paso debe comenzar con un valor de la lista!"
    dgdx1 = 48 * 0 + 0
    RES1 = MsgBox(msg1, dgdx1, título)
End Sub

Sub salir_Click ()
    %1 ndk.ya "ex1", True
    DoCmd.Close
End Sub

Sub Consulta_Click ()
    DoCmd.OpenForm "Catálogo de salones"
End Sub

Sub Dar_de_Baja_Click ()
    p1 = "¿A qué dar de Baja el Salón?"
    d1 = 4 * 32 + 0
    r1 = MsgBox(p1, d1, título)
    If r1 = 0 Then
        Dim miDB As Database, MiTabla As Recordset
        %1 miDB = D:\Program Works\pac\01 Database\sf01
        %1 MiTabla = miDB.OpenRecordset("CLAVES", DB_OPEN_TABLE)
        MiTabla.Seek "=", Form!Baja de salones!salon a dar de Baja!
        If Not (MiTabla.Recordset) Then
            msg1 = "El registro no se pudo eliminar ya que al menos un Clave o impartida en este salón!"
            dgdx1 = 48 * 0 + 0
            RES1 = MsgBox(msg1, dgdx1, título)
        End Sub
    Else
        msg1 = "El salón se es eliminado definitivamente!"
        dgdx1 = 48 * 0 + 0
        RES1 = MsgBox(msg1, dgdx1, título)
        DoCmd.ShowWarnings True
        DoCmd.DeleteInTable A_FORMBAR, A_EDITITEM1, A_DELETE_VZ, A_MENU_VR20
        DoCmd.DeleteInTable A_FORMBAR, A_EDITITEM1, A_DELETE_VZ, A_MENU_VR20
        DoCmd.DeleteInTable A_RECORDSMENU, A_REFRESH1, A_MENU_VR20
        Me!Bar de Baja!Enabled = False
    End If
End Sub

Sub Atable2_Close
    End If
    MiTabla.Close
End Sub

Sub Form_Open (Cancel As Integer)
    Me!Etiqueta clave1.Visible = False
    Me!Etiqueta nombre1.Visible = False

```

```

Me!{N_SALON} Visible = False
Me!{TRUCACION} Visible = False
Me!{marco} Visible = False
Me!{Dar de Baja} Enabled = False
End Sub
Sub Salir_Click ()
SendKeys "{esc}", True
DoCmd Close
End Sub
Sub Salon_a_dar_de_baja_AfterUpdate ()
Me!{etiqueta clave} Visible = True
Me!{etiqueta nombre} Visible = True
Me!{N_SALON} Visible = True
Me!{TRUCACION} Visible = True
Me!{marco} Visible = True
Me!{Dar de Baja} Enabled = True
DoCmd ApplyFilter "TIPO Baja de salones"
End Sub
Sub Salon_a_dar_de_baja_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El salon debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub Consulta_Click ()
DoCmd OpenForm "Catalogo de titulos"
End Sub
Sub Dar_de_baja_Click ()
pt = 7/15 sea dar de Baja la Carrera
d1 = 4 + 32 + 0
r1 = MsgBox(pt, d1, TIT)
If r1 = 1 Then
Dim miTab As Database, miTabla As Recordset
Set miTab = Databases.Workspaces(0).Databases(0)
Set miTabla = miTab.OpenRecordset("ALUMNOS", DB_OPEN_TABLE)
miTabla Index = "TITULO" 'establece el indice activo.
miTabla Seek "=", form!{baja de titulos}||Titulo a dar de baja
If Not miTabla.NoMatch() Then
msg1 = "El registro no se puede eliminar ya que al menos un Alumno cursó esta licenciatura!"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, TIT)
Exit Sub
Else
msg1 = "La Carrera será eliminado definitivamente!"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, TIT)
DoCmd SetWarnings True
DoCmd DoMenuItem A_FORMATAR, A_EDITAR, A_SILECIOS, V2, , A_MENU_VERZO
DoCmd DoMenuItem A_FORMATAR, A_EDITAR, A_DELETE, V2, , A_MENU_VERZO
DoCmd DoMenuItem A_FORMATAR, A_RECORDSMENU, A_RESET, , A_MENU_VERZO
Me!{Dar de Baja} Enabled = False
End If
miTabla Close
End If 'En caso de no eliminar
End Sub
Sub Form_Open (Cancel As Integer)
Me!{etiqueta clave} Visible = False
Me!{etiqueta nombre} Visible = False
Me!{CLAVE_TIT} Visible = False
Me!{TIPO} Visible = False
Me!{marco} Visible = False
Me!{Dar de Baja} Enabled = False
End Sub
Sub Dar_de_baja_AfterUpdate ()
End Sub
Sub Salir_Click ()
SendKeys "{esc}", True
DoCmd Close
End Sub
Sub Titulo_a_dar_de_baja_AfterUpdate ()
Me!{etiqueta clave} Visible = True
Me!{etiqueta nombre} Visible = True
Me!{CLAVE_TIT} Visible = True
Me!{TIPO} Visible = True
Me!{marco} Visible = True
Me!{Dar de Baja} Enabled = True
DoCmd ApplyFilter "TIPO Baja de titulos"
End Sub
Sub Titulo_a_dar_de_baja_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "La Carrera debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, TIT)
End Sub
Sub cancelar_Click ()
DoCmd Close
DoCmd Close a_form, "La consulta para baja definitiva"
End Sub
Sub EJECUTAR_Click ()
pt = 7/11, AUMENTO SI RA ELIMINADO AL IGU AL QUE SUS MATEMATICAS Y PLAN DE ESTUDIOS, ¿DESEA CONTINUAR?
d1 = 4 + 0 + 0
r1 = MsgBox(pt, d1, titulo)
If r1 = 1 Then
DoCmd DoMenuItem A_FORMATAR, A_EDITAR, A_SILECIOS, V2, , A_MENU_VERZO
DoCmd DoMenuItem A_FORMATAR, A_EDITAR, A_DELETE, V2, , A_MENU_VERZO
DoCmd DoMenuItem A_FORMATAR, A_RECORDSMENU, A_RESET, , A_MENU_VERZO
Else
SendKeys "{ESC}", True
End If
End Sub
Sub Iniciar_en_Click ()
Dim tex As Database, miTabla As Recordset
Set tex = Databases.Workspaces(0).Databases(0)
Set miTabla = tex.OpenRecordset("inscrito", DB_OPEN_TABLE)
miTabla Index = "inscrito"
miTabla Seek "=", formularios||Lee consulta para baja definitiva||consultar
If miTabla.NoMatch() Then 'si NO está mandando un mensaje

```

```

msg3 = "El alumno no tiene materias en el semestre actual"
dgdef3 = 48 + 0 + 0
RES3 = MsgBox(msg3, dgdef3, titulo)
mitabla Close
Exit Sub
Else
Do Until mitabla.eof
If mitabla!semestre1 = formularios!Semestre Actual||Semestre1 Then
GoTo abrir:
Exit Do
Else
mitabla MoveNext
End If
Loop
m1 = "El alumno no tiene materias en el semestre actual"
f1 = 48 + 0 + 0
r1 = MsgBox(m1, f1, titulo)
mitabla Close
Exit Sub
Exit Sub
End If
mitabla Close
abrir:
DoCmd OpenForm "Ver lista de materias...", "it_cuenta"|> formularios|lee cuenta para baja definitiva|||ccuenta"
mitabla Close
Exit Sub
End Sub

Sub Consultar_Profesores_Check ()
DoCmd OpenForm "Consulta de Profesores"
End Sub
Sub Dar_de_Alta_Click ()
If IsNull(rfo) Then
msg0 = "El RFL es incorrecto"
dgdef0 = 48 + 0 + 0
RES0 = MsgBox(msg0, dgdef0, titulo)
DoCmd GoToControl "rfa"
Exit Sub
End If
Dim mtbl As Database, Mitabla As Recordset
Set mtbl = DSRigree.Workspaces(0).Databases(f0)
Set Mitabla = mtbl.OpenRecordset("PROFESOR", DB_OPEN_TABLE)
Mitabla.Index = "NRRFL" 'Establece el índice activo.
Mitabla.Seek "=", form!Alias de profesor|NRRFL
If (Mitabla.IsMatch) Then
msg1 = "¿Quieres dar de alta al Profesor?"
dgdef1 = 4 + 32 + 0
res1 = MsgBox(msg1, dgdef1, titulo)
If res1 = 6 Then
If IsNull(PATERNO) Then
msg1 = "El campo Apellido Paterno del Profesor Vacío"
dgdef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dgdef1, titulo)
DoCmd GoToControl "paterno"
Exit Sub
End If
If IsNull(MATERNO) Then
msg1 = "El campo Apellido Materno del Profesor Vacío"
dgdef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dgdef1, titulo)
DoCmd GoToControl "materno"
Exit Sub
End If
If IsNull(nom_prof) Then
msg1 = "El campo Nombre del Profesor Vacío"
dgdef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dgdef1, titulo)
DoCmd GoToControl "nombre"
Exit Sub
End If
DoCmd DoMenuItem A_FONMARR, A_BBJ, A_SAVERICORDI, A_MENU_VER20
msg2 = "¿Dar de alta otro Profesor?"
dgdef2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdef2, titulo)
If res2 = 7 Then
DoCmd Close
f0e
DoCmd GoToRecord a_form, "Alias de Profesor", A_NEWREC
DoCmd GoToControl "RFL"
End If
Else
DoCmd DoMenuItem A_FONMARR, A_LIYMENU, A_UNIDOFIELD, A_MENU_VFR20
msg2 = "¿Dar de alta otro Profesor?"
dgdef2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdef2, titulo)
If res2 = 7 Then
DoCmd Close
f0e
DoCmd GoToRecord a_form, "Alias de Profesor", A_NEWREC
DoCmd GoToControl "RFL"
End If
End If
Else
m2 = "¿Existe un Profesor con el mismo RFL?"
f2 = 48 + MB_ICONQUESTION + MB_YESNOCANCEL
r2 = MsgBox(m2, f2, titulo)
DoCmd DoMenuItem A_FONMARR, A_LIYMENU, A_UNIDOFIELD, A_MENU_VER20
DoCmd CancelEvent
msg2 = "¿Dar de alta otro Profesor?"
dgdef2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdef2, titulo)
If res2 = 7 Then
DoCmd Close
Else
DoCmd GoToRecord a_form, "Alias de Profesor", A_NEWREC
DoCmd GoToControl "RFL"
End If
End If
End Sub

```

```

Sub Dar_de_Baja_Click ()
msg1 = "¿EL PROFESOR SERÁ ELIMINADO DEFINITIVAMENTE, ASI COMO LAS CLASES Y TUTORIAS A SU CARGO? ¿DESEA CONTINUAR?"
dgdef1 = 4 + 32 + 0
res1 = MsgBox(msg1, dgdef1, titulo)
If res1 = 0 Then
DoCmd SendWarnings False
DoCmd DoMenuItem A_FORMBAR, A_FITMENU, A_DELETECORD_V2,, A_MENU_VIERZO
DoCmd DoMenuItem A_FORMBAR, A_FITMENU, A_DELETE_V2,, A_MENU_VIERZO
DoCmd DoMenuItem A_FORMBAR, A_RECORDMENU, A_REMOVE,, A_MENU_VIERZO
Me!Bar.de.Baja.Enabled = False
Me!Etiqueta RFC.Visible = False
Me!Etiqueta Paterno.Visible = False
Me!Etiqueta materno.Visible = False
Me!Etiqueta Profesor.Visible = False
Me!Etiqueta campo.Visible = False
Me!PATERNO.Visible = False
Me!materno.Visible = False
Me!nombre.Visible = False
Me!HISTORIAL.Visible = False
Me!sombra paterno.Visible = False
Me!sombra historial.Visible = False
Me!sombra materno.Visible = False
Me!sombra nombre.Visible = False
Me!Etiqueta RFC.Visible = False
Me!sombra RFC.Visible = False
Me!RFC.Visible = False
Else
SendKeys "{esc}", True
End If 'En caso de no eliminar
End Sub

Sub Form_Open (Cancel As Integer)
Me!Bar.de.Baja.Enabled = False
Me!Etiqueta RFC.Visible = False
Me!Etiqueta Paterno.Visible = False
Me!Etiqueta materno.Visible = False
Me!Etiqueta Profesor.Visible = False
Me!Etiqueta campo.Visible = False
Me!PATERNO.Visible = False
Me!materno.Visible = False
Me!nombre.Visible = False
Me!HISTORIAL.Visible = False
Me!sombra paterno.Visible = False
Me!sombra historial.Visible = False
Me!sombra materno.Visible = False
Me!sombra nombre.Visible = False
Me!Etiqueta RFC.Visible = False
Me!sombra RFC.Visible = False
Me!RFC.Visible = False
End Sub

Sub Lista_de_Profesores_Actualizade ()
DoCmd RunMacro "Rellenar_profesor"
End Sub

Sub lista_de_profesores_BeforeUpdate (Cancel As Integer)
Me!Etiqueta RFC.Visible = True
Me!Etiqueta Paterno.Visible = True
Me!Etiqueta materno.Visible = True
Me!Etiqueta Profesor.Visible = True
Me!Etiqueta campo.Visible = True
Me!PATERNO.Visible = True
Me!materno.Visible = True
Me!nombre.Visible = True
Me!HISTORIAL.Visible = True
Me!sombra paterno.Visible = True
Me!sombra historial.Visible = True
Me!sombra materno.Visible = True
Me!sombra nombre.Visible = True
Me!Etiqueta RFC.Visible = True
Me!sombra RFC.Visible = True
Me!RFC.Visible = True
End Sub

Sub lista_de_profesores_Nullinitial (NewData As String, Response As Integer)
Response = DATA_ERR_CONTINUE
msg1 = "El RFC del Profesor debe coincidir con un valor de la lista"
dgdef1 = 0 + 48 + 0
res1 = MsgBox(msg1, dgdef1, titulo)
End Sub

Sub RFC_BeforeUpdate (Cancel As Integer)
If Len(rfc) < 10 Then
msg1 = "RFC incorrecto"
dgdef1 = 0 + 48 + 0
res1 = MsgBox(msg1, dgdef1, titulo)
DoCmd CancelEvent
End If
Dim miRD As Database, MiTabla As Recordset
Set miRD = DBEngine.Workspaces(0).Database(0)
Set MiTabla = miRD.OpenRecordset("PROFESOR", DB_OPEN_TABLE)
MiTabla.Index = "INDRC" 'Establece el índice activo
MiTabla.Seek "=", Form1.Modificar_profesor(RFC)
If (MiTabla.NoMatch) Then
Else
m2 = "¿Existe un Profesor con el mismo RFC?"
f2 = 48 + MB_CONFIRM + MB_OKCANCEL
r2 = MsgBox(m2, f2, titulo)
DoCmd CancelEvent
End If
End Sub

Sub Bar_Click ()
msg2 = "¿Bar de Baja u otro Profesor?"
dgdef2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdef2, titulo)
If res2 = 7 Then
DoCmd Close
Else
Me!Etiqueta RFC.Visible = False
Me!Etiqueta Paterno.Visible = False
Me!Etiqueta materno.Visible = False
Me!Etiqueta Profesor.Visible = False
Me!Etiqueta campo.Visible = False
Me!PATERNO.Visible = False

```

```

Me|materno|visible = False
Me|nombr|visible = False
Me|historia|visible = False
Me|sombra.paterno|visible = False
Me|sombra.historia|visible = False
Me|sombra.materno|visible = False
Me|señal.nombre|visible = False
Me|etiqueta.RFC|visible = False
Me|sombra.RFC|visible = False
Me|RFC|visible = False
End If
End Sub

Sub Botón644_Click ()
On Error GoTo Err_Botón644_Click
msg1 = "¿Realmente desea deshacer el registro?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 0 Then
DoCmd DoMenuItem A_FORMBAR, A_FHTMENU, A_UNDOFIELD, , A_MENU_VER20
DoCmd Close
Exit Botón644_Click
End Sub
Err_Botón644_Click:
Resume Exit_Botón644_Click
Else
Exit Sub
End If
End Sub

Sub Botón719_Click ()
On Error GoTo Err_Botón719_Click
msg1 = "¿Realmente desea deshacer el registro?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 0 Then
DoCmd DoMenuItem A_FORMBAR, A_FHTMENU, A_UNDOFIELD, , A_MENU_VER20
DoCmd Close
Exit Botón719_Click
End Sub
Err_Botón719_Click:
Resume Exit_Botón719_Click
Else
Exit Sub
End If
End Sub

Sub Botón727_Click ()
On Error GoTo Err_Botón727_Click
msg1 = "¿Realmente desea deshacer el registro?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 0 Then
DoCmd DoMenuItem A_FORMBAR, A_FHTMENU, A_UNDOFIELD, , A_MENU_VER20
DoCmd Close
Exit Botón727_Click
End Sub
Err_Botón727_Click:
Resume Exit_Botón727_Click
Else
Exit Sub
End If
End Sub

Sub Botón811_Click ()
msg1 = "¿Realmente desea cancelar alta?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 6 Then
DoCmd Close
End If
End Sub

Sub Botón813_Click ()
On Error GoTo Err_Botón813_Click
msg1 = "¿Realmente desea deshacer el registro?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 0 Then
DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, A_UNDOFIELD, , A_MENU_VER20
DoCmd Close
Exit Botón813_Click
End Sub
Err_Botón813_Click:
Resume Exit_Botón813_Click
Else
Exit Sub
End If
End Sub

Sub Botón821_Click ()
msg1 = "¿Realmente desea cancelar el cambio?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 6 Then
SendKeys "{ESC}"; True
DoCmd Close A_FORM, "Lee cuenta a modificar"
DoCmd Close A_FORM, "Cambiar datos generales"
End If
End Sub

Sub Botón823_Click ()
msg1 = "¿Realmente desea cancelar el cambio?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 0 Then
SendKeys "{ESC}"; True
DoCmd Close A_FORM, "Cambiar datos generales"
DoCmd Close A_FORM, "Lee cuenta a modificar"
End If
End Sub

Sub Botón825_Click ()
msg1 = "¿Realmente desea cancelar el cambio?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)

```

```

If RES1 = 0 Then
    SendKeys "{ESC}", True
    DoCmd Close A_FORM, "Lee cuenta a modificar"
    DoCmd Close A_FORM, "Cambiar datos generales"
End If
End Sub
Sub Botón827_Click ()
    msg1 = "¿Realmente desea cancelar el cambio?"
    dgdef1 = 4 + 32 + 0
    RES1 = MsgBox(msg1, dgdef1, título)
    If RES1 = 41 Then
        SendKeys "{ESC}", True
        DoCmd Close A_FORM, "Lee cuenta a modificar"
        DoCmd Close A_FORM, "Cambiar datos generales"
    End If
End Sub
Sub Botón840_Click ()
    On Error GoTo Err_Botón840_Click
    DoCmd DoMenuItem A_FORMBAR, A_EDITMENU1, 10, , A_MENU_VLR20
    Exit_Botón840_Click
End Sub
Err_Botón840_Click
MsgBox Error!
Resume Err_Botón840_Click
End Sub
Sub estado_de_titulo_NotInList (NewData As String, Response As Integer)
    Response = DATA_ERRCONTINUE
    msg1 = "La carrera debe coincidir con un valor de la lista"
    dgdef1 = 48 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, título)
End Sub
Sub codigo_postal_BeforeUpdate (Cancel As Integer)
    If Len(codigo_p) < 3 Then
        msg1 = "El código postal debe ser de 3 dígitos"
        dgdef1 = 48 + 0 + 0
        RES1 = MsgBox(msg1, dgdef1, título)
        DoCmd CancelEvent
    End If
End Sub
Sub estado_civil_NotInList (NewData As String, Response As Integer)
    Response = DATA_ERRCONTINUE
    msg1 = "El estado civil debe coincidir con un valor de la lista"
    dgdef1 = 48 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, título)
End Sub
Sub estado_hocenciatura_NotInList (NewData As String, Response As Integer)
    Response = DATA_ERRCONTINUE
    msg1 = "El estado debe coincidir con un valor de la lista"
    dgdef1 = 48 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, título)
End Sub
Sub estado_posgrado_autor_NotInList (NewData As String, Response As Integer)
    Response = DATA_ERRCONTINUE
    msg1 = "El estado debe coincidir con un valor de la lista"
    dgdef1 = 48 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, título)
End Sub
Sub estado1_NotInList (NewData As String, Response As Integer)
    Response = DATA_ERRCONTINUE
    msg1 = "El estado debe coincidir con un valor de la lista"
    dgdef1 = 48 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, título)
End Sub
Sub fecha_de_nacimiento_error ()
    If Error = 2118 Then
        msg1 = "Fecha incorrecta"
        dgdef1 = 48 + 0 + 0
        RES1 = MsgBox(msg1, dgdef1, título)
        DoCmd CancelEvent
    Else
        msg1 = "Fecha incorrecta" & " " & prueba
        dgdef1 = 0 + 0 + 0
        RES1 = MsgBox(msg1, dgdef1, título)
        DoCmd CancelEvent
    End If
End Sub
Sub Form_BeforeUpdate (Cancel As Integer)
    msg1 = "¿Realmente desea dar de alta a este alumno?"
    dgdef1 = 4 + 32 + 0
    RES1 = MsgBox(msg1, dgdef1, título)
End Sub
Sub Form_Error (dataerr As Integer, Response As Integer)
    Response = DATA_ERRCONTINUE
    FECHAINCORRECTA = 2118
    If dataerr = FECHAINCORRECTA Then
        msg1 = "La fecha es incorrecta"
        dgdef1 = 0 + 48 + 0
        RES1 = MsgBox(msg1, dgdef1, título)
    End If
End Sub
Sub guardar_registro_Click ()
    msg1 = "¿Desea calbar el cambio?"
    dgdef1 = 4 + 32 + 0
    RES1 = MsgBox(msg1, dgdef1, título)
    If RES1 = 0 Then
        If IsNull(paterno) Or IsNull(materno) Or IsNull(nombre) Then
            msg1 = "Si no se apunta nombre completo del alumno, el registro no será actualizado"
            dgdef1 = 0 + 48 + 0
            RES1 = MsgBox(msg1, dgdef1, título)
        End If
        On Error GoTo Err_guardar_registro_Click
        DoCmd DoMenuItem A_FORMBAR, A_EDITMENU1, A_UNDOHELP1, A_MENU_VLR20
    End Sub
    Exit_guardar_registro_Click
    msg1 = "¿Desea modificar datos de otro alumno?"
    dgdef1 = 4 + 32 + 0
    RES1 = MsgBox(msg1, dgdef1, título)
    If RES1 = 7 Then
        DoCmd Close
        DoCmd Close A_FORM, "Lee cuenta a modificar"
    Else

```



```

DoCmd Close
DoCmd Close A_FORM, "lee cuenta a modificar"
DoCmd OpenForm "lee cuenta a modificar"
End If
Sub Sub
Frg_guardar_registro_Click
Resume Exit_guardar_registro_Click
End If
DoCmd DoMenuItem A_FORMBAR, A_MENU, A_OVERLAYORIG, A_MENU, VERZO
msg1 = "¿Quieres modificar datos de otro alumno?"
dgl%16 = 4 + 32 + 0
res1 = MsgBox(msg1, dgl%16, titulo)
If res1 = 7 Then
DoCmd Close
DoCmd Close A_FORM, "lee cuenta a modificar"
Else
DoCmd Close
DoCmd Close A_FORM, "lee cuenta a modificar"
DoCmd OpenForm "lee cuenta a modificar"
End If
Else
On Error GoTo Frg_guardar_registro_Click
DoCmd DoMenuItem A_FORMBAR, A_MENU, A_UNINDIRECT, A_MENU, VERZO
msg1 = "¿Quieres modificar datos de otro alumno?"
dgl%16 = 4 + 32 + 0
res1 = MsgBox(msg1, dgl%16, titulo)
If res1 = 7 Then
DoCmd Close
DoCmd Close A_FORM, "lee cuenta a modificar"
Else
DoCmd Close
DoCmd Close A_FORM, "lee cuenta a modificar"
DoCmd OpenForm "lee cuenta a modificar"
End If
End If
End Sub
Sub Imprimir_Click ()
End Sub
Sub Inclusion_Incencia_Notifias (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "La inclusión debe comenzar con un valor de la lista"
dgl%1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgl%1, titulo)
End Sub
Sub Inclusion_Posgrado_Notifias (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "La inclusión del posgrado con un valor de la lista"
dgl%1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgl%1, titulo)
End Sub
Sub Incum_Incumtatura_Notifias (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El país debe comenzar con un valor de la lista"
dgl%1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgl%1, titulo)
End Sub
Sub Incum_Posgrado_Inferi_Notifias (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El país debe comenzar con un valor de la lista"
dgl%1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgl%1, titulo)
End Sub
Sub Incum_Plan_dcl_nuevo_posgr_Notifias (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El plan debe comenzar con un valor de la lista"
dgl%1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgl%1, titulo)
End Sub
Sub Incum_Ric_dcl_Tutor_Notifias (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El RIC debe comenzar con un valor de la lista"
dgl%1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgl%1, titulo)
End Sub
Sub siguiente1_Click ()
msg4 = "¿Ehales conatos del Alumno correctos?"
dgl%14 = 4 + 32 + 0
res4 = MsgBox(msg4, dgl%14, titulo)
If res4 = 0 Then
DoCmd GoToControl "numero de cuenta1"
DoCmd GoToControl "lave de titulo"
Else
DoCmd GoToControl "numero de cuenta2"
End If
End Sub
Sub siguiente2_Click ()
msg4 = "¿Amuecedatos Academicos de Posgrado correctos?"
dgl%14 = 4 + 32 + 0
res4 = MsgBox(msg4, dgl%14, titulo)
If res4 = 0 Then
DoCmd GoToControl "numero de cuenta1"
DoCmd GoToControl "plan"
Else
DoCmd GoToControl "numero de cuenta2"
End If
End Sub
Sub siguiente3_Click ()
msg4 = "¿Amuecedatos Academicos de Posgrado correctos?"
dgl%14 = 4 + 32 + 0
res4 = MsgBox(msg4, dgl%14, titulo)
If res4 = 0 Then
DoCmd GoToControl "ultima"
DoCmd GoToControl "ingrese"
Else
DoCmd GoToControl "previ"
End If
End Sub
Sub Incum_de_Incum_Notifias (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE

```

```

msg1 = "El nombre de la beca debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RDS1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub tutor_MostrarInst (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El RFC debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RDS1 = MsgBox(msg1, dgdef1, titulo)
End Sub
4/11/97 10:41:09 AM
6/01/98 11:00:19 AM
Sub CancelarCambios_Click()
msg1 = "Los cambios realizados serán guardados en la base de datos"
dgdef1 = 48 + 0 + 0
RDS1 = MsgBox(msg1, dgdef1, titulo)
DoCmd Close
DoCmd Close A_FORM, "Lee cuenta para modificar plan"
msg2 = "¿Desea modificar el Plan de Estudios de otro alumno?"
dgdef2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdef2, titulo)
If res2 = 6 Then
DoCmd Close
DoCmd Close A_FORM, "Lee cuenta para modificar plan"
DoCmd OpenForm "Lee cuenta para modificar plan"
Else
DoCmd Close
DoCmd Close A_FORM, "Lee cuenta para modificar plan"
End If
End Sub
Sub term_AfterUpdate()
Dim tes As Database, mitabla As Recordset
Set tes = DBEngine.Workspaces(0).Databases(0)
Set mitabla = tes.OpenRecordset("mat_plan", DB_OPEN_TABLE)
mitabla index = "materia"
mitabla Seek "=", nueva
If mitabla.NoMatch Then
DoCmd CancelEvent
Else
If mitabla("n_cuenta") = forms![Cambiar plan de estudios]!n_cuenta Then
m2 = "El alumno ya tiene esta Materia"
f2 = 48 + 0 + 0
r2 = MsgBox(m2, f2, titulo)
DoCmd CancelEvent
End If
End Sub
Sub term_Error (DataErr As Integer, Response As Integer)
Const sinrelacion = 5101
If Err = sinrelacion Then
SendKeys "{ESC}", True
End If
End Sub
Sub NUEVA_AfterUpdate()
Dim tes As Database, mitabla As Recordset
Set tes = DBEngine.Workspaces(0).Databases(0)
Set mitabla = tes.OpenRecordset("mat_plan", DB_OPEN_TABLE)
mitabla index = "materia"
mitabla Seek "=", nueva
If mitabla.NoMatch Then
m1 = "El estudiante de esta materia al plan de estudios del alumno"
f1 = 1 + 32 + MB_DIALOGBOX_YES
r1 = MsgBox(m1, f1, titulo)
Else
If mitabla("n_cuenta") = forms![Cambiar plan de estudios]!n_cuenta Then
m2 = "El alumno ya tiene esta Materia"
f2 = 48 + 0 + 0
r2 = MsgBox(m2, f2, titulo)
DoCmd CancelEvent
End If
End Sub
Sub nueva_BeforeUpdate (Cancel As Integer)
Dim miB1 As Database, mitabla As Recordset
Set miB1 = DBEngine.Workspaces(0).Databases(0)
Set mitabla = miB1.OpenRecordset("MAT_PLAN", DB_OPEN_TABLE) ' Abrir la tabla
mitabla index = "n_cuenta" ' Establece el índice activo
mitabla Seek "=", forms![Cambiar plan de estudios]!n_cuenta
If (mitabla.NoMatch) Then
Else
Do Until mitabla.EOF
If (mitabla!MATERIA) = forms![Cambiar plan de estudios]!nueva And (mitabla!n_cuenta) = forms![Cambiar plan de estudios]!n_cuenta Then
m2 = "El alumno ya tiene esta Materia, presionar ESC para continuar !"
f2 = 48 + 0 + 0
r2 = MsgBox(m2, f2, titulo)
SendKeys "{ESC}", True
DoCmd CancelEvent
Exit Do
Else
mitabla.MoveNext
End If
Loop
End If
SendKeys "{ESC}", True
mitabla Close ' Cierra la tabla
End Sub
Sub NUEVA_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
m2 = "La clave de la materia debe coincidir con un valor de la lista"
f2 = 48 + 0 + 0
r2 = MsgBox(m2, f2, titulo)
End Sub
Sub NUMERO_BeforeUpdate (Cancel As Integer)
If (NUMERO) > 6 Or (NUMERO) < 1 Then
m2 = "Número Incorrecto"
f2 = 48 + 0 + 0
r2 = MsgBox(m2, f2, titulo)
SendKeys "{ESC}"
DoCmd CancelEvent
End Sub

```

```

End If
End Sub
Sub Actar_Click ()
    m2 = "¿Hacer Matricular el Plan de Estudios de otro alumno?"
    r2 = 4 + 32 + 0
    r2 = MsgBox(m2, vbYesNo)
    If r2 = 0 Then
        DoCmd Close
        DoCmd Close A_PORM, "Lee cuenta para modificar plan"
        DoCmd OpenForm "Lee cuenta para modificar plan"
    Else
        DoCmd Close
        DoCmd Close A_PORM, "Lee cuenta para modificar plan"
    End If
End Sub

Sub Aceptar_Click ()
    m1 = "¿Desca Iniciar el Cambiio?"
    r1 = 4 + 32 + 0
    r1 = MsgBox(m1, vbYesNo)
    If r1 = 0 Then Inicia cambio
    Dim dbs As Database, mitable As Recordset
    Set dbs = CurrentProject.Database
    Set mitable = dbs.OpenRecordset("INSCRITO", db_OPEN_TABLE)
    mitable.Index = "ACTA"
    mitable.Seek "=", mtable.Acta Nueva
    If Not (mtable.Recordset.EOF) Then
        m1 = "¿El número de Acta ya existe?"
        r1 = 48 + 0 + 0
        r1 = MsgBox(m1, vbYesNo)
        SendKeys "Esc", True
        mitable.Close
        Exit Sub
    Else
        DoCmd RunMacro "macro de acta cambiar acta"
        "ACTUAR CONSULTA DE REEMPLAZO"
        mitable.Close
    End If
Else
    Me!Cerrar.SetFocus
    Me!Acta Nueva.Visible = False
    Me!Aceptar.Visible = False
    SendKeys "Esc", True
End If
End Sub
Sub Boton41_Click ()
End Sub
Sub CALIF_AfterUpdate ()
    If Val(Me!Calif) < 4 Or Val(Me!Calif) < 0 Then
        Me!Aprobada = "NA"
    Else
        Me!Aprobada = "A"
    End If
End Sub
Sub Cambiar_acta_per_Click ()
    Me!Acta Nueva.Visible = True
    Me!Aceptar.Visible = True
End Sub
Sub Form_Open (Cancel As Integer)
    Me!Acta Nueva.Visible = False
    Me!Aceptar.Visible = False
End Sub
Sub aceptar_Click ()
    m2 = "¿Otro semestre?"
    r2 = 4 + 32 + MB_DIALOGBOX
    r2 = MsgBox(m2, vbYesNo)
    If r2 = 0 Then
        Formulario1 Alta del Plan de Estudios(Primer) ENABLER = True
        Formulario1 Alta del Plan de Estudios(Primer) SetFocus
        Formulario1 Alta del Plan de Estudios(segundo) Visible = False
        Formulario1 Alta del Plan de Estudios(segundo) ENABLER = True ' Hago visibles
        Formulario1 Alta del Plan de Estudios(tercer) ENABLER = True ' botones de semestres
        Formulario1 Alta del Plan de Estudios(cuarta) ENABLER = True
    Else
        m2 = "¿Bar de Alta Plan de Estudios para otro alumno?"
        r2 = 4 + 32 + MB_DIALOGBOX
        r2 = MsgBox(m2, vbYesNo)
        If r2 = 0 Then
            DoCmd Close
            DoCmd Close A_PORM, "Alta del plan de estudio"
            DoCmd Close A_PORM, "Lee cuenta para plan"
            DoCmd OpenForm "Lee cuenta para plan"
            DoCmd GoToControl "Aceptar"
        Else
            DoCmd Close
            DoCmd Close A_PORM, "Alta del plan de estudio"
            DoCmd Close A_PORM, "Lee cuenta para plan"
        End If
    End If
End Sub
Sub Boton41_Click ()
    m2 = "¿Bar de Alta Plan de Estudios para otro alumno?"
    r2 = 4 + 32 + MB_DIALOGBOX
    r2 = MsgBox(m2, vbYesNo)
    If r2 = 0 Then
        DoCmd Close
        DoCmd Close A_PORM, "Alta del plan de estudio"
        DoCmd Close A_PORM, "Lee cuenta para plan"
        DoCmd OpenForm "Lee cuenta para plan"
        DoCmd GoToControl "Aceptar"
    Else
        DoCmd Close
        DoCmd Close A_PORM, "Alta del plan de estudio"
        DoCmd Close A_PORM, "Lee cuenta para plan"
    End If
End Sub
Sub NUEVA_AfterUpdate ()
    Dim mtdb As Database, mitable As Recordset

```

```

Set m1(0) = DNGEngine.Workspace(0).Database(0)
Set m1tabla = m1(0).OpenRecordset("MAT_TABLA", DB_OPEN_TABLE) ' Abre la tabla.
m1tabla.index = "M_CUENTA" ' Establece el índice activo
m1tabla.Seek "=", Form!captura_sistema(1)(cucenta)
m1tabla.Seek "=", Me!(cucenta)
If (m1tabla.NoMatch) Then
    Else
Do Until m1tabla.EOF
If m1tabla(MATERIA) = Form!captura_sistema(1)(materia) And m1tabla(n_cuenta) = Form!captura_sistema(1)(cucenta) Then
If m1tabla(MATERIA) = Me!(materia) And m1tabla(n_cuenta) = Me!(cucenta) Then
m2 = "¿El alumno ya tiene esta Materia?"
f2 = 48 + 0 + MR_DEFECTO2
r2 = MsgBox(m2, f2, titulo)
DoCmd DoMenuItem A.FORMAR, A.EDITMENU, A.UNIFORME10, , A.MENU_VI20
DoCmd CancelEvent
Exit Do
Else
m1tabla.MoveNext
End If
Loop
Exit If
m1tabla.Close ' Cierra la tabla.
End Sub
Sub NUEVA_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRORCONTINUE
m2 = "La clave de la materia debe de coincidir con un valor de la lista"
f2 = 48 + 0 + MR_DEFECTO2
r2 = MsgBox(m2, f2, titulo)
End Sub

Sub aceptar_Click ()
If IsNull(leeclave) Then
msg4 = "Clave Incorrecta!"
dgDef4 = 48 + 0 + 0
RES4 = MsgBox(msg4, dgDef4, titulo)
If RES4 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "leeclave"
End If
Else
Dim tes As Database, m1tabla As Records1
Set tes = dngengine.workspace(0).database(0)
Set m1tabla = tes.OpenRecordset("materia", DB_OPEN_TABLE)
m1tabla.index = "clave"
m1tabla.Seek "=", leeclave
If m1tabla.NoMatch Then 'si NO esta manda un mensaj
msg3 = "La Materia no existe!"
dgDef3 = 48 + 0 + 0
RES3 = MsgBox(msg3, dgDef3, titulo)
If RES3 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "leeclave"
End If
Else
DoCmd OpenForm "Consulta Materia especifica", , "tbl_mate", Formulario!(leeclave), , "materia a consultar"(leeclave)
DoCmd Close _a_form, "Clave de materia a consultar"
End If
m1tabla.Close
End If
End Sub
Sub sub_aceptar_Click ()
DoCmd Close
End Sub
Sub Form_Error (DataErr As Integer, Response As Integer)
Response = DATA_ERRORCONTINUE
Const ERRORMASCARA = 2273
If (DataErr = ERRORMASCARA) Then
m1 = "Clave Incorrecta!"
f1 = 48 + 0 + MR_DEFECTO1
r1 = MsgBox(m1, f1, titulo)
If r1 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "leeclave"
End If
End If
End Sub
Sub leeclave_AfterUpdate ()
If IsNull(leeclave) Then
msg4 = "Clave Incorrecta!"
dgDef4 = 48 + 0 + 0
RES4 = MsgBox(msg4, dgDef4, titulo)
If RES4 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "leeclave"
End If
Else
Dim tes As Database, m1tabla As Records1
Set tes = dngengine.workspace(0).database(0)
Set m1tabla = tes.OpenRecordset("materia", DB_OPEN_TABLE)
m1tabla.index = "clave"
m1tabla.Seek "=", leeclave
If m1tabla.NoMatch Then 'si NO esta manda un mensaje
msg3 = "La Materia no existe!"
dgDef3 = 48 + 0 + 0
RES3 = MsgBox(msg3, dgDef3, titulo)
If RES3 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "leeclave"
End If
Else
DoCmd OpenForm "Consulta Materia especifica", , "tbl_mate", Formulario!(leeclave), , "materia a consultar"(leeclave)
DoCmd Close _a_form, "Clave de materia a consultar"
End If
m1tabla.Close

```

```

End If
End Sub

Sub Acceptar_Click ()
m3 = "¿Se cancela el Cambio?"
f1 = 4 + 32 + 0
r1 = MsgBox(m3, f1, titulo)
if r1 = 0 Then 'hace cambio
Dim lrs As Database, mitable As Recordset
Set lrs = db.open(workspaces(0).database(0))
Set mitable = lrs.OpenRecordset("INSCRITO", lrs.OPEN_TABLE)
mitable.index = "ACTA"
mitable.Seek "=", Mellacta.nueva
If Not (mitable.NoMatch) Then
n1 = "El número de Acta ya existe!"
f1 = 48 + 0 + 0
c1 = MsgBox(m1, f1, titulo)
SendKeys "{esc}", True
mitable.Close
Exit Sub
Else
'DoCmd RunMacro "Consulta para cambio de acta"
'EXECUTAR CONSULTA DE REEMPLAZO
mitable.Close
End If
Else
MellcerrarJSelfbus
Mellacta.nueva.Visible = False
Mellacceptar.Visible = False
SendKeys "{esc}", True
End If
End Sub
Sub Botón41_Click ()
End Sub
Sub CALIF_ActaUpdate ()
if Val(Mellcalif) < 0 Then
Mellaprobada1 = "N"
Else
Mellaprobada1 = "A"
End If
End Sub
Sub cancelar_acta_pot_41_Click ()
Mellacta.nueva.Visible = True
Mellacceptar.Visible = True
End Sub
Sub Imprimir_Click ()
On Error GoTo Err_Imprimir_Click
Dim DocName As String
DocName = "actas"
'DoCmd OpenReport "DocName, A_normal, "[acta]=formularios[Consulta de acta exp.cifun][acta]"
Exit_Imprimir_Click
Err_Imprimir_Click:
m3 = "¿Se cancela impresión?"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, f3, titulo)
Resume Exit_Imprimir_Click
End Sub
Sub Imprimir_Click ()
On Error GoTo Err_Imprimir_Click
Dim DocName As String
DocName = "actas generales"
'DoCmd OpenReport "DocName, A_preview"
Exit_Imprimir_Click
Err_Imprimir_Click:
m3 = "¿Se cancela impresión?"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, f3, titulo)
Resume Exit_Imprimir_Click
End Sub
Sub Botón29_Click ()
On Error GoTo Err_botón29_Click
'DoCmd DoMenuItem A_FORMIBAR, A_IDIOMAS, IO, A_MFNU_VER20
Exit_botón29_Click
Err_botón29_Click:
MsgBox Error$
Resume Exit_botón29_Click
End Sub
Sub Botón28_Click ()
End Sub
Sub Botón29_Click ()
On Error GoTo Err_botón29_Click
'DoCmd DoMenuItem A_FORMIBAR, A_IDIOMAS, A_SAVE RECORDS, A_MENU_VER20
Exit_botón29_Click
Err_botón29_Click:
MsgBox Error$
Resume Exit_botón29_Click
End Sub
Sub Botón30_Click ()
On Error GoTo Err_botón30_Click
'DoCmd DoMenuItem A_FORMIBAR, A_IDIOMAS, A_IDIOMAS, A_IDIOMAS, A_MFNU_VER20
Exit_botón30_Click
Err_botón30_Click:
MsgBox Error$
Resume Exit_botón30_Click
End Sub
Sub Imprimir_Click ()
On Error GoTo Err_Imprimir_Click
'DoCmd OpenReport "Lista de matriculas, A_normal"
Exit_Imprimir_Click
Err_Imprimir_Click:
m3 = "¿Se cancela impresión?"
f3 = 48 + 0 + 0

```

```

r3 = MsgBox(m3, f3, titulo)
Resume Err_imprimir_Click
End Sub

Sub cancelar_consulta_pla_Click ()
msg1 = "¿Desea cancelar la consulta?"
dglNf1 = 4 + 32 + 0
res1 = MsgBox(msg1, dglNf1, titulo)
If res1 = 0 Then
    msg2 = "¿Desea consultar el Plan de Estudios de otro Alumno?"
    dglNf2 = 4 + 32 + 0
    res2 = MsgBox(msg2, dglNf2, titulo)
    If res2 = 7 Then 'si res2 es NO
        DoCmd Close
    DoCmd Close A_FORM, "lee cuenta para consultar plan"
    Else
        DoCmd Close
    DoCmd Close A_FORM, "lee cuenta para consultar plan"
    DoCmd Openform "lee cuenta para consultar plan"
    End If
End If
End Sub

Sub imprimir_Click ()
On Error GoTo Err_imprimir_Click
DoCmd Openform "plan de estudios del alumno", A_normal, , "In_cuenta" & formularios\consulta de plan de estudios\In_cuenta"
Err_imprimir_Click:
Exit Sub
Err_imprimir_Click:
m3 = "¿Se canceló impresión?"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, f3, titulo)
Resume Err_imprimir_Click
End Sub

Sub salir_Click ()
msg2 = "¿Desea consultar el Plan de Estudios de otro Alumno?"
dglNf2 = 4 + 32 + 0
res2 = MsgBox(msg2, dglNf2, titulo)
If res2 = 7 Then 'si res2 es NO
    DoCmd Close
DoCmd Close A_FORM, "lee cuenta para consultar plan"
Else
    DoCmd Close
DoCmd Close A_FORM, "lee cuenta para consultar plan"
DoCmd Openform "lee cuenta para consultar plan"
End If
End Sub

Sub cancelar_consulta_pla_Click ()
msg1 = "¿Desea cancelar la consulta?"
dglNf1 = 4 + 32 + 0
res1 = MsgBox(msg1, dglNf1, titulo)
If res1 = 0 Then
    msg2 = "¿Desea consultar el Plan de Estudios de otro Alumno?"
    dglNf2 = 4 + 32 + 0
    res2 = MsgBox(msg2, dglNf2, titulo)
    If res2 = 7 Then 'si res2 es NO
        DoCmd Close
    DoCmd Close A_FORM, "lee cuenta para consultar plan"
    Else
        DoCmd Close
    DoCmd Close A_FORM, "lee cuenta para consultar plan"
    DoCmd Openform "lee cuenta para consultar plan"
    End If
End If
End Sub

Sub Salir_Click ()
DoCmd Close
End Sub

Sub Form_Load ()
Me!Subformulario para consultar grupos visible = False
End Sub

Sub lista_de_semestres_Abrir_Click ()
Dim miDBO As Database, miTabla0 As Recordset
Set miDBO = DBEngine.Workspaces(0).Database(0)
Set miTabla0 = miDBO.OpenRecordset("class", DB_OPEN_TABLE)
miTabla0.Index = "semestre" ' establece el índice activo
miTabla0.Seek "=", acFirst
If miTabla0.RecordCount > 0 Then
    reemplazar\consulta historica de clases\Subformulario para consultar grupos visible = True
    m2 = "¿No hay grupos para el semestre seleccionado?"
    f2 = 48 + 0 + 0 + MB_YESNO
    r2 = MsgBox(m2, f2, titulo)
    DoCmd CancelDoCmd
End Sub
Else
DoCmd RunMacro "rellenar busca semestra"
End If
miTabla0.Close
End Sub

Sub lista_de_semestres_NoInList (NewData As String, Response As Integer)
Response = DATA, ERRCONTINUE
msg1 = "El Semestre debe coincidir con un valor de la lista"
dglNf1 = 48 + 0 + 0
RfN1 = MsgBox(msg1, dglNf1, titulo)
DoCmd CancelDoCmd
End Sub

Sub Abrir_Semestre_Click ()
Me!Subformulario para consultar grupos visible = False
DoCmd GoToControl "lista de semestres"
End Sub

Sub consultar_a_ave_Click ()
DoCmd Openform "Consulta de Materias"
End Sub
Sub consultar_estu_mater_Click ()
DoCmd Close a_form, "consulta materia específica"
DoCmd Openform "ave de materia a consultar"

```

```

End Sub
Sub Form_BeforeUpdate (Cancel As Integer)
If IsNull(mom_mat) Then
msg1 = "El campo Nombre de Materia Vacío!"
dgdef1 = 41 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, Título)
DoCmd CancelEvent
DoCmd GoToControl "nombre_materia"
End If
End Sub
Sub Form_Error (DefErr As Integer, Response As Integer)
Const VALORSNULO = 2107
Const ERRORESEMETRE = 2501
If Err = VALORSNULO Then
msg1 = "El campo Nombre de Materia Vacío!"
dgdef1 = 46 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, Título)
DoCmd CancelEvent
DoCmd GoToControl "nombre_materia"
End If
If Err = ERRORESEMETRE Then
msg1 = "Elegir primero semestre!"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, Título)
DoCmd CancelEvent
End If
End Sub
Sub lista_de_materias_BeforeUpdate (Cancel As Integer)
Me!clave de la materia visible = True
Me!numero de clase visible = True
Me!etiqueta clave visible = True
Me!etiqueta nombre materia visible = True
Me!numero de la materia visible = True
Me!nombre nombre visible = True
Me!etiqueta credits visible = True
Me!CREDITOS visible = True
Me!numero credits visible = True
End Sub
Sub lista_de_materias_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El nombre de la materia debe coincidir con la lista!"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, Título)
End Sub
Sub mat_nueva_BeforeUpdate (Cancel As Integer)
If Len(mat_nueva) < 6 Then
msg1 = "La clave debe ser de 6 dígitos!"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, Título)
DoCmd CancelEvent
End If
End Sub
Sub MATERIA_AfterUpdate ()
DoCmd RunMacro "Relevar nombres"
End Sub
Sub nombre_de_la_materia_AfterUpdate ()
If Me!NOM_MAT = "" Then
msg1 = "El campo Nombre de Materia Vacío!"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, Título)
SendKeys "{esc}", True
End If
End Sub
Sub nombre_materia_BeforeUpdate (Cancel As Integer)
If IsNull(mom_mat) Then
msg1 = "El campo Nombre de Materia Vacío!"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, Título)
DoCmd CancelEvent
DoCmd GoToControl "nombre_materia"
End If
End Sub
Sub Salir_Click ()
DoCmd Close
End Sub
Sub Correr_Click ()
msg1 = "Dar de alta Otra Clase?"
dgdef1 = 4 + 32 + 0
res = MsgBox(msg1, dgdef1, título)
If res = 0 Then
DoCmd GoToRecord , , A_NEWREC
Else
DoCmd Close
End If
End Sub
Sub CLAVE_DE_MATERIA_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
Loop
msg1 = "La Clave de la Materia debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, título)
End Sub
Sub Dar_de_alta_clase_Click ()
If IsNull(Me!GRUPO) Or IsNull(Me!gruppo) Or IsNull(Me!CLAVE_DE_MATERIA) Or IsNull(Me!salon) Then
msg1 = "Alguno de los campos (Profesor, Materia, Grupo, salón) está vacío!"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, título)
Exit Sub
End If
msg2 = "¿Datos correctos?"
dgdef2 = 4 + 32 + 0
RES2 = MsgBox(msg2, dgdef2, título)
If Me!dia1 <> -1 Or IsNull(hora1) And Me!dia1 = -1 Or Me!dia2 <> -1 Or IsNull(hora2) And Me!dia2 = -1 Or Me!dia3 <> -1 Or IsNull(hora3) And Me!dia3 = -1 Or Me!dia4 <> -1 Or IsNull(hora4) And Me!dia4 = -1 Or Me!dia5 <> -1 Or IsNull(hora5) And Me!dia5 = -1 Or Me!dia6 <> -1 Or IsNull(hora6) And Me!dia6
= -1) Then
msg = "¿Debe Marcar el día y hora de clase?"
r2 = 48 + 0 + MR_DEBTOTONZ
r2 = MsgBox(m2, r2, título)

```

```

End Sub
End If
If RES2 = 0 Then
    msg3 = "¿Por de alta otra Clase?"
    dgdef3 = 4 + 32 + 0
    RES3 = MsgBox(msg3, dgdef3, título)
    If RES3 = 1 Then
        DoCmd DoMenuItem A_FORMAR, A_FILE, A_SAVRECORD, , A_MENU_VERZO
        msg4 = "¿Por de alta Otra Clase?"
        dgdef4 = 4 + 32 + 0
        res4 = MsgBox(msg4, dgdef4, título)
        If res4 = 0 Then
            DoCmd GoToRecord , , A_NEWREC
        Else
            DoCmd Close
        End If
    Else
        DoMenuItem A_FORMAR, A_EDITMENU, A_UNIONHELI, , A_MENU_VERZO
    End If
Else
    DoCmd GoToControl "RFC PROFESOR"
End If
End Sub
Sub estado1_Notifia1 (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "La clave de la Materia debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, título)
End Sub
Sub Form_Error (Database As Integer, Response As Integer)
Response = DATA_ERRCONTINUE
Const NOHAYNINGUN = 5101
CONST L1TIEMPODE:
If Database = NOHAYNINGUN Then
    m2 = "Existe algún campo vacío? Promonar ESC para continuar."
    f2 = 48 + 0 + MI_ICONALTO + MI_UIBOTON2
    r2 = MsgBox(m2, f2, título)
    DoCmd CancelEvent
    SendKeys "{ESC}", True
End If
If Database = NOHAYNINGUN Then
    m2 = "Existe algún campo vacío? Promonar ESC para continuar."
    f2 = 48 + 0 + MI_ICONALTO + MI_UIBOTON2
    r2 = MsgBox(m2, f2, título)
    DoCmd CancelEvent
    SendKeys "{ESC}", True
End If
End Sub
Sub Form_Open (Cancel As Integer)
DoCmd ApplyFilter "Filtro de semestre"
End Sub
Sub REGISTRO_ANTERIOR_Click ()
On Error GoTo Err_REGISTRO_ANTERIOR_Click
DoCmd GoToRecord , , A_PREVIOUS
Exit_REGISTRO_ANTERIOR_Click
End Sub
Err_REGISTRO_ANTERIOR_Click
msg1 = "¿Este es el primer registro?"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, título)
Resume Exit_REGISTRO_ANTERIOR_Click
End Sub
Sub RFC_PROFESOR_Notifia1 (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El RFC debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, título)
End Sub
Sub Salir_Click ()
m2 = "¿Esco realizar otra consulta?"
f2 = 32 + 4 + 0
r2 = MsgBox(m2, f2, título)
If r2 = 0 Then
    DoCmd Close
    DoCmd OpenForm "Consultar Clase por Materia"
    DoCmd GoToControl "materia a buscar"
Else
    DoCmd Close
    DoCmd Close a_form, "Consultar Clase por Materia"
End If
End Sub
Sub NADON_Notifia1 (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El salon debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, título)
End Sub
Sub Siguiente_Registro_Click ()
On Error GoTo Err_Siguiente_Registro_Click
DoCmd GoToRecord , , A_NEXT
Exit_Siguiente_Registro_Click
End Sub
Err_Siguiente_Registro_Click
msg1 = "¿Este es el último registro?"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, título)
Resume Exit_Siguiente_Registro_Click
End Sub
Sub lista_de_nombres_Afieri (data1)
If IsNull(data1) Then
    m2 = "La Clave de la Materia es incorrecta!"
    f2 = 0 + 48 + 0
    r2 = MsgBox(m2, f2, título)
    DoCmd CancelEvent
    Exit Sub
End If
Dim m1(1) As Database, m1(1) As Recordset

```



```

Set mIRDO = DBEngine.Workspace(0) Database(0)
Set mtabla1 = mIRDO1.OpenRecordset("MATERIA", DB_OPEN_TABLE)
mtabla1.index = "CLAVE" ' Establece el índice activo
mtabla1.Seek "=", Me!lista de nombres
If (mtabla1.nomatch) Then
    m2 = "¡La materia no existe!"
    f2 = 48 + 0 + MB_DEFBOTÓN2
    r2 = MsgBox(m2, f2, titulo)
    DoCmd CancelEvent
    Exit Sub
End If
Dim mIRDO As Database, mtabla0 As Records1
Set mIRDO = DBEngine.Workspace(0) Database(0)
Set mtabla0 = mIRDO.OpenRecordset("clase", DB_OPEN_TABLE)
mtabla0.index = "semestre" ' Establece el índice activo
mtabla0.Seek "=", Formulario0[Semestre Actual][Semestre]
If (mtabla0.nomatch) Then
    m2 = "¡No hay grupos para el semestre actual!"
    f2 = 48 + 0 + MB_DEFBOTÓN2
    r2 = MsgBox(m2, f2, titulo)
    DoCmd CancelEvent
    Exit Sub
End If
mtabla1.Close
Dim mIR0 As Database, mtabla As Recordset
Set mIR0 = DBEngine.Workspace(0) Database(0)
Set mtabla = mIR0.OpenRecordset("clase", DB_OPEN_TABLE)
mtabla.index = "materia" ' Establece el índice activo
mtabla.Seek "=", Me!lista de nombres
If Not (mtabla.nomatch) Then
    Do Until mtabla.eof
        If (mtabla[clv_mai] = Me!lista de nombres And mtabla[Semestre] = Formulario0[Semestre Actual][Semestre]) Then
            GoTo abrrz2
        Else
            mtabla.MoveNext
        End If
    Loop
abrrz2
DoCmd OpenForm "para consultar grupos"
DoCmd Close a_form, "Consultar Clase por grupo"
Exit Sub
Else
    m2 = "¡No existen clases para esa materia!"
    f2 = 48 + 0 + MB_DEFBOTÓN2
    r2 = MsgBox(m2, f2, titulo)
    DoCmd CancelEvent
    Exit Sub
End If
End Sub
Sub lista_de_nombres_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
Const titulo = "MCA"
msg1 = "¡La Materia debe ocurrir con un valor de la lista!"
dgd1 = 48 + 0 + 0
RLX1 = MsgBox(msg1, dgd1, titulo)
DoCmd CancelEvent
End Sub
Sub acc_ptar_Click ()
If IsNull(Me!materia a buscar) Then
    m2 = "¡La Clave de la Materia es incorrecta!"
    f2 = 48 + 0 + 0
    r2 = MsgBox(m2, f2, titulo)
    DoCmd CancelEvent
    Exit Sub
End If
Dim mIRDO1 As Database, mtabla1 As Recordset
Set mIRDO1 = DBEngine.Workspace(0) Database(0)
Set mtabla1 = mIRDO1.OpenRecordset("MATERIA", DB_OPEN_TABLE)
mtabla1.index = "CLAVE" ' Establece el índice activo.
mtabla1.Seek "=", Me!materia a buscar
If (mtabla1.nomatch) Then
    m2 = "¡La materia no existe!"
    f2 = 48 + 0 + MB_DEFBOTÓN2
    r2 = MsgBox(m2, f2, titulo)
    DoCmd CancelEvent
    Exit Sub
End If
Dim mIRDO As Database, mtabla0 As Records1
Set mIRDO = DBEngine.Workspace(0) Database(0)
Set mtabla0 = mIRDO.OpenRecordset("clase", DB_OPEN_TABLE)
mtabla0.index = "semestre" ' Establece el índice activo
mtabla0.Seek "=", Formulario0[Semestre Actual][Semestre]
If (mtabla0.nomatch) Then
    m2 = "¡No hay grupos para el semestre actual!"
    f2 = 48 + 0 + MB_DEFBOTÓN2
    r2 = MsgBox(m2, f2, titulo)
    DoCmd CancelEvent
    Exit Sub
End If
mtabla1.Close
Dim mIR0 As Database, mtabla As Recordset
Set mIR0 = DBEngine.Workspace(0) Database(0)
Set mtabla = mIR0.OpenRecordset("clase", DB_OPEN_TABLE)
mtabla.index = "materia" ' Establece el índice activo
mtabla.Seek "=", Me!materia a buscar
If Not (mtabla.nomatch) Then
    Do Until mtabla.eof
        If (mtabla[clv_mai] = Me!materia a buscar And mtabla[Semestre] = Formulario0[Semestre Actual][Semestre]) Then
            GoTo abrrz2
        Else
            mtabla.MoveNext
        End If
    Loop
abrrz2
m2 = "¡No hay grupos para el semestre actual!"

```

```

f2 = 48 + 0 + MB_DIFEROTONZ
r2 = MsgBox(m2, f2, título)
DoCmd CancelEvent
Exit Sub

'abrirZ:
DoCmd OpenForm "consultar clase"
Exit Sub

f2 = "¿No existen clases para esa materia?"
f2 = 48 + 0 + MB_DIFEROTONZ
r2 = MsgBox(m2, f2, título)
DoCmd CancelEvent
End If
End Sub

Sub CLAVE_DE_MATERIA_NotInList (NewData As String, Response As Integer)
Response = DATA_CONTINUE
If Not IsNull(materia) Then
    MsgBox "La Clave de la Materia debe coincidir con un valor de la lista"
    r2 = 48 + 0 + 0
    r2 = MsgBox(m2, f2, título)
End Sub

Sub estado_NotInList (NewData As String, Response As Integer)
Response = DATA_CONTINUE
If Not IsNull(materia) Then
    MsgBox "La Clave de la Materia debe coincidir con un valor de la lista"
    r2 = 48 + 0 + 0
    r2 = MsgBox(m2, f2, título)
End Sub

Sub Form_Load ()
Me.Filter = ""
End Sub

Sub CRUPO_Refresh (Cancel As Integer)
If IsNull(grupos) < 4 Then
    If IsNull(grupos) Then
        m2 = "El número de grupos debe ser de 4 dígitos"
        f2 = 48 + 0 + MB_DIFEROTONZ
        r2 = MsgBox(m2, f2, título)
        SendKeys "{esc}", True
        DoCmd CancelEvent
    End If
    Dim mIH1 As Database, mItabla As Recordset
    Set mIH1 = DSEngine.Workspaces(0).Database(0)
    Set mItabla = mIH1.OpenRecordset("CLASE", DB_OPEN_TABLE) 'Abre la tabla
    mItabla.Index = "materia" 'Establece el índice activo
    mItabla.Seek "=", Me!CLAVE_DE_MATERIA
    If Not (mItabla.NoMatch) Then
        Do Until mItabla.EOF
            If mItabla!clv_mai = Me!CLAVE_DE_MATERIA And mItabla!numrol = Me!grupol And mItabla!semestre = Me!semestre Then
                m2 = "El grupo ya existe"
                f2 = 48 + 0 + MB_DIFEROTONZ
                r2 = MsgBox(m2, f2, título)
                SendKeys "{esc}", False
                DoCmd CancelEvent
                Exit Do
            Else
                mItabla.MoveNext
            End If
        Loop
    End If
    mItabla.Close 'Cierra la tabla
End Sub

Sub materia_a_buscar_AfterUpdate ()
If IsNull(Me!materia a buscar) Then
    m2 = "La Clave de la Materia es incorrecta"
    f2 = 48 + 0 + 0
    r2 = MsgBox(m2, f2, título)
    DoCmd CancelEvent
    Exit Sub
End If
Dim mIH1 As Database, mItabla As Recordset
Set mIH1 = DSEngine.Workspaces(0).Database(0)
Set mItabla = mIH1.OpenRecordset("MATERIA", DB_OPEN_TABLE)
mItabla.Index = "CLAVE" 'Establece el índice activo
mItabla.Seek "=", Me!materia a buscar
If (mItabla.NoMatch) Then
    m2 = "La materia no existe"
    f2 = 48 + 0 + MB_DIFEROTONZ
    r2 = MsgBox(m2, f2, título)
    DoCmd CancelEvent
    Exit Sub
End If
Dim mIH0 As Database, mItabla As Recordset
Set mIH0 = DSEngine.Workspaces(0).Database(0)
Set mItabla = mIH0.OpenRecordset("CLASE", DB_OPEN_TABLE)
mItabla.Index = "semestre" 'Establece el índice activo
mItabla.Seek "=", Formulario!Semestre Actual||Semestre
If (mItabla.NoMatch) Then
    m2 = "No hay grupos para el semestre actual"
    f2 = 48 + 0 + MB_DIFEROTONZ
    r2 = MsgBox(m2, f2, título)
    DoCmd CancelEvent
    Exit Sub
End If
mItabla.Close
Dim mIH1 As Database, mItabla As Recordset
Set mIH1 = DSEngine.Workspaces(0).Database(0)
Set mItabla = mIH1.OpenRecordset("CLASE", DB_OPEN_TABLE)
mItabla.Index = "materia" 'Establece el índice activo.
mItabla.Seek "=", Me!materia a buscar
If Not (mItabla.NoMatch) Then
    Do Until mItabla.EOF
        If (mItabla!clv_mai = Me!materia a buscar) And mItabla!semestre = Formulario!Semestre Actual||Semestre Then
            GoTo abrir
        Else
            mItabla.MoveNext
        End If
    Loop
    m2 = "No hay grupos para el semestre actual"
    f2 = 48 + 0 + MB_DIFEROTONZ
    r2 = MsgBox(m2, f2, título)

```

```

DoCmd CancelEvent
Exit Sub
abrir1:
DoCmd OpenForm "consultar clase"
Exit Sub
Else
m2 = "No existen clases para ese material"
f2 = 48 + 0 + MB_DEFBUTTON2
r2 = MsgBox(m2, f2, titulo)
DoCmd CancelEvent
Exit If
End Sub
Sub nombre_de_la_materia_NoInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
Loop
msg1 = "El nombre de la Materia debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub otra_clase_Click ()
DoCmd GoToControl "Materia a buscar"
End Sub
Sub RFC_PROFESOR_NoInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
Loop
msg1 = "El RFC debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub SALON_NoInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El salón debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub cerrar_Click ()
msg4 = "¿Consultar comprobante de inscripción para otro alumno?"
dgDef4 = 4 + 32 + 0
RES4 = MsgBox(msg4, dgDef4, titulo)
If RES4 = 6 Then
DoCmd Close
DoCmd Close #.form, "Lee cuenta o consultar tira"
DoCmd OpenForm "Lee cuenta o consultar tira"
Else
DoCmd Close
DoCmd Close #.form, "Lee cuenta o consultar tira"
End If
End Sub
Sub imprimir_Click ()
On Error GoTo Err_imprimir_Click
DoCmd OpenReport "tira de materias", A_preview, "In_cuenta" #formularios\Consultar comprobante de Ins. rpección\In_cuenta"
Exit Sub
Err_imprimir_Click
Exit Sub
Err_imprimir_Click
m3 = "¿Se canceló impresión?"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, f3, titulo)
Resume Err_imprimir_Click
End Sub
Sub Consultar_otro_Prof_Click ()
Me!Etiqueta RFC1.Visible = False
Me!Etiqueta Paterno1.Visible = False
Me!Etiqueta mat.rno1.Visible = False
Me!Etiqueta Profesor1.Visible = False
Me!Etiqueta campo1.Visible = False
Me!PATERNO1.Visible = False
Me!matrno1.Visible = False
Me!nombre1.Visible = False
Me!HISTORIA1.Visible = False
Me!sombra paterno1.Visible = False
Me!sombra historial1.Visible = False
Me!sombra materno1.Visible = False
Me!sombra nombre1.Visible = False
Me!Etiqueta RFC1.Visible = False
Me!sombra RFC1.Visible = False
Me!RFC1.Visible = False
End Sub
Sub Consultar_Profesores_Click ()
DoCmd OpenForm "Consulta de Profesores"
End Sub
Sub Dar_de_Alta_Click ()
If IsNull(rfc) Then
msg0 = "El RFC es incorrecto"
dgdef0 = 48 + 0 + 0
RES0 = MsgBox(msg0, dgdef0, titulo)
DoCmd GoToControl "rfc"
Exit Sub
End If
Dim mIBD As Database, mItabla As Recordset
Set mIBD = DBEngine.Workspaces(0).Database(0)
Set mItabla = mIBD.OpenRecordset("PROFESOR", DB_OPEN_TABLE)
mItabla.Index = "INDEX" 'Establece el índice activo
mItabla.Seek "=", Formulario8.de profesor\RFC
If (mItabla.NoMatch) Then
msg1 = "¿Desea dar de alta al Profesor?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 6 Then
If IsNull(PATERNO) Then
msg1 = "El campo Apellido Paterno del Profesor Vacío"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
DoCmd GoToControl "paterno"
Exit Sub
End If
If IsNull(MATERNO) Then

```

```

msg1 = "El campo Apellido Materno del Profesor Vacío"
dgdef1 = 48 + 0 + 0
RFE1 = MsgBox(msg1, dgdef1, titulo)
DoCmd GoToControl "Materno"
Exit Sub
End If
If IsNull(mom_prof) Then
msg1 = "El campo Nombre del Profesor Vacío"
dgdef1 = 48 + 0 + 0
RFE1 = MsgBox(msg1, dgdef1, titulo)
DoCmd GoToControl "nombre"
Exit Sub
End If
DoCmd DoMenuItem A_KORBAR, A_HLE, A_SAVERE, ORI, , A_MENU_VERZO
msg2 = "¿Jir de alta otro Profesor?"
dgdef2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdef2, titulo)
If res2 = 7 Then
DoCmd Close
Else
DoCmd GoToControl a_form, "Alias de Profesor", A_NEWREC
DoCmd GoToControl "RFC"
End If
Else
DoCmd DoMenuItem A_KORBAR, A_EHTAMU, A_USINWELI, , A_MENU_VERZO
msg2 = "¿Jir de alta otro Profesor?"
dgdef2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdef2, titulo)
If res2 = 7 Then
DoCmd Close
Else
DoCmd GoToControl a_form, "Alias de Profesor", A_NEWREC
DoCmd GoToControl "RFC"
End If
End If
Else
m2 = "¿Existe un Profesor con el mismo RFC?"
f2 = 48 + 0 + MR_HETOTONZ
r2 = MsgBox(m2, f2, titulo)
DoCmd DoMenuItem A_KORBAR, A_EHTAMU, A_USINWELI, , A_MENU_VERZO
DoCmd CancelEvent
msg2 = "¿Jir de alta otro Profesor?"
dgdef2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdef2, titulo)
If res2 = 7 Then
DoCmd Close
Else
DoCmd GoToControl a_form, "Alias de Profesor", A_NEWREC
DoCmd GoToControl "RFC"
End If
End If
End Sub
Sub form_Open (Cancel As Integer)
Me!etiqueta RFC.Visible = False
Me!etiqueta Paterno.Visible = False
Me!etiqueta maternal.Visible = True
Me!etiqueta Profesor.Visible = False
Me!etiqueta campo.Visible = False
Me!PATERNO.Visible = False
Me!materno.Visible = False
Me!nombr1.Visible = False
Me!HISTORIA.Visible = False
Me!sombra paterno.Visible = False
Me!sombra historial.Visible = False
Me!sombra maternal.Visible = False
Me!sombra nombr1.Visible = False
Me!etiqueta RFC.Visible = False
Me!sombra RFC.Visible = False
Me!RFC.Visible = False
End Sub
Sub lista_de_profesores_AfterUpdate ()
DoCmd RunMacro "Wolkenar profesor"
End Sub
Sub lista_de_profesores_detalle (Cancel As Integer)
Me!etiqueta RFC.Visible = True
Me!etiqueta Paterno.Visible = True
Me!etiqueta maternal.Visible = True
Me!etiqueta Profesor.Visible = True
Me!etiqueta campo.Visible = True
Me!PATERNO.Visible = True
Me!materno.Visible = True
Me!nombr1.Visible = True
Me!HISTORIA.Visible = True
Me!sombra paterno.Visible = True
Me!sombra historial.Visible = True
Me!sombra maternal.Visible = True
Me!sombra nombr1.Visible = True
Me!etiqueta RFC.Visible = True
Me!sombra RFC.Visible = True
Me!RFC.Visible = True
End Sub
Sub lista_de_profesores_Nuevas (NewData As String, R_Quitar As Integer)
Response = DATA_DIALOG("NUEVA")
msg1 = "El RFC del Profesor debe comenzar con un valor de 4 hasta"
dgdef1 = 48 + 0 + 0
RFE1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub salir_Click ()
msg2 = "¿Consultar otro Profesor?"
dgdef2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdef2, titulo)
If res2 = 7 Then
DoCmd Close
Else
Me!etiqueta RFC.Visible = False
Me!etiqueta Paterno.Visible = False
Me!etiqueta maternal.Visible = False
Me!etiqueta Profesor.Visible = False
Me!etiqueta campo.Visible = False

```

```

Me[[FATERNO].visible = False
Me[[materno].visible = False
Me[[nombre].visible = False
Me[[HISTORIA].visible = False
Me[[nombre_paterno].visible = False
Me[[nombre_materno].visible = False
Me[[nombre_materno].visible = False
Me[[etiqueta RFC].visible = False
Me[[nombre RFC].visible = False
Me[[RFC].visible = False
End If
End Sub

Sub Botón044_Click ()
On Error GoTo Err_Botón044_Click
msg1 = "Realmente desea deshacer el registro?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 0 Then
DoCmd DoMenuItem A_NORMALAR, A_EDITMENIU, A_UNIDONDEJA, , A_MENU_VERZO
DoCmd Close
Exit_Botón044_Click:
Exit Sub
Err_Botón044_Click:
Resume Exit_Botón044_Click
Else
Exit Sub
End If
End Sub

Sub Botón710_Click ()
On Error GoTo Err_Botón710_Click
msg1 = "Realmente desea deshacer el registro?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 0 Then
DoCmd DoMenuItem A_NORMALAR, A_EDITMENIU, A_UNIDONDEJA, , A_MENU_VERZO
DoCmd Close
Exit_Botón710_Click:
Exit Sub
Err_Botón710_Click:
Resume Exit_Botón710_Click
Else
Exit Sub
End If
End Sub

Sub Botón727_Click ()
On Error GoTo Err_Botón727_Click
msg1 = "Realmente desea deshacer el registro?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 0 Then
DoCmd DoMenuItem A_NORMALAR, A_EDITMENIU, A_UNIDONDEJA, , A_MENU_VERZO
DoCmd Close
Exit_Botón727_Click:
Exit Sub
Err_Botón727_Click:
Resume Exit_Botón727_Click
Else
Exit Sub
End If
End Sub

Sub Botón811_Click ()
msg1 = "Realmente desea cancelar alta?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 0 Then
DoCmd Close
End If
End Sub

Sub Botón813_Click ()
On Error GoTo Err_Botón813_Click
msg1 = "Realmente desea deshacer el registro?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 0 Then
DoCmd DoMenuItem A_NORMALAR, A_EDITMENIU, A_UNIDONDEJA, , A_MENU_VERZO
DoCmd Close
Exit_Botón813_Click:
Exit Sub
Err_Botón813_Click:
Resume Exit_Botón813_Click
Else
Exit Sub
End If
End Sub

Sub Botón821_Click ()
msg1 = "Realmente desea cancelar alta del alumno?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 0 Then
SendKeys "(ESC)", True
DoCmd Close
DoCmd Close a_form, "lee cuanta a inscribir"
Exit Sub
End If
End Sub

Sub Botón823_Click ()
msg1 = "Realmente desea cancelar alta del alumno?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 0 Then
SendKeys "(ESC)", True
DoCmd Close
DoCmd Close a_form, "lee cuanta a inscribir"
End If
End Sub

Sub Botón825_Click ()
msg1 = "Realmente desea cancelar alta del alumno?"

```

```

dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 6 Then
    SendKeys "(ESC)", True
    DoCmd Close
    DoCmd Close a_form, "lee cuenta a inscribir"
End If
End Sub
Sub Botón827_Click ()
msg1 = "¿Realmente desea cancelar alta del alumno?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 0 Then
    SendKeys "(ESC)", True
    DoCmd Close
    DoCmd Close a_form, "lee cuenta a inscribir"
End If
End Sub
Sub Botón840_Click ()
On Error Goto Err_Botón840_Click
DoCmd DoMenuItemem A_FORMBAR, A_EDITMENU, 10, , A_MENU_VER20
Exit_Botón840_Click:
Exit Sub
Err_Botón840_Click:
MsgBox Error$
Resume Exit_Botón840_Click
End Sub
Sub estado_civil_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRECONTINUE
msg1 = "El estado debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub codigo_postal_BeforeUpdate (Cancel As Integer)
If Len(codigo_p) < 3 Then
msg1 = "El código postal debe ser de 3 dígitos"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
DoCmd CancelEvent
End If
End Sub
Sub estado_civil_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRECONTINUE
msg1 = "El estado civil debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub estado_occupacion_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRECONTINUE
msg1 = "El estado debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub estado_postgrado_ente_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRECONTINUE
msg1 = "El estado debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub estado1_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRECONTINUE
msg1 = "El estado debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub fecha_de_nacimiento_error ()
If Error = 2113 Then
msg1 = "Fecha incorrecta"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
DoCmd CancelEvent
Else
msg1 = "Fecha incorrecta" & " " & prueba
dgdef1 = 0 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
DoCmd CancelEvent
End If
End Sub
Sub Form_BeforeUpdate (Cancel As Integer)
msg1 = "¿Realmente desea dar de alta a este alumno?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub Form_Error (dataerr As Integer, Response As Integer)
Response = DATA_ERRECONTINUE
PECIAINCORRECTA = 2113
If dataerr = PECIAINCORRECTA Then
msg1 = "La fecha es incorrecta"
dgdef1 = 0 + 48 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
DoCmd CancelEvent
End If
End Sub
Sub Form_Load ()
DoCmd GoToRecord a_form, "Datos Generales", A_NEWREC
End Sub
Sub guardar_registro_Click ()
msg1 = "¿Realmente desea dar de alta a este alumno?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 0 Then
If IsNull(paterno) Or IsNull(materno) Or IsNull(nombre) Then
msg1 = "No se captura nombre completo del alumno, el registro no será almacenado"
dgdef1 = 0 + 48 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
On Error Goto Err_guardar_registro_Click
DoCmd DoMenuItemem A_FORMBAR, A_EDITMENU, A_UNDOHELP, , A_MENU_VER20
Exit_guardar_registro_Click:
msg4 = "¿Desea dar de alta otro alumno?"

```

```

dgdDef4 = 4 + 32 + 0
res4 = MsgBox(msg4, dgdDef4, titulo)
If res4 = 7 Then
    DoCmd Close
    DoCmd Close a_form, "lee cuenta a inscribir"
Else
    DoCmd Close
    DoCmd Close a_form, "lee cuenta a inscribir"
    DoCmd OpenForm "lee cuenta a inscribir"
End If
End Sub
Err_guardar_registro_Click
Resume Err_guardar_registro_Click
End If
DoCmd DoMenuItem A_FORMBAR, A_FILE, A_SAVERECORE, , A_MENU_VERRO
msg1 = "¿Debes dar de alta otro alumno?"
dgdDef1 = 4 + 32 + 0
res1 = MsgBox(msg1, dgdDef1, titulo)
If res1 = 7 Then
    DoCmd Close
    DoCmd Close a_form, "lee cuenta a inscribir"
Else
    DoCmd Close
    DoCmd Close a_form, "lee cuenta a inscribir"
    DoCmd OpenForm "lee cuenta a inscribir"
End If
Else
    On Error GoTo Err_guardar_registro_Click
DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, A_UNDORELU, , A_MENU_VERRO
msg1 = "¿Debes dar de alta otro alumno?"
dgdDef1 = 4 + 32 + 0
res1 = MsgBox(msg1, dgdDef1, titulo)
If res1 = 7 Then
    DoCmd Close
    DoCmd Close a_form, "lee cuenta a inscribir"
Else
    DoCmd Close
    DoCmd Close a_form, "lee cuenta a inscribir"
    DoCmd OpenForm "lee cuenta a inscribir"
End If
End If
End Sub
Sub Institucion_licencia_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "La Institución debe coincidir con un valor de la lista!"
dgdDef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdDef1, titulo)
End Sub
Sub Institucion_posgrado_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "La Institución debe coincidir con un valor de la lista!"
dgdDef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdDef1, titulo)
End Sub
Sub pais_licenciatura_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El país debe coincidir con un valor de la lista!"
dgdDef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdDef1, titulo)
End Sub
Sub pais_posgrado_anterior_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El país debe coincidir con un valor de la lista!"
dgdDef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdDef1, titulo)
End Sub
Sub plan_del_nuevo_posgr_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El plan debe coincidir con un valor de la lista!"
dgdDef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdDef1, titulo)
End Sub
Sub RFC_del_tutor_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El RFC debe coincidir con un valor de la lista!"
dgdDef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdDef1, titulo)
End Sub
Sub siguiente1_Click ()
msg4 = "Datos Generales del Alumno correctos?"
dgdDef4 = 4 + 32 + 0
res4 = MsgBox(msg4, dgdDef4, titulo)
If res4 = 0 Then
    DoCmd GoToControl "numero de cuentas"
    DoCmd GoToControl "clave de titular"
Else
    DoCmd GoToControl "numero de cuenta"
End If
End Sub
Sub siguiente2_Click ()
msg4 = "Antecedentes Académicos de Licenciatura correctos?"
dgdDef4 = 4 + 32 + 0
res4 = MsgBox(msg4, dgdDef4, titulo)
If res4 = 0 Then
    DoCmd GoToControl "numero de cuentas"
    DoCmd GoToControl "plan"
Else
    DoCmd GoToControl "numero de cuenta"
End If
End Sub
Sub siguientes_Click ()
msg4 = "Antecedentes Académicos de Posgrado correctos?"
dgdDef4 = 4 + 32 + 0
res4 = MsgBox(msg4, dgdDef4, titulo)
If res4 = 0 Then
    DoCmd GoToControl "ultimo"
    DoCmd GoToControl "ingreso"
Else
    DoCmd GoToControl "previ"

```

```

End If
End Sub
Sub tipo_de_beca_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El nombre de la beca debe coincidir con un valor de la lista"
dgDef1 = 4 + 0 + 0
RES1 = MsgBox(msg1, dgDef1, título)
End Sub
Sub tutor_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El RUC debe coincidir con un valor de la lista"
dgDef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgDef1, título)
End Sub

Sub Botón644_Click ()
On Error GoTo Err_Botón644_Click
msg1 = "¿Realmente desea deshacer el registro?"
dgDef1 = 4 + 32 + 0
res1 = MsgBox(msg1, dgDef1, título)
If res1 = 0 Then
DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, A_UNDOFIELD, A_MENU_VER20
DoCmd Close
Exit_Botón644_Click:
Exit Sub
Err_Botón644_Click:
Resume Exit_Botón644_Click
Else
Exit Sub
End If
End Sub
Sub Botón719_Click ()
On Error GoTo Err_Botón719_Click
msg1 = "¿Realmente desea deshacer el registro?"
dgDef1 = 4 + 32 + 0
res1 = MsgBox(msg1, dgDef1, título)
If res1 = 0 Then
DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, A_UNDOFIELD, A_MENU_VER20
DoCmd Close
Exit_Botón719_Click:
Exit Sub
Err_Botón719_Click:
Resume Exit_Botón719_Click
Else
Exit Sub
End If
End Sub
Sub Botón727_Click ()
On Error GoTo Err_Botón727_Click
msg1 = "¿Realmente desea deshacer el registro?"
dgDef1 = 4 + 32 + 0
res1 = MsgBox(msg1, dgDef1, título)
If res1 = 0 Then
DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, A_UNDOFIELD, A_MENU_VER20
DoCmd Close
Exit_Botón727_Click:
Exit Sub
Err_Botón727_Click:
Resume Exit_Botón727_Click
Else
Exit Sub
End If
End Sub
Sub Botón754_Click ()
End Sub
Sub Botón811_Click ()
msg1 = "¿Realmente desea cancelar alta?"
dgDef1 = 4 + 32 + 0
res1 = MsgBox(msg1, dgDef1, título)
If res1 = 0 Then
DoCmd Close
End If
End Sub
Sub Botón819_Click ()
On Error GoTo Err_Botón819_Click
msg1 = "¿Realmente desea deshacer el registro?"
dgDef1 = 4 + 32 + 0
res1 = MsgBox(msg1, dgDef1, título)
If res1 = 0 Then
DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, A_UNDOFIELD, A_MENU_VER20
DoCmd Close
Exit_Botón819_Click:
Exit Sub
Err_Botón819_Click:
Resume Exit_Botón819_Click
Else
Exit Sub
End If
End Sub
Sub Botón821_Click ()
msg1 = "¿Realmente desea cancelar la consulta?"
dgDef1 = 4 + 32 + 0
res1 = MsgBox(msg1, dgDef1, título)
If res1 = 0 Then
msg2 = "¿Desea consultar otro alumno?"
dgDef2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgDef2, título)
If res2 = 7 Then
DoCmd Close
DoCmd Close A_FORM, "Lee cuenta a consultar"
Else
DoCmd Close
DoCmd Close A_FORM, "Lee cuenta a consultar"
DoCmd OpenForm "Lee cuenta a consultar"
End If
End If
End Sub
Sub Botón823_Click ()
msg1 = "¿Realmente desea cancelar la consulta?"

```



```

dgDef1 = 4 + 32 + 0
res1 = MsgBox(msg1, dgDef1, título)
    If res1 = 0 Then
        msg2 = "¿Debería consultar otro alumno?"
        dgDef2 = 4 + 32 + 0
        res2 = MsgBox(msg2, dgDef2, título)
        If res2 = 7 Then
            DoCmd Close
            DoCmd Close A_FORM, "lee cuenta a consultar"
        Else
            DoCmd Close
            DoCmd Close A_FORM, "lee cuenta a consultar"
            DoCmd OpenForm "lee cuenta a consultar"
        End If
    End If
End Sub

Sub Botón823_Click ()
    msg1 = "¿Realmente desea cancelar la consulta?"
    dgDef1 = 4 + 32 + 0
    res1 = MsgBox(msg1, dgDef1, título)
    If res1 = 0 Then
        msg2 = "¿Debería consultar otro alumno?"
        dgDef2 = 4 + 32 + 0
        res2 = MsgBox(msg2, dgDef2, título)
        If res2 = 7 Then
            DoCmd Close
            DoCmd Close A_FORM, "lee cuenta a consultar"
        Else
            DoCmd Close
            DoCmd Close A_FORM, "lee cuenta a consultar"
            DoCmd OpenForm "lee cuenta a consultar"
        End If
    End If
End Sub

Sub Botón827_Click ()
    msg1 = "¿Realmente desea cancelar esta consulta?"
    dgDef1 = 4 + 32 + 0
    res1 = MsgBox(msg1, dgDef1, título)
    If res1 = 0 Then
        SendKeys "{ESC}", True
        DoCmd Close
        DoCmd Close A_FORM, "lee cuenta a usar ribir"
    End If
End Sub

Sub Botón840_Click ()
On Error GoTo Err_Botón840_Click
    DoCmd DoMenuItem A_FORMUAR, A_MENUITEM1, 10, , A_MENU_VER20
Exit_Botón840_Click:
Exit Sub
Err_Botón840_Click:
MsgBox Error!
Resume Exit_Botón840_Click
End Sub

Sub Botón870_Click ()
End Sub

Sub Botón872_Click ()
On Error GoTo Err_Botón872_Click
    DoCmd OpenReport "datos de alumnos", A_normal, , "[n_cuenta] = formularios[Datos Generales a consultar][numero de cuenta]"
Exit_Botón872_Click:
Exit Sub
Err_Botón872_Click:
m3 = "¿Se canceló impresión?"
f3 = 48 + 0 + 0
f3 = MsgBox(m3, f3, título)
Resume Exit_Botón872_Click
End Sub

Sub Botón874_Click ()
On Error GoTo Err_Botón874_Click
    DoCmd OpenReport "datos de alumnos", A_normal, , "[n_cuenta] = formularios[Datos Generales a consultar][numero de cuenta]"
Exit_Botón874_Click:
Exit Sub
Err_Botón874_Click:
m3 = "¿Se canceló impresión?"
f3 = 48 + 0 + 0
f3 = MsgBox(m3, f3, título)
Resume Exit_Botón874_Click
End Sub

Sub Botón876_Click ()
On Error GoTo Err_Botón876_Click
    DoCmd OpenReport "datos de alumnos", A_normal, , "[n_cuenta] = formularios[Datos Generales a consultar][numero de cuenta]"
Exit_Botón876_Click:
Exit Sub
Err_Botón876_Click:
m3 = "¿Se canceló impresión?"
f3 = 48 + 0 + 0
f3 = MsgBox(m3, f3, título)
Resume Exit_Botón876_Click
End Sub

Sub Botón878_Click ()
    DoCmd DoMenuItem A_FORMUAR, A_MENUITEM1, A_SELECTRECORD_V2, , A_MENU_VER20
    DoCmd Print A_SELECTRECORD
On Error GoTo Err_Botón878_Click
Exit_Botón878_Click:
Exit Sub
Err_Botón878_Click:
MsgBox Error!
Resume Exit_Botón878_Click
End Sub

Sub Botón879_Click ()
On Error GoTo Err_Botón879_Click
    Dim DocName As String
    Dim MyForm As Form
    DocName = "ARCC Proceso Semestral"
    Set MyForm = Screen.ActiveForm
    DoCmd SelectObject A_FORM, DocName, True
    DoCmd Print
    DoCmd SelectObject A_FORM, MyForm.Name, False
Exit_Botón879_Click:
Exit Sub

```

```

Err_botón879_Click
MsgBox Error8
End Sub
Sub clave_de_título_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "La carrera debe coincidir con un valor de la lista"
dglDef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dglDef1, título)
End Sub
Sub código_postal_BeforeUpdate (Cancel As Integer)
If Len(código_p) < 5 Then
msg1 = "El código postal debe ser de 5 dígitos"
dglDef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dglDef1, título)
DoCmd CancelEvent
End If
End Sub
Sub estado_civil_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El estado civil debe coincidir con un valor de la lista"
dglDef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dglDef1, título)
End Sub
Sub estado_inscripción_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El estado debe coincidir con un valor de la lista"
dglDef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dglDef1, título)
End Sub
Sub estado_programa_mie_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El estado debe coincidir con un valor de la lista"
dglDef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dglDef1, título)
End Sub
Sub estado_papel_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El estado debe coincidir con un valor de la lista"
dglDef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dglDef1, título)
End Sub
Sub fecha_de_nacimiento_error ()
If Error = 2113 Then
msg1 = "Fecha incorrecta"
dglDef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dglDef1, título)
DoCmd CancelEvent
Else
msg1 = "Fecha incorrecta" & " - & prueba"
dglDef1 = 0 + 0 + 0
res1 = MsgBox(msg1, dglDef1, título)
DoCmd CancelEvent
End If
End Sub
Sub form_BeforeUpdate (Cancel As Integer)
msg1 = "Realmente desea dar de alta a este alumno?"
dglDef1 = 4 + 32 + 0
res1 = MsgBox(msg1, dglDef1, título)
End Sub
Sub form_Error (dataerr As Integer, Response As Integer)
Response = DATA_ERRCONTINUE
FUNCTIONCORRECTA = 2113
If dataerr = FUNCTIONCORRECTA Then
msg1 = "La fecha es incorrecta"
dglDef1 = 0 + 48 + 0
res1 = MsgBox(msg1, dglDef1, título)
DoCmd CancelEvent
End If
End Sub
Sub guardar_registro_Click ()
msg1 = "Realmente desea dar de alta a este alumno?"
dglDef1 = 4 + 32 + 0
res1 = MsgBox(msg1, dglDef1, título)
If res1 = 6 Then
If IsNull(paterno) Or IsNull(materno) Or IsNull(nombre) Then
msg1 = "No se capturan nombre completo del alumno, el registro no será almacenado"
dglDef1 = 0 + 48 + 0
res1 = MsgBox(msg1, dglDef1, título)
On Error GoTo Err_guardar_registro_Click
DoCmd DoMenuItem A_FORMBAR, A_FITMENCU, A_INNOVEX, A_MENU_VIR20
End Sub
Sub guardar_registro_Click
msg1 = "¿Desea dar de alta otro alumno?"
dglDef1 = 4 + 32 + 0
res1 = MsgBox(msg1, dglDef1, título)
If res1 = 7 Then
DoCmd Close
DoCmd Close A_FORM, "Tec cuenta a inscribir"
Else
DoCmd Close
DoCmd Close A_FORM, "Tec cuenta a inscribir"
DoCmd OpenForm "Tec cuenta a inscribir"
End If
End Sub
Sub guardar_registro_Click
Resume Err_guardar_registro_Click
End If
DoCmd DoMenuItem A_FORMBAR, A_MIE, A_SAVREFCORR, A_MENU_VIR20
msg1 = "¿Desea dar de alta otro alumno?"
dglDef1 = 4 + 32 + 0
res1 = MsgBox(msg1, dglDef1, título)
If res1 = 7 Then
DoCmd Close
DoCmd Close A_FORM, "Tec cuenta a inscribir"
Else
DoCmd Close
DoCmd Close A_FORM, "Tec cuenta a inscribir"
DoCmd OpenForm "Tec cuenta a inscribir"
End If
Else

```

```

On Error Goto Err_Quitar_Registro_Click
DoCmd DoMenuItem A_FORMAR, A_ENTRADA, A_UNDOFIELD, , A_MENU_VERZO
msgB = "¿Deuda dar de otro alumno?"
dgDef1 = 4 + 32 + 0
res1 = MsgBox(msgB, dgDef1, titulo)
If res1 = 7 Then
DoCmd Close
DoCmd Close A_FORM, "lea cuenta a inscribir"
Else
DoCmd Close
DoCmd Close A_FORM, "lea cuenta a inscribir"
DoCmd Openform "lea cuenta a inscribir"
End If
End If
End Sub
Sub Imprimir_Click ()
On Error Goto Err_Imprimir_Click
DoCmd Openform! "datos de alumnos", A_normal, , "In_cuenta = formularios[Datos Generales a consultar]||numero de cuenta!"
Exit Sub
Exit Imprimir_Click
Exit Sub
Err_Imprimir_Click:
res = "¿Se canceló impresión?"
rs = 48 + 0 + 0
rs = MsgBox(rs, rs, titulo)
Resume Exit_Imprimir_Click
End Sub
Sub Institucion_licencia_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "¿La Institución debe coincidir con un valor de la lista?"
dgDef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dgDef1, titulo)
End Sub
Sub Institucion_posgrado_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "¿La Institución debe coincidir con un valor de la lista?"
dgDef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dgDef1, titulo)
End Sub
Sub pala_licenciatura_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "¿El pala debe coincidir con un valor de la lista?"
dgDef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dgDef1, titulo)
End Sub
Sub pala_posgrado_anti_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "¿El pala debe coincidir con un valor de la lista?"
dgDef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dgDef1, titulo)
End Sub
Sub plan_del_nuevo_pogra_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "¿El plan debe coincidir con un valor de la lista?"
dgDef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dgDef1, titulo)
End Sub
Sub RFC_dol_Tutor_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "¿El RFC debe coincidir con un valor de la lista?"
dgDef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dgDef1, titulo)
End Sub
Sub salir_Click ()
msg1 = "¿Debes consultar otro alumno?"
dgDef1 = 4 + 32 + 0
res1 = MsgBox(msg1, dgDef1, titulo)
If res1 = 7 Then
DoCmd Close
DoCmd Close A_FORM, "lea cuenta a consultar"
Else
DoCmd Close
DoCmd Close A_FORM, "lea cuenta a consultar"
DoCmd Openform "lea cuenta a consultar"
End If
End Sub
Sub tipo_de_beca_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "¿El nombre de la beca debe coincidir con un valor de la lista?"
dgDef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dgDef1, titulo)
End Sub
Sub tutor_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "¿El RFC debe coincidir con un valor de la lista?"
dgDef1 = 48 + 0 + 0
res1 = MsgBox(msg1, dgDef1, titulo)
End Sub
Sub Aceptar_Click ()
m1 = "¿Debes hacer el Cambio?"
f1 = 4 + 32 + 0
r1 = MsgBox(m1, f1, titulo)
If r1 = 0 Then 'hace cambio
Dim tea As Database, mitabla As Recordset
Set tea = dbengine.workspaces(0).databases(0)
Set mitabla = tea.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
mitabla.Index = "ACTA"
mitabla.Seek "=", Me!lacta.nuevo!
If Not (mitabla.NoMatch) Then
m1 = "¿El número de Acta ya existe?"
f1 = 48 + 0 + 0
r1 = MsgBox(m1, f1, titulo)
SendKeys "[esc]", True
mitabla.Close
Exit Sub
Else
DoCmd RunMacro "Consulta para cambio de nota"
'Ejecutar CONSULTA-48-REZAFI-AZ0-

```

```

mitabla.Close
End If
Else
  Me.Hoverari.SetFocus
  Me.Letra_muestra.Visible = False
  Me.Letras.Visible = False
  SendKeys "{esc}", True
  End If
End Sub
Sub Boton41_Click ()
End Sub
Sub CANT_AfterUpdate ()
If Val(Me.Letra) < 0 Then
  Me.Letras.Visible = "NA"
Else
  Me.Letras.Visible = "A"
End If
End Sub
Sub Eliminar_acta_Click ()
m1 = ?"¿ número de acta será cancelado? (Presione Continuar)"
f1 = 48 + 0 + 0
r1 = MsgBox(m1, f1, titulo)
If r1 = 0 Then
  GoTo reemplazo
Else
  DoCmd CancelEvent
  SendKeys "{esc}", True
  Exit Sub
End Sub
DoCmd Close
End If
reemplazo:
DoCmd RunMacro "macro de actas eliminar actas"
Exit Sub
End Sub

Sub Boton24_Click ()
On Error GoTo Err_Boton24_Click
DoCmd Quit
Exit_Boton24_Click
Err_Boton24_Click:
MsgBox Error
Resume Err_Boton24_Click
End Sub
Sub ENTRADA_Click ()
DoCmd OpenForm "Formulario Entrada"
End Sub
Sub Ingresado1_Click ()
DoCmd OpenForm "Semestre Actual"
End Sub
Sub salir_Click ()
DoCmd Quit A_Save
End Sub

Sub cerrar_Click ()
msg4 = ?"¿Bar de alta Otra Clase?"
dgedf4 = 4 + 32 + 0
res4 = MsgBox(msg4, dgedf4, titulo)
If res4 = 0 Then
  SendKeys "{esc}", True
  DoCmd RunMacro "entradas otra clase"
Else
  DoCmd Close
End If
End Sub

Sub CLAVE_DE_MATERIA_NotaFinal (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
Res p
msg1 = ?"La clave de la Materia debe coincidir con un valor de la lista"
dgedf1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgedf1, titulo)
End Sub
Sub Consultar_Clasas_Click ()
Dim res As Database, mitabla As Recordset
set res = dbeengine.workspaces(0).databases(0)
set mitabla = res.OpenRecordset("CLASE", DB_OPEN_TABLE)
total = mitabla.RecordCount
If total = 0 Then
  m3 = ?"No hay clases dadas de alta para el semestre actual"
  f3 = 48 + 0 + 0
  r3 = MsgBox(m3, f3, titulo)
  Exit Sub
Else
  mitabla.Index = "semestre"
  mitabla.Seek "=", Formulario(Semestre Actual)|Semestre|
  If Not (mitabla.NoMatch) Then
    DoCmd OpenForm "Formulario de consulta de clase en alta y modificaciones"
  Else
    m3 = ?"No hay clases dadas de alta para el semestre actual"
    f3 = 48 + 0 + 0
    r3 = MsgBox(m3, f3, titulo)
  Exit Sub
End If
End If
mitabla.Close
End Sub
Sub Entr_de_alta_clase_Click ()
If IsNull(Me.HERRamientas) Or IsNull(Me.Letra) Or IsNull(Me.Letra) Or IsNull(Me.Letra) Then
  msg1 = ?"Alguno de los campos (Profesor, Materia, Grupo, sesión) está vacío"
  dgedf1 = 48 + 0 + 0
  RES1 = MsgBox(msg1, dgedf1, titulo)
  Exit Sub
End Sub
Msg 2 = ?"¿Habitó correctos?"
dgedf2 = 4 + 32 + 0
RES2 = MsgBox(msg2, dgedf2, titulo)
If RES2 = 0 Then
  msg3 = ?"¿Bar de alta esta Clase?"
  dgedf3 = 4 + 32 + 0

```

```

RES4 = MsgBox(msg4, dgdef4, titulo)
If RES4 = 8 Then
Dim mIBD As Database, mItabla As Recordset
Set mIBD = dbengine.workspaces(0).databases(0)
Set mItabla = mIBD.OpenRecordset("CLASE", DB_OPEN_TABLE) ' Abre la tabla.
mItabla.Index = "materia" ' Establece el índice activo.
mItabla.Seek "=", Me!CLAVE DE MATERIA
If Not (mItabla.NoMatch) Then
Do Until mItabla.EOF
If mItabla[clv_mai] = Me!CLAVE DE MATERIA And mItabla[numero] = Me!grupo And Me!actual = mItabla[semestre] Then
m2 = "El grupo ya existe!"
f2 = 48 + 0 + MI_DEBOTOÑZ
r2 = MsgBox(m2, f2, titulo)
SendKeys "{esc}", True
DoCmd CancelEvent
Exit Do
Else
mItabla.MoveNext
End If
Loop
End If
mItabla.Close ' Cierra la tabla.
If Len(numero) < 4 Then
msg4 = "El grupo es incorrecto"
dgdef4 = 48 + 0 + 0
res4 = MsgBox(msg4, dgdef4, titulo)
DoCmd GoToControl "grupo"
Exit Sub
End If
DoCmd DoMenuItem A_FORMBAR, A_FILE, A_SAVERECORD, , A_MENU_VERZO
msg4 = "¿Var de otra Clase?"
dgdef4 = 4 + 32 + 0
res4 = MsgBox(msg4, dgdef4, titulo)
If res4 = 0 Then
DoCmd GoToRecord , , A_NEWREC
DoCmd RunMacro "entradas.otra clase"
Else
DoCmd Close
End If
Else
SendKeys "{ESC}", True
msg4 = "¿Var de otra Clase?"
dgdef4 = 4 + 32 + 0
res4 = MsgBox(msg4, dgdef4, titulo)
If res4 = 0 Then
DoCmd RunMacro "otramateria.otra clase"
Else
DoCmd Close
End If
End If
Else ' De datos correctos
DoCmd GoToControl "RFC PROFESOR"
End If
End Sub
Sub estudio_NoInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "La Clave de la Materia debe coincidir con un valor de la lista"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub Form_Error (DataErr As Integer, Response As Integer)
Response = DATA_ERRCONTINUE
Const NCHAYNINGUN = 3101
If DataErr = NCHAYNINGUN Then
m2 = "Existe algún campo vacío? Presionar ESC para continuar."
f2 = 48 + 0 + MI_DEBOTOÑZ
r2 = MsgBox(m2, f2, titulo)
DoCmd CancelEvent
SendKeys "{ESC}", True
SendKeys "{ESC}", True
End If
End Sub
Sub Form_Open (Cancel As Integer)
Dim (IpoObjeto As Integer, nombreObjeto As String, estadoObjeto As Integer)
IpoObjeto = A_form
nombreObjeto = "semestre Actual"
estadoObjeto = SysCmd(SYSCMD_GETOBJECTSTATE, (IpoObjeto, nombreObjeto)
If estadoObjeto = 0 Then
msg4 = "No hay semestre actual. Elegir semestre y repetir la acción que desea!"
dgdef4 = 0 + 0 + 0
res4 = MsgBox(msg4, dgdef4, titulo)
DoCmd OpenForm "Semestre Actual"
DoCmd CancelEvent
End If
End Sub
Sub DRUPO_BeforeUpdate (Cancel As Integer)
Rem If Len(grupo) < 4 Then
If IsNull(grupo) Then
m2 = "El número de grupo debe ser de 4 dígitos!"
f2 = 0 + 48 + MI_DEBOTOÑZ
r2 = MsgBox(m2, f2, titulo)
SendKeys "{esc}", True
DoCmd CancelEvent
End If
Dim mIBD As Database, mItabla As Recordset
Set mIBD = dbengine.workspaces(0).databases(0)
Set mItabla = mIBD.OpenRecordset("CLASE", DB_OPEN_TABLE) ' Abre la tabla.
mItabla.Index = "materia" ' Establece el índice activo.
mItabla.Seek "=", Me!CLAVE DE MATERIA
If Not (mItabla.NoMatch) Then
Do Until mItabla.EOF
If mItabla[clv_mai] = Me!CLAVE DE MATERIA And mItabla[numero] = Me!grupo And Me!actual = mItabla[semestre] Then
m2 = "El grupo ya existe!"
f2 = 48 + 0 + MI_DEBOTOÑZ
r2 = MsgBox(m2, f2, titulo)
SendKeys "{esc}", False
DoCmd CancelEvent
Exit Do
Else

```

```

        mtabla.MoreNext
    End If
    Loop
End If
mtabla.Close ' Cierre la tabla.
End Sub
Sub guarda_Click ()
On Error GoTo Err_guarda_Click
    DoCmd ExecuteSQL A_FORMUAR, A_STLE, A_BAVERECORD, , A_MENUL_VERZO
Exit_guarda_Click:
    Exit Sub
Err_guarda_Click:
    MsgBox Error#
    Resume Exit_guarda_Click
End Sub
Sub nombre_de_la_materia_NoInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
    Keep
    msg1 = "El nombre de la Materia debe coincidir con un valor de la lista"
    dgdef1 = 48 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub RFC_PROFESOR_NoInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
    Keep
    msg1 = "El RFC debe coincidir con un valor de la lista"
    dgdef1 = 0 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub RACION_NoInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
    Keep
    msg1 = "El RACION debe coincidir con un valor de la lista"
    dgdef1 = 48 + 0 + 0
    RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub Acepta_Opcion_Click ()
If Form!Proceso semestral!bandera = "4" And Me!tipo de alumno = 1 And Me!Para primer ingreso = 1 And Me!Unam o no Unam = 1 Then
    DoCmd OpenForm "Tec cuenta a Iniciar"
    Exit Sub
End If
If Form!Proceso semestral!bandera = "4" And Me!tipo de alumno = 1 And Me!Para primer ingreso = 2 And Me!Unam o no Unam = 1 Then
    DoCmd OpenForm "Tec cuenta para plan"
    Exit Sub
End If
If Form!Proceso semestral!bandera = "4" And Me!tipo de alumno = 1 And Me!Para primer ingreso = 3 And Me!Unam o no Unam = 1 Then
    DoCmd OpenForm "Tec cuenta a dar de alta materias"
    Exit Sub
End If
If Form!Proceso semestral!bandera = "4" And Me!tipo de alumno = 2 Then
    DoCmd OpenForm "Tec cuenta a dar de alta materias"
    Exit Sub
End If
If Form!Proceso semestral!bandera = "4" And Me!tipo de alumno = 1 And Me!Para primer ingreso = 1 And Me!Unam o no Unam = 2 Then
    DoCmd OpenForm "Asigna numero de cuenta provisional"
    Exit Sub
End If
If Form!Proceso semestral!bandera = "4" And Me!tipo de alumno = 1 And Me!Para primer ingreso = 2 Then
    DoCmd OpenForm "Tec cuenta para plan"
    Exit Sub
End If
If Form!Proceso semestral!bandera = "4" And Me!tipo de alumno = 1 And Me!Para primer ingreso = 4 Then
    DoCmd OpenForm "Tec cuenta a dar de alta materias"
    Exit Sub
End If
Sub cancela_opcion_Click ()
Form!Proceso semestral!cancelar proceso semestral.Enabled = True
Form!Proceso semestral!cancelar proceso semestral.SetFocus
Form!Proceso semestral!finl.Visible = False
Form!Proceso semestral!materias.Enabled = True
Form!Proceso semestral!profesor.Enabled = True
Form!Proceso semestral!horarios.Enabled = True
Form!Proceso semestral!inscripciones.Enabled = True
Form!Proceso semestral!lecta.Enabled = True
End Sub
Sub cancela_opcion_Leafocus ()
DoCmd Close
End Sub
Sub Form_Loadfocus ()
DoCmd Close
End Sub
Sub Form_Open (Cancel As Integer)
Me!Para primer ingreso.Visible = False
Me!Unam o no Unam.Visible = False
Me!barra1.Visible = False
Me!barra2.Visible = False
End Sub
Sub Para_primer_ingreso_AfterUpdate ()
If Me!tipo de alumno = 1 And Me!Para primer ingreso = 1 Or Me!Para primer ingreso = 2 Or Me!Para primer ingreso = 3 Then
    Me!Unam o no Unam.Visible = True
    Me!barra2.Visible = True
Else
    Me!Unam o no Unam.Visible = False
    Me!barra2.Visible = False
End If
If Me!Para primer ingreso = 2 Or Me!Para primer ingreso = 3 Then
    Me!Unam o no Unam.Visible = False
End If
End Sub
Sub tipo_de_alumno_AfterUpdate ()
If Me!tipo de alumno = 1 And Form!Proceso semestral!bandera = 4 Then
    Me!Para primer ingreso.Visible = True
    Me!Unam o no Unam.Visible = True
    Me!barra1.Visible = True
    Me!barra2.Visible = True
Else
    Me!Para primer ingreso.Visible = False
    Me!Unam o no Unam.Visible = False
    Me!barra1.Visible = False
    Me!barra2.Visible = False
End If
End Sub

```

```

Sub abrir_formulario_Click ()
On Error GoTo Err_abrir_formulario_Click
Dim DocName As String
Dim LinkCriteria As String
DocName = "tabla de inscripciones"
DocCmd.OpenForm DocName, , LinkCriteria
Err_abrir_formulario_Click:
Exit Sub
Err_abrir_formulario_Click:
MsgBox Error
Resume Err_abrir_formulario_Click
End Sub

Sub Aceptar_AfterUpdate ()
msg1 = "¿Inscribir al Alumno en esta Materia?"
dgdof1 = 4 + 32 + 0
res1 = MsgBox(msg1, dgdof1, titulo)
If res1 = 0 Then
Dim m1RD As Database, m1tabla As Recordset
Set m1RD = DSEngine.Workspaces(0).Database(0)
Set m1tabla = m1RD.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
m1tabla.Index = "N_CUENTA" 'Indicase el índice activo.
m1tabla.Seek "=", M1IN_CUENTA
If Not (m1tabla.NameMatch) Then
Do Until m1tabla.EOF
If m1tabla[CLV_MAT] = M1[CLAVE] And m1tabla[N_CUENTA] = M1[N_CUENTA] And m1tabla[Semestre] = formulario[Semestre actual][Semestre] Then
m2 = "¿El Alumno ya está inscrito en esta Materia?"
f2 = 48 + 0 + M1L_DEPTOSN2
r2 = MsgBox(m2, f2, titulo)
SendKeys "{esc}", True
Exit Do
Else
m1tabla.MoveNext
End If
Loop
msg1 = "¿Inscribir al alumno en otra Materia?"
dgdof1 = 4 + 32 + 0
res1 = MsgBox(msg1, dgdof1, titulo)
If res1 = 0 Then
DocCmd.Close
Else
DocCmd.Close
DocCmd.Close A_FORM, "tabla de materias y grupos"
DocCmd.Close A_FORM, "ace cuenta a dar de alta materias"
End If
Exit Sub
Else
'Aquí guarda
msg2 = "¿Inscribir al alumno en otra Materia?"
dgdof2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdof2, titulo)
If res2 = 0 Then
DocCmd.Close
Else
DocCmd.Close
DocCmd.Close A_FORM, "tabla de materias y grupos"
DocCmd.Close A_FORM, "ace cuenta a dar de alta materias"
End If
End If
Else 'en caso de no aceptar inscripción
SendKeys "{ESC}", True
DocCmd.CancelEvent
msg2 = "¿Inscribir al alumno en otra Materia?"
dgdof2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdof2, titulo)
If res2 = 0 Then
DocCmd.Close
Else
DocCmd.Close
DocCmd.Close A_FORM, "tabla de materias y grupos"
DocCmd.Close A_FORM, "ace cuenta a dar de alta materias"
End If
End If
m1tabla.Close
End Sub

Sub Boton100_Click ()
DocCmd.GoToRecord A_FORM, "formularios", A_NEXT
msg1 = "¿Inscribir al Alumno en esta Materia?"
dgdof1 = 4 + 32 + 0
res1 = MsgBox(msg1, dgdof1, titulo)
If res1 = 0 Then
Dim m1RD As Database, m1tabla As Recordset
Set m1RD = DSEngine.Workspaces(0).Database(0)
Set m1tabla = m1RD.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
m1tabla.Index = "N_CUENTA" 'Indicase el índice activo.
m1tabla.Seek "=", M1IN_CUENTA
If Not (m1tabla.NameMatch) Then
Do Until m1tabla.EOF
If m1tabla[N_CUENTA] = M1[N_CUENTA] And m1tabla[CLV_MAT] = M1[CLAVE] Then
m2 = "¿El Alumno ya está inscrito en esta Materia?"
f2 = 48 + 0 + M1L_DEPTOSN2
r2 = MsgBox(m2, f2, titulo)
Exit Do
Else
m1tabla.MoveNext
End If
Loop
End If
m1tabla.Close 'Cierra la tabla.
msg2 = "¿Inscribir al alumno en otra Materia?"
dgdof2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdof2, titulo)
If res2 = 0 Then
DocCmd.Close
formulario[tabla de MATERIAS Y GRUPOS].Visible = True
Else
DocCmd.Close
DocCmd.Close A_FORM, "tabla de materias y grupos"
End If

```

```

Else
Dim mIBV2 As Database, mtabla2 As Recordset
Set mIBV2 = DBEngine.Workspaces(0).Databases(0)
Set mtabla2 = mIBV2.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
mtabla2.Index = "RCURCITA" ' Establece el índice activo
mtabla2.Seek "=", Ase([C_CURCITA])
If Not (mtabla2.NoMatch) Then
If mtabla2([C_CURCITA]) = Ase([C_CURCITA]) And mtabla2([C_IV_MATE]) = Ase([C_IV_MATE]) Then
msg2 = "Inscribir al alumno en otra Materia?"
dgdcl2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdcl2, titulo)
If res2 = 1 Then
Docmd.Close
Formulario([Tabla de MATERIAS Y GRUPOS]) visible = True
Exit Sub
Else
Docmd.Close
Docmd.Close A_FORM, "Tabla de materias y grupos"
Exit Sub
End If
End If
End If
mtabla2.Close
Docmd.DoMenuItem A_FORMBAR, A_EDITMENU, A_ZELZ.TRECORR1_VZ, A_MENU_VER20
Docmd.DoMenuItem A_FORMBAR, A_EDITMENU, A_ZELZ.VZ, A_MENU_VCR20
Docmd.SetWarnings False
msg2 = "Inscribir al alumno en otra Materia?"
dgdcl2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdcl2, titulo)
If res2 = 1 Then
Docmd.Close
Formulario([Tabla de MATERIAS Y GRUPOS]) visible = True
Exit Sub
Else
Docmd.Close
Docmd.Close A_FORM, "Tabla de materias y grupos"
Exit Sub
End If
End If
End Sub
Sub Form_Error (Output As Integer, Response As Integer)
Response = DATA_CONTINUE
Const CLAVECORRECTA = 3022
If Output = CLAVECORRECTA Then
m2 = "El alumno ya está inscrito"
f2 = 0 + MR_LCONOALTO + MB_DIFROTÓN2
r2 = MsgBox(m2, f2)
Sendkeys "{ESC}", True
End If
End Sub
Sub Form_Load ()
Docmd.GoToRecord A_FORM, "Inscripción actual", A_NLWRK
End Sub
Sub Potenti_Click ()
Const MB_ACEPTAR = 0, MB_ACEPTARANCELAR = 1 ' 1x fine botones
Const MB_SINOCANCELAR = 3, MR_SINCO = 4, MR_REPCAN = 5
Const MB_LCONOALTO = 16, MB_LCONOINTERROGACION = 32 ' 1x fine iconos
Const MB_LCONOADMIRACION = 48
Const MB_DIFROTÓN2 = 256, ID4 = 6, IDno = 7, IDCAN = 5, MB_DIFROTÓN1 = 0 ' 1x fine otros.
If (IsNull(clavemodificar)) Then
msg4 = "Clave incorrecta!"
dgdcl4 = MB_REPCAN + MB_LCONOADMIRACION + MB_DIFROTÓN1
RES4 = MsgBox(msg4, dgdcl4, titulo)
If RES4 = 2 Then
Docmd.Close
Else
Docmd.GoToControl "clavemodificar"
End If
Else
Dim tes As Database, mtabla As Recordset
Set tes = dbengine.workspace(0).databases(0)
Set mtabla = tes.OpenRecordset("Materia", DB_OPEN_TABLE)
mtabla.Index = "clave"
mtabla.Seek "=", clavemodificar
If mtabla.NoMatch Then
msg3 = "La Materia no existe!"
dgdcl3 = MB_REPCAN + MB_LCONOADMIRACION + MB_DIFROTÓN1
RES3 = MsgBox(msg3, dgdcl3, titulo)
If RES3 = 2 Then
Docmd.Close
Else
Docmd.GoToControl "clavemodifiaur"
End If
Else
Docmd.OpenForm "Modificar Materia", , "[civ_mate]=formularios\lee clave materia para modificar\llavemodificar"
End If
mtabla.Close
End If
End Sub
Sub Cancelar_Click ()
Docmd.Close
End Sub
Sub clavemodificar_AfterUpdate ()
Const MB_ACEPTAR = 0, MB_ACEPTARANCELAR = 1 ' 1x fine botones
Const MB_SINOCANCELAR = 3, MR_SINCO = 4, MR_REPCAN = 5
Const MB_LCONOALTO = 16, MB_LCONOINTERROGACION = 32 ' 1x fine iconos
Const MB_LCONOADMIRACION = 48
Const MB_DIFROTÓN2 = 256, ID4 = 6, IDno = 7, IDCAN = 5, MB_DIFROTÓN1 = 0 ' 1x fine otros
If (IsNull(clavemodificar)) Then
msg4 = "Clave incorrecta!"
dgdcl4 = MB_REPCAN + MB_LCONOADMIRACION + MB_DIFROTÓN1
RES4 = MsgBox(msg4, dgdcl4, titulo)
If RES4 = 1 Then
Docmd.Close
Else
Docmd.GoToControl "clavemodificar"
End If
Else
End If
End Sub

```



```

Dim lex As Database, mifabla As Recordset
Set lex = dbengine.workspaces(0).databases(0)
Set mifabla = lex.OpenRecordset("materias", DB_OPEN_TABLE)
mifabla.Index = "clave"
mifabla.Seek "=", claveModificar
If mifabla.NoMatch Then
    msg3 = "¡La Materia no existe!"
    dgDef3 = MB_REPCAN + MB_ICONADMIRACION + MB_DEFBOTON1
    RES3 = MsgBox(msg3, dgDef3, titulo)
    If RES3 = 2 Then
        DoCmd Close
    Else
        DoCmd GoToControl "claveModificar"
    End If
Else
    DoCmd OpenForm "Modificar Materia", , "[tbl_materias]formularios[[Le clave materia para modificar]](claveModificar)"
End If
mifabla.Close
End If
End Sub

Sub Form_Error (DataErr As Integer, Response As Integer)
Response = DATA_ERRCONTINUE
Const ERRORMASCARA = 2279
If DataErr = ERRORMASCARA Then
If DataErr = ERRORMASCARA Then
    m1 = "¡Clave Incorrecta!"
    f1 = 3 + 16 + MB_DEFBOTON1
    e1 = MsgBox(m1, f1, titulo)
    If e1 = 2 Then
        DoCmd Close
    Else
        DoCmd GoToControl "claveModificar"
    End If
End If
End Sub

Sub Iconos_mis_AfterUpdate ()
Const MB_ACEPTAR = 0, MB_ACEPTARCANCELAR = 1 ' Define botones.
Const MB_SINOCANCELAR = 3, MB_SINO = 4, MB_REPCAN = 5
Const MB_ICONALTO = 16, MB_ICONINTERROGACION = 32 ' Define iconos
Const MB_ICONADMIRACION = 48
Const MB_DEFBOTON2 = 256, IDSI = 0, IDNO = 7, IDCAN = 8, MB_DEFBOTON1 = 0 ' Define otros
If IsNull(iconos) Then
    msg4 = "¡Número de Cuenta Incorrecto!"
    dgDef4 = MB_REPCAN + MB_ICONADMIRACION + MB_DEFBOTON1
    RES4 = MsgBox(msg4, dgDef4, titulo)
    If RES4 = 2 Then
        DoCmd Close
    End If
Else
    DoCmd OpenForm "Datos Generales a consultar", , "[tbl_cuentas] formularios[[Le cuenta a consultar]](iconos)"
End If
mifabla.Close
End If
End Sub

Sub Botón6_Click ()
Const MB_ACEPTAR = 0, MB_ACEPTARCANCELAR = 1 ' Define botones
Const MB_SINOCANCELAR = 3, MB_SINO = 4, MB_REPCAN = 5
Const MB_ICONALTO = 16, MB_ICONINTERROGACION = 32 ' Define iconos
Const MB_ICONADMIRACION = 48
Const MB_DEFBOTON2 = 256, IDSI = 0, IDNO = 7, IDCAN = 8, MB_DEFBOTON1 = 0 ' Define otros
If IsNull(iconos) Then
    msg4 = "¡Número de cuenta incorrecto!"
    dgDef4 = MB_REPCAN + MB_ICONADMIRACION + MB_DEFBOTON1
    RES4 = MsgBox(msg4, dgDef4, titulo)
    If RES4 = 2 Then
        DoCmd Close
    Else
        DoCmd GoToControl "leconsultar"
    End If
End If
Else
    DoCmd OpenForm "Datos Generales a consultar", , "[tbl_cuentas] formularios[[Le cuenta a consultar]](leconsultar)"
End If
mifabla.Close
End If
End Sub

Sub Cancelar_Click ()
DoCmd Close
End Sub

```

```

Sub Form_Error (DataErr As Integer, Response As Integer)
    Response = DATA_ERROCONTINUE
    Const ERROCONTINUE = 2271
    If DataErr = ERROCONTINUE Then
        If DataErr = ERROCONTINUE Then
            m1 = "Número de Cuenta No Válido"
            f1 = 8 + 16 + MB_DEBOTOÑ1
            r1 = MsgBox(m1, f1, título)
            If r1 = 2 Then
                DoCmd Close
            Else
                DoCmd GoToControl "seconsulta"
            End If
        End If
    End Sub

Sub seconsulta_AfterUpdate ()
    Const MB_ACEPTAR = 0, MB_CANCELAR = 1 ' Define botones.
    Const MB_RECANCELAR = 2, MB_SINO = 4, MB_SI = 5
    Const MB_CONCONFIRMACION = 16, MB_CONCONFIRMACION = 32 ' Define iconos.
    Const MB_CONCONFIRMACION = 48
    Const MB_DEBOTOÑ2 = 256, f1a = 1, f1b = 7, f1CAN = 5, MB_DEBOTOÑ1 = 0 ' Define otros.
    If IsNull(seconsulta) Then
        msg4 = "Número de Cuenta incorrecto"
        dgDef4 = MB_RECANCELAR + MB_CONCONFIRMACION + MB_DEBOTOÑ1
        RES4 = MsgBox(msg4, dgDef4, título)
        If RES4 = 2 Then
            DoCmd Close
        End If
    Else
        Dim les As Database, mitable As Recordset
        Set les = dbengine.workspaces(0).databases(0)
        Set mitable = les.OpenRecordset("alumnos", DB_OPEN_TABLE)
        mitable.index = "cuenta"
        mitable.Seek "=", seconsulta
        If (mitable.NoMatch) Then 'si NO está manda un mensaje
            msg3 = "El alumno no existe!"
            dgDef3 = MB_RECANCELAR + MB_CONCONFIRMACION + MB_DEBOTOÑ1
            RES3 = MsgBox(msg3, dgDef3, título)
            If RES3 = 2 Then
                DoCmd Close
            End If
        Else
            DoCmd OpenForm "datos generales = consultar", , "seconsulta" = formularios\lee cuenta a consultar\seconsulta"
        End If
        mitable.Close
    End If
End Sub

Sub aceptar_Click ()
    If IsNull(lectra) Then
        msg4 = "Número de Cuenta incorrecto"
        dgDef4 = 48 + 0 + 0
        RES4 = MsgBox(msg4, dgDef4, título)
        If RES4 = 2 Then
            DoCmd Close
        Else
            DoCmd GoToControl "lectra"
        End If
    End Sub

Dim les As Database, mitable As Recordset
Set les = dbengine.workspaces(0).databases(0)
Set mitable = les.OpenRecordset("alumnos", DB_OPEN_TABLE)
mitable.index = "cuenta"
mitable.Seek "=", lectra
If (mitable.NoMatch) Then 'si NO está manda un mensaje
    msg3 = "El alumno no existe!"
    dgDef3 = 48 + 0 + 0
    RES3 = MsgBox(msg3, dgDef3, título)
    If RES3 = 2 Then
        DoCmd Close
    Else
        DoCmd GoToControl "lectra"
    End If
Else
    'esta en tabla alumnos
    'Verificar si esta inscrito en semestre actual
    Dim les2 As Database, mitable2 As Recordset
    Set les2 = dbengine.workspaces(0).databases(0)
    Set mitable2 = les2.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
    mitable2.index = "cuenta"
    mitable2.Seek ">", lectra
    If Not (mitable2.NoMatch) Then 'si NO está manda un mensaje
        If mitable2("semestre") = Formularios\Semestre Actual\semestre Then
            DoCmd OpenForm "consultar comprobante de inscripción", , "seconsulta" = formularios\lee cuenta a consultar\trafflectra"
            Formulario.cuenta a consultar\trafflectra = les2
        Else
            msg3 = "El alumno no está inscrito en materias del semestre actual"
            dgDef3 = 48 + 0 + 0
            RES3 = MsgBox(msg3, dgDef3, título)
            If RES3 = 2 Then
                DoCmd Close
            Else
                DoCmd GoToControl "lectra"
            End If
        End If
    Else
        msg3 = "El alumno no está inscrito en materias del semestre actual"
        dgDef3 = 48 + 0 + 0
        RES3 = MsgBox(msg3, dgDef3, título)
        If RES3 = 2 Then
            DoCmd Close
        Else
            DoCmd GoToControl "lectra"
        End If
    End If
End Sub

mitable3.Close 'Tabla inscrita
End If
mitable.Close 'Tabla alumnos

```

```

End Sub
Sub Cancelar_Click ()
DoCmd Close a_form, "Lee cuenta a consultar tira"
End Sub
Sub Form_Error (DataErr As Integer, Response As Integer)
Response = DATA_ERRCONTINUE
Const ERRORMASCARA = 2270
If DataErr = ERRORMASCARA Then
If DataErr = ERRORMASCARA Then
m1 = "Número de cuenta incorrecto"
f1 = 8 + 13 + MB_DEFROTÓN1
r1 = MsgBox(m1, 1, título)
If r1 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "lectira"
End If
End If
End If
Sub lectira_AfterUpdate ()
If IsNull(lectira) Then
msg4 = "Número de Cuenta Incorrecto"
dgDef4 = 48 + 0 + 0
RES4 = MsgBox(msg4, dgDef4, título)
If RES4 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "lectira"
End If
End If
End Sub
Dim tea As Database, mtabla As Recordset
Set tea = dbengine.workspaces(0).databases(0)
Set mtabla = tea.OpenRecordset("alumnos", DB_OPEN_TABLE)
mtabla.Index = "n_cuenta"
mtabla.Seek "=", lectira
If (mtabla.nomatch) Then 'si NO está manda un mensaje
msg3 = "El alumno no existe"
dgDef3 = 48 + 0 + 0
RES3 = MsgBox(msg3, dgDef3, título)
If RES3 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "lectira"
End If
Else 'está en tabla alumnos
' Verificar si esta inscrito en semestre actual
Dim tea2 As Database, mtabla2 As Recordset
Set tea2 = dbengine.workspaces(0).databases(0)
Set mtabla2 = tea2.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
mtabla2.Index = "cuenta"
mtabla2.Seek "=", lectira
If Not (mtabla2.nomatch) Then 'And Then 'si NO está manda un mensaje
If mtabla2(!semestre) = Formularios[Semestre Actual][!semestre] Then
DoCmd OpenForm "consultar comprobante de inscripción", , , "cuenta" = Formularios[lee cuenta a consultar tira][!lectira]
Forms[lee cuenta a consultar tira].Visible = False
Else
msg0 = "El alumno no está inscrito en materias del semestre actual"
dgDef0 = 48 + 0 + 0
RES0 = MsgBox(msg0, dgDef0, título)
If RES0 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "lectira"
End If
End If
Else
msg0 = "El alumno no está inscrito en materias del semestre actual"
dgDef0 = 48 + 0 + 0
RES0 = MsgBox(msg0, dgDef0, título)
If RES0 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "lectira"
End If
End If
mtabla2.Close 'Tabla inscrito
End If
mtabla.Close 'tabla alumnos
End Sub
Sub Retirar_Click ()
Const MB_REFCAN = 8, MB_ICONOALTO = 16
If IsNull(lectuenta) Then
m1 = "Número de cuenta incorrecto"
f1 = MB_REFCAN + MB_ICONOALTO + MB_DEFROTÓN1
r1 = MsgBox(m1, f1, título)
If r1 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "lectuenta"
End If
Else
Dim tea As Database, mtabla As Recordset
Set tea = dbengine.workspaces(0).databases(0)
Set mtabla = tea.OpenRecordset("alumnos", DB_OPEN_TABLE)
mtabla.Index = "n_cuenta"
mtabla.Seek "=", lectuenta
If mtabla.nomatch Then
m2 = "El Alumno No Existió"
f2 = MB_REFCAN + MB_ICONOALTO + MB_DEFROTÓN2
r2 = MsgBox(m2, f2, título)
If r2 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "lectuenta"
End If
End If
Else
End If
End Sub

```

```

Dim tipoObjeto As Integer, nombreObjeto As String, estadoObjeto As Integer
tipoObjeto = A_Form
nombreObjeto = "mensaje Actual"
estadoObjeto = SysCmd(SYSCMD_DIALOGOBJECTSTATE, tipoObjeto, nombreObjeto)
If estadoObjeto = 0 Then
    msg4 = "No hay mensajes actuales. Elegir mensajes y repetir la acción que desee"
    dlg4f4 = 48 + 0 + 0
    r4 = MsgBox(msg4, dlg4f4, título)
    DoCmd OpenForm "Mensajes Actual"
    DoCmd CancelEvent
    Exit Sub
Else
    Alumno10 = miTabla1[nom_alum] & " & miTabla1[ape_pat] & " & miTabla1[ape_mat]
    *****
    Dim tes2 As Database, miTabla2 As Recordset
    Set tes2 = dbengine.workspaces(0).databases(0)
    Set miTabla2 = tes2.OpenRecordset("CLASE2", DB_OPEN_TABLE)
    total = miTabla2.RecordCount
    If total = 0 Then
        m3 = "No hay grupos disponibles"
        f3 = 48 + 0 + 0
        r3 = MsgBox(m3, f3, título)
        Exit Sub
    Else
        DoCmd OpenForm "Tabla de materias y grupos"
        Me.Visible = False
    End If
    miTabla2.Close
    End If
    miTabla1.Close
    End If
    Sub cancelar_Click ()
    DoCmd Close
    End Sub
    Sub Form_Error (DataErr As Integer, Response As Integer)
    Response = DATA_ERRORCONTINUE
    Const ERRORMASCARA = 2279
    If DataErr = ERRORMASCARA Then
        m1 = "Número de cuenta incorrecto"
        f1 = 3 + 16 + MR_DIALOGCONTI
        r1 = MsgBox(m1, f1, título)
        If r1 = 2 Then
            DoCmd Close
        Else
            DoCmd GoToControl "leccuonia"
        End If
    End If
    End Sub
    Sub leccuonia_AfterUpdate ()
    Const MR_RETGAN = 5, MR_ICONOALTO = 11
    If (IsNull(leccuonia)) Then
        m1 = "Número de cuenta incorrecto"
        f1 = MR_RETGAN + MR_ICONOALTO + MR_DIALOGCONTI
        r1 = MsgBox(m1, f1, título)
        If r1 = 2 Then
            DoCmd Close
        Else
            DoCmd GoToControl "leccuonia"
        End If
    Else
        Dim tes1 As Database, miTabla As Recordset
        Set tes1 = dbengine.workspaces(0).databases(0)
        Set miTabla = tes1.OpenRecordset("Alumnos", DB_OPEN_TABLE)
        miTabla.Index = "leccuonia"
        miTabla.Seek "=", leccuonia
        If miTabla.NoMatch Then
            m2 = "El Alumno No Existe"
            f2 = MR_RETGAN + MR_ICONOALTO + MR_DIALOGCONTI
            r2 = MsgBox(m2, f2, título)
            If r2 = 2 Then
                DoCmd Close
            Else
                DoCmd GoToControl "leccuonia"
            End If
        Else
            Exit Sub
        End If
    End If
    Dim tipoObjeto As Integer, nombreObjeto As String, estadoObjeto As Integer
    tipoObjeto = A_Form
    nombreObjeto = "Mensajes Actual"
    estadoObjeto = SysCmd(SYSCMD_DIALOGOBJECTSTATE, tipoObjeto, nombreObjeto)
    If estadoObjeto = 0 Then
        msg4 = "No hay mensajes actuales. Elegir mensajes y repetir la acción que desee"
        dlg4f4 = 48 + 0 + 0
        r4 = MsgBox(msg4, dlg4f4, título)
        DoCmd OpenForm "Mensajes Actual"
        DoCmd CancelEvent
        Exit Sub
    Else
        Alumno10 = miTabla1[nom_alum] & " & miTabla1[ape_pat] & " & miTabla1[ape_mat]
        *****
        Dim tes2 As Database, miTabla2 As Recordset
        Set tes2 = dbengine.workspaces(0).databases(0)
        Set miTabla2 = tes2.OpenRecordset("CLASE2", DB_OPEN_TABLE)
        total = miTabla2.RecordCount
        If total = 0 Then
            m3 = "No hay grupos disponibles"
            f3 = 48 + 0 + 0
            r3 = MsgBox(m3, f3, título)
            Exit Sub
        Else
            DoCmd OpenForm "Tabla de materias y grupos"
            Me.Visible = False
        End If
        miTabla2.Close
        End If
        miTabla1.Close
        End If
    End If

```

```

End Sub

Sub Aceptar_Click ()
If (IsValid(cuenta)) Then
m1 = "Número de cuenta incorrecto"
f1 = 5 + 16 + 0
r1 = MsgBox(m1, f1, título)
If r1 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "lccuenta"
End If
Else
Dim tex As Database, mitabla As Recordset
Set tex = dbengine.workspaces(0).databases(0)
Set mitabla = tex.OpenRecordset("alumnos", DB_OPEN_TABLE)
mitabla.index = "n_cuenta"
mitabla.Seek "=", lccuenta
If mitabla.noMatch Then
m2 = "El Alumno No Existe!"
f2 = 5 + 16 + 0
r2 = MsgBox(m2, f2, título)
If r2 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "lccuenta"
End If
mitabla.Close
Else
'Busca si está inscrito en semestre actual
Dim tex1 As Database, mitabla1 As Recordset
Set tex1 = dbengine.workspaces(0).databases(0)
Set mitabla1 = tex1.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
mitabla1.index = "n_cuenta"
mitabla1.Seek "*" & lccuenta
If (mitabla1.noMatch) Then
m2 = "El Alumno No Está inscrito en ningún semestre!"
f2 = 5 + 16 + 0
r2 = MsgBox(m2, f2, título)
Else
Do Until mitabla1.EOF
If mitabla1!n_cuenta = Me!lccuenta And mitabla1!semestre = Formulario0!Semestre Actual Then
GoTo abre
Else
mitabla1.MoveNext
End If
Loop
m2 = "El Alumno No Está inscrito en el semestre actual!"
f2 = 48 + 0 + 0
r2 = MsgBox(m2, f2, título)
Exit Sub
abre:
DoCmd OpenForm "faja de inscripción", , "n_cuenta", formulario0!lccuenta & dar de baja inscripcion!lccuenta!
Exit Sub
End If
mitabla1.Close
End If
End If
End Sub
Sub cancelar_Click ()
DoCmd Close
End Sub
Sub form_Error (DataErr As Integer, Response As Integer)
Response = DATA_ERRORCONTINUE
Const ERRORMASCARA = 2270
If DataErr = ERRORMASCARA Then
m1 = "Número de cuenta incorrecto!"
f1 = 5 + 16 + MR_DETROTON1
r1 = MsgBox(m1, f1, título)
If r1 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "lccuenta"
End If
End If
End Sub
Sub form_Open (Cancel As Integer)
Dim tipoObjeto As Integer, nombreObjeto As String, estadoObjeto As Integer
tipoObjeto = A_form
nombreObjeto = "semestre Actual"
estadoObjeto = SysCmd(SYSCMD_OBJECTACTIVATE, tipoObjeto, nombreObjeto)
If estadoObjeto = 0 Then
msg4 = "No hay semestre actual. Elegir semestre y repetir la acción que desea!"
dgdcf4 = 48 + 0 + 0
res4 = MsgBox(msg4, dgdcf4, título)
DoCmd OpenForm "Semestre Actual"
DoCmd CancelEvent
Exit Sub
End If
End Sub
Sub lccuenta_AfterUpdate ()
If (IsValid(lccuenta)) Then
m1 = "Número de cuenta incorrecto!"
f1 = 5 + 16 + 0
r1 = MsgBox(m1, f1, título)
If r1 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "lccuenta"
End If
Else
Dim tex As Database, mitabla As Recordset
Set tex = dbengine.workspaces(0).databases(0)
Set mitabla = tex.OpenRecordset("alumnos", DB_OPEN_TABLE)
mitabla.index = "n_cuenta"
mitabla.Seek "=", lccuenta
If mitabla.noMatch Then
m2 = "El Alumno No Existe!"
f2 = 5 + 16 + 0

```

```

r2 = MsgBox(mZ, fZ, titulo)
If r2 = 2 Then
    DoCmd Close
Else
    DoCmd GoToControl "Teccuenta"
End If
mitabla Close
Else
    "Busca si está inscrito en semestre actual"
Dim fca1 As Database, mitabla1 As Recordset
Set fca1 = dbengine.workspaces(0).databases(0)
Set mitabla1 = fca1.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
mitabla1.Index = "m_cuenta"
mitabla1.Seek "=", fccuenta
If (mitabla1.NoMatch) Then
    mZ = "El Alumno No Está inscrito en ningún semestre!"
    fZ = 5 + f0 + 0
    rZ = MsgBox(mZ, fZ, titulo)
Else
    Do Until mitabla1.EOF
    If mitabla1(f_cuenta) = Me!f_cuenta1 And mitabla1(f_semestre) = Formulario1[Semestre Actual][f_semestre] Then
        GoTo sde1
    Else
        mitabla1.MoveNext
    End If
    Loop
    mZ = "El Alumno No Está inscrito en el semestre actual!"
    fZ = 48 + 0 + 0
    rZ = MsgBox(mZ, fZ, titulo)
    Exit Sub
sde1:
    DoCmd OpenForm "Faja de inscripción...", "In_cuenta" & Formulario1[f_cuenta] & " de faja inscripción" & f_cuenta1
End If
mitabla1.Close
End If
End Sub

Sub Botón1_Click ()
Const MR_REDCAN = 5, MR_CONOALTO = 16
If (IsNull(fccuenta)) Then
    m1 = "Número de cuenta incorrecto!"
    f1 = MR_REDCAN + MR_CONOALTO + MR_DEFBOTON1
    r1 = MsgBox(m1, f1, titulo)
    If r1 = 2 Then
        DoCmd Close
    Else
        DoCmd GoToControl "Teccuenta"
    End If
Else
    "TRABAJON PARA VALIAR EL NUMERO DE cuenta"
    uno = Left(fccuenta, 1)
    tres = Left(Right(fccuenta, 6), 1)
    cinco = Left(Right(fccuenta, 4), 1)
    siete = Left(Right(fccuenta, 2), 1)
    dos = Left(Right(fccuenta, 7), 1)
    cuatro = Left(Right(fccuenta, 9), 1)
    seis = Left(Right(fccuenta, 3), 1)
    parte1 = (Val(uno) + Val(tres) + Val(cinco) + Val(siete)) * 7
    parte2 = (Val(dos) + Val(cuatro) + Val(seis)) * 3
    suma = parte1 + parte2
    resultado = (suma Mod 10)
    If Val(resultado) <> Right(fccuenta, 1) Or (fccuenta = Null) Then
        mZ = "Número de cuenta incorrecto!"
        fZ = MR_REDCAN + MR_CONOALTO + MR_DEFBOTON1
        rZ = MsgBox(mZ, fZ, titulo)
        If rZ = 2 Then
            DoCmd Close
        Else
            DoCmd GoToControl "Teccuenta"
        End If
    Else
Dim fca As Database, mitabla As Recordset
Set fca = dbengine.workspaces(0).databases(0)
Set mitabla = fca.OpenRecordset("alumnos", DB_OPEN_TABLE)
mitabla.Index = "n_cuenta"
mitabla.Seek "=", fccuenta
If mitabla.NoMatch Then
    DoCmd OpenForm "Datos Generales", "In_cuenta" & Formulario1[f_cuenta] & " de datos"
Else
    mZ = "El Número de cuenta ya existe!"
    fZ = MR_REDCAN + MR_CONOALTO + MR_DEFBOTON2
    rZ = MsgBox(mZ, fZ, titulo)
    If rZ = 2 Then
        DoCmd Close
    Else
        DoCmd GoToControl "Teccuenta"
    End If
End If
End If
End Sub

Sub Cancelar_Click ()
DoCmd Close
End Sub

Sub Form1_Deactivate ()
DoCmd Close
End Sub

Sub Form_Error (DataErr As Integer, Response As Integer)
Response = DATA_ERRORCONTINUI
Const ERRORESMASCARA = 2271
If DataErr = ERRORESMASCARA Then
    m1 = "Número de cuenta incorrecto!"
    f1 = 3 + 16 + MR_DEFBOTON1
    r1 = MsgBox(m1, f1, titulo)
    If r1 = 2 Then

```

```

DoCmd Close
Else
    DoCmd GoToControl "lccuenta"
End If
End If
End Sub
Sub lccuenta_AfterUpdate ()
Const MB_REFCAN = 5, MB_ICONOALTO = 16
If (IsNull(lccuenta)) Then
m1 = "Número de cuenta incorrecto"
f1 = MB_REFCAN + MB_ICONOALTO + MB_DEBTOTÓN1
r1 = MsgBox(m1, f1, título)
If r1 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "lccuenta"
End If
Else
'FUNCION PARA VALIDAR EL NUMERO DE cuenta
uno = Left(lccuenta, 1)
tres = Left(Right(lccuenta, 6), 1)
cinco = Left(Right(lccuenta, 4), 1)
siete = Left(Right(lccuenta, 2), 1)
dos = Left(Right(lccuenta, 7), 1)
cuatro = Left(Right(lccuenta, 5), 1)
seis = Left(Right(lccuenta, 3), 1)
parte1 = (Val(uno) + Val(tres) + Val(cinco) + Val(siete)) * 7
parte2 = (Val(dos) + Val(cuatro) + Val(seis)) * 3
suma = parte1 + parte2
resultado = (suma Mod 10)
If Val(resultado) <= Right(lccuenta, 1) Or (lccuenta = Null) Then
m2 = "Número de cuenta incorrecto"
f2 = MB_REFCAN + MB_ICONOALTO + MB_DEBTOTÓN1
r2 = MsgBox(m2, f2, título)
If r2 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "lccuenta"
End If
Else
Dim tes As Database, miTabla As Recordset
Set tes = dberpgete.workspaces(0) databases(0)
Set miTabla = tes.OpenRecordset("alumnos", DIR_OPEN_TABLE)
miTabla.index = "n_cuenta"
miTabla.Seek "a", lccuenta
If miTabla.NoMatch Then
miTabla.Close
Formulario.lccuenta & "Inscribir" Visible = False
DoCmd OpenForm "Datos Generales", , "In_cuenta]= formulario[lcc cuenta a inscribir]]lccuenta]"
Else
m3 = "El número de cuenta ya existe"
f3 = MB_REFCAN + MB_ICONOALTO + MB_DEBTOTÓN2
r3 = MsgBox(m3, f3, título)
If r3 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "lccuenta"
End If
End If
End If
End Sub

Sub Botón1_Click ()
Const MB_ACEPTAR = 0, MB_ACEPTARCANCELAR = 1 'define botones.
Const MB_SINOCANCELAR = 3, MB_SINO = 4, MB_REFCAN = 5
Const MB_ICONOALTO = 16, MB_ICONOINTERROGACION = 32 'Define iconos.
Const MB_ICONOADMIRACION = 48
Const MB_DEBTOTÓN2 = 256, ID1 = 0, ID10 = 7, IDCAN = 5, MB_DEBTOTÓN1 = 0 'Define etras.
If (IsNull(lccmodifica)) Then
msg4 = "Número de cuenta incorrecto"
dgdDef4 = MB_REFCAN + MB_ICONOADMIRACION + MB_DEBTOTÓN1
RES4 = MsgBox(msg4, dgdDef4, título)
If RES4 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "lccmodifica"
End If
Else
Dim tes As Database, miTabla As Recordset
Set tes = dberpgete.workspaces(0) databases(0)
Set miTabla = tes.OpenRecordset("alumnos", DIR_OPEN_TABLE)
miTabla.index = "n_cuenta"
miTabla.Seek "a", lccmodifica
If miTabla.NoMatch Then 'si NO está manda un mensaj.
msg5 = "El alumno no existe"
dgdDef5 = 48 + 0 + 0
RES5 = MsgBox(msg5, dgdDef5, título)
If RES5 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "lccmodifica"
End If
Else
DoCmd OpenForm "Cambiar datos Generales", , "In_cuenta]= formulario[lcc cuenta a modificar]]lccmodifica]"
End If
miTabla.Close
End If
End Sub
Sub Cancelar_Click ()
DoCmd Close
End Sub
Sub Form_Error (DataErr As Integer, Response As Integer)
Response = DATA_ERRCONTINUE
Const ERRORAMASCARA = 2279
If DataErr = ERRORAMASCARA Then
m1 = "Número de cuenta incorrecto"
f1 = 5 + 16 + MB_DEBTOTÓN1
r1 = MsgBox(m1, f1, título)

```

```

If r1 = 2 Then
    DoCmd Close
Else
    DoCmd GoToControl "tecmofila"
End If
End If
End Sub

Sub tecmodifica_AfterUpdate ()
Const MB_ACEPTAR = 0, MB_ACEPTARCANCELAR = 1 'Define botones
Const MB_SINOCANCELAR = 3, MB_SINO = 4, MB_REFCAN = 5
Const MB_ICONOCALTO = 10, MB_ICONONINTERROGACION = 32 'Define iconos
Const MB_ICONOADMIRACION = 48
Const MB_DEFBOTON2 = 256, ID1 = 6, IDno = 7, IDCAN = 8, MB_DEFBOTON1 = 0 'Define otros
If IsNull(tecmodifica) Then
msg4 = "Número de cuenta incorrecto!"
dgbf4 = MB_REFCAN + MB_ICONOADMIRACION + MB_DEFBOTON1
RES4 = MsgBox(msg4, dgbf4, titulo)
If RES4 = 2 Then
    DoCmd Close
Else
    DoCmd GoToControl "tecmofila"
End If
Else
Dim tex As Database, mtabla As Recordset
Set tex = db.engine.workspaces(0).databases(0)
Set mtabla = tex.OpenRecordset("alumnos", DB_OPEN_TABLE)
mtabla.Index = "n_cuenta"
mtabla.Seek "=", tecmodifica
If mtabla.NoMatch Then 'si NO está manda un mensaje
msg3 = "El alumno no existe!"
dgbf3 = MB_REFCAN + MB_ICONOADMIRACION + MB_DEFBOTON1
RES3 = MsgBox(msg3, dgbf3, titulo)
If RES3 = 2 Then
    DoCmd Close
Else
    DoCmd GoToControl "tecmofila"
End If
Else
DoCmd OpenForm "Cambiar datos Generales", , In_cuenta = formularios\Ive cuenta a modificar\Itecmofila"
End If
mtabla.Close
End If
End Sub

Sub Aceptar_Click ()
Const MB_ACEPTAR = 0, MB_ACEPTARCANCELAR = 1 'Define botones
Const MB_SINOCANCELAR = 4, MB_SINO = 4, MB_REFCAN = 5
Const MB_ICONOCALTO = 10, MB_ICONONINTERROGACION = 32 'Define iconos
Const MB_ICONOADMIRACION = 48
Const MB_DEFBOTON2 = 256, ID1 = 6, IDno = 7, IDCAN = 8, MB_DEFBOTON1 = 0 'Define otros
If IsNull(tecuenta) Then
msg4 = "Número de cuenta incorrecto!"
dgbf4 = MB_REFCAN + MB_ICONOADMIRACION + MB_DEFBOTON1
RES4 = MsgBox(msg4, dgbf4, titulo)
If RES4 = 2 Then
    DoCmd Close
Else
    DoCmd GoToControl "tecuenta"
End If
Else
Dim tex As Database, mtabla As Recordset
Set tex = db.engine.workspaces(0).databases(0)
Set mtabla = tex.OpenRecordset("alumnos", DB_OPEN_TABLE)
mtabla.Index = "n_cuenta"
mtabla.Seek "=", tecuenta
If mtabla.NoMatch Then 'si NO está manda un mensaje
msg3 = "El alumno no existe!"
dgbf3 = MB_REFCAN + MB_ICONOADMIRACION + MB_DEFBOTON1
RES3 = MsgBox(msg3, dgbf3, titulo)
If RES3 = 2 Then
    DoCmd Close
Else
    DoCmd GoToControl "tecuenta"
End If
Else
DoCmd OpenForm "Baja definitiva del alumno", , In_cuenta = formularios\Ive cuenta por baja definitiva\Itecuenta"
End If
mtabla.Close
End If
End Sub

Sub cancelar_Click ()
DoCmd Close
End Sub

Sub Form_Error (Button As Integer, Response As Integer)
Response = DATA_ERRORCONTINUE
Const ERRORMANCARRA = 2279
If DataErr = ERRORMANCARRA Then
m1 = "Número de cuenta incorrecto!"
f1 = 9 + 16 + MB_DEFBOTON1
r1 = MsgBox(m1, f1, titulo)
If r1 = 2 Then
    DoCmd Close
Else
    DoCmd GoToControl "tecuenta"
End If
End If
End Sub

Sub tecuenta_AfterUpdate ()
Const MB_ACEPTAR = 0, MB_ACEPTARCANCELAR = 1 'Define botones
Const MB_SINOCANCELAR = 3, MB_SINO = 4, MB_REFCAN = 5
Const MB_ICONOCALTO = 10, MB_ICONONINTERROGACION = 32 'Define iconos
Const MB_ICONOADMIRACION = 48
Const MB_DEFBOTON2 = 256, ID1 = 6, IDno = 7, IDCAN = 8, MB_DEFBOTON1 = 0 'Define otros
If IsNull(tecuenta) Then
msg4 = "Número de cuenta incorrecto!"
dgbf4 = MB_REFCAN + MB_ICONOADMIRACION + MB_DEFBOTON1
RES4 = MsgBox(msg4, dgbf4, titulo)
If RES4 = 2 Then
    DoCmd Close

```



```

Else
  DoCmd GoToControl "leecuenta"
End If
Else
Dim lca As Database, mifabla As Recordset
Set lca = dbeengine.workspaces(0).databases(0)
Set mifabla = lca.OpenRecordset("alumnos", DB_OPEN_TABLE)
mifabla.index = "n_cuenta"
mifabla.Seek "=", leecuenta
If mifabla.NoMatch Then 'si NO está manda un mensaje
  msg$ = "El alumno no existe!"
  dgDef5 = MB_REFCAN + MB_ICONOADMIRACION + MR_DEFBOTÓN1
  RES3 = MsgBox(msg$, dgDef5, título)
  If RES3 = 2 Then
    DoCmd Close
  Else
    DoCmd GoToControl "leecuenta"
  End If
Else
  DoCmd OpenForm "Baja definitiva del alumno", , "In_cuenta]=formularios|[lee cuenta para baja definitiva|[leecuenta]"
End If
mifabla.Close
End If
End Sub

Sub Botón1_Click ()
Const MB_REFCAN = 5, MB_ICONOALTO = 10
If (IsNull(leconsultaplan)) Then
  m1 = "Número de cuenta incorrecto!"
  f1 = MR_REFCAN + MB_ICONOALTO + MR_DEFBOTÓN1
  r1 = MsgBox(m1, f1, título)
  If r1 = 2 Then
    DoCmd Close
  Else
    DoCmd GoToControl "leconsultaplan"
  End If
Else
Dim lca As Database, MITABLA As Recordset
Set lca = dbeengine.workspaces(0).databases(0)
Set MITABLA = lca.OpenRecordset("alumnos", DB_OPEN_TABLE)
MITABLA.index = "n_cuenta"
MITABLA.Seek "=", leconsultaplan
If MITABLA.NoMatch Then
  m2 = "El Alumno No Existe!"
  f2 = MR_REFCAN + MB_ICONOALTO + MR_DEFBOTÓN2
  r2 = MsgBox(m2, f2, título)
  If r2 = 2 Then
    DoCmd Close
  Else
    DoCmd GoToControl "leconsultaplan"
  End If
Else
Dim lca2 As Database, MITABLA2 As Recordset
Set lca2 = dbeengine.workspaces(0).databases(0)
Set MITABLA2 = lca2.OpenRecordset("mat_plan", DU_OPEN_TABLE)
MITABLA2.index = "n_cuenta"
MITABLA2.Seek "=", leconsultaplan
If MITABLA2.NoMatch Then
  m2 = "El Alumno No Tiene Plan de Estudios!"
  f2 = MR_REFCAN + MB_ICONOALTO + MR_DEFBOTÓN2
  r2 = MsgBox(m2, f2, título)
  If r2 = 2 Then
    DoCmd Close
  Else
    DoCmd GoToControl "leconsultaplan"
  End If
Else
  DoCmd OpenForm "consulta d. plan de estudios", , "In_cuenta]=formularios|[lee cuenta para consultar plan|[leconsultaplan]"
  Forms![lee cuenta para consultar plan].Visible = False
End If
End If
End Sub
MITABLA.Close
MITABLA2.Close
End Sub
Sub Cancelar_Click ()
DoCmd Close
End Sub
Sub Form_Error (DataErr As Integer, Response As Integer)
Response = DATA_ERRCONTINUE
Const ERRORMASCARA = 2273
If DataErr = ERRORMASCARA Then
  m1 = "Número de cuenta incorrecto!"
  f1 = 5 + 10 + MR_DEFBOTÓN1
  r1 = MsgBox(m1, f1, título)
  If r1 = 2 Then
    DoCmd Close
  Else
    DoCmd GoToControl "leconsultaplan"
  End If
End If
End Sub
Sub leconsultaplan_AfterUpdate ()
Const MB_REFCAN = 5, MB_ICONOALTO = 10
If (IsNull(leconsultaplan)) Then
  m1 = "Número de cuenta incorrecto!"
  f1 = MR_REFCAN + MB_ICONOALTO + MR_DEFBOTÓN1
  r1 = MsgBox(m1, f1, título)
  If r1 = 2 Then
    DoCmd Close
  Else
    DoCmd GoToControl "leconsultaplan"
  End If
Else
Dim lca As Database, MITABLA As Recordset
Set lca = dbeengine.workspaces(0).databases(0)
Set MITABLA = lca.OpenRecordset("alumnos", DB_OPEN_TABLE)
MITABLA.index = "n_cuenta"
MITABLA.Seek "=", leconsultaplan

```

```

If MITABLA.nomatch Then
m2 = "El Alumno No Existe!"
f2 = MB_REFCAN + MB_ICONOALTO + MB_DEFBOTÓN2
r2 = MsgBox(m2, f2, título)
If r2 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "leeconsultaplan"
End If
Else
Dim tex2 As Database, MITABLA2 As Recordset
Set tex2 = dbe.engine.workspaces(0).databases(0)
Set MITABLA2 = tex2.OpenRecordset("mat_plan", DB_OPEN_TABLE)
MITABLA2.index = "n_cuenta"
MITABLA2.Seek "=", leeconsultaplan
If MITABLA2.nomatch Then
m2 = "El Alumno No Tiene Plan de Estudios!"
f2 = MB_REFCAN + MB_ICONOALTO + MB_DEFBOTÓN2
r2 = MsgBox(m2, f2, título)
If r2 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "leeconsultaplan"
End If
Else
DoCmd OpenForm "consulta de plan de estudios", , "n_cuenta" = (formulario) lee cuenta para consultar plan | leeconsultaplan
Form!lee cuenta para consultar plan.Visible = False
End If
End If
End Sub

Sub Boton1_Click ()
Const MB_REFCAN = 5, MB_ICONOALTO = 16
If (Not(IsNumeric(modificaplan))) Then
f1 = "Número de cuenta incorrecto!"
f1 = MB_REFCAN + MB_ICONOALTO + MB_DEFBOTÓN1
r1 = MsgBox(m1, f1, título)
If r1 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "leemodificaplan"
End If
Else
Dim tex As Database, mitabla As Recordset
Set tex = dbe.engine.workspaces(0).databases(0)
Set mitabla = tex.OpenRecordset("alumno", DB_OPEN_TABLE)
mitabla.index = "n_cuenta"
mitabla.Seek "=", leemodificaplan
If mitabla.nomatch Then
m2 = "El Alumno No Existe!"
f2 = MB_REFCAN + MB_ICONOALTO + MB_DEFBOTÓN2
r2 = MsgBox(m2, f2, título)
If r2 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "leemodificaplan"
End If
Else
Dim tex2 As Database, mitabla2 As Recordset
Set tex2 = dbe.engine.workspaces(0).databases(0)
Set mitabla2 = tex2.OpenRecordset("mat_plan", DB_OPEN_TABLE)
mitabla2.index = "n_CUENTA"
mitabla2.Seek "=", leemodificaplan
If mitabla2.nomatch Then
m2 = "El Alumno No Tiene Plan de Estudios!"
f2 = MB_REFCAN + MB_ICONOALTO + MB_DEFBOTÓN2
r2 = MsgBox(m2, f2, título)
If r2 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "leemodificaplan"
End If
Else
DoCmd OpenForm "Cambiar plan de estudios", , "n_cuenta" = (formulario) lee cuenta para modificar plan | leemodificaplan
Form!lee cuenta para modificar plan.Visible = False
End If
mitabla2.Close
End If
mitabla.Close
End If
End Sub
Sub Cancelar_Click ()
DoCmd Close
End Sub
Sub Form_Error (DataErr As Integer, Response As Integer)
Response = DATA_ERRCONTINUE
Const ERRORMASCARA = 2271
If DataErr = ERRORMASCARA Then
m1 = "Número de cuenta incorrecto"
f1 = 5 + 11 + MB_DEFBOTÓN1
r1 = MsgBox(m1, f1, título)
If r1 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "leemodificaplan"
End If
End If
End Sub
Sub leemodificaplan_AfterUpdate ()
Const MB_REFCAN = 5, MB_ICONOALTO = 16
If (Not(IsNumeric(modificaplan))) Then
m1 = "Número de cuenta incorrecto!"
f1 = MB_REFCAN + MB_ICONOALTO + MB_DEFBOTÓN1
r1 = MsgBox(m1, f1, título)
If r1 = 2 Then
DoCmd Close
Else

```

```

DoCmd GoToControl "ecmodificaplan"
End If
Else
Dim tex As Database, mitabla As Recordset
Set tex = dbengine.workspace(0).databases(0)
Set mitabla = tex.OpenRecordset("alumno", DB_OPEN_TABLE)
mitabla.index = "n_cuenta"
mitabla.Seek "=", locmodificaplan
If mitabla.NoMatch Then
m2 = "¿El Alumno No Existe?"
f2 = MR_REFCAN + MR_ICONOALTO + MR_DETROTÓN2
r2 = MsgBox(m2, f2, titulo)
If r2 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "kanmodificaplan"
End If
Else
Dim tex2 As Database, mitabla2 As Recordset
Set tex2 = dbengine.workspace(0).databases(0)
Set mitabla2 = tex2.OpenRecordset("mat_plan", DB_OPEN_TABLE)
mitabla2.index = "n_cuenta"
mitabla2.Seek "=", locmodificaplan
If mitabla2.NoMatch Then
m2 = "¿El Alumno No Tiene Plan de Estudios?"
f2 = MR_REFCAN + MR_ICONOALTO + MR_DETROTÓN2
r2 = MsgBox(m2, f2, titulo)
If r2 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "ecmodificaplan"
End If
Else
DoCmd OpenForm "Cambiar plan de estudios", , "[n_cuenta]= formularios\lee cuenta para modificar plan\ecmodificaplan"
Forms!lee cuenta para modificar plan.Visible = False
End If
mitabla2.Close
End If
mitabla.Close
End If
End Sub

Sub Botón_Click ()
Const MR_REFCAN = 5, MR_ICONOALTO = 10
If (IsNull(iceplan)) Then
m1 = "Número de cuenta incorrecto!"
f1 = MR_REFCAN + MR_ICONOALTO + 0
r1 = MsgBox(m1, f1, titulo)
If r1 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "iceplan"
End If
Else
Dim tex As Database, mitabla As Recordset
Set tex = dbengine.workspace(0).databases(0)
Set mitabla = tex.OpenRecordset("alumno", DB_OPEN_TABLE)
mitabla.index = "n_cuenta"
mitabla.Seek "=", iceplan
If mitabla.NoMatch Then
m2 = "¿El Alumno No Existe?"
f2 = MR_REFCAN + MR_ICONOALTO + 0
r2 = MsgBox(m2, f2, titulo)
If r2 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "iceplan"
End If
Else
DoCmd OpenForm "Alta del Plan de estudios", , "[n_cuenta]= formularios\lee cuenta para plan\iceplan"
Forms!lee cuenta para plan.Visible = False
End If
mitabla.Close
End If
End Sub

Sub Cancelar_Click ()
DoCmd Close
End Sub

Sub Form_Error (DataErr As Integer, Response As Integer)
Response = DATA_ERRCONTINUE
Const ERRORMASCARA = 2273
If DataErr = ERRORMASCARA Then
m1 = "Número de cuenta incorrecto!"
f1 = 5 + 16 + MR_DETROTÓN1
r1 = MsgBox(m1, f1, titulo)
If r1 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "iceplan"
End If
End If

Sub iceplan_AfterUpdate ()
Const MR_REFCAN = 5, MR_ICONOALTO = 10
If (IsNull(iceplan)) Then
m1 = "Número de cuenta incorrecto!"
f1 = MR_REFCAN + MR_ICONOALTO + 0
r1 = MsgBox(m1, f1, titulo)
If r1 = 2 Then
DoCmd Close
Else
DoCmd GoToControl "iceplan"
End If
Else
Dim tex As Database, mitabla As Recordset
Set tex = dbengine.workspace(0).databases(0)
Set mitabla = tex.OpenRecordset("alumno", DB_OPEN_TABLE)
mitabla.index = "n_cuenta"
mitabla.Seek "=", iceplan

```

```

If mitabla nomatch Then
m2 = "El Alumno No Existe"
f2 = AH_RFV\AN + MR_{C\NOA}TC + 0
r2 = MsgBox(m2, f2, titulo)
If r2 = 2 Then
    EscMd Close
Else
    EscMd GoTo Control "Aceptar"
End If
Else
    EscMd OpenForm "Abra del Plan de Estudios", , , Tu_cuenta = formularios\Fee cuenta para plan\|Aceptar"
FormDef Fee cuenta para plan Visible = False
End If
End If
mitabla Close
End Sub

Sub Aceptar_provisional_Click ()
If IsNull(Me!facta) Then
m1 = "Número de Acta incorrecto"
f1 = 48 + 0 + 0
r1 = MsgBox(m1, f1, titulo)
Exit Sub
End If
Dim tes As Usuario, mitabla As Recordset
Set tes = db.OpenWorkspace(0) databases(0)
Set mitabla = tes.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
mitabla index = "ACTA"
mitabla Seek "=", Me!facta
If (mitabla nomatch) Then
m1 = "El número de Acta no existe"
f1 = 48 + 0 + 0
r1 = MsgBox(m1, f1, titulo)
mitabla Close
Exit Sub
Else
    If mitabla[s_mestre] = formularios[s_mestre_actual][s_mestre] Then
        EscMd OpenForm "Consulta de Acta específica", , , "Acta"=formularios\Fee foto a consultar\|Acta"
    Else
        m1 = "El número de Acta no existe para el semestre actual"
        f1 = 48 + 0 + 0
        r1 = MsgBox(m1, f1, titulo)
        mitabla Close
        Exit Sub
    End If
    mitabla Close
End If
End Sub

Sub Cancelar_Click ()
EscMd Close
End Sub

Sub Aceptar_provisional_Click ()
If IsNull(Me!facta) Then
m1 = "Número de Acta incorrecto"
f1 = 48 + 0 + 0
r1 = MsgBox(m1, f1, titulo)
Exit Sub
End If
Dim tes As Usuario, mitabla As Recordset
Set tes = db.OpenWorkspace(0) databases(0)
Set mitabla = tes.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
mitabla index = "ACTA"
mitabla Seek "=", Me!facta
If Not (mitabla nomatch) Then repetido
m1 = "El número de acta ya existe, por lo que sera reemplazado", "Vea Continuar"
f1 = 32 + 4 + 0
r1 = MsgBox(m1, f1, titulo)
If r1 = 6 Then
    GoTo reemplazo
Else
    SendKeys "{esc}", True
    EscMd Close
    mitabla Close
    Exit Sub
End If
End If
Else
Dim tipoObjeto As Integer, nombresObjeto As String, ciudadesObjeto As Integer
tipoObjeto = A_form
nombresObjeto = "Semestre Actual"
ciudadesObjeto = srsCmd("SELECT * FROM ESTADIST", tipoObjeto, nombresObjeto)
If ciudadesObjeto = 0 Then
mmsg4 = "No hay semestre actual. Ejecutar semestre y ejecutar la acción que desea"
dgk4f4 = 0 + 0 + 0
rmsg4 = MsgBox(mmsg4, dgk4f4, titulo)
EscMd OpenForm "Semestre Actual"
Exit Sub
Else
Dim test As Database, mitabla1 As Recordset
Set test = db.OpenWorkspace(0) databases(0)
Set mitabla1 = test.OpenRecordset("inscrito", DB_OPEN_TABLE)
mitabla1 index = "sem"
mitabla1 Seek "=", formularios\Semestre Actual\|s_mestre]
If (mitabla1 nomatch) Then
    Si no hay alumnos inscritos en el semestre actual
    dgk4f1 = 0 + 48 + 0
    RMSG1 = MsgBox(mmsg1, dgk4f1, titulo)
    Else
        GoTo reemplazo
    End If
    mitabla1 Close
End If
mitabla Close
End If
EscMplaza.
EscMd OpenForm "Renovar actas"
Exit Sub
End Sub
Sub Cancelar_Click ()

```

```

DoCmd Close
Formulario1([Yooco semestral],[Visible = True
End Sub
Sub Form_Open (Cancel As Integer)
Dim tex As Database, mitabla As Recordset
Set tex = dbrngine.workspaces(0).databases(0)
Set mitabla = tex.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
mitabla.Seek = "ACTA"
Mitabla.Seek = "1000"
If (mitabla.nomatch) Then 'Si no hay provisionales, crea el primero
Me!facta) = "1000"
Else 'Buscamos el último número para generar el siguiente
mitabla.MoveLast
numero = (Val(mitabla!acta) + 1)
Me!facta) = numero
End If
mitabla.Close
End Sub

Sub Aceptar_provisional_Click ()
If IsNull(Me!facta) Then
m) = "Número de Acta incorrecto!"
f) = 48 + 0 + 0
r) = MsgBox(m), f), título)
Exit Sub
End If
Dim tex As Database, mitabla As Recordset
Set tex = dbrngine.workspaces(0).databases(0)
Set mitabla = tex.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
mitabla.Seek = "ACTA"
mitabla.Seek = "Me!facta)
If (mitabla.nomatch) Then
m) = "El número de Acta no existe!"
f) = 48 + 0 + 0
r) = MsgBox(m), f), título)
mitabla.Close
Exit Sub
Else
If mitabla!semestre) = formulario1(semestre actual))semestre) Then
DoCmd OpenForm "Cambio de Actas", , , [acta]=formulario1(lee folio para cambiar acta))facta)
Else
m) = "El número de Acta no existe para el semestre actual!"
f) = 48 + 0 + 0
r) = MsgBox(m), f), título)
mitabla.Close
Exit Sub
End If
mitabla.Close
End Sub
Sub Cancelar_Click ()
DoCmd Close
End Sub

Sub Aceptar_provisional_Click ()
If IsNull(Me!facta) Then
m) = "Número de Acta incorrecto!"
f) = 48 + 0 + 0
r) = MsgBox(m), f), título)
Exit Sub
End If
Dim tex As Database, mitabla As Recordset
Set tex = dbrngine.workspaces(0).databases(0)
Set mitabla = tex.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
mitabla.Seek = "ACTA"
mitabla.Seek = "Me!facta)
If (mitabla.nomatch) Then
m) = "El número de Acta no existe!"
f) = 48 + 0 + 0
r) = MsgBox(m), f), título)
mitabla.Close
Exit Sub
Else
If mitabla!semestre) = formulario1(semestre actual))semestre) Then
DoCmd OpenForm "Eliminar Acta", , , [acta]=formulario1(lee folio para eliminar acta))acta)
Else
m) = "El número de Acta no existe para el semestre actual!"
f) = 48 + 0 + 0
r) = MsgBox(m), f), título)
mitabla.Close
Exit Sub
End If
mitabla.Close
End Sub
Sub Cancelar_Click ()
DoCmd Close
End Sub

Sub Aceptar_Click ()
DoCmd OpenForm "Inscripción Actual"
End Sub
Sub acta_AfterUpdate ()
Dim tex As Database, mitabla As Recordset
Set tex = dbrngine.workspaces(0).databases(0)
Set mitabla = tex.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
mitabla.Seek = "ACTA"
mitabla.Seek = "0000"
If (mitabla.nomatch) Then 'Si no hay provisionales, crea el primero
Me!ACTA) = "0000"
Else 'Buscamos el último número para generar el siguiente
mitabla.MoveLast
numero = (Val(mitabla!ACTA) + 1)
Me!ACTA) = numero
End If
mitabla.Close
End Sub
Sub Botón172_Click ()
DoCmd RunMacro "consulta de actualizacion"

```

```

End Sub
Sub Validacion_AfterUpdate ()
If Val(Me!fecha) < 0 Or Val(Me!edad) = 0 Then
Me!aprobada = "N"
Else
Me!aprobada = "A"
End If
End Sub
Sub Validacion_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "La calificación debe coincidir con un valor de la lista"
dgdef1 = 0 + 48 + 0
R241 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub Cancelar_inscripcion_Click ()
DoCmd Close
DoCmd Close A_form, "se cuentas a dar de alta materias"
End Sub
Sub Form_AfterUpdate ()
Me!Seleccionar grupo! Required
End Sub
Sub Form_Close ()
DoCmd RunMacro "macro de altas genero actas"
End Sub
Sub Otra_acta_Click ()
DoCmd Close A_form, "LEENAR ACTAS"
DoCmd Close A_form, "se folio para acta"
DoCmd OpenForm "Los folio para acta"
End Sub
Sub Precede_inscripcion_Click ()
Me!Seleccionar grupo! Enabled = False
DoCmd OpenForm "inscripcion actual"
End Sub
Sub Salir_Click ()
DoCmd Close
DoCmd Close A_form, "se folio para acta"
Formulario5!Formulario5.Visible = True
End Sub
Sub Seleccionar_grupo_AfterUpdate ()
DoCmd ApplyFilter "para actas"
DoCmd GoToControl "Calificacion"
Me!Seleccionar Materia! Enabled = False
Me!Seleccionar grupo! Enabled = False
End Sub
Sub Seleccionar_grupo_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El grupo debe coincidir con un valor de la lista"
dgdef1 = 0 + 48 + 0
R241 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub Seleccionar_Materia_AfterUpdate ()
Dim temp As Variant
Me!Seleccionar grupo! Enabled = True
Me!Seleccionar grupo! Required
temp = activarcontrol("detalle", False)
End Sub
Sub Seleccionar_Materia_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El nombre de la Materia debe coincidir con un valor de la lista"
dgdef1 = 0 + 48 + 0
R241 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub Ver_lista_de_materias_Click ()
Dim m1() As Database, m1tabla As Recordset
Set m1() = dbeqmc.workspaces(0).databases(0)
Set m1tabla = m1().OpenRecordset("DESCRIPCION", DIR_OPEN_TABLE)
m1tabla.Index = "NUMCuenta" & " Establece el indice activo"
m1tabla.Seek "=", Me!cuenta1
If (m1tabla.NoMatch) Then
m1 = "El Alumno no está inscrito en ningún material"
f1 = 48 + 0 + 0
r1 = MsgBox(m1, f1, titulo)
Else
DoCmd OpenForm "Ver lista de Materias", , , "n_cuenta1= formularios\Tabla de materias y grupos\CUENTA1"
End If
m1tabla.Close
End Sub
Sub Form_Open (Cancel As Integer)
Me!Imprimir! Enabled = False
End Sub
Sub Imprimir_Click ()
On Error GoTo Err_imprimir_Click
DoCmd OpenReport "materias por profesor", A_preview, True, "formularios\materias impartidas\profesor\seleccionar profesor"
Err_imprimir_Click:
Exit Sub
Err_imprimir_Click:
msg = "se canceló impresión"
f1 = 48 + 0 + 0
r1 = MsgBox(msg, f1, titulo)
Resume Err_imprimir_Click
End Sub
Sub Seleccionar_Profesor_AfterUpdate ()
Dim tes As Database, m1tabla As Recordset
Set tes = dbeqmc.workspaces(0).databases(0)
Set m1tabla = tes.OpenRecordset("CLASIF", DIR_OPEN_TABLE)
m1tabla.Index = "PROFESOR"
m1tabla.Seek "=", Me!Seleccionar profesor
If (m1tabla.NoMatch) Then
m1 = "Profesor sin materias para ningún semestre"
f1 = 48 + 0 + 0
r1 = MsgBox(m1, f1, titulo)
m1tabla.Close
Me!Imprimir! Enabled = False
Exit Sub
Else
Dim Util m1tabla cod
If m1tabla.Fields("se") = formularios\Semestre Actual\Exmedrecl Then
GoTo abrir

```

```

Exit Do
Else
    miTabla.MoveNext
End If
Loop
    m1 = "Profesor sin materias para el semestre actual"
    f1 = 48 + 0 + 0
    r1 = MsgBox(m1, f1, f1ulo)
    miTabla.Close
    Me.Refresh.Enabled = False
Exit Sub

Exit Sub
End If
Abrir:
    Me.Refresh.Enabled = True
    DoCmd ApplyFilter "Para consultar materias impartidas/profesor"
    Exit Sub

End Sub
Sub Seleccionar_Profesor_MostrarMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
DoCmd OpenForm "acordeado"
End Sub
Sub Seleccionar_Profesor_NoImpartidas (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "El nombre del Profesor debe coincidir con un valor de la lista"
dqdef1 = 0 + 48 + 0
RES1 = MsgBox(msg1, dqdef1, f1ulo)
End Sub

Sub Aceptar_Click ()
DoCmd OpenForm "Inscripcion Actual"
End Sub
Sub Botón129_Click ()
DoCmd SetWarnings True
End Sub
Sub Cancelar_Inscripcion_Click ()
DoCmd Close
DoCmd Close A_FORM, "Le cuenta a dar de alta materias"
End Sub
Sub consulta_grupos_Click ()
DoCmd OpenForm "QUE MATERIA"
End Sub
Sub Form_AfterUpdate ()
Me.Refresh.Enabled = True
End Sub
Sub Form_Error (DataErr As Integer, Response As Integer)
Response = DATA_ERRCONTINUE
Const noeliminar = 3300
If DataErr = noeliminar Then
    m2 = "Existen alumnos inscritos en esta materia!"
    f2 = 48 + 0 + 0
    r2 = MsgBox(m2, f2, f1ulo)
End If
End Sub
Sub Form_Open (Cancel As Integer)
DoCmd SetWarnings False
Dim tipoObjeto As Integer, nombreObjeto As String, estadoObjeto As Integer
tipoObjeto = A_FORM
nombreObjeto = "Semestre Actual"
estadoObjeto = SysCmd(SYSCMD_GETOBJECTSTATE, tipoObjeto, nombreObjeto)
If estadoObjeto = 0 Then
    msg4 = "No hay semestre actual. Elegir semestre y repetir la acción que desea."
    dqdef4 = 48 + 0 + 0
    res4 = MsgBox(msg4, dqdef4, f1ulo)
    DoCmd OpenForm "Semestre Actual"
    DoCmd CancelEvent
End If
Me.Refresh.Enabled = False
Sub grupo_BeforeUpdate (Cancel As Integer)
If Len(grupo) < 4 Then
    m2 = "El número de grupo debe ser de 4 dígitos"
    f2 = 0 + 48 + MR_DIALOGONZ
    r2 = MsgBox(m2, f2, f1ulo)
    SendKeys "{esc}", True
    DoCmd CancelEvent
End If
If IsNull(grupo) Then
    m2 = "El número de grupo debe ser de 4 dígitos"
    f2 = 0 + 48 + MR_DIALOGONZ
    r2 = MsgBox(m2, f2, f1ulo)
    SendKeys "{esc}", True
    DoCmd CancelEvent
End If
End Sub
Sub hacer_oL_cambio_Click ()
Me.Refresh.Enabled = False
If IsNull(Me.Refresh.Enabled) Or IsNull(Me.Refresh.Enabled) Or IsNull(Me.Refresh.Enabled) Or IsNull(Me.Refresh.Enabled) Then
    msg1 = "Alguno de los campos (Profesor, Materia, Grupo, año) está vacío"
    dqdef1 = 0 + 48 + 0
    RES1 = MsgBox(msg1, dqdef1, f1ulo)
Exit Sub
End If
msg2 = "¿Datos correctos?"
dqdef2 = 4 + 32 + 0
res2 = MsgBox(msg2, dqdef2, f1ulo)
If res2 = 0 Then
    msg3 = "¿Realizar el Cambio?"
    dqdef3 = 4 + 32 + 0
    RES3 = MsgBox(msg3, dqdef3, f1ulo)
    If RES3 = 0 Then
        Dim mIBD As Database, miTabla As Recordset
        Set mIBD = DBEngine.Workspaces(0).Databases(0)
        Set miTabla = mIBD.OpenRecordset("CLASE", DB_OPEN_TABLE) ' Abre la tabla.
        miTabla.Index = "matas" ' Establece el índice activo.
        miTabla.Seek "=", Me(CLV_MATI)
        If Not (miTabla.NoMatch) Then
            miTabla.MoveNext
Do Until miTabla.EOF
    If miTabla(CLV_MATI) = Me(CLV_MATI) And miTabla[numero] = Me(GRUPON) And miTabla[semestre] = Formularios[Semestre Actual][Semestre] Then

```

```

m2 = "El grupo ya existe!"
f2 = 0 + 0 + MR_INTERO2
f2 = MsgBox(m2, f2, titulo)
SendKeys "{Tab}"
DoCmd CancelEvent
Exit Do
Else
mitabla MoveNext
End If
Loop
End If
mitabla.Close 'Cierra la tabla
DoCmd DoMenuItem A_FORMARAR, A_FILE, A_SAVELASORD, , A_MENU_VLR20 09000
msg4 = "Modificar Otra Clase?"
dgdef4 = 4 + 32 + 0
res4 = MsgBox(msg4, dgdef4, titulo)
If res4 = 7 Then
DoCmd Close
End If
Else
SendKeys "{ESC}", True
msg4 = "Modificar Otra Clase?"
dgdef4 = 4 + 32 + 0
res4 = MsgBox(msg4, dgdef4, titulo)
If res4 = 7 Then
DoCmd Close
End If
End If
Else 'De datos correctos
DoCmd GoToControl "R/C PROMOTOR"
End If
End Sub
Sub Seleccionar_grupo_AfterUpdate ()
Me!Buzcar el cambio Enabled = True
DoCmd ApplyFilter "prueba2"
End Sub
Sub Seleccionar_materia_AfterUpdate ()
Me!Buzcar grupo! Enabled = True
Me!Buzcar grupo! Requery
End Sub
Sub Ver_tira_de_materias_Click ()
Dim miDB As Database, mitabla As Recordset
Set miDB = DDEngine.Workspaces(0).Databases(0)
Set mitabla = miDB.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
mitabla.Index = "INCLASIFICA" 'Establece el índice activo.
mitabla.Seek "=", Me!Inscripcion
If (mitabla.NoMatch) Then
m2 = "El Alumno no está inscrito en ninguna materia!"
f2 = 48 + 0 + 0
r2 = MsgBox(m2, f2)
Else
DoCmd OpenForm "Ver tira de Materias", , "Inscripcion" & "Tabla de materias y grupos" & "INCLASIFICA"
End If
End Sub
Sub clave_de_la_materia_AfterUpdate ()
If IsNull(mat_nueva) Then
msg1 = "La clave es incorrecta!"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
SendKeys "{esc}", True
DoCmd CancelEvent
End If
Dim miDB As Database, mitabla As Recordset
Set miDB = DDEngine.Workspaces(0).Databases(0)
Set mitabla = miDB.OpenRecordset("MATERIA", DB_OPEN_TABLE)
mitabla.Index = "CLAVE" 'Establece el índice activo
mitabla.Seek "=", format(modificar materia) & "clave de la materia"
If Not (mitabla.NoMatch) Then
m2 = "Existe una materia con la misma clave!"
f2 = 48 + 0 + 0
r2 = MsgBox(m2, f2, titulo)
SendKeys "{esc}", False
DoCmd CancelEvent
End If
End Sub
Sub consultar_claves_Click ()
DoCmd OpenForm "Consulta de Materias"
End Sub
Sub form_dctore(pdate (Cancel As Integer)
If IsNull(trim_mai) Then
msg1 = "El campo Nombre de Materia Vacío!"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
DoCmd CancelEvent
DoCmd GoToControl "nombre_materia"
End If
End Sub
Sub form_Error (DataErr As Integer, Response As Integer)
Const VALORNULL = 2107
If Err = VALORNULL Then
msg1 = "El campo Nombre de Materia Vacío!"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
DoCmd CancelEvent
DoCmd GoToControl "nombre_materia"
End If
End Sub
Sub form_Open (Cancel As Integer)
DoCmd SetWarnings False
Dim tipoObjeto As Integer, nombreObjeto As String, estadoObjeto As Integer
tipoObjeto = A_form
nombreObjeto = "Semestre Actual"
estadoObjeto = SysCmd(SysCmd_GETOBJECTSTAT, tipoObjeto, nombreObjeto)
If estadoObjeto = 0 Then
msg4 = "No hay semestre actual. Llegr semestre y repetir la accion que desee!"
dgdef4 = 48 + 0 + 0
res4 = MsgBox(msg4, dgdef4, titulo)
DoCmd OpenForm "Semestre Actual"

```



```

DoCmd CancelEvent
End If
Me!etiqueta campo\visible = False
Me!etjje campo\visible = False
Me!clave de la materia\visible = False
Me!sombra de clave\visible = False
Me!etiqueta clave\visible = False
Me!etiqueta nombre materia\visible = False
Me!nombre de la materia\visible = False
Me!sombra nombre\visible = False
Me!etiqueta creditos\visible = False
Me!CREDITOS\visible = False
Me!sombra creditos\visible = False
End Sub
Sub HACER_CAMBIO_Click ()
msg1 = "¿Dece hacer el cambio?"
dgdef1 = 4 + 32 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
If RES1 = 6 Then
If IsNull(nom_mat) Then
msg1 = "¿El campo Nombre de Materia Vacío?"
dgdef1 = 0 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
DoCmd GoToControl "nombre de la materia"
Exit Sub
End If
If IsNull(creditos) Then
msg1 = "¿Debe especificar Número de Créditos?"
dgdef1 = 0 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
DoCmd GoToControl "creditos"
Exit Sub
End If
DoCmd DoMenuItem A_FORMBAR, A_FILE, A_SAVEREORD, , A_MENU_VERZO
msg2 = "¿Modificar otra Materia?"
dgdef2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdef2, titulo)
If res2 = 7 Then
DoCmd Close
Else
Me!clave de la materia\visible = False
Me!sombra de clave\visible = False
Me!etiqueta clave\visible = False
Me!etiqueta nombre materia\visible = False
Me!nombre de la materia\visible = False
Me!sombra nombre\visible = False
Me!etiqueta creditos\visible = False
Me!CREDITOS\visible = False
Me!sombra creditos\visible = False
Me!etiqueta campo\visible = False
Me!etjje campo\visible = False
End If
Else
SendKeys "[ESC]", True
DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, A_UNDOFIELD, , A_MENU_VERZO
msg2 = "¿Modificar otra Materia?"
dgdef2 = 4 + 32 + 0
res2 = MsgBox(msg2, dgdef2, titulo)
If res2 = 7 Then
DoCmd Close
Else
Me!clave de la materia\visible = False
Me!sombra de clave\visible = False
Me!etiqueta clave\visible = False
Me!etiqueta nombre materia\visible = False
Me!nombre de la materia\visible = False
Me!sombra nombre\visible = False
Me!etiqueta creditos\visible = False
Me!CREDITOS\visible = False
Me!sombra creditos\visible = False
Me!etiqueta campo\visible = False
Me!etjje campo\visible = False
End If
End If
End Sub
Sub lista_de_materias_beforeUpdate (Cancel As Integer)
Me!clave de la materia\visible = True
Me!sombra de clave\visible = True
Me!etiqueta clave\visible = True
Me!etiqueta nombre materia\visible = True
Me!nombre de la materia\visible = True
Me!sombra nombre\visible = True
Me!etiqueta creditos\visible = True
Me!CREDITOS\visible = True
Me!sombra creditos\visible = True
Me!etiqueta campo\visible = True
Me!etjje campo\visible = True
End Sub
Sub lista_de_materias_NotInList (NewData As String, Response As Integer)
Response = DATA_CONTINUE
msg1 = "¿El nombre de la materia debe coincidir con la lista?"
dgdef1 = 0 + 48 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
End Sub
Sub lista_nueva_beforeUpdate (Cancel As Integer)
If Len(msg1_nueva) < 6 Then
msg1 = "¿La clave debe ser de 6 dígitos?"
dgdef1 = 48 + 0 + 0
RES1 = MsgBox(msg1, dgdef1, titulo)
DoCmd CancelEvent
End If
End Sub
Sub MATERIA_AfterUpdate ()
DoCmd RunMacro "Rellenar nombres"
End Sub
Sub nombre_de_la_materia_AfterUpdate ()
If Me!NOA_MATI = "" Then
msg1 = "¿El campo Nombre de Materia Vacío?"
dgdef1 = 48 + 0 + 0

```



```

Me[[etiqueta RFC].visible = False
Me[[sombra RFC].visible = False
Me[[RFC].visible = False
    End If
Else
    SendKeys "[ESC]", True
    DoCmd DoMenuItem A_FORMATAR, A_CORTAR, A_UNPOHELI, , A_MENU_VERZO
    msg2 = "¿Modificar otro Profesor?"
    dgde12 = 4 + 32 + 0
    res2 = MsgBox(msg2, dgde12, titulo)
    If res2 = 7 Then
        DoCmd Close
    Else
        Me[[etiqueta RFC].visible = False
        Me[[etiqueta Internol].visible = False
        Me[[etiqueta Internol].visible = False
        Me[[etiqueta Profesor].visible = False
        Me[[etiqueta campo].visible = False
        Me[[PATERNO].visible = False
        Me[[maternol].visible = False
        Me[[sombre].visible = False
        Me[[HISTORIA].visible = False
        Me[[sombra paternol].visible = False
        Me[[sombra historial].visible = False
        Me[[sombra maternol].visible = False
        Me[[sombra nombrel].visible = False
        Me[[etiqueta RFC].visible = False
        Me[[sombra RFC].visible = False
        Me[[RFC].visible = False
    End If
End If
End If
End Sub
Sub Hacer_cl_cambio_Click ()
End Sub
Sub lista_de_profesores_AfterUpdate ()
DoCmd RunMacro "Rellevar profesor"
End Sub
Sub lista_de_profesores_BeforeUpdate (Cancel As Integer)
Me[[etiqueta RFC].visible = True
Me[[etiqueta Paternol].visible = True
Me[[etiqueta maternol].visible = True
Me[[etiqueta Profesor].visible = True
Me[[etiqueta campo].visible = True
Me[[PATERNO].visible = True
Me[[maternol].visible = True
Me[[nombrel].visible = True
Me[[HISTORIA].visible = True
Me[[sombra paternol].visible = True
Me[[sombra historial].visible = True
Me[[sombra maternol].visible = True
Me[[sombra nombrel].visible = True
Me[[etiqueta RFC].visible = True
Me[[sombra RFC].visible = True
Me[[RFC].visible = True
End Sub
Sub lista_de_profesores_NotInList (NewData As String, Response As Integer)
Response = DATA_ERRCONTINUE
msg1 = "¿L RFC del Profesor debe coincidir con un valor de la lista"
dgde11 = 0 + 48 + 0
RES1 = MsgBox(msg1, dgde11, titulo)
End Sub
Sub RFC_AfterUpdate ()
msg4 = "¿Cambiar RFC?"
dgde14 = 4 + 32 + 0
res4 = MsgBox(msg4, dgde14, titulo)
If res4 = 6 Then 'si acepta cambio
Else
    SendKeys "[esc]", True
    DoCmd CancelEvent
End Sub
End If
End Sub
Sub RFC_BeforeUpdate (Cancel As Integer)
If Len(rfc) < 10 Then
    msg1 = "RFC incorrecto!"
    dgde11 = 0 + 48 + 0
    RES1 = MsgBox(msg1, dgde11, titulo)
    DoCmd CancelEvent
End If
Dim mIBD As Database, mitabla As Recordset
Set mIBD = DDEngine.Workspace(0).Database(0)
Set mitabla = mIBD.OpenRecordset("TROF.SOR", DB_OPEN_TABLE)
mitabla.index = "INRFC" 'Establece el índice a suvo.
mitabla.seek "=", formID[Modificar profesor]([RFC])
If (mitabla.NoMatch) Then
Else
    m2 = "¿Existe un Profesor con el mismo RFC?"
    f2 = 48 + MB_ICONALTO + MB_DEFTONZ
    r2 = MsgBox(m2, f2, titulo)
    SendKeys "[esc]", False
    DoCmd CancelEvent
End If
End Sub
Sub Suhr_Click ()
msg2 = "¿Modificar otro Profesor?"
dgde12 = 4 + 32 + 0
res2 = MsgBox(msg2, dgde12, titulo)
If res2 = 7 Then
    DoCmd Close
Else
        Me[[etiqueta RFC].visible = False
        Me[[etiqueta Internol].visible = False
        Me[[etiqueta Internol].visible = False
        Me[[etiqueta Profesor].visible = False
        Me[[etiqueta campo].visible = False
        Me[[PATERNO].visible = False
        Me[[maternol].visible = False
        Me[[sombre].visible = False
    End If
End Sub

```

```

MeIf(nombre).visible = false
MeIf(HISTORIA).visible = False
MeIf(sombra.paternol).visible = false
MeIf(sombra.historial).visible = false
MeIf(sombra.maternol).visible = false
MeIf(sombra.nombre).visible = false
MeIf(etiqueta.RFC).visible = false
MeIf(sombra.RFC).visible = False
MeIf(RFC).visible = false
End If
End Sub

Sub Form_Open (Cancel As Integer)
DoCmd Applyfilter "Filtro semestre actual"
End Sub

Sub Form_Load ()
DoCmd Applyfilter "Filtro de semestre y materia"
End Sub
Sub imprimir_Click ()
On Error GoTo Err_imprimir_Click
DoCmd OpenReport "grupos actuales", , "clv_mat" = formularios!para consultar grupos!clv_mat"
Exit Sub
Err_imprimir_Click:
Exit Sub
Err_imprimir_Click:
m3 = "Se canceló impresión!"
fs = 48 + 0 + 0
rs = MsgBox(m3, fs, titulo)
Resume Err_imprimir_Click
End Sub
Sub salir_Click ()
m2 = "¿Desea realizar otra consulta?"
f2 = 42 + 4 + 0
r2 = MsgBox(m2, f2, titulo)
If r2 = 6 Then
DoCmd Close
DoCmd OpenForm "Consultar Clase por Grupo"
Else
DoCmd Close
End If
End Sub

Sub aceptar_Click ()
DoCmd OpenForm "Inscripcion Actual"
End Sub
Sub consulta_grupos_Click ()
DoCmd OpenForm "QUE MATERIA"
End Sub
Sub Form_AfterUpdate ()
Me!Seleccionar grupo!Requery
End Sub
Sub Form_Open (Cancel As Integer)
Me!Vista preliminar.Enabled = false
Me!Imprimir.Enabled = false
End Sub
Sub Imprimir_Click ()
On Error GoTo Err_imprimir_Click
DoCmd OpenReport "Lista de asistencia", , "para lista de asistencia"
Exit Sub
Err_imprimir_Click:
m3 = "Se canceló impresión!"
fs = 48 + 0 + 0
rs = MsgBox(m3, fs, titulo)
Exit Sub
End Sub
Sub Proceed_inscripcion_Click ()
Me!Seleccionar grupo.Enabled = false
DoCmd OpenForm "inscripcion actual"
End Sub
Sub salir_Click ()
DoCmd Close
Formulario!Menu de Reportes!cancelar reportes.Enabled = True
Form!Menu de Reportes!cancelar reportes!Selfocus
Formulario!Menu de Reportes!Estadisticos.Enabled = True
Formulario!Menu de Reportes!Nómeros pronosticados.Enabled = True
Formulario!Menu de Reportes!Historial Académico.Enabled = True
End Sub
Sub Seleccionar_grupo_AfterUpdate ()
Dim mBU() As Database, mtTabla As Recordset
Set mBU() = DDEngine.Workspace(0) Databases(0)
Set mtTabla = mBU.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
mtTabla.index = "CLV_MAT" Establece el índice activo
mtTabla.Seek "=", Me!Seleccionar material
If (mtTabla.NoMatch) Then
Else
Do Until mtTabla.EOF
If mtTabla!clv_mat = Me!Seleccionar material And mtTabla!num_grupo = Me!Seleccionar grupo And mtTabla!s_mestre1 = formularios!Semestre Actual!Semestre1 Then
GoTo abrir
Exit Do
Else
mtTabla.MoveNext
End If
Loop
m2 = "¿No hay alumnos inscritos en este grupo para el semestre actual?"
f2 = 48 + 0 + 0
r2 = MsgBox(m2, f2, titulo)
Me!Vista preliminar.Enabled = True
Me!Imprimir.Enabled = True
mtTabla.Close
Exit Sub
End If
mtTabla.Close
abrir:
DoCmd Applyfilter "para lista de asistencia"
Me!Vista preliminar.Enabled = True
Me!Imprimir.Enabled = True
Exit Sub
End Sub
End Sub

```

```

Sub Seleccionar_Materia_AfterUpdate ()
Dim temp As Variant
Me!Seleccionar_grupo!.Enabled = True
Me!Seleccionar_grupo!.Requery
End Sub
Sub Vista_preliminar_Click ()
DoCmd OpenReport "Lista de asistencia", A_PREVIEW, "para lista de asistencia"
End Sub

Sub aceptar_Click ()
DoCmd OpenForm "Inscripcion Actual"
End Sub
Sub consulta_grupo_Click ()
DoCmd OpenForm "QUE MATERIA"
End Sub
Sub Form_AfterUpdate ()
Me!Seleccionar_grupo!.Requery
End Sub
Sub Impedir_Click ()
On Error GoTo Err_Impedir_Click
DoCmd OpenReport "Reporte de provisionales"
Exit Sub
Err_Impedir_Click:
m3 = "Se canceló impedido!"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, f3, título)
Exit Sub
Sub Procede_Inscripcion_Click ()
Me!Seleccionar_grupo!.Enabled = False
DoCmd OpenForm "Inscripcion actual"
End Sub
Sub salir_Click ()
DoCmd Close
Formulario!Menu de Reportes!Cancelar reportes!.Enabled = True
Form!Menu de Reportes!Cancelar reportes!.SetFocus
Formulario!Menu de Reportes!Estadísticas!.Enabled = True
Formulario!Menu de Reportes!Lista de asistencia!.Enabled = True
Formulario!Menu de Reportes!Historial Académico!.Enabled = True
End Sub
Sub Seleccionar_grupo_AfterUpdate ()
Dim mIBD As Database, mITABLA As Recordset
Set mIBD = dbengine.workspaces(0).databases(0)
Set mITABLA = mIBD.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
mITABLA.Index = "CLV_MAT" ' Establece el índice activo.
mITABLA.Seek "=", Me!Seleccionar_Materia
If Not (mITABLA.NoMatch) Then
If mITABLA!nombre = Me!Seleccionar_grupo And mITABLA!semestre = Formulario!Semestre Actual!Semestre!Then
DoCmd ApplyFilter "para lista de asistencia"
Me!Vista_preliminar!.Enabled = True
Else
m2 = "No hay alumnos inscritos en este grupo para el semestre actual"
f2 = 48 + 0 + 0
r2 = MsgBox(m2, f2, título)
Me!Vista_preliminar!.Enabled = False
End If
Else
m2 = "No hay alumnos inscritos"
f2 = 48 + 0 + 0
r2 = MsgBox(m2, f2, título)
Me!Vista_preliminar!.Enabled = False
End If
mITABLA.Close
End Sub
Sub Seleccionar_Materia_AfterUpdate ()
Dim temp As Variant
Me!Seleccionar_grupo!.Enabled = True
Me!Seleccionar_grupo!.Requery
End Sub
Sub Vista_preliminar_Click ()
DoCmd OpenReport "reporte de provisionales", A_PREVIEW
End Sub

Sub aceptar_Click ()
Dim tipoObjeto As Integer, nombreObjeto As String, estadoObjeto As Integer
tipoObjeto = A_form
nombreObjeto = "Semestre Actual"
estadoObjeto = SysCmd(SYSCMD_CUSTOMOBJECTSTATE, tipoObjeto, nombreObjeto)
If estadoObjeto = 0 Then
msg4 = "No hay semestre actual. Elegir semestre y repetir la acción que desea!"
dlg4f4 = 0 + 0 + 0
rst4 = MsgBox(msg4, dlg4f4, título)
DoCmd OpenForm "Semestre Actual"
Exit Sub
End If
Select Case Formulario!Proceso Semestral!bandera)
Case 1
Dim mIBD As Database, MITABLA As Recordset
Set mIBD = dbengine.workspaces(0).databases(0)
Set MITABLA = mIBD.OpenRecordset("PASSWORD", DB_OPEN_TABLE)
MITABLA.Index = "CLAVE" ' Establece el índice activo.
MITABLA.Seek "=", "MATERIAS"
If Not (MITABLA.NoMatch) Then
If UCCase(MITABLA!CLAVE) = "MATERIAS" And MITABLA!PASSWORD = UCCase(Me!CLAVE) Then
DoCmd OpenForm "Alta de Materias"
DoCmd Close A_form, "PASSWORD"
MITABLA.Close
Exit Sub
Else
If (Formulario!Proceso Semestral!bandera) = 1 And Formulario!Proceso Semestral!sub.formulario!DATOS A SELECCIONAR = 1) Then
DoCmd OpenForm "Alta de Materias"
DoCmd Close A_form, "PASSWORD"
MITABLA.Close
Exit Sub
Else
If (Formulario!Proceso Semestral!bandera) = 1 And Formulario!Proceso Semestral!sub.formulario!DATOS A SELECCIONAR = 2) Then
Dim ts1 As Database, MITABLA1 As Recordset
Set ts1 = dbengine.workspaces(0).databases(0)
Set MITABLA1 = ts1.OpenRecordset("MATERIA", DB_OPEN_TABLE)
total = MITABLA1.RecordCount
If total = 0 Then
m3 = "No hay materias dadas de alta!"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, f3, título)

```

```

MITABLA1 Close
Docmd Close A_form, "PASSWORD"
MITABLA1Close
MITABLA1Close
Exit Sub
Else
Docmd OpenForm "Bajas de Materias"
Docmd Close A_form, "PASSWORD"
MITABLA1 Close
MITABLA1 Close
Exit Sub
End If
Else
If (Formulario1)Processo Semestral()|bandera1 = 1 And Formulario1|Processo Semestral()|sub|formulario1|DATOS A SELECCIONAR1 = 31 Then
Dim test2 As Database, mitabla2 As Recordset
Set test2 = dbengine.workspaces(0).databases(0)
Set mitabla2 = test2.OpenRecords("MATERIAS", DB_OPEN_TABLE)
total = mitabla2.RecordCount
If total = 0 Then
m3 = "No hay materias dadas de alta"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, 15, titulo)
mitabla2 Close
MITABLA1 Close
Docmd Close A_form, "PASSWORD"
Exit Sub
Else
Docmd OpenForm "Modificar Materias"
Docmd Close A_form, "PASSWORD"
mitabla2 Close
MITABLA1 Close
Exit Sub
End If
End If
End If
Else
msg4 = "La clave es incorrecta"
dgedf4 = 48 + 0 + 0
rc4 = MsgBox(msg4, dgedf4, titulo)
Docmd GoToControl "cancelar"
MITABLA1 Close
Exit Sub
End If
Else
msg4 = "La clave es incorrecta"
dgedf4 = 48 + 0 + 0
rc4 = MsgBox(msg4, dgedf4, titulo)
Docmd GoToControl "cancelar"
MITABLA1 Close
Exit Sub
End If
Case 2
Dim m1013 As Database, MITABLA3 As Recordset
Set m1013 = dbengine.workspaces(0).databases(0)
Set MITABLA3 = m1013.OpenRecordset("PASSWORD", DB_OPEN_TABLE)
MITABLA3 index = "CLAVE" ' Establece el índice activo
MITABLA3 <<< " ", "PROFESOR"
If Not (MITABLA3.isnull) Then
If (Formulario1)Processo Semestral()|bandera1 = 2 And Formulario1|Processo Semestral()|sub|formulario1|DATOS A SELECCIONAR1 = 1) Then
Docmd OpenForm "Altas de Profesor"
Docmd Close A_form, "PASSWORD"
Exit Sub
Else
If (Formulario1)Processo Semestral()|bandera1 = 2 And Formulario1|Processo Semestral()|sub|formulario1|DATOS A SELECCIONAR1 = 2) Then
Dim test4 As Database, MITABLA4 As Recordset
Set test4 = dbengine.workspaces(0).databases(0)
Set MITABLA4 = test4.OpenRecords("PROFESOR", DB_OPEN_TABLE)
total = MITABLA4.RecordCount
If total = 0 Then
m3 = "No hay profesores dados de alta"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, 15, titulo)
MITABLA4 Close
MITABLA4 Close
Docmd Close A_form, "PASSWORD"
Exit Sub
Else
Docmd OpenForm "Bajas de Profesor"
Docmd Close A_form, "PASSWORD"
MITABLA4 Close
MITABLA4 Close
Exit Sub
End If
Else
If (Formulario1)Processo Semestral()|bandera1 = 2 And Formulario1|Processo Semestral()|sub|formulario1|DATOS A SELECCIONAR1 = 3) Then
Dim test5 As Database, MITABLA5 As Recordset
Set test5 = dbengine.workspaces(0).databases(0)
Set MITABLA5 = test5.OpenRecords("PROFESOR", DB_OPEN_TABLE)
total = MITABLA5.RecordCount
If total = 0 Then
m3 = "No hay profesores dados de alta"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, 15, titulo)
MITABLA5 Close
MITABLA5 Close
Docmd Close A_form, "PASSWORD"
Exit Sub
Else
Docmd OpenForm "Modificar Profesor"
Docmd Close A_form, "PASSWORD"
MITABLA5 Close
MITABLA5 Close
Exit Sub
End If
End If
End If
End If
End If

```

```

Else
    msg4 = "¡La clave es incorrecta!"
    dgdef4 = 48 + 0 + 0
    res4 = MsgBox(msg4, dgdef4, título)
    DoCmd GoToControl "cancelar"
    MITABLA3.Close
    Exit Sub
End If
Else
    msg4 = "¡La clave es incorrecta!"
    dgdef4 = 48 + 0 + 0
    res4 = MsgBox(msg4, dgdef4, título)
    DoCmd GoToControl "cancelar"
    MITABLA3.Close
    Exit Sub
End If
Case 3
    Dim mIBD1 As Database, MITABLA1 As Recordset
    Set mIBD1 = dbengine.workspaces(0).databases(0)
    Set MITABLA1 = mIBD1.OpenRecordset("PASSWORDS", DII_OPEN_TABLE)
    MITABLA1.Index = "CLAVE" ' Establece el índice activo.
    MITABLA1.Seek "=", "TORARIOS"
    If Not (MITABLA1.NoMatch) Then
        If UCCase(MITABLA1[CLAVE]) = "TORARIOS" And MITABLA1[PASSWORD] = UCCase(MITABLA1[CLAVE]) Then
            DoCmd OpenForm "Formar Clase"
            DoCmd Close A_form, "PASSWORD"
            MITABLA1.Close
            Exit Sub
        Else
            If (Formulario1[Proceso Semestral][bandera] = 3 And Formulario1[Proceso Semestral][sub].formulario1[DATOS A SELECCIONAR] = 1) Then
                DoCmd OpenForm "Formar Clase"
                DoCmd Close A_form, "PASSWORD"
                Exit Sub
            Else
                If (Formulario1[Proceso Semestral][bandera] = 3 And Formulario1[Proceso Semestral][sub].formulario1[DATOS A SELECCIONAR] = 2) Then
                    DoCmd OpenForm "Baja de clase"
                    DoCmd Close A_form, "PASSWORD"
                    Exit Sub
                Else
                    If (Formulario1[Proceso Semestral][bandera] = 3 And Formulario1[Proceso Semestral][sub].formulario1[DATOS A SELECCIONAR] = 3) Then
                        DoCmd OpenForm "Modificar Clase"
                        DoCmd Close A_form, "PASSWORD"
                        Exit Sub
                    End If
                End If
            End If
        End If
    Else
        msg4 = "¡La clave es incorrecta!"
        dgdef4 = 48 + 0 + 0
        res4 = MsgBox(msg4, dgdef4, título)
        DoCmd GoToControl "cancelar"
        MITABLA3.Close
        Exit Sub
    End If
End If
Else
    msg4 = "¡La clave es incorrecta!"
    dgdef4 = 48 + 0 + 0
    res4 = MsgBox(msg4, dgdef4, título)
    DoCmd GoToControl "cancelar"
    MITABLA1.Close
    Exit Sub
End If
End Select
End Sub
Sub Botón21_Click ()
    [Formulario1][Semestre Actual].[Visible] = False
End Sub
Sub cancelar_Click ()
    DoCmd Close
    DoCmd CancelEvent
End Sub
Sub aceptar_Click ()
    Dim mIBD As Database, MITABLA As Recordset
    Set mIBD = dbengine.workspaces(0).databases(0)
    Set MITABLA = mIBD.OpenRecordset("PASSWORDS", DII_OPEN_TABLE)
    MITABLA.Index = "CLAVE" ' Establece el índice activo.
    MITABLA.Seek "=", "ENTRADA"
    If Not (MITABLA.NoMatch) Then
        If UCCase(MITABLA[CLAVE]) = "ENTRADA" And MITABLA[PASSWORD] = Me[CLAVE] Then
            Dim tipoObjeto As Integer, nombreObjeto As String, estadoObjeto As Integer
            tipoObjeto = a_form.nombreObjeto
            estadoObjeto = "Semestre Actual"
            nombreObjeto = SysCmd(SYSCMD_GETOBJECTSTAT, tipoObjeto, nombreObjeto)
            If estadoObjeto = 0 Then
                DoCmd Close a_form, "entrada"
                DoCmd Close
                DoCmd OpenForm "Semestre Actual"
                Exit Sub
            End If
        End If
    Else
        msg4 = "¡Password incorrecto!"
        dgdef4 = 48 + 0 + 0
        res4 = MsgBox(msg4, dgdef4, título)
        MITABLA.Close
    End If
End Sub
Sub Botón21_Click ()
    [Formulario1][Semestre Actual].[Visible] = False
End Sub
Sub cancelar_Click ()
    DoCmd Close
End Sub
Sub aceptar_Click ()
    If UCCase(MITABLA[CLAVE]) = "SISTEMAS" Then
        If Formulario1[Menu de Catalogo][Opciones password]/formulario1[seleccion de password] = 1 Then

```

```

ProcMd Close
ProcMd OpenForm "Cambio de password"
Libc
ProcMd Close
ProcMd OpenForm "Catálogo de passwords"
End If
Vbc
msg4 = "¡password incorrecto!"
dgdef4 = 48 + 0 + 0
rc4 = MsgBox(msg4, dgdef4, título)
End If
End Sub
Sub BotónZ1_Click ()
If form1[Opciones de Proceso Actual][Visible] = False
End Sub
Sub cancelar_Click ()
ProcMd Close
End Sub
Sub aceptar_datos_o_plan_Click ()
If form1[datos o plan][Opciones_alumnos] = 1 Then
ProcMd OpenForm "Opciones alumnos"
End If
If form1[datos o plan][Opciones_alumnos] = 2 Then
ProcMd OpenForm "Opciones alumnos"
End If
If form1[datos o plan][Opciones_alumnos] = 3 Then
ProcMd OpenForm "Opciones alumnos"
End If
If form1[datos o plan][Opciones_alumnos] = 4 Then
ProcMd OpenForm "Opciones alumnos (consultar)"
End If
End Sub
Sub aceptar_proceso_seme_Click ()
If form1[Opciones de Proceso semestral][Opciones_proceso] = 1 Or form1[Opciones de Proceso semestral][Opciones_proceso] = 2 Or form1[Opciones de Proceso semestral][Opciones_proceso] = 3 Then
ProcMd OpenForm "ABC Proceso Semestral"
End If
If form1[Opciones de Proceso semestral][Opciones_proceso] = 4 Then
ProcMd OpenForm "Opciones alumnos"
End If
If form1[Opciones de Proceso semestral][Opciones_proceso] = 5 Then
ProcMd OpenForm "Opciones alumnos"
End If
End Sub
Sub Actas_Click ()
Me[txtActas].Visible = False
Me[txtMater].Visible = False
Me[Subformulario actas].Visible = True
Me[profesores].Enabled = False
Me[horarios].Enabled = False
Me[inscripciones].Enabled = False
Me[materias].Enabled = False
Me[cancelar_proceso_semestral].Enabled = False
Me[bandera] = 5
End Sub
Sub cancelar_datos_o_plan_Click ()
ProcMd Close
End Sub
Sub cancelar_proceso_sem_Click ()
ProcMd Close
form1[datos][MENUC_PRINCIPAL].Visible = True
End Sub
Sub Form_Open (Cancel As Integer)
Me[Subformulario actas].Visible = False
Me[txtMater].Visible = False
Me[txtActas].Visible = False
Me[foto para acta].Visible = False
End Sub
Sub Horarios_Click ()
Me[txtMater].Visible = False
Me[txtMater].Visible = True
Me[materias].Enabled = False
Me[profesores].Enabled = False
Me[inscripciones].Enabled = False
Me[actas].Enabled = False
Me[cancelar_proceso_semestral].Enabled = False
Me[bandera] = 4
End Sub
Sub Inscripciones_Click ()
Me[txtMater].Visible = False
Me[txtMater].Visible = True
Me[profesores].Enabled = False
Me[horarios].Enabled = False
Me[materias].Enabled = False
Me[actas].Enabled = False
Me[cancelar_proceso_semestral].Enabled = False
Me[bandera] = 4
End Sub
Sub Materias_Click ()
Me[txtMater].Visible = False
Me[txtMater].Visible = True
Me[profesores].Enabled = False
Me[horarios].Enabled = False
Me[inscripciones].Enabled = False
Me[actas].Enabled = False
Me[cancelar_proceso_semestral].Enabled = False
Me[bandera] = 1
End Sub
Sub Materias_Exit (Cancel As Integer)
Me[txtMater].Visible = False
End Sub
Sub Materias_MouseDown (Button As Integer, Shift As Integer, X As Single, Y As Single)
Me[txtMater].Visible = False
End Sub
Sub opciones_proceso_alterUpdate ()
If form1[Opciones de Proceso semestral][Opciones_proceso] = 1 Then
ProcMd OpenForm "ABC Proceso Semestral"
End If

```



```

If Form1.Opciones de Proceso semestral(Opciones_proceso) = 2 Then
  DoCmd OpenForm "ABC Proceso Semestral"
End If
If Form1.Opciones de Proceso semestral(Opciones_proceso) = 3 Then
  DoCmd OpenForm "ABC Proceso Semestral"
End If
If Form1.Opciones de Proceso semestral(Opciones_proceso) = 4 Then
  DoCmd OpenForm "Opciones alumnos"
End If
If Form1.Opciones de Proceso semestral(Opciones_proceso) = 5 Then
  DoCmd OpenForm "Opciones alumnos"
End If
End Sub
Sub Profesores_Click ()
  Me!m1.Visible = False
  Me!m2.Visible = True
  Me!materias.Enabled = False
  Me!honorarios.Enabled = False
  Me!materia(Opciones).Enabled = False
  Me!materia(Opciones).Enabled = False
  Me!cancelar proceso semestral.Enabled = False
  Me!bandera1 = 2
End Sub

Sub Boton70_Click ()
  Dim tex As Database, mitable As Recordset
  Set tex = DBaseUtils.Workspaces(0).databases(0)
  Set mitable = tex.OpenRecordset("ALUMNOS", DB_OPEN_TABLE)
  mitable.Index = "TUTOR"
  mitable.Seek "=", Me!registro
  If mitable.NoMatch Then
    m1 = "Profesor sin alumnos bajo su tutela"
    f1 = 48 + 0 + 0
    r1 = MsgBox(m1, f1, titulo)
  mitable.Close
  Else
  DoCmd OpenQuery "para tutorias"
  DoCmd ShowToolbar "diseño de la consulta", A_TOOLBAR_no
  DoCmd ShowToolbar "hoja de datos de la consulta", A_TOOLBAR_no
  DoCmd MoveSize 1440, 4000, 7500, 2000
  End If
End Sub
Sub Boton70_Click ()
End Sub
Sub Boton73_Click ()
End Sub
Sub clase_Click ()
  Dim tex As Database, mitable As Recordset
  Set tex = DBaseUtils.Workspaces(0).databases(0)
  Set mitable = tex.OpenRecordset("CLASE", DB_OPEN_TABLE)
  mitable.Index = "PROFESOR"
  mitable.Seek "=", Me!registro
  If (mitable.NoMatch) Then
    m1 = "Profesor sin materias para el semestre actual"
    f1 = 48 + 0 + 0
    r1 = MsgBox(m1, f1, titulo)
  mitable.Close
  Exit Sub
  Else
  Do Until mitable.EOF
    If mitable.Fields("Semestre Actual") = Semestre Then
      LetTo Abrir.
      Exit Do
    Else
      mitable.MoveNext
    End If
  Loop
  m1 = "Profesor sin materias para ningún semestre"
  f1 = 48 + 0 + 0
  r1 = MsgBox(m1, f1, titulo)
  mitable.Close
  Exit Sub
End If
mitable.Close
Abrir
DoCmd OpenQuery "para clases del profesor"
DoCmd ShowToolbar "diseño de la consulta", A_TOOLBAR_no
DoCmd ShowToolbar "hoja de datos de la consulta", A_TOOLBAR_no
DoCmd MoveSize 1440, 4000, 8000, 2000
Exit Sub
End Sub
Sub Imprimir_Click ()
On Error GoTo Err_Imprimir_Click
DoCmd OpenReport "Lista de Profesores"
Exit Sub
Err_Imprimir_Click
Exit_Imprimir_Click
End Sub
Err_Imprimir_Click
m3 = "Se canceló impresión"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, f3, titulo)
Resume Err_Imprimir_Click
End Sub
Sub salir_Click ()
On Error GoTo Err_salir_Click
DoCmd Close
Exit_salir_Click
Exit Sub
Err_salir_Click
MsgBox Error$
Resume Err_salir_Click
End Sub

Sub base_NotFound (NewData As String, Response As Integer)
Response = DATA_ERRORCONTINUE
msg1 = "El nombre de la Base debe coincidir con un valor de la lista"
ddef1 = 48 + 0 + 0
rdef1 = MsgBox(msg1, ddef1, titulo)
End Sub

```

```

Sub Canceled_Click ()
DoCmd Close
Formularioes[Menu de Reportes][Lista de Asistencia] Enabled = True
Formularioes[Menu de Reportes][Historial Académico] Enabled = True
Formularioes[Menu de Reportes][Número Provisional] Enabled = True
Formularioes[Menu de Reportes][Estadístico] Enabled = True
Formularioes[Menu de Reportes][Lista de Asistencia] Visible = False
Formularioes[Menu de Reportes][Subformulario para reportes] Visible = False
Formularioes[Menu de Reportes][Subformulario para historial] Visible = False
End Sub
Sub excel_Click ()
Dim tes As Database, mitable As Recordset
Set tes = dbengine.workspaces(0).databases(0)
Set mitable = tes.OpenRecordset("ALUMNOS", DB_OPEN_TABLE)
total = mitable.RecordCount
If total = 0 Then
m3 = "No hay alumnos dados de alta!"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, vb, titulo)
Exit Sub
Else
DoCmd RunMacro "salidas.beca.excel"
End If
End Sub
Sub Form_Open (Cancel As Integer)
Me[!beca].Visible = False
Me[!excel].Visible = False
Me[word].Visible = False
Me[!etiqueta excel].Visible = False
Me[!Etiqueta Word].Visible = False
End Sub
Sub imprimir_Click ()
Dim tes As Database, mitable As Recordset
Set tes = dbengine.workspaces(0).databases(0)
Set mitable = tes.OpenRecordset("ALUMNOS", DB_OPEN_TABLE)
total = mitable.RecordCount
If total = 0 Then
m3 = "No hay alumnos dados de alta!"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, vb, titulo)
Exit Sub
End If
If Me[!imprimir_reportes] = 1 Then
On Error Goto Err_imprimir_Click
DoCmd OpenReport "Reportes por beca general"
Exit Sub
Err_imprimir_Click:
m3 = "Se canceló impresión!"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, vb, titulo)
Exit Sub
Else
If Me[!imprimir_reportes] = 2 Then
mitable.Index = "Beca"
mitable.Seek "=", beca
If (mitable.NoMatch) Then
m3 = "No existen alumnos que tengan esta beca!"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, vb, titulo)
Exit Sub
Else
On Error Goto Err_imprimir2_Click
DoCmd OpenReport "Reportes por beca específica"
Exit Sub
Err_imprimir2_Click:
m3 = "Se canceló impresión!"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, vb, titulo)
Exit Sub
End If
End If
mitable.Close
End Sub
Sub imprimir_etiquetas_AfterUpdate ()
End Sub
Sub imprimir_reportes_AfterUpdate ()
Me[!vista preliminar] Enabled = True
Me[!imprimir] Enabled = True
If Me[!imprimir_reportes] = 2 Then
Me[!beca].Visible = True
Else
Me[!beca].Visible = False
End If
If Me[!imprimir_reportes] = 3 Then
Me[!excel].Visible = True
Me[word].Visible = True
Me[!etiqueta excel].Visible = True
Me[!Etiqueta Word].Visible = True
Me[!Vista preliminar] Enabled = False
Me[!Imprimir].Enabled = False
Else
Me[!excel].Visible = False
Me[word].Visible = False
Me[!etiqueta excel].Visible = False
Me[!Etiqueta Word].Visible = False
End If
End Sub
Sub Vista_preliminar_Click ()
Dim tes As Database, mitable As Recordset
Set tes = dbengine.workspaces(0).databases(0)
Set mitable = tes.OpenRecordset("ALUMNOS", DB_OPEN_TABLE)
total = mitable.RecordCount
If total = 0 Then
m3 = "No hay alumnos dados de alta!"
f3 = 48 + 0 + 0
r3 = MsgBox(m3, vb, titulo)
Exit Sub
End If

```

```

If Me!ImpprimirReportes = 1 Then
    DoCmd OpenReport "Reportes por beca general", A_preview
Else
    If Me!ImpprimirReportes = 2 Then
        mitable.index = "Beca"
        mitable.seek = "m", beca
        If (mitable.nomatch) Then
            m3 = "No existen alumnos que tengan esta beca!"
            f3 = 48 + 0 + 0
            r3 = MsgBox(m3, 5, titulo)
        End Sub
    Else
        DoCmd OpenReport "Reportes por beca especifica", A_preview
    End If
End If
End Sub
mitable.Close
Sub word_Click ()
Dim lsa As Database, mitable As Recordset
Set lsa = dbengine.workspaces(0).databases(0)
Set mitable = lsa.OpenRecordset("ALUMNOS", DB_OPEN_TABLE)
total = mitable.RecordCount
If total = 0 Then
    m3 = "No hay alumnos dados de alta!"
    f3 = 48 + 0 + 0
    r3 = MsgBox(m3, 5, titulo)
End Sub
Exit Sub
Else
DoCmd RunMacro "midades.becas word"
End If
End Sub

Sub Aceptar_Click ()
If (IsNull(semester) Or (Right(semester, 1) > 2 Or Right(semester, 1) = 0) Or Left(semester, 1) > 2 Or Left(semester, 1) = 0) Then
    m1 = "Semestre incorrecto!"
    d1 = 48 + 0 + 0
    r1 = MsgBox(m1, 4, 1, titulo)
    Exit Sub
Else
    Formulario!Semestre Actual.Visible = False
    DoCmd OpenForm "MENU_PRINCIPAL"
End If
End Sub

Sub sem_AfterUpdate ()
If (IsNull(semester) Or (Right(semester, 1) > 2) Or (Left(semester, 1) > 2)) Then
    m1 = "Semestre incorrecto!"
    d1 = 48 + 0 + 0
    r1 = MsgBox(m1, 4, 1, titulo)
    Exit Sub
Else
    Formulario!Semestre Actual.Visible = False
    DoCmd OpenForm "MENU_PRINCIPAL"
End If
End Sub

Sub Aceptar_Click ()
DoCmd OpenForm "Inscripcion Actual"
End Sub
Sub Cancelar_Inscripcion_Click ()
DoCmd Close
DoCmd Close a_form, "lee cuenta a dar de alta materias"
End Sub
Sub consulta_grupo_Click ()
DoCmd OpenForm "QUE MATERIA"
End Sub
Sub Form_AfterUpdate ()
Me!Seleccionar Grupo.Requery
End Sub
Sub Form_Open (Cancel As Integer)
Me!Procede inscripcion.Enabled = False
End Sub
Sub Procede_Inscripcion_Click ()
Me!Seleccionar Grupo.Enabled = False
DoCmd OpenForm "Inscripcion actual"
End Sub
Sub Seleccionar_grupo_AfterUpdate ()
Me!Procede inscripcion.Enabled = True
DoCmd ApplyFilter "prueba"
End Sub
Sub Seleccionar_Materia_AfterUpdate ()
Dim temp As Variant
Me!Seleccionar Grupo.Enabled = True
Me!Seleccionar Grupo.Requery
temp = activarcontrol("detalle", False)
End Sub
Sub Ver_lista_de_materias_Click ()
Dim m1RD As Database, mitable As Recordset
Set m1RD = DBEngine.Workspaces(0).Databases(0)
Set mitable = m1RD.OpenRecordset("INSCRITO", DB_OPEN_TABLE)
mitable.index = "CURSOS" ' Establece el indice activo.
mitable.seek = "m", Me!Cuenta1
If (mitable.nomatch) Then
    m2 = "El Alumno no está inscrito en ninguna materia del semestre actual!"
    f2 = 48 + 0 + 0
    r2 = MsgBox(m2, 5, titulo)
    mitable.Close
    Exit Sub
Else
    Do Until mitable.eof
        If mitable[Semestre] = Formulario!Semestre Actual Then
            Do To abort:
                Exit Do
            Else
                mitable.MoveNext
            End If
        Loop
    m2 = "El Alumno no está inscrito en ninguna materia del semestre actual!"
    f2 = 48 + 0 + 0

```

```

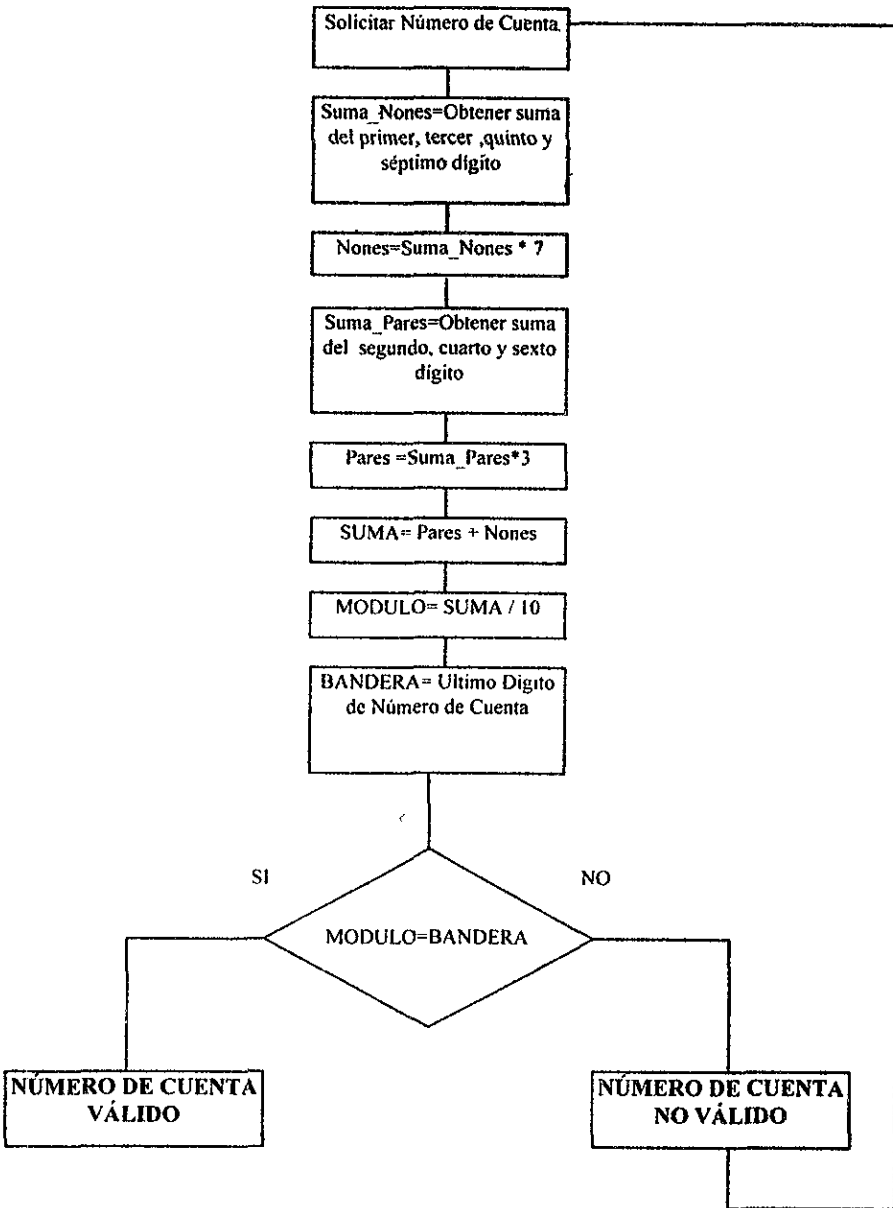
r2 = MsgBox(m2, f2, titulo)
mitabla.Close
Exit Sub

End If
)BUIF
DocCmd OpenForm "Ver tira de Materias", , , "In_cuenta)= formularios\Tabla de materias y grupos\ICUENTA"
mitabla.Close
Exit Sub
End Sub

Sub Botón50_Click ()
End Sub
Sub Imprimir_Click ()
    On Error GoTo Err_impresor_Click
    DocCmd OpenReport "Lista de tutorías individuales"
    Exit Sub
Err_impresor_Click:
    m3 = "Se canceló impresión!"
    f3 = 48 + 0 + 0
    r3 = MsgBox(m3, f3, titulo)
    Exit Sub
End Sub
Sub Imprimir_lista_gener_Click ()
    On Error GoTo Err_impresor2_Click
    DocCmd OpenReport "Lista general de tutorías"
    Exit Sub
Err_impresor2_Click:
    m3 = "Se canceló impresión!"
    f3 = 48 + 0 + 0
    r3 = MsgBox(m3, f3, titulo)
    Exit Sub
End Sub
Sub Seleccionar_Profesor_AfterUpdate ()
Dim lca As Database, mitabla As Recordset
Set lca = dbengine.workspaces(0).databases(0)
Set mitabla = lca.OpenRecordset("ALUMNOS", DB_OPEN_TABLE)
mitabla.Index = "TUTOR"
mitabla.Seek "a", Me!Seleccionar_Profesor!
If mitabla.NoMatch Then
    m1 = "Profesor sin alumnos bajo su tutela!"
    f1 = 48 + 0 + 0
    r1 = MsgBox(m1, f1, titulo)
Else
    DocCmd ApplyFilter "para consultar tutores a cargo"
End If
mitabla.Close
End Sub
Sub Seleccionar_Profesor_NoInList (NewData As String, Response As Integer)
Response = DATA_ERRORCONTINUE
msg1 = "El Nombre del Profesor debe coincidir con un valor de la lista!"
dgdDef1 = 48 + 0 + 0
r2d1 = MsgBox(msg1, dgdDef1, titulo)
End Sub

```

ALGORITMO PARA VALIDAR NÚMEROS DE CUENTA



BIBLIOGRAFÍA

📖 Diseño de Sistema de Información: Teoría y Práctica
Burch, John G. Y Grunitski, Gary.
Limusa, 1992

📖 Análisis y Diseño de Sistemas
Kenneth E. Kendall, Julie E. Kendall
Prentice-Hall, 1991

📖 Sistemas de Información Administrativa
Murdick, Robert G.
Prentice Hall

📖 Essentials of Management Information Systems
Laudon, Kenneth C. Y Laudon, Jane P.
Prentice- Hall, 1995

📖 Análisis y Diseño de Sistemas de Información
Seen, James A.
McGraw-Hill, 1995

📖 Ingeniería de Software, Un enfoque práctico
Pressman, Roger S.
McGraw-Hill

📖 Ingeniería de Software
Failley, Richard
McGraw-Hill

📖 Database System Concepts
Henry F. Korth y Silbertschatz
McGraw-Hill, 1991

📖 Sistemas de Bases de Datos
Alice Y. H. Tsai
Prentice- Hall, 1990.

📖 Object-Oriented Databases A semantic Data Model Approach
Gray, P.M.D. and Krishnarao, G.K. and Paton, N.W.
Prentice-Hall International Series in Computer Science, Herfordshire, UK, 1992.

📖 Microsoft Access para Windows, a su alcance.
Sánchez Navarro, José Daniel
McGraw-Hill, 1994

- 📖 **Cómo usar Microsoft Access**
Chordá, Ramón M.
Addison-Wesley Iberoamericana, 1994.
- 📖 **Access para Windows 95, paso a paso**
Schneider, Bob.
Prentice- Hall, 1996.
- 📖 **Todo el Ms-Access 2.0 para Windows: en un solo libro**
Martins Carrizo, Marta Beatriz.
Trillas, 1996
- 📖 **Guía de campo Access 95**
Pascual González, Francisco.
Alfaomega, 1997.
- 📖 **El gran libro para Access 95**
Bar, Jurgen.
Marcombo, 1996.
- 📖 **Access para Windows 95**
Peña, Jaime
Paraninfo, 1996.
- 📖 **Fundamentos de Access 7.0**
Rosskamp, Alfred.
Marcombo, 1996.
- 📖 **Todo el Ms-Access 95: Versión 7.0 en un solo libro**
Martins Carrizo, Marta Beatriz.
Trillas, 1996
- 📖 **Access Office 97, Guía rápida**
González Mangas, Antonia
Paraninfo, 1997.

REFERENCIAS

- [1] Internet <http://www.microsoft.com>
- [2] Internet
<http://www.windows.com/office/office97/documents/o97revgd/pegs/acc97peg>
- [3] Conéctate al mundo de Internet
Krol, Ed., García, Hugo Edmundo, Tr.
Editorial McGraw-Hill México, 1995.
- [4] Word Wide Web Unleashed
Ransall, Neil
Editorial SAMS
Indianapolis, 1994.
- [5] Aprendiendo HTML para Web en una semana
Lemy, Laura
Editorial Prentice-Hall Hispanoamericana
México, 1994.

GLOSARIO**ALGORITMO**

Procedimiento matemático o lógico para resolver un problema.

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 alto/bajo, sí/no se conocen como bits.

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.

CURSOR

Un carácter parpadeante en pantalla que muestra dónde aparecerá el siguiente carácter.

FTP

Protocolo de transmisión de archivo, protocolo que permite que el usuario transfiera archivos de una ubicación a otra por medio de Internet) y HTTP (Protocolo de transmisión de hipertexto, protocolo de Internet subyacente que distribuye información por medio de World Wide Web. El protocolo permite al usuario utilizar un programa cliente para obtener acceso a una dirección URL (o hacer clic en un hipervínculo) y recuperar texto, gráficos, sonido y otra información digital de un servidor 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.

HTML

Lenguaje de Marcado de Hipertexto (Hiptertext 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 demás documentos).

ICONO

En una interfaz gráfica para usuario (GUI), símbolo en pantalla que representa un programa, un archivo de datos u otra entidad o función de la computadora.

INTERNET

“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.

OLE (Object linking and embedding, vinculación e incrustación de objetos)

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. A diferencia del Portapapeles, OLE permite crear una vinculación dinámica entre el documento fuente (creado por una aplicación de servidor) y el documento destino (creado por una aplicación de cliente), de modo que las modificaciones que le haga al primero se reflejen en el segundo. 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. Este archivo fuente puede editarse con la frecuencia que se necesite, pero su localización no debe cambiar. 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 (o parte de éste) en el archivo destino, lo que da por resultado un archivo compuesto.

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.

REINGENIERIA

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 DE COMPUTACION

Instalación completa de una computadora (incluyendo periféricos, como unidades de disco duro y flexible, monitor, ratón, sistema operativo, software e impresora), en la que todos los componentes están diseñados para trabajar en conjunto.

SISTEMA OPERATIVO

Programa de control maestro que maneja las funciones internas de la computadora y le proporciona los medios para controlar la operatividad de la misma.

VISUAL BASIC

Un lenguaje de programación de alto nivel para el desarrollo de aplicaciones diseñadas para ejecutarse bajo Microsoft Windows 95.
SQL (Structured Query Language, Lenguaje de Consulta Estructurado), es un lenguaje de consulta desarrollado por IBM y de uso extendido en sistemas de macrocomputadoras y minicomputadoras.

WORD WIDE WEB

Es un sistema para explorar Internet mediante hipervínculos. Los campos Hipervínculo contienen texto o combinaciones de texto y número almacenados como texto que se utilizan para las direcciones de los hipervínculos. World Wide Web se puede abreviar como Web .

WORKSTATION (Estación de trabajo.)

En una red de área local (LAN), es una computadora de escritorio que corre programas de aplicación y sirve como un punto de acceso a la red.