

46  
2ij



UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO

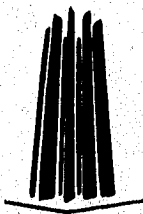
ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
"ARAGON"

IMPLEMENTACION DE UN SISTEMA PARA LA  
ADMINISTRACION Y CONTROL ESCOLAR EN LA  
FACULTAD DE ESTUDIOS SUPERIORES (FES)  
ZARAGOZA.

**T E S I S**

QUE PARA OBTENER EL TITULO DE:  
**INGENIERO EN COMPUTACION**  
P R E S E N T A ;  
MONCAYO LOPEZ LUIS MIGUEL

DIRECTOR DE TESIS: M. EN I. JUAN CARLOS ROA REIZA.



MEXICO

1996

**TESIS CON  
FALLA DE ORIGEN**

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

**TESIS**

**COMPLETA**

## **A MIS PADRES.**

Por mostrarme el camino hacia el éxito, por enseñarme a valorar la vida, por ayudarme a levantarme de tropiezos difíciles, quiero mostrar mi más sincero agradecimiento a las personas que hicieron posible este sueño, a quienes con su paciencia, amor, y sabiduría crearon el águila que hoy puede volar y contemplar el mundo con un horizonte lleno de fé y esperanzas en un futuro mejor.

Por esto y muchas cosas más gracias Papá y Mamá.

Luis Miguel.



**IMPLEMENTACIÓN  
DE UN SISTEMA  
PARA LA  
ADMINISTRACIÓN  
Y CONTROL  
ESCOLAR EN LA  
FACULTAD DE  
ESTUDIOS  
SUPERIORES (FES)  
ZARAGOZA.**

**OBJETIVO:**

**Diseñar un sistema de administración y control escolar para facilitar el trabajo administrativo del plantel y brindar un mejor servicio a la comunidad estudiantil en la FESZ, utilizando nuevas tecnologías**

---

# **ÍNDICE**

---

---

**CAPÍTULO I "FUNDAMENTACIÓN TEÓRICA" \_\_\_\_\_ 1**

**1.1 Características y funcionamiento de las bases de datos  
relacionales. \_\_\_\_\_ 3**

**1.2 Bases de la Ingeniería del Software. \_\_\_\_\_ 59**

**1.2.1 Modelo Ambiental. \_\_\_\_\_ 60**

a) **Objetivo. \_\_\_\_\_ 61**

b) **Lista de eventos. \_\_\_\_\_ 61**

c) **Diagrama de Contexto. \_\_\_\_\_ 62**

d) **Flujo de datos. \_\_\_\_\_ 65**

**1.2.2 Modelo de Comportamiento \_\_\_\_\_ 66**

a) **Diccionario de Datos \_\_\_\_\_ 66**

b) **Diagrama de flujo de datos \_\_\_\_\_ 86**

c) **Archivos \_\_\_\_\_ 101**

d) **Mini-especificaciones \_\_\_\_\_ 125**

---

Índice

---

1.3	Características del manejador de bases de datos ACCESS	133
1.4	Características de Visual Basic 3.0 Profesional	145
1.5	Seguridad de datos	161

**CAPÍTULO II "PLANTEAMIENTO DE LA PROBLEMÁTICA Y  
PROPUESTA DE SOLUCIÓN" 175**

2.1	Antecedentes (¿Cómo está operando actualmente el sistema de Servicios Escolares de la Fesz?)	177
2.1.1	Esquema actual de inscripción	177
2.1.2	Esquema de cómo se pretende que opere el sistema	181
2.2	Estrategia de solución	185
2.2.1	Plan de trabajo	185
2.2.2	Recopilación de Información	191
2.2.3	Clasificación de la Información	197

---

---

2.3 Análisis de los requerimientos del usuario. 199

2.4 Diagrama de Descomposición Funcional. 205

2.5 Opciones de Solución. 211

**CAPÍTULO III "DESARROLLO E IMPLEMENTACIÓN**

**DEL SISTEMA" 221**

3.1 Especificaciones del Análisis y Diseño (Establecimiento de Estándares). 223

3.2 Diagrama de Flujo de Datos. 233

3.3 Diccionario de Datos. 251

3.4 Diagrama Entidad-Relación. 255

3.5 Diseño e implementación de los diversos módulos de programación en Visual Basic 3.0. 257

---

*índice*

---

<b>3.6 Pruebas e Integración de los módulos.</b>	<b>266</b>
3.6.1 Prueba de unidad.	272
3.6.2 Prueba de la caja blanca.	295
3.6.3 Prueba de la caja negra.	297
3.6.4 Prueba de integración.	299
3.6.5 Prueba de validación.	304
3.6.6 Repaso de la configuración.	306
3.6.7 Pruebas Alfa y Beta.	308
3.6.8 Integración de módulos.	310

<b>CONCLUSIONES</b>	<b>315</b>
---------------------	------------

**DOCUMENTACIÓN:**

- **Manual de usuario.** \_\_\_\_\_ **I**
- **Manual de mantenimiento.** \_\_\_\_\_ **XXXV**
- **Apéndice "A" Planes de estudio.** \_\_\_\_\_ **A-1**
- **Apéndice "B" Reportes del Sistema:**
  - **Listas de grupo.** \_\_\_\_\_ **B-1**
  - **Declaración Grupo/Profesor.** \_\_\_\_\_ **B-11**
  - **Hitoria Académica.** \_\_\_\_\_ **B-15**
  - **Avance Académico.** \_\_\_\_\_ **B-19**

---

- 
- Apéndice "C" Manejo la Base de Datos en VB 3. \_\_\_\_\_ C-1
  - Apéndice "D" Ejemplo de Desarrollo en VB. \_\_\_\_\_ D-1
  - Apéndice "E" Código Fuente \_\_\_\_\_ E-1

## **BIBLIOGRAFÍA**

## **GLOSARIO.**

---

---

# ***CAPÍTULO I***

## ***“FUNDAMENTACIÓN TEÓRICA”***

**OBJETIVO:**

Definir los conceptos básicos de los elementos computacionales que serán utilizados en el diseño del sistema.

---



## 1.1 CARACTERÍSTICAS Y FUNCIONAMIENTO DE LAS BASES DE DATOS RELACIONALES.

Una base de datos es una fuente central de datos significativos, los cuales son compartidos por numerosos usuarios para diversas aplicaciones.

Por su alto grado de complejidad muchas bases de datos parecen crecer sin límites previsibles. A menos que los diseñadores tengan un claro concepto de lo que está ocurriendo, si no, estos sistemas se transformarán en una maraña de datos e interrelaciones.

Es posible evitar la complejidad a que dan lugar las estructuras plex y ramificadas recurriendo a técnicas de normalización. Estas técnicas fueron ideadas y recomendadas por E.F. Codd. Estos principios que Codd aplica para el diseño de bases de datos, se refieren a la vista de datos propia del usuario, es decir, a la descripción lógica de éstos. Es importante observar que las descripciones lógicas no son directamente aplicables a las representaciones físicas.

Por medio de un software de administración adecuado, la vista de los datos que se presentan al usuario se mantendrán independientes de la representación física, de modo que ésta y el hardware, podrán alterarse sin afectar la descripción lógica de los datos que interesan al usuario.

Para el diseño de la descripción lógica debemos procurar una manera de describir los datos de forma que:

- 1.- Sea entendida fácilmente por los usuarios que no tienen preparación previa como programadores.

## Capítulo I

---

- 2.- Haga posible ampliar la base de datos sin modificación de la estructura lógica existente y , por tanto, sin modificación de los programas de la aplicación.
- 3.- Permita la máxima flexibilidad en la formulación de interrogantes de forma no prevista, o espontánea, en las terminales.

El manejo y organización en una base de datos puede ser visto en 4 niveles de Arquitectura que van desde el nivel de Organización Lógica hasta su correspondiente en el nivel de Organización Física, desde el más independiente del equipo hasta el más dependiente, éstos niveles se describen a continuación:

**Cuarto Nivel:** Es el más alto, es donde se hacen sólo las consideraciones lógicas de datos y el análisis de necesidades del usuario, de los datos con que cuenta y los que necesita, se especifica el flujo de la información y sus transformaciones. Posteriormente se definen entidades u objetos y sus relaciones funcionales. A partir de éstas entidades y relaciones funcionales se construye o modela la base utilizando los modelos lógicos de organización. Cabe mencionar que éste nivel es el más independiente del computador donde se va a implementar la base de datos.

**Tercer Nivel:** Orienta a la implementación de los modelos de organización, se abordan acciones orientadas al desempeño, se emplean a tomar en cuenta las limitaciones de los dispositivos físicos. Aquí se definen las estrategias de acceso para mejorar el desempeño en el manejo de datos, en este nivel se definen que organización de archivos se utilizará, si son de acceso secuencial, random, etc. A los niveles tres y cuatro se les llama interface física-lógica.

**Segundo Nivel:** Refleja alternativas de organización física y formatos de almacenamiento de datos en dispositivos de almacenamiento lineal para cada una de las organizaciones definidas en el nivel tres. Algunos autores sólo reconocen tres niveles y manejan como uno solo al tres y al dos.

**Primer Nivel:** Es más dependiente que los anteriores del equipo donde se implementa la base de datos, se refieren a las características y aspectos particulares de los dispositivos físicos de almacenamiento. En este nivel se ven aspectos como: la organización de los dispositivos en subdivisiones tales como bloques y sectores; el mapeo de datos dentro de estas subdivisiones; el almacenamiento de áreas de sobreflujo y características dependientes de cada máquina, así como la localización de datos de control y verificación de integridad.

### ESQUEMAS Y SUBESQUEMAS

Antes de intentar la descripción de las representaciones físicas de las relaciones entre datos, debemos considerar la manera en cómo los usuarios de la base (usuarios terminales y programadores de aplicaciones) ven esas relaciones.

La vista que el usuario tiene de los datos debería adoptar siempre la forma que resulta más conveniente para él y sus asociados (en el presente y en el futuro), quedando a cargo del software de administración de datos, al hacer la "traducción" entre esta organización lógica y la organización física, cualquiera que sea la elegida, teniendo en cuenta la eficiencia del sistema.

## ESQUEMAS

Llamamos esquema a la descripción lógica de la base de datos, El esquema es un diagrama de los tipos de datos que se usan. Proporciona los nombres de las entidades y sus atributos y especifica las relaciones que existen entre ellos. Es un marco en el que se inscriben los valores de los ítems de datos.

Cuando el marco del esquema se completa con valores de ítems, entonces hablamos de una instancia del esquema.

### RELACIONES Y REFERENCIAS CRUZADAS

Los esquemas se presentan a menudo a modo de diagramas de bloques. La figura. 1.1.1 ilustra un esquema típico.

Las líneas llenas que unen ciertos bloques representan relaciones. Así, el registro **ORDEN-DE-COMPRA** está conectado con registros **RENLÓN-DE-COMPRA**, ambos en orden de compra. El registro **PROVEEDOR** está conectado con los registros de **COTIZACIÓN**, en los que se indican los artículos que el proveedor puede suministrar y las cotizaciones que ha presentado.

Las líneas discontinuas representan referencias cruzadas. **NOMBRE-DE-ARTÍCULO** y **DETALLES-DE-LOS-ARTÍCULOS** no aparecen en el registro **RENLÓN-DE-COMPRA** sino en un registro separado, **ARTÍCULOS**. De igual modo, **NOMBRE-DEL-PROVEEDOR** y **DIRECCIÓN-DEL-PROVEEDOR** no se encuentran en el registro **ORDEN-DE-COMPRA** sino en el registro independiente **PROVEEDOR**.

Las relaciones (líneas continuas) indicadas en el esquema agregan una información que no es inherente a los ítems de datos indicados en el esquema.

## ESQUEMAS

Llamamos esquema a la descripción lógica de la base de datos, El esquema es un diagrama de los tipos de datos que se usan. Proporciona los nombres de las entidades y sus atributos y especifica las relaciones que existen entre ellos. Es un marco en el que se inscriben los valores de los ítems de datos.

Cuando el marco del esquema se completa con valores de ítems, entonces hablamos de una instancia del esquema.

### RELACIONES Y REFERENCIAS CRUZADAS

Los esquemas se presentan a menudo a modo de diagramas de bloques. La figura. 1.1.1 ilustra un esquema típico.

Las líneas llenas que unen ciertos bloques representan relaciones. Así, el registro **ORDEN-DE-COMPRA** está conectado con registros **REGLÓN-DE-COMPRA**, ambos en orden de compra. El registro **PROVEEDOR** está conectado con los registros de **COTIZACIÓN**, en los que se indican los artículos que el proveedor puede suministrar y las cotizaciones que ha presentado.

Las líneas discontinuas representan referencias cruzadas. **NOMBRE-DE-ARTÍCULO** y **DETALLES-DE-LOS-ARTÍCULOS** no aparecen en el registro **REGLÓN-DE-COMPRA** sino en un registro separado, **ARTÍCULOS**. De igual modo, **NOMBRE-DEL-PROVEEDOR** y **DIRECCIÓN-DEL-PROVEEDOR** no se encuentran en el registro **ORDEN-DE-COMPRA** sino en el registro independiente **PROVEEDOR**.

Las relaciones (líneas continuas) indicadas en el esquema agregan una información que no es inherente a los ítems de datos indicados en el esquema.

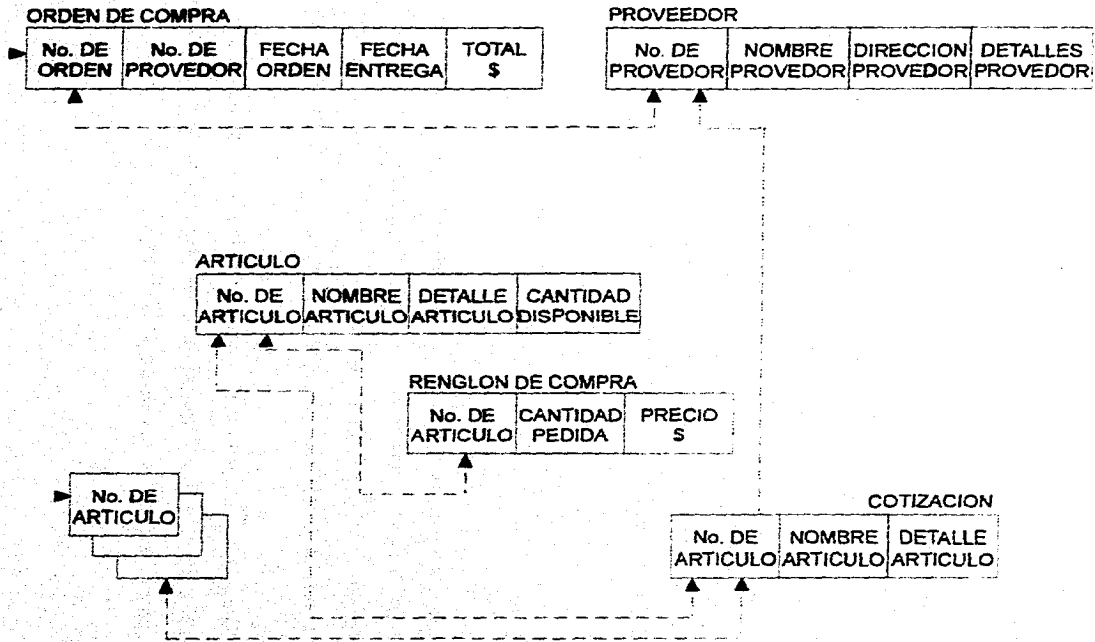


Figura 1.1.1 Esquema. Las líneas discontinuas entre bloques representan referencias cruzadas.

## Capítulo I

---

Por ejemplo, no hay nada en el registro ORDEN-DE-COMPRA de la Figura. 1.1.1 que diga qué artículos y en qué cantidades se compran. Esta Información se completa sólo cuando se vinculan los registros RENGLÓN-DE-COMPRA con los registros ORDEN-DE-COMPRA que corresponden.

Las referencias cruzadas, no aportan nueva información. El registro ORDEN-DE-COMPRA no contiene el nombre del proveedor ni su dirección, pero sí el número del proveedor, de modo que aprovechando esto pueden recabarse el nombre y la dirección del proveedor del archivo PROVEEDOR. Las líneas de referencia cruzada en el esquema indican vinculaciones existentes en el archivo que permiten localizar los datos más fácilmente.

### **SUBESQUEMAS**

El término esquema se aplica al diagrama general de todos los tipos de ítems de datos y de registros almacenados en la base de datos. El término subesquema, en cambio, se refiere a la vista que tiene el programador de aplicaciones de los datos que utiliza. Del mismo esquema se pueden derivar muchos subesquemas.

La Figura. 1.1.2. muestra los subesquemas correspondientes a dos programas de aplicación. Cada programador tiene su propia vista de los datos, pero ambos derivan su subesquema del esquema de la Figura. 1.1.1

## SUBESQUEMA PARA EL PROGRAMADOR A ARCHIVO-MAESTRO DE ORDENES DE COMPRA

No. ORDEN	No. PROVEEDOR	NOMBRE PROVEEDOR	FECHA DE ENTREGA	FECHA DE ORDEN	TOTAL "
-----------	---------------	------------------	------------------	----------------	---------

DETALLES ORDEN DE COMPRA

No. ARTICULO	NOMBRE ARTICULO	CANTIDAD PEDIDA	PRECIO

Figura 1.1.2.A Dos diferentes programadores necesitan diferentes archivos, se ilustran sus respectivos subesquemas, ambos derivados del esquema de la figura 1.1 (Esquema A).



## SUBESQUEMA PARA EL PROGRAMADOR B ARTICULOS - EN - EXISTENCIA

No ARTICULO	NOMBRE ARTICULO	CANTIDAD DISPONIBLE
----------------	--------------------	------------------------

DETALLES PEDIDOS-PENDIENTES

No. ARTICULO	NOMBRE ARTICULO	CANTIDAD PEDIDA	FECHA DE ORDEN	FECHA DE ENTREGA

Figura 1.1.2.B (Esquema B)

El registro MAESTRO-DE-ORDEN-DE-COMPRA del programador A contiene ahora NOMBRE-DEL-PROVEEDOR y DIRECCIÓN-DEL-PROVEEDOR, mientras que el registro DETALLE-DE-ORDEN-DE-COMPRA contiene NOMBRE-DE-ARTÍCULO, CANTIDAD-PEDIDA y PRECIO. El registro DETALLE-DE-ÓRDENES-PENDIENTES del programador B contiene ítems de datos tomados de tres diferentes registros de la Figura. 1.1.1

Los programadores de aplicación no tienen por qué conocer la totalidad del esquema. En cambio, el administrador de datos debe asegurar que los esquemas que aquéllos usan son derivables del esquema. El software de administración de datos deriva los datos de los subesquemas de los datos del esquema, automáticamente, y los pone a disposición del programa de aplicación.

### DESCRIPCIÓN DE DATOS

Ni los esquemas ni los subesquemas reflejan la forma en que los datos se almacenan físicamente. Tenemos así, en total, cuatro distintas descripciones de datos:

**1.- El subesquema.**

El diagrama de una porción de datos, orientado a satisfacer uno o más programas de aplicación, esto es, una organización de archivos del programador.

**2.- La descripción lógica global de la base de datos o esquema.**

El diagrama lógico de la base de datos completa. Es ésta la vista general de los datos como los contempla el administrador o los analistas de sistemas que usan toda la base.

**3.- La descripción de la base de datos.**

Es un diagrama de la distribución física de los datos en los dispositivos de almacenamiento que es utilizado por los programadores y diseñadores encargados de la ubicación de los datos en el hardware y técnicas de compactación para lograr así un óptimo desempeño.

**4.- El usuario de terminal.**

Tiende a ser una persona que conoce poco o nada a cerca de las técnicas del procesamiento de datos. La lista de los datos que el sistema le presenta debe asemejarse tanto como sea posible a la que es inherente al trabajo del usuario. El modo de presentación de esta vista depende del diseño del diálogo hombre-terminal. La figura 1.1.3 resume estas cuatro distintas vistas de los datos.

Existen diferentes Arquitecturas que manejan diferentes niveles de organización por ejemplo: La Arquitectura CODASYL maneja 3 niveles con su lenguaje asociado, más un lenguaje para procesar esos datos que son:

- 1.- Lenguaje de definición de datos a nivel de esquema.
- 2.- Lenguaje de definición de datos a nivel de subesquema.
- 3.- Lenguaje de Manipulación de datos.
- 4.- Lenguaje de control de dispositivos/medios o lenguaje de descripción de almacenamiento de datos.

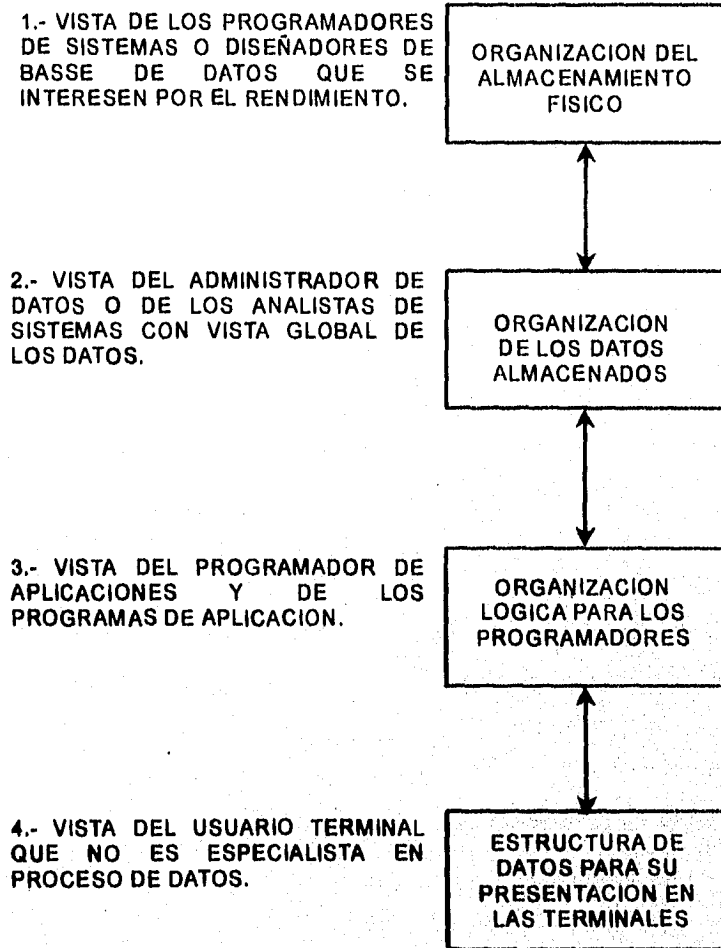


Figura 1.1.3 Cuatro vistas diferentes de los datos.

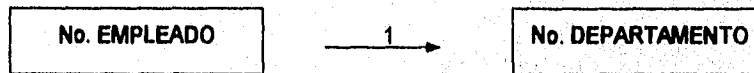
## Capítulo I

Otra Arquitectura es la ANSI/PARC en la cual se contemplan 3 niveles:

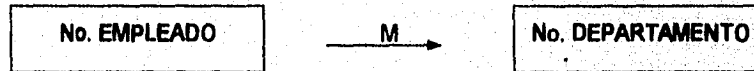
- 1.- El nivel externo.: En este nivel se aborda la definición de subesquemas de la base de datos.
- 2.- El nivel conceptual: En este esquema se implanta la base de datos en su forma más general se define el esquema de la base de datos.
- 3.- El nivel interno: En este nivel se implantan las definiciones de almacenamiento.

La relación existente entre dos tipos de datos puede ser simple o compleja.

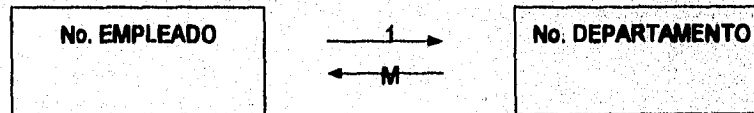
La correspondencia simple se representa por medio de una flecha con el dígito 1 escrito sobre ella:



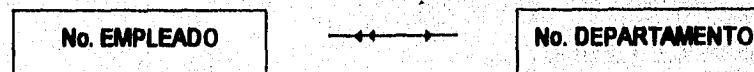
Mientras que la correspondencia de uno a muchos se indica con una flecha con la letra M en lugar del dígito 1:



Ambas representaciones pueden fundirse en una:



Otro estilo de representación que evita tener que escribir dígitos y letras sobre un diagrama ya complejo consiste en indicar las relaciones simples por medio de una flecha simple y las relaciones complejas mediante una doble flecha:

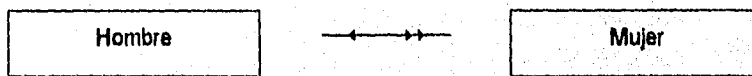


Hay cuatro tipos de relación entre dos conjuntos de ítems A y B. La correspondencia de A a B puede ser simple y la recíproca compleja. O la correspondencia de A a B puede ser compleja y recíproca simple. Ambas correspondencias pueden ser simples o ambas correspondencias pueden ser complejas. La figura 1.1.4 ilustra cuatro posibles correspondencias entre HOMBRE y MUJER sobre la base de la relación <<matrimonio>>.

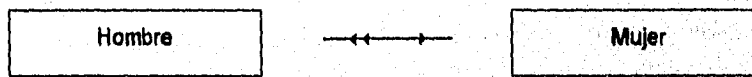
**Matrimonio Convencional:**



**Poligamia:**



**Pollandria:**



**Matrimonio grupal:**

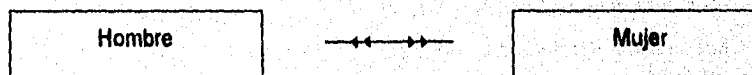


Figura 1.1.4 Formas de representar correspondencias simples y complejas, basadas en el empleo de la relación <<matrimonio>>.

### COMO DIBUJAR UN ESQUEMA.

Para dibujar un esquema se deben seguir las siguientes reglas:

- 1.- El diagrama debe distinguir claramente los nombres de registro de los ítems de datos.
- 2.- Debe quedar clara la reunión de ítems de datos en agregados de datos.
- 3.- Debe aclararse la distinción entre agregados de datos y registros.
- 4.- Deben mostrarse los identificadores de registro.
- 5.- El diagrama debe indicar claramente las relaciones simples y las complejas.
- 6.- Las relaciones deben diferenciarse de las referencias cruzadas.
- 7.- Las relaciones entre registros deben nominarse o numerarse.
- 8.- No deben utilizarse nombres duplicados.

Se tiene la figura 1.1.5 donde se muestra una forma de dibujar los esquemas: Los registros aparecen en bloques de línea llena, mientras que los bloques correspondientes a los agregados de datos repetitivos se dibujan con línea discontinua; los nombres de los registros y los agregados de datos se anotan fuera de los bloques correspondientes. Los nombres relacionales identificadores de registro se subrayan.

Las flechas dobles que indican relaciones de uno a muchos no son estrictamente necesarias, pues la separación entre niveles en este diagrama indica precisamente esa relación. Pero suelen resultar indispensables en diagramas más complejos.

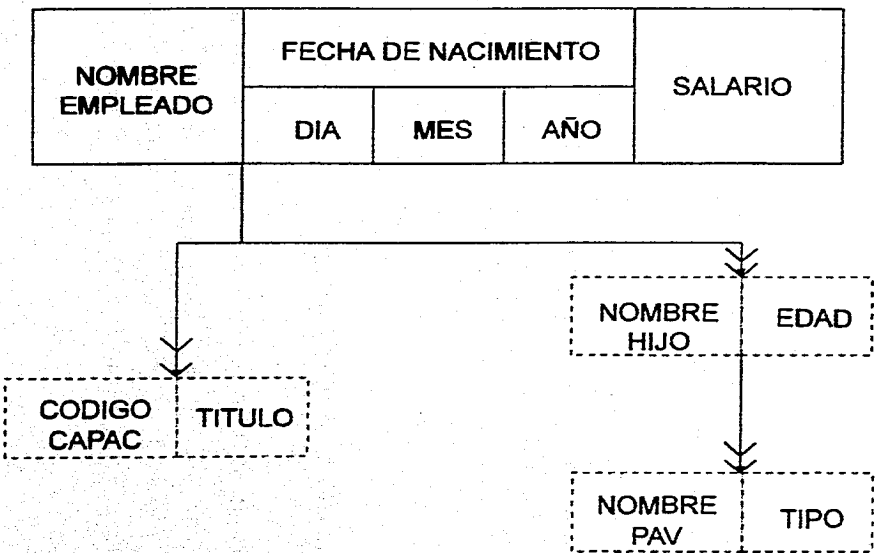


Figura 1.1.5 De acuerdo con las reglas mencionadas se muestra como diseñar un esquema.



## MODELOS LÓGICOS

Una base de datos es un modelo en computadora de un sistema del mundo real. El contenido de la base de datos corresponde al estado del sistema de aplicación, mientras que los cambios a la base de datos corresponden a eventos del sistema. En general, si el modelo puede ser descrito en términos de estructuras naturales, el trabajo se simplificará y esto provocará que se realicen formulaciones más fáciles de consultar.

El mecanismo formal utilizado para expresar la estructura lógica de los datos y la semántica asociada, es llamado modelo lógico de datos. Para que un modelo sea efectivo, debe representar lo más cercanamente posible los conceptos del mundo real que es usado para estructurar la información en una organización y son:

### 1.- MODELO JERÁRQUICO O DE ÁRBOL:

En las estructuras de árbol, se tienen nodos y una relación de jerarquía de un nodo padre a un nodo hijo. Donde el registro padre es llamado propietario y el registro hijo es llamado miembro o miembros. Las relaciones de propietario a miembro pueden ser 1:1 (uno a uno), 1:N (uno a muchos) pero no M:M (muchos a muchos).

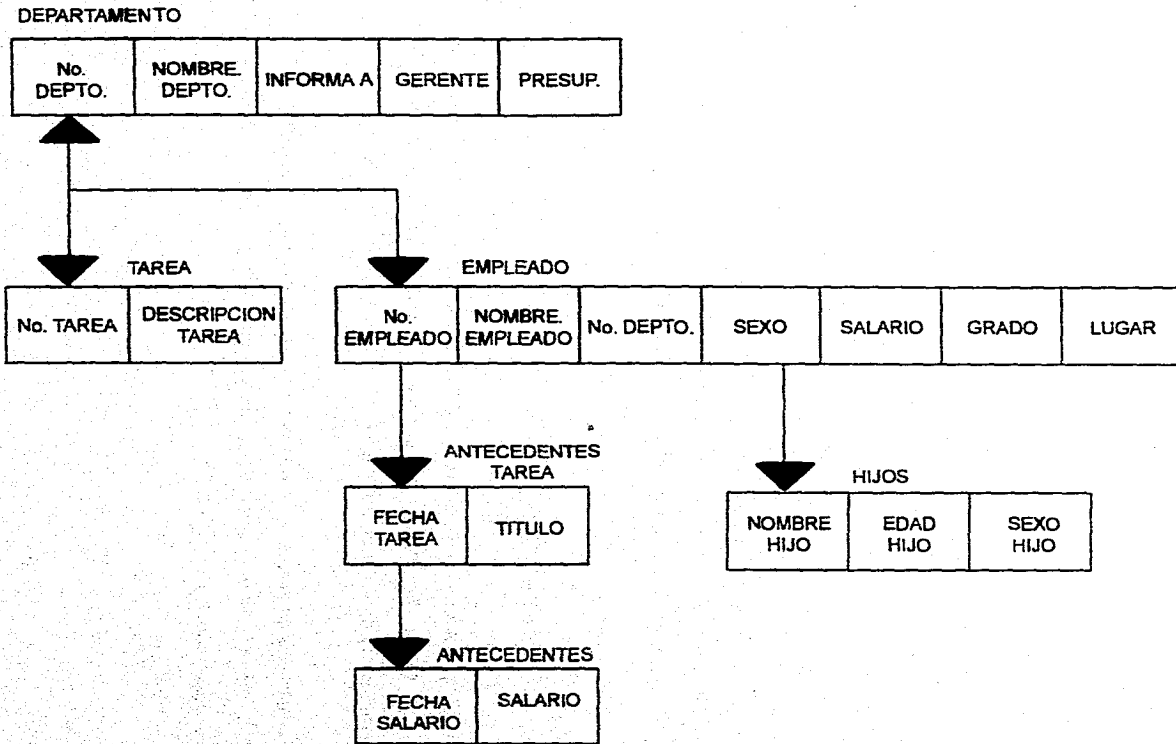


FIGURA 1.6 NORMALIZACION DE UNA ESTRUCTURA RAMIFICADA

UNA FORMA NORMALIZADA DE ESTE ESQUEMA

DEPARTAMENTO	(No-DEPTO, INFORMA-A, GERENTE, PROPUESTA).
TAREA	(No-DEPTO, No-TAREA, DESCRIPCION-TAREA).
EMPLEADO	(No-EMPLEADO, NOMBRE-EMPLEADO, No-DEPTO, SEXO, SALARIO, GRADO, LUGAR).
ANTECEDENTES-TAREA	(No-EMPLEADO, FECHA, TAREA, TITULO).
ANTECEDENTES-SALARIO	(No-EMPLEADO, FECHA-SALARIO, SALARIO).
HIJOS	(No-EMPLEADO, NOMBRE-HIJO, EDAD-HIJO, SEXO-HIJO).

Del mismo modo es posible normalizar un árbol. La figura 1.1.6 ilustra un árbol de cuatro niveles y la manera de reemplazarlo por medio de seis relaciones. La clave de cada relación puede incorporar la de la relación precedente en el árbol.

**2.- MODELO DE RED O RETICULAR (PLEX).**

En una estructura de árbol no se permite que un nodo hijo tenga más de un nodo padre. según CODASYL un miembro si puede tener más de un propietario, siempre y cuando cada uno esté en un conjunto diferente.

En este esquema aparecen una variedad de vinculaciones dirigidas, algunas con una flecha simple (correspondencia simple) y otras con flecha doble (correspondencia compleja).

En una relación entre datos un hijo tiene más de un padre, la relación no puede ser descrita por medio de un árbol o estructura jerárquica, se le describe en cambio por medio de una estructura plex o una red. En una estructura plex cualquier componente puede vincularse con cualquier otro. Como en el caso del árbol, la estructura plex puede ser descrita en términos de padres e hijos, y dibujada de tal manera que los hijos aparezcan debajo de los padres. Pero en la estructura plex un hijo puede tener más de un padre.

En muchas estructuras plex que representan relaciones entre tipos de registros o tipos de agregados de datos, la correspondencia entre padres e hijos es similar a la que existe en un árbol: la correspondencia de padre a hijo es compleja y la de hijo a padre simple. Llamaremos estructura plex simple a aquella que no tiene doble punta de flecha en los dos extremos de ninguna línea. Y llamaremos estructura plex compleja a aquella que tiene doble flecha en algún extremo de la línea.

La figura 1.1.7 muestra un esquema plex ejemplificado y una de las varias formas normalizadas.

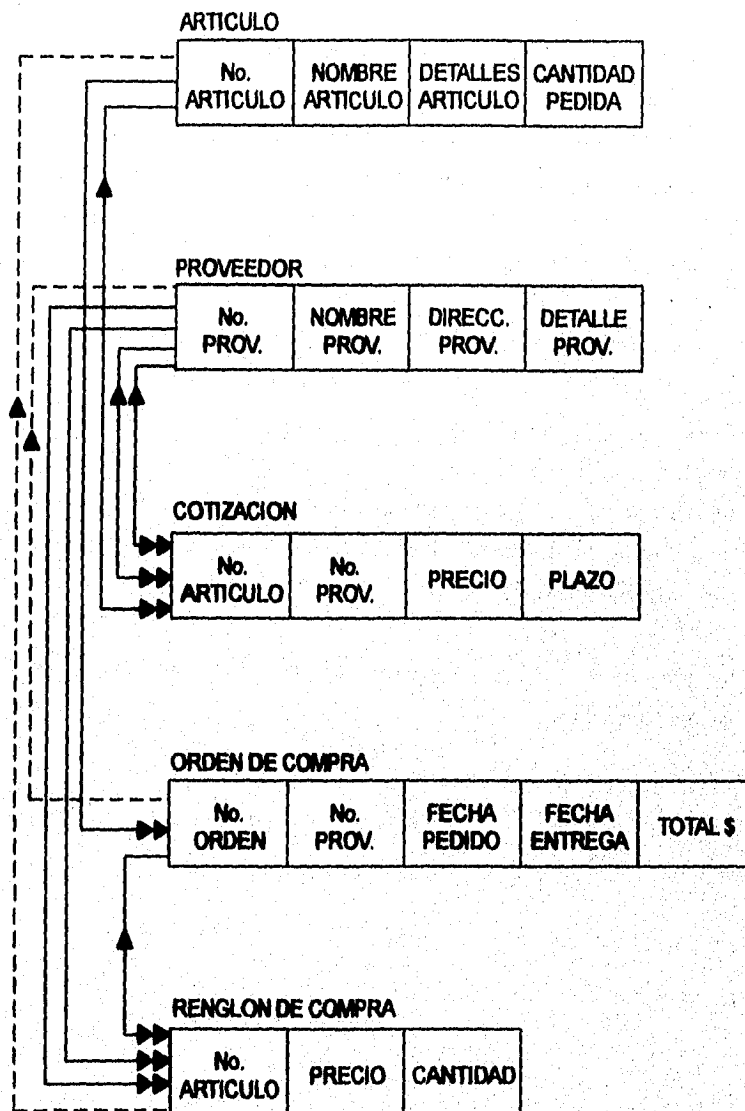


FIGURA 1.1.7 Esquema para estructura plex con forma normalizada.

Algunas de las vinculaciones de la figura 1.1.7, son innecesarias, la vinculación entre RENGLÓN-DE-COMPRA y ARTÍCULO es superflua, porque el registro RENGLÓN-DE-COMPRA contiene el atributo No-DE-ARTÍCULO, que podría utilizarse para tener acceso al registro ARTÍCULO.

En la mayoría de las bases de datos existen muchas asociaciones de referencias cruzadas que se prestan a su inclusión en el esquema, la representación de todas ellas mediante líneas y flechas resultaría en una maraña de conexiones.

Los enlaces incluidos gráficamente en un esquema implican por lo general una relación de grado 2, la línea rotulada RENGLONES-OC que va de ORDEN-DE-COMPRA al tupla RENGLÓN-DE-COMPRA, podría ser representada mediante la siguiente relación de grado 2:

RENGLONES-OC (No-ORDEN, No-ARTÍCULO).
---------------------------------------

Las vinculaciones paralelas pero diferentemente rotuladas, por ejemplo, PENDIENTE y DEMORADA entre PROVEEDOR y RENGLÓN-DE-COMPRA, implican por lo general una relación de grado 3 (PROVEEDOR, RENGLÓN-DE-COMPRA, ESTADO). Esos enlaces implican un tratamiento diferente de las relaciones de grado 2 (o de grado 3) en comparación con las relaciones de mayor grado, siendo que pueden ser tratadas idénticamente. Si estas son tratadas de distinta forma por la computadora, digamos con distintos métodos de direccionamiento, eso es responsabilidad del diseñador de la estructura física de los datos, no del diseñador de la estructura lógica.

### 3.- MODELO ENTIDAD-RELACIÓN.

Una relación es aquella asociación que describe el enlace entre entidades. El formato establecido para dibujar un diagrama de entidad-relación se presenta en la figura 1.1.8.

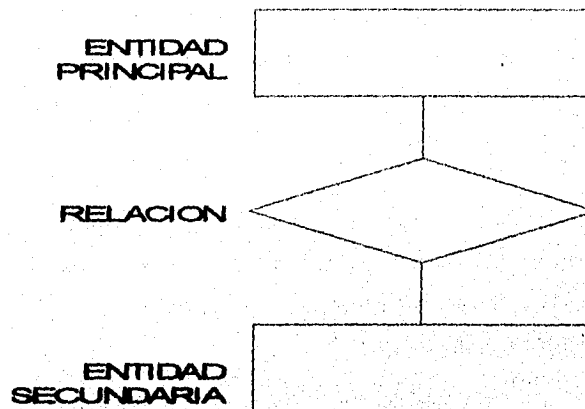


Figura 1.1.8 Un diagrama entidad-relación consiste en un rectángulo un diamante o rombo y otro rectángulo.

La figura 1.1.9 presenta los cuatro tipos de diagramas E-R diferentes. El primero corresponde a una relación de uno a uno (que abreviaremos 1:1). En este caso a cada EMPLEADO se le asigna sólo una EXTENSIÓN TELEFÓNICA y por cada EXTENSIÓN TELEFÓNICA hay un solo EMPLEADO. El segundo diagrama presenta la relación de muchos a uno (M:1). Un DEPARTAMENTO puede tener a muchos EMPLEADO(s), pero un EMPLEADO pertenece a un solo DEPTO.

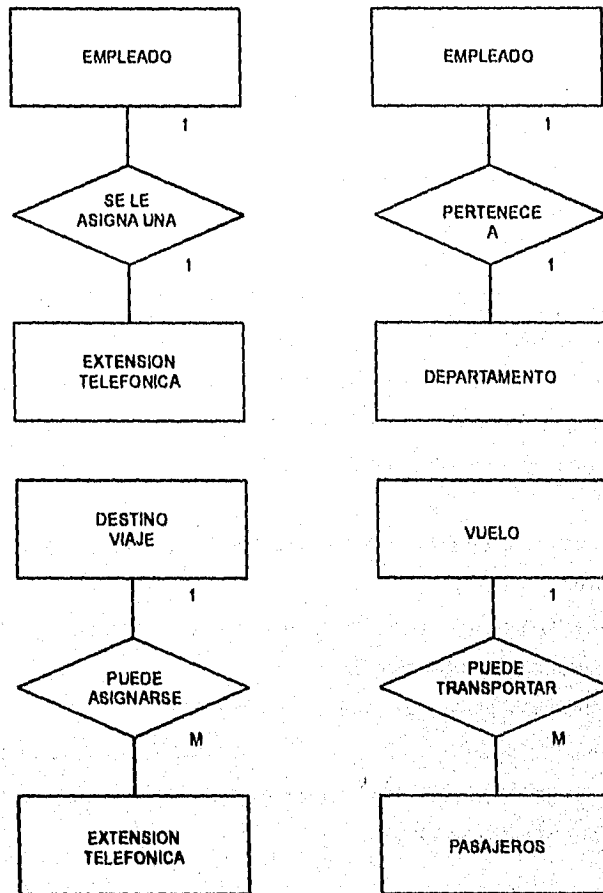


Figura 1.1.9 Los diagramas entidad-relación pueden ser:  
(1:1), (1:M), (M:1) y (M:M).



El tercer tipo de diagrama presenta la relación de uno a muchos (1:M). En este ejemplo se puede asignar un DESTINO DE VIAJE a numerosos PASAJEROS, pero un PASAJERO sólo tiene un DESTINO DE VIAJE final. Es válido distinguir entre las relaciones 1:M y M:1, ya que el diagrama siempre tendrá en la parte superior la entidad principal. Por último, el cuarto diagrama muestra una relación de muchos a muchos (M:M). Un VUELO transporta a numerosos PASAJEROS y un PASAJERO puede disponer de muchos VUELOS dentro de su itinerario.

A menudo, los diseñadores de sistemas hacen uso de los diagramas de entidad-relación para auxiliarse al modelar el archivo o la base de datos. Sin embargo, es más importante aún que el analista de sistemas se involucre pronto con las entidades y relaciones del sistema de organización. Al delinear algunos diagramas E-R básicos, el analista necesita:

1. Enumerar las entidades de la organización, con el fin de comprenderlas mejor.
2. Elegir las entidades clave para reducir la visión del problema a dimensiones manejables pero con significado.
3. Identificar la entidad principal.
4. Confirmar lo anterior mediante otros métodos de captura de información (investigación, entrevistas, entrega de cuestionarios, observación y desarrollo de prototipos).

Tomemos como ejemplo el caso del Festival del Lago, una organización que produce teatro de verano. El papel del analista de sistemas consistirá en estudiar el sistema actual y diseñar un sistema de boletaje. Conforme vaya avanzando en el problema, listará entidades tales como PÚBLICO, BUTACAS, FUNCIONES, FECHA, ACTORES y SUPLENTES.

Se debe tener una buena visión de la organización al listar las entidades y pronto se verá en la necesidad de reducir tal visión del problema. Aunque trabajan en el teatro, los ACTORES y los SUPLENTEs no tienen participación en el sistema de reservaciones. En cambio, BUTACAS, FUNCIONES y FECHA pudieran tratarse como atributos de una sola entidad que podría definirse como BOLETOS.

La siguiente etapa se involucra con la selección de la entidad principal. En este ejemplo, la elección con seguridad sería el PÚBLICO. Es el PÚBLICO quien reserva los BOLETOS del espectáculo; los BOLETOS no eligen al PÚBLICO. Cuando el PÚBLICO llega a la taquilla a recoger los BOLETOS, el PÚBLICO da su NOMBRE como identificación y no alguno de los atributos del BOLETO, como sería el número de asiento o la función.

En este ejemplo, la selección de la entidad principal parece ser muy simple, pero en otros problemas puede dificultarse, no se debe olvidar que cada persona de la organización la contempla de distinta forma.

El diagrama entidad-relación para el proyecto del Festival del Lago se muestra en la figura 1.1.10. Observe que el diagrama E-R puede ampliarse para incluir ciertos atributos descriptivos de cada entidad o relación. El PÚBLICO tiene atributos tales como su nombre, dirección, teléfono y número de tarjeta de crédito, los cuales conviene registrar para una correspondencia futura. El BOLETO tiene atributos tales como el número del boleto, fecha, hora, función, teatro y precio. La relación de reservación de boletos tiene atributos en común con las dos entidades. Estos atributos incluyen el nombre del asistente y el número del boleto.

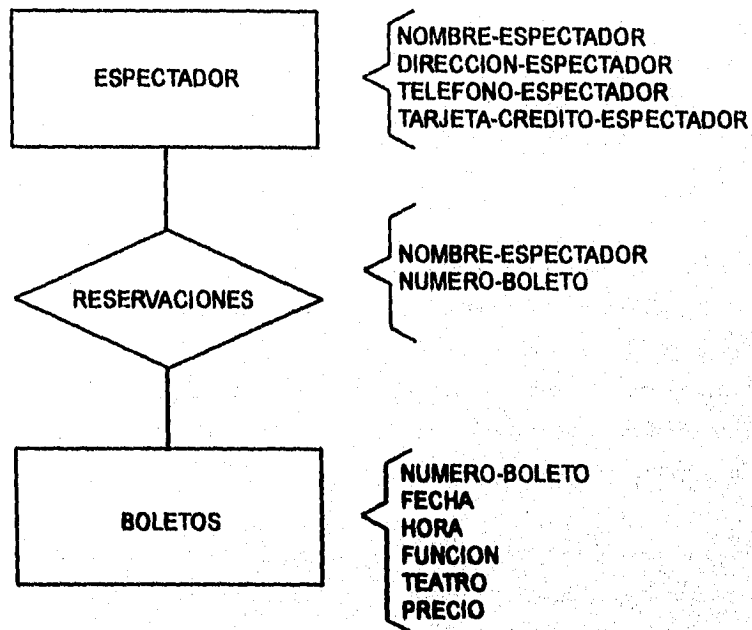


Figura 1.1.10 Diagrama E-R del Festival en el Lago, en el se muestran los atributos de cada entidad y de sus relaciones.

Es primordial que el analista de sistemas inicie la elaboración del diagrama E-R tan pronto como se incorpore a la organización. Será muy tarde esperar hasta que se diseñe la base de datos, ya que los diagramas E-R auxilian al analista a comprender el giro actual de la organización; ayudan a determinar las dimensiones de los problemas y auxilian a discernir si el problema abordado es el correcto. Los diagramas E-R requieren de una revisión continua, conforme se lleva a cabo el proceso de captura de datos.

Dentro de los factores de la organización que influyen en el diseño y el análisis de los sistemas de información, se encuentran los niveles de administración, el diseño de organización, el estilo de liderazgo, las consideraciones tecnológicas, la cultura organizacional y las relaciones humanas. A continuación, discutiremos cada uno de estos factores, así como las implicaciones que tienen sobre el análisis y el diseño de los sistemas de información.

#### 4.- MODELO RELACIONAL

Este modelo está basado en el álgebra relacional.

##### MANIPULACIONES DE RELACIONES.

El secreto de la flexibilidad de las bases de datos relacionales reside en la facilidad con que las relaciones se presentan. Se han definido una gran variedad de operadores para manipular relaciones. Por ejemplo, puede ser necesario descartar columnas en una relación. Esta operación, ilustrada por la fig. 1.1.11, se denomina proyección y su operador es el símbolo  $\lceil$ . La relación EMPLEADO(No-EMPLEADO, NOMBRE-EMPLEADO, No-DEPT, SALARIO, LUGAR) se proyecta mediante el enunciado:

EMP =  $\lceil$ EMPLEADO(No-EMPLEADO, NOMBRE-EMPLEADO, No-DEPT, SALARIO).

Para formar una nueva relación llamada EMP y que es la siguiente:

EMP(No-EMPLEADO, NOMBRE-EMPLEADO, No-DEPT, SALARIO)

## Capítulo I

---

De modo similar, el enunciado:

DEPT= EMPLEADO (No-DEPT, LUGAR)

Forma la relación:

DEPT (No-DEPT, LUGAR)

En este caso, el resultado contiene menos tuplas que el original porque se han eliminado todos los tuplas que resultaron duplicadas.

La inversa de dividir relaciones es la operación de conectar diferentes relaciones formando una unión natural. El símbolo \* representa el operador UNIÓN. El enunciado  $EMPLEADO = EMP * DEPT$  ejecuta la operación opuesta a la que hemos visto en la figura 1.1.11 y forma la relación EMPLEADO a partir de las relaciones EMP y DEPT.

## EMPLEADO

No. EMPLEADO	NOMBRE-EMPLEADO	No. DEPTO	SALARIO	LUGAR
53702	FERNÁNDEZ J	721	1200	B. ARIES
53703	ROPLEY E S	721	2300	B. ARIES
53791	MUSSOLINI M	007	5000	CORDOBA
53800	NEURINGER H N W	402	1100	ROSARIO
53805	FABRI J	721	1900	B. ARIES
53806	PEREZ P L	402	1000	ROSARIO

DEPT:

No-DEPT	LUGAR
721	B. ARIES
007	CORDOBA
402	ROSARIO

EMP

No. EMPLEADO	NOMBRE-EMPLEADO	No. DEPTO	SALARIO
53702	FERNÁNDEZ J	721	1200
53703	ROPLEY E S	721	2300
53791	MUSSOLINI M	007	5000
53800	NEURINGER H N W	402	1100
53805	FABRI J	721	1900
53806	PEREZ P L	402	1000

FIGURA 1.1.11 Ilustración de la proyección.

## ÁLGEBRA RELACIONAL Y CÁLCULO RELACIONAL.

Las notaciones que acabamos de emplear y otras similares son propias del álgebra relacional. Una operación del álgebra relacional tiene como operandos una o más relaciones y las manipula de modo que forman una nueva relación.

Una manera más automática de atacar el problema consiste en acudir a lo que se llama cálculo relacional. En este caso, el usuario se limita a definir el resultado que desea y deja que el sistema decida qué operaciones se requieren para obtener ese resultado a partir de la base de datos. El cálculo relacional es una notación que sirve para definir una relación que se deriva de las relaciones ya existentes en la base de datos.

Son muchas las formas que puede adoptar el cálculo relacional y presenta éste eventualmente varias ventajas sobre el álgebra relacional:

- 1.- El usuario no declara nada acerca de cómo el sistema se las arreglará para obtener el resultado deseado; por lo tanto el sistema está en libertad para optimizar el método.
- 2.- Permite procedimientos de seguridad más discriminativos, pues esos procedimientos pueden basarse en una definición de las propiedades de los datos requeridos, más bien que en un procedimiento declarado para leerlos.
- 3.- El requerir datos por sus propiedades es más natural para el usuario lego que el requerirlos mediante la especificación de una serie de operaciones. El usuario no tiene por qué presentar sus enunciados de la forma ejemplificada, sino a modo de un diálogo psicológicamente planeado que la máquina ha de traducir a un cálculo relacional formal.

El cálculo relacional tiene la desventaja frente al álgebra, de ser más difícil de implementar ya que representa un más alto nivel de automatización.

En la figura 1.1.12 muestra los símbolos más comúnmente empleados en el cálculo relacional.

SÍMBOLO	EXPLICACIÓN.
$x \in V$	El conjunto de los valores de los ítems de datos en el dominio y de la relación $x$ .
$A(x_1 \in y_1, x_2 \in y_2, \dots)$	Una relación llamada $A$ formada por los dominios que contienen los conjuntos de los valores $x_1 \in y_1, x_2 \in y_2, \dots$
:	<<De modo que>>. La expresión escrita a la izquierda de los dos puntos indica qué es lo que debe obtenerse, mientras que la expresión escrita a la derecha es un calificador
$\exists$	<<Existe.>>
$\forall$	<<Para todo.>>
$\wedge$	<<Y.>>
$\vee$	<<O.>>
$\neg$	<<NO.>>
$=, \neq, <, >$	Igual a, no igual a, menor que, mayor que.
'x'	El valor literal de $x$ .

Figura 1.1.12 Símbolos y operadores utilizados.



### TRES NIVELES DE AUTOMATIZACIÓN

En los sublenguajes de base de datos, se encuentran tres niveles de automatización:

- 1.- **Cálculo:** El más alto nivel de automatización. El usuario manifiesta lo que quiere y el sistema se ocupa de satisfacerlo.
- 2.- **Álgebra:** El usuario declara un conjunto de operaciones de alto nivel que deben ejecutarse sobre ciertas relaciones (u otros agrupamientos de ítems de datos).
- 3.- **Una tupla por vez:** El nivel más bajo. El programador procesa un registro o tupla por vez.

### REPRESENTACIÓN TABULAR

Una de las maneras más naturales de representar datos para el usuario es el que se basa en las tablas bidimensionales. La normalización es un proceso que paso a paso transforma las complejas presentaciones de usuarios y de los almacenamientos de datos, en conjuntos estables de estructuras de datos de menor tamaño que permiten reemplazar relaciones de datos de los usuarios por relaciones de la forma plana bidimensional. Las tablas deben organizarse de forma tal, que no se pierda ninguna relación existente entre datos.

Las tablas en cuestión son matrices rectangulares que pueden ser descritas matemáticamente poseen las siguientes propiedades generales:

- 1.- Cada entrada de la tabla representa un ítem de datos no hay grupos repetitivos.

- 2.- Son homogéneas por columna, es decir, todos los ítems de una columna son de la misma clase.
- 3.- Cada columna tiene nombre propio.
- 4.- Todas las filas son diferentes; no se admiten filas duplicadas.
- 5.- Tanto las filas como las columnas, pueden ser consideradas en cualquier secuencia y en cualquier momento, sin afectar por ello ni el contenido de información ni la semántica de cualquier función que utiliza la tabla.

La base de datos construida por medio de relaciones, es una base de datos relacional. Por lo tanto, una base de datos es relacional, cuando está construida con matrices planas de ítems de datos.

La relación, o tabla, es un conjunto de tuplas. Si se trata de  $n$ -tuplas (una tabla con  $n$  columnas) se dice que la relación es de grado  $n$ . Las relaciones de grado 2 se llaman binarias las de grado 3, ternarias y las de grado  $n$ -enarias.

El conjunto de valores de un mismo tipo, esto es, cada columna de la relación, constituye por definición un dominio. La columna  $j$ -ésima es el dominio  $j$ -ésimo de la relación.

En el lenguaje matemático, dados los conjuntos  $S_1, S_2, \dots, S_n$  (no necesariamente distintos),  $R$  es una relación entre esos  $n$  conjuntos si  $R$  es un conjunto de  $n$ -tuplas cada una de las cuales tiene como primer elemento un elemento de  $S_1$ , como segundo elemento un elemento de  $S_2$  y así sucesivamente.

Para describir estas relaciones y operar con ellas, se recurre a las notaciones y las reglas del álgebra o cálculo relacionales. Codd ha ideado un sublenguaje para manipular las bases de datos de este tipo. Podría traducirse a este sublenguaje cualquier diálogo hombre-computadora convencional.

Los diferentes usuarios de la misma base de datos, percibirán diferentes conjuntos de datos y diferentes relaciones entre ellos, por lo tanto es necesario extraer de las columnas de las tablas los subconjuntos pedidos por algunos usuarios, creando así tablas de menor grado, o por el contrario, a veces es preciso fundir dos o más tablas en una, creando una de mayor grado.

La vista lógica de la base de datos, consistirá en conjuntos de tablas bidimensionales con operaciones para extraer columnas y combinarlas indistintamente.

### NORMALIZACIÓN

La relación derivada de la presentación del usuario o del almacenamiento de datos, generalmente se encontrará no normalizada. El primer paso del proceso incluye la eliminación de grupos repetidos y la identificación de la llave que define al criterio primario. Con este fin, la relación necesita desglosarse en dos o más relaciones. En este punto, las relaciones pueden encontrarse en la forma normal tercera, pero quizás sean necesarios más pasos para transformar las relaciones a la forma normal tercera.

El segundo paso asegura que todos los atributos no-llave, o sin llave, sean completamente dependientes de la llave del criterio primario, las dependencias normales se eliminan y se colocan en otra relación.

El tercer paso elimina cualquier dependencia transitoria. Una dependencia transitoria es aquella en la cual sus atributos no-llave son dependientes de otros atributos no-llave.

## Ejemplo de normalización

La figura 1.1.13 es una presentación para el usuario de la Compañía "Manufacturera de Equipo A1S. Well Hydraulic". El reporte muestra: 1) el NÚMERO-VENDEDOR; 2) el NOMBRE-VENDEDOR y 3) el ÁREA-VENTAS.; La parte central del reporte muestra: 4) el NÚMERO-CLIENTE y 5) el NOMBRE-VENDEDOR. Más adelante : 6) el NÚMERO-ALMACÉN el cual le dará servicio al cliente como se indica, seguirá por 7) la UBICACIÓN-ALMACÉN, la cual es la ciudad en la cual se localiza la compañía. La información final que contendrá la presentación para el usuario es. 8) VALOR-VENTA. Los renglones (uno para cada cliente) en la presentación del usuario muestran que los artículos del 4 al 8, forman un grupo repetido.

Compañía Manufacturera de Equipos  
S. Well Hydraulic  
Spring Valley Minnesota.

Vendedor #: 3462  
Nombre: Waters.  
Área de Ventas: Occidente.

NUMERO DE CLIENTE	NOMBRE CLIENTE	NUMERO ALMACÉN	UBICACIÓN ALMACÉN	VENTAS
18765	Delta Services	4	Fargo	13,540
18830	M. Levy and Sons	3	Bismarck	10,6000
etc.				

Figura 1.1.13 Un reporte de usuario de la Compañía Hidráulicos Al. S. Well

Si se utiliza un enfoque de diccionario de flujo de datos, la misma información aparecerá, tanto en la presentación para el usuario, como en la estructura de datos. La figura 1.1.14 muestra cómo aparecería la estructura de datos, en el diccionario de datos, durante la etapa del análisis. Los grupos repetidos también se anotan en la estructura de datos por medio de un asterisco (\*) y se marca una sangría en los siguientes renglones.

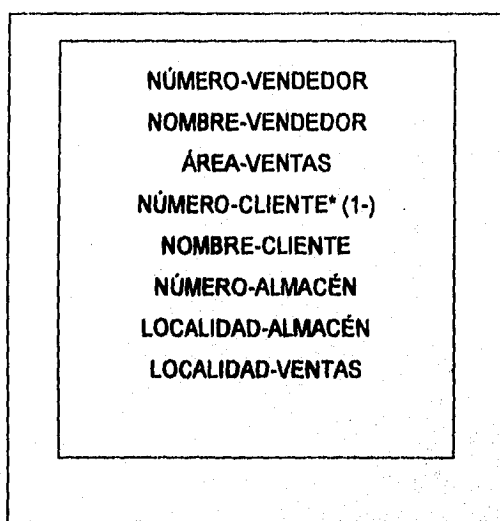


Figura 1.1.14 Estructura de datos para el desarrollo de un base de datos a partir de un diccionario de datos.

La figura 1.1.15 muestra las asociaciones existentes entre los datos elementales. Este tipo de ilustración se denomina diagrama de burbuja o diagrama de modelo de datos. Cada entidad se encierra en una elipse y se utilizan flechas para indicar las relaciones. Aunque es posible dibujar estas relaciones en un diagrama E-R, en ocasiones es más fácil utilizar un sencillo diagrama de burbuja para modelar los datos.

En este ejemplo existe sólo un NÚMERO-VENDEDOR asignado a cada NOMBRE-VENDEDOR y tal persona cubrirá sólo un ÁREA-VENTAS, pero cada ÁREA-VENTAS puede asignarse a muchos vendedores; es por ello, que se utiliza la notación de doble flecha de ÁREA-VENTAS a NÚMERO-VENDEDOR. Y para cada NÚMERO-VENDEDOR puede haber muchos NÚMERO-CLIENTE(s).

Y es por ello que habrá una correspondencia uno a uno entre NÚMERO-CLIENTE y NOMBRE-CLIENTE; lo mismo es cierto para NÚMERO-ALMACÉN y UBICACIÓN-ALMACÉN. NÚMERO-CLIENTE tendrá sólo un NÚMERO-ALMACÉN y ALMACÉN-UBICACIÓN, pero cada NÚMERO-ALMACÉN o ALMACÉN-UBICACIÓN puede dar servicio a numerosos NÚMERO-CLIENTE. Finalmente, para determinar el VALOR-VENTAS para una petición del vendedor de una compañía particular, es necesario saber tanto el NÚMERO-VENDEDOR como el NÚMERO-CLIENTE.

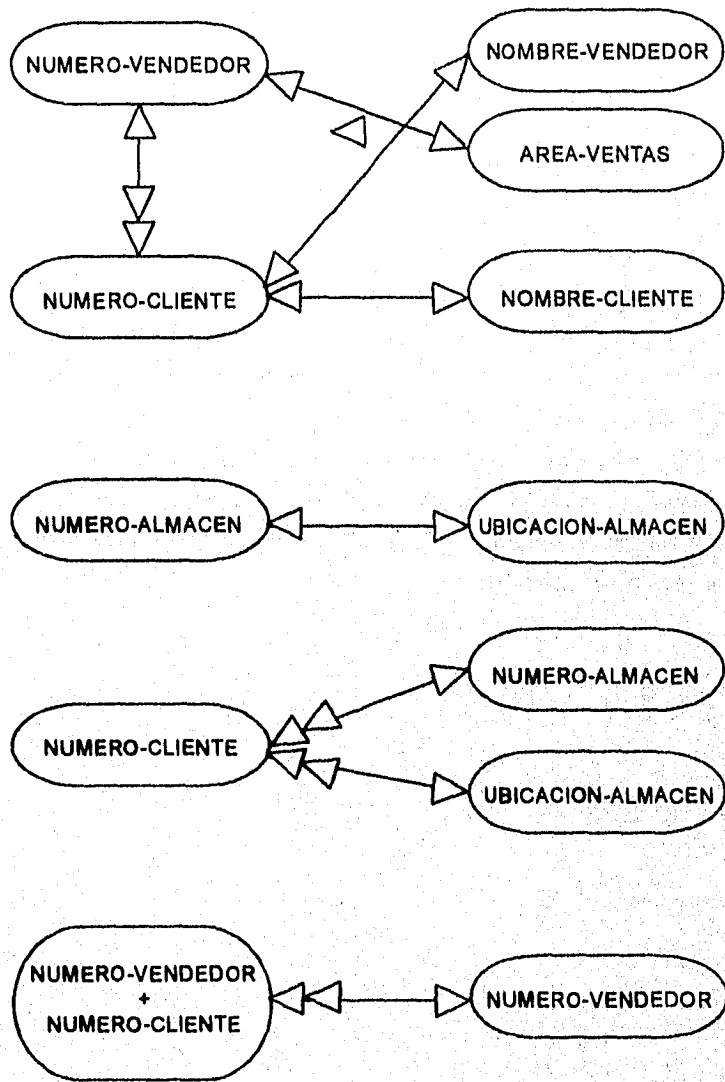


FIGURA 1.1.15 Elaboración de diagramas de datos.

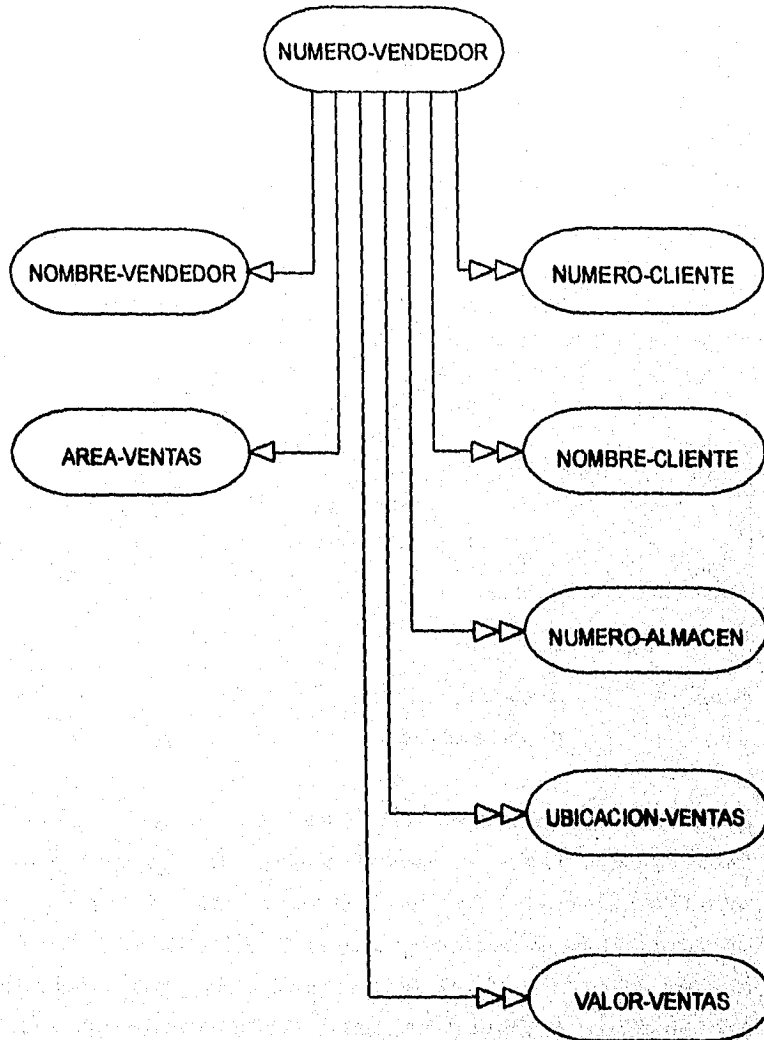
El principal objetivo del proceso de normalización, es simplificar toda la complejidad existente de los datos, en las aplicaciones de los usuarios. Por ejemplo, si el analista considera utilizar la presentación antes expuesta del usuario e intentara desarrollar una tabla de relación a partir de ella, la tabla se asemejaría a la de la figura 1.1.16; ya que ésta es una relación que se basa en nuestra presentación inicial para el usuario, nos referiremos a ella como **REPORTE-VENTAS**.

No. VENDEDOR	NOMBRE VENDEDOR	ÁREA VENTAS	No. CLIENTE	NOMBRE CLIENTE	No. ALM	UBICACIÓN ALMACÉN	VALOR VENTA
3482	WATERS	WEST	18766	DELTA S	4	FARGO	13540
			18830	A LEVY	3	BISMARCK	10800
			18242	RANIER	3	BISMARCK	9700
3683	DRYNE	EAST	18841	R.W.	2	SUPERIOR	11580
			18899	SEWARD	2	SUPERIOR	2580
			19565	STODOLA	1	PLYMOUTH	8800
ETC.							

**FIGURA 1.1.16** Si los datos se listaran en una tabla no normalizada habría grupos repetidos.

El **REPORTE-VENTAS** es una relación no normalizada, ya que cuenta aún con grupos repetidos. También es importante observar que un atributo sencillo, tal como **NÚMERO-VENDEDOR** no puede servir como llave. La razón será obvia cuando examinemos las relaciones existentes entre el **NÚMERO-VENDEDOR** y otros atributos de la figura 1.1.17. Aunque existe una correspondencia uno a uno entre **NÚMERO-VENDEDOR** y dos atributos (**NOMBRE-VENDEDOR** y **ÁREA-VENTAS**), existe una relación de uno a muchos entre **NÚMERO-VENDEDOR** y los otros cinco atributos (**NÚMERO-CLIENTE**, **NOMBRE-CLIENTE**, **NÚMERO-ALMACÉN**, **UBICACIÓN-ALMACÉN** y **VALOR-VENTAS**).





**Figura 1.1.17** Un diagrama de datos modelo muestra que una relación no normalizada, el NÚMERO-VENDEDOR tiene una asociación 1:M con ciertos atributos.

El REPORTE-VENTAS puede expresarse mediante la siguiente notación taquigráfica:

**REPORTE-VENTAS:** (NÚMERO-VENDEDOR, NOMBRE-VENDEDOR, ÁREA-VENTAS, (NÚMERO-CLIENTE, NOMBRE-CLIENTE, NÚMERO-ALMACÉN, UBICACIÓN-ALMACÉN, VALOR-VENTAS).

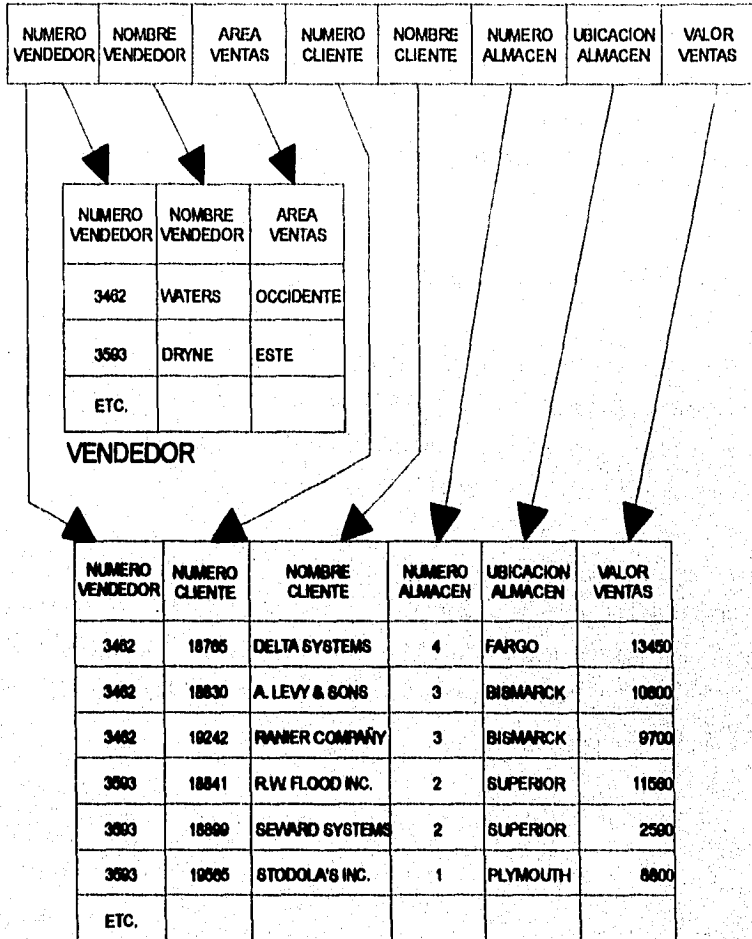
El conjunto comprendido dentro de los paréntesis internos representa al grupo repetido.

#### Primera forma normal (FN1)

El primer paso para normalizar una relación es eliminar los grupos que están repetidos. En nuestro ejemplo, la relación no normalizada REPORTE-VENTAS se descompondrá en dos relaciones separadas. Esas nuevas relaciones se denominarán VENDEDOR y CLIENTE-VENDEDOR.

La figura 1.1.18 muestra la relación original no normalizada de REPORTE-VENTAS, la cual se normaliza al separar la relación en dos nuevas relaciones. Observe que la relación VENDEDOR contiene la llave primaria NÚMERO-VENDEDOR y todos los atributos que no se repiten (NOMBRE-VENDEDOR y ÁREA-VENTAS).

REPORTE- VENTAS



VENDEDOR-CLIENTE

Figura 1.1.18 La rel. original no normalizada VENTAS-REPORTES se separa en 2 relaciones VENDEDOR (FN3) y VENDEDOR-CLIENTE (FN1).

La segunda relación, VENDEDOR-CLIENTE, contiene el criterio o llave principal de la relación VENDEDOR (el primer criterio de VENDEDOR es NÚMERO-VENDEDOR), así como todos los atributos que formaron parte del grupo repetido (NÚMERO-CLIENTE, NOMBRE-CLIENTE, NÚMERO-ALMACÉN, UBICACIÓN-ALMACÉN y VALOR-VENTAS). Sin embargo, al saber que el NÚMERO-VENDEDOR no es suficiente para conocer el NOMBRE-CLIENTE, VALOR-VENTAS, UBICACIÓN-ALMACÉN, etc. En ésta relación, debe utilizarse una llave concatenada ( NÚMERO-VENDEDOR y NOMBRE-CLIENTE para acceder el resto de la información. Es posible escribir tales relaciones en notación taquigráfica, de la siguiente manera:

**VENDEDOR:** (NÚMERO-VENDEDOR, NOMBRE-VENDEDOR, ÁREA-VENTAS)  
y VENDEDOR.

**CLIENTE:** (NÚMERO-VENDEDOR, NÚMERO-CLIENTE, NOMBRE-CLIENTE,  
NÚMERO-ALMACÉN, UBICACIÓN-ALMACÉN, VALOR-VENTAS)

La relación VENDEDOR-CLIENTE es una primera relación de normalización, pero no se encuentra en una forma ideal. Los problemas emergen a partir del hecho de que ciertos atributos no son funcionalmente dependientes del criterio o llave primario, NÚMERO-VENDEDOR, NÚMERO-CLIENTE. En otras palabras, ciertos atributos no claves son dependientes sólo de NÚMERO-CLIENTE y no del criterio concatenado.

El diagrama de datos modelo de la figura 1.1.19., muestra que VALOR-VENTAS es dependiente, tanto de NÚMERO-VENDEDOR como de NÚMERO-CLIENTE, pero los otros tres atributos son dependientes solamente de NÚMERO-CLIENTE.

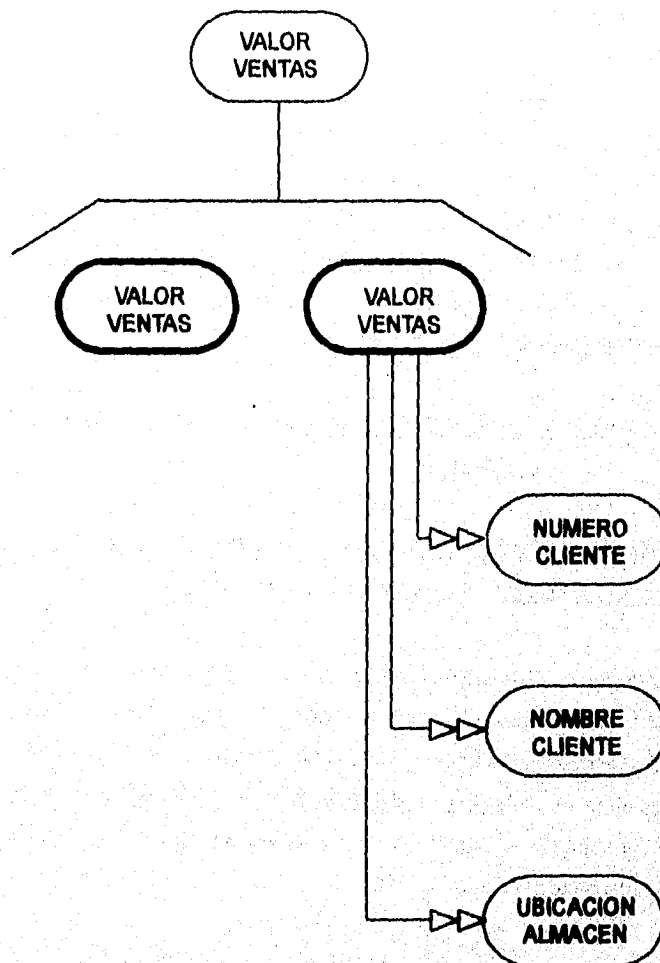


Figura 1.1.19 Un diagrama de datos modelo muestra que tres atributos son dependientes del NÚMERO-CLIENTE, de tal forma que la relación aún no se encuentra normalizada. Tanto NÚMERO-VENDEDOR como NÚMERO-CLIENTE se requieren para localizar VALOR-VENTAS.

**Segunda forma normal (FN2)**

En la forma normal secundaria, todos los atributos serán funcionalmente dependientes del criterio o llave primaria. Además, el siguiente paso sería eliminar todas las dependencias parciales y colocarlas en otra relación. La figura 1.1.20 se muestra cómo la relación VENDEDOR-CLIENTE se separa en dos nuevas relaciones, VENTAS y CLIENTE-ALMACÉN.

Estas relaciones pueden expresarse de la siguiente manera:

**VENTAS:** (No-VENDEDOR, No-CLIENTE, VALOR-VENTAS).

**CLIENTE-ALMACÉN:** (No-CLIENTE, NOMBRE-CLIENTE, UBICACIÓN-ALMACÉN, No-ALMACÉN).

La relación CLIENTE-ALMACÉN se encuentra en una segunda forma normal. Esto puede simplificarse aún más, al disponer de tres dependencias adicionales dentro de la relación. Algunos de los atributos no-primarios son dependientes no sólo del criterio o llave primaria, sino también de atributos no primarios. A esto se le denomina como una dependencia transitiva.

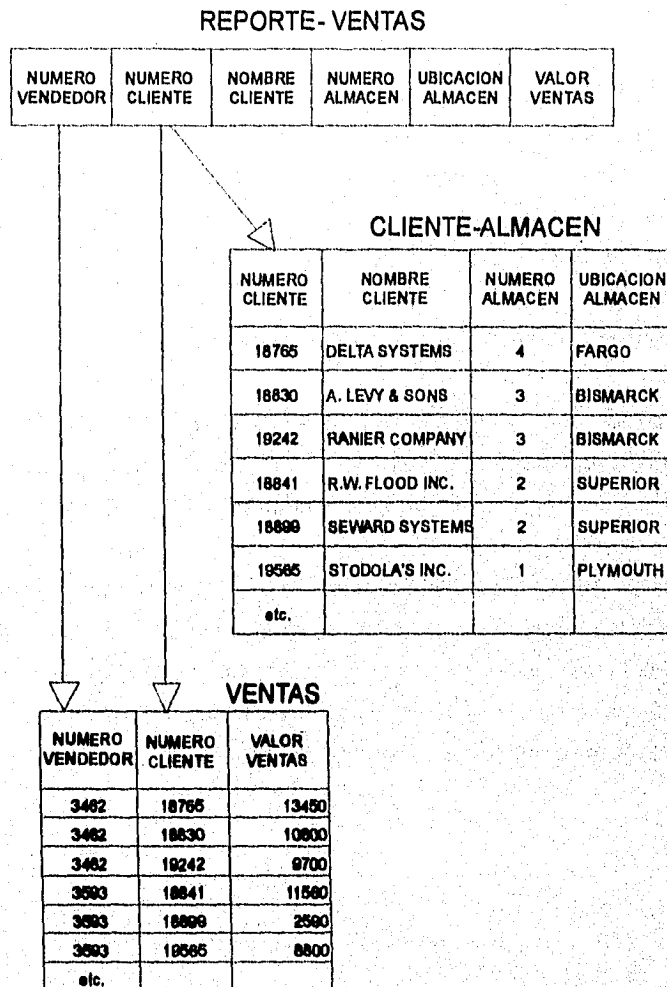


Figura 1.1.20 La relación VENDEDOR-CLIENTE se encuentra en una relación denominada CLIENTE-ALMACÉN (FN2) y una relación denominada VENTAS(FN1).

La figura 1.1.21 muestra las posibles dependencias dentro de la relación **CLIENTE-ALMACÉN**. Con el fin de que la relación se encuentre en una forma normal secundaria, todos los atributos deben ser dependientes del criterio o llave primaria **NÚMERO-CLIENTE**, como se muestra en el diagrama.

Sin embargo, también **UBICACIÓN-ALMACÉN** es obviamente dependiente de **NÚMERO-ALMACÉN** para simplificar ésta relación se requiere de otro paso adicional.

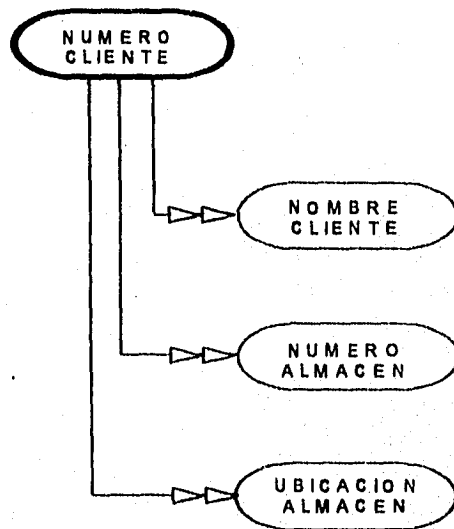


Figura 1.1.21 Diagrama de datos modelo que muestra que existe una dependencia transitoria entre **NÚMERO-ALMACÉN** y **UBICACIÓN-ALMACÉN**.



**Tercera forma normal (FN3)**

Una relación normalizada es terciaria si todos los atributos no fundamentales son completamente dependientes desde un punto de vista funcional del criterio o llave primaria y no hay dependencias transitivas (no claves). De manera similar a los pasos anteriores es posible descomponer la relación CLIENTE-ALMACÉN en dos relaciones, tal y como se muestra en la figura 1.1.22. Las dos nuevas relaciones se denominan CLIENTE y ALMACÉN, y pueden escribirse de la siguiente manera:

CLIENTE	(NÚMERO-CLIENTE, NOMBRE-CLIENTE, <u>NÚMERO-ALMACÉN</u> )
ALMACÉN	( <u>NÚMERO-ALMACÉN</u> , UBICACIÓN-ALMACÉN)

El criterio o llave primaria para la relación CLIENTE es NÚMERO-CLIENTE y el criterio o llave primaria para la relación ALMACÉN es NÚMERO-ALMACÉN. Además de estos criterios primarios, podemos identificar a NÚMERO-ALMACÉN como un criterio externo a la relación CLIENTE.

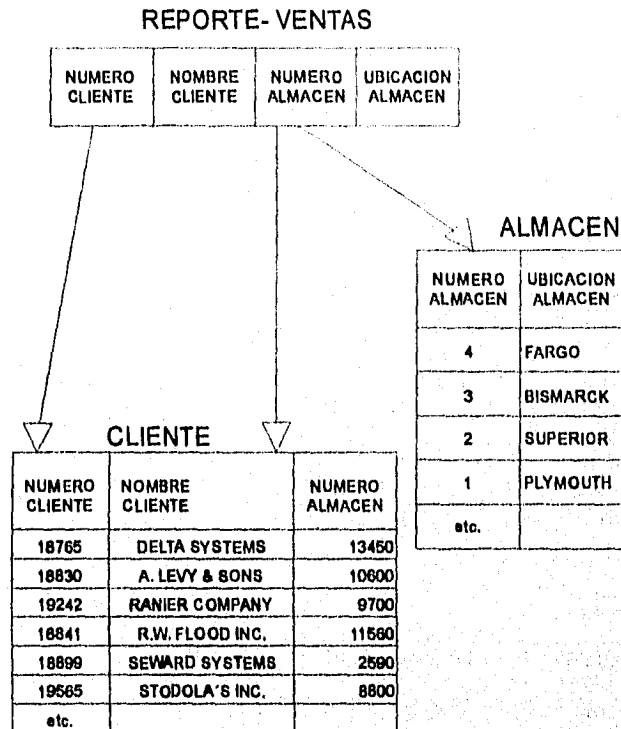


Figura 1.1.22 La relación **VENDEDOR-CLIENTE** se encuentra en una relación denominada **CLIENTE-ALMACÉN (FN2)** y una relación denominada **VENTAS(FN1)**.

Un criterio externo es cualquier atributo que no es llave en tal relación, pero es un criterio o llave primario en otra relación.

**VENDEDOR**

NÚMERO VENDEDOR	NOMBRE VENDEDOR	ÁREA VENTAS
3462	WATERS	OCCIDENTE
3593	DRYNE	ESTE
etc.		

**VENTAS**

NÚMERO VENDEDOR	NÚMERO CLIENTE	VALOR VENTAS
3462	18765	13540
3462	18830	106000
3462	19242	9700
3593	18841	11560
3593	18899	2590
3593	19565	8800
ETC.		

FIGURA 1.1.23. La base de datos completa consiste de cuatro relaciones FN1 denominadas VENDEDOR, VENTAS, CLIENTES Y ALMACÉN.

**CLIENTE**

NÚMERO CLIENTE	NOMBRE CLIENTE	NÚMERO ALMACÉN
18765	DELTA SYSTEMS	4
18830	A. LEVY AND SONS	3
19242	RANIER COMPANY	3
18841	R.W. FLOOD INC.	2
18899	SEWARD SYSTEMS	2
19565	STODOLA'S INC.	1
etc.		

**ALMACÉN**

NÚMERO ALMACÉN	UBICACIÓN ALMACÉN
4	FARGO
3	BISMARCK
2	SUPERIOR
1	PLYMOUTH
etc.	

Hemos designado con anterioridad a NÚMERO-ALMACÉN como un criterio externo por medio del subrayado en línea separada \_\_\_\_\_. Finalmente, la relación no normalizada REPORTE-VENTAS se transforma en cuatro relaciones normales terciarias (FN3). Al revisar las relaciones que se muestran en la figura 1.1.23, uno puede observar que la relación sencilla REPORTE-VENTAS se transforma en las siguientes cuatro relaciones:

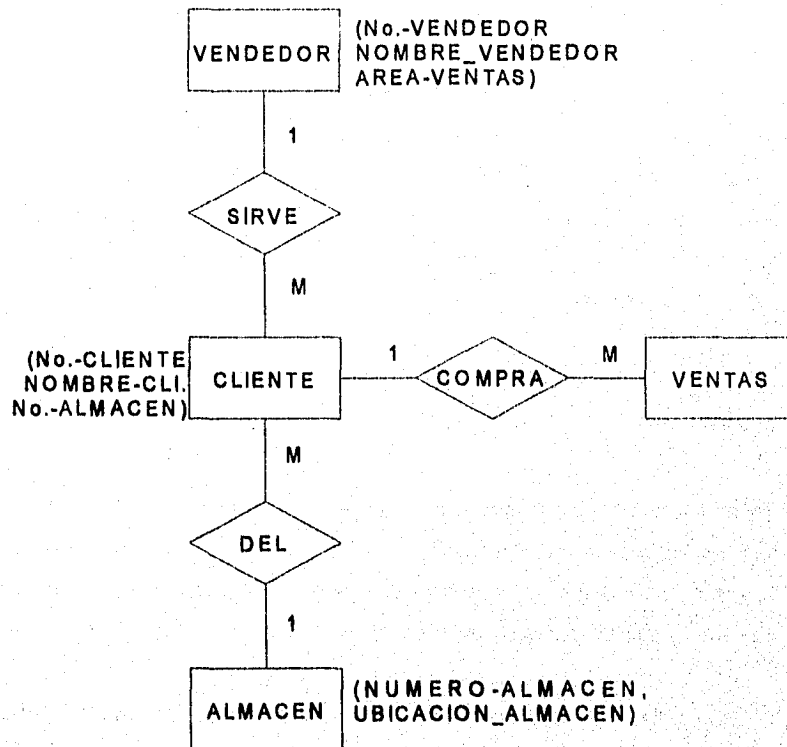


Figura 1.1.24 Diagrama de relación-entidad para la base de datos de la Compañía Hidráulica Al. S. Well.

## Capítulo I

---

VENDEDOR: (NÚMERO-VENDEDOR, NOMBRE-VENDEDOR, ÁREA-VENTAS)

VENTAS: (NÚMERO-VENDEDOR, NÚMERO-CLIENTE, VALOR-VENTAS)

CLIENTE: (NÚMERO-CLIENTE, NOMBRE-CLIENTE, NÚMERO-ALMACÉN)

ALMACÉN: (NÚMERO-ALMACÉN, UBICACIÓN-ALMACÉN)

La forma de normalización terciaria es adecuada para la mayoría de los problemas del diseño de base de datos. La simplificación obtenida al transformar una relación no normalizada en relaciones normales terciarias redundante en un amplio beneficio para la inserción, supresión y actualización de la información de la base de datos. En la figura 1.1.24 se muestra un diagrama para una base de datos.

### CLAVES

Toda tupla debe estar asociada con una clave que permita su identificación. A veces la tupla se identifica mediante un único atributo pero en otras ocasiones hay que recurrir a más de un atributo para lograr la identificación inequívoca de una tupla. Así, ningún atributo único es suficiente para identificar una tupla.

En cada tupla puede existir más de un conjunto de atributos capaz de satisfacer estos requisitos. Esos conjuntos se denominan claves candidatas y de ellas una debe ser designada como la clave primaria usada para efectivamente identificar al registro. Cuando hay opción, la clave primaria se elegirá:

- 1.- De modo que ninguno de los atributos que la forman tengan un valor indefinido.
- 2.- de modo que el número de atributos sea el mínimo.

#### **NORMALIZACIÓN DE LAS LISTAS DE COMPONENTES O MATERIALES.**

A menudo la forma normalizada tiene una apariencia mucho más simple que el esquema convencional. Los productos fabricados por una planta, consisten en este caso hipotético, en piezas tales como bulones o varillas roscadas y subconjuntos de unidades de mando y pestillos para puertas. Cada subconjunto estará eventualmente compuesto de piezas sueltas y otros subconjuntos. Los subconjuntos de nivel 4 contienen quizás subconjuntos de nivel 3 y éstos, subconjuntos de nivel 2 y así sucesivamente. La lista de materiales indica de qué están compuestos los productos y los subconjuntos.

La forma normalizada de este archivo contiene solo dos relaciones:

- 1.- La relación ÍTEM que contiene No.-ÍTEM, NOMBRE y un atributo CATEGORÍA que expresa si el rubro es un producto, una pieza suelta o un subconjunto de nivel 1, 2, 3 o 4
- 2.- Una relación DESPIEZO que da la cantidad de cada rubro que es componente de otro.

La lista de materiales tiene la característica de estar expuesta a frecuentes cambios por razones de ingeniería. Tendrá que almacenarse el despiezo del producto tal como era antes del cambio y tal como queda después. Si el archivo está representado por medio de una estructura plex, la inclusión de estos cambios es casi siempre difícil, en cambio en el archivo normalizado es cosa simple. Todo nuevo renglón introducido por el cambio se agrega simplemente a las relaciones ÍTEM y DESPIEZO, agregando además una relación CAMBIO-ING (No-CAMBIO, No-ÍTEM).

#### RESUMEN DE LAS VENTAJAS

A continuación se resumen las ventajas que reporta la representación de datos de la forma normalizada y particularmente en la tercera forma normalizada.

- **Facilidad de uso.-**

La manera más fácil de representar la mayor parte de los datos para uso del usuario es la que se basa en el empleo de tablas bidimensionales.

- **Flexibilidad.-**

Las operaciones del tipo PROYECCIÓN y UNIÓN permiten partir y pegar relaciones de modo que se pueda dar a los usuarios los archivos que necesitan para sus aplicaciones y de la forma en que los requieren.

- **Precisión.-**

Los enlaces dirigidos, tan comunes en las representaciones lógicas actuales, son a menudo engañosos. Las relaciones (tablas) tienen un significado preciso y pueden ser manipuladas con la matemática del álgebra o el cálculo relacional.

- **Seguridad.-**

Es más fácil implementar los controles de seguridad. Las autorizaciones de seguridad se refieren a relaciones. Los atributos más sensibles, por ejemplo SALARIO en un archivo de empleados, pueden ser trasladados a una relación aparte, con sus propios controles de autorización. De satisfacerse estos controles, SALARIO se reunirá por UNIÓN con otros atributos del empleado.

- **Relacionabilidad.-**

Se tiene la máxima flexibilidad para relacionar atributos de diferentes conjuntos de tuplas o diferentes archivos.

- **Facilidad de implementación.-**

El almacenamiento físico de los archivos planos, es eventualmente menos complejo que el de los árboles o las estructuras plex. A medida que se popularizan técnicas físicas más elaboradas, tales como la de las jerarquías de almacenamiento o las memorias asociativas, la simplificación que la normalización aporta va redituando en mayores dividendos. Los dispositivos capaces de llevar a la práctica la exploración rápida de los archivos son más factibles en el caso de los archivos que carecen de complejos sistemas de apuntadores.

- **Independencia de datos.-**

Las bases de datos están obligadas a crecer, a causa de la adición de nuevos atributos y nuevas relaciones. Se les usará, también, de distintas maneras. Se agregarán y eliminarán tuplas y tipos de ítems de datos. Si la base de datos es de la forma normalizada con independencia de datos en el software, será posible reestructurar los datos y la base de datos crecerá en la mayoría de los casos sin tener que volver a escribir los programas de aplicación. Esto es de suma importancia a causa de los excesivos y crecientes costos de mantenimiento de los programas de aplicación de una empresa y de sus datos, los que tienden a



alcanzar niveles prohibitivos. Es probable que la adecuada independencia de datos sea más fácil de lograr en las bases relacionales.

- **Lenguaje para la manipulación de datos.-**

El lenguaje para la manipulación de datos podrá basarse en el álgebra o el cálculo relacional. Si los datos se organizaran de modo que permitieran una variedad de estructuras no planas, el sublenguaje de datos resultaría innecesariamente complicado para el usuario o se vería limitado en cuanto a su capacidad.

- **Claridad.-**

Las representaciones lógicas basadas en el uso de flechas, pueden resultar claras mientras sea pequeño el número de registros. El objetivo final del desarrollo de las bases de datos es, sin embargo, el producir bases que abarquen la mayor parte de las actividades interrelacionadas de un organismo. Con el crecimiento de las bases de datos y su tendencia a abarcar más y más actividades, es imperativo que terminemos con las representaciones lógicas basadas en el uso de punteros de punteros de punteros. Las bases relacionales ofrecen, al parecer, la mejor solución.

## **1.2 BASES DE LA INGENIERÍA DEL SOFTWARE.**

La ingeniería del software consiste en el establecimiento y uso de principios y metodologías de ingeniería, orientados a obtener software que sea fiable y funcione eficientemente sobre máquinas reales.

El propósito fundamental de la ingeniería del software es modelar las necesidades del usuario y sus políticas de funcionamiento a través de herramientas gráficas y textuales que faciliten la comprensión del sistema a los usuarios y permitan enfatizar los aspectos críticos del mismo.

Se construyen modelos de sistemas por las siguientes razones:

- 1.- Para enfocar características importantes de sistemas y minimizar los aspectos menos importantes.
- 2.- Discutir cambios y correcciones a los requerimientos de los usuarios a bajos costos y riesgos mínimos.
- 3.- Verificar que se entiende el ambiente del usuario y que se documentó de una manera clara para los diseñadores y programadores.

El primer modelo que se presenta en la etapa de análisis es el "modelo esencial", el cual muestra lo que debe hacer el sistema para satisfacer adecuadamente las necesidades del usuario, indicando lo mínimo posible acerca de cómo se implantará. El modelo supone que se tiene disponible una tecnología capaz y que se puede obtener fácilmente y sin costos elevados. Este modelo está integrado a su vez por el modelo ambiental y el modelo de comportamiento.

### 1.2.1 MODELO AMBIENTAL

Cualquier sistema tiene relación con el medio ambiente en el que se sitúa. Esta interrelación dificulta la visión del sistema que se está desarrollando, por lo cual es importante definir con claridad ¿qué forma parte del sistema? y ¿qué forma parte del medio ambiente?. Podemos decir que "el modelo ambiental define la frontera entre el sistema y el resto del mundo". La definición de los límites del sistema comienza por la elaboración del objetivo a través de la redacción de las actividades que realizará el sistema.

Dentro del análisis de un sistema es necesario determinar ¿qué es parte del sistema? y ¿que no?. Cualquier sistema que se desarrolle, será parte de un sistema aún mayor.

Además de determinar ¿qué está en el interior? y ¿qué en el exterior? (lo que se logra definiendo la frontera entre el sistema y el medio ambiente), también se necesita saber qué información entra al sistema desde el exterior y cuál es la que produce como salida.

Las entradas y salidas siempre se presentan en forma racional, tienen un propósito específico como evento derivado de algún acontecimiento o estímulo en el medio ambiente. Por ello, otro aspecto importante del modelo ambiental es identificar los eventos que ocurren en el ambiente al cual debe responder el sistema. Sólo se deben considerar eventos que ocurren en el medio ambiente exterior y que requieren respuesta del sistema.

La frontera entre un sistema y su medio ambiente es arbitraria; sin embargo, puede considerarse alguna política administrativa, negociaciones políticas u otras estrategias para definir una frontera.

**A) OBJETIVO**

El primer componente del modelo ambiental es una declaración textual breve y concisa del propósito del sistema (objetivo), dirigida al nivel administrativo superior, la administración de los usuarios, y otros que no están directamente involucrados con el desarrollo del sistema.

El establecimiento del objetivo puede constar de una, dos o varias frases, sin embargo, jamás debe llegar a más de un párrafo, ya que la intención no es proporcionar una descripción completa y detallada del sistema, tal esfuerzo iría en contra del objetivo.

**B) LISTA DE EVENTOS**

La descripción de los estímulos que recibe el sistema del medio ambiente y que provocan una reacción en él, se define como lista de eventos.

Cada uno de los eventos es analizado y clasificado como un evento de tipo flujo, temporal o de control.

Los eventos orientados a flujos se identifican como aquellos que proporcionan datos al sistema y que al ser captados hacen que se realice algún proceso, estos eventos son marcados con (F).

Los eventos de tipo temporal, son aquellos que periódicamente proporcionan información a las entidades con las cuales se relaciona el sistema, estos eventos se marcan con (T). Debe notarse que los eventos temporales no se inician con flujos de datos de entrada; puede imaginarse que el sistema tiene un reloj interno con el cual puede determinar el paso del tiempo, sin embargo, debe tenerse en mente también que un evento temporal podría requerir que el sistema solicite entradas de uno o más terminales, por ello, podrían asociarse uno o

más flujos de datos con un evento temporal, aunque los flujos de datos en sí, no representan el evento mismo.

Los eventos de control deben considerarse un caso especial del evento temporal: un estímulo externo que ocurre en algún momento impredecible. A diferencia de un evento temporal normal, el evento de control no se asocia con el paso regular del tiempo, por lo que el sistema no puede anticiparlo utilizando un reloj interno. Y a diferencia de un evento de flujo normal, el de control no indica su presencia con el arribo de datos.

### **C) DIAGRAMAS DE CONTEXTO**

El diagrama de contexto es la siguiente etapa del modelo ambiental es un caso especial del diagrama de flujo de datos, en donde una sola burbuja representa todo el sistema.

Para poder cumplir con el objetivo planteado y realizar las funciones requeridas, es necesario que el sistema interactúe con otros sistemas u organizaciones, los cuales son llamados terminadores o entidades y están representadas en el diagrama de contexto a través de rectángulos, dentro de los cuales se coloca el nombre de la entidad a la que representan. Los datos que proporciona cada una de las entidades se muestran mediante flechas que representan a su vez los principales flujos de entrada y salida del sistema.

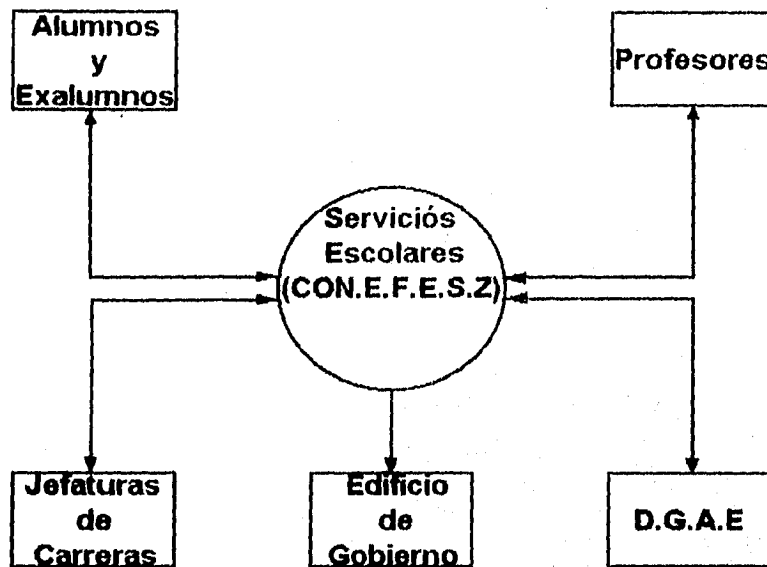


Figura 1.1.3.1 Diagrama de contexto para el sistema de control escolar para la Facultad de estudios Superiores Zaragoza.

La parte más difícil de un diagrama de contexto es el proceso; que consiste de un círculo en un Diagrama de flujo de datos. El nombre de tal proceso suele ser el nombre completo del sistema o un acrónimo convenido. En un caso extremo, el sistema puede representar una organización completa; el nombre del proceso sería el de la organización misma. Los nodos terminales, se representan con rectángulos en el diagrama de contexto. Se comunican con el sistema a través de flujos de datos o de control, los cuales no deben de comunicarse directamente entre sí, esto es incorrecto dentro de un diagrama de contexto.

## Capítulo I

---

Además se deben considerar las siguientes aclaraciones sobre los terminales:

- Algunos tienen un amplio número de entradas y salidas. Para evitar un diagrama innecesariamente saturado, conviene dibujar el terminal más de una vez y ser marcado por un asterisco o una diagonal.
- Cuando el terminal es una persona individual, generalmente es preferible indicar el rol que desempeña, más que su entidad.
- Es importante distinguir entre almacenes de datos y manejadores de bases de datos. Se entiende por almacenes de datos a bases de datos y archivos; mientras un manejador es un mecanismo, dispositivo o medio físico usado para transportar datos hacia dentro o fuera del sistema.

Los flujos que aparecen en el diagrama de contexto se incluyen para:

- Detectar un acontecimiento en el medio ambiente en que deba responder el sistema, o si se ocupan (como datos) para producir una respuesta.
- Ilustrar datos que son transportados entre los terminales y el sistema.

El diagrama de contexto de un modelo esencial debe evitar (hasta donde sea posible) mostrar los manejadores cercanos a la implementación que introducen y sacan datos del sistema. Además, tampoco deberá mostrar los mensajes y medios específicos de coordinación que el sistema y los terminales pasan entre sí para indicar que están listos para las entradas o salidas.

En lugar de lo anterior, es conveniente dibujar el diagrama de contexto bajo el supuesto de que las entradas son causadas e iniciadas por los terminales y que las salidas son causadas e iniciadas por el sistema.

Aún así, habrá ocasiones en que el terminal no inicie las entradas, pues aún con la tecnología capaz, éste no sabe que el sistema requiere entradas, similarmente, hay ocasiones en que el sistema no inicia la generación de salidas, debido a que no sabe que el terminal las necesita o desea. En ambos casos, el mensaje es una parte esencial del sistema. A veces resulta conveniente mostrar el mensaje y el correspondiente flujo de entrada o salida con un flujo de diálogo (una flecha de dos cabezas).

#### **D) FLUJO DE DATOS**

A medida que fluye la información en un sistema, éste se transforma. El sistema acepta entradas en una gran variedad de formas, las procesa y proporciona flujos de datos de salida.

A medida que la información se mueve a través del software, es modificada por una serie de transformaciones. Los flujos de datos se representan por medio de un diagrama de flujo de datos, el cual muestra como se transforman los flujos de información en flujos de datos.

Entonces podemos tener las siguientes clases de flujos de datos:

##### **1.- Flujo de datos de entrada.**

Son aquellos datos que el sistema requiere para un determinado proceso y que fluyen de una entidad externa.

##### **2.- Flujo de datos de salida.**

Son los datos que el sistema produce como resultado del procesamiento de datos de entrada y control, y que fluyen hacia una entidad externa al sistema.



**3.- Flujo de control.**

El flujo de control puede considerarse como un flujo de datos binario: está encendido o apagado, y puede cambiar de un estado al otro en cualquier momento, señalando así al sistema que se necesita tomar alguna acción inmediata.

**1.2.2 MODELO DE COMPORTAMIENTO**

**A) DICCIONARIO DE DATOS**

**1 ¿Que es un diccionario de datos?**

Un diccionario de datos es un depósito central de información acerca de las entidades, los campos de datos que representan a las entidades, las relaciones entre éstas, sus orígenes, significado, uso y formatos de representación.

En ninguna empresa se instala una base de datos de la noche a la mañana. La base de datos se expande a medida que se desarrollan e integran las aplicaciones y se introducen nuevos campos de datos, además puede ser necesario modificar los elementos utilizados durante el diseño de la base de datos. El recurso que proporciona información uniforme y centralizada acerca de todos los datos se llama diccionario de datos (DD).

Los beneficios de usar un diccionario de datos están relacionados con la recopilación, especificación y manejo efectivo de los recursos totales de datos de una empresa. Un diccionario de datos debe ayudar al usuario de una base de datos a:

- Comunicarse con otros usuarios.
- Controlar los campos de datos de manera simple y efectiva, esto es, al introducir nuevos campos de datos en los sistemas o al cambiar las descripciones de estos.
- Reducir la redundancia e inconsistencia de los datos.
- Determinar el impacto de los cambios en los campos de datos sobre la base de datos total.
- Centralizar el control de los campos de datos, como una ayuda en el diseño y en la expansión del diseño de la base de datos.

Además, un diccionario de datos ideal contiene información sobre otras entidades, almacena información sobre grupos de campos de datos, sobre la base de datos y sobre las referencias cruzadas entre los grupos de campos de datos y las bases, también indica qué programas se usan con las bases de datos y conserva información concerniente a los códigos de autorización y de seguridad.

El primer paso en el diseño de una base de datos es recabar información sobre la empresa, es decir, acerca del uso de las relaciones y significado de los datos. Al avanzar el proceso de diseño es necesario almacenar información sobre los modelos conceptuales, lógico interno y externo, en un lugar central. La herramienta que da la posibilidad de controlar y manejar la información sobre los datos en la fase de diseño, implantación, operación y expansión de una base de datos se llama diccionario de datos.

## Capítulo I

---

En la etapa inicial de recolección de información sobre los datos se deben:

- Establecer criterios para nombrar los campos de datos.
- Resolver los conflictos entre distintos departamentos acerca de los significados, las fuentes y los criterios.
- Llegar a un acuerdo sobre las descripciones de los campos de datos y reconocer sus sinónimos.

Cada departamento o parte de la organización se considera que su misión es la más importante, y que su perspectiva de la organización es la correcta. Esto trae como consecuencia conflictos en la recolección de información acerca de los datos. En ésta situación, el administrador de la base de datos tiene que desempeñar el papel del que encuentra soluciones. Para seguir y registrar efectivamente la información sobre los datos, es deseable que todo el que tenga participación en la base de datos use un diccionario de datos automatizado.

El diccionario de datos almacena información relativa al origen de los datos, su descripción, relación con otros datos, uso, responsabilidad y formato. Es la misma base de datos la que almacena "datos sobre datos". El diccionario de datos es una guía y contiene el "mapa de la ruta" hacia la base de datos en lugar de "datos de la base".

### 2 Objetivos básicos de un diccionario de datos.

Uno de los objetivos básicos de un diccionario de datos es permitir el manejo y la documentación de los datos, puesto que la base de datos sirve a varios usuarios, es vital que cada uno de ellos entienda exactamente ¿qué son los datos? y ¿qué significan?.

Debe existir una definición o descripción entendible de cada uno de los datos. Sin embargo, si tres departamentos distintos están usando los mismos datos en tres contextos diferentes, no es nada fácil para los tres departamentos ponerse de acuerdo sobre una definición o una descripción para el campo compartido. Los problemas de esta naturaleza tienen que ser resueltos por la función de la administración de la base de datos.

En esta etapa, cuando se está intentando llegar a una descripción textual de cada parte de los datos, no se necesita poner atención a la presentación física actual. El diseñador no tiene que determinar si los datos se van a almacenar empaquetados o en forma de carácter o en algún otro formato.

Es aconsejable comenzar a recabar la información sobre los datos en un diccionario de datos en el mismo día en que el proyecto se echa a andar. Tan pronto como el proyecto se pone en marcha, el diseñador empieza a hacerle a cada usuario preguntas tales como qué clase de sistemas desea, qué información requiere del sistema y que tipo de entrada puede proporcionar. Tan pronto como el usuario y el diseñador empiecen a platicar sobre las necesidades del primero, van a usar los nombres de los campos de datos; en una empresa bancaria son ejemplos "numero de cuenta", "saldo" e "interés". El diseñador y el usuario deben estar convencidos de que cuando usa un término se refieren precisamente a lo mismo; de otra manera el diseñador puede construir un sistema que no es el que el usuario desea. Esto destaca otro objetivo básico de un diccionario de datos: ayudar a establecer una comunicación efectiva entre el diseñador y los usuarios.

Dos objetivos básicos de un dicc. de datos son la administración y el control de los datos como un recurso, en un lugar central, a través de la fase de diseño, realización y operación, así como el establecimiento de una comunicación efectiva entre todos los que estén interesados en la base de datos.

## Capítulo I

---

En el caso de un medio de base de datos distribuido, las bases de datos o porciones de éstas pueden estar almacenadas físicamente en diferentes lugares o computadoras y conectados por medio de instalaciones de comunicación de datos. Es posible que algunas instalaciones puedan almacenar y acceder datos solamente en los lugares locales, mientras que otras instalaciones pueden almacenar y acceder datos locales y remotos.

Un diccionario de datos instalado en cada lugar puede proveerse con la información sobre los lugares físicos donde los datos están almacenados, así como el control de acceso y las restricciones de seguridad y privacidad. Con ayuda de esta información, el diccionario de datos puede decidir "inteligentemente" si satisfacer una consulta a la base de datos local o si el usuario está autorizado para acceder los datos remotos, en cuyo caso dicha solicitud se manda al lugar remoto.

Algunas personas argumentan que el diccionario de datos también se puede poner en papel, ya sea a mano o a máquina. Pero el problema con un diccionario de datos manual es que es difícil satisfacer al diseñador que va a necesitar la definición de un campo de datos clasificado de diferente manera. El mismo campo puede ser usado por diferentes usuarios en diferentes contextos. En la etapa inicial habrá una lista de los campos que de alguna manera están semirrelacionados. Sin embargo, el diseñador, al proceder a través de la fase de diseño, se encontrará con que los campos se usarán en varios lugares distintos; aparecerán en los formatos de entrada y en los de salida, relacionados uno con otro. En cada lugar los campos de datos aparecerán en un contexto distinto, lograr ésto implica un trabajo tedioso y al poco tiempo las listas serán difíciles de manejar. Una versión automatizada sería más fácil de manejar y controlar.

Con el fin de lograr que el diccionario de datos sea aceptado y utilizado para el procesamiento de datos, necesita tener una entrada central. La información puesta en esta unidad fuente se puede usar a lo largo de todas las fases de diseño y los programadores la pueden copiar en sus programas para utilizarla en estructuras de datos. Si se utiliza un diccionario manual o no integrado, con el tiempo la información que contiene se vuelve inconsistente. A un diccionario de datos también se le llama meta base de datos que almacena información sobre la base de datos.

### 3 El sistema de manejo de la base de datos y el diccionario de datos.

El diccionario de datos también se puede usar en un medio ajeno a una base de datos, es un lugar central de información de descripciones de los datos, tales como significado y relaciones con otros datos, es responsabilidad del diseñador tener los datos actualizados, así como tener registrado el origen de estos. En un medio de base de datos, la información almacenada en un diccionario de datos es sobre los datos almacenados en la base, mientras que en un medio ajeno a una base de datos, la información almacenada en el diccionario es sobre los datos almacenados en archivos de datos.

Es necesario instalar software para crear y manejar el diccionario de datos de una base de datos. El software también se conoce como diccionario de datos. El paquete del diccionario de datos se puede integrar dentro de un sistema de manejo de base de datos o tratarse aisladamente.

En el caso de un diccionario de datos integrado, las descripciones de los datos existen sólo una vez en el sistema y están almacenadas en el diccionario de datos. En un diccionario de datos independiente existe la opción de recuperar las descripciones apropiadas de los datos del diccionario mismo, o bien proporcionarle al diccionario las descripciones.

## **Capítulo I**

---

En caso de un diccionario de datos integrado, no hay opción; es necesario verificar la exactitud de las descripciones de los datos antes de la ejecución de un programa. Un diccionario de datos independiente puede requerir o no una verificación de la actualización de las descripciones de los datos antes de ejecutar un programa.

### **Ventajas de un sistema de diccionario de datos integrado.**

- Las descripciones de los datos no están almacenadas redundantemente en un paquete de diccionario de datos y en el sistema de manejo de la base de datos. esto reduce la ocurrencia de errores debidos a fallas en la actualización de ambos.
- El diccionario de datos tiene acceso a los datos de la base de datos. Un uso potencial del diccionario de datos puede ser en el área de seguimiento del acceso a los datos, al proporcionar estadísticas valiosas para mejorar el funcionamiento.
- Un diccionario de datos puede servir como una herramienta de control mucho más poderosa cuando está integrado con el DBMS, ya que el diseñador de la base de datos y los usuarios tendrán que reforzar el diccionario de datos como una herramienta para la documentación y el control de los datos.

**Ventajas de un diccionario de datos independiente**

- Hay menos riesgo al implantar en el sistema de manejo de la base de datos, un diccionario independiente que uno integrado. También la implantación de un diccionario independiente es más sencilla, ya que el diccionario no tiene que ajustarse a las características de implantación de un DBMS.
- Un diccionario de datos integrado necesita al mismo tiempo todas las descripciones de los datos requeridas para una base de datos, mientras que estas descripciones se le pueden proporcionar por etapas al diccionario de datos independiente. En el medio actual de procesamiento de datos no todos los archivos de datos se convierten al mismo tiempo a la base de datos. Una base de datos existe junto con algunos archivos de datos convencionales. Es deseable tener un paquete de diccionario de datos con el que se puede vigilar a los datos en la base de datos tal como lo haría un diccionario de datos integrado y que además vigile los datos contenidos en archivos como lo haría un diccionario independiente.

**4 El diccionario de datos y sus interfaces.**

Consideremos las interfaces de un diccionario de datos en un medio que dispone de un único sistema de manejo de la base de datos como en la figura 1.2.2.A.1. En la etapa inicial del proceso de diseño de la base de datos, la función de administración de la base de datos interactuará con el diccionario de datos. Con la ayuda de un generador de informes (que probablemente es una parte integral del paquete del diccionario de datos, a la dirección y los usuarios se les proveerá de informes adaptados a las necesidades individuales de cada uno.



Existen dos tipos de interfaces:

1. La interface con las personas participantes, por ejemplo: el administrador de la base de datos, el programador de sistemas, el analista de sistemas, el programador de aplicaciones, la administración, el usuario final y el auditor.
2. La interface con el software, por ejemplo, el sistema de manejo base de datos, los compiladores, el sistema operativo y los generadores de informes.

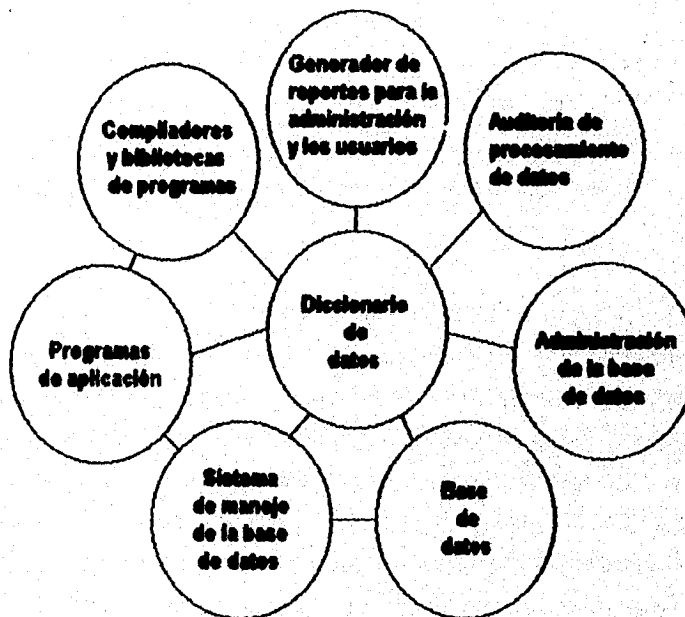


Figura 1.2.2.A.1. Interfaces del diccionario de datos en un medio de base de datos ideal.

Los informes pueden contener información sobre lo siguiente:

- Los campos de datos y las entidades.
- Las relaciones entre los campos de datos y entre las entidades.
- Responsabilidad de los usuarios de proporcionar datos correctos.
- Frecuencia del uso y descripción textual de los elementos.
- Información de control de acceso.
- Informes de contabilidad del sistema.
- Informes sumarios predefinidos.
- Informes adecuados diseñados por los usuarios.
- Informes de referencia cruzada.
- Informes sobre cambios.
- Informes de error.

Un diccionario de datos también puede generar descripciones de datos para lenguajes de alto nivel, en forma de dialectos de lenguajes apropiados para las bibliotecas de programas y para los compiladores. La información sobre las relaciones de referencias cruzadas entre los campos de datos y los programas de aplicación se pueden almacenar en el diccionario de datos. La lista de referencias cruzadas posibilitará a cualquier persona autorizada para informar a cualquier programador de aplicaciones o departamento sobre cualquier posible modificación antes de que se lleve a cabo, llegar a un acuerdo si es necesario, entonces, implantar el cambio. El cambio puede comprender, formato o relaciones con otros campos de datos.

Idealmente, la interface entre el sistema de manejo de la base de datos y el diccionario de datos debería ser tal que los directorios del DBMS fueran legibles para el sistema de diccionario de datos, con el fin de proporcionar el estado actual del DBMS. Los tipos de datos en la base de datos deberán actualizarse después de que el sistema de diccionario de datos tome nota de los cambios y

## Capítulo I

---

los datos deberán actualizarse sólo cuando el DBMS los ha encontrado aceptables. Así, el diccionario de datos, el DBMS y la base de datos deberán formar un círculo estrecho.

El diccionario de datos puede ser utilizado como una herramienta efectiva para la función de administración de la base de datos en la fase de diseño, implantación y operación de la base de datos es responsabilidad del ABD (Administrador de la Base de Datos) proteger el diccionario de datos imponiendo normas y medidas de seguridad y privacidad. En un medio donde la base de datos se comparte, la supervisión se ha vuelto obligatoria.

Un diccionario de datos es el lugar ideal para encontrar respuestas a preguntas como ¿dónde se usó?, ¿quien lo usó? y ¿cuando se usó?, tomando en cuenta los campos de datos y las entidades. Es también el lugar ideal para recabar información para un estudio de costo/eficiencia de los sistemas de la organización, así como para satisfacer las necesidades de privacidad particular y de confiabilidad.

Esta interface muestra que existen dos tipos de funciones de diccionario de datos. El primer tipo de uso lo realizan personas en sus funciones de administrador de base de datos, programador de sistemas, analistas de sistemas, programadores de aplicación, dirección, usuario final y auditor. El otro tipo de uso lo realiza el software en las áreas de: sistema de manejo de la base de datos, compiladores, sistemas de operación y generadores de informes. Estos dos tipos de interfaces dan realce a la dirección y control de la base de datos como resultado de la comunicación efectiva entre los participantes.

**5 El diccionario de datos ideal, sus requisitos y organización.**

La siguiente es una lista de los requisitos deseables para formar un diccionario de datos que los describa. Esto no significa que cualquier diccionario de datos, disponible en la actualidad, contenga estos requisitos en su totalidad.

**Modelo conceptual.** La información sobre los datos que se necesitan en el proceso de diseño del modelo conceptual incluye entidades, campos de datos o atributos que representan a las entidades y a las relaciones entre los campos. La información sobre cuáles departamentos y usuarios usan normalmente o intentan usar, qué campos y con qué frecuencia se usan éstos, junto con las descripciones textuales con significado y propósito, también deben ser almacenadas en un diccionario de datos. Los campos de datos, entidades y relaciones deben tener etiquetas, versiones, condiciones (propuestas convenidas, aprobadas, efectivas), descripciones textuales, sinónimos (otras entidades de diccionario de datos con el mismo o diferente significado), miembros (entidades de las que un campo de datos particular es miembro y por medio de éste se hace referencia a la entidad) y campos de datos agrupados con los campos clave apropiados.

**Modelo lógico.** La siguiente información sobre el modelo lógico de la base de datos debe ser almacenada en el diccionario de datos. Las agrupaciones de campo de datos con los elementos clave (las cuales pueden ser subconjuntos de las agrupaciones específicas en el modelo conceptual), el modelo de datos fundamental, las relaciones de las agrupaciones basadas en el modelo de datos, los modelos externos contenidos en el modelo lógico (esto es, diferentes trayectorias lógicas para el procesamiento de información), las transacciones, programas y módulos. Deben almacenar también las interdependencias de las transacciones, programas y módulos. La información de las referencias cruzadas entre transacciones, programas, módulos y departamentos o las funciones a las que éstos sirven, deben ser incluidos. Otra información necesaria es el lenguaje

de programación y el tipo de programa (por lote, teleprocesamiento) para los programas y las transacciones.

**Modelo interno.** La representación física de la información sobre los campos de datos por ejemplo, longitud (caracteres) forma (cadena de bits, cadena de caracteres, punto flotante simple), precisión (para los elementos numéricos), justificación o alineación (derecha o izquierda), imagen (solamente para propósitos de representación), reglas de edición (constante, rango de valores), algoritmo de derivación (para datos calculados), secuencia (la posición secuencial que un campo de datos particular ocupa entre los miembros), seguridad (código de seguridad para la lectura, actualización), medio (tarjeta, disco, cinta, video), dispositivos donde se almacenan las bases de datos y la información de control de acceso deben ser almacenadas en el diccionario de datos. Un diccionario de datos ideal debe ser parte integral del medio de la base de datos completa. El administrador de la base de datos debe ser responsable de la implantación del diccionario de datos. Siendo el diccionario de datos la parte central del medio de la base de datos, debe ponerse a salvo mediante copias de respaldo para evitar efectos desastrosos debido a un mal funcionamiento del sistema o la destrucción accidental de la versión activa del diccionario. La función de administración de la base de datos es responsable de la protección de la parte vital del medio de la base de datos: el diccionario de datos.

Un paquete ideal de diccionario de datos debe apoyar también las siguientes características:

**Recursos de recuperación y de informe.** Cualquier herramienta de comunicación y documentación puede ser efectiva si proporciona recursos sencillos de recuperación y de informe. Los recursos deben contener requisitos tanto formales como adecuados.

Los informes pueden incluir las siguientes partes:

- Lista de los campos en secuencia alfabética o en cualquier otra deseable.
- Listado de referencia cruzada entre los campos de datos, grupos de campos de datos, programas que lo están usando y departamento o gente responsable de mantener válidos los contenidos.
- Descripciones de datos para los programas en lenguaje huésped y los enfoques lógicos para los modelos externos que contienen a los programas de aplicación.

El diccionario de datos también debe contener la información suficiente para proporcionar alguna indicación sobre cualquier efecto en los programas o transacciones como resultado de algún cambio en el modelo de datos. La previsión de las estimaciones de costos para implantar el cambio propuesto es una característica muy deseable.

**Captura de datos como entrada al diccionario de datos.** Idealmente, el diccionario de datos debe ser usado como herramienta por:

- El administrador de la base de datos.
- Los programadores de aplicaciones.
- Los usuarios.
- La dirección y cualquier programa que interactúe con la base de datos.

Los datos almacenados en el dicc. de datos deben ser introducidos por personas que dominen un lenguaje de entrada orientado al usuario o por el software proveniente de las definiciones de programas, de las definiciones del sistema de manejo de la base de datos y de procedimientos establecidos en el programa.

## Capítulo I

---

**Información de control de acceso.** El diccionario de datos puede contener información tomando en cuenta el control de acceso, especificando quién puede tener acceso a qué parte de la base de datos y de qué forma. La información puede usarse como sigue:

- **Antes de compilar programas e imprimir informes revise si existe cualquier contradicción entre las especificaciones para la recuperación de datos dadas por el solicitante y las especificaciones proporcionadas por el administrador de la base de datos.**
- **Para localizar con precisión cualquier mal uso de los datos por parte de la base de datos compartida por múltiples usuarios, implántese un conjunto riguroso de restricciones de seguridad.**

**Si el diccionario de datos va a ser usado para control de acceso, el mismo debe estar sujeto a este último. El número de personas elegibles para hacer modificaciones al diccionario de datos debe restringirse. Deben implantarse las restricciones de privacidad en los contenidos del diccionario de datos.**

### **Apoyo de programas de servicios.**

**Ya que el diccionario de datos es la parte central del medio de la base de datos, debe ser capaz de apoyar a los diferentes programas de edición, actualización y reestructuración de la base de datos, generadores de informes, etc.**

### **Generación de programas y código de descripción de datos.**

**El almacenar la información sobre los campos de datos, sus representaciones físicas, relaciones con los otros campos, y relaciones sobre los modelos lógicos ó externos, debe capacitar a un diccionario de datos para generar el código de descripción de datos, así como también algunos módulos del programa general tal como los módulos de acceso o entrada/salida.**

**Consistencia.**

La información contenida en un diccionario de datos debe estar completa, con un formato correcto y con las referencias cruzadas apropiadas, para satisfacer estos requisitos, debe revisarse la consistencia de cualquier entrada al diccionario de datos. La revisión debe basarse también en la información convertida entre los modelos conceptual, lógico, externo e interno almacenados en el diccionario de datos.

**Sistema ideal de diccionario de datos.**

1. Debe apoyar a los modelos conceptuales, lógico, interno y externo.
2. Debe estar integrado en el sistema de manejo de la base de datos.
3. Debe contener varias versiones de la documentación (por ejemplo, las versiones de prueba, la versión de producción).
4. Debe prestar apoyo a la transferencia eficiente de información al sistema de manejo de la base de datos, idealmente la conexión entre los modelos externos y el modelo interno debe hacerse en el momento de ejecución, cuando el diccionario de datos debe tomar la información de la versión de producción y crear en forma dinámica la descripción de la base de datos y la descripción de programa.
5. Debe iniciar la reorganización de la versión de producción de la base de datos como resultado de los cambios a la descripción de la base de datos. Similarmente, cualquier cambio en la descripción del programa debe reflejarse automáticamente en la biblioteca de descripción de programas con la ayuda del diccionario de datos. Esto sucederá cuando el sistema de diccionario de datos sea una parte integral del DBMS.



## Capítulo I

---

### **Cuando aplicar el diccionario de datos**

Es importante la elección de un intervalo razonable para la instalación del diccionario de datos ya que una elección errónea puede arriesgar la totalidad del proyecto de la base de datos, sin embargo, no se debe olvidar que no existe una "receta de cocina" para cada empresa. El plan de implantación del diccionario de datos, como cualquier otra cosa en el procesamiento de datos, depende del ambiente de la empresa.

Puede resultar ambicioso instalar un diccionario para la base de datos de toda la empresa, así como lo es el instalar una base de datos para toda la empresa.

El proyecto de instalar un diccionario de datos para toda la empresa no debe ser llevado a cabo antes de alcanzar un punto de altura adecuada en la curva. Debe elegirse una de las siguientes posibilidades para la primera aplicación con el paquete del diccionario de datos.

### **Un sistema de aplicación convencional.**

Se puede encontrar un sistema de aplicación que no afecte a los otros sistemas de aplicación de manera importante y que su tamaño sea moderado, puede considerarse como un posible candidato. Al mismo tiempo, esta aplicación también debe tener algo de viabilidad e importancia en el funcionamiento de la empresa. Más aún, debe ser dinámico, con cambios, ya que de este modo el diccionario de datos puede demostrar sus beneficios y efectividad potencial en un medio variable.

### **Un conjunto de aplicaciones existentes del sistema de manejo de base de datos.**

El funcionamiento de las aplicaciones del sistema de manejo de base de datos parece tener gran viabilidad en una empresa y tiende también a ser dinámico. Si se pretende ampliar el conjunto de datos de aplicación, la introducción de un

sistema de diccionario de datos puede marcar la ocasión correcta para mostrar la fuerza del diccionario de datos en las áreas de comunicaciones entre usuarios, documentación y almacenamiento de información sobre cierto número de aplicaciones almacenadas en un lugar central ligado.

**Un nuevo conjunto de aplicaciones.**

Si una empresa planea ampliar sus aplicaciones o convertir sus aplicaciones convencionales existentes a un medio de base de datos, las nuevas aplicaciones significarán una elección excelente. A los programadores, analistas y usuarios no necesariamente les gustará la idea de documentar las aplicaciones que ya están en ejecución, sin embargo, para los nuevos sistemas de aplicaciones que se van a desarrollar, el trabajo aún no se ha hecho y la documentación de las aplicaciones por vez primera es un incentivo que no se debe soslayar.

**Directrices para la definición de datos.**

Si el diccionario de datos ha de fungir como una herramienta de comunicación eficaz, debe existir un entendimiento común entre los usuarios en lo relativo a los bloques componentes de su construcción. Las partes básicas de la(s) base(s) de datos de diccionario son las siguientes:

**Campo de datos.** Atributo que describe a una entidad, cada campo tiene un nombre o etiqueta única. Estos deben ser descriptivos y consistir de palabras claves o abreviaciones tomadas de listas autorizadas.

**Campo de grupo.** Agrupación de campo de datos lógicamente relacionados. Para un campo de grupo todos los subelementos (elementales y campos de grupo), deben estar identificados explícitamente. Para los elementos básicos, todos los campos miembros de un grupo deben estar identificados explícitamente.

**Campo derivado.** Es un campo de datos que se origina como resultado de una fórmula o de un cálculo. Para todos los campos derivados, la fórmula o el cálculo deben estar plenamente identificados, junto con todos los campos de datos usados en la derivación. Estos campos también deben estar definidos en el diccionario.

**Sinónimo o alias.** Es un campo de datos con designación distinta pero con el mismo significado que otro campo. La descripción de un sinónimo debe identificar las designaciones de otros campos de datos de los cuales éste es sinónimo.

**Homónimos.** Son dos campos de datos con la misma designación pero con distinto significado. Al introducir las designaciones del campo en el diccionario, los problemas de sinónimos y homónimos deben ser identificados por el diccionario y resueltos por el equipo de diseño de la base de datos.

**Descripción del modelo conceptual.** El modelo describe las entidades de la empresa, las relaciones entre ellas y el flujo de información.

**Descripción del modelo lógico.** El modelo es un posible subconjunto del modelo conceptual y para utilizarlo se transforma al paquete del sistema de manejo de base de datos.

**Descripción del (los) modelo(s) externo(s).** Los modelos externos describen los puntos de vista de los programas de aplicación, esto es, las formas en las que las distintas aplicaciones ven las relaciones entre las entidades.

**Descripción del modelo interno.** El modelo interno es la conversión física de la DB. Describe las relaciones entre las entidades, los métodos de acceso que se han de usar, los elementos de grupo y otros detalles de conversión física.

Además de las descripciones de los bloques y componentes dadas anteriormente, se debe proporcionar la siguiente descripción de las relaciones entre ellos:

**Texto.** El texto proporciona información que capacita al usuario del diccionario para decidir si un campo de datos es aplicable o potencialmente aplicable a sus requerimientos. Esta descripción debe ser comprensible para todos los usuarios potenciales del diccionario, pero a la vez debe ser lo suficientemente específico para distinguir un campo de datos particular de otros semejantes.

**Deben observarse las siguientes reglas:**

1. **Usar palabras específicas y no ambiguas.**
  - Evitar frases como:
    - Algunos empleados.
    - Ciertos productos.
  - En lugar de lo anterior usar por ejemplo:
    - Empleados retirados.
    - Productos de papel.
  
2. **Abreviar sólo en caso necesario debido a limitaciones de espacio.**  
**Ser específico en las abreviaciones.**
  
3. **Evitar un lenguaje con audiencia limitada**  
**Por ejemplo, procesamiento de datos, contabilidad, relaciones laborales, etc.**
  
4. **Establecer unidades de medida**  
**(miles de dólares, metros, centímetros, etc.) o frecuencias de ocurrencia**  
**(por año, turno, etc.) de cantidades, promedios, etc.**

5. Indicar la fuente de los datos.

El texto también debe especificar el (los) documento (s) donde se usan los datos, así como el (los) programas (s) y el (los) departamento (s) o usuario (s) que requieren los datos.

**B) DIAGRAMA DE FLUJO DE DATOS**

El diagrama de flujo de datos (DFD). Es una de las herramientas gráficas de modelado más importantes del análisis estructurado. Esta es una herramienta que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por "conductos y tanques de almacenamiento" de datos.

El DFD es una de las herramientas más comúnmente usadas, sobre todo por sistemas operacionales en los cuales las funciones del sistema son de gran importancia y son más complejas que los datos que éste maneja.

Los DFD no sólo se puede utilizar para modelar sistemas de procesos de información, sino también como manera de modelar organizaciones enteras, es decir, como una herramienta para la planeación estratégica y de negocios. Los componentes de un diagrama típico de flujo de datos son: el proceso, el flujo, el almacén y el terminador.

**El proceso.**

Muestra una parte del sistema que transforma entradas en salidas, es decir, muestra cómo es que una o más entradas se transforman en salida, el proceso se representa gráficamente como un círculo como se muestra en la figura 1.2.2.B.2. El proceso se nombra o describe con una sola palabra, frase u oración sencilla, (verbo-objeto) tal como VALIDAR ENTRADA.

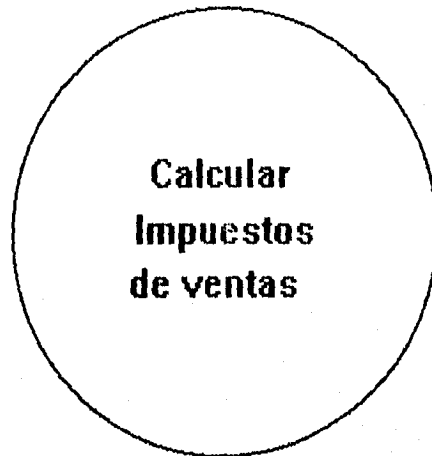


Figura 1.2.2.B.2. Ejemplo de un proceso

En algunos casos, el proceso contiene el nombre de una persona o un grupo (por ejemplo un departamento o división de una organización), computadora o aparato mecánico. Es decir, el proceso a veces describe quién o qué lo está efectuando, más que describir el proceso mismo.

**El flujo.**

Un flujo se representa gráficamente por medio de una flecha que entra o sale de un proceso; un ejemplo se muestra en la figura 1.2.2.B.3. El flujo se usa para describir el movimiento de bloques o paquetes de información de una parte del sistema a otra.



Figura 1.2.2.B.3 Ejemplo de un flujo.



Figura 1.2.2.B.4. Ejemplo de un almacén

Por ello, los flujos representan datos en movimiento, mientras que los almacenes representan datos en reposo. Figura 1.2.2.B.4.

En la mayoría de los sistemas, los flujos realmente representan datos, es decir, bits, caracteres, mensajes, números de punto flotante y los diversos tipos de información con los que las computadoras pueden tratar. Pero los DFD también pueden utilizarse para modelar otros sistemas aparte de los automatizados y computarizados.

Nótese también que los flujos muestran la dirección: una cabeza de flecha en cualquier extremo del flujo indica si los datos se están moviendo hacia adentro y/o hacia afuera de un proceso.

El flujo que se muestra en la figura 1.2.2.B.5, indica claramente que el número se está mandando hacia el proceso denominado VALIDA NUMERO TELEFÓNICO. Y el flujo denominado ITINERARIO DE ENTREGA DE CONDUCTOR de la figura 1.2.2.B.6.

Claramente indica que es una salida generada por el proceso GENERAR ITINERARIO DE ENTREGA DE CONDUCTOR los datos que se mueven a lo largo de dicho flujo viajarán ya sea a otro proceso (como entrada) o a un almacén o a un terminador.

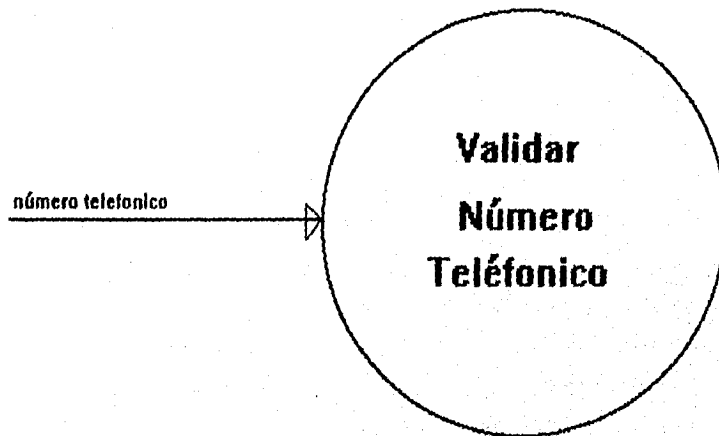


Figura 1.2.2.B.5. Flujo de entrada

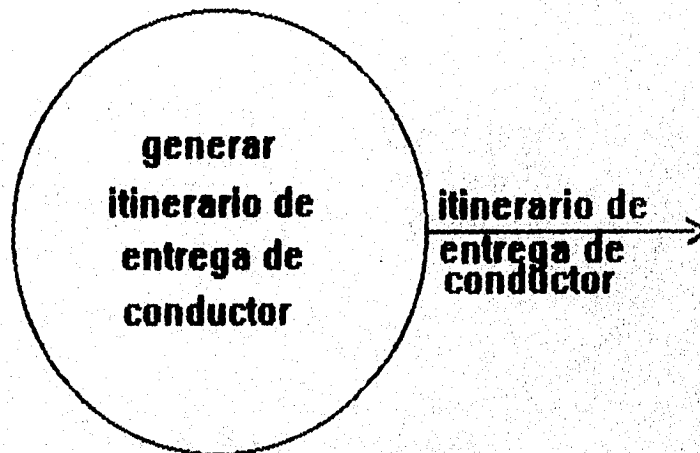


Figura 1.2.2.B.6. Flujo de salida



El flujo de dos cabezas que se muestra en la figura 1.2.2.B.7, es un diálogo, es decir, (una pregunta y una respuesta) en el mismo flujo. En el caso de un diálogo, los paquetes en cada extremo de la flecha deben nombrarse, como se ilustra en la figura 1.2.2.B.7.

Los flujos de datos pueden divergir o converger en un DFD; conceptualmente ésto es algo así como un río principal que se divide en varios más pequeños, o varios pequeños que se unen, sin embargo, esto tiene un significado especial en un DFD típico, en el cual hay paquetes de datos que se mueven a través del sistema; en el caso de un flujo divergente, esto significa que se están mandando copias por duplicado de un paquete de datos a diferentes partes del sistema, o bien que un paquete complejo de datos se está dividiendo en varios paquetes individuales, cada uno de los cuales se está mandando a diferentes partes del sistema, o que el ducto de flujo de datos lleva artículos con distintos valores (por ejemplo, vegetales cuyos valores pudieran ser "papa", "col", "tomate", "lima" o "ejotes) que están siendo separados.

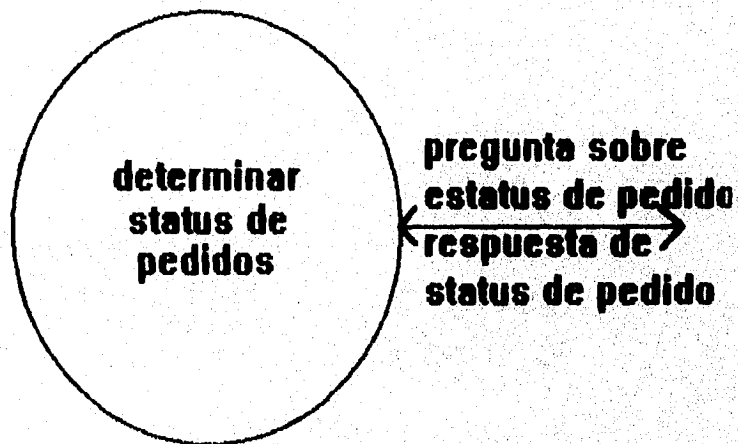


Figura 1.2.2.B.7. Flujo de diálogo

De manera inversa, en el caso de un flujo convergente, significa que varios paquetes elementales de datos se están uniendo para formar agregados más complejos de paquetes de datos. Por ejemplo, la figura 1.2.2.B.8, muestra un DFD en el cual el flujo denominado DETALLES DE PEDIDOS diverge y lleva copias de los mismos paquetes a los procesos GENERAR DOCUMENTOS DE ENVÍO, ACTUALIZAR INVENTARIO y GENERAR FACTURAS.

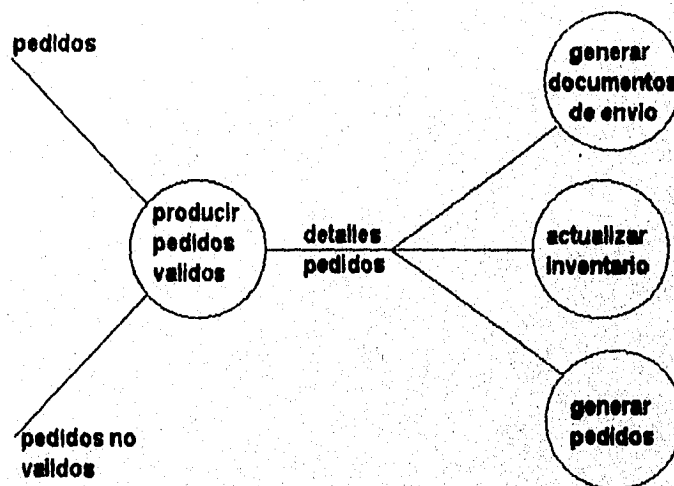


Figura 1.2.2.B.8. Flujo divergente

La figura 1.2.2.B.9 muestra como el flujo DOMICILIO DE CLIENTE se divide en tres paquetes más elementales NÚMERO TELEFÓNICO, CÓDIGO POSTAL, CALLE Y NÚMERO, los cuales se mandan a tres procesos de validación diferentes.

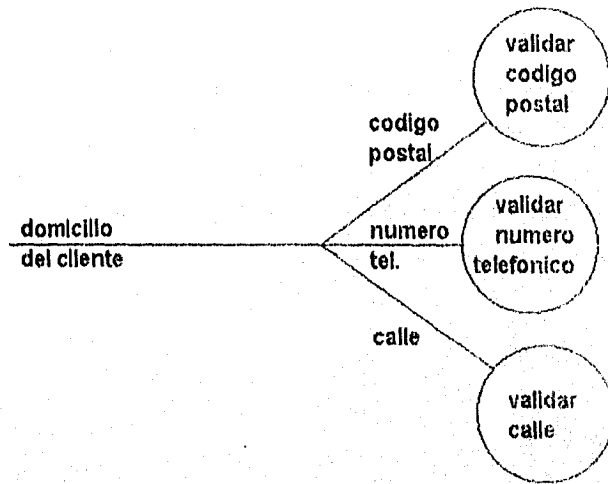


Figura 1.2.2.B.9. Flujo convergente

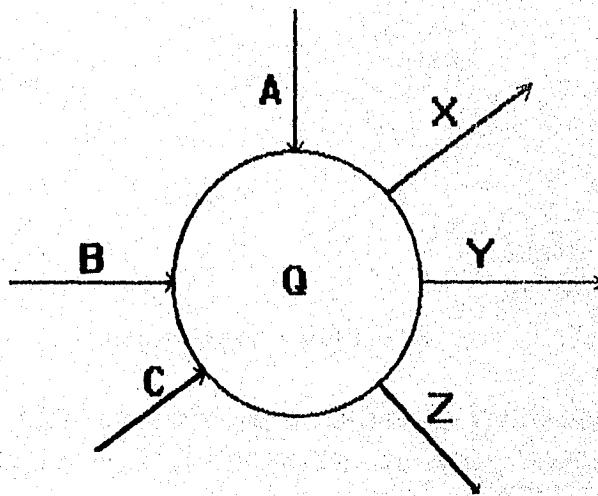


Figura 1.2.2.B.10. Combinación de flujos de salida y entrada

Finalmente, considere la situación más común que se muestra en la figura 1.2.2.B.10 en donde hay múltiples flujos de entrada y de salida: ¿En qué secuencia llegan los paquetes de datos y en qué secuencia se generan los paquetes de salida? Es decir, ¿El proceso Q requiere exactamente un paquete de los flujos A, B y C para producir exactamente un paquete de salida para los flujos X, Y y Z? ¿O existen dos A's para cada tres B's? La respuesta a todas estas preguntas es muy sencilla: no sabemos, si todas estas interrogantes acarrearán detalles de tipo procedimiento, que son el tipo de preguntas que se modelaría normalmente con un diagrama de flujo de datos o alguna otra herramienta de modelado de tipo procedimiento. El DFD simplemente no intenta abordar estas cuestiones. Si estas preguntas se vuelven importantes, entonces tendrá que modelarse el procedimiento interno de los diversos procesos las herramientas para hacer esto se discuten más adelante.

#### **El almacén.**

El almacén se utiliza para modelar una colección de paquetes de datos en reposo. Se denota por dos líneas paralelas, como lo muestra la figura 1.2.2.B.4. De modo característico el nombre que se utiliza para identificar el almacén es el plural del que se utiliza para los paquetes de entrada y salen del almacén por medio de flujos. Por lo general los almacenes son referidos como archivos o bases de datos, pero un almacén también pudiera consistir en datos almacenados en tarjetas perforadas, microfilms, microfichas, disco óptico o alguna más de otras posibles formas electrónicas. Y un almacén también puede ser un conjunto de fichas de papel en una caja de cartón, nombres y domicilios en un directorio, diversos archivos en un archivero, o varias formas no computarizadas. Es precisamente debido a la variedad de formas de implantación posibles de un almacén que deliberadamente escogimos una notación gráfica simple y abstracta así como el término almacén en lugar de, por ejemplo base de datos.

Aparte de la forma física que toma el almacén, también existe la cuestión de su propósito: ¿Existe el sistema por causas de un requerimiento fundamental del usuario o por algún aspecto conveniente en la realización del sistema? En el primer caso, la base de datos existe como un área de almacenamiento diferida en el tiempo, necesaria entre dos procesos que ocurren en momentos diferentes.

Por ejemplo la figura 1.2.2.B.11, muestra un fragmento de un sistema en el cual, como política del usuario, el proceso de entrada de órdenes puede operar en tiempos diferentes que el proceso de investigación de órdenes.

El almacén de PEDIDOS debe existir en alguna forma, ya sea en disco, cinta, tarjetas o inscritos en piedra.

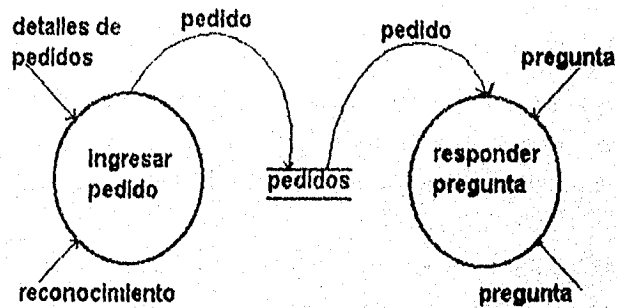


Figura 1.2.2.B.11. Un almacén necesario

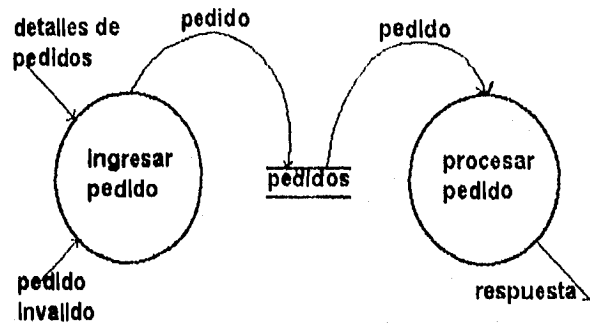


Figura 1.2.2.B.12.

Almacén "de implantación"

La figura 1.2.2.B.12 muestra un tipo distinto de almacén: el almacén de Implantación, podemos imaginar que el diseñador del sistema interpondrá un almacén de PEDIDOS entre INGRESAR PEDIDO y PROCESAR PEDIDO porque:

- Se espera que ambos procesos se ejecuten en la misma computadora, pero no hay suficiente memoria para cubrir ambos al mismo tiempo. Así el almacén de ÓRDENES se crea como archivo intermedio, pues la tecnología de Implantación disponible ha forzado a que los procesos se ejecuten en tiempos distintos.
- Se espera que cualquiera de los procesos, se ejecuten en una configuración de hardware que es poco confiable. Así el almacén de ÓRDENES se crea como respaldo en caso de que cualquiera de los procesos se aborte.

- Se espera que diferentes programadores implanten los dos procesos (o en caso más extremo, que lo hagan diferentes grupos de programadores que trabajan en lugares geográficos distintos). Así, el almacén de ÓRDENES se crea para probar y corregir, de manera que si el sistema completo no trabaja, ambos grupos puedan ver los contenidos del almacén y detectar el problema.
- El analista o el diseñador pensaron que el usuario podría algún día hacer accesos al almacén de ÓRDENES por alguna razón, aún cuando no haya expresado tal interés. En éste caso, el almacén se crea anticipando necesidades futuras del usuario (y dado que costará algo implantar el sistema de esta manera, el usuario acabará pagando por algo que no se pidió).

Si fuera a excluirse los asuntos y modelar sólo los requerimientos esenciales del sistema, no existiría necesidad de un almacén de PEDIDOS; en lugar de eso, se tendría un DFD como el que se muestra en la figura 1.2.2.B.13.

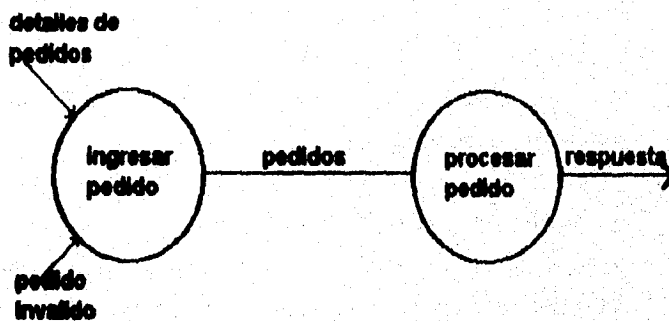


Figura 1.2.2.B.13. Almacén de implantación eliminado

Normalmente se interpreta un flujo que procede de un sistema como una lectura o un acceso a la información del almacén. Esto significa específicamente que:

- Se recupera del almacén un solo paquete de datos, esto es, de hecho, el ejemplo más común del flujo desde un almacén. Imagínese, por ejemplo, un almacén llamado CLIENTES, donde cada paquete contiene nombre, domicilio y número telefónico de los clientes individuales. Así, un flujo típico del almacén podría implicar la recuperación de un paquete completo de información acerca de un cliente.
- Se ha recuperado más de un paquete del almacén. Por ejemplo, el flujo podría recuperar paquetes de información acerca de todos los clientes de la ciudad de Nueva York del almacén CLIENTES.
- Se tiene una porción de un paquete del almacén. En algunos casos, por ejemplo, sólo se podría recuperar la información del número telefónico del cliente del almacén CLIENTES.
- Se tienen porciones de más de un paquete del almacén. Por ejemplo, un flujo podría recuperar del almacén CLIENTES la porción del código postal de todos los clientes que viven en el estado de Nueva York.

Existe un detalle de tipo procedimiento del cual podemos estar seguros: el almacén es pasivo, y los datos no viajarán a lo largo del flujo a menos que el proceso lo solicite explícitamente. Existe otro detalle de tipo procedimiento que suponen por convenio, los sistemas de procesos de datos: el almacén no cambia cuando un paquete se mueve del almacén a lo largo del flujo. Un programador pudiera referirse a esto como una lectura no destructiva o en otras palabras, del almacén se recupera una copia del paquete y el almacén mantiene su condición original.



## Capítulo I

---

Un flujo hacia un almacén habitualmente se describe como una escritura, actualización o posiblemente eliminación. Específicamente, sólo puede significar que se tiene una de las situaciones siguientes:

- Se está guardando uno o más paquetes nuevos en el almacén. Dependiendo de la naturaleza del sistema, los paquetes nuevos pudieran anexarse; o colocarse en algún lado entre los paquetes. Esto es a menudo un asunto de la implantación (es decir, controlado por el sistema específico de administración de base de datos), por lo que el analista no debiera preocuparse acerca de ello. podría ser, sin embargo, cuestión de una política del usuario.
- Uno o más paquetes se están modificando o cambiando. Esto pudiera traer consigo un cambio de todo un paquete, o (más comúnmente), de solo una porción o de una porción de múltiples paquetes. Por ejemplo, suponga que la policía tiene un almacén de sospechosos y que cada paquete contiene sus nombres y domicilios, puede ofrecérsele una nueva "identidad" a un sospechoso que coopera, en cuyo caso toda la información relacionada con su paquete cambiaría. Como alternativa, considere un almacén CLIENTES que contenga información acerca de los clientes que residen en la ciudad de Puebla y que se les cambie el código postal se necesitaría un cambio a una porción de los diversos paquetes.

En todos estos casos es evidente que el almacén cambió como resultado del flujo que ingresa. El proceso conectado con el otro extremo del flujo es el responsable de realizar el cambio al almacén.

Los flujos conectados a un almacén, sólo pueden transportar paquetes de información que el almacén sea capaz de guardar.

**El Terminador.**

El siguiente componente del DFD es un terminador, gráficamente se representa como un rectángulo como se muestra en la figura 1.2.2.B.14. Los terminadores representan entidades externas con las cuales el sistema se comunica, comúnmente, un terminador es una persona o grupo, por ejemplo, una organización externa o una agencia gubernamental, o un grupo, o departamento que esté dentro de la misma compañía u organización, pero fuera del control del sistema que se está modelando.

En algunos casos, un terminador puede ser otro sistema, como algún otro sistema computacional con el cual se comunica éste.

Suele ser muy fácil identificar los terminadores en el sistema que se está modelando.

A veces el terminador es el usuario, es decir, en sus discusiones con el usuario, éste dirá "Pretendo suministrar al sistema los datos X, Y y Z, y espero que me regrese los datos A, B y C". En otros casos, el usuario se considera parte del sistema y ayudará a identificar los terminadores relevantes.

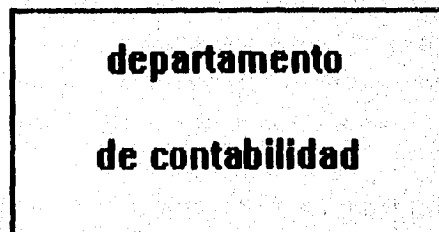


Figura 1.2.2.B.14. Representación gráfica de un terminador

## **Capítulo I**

---

Existen tres cosas importantes que debemos recordar acerca de los terminadores:

1. Son externos al sistema que se está modelando; los flujos que conectan los terminadores a diversos procesos (o almacenes) en el sistema representan la interfaz entre él y el mundo externo.
2. Como consecuencia, es evidente que ni el analista ni el diseñador del sistema están en posibilidades de cambiar los contenidos de un terminador o la manera en que éste trabaja. Lo que esto significa es que el analista está modelando un sistema con la intención de permitir una considerable flexibilidad y libertad al diseñador para elegir la mejor implantación posible (la más eficiente o la más confiable, etc.). Sin embargo, el analista de sistemas no puede modificar los contenidos, la organización ni los procedimientos internos asociados con los terminadores.
3. Las relaciones que existen entre los terminadores no se muestran en el modelo de DFD. Pudieran existir de hecho diversas relaciones, pero, por definición, no son parte del sistema que se está estudiando. De manera inversa, si existen relaciones entre los terminadores y si es esencial para el analista modelarlos para poder documentar los requerimientos del sistema, entonces por definición, los terminadores son en realidad parte del sistema y deberían modelarse como procesos.

En un sistema real típico pueden existir docenas de terminadores diferentes interactuando con él. Identificar los terminadores y su interacción con el sistema es parte del proceso de construir el modelo del ambiente.

#### **Guía para la construcción de DFD.**

Existe un número de reglas adicionales que se requieren para poder utilizar DFD con éxito. Algunas de estas reglas ayudarán para no elaborar DFD erróneos (por ejemplo, incompletos o lógicamente inconsistentes). Algunas de estas reglas tienen la finalidad de ayudar para dibujar un DFD grato a la vista, y por tanto, tenga más probabilidades de que lo lea con cuidado el usuario.

1. Escoger nombres con significado para los procesos, flujos, almacenes y terminadores.
2. Numerar los procesos.
3. Redibujar el DFD tantas veces como sea necesario estéticamente.
4. Evitar los DFD excesivamente complejos.
5. Asegúrese de que el DFD sea internamente consistente y que también lo sea con cualesquiera DFD relacionados con él.

#### **C) ARCHIVOS**

La principal tarea de manejo de datos es el almacenar y recopilar datos de manera eficiente y adecuada. La habilidad para localizar registros en un archivo con efectividad de costo, dados cualquier campo, llave o combinaciones de ellos, es esencial para la operación efectiva de los sistemas de información actuales que se centran en una computadora. Las organizaciones básicas de archivos (llamadas también métodos básicos de acceso o de indexación primaria) están disponibles en los sistemas de cómputo actuales, estas bien conocidas organizaciones para archivo son la secuencial, aleatoria y secuencial indexada. Aún cuando los detalles del funcionamiento de estas organizaciones pueden variar en los niveles inferiores del lenguaje de programación, sistema operativo o máquina específicos (particularmente para las organizaciones aleatoria e indexada), proporcionan una capacidad limitada de entrada al espacio de datos almacenados; es decir, el acceso directo a los registros es en

## Capítulo I

---

base al campo seleccionado como única llave de acceso, cuando se crea inicialmente el archivo. la facilidad señalada es mínima, ya que en la mayoría de los medios ambientes actuales, es esencial la habilidad de tener acceso a los datos en base a cualquiera de las llaves o combinaciones de ellas (cada pieza de datos elemental o campo es candidato a ser llave).

Considérese la consulta o transacción: "LISTAR A TODOS LOS CLIENTES CUYO SALDO ADEUDADO SEA MAYOR QUE 1000, Y CUYA COMPAÑIA ASEGURADORA SEA XXXXX", que pueda provenir de una consola en línea o de un programa por lotes (batch). Si ninguna de las llaves de acceso k1 y k2 de la consulta, que podrían ser SALDO Y COMPAÑIA-ASEGURADORA, coincide con la llave k0 de acceso al sistema, que podría ser NO\_DE\_CLIENTE, entonces será necesario que el programa de aplicación o el usuario examinen todos los N registros del archivo para identificar a los que satisfagan la consulta. Si cualquiera de las llaves k1 o k2 es en efecto la llave k0 del sistema, existen entonces dos posibilidades:

1. Si la consulta abarca la conjunción lógica k1 Y k2 (k1 AND k2), entonces de hecho se recopila solamente un subconjunto de los N registros y el programa o el usuario examina ese conjunto para localizar a los que satisfagan la condición sobre la otra llave.
2. Si la consulta abarca la disyunción lógica k1 O k2 (k1 OR k2) entonces es necesario que se obtenga y examine cada uno de los N registros.

En cualquier caso el hecho importante es que se obtienen muchos registros que no satisfacen el criterio de la consulta.

La complejidad de la recopilación o acción de obtener los registros, aumenta con el número de operadores lógicos Y y O (AND y OR) y con los operadores de comparación <, >, =, <> involucrados en la consulta. Una consulta o solicitud transacción de entrada/salida es de la forma: <parte operacional><parte calificadora>. Por ejemplo:

**Parte Operacional**

LISTAR NOMBRE, DEPARTAMENTO y SALARIO

**Parte Calificadora**

DONDE SALDO > 1000 Y CIA. ASEGURADORA = XXXX

A través de los años ha surgido un gran número de conceptos de esquemas importantes relacionados con las organizaciones físicas de datos y archivos; se les ha llamado organizaciones secundarias o de alto nivel para archivos y su propósito ha sido compensar la disparidad entre las complejas demandas de los usuarios para recopilación de información en base a llaves múltiples, y las facilidades básicas de llave única que proporcionan los sistemas operativos. El uso del término "secundarias" es inadecuado, ya que tales organizaciones son de primordial importancia en el manejo de datos. Las principales son:

1. Organizaciones de listas y multilistas.
2. Organizaciones parciales y completamente invertidas.
3. Organizaciones anilladas y arbóreas encadenadas.

## Capítulo I

### **Apuntadores**

Un apuntador es simplemente un indicador que guía hasta un dato elemental o registro desde algún otro punto en el espacio de datos. Un apuntador puede ser:

1. La dirección de la máquina del registro que se señala.

2. La dirección relativa numérica del registro que se señala.

Los registros podrían estar numerados en secuencia, de manera que el apuntador de registro sería un número relativo en el archivo. Por ejemplo, el apuntador que apunta a un registro, es un número secuencial relativo y la dirección real en la máquina se logra internamente mediante el módulo para acceso aleatorio del sistema operativo y es transparente al usuario del apuntador.

3. El identificador lógico del registro indicado.

A un apuntador cuyo valor es un identificador lógico y no una dirección, se le llama un apuntador simbólico. Podría ser un dato elemental numérico o no numérico que identifica en forma única a cada registro en el archivo. Observe que un número de identificación tal como un número de seguridad social, puede considerarse como apuntador simbólico siempre y cuando no implique nociones de dirección relativa. El identificador se convierte a la dirección real del registro mediante alguna transformación tal como una simple búsqueda en una tabla o algún otro esquema de transformación de llaves; esta conversión se realiza internamente en la máquina y es transparente al usuario.

### **Organizaciones de listas y multilistas**

Una estructura de lista simple es una secuencia de datos elementales o registros encadenada por apuntadores. Al incluirse un apuntador en cada registro para señalar al registro que sigue lógicamente, las organizaciones física y lógica pueden resultar completamente diferentes. En la figura 1.2.2.C.1 se muestra una serie de registros de personas en localidades físicas no contiguas y encadenados lógicamente por una sola lista de acuerdo a su orden alfabético. La longitud de la lista se define como el número de entradas encadenadas por la misma. Observe que una lista implica un costo de un espacio para el apuntador por cada registro encadenado.

Todas las estructuras de lista y los esquemas de indexación secundarios requieren del uso de dispositivos de acceso aleatorio; excluyen el uso de implementación usando cintas bajo casi todas las circunstancias. Un archivo en cinta puede considerarse una serie de registros en posiciones físicas no contiguas encadenados lógicamente por una lista sencilla como una secuencia de registros en la cual el apuntador se ha implementado por la contiguidad física, es decir, la contiguidad física y lógica coinciden. Los registros en el área prima de un archivo secuencial indexado, están ordenados físicamente de acuerdo a la secuencia

La figura 1.2.2.C.1 muestra la lógica de la llave que se ha escogido (los apuntadores resultan innecesarios), mientras que en las áreas de sobreflujo sus posiciones físicas y lógicas no coinciden y la secuencia lógica se mantiene mediante apuntadores.



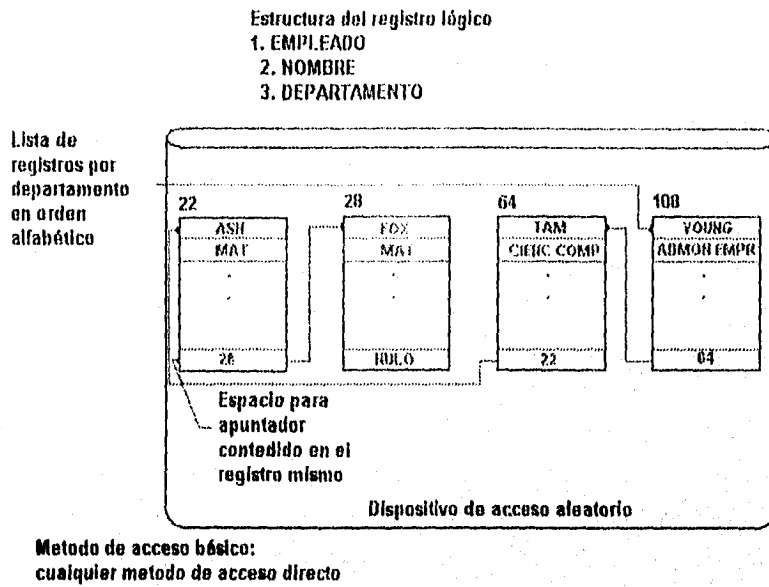


Figura 1.2.2.C.1

**Organización de multilista.**

Se puede hacer pasar prácticamente una infinidad de listas a través de un conjunto de un número N de registros. Una organización de multilista es simplemente la que abarca varias listas. En la figura 1.2.2.C.2 se muestra la organización de multilista básica para archivos. En la figura 1.2.2.C.3 se muestra la lista de registros de personal constituida por dos listas, una que ordena todos los registros por apellido y una que encadena a todos los registros en un rango de edad determinado. El uso del termino "enhebrado" ha conducido en ocasiones al uso de "organización enhebrada" como sinónimo de organización de multilista.

**Diseño y compromisos.**

Una de las tareas del diseñador de un banco de datos es el establecer el mejor balance entre las rutas de acceso y los requerimientos de las transacciones. Sin embargo, la tarea es más compleja cuando se incluyen inserciones, eliminaciones y actualizaciones (cambiar un registro que ya se encuentra almacenado). Existen tres factores que se encuentran en prácticamente todas las organizaciones de datos:

1. Los mecanismos de búsqueda y mantenimiento para la inserción, eliminación y actualización son más costosos e intrincados que para la recopilación de registros; frecuentemente son dos veces más costosos en términos de rapidez de procesamiento.
2. Al optimizarse el tiempo requerido para recopilación de registros, se incrementa el tiempo necesario para las demás operaciones.
3. La optimización del almacenamiento es usualmente a expensas del tiempo de ejecución y viceversa.

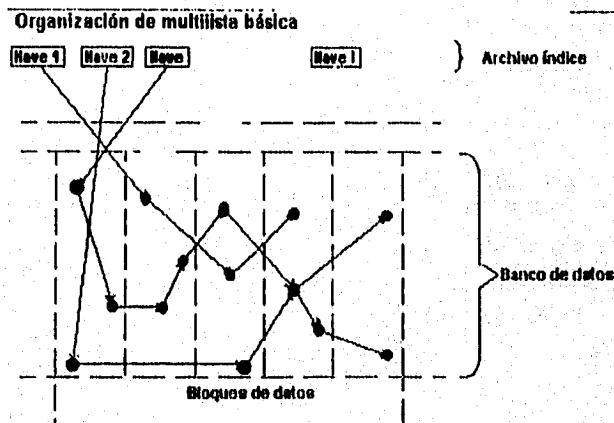


Figura 1.2.2.C.2. Organización de archivos como multilista básica.

Organización Invertida.

Una inconveniencia seria de las organizaciones de multilistas, así como de variantes de las mismas, tales como las diversas estructuras de anillo, es que la búsqueda en la lista involucra localizar efectivamente registros de la lista, para cada uno para obtener su(s) apunador(es) al siguiente registro y finalmente obtener éste. Esto implica tiempo de acceso en el rango de los milisegundos a segundos. El mejor funcionamiento de una lista de N registros se logra, ya sea cuando el valor de las llaves es único para cada registro, o cuando los registros están agrupados y ordenados por el valor de las llaves si éstos no son únicos, entonces se tendrá que recopilar  $(N+1)/2$  registros para encontrar los que se desean.

Por que a lo largo de nuestra experiencia, nos brinda el apoyo necesario para realizar la implementación de los que se desean.

- 1. EMPLEADOS
- 2. EMPLEADO
- 2. NOMBRE
- 2. DEPARTAMENTO
- 2. EDAD
- 2. SEXO
- 2. ESTADO
- 2. CIUDAD

- L1 Lista de aquellos de edad entre 24-26
- L2 Lista de aquellos de edad entre 26-30
- L4 Lista de aquellos de edad entre 30-40
- L5 Lista ordenada por apellido (el registro inicial de la lista tiene la dirección 22)

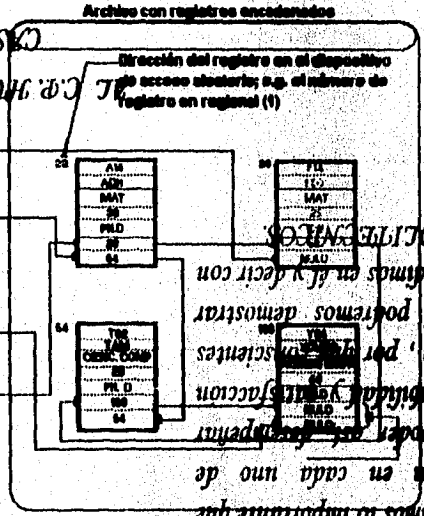


Figura 1.2.2.C.3. Ejemplo de la organización de archivos en multilista

orgullo somos: POLITECNICOS  
 todo lo que aprendimos en él y decir con  
 de nosotros demostramos  
 por los estudiantes  
 con nosotros y satisfacción  
 nosotros, para poder aprender  
 es la superación en cada uno de  
 desvelos, aprendimos lo importante que  
 todos los momentos de angustia y  
 Por que gracias a la estancia en él y a

Una alternativa mejor es la llamada *organización invertida*, mediante la cual el tiempo de acceso vía los campos elegidos para ser "invertidos" puede reducirse significativamente en varios órdenes de magnitud, respecto al enfoque de multilistas.

Su estructura básica comprende:

1. Los registros originaies sin contener apuntadores.
- 2.
3. El *diccionario invertido o directorio o índice* que contiene ciertos valores de los campos, seguidos de la lista de apuntadores a los registros caracterizados precisamente por esos valores de campos.

Los campos que se colocan en el directorio para proporcionar acceso rápido a los registros se llaman *llaves invertidas, llaves indexadas, llaves de acceso* o simplemente *llaves*. Con frecuencia se llama a la lista de apuntadores o lista apuntadora *lista de acceso*.

Es deseable que los apuntadores en el directorio invertido, o los apuntadores eslabón, almacenados dentro de los registros en las organizaciones de multilistas, sean direcciones relativas o identificadores en vez de direcciones físicas verdaderas. Esto permite cambios en la localización física de los registros a los que se apunta, sin requerir de cambios en el directorio invertido o en los apuntadores de eslabón, por supuesto que los mecanismos o tablas internos tendrán que reflejar la correspondencia entre direcciones relativas o identificadores y las nuevas localidades físicas.

*El grado de inversión se refiere a la medida en la que los valores de campo y los apuntadores a los registros correspondientes se han colocado en el directorio, en otras palabras, la medida en la que se han establecido rutas de acceso. Entre más alto sea el grado de inversión, mayor será el número de valores de campo que se invertirán. Cero por ciento de inversión es simplemente el método básico de acceso de soporte, ya sea el aleatorio o el secuencial indexado, que tiene una sola llave de acceso. Un 100% de inversión significa que se han invertido cada valor de campo de cada nombre de campo, proporcionando de esta manera acceso muy rápido vía cualquier llave (0 llaves) de acceso del usuario.*

Una de las grandes ventajas de las organizaciones invertidas sobre otras organizaciones es la habilidad para satisfacer varios tipos de consultas que involucran a los valores de campo invertidos, sin tener que recopilar los registros mismos. Consultas del tipo *¿Existen registros que satisfagan los criterios siguientes?* y *¿Cuántos registros existen que satisfagan los criterios siguientes?*, puede satisfacerse rápidamente mediante una búsqueda en el directorio únicamente y realizando un Y o un O lógico de las listas de apuntadores, correspondientes a las llaves de acceso dadas en la consulta.

El proceso invertido de búsqueda en el directorio y de la manipulación de la lista de apuntadores, es obviamente un mecanismo de búsqueda más eficiente que el proceso de multilistas de buscar en el directorio, recopilar el primer registro de la lista, analizar su contenido para obtener el apuntador y verificar si hay información común en la lista, y recopilar el siguiente registro y examinar su contenido y así sucesivamente. Recuerde que el espacio que requiere un apuntador es de unos cuantos bits y es mucho menor que los valores de las llaves. El esfuerzo inicial de construir las estructuras de listas y multilistas no es muy diferente.

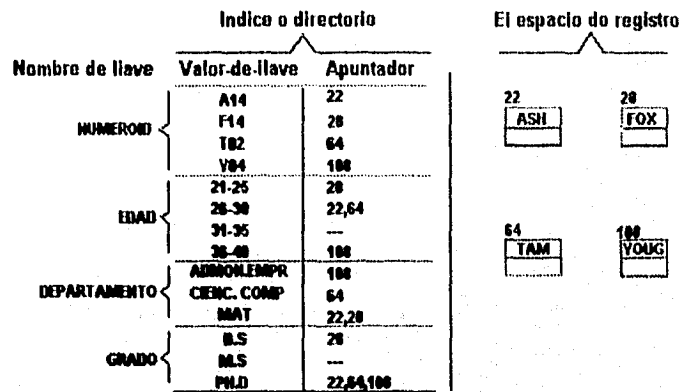
La inserción, eliminación y actualización de registros son más costosos que la recopilación en cualquiera de las organizaciones de indicación primaria o secundaria.

Las siguientes descripciones se refieren específicamente a la organización invertida usando el ejemplo en la figura 1.2.2.C.4

**1. Inserción**

Se almacena el registro y después se toma en cuenta en el directorio cada valor de campo del nuevo registro que corresponda a un campo invertido. Si el valor del campo ya existe como un valor invertido, entonces el identificador podría:

- a) agregarse al final de la lista de apuntadores; o
- b) si los apuntadores están ordenados en secuencia para optimizar la conjunción y disyunción de listas apuntador podría insertarse el identificador en su lugar adecuado después de la búsqueda binaria en la lista (recuerde que cada registro tiene un identificador único). Si el valor del campo no está en el directorio se agrega y se inicia su lista.



Note: No hay apuntadores en los registros mismos: cada valor-de-llave indexado aparece en el directorio.

Figura 1.2.2.C.4 Ejemplo de la organización de archivos invertida.

## 2. Eliminación

Primeramente debe localizarse el registro que va a borrarse según sea cualquier criterio de búsqueda que se haya especificado por ejemplo, "Eliminar todas las personas con grado = bs. ". En este caso, cada apuntador a esos registros también deben eliminarse. Esto requiere nuevamente el directorio y buscar las listas en las que aparecen el identificador o la dirección del registro para eliminarlo también. Si el registro solamente se marco como eliminado, el directorio podría contener apuntadores a un registro lógicamente eliminado y se obtendrían respuestas erróneas a preguntas que involucran normalmente búsqueda en el directorio exclusivamente.

3. Actualización (cambio en el mismo sitio).

Primeramente deberá hacerse el acceso al registro que se actualizará, mediante un criterio de búsqueda, "actualizar departamento = cienc. comp. donde grado = ph.d Y edad > 36", el cual se aplicará cuando todas las personas que poseyeran un grado ph.d. y tuvieran más de 35 años de edad se transfirieran al departamento de ciencias computacionales. Cada registro que satisfaga los criterios de búsqueda debe examinarse para ver su valor en el campo departamento a manera de eliminar su identificador de la lista apuntadora de este valor y agregarlo a la lista de cienc. Comp.

**Indexación de niveles múltiples**

Cuando los niveles de inversión son altos, el directorio es un archivo grande que presenta los mismos problemas para búsquedas que el conjunto original de registros. Entonces, ¿por qué no invertir al directorio mismo lo que resultará en un segundo directorio e invertir el segundo directorio lo que resultará en un tercer directorio; y así sucesivamente de ser posible? En otras palabras, la estrategia es una organización indexada de niveles múltiples, como se ejemplifica en la figura 1.2.2.C.5. La mayoría de las implementaciones de archivos secuenciales indexados (ISAM) usan internamente una organización similar de indexación en niveles múltiples, que se usa para mejorar significativamente la rapidez para el acceso por llave (el nombre de la llave de acceso es el que se elige a priori cuando se crea el archivo) ISAM puede considerarse como un sistema con inversión de una llave Selección del índice invertido El problema de qué campo invertir es un tema crucial y difícil para los diseñadores.

Anteriormente todos los bancos de datos eran completamente invertidos y no se permitía inversión parcial alguna. Actualmente todos los bancos de datos comerciales permiten a los usuarios especificar cualquier nivel de inversión que



se desee. Esta flexibilidad es una necesidad en la práctica real. Entre más rutas de acceso se establezcan para optimizar la recopilación, generalmente es más costoso actualizar, insertar y eliminar. Así los campos que rara vez se usan como llaves de recopilación no deberán indexarse. En la práctica actual la selección de la inversión inicial es a lo más rudimentaria; la selección de los campos invertidos se mejora sólo después que se han hecho costosas pruebas por tanteo al tiempo del funcionamiento. El cambiar el grado de inversión puede resultar bastante caro, de hecho en muchos sistemas comerciales puede hacerse sólo mediante una costosa recarga del banco de datos completo. Son pocos los autores que han sugerido guías prácticas y específicas para seleccionar la inversión de campos óptima o casi óptima.

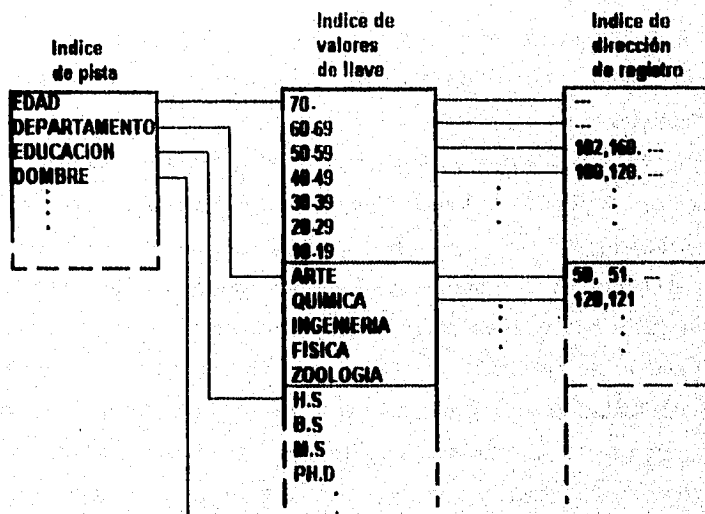


Figura 1.2.2.C.5. Un índice triple, de la organización de archivos invertida

**Indexación de atributos múltiples.**

La Indexación de atributos múltiples se refiere a la Indexación sobre varios campos combinados, proporcionando así rutas de acceso de resolución superior. Se escucha con frecuencia varios términos confusos referentes a esta práctica. Para ilustrar la Indexación de atributos múltiples considérese consultas que hagan uso de las llaves de acceso K1, K2 ; K3. En algunos casos específicos el acceso puede resultar más rápido al invertirse sobre las tres llaves combinadas (por ejemplo, una sola lista que apunte a los registros con los tres atributos K1, K2, K3), mientras que en otros casos podrá incrementarse la rapidez invirtiendo sobre cada una de las tres llaves individualmente, es decir, teniendo tres listas. Las cosas se complican cuando se considera la actualización.

No se ha reportado en la práctica real el uso de Indexación de atributos múltiples. La razón es, en resumen, que el rendimiento en cuanto al tiempo es muy sensible a cambios en el contenido del banco y en el tipo de complejidad lógica de las consultas.

Los sistemas de atributos múltiples pueden afinarse muy bien pero solamente en medios ambientes muy estáticos. Aún en los sistemas de recopilación manuales o automatizados para bibliotecas, que son relativamente muy estáticos (ya que el banco de datos, o sean los documentos y los términos de Indexación seleccionados para referirse a ellos no cambian con el tiempo), no se ha usado mucho la Indexación de atributos múltiples; la entrada clásica vía la tarjeta de catálogo o el índice es el tipo de llave sencilla: por autor o por título del documento.

**Organización híbrida multilista-invertida.**

Considere la organización de multilistas básica de la figura 1.2.2.C.2. Una de las maneras de reducir el problema de tener que buscar innecesariamente en listas largas es limitar su longitud. Los dispositivos de almacén externo y los programas (software) del sistema están orientados al uso de casillas o páginas, de manera que por lo general es menos costoso buscar una lista restringida a una página o celda (que puede obtenerse tal vez en una sola operación de entrada/salida) que en una lista de la misma longitud, pero que se extiende sobre varias páginas.

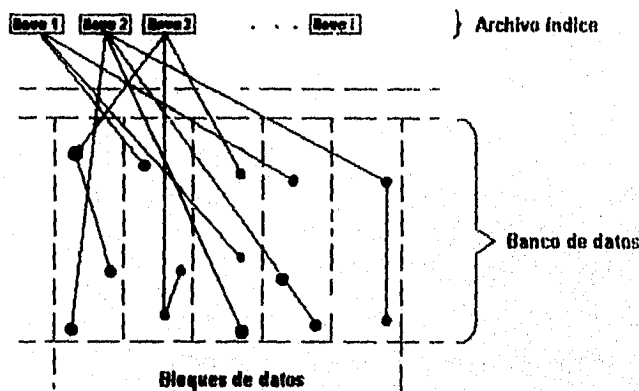


Figura 1.2.2.C.6. Organización de multilista celular básica.

Esto conduce a la estructura llamada con frecuencia multilista celular o multilista invertida que se muestra en la figura 1.2.2.C.6. En este caso el directorio contiene una entrada, el apuntador al primer registro de cada sublistas, donde cada sublistas está restringida a una página o celda. Existen aún apuntadores dentro de los registros excepto por el último de la sublistas. Si la longitud de cada sublistas se limita aún más hasta la de un registro, resulta entonces la organización de archivo invertida, como en la figura 1.2.2.C.7. La multilista celular es un enfoque híbrido de las organizaciones de multilista e invertida.

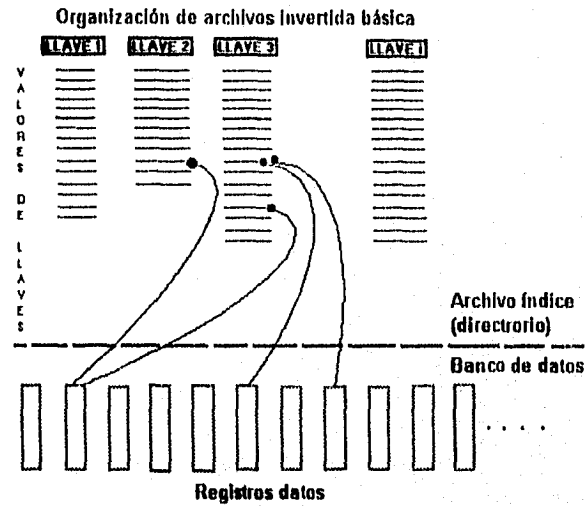


Figura 1.2.2.C.7. Organización de archivos invertida básica

**Organización de datos anillada y arbórea encadenada.**

Una organización de datos anillada es simplemente una organización de listas en la cual el último registro apunta de regreso al primero de la lista, como se muestra en la figura 1.2.2.C.8. En una organización de anillos múltiples (o multianillada), llamada también simplemente organización anillada, pasan anillos múltiples a través de los registros. Hay apuntadores en cada registro. Como ejemplo, vea la figura 1.2.2.C.2. donde el apuntador nulo de cada registro final ahora apunta a la cabeza de la lista. La cabeza de cada anillo es la llave de entrada a los datos almacenados. cada anillo es una ruta de acceso.

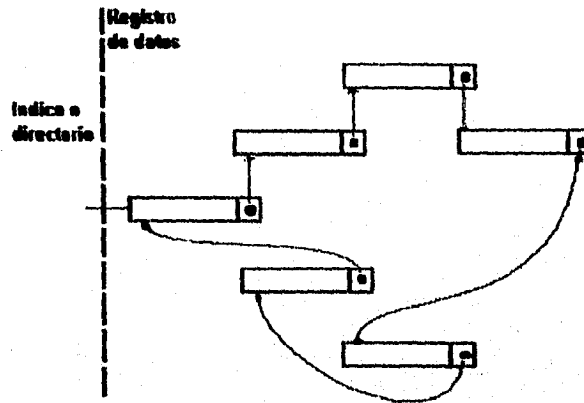


Figura 1.2.2.C.8. Organización de anillo.

Pueden usarse también listas o anillos hacia atrás como se muestra en la figura 1.2.2.C.9. Una de las justificaciones para este gasto adicional de almacén es el ahorro de tiempo de acceso. Si se proporcionan apuntadores hacia atrás, es fácil encontrar registros que anteceden; por el contrario, si únicamente se tienen apuntadores hacia adelante, el resto del anillo tendrá que recorrerse hacia adelante desde la posición original, para llegar a los registros anteriores. También podrían almacenarse apuntadores en cada registro, o cada cierto número de registros, para indicar el inicio del anillo, mejorando así la habilidad para navegar en el espacio de registros. Esta navegación y el software que le da soporte se hacen más intrincados conforme aumenta el número de anillos o rutas de acceso.

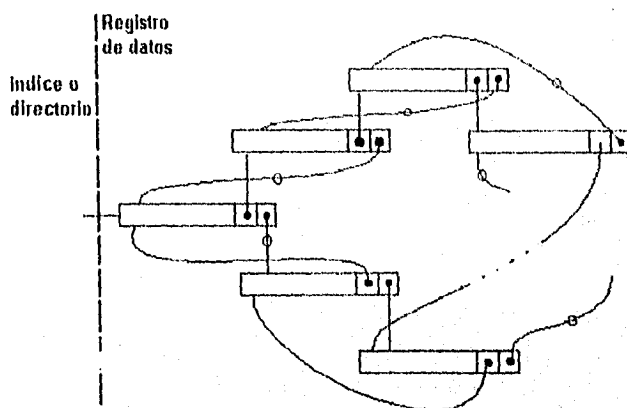


Figura 1.2.2.C.9. Organización de anillos múltiples con apuntadores hacia atrás y hacia adelante.

La organización de datos básica arbórea doblemente encadenada, se ilustra en la figura 1.2.2.C.10. Se han usado otros sinónimos para ella como "árbol enhebrado (sin y con apuntadores de línea punteada)" y "árbol triplemente encadenado". En la figura 1.2.2.C.10, cada nivel representa un nombre de llave y cada nodo un valor de llave. Las hojas del árbol son los registros de datos.

Por lo general tres apuntadores están asociados con cada nodo. El apuntador F apunta a un conjunto de valores de llave en el siguiente nivel inferior, contenidos en aquellos registros que tienen el valor de llave que denota el nodo. A este conjunto de valores de llaves, generalmente se le llama conjunto filial. El apuntador C apunta horizontalmente al siguiente valor de llave en el conjunto filial.

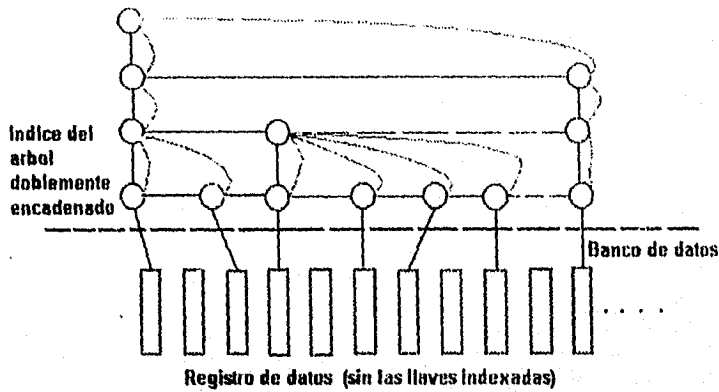


Figura 1.2.2.C.10. Organización de archivo arbórea doblemente encadenada básica

También puede usarse para atravesar horizontalmente de un conjunto filial a otro en el caso especial de que esté contenido en el último nodo de un conjunto filial. El apuntador P apunta a el valor de llave en el nivel superior, que sea el padre del conjunto filial del cual el nodo es miembro.

Fisicamente la organización contiene dos tipos de bloques: 1) bloque índice que contiene valores de llaves y apuntadores asociados y 2) bloques de datos que contienen los registros sin esos valores de llaves. Los bloques índice contienen los valores de llaves y los apuntadores que se han seleccionado para mejorar el acceso a los registros. Solamente aquellas llaves que aparecen con frecuencia como llaves de acceso en las consultas deberán indexarse o encadenarse doblemente. Las llaves doblemente encadenadas son análogas a las llaves invertidas en la organización de archivos invertida. Para lograr encadenamiento doblemente completo, todos los valores de llave se colocan en el índice, por lo que cada registro se convierte precisamente en una dirección. En el directorio doblemente encadenado se proporcionan suficientes apuntadores para construir

el registro original desde cualquier punto del directorio. Aunque las representaciones arbóreas para datos se han usado desde hace largo tiempo en numerosas aplicaciones, el enfoque doblemente encadenado como una organización de alto nivel para archivos no ha sido considerado o ejercitado lo suficiente.

El ejemplo ilustrativo de la figura 1.2.2.C.11 muestra en la parte (a) los valores de los datos actuales de un subconjunto de 1) nombres de llaves de un conjunto de registros, o 2) dominio de tuplas de una relación; en la parte (b) su organización doblemente encadenada; y en la parte (c) su estructura física real. La figura 1.2.2.C.11. (c) es una de las formas físicas posibles en los dispositivos de almacén actuales de una dimensión.

Observe que el apuntador F [apuntador hacia abajo en la figura 1.2.2.C.11. (b) ] se logra con contigüidad Física.

**Organización jerárquica o completamente anillada.**

Si un conjunto de registros de datos se fragmenta de manera que todos sus datos elementales existan por sí solos, pero conectados a otros por un gran número de anillos, el resultado es una estructura de multianillos o completamente anillada.





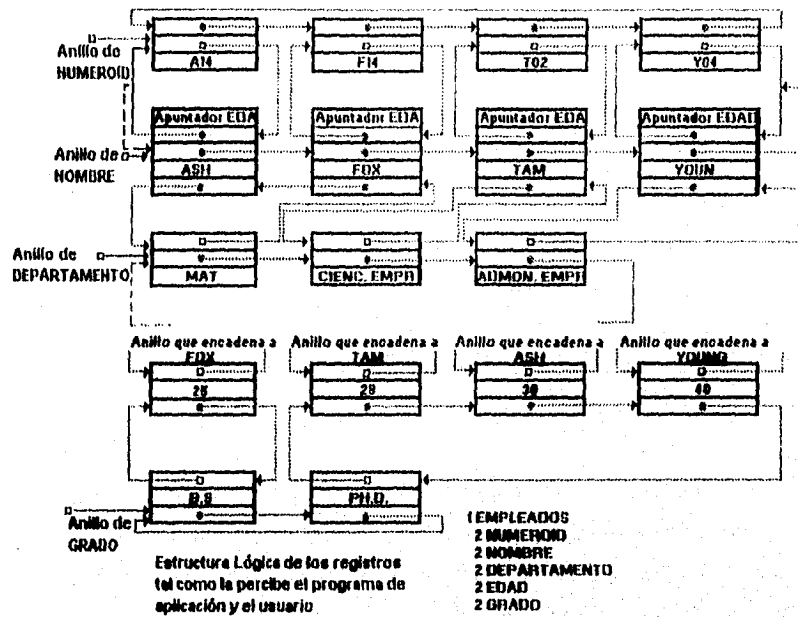


Figura 1.2.2.C.12. Ejemplo de la organización completamente anillada.

El lector deberá recorrer la jerarquía de la figura 1.2.2.C.12. hacia arriba y hacia abajo y comprender cómo se busca en la estructura, observe cómo se ha presentado realmente el contenido lógico de cada registro vía los multianillos, aunque esto no sea aparente inicialmente. Los puntos de entrada a la estructura son los apuntadores iniciales de los anillos número\_id, nombre, departamento y grado. Los puntos de entrada deben establecerse de acuerdo a los requisitos de acceso de las consultas.

Suponga que requiere listar toda la información de empleados en los que " grado = b.s ", el proceso para satisfacer la consulta sería:

1. Entrar al anillo de grado y detenerse en b.s.
2. Ir a la jerarquía edad; dice edad = 25
3. Ir a la jerarquía nombre; dice nombre = fox.
4. Ir a la jerarquía departamento ; dice departamento = mat
5. Ir a la jerarquía número\_id ; dice número\_id = f14

Así la satisfacción de la consulta consiste de una sola persona que califica:

f14, fox, mat, 25, b.s

La estructura de multianillo no es la misma que la doblemente encadenada. Ambas usan un nivel del árbol o jerarquía para un nombre de llave dado y apuntadores hacia adelante y al siguiente nivel. Esto las hace aparecer como similares. Observe que un nivel dado un valor de llave puede repetirse en el árbol doblemente encadenado, pero no en la estructura anillada. Otra diferencia es que la organización de multianillo implica anillos en el espacio de registros, mientras que la organización arbórea doblemente encadenada implica cadenas (parecidas a anillos) en el índice. La estructura de multianillos es más complicada y su programación presenta una considerable carga de trabajo para manejo de los apuntadores, particularmente para actualizaciones, inserciones y eliminaciones.

**D) MINIESPECIFICACIONES**

Existe una variedad de herramientas que podemos utilizar para producir una especificación de proceso. Tablas de decisión, lenguaje estructurado, pre/post condiciones, diagramas etc.

Se puede usar cualquier método mientras satisfaga dos requerimientos cruciales:

- La especificación del proceso debe expresarse de una manera que pueda ser verificada tanto por el usuario como por el analista.
- El proceso debe especificarse de forma que puede ser comunicado efectivamente al público amplio que esté involucrado.

Las pre/pos condiciones son una manera conveniente de describir la función que debe realizar el proceso, sin decir mucho acerca del algoritmo o procedimiento que se utilizará.

Resulta ser un enfoque particularmente útil cuando:

- 1) El usuario tiene tendencia a expresar la política llevada a cabo por la burbuja en términos de un algoritmo particular que ha estado utilizando durante décadas.
- 2) El analista está razonablemente seguro de que existen muchos algoritmos distintos que podrían usarse.
- 3) El analista desea que el programador explore varios de estos algoritmos pero no quiere involucrarse personalmente con tales detalles y, sobre todo, no quiere enredarse en discusiones con el usuario acerca del mérito relativo de cada uno.

Un ejemplo de una especificación de proceso escrita con el enfoque de la pre/post condición se muestra a continuación:

**Especificación de proceso 3X.5X Calcular el impuesto sobre ventas**

**Precondición 1**

Ocurre Datos-Venta con Tipo-Item que corresponde con Categoría-Item en Categorías-Impuesto.

**Postcondición 1**

Impuesto-Sobre-Venta se hace igual a Monto-venta Impuesto.

**Precondición 2**

Ocurre Datos-Venta con Tipo-Item que no concuerda con Categoría-Item en Categorías-Impuesto.

**Postcondición 2**

Se genera Mensaje-Error

Como puede verse, existen dos partes del proceso: precondiciones y postcondiciones.

Las precondiciones describen todas las cosas que deben darse antes de que el proceso pueda comenzar a ejecutarse.

Puede imaginar a las precondiciones como una garantía del usuario. "Garantizo que cuando se active este proceso se cumplirán las siguientes cosas".

Típicamente, las precondiciones describirán lo siguiente:

- Qué entradas se encuentran disponibles. Estas entradas llegan mediante un flujo conectado con un proceso, como se muestra en el DFD. Nótese que puede haber casos en que los diversos flujos entran a un proceso, pero sólo uno de ellos es precondición necesaria para que se active el proceso. Por ejemplo, si hubiera una especificación que empieza con:

**Precondición**

ocurre el dato X

asociada con el DFD que se muestra en la figura 1.2.2.D.1, se interpretaría de la siguiente forma:

la llegada del estado X es el estímulo activador que hace que el proceso empiece a trabajar. Como parte de su trabajo, busca entradas de los flujos Y o Z, o ambos, pero Y y Z no son necesarios para que el proceso comience su trabajo.

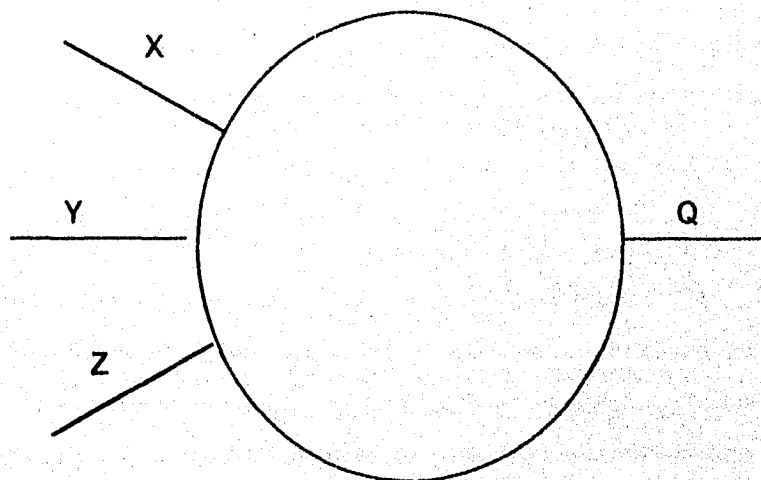


Figura 1.2.2.D.1. DFD con entradas X, Y y Z

- ¿Que relación debe existir entre las entradas?. Muy a menudo una precondición especificará que deben llegar dos entradas con campos que correspondan (por ejemplo, detalles de pedidos y detalles de envío y, concretamente, las ganancias derivadas de este tipo de operaciones, sin que las con el mismo número de cuenta). O bien la precondición puede especificar que una componente de un lado de entrada debe estar dentro de cierto intervalo (por ejemplo, "pedido con fecha de entrega a más de 60 días").

2.2.1 Ley del Impuesto Sobre la Renta

- ¿Que relaciones deben existir entre entradas y almacenes de datos?. Una precondición pudiera estipular que "existe un pedido en el almacén de pedidos cuyo número-de-cuenta-de-cliente corresponde con el número-de-cuenta-de-cliente del almacén de clientes". O bien, la precondición podría establecer que "hay un pedido en el almacén de pedidos cuyo número-de-cuenta-de-cliente corresponde con un número-de-cuenta-de-cliente que corresponde con un número-de-clientes de otros bienes muebles e inmuebles que están sujetos a la adquisición de las acciones".

En el régimen fiscal de enajenación de acciones no tiene gran importancia el valor nominal de los títulos, porque generalmente no es significativo para obtener el valor de adquisición de la participación por el comprador. El régimen de enajenación de acciones no tiene gran importancia el valor nominal de los títulos, porque generalmente no es significativo para obtener el valor de adquisición de la participación por el comprador.

Es decir, la precondición podría establecer que "hay un pedido en el almacén de pedidos cuyo número-de-cuenta-de-cliente corresponde con el número-de-cuenta-del-cliente en el almacén de clientes". O bien, la precondición podría establecer que "existe un pedido en el almacén de pedidos con fecha-de-envío igual a la fecha actual".

De manera similar, las postcondiciones describen lo que debe darse cuando el proceso ha concluido. Nuevamente, esto puede imaginarse como una garantía: ordena que los hechos que se describen en la postcondición se cumplan al momento de la enajenación de acciones.

En los casos en que se haga referencia a acciones se entenderán incluidos los certificados de participación patrimonial emitidos por las sociedades nacionales de crédito, las partes sociales, las participaciones en asociaciones civiles y los certificados de participación social, las participaciones en sociedades nacionales de crédito, las partes sociales, las participaciones en asociaciones civiles y los certificados de participación social, las participaciones en sociedades nacionales de crédito, las partes sociales, las participaciones en asociaciones civiles y los certificados de participación social.

En los casos en que se haga referencia a acciones se entenderán incluidos los certificados de participación patrimonial emitidos por las sociedades nacionales de crédito, las partes sociales, las participaciones en asociaciones civiles y los certificados de participación social, las participaciones en sociedades nacionales de crédito, las partes sociales, las participaciones en asociaciones civiles y los certificados de participación social.

Las postcondiciones típicamente describen lo siguiente:

- Las salidas que generará o producirá el proceso. Ésta es la forma más común de postcondición (por ejemplo, "se producirá una factura").
- Las relaciones que existirán entre los valores de salida y los valores originales de entrada. Esto es común para la situación donde una salida es una función matemática directa de un valor de entrada. De ésta forma una postcondición pudiera afirmar que "la factura-total se calcula como la suma de precios unitarios-de-artículos más costos-de-envío".
- Las relaciones que existirán entre valores de salida y los valores en uno o varios de los almacenes. Esto es común cuando la información debe recuperarse de un almacén y utilizarse como parte de la salida de un proceso.
- Por ejemplo una especificación de proceso pudiera tener como postcondición la siguiente afirmación: "el balance-actual en el almacén INVENTARIO se incrementará con cantidad-recibida, y el nuevo balance-actual se producirá como salida de este proceso."
- Los cambios que se hayan dado en los almacenes: nuevos artículos añadidos, artículos existentes que se hayan modificado, o artículos existentes que se hayan eliminado. Así pudieran darse afirmaciones tales como "el pedido se anexará al almacén de PEDIDOS", o "el registro de clientes se eliminará del almacén de CLIENTES".



## Capítulo I

---

Cuando se esté construyendo una especificación de pre/post condiciones, se debe comenzar por describir las situaciones normales de proceso. Pudieran existir diversas situaciones normales diferentes (por ejemplo, combinaciones únicas de relaciones de entrada/almacén válidas), cada una de las cuales se expresa como precondición distingible e individual. Para cada una de estas precondiciones se debe describir la condición de la burbuja del proceso cuando se han producido las salidas y se han modificado los almacenes. Después de haber descrito las situaciones formales del proceso, deben incluirse precondiciones y postcondiciones apropiadas para los casos de error y casos anormales.

Aunque el enfoque de pre/post condiciones sea bastante útil y tenga un gran número de ventajas. Hay ocasiones en las cuales puede no ser apropiado. La falta de pasos intermedios entre entradas (precondiciones) y salidas (postcondiciones) es deliberada y consciente, pero puede volverse difícil de entender si el lector no visualiza algún tipo de procedimiento que lleve de las entradas a las salidas, podía ser más fácil escribir una especificación utilizando lenguaje estructurado. Un ejemplo de especificación de pre/postcondición que probablemente sea demasiado compleja se muestra a continuación:

**Determinar tasa de préstamo según factores de compradores**

**Precondición 1**

Ocurre una solicitud-de-préstamo  
y antigüedad > 5 o valor-neto > monto-del-préstamo  
y gastos-mensuales <  $0.25 * \text{monto-del-préstamo}$  o  
garantía-colateral >  $2 * \text{monto-del-préstamo}$   
y edad > 25 o  
garantía-colateral > monto-del-préstamo  
y edad > 30 o  
antigüedad > 2  
y valor-neto >  $2 * \text{monto-del-préstamo}$   
y edad > 21  
y gastos-mensuales <  $0.5 * \text{monto-del-préstamo}$

**Postcondición 1**

monto-aprobado = monto del préstamo

Como con todas las formas de especificación de procesos, permite que su propio juicio y las reacciones del usuario lo guíen; si el usuario encuentra la especificación de pre/postcondición demasiado difícil de leer, escoja otro formato.

### 1.3 CARACTERÍSTICAS DEL MANEJADOR DE BASES DE DATOS ACCESS

Microsoft Access, puede ordenar rápidamente los registros de tablas ,consultas y formularios, en forma ascendente o descendente, puede ajustar rápidamente el ancho de una columna de la hoja de datos para que se adapte a su valor más ancho, Además de poder imprimir información detallada sobre el diseño de cualquier objeto de la base de datos.

Las bases de datos en Microsoft Access están constituidas por diversos objetos , tales como: tablas, consultas, formularios, informes, macros y módulos. Éstos objetos se listan en la ventana de la base de datos ,donde se crean , abren y utilizan.

#### • DISEÑO DE UNA BASE DE DATOS

Para poder obtener el máximo de 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 la base de datos estriba en tener bases sólidas acerca de la organización de las Bases de datos.

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

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

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 en la base de datos debe incluir un campo o conjunto de campos que identifiquen únicamente cada renglón o registro almacenado en la tabla.

- **CARACTERÍSTICAS DE LAS TABLAS**

Antes de crear cualquier otro objeto de una base de datos como lo son: 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 los nombres de varias categorías de productos, descripción de la categoría ,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 puede haber muchas tablas, cada una de las cuales almacena información sobre un tema distinto. Cada tabla puede contener datos de muchos tipos, incluyendo texto, números, fechas e imágenes.

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 en la base de datos debe incluir un campo o conjunto de campos que identifiquen únicamente cada renglón o registro almacenado en la tabla.

- **CARACTERÍSTICAS DE LAS TABLAS**

Antes de crear cualquier otro objeto de una base de datos como lo son: 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 los nombres de varias categorías de productos, descripción de la categoría ,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 puede haber muchas tablas, cada una de las cuales almacena información sobre un tema distinto. Cada tabla puede contener datos de muchos tipos, incluyendo texto, números, fechas e imágenes.

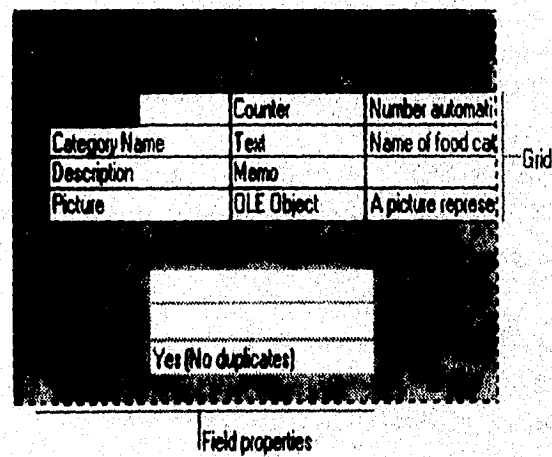
Con MS Access, se puede usar la ventana tabla para crear y ver las tablas. Puede abrir la ventana tabla en cualquiera de las presentaciones existentes: Diseño y Hoja de datos.

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.

Las reglas de validación que se establezcan para las tablas y sus campos estarán siempre en vigor, tanto si cambia los datos en un formulario, como si importa nuevos datos o utiliza una consulta de acciones, una macro o código de Access Basic para cambiar datos. También puede establecer una propiedad de campo para solicitar la introducción de datos en un campo.

### Table Window - Design View

See Also



Use a table's Design view to create and modify the structure of a table.

Figura 1.3.1 Ventana Tabla en Modo Presentación Diseño.

## Capítulo 1

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 incorrecta introducción, especificar valores predeterminados y acelerar la búsqueda y la ordenación en la tabla.

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

Microsoft Access, cuenta con el asistente para tablas, en donde puede elegir entre docenas de ejemplos: tablas de ejemplos y cientos de campos prácticos así como dejar que el asistente cree las tablas automáticamente.

Microsoft Access, puede filtrar los registros de una tabla para ver únicamente los registros deseados. Además puede presentar y editar simultáneamente todos los índices de una tabla, en la ventana índices.

### • CARACTERÍSTICAS DE LAS RELACIONES ENTRE TABLAS

Cuando se crea una tabla, ésta no está relacionada 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 desee 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 de la base de datos.

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

• **CARACTERÍSTICAS EN CONSULTAS**

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

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

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

Cuando una consulta incluye campos de varias tablas relacionadas, se pueden actualizar los datos a ambos lados de la combinación. Puede definir una consulta de selección para limitar el número de registros devueltos.

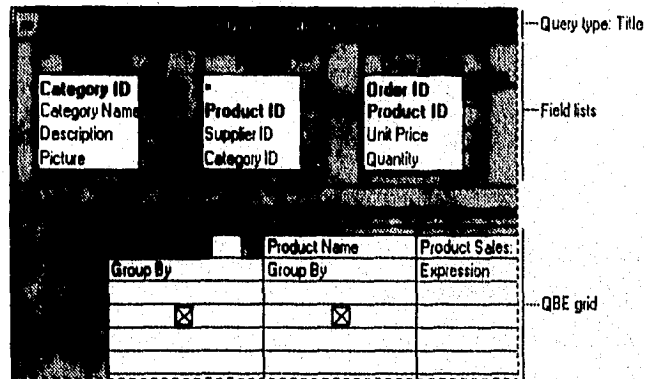


• MODOS DE PRESENTACIÓN DE LAS CONSULTAS

Utilice el modo de presentación Diseño para crear una nueva consulta o modificar el diseño de una ya existente. En el modo presentación diseño podrá emplear herramientas gráficas para la creación de consultas.

Query Window – Design View

See Also



Use a query's Design view to create and modify the structure of a query.

Figura 1.3.2 Ventana Consulta en modo de Presentacion Diseño

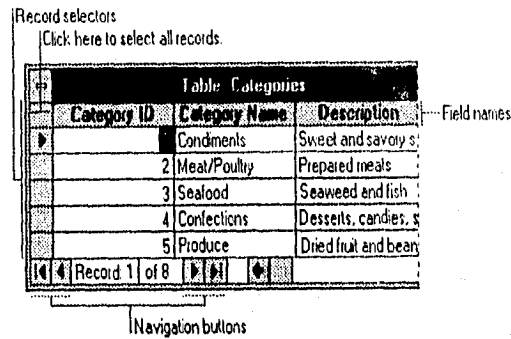
Utilice el modo de presentación Hoja de Datos para mostrar los datos recuperados por la consulta.

Utilice el modo de presentación preliminar para ver la apariencia que tendrán los datos de la consulta una vez impresos.

Utilice el modo de presentación SQL cuando desee crear o modificar una consulta mediante inscripciones SQL.

## Datasheet View

See Also

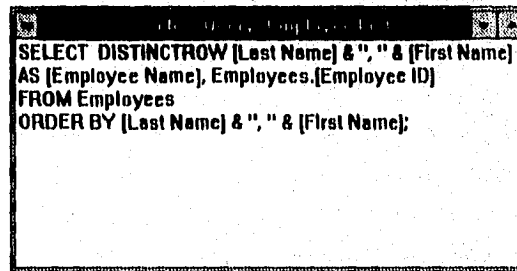


Datasheet view displays multiple records in a row-and-column format, enabling you to view many records at one time. You can also use a datasheet to add and edit data, although a form usually provides more flexibility for these operations.

Figura 1.3.3 Ventana Consulta, Modo de Presentación Hoja de Datos.

## Query Window - SQL View

See Also



The SQL view displays the SQL statement for the current query in the Query window. When you create a query (either directly in the Query window or by using one of the Query Wizards), Microsoft Access constructs the SQL equivalent in the SQL view of the Query window.

Figura 1.3.4 Ventana Consulta, Modo de Presentación SQL

Microsoft Access dentro de la ventana de presentación SQL puede introducir cualquier instrucción del SQL. Mientras edita la instrucción del SQL también dispone de acceso a los comandos del menú y a otras ventanas de MS Access.

- **TIPOS DE CONSULTA.**

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

**Selección:** Son los 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.

**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 reducir 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.

**Acción:** Modifican muchos registros en una sola operación. Se utiliza para crear una nueva tabla, eliminar registros de una tabla, añadir nuevos registros a una tabla o modificarlos. A partir de la selección actualizan, agregan y eliminan datos, Access provee cuatro tipos de consultas de este tipo que son:

- consultas de creación de tablas
- consultas de eliminación
- consulta de datos añadidos
- consulta de actualización.

**Porque se utilizan consultas?**

Microsoft Access ofrece una gran flexibilidad, a la hora de diseñar sus consultas podrá:

**Elegir campos:** no es necesario incluir en la consulta todos los campos de una tabla.

**Elegir registros:** puede especificar los criterios que deben cumplir los registros para que se incluyan en la hoja de respuesta dinámica de la consulta.

**Ordenar registros:** puede ver los registros en un orden determinado. Para mostrar los datos ordenados en un formulario, 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:** puede utilizar una consulta para responder a una pregunta sobre datos procedentes de mas de una tabla y ver los resultados en una sola hoja de datos.

**Realizar cálculos:** puede crear nuevos campos llamados campos calculados, que contengan el resultados de un calculo. Para mostrar campos calculados en formularios o informes, 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 formularios, informes y otras consultas:** Para seleccionar solo los datos que deben aparecer en un formulario o informe, puede crear una consulta de selección y utilizarla como origen de los datos del formulario o informe. Utilizando la consulta para incluir datos de varias tablas y establecer criterios para mostrar solo un conjunto limitado de datos. Cada vez que abra el formulario o imprima el informe, la

consulta recuperará información actualizada de las tablas. También puede introducir nuevos datos o modificar los existentes directamente en la hoja de datos de la consulta, o bien en un formulario que este basado en esta consulta.

**Modificar los datos de las tablas:** con las consultas de acciones, puede actualizar, eliminar o añadir un grupo de registros simultáneamente. También puede utilizarlas para crear una nueva tabla que contenga registros de otra tabla o de un grupo de tablas existentes.

#### DESVENTAJAS

1.- **NO ES ORIENTADO A OBJETOS.**

El ambiente de desarrollo de Microsoft Access no es orientado a objetos, lo que repercute en mayor inversión de tiempo para el desarrollo, duplicidad de código y que se traslapen los esfuerzos de desarrollo.

2.- **NO MANEJA OBJETOS O LIBRERIAS COMPARTIDAS.**

Los objetos compartidos entre aplicaciones son almacenados en módulos que tienen que ser llamados o cargados desde cada aplicación, por lo que se pierden los beneficios de un solo punto de control.

3.- **USO DE FUNCIONES NO ESTANDARIZADAS.**

El soporte SQL esta disponible únicamente a través del diálogo del query, los controladores ODBC no proveen el performance (desempeño) que proporcionan los controladores nativos, por lo que no se pueden explotar todas las características únicas de los servidores de bases de datos. (Entre los que destacan: vistas, procedimientos interconstruidos, sinónimos y números de secuencia).

**4.- VULNERABILIDAD DE LAS BASES DE DATOS E ÍNDICES.**

Las bases de datos e índices de access, son susceptibles de corromperse debido a desconexiones de la red o problemas del servidor, además de que las operaciones necesarias para la reparar las bases de datos y reconstruir los índices son extremadamente lentas.

#### **1.4 CARACTERÍSTICAS DE VISUAL BASIC 3.0 PROFESIONAL**

La tendencia dominante de hoy día es las herramientas de desarrollo de aplicaciones generadas mediante lo que se denomina programación visual. Por programación visual, se entiende el uso de expresiones visuales (tales como gráficas, dibujos, iconos, barras de menús ,etc. ) en el proceso de la programación de aplicaciones. Aunque existe una gran variedad de lenguajes visuales uno de los que han destacado sobre todo para el desarrollo de aplicaciones 'front-end' es el Microsoft Visual Basic.

Microsoft liberó el Visual Basic 1.0 en 1987 y fue el primer lenguaje visual desarrollado por esta compañía. Visual Basic fue un éxito y poco después salió al mercado la versión 2.0 (1991), sin embargo el verdadero Potencial de esta herramienta fue vista por los usuarios hasta la versión 3.0. Algunas de las características de esta herramienta se enlista a continuación:

PASOS PARA LA GENERACIÓN DE APLICACIONES EN VISUAL BASIC V.3.0

Los pasos recomendables para la creación de una aplicación en Visual Basic son:

- 1.- Dibuje o aplique los objetos que utilizara en su Interfaz de usuario. Esta parte se convierte en un problema de diseño gráfico, y consiste en colocar los botones, textos, gráficas, celdas, etc. necesarios que utilizará nuestra aplicación mediante el 'Visual Basic Toolbox'

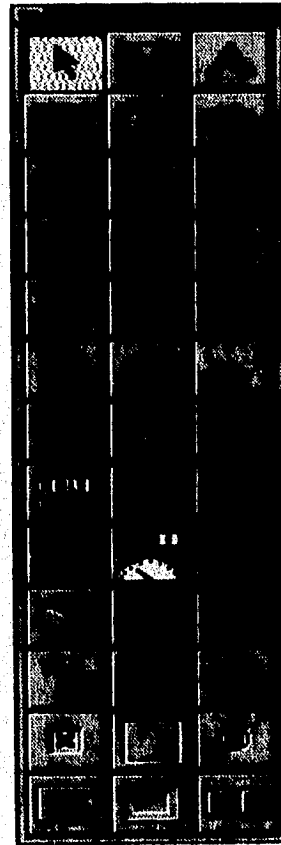


Fig. 1.4.1 Visual Basic  
Toolbox



Descripción de cada control( de izquierda a derecha):

<b>Pointer.</b>	Se utiliza para cambiar el tamaño de un control, moverlo y seleccionar conjunto de controles.
<b>Picture Box.</b>	Despliega imágenes con formato BMP ó PCX.
<b>Label.</b>	Sirve para colocar letreros que no requieran ser cambiados.
<b>Text Box.</b>	Se utiliza para desplegar o capturar texto.
<b>Frame</b>	Se utiliza para crear gráficas o agrupar conjuntos de controles dentro de este control.
<b>Command Button.</b>	Crea un botón que controla alguna acción.
<b>Check Box.</b>	Se utiliza para crear cajas de selección de opciones, en la cual se necesita seleccionar múltiples opciones.
<b>Option Button.</b>	Se utiliza para crear grupos de opciones en la cual se necesita seleccionar una y solo una de ellas.
<b>Combo Box.</b>	Es una combinación de un 'list box' y un 'text box'. El usuario puede realizar una selección de una lista o capturar un texto.
<b>List Box.</b>	Se utiliza para desplegar una lista de artículos en la cual el usuario puede seleccionar uno.
<b>Horizontal Scroll Bar,</b> <b>Vertical Scroll Bar.</b>	Se utilizan para navegar a través de una lista muy larga de información que no puede ser desplegada en una sola pantalla.
<b>Timer.</b>	Se utiliza en aplicaciones que requieran eventos de tiempo.
<b>Drive List Box.</b>	Se utiliza para desplegar las unidades de disco válidas.
<b>Directory List Box.</b>	Despliega el directorio de alguna unidad

	seleccionada, incluyendo sus subdirectorios.
<b>File List Box</b>	Despliega la lista de archivos que pueden ser cargados o salvados, de acuerdo a un conjunto de opciones.
<b>Shape.</b>	Se utiliza para dibujar rectángulos, triángulos o círculos.
<b>Line.</b>	Se utiliza para dibujar líneas con ciertas variedades de estilo.
<b>Image.</b>	Despliega imágenes adicionando cierto control sobre ella.
<b>Data Control.</b>	Permite el uso de bases de datos en la aplicación.
<b>Grid.</b>	Se utiliza para crear hojas de trabajo, en la cual los datos se almacenan en celdas ordenados por filas y columnas.
<b>OLE Control.</b>	Permite el uso de OLE (object linking and embedding) en la aplicación.
<b>Animated Button Control.</b>	Permite el control de iconos o gráficas para realizar un efecto de animación en un botón.
<b>Common Dialog.</b>	Se utiliza para utilizar los diálogos utilizados en windows para por ejemplo seleccionar un font o cambiar colores.
<b>Crystal Custom Control.</b>	Utilería que permite la generación de reportes en una base de datos.
<b>Gauge Control.</b>	Se utiliza para mostrar el avance de un proceso, mediante una figura animada.
<b>Graph Control.</b>	Permite el diseño de gráficas interactivas con nuestras aplicaciones.
<b>Key Status Control.</b>	Se utiliza para desplegar o modificar el estado de las teclas CAPS LOCK, INS, SCROLL LOCK, etc.

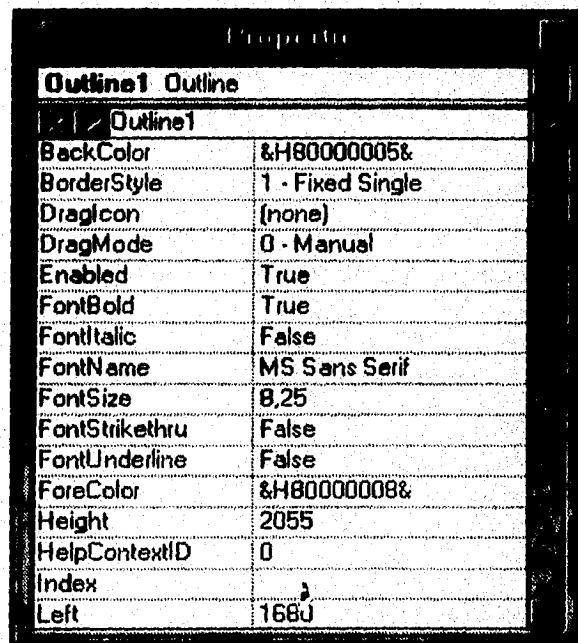
<b>Communications Control.</b>	Provee comunicación por puerto serial.
<b>Masked Edit.</b>	Se utiliza para restringir la captura de datos de acuerdo a un formato seleccionado por el usuario.
<b>Outline Custom Control.</b>	Permite la creación de listas jerárquicas.
<b>Picture Clip Control.</b>	Permite identificar áreas en gráficas de bits.
<b>Spin Button Control.</b>	Es un control de avance el cual permite incrementar o decrementar un valor numérico.
<b>3D check Box Control.</b>	Similar a 'Check box' pero con opciones para dar efecto de tres dimensiones.
<b>3D Frame Control.</b>	Similar a 'Frame Control' pero con opciones para dar efecto de tres dimensiones.
<b>3D Option Button Control.</b>	Similar a 'Option Button Control' pero con opciones para dar efecto de tres dimensiones.
<b>3D Command Button Control.</b>	Similar a 'Command Button Control' pero con opciones para dar efecto de tres dimensiones e inserción de gráficas.
<b>3D Panel Control.</b>	Se utiliza para desplegar texto y grupos de otros controles en tres dimensiones.
<b>3D Group Push Button Control.</b>	Se utiliza para generar grupos de botones que emulan la funcionalidad de una barra de herramientas.

2.- Asignación de nombre a cada uno de los objetos.

Consiste en dar un nombre a los objetos utilizados. Mediante este nombre se hará referencia a cada objeto desde un programa.

3.- Selección de propiedades de cada objeto

Cada objeto (forma o control) en Visual Basic posee un conjunto de propiedades predeterminadas. Estas propiedades determinan por ejemplo, la apariencia del objeto, color, tipo de fuente, Nombre, etc. Para modificar el conjunto de propiedades de un objeto puede realizarse desde el diseño del programa aunque también puede realizarse desde la ejecución (run time) a través de un código.



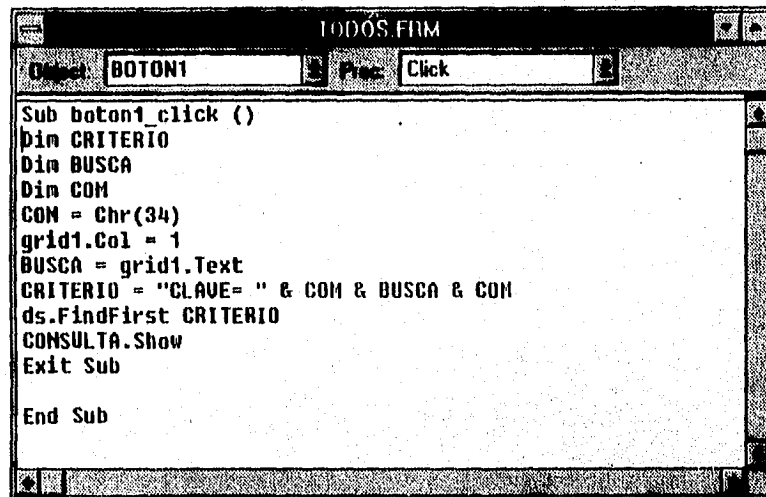
The image shows a screenshot of the Visual Basic Properties window. The title bar reads 'Propiedades'. The window is titled 'Outline1 Outline'. Below the title bar, there is a tree view showing 'Outline1' selected. The main area of the window displays a list of properties and their values for the selected object. The properties listed are: BackColor (&H80000005&), BorderStyle (1 - Fixed Single), DragIcon (none), DragMode (0 - Manual), Enabled (True), FontBold (True), FontItalic (False), FontName (MS Sans Serif), FontSize (8,25), FontStrikethru (False), FontUnderline (False), ForeColor (&H80000008&), Height (2055), HelpContextID (0), Index (1), and Left (1680).

Property	Value
BackColor	&H80000005&
BorderStyle	1 - Fixed Single
DragIcon	(none)
DragMode	0 - Manual
Enabled	True
FontBold	True
FontItalic	False
FontName	MS Sans Serif
FontSize	8,25
FontStrikethru	False
FontUnderline	False
ForeColor	&H80000008&
Height	2055
HelpContextID	0
Index	1
Left	1680

Fig. 1.4.2 Propiedades de un objeto

4.- Conexión de código para cada objeto.

Para este paso se requiere conocer cuales van hacer los eventos que afectarán a cada objeto y que se realizará cuando estos sucedan. Cada objeto posee un conjunto de eventos predeterminado en los cuales puede incurrir.



The image shows a screenshot of a Visual Basic IDE. At the top, there is a window title bar that reads "TODOS.FRM". Below the title bar, there is a toolbar with a "Click" event selected. The main area of the window contains the following VBA code:

```
Sub boton1_click ()  
Dim CRITERIO  
Dim BUSCA  
Dim COM  
COM = Chr(34)  
grid1.Col = 1  
BUSCA = grid1.Text  
CRITERIO = "CLAVE= " & COM & BUSCA & COM  
ds.FindFirst CRITERIO  
CONSULTA.Show  
Exit Sub  
  
End Sub
```

Fig. 1.4.3 Escritura de código a un evento

5.- Prueba y Depuración de una aplicación

CARACTERÍSTICAS DEL EDITOR

1. Una de las características mas importantes de Visual Basic 3.0 es la gran facilidad que se tiene para el desarrollo de las Pantallas (Denominadas en VB como formas) , estas se realizan a través de una barra de herramientas la cual posee una serie de objetos Visuales como por ejemplo; Botones, Etiquetas de textos, Combos, Barras de Menús, etc. El primer paso para la realización de programas es precisamente la colocación de los objetos necesarios en una 'forma' para luego asociar el código correspondiente a cada objeto. Esta forma de programación se denomina Programación por eventos.

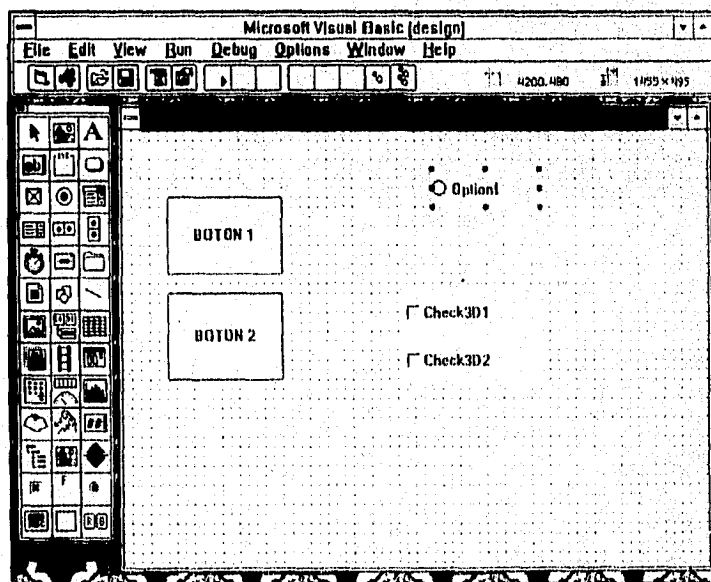


FIG. 1.4.4. EDITOR DEL VISUAL BASIC 3.0

2. Herramienta para la generación de ayudas: A través de esta herramienta se puede crear los archivos de Ayudas en Línea necesarios, utilizando el ambiente que proporciona Windows.
3. Herramienta para la generación de discos de instalación: Mediante esta herramienta se puede generar los discos de instalación de nuestra aplicación teniendo como ventajas lo siguiente: todos los archivos que son necesarios para que nuestra aplicación trabaje sean incluidos (iconos, fuentes, custom control, etc.), los archivos son compactado y se anexa un sencillo programa de instalación similar al que existe en todos las aplicaciones de la Compañía Microsoft.
4. Contiene librerías de Iconos y Gráficas que pueden ser utilizados en el diseño de las aplicaciones.
5. Posee una herramienta para el análisis de como operan las aplicaciones (Debugging). Esta herramienta de Depuración es usada particularmente para la localización de errores en el código fuente.
6. Contiene programas de ejemplo para cada una de las características del Visual Basic y un sistema de Aprendizaje.

• PROGRAMACIÓN

7. Es un lenguaje de Programación completo y permite la construcción de programas estructurados.
8. No es un lenguaje Orientado a Objetos por que carece de las componentes de herencia y Polimorfismo, aunque no de encapsulamiento.
9. Su compatibilidad con las antiguas versiones de Basic es casi completa acepta instrucciones tradicionales como PRINT, INPUT o DATA aunque estas instrucciones no son muy útiles.
10. Utiliza el Tipo de Dato 'Variant' el cual es un tipo de variable de datos que puede almacenar Datos de cualquier tipo, ya sea numérico, cadenas o de fecha/tiempo.
11. La Programación en Visual Basic es por eventos, es decir, que se asocio código a las acciones posibles de los objetos de nuestra 'forma'.
12. Utilización de las denominadas Visual Basic eXtencion (VBX). Los VBX programas normalmente realizados en lenguaje C que realizan una función determinada y que pueden ser utilizados de manera directa en las Aplicaciones Realizadas en Visual Basic.
13. Posee una librería de rutinas especializadas con la cual el programador puede crear sus propios VBX, estas librerías están basadas en el software llamado 'Microsoft Windows Software Developed Kit (SDK) '.



• CARACTERÍSTICAS AVANZADAS

14. Permite la creación de aplicaciones MDI (Multiple Document Interface): Una aplicación MDI permite al usuario desplegar múltiples documentos (formas) al mismo tiempo cada uno de ellos en su propia ventana, estos documentos están contenidos en una ventana denominada 'Ventana Padre' que provee el espacio de trabajo para todas las 'Ventanas Hijas'. Ejemplo de esto es, el Manejador de Programas de Windows (Windows Program Manager) el cual contiene muchos diferentes grupos de programas. Cada grupo de programas es visualizado en su propia ventana y esta confinada a la 'Ventana Padre' que es el Manejador de Programas de Windows.
  
15. Permite el Intercambio Dinámico de Datos (DDE) con otras aplicaciones: Debido a que las Aplicaciones Generadas con Visual Basic se ejecutan en un Sistema Operativo Multitareas, es decir, que corren varias aplicaciones al mismo tiempo el intercambio de datos con estas aplicaciones podría ser con el conocido 'Cut and Paste'. Sin embargo Visual Basic nos permite automatizar este proceso usando el DDE (Dynamic Data Exchange) para extraer datos de otras aplicaciones y automáticamente actualizar en nuestra aplicación.
  
16. Permite el uso de OLE (Object Linking and Embedding): OLE es un método de Intercambio e Interoperatividad entre aplicaciones. A través de OLE las aplicaciones generadas en Visual Basic Podrán desplegar y manipular Datos de otras aplicaciones Windows.

17. **Llamadas a Procedimientos en DDLS: DLL (Dynamic-link libraries)** son una característica básica de Microsoft Windows. Los DDLS son librerías de procedimientos que las aplicaciones pueden enlazar y usar en el tiempo de ejecución, pueden ser compartidas por múltiples aplicaciones. En estas librerías están contenidos los procedimientos de manejo de memoria, graficación, etc. Las aplicaciones en Visual Basic pueden Realizar llamadas a estos procedimientos para el desempeño de acciones especiales que involucren DLLs del Ambiente Windows (USER.DLL, GDI.DLL, KERNEL.DLL, MMSYSTEM.DLL) u otros DLLs

- **BASES DE DATOS:**

1. Visual Basic incorpora un acceso directo a bases de datos de los siguientes formatos; Microsoft Access, Dbase III, Dbase IV, Btrieve, Microsoft Foxpro, Oracle, Paradox y Microsoft SQL server, sin embargo hay que mencionar que el formato de Base de Datos nativa es Microsoft Access dándole a este la mayor flexibilidad y velocidad en el acceso a los Datos.
2. Permite el uso de ODBC (Open Databases Connectivity) con esto permite la utilización de bases de Datos Externas, esta categoría incluye las Bases de Datos en el modelo Cliente Servidor.
3. La Arquitectura de Bases de Datos es extensible, es decir se pueden utilizar Drivers (Manejadores) para Adicionar formatos de Bases de Datos de otros Fabricantes y sin la necesidad de realizar cambios en el código original.

- Incorpora el Uso del Lenguaje SQL (Structure Query Language), este lenguaje es utilizado para la realización de consultas, actualizaciones y el manejo de Bases de Datos Relacionales. Para el caso de las Bases de Datos Externas (ODBC) los comandos SQL son mandados al Servidor para su Procesamiento.

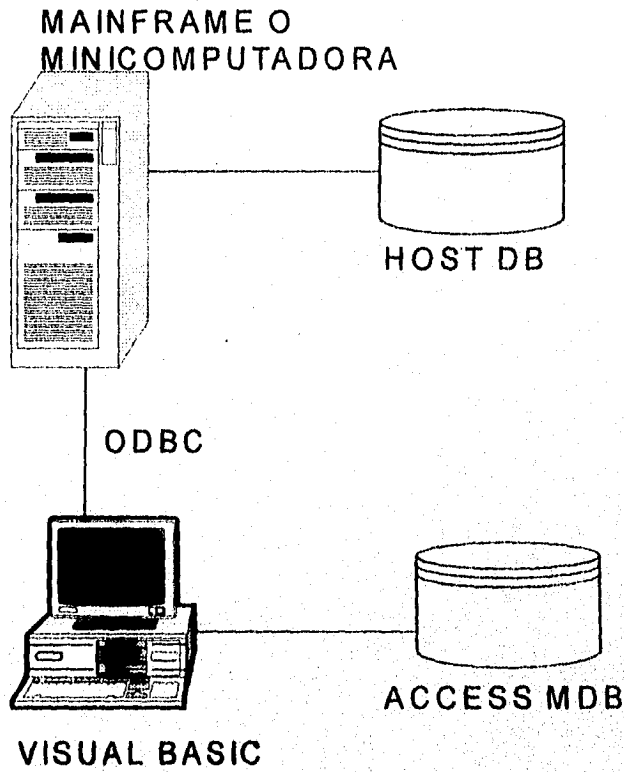


Fig. 1.4.5. Acceso a bases de datos externas

- El lenguaje SQL de Visual Basic cumple con las normas de ANSI (ANSI-86, ANSI-89 y ANSI-9).

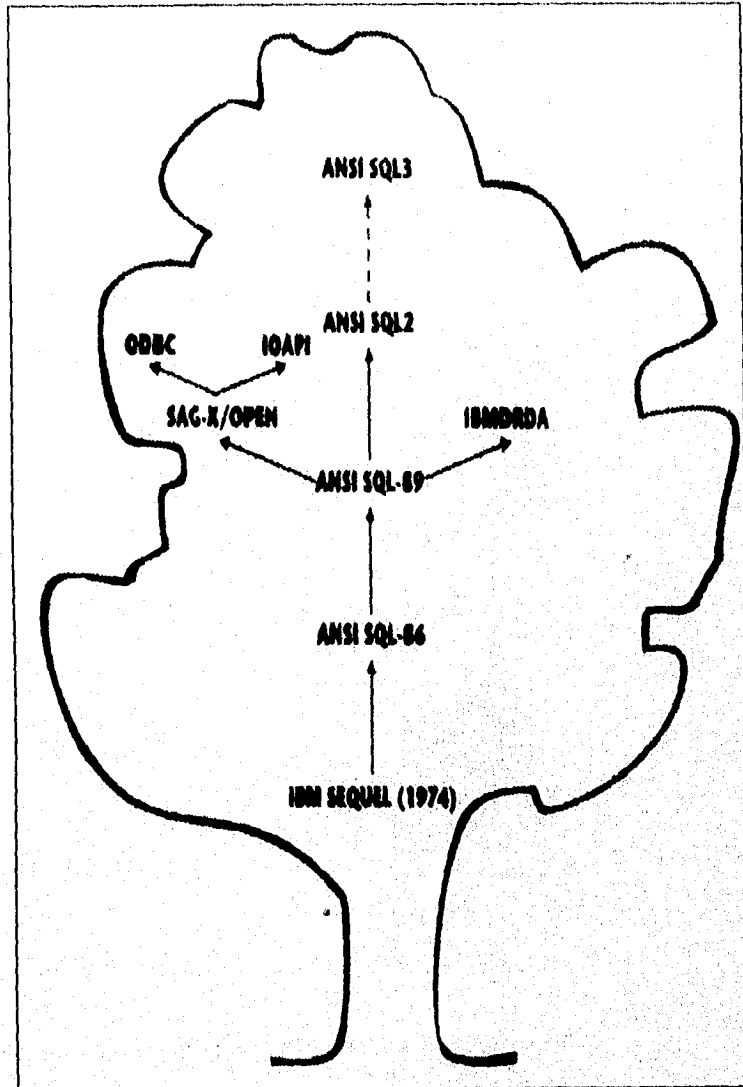


fig. 1.4.6. Evolución del lenguaje SQL.

6. Posee una Poderosa Herramienta para la Generación de Reportes denominada Crystal Reports. Mediante esta herramienta nos permitirá:
- Realizar las conexiones necesarios entre Bases de Datos
  - Realizar cálculos de las distintas bases de Datos incluyendo cálculos de subtotales, comparaciones, conversiones de datos.
  - Realizar el Formato de nuestros Reportes incluyendo fuentes, márgenes, tamaño de hoja. etc.
  - Fijar encabezados y pies de Paginas, títulos de columnas, etc.

• **VENTAJAS**

1. Los requerimientos de hardware para Visual Basic son comparativamente menores que otras herramientas de Programación Visual que existen en el mercado. Los requerimientos mínimos para que funcione Visual Basic son:
- Computadora IBM compatible con procesador 286,
  - 4 MB de RAM,
  - Disco Duro con por lo menos 10MB de espacio disponible.
  - Sistema Operativo 3.1 ó superiores
  - Microsoft Windows 3.0 ó superiores

2. Otra de las grandes ventajas de Visual Basic es la utilización de las denominadas Visual Basic eXtension (VBX) ó 'Custom Controls'. Mediante un VBX se puede desarrollar funciones que no están integradas en el Visual Basic. Por ejemplo, si un programador necesita el uso en alguna aplicación de gráficas que se encuentran en formatos distintos a los que maneja Visual Basic (por ejemplo GIF ó TIFF), mediante un VBX puede ser programado en un lenguaje distinto, y ser utilizado en las Aplicaciones Visual Basic. Y aun mejor existen actualmente en el mercado mas de 1000 VBX con funciones que van desde la graficación, cálculos financieros o animación hasta la inteligencia artificial y el reconocimiento de voz.
  
3. Las licencias de Distribución de las Aplicaciones Generadas en Visual Basic, son fáciles de obtener en comparación con algunos otros lenguajes que hay que comprar una licencia por cada aplicación.

• **DESVENTAJAS**

- 1.- La capacidad de los 'custom controls' que traen de fábrica es limitada, sobre todo si el Visual Basic va a ser utilizado para el manejo de Bases de Datos de gran tamaño, y obliga a usuarios la adquisición de 'Custom Controls' de terceros Fabricantes.
  
- 2.- Los requerimientos de Hardware aumentan considerablemente si se utilizan demasiados 'Custom Controls' en un solo programa.

## **1.5 SEGURIDAD DE DATOS**

Con la evolución de la computación en los últimos años, ésta se ha hecho mas accesible. El número de instalaciones de computadoras ha aumentado de una forma colosal durante este tiempo. Hoy en día, la vulnerabilidad se está convirtiendo en algo cada vez más serio debido a la proliferación de las redes de computadoras.

Otra tendencia de este periodo ha sido asignar a las computadoras un número cada vez mayor de tareas de administración de nuestras actividades personales y de negocios. Las computadoras manejan de forma rutinaria la correspondencia confidencial, transferencias de fondos haciendo circular grandes cantidades de dinero por todo el mundo en cuestión de segundos. Los sistemas de control de tráfico aéreo se han hecho cargo de gran parte de las tareas que antes eran dirigidas por los controladores de tráfico.

Debido a esto es importante implementar medidas de seguridad para asegurar el funcionamiento ininterrumpido y libre de problemas de estos sistemas. Son millones los sistemas de computación que están conectados por medio de redes de comunicación, con frecuencia a través del sistema telefónico. Donde en el pasado la simple seguridad física parecía ser suficiente para proteger a una computadora de un ataque externo, ahora deben introducirse complicados controles. Los sistemas se enfrentan a una gran encrucijada, por un lado deben ser favorables al usuario, es decir, si el uso de la computadora va a ponerse al servicio del público en general de una manera eficiente, los sistemas deberán hacerse accesibles y fáciles de usar. Pero por su propia naturaleza, los beneficios hacia el usuario pueden implicar un aumento de la vulnerabilidad.

## Capítulo I

---

Un programa de seguridad comienza con la identificación de las amenazas potenciales. Las amenazas pueden proceder de fuentes maliciosas o no. Lo mejor que se puede esperar, en este caso es identificar la gran mayoría de las amenazas potenciales aunque siempre habrá algunas que no podrán ser previstas.

La facilidad de uso y la protección de sistemas son metas contradictorias. Los programas pueden estar aislados unos de otros ejecutándolos en computadoras separados (no conectadas en red) pero esto impide el compartimiento de datos.

El nivel de seguridad que deben proporcionarse a un sistema depende mucho del valor de los recursos que hay que asegurar. Por ejemplo, la información de un sistema bancario puede tener un gran valor financiero.

A continuación se analizan los principales métodos para conservar la seguridad en un sistema de computo.

### • SEGURIDAD EXTERNA

La seguridad externa consiste en seguridad física y seguridad operacional. La seguridad física incluye la protección contra desastres. La utilización de mecanismos de detección son importantes para la seguridad física; los detectores de humo y sensores de calor ofrecen una alarma anticipada contra incendios. La protección contra desastres pueden ser costosa y a menudo se trata con demasiada ligereza. Depende en gran medida de las consecuencias de la pérdida; la mayoría de organizaciones no pueden costear un esquema de protección completo contra desastres. Los principales esfuerzos de la seguridad física están dirigidos a impedir la entrada a intrusos. Para esto, es común la utilización de sistemas de identificación física como tarjetas de identificación magnéticas o sistemas de reconocimiento de huellas digitales o voz.



- **SEGURIDAD OPERACIONAL**

La seguridad operacional consiste en las diferentes políticas y procedimientos implementados para la administración de los sistemas computacionales. Las autorizaciones determina qué acceso se permite y a quien (clasificación).

La clasificación divide el problema en subproblemas, los datos del sistema y los usuarios se dividen en clases, a los cuales se concede diferentes derechos de acceso. Un tratamiento común es este aspecto es la división de responsabilidad, a los usuarios se le dan distintos conjuntos de responsabilidades, que realizan sin necesidad de conocer la totalidad de un sistema. Con esto, los individuos no pueden ver el panorama total. Para comprometer a un sistema, puede ser necesaria la cooperación entre muchos empleados, con la cual se reduce en gran medida la probabilidad de violar la seguridad. En estas instalaciones computacionales, por ejemplo, solo los operadores del computador tienen acceso al sistema, sólo los programadores pueden escribir programas, de manera similar las entradas a los sistemas de producción son controlados por un grupo aparte y las salidas son comprobadas y verificadas por un grupo diferente.

Deben controlarse un gran número de verificaciones y balances en el sistema para ayudar a la detección de cualquier brecha en la seguridad. El personal de operación debe desconocer la naturaleza de estos controles; esto reduce la probabilidad de poder evitarlos. Pero deben estar al tanto de que el sistema dispone de elaborados controles; este conocimiento es suficiente para disuadir a muchas personas de intento de violar la seguridad.

- VERIFICACIÓN DE AMENAZAS.

Una forma de reducir al mínimo los riesgos de seguridad es hacer que los sistemas de computo controlen las operaciones delicadas, en vez de darle el control directamente al usuario. Si un usuario desea acceder a un archivo importante, por ejemplo, algunos sistemas pueden permitir acceder al archivo directamente. El usuario puede tener el control y el sistema operativo puede no tener la capacidad de verificar la actividad contra el archivo mientras este tiene lugar. Una técnica más segura es la llamada 'verificación de amenazas'. En la verificación de amenazas, los usuarios no pueden tener acceso directo a un recurso. Tan sólo las rutinas del sistema operativo llamados 'programas de vigilancia' pueden hacerlo. El usuario que desee acceder a un recurso tiene que pedir tal acceso al sistema operativo. El sistema operativo niega o permite el acceso; entonces, un programa de vigilancia realiza el acceso al archivo y pasa los resultados al programa, del usuario. La verificación de amenazas es una actividad que se da sobre la marcha; los intentos de penetración son detectados en el momento en que se producen y el administrador del sistema es inmediatamente informado.

- **PROTECCIÓN POR CONTRASEÑA.**

Tres son las clases de elementos de autenticación para establecer la identidad de una persona:

- *Algo sobre la persona.* Esto incluye huellas digitales, registro de voz, fotografías y firmas.
- *Algo poseído por la persona.* Esto incluye insignias especiales, tarjetas de identificación, llaves, etc.
- *Algo conocido por la persona.* Esto incluye contraseñas, combinaciones de cerraduras, etc.

El esquema más común de autenticación en uso, es la simple protección con contraseña. El usuario elige una palabra clave de varios caracteres de longitud, la guarda en su memoria, y después la teclea para ser admitido en un sistema de computación. Muchos sistemas suprimen la visualización de la contraseña para que no aparezca sobre la pantalla al ser tecleado.

La protección mediante contraseña tiene muchas desventajas. Los usuarios tienden a elegir contraseñas fáciles de recordar, como el nombre de un amigo o pariente. Alguien que conozca al usuario puede intentar la entrada varias veces usando los nombres de diversas personas conocidas por el usuario. Esto puede dar como resultado una brecha en la seguridad por el uso de intentos repetidos de una clave. Algunos de los primeros sistemas usaban contraseñas cortas; estos sistemas eran fáciles de superar por la simple prueba de todas las contraseñas posibles. Muchos de los sistemas actuales usan contraseñas más largas para impedir tales intentos de penetración.

## Capítulo I

---

El uso de contraseñas largas tampoco es demasiado inteligente; las contraseñas largas son difíciles de recordar, por lo que trae como consecuencia que los usuarios tenderán a anotarias, lo cual, claro está, puede facilitar que alguien se entere de una contraseña. Por tal motivo, los diseñadores deben elegir un esquema que use contraseñas lo suficientemente largas como para hacer que los ensayos repetitivos resulten estériles, pero lo suficientemente cortas como para que sean fáciles de recordar.

Los usuarios de la mayoría de los sistemas son instados a cambiar sus contraseñas de manera periódica. aunque el intruso consiga una contraseña, ésta puede ser cambiada antes de que se produzca un daño real. Una defensa sencilla contra los intentos repetidos es limitar el número de intentos de entrada que puede realizarse en un periodo de tiempo.

## Capítulo I

---

El uso de contraseñas largas tampoco es demasiado inteligente; las contraseñas largas son difíciles de recordar, por lo que trae como consecuencia que los usuarios tenderán a anotarlas, lo cual, claro está, puede facilitar que alguien se entere de una contraseña. Por tal motivo, los diseñadores deben elegir un esquema que use contraseñas lo suficientemente largas como para hacer que los ensayos repetitivos resulten estériles, pero lo suficientemente cortas como para que sean fáciles de recordar.

Los usuarios de la mayoría de los sistemas son instados a cambiar sus contraseñas de manera periódica. aunque el intruso consiga una contraseña, ésta puede ser cambiada antes de que se produzca un daño real. Una defensa sencilla contra los intentos repetidos es limitar el número de intentos de entrada que puede realizarse en un período de tiempo.

• **AUDITORIA**

La auditoria son llamados periódicos para examinar las recientes transacciones de una organización y determinar si se ha producido alguna actividad fraudulenta. La auditoria en un sistema de computo puede implicar un procesamiento computacional inmediato para verificar las transacciones que se acaban de producir. Un registro de auditoria es un registro permanente de acontecimientos importantes ocurridos en el sistema. Este registro se realiza de manera automática cada vez que tiene lugar un evento elegido y se almacena en un área altamente protegida del sistema; si el sistema se ve comprometido, los registros de auditoría permanecen intactos. Por ejemplo, se pueden llevar registros de Auditoria de acciones como: Entrada y salida al sistema de usuarios denominados administradores, acceso a algún programa de alta importancia para la organización, ejecución de comandos del sistema operativo peligrosos como es el caso del comando para borrar archivos o cintas, modificaciones de los derechos de acceso a los usuarios, etc.

El registro de auditoría es un importante mecanismo de detección, aunque las defensas del sistema pueden ser penetradas, la gente puede sentirse poco inclinada a intentarlo por el temor a la detección después del hecho.

El solo hecho de producir un registro de auditoría no garantiza una buena seguridad. Es necesario revisar frecuentemente y cuidadosamente el registro. Tales revisiones deben realizarse de manera periódica y aleatoriamente. Las auditorías periódicas prestan atención regular a los problemas de seguridad; las auditorías al azar ayudan a atrapar a los intrusos desprevenidos.

- **DIARIOS**

Los diarios son registros similares a los registros de auditoría, excepto que este se lleva a cabo en las bases de datos de algún sistema. Es decir, lleva registros de las transacciones tales como altas, bajas y modificaciones de alguna base de datos que se considere crítica, quien la realizó y como estaban los datos antes de esta transacción. Mediante los diarios también nos permitirá la recuperación de datos en caso de fallas en el sistema.

- **DERECHOS DE ACCESO**

La clave de acceso a un sistema solo limita la entrada a éste. Con los derechos de acceso se define qué acceso tienen varios sujetos a varios objetos. Los objetos son entidades que contienen información, pueden ser objetos concretos como discos, cintas, procesadores o impresoras, o pueden ser objetos abstractos que corresponden a las estructuras de datos de algún archivo.

Los derechos o autorizaciones de un sistema se conceden a los sujetos. Los sujetos pueden ser varios tipos de entidades, incluyendo a usuarios, procesos u otras entidades.

Los derechos de acceso más comunes son:

- **Derecho de lectura:** El usuario o entidad solo puede leer el objeto y no modificarlo.
- **Derecho de escritura:** El usuario o entidad puede modificar y agregar datos al objeto.

- **Derechos de ejecución:** Se utiliza en el caso en que el objeto sea un programa, con este derecho el usuario puede ejecutar dicho programa.
- **Derechos de Modificación.** Es un caso particular del derecho de escritura en el cual el usuario solo puede modificar datos pero no agregarlos.

Para la implementación de los derechos de acceso se utiliza una matriz de control de accesos, en la cual los distintos sujetos son listados y los diferentes objetos a los cuales tienen acceso. Una matriz de control de acceso debe, desde luego, ser una de las entidades más celosamente guardada de un sistema.



• CRIPTOGRAFÍA.

La criptografía fue solo de interés casi exclusivo para la comunidad militar y política. En la actualidad es especialmente importante en sistemas de computación. El espionaje es más fácil ahora. Grandes volúmenes de transacciones comerciales se tratan a través del teléfono. El uso de redes está creciendo a marchas forzadas. El uso del correo electrónico y transferencia electrónica de fondos está en aumento.

La criptografía es el uso de la transformación de datos, para hacerlos incomprensibles a todos, excepto a los usuarios a quienes están destinados. El 'problema de la intimidad' trata de cómo evitar la obtención no autorizada de información de un canal de comunicaciones. El 'problema de autenticación' trata sobre cómo evitar que un oponente modifique una transmisión o le introduzca datos falsos.

En un sistema de intimidad criptográfica, el remitente desea transmitir cierto mensaje no encriptado (llamado texto simple) a un receptor legítimo; la transmisión se producirá a través de un canal inseguro. El remitente pasa el texto simple a una unidad de codificación que transforma el texto simple en un texto encriptado o criptograma; el texto no es comprensible para los eventuales espías. En el extremo receptor, el texto encriptado pasa por una unidad de descifrado que regenera el texto simple.

La criptografía es especialmente útil en los sistemas de tiempo compartido y en las redes de computadoras. Las contraseñas deben ser almacenadas en forma encriptada a fin de que la lista de contraseñas solo tenga utilidad para el sistema operativo. Esta es la aplicación más común de la criptografía, en los sistemas actuales.

La criptografía puede usarse también para proteger todos los datos almacenados en un sistema. Pero la carga de trabajo a una computadora para realizar el encriptado/desencriptado es muy grande como para hacerlo habitualmente. Aunque algunos sistemas han incorporado hardware especial para realizar estas tareas, como es el caso de unidades de cintas y discos que tienen la opción de encriptamiento al momento de grabar cintas o discos.

Las redes de computadoras también han introducido técnicas de Criptografía. Los protocolos de redes en capas ofrecen varios niveles de encriptado. En el encriptado de enlace, la red asume la responsabilidad de encriptado/desencriptado de cada nodo. Los datos son transmitidos entre los nodos, en forma encriptada, decifrándolos en cada nodo para determinar a dónde pasarlo a continuación.

- **SISTEMAS SUPERVIVIENTES.**

Los sistemas supervivientes ofrecen la posibilidad de continuar operando, aún cuando se haya presentado fallas en alguno de sus componentes. Los sistemas supervivientes están diseñados de manera que un componente fallido pueda quitarse de línea, ser reparado y colocado de nuevo en línea sin interrumpir el servicio del sistema.

Una clave para la capacidad de supervivencia es la redundancia. Si un componente falla, otro equivalente toma su puesto. Algunos sistemas utilizan una bolsa de recursos redundantes sólo cuando se produce un fallo.

En los últimos años las arquitecturas de microprocesadores han evolucionado constantemente hacia diseños más libres de fallo. Muchos de ellos usan en la actualidad estructuras de bus doble para la comunicación. El multiprocesamiento es importante en cualquier sistema superviviente. Algunas características de supervivencia son:

- La incorporación de mecanismos contra fallos en el hardware en vez de en el software.
- El uso de multiprocesamiento transparente, esto permite mejorar el rendimiento sin modificar el software.
- El uso de subsistemas múltiples de entrada/salida.
- La incorporación de mecanismos de detección de fallos en el hardware y el software.

Uno de los ejemplos típicos de los sistemas supervivientes es el denominado RAID (Redundant Array of Inexpensive Disk). El sistema consta de un arreglo de 3,5 o 7 discos en los cuales permite la falla de un único disco y puede reconstruir de forma automática los datos perdidos como consecuencia de la falla. Esto se realiza por medio de una suma de comprobación (checksum) que se almacena en partes designadas como redundantes (generadas por la aplicación de un OR exclusivo entre datos de los diferentes discos) en cada disco. Si alguno de los discos fallara automáticamente su información se reconstruiría, por lo menos mientras es sustituido por un nuevo disco, esto sin afectar la operación normal del sistema.

Como se puede observar existen varios métodos para salvaguardar la seguridad de un sistema; una organización puede utilizar alguno de ellos o todos ellos, dependiendo de la seguridad necesaria en cada organización, sin embargo hay que tomar en cuenta que cada método de seguridad consume recursos tanto en los equipos de cómputo como en las personas que lo administran, realizar un balance de seguridad y el costo asociado, es importante para los administradores del sistema.

**“CAPÍTULO II  
“PLANTEAMIENTO DE  
LA PROBLEMÁTICA Y  
PROPUESTA DE  
SOLUCIÓN”**

**OBJETIVO:**

**Determinar la problemática actual dentro de los procesos administrativos de la unidad de servicios escolares, para implementar posibles soluciones.**

## **2.1 ANTECEDENTES**

En la actualidad, la Universidad Nacional Autónoma de México vive una dinámica que exige que los responsables de la administración y manejo de información estratégica en todos los sectores, se anticipen en estimular todos los procesos de mejora continua hacia la calidad total y la excelencia, promoviendo una mayor productividad y el consecuente incremento de la eficiencia en sus procesos. Todo ello requiere crear e impulsar una nueva cultura, desarrollando las habilidades y aptitudes directivas, y continuamente estar en busca de un futuro cada vez mejor, para que los procesos que se realizan dentro de la institución y el aprovechamiento de los recursos hagan posible alcanzar la excelencia que persigue la UNAM.

Es por ello, que la UNAM, como todas las instituciones educativas, requieren que la información que se maneje dentro de ellas esté lo más seguro posible, es por eso que la institución ha ido adoptando modernos mecanismos y sistemas, que poco a poco irán dando sus frutos.

### **2.1.1 ESQUEMA ACTUAL DE INSCRIPCIÓN**

Para lograr la plena identificación con el proceso de modernización, se tiene invariablemente que presentar un análisis a detalle de cuáles son las condiciones, los requerimientos y en general el ambiente que se respira dentro de la FESZ en materia de manejo de información.

## Capítulo II

Por mucho tiempo la Facultad de Estudios Superiores Zaragoza, ha estado buscando la manera de que el tratamiento de toda la información que se maneja dentro de la institución sea lo más seguro y eficaz que se pueda.

Desde su creación, en 1976 la FESZ, ha tenido que sujetarse a las condiciones que en materia de manejo de información se refiere, a lo que dictamine el nivel central, ajustándose a los procesos que de él emanan y necesariamente cumplirlos adecuadamente y en el tiempo señalado.

La situación actual de la FESZ en cuestión informática es la siguiente:

1. Todos los trámites que se realizan actualmente dentro del seno de la FESZ, son realizados mediante procesos manuales y auxiliados de Programas aislados de cómputo.
2. Los procesos, hasta el momento se realizan de manera ordenada y con un nivel de eficacia aceptable (aunque lentamente), el problema se presenta cuando los diferentes departamentos requieren de alguna información referente a alguno de éstos procesos, estadísticas ó simplemente un reporte de actividades, es entonces cuando el tiempo se vuelve un enemigo insuperable. Este tipo de problemas se agrava en lo que actualmente es la unidad de servicios escolares, donde el volumen de información excede y con mucho a la capacidad que pudiesen tener los encargados de dicha unidad, ya que ahí se maneja toda la información referente a los alumnos, profesores, grupos, asignaturas, historiales académicos, estadísticas, inscripciones, y en general toda la información que sirve como base para que la FESZ opere de manera exitosa, es por ello que cada vez que un evento se realiza (evento se refiere a una actividad, por ejemplo una inscripción de alumnos) las aglomeraciones de alumnos no se hacen esperar y no solo eso, sino que es necesaria la

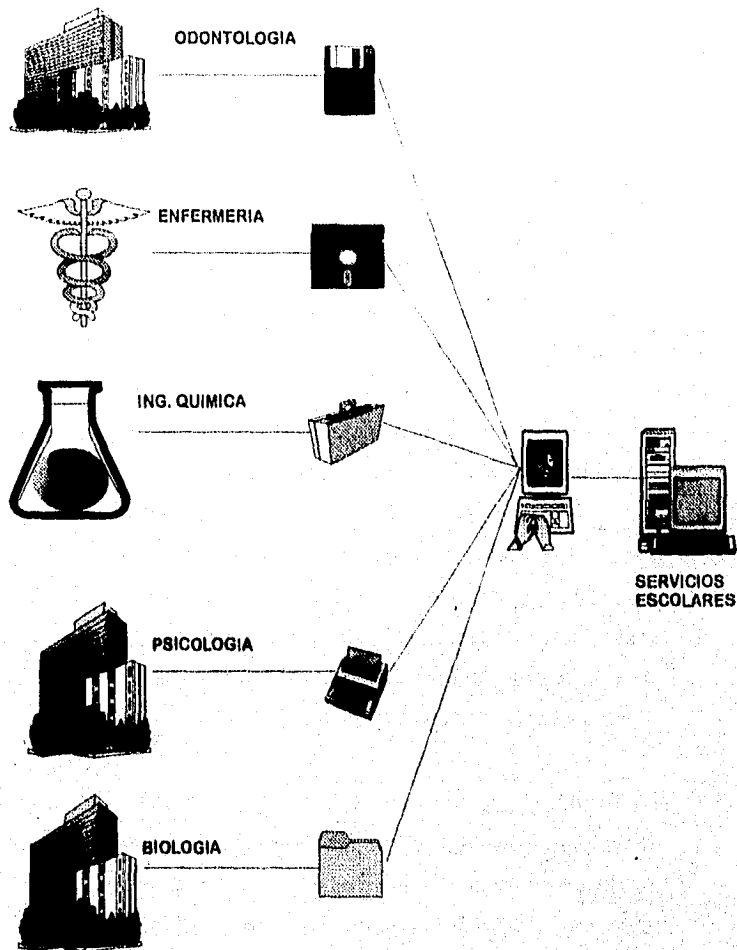
---

*Planteamiento de la problemática y propuesta de solución*

intervención de personal que auxilie en las labores de revisión de documentos necesarios, recibos de pago, de validaciones diversas.

3. La FESZ actualmente cuenta con una pequeña red basada en Novell Netware 3.11 y cuenta con el siguiente equipo: Un servidor Pentium 100MHZ, 32 Mb en RAM, Disco interno de 1Gb; Un servidor 486 DX2, 8 Mb en Ram, Disco interno de 1 Gb; 6 Estaciones 486 DX , 4Mb en Ram, Disco duro de 540Mb, Una computadora dedicada a servidor de impresión; 2 impresoras de alta Velocidad.
4. Un sistema programado en un interprete de Progress 6.2. El manejo de este sistema es complicado y en ambiente texto, se requiere de personal de sistemas especializado para la realización de algunos procesos como consultas, reportes y carga de información necesaria para el correcto funcionamiento del sistema; es fácil capturar datos erróneos a las bases, dado que no existe una secuencia de validación de datos apropiada. Adicionalmente no se cuenta con todos los Módulos del Manejador de Bases de Datos de Progress 6.2.
5. También hay que considerar la posible estandarización de la información dentro de la UNAM. De acuerdo a la política de Manejo de Información referente al control escolar se pretende migrar las bases de datos a una plataforma Unix con un Manejador de Bases de Datos SYBASE a corto plazo.





## SITUACION ACTUAL

Figura 2.1.1 Situación Actual

## **2.1.2 ESQUEMA DE COMO SE PRETENDE QUE OPERE EL SISTEMA**

En el caso particular de la FES Zaragoza, a finales del año de 1992, se le otorgaron recursos en materia de cómputo; dichos recursos están formados por equipo de cómputo: Pc's, impresoras, interfaces de red (ethernet), etc.; así como del software correspondiente para la instalación de una red local (netware 3.11).

Debido a la importancia que ha tenido éste proyecto, la demanda en materia de desarrollo de sistemas ha ido creciendo, así como las necesidades en el manejo de la información. Es decir, la disponibilidad de bases de datos con información única, la cual pueda ser accesada por todos los que la requieran, evitándose la duplicidad de información y datos y su manejo de manera errónea.

Nuestro proyectos llamado "CONEFESZ", el cual pretende utilizar la red de cómputo de la unidad de servicios escolares y lograr la automatización completa de los procesos que se realizan dentro de dicha unidad; eliminando así muchos problemas que se han venido presentando durante la realización de los diferentes eventos que se llevan a cabo en la institución.

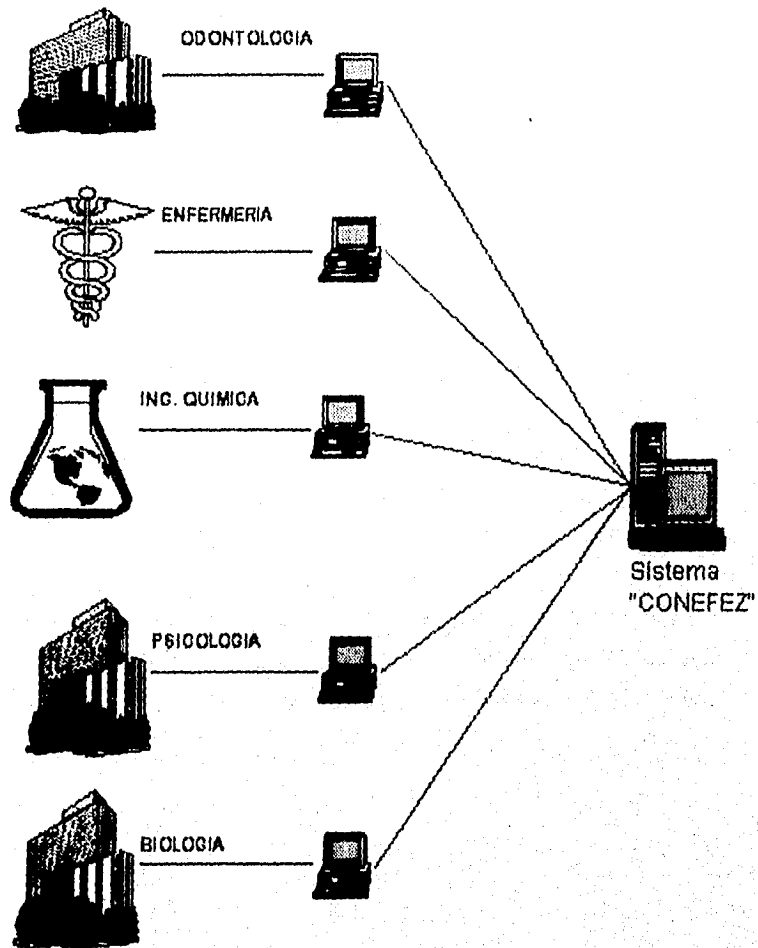
La perspectiva de como se pretende que opere el sistema es la siguiente:

1. El sistema pretende que los procesos de inscripción ordinaria de alumnos, extraordinaria, alta de grupos, profesores, asignaturas, etc., se realice de manera automatizada, aprovechando la infraestructura existente dentro de la unidad de servicios escolares, también reducirá la duración de las jornadas de trabajo de los empleados asignados a dichos procesos, elevando con esto la productividad, y reduciendo al máximo los tiempos de espera para los usuarios finales.

## Capítulo II

---

2. A su vez, se tendrá la ventaja que otorga el manejo de la información de manera automatizada, se podrán obtener estadísticas, informes, listas de alumnos, resultados de los eventos, etc., en el momento que se requieran.
3. Este sistema será diseñado en ambiente gráfico con el objeto de ser de fácil manejo para el usuario y se eliminara la necesidad de personal especializado en algunos procesos.
4. En el diseño del sistema se utilizarán elementos visuales como por ejemplo iconos, gráficas, botones. Con la finalidad de que el manejo del sistema sea intuitivo para el usuario y la capacitación de los operadores sea mínima.
5. El sistema se desarrollará optimizando la captura de datos mediante la utilización de botones, menús desplegables, teclas de acceso rápido, etc.
6. La identificación del alumno podrá realizarse a través de una imagen digitalizada, si así se desea, acelerando el trámite de inscripción.
7. Se utilizarán los datos acumulados durante 1 año en el sistema antiguo. Esto involucra bases de datos en PROGRESS v6.2 con Alumnos, Maestros e Inscripciones. Esto a petición expresa de la FESZ.



## SITUACION QUE SE PRETENDE

Figura 2.1.2 Situación que se pretende

## Capítulo II

---

f En lo que concierne a los elementos necesarios que requerirá el sistema (CONFESZ) para su correcto funcionamiento serán los siguientes:

- *Una red.*
  - Servidor de archivos.
  - Sistema Operativo de Red (Netware 3.11 de Novell).
  - Manejador de Bases de datos Access 2.0
  - Estaciones de trabajo (Pc's).
  - Supervisor de red.
  
- *Administrador del sistema.*
  
- *Información necesaria.*

Los recursos que se listan ya se tienen, sólo hace falta la información necesaria que las jefaturas de cada carrera que se imparte en la FESZ debe proporcionar, como son las declaraciones de grupos, profesores, etc.

## **2.2 ESTRATEGIA DE SOLUCIÓN**

Se denomina así a la metodología que se sigue para dar solución a un problema. En este caso se plantea el análisis y el diseño de los sistemas. Una gran parte de éste, se encuentra en lo que se denomina como ciclo de desarrollo de los sistemas o ciclo de vida de los sistemas. Este es un enfoque por etapas de análisis y de diseño, postula que el desarrollo de los sistemas mejora cuando existe un ciclo específico de actividades.

### **2.2.1 PLAN DE TRABAJO**

Se considero que el ciclo de desarrollo es el más apto para realizar el sistema.

El ciclo de desarrollo por lo general se divide en siete etapas como se muestra en la figura 2.2.1.1 Aunque cada etapa se presenta de manera discreta, nunca se lleva a cabo como un elemento independiente. Por lo que se realizarán diversas actividades, y éstas llegan a repetirse. Por esto es de mayor utilidad suponer que el ciclo de desarrollo de sistemas transcurre en etapas (con actividades en plena acción que luego cesan poco a poco) y no como elementos separados.

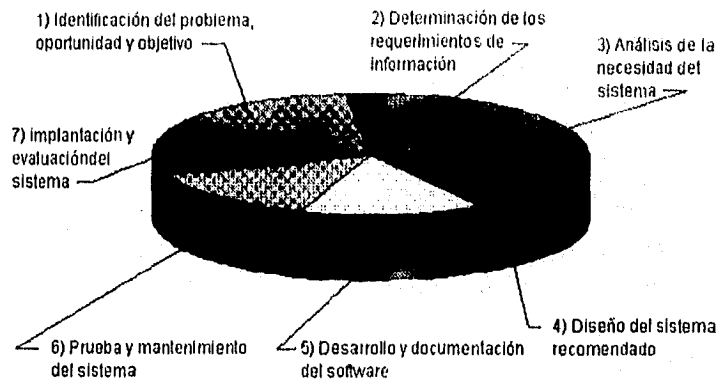


Figura 2.2.1.1

La primera etapa nos indica la identificación del problema, oportunidades y objetivos. Se identificó el problema el cual es que la información que se tiene de los alumnos no se encuentra estandarizada, existen formatos incompatibles con complicados procesos en la integración.

Las oportunidades que se pueden obtener a través del sistema es que no se generará duplicidad de información y se podrá tener un mejor control, se podrá obtener un retraso menor en la generación de documentos para los diferentes usuarios, así como la obtención de un solo formato.

El objetivo es desarrollar un sistema que maneje los datos de los alumnos por su registro y reinscripción ordinaria y extraordinaria, además de poder realizar cambios y actualizaciones a los historiales académicos, para proporcionar información a los alumnos de la FESZ, DGAE y departamentos internos que lo requieran.

El segundo punto del desarrollo del sistema es la determinación de los requerimientos de información, para obtener esta información utilizamos diferentes instrumentos como es el muestreo, estudios de los datos y formas usadas por la organización, la entrevista y los cuestionarios.

El tercer punto es el análisis de las necesidades del sistema, en donde se utilizan los diagramas de flujo de datos que cuentan con una técnica estructurada para representar en forma gráfica la entrada de los datos a la institución, los procesos y la salida de la información. A partir del diagrama de flujo de datos se desarrolla un diccionario de datos que contiene todos los elementos que utiliza el sistema, así como especificaciones, si son alfanuméricos y el espacio de impresión que requieren.

En éste punto también se analizan las decisiones estructuradas, semiestructuradas y de criterio múltiple.

El cuarto punto es el diseño del sistema recomendado, en éste se utiliza la información recolectada con anterioridad y se elabora el diseño lógico del sistema. En éste punto se diseñan los procedimientos de captura de datos y el acceso al sistema de información, mediante técnicas de diseño de formas y de pantallas. Así también se realizará la interfaz con el usuario que en este caso es la interfaz gráfica realizada en Visual Basic. En este ciclo se incluye el diseño de los archivos o la base de datos que almacenará aquellos datos requeridos. Así como la salida de estos datos ya sea en la pantalla o en la impresora según sea el caso.

La quinto punto es el desarrollo y documentación del software aquí se describe de manera detallada el programa que se desarrollo en Visual Basic.



## Capítulo II

---

Dentro de las técnicas estructuradas para el diseño y documentación del software se tiene los diagramas de flujo. Durante esta fase los usuarios colaboran para desarrollar la documentación indispensable del software, incluyendo los manuales de procedimientos, esta documentación le dirá al usuario cómo manejar el software, y así también, qué hacer en caso de presentarse algún problema.

En el sexto punto denominado pruebas y mantenimiento del sistema, se debe probar el sistema antes de ser liberado. En un principio se hace una serie de pruebas, con datos tipo, para identificar las posibles fallas del sistema; más adelante, se utilizarán los datos del sistema real. El mantenimiento del sistema y de su documentación empiezan aquí.

La última etapa es la implantación y evaluación del sistema; ésta se llevará a cabo dentro del servidor de red Novell 3.11. Dentro de esta parte se encuentra la capacitación a los usuarios del sistema, se definen responsabilidades y el alcance del sistema.

A continuación se explica por medio de un diagrama de flujo los pasos a seguir para la elaboración de un sistema:

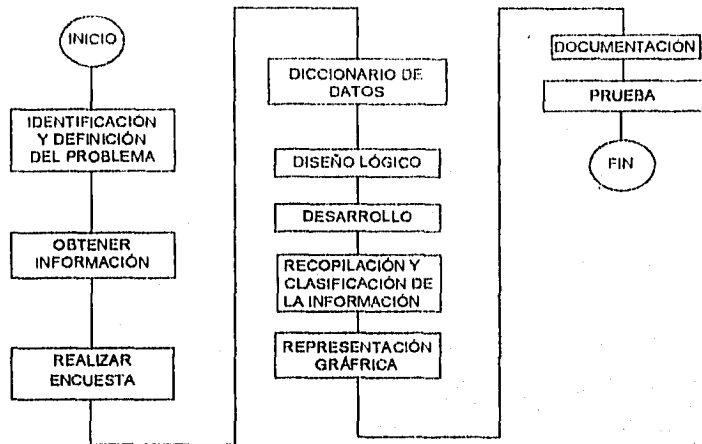


Figura 2.2.2.2 El ciclo de vida que se siguió del sistema.

- 1) **Identificación y definición del problema:** Como ya se mencionó se tiene un problema con la administración escolar de la FES Zaragoza. Para esto nuestro objetivo es solucionar este problema por medio de un sistema que sea estándar en todas las facultades, así como amigable.
- 2) **Obtener información:** Obtener información existente del manejo de la administración escolar como es papeleo, la forma de como controlan hasta el momento a los alumnos, obtención de la base de todos los alumnos.
- 3) **Realizar Encuesta:** Esta encuesta será dividida en cuatro categorías:
  - 1.- Alumnos.
  - 2.- Profesores.
  - 3.- Administradores de Servicios Escolares.

## Capítulo II

---

### 4.- Operadores del sistema.

Esto se hace con el fin de conocer las necesidades de las diferentes categorías y tener una idea de como se realizará el sistema.

- 4) **Diccionario de datos:** Se obtiene tanto la información de las encuestas como la información existente. La información se clasificará en tres grupos, que son : alumnos, Profesores y directivos.
- 5) **Diseño lógico:** Se utiliza un diagrama de flujo de datos donde se explica como funciona el sistema, es decir, se identifican las entradas y salidas de información de la base de datos implementada en Access 2.
- 6) **Desarrollo:** Se elabora un diccionario de datos a partir del diagrama de flujo de datos.
- 7) **Recopilación y clasificación de la Información:** Una vez obtenida toda la información se llega a la conclusión de que el sistema tenga una sola ventana en donde se encuentre la ayuda automáticamente.
- 8) **Representación gráfica:** Se realiza la programación del sistema en Visual Basic utilizando la base de datos implementada en Access 2.
- 9) **Documentación:** Se realiza un manual de usuario, así como la documentación interna del programa.
- 10) **Prueba:** Esta prueba se realiza dentro de la red Novell 3.11 de la FES Zaragoza.

### **2.2.2. RECOPIACIÓN DE LA INFORMACIÓN**

Como antecedentes de cualquier método de recopilación de información (la investigación, la entrevista o la observación), se tiene que decidir qué es lo que se examinará y a quiénes se entrevistarán u observarán, Basándonos primero en el enfoque estructurado llamado muestreo.

El muestreo es el proceso por el cual se selecciona de manera sistemática elementos representativos de una población. El muestreo es muy importante por que reduce costos, evita duplicidad en cuestionarios, además de agilizar el proceso.

Para lograr un buen diseño se deben seguir los cuatro pasos siguientes:

- 1.- Determinar con precisión los datos que se van a recopilar o a describir.
- 2.- Delimitar la población sujeta a selección de muestras.
- 3.- Elegir el tipo de muestra
- 4.- Decidir el tamaño de la muestra

A los directivos de servicios escolares se les pidió su colaboración para dar toda la información que se necesita para las diferentes solicitudes de servicio de los alumnos y profesores, así como, a los operadores del sistema se les pidió sus requerimientos para que puedan dar servicio. La información mínima necesaria es la siguiente:

- a).- Número de Cuenta.
- b).- Nombre del alumno.
- c).- Nombre de asignatura.
- d).- Clave de Asignatura.

## Capítulo II

---

- e).- Facultad.
- f).- Tipo de alumno (regular o irregular).
- g).- Calificación.
- h).- Primer ingreso, reinscripción o inscripción extraordinaria.

La población que esta sujeta a muestra es la siguiente:

- 1.- Alumnos.
- 2.- Profesores.
- 3.- Personal Administrativo.
- 4.- Operadores del sistema.

En la elección del tipo de muestra se escogió un tipo de muestra básico; llamada muestra de oportunidad, donde se cuenta con toda la información de los alumnos, Profesores, Personal Administrativo y Operadores del sistema, esta información no tiene restricciones ni soporte probabilístico.

La decisión sobre el tamaño de la muestra es tomar cinco personas de los cuatro diferentes tipos de población.

Durante la investigación de documentos se visualizaron documentos duplicados de los alumnos, información que no estaba actualizada (como domicilio, materias, etc), algunos alumnos tienen que esperar a que esta sea enviada a la DGAE para poder hacer una actualización u obtener el historial académico.

La búsqueda de información que el alumno requiere es manual y existen enormes archivos en donde se guarda la información de los alumnos.

El control de calificaciones y el formato son diferentes en todas las facultades, y existe un excesivo acumulamiento de información.

## LA ENTREVISTA

Una entrevista se utiliza para la recopilación de información. Es una conversación dirigida con un propósito, que se basa en un formato de preguntas y respuestas.

Para los Alumnos se formularon las siguientes preguntas:

- 1.- ¿Qué trámites realiza con más frecuencia en la unidad de servicios escolares?
- 2.- ¿Considera que el control de la información en la unidad de servicios escolares es adecuado?
- 3.- ¿Qué opina sobre el sistema de cómputo actual?
- 4.- ¿En base a su experiencia, qué tipo de problemas ha encontrado en la realización de sus trámites en la unidad de servicios escolares?
- 6.- ¿Cómo le gustaría realizar sus trámites en la unidad de servicios escolares?
- 7.- ¿Qué te gustaría que se mejorara dentro de la unidad de servicios escolares?
- 8.- ¿Cuántos trámites aproximadamente realizas en el semestre?
- 9.- ¿Cuánto tiempo crees que te ahorrarías si se implantara un sistema de cómputo para el control escolar?
- 10.- ¿Quisieras que hubiese consulta de calificaciones en cualquier momento?.

Para los Profesores:

- 1.- ¿Considera que el sistema de entrega de calificaciones es el adecuado?
- 2.- ¿Cómo le gustaría realizar sus trámites que involucren a la unidad de servicios escolares?

## Capítulo II

---

- 3.- ¿En base a su experiencia, qué tipo de problemas ha encontrado en la realización de sus trámites en la unidad de servicios escolares?
- 4.- ¿Qué te gustaría que se mejorara dentro de la unidad de servicios escolares?
- 5.- ¿Cuántos trámites realiza aproximadamente en el semestre?
- 7.- ¿Quisiera usted corregir una calificación desde una terminal?
- 8.- ¿Está de acuerdo en que los trámites que se realizan en la unidad de administración escolar se lleven a cabo a través de computadora?
- 9.- ¿Quisiera usted que hubiese consulta de calificaciones en cualquier momento?

### Para los Directivos:

- 1.- ¿Qué deficiencias tiene la unidad de administración escolar en cuanto a control escolar se refiere?
- 2.- ¿Se podría aumentar la eficiencia en los procesos de control escolar con un sistema automatizado?
- 3.- ¿Le gustaría que el sistema tuviera ayuda instantánea?
- 4.- ¿Le gustaría que el sistema fuera en base a iconos y que no existiera más que un solo menú?
- 5.- ¿Qué trámites son los más realizados por los alumnos y profesores?
- 6.- ¿Le gustaría que los alumnos y profesores tuvieran acceso a la información que se maneja en servicios escolares?

### Para los Operadores del Sistema:

- 1.- ¿Le gustaría un sistema que fuera amigable?
- 2.- ¿Qué te gustaría que se mejorara dentro de la unidad de servicios escolares?

- 3.- ¿Cuales son los servicios más solicitados por parte de los alumnos y profesores?
- 4.- ¿En su opinión, como desearía que fuese el sistema de control escolar que se implemente en la Unidad de servicios escolares?

La encuesta se realizó tomando una muestra de 30 alumnos, 20 Profesores, 10 directivos y 5 usuarios (operadores del sistema), y los resultados que se obtuvieron fueron los siguientes:

**Para los Alumnos:**

Los alumnos consideraron que la cantidad promedio de trámites que realizan durante un semestre son tardados y excesivos.

El control de la información de los alumnos que se tiene de la FESZ es muy inseguro debido a que las facultades llevan de diferente forma el control de los alumnos y esto produce diferentes errores y retrasos en la entrega de cualquier documento o realización del mismo.

**Para los directivos:**

Los directivos de la unidad de servicios escolares mencionan que tienen los siguientes problemas:

- 1.- La misma información con que se cuenta se encuentra en diferentes formatos.
- 2.- La base de datos sirve básicamente para guardar la información de todos los alumnos.
- 3.- El procesamiento de información es muy lento.



## *Capítulo II*

---

### **Para los Operadores del Sistema:**

- 1.- El sistema actual es poco amigable para el usuario ya que está desarrollado en ambiente texto.
- 2.- No se cuenta con el módulo de progress para generar programas ejecutables.
- 3.- El sistema actual tiene diferentes modulos para una sola función, generando retraso en la información.

Dado lo anterior se decidió implementar un sistema de cómputo desarrollado en un ambiente gráfico, con ayudas en línea y sobre todo amigable al usuario.

También, se homogenizo el sistema para que todas las facultades tuvieran un mismo sistema a través del cual capturar toda la información referente al control escolar.

### 2.2.3. CLASIFICACIÓN DE LA INFORMACIÓN

Una vez reunida toda la información de alumnos, profesores, personal administrativo y operadores del sistema, se clasificó en tres tipos:

#### 1.- Alumnos:

- a).- Número de Cuenta.
- b).- Nombre del alumno.
- c).- Nombre de asignatura.
- d).- Clave de Asignatura.
- e).- Facultad.
- f).- Periodo.
- g).- Tipo de alumno (regular e irregular).
- h).- Calificación.
- i).- Tipo de inscripción: Primer ingreso, reinscripción e inscripción extraordinaria.
- j).- Créditos.
- k).- Clave de la carrera.
- l).- Folio.

#### 2.- Profesores:

- a).- Plantel.
- b).- Grupos.
- c).- Calificaciones.
- d).- Facultad.
- e).- Periodo.
- f).- Clave de asignatura.
- g).- Nombre de profesores.
- h).- Folio.

## **Capítulo II**

---

### **3.- Directivos.**

- a).- Facultad.
- b).- Folio.
- c).- Período.
- d).- Clave de usuario.

### **2.3 ANÁLISIS DE LOS 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, controlar una actividad de la empresa o brindar soporte a las gerencias.

Es así como la determinación de requerimientos vincula el estudio de un sistema existente con la recopilación de detalles relacionados con él.

El éxito o el fracaso de un proyecto depende en gran medida de la calidad de los datos recolectados durante esta fase. La duración del proyecto puede verse afectada si el análisis carece de proporciones significativas de los requerimientos o bien, el usuario falló al proporcionar las respuestas detalladas durante el análisis.

Esta es la fase más importante en la etapa del desarrollo de sistemas. Es durante esta fase que la información sobre la factibilidad y alcance del proyecto, el control, los requerimientos de operatividad de datos y los datos requeridos son obtenidos con el fin de diseñar la Base de Datos adecuada.

Una de las tareas en la definición de un sistema es la de enfocar y comprender el problema que viene a solucionar, para poder emitir una justificación de la solución propuesta. Este proceso requiere de una intensiva comunicación entre el usuario final y el diseñador del sistema.

La mejor manera de comprender la problemática es considerar los tópicos que deberán analizarse, que permitan al diseñador determinar los requerimientos del usuario, considerando lo siguiente:

## **Capítulo II**

---

- Identificar a todas las personas involucradas y su relación organizacional dentro de la unidad de Servicios Escolares.
- Que persona(s) está(n) en contacto directo con los procedimientos de operación de todos los servicios que proporciona la Unidad de Servicios Escolares.
- Identificar las aplicaciones existentes y si éstas satisfacen las necesidades de proceso de información de la Unidad.
- Identificar las áreas ajenas a la Unidad que están involucradas en el uso y distribución de la información que se necesita.
- Los beneficios que puede aportar la implementación del sistema.

De acuerdo a lo anterior, se identificó en primera instancia, al personal que podría proporcionar la lista de procedimientos que habrán de automatizarse, y que además conociera el seguimiento que se le debe de dar a cada uno de los servicios proporcionados, incluyendo todos los procedimientos manuales involucrados en ellos.

Se aplicó la técnica en Grupo en donde el usuario describió la problemática general alrededor de sus funciones así como sus requerimientos.

La mayoría de documentos que emite la Unidad de Servicios Escolares están basados en los datos contenidos en la Historia Académica del alumno. La consulta de estos datos es realizada actualmente en forma manual, lo cual requiere invertir demasiado tiempo en atención personalizada. Por lo tanto, el usuario espera que la implementación del sistema le proporcione lo siguiente:

### Planteamiento de la Problemática y Propuesta de Solución

- Poseer la información necesaria de manera concentrada en una computadora central, de tal modo que le sea posible realizar consultas sobre esta de modo automatizado y consumiendo el mínimo de tiempo, además de garantizar niveles de seguridad en el acceso y actualización de la misma, así mismo garantizar su integridad y confiabilidad.
- Proporcionar a los alumnos información actualizada respecto a su avance académico, de acuerdo al semestre inmediato anterior cursado.

Para tal efecto, el mismo usuario solicitó se implementara dentro del mismo sistema alguna opción que permita capturar mediante un proceso las calificaciones de las actas rezagadas. Esto se hará con el objeto de proporcionar información actualizada en tanto que las actas no entregadas a tiempo son procesadas. Esto implica que la información capturada debe permanecer archivada en forma temporal ya que cada semestre será necesario actualizarla. Dado que este proceso se realizará de manera independiente a la DGAE, la información que se obtenga de este, deberá ser consistente antes y después de que las actas hayan sido procesadas.

- Obtener el número de créditos que corresponden al avance académico del alumno, la razón de esto es saber el número de las materias que han sido acreditadas, el promedio general, la carrera que está cursando o el semestre que está cursando, también se debe de considerar el número de créditos acumulados de acuerdo al plan de estudios que le corresponde cursar.
- El usuario solicita que se automatice de manera primordial todo este proceso, mediante la captura inicial de todos los planes de estudio, almacenados en algún archivo que pueda ser accedido todo el tiempo utilizando alguna estructura de datos permitiéndole al sistema realizar un

## Capítulo II

---

tipo de mapeo entre planes de estudio, obtener asignaturas y créditos equivalentes y finalmente obtener el número real de créditos acumulados. Los planes de estudio que conformarían este archivo son aquellos que van desde 1989 hasta la fecha ya que apartir de esta, se introdujeron los sistemas de cómputo en esta Facultad que vinieron a satisfacer las necesidades de proceso de toda su información.

- Llevar estadísticas por evento de cuales servicios fueron proporcionados, pudiendo generar estadísticas en periodos trimestrales, semestrales y anuales.

La intención de emitir estadísticas es la de proporcionar a la unidad de administración escolar un medio de comparación que la permitan obtener la cantidad de los servicios proporcionados y posteriormente llevar a cabo las consideraciones necesarias para mejorarlos, además de orientar al personal hacia otras funciones que le reporten mayores beneficios a sí misma.

Debemos mencionar que el termino Usuario se refiere a personal administrativo, alumnos y departamentos internos de la FESZ.

A continuación mencionaremos y describiremos algunos de los requerimientos del usuario para el desarrollo del sistema (CONEFESZ).

- Verificar semestralmente si el alumno está o no inscrito y cual es su respectivo horario. Esto implica la existencia de algún archivo temporal que sea actualizado semestralmente con los respectivos horarios.
- Los colores también forman parte importante ya que de hacer una buena selección, el usuario no presentará síntomas de cansancio por el uso del mismo.

*Planteamiento de la Problemática y Propuesta de Solución*

---

- Las pantallas deberán ser legibles, es decir, que los conceptos deberán ser claros y entendibles para el usuario.
- La ayuda en el sistema en forma inmediata ofrecerá una gran seguridad en el usuario, sin tener que distraerse en buscar en manuales muy voluminosos y complejos.
- El sistema debe tener un acceso sencillo sin tener que hacer muchas preguntas al usuario, salvo las necesarias para poder operarlo en una forma clara y concisa.
- El sistema deberá tener capacidad de mantenimiento sencillo para futura actualización, ya que la información que se va a utilizar varía continuamente. Esto proporcionará al usuario información actualizada y veraz.

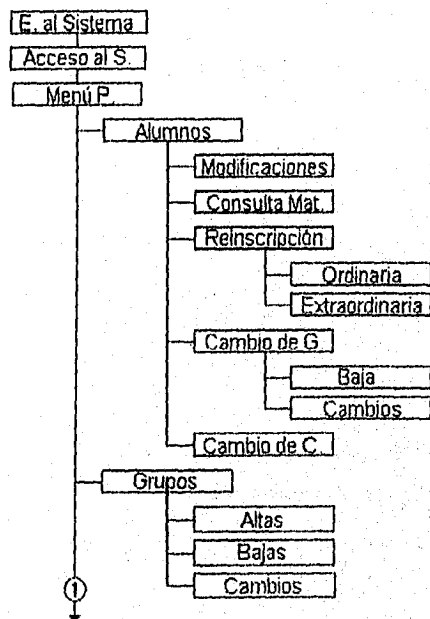
Observadas las condiciones anteriores se optó por utilizar la llamada "programación visual" donde se pretende simplificar al máximo todo el proceso evitando cuanto paso intermedio sea posible. Esta programación visual es atractiva y representa un pequeño reto para el usuario dada la facilidad al utilizarla.

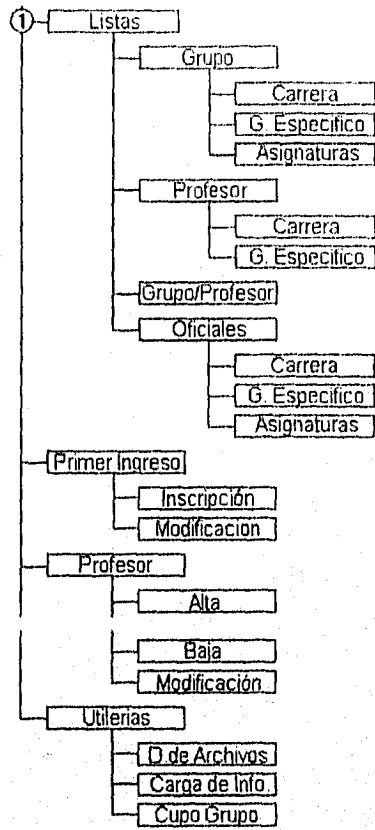


## 2.4. DIAGRAMA DE DESCOMPOSICIÓN FUNCIONAL.

El diagrama de descomposición funcional muestra las partes fundamentales de la estructura del sistema. En él se muestran las diferentes entidades y como podemos llegar a ellas de acuerdo al flujo de información ya sea manual o automatizada, incluyendo procesos y retrocesos de información.

Para una mejor comprensión del sistema a continuación se presentan los procesos esenciales que lo integran. Estos procesos se han estandarizado con la colaboración del personal de Servicios Escolares, dado que ellos serán uno de los usuarios finales





A continuación se presentan las partes esenciales de la estructura del sistema, en donde se lograron identificar los siguientes módulos:

**Módulo de entrada al sistema.** Como el sistema se encuentra corriendo en red es necesario que el administrador de red tenga listo el servidor. El primer nivel de seguridad de los sistemas lo brinda la red. Antes de que un usuario pueda hacer uso de la red, y establecer una sesión en el servidor, debe conectarse a ella desde una estación de trabajo, mediante un login y un password. La red con que se cuenta es Netware 3.11

**E. al Sistema**

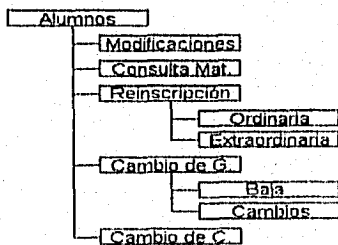
**Módulo de Acceso al Sistema.** El sistema CONEFEZ cuenta con sus propios niveles de seguridad, el acceso a estos dos niveles se realiza a través de una clave de usuario del sistema y de una clave de Administrador del sistema.

**Acceso al S.**

**Módulo del menú Principal.** Este módulo permite el acceso a todos los módulos que se desarrollan en este sistema.

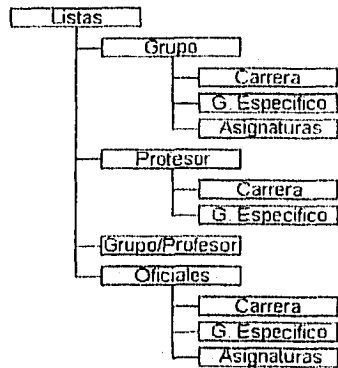
**Menú P.**

**Módulo de Alumnos.** En este modulo se desarrollan la mayoría de Servicios que un alumno solicita a las oficina de Servicios Escolares en su estancia en ésta facultad. Por ejemplo: la actualización de sus datos generales como son: Dirección, Teléfono, etc.

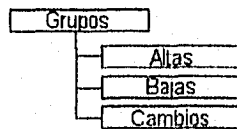


## Capítulo II

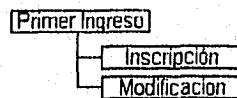
**Módulo de listas.** Este modulo permitirá obtener todos los tipos de listas que se requieren en esta facultad



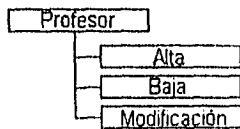
**Módulo de Grupos.** La realización de cambios es algo muy común en las oficinas de Servicios Escolares. En este modulo se detectan los tres tipos de cambios básicos.



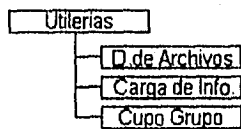
**Módulo de Primer Ingreso:** Este modulo permitirá realizar la Inscripción de Primer Ingreso para los alumnos que se presenten a Servicios Escolares a confirmar su primera inscripción.



**Módulo de Profesor.** Este modulo permite mantener actualizados los datos de los profesores que imparten sus clases en la facultad.



**Módulo de utilerías del sistema.** Este modulo permitirá darle mantenimiento al sistema y solo será accesado por la persona indicada.



El cupo de un grupo lo determina la autoridad correspondiente en teoría el cupo debe ser de 40 alumnos regulares y 10 alumnos irregulares.

## **2.5 OPCIONES DE SOLUCIÓN**

### **2.5.1 Elección del manejador de base de datos.**

De acuerdo a las necesidades del usuario en la elección de la base de datos se deben tomar en cuenta los siguientes parámetros:

- 1) Como se menciona en puntos anteriores, la migración de la base de datos a un **Manejador SYBASE** se realizara en **menos de 2 años**; esto trae como consecuencia que la inversión en una base de datos cuyo costo sea elevado es **inconveniente**.
- 2) El **manejador de bases de datos debe contar con las siguientes características**:
  - **Trabajar en modo multiusuario.**
  - **Tener seguridad y permitir distintos niveles de acceso a la base de datos**
  - **Manejo sencillo**
  - **Permitir conexión con bases de datos externas**
  - **Poder importar datos en ASCII**
  - **Búsquedas eficientes**
- 3) Tomando como **parámetro las bases de datos con que cuenta actualmente la FESZ y el crecimiento anual del 7%**. La **base de datos elegida debe tener capacidad para el manejo eficiente de 400.000 registros**.
- 4) **Disponibilidad en el mercado de manera inmediata.**

## Capítulo II

---

Para la elección de la base de datos se evaluó el siguiente Software:

- Dbase 5.0 for Windows
- Oracle 7 server
- Approach para Windows Ver 2.1
- Paradox para Windows Versión 5
- Access Versión 2.0 para Windows

El análisis se presenta a continuación:

- Dbase 5.0 for Windows

Los programadores de herramientas siempre están alabando las virtudes de la orientación hacia los objetos, pero Dbase 5.0 para Windows fue uno de los pocos productos que la ofreció este año. Sus sólidos modelos de objetos y sus productivas herramientas visuales son un caso de la orientación hacia los objetos. Quizá la más impresionante de sus nuevas características son sus herramientas visuales de dos vías. Estas permiten a los programadores moverse fácilmente para crear una interfaz con pintores visuales y editar una representación de código fuente de la misma interfaz. Los cambios de código fuente son reflejados inmediatamente en el objeto visual y viceversa, los objetos de interfaz se convierten a código en forma incrementada a medida que se guarda el trabajo, lo que permite observar y editar el código en cualquier momento. Esta presentación de un sucesor visual y orientado hacia objetos del Dbase for DOS, marca una vuelta impresionante para este producto clásico de la PC. El costo de este paquete es actualmente de 755 usd. por usuario.

• **Oracle 7 server, versión 7.3 (Oracle Corp.)**

Oracle 7 server, versión 7.3, la versión mas reciente del Oracle 7, ofrece potentes funciones de multiprocesamiento simétrico, (tal vez la tendencia mas comentada en la programación de SQL hoy en día) a las bases de datos de SQL. Aunque muchas bases de datos de SQL actualmente apoyan el hardware SMP de múltiples CPU'S, Oracle 7 se adelanta a su capacidad de ejecutar en paralelo operaciones importantes de bases de datos, incluyendo las indagaciones, las cargas y la creación de índices. Bajo condiciones ideales de laboratorio, hemos visto ciertas operaciones ejecutarse al doble de la velocidad cuando se añade un segundo CPU y recursos de E/S.

El grupo de funciones de la versión 7.3 es como un compendio de todo lo bueno que hay en las actuales bases de datos de SQL. Su inusual modelo de multiversiones concurrentes, donde los lectores no bloquean a los escritores, provee un buen rendimiento en los entornos de transacciones mixtas. Tal pareciera que Oracle 7 define el estándar de alguna forma que los demás quisieran alcanzar. Aunque Oracle 7 se define como uno de los manejadores de bases de datos más eficientes la desventaja es su alto costo de casi 3000 usd.



- **Approach para Windows Ver 2.1 (Lotus Development Corporation)**

**Requerimientos del sistema:** Un sistema pc basado en un 386 o superior, 4MB en RAM y windows 3.1 o posterior. Su buena reputación se debe a su limpieza y facilidad de uso, es excelente para realizar tareas de bases de datos mas comunes, tales como crear formularios de entrada de datos, también provee un acceso rápido a la información existente. Sin embargo, existe una limitante por la falta de herramientas actualizadas y un servicio de programación de aplicaciones mas potentes, la versión 3.0 incorpora mejoras considerables a este respecto y a otros. No presenta un formato nativo de base de datos, pero presenta opción de varios formatos estándares en la industria, posee una útil barra dinámica de estado.

Ofrece conectividad magnífica para examinar e importar datos de formatos de Dbase III+, Dbase IV, FoxPro, Paradox, Oracle, SQL Server y DB2. También puede importar datos de archivos ASCII delimitados y hojas de calculo de Lotus 123 y Excel. Approach no tiene apoyo en sí para crear una llave primaria, pero tiene facilidad para crear macros y el modo de visualización permite ver la apariencia final de su formulario, reporte, etiqueta o carta. En la versión 3.0 incorpora varios diagramas en 3D y un diseño elegante, ideal para usuarios principiantes. Su costo actual es de casi 700 usd.

- **Paradox para Windows Versión 5 (Borland International Inc.).**

Requerimientos del sistema: PC basada en un 386 o posterior 4MB en RAM, 20 MB de espacio en el disco duro y Windows 3.1 o posterior. Posee un intuitivo entorno de trabajo y una rica plataforma de programación, potentes capacidades de indagación que permiten navegar por datos de múltiples tablas y tablas SQL basadas en servidores. Sin embargo no es tan fácil de aprender como sus competidores. Es una potente base de datos por relación, no ofrece suficiente soporte para usuarios remotos en la LAN. Se puede usar un formulario personalizado y permite definir rápidamente búsquedas demasiado complejas y guardarlas para uso posterior y se facilita aun mas con la speedbar. Se pueden crear uniones entre tablas fácilmente. Los resultados de cada búsqueda generan una nueva tabla, pero asume que el usuario entiende la estructura de la base de datos, posee en formato FDB nativo, pero se puede utilizar para crear y manipular bases de datos de Dbase III y IV. Puede utilizar una gran variedad de tipos de campos incluyendo BLOB (Binary Large Objects u objetos binarios grandes), de imagen y objetos OLE (Object Linking and Embedding), también puede importar texto fijo y delimitado y varios formatos de hojas de calculo.

Paradox es una herramienta potente para la creación de aplicaciones. El lenguaje de programación ObjectPAL, es una versión basada en eventos del PAL (Programming Application Languaje o Lenguaje de Programación de Aplicaciones) usado en la versión DOS del producto. El modelo de eventos de paradox es un mecanismo complejo que pasa eventos por varios niveles de objetos y el propio formulario recipiente. No es fácil dominar el sistema, requiere de grandes conocimientos en programación. Posee una gran ventaja, la WorkGroup Edition provee acceso directo a la base de datos SQL de Borland, Microsoft, Oracle y Sybase. El costo de este paquete es de 875 usd.

**Access Versión 2.0 para Windows (Microsoft Corporation).**

Requerimientos del sistema: pc basado en 386sx o posterior, 6MB de RAM, 19 MB libres en disco duro y Windows 3.1 o posterior. Es el producto mas escalable en su clase, implanta un impresionante grupo de características nuevas y mejoras a la facilidad de uso, incluyendo varios wizards nuevos. Tiene el mejor rendimiento en búsqueda, impone reglas de integridad de datos a nivel de bases de datos que le permite determinar como los registros en tablas asociadas se ven afectados cuando se modifica una tabla matriz, reduciendo así la cantidad de código necesaria para construir aplicaciones.

Cuando se crea una base de datos en access todos sus componentes se guardan en un archivo único. MDB, pulsando sobre el botón New se comienza el proceso de crear un objeto del tipo seleccionado y ahí es donde los wizards entran en acción, facilitando la creación de registros funcionales. De los asistentes mas potentes esta el table wizard, que provee mas de cuarenta modelos de tablas personales y de negocios, cada una con su propio juego de campos con formato para usar en su propia base de datos. Tiene una ventana gráfica de relationship que muestra todas las tablas en la base de datos con líneas entre ellas para representar las relaciones activas. Simplemente se arrastra de un campo en una tabla a otro en otra tabla para definir una relación y luego se personaliza el enlace.

Puede usar datos de varias bases de datos como: Dbase III+ y IV, FoxPro 2.0 y 2.5, Paradox 3.X y 4.X, así como Microsoft, Oracle, Sybase y bases de datos SQL. Puede importar todos estos datos, pero también puede acoplarse a las tablas externas y trabajar con sus formatos existentes.

Tiene una gran potencia de búsqueda, se define el criterio en pocos pasos y es fácilmente modificable, soporta facilidades de macros, así como búsquedas en

tablas cruzadas, registros duplicados, etc. También presenta una potente actualización de tablas cuando se modifica el resultado de una búsqueda, característica todavía ausente en las demás bases de datos. Los wizards de botones de comandos frecuentemente determinan de antemano las respuestas a los eventos del ratón. Contiene características que permiten generar rápidamente reportes y formularios. Posee apoyo para OLE 2.0, que se puede incluir en formularios y reportes, así como para BOLB's. Los wizards son módulos orientados a ciertas tareas, que hacen una serie de preguntas y luego generan los objetos apropiados, según las respuestas. También posee un lenguaje de programación ObjectPal. Es el más barato del mercado, pudiéndose adquirir en 100 usd De acuerdo a los requerimientos del usuario, se evaluaron los puntos mostrados en la tabla 2.5.1, misma que muestra que por sus características de bajo costo, disponibilidad, facilidad de manejo, portabilidad (compatible con otras bases de datos) y acceso a bases de datos externas, la mejor elección para el desarrollo del sistema es Microsoft Access.

## Capítulo II

---

### Elección Del Lenguaje De Programación.

En base al tipo de trabajo a realizar y tomando en cuenta la capacidad de Microsoft Windows, así como la previa selección de Access for windows, se ve en la necesidad de encontrar un lenguaje de programación en y para la plataforma gráfica, así como de fácil manejo de los objetos. Por lo cual se tomo la determinación de trabajar con lenguaje notablemente mas amigable que Visual C++ y este fue Visual Basic 3.0 para Windows. Ya que es un lenguaje de programación con herramientas que permiten el manejo de bases de datos, archivos, gráficas, controles tridimensionales, y animados, además de poder crear herramientas de control personalizadas, todo esto en conjunto con la capacidad de generar programas ejecutables como producto final.

Este es un lenguaje basado en el basic antiguo y por lo tanto presenta muchas similitudes, pero mejoradas, es por eso que ahora con un tipo de programación mas sutil se pueden lograr resultados mas flexibles; es un lenguaje orientado a determinados eventos, es decir, si el usuario hace "Click" en ese botón, se ejecutara este sector de códigos del programa, si maximiza o minimiza esta ventana, se ejecutará este otro sector de código del programa.

Otra de las grandes ventajas de Visual Basic es el manejo de ODBC que permite la conexión a bases de datos externas. Esta característica permitirá utilizar la aplicación aún cuando las bases de datos se emigren a SYBASE sin realizar cambios al sistema.

	D B A S E 5.0	O R A C L E 7	A P R O A C H	P A R A D O X	A C C E S S
<b>COSTO EN U.S.D.</b>	755	3000	700	875	100
<b>MULTIUSUARIO</b>	S	S	S	S	S
<b>DIFERENTES NIVELES DE ACCESO</b>	N	S	N	N	S
<b>MANEJO SENCILLO</b>	S	N	N	N	S
<b>CONEXION A BASES DE DATOS EXTERNAS</b>	N	S	S	S	S
<b>IMPORTAR DATOS EN ASCII</b>	S	S	S	S	S
<b>BUSQUEDAS EFICIENTES</b>	N	S	N	S	S
<b>DISPONIBILIDAD INMEDIATA</b>	N	N	N	S	S
<b>ORIENTADO A OBJETOS</b>	S	N	N	N	N
<b>MULTIPROCESAMIENTO</b>	N	S	N	N	N
<b>CPU</b>	486	486	486	386	386
<b>MEMORIA RAM (MB)</b>	16	32	4	4	6
<b>ESPACIO EN DISCO DURO (MB)</b>	30	60	20	20	19
<b>VERSION WINDOWS</b>	95	-	3.1	3.1	3.1

TABLA 2.5.1 COMPARATIVO DE BASES DE DATOS

**“CAPÍTULO III**  
**“DISEÑO E**  
**IMPLEMENTACIÓN**  
**DEL SISTEMA**

**OBJETIVO:**

**Desarrollar e implementar el sistema  
para resolver la problemática de la  
administración escolar en la Facultad de  
Estudios Superiores Zaragoza.**

---

### 3.1 ESPECIFICACIONES DEL ANÁLISIS Y DISEÑO (ESTABLECIMIENTO DE ESTÁNDARES).

En la Facultad de Estudios Superiores Zaragoza se tiene como objetivo formar profesionistas de una excelente calidad y eficiencia. Para ello se le ofrecen a la población estudiantil, actualmente seis carreras a nivel licenciatura, las cuales han sufrido modificaciones en sus planes de estudio a partir de 1976, que es el año en el cual la institución comenzó a brindar sus servicios.

A continuación se describen las características de cada licenciatura a partir del año de 1976:

Carrera	Desc-Carrera	Duración	Plan	Créditos	
				oblig.	optativos
503	Biólogo	9	22	441	0
505	Ingeniería Química	9	21	384	0
505	Químico Farmacéutico Biólogo	9	24	384	0
512	Técnico en Enfermería	6	01	281	0
512	Médico Cirujano	0	06	0	0
512	Médico Cirujano Modular	8	07	446	0
512	Médico Cirujano	9	08	446	0
514	Cirujano Dentista	8	21	369	0
519	Psicología	9	21	320	0

La duración de cada carrera está manifestada en semestres.



### Capítulo III

En la tabla anterior se puede apreciar que las claves de las carreras 505 y 512 se repiten, esto es debido a que han sufrido cambios en sus planes de estudio. Actualmente las carreras que se imparten en esta Facultad a nivel licenciatura son: Biólogo plan 22, Químico farmacéutico Biólogo plan 24, Técnico en enfermería plan 01, Médico Cirujano plan 08, Cirujano Dentista plan 21 y Psicología plan 21. Los planes de estudio que se cubren actualmente se enlistan en el anexo A.

Para lograr el objetivo de esta institución es necesario coordinar las acciones de cada uno de los participantes en este proceso. Esta función la realizan las Oficinas de Servicios Escolares, manifestándose así como un participante más. Así pues se identifican los siguientes entes:

- DGAE.
- Profesores.
- Alumnos y Exalumnos.
- Jefaturas de Carreras.
- Servicios Escolares.

Analizando cada uno de estos elementos, desde el punto de vista administrativo y de control escolar tenemos que sus acciones más representativas se pueden enlistar de la siguiente forma:

#### **DGAE (Dirección General de Administración Escolar).**

##### **Solicita a Servicios Escolares:**

- Cupo por carrera primer ingreso.
- Relación de grupos-profesor.
- Datos actualizados de los alumnos.
- Calificaciones.
- Planes de estudio de cada carrera.

**Proporciona a Servicios Escolares:**

- Relación de alumnos primer ingreso.
- Historiales académicos de alumnos.

**Alumno.**

**Solicita a Servicios Escolares:**

- Inscripción de primer ingreso.
- Reinscripción semestral.
- Inscripción extraordinaria.
- Cambio de grupo.
- Baja de alumno en un grupo.
- Actualización de datos personales.
- Solicitud de informes.
- Consultas.
- Historia académica.
- Revisión de historia académica.
- Revisión de estudios.
- Carrera simultánea.
- Cambio interno de carrera.
- Cambio externo de carrera.

**Proporciona a servicios escolares**

- Identificación vigente.
- Recibos de pago correspondientes.
- Formatos previamente solicitados, con sus datos y autorizaciones requeridas.

### **Capítulo III**

---

#### **Profesores**

##### **Solicita a Servicios Escolares:**

- Listas de grupos en los que impartirá su cátedra.
- Actas de las materias para evaluar a sus alumnos.

##### **Proporciona a servicios escolares**

- Identificación vigente.
- Actas de las materias ya evaluadas y firmadas.

#### **Jefaturas de Carreras**

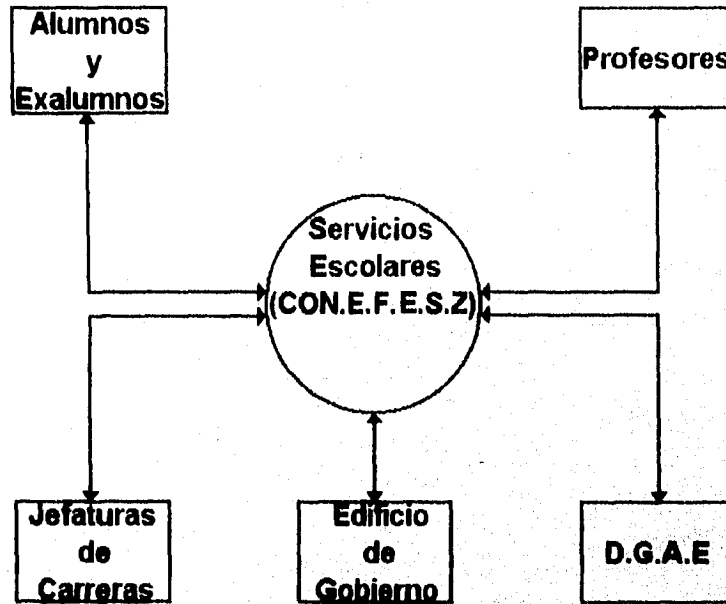
Cada carrera es coordinada por una jefatura de carrera. Esta jefatura (para nuestro objetivo) proporciona semestralmente a la oficina de Servicios Escolares los siguientes documentos:

- Plan de estudios actual de dicha carrera.
- Cupo máximo de alumnos de primer ingreso y reingreso.
- Listas de grupos disponibles para los alumnos de primer ingreso.
- Listas de grupos disponibles para la reinscripción de los alumnos de la carrera en sus distintos semestres.

#### **Servicios Escolares**

Dado que nuestro interés está enfocado en la automatización de la mayoría de procesos que se realizan en estas oficinas, a continuación presentamos un diagrama de contexto que nos muestra una visión general de lo significativo que es el sistema de computo para la administración y control escolar, que facilitará la operatividad de estas oficinas y algunos departamentos de esta Facultad en el manejo de su información.

La elaboración del sistema se ha podido establecer en base al análisis de la información que se tiene, los estándares de los archivos que se utilizarán y estos se muestran en el siguiente esquema:



Dentro de este esquema tendremos archivos maestros como son: Carrera, Asign, Grupo, Alumno, Historia, Inscip, Profesores y archivos auxiliares como son: Cambios, Telu

Estos últimos se requieren para mantener actualizados los anteriores.

A continuación se muestra el esquema global de la base de datos, que es la culminación del modelo conceptualizado en la programación.

Capítulo III

ALUMNO:

Alumno.	Ingreso.	Nación.
Cuenta.	Origen.	Exalum.
Carrera.	Dirección.	Tel.
Plantel.	Folio.	CP.
Insc.	Comprobante.	Esc-Proc.
Promedio.	Sexo.	

ASIGN.

Asign.	Carrera.	Créditos.
Desc-Asign.	Semestre.	Seriación.
Plan.		

CAMBIO.

Cuenta.	Cambio.	Fecha.
Origen.	Destino.	RFC.
Carrera.	Asign.	

CARRERA.

Carrera.	Desc-Carr.	Duración.
Plan.	Oblig.	Dpta.

GRUPO.

Grupo.	Asign.	RFC.
Cupo.	Plan.	RFC-Asis.
Carrera.		

HISTORIA.

Cuenta.	Asign.	Grupo.
Plantel.	Período.	Calif.
Acta.	Num-Ord.	Num-Ext.

INSCRIP.

Cuenta.	Grupo.	Asign.
Tipo.	Carrera.	

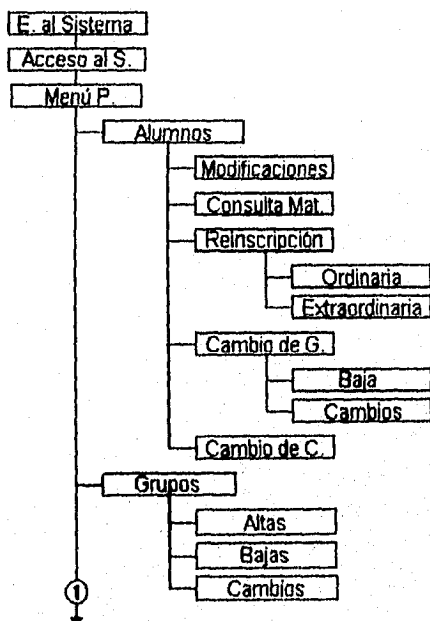
PROFESOR.

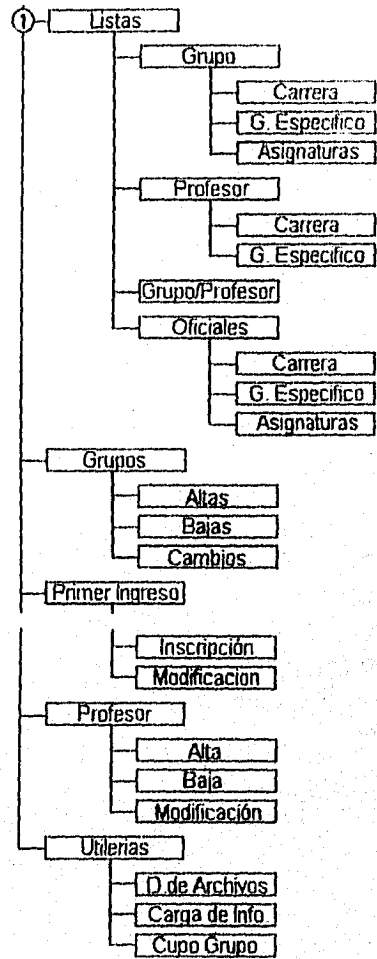
RFC.	Profesor.
------	-----------

TALU.

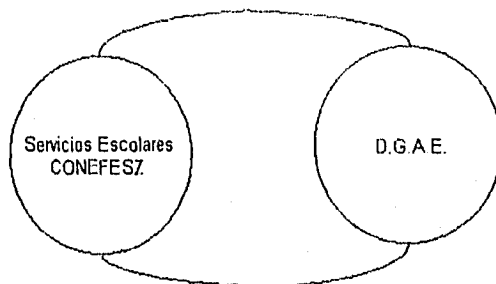
Cuenta.	Alumno.	Esc-Proc.
Sinuso.	Carrera.	Grupo.
Plan.	Comprobante.Domicilio.	
CP.	Teléfono.	Nac.
Sexo.	Nacion.	Inscrito.
Plantel.	Promedio.	

A continuación se muestra la forma en que se integran las posibilidades reales del sistema. Como se puede apreciar cada modulo tiene una total independencia en su programación. Esta característica permite utilizar la programación de Visual Basic que tiene como atributo la programación por eventos.





**INTEGRACIÓN MASIVA DE INFORMACIÓN**



Cada semestre académico se lleva a cabo un ciclo de actualización de información entre la Oficina de Servicios Escolares y la D.G.A.E., este ciclo de ahora en adelante lo realiza CONEFESZ (Control Escolar Facultad de Estudios Superiores Zaragoza) el cual consiste a grandes rasgos en lo siguiente:

1.- Preparación del sistema antes de la inscripción de cada período.(semestral).

• Depuración de archivos

**INSCRIP** (borrar la inscripción del periodo anterior)

**HISTORIA** (carga historia procedente de C.U.)

**CAMBIOS** (borrar toda la información existente)

**PROFESORES** (checa dependiendo baja/alta/modificación)

**GRUPOS** (asignación de cupos, y profesores a grupos)

**ASIGN** (checa clave de la asignaturas)

2. - Arranque del sistema

Inscripciones primer ingreso

Inscripciones ordinarias



### Capítulo III

---

#### 3. Termina el ciclo del sistema

- Manda información o resultados del evento a la D.G.A.E.  
Inscripciones, Cambios, etc.

A Continuación describimos el proceso entre CONEFESZ Y DGAE para la inscripción de primer ingreso.

#### 1.- DGAE. envía a CONEFESZ.

- Alumnos por carrera
- Grupos de carreras
- Planes de estudios vigentes

#### 2.- CONEFESZ envía a DGAE.:

- Datos generales de alumnos, inscripciones y cambios

#### 3.- CONEFESZ:

- Realiza la inscripción de los alumnos que se presenten
- Manda a DGAE los resultados del evento:
  - Inscripciones:
    1. 1er Ingreso.
    2. Ordinarias.
    3. Extraordinarias.
  - Cambios:
    4. Grupo.
    5. Carrera.
  - Alumnos:
    6. Actualización de datos.

### **3.2 DIAGRAMAS DE FLUJO DE DATOS.**

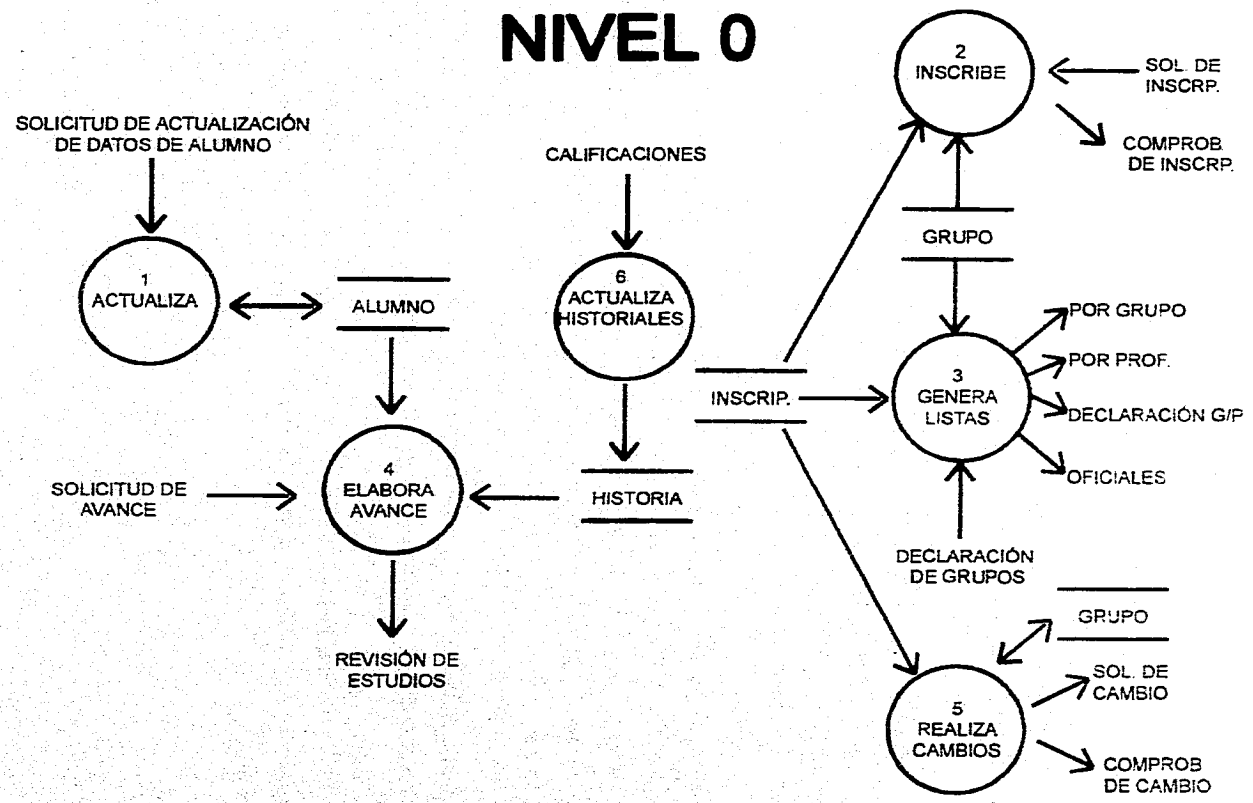
Los diagramas de flujo de datos describen los procesos que realiza el sistema, las entradas que recibe y las salidas que proporciona. Es una herramienta que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por flujos y almacenes de datos.

El punto de partida de los diagramas de datos es la creación del nivel 0, el cual muestra una vista global del sistema a través de burbujas que representan las actividades identificadas en el modelo ambiental; flujos de datos que muestran el movimiento de datos y almacenes que representan los datos que el sistema debe conservar.

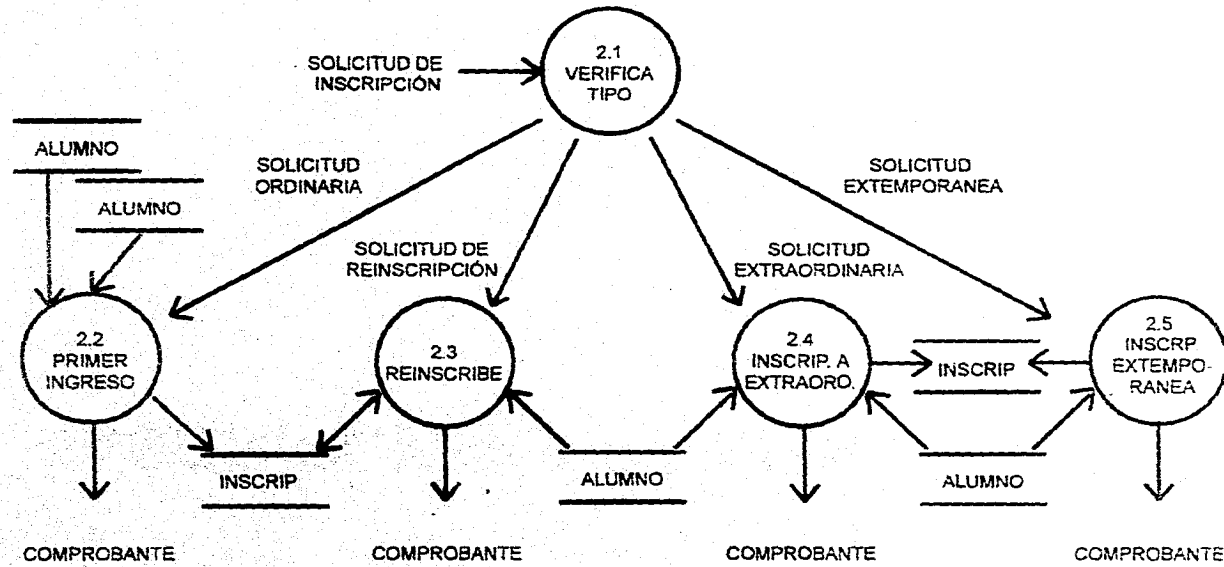
El detalle de cada uno de los procesos definidos en el nivel 0, se lleva a cabo a través de la elaboración de los niveles siguientes de los diagramas de flujo de datos.

Cada una de las burbujas de los diagramas es identificada mediante un número consecutivo, sin embargo dicha numeración no implica una secuencia lógica de actividades.

# NIVEL 0

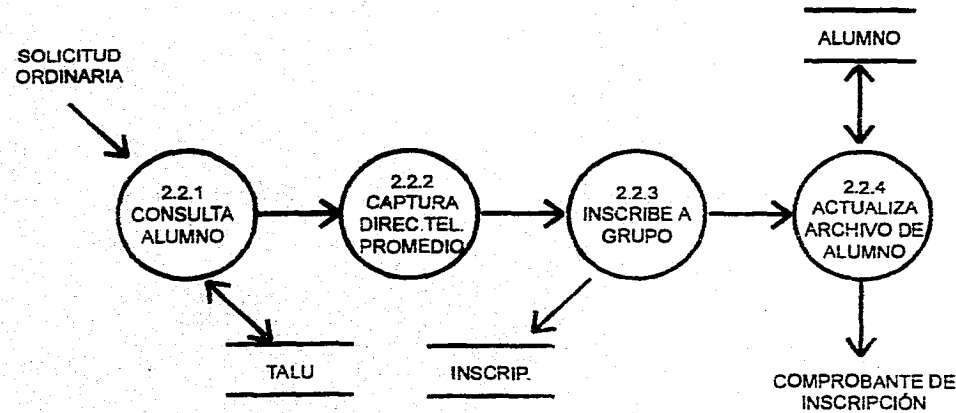


# INSCRIPCIÓN

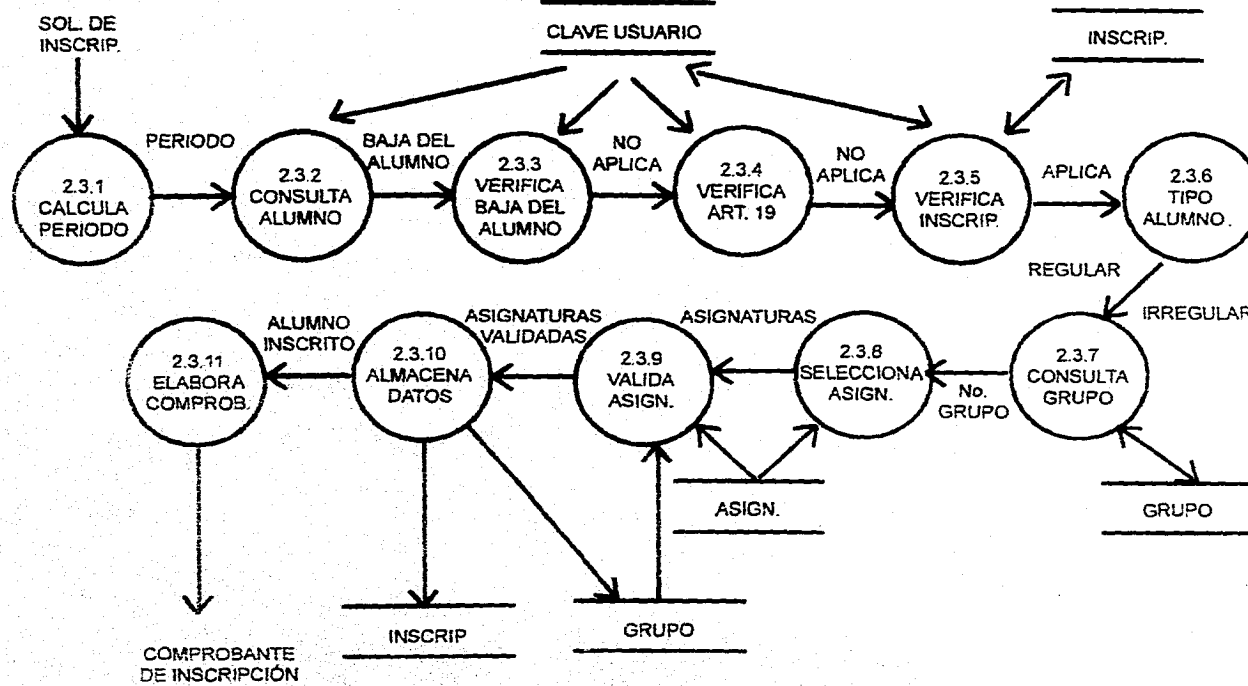


# INSCRIPCIÓN A PRIMER INGRESO

236



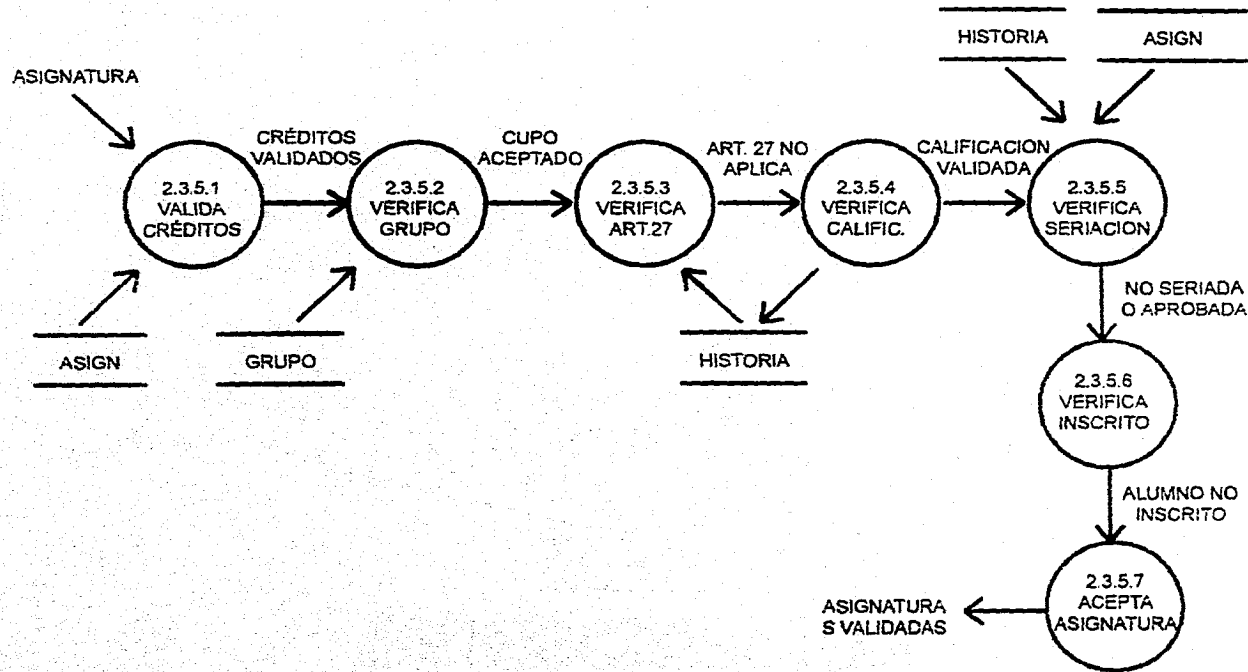
# REINSCRIPCIÓN



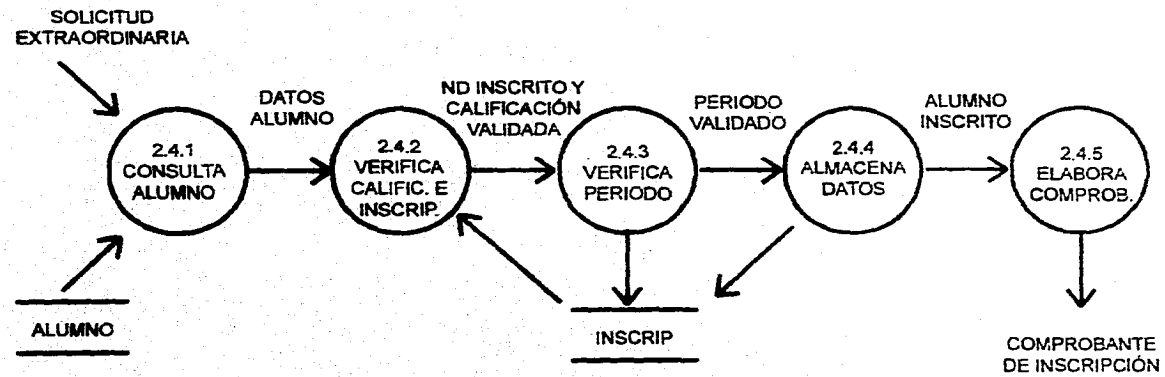
237

Desarrollo e Implementación del Sistema

# VALIDA ASIGNATURA



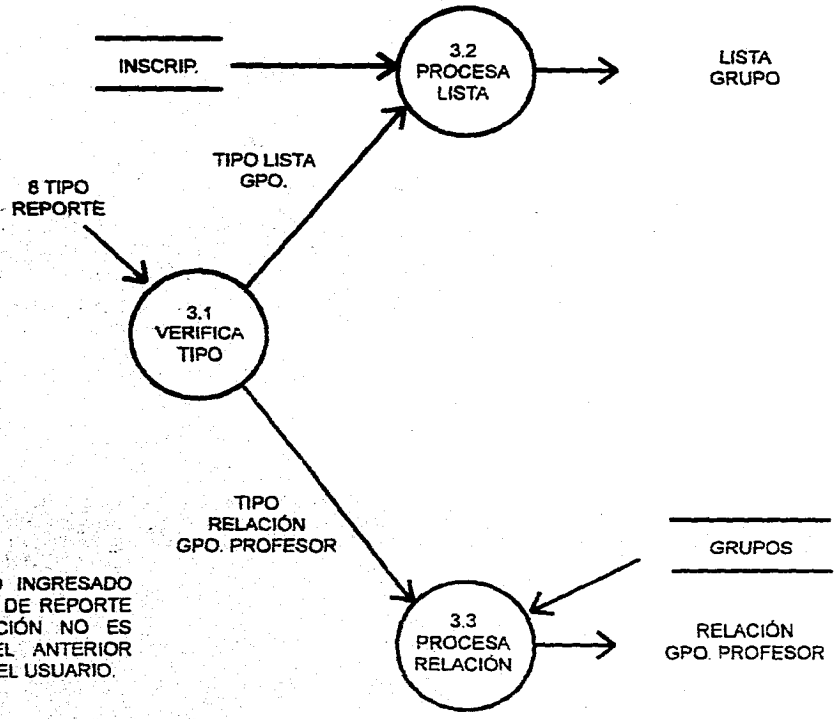
# INSCRIPCIÓN A EXTRAORDINARIO





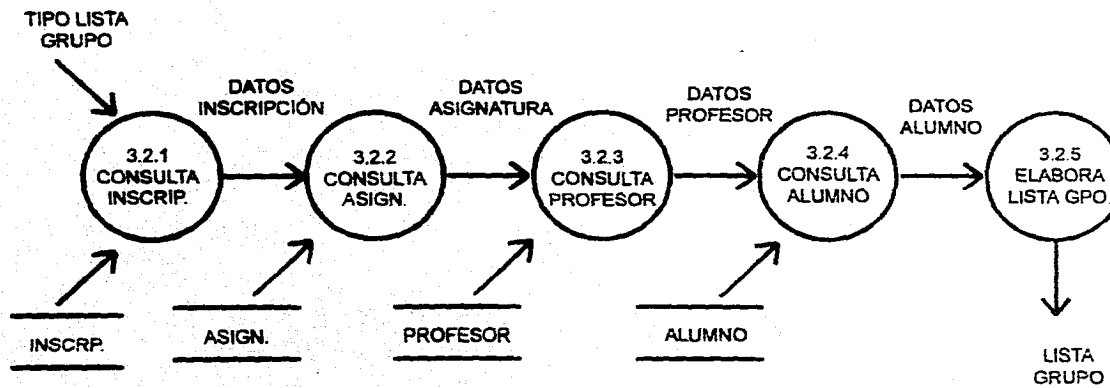
# GENERA LISTAS

240

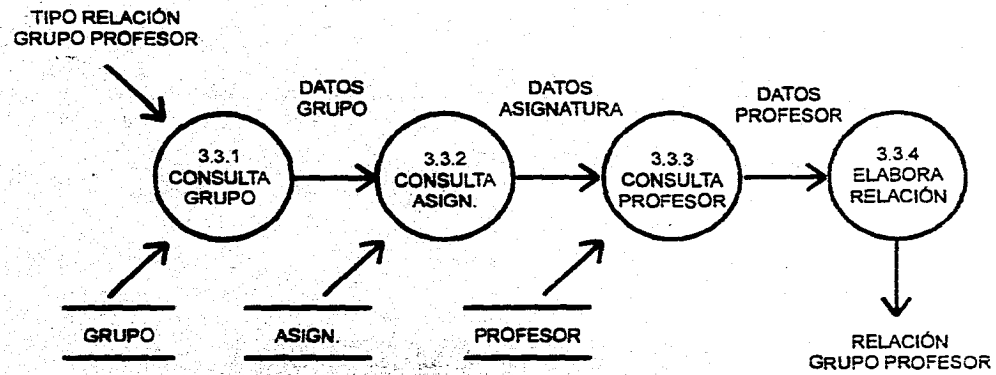


8 TIPO REPORTE: FLUJO INGRESADO QUE DETERMINA EL TIPO DE REPORTE A IMPRIMIR. SU DEFINICIÓN NO ES ESCENCIAL EN EL NIVEL ANTERIOR PARA LA COMPRESIÓN DEL USUARIO.

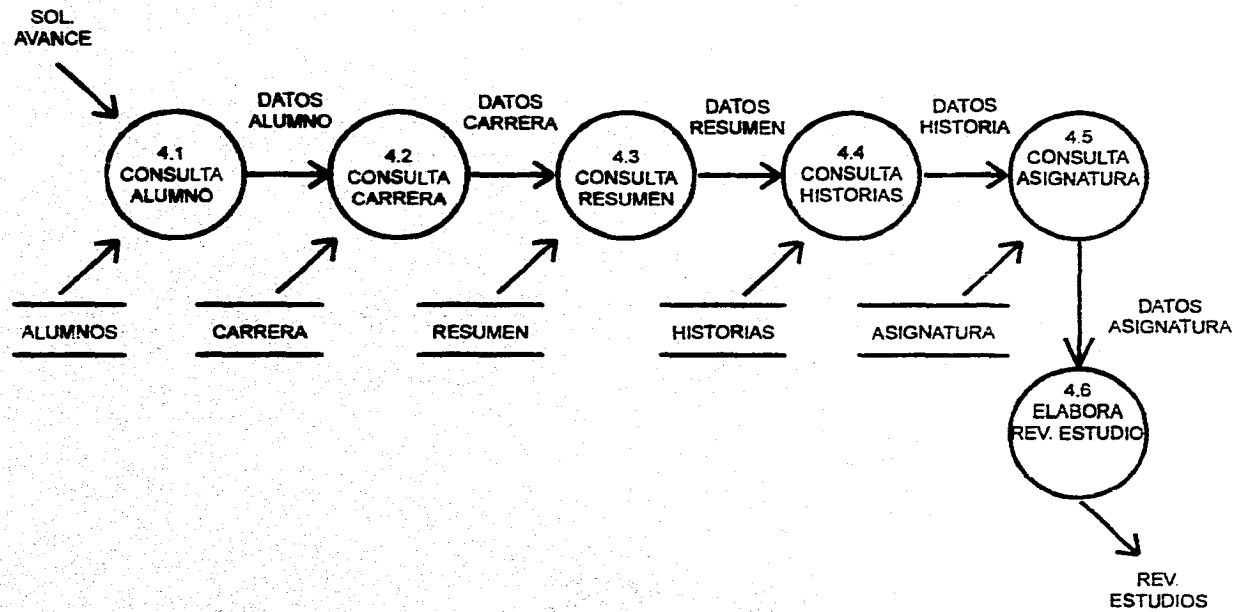
# PROCESA LISTA



# PROCESA RELACIÓN

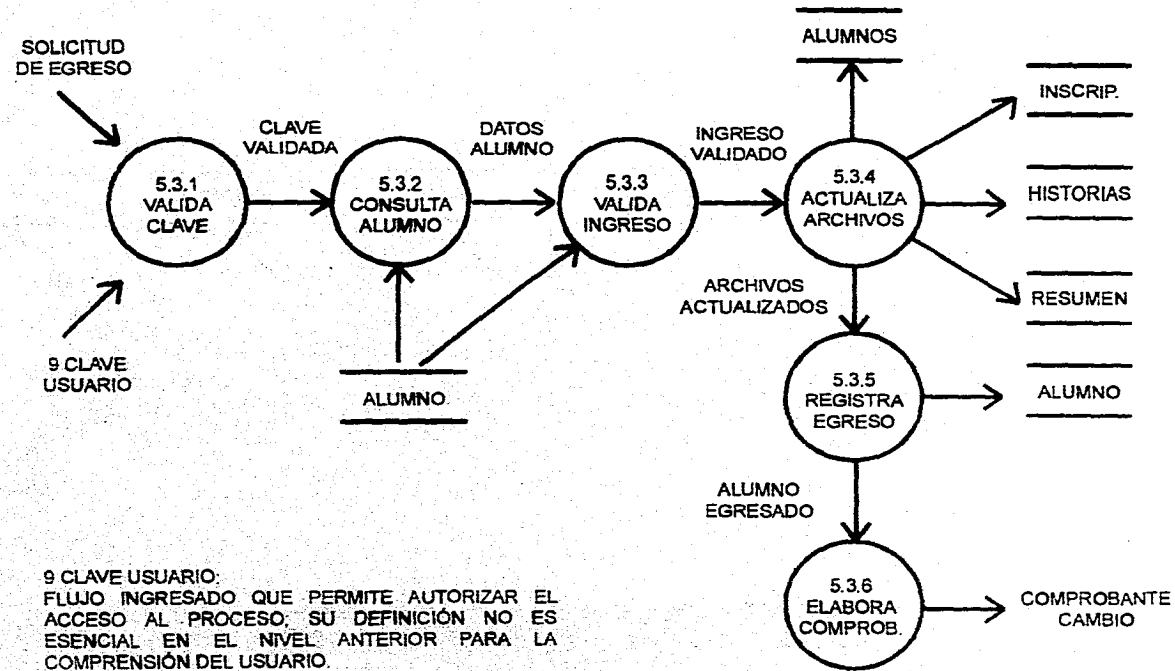


# ELABORA AVANCE

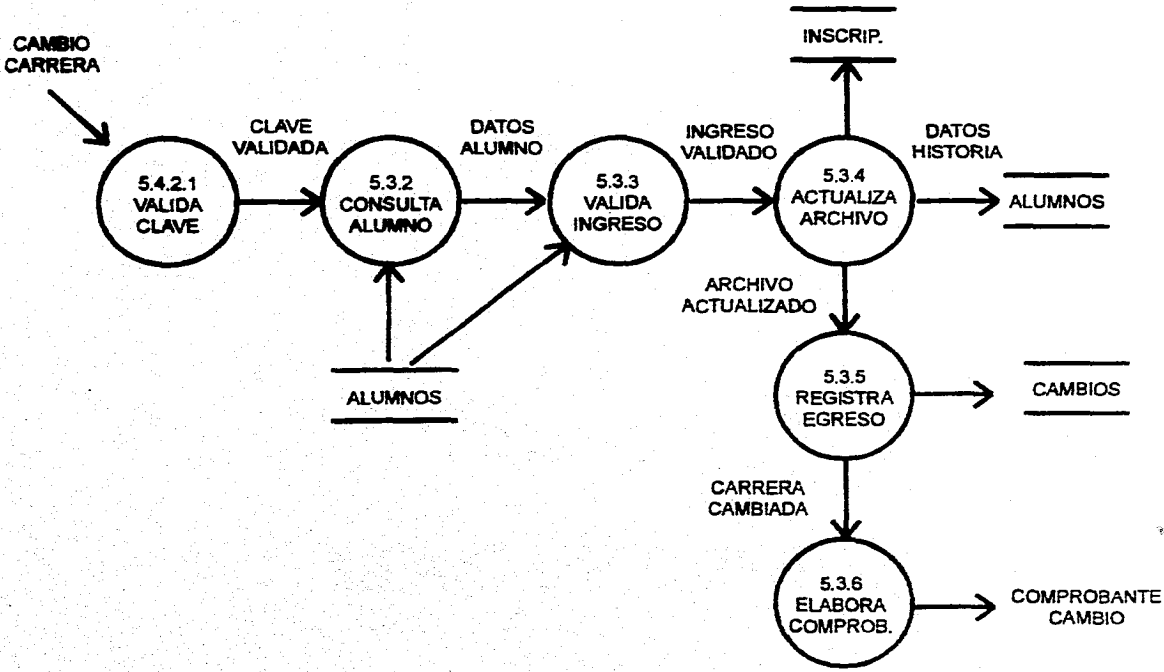


243

# BORRA



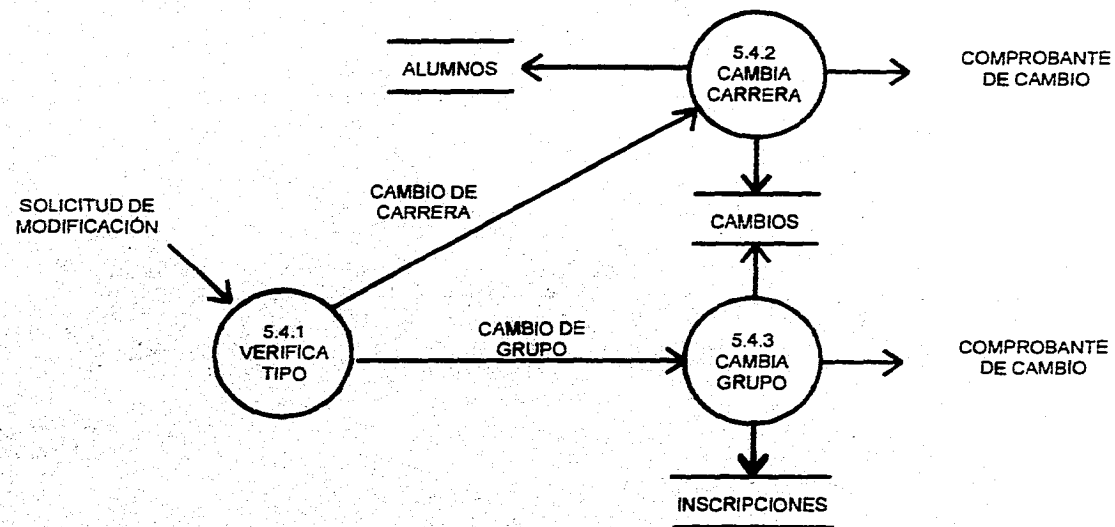
# CAMBIA CARRERA



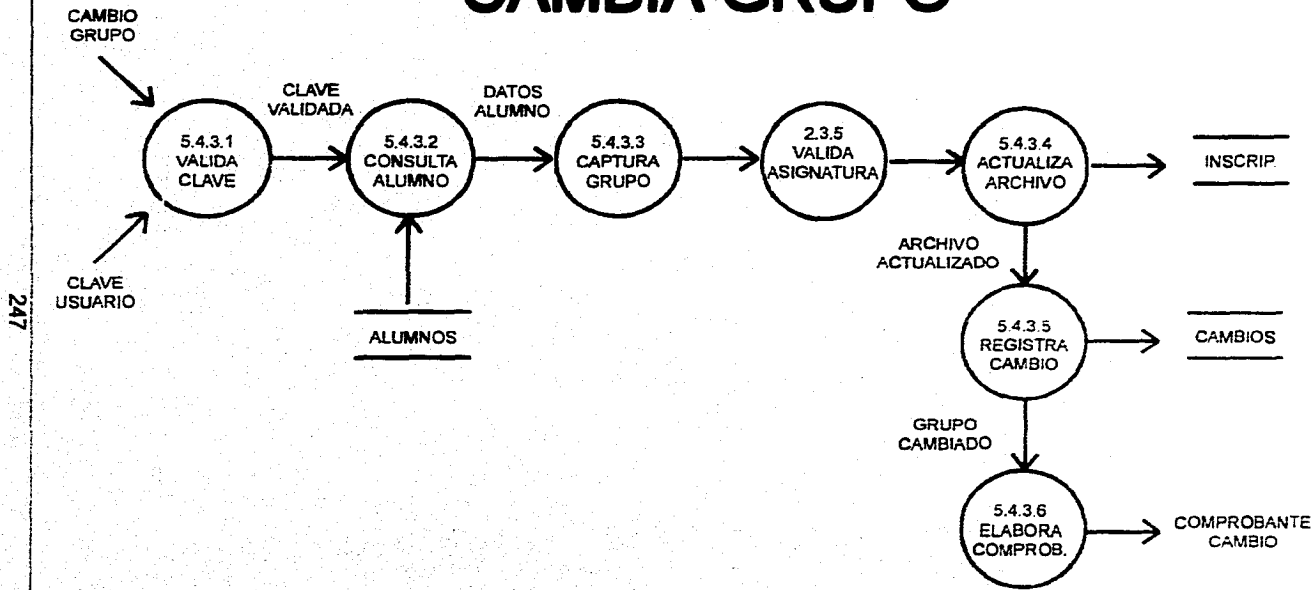
245

Desarrollo e Implementación del Sistema

# REALIZA CAMBIO



# CAMBIA GRUPO

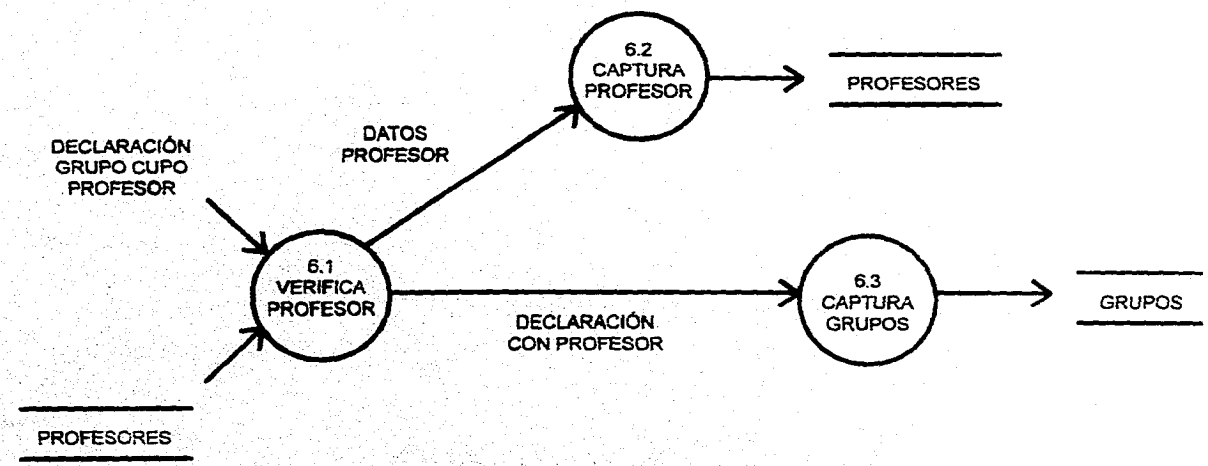


247

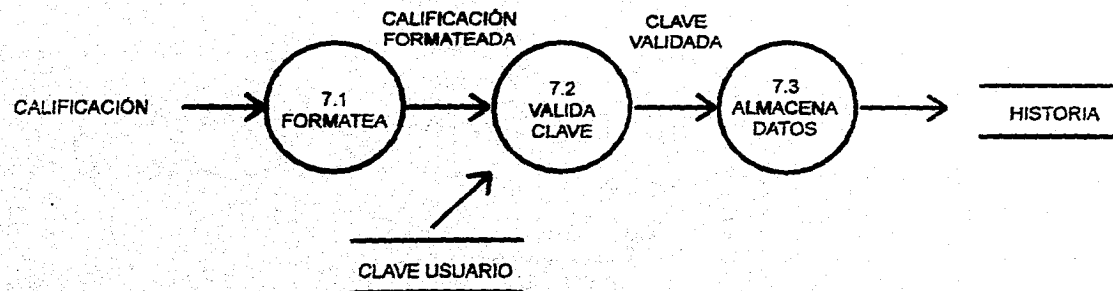


# REGISTRA GRUPOS

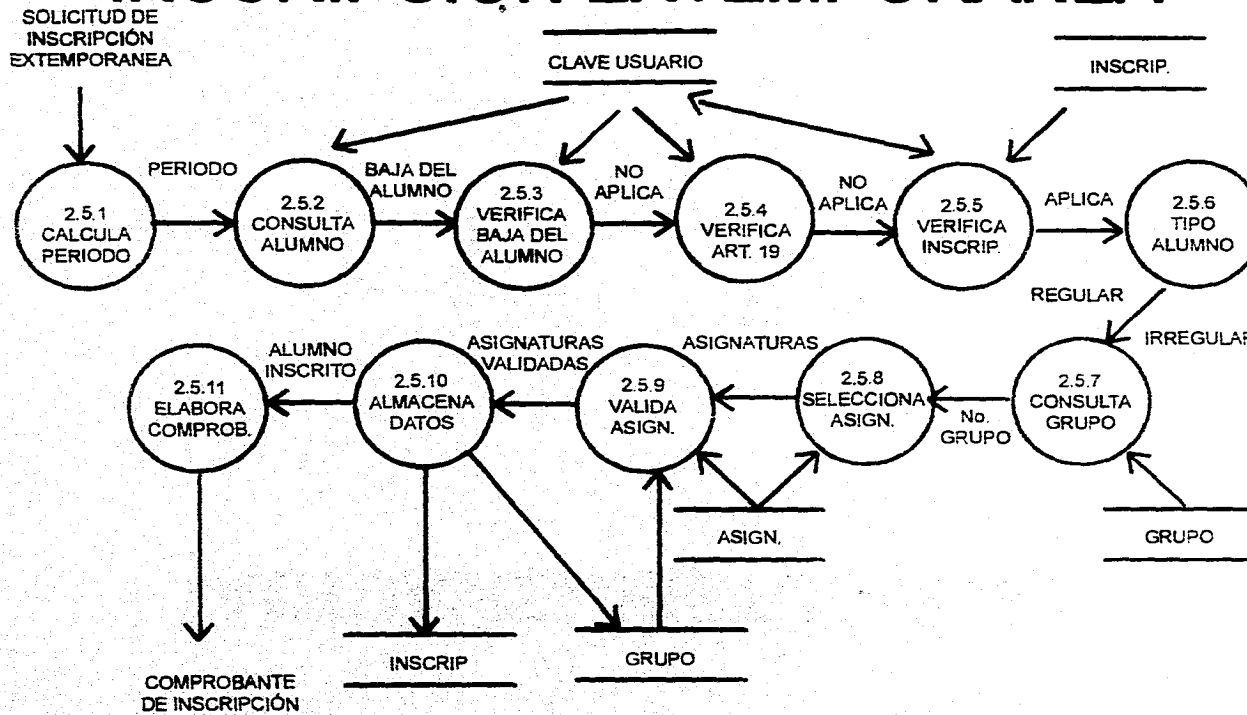
248



# ACTUALIZA HISTORIALES



# INSCRIPCIÓN EXTEMPORANEA



### 3.3 DICCIONARIO DE DATOS.

El Diccionario de Datos (DD) se utiliza para definir el significado, uso, características y otros datos relevantes de todos los data items, campos, entidades, referencias cruzadas y las relaciones que existen entre ellos. El DD es una Base de Datos que contiene datos acerca de los datos. Es importante señalar que el DD, se tiene que analizar para saber como esta formada la Base de Datos en su estructura.

A continuación se detalla el DD del sistema en cuestión:

**Tabla: ALUMNO  
CATALOGO DE ALUMNOS**

Orden	Campo	Tipo	Figs	Formato	Inicial
1	ALUMNO	char	lm	x(32)	
10	CUENTA	char	lm	99999999	
20	CARRERA	inte	i	999	0
25	PLANTEL	inte		>>9	0
40	INGRESO	inte		99	0
50	NACION	inte		9	0
55	MORIGEN	inte		>9	0
58	EXALUM	inte		99	0
60	SEXO	char		X	
70	NACIM	date		99/99/99	?
80	DIRECCION	char		x(67)	
90	TEL	char		9999999	
100	FOLIO	inte		>>>>9	0
110	CP	inte		>>>>9	0
130	INSC	inte		99	0
140	COMPROBANTE	inte		>>>>9	0
150	ESC PROC	char		x(10)	
180	PROMEDIO	dec12		>9.99	0

**Tabla: ASIGN**  
**CATALOGO DE ASIGNATURAS**

Orden	Campo	Tipo	Figs	Formato	Inicial
10	ASIGN	inte	i	9999	0
20	CARRERA	inte	i	999	0
30	DESC_ASIGN	char		x(25)	
40	CREDITOS	inte		>>9	0
50	SEMESTRE	inte		99	0
60	SERIACION	inte		9999	0
70	PLAN	inte		99	0

**Tabla: CAMBIO**  
**RELACION DE CAMBIOS REALIZADOS**

Orden	Campo	Tipo	Figs	Formato	Inicial
10	CUENTA	char	lm	99999999	
20	CAMBIO	inte	i	999	0
30	FECHA	date	i	99/99/99	?
40	ORIGEN	char		x(6)	
50	DESTINO	char		x(6)	
60	RFC	char		AAA999999xxx	
70	CARRERA	inte		999	0
80	ASIGN	inte		9999	0

**Tabla: CARRERA**  
**CATALOGO DE CARRERAS**

Orden	Campo	Tipo	Figs	Formato	Inicial
10	CARRERA	inte	i	999	0
20	DESC_CARR	char		x(30)	
30	DURACION	inte		9	0
40	PLAN	inte	i	99	0
50	OBLIG	inte		>>9	0
60	OPTAT	inte		>>9	0

**Tabla: GRUPO**  
**CATALOGO DE GRUPOS**

Orden	Campo	Tipo	Flgs	Formato	Inicial
10	GRUPO	char	i	x(4)	
20	ASIGN	inte	i	9999	
30	RFC	char	i	AAAA999999xxx	
40	CUPO	inte		>>9	0
50	RFC_ASIS	char		AAAA999999xxx	
60	CARRERA	inte			999
62	PLAN	inte			99
70	CUPOI	inte		>>9	0

**Tabla: HISTORIA**  
**ARCHIVO PARA HISTORIAS ACADÉMICAS DE LOS ALUMNO.**

Orden	Campo	Tipo	Flgs	Formato	Inicial
10	CUENTA	char	im		99999999
20	ASIGN	inte	i	9999	0
30	GRUPO	char	i	X(4)	
40	PLANTEL	inte	i	999	0
50	PERIODO	char		999	
60	CALIF	char		X(2)	
70	ACTA	inte		>>>>>9	0
80	NUMORD	inte		>9	0
90	NUMEXT	inte		>9	0

**Tabla: INSCRIP**  
**DETALLADO DE INSCRIPCIONES**

Orden	Campo	Tipo	Ext	Flgs	Formato	Inicial
10	CUENTA	char		im	99999999	
20	GRUPO	char		i	X(4)	
30	ASIGN	inte		i	9999	0
40	TIPO	inte			9	0
50	CARRERA	inte		i	999	0

**Tabla: PROFESOR  
CATALOGO DE PROFESORES**

Orden	Campo	Tipo	Ext	Flgs	Formato	Inicial
10	RFC	char		i	!!!I999999xxx	
20	PROFESOR	char		i	x(32)	

**Tabla: TALU  
TEMPORAL DE ALUMNOS**

Orden	Campo	Tipo	Ext	Flgs	Formato	Inicial
10	CUENTA	char		im	99999999	
20	ALUMNO	char		im	x(32)	
30	ESCPROC	char			X(10)	0
40	SINUSO	char			x(4)	
50	CARRERA	inte		i	99999	0
60	GRUPO	char			x(4)	
70	PLAN	inte			99	0
80	COMPROBANTE	inte			>>>>9	0
90	DOMICILIO	char			x(62)	
100	CP	inte			>>>>9	0
110	TELEFONO	char			9999999	
130	NACIM	date			99/99/99	?
140	SEXO	char			X	
150	NACION	inte			9	0
160	INSCRITO	inte			9	0
170	PLANTEL	inte			>>9	0
180	PROMEDIO	dec12			>9.99	0

**Nota:**

i = Llave primaria.

im = Llave foránea.

## DIAGRAMA ENTIDAD-RELACIÓN

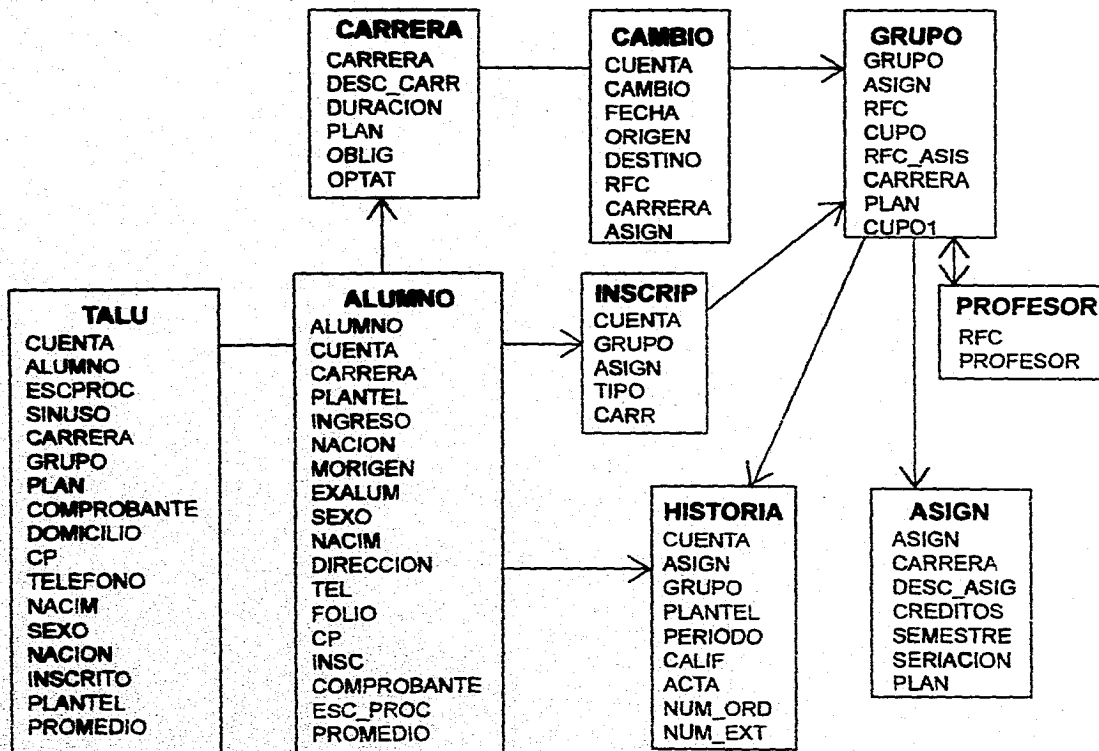


Figura 3.4.1 Diagrama Entidad-Relación.



### **3.5 DISEÑO E IMPLEMENTACION DE LOS DIVERSOS MODULOS DE PROGRAMACION EN VISUAL BASIC 3.0**

La siguiente fase en el desarrollo del sistema es la programación, la cual "involucra la escritura de instrucciones en algún lenguaje de programación para implantar lo que el análisis ha especificado y el diseñador ha organizado en módulos". En el desarrollo de CONEFESZ se utilizó el lenguaje de programación Visual Basic 3.0.

El sistema se dividió en los siguientes módulos, denominados en Visual Basic como "formas":

- **ACCESO.FRM**
- **INICIO.FRM**
- **ALUMNOS.FRM**
- **REINSCRI.FRM**
- **CCARRERA.FRM**
- **CGRUPO.FRM**
- **GRUPOS.FRM**
- **MATERIAS.FRM**
- **PROFESOR.FRM**
- **PRIINGRE.FRM**
- **REPORTES.FRM**
- **FESZ.BAS**

A continuación se resumen las funciones de cada módulo, el código completo de cada módulo puede ser consultado en el Apéndice "E".

### **Capítulo III.**

---

#### **Módulo ACCESO.FRM:**

Contiene la carátula de presentación. Permite el control de acceso al sistema por medio de un Password. Este módulo identifica al usuario como "Administrador" u "Operador" de acuerdo a la clave tecleada.

#### **Módulo INICIO.FRM:**

Contiene el menú principal.

#### **Módulo ALUMNOS.FRM**

Muestra los datos del Alumno Seleccionado. A través de botones nos permite llamar a los módulos de Inscripción (REINSCRIP), Cambio de Carrera (CCARRERA), Cambio de grupos (CGRUPO) y Consulta de Materias (MATERIAS).

#### **Módulo REINSCRIP.FRM**

Este se podría decir que es el módulo más importante. En este se realiza la selección de las Materias a las que el alumno se inscribira. Y se realizan las Validaciones necesarias (Materia ya se aprobo?, Materia ya se curso 2 veces?, Se excede de créditos?, El alumno tiene derecho a inscripción, el grupo tiene cupo?). Como se muestra en la siguiente subrutina:

```
Sub bselecciona_Click () ' Realiza Inscripción
    GRID1.Col = 1
    vassign = GRID1.Text
    GRID1.Col = 2
    VDESCRIP = GRID1.Text
    GRID1.Col = 3
    vcreditos = GRID1.Text
    If checa_aprobada(vassign) Then
        If checa_art27(vassign) Then
            If materia_repetida() Then
                If checa_creditos() Then
                    verifica_cupo
                End If
            End If
        End If
    End If
End sub
```

**En el caso de que se pasen todas las Validaciones necesarias, se entrara a la última validación en la cual checa el cupo disponible del grupo si existe cupo disponible se generará la inscripción afectando las base de datos y generando el comprobante, como se muestra a continuación.**

```
Sub verifica_cupo ()
```

**Capítulo III.**

---

```
dtgrupo.Index = "primarykey"
dtgrupo.Seek "=", carrera, GRUPO.Text, vassign
If dtgrupo.NoMatch Then
    MsgBox ("no existe")
End If
If opcion(0).Value Then
    If dtgrupo!CUPO = 0 Then
        c = MsgBox(" Grupo saturado ", 48, "Aviso")
    Else
        dtgrupo.Edit
        dtgrupo!CUPO = dtgrupo!CUPO - 1
        dtgrupo.Update
        pasa_materia
    End If
Else
    If dtgrupo!CUPOI = 0 Then
        c = MsgBox(" Grupo saturado!!!! ", 48,
"Aviso")
    Else
        dtgrupo.Edit
        dtgrupo!CUPOI = dtgrupo!CUPOI - 1
        dtgrupo.Update
        pasa_materia
    End If
End If
End Sub
```

**Módulo CCARRERA.FRM**

Este módulo solo esta disponible si se tiene clave de Administrador del Sistema y permite el Cambio Interno de Carrera.

```
Sub CAMBIA_CARRERA ()  
    dsalumno.Edit  
    dsalumno!carrera = NCARRERA  
    dsalumno!PLANTEL = NPLAN  
    dsalumno.Update  
    c = MsgBox("Cambio Realizado", 48, "Aviso")  
End Sub
```

**Módulo CGRUPO.FRM**

Este módulo se encarga de realizar Cambios y Bajas de Materias después del periodo ordinario de inscripción. Las materias seleccionadas para cambio nuevamente son validadas (Por todas las validaciones mencionadas en el módulo de inscripción) antes de efectuar la subrutina de cambio de Materia, que se muestra a continuación.

### Capítulo III.

---

```
Sub cambia_materia () 'Cambio de Materia
  grupoini = text1(0).Text
  asignini = Val(text1(1).Text)
  dtinscrip.Index = "Primarykey"
  dtinscrip.Seek "=", numcuenta, carrera, grupoini, asignini
If dtinscrip.NoMatch Then
  MsgBox ("Registro no encontrado")
Else
  dtinscrip.Edit
  dtinscrip!GRUPO = GRUPOM.Text
  dtinscrip!ASIGN = text1(4).Text
  dtinscrip.Update
  DTCAMBIO.AddNew
  DTCAMBIO!CUENTA = numcuenta
  DTCAMBIO!cambio = 1
  DTCAMBIO!fecha = Format(Now, "dd-mm-yy")
  DTCAMBIO!origen = text1(0).Text
  DTCAMBIO!destino = GRUPOM.Text
  DTCAMBIO!carrera = carrera
  DTCAMBIO!ASIGN = text1(1).Text
  DTCAMBIO.Update
  Actualiza_pantalla
  C = MsgBox("Cambio Realizado", 64, "Aviso")
  banlisto = True
End If
End Sub
```

**Módulo GRUPOS.FRM**

Esta rutina da altas, bajas y cambios de las materias asignadas a un grupo.

**Módulo MATERIAS.FRM**

Este módulo muestra en pantalla los grupos y las materias en que el alumno se encuentra inscrito.

**Módulo PROFESOR.FRM**

Este modulo da Altas, Bajas y Cambios de Profesores.

**Módulo PRIINGRE.FRM**

Los alumnos de primer ingreso son tratados de una manera especial dado que los datos del alumno y el grupo en que esta inscrito son enviados desde Ciudad Universitaria. Este módulo permite la actualización de la información de los alumnos de primer ingreso y la generación de la inscripción.

**Módulo REPORTES.FRM**

Este modulo imprime los siguientes reportes de acuerdo a la selección del usuario:

- Historial académica
- Avance académico
- Grupos por carrera
- Grupos por Asignatura
- Grupo
- Listas oficiales

**Módulo FESZ.BAS**

Este módulo se utiliza exclusivamente para las declaración de variables y subrutinas globales del sistemas.

### **3.6.- PRUEBAS E INTEGRACIÓN DE LOS MÓDULOS.**

La prueba es un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente. Por esta razón se debe definir el proceso de la ingeniería del software además de una plantilla para la prueba del software un conjunto de pasos en la que podamos situar las técnicas específicas de diseño de casos de prueba y los métodos de prueba.

Se han propuesto varias estrategias de prueba del software en distintos libros. Todas proporcionan al ingeniero de software una plantilla para la prueba y todas tienen las siguientes características generales:

- La prueba comienza en el nivel de módulo y trabaja "hacia afuera", hacia la integración de todo el sistema basado en computadora.
- Diferentes técnicas de prueba son apropiadas en diferentes momentos.
- La prueba la lleva a cabo el que desarrolla el software y (para grandes proyectos) un grupo de prueba independiente.
- La prueba y la depuración son actividades diferentes, pero la depuración se puede incluir en cualquier estrategia de prueba.

Una estrategia para la prueba del software debe acomodar pruebas de bajo nivel que verifique que cada pequeño segmento de código fuente se ha implementado correctamente, así como pruebas de alto nivel que muestren la validez de las principales funciones del sistema frente a los requisitos del cliente. Una estrategia debe constituir una guía para el profesional y proporcionar un conjunto de referencias para el gestor o director. Debido a que los pasos de la estrategia de prueba se dan cuando surge la presión de los plazos fijados, se



debe poder medir el progreso, y los problemas deben aparecer lo antes posible.

La prueba de software es un elemento de un concepto más amplio que, a menudo, se referencia como verificación y validación (V & V). La verificación se refiere al conjunto de actividades que aseguran que el software implementa correctamente una función específica. La validación se refiere a un conjunto diferente de actividades que aseguran que el software construido se ajusta a los requisitos del cliente.

En cualquier proyecto de software existe un conflicto de intereses inherente que aparece cuando comienza la prueba. Se pide a la gente que ha construido el software que lo pruebe. Esto parece totalmente inofensivo; después de todo, ¿quién puede conocer mejor un programa que los que lo han desarrollado? Desgraciadamente, esos mismos programadores tienen un gran interés en demostrar que el programa está libre de errores, que funciona de acuerdo a las especificaciones del cliente y que estará listo de acuerdo con los plazos y el presupuesto. Cada uno de éstos intereses se convierte en inconveniente a la hora de encontrar errores a lo largo del proceso de prueba.

El desarrollador del software siempre es responsable de probar las unidades individuales (módulos) del programa, asegurándose de que cada una lleva a cabo la función para la que fue diseñada. En muchos casos, también se encargará de la prueba de integración - el paso de prueba que lleva a la construcción (y prueba) de la estructura total del sistema. Solo una vez que la arquitectura del software esté completa se da paso a eliminar los errores inherentes que se generan al permitir que el constructor del sistema pruebe el mismo.

El proceso de ingeniería del software se puede ver como una espiral, como se muestra en la figura 3.6.1. Inicialmente la ingeniería del sistema define el papel del software y conduce al análisis de los requisitos del software, donde se establece el campo de información, la función, el comportamiento, el rendimiento, las restricciones y los criterios de validación del software. Al movernos hacia el interior de la espiral, llegamos al diseño y, por último, a la codificación. Para desarrollar software de computadora, damos vueltas en espiral a través de una serie de flujos o líneas que disminuyen el nivel de abstracción de cada vuelta.

También podemos imaginar una estrategia para la prueba de software si nos movemos hacia afuera de la espiral de la figura 3.6.1.

La prueba de unidad comienza en el vértice de la espiral y se centra en cada unidad del software, tal como está implementada en código fuente. La prueba avanza, al movernos hacia afuera de la espiral, hasta llegar a la prueba de integración, donde el foco de atención es el diseño y la construcción de la arquitectura del software.

Dando otra vuelta por la espiral hacia afuera, encontramos la prueba de validación, donde se validan los requisitos establecidos por parte del análisis de requisitos del software, comparándolos con el sistema que ha sido construido.

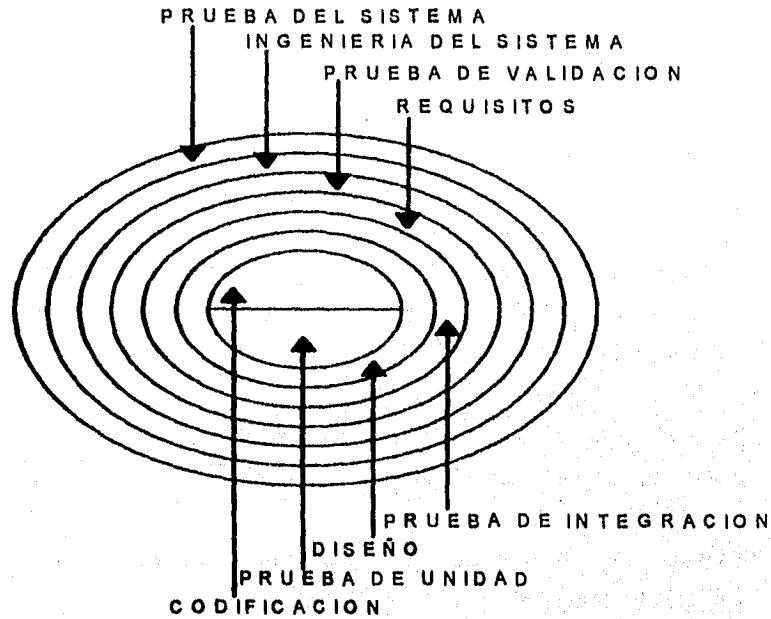


FIGURA 3.6.1 ESTRATEGIA DE PRUEBA.

Finalmente, llegamos a la prueba del sistema, en la que se prueban como un todo el software y otros elementos del sistema. Para probar el software de computadora nos movemos hacia afuera por una espiral que, a cada vuelta, aumenta el alcance de la prueba.

Si consideramos el proceso desde el punto de vista del procedimiento, la prueba, en el contexto de la ingeniería del software, realmente es una serie de tres pasos que se llevan a cabo secuencialmente. Estos pasos se muestran en la figura 3.6.2. Inicialmente, la prueba se centra en cada módulo individual, asegurando que funcionen adecuadamente como una unidad. De ahí el nombre de prueba de unidad. La prueba de unidad hace un uso intensivo de las técnicas de prueba de la caja negra, ejercitando caminos específicos de la estructura de

control del módulo para asegurar un alcance completo y una detección máxima de errores. A continuación se deben ensamblar o integrar los módulos para formar el paquete de software completo. La prueba de integración se dirige a todos los aspectos asociados con el doble problema de verificación y construcción del programa. Durante la integración, las técnicas que más prevalecen son las de diseño de casos de prueba de la caja negra, aunque se pueden llevar a cabo unas pocas pruebas de la caja blanca con el fin de asegurar que se cubren los principales caminos de control. Después de que el software se ha integrado (construido), se dirigen un conjunto de pruebas de alto nivel.

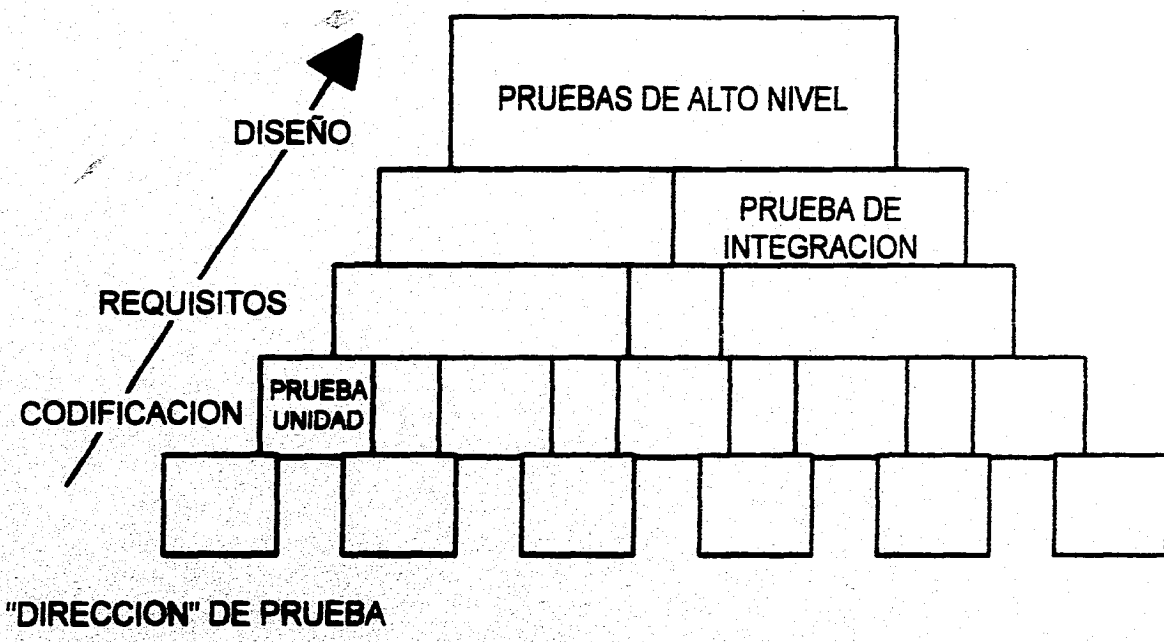


Figura 3.6.2

Se deben comprobar los criterios de validación. La prueba de validación proporciona una seguridad final de que el software satisface todos los requisitos funcionales, de comportamiento y de rendimiento. Durante la validación se usan exclusivamente técnicas de prueba de la caja negra.

El último paso de prueba queda fuera de los límites de la ingeniería del software, entrando en el más amplio contexto de la ingeniería de sistemas de computadora. El software una vez validado, se debe combinar con otros elementos del sistema (p. ej. hardware, gente, bases de datos). La prueba del sistema verifica que cada elemento encaja en forma adecuada y que se alcanza la funcionalidad y el rendimiento del sistema total.

### Capítulo III.

---

#### **3.6.1.-PRUEBA DE UNIDAD.**

La prueba de unidad centra el proceso de verificación en la menor unidad del diseño del software - el módulo. Usando la descripción del diseño detallado como guía, se prueban los caminos de control importantes, con el fin de descubrir errores dentro del ámbito del módulo. La complejidad relativa de las pruebas y de los errores descubiertos estuvo limitada por el alcance estricto establecido por la prueba de unidad. La prueba de unidad estuvo orientada a la caja blanca y este paso se llevó a cabo en paralelo para múltiples módulos.

Las pruebas que se dan como parte de la prueba de unidad están esquemáticamente ilustradas en la figura 3.6.1.1. Se probaron las interfaces de cada módulo para asegurar que la información fluyera de forma adecuada hacia y desde la unidad del programa que está siendo probada. Se examinaron las estructuras de datos locales para asegurar que los datos que se mantienen temporalmente conservan su integridad durante todos los pasos de ejecución del algoritmo. Se probaron las condiciones límite para asegurar que los módulos funcionaran correctamente en los límites establecidos como restricciones de procesamiento. Se ejercitaron todos los caminos independientes (caminos básicos) de la estructura de control con el fin de asegurar que todas las sentencias del módulo se ejecutasen por lo menos una vez. Y finalmente, se probaron todos los caminos de manejo de errores.

Antes de iniciar cualquier otra prueba, fue preciso probar el flujo de datos de la interfaz de cada módulo, porque si los datos no entraban correctamente, todas las demás pruebas no tendrían sentido.

---

*Diseño e Implementación del Sistema.*

Se aplicó una estrategia de enfoque descendente, para lo cual se probó que los módulos de nivel más alto ejecutaran correctamente los módulos subordinados; posteriormente se probaron los módulos subordinados hasta llegar a los módulos de nivel más bajo. Este tipo de prueba conocida como "TOP-DOWN" es una estrategia que prueba los módulos de más alto nivel, antes de que hayan sido codificados (y tal vez diseñados) los módulos de bajo nivel.



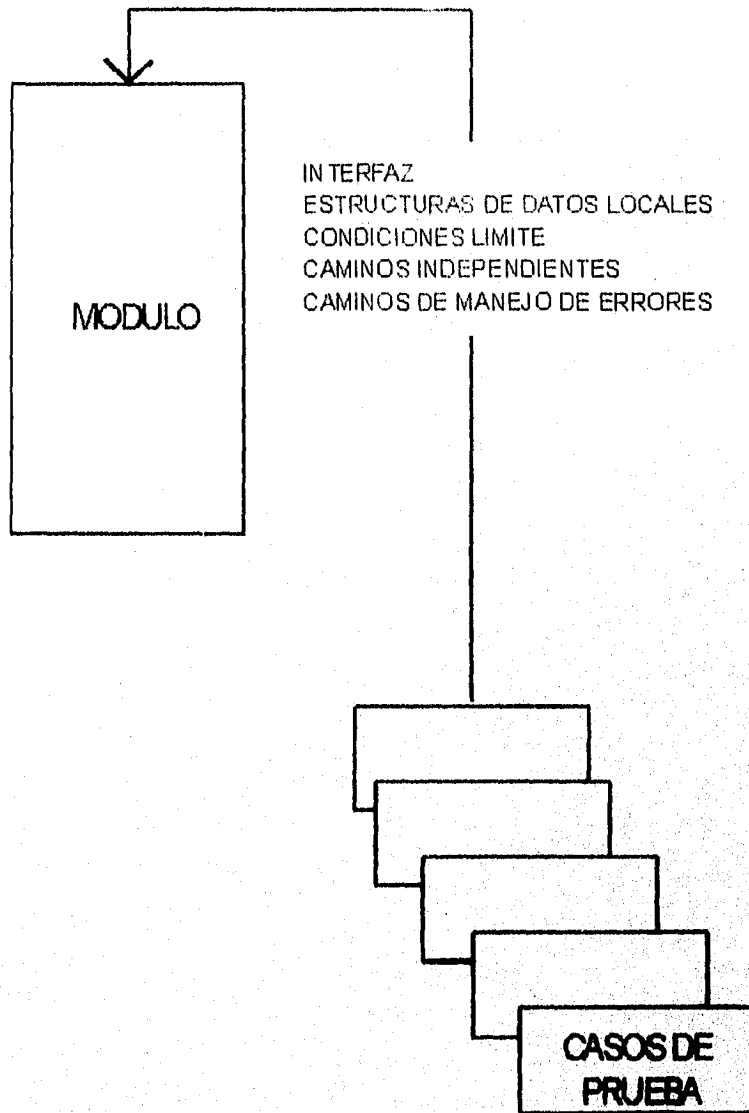


FIGURA 3.6.1.1 PRUEBA DE UNIDAD

Para poner en práctica esta estrategia se comenzó con la programación de los módulos jerárquicamente más altos con el fin de presentarlos periódicamente a los directivos, para así programar los menús, pantallas principales y formatos tal como ellos los solicitaban. Este proceso fue utilizado durante toda la etapa de programación. Al pasar de los módulos ejecutivos a los módulos subordinados, se realizaron casos de prueba a partir de las especificaciones o salidas deseadas en la etapa de análisis. Una vez obtenidos los resultados esperados, se presentó a los usuarios que operarían el sistema, los cuales dieron sus puntos de vista y se les pidió que generaran casos de prueba excepcionales que con su experiencia han observado.

El sistema CONEFESZ está compuesto por los módulos que se muestran en la figura 3.6.1.2.

El módulo central o principal permite tener acceso al sistema mediante un menú de opciones, las cuales son la actualización de datos del alumno, inscripciones, obtención de listas, realización de cambios, utilerías del sistema y salir del mismo.

Se probó cada uno de los menús de opciones y que las pantallas de captura cumpliera con los puntos que se muestran a continuación:

- **Nombre del sistema.**  
Aparece en la pantalla de inicio del sistema.
- **Ortografía correcta.**  
Todos los mensajes y pantallas.

### Capítulo III.

---

- **Nombre de la opción.**  
Cuando se selecciona un menú se mantiene el nombre de este en la parte superior izquierda de la pantalla y centrado con el fin de saber en todo momento en la opción en que nos encontramos.
  
- **Descripción de la opción seleccionada.**  
Cuando se esté posicionado en una opción deberá aparecer una descripción en la parte inferior de la pantalla que nos indica la función de dicha opción.
  
- **Tecla de tabulador.**  
Se podrá hacer uso de la tecla de tabulador para moverse a través de todos los botones, opciones o campos que se desplieguen en la pantalla.

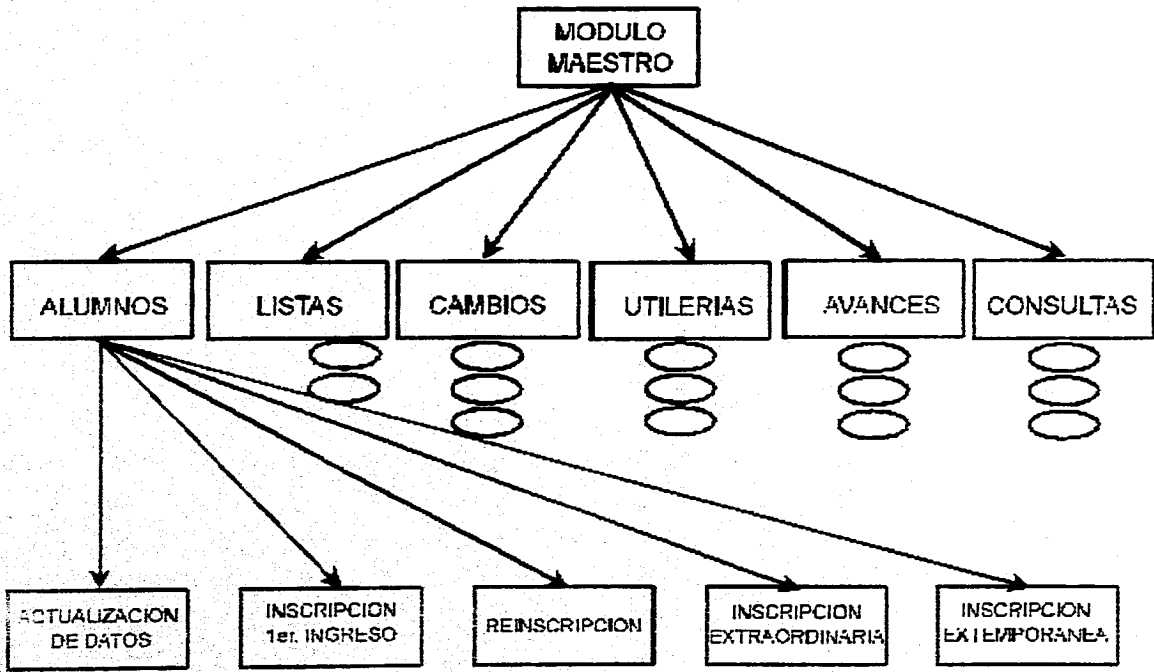


FIGURA 3.6.1.2 MÓDULOS DEL SISTEMA

### **Capítulo III.**

---

- **Botón de aceptación.**  
Se podrá hacer uso de un botón para indicar que estamos de acuerdo con la selección que se escogió o con los datos introducidos que nos solicite alguna de las opciones y se ejecuta el módulo del programa que le corresponde.
  
- **Botón de regresar.**  
Se podrá hacer uso de un botón para abandonar la selección y regresarnos al menú anterior.
  
- **Uso del Mouse.**  
Se podrá hacer uso del Mouse oprimiendo el botón izquierdo con el fin de activar a un botón de aceptación, un botón de cancelación, una selección, o para ubicarse en un campo en específico para introducir información.

**PRUEBA DEL MÓDULO DE ALUMNOS.**

En éste módulo se concentra todo el manejo de la información referente a los alumnos.

**PRUEBAS DEL SUBMÓDULO ACTUALIZACIÓN DE DATOS DEL MÓDULO DE ALUMNOS.**

En éste submódulo se podrán hacer actualizaciones a los datos personales de los alumnos. Podrán actualizarse datos hasta que se active el botón de regresar para volver al módulo principal de CONEFESZ. El campo en el que se aplicó la prueba fué:

- **Número de cuenta.**

Al proporcionar el número de cuenta del alumno el sistema valida si existe, si no enviará un mensaje que el usuario no existe en la base de datos.

**PRUEBAS DE LA OPCIÓN DE INSCRIPCIÓN DE PRIMER INGRESO DEL  
MÓDULO DE ALUMNOS.**

Si se desea inscribir a algún alumno de nuevo ingreso se solicitan los siguientes datos:

- **Número de cuenta.**

Al proporcionar el número de cuenta del alumno el sistema valida si existe, si no enviará un mensaje de que el alumno no existe en la base de datos.

A continuación el sistema pedirá que el usuario cheque los siguientes datos, para su corrección o archivación en el sistema:

- **Carrera.**
- **Plantel.**
- **Dirección.**
- **Año de ingreso.**
- **Escuela de procedencia.**
- **Promedio.**
- **Código postal.**
- **Teléfono.**
- **Grupo.**

**PRUEBAS DE LA OPCIÓN DE REINSCRIPCIÓN DEL MÓDULO DE ALUMNOS.**

Si se desea reinscribir a algún alumno se solicitan los siguientes datos:

- **Número de cuenta.**
- **Carrera.**
- **Plantel.**

Al proporcionar el número de cuenta, la carrera y el plan de estudios en que el alumno está inscrito el sistema valida si existe, si no enviará un mensaje de que el alumno no existe en la base de datos.

- **Grupo.**  
El usuario deberá proporcionar el grupo al que desea inscribirse. El sistema validará automáticamente si éste existe o no.
- **R/I.**  
El sistema debe de saber que tipo de alumno es el que en ese momento desea inscribirse (R=regular/I=irregular).
- **Asignatura.**  
El alumno deberá escoger la asignatura a la que desea inscribirse. En éste punto el sistema validará automáticamente que el alumno no se encuentre en artículo 27 o 19, es decir, que el alumno no haya excedido el tiempo que le permite la institución en terminar su carrera, o que no haya cursado la asignatura 2 veces, a su vez el sistema también validará el límite de créditos permitido.



**Capítulo III.**

**PRUEBAS DE LA OPCIÓN DE INSCRIPCIÓN A EXAMEN EXTRAORDINARIO DEL MÓDULO DE ALUMNOS.**

Si se desea inscribir a algún alumno a examen extraordinario se solicitan los siguientes datos:

- **Número de cuenta.**
- **Carrera.**  
Al proporcionar el número de cuenta, la carrera en que el alumno está inscrito el sistema valida si existe, si no enviará un mensaje de que el alumno no existe en la base de datos.
- **Grupo.**  
El usuario deberá proporcionar el grupo al que desea inscribirse. El sistema validará automáticamente si éste existe o no.
- **Asignatura.**  
El alumno deberá escoger la asignatura a la que desea inscribirse. En éste punto el sistema validará automáticamente que el alumno no rebese el límite de dos inscripciones a examen extraordinario, además el sistema validará los siguientes datos:
  - Que el alumno no se haya inscrito de manera normal en dicha asignatura.
  - Que el alumno no haya aprobado aún la materia.

### **PRUEBA DE LA OPCIÓN DE INSCRIPCIÓN EXTEMPORÁNEA DEL MÓDULO DE ALUMNOS.**

A éste MÓDULO se le aplicaron exactamente las mismas pruebas que se le aplicaron al submódulo de reinscripciones, ya que están constituidos por los mismos programas, con la salvedad de que su ejecución se diferencia por el valor de una variable que se captura al momento de seleccionar dicha opción en el menú de inscripciones de alumnos.

### **PRUEBA DEL MÓDULO DE OBTENCIÓN DE LISTAS.**

Este módulo está formado por los submódulos que se muestran en la figura 3.6.1.4.

### **PRUEBAS DE LA OPCIÓN DE LISTAS POR GRUPO DEL MÓDULO DE LISTAS.**

En ésta opción se solicitará cada uno de los datos que formarán los datos que constituyen la llave compuesta de búsqueda, se probó que cada uno de los campos y botones cumpliera con:

- **Carrera.**  
Se proporciona la clave de la carrera deseada y el sistema validará si existe o no.
- **Grupo.**  
Se proporciona el grupo que se desee su impresión, sin antes ser validado por el sistema.
- **Asignatura.**  
Se debe de proporcionar la clave de la asignatura en cuestión, antes, el sistema la debe validar.

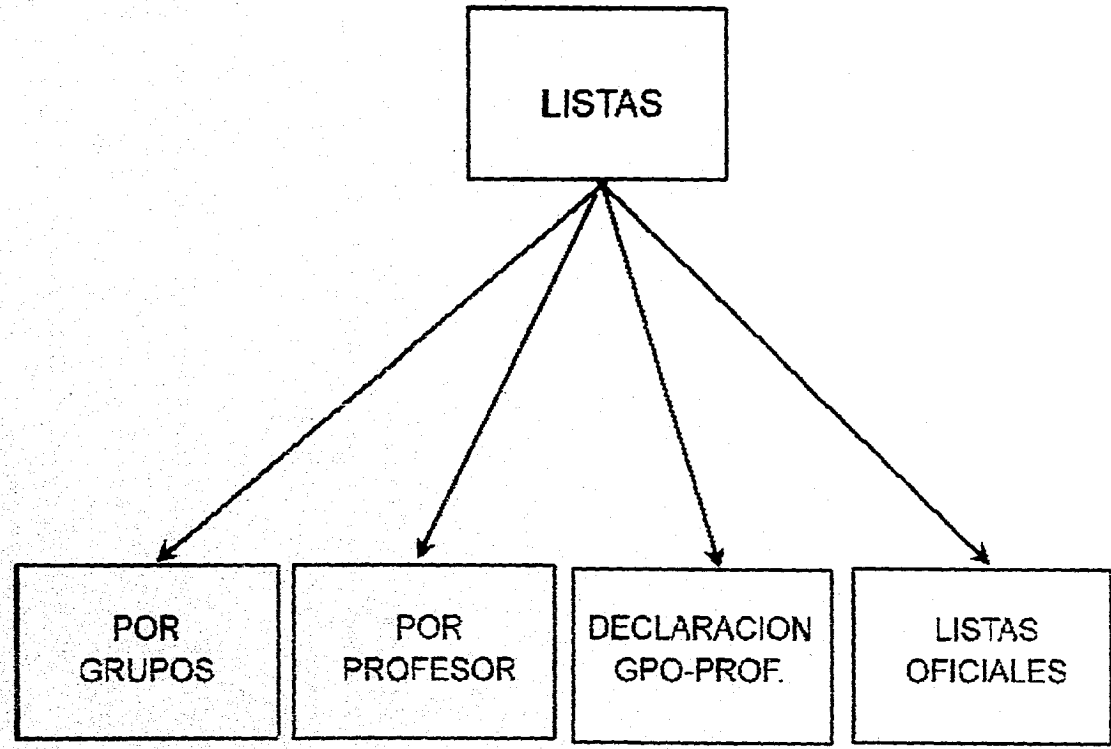


FIGURA 3.6.1.4 MÓDULO DE LISTAS

**PRUEBAS DE LA OPCIÓN DE LISTAS POR PROFESOR DEL MÓDULO DE LISTAS.**

En ésta opción se solicitará cada uno de los datos que formarán los datos que constituyen la llave compuesta de búsqueda, se probó que cada uno de los campos y botones cumpliera con:

- **Carrera.**  
Se proporciona la clave de la carrera deseada y el sistema validará si existe o no.
- **Grupo.**  
Se proporciona el grupo que se desee su impresión, sin antes ser validado por el sistema.
- **Profesor.**  
Se proporciona el RFC del Profesor que quiera imprimir los grupos que le estén asignados, validados previamente por el sistema.

### **Capítulo III.**

---

#### **PRUEBAS DE LA OPCIÓN DE DECLARACIÓN DE GRUPO/PROFESOR DEL MÓDULO DE LISTAS.**

En ésta opción se solicitará cada uno de los datos que formarán los datos que constituyen la llave compuesta de búsqueda, se probó que cada uno de los campos y botones cumpliera con:

- **Carrera.**  
Se proporciona la clave de la carrera deseada y el sistema validará si existe o no.
- **Plan.**  
Se proporciona el plan de estudios que junto con la carrera constituirán la llave de búsqueda.

#### **PRUEBAS DE LA OPCIÓN DE LISTAS OFICIALES DEL MÓDULO DE LISTAS.**

En ésta opción se solicitará cada uno de los datos que formarán los datos que constituyen la llave compuesta de búsqueda, se probó que cada uno de los campos y botones cumpliera con:

- **Carrera.**  
Se proporciona la clave de la carrera deseada y el sistema validará si existe o no.
- **Grupo.**  
Se proporciona el grupo que se desee su impresión, pero antes debe ser validado por el sistema.

- **Asignatura.**  
Se debe de proporcionar la clave de la asignatura en cuestión, antes, el sistema la debe validar.

**NOTA:** Esta opción maneja los mismos programas, con la salvedad del formato de salida que es diferente.

**PRUEBA DEL MÓDULO DE OBTENCIÓN DE CAMBIOS.**

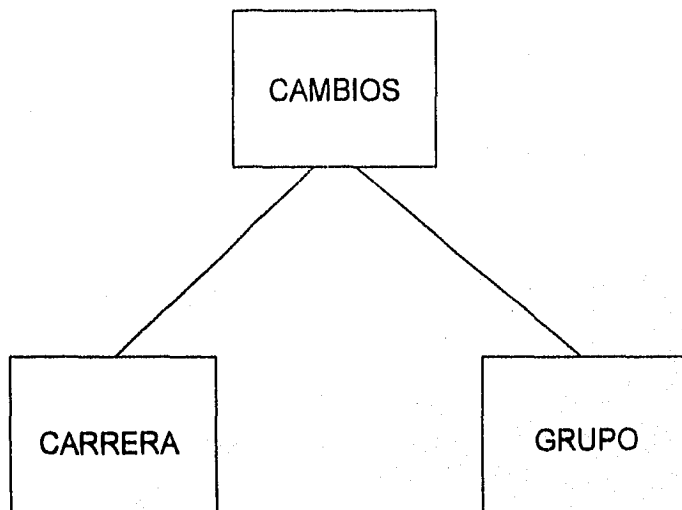
Este módulo está formado por los submódulos que se muestran en la figura 3.6.1.5.

**PRUEBAS DE LA OPCIÓN DE CAMBIOS DE CARRERA DEL MÓDULO DE CAMBIOS.**

En éste módulo se podrán hacer cambios de carrera que algún alumno solicite. Los campos en los que se aplicaron las pruebas fueron:

- **Carrera.**  
Se proporciona la clave de la carrera deseada y el sistema validará si existe o no.
- **Plantel.**  
Se proporciona la clave del plan de estudios de la carrera en la que el alumno está inscrito actualmente.
- **Número de cuenta.**  
Se proporciona el número de cuenta del alumno en cuestión.

Una vez que se tengan los tres datos, el sistema buscará en el archivo de alumnos, al alumno en cuestión, si no lo encontrase, se desplegará un mensaje de error; en el caso contrario se puede cambiar la clave de la carrera y la clave del plan de estudios a la que el alumno se desea cambiar.



**FIGURA 3.6.1.5 MÓDULO DE CAMBIOS**



**PRUEBAS DE LA OPCIÓN DE CAMBIOS DE GRUPO DEL MÓDULO DE CAMBIOS.**

En éste módulo se podrán hacer cambios de grupo que algún alumno solicite. Los campos en los que se aplicaron las pruebas fueron:

- **Carrera.**  
Se proporciona la clave de la carrera deseada y el sistema validará si existe o no.
- **Plantel.**  
Se proporciona la clave del plan de estudios de la carrera en la que el alumno está inscrito actualmente.
- **Número de cuenta.**  
Se proporciona el número de cuenta del alumno en cuestión.

Una vez que se tengan los tres datos, el sistema buscará en el archivo de alumnos, al alumno en cuestión, si no lo encontrase, se desplegará un mensaje de error; en el caso contrario se desplegarán los datos del mismo, y se pedirán los siguientes datos:

- **Grupo.**  
Se proporciona la clave del grupo del cual el alumno quiera cambiarse, el cual el sistema lo validará en el archivo de grupos.
- **Asignatura.**  
Se proporciona la clave de la asignatura de la cual el alumno desea cambiarse, la cual el sistema tendrá que checar en el archivo de asignaturas.

Una vez que se tengan los dos datos, junto con el número de cuenta, el sistema buscará en el archivo de inscripción, al alumno en cuestión, si no lo encontrase, se desplegará un mensaje de error; en el caso contrario se desplegarán los datos del mismo y se pedirán los datos del grupo y asignatura al que el alumno desee cambiarse, no sin antes ser validados por el sistema.

### **Capítulo III.**

---

#### **PRUEBA DEL MÓDULO DE AVANCE.**

Este módulo está formado por los submódulos que se muestran en la figura 3.6.1.6.

#### **PRUEBAS DE LA OPCIÓN DE AVANCE ACADÉMICO DEL MÓDULO DE AVANCE.**

En éste módulo se podrá obtener una impresión del avance académico que el alumno solicite. Los campos en los que se aplicaron las pruebas fueron:

- **Carrera.**  
Se proporciona la clave de la carrera deseada y el sistema validará si existe o no.
  
- **Plantel.**  
Se proporciona la clave del plan de estudios de la carrera en la que el alumno está inscrito actualmente.
  
- **Número de cuenta.**  
Se proporciona el número de cuenta del alumno en cuestión.

Una vez que se tengan los tres datos, el sistema buscará en el archivo de alumnos, al alumno en cuestión, si no lo encontrase, se desplegará un mensaje de error; en el caso contrario el sistema comenzará a realizar el avance académico del alumno, para posteriormente desplegarlo en la impresora.

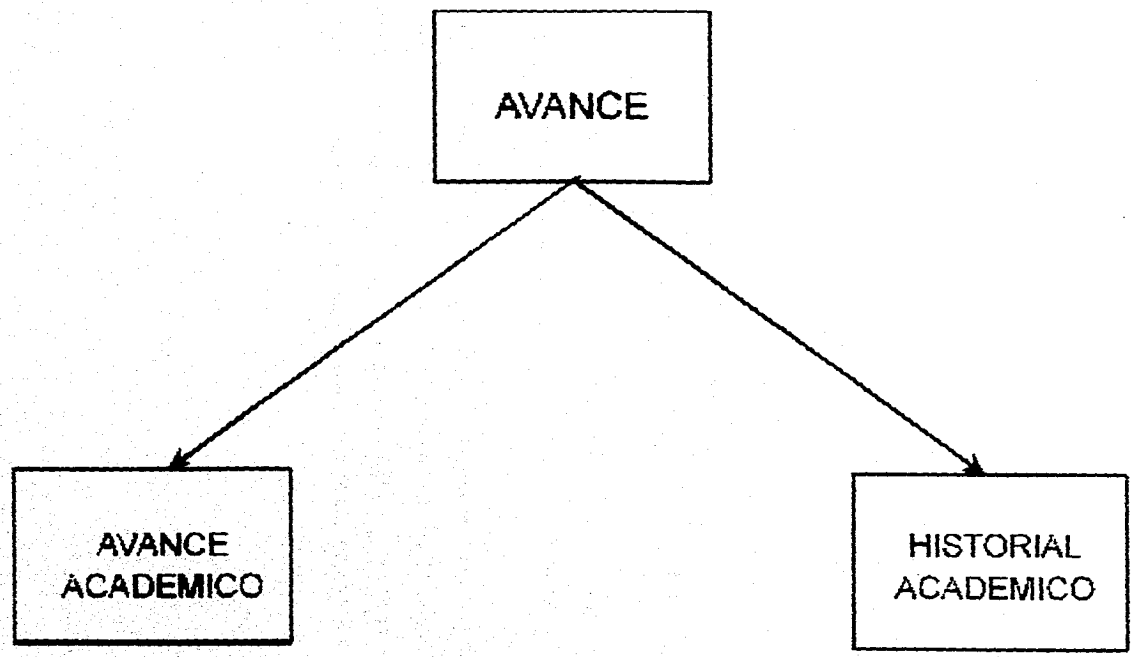


FIGURA 3.6.16 MÓDULO DE AVANCE

**PRUEBAS DE LA OPCIÓN DE HISTORIAL ACADÉMICO DEL MÓDULO DE AVANCE.**

En éste módulo se podrá obtener una impresión del avance académico que el alumno solicite. Los campos en los que se aplicaron las pruebas fueron:

- **Carrera.**  
Se proporciona la clave de la carrera deseada y el sistema validará si existe o no.
  
- **Plantel.**  
Se proporciona la clave del plan de estudios de la carrera en la que el alumno está inscrito actualmente.
  
- **Número de cuenta.**  
Se proporciona el número de cuenta del alumno en cuestión.

Una vez que se tengan los tres datos, el sistema buscará en el archivo de alumnos, al alumno en cuestión, si no lo encontrase, se desplegará un mensaje de error; en el caso contrario el sistema comenzará a realizar el avance académico del alumno, para posteriormente desplegarlo en la impresora.

### **3.6.2.-PRUEBA DE LA CAJA BLANCA.**

La prueba de la caja blanca es un método de diseño de casos de prueba que usa la estructura de control del diseño procedural para derivar los casos de prueba. Mediante los métodos de prueba de la caja blanca, se pudieron obtener casos de prueba que:

- 1.- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo.
- 2.- Ejerciten todas las decisiones lógicas en sus vértices verdadera y falsa.
- 3.- Ejecuten todos los bucles en sus límites y con sus límites de operaciones.
- 4.- Ejerciten las estructuras internas de datos para asegurar su validez.

En este momento, se puede plantear una pregunta razonable: ¿Por qué gastar tiempo y energía preocupándose de (y de probarlo) los detalles cuando podríamos gastar mejor el esfuerzo asegurando que se han alcanzado los requisitos del programa? O, dicho de otra forma, ¿Por qué no gastamos todas nuestras energías en las pruebas de la caja negra? La respuesta se encuentra en la naturaleza misma del software:

- Los errores lógicos y las suposiciones incorrectas son inversamente proporcionales a la probabilidad de que se ejecute un camino del programa. Los errores tienden a reproducirse en nuestro trabajo cuando diseñamos e implementamos funciones, condiciones o controles que se encuentran fuera de lo normal. El procedimiento habitual tiende a hacerse más comprensible (y bien examinado), mientras que el procedimiento de "casos especiales" tiende a caer en el caos.

### Capítulo III.

---

- A menudo creemos que un camino lógico tiene pocas posibilidades de ejecutarse cuando, de hecho, se puede ejecutar de forma regular. El flujo lógico de un programa a veces no es nada intuitivo, lo que significa que nuestras suposiciones intuitivas sobre el flujo de control y los datos nos pueden llevar a tener errores de diseño que sólo se descubren cuando comienza la prueba del camino.
- Los errores tipográficos son aleatorios. Cuando se traduce un programa a código fuente en un lenguaje de programación, es muy probable que se den algunos errores de escritura. Muchos serán descubiertos por los mecanismos de comprobación de sintaxis, pero otros permanecerán indetectados hasta que comience la prueba, así como es igual de probable que haya un error tipográfico en un oscuro camino lógico.

Cada una de estas razones nos da un argumento para llevar a cabo las pruebas de la caja blanca. La prueba de la caja negra, sin tener en cuenta como sea de completa, puede pasar por alto los tipos de errores que acabamos de señalar. Es mucho más fácil descubrirlos con la prueba de la caja blanca.

### **3.6.3.-PRUEBA DE LA CAJA NEGRA.**

Los métodos de prueba de la caja negra se encuentran en los requisitos funcionales del software. O sea, la prueba de la caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de la caja negra no es una alternativa a las técnicas de prueba de la caja blanca. Más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de la caja blanca.

La prueba de la caja negra intenta encontrar errores de las siguientes características:

- 1.- Funciones incorrectas o ausentes.
- 2.- Errores de interfaz.
- 3.- Errores de estructuras de datos o en accesos a bases de datos externas.
- 4.- Errores de rendimiento.
- 5.- Errores de inicialización y de terminación.

A diferencia de la prueba de la caja blanca, que se llevó a cabo en el proceso de prueba, la prueba de la caja negra se aplicó durante posteriores fases de prueba. Ya que la prueba de la caja negra ignora intencionalmente la estructura de control, centra su atención en el campo de información. Los cuestionamientos se diseñaron en éste caso para responder a las siguientes preguntas:

- ¿Cómo se prueba la validez funcional?
- ¿Qué clases de entrada compondrán unos buenos casos de prueba?
- ¿Es el sistema particularmente sensible a ciertos errores de entrada?
- ¿De qué forma están aislados los límites de una clase de datos?



### **Capítulo III.**

---

- ¿Qué volúmenes y niveles de datos tolerará el sistema?
- ¿Qué efectos sobre la operación del sistema tendrán combinaciones específicas de datos?

Mediante las técnicas de prueba de la caja negra se derivaron un conjunto de casos de prueba que satisfacen los siguientes criterios:

- 1.- Casos de prueba que reducen, en un coeficiente que es mayor que uno, el número de prueba adicionales que se deben diseñar para enlazar una prueba razonable y,
- 2.- Casos de prueba que nos dicen algo sobre la presencia o ausencia de clases de errores en lugar de un error asociado solamente con la prueba, en particular que se encuentre disponible.

### **3.6.4.-PRUEBA DE INTEGRACIÓN.**

Se utilizó esta técnica para construir la estructura del programa y para realizar pruebas para detectar errores asociados con la integración de los módulos. Se tomaron los módulos probados en unidad y nos aseguramos que la estructura del programa estuviera de acuerdo con lo que dictaba el diseño.

Se utilizó la integración incremental en el cual el programa se construye y se prueban los módulos en los que los errores son más fáciles de aislar y de corregir con el fin de probar completamente todas las interfaces y aplicar una aproximación de prueba sistemática.

Existen dos estrategias de integración incremental:

- Integración ascendente.
- Integración descendente.

Empleamos integración descendente de forma primero-en-profundidad, integramos los módulos moviéndonos hacia abajo por la jerarquía de control, comenzando por el módulo de control principal se van incorporando en la estructura de forma primero-en-profundidad.

En la figura 3.6.4.1 se eligió el camino a mano izquierda, se integraron los módulos M1, M2 y M3. A continuación será integrado M4. Acto seguido se construyen los caminos de control central y derecho.

Llevamos a cabo la integración en cinco pasos:

### **Capítulo III.**

---

Se usó el módulo principal como conductor de la prueba, disponiendo de los resguardos para todos los módulos directamente subordinados al módulo principal (los resguardos sirven para reemplazar módulos que están subordinados a el módulo a ser probado; un resguardo o "subprograma mudo" usa la interface del módulo subordinado, lleva a cabo la mínima manipulación de datos e imprime una verificación de la entrada y vuelve).

- 1.- Se usa el módulo de control principal como conductor de la prueba, disponiendo resguardos para todos los módulos directamente subordinados al módulo de control principal.
- 2.- Se fueron sustituyendo los resguardos subordinados uno a uno por los módulos reales.
- 3.- Se llevaron a cabo pruebas de caja negra cada vez que integramos un nuevo módulo.
- 4.- Después de las pruebas reemplazamos otro resguardo con un módulo real.
- 5.- Realizamos la prueba de regresión ( o sea, todas las pruebas anteriores) para asegurarnos de que no hayan introducido nuevos errores.

El proceso continuó desde el paso 2 hasta construir la estructura del programa entero. En cada reemplazamiento se llevaron a cabo pruebas para verificar la interface.

La principal desventaja de la aproximación descendente es la necesidad de resguardos y las dificultades de prueba que pueden estar asociadas con ellos.

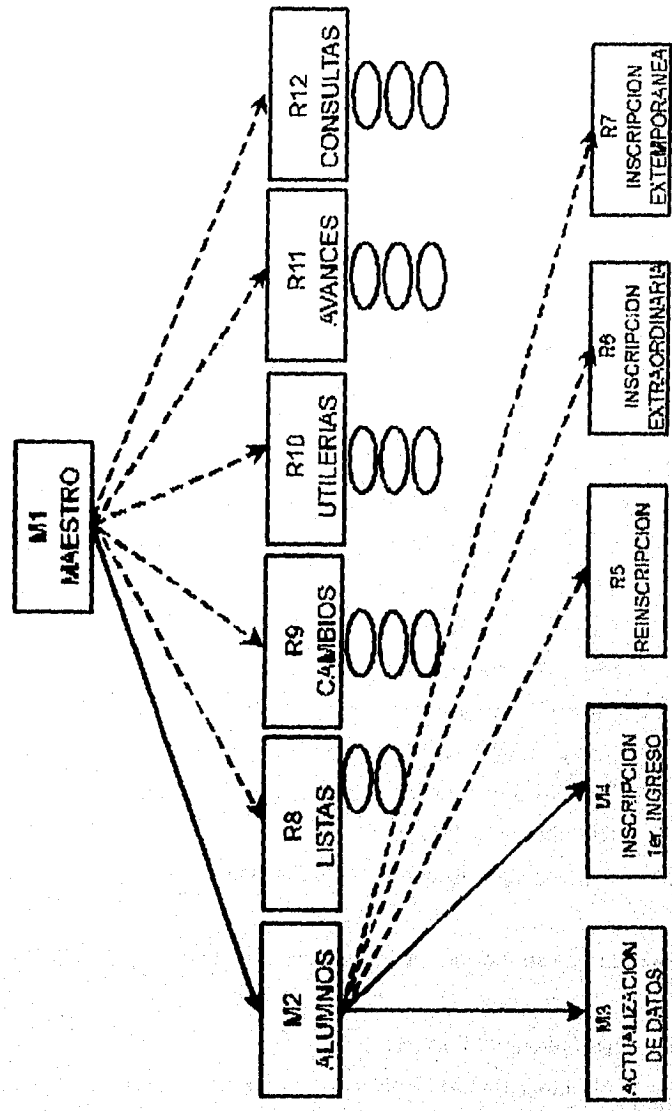


FIGURA 3.6.4.1 INTEGRACIÓN DESCENDENTE DE FORMA PRIMERO EN PROFUNDIDAD.

**ESPECIFICACIONES DE LA PRUEBA DE INTEGRACIÓN.**

**Alcance de la prueba.**

Se probaron características tales como:

- Fácil uso por parte del usuario.
- Desplegado de ayuda.
- El fácil acceso utilizando los dispositivos de entrada: el Mouse y el teclado.
- Además que el número de módulos cumpliera con las especificaciones solicitadas.

**Plan de prueba.**

La estrategia en general para la integración se dividió en cuatro fases y subfases, dirigidas a características específicas funcionales del software.

- Interacción con el usuario.  
Selección de órdenes.  
Representación visual.  
Procesamiento y representación de errores.
- Manipulación y análisis de datos.
- Procesamiento y generación de información visual.  
Reportes y estadísticas.
- Estructura y contenido de la base de datos.

En cada fase se sugirieron los criterios con sus correspondiente pruebas:

- **Integridad de la interface.**  
Se probaron las interfaces Internas y externas a medida que se incorporaron los módulos.

- **Validez funcional.**  
Se llevaron a cabo pruebas diseñadas para descubrir errores funcionales.
- **Contención de información.**  
Pruebas para descubrir errores asociados con las estructuras globales y locales.
- **Rendimiento.**  
Pruebas para verificar los límites de rendimiento establecidos durante el diseño del software.

Los recursos empleados fueron una PC AT 486 DX4 con 340 MB en disco duro disponible, con conexión a red.

### 3.6.5.-PRUEBA DE VALIDACIÓN.

Tras la culminación de la prueba de integración, el sistema está completamente ensamblado como un paquete, se han encontrado y corregido los errores de interfaz y puede comenzar una serie final de pruebas del software -la prueba de validación. La validación puede definirse de muchas formas, pero una muy simple es que la validación se logra cuando el sistema funciona de acuerdo con las expectativas razonables del cliente.

Las expectativas razonables estuvieron definidas en la especificación de requisitos del sistema - un documento que describe todos los atributos del software que son visibles al usuario. La especificación contiene una sección denominada criterios de validación. La información contenida en esa sección forma la base del enfoque a la prueba de validación.

La validación del sistema se consigue mediante una serie de pruebas de la caja negra que demuestran la conformidad de los requisitos. Un plan de prueba trazó las pruebas que se llevaron a cabo y un procedimiento para éstas definió los casos de prueba específicos que se usaron para demostrar la conformidad con los requisitos. Tanto el plan como el procedimiento estuvieron diseñados para asegurar la satisfacción de todos los requisitos funcionales, para alcanzar los requisitos de rendimiento, que la documentación sea correcta e inteligible y otros requisitos (p. ej.: portabilidad, compatibilidad, facilidad de mantenimiento, etc.).

Una vez que se procede con cada caso de prueba de validación, puede darse una de dos condiciones:

- 1.- Las características de funcionamiento o de rendimiento están de acuerdo con las especificaciones y son aceptables, o,

- 2.- Se descubre una desviación de las especificaciones y se crea una lista de deficiencias.

Las desviaciones o errores descubiertos en esta fase del proyecto raramente se pueden corregir antes de terminar el plan. A menudo es necesario negociar con el cliente o usuario un método para resolver las deficiencias.



### 3.6.6.-REPASO DE LA CONFIGURACIÓN.

Un elemento importante en el proceso de validación es el repaso de la configuración. Tal como se muestra en la figura 3.6.6.1, el repaso intenta asegurar que todos los elementos de la configuración del sistema se han desarrollado en forma adecuada, están catalogados y tienen el suficiente detalle para facilitar la fase de mantenimiento dentro del ciclo de vida del sistema.

Es virtualmente imposible que un encargado del desarrollo del software pueda prever como un cliente pueda usar realmente un programa. Se pueden interpretar mal las instrucciones de uso, se pueden usar regularmente extrañas combinaciones de datos y una salida que puede estar clara para el que realiza la prueba, pero puede resultar inteligible para un usuario final.

Cuando se construye software a la medida para un cliente, se lleva a cabo una serie de pruebas de aceptación para permitir que el cliente valide todos los requisitos. Llevado a cabo por el usuario final en lugar del equipo de desarrollo.

Una prueba de aceptación puede ir desde el informal "paso de prueba" hasta la ejecución sistemática de una serie de pruebas bien planificadas. De hecho, la prueba de aceptación puede tener lugar a lo largo de semanas o meses, descubriendo así errores acumulados que pueden ir degradando el sistema.

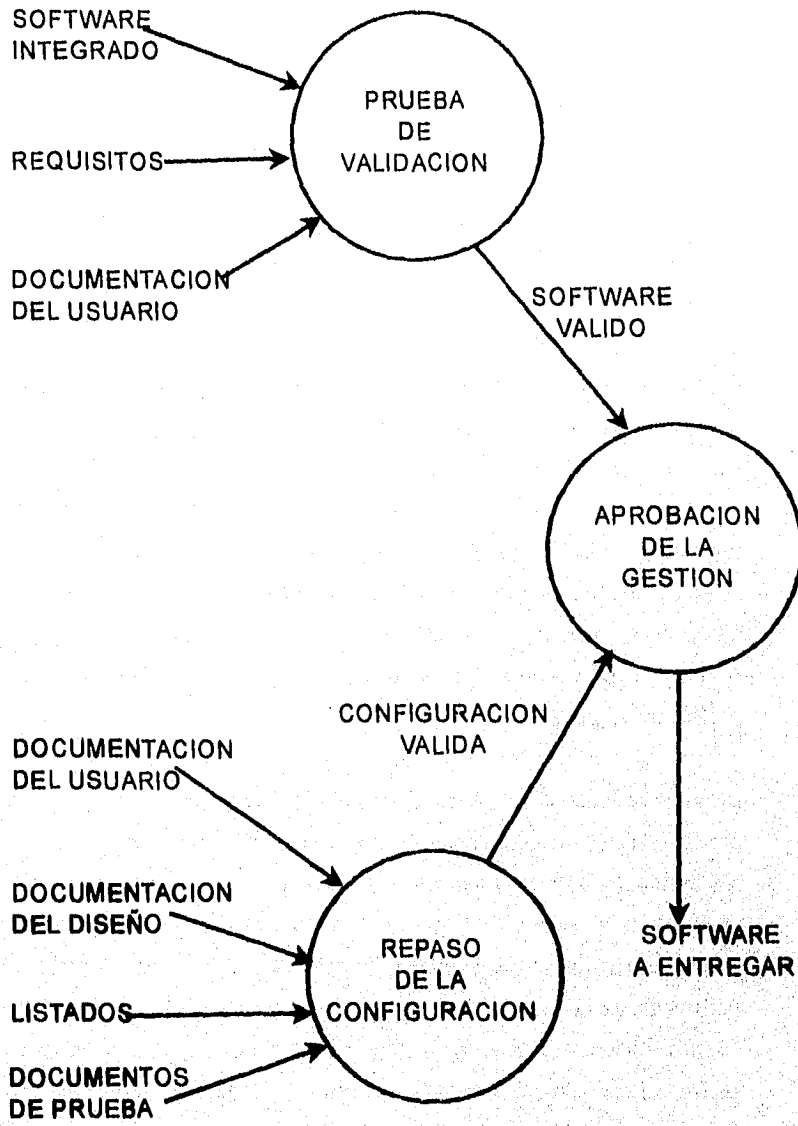


FIGURA 3.6.6.1 REPASA DE LA CONFIGURACIÓN

### 3.6.7.-PRUEBAS ALFA Y BETA.

Si el software se desarrolla como un producto que se va a usar por muchos clientes, no es práctico realizar pruebas de aceptación formales para cada uno de ellos. La mayoría de los constructores de productos de software llevan a cabo un proceso denominado "prueba alfa y beta" para descubrir errores que parezca que solo el usuario final puede descubrir.

La prueba alfa es conducida por un cliente en lugar del desarrollador. Se usa el software de forma normal, con el encargado del desarrollo "mirando por encima del hombro" del usuario y registrando errores y problemas de uso. Las pruebas alfa se llevan a cabo en un entorno controlado.

En nuestro caso, ésta prueba se llevó a cabo en presencia de los usuarios, el Jefe de Servicios Escolares, El Jefe del Departamento de Avance Académico y Titulación, La Jefa del Departamento de Actas y los desarrolladores del sistema. Al final de la prueba se obtuvieron resultados satisfactorios, por lo que se dio por pasada dicha prueba.

La prueba beta se lleva a cabo en uno o más lugares que los clientes decidan, por ser ellos los usuarios finales del software. A diferencia de la prueba alfa, el encargado del desarrollo del software, normalmente, no esta presente. Así, la prueba beta es una aplicación "en vivo" del software en un entorno que no puede ser controlado por el equipo de desarrollo. El cliente registra todos los problemas (reales o imaginarios) que encuentra durante la prueba beta e informa a intervalos regulares al equipo de desarrollo. Como resultado de los problemas anotados durante la prueba beta, el equipo de desarrollo del software lleva a cabo modificaciones y así prepara una versión del producto de software para toda la base de clientes.

Para la realización de ésta prueba, se capacitó al personal y durante esta etapa se adicionaron los programas y ayuda en una barra de estado. Esta última fase se efectuó con la finalidad de hacer más comprensible y amigable el sistema y preparar al personal para hacer una prueba en paralelo durante las inscripciones del periodo 96-1. Para asegurar la eficiencia de CONEFESZ se aplicó este tipo de pruebas el cual tiene como objetivo "asegurar que el sistema pueda manejar el volumen de datos y transacciones de entrada especificados en el modelo de implantación de usuario, y asegurar que tenga el tiempo de respuesta requerido".

Habiendo terminado de programar los módulos subordinados se solicitaron los catálogos a la Dirección General de Administración Escolar (DGAE) para cargarlos al sistema. Se hicieron pruebas con volúmenes de 4,000 registros en el catálogo de alumnos, 190,000 registros en el catálogo de historias, y 9 estaciones de trabajo operando simultáneamente para obtener los tiempos de respuesta. En ésta etapa se realizaron modificaciones a los índices de los archivos para hacer más eficiente el desempeño del sistema.

Una vez probados los procesos de funcionamiento y desempeño, se realizó una prueba en paralelo con la carrera de Psicología para las inscripciones del periodo 96-1. Estos resultados fueron enviados a la "DGAE" para ser procesados y posteriormente recibir el diagnóstico correspondiente. En base a esta prueba CONEFESZ logró reducir en un 80 % el tiempo empleado para las inscripciones con respecto al promedio observado en periodo anteriores.

**3.6.8.-Integración de módulos.**

Después del resultado exitoso de las diferentes pruebas antes citadas, procederemos ahora a la descripción de cómo quedaron integrados los diferentes módulos que constituyen al sistema.

El módulo principal contiene los siguientes submódulos:

**Módulo de actualización de datos.**

Este módulo se encarga de realizar el proceso de actualización de datos en el archivo de alumnos.

**Módulo de Inscripciones.**

Este módulo se encarga del procesamiento general de toda la información que los procesos de inscripción requieren para su correcto funcionamiento, estos procesos de inscripción son los siguientes:

- **Inscripción a primer Ingreso.**  
Realiza la inscripción de los alumnos que ingresan a la FESZ por primera vez, éste proceso requiere información procedente de C.U. como es el catálogo de alumnos, en donde se encuentran los datos generales de los alumnos.
  
- **Inscripción de alumnos a exámenes extraordinarios.**  
Este módulo realiza la inscripción de alumnos que deseen inscribirse a algún examen extraordinario en alguna asignatura. Este proceso requiere de la información referente a la historia académica del alumno, procedente de C.U.

- **Reinscripción del alumnado a sus diferentes asignaturas.**  
Este proceso realiza la reinscripción de los alumnos a sus diferentes asignaturas.

Estos procesos requieren de la información que las historias académicas contienen y que necesariamente procede de C.U.

Los tres procesos anteriores requieren además de la información que se cita, la siguiente:

- Declaración de grupos.
- Códigos de Asignaturas.
- Códigos de Carreras.

**Módulo de obtención de listas.**

Este módulo se encarga de realizar todos los procesos que conllevan a la obtención de listados en papel. Este módulo se subdivide a su vez en los siguientes submódulos:

- **Módulo de obtención de listas por grupo.**  
Este módulo está enfocado a la obtención de listados en papel y contiene las siguientes opciones:
  - Por Carrera.
  - Por Grupo específico.
  - Por Asignatura.

### **Capítulo III.**

---

- **Módulo de obtención de listas por Profesor.**

Está enfocado a la obtención de listados por Profesor, y a su vez, éste tiene las siguientes opciones:

- **Por carrera.**
- **Por Grupo específico.**
- **Obtención de listados por Grupo/Profesor.**

Se enfoca a la obtención de listas, haciendo énfasis en la asignación de Profesores a los diferentes grupos.

- **Módulo de obtención de listas oficiales.**

Este módulo está diseñado específicamente para la obtención de listados en los formatos oficiales, y contiene las siguientes opciones:

- **Por Carrera.**
- **Por Grupo.**
- **Por Asignatura.**

- **Módulo de obtención de Avances.**

Este módulo se diseñó para que el usuario pueda obtener tanto el avance como el historial académico de algún alumno en especial, y consecuentemente contiene las siguientes opciones:

- **Obtención del avance académico del alumno.**
- **Obtención del historial académico del alumno.**

**Módulo de realización de cambios.**

Este módulo está diseñado para la realización y registro de los diversos cambios que se requieran hacer con la información que CONEFESZ maneje o genere, éste a su vez se subdivide en los siguientes submódulos:

- **Realización de cambios de Grupo.**

Este comprende la realización de todos los movimientos que un cambio de grupo genera, como son:

- **Bajas de Alumnos en un grupo.**
- **Altas de Alumnos en el nuevo grupo.**
- **Registro del movimiento.**

- **Realización de cambios de carrera.**

En éste módulo se registra todo el proceso que un cambio de carrera genere, y son:

- **Baja del alumno en la carrera.**
- **Alta del alumno en la nueva carrera.**
- **Registro del movimiento.**

**Módulo de modificación de cupo a grupos.**

Aquí se realiza el proceso de la modificación del cupo máximo de alumnos que puede tener un grupo, con la previa autorización del funcionario correspondiente.



# **CONCLUSIÓN**

Para poder ofrecer una amplia expectativa de servicio a los alumnos y Profesores, la Facultad de Servicios Superiores Zaragoza (FESZ) tomó la decisión de hacer innovaciones. Debido a su transición de una organización totalmente dependiente de Ciudad Universitaria, a otra basada en un esquema en donde la FESZ comienza a tomar vida propia, dependiendo aún en algunos procesos de Ciudad Universitaria, la Institución evaluó e implantó una estrategia totalmente nueva en materia de manejo de información, mediante el uso de un sistema de control escolar que funcionase sobre una red integral de cómputo que la Facultad se encuentra implementando.

Aunque en la FESZ se tenía conciencia de las ventajas que este proyecto ofrece (la realización de todos los trámites que se llevan a cabo en la oficina de servicios escolares de forma automatizada, la agilización de todos los procesos de manejo y control de la información referente a alumnos, profesores, grupos, etc.), existía gran preocupación acerca de la dificultad de manejo y administración del sistema, tiempos de planeación, desarrollo e implementación demasiado prolongados y por lo mismo difíciles de alcanzar; Quizás lo más preocupante era que el software y el equipo de desarrollo de sistemas no se tenía.

## Conclusión

Sin embargo en la FES Zaragoza se aprobó la propuesta de desarrollar el sistema CONEFESZ y se decidió implementar un nuevo concepto cocido como "home run" (hogar en marcha) el cual permite la eliminación o adecuación de los procesos existentes, y la creación de un Sistema de Control Escolar, por lo que no solo se simplificará el mantenimiento y localización de fallas en los procesos, sino que requerirá menos personal para su realización.

El objetivo de la institución al implementar y poner en marcha el sistema CONEFESZ, es alcanzar una notable mejoría en todos los procesos en que se involucre información que genere la Unidad de Servicios Escolares. Dados los resultados en términos de desempeño, eficiencia y sencillez de operación, se planea la adecuación del sistema para que a corto plazo los alumnos, profesores, dirigentes universitarios, etc., puedan hacer cualquier trámite o consulta a través del sistema, que se pretende funcione sobre una red integral de cómputo, conectada a RED-UNAM.

La utilización de lenguajes visuales para el desarrollo de este sistemas, ofrece grandes ventajas ya que: permite presentar mayor cantidad de datos an pantalla lo que repercute en una notable mejoría en la presentación de la información, su fácil manejo elimina la necesidad de una capacitación especializada y costosa, sin embargo, la cantidad de recursos de hardware requeridos, hacen que los sistemas requieran computadoras con procesadores de alto rendimiento.

Esta disyuntiva representa actualmente un obstáculo para muchas empresas e instituciones renuentes a renovar sus equipos, sin considerar que tiene un costo más elevado el desarrollo de sistemas y capacitación en lenguajes sin estas características, que la actualización de los equipos para hacer posible la utilización de lenguajes visuales.

Al realizar esta tesis nos queda la grata experiencia de el trabajo en equipo, pero un equipo integrado no por personas de una escuela ó empresa en particular, sino un equipo de personas que egresan de diferentes escuelas, que laboran en empresas y departamentos dentro de la misma carrera. Todos nosotros con el mismo objetivo.

Las experiencias de uno y otro enriquecen y complementan muchos aspectos de nuestra vida profesional.

Estamos convencidos que el funcionamiento de este sistema (CONFESZ) les facilita tanto a los alumnos como a el personal de Servicios Escolares y de más usuarios los tramites que desean realizar en esta facultad. Seguro es que las largas colas de espera disminuirán de las ventanillas de Servicios Escolares sobre todo en los periodos de reinscripción.

**LUIS MIGUEL MONCAYO LÓPEZ.**

---

# ***MANUAL DE USUARIO***

---

Índice.

Introducción	II
Ejecutar Sistema	IV
Menú Principal	V
Selección de Alumno	VI
Modificar Datos de Alumno	IX
Consultar Asignaturas de Alumno	XI
Reinscripción de Alumno	XII
Cambio de Grupo	XIII
Baja de materias	XVII
Cambio de carrera	XIX
Primer Ingreso	XXI
Modificar Alumno de Primer Ingreso	XXIII
Reportes	XXIV
Altas de Grupos .	XXV
Altas de Materias.	XXVII
Altas Profesores	XXIX
Modificaciones Profesores	XXXI
Bajas Profesores	XXXII
Salida	XXXIII

---

**Cuadros de diálogo:**

Estos cuadros muestran mensajes que el usuario debe contestar.

**Interruptor de selección:**

Estos interruptores le permiten realizar una selección entre distintas opciones de las cuales sólo es posible una al mismo tiempo

**Menú desplegable:**

Son cuadros que presentan una lista de opciones, para ser elegidas por el usuario.

A continuación explicamos los términos básicos asociados con el uso del Mouse.

**Puntero del Mouse:**

El Mouse generalmente es un elemento gráfico (imagen) con forma de flecha.

**Señalar:**

Colocar el puntero sobre un elemento.

**Dar un clic:**

Señale un elemento y a continuación, presione y suelte el botón izquierdo.

**Dar dos clic:**

Señale un elemento y, a continuación, presione y suelte rápidamente el botón izquierdo del Mouse dos veces.

**Arrastrar:**

Señale un elemento. Presione y mantenga presionado el botón izquierdo del Mouse mientras desplace el Mouse a otra posición. Seguidamente, suelte el botón del Mouse.

### Introducción.

El siguiente documento describe los elementos que integran cada una de las pantallas del sistema. Además de cada una de las acciones que el usuario puede realizar en cada pantalla. Dado que hay acciones y elementos que están siempre presentes se describirán primeramente.

**Iconos:**

Elemento gráfico (imagen) que sugiere una acción.

**Botones:**

Son elementos que ejecutan la acción representada por el icono.

**Ventana:**

Son elementos que se presentan en un segmento de la pantalla generalmente después de que el usuario ejecuto una acción previa y continúa su aplicación.

**Barra de títulos:**

Esta es la línea superior de la pantalla y le recuerda en que opción se encuentra.

**Barras de desplazamiento:**

Esta barra le permite desplazar una mayor cantidad de información en la ventanas que despliegan esta.

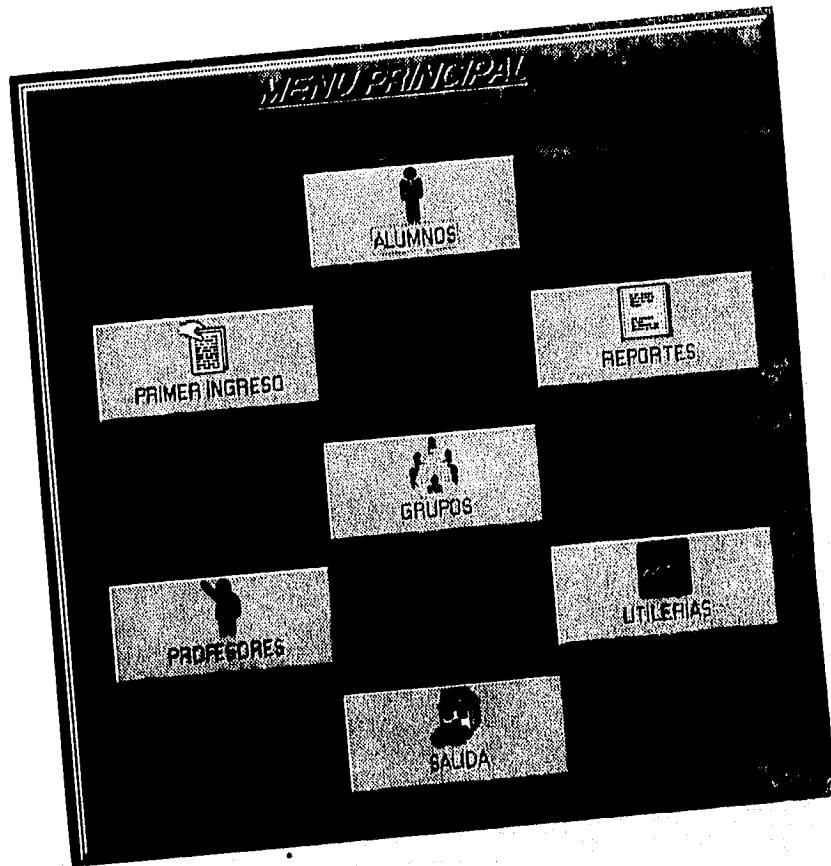
**Barra de estado:**

Esta es la línea inferior de la pantalla. CONEFESZ utiliza esta zona para darle mensajes informativos.

**Cuadros de Texto:**

En estos cuadros el usuario tendrá que teclear el texto correspondiente.





## Documentación

---

Hay tres maneras de Seleccionar un botón de Menú:

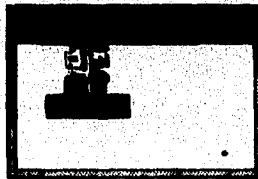
1. Dando un clic en el botón deseado.
2. Posicionarse mediante la tecla <Tab> o cursores y dar <Enter>
3. Utilizando las teclas rápidas (Alt+<tecla>)

### **Ejecutar Sistema**

Para ejecutar el sistema CONEFESZ. Realice los siguientes pasos:

- 1) Entre al ambiente Windows (Teclee win desde el sistema operativo MS-DOS).
- 2) Del administrador de programas (Program Manager) seleccione el grupo FESZ.

A continuación para activar el sistema de doble clic sobre el icono CONEFESZ.



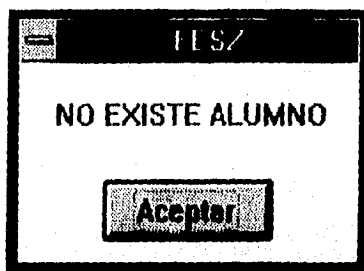
El sistema le pedirá que teclee una clave de acceso. Si la clave es correcta se desplegará el menú principal. De lo contrario se desplegará el mensaje:

"Clave Invalida".

**Nota:**

El sistema CONEFESZ reconoce 2 tipos de claves: "Administradores" y "Operadores". de acuerdo a esto le dará acceso a ciertos módulos.

- 3) Teclee el número de cuenta del alumno que desea hacer algún movimiento y, una vez que se haya capturado el número de cuenta, el usuario deberá confirmar la acción mediante la Tecla <Tab> ó <Enter>.
- 4) Si al confirmar la captura del número de cuenta del alumno aparece la siguiente ventana.



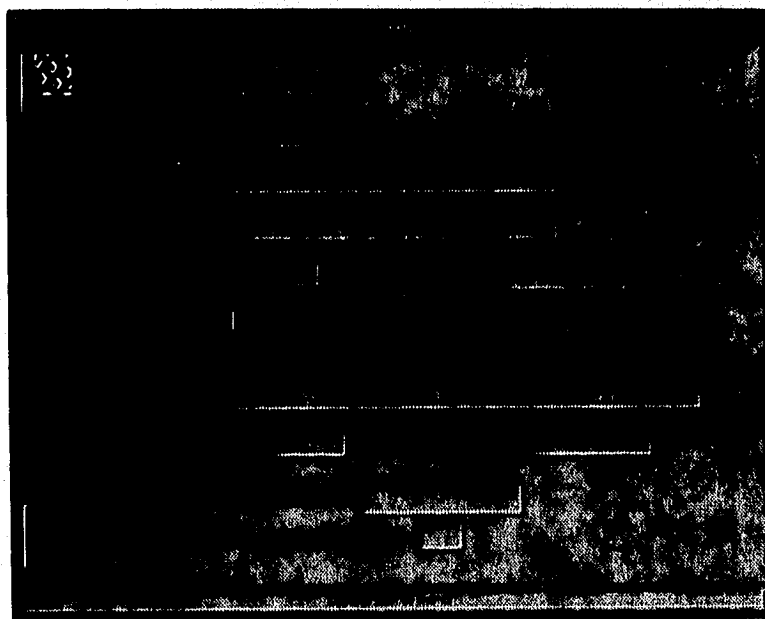
el número de cuenta no se encuentra en la base de datos, verifique. De lo contrario se mostrarán los datos del alumno.

### Selección de un Alumno

- 1) Del menú principal Selecione el botón de alumnos.



Entrará al Modulo de Alumnos desplegándose la siguiente pantalla:



- 2) Seleccione el botón <cuanta>.

Modificar Datos del Alumno.

- 1) Seleccione un alumno.
- 2) Seleccione el Botón Modificar

Al Seleccionar el botón Modificar se desplegará la siguiente pantalla:

The screenshot shows a window titled 'ALUMNO' with a 'Cancelar' button on the left. The form contains the following fields:

- CUENTA: 3027435
- NOMBRE: Juan Pedro del Campo
- CARRERA: LIC. EN PSICOLOGIA
- AÑO DE INGRESO: 19
- NACIONALIDAD: ARGENTINA
- SEXO: M
- FECHA DE NACIMIENTO: 14/05/75
- DIRECCION: Calle loma linda No 54 Col. Colina del Sur
- CODIGO POSTAL: 98760
- Nº. DE TELEFONO: 7775533
- ESCUELA DE PROCEDENCIA: NO INCOR.
- PROGRAMA ESC. DE PROCEDENCIA: 10

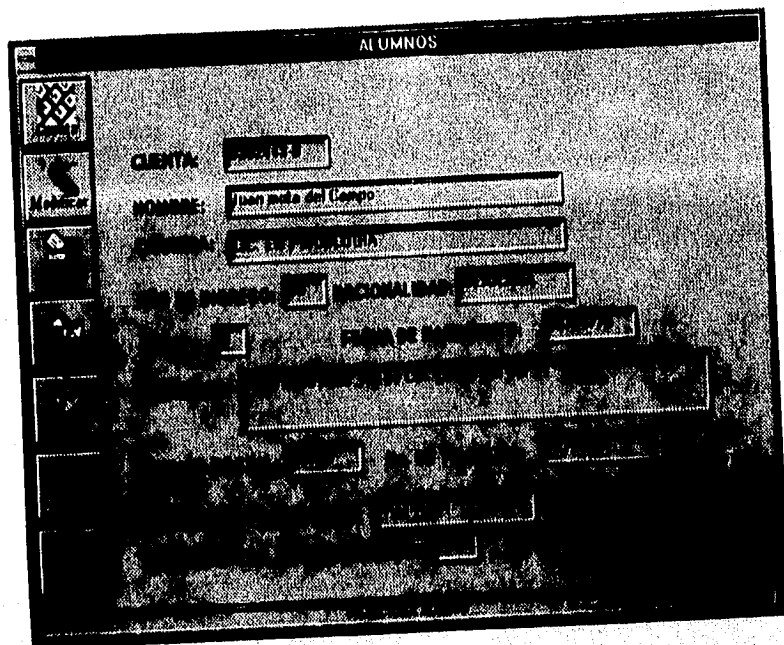
cambie los datos necesarios, solo podrá modificar aquellos cuadros de texto que aparecen con fondo blanco. Para navegar por los cuadros de texto utilice la tecla <Tab> ó el ratón.

En el cuadro de texto:

**sexo:** deberá teclear

F = Femenino

M = Masculino

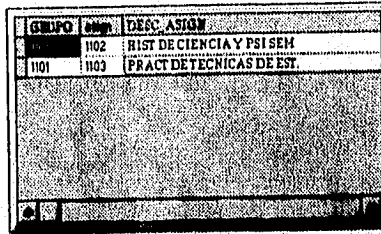


- 6) Si ya se encuentra en la pantalla de ALUMNOS, puede Seleccionar un nuevo alumno dando un clic en el botón cuenta.

**Botón Cancelar:** Si el usuario no desea alterar la información previa, aún cuando ya realizó cambios en los cuadros de texto. De un clic sobre este botón. Esto también lo regresará a la pantalla anterior.

**Consultar Asignaturas de Alumno.**

- 1) Seleccione un Alumno.
- 2) Seleccione el botón Materias. Se desplegará la siguiente ventana:



GRUPO	COD	DESC ASIGN
	1102	HIST DE CIENCIA Y PSISEM
1101	1103	PRACT DE TECNICAS DE EST

La ventana mostrará las materias a las que el alumno en cuestión se haya inscrito. Para regresar el usuario solo necesita dar un clic sobre la ventana anterior.

## Documentación

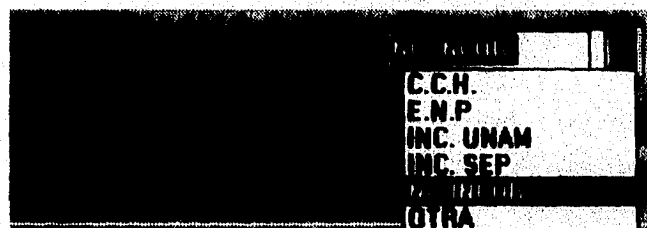
**Fecha de nacimiento:** Este campo es numérico y su formato es el siguiente día/mes/año Ejemplo 29/11/94

**Dirección:** Este campo es de tipo texto. Se recomienda primero posicionarse en el lugar de la corrección y borrar el texto de error para realizar la corrección.

**Código Postal:** Este campo es numérico de 5 cifras. ejemplo: 34529

**No de Teléfono:** Este campo es numérico y almacena hasta 10 cifras.  
ejemplo. 9180167518

**Escuela de Procedencia:** Cuenta con un menú desplegable que se muestra a continuación:



Para activarlo solo de un clic en la flecha de este menú en el cual usted puede elegir la procedencia del alumno dando un clic en el renglón correspondiente.

**Promedio de Esc. de Procedencia:** Este campo es numérico  
ejemplo 9.45

**Botón Grabar:** Si el usuario ya terminó de modificar la información tendrá que Confirmar éstas modificaciones dando un clic sobre éste botón



Si el grupo no es válido se desplegará el siguiente mensaje: "Grupo no existe"

- 5) Seleccione la Asignatura y presione el botón de "Selecciona Asignatura". El sistema realizará las validaciones necesarias para esa asignatura y dará de alta la asignatura en la tira de materia como se muestra.

CLAVE	ASIGNATURA	CRED.
1101	HIST DE CIENCIA Y PSI(SB)	4
1102	HIST DE CIENCIA Y PSI SEM	12
1103		2
1104	MATEMATICAS (TEORIA)	4

Grupo	CLAVE	ASIGNATURA	CRED.
	1102	HIST DE CIENCIA Y PSI SEM	12
1101	1103	FRACCT DE TECNICAS DE EST.	2

- 6) Repita el paso 5 y 4 las veces que sea necesario.
- 7) Para concluir la inscripción de manera correcta deberá dar un clic sobre el botón **Comprobante**. El sistema imprimirá el comprobante y desplegará el siguiente aviso:

Reinscripción de Alumno.

- 1) Seleccione un Alumno
- 2) Seleccione el botón Inscripción, se mostrarán todos los elementos necesarios para la realización de una reinscripción, además de el botón de impresión de comprobante y el botón de regresar a la pantalla anterior. Para continuar con la reinscripción realice los siguientes pasos:
- 3) El usuario deberá especificar el tipo de alumno que en ese momento se va a reinscribir dando un clic en el interruptor de selección:
  - Regular.
  - Irregular.
- 4) Teclee el grupo al cual el alumno desea inscribirse. El sistema responderá con la siguiente ventana, indicando todas las asignaturas a las que el alumno pueda inscribirse y que estén dentro de ese grupo:

1101	FIS DE CIENCIA Y PRIMO	6
1102	FIS DE CIENCIA Y PSI SER	6
1103	PRACT DE TECNICAS DE FIS	2
1104	MATEMATICAS (PRIMA)	6

## Cambio de grupo.

- 1) Seleccione un Alumno.
- 2) Seleccione el Botón Cambios. Se desplegará la siguiente ventana, mostrando las materias en que esta inscrito el alumno.

**CAMBIO DE GRUPO**

**Asignatura a Cambiar ó Baja:**

Grupo:

Clave:

Asignatura:

**Cambiar por:**

Grupo:

Clave:

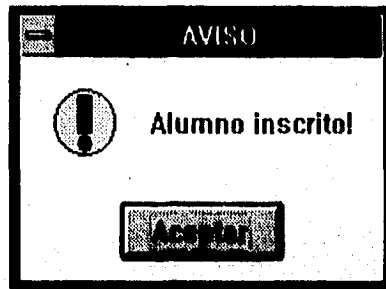
Asignatura:

**MATERIAS ACTUALES**

GRUPO	CLAVE	DESCRIPCIÓN
1102		EST. DE CIENCIA Y EN SEN
2653	1600	ANA Y DES. MIVGRUP (CI)
2653	1601	ANA Y DES. MIVGRUP (SR)
2653	1602	ANA Y DES. MIV. GRU. STIC

**SELECCIÓN DE MATERIA A CAMBIAR O BAJA**

- 3) Deberá seleccionar la asignatura a cambiar dando un clic en la ventana de Materias Actuales. Los datos de esta asignatura se desplegarán en la parte de "Asignatura a Cambiar ó Baja"
- 4) El cursor se posicionará en el cuadro de texto de grupo. Seleccione al grupo al cual desea cambiarse. Se desplegarán en la ventana correspondiente las materias del grupo como se muestra a continuación:



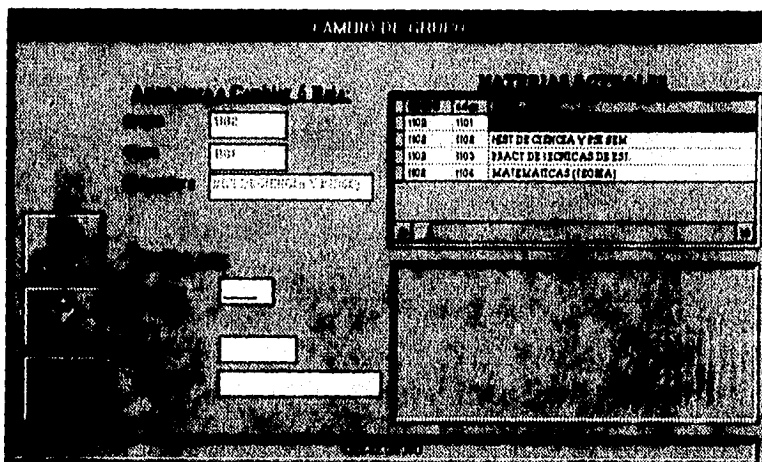
8) De un clic al botón regresar para salir del modulo de Inscripciones.

**NOTA:** En el proceso de "Selecciona Asignatura" se pueden presentar los siguientes mensajes.

<b>Alumno en Artículo 27</b>	<b>El alumno se ha inscrito 2 veces a esta materia</b>
<b>Materia ya aprobada</b>	<b>El alumno ya acreditó la materia seleccionada</b>
<b>Materia ya seleccionada</b>	<b>La materia se seleccionó previamente</b>
<b>Rebasa los créditos permitidos.</b>	<b>Se rebasa el límite de créditos por semestre</b>

Baja de Materia.

- 1) Seleccione un alumno
- 2) Seleccione el Botón Cambios. Se desplegará la siguiente ventana:



- 3) Como en este caso el proceso es de Baja de Asignatura. Deberá seleccionar la asignatura a dar de baja dando un clic en la ventana de **Materias Actuales**. Los datos de esta asignatura se desplegarán en "Asignatura a Cambiar o Baja"
- 4) El botón **Baja** se activará. Para realizar la baja solo de un clic en este botón, se desplegará la siguiente ventana:

**CAMBIO DE GRUPO**

**Asignatura a Cambiar o Borrar:**

Grupo:   
 Clave:   
 Asignatura:

**Cambiar por:**

GRUPO:   
 Clave:   
 Asignatura:

**MATERIAS ACTUALES:**

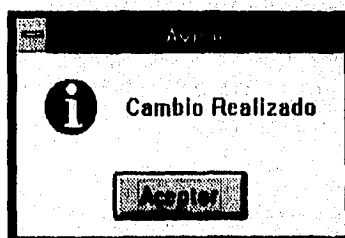
GRUPO	Clave	DESCRIPCION
1101	1102	ANA. Y DES. NIV. GRUPO (CI)
2453	1600	ANA. Y DES. NIV. GRUPO (CI)
2453	1601	ANA. Y DES. NIV. GRUPO (PB)
2453	1602	ANA. Y DES. NIV. GRUPO (SB)

**ASIGNATURAS:**

ASIGNATURA	DESCRIPCION
1100	HIST DE CIENCIA Y PSI(CT)
1101	HIST DE CIENCIA Y PSI(SB)
1102	HIST DE CIENCIA Y PSI(EM)
1103	FRAC DE TECNICAS DE EST.

**SELECCIONE MATERIA A CAMBIAR**

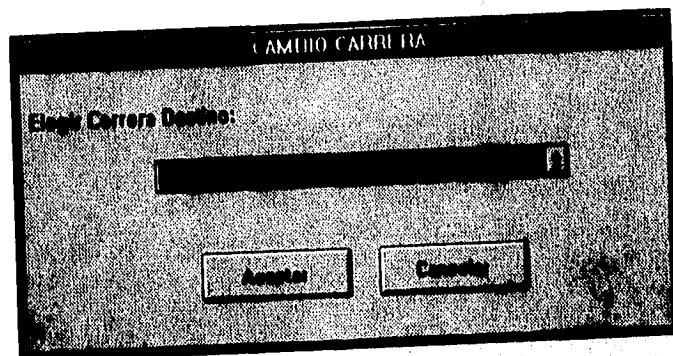
- 5) Seleccione la nueva materia que sustituirá a la marcada anteriormente, dando un clic en ésta. Se desplegarán los datos de esta materia en "Cambiar por" y el botón de Cambio se habilitará en este momento.
- 6) De un clic en el botón "Cambio". Se desplegará el siguiente Aviso.



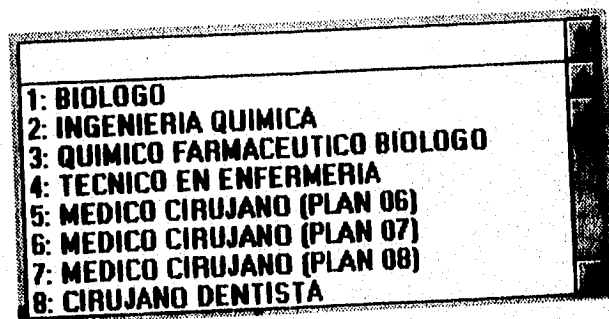
- 7) Si desea realizar más cambios repita el proceso anteriormente descrito.

Cambio de Carrera.

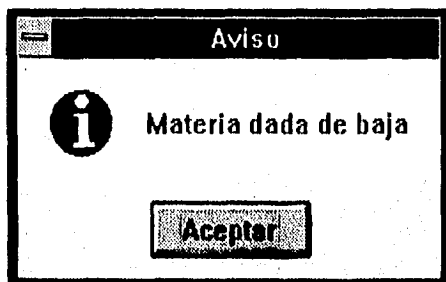
- 1) Seleccioné un Alumno
- 2) Seleccione el botón C Carrera se desplegará la siguiente ventana:



- 3) La ventana presenta un menú desplegable con todas las carreras que se imparten en esta facultad. Para desplegarlo de un clic en la flecha de menú, se desplegará el menú siguiente:



- 4) El usuario puede seleccionar con el Mouse la nueva carrera que cursará el alumno. Al Seleccionaria del menú, se desplegará la nueva carrera:



- 5) Para realizar más bajas de asignaturas repita los pasos 2 y 3. Puede regresar a la pantalla anterior mediante el botón Regresar.



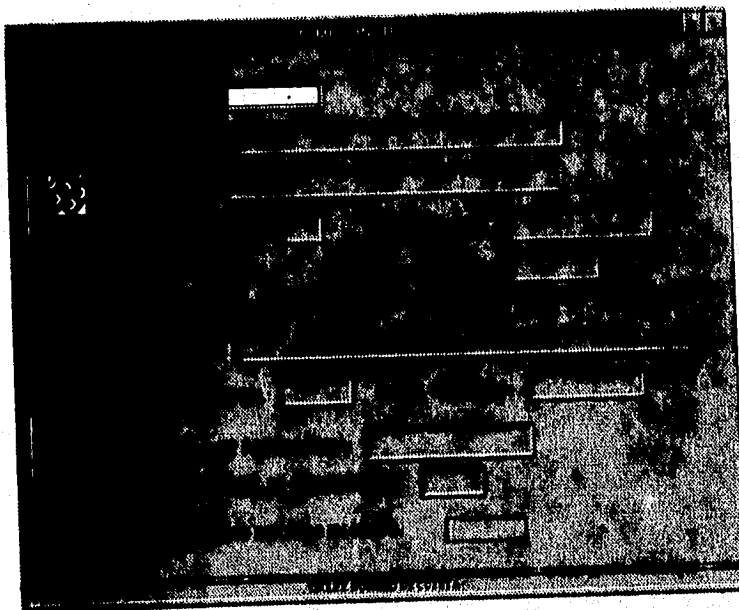
## INSCRIPCIÓN DE ALUMNOS DE PRIMER INGRESO.

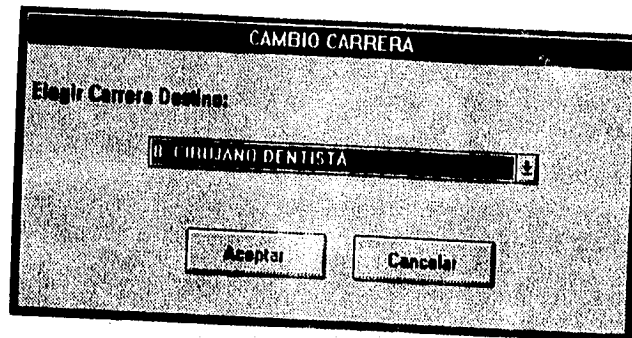
Esta opción está destinada a los alumnos que realizan su inscripción a la FESZ por primera vez.

- 1) Seleccione el botón primer ingreso del menú principal.



- 2) Seleccione el botón Cuenta. Se activará el cuadro de texto cuenta. Teclee el número de cuenta del alumno de primer ingreso.





- 5) De un clic en el botón Aceptar para realizar el cambio.

**Nota:** Esta opción solo esta Disponible para los Administradores del Sistema.

Modificaciones a Alumno de Primer Ingreso.

- 1) Seleccione el botón Modificar. Se desplegará la siguiente pantalla:

The screenshot shows a window titled "PRIMER INGRESO" with a form for entering student data. The fields are as follows:

- Cuenta: 0001073
- Nombre: LUISMI
- Carrera: TECNICO EN ENFERMERIA
- Año de Ingreso: 95
- Nacionalidad: EXTRANJERO
- Sexo: (empty)
- Fecha de Nacimiento: / /
- Dirección: (empty)
- No. de Teléfono: (empty)

At the bottom of the window, there are buttons for "Grabar" and "Cancelar".

En esta pantalla el usuario puede realizar Modificaciones en algunos de los cuadros de texto que contienen los datos generales del alumno de nuevo ingreso

- 2) Teclee las modificaciones en los cuadros de texto, para desplazarse utilice la tecla <Tab> ó el Mouse.
- 3) Confirme las modificaciones dando un clic en el botón Grabar. o de un clic en el botón Cancelar.

- 3) Confirme la acción mediante la tecla <Tab> ó <Enter> se desplegará la siguiente pantalla:

The screenshot shows a window titled "PRIMER INGRESO" with the following fields and values:

- CUENTA: 654001-1
- NOMBRE: EDISM
- CARRERA: TECNICO EN ENFERMERIA
- AÑO DE INGRESO: 99
- NACIONALIDAD: EXTRANJERO
- SEXO: (empty)
- FECHA DE NACIMIENTO: (empty)
- DIRECCION: (empty)
- CODIGO POSTAL: (empty)
- Nº. DE TELEFONO: (empty)
- ESCUELA DE PROCEDENCIA: (empty)
- EMPLEADO S.C. DE PROCEDENCIA: (empty)
- ALUMNO REGISTRO EN GRUPO: 123

On the left side of the form, there are four small icons: a flag, a person, a building, and a tree. At the bottom of the window, there is a label "PANTALLA 1000x750".

- 4) Para confirmar la Inscripción del alumno de un clic en el botón **Comprobante**, el sistema dará de alta al alumno y generará su Tira de Materias.

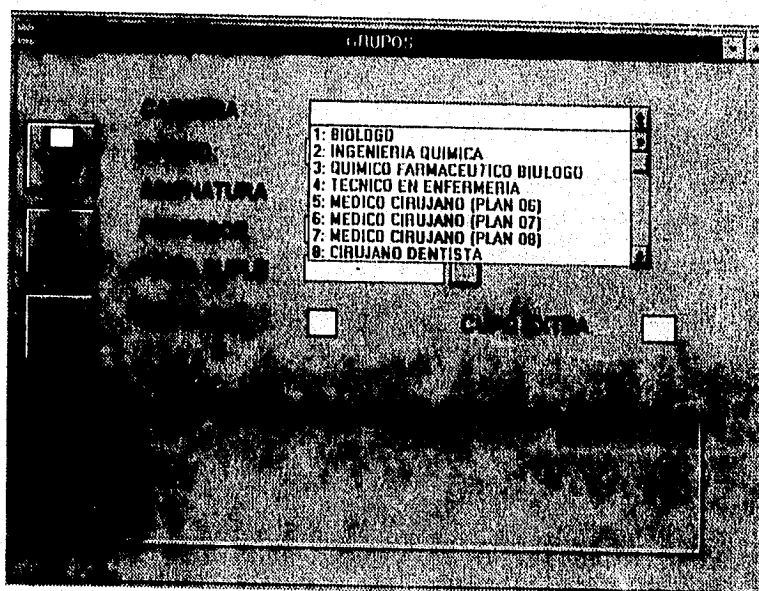
## GRUPOS.

Altas de Grupos. El usuario puede dar de alta nuevos grupos

- 1) Seleccione el botón grupos del menú principal.



- 2) Seleccione del menú desplegable la carrera.



- 4) Confirme la acción con la tecla <Tab>

## REPORTES.

El sistema cuenta con los siguientes reportes:

Historial Académico

Avance Académico

Reportes Por:

Grupo

Carrera

profesor

Asignatura

Listas Oficiales Por:

Carrera

Profesor

Asignatura

Para Imprimir un reporte:

- 1) Seleccione del Menú Principal el Botón **Reportes**.



Alta de materias en un nuevo grupo

- 6) Teclee la información correspondiente en el cuadro de texto Asignatura
- 7) Confirme la acción con la tecla <Tab>
- 8) Teclee el RFC del Profesor. Si lo desea puede utilizar el botón de Menú para desplegar la lista de profesores.

The screenshot shows a window titled "GRUPOS" with a form on the left and a list of professors on the right. The form fields are:

- CARRERA: 7: MEDICO CIRUJANO (P)
- GRUPO: 7201
- ASIGNATURA: 1202 PSIC. PSICOL. M
- PROFESOR: [Empty field]
- PROF. SUPLE: [Empty field]
- CUPO ORD: [Empty field]

On the right, there is a scrollable list of professors with columns for "RFC" and "NOMBRE". The list includes names such as AYALA AGUILAR MODESTO, ALVAREZ BERBER, ARAUJO CONTRERAS, ALVARADO DOMINGUEZ, AYALA ESCOBAR MA, ALTAMIRANO GARCIA, ALCANTARA NIESTA, ALTAMIRANO LOZANO, ARAGON MENDOZA, ALMANZA HUIZ, ALCANTARA ORTIZ, ALVARADO PEÑA CARMEN, ALVARADO PEREZ, ANAYA ROSA JOSE A, AVALES RAMIREZ, AVALES REYES JOSE LUIS, ADESCAL RIVERA RAMON, and ALVAREZ BANCHEZ LAURA.

- 9) Localice el profesor de la asignatura en la ventana mediante la barra de desplazamiento
- 10) Seleccione el profesor de la asignatura en la ventana dando doble clic
- 11) Confirme la Acción con la tecla <Tab>

- 5) Teclee la información correspondiente en el cuadro de texto grupo. Utilice la tecla <Tab> para confirmar la acción.

Si el grupo ya existe se desplegará la información de dicho grupo como lo muestra la siguiente pantalla:

The screenshot shows a software window with a form for entering group information. The form includes the following fields:

- CARRERA: 1: BIOLOGO
- GRUPO: 1101
- ASIGNATURA: ...
- PROFESOR: [ ]
- PROF. SUPLE: [ ]
- ALUMNOS: [ ]
- GRUPO EXTRA: [ ]

Below the form, the text "GRUPO: 1101" is displayed. A table shows the following data:

ASIGNATURA	ALUMNO	NOTA
HIST DE CIENCIA Y PSI (BB)	DURAND ALCANTARA CAMLOS	20
HIST DE CIENCIA Y PSI SEM	MACIAS PEREZ JOSELUIS	20
PRACT DE TECNICAS DE EST.	BALDERAS RAMIREZ ROBERTO	21
MATEMATICAS (TEORIA)	CASTILLO CRUZ CARLOS	25

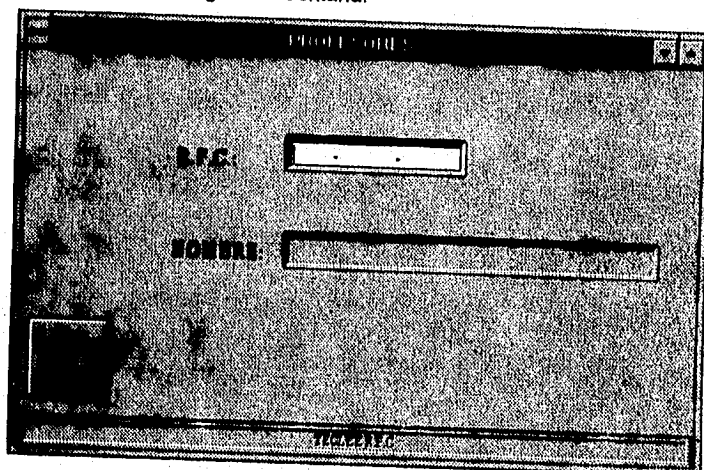


### ALTA DE PROFESORES.

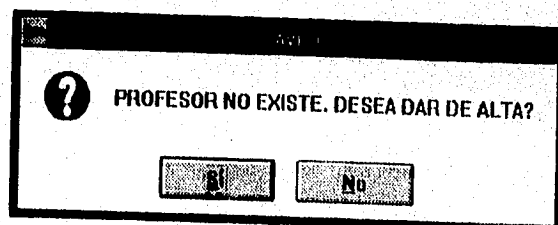
- 1) Seleccione el botón Maestros de el Menú Principal



Se desplegará la siguiente Ventana:



- 2) Teclee el Registro Federal del Contribuyente del profesor. Si el profesor **NO** existe en la base de datos, se desplegará la siguiente ventana.



- 12) Para el Profesor suplente repite el proceso desde el punto 8.
- 13) Teclee la información de cupo ordinario y extraordinario en los cuadros de texto correspondientes
- 14) Confirme los datos de la nueva materia con un clic en el botón Grabar se desplegará los nuevos datos del grupo en la ventana inferior.
- 15) Si existe algún error, colóquese en el cuadro de texto del error y de un clic en el botón Cancelar. Repita los procedimientos. Finalmente deberá Tener una pantalla como la siguiente;

The screenshot shows a software window titled "GRUPO" with the following fields and values:

- CARRERA: 1: BIOLOGO
- GRUPO: 1101
- MATERIA: 1101
- PROFESOR: DUAC540104-XXX
- PROF. SUPLENTE: CUMR540429-XXX
- CUPO ORD: 30

Below the fields is a table with the following data:

	BISI DE CIENCIA Y PSI(58)	EDWARD ALCANTARA CARLOS 26
002	BISI DE CIENCIA Y PSI(58)	MACIAS PEREZ JOSE LUIS 28
003	PPACT DE TECNICAS DE EST.	BALDEAS RAMIREZ ROBERTO 21
004	MATEMATICAS (TEORIA)	CARTILLO CINE CARLOS 25

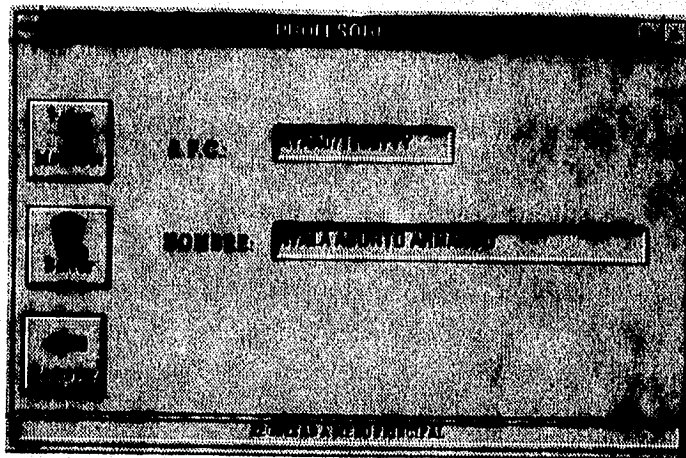
### MODIFICACIONES PROFESORES.

- 1) Seleccione el botón Maestros de el Menú Principal



Se desplegará la 1er Ventana de Profesores:

- 2) Teclee el Registro Federal del Contribuyente del profesor. Si el profesor **YA** existe en la base de datos, se desplegará la siguiente ventana.



- 3) Seleccione el botón Modificar y realice las correcciones correspondientes.
- 4) Confirme las modificaciones con el botón Grabar.

3) De un clic en el Botón Si Para darlo de Alta. Se desplegará la siguiente Ventana:

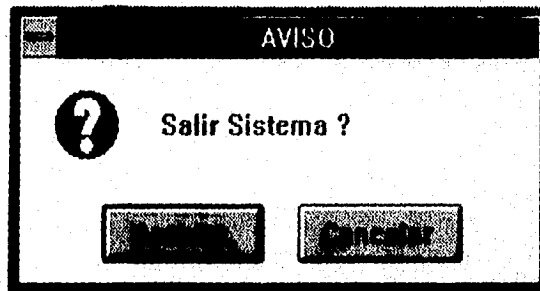
The screenshot shows a window titled "PROFESORES" with a standard Windows-style title bar. On the left side, there are two buttons: a "Cancelar" button with a trash can icon and a "Grabar" button with a floppy disk icon. The main area of the window contains two text input fields. The first field is labeled "R.F.C." and contains the text "AYAA071169YYY". The second field is labeled "NOMBRE" and contains the text "AYALA ABURTO ARMANDO". At the bottom of the window, there is a label that reads "TECLEE EL NOMBRE DEL PROFESOR".

- 4) Teclee el nombre del Profesor en el cuadro de texto correspondiente
- 5) Confirme la Alta dando un clic en el botón Grabar.

SALIDA.



cuando desee dar por concluido el trabajo con el sistema CONEFESZ tendrá que seleccionar el botón <SALIDA> del menú principal. El sistema le responderá con la siguiente ventana:

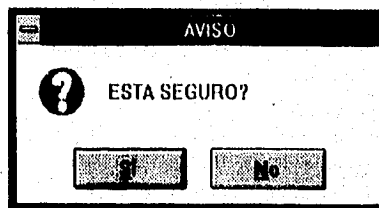


En esta ventana el usuario tiene dos alternativas:

- 1) Seleccionar el botón Aceptar. Para dar por concluido el servicio del sistema.
- 2) Seleccionar el botón Cancelar. Si desea continuar en el Sistema.

**BAJA DE PROFESORES.**

- 1) Repita los pasos 1 y 2 anteriores.
- 2) Seleccione el botón Baja. Se desplegará la siguiente ventana:



- 3) Confirme la acción dando un clic en el botón Si o <Enter> la baja se realizara en la base de datos.

---

***MANUAL DE  
MANTENIMIENTO***

---

Todo sistema que se precie de serio y que maneje información trascendental de la institución en la que se implemente, tendrá la tendencia al crecimiento, es el caso de "CONFESZ", que dada su importancia dentro de la Facultad de Estudios Superiores Zaragoza, su crecimiento anual está calculado en un 7 %, por lo que se tiene que sujetar a diversos procesos de mantenimiento y depuración para que su funcionamiento sea de niveles de excelencia y con esto alcanzar todos o la mayoría de los objetivos y metas que fueron trazadas cuando se comenzó la etapa de planeación del sistema.

A continuación se explican los procesos de mantenimiento y depuración que tienen que ser llevados a cabo para que el sistema funcione dentro de los parámetros convenidos:

#### **MANTENIMIENTO.**

CONFESZ, dentro de la Facultad ocupa un lugar muy importante en lo que a sistemas se refiere, y en consecuencia se tiene que dar mantenimiento constante a los archivos que constituyen la base de datos del sistema, los archivos a los que se les aplica un proceso de mantenimiento constante son los siguientes:

- **ALUMNO.**

A éste archivo se le deberá aplicar un proceso de mantenimiento anual, el cual comprende los siguientes pasos:



## Documentación

---

- *Depuración del archivo.*

En éste punto se eliminan todos los registros de los alumnos que hayan ingresado a la institución diez años atrás, tomando la fecha en que se requiera realizar el proceso de depuración.

Este proceso se realiza con el fin de tener archivados los datos de aquellos alumnos que se consideran como activos dentro de la institución (que no se encuentren afectados por el artículo 19 de la legislación universitaria).

- *Carga de datos de nuevos alumnos.*

En éste proceso se ingresan al archivo los datos generales de aquellos alumnos de nuevo ingreso que Ciudad Universitaria (DGAE) proporciona a la institución. Este proceso se realiza de la siguiente manera:

- 1.- Se cargan los datos de los alumnos de nuevo ingreso a un archivo temporal de alumnos (TALU).
- 2.- Conforme los alumnos se presentan a la ventanilla de servicios escolares (en el evento de inscripciones de primer ingreso) a rectificar sus datos, estos se transfieren del archivo temporal de alumnos (TALU) al archivo permanente de alumnos (ALUMNO) donde quedarán almacenados los datos del alumno para posteriores movimientos que éste desee realizar.

• **HISTORIA.**

A éste archivo se le deberá aplicar un proceso de mantenimiento semestral, el cual comprende los siguientes pasos:

- *Depuración del archivo.*

En éste punto se eliminan todos los registros de los alumnos que hayan ingresado a la institución diez años atrás, tomando la fecha en que se requiera realizar el proceso de depuración.

Este proceso se realiza con el fin de tener archivadas las calificaciones de aquellos alumnos que se consideran como activos dentro de la institución (que no se encuentren afectados por el artículo 19 de la legislación universitaria).

**Carga de nuevas calificaciones de alumnos.**

En éste proceso se agregan al archivo las calificaciones que Ciudad Universitaria (DGAE), proporciona a la institución. Este proceso se realiza de la siguiente manera:

- 1.- Se le da formato a la información, para que el sistema CONEFESZ la pueda entender.
- 2o.- Se cargan las calificaciones de los alumnos a un archivo de historias.

Documentación

---

- **GRUPO.**

A éste archivo se le aplicará un proceso de mantenimiento semestral, el cual consiste en asignar a todos los grupos un cupo para alumnos regulares y un cupo para alumnos irregulares, a su vez, se asigna a cada grupo un Profesor titular y un Profesor asistente, éste proceso se realiza con una lista de asignación de Profesores a grupos con sus respectivos cupos que cada jefatura de carrera hace llegar al departamento de servicios escolares antes de cada evento de inscripción.

- **PROFESOR.**

A este archivo se le aplica un proceso de mantenimiento semestral, el cual consiste en checar que los datos de los Profesores que se encuentren en dicho archivo sean verdaderos.

A su vez se realiza la alta o baja de algún Profesor en la Institución.

PROFESORES

Modificar

R.F.C.: BOSL500130000

Borrar

NOMBRE: SORIANO SANCHEZ LUIS

Regresar

REGRESAR A MENU PRINCIPAL

- **CAMBIOS.**

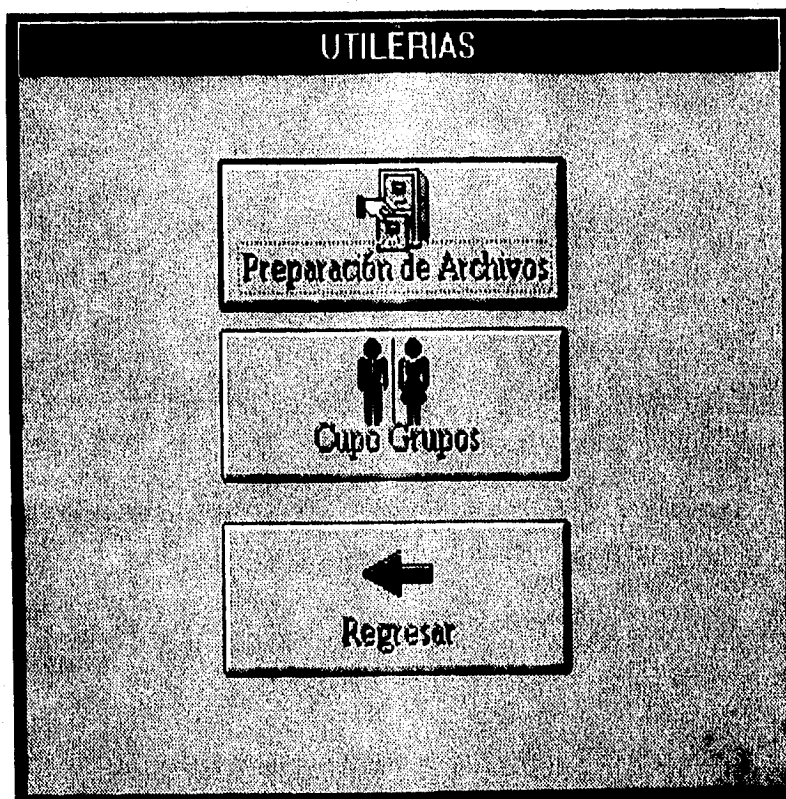
A éste archivo se le aplicará un proceso de mantenimiento semestral, el cual consiste en borrar toda la información que contenga éste archivo.

- **INSCRIPCIONES.**

A éste archivo (INSCRIP) se le aplicará un proceso de depuración de la información que en él se encuentra, que consiste en borrar las inscripciones del periodo anterior, con el fin de no mezclar información de inscripciones de dos periodos diferentes.

Documentación

Estos procesos de depuración se realizan cada semestre y se ejecuta seleccionando el botón de Preparación de Archivos de la Opción de Utilerías



---

***APÉNDICE “A”***  
***PLANES DE ESTUDIO***

---

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
 FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
 UNIDAD DE ADMINISTRACION ESCOLAR  
 PLAN DE ESTUDIOS DE LA CARRERA  
 MEDICO CIRUJANO 512-06

0001	ANATOMIA HUMANA	17
0040	BIOQUIMICA	17
0080	CLINICA I A (NEUMOLOGIA)	15
0081	CLINICA I B (OTORRINOLAR)	3
0082	CLINICA 2 A (CARDIOLOGIA)	15
0083	CLINICA 2 B (OFTALMOLOGIA)	3
0084	CLINICA 3 (TRAUMATOLOGIA)	15
0085	CLINICA 4 A (GASTROENTE)	15
0086	CLINICA 4 B (DERMATOLOGIA)	3
0087	CLINICA 5 A (NEUROLOGIA)	8
0088	CLINICA 5 B (PSIQUIATRIA)	8
0089	CLINICA 6 A (INFECTOLOGIA)	8
0090	CLINICA 6 B (ENDOCRINOLO)	8
0091	CLINICA 7 A (GINECO-OBST)	15
0092	CLINICA 7 B (UROLOGIA)	3
0093	CLINICA 8 (PEDIATRIA)	18
0094	CLINICA 9-10 (MED. CIRUGIA)	72
0160	ECOLOGIA HUMANA	23
0161	EDUCACION TECNICAS QUIRUR.	4
0162	EMBRIOLOGIA HUMANA	8
0200	FARMACOLOGIA	16
0201	FISIOLOGIA HUMANA	17
0280	HISTOLOGIA HUMANA	12
0281	HISTORIA FILOSOF MEDICINA	3
0320	INTRODUCCION CLINICA	24
0321	INTROD. A PRACTICA MEDICA	2
0480	MEDICINA FORENSE	3
0481	MEDICINA HUMANISTICA	6
0482	MEDICINA PREVENTIVA	6
0483	MEDICINA SOCIAL	6
0520	NASOLOGIA BASICA INTEGRAL	9
0600	PATOLOGIA	13
0601	PSICOLOGIA MEDICA	6
0720	SALUD PUBLICA	6

Apéndice "A"

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
UNIDAD DE ADMINISTRACION ESCOLAR  
PLAN DE ESTUDIOS DE LA CARRERA  
MEDICO CIRUJANO 512-07

1100	INTODUCCION A LA MEDICINA	22
1101	ECOLOGÍA	18
1102	NUTRICIÓN	21
1103	CREC. Y DES. INTRAUTERINO	21
1104	PARTO PER. PERINATAL	12
1105	CREC. Y DES. EXTRAUTERINO	22
1300	INTROD. A LA CLINICA	19
1301	GENT. OBSTET. Y DESA.	40
1302	VIDA DE RELACION	76
1303	INTRODUCCION	14
1304	PIEL Y MUSCULO ESQ.	8
1305	APARATO REPRODUCTOR	8
1306	APARATO DIGESTIVO	13
1307	APARATO CARDIOVASCULAR	14
1400	APARATO RESPIRATORIO	11
1401	APARATO URINARIO	8
1402	SISTEMA NERVIOSO	11
1403	SISTEMA ENDOCRINO	11
1404	REGU. DE LIQ. Y NUTRIENTE	46
1405	REGULACION DE GASES	10
1406	GINECOLOGIA	7
1407	PEDIATRIA	7
1408	ORTOPEDIA Y TRAUMATOLOGIA	11
1409	LA MEDICINA Y LA SOCIEDAD	2



UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
UNIDAD DE ADMINISTRACION ESCOLAR  
PLAN DE ESTUDIOS DE LA CARRERA  
MEDICO CIRUJANO 512-08

PRIMER AÑO		
1106	LA SALUD DEL HOMBRE Y SU AMB.	50
1107	CREE. Y DES. INTRAUTERINO	20
1108	PARTO PUERPERIO Y PERINATAL	10
1109	CREE. Y DES. EXTRAUTERINO	20
SEGUNDO AÑO		
1212	INTRODUCTORIO	05
1213	PIEL Y MUSC. ESQUELETICO	10
1214	APARATO RESPIRATORIO	10
1215	APARATO CARDIOVASCULAR	20
1216	APARATO DIGESTIVO	15
1217	APARATO UROGENITAL	18
1218	SIST. NERV. Y ORG. DE LOS SENTIDOS	20
1219	SISTEMA ENDOCRINO	18
TERCER AÑO		
1308	ATEN. MED. DEL ADUL EN C. EX.	17
1309	ATEN. MED. DEL N. EN CON. EXT.	17
1310	ATEN. GINEC. Y OBST. EN C. EXT.	17
1311	ATEN. MED. GEN. E INT. DE C. EXT.	17
1312	EST. MED. INT. DE LA FAMILIA	06
1313	TERAPEUTICA	01
1314	FISOPATOLOGIA EXPERIMENTAL	12
1315	SALUD OCUPACIONAL	02

Apéndice "A"

NIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
UNIDAD DE ADMINISTRACION ESCOLAR  
PLAN DE ESTUDIOS DE LA CARRERA  
MEDICO CIRUJANO 512-08

(HOJA 2)

CLAVE	ASIGNATURA	CREDITOS
1410	ATEN. MED. ADUL. EN URG. Y HOSP.	35
1411	ATEN. DEL NIÑO EN URG. Y HOSP.	17
1412	ATEN. GIN. Y OBST. EN URG. Y HOSP.	17
1413	FISIOPATOLOG. EXPERIMENTAL	06
1414	MEDICINA LEGAL EN MÉXICO	06
1500	INTERNADO	
	SERVICIO SOCIAL	70

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
 FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
 UNIDAD DE ADMINISTRACION ESCOLAR  
 PLAN DE ESTUDIOS DE LA CARRERA  
 QUIMICO FARMACEUTICO BIOLOGO

PRIMER SEMESTRE		
1101	SEM. DE PROB. SOCIO ECONOM.	06
1102	MATEMATICAS I	18
1103	QUIMICA I	14
1104	LAB. DE CIENCIA BASICA I	10
SEGUNDO SEMESTRE		
1200	MATEMATICAS II	10
1201	QUIMICA II	12
1202	FISICOQUIMICA I	14
1203	LAB. DE CIENCIA BASICA II	10
TERCER SEMESTRE		
1300	BIOESTADISTICA	12
1301	QUIMICA II	12
1302	FISICOQUIMICA II	14
1303	LAB. DE CIENCIA BASICA III	10
CUARTO SEMESTRE		
1401	MATS. PRIMAS Y SINTESIS DE MED. I	24
1402	BIOQUIM. CELU. Y DE TEJIDOS 1	24

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
UNIDAD DE ADMINISTRACION ESCOLAR  
PLAN DE ESTUDIOS DE LA CARRERA  
BIOLOGO

PRIMER SEMESTRE		
1101	SEM. DE PROB. SOCIOECONOM.	06
1102	MATEMATICAS	18
1103	QUIMICA I	14
1104	LAB. DE CIENCIA BASICA Y	10
SEGUNDO SEMESTRE		
1200	MATEMATICAS II	10
1201	QUIMICQ II	12
1202	FISICOQUIMICA I	14
1203	LAB. DE CIENCIA BASICA II	10
TERCER SEMESTRE		
1300	BIOESTADISTICA	12
1301	QUIMICA III	12
1302	FISICOQUIMICA II	14
1303	LAB. DE CIENCIA BASICA III	10
CUARTO SEMESTRE		
1400	INTROD. A LA BIOL. MOLECULAR	10
1401	DIFERENCIACION	10
1402	LAB. INTEGRAL DE BIOLOGIA I	10
1403	GENETICA	10
1404	SEM. DE BIOLOGIA MOLECULAR	06
1405	TALLER DE DISCUSION	02

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
 FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
 UNIDAD DE ADMINISTRACION ESCOLAR  
 PLAN DE ESTUDIOS DE LA CARRERA  
 QUIMICO FARMACEUTICO BIOLOGO

(HOJA 2)

QUINTO SEMESTRE		
1501	MATS. PRIMAS Y SINTESIS DE MED. II	24
1502	BIOQUIM. CELU. Y DE TEJIDOS II	24
SEXTO SEMESTRE		
1601	TECNOLOGIA FARMACEUTICA I	16
1602	EVA. DE FARMACOS Y MEDICAMENTOS I	14
1603	MICROBIOLOGIA GENERAL I	18
SEPTIMO SEMESTRE		
1701	TECNOLOGIA FARMACEUT. II	14
1702	EVAL. DE FARM. Y MEDIC. II	14
1703	MICROBIOLOGIA GENERAL II	10
1704	BROMATOLOGIA	10
OCTAVO SEMESTRE (FARMACIA)		
1801	TECNOLOGIA FARMACEUTICA III	26
1802	DESARROLLO ANALITICO	12
1803	SEMINARIO DE FARMACIA I	08

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
UNIDAD DE ADMINISTRACION ESCOLAR  
PLAN DE ESTUDIOS DE LA CARRERA  
QUIMICO FARMACEUTICO BIOLOGO

(HOJA 3)

OCTAVO SEMESTRE (CLINICOS)		
1804	GENETICA CLINICA	12
1805	INMUNOLOGIA	15
1806	ANALISIS BIOQUIM. CLINICO I	16
1807	SEMINARIO BIOQUIMICO CLIN. I	03
NOVENO SEMESTRE (FARMACIA)		
1901	MICROBIOLOGIA FARMACEUTICA	14
1902	BIOFARMACIA	12
1903	SEMINARIO DE FARMACIA II	06
1904	DISEÑO Y ESTABIL. DE MEDICAMENTOS	14
NOVENO SEMESTRE (CLINICOS)		
1905	ANALISIS BIOQUIM. CLINICO II	16
1906	SEM. BIOQUIMICO CLINICO II	03
1907	BIOLOGIA MEDICA	27

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
UNIDAD DE ADMINISTRACION ESCOLAR  
PLAN DE ESTUDIOS DE LA CARRERA  
CARRERA: BIÓLOGO  
HOJA (2)

QUINTO SEMESTRE		
1503	LAB. INTEG. DE BIOLOGIA II	10
1506	TAXONOMIA Y EVOLUCION	10
1507	BIOGEOGRAFIA I	10
1508	DIVERSIDAD VEGETAL	10
1509	EMBRIOL. ANAT. FISIOL. ANIM. COMP. I	10
1510	SEMINARIO Y TALLER DE 5o.	04
SEXTO SEMESTRE		
1603	LAB. INTEG. DE BIOL. I	10
1603	ECOLOGIA I	10
1607	BIOGEOGRAFIA II	10
1608	DIVERSIDAD ANIMAL	10
1609	EMBRIOL. ANAT. FISIOL. ANIM. COMP. II	10
1610	SEMINARIO Y TALLER DE 6o.	04
SEPTIMO SEMESTRE		
1701	INTROD. Y PROPAG. EN BIOL.	10
1702	LAB. INTEGRAL DE BIOL. IV	10
1703	ECOLOGIA I	10
1704	SUELO	10
1705	AGUA	10
1706	SEMINARIO Y TALLER DE 7o.	04

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
UNIDAD DE ADMINISTRACION ESCOLAR  
PLAN DE ESTUDIOS DE LA CARRERA  
CARRERA: BIOLOGO

HOJA (3)

OCTAVO SEMESTRE		
1800	ECONOMIA Y ADMINISTRACION	10
1801	LAB. INTEGRAL DE BIOLOGIA	20
1802	PROG. EST. SIN. EN BIOLOGIA	10
1803	MEDIO AMBIENTE Y LEGISLACION	10
1804	SEMINARIO DE CAMPO	04
NOVENO SEMESTRE		
1900	LAB. INTEGRAL DE BIOL. IV	30



UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
 FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
 UNIDAD DE ADMINISTRACION ESCOLAR  
 PLAN DE ESTUDIOS DE LA CARRERA  
 INGENIERO QUIMICO

PRIMER SEMESTRE		
1101	SEM DE PROB SOCIOECONOM	06
1102	MATEMATICAS I	18
1103	QUIMICA I	14
1104	LAB DE CIENCIA BASICA I	10
SEGUNDO SEMESTRE		
1200	MATEMATICAS II	10
1201	QUIMICA II	12
1202	FISICOQUIMICA I	14
1203	LAB. DE CIENCIA BASICA II	10
TERCER SEMESTRE		
1300	BIOESTADISTICA	12
1301	QUIMICA III	12
1302	FISICOQUIMICA II	14
1303	LAB. DE CIENCIA BASIC. III	10
CUARTO SEMESTRE		
1400	ANALISIS DE PROCESOS	48
1403	BALANCE DE MASA Y ENERGIA	12
1404	FENOMENOS DE TRANSPORTE	11
1405	QUIMICA INDUSTRIAL	8
1406	METODOS NUMERICOS	7
1407	LAB. TALLER DE PROYECTOS	10

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
UNIDAD DE ADMINISTRACION ESCOLAR  
PLAN DE ESTUDIOS DE LA CARRERA  
CARRERA: INGENIERO QUIMICO

(HOJA 2)

QUINTO SEMESTRE		
1500	MANEJO DE MATERIALES	48
1503	FLUJO DE FLUIDOS	12
1504	SEPARACION MEC-MEZCLADOS	12
1505	DISEÑO DE EQUIPO	12
1506	LAB. TALLER DE PROYECTOS	12
SEXTO SEMESTRE		
1600	MANEJO DE ENERGIA	48
1604	INGENIERIA DE SERVICIOS	12
1605	INGENIERIA ELECTRICA	12
1606	TRANSFERENCIA DE CALOR	12
1607	LAB. TALLER DE PROYECTOS	12
SEPTIMO SEMESTRE		
1700	PROCESO DE SEPARACION	48
1705	TERMODINAMICA QUIMICA	12
1706	DISEÑO DE EQUIPO DE SEPARACION	12
1707	TRANSFERENCIA DE MASA	12
1708	LAB. TALLER DE PROYECTOS	12
OCTAVO SEMESTRE		
1800	DISEÑO DE PROCESOS	48
1808	INGENIERIA DE REACTORES	12
1809	INGENIERIA DE PROCESOS	12
1810	DINAMICA CONTROL DE PROCE	12
1811	LAB. TALLER DE PROYECTOS	12

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
UNIDAD DE ADMINISTRACION ESCOLAR  
PLAN DE ESTUDIOS DE LA CARRERA  
INGENIERO QUIMICO

(HOJA 3)

NOVENO SEMESTRE		
1900	DESARROLLO DE PROYECTOS	48
1908	INGENIERIA ECONOMICA	12
1909	INGENIERIA DE PROYECTOS	12
1910	ADMINISTRACION DE PROYECTOS	12
1911	LAB. TALLER DE PROYECTOS	12

Apéndice 'A'

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
UNIDAD DE ADMINISTRACIÓN ESCOLAR  
PLAN DE ESTUDIOS DE LA CARRERA  
TÉCNICO EN ENFERMERÍA

PRIMER SEMESTRE		
1100	INTRODUCCIÓN A LA ENFERMERÍA	05
1101	EL HOMBRE Y SU AMBIENTE	16
1102	NUTRICIÓN	11
1103	ENFERMERÍA MATERNO INFANTIL I	11
SEGUNDO SEMESTRE		
1200E	ENFERMERÍA MATERNO INFANTIL II	42
TERCER SEMESTRE		
1300	ENFERMERÍA MATERNO INFANTIL III	48
CUARTO SEMESTRE		
1400	ENFERMERÍA MÉDICO QUIRÚRGICA	48
QUINTO SEMESTRE		
1500	ENFERMERÍA MÉDICO QUIRÚRGICA II	50

**Apéndice "A"**

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
UNIDAD DE ADMINISTRACION ESCOLAR  
PLAN DE ESTUDIOS DE LA CARRERA  
TECNICO EN ENFERMERIA**

<b>PRIMER SEMESTRE</b>		
<b>1100</b>	<b>INTRODUCCION A LA ENFERMERIA</b>	<b>05</b>
<b>1101</b>	<b>EL HOMBRE Y SU AMBIENTE</b>	<b>16</b>
<b>1102</b>	<b>NUTRICIÓN</b>	<b>11</b>
<b>1103</b>	<b>ENFERMERIA MATERNO INFANTIL I</b>	<b>11</b>
<b>SEGUNDO SEMESTRE</b>		
<b>1200E</b>	<b>ENFERMERIA MATERNO INFANTIL II</b>	<b>42</b>
<b>TERCER SEMESTRE</b>		
<b>1300</b>	<b>ENFERMERIA MATERNO INFANTIL III</b>	<b>48</b>
<b>CUARTO SEMESTRE</b>		
<b>1400</b>	<b>ENFERMERIA MEDICO QUIRURGICA</b>	<b>48</b>
<b>QUINTO SEMESTRE</b>		
<b>1500</b>	<b>ENFERMERIA MEDICO QUIRURGICA II</b>	<b>50</b>

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
UNIDAD DE ADMINISTRACION ESCOLAR  
PLAN DE ESTUDIOS DE LA CARRERA  
TECNICO EN ENFERMERIA

(HOJA 2)

SEXTO SEMESTRE		
1600	ENFERMERIA MEDICO QUIRURGICA III	18
1601	ENFERMERIA PSIQUIATRICA	16
1602	ADMN. DE SERVICIOS DE ENFERMERIA	16

Apéndice "A"

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
UNIDAD DE ADMINISTRACION ESCOLAR  
PLAN DE ESTUDIOS DE LA CARRERA  
CIRUJANO DENTISTA

PRIMER SEMESTRE		
1100	MODULO INTRODUCTORIA	07
1101	APARATO ESTOMATOCNATICO	13
1102	SALUD BUCAL	14
1103	PLACA DENTOBACTERIANA	10
SEGUNDO SEMESTRE		
1200	CRECIMIENTO Y DESARROLLO	16
1201	ODONT. PREVENT. Y RESTAURADORA	16
1202	ESTUDIO CLINICO Y RADIOGRAFICO	14
TERCER SEMESTRE		
1300	SISTEMAS DE MANTENIMIENTO	16
1301	PREP. DEL ACTO OPERATIVO	11
1302	TEORIA ODONTOLOGICA I	09
1303	CLINICA INTEGRAL I	09
CUARTO SEMESTRE		
1400	SISTEM. DE REGULACION Y RELACION	16
1401	TEORIA ODONTOLOGICA II	12
1402	CLINICA INTEGRAL II	18

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
UNIDAD DE ADMINISTRACION ESCOLAR  
PLAN DE ESTUDIOS DE LA CARRERA  
CIRUJANO DENTISTA

( HOJA 2)

QUINTO SEMESTRE		
1500	AGRESION Y DEFENSA I	16
1501	TEORIA ODONTOLOGICA III	12
1502	CLINICA INTEGRAL III	18
SEMESTRE		SEXTO
1600	AGRESION Y DEFENSA II	16
1601	TEORIA ODONTOLOGICA IV	12
1602	CLINICA INTEGRAL IV	18
SEPTIMO SEMESTRE		
1700	SEMINARIO DE INTEGRACION I	18
1701	TEORIA ODONTOLOGICA V	12
1702	CLINICA INTEGRAL V	18
OCTAVO SEMESTRE		
1800	SEMINARIO DE INTEGRACION II	18
1801	TEORIA ODONTOLOGICA VI	12
1802	CLINICA INTEGRAL VI	18



Apéndice "A"

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
 FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
 UNIDAD DE ADMINISTRACION ESCOLAR  
 PLAN DE ESTUDIOS DE LA CARRERA  
 LIC. EN PSICOLOGIA

CLAVE	DESCRIPCION	CREDITOS
<b>PRIMER SEMESTRE</b>		
1100	HIST. DE LA CIENCIA Y PSI. (CT)	04
1101	HIST. DE LA CIENCIA Y PSI. (SB)	04
1102	HIST. DE LA CIENCIA Y PSI. (SEM.)	12
1103	PRACTICA DE TECNICAS DE EST.	02
1104	MATEMATICAS (TEORIA)	04
1105	MATEMATICAS (PRACTICA)	02
1106	MORFOL. Y FIS. DEL SIS. NERV. (T)	04
1107	MORFOL. Y FIS. DEL SIS. NERV. (P)	02
<b>SEGUNDO SEMESTRE</b>		
1200	PROCESOS PSICOL. BASICOS (CT)	04
1201	PROCESOS PSICOL. BASICOS (SB)	04
1202	PROCESOS PSICOL. BASICOS SEM.	12
1203	PRACTICAS DE LAB. EXPERIM. I	08
1204	ESTADISTICA DESCRIPTIVA (TEO.)	04
1205	ESTADISTICA DESCRIPTIVA (PRAC.)	02
1206	BASES BIOL. DE CONDUCTA (TEO)	04
1207	BASES BIOL. DE CONDUCTA (PRAC.)	02
<b>TERCER SEMESTRE</b>		
1300	DETEC. EN PROC. PSIC. APLDOS. (CT)	04
1301	DETEC. EN PROC. PSIC. APLDOS. (SB)	04
1302	DETEC. EN PROC. PSIC. APLDOS. (SEM.)	12
1303	PRACTICAS DE LAB. EXPERIM. II	06
1304	PRACTICAS DE DETEC. PROC. PSIC. APL.	04
1305	ESTADISTICA INFERENCIAL (T)	04
1306	ESTADISTICA INFERENCIAL (P)	02

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
 FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
 UNIDAD DE ADMINISTRACION ESCOLAR  
 PLAN DE ESTUDIOS DE LA CARRERA  
 LIC. EN PSICOLOGIA  
 (HOJA 2)

CUARTO SEMESTRE		
1400	ANALISIS Y DISEÑO NIV. INT. (CT)	04
1401	ANALISIS Y DISEÑO NIV. INT. (SB)	04
1402	ANALISIS Y DISEÑO NIV. INT. SEM	12
1403	ANALISIS Y DISEÑO NIV. INT. SER.	10
1404	ANALISIS DE DATOS (TEO)	04
1405	ANALISIS DE DATOS (PRAC.)	02
QUINTO SEMESTRE		
1500	INTERV. Y EVAL. NIV. IND. (CT)	04
1501	INTERV. Y EVAL. NIV. IND. (SB)	06
1502	INTERV. Y EVAL. NIV. IND. SEM	08
1503	INTERV. Y EVAL. NIV. IND. SERV.	10
1504	PROGRAMACION (TEO)	04
1505	INTROD. A LA NEUROPATOL. (T)	04
SEXTO SEMESTRE		
1600	ANALISIS Y DISEÑO NIV. GRUP. (CT)	04
1601	ANALISIS Y DISEÑO NIV. GRUP. (SB)	04
1602	ANALISIS Y DISEÑO NIV. GRUP. SEM.	12
1603	ANALISIS Y DISEÑO NIV. GRUP. SERV.	10
1604	PROGRAMACION (PRAC.)	02
1605	NEUROLOGIA (TEO)	04

Apéndice "A"

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO  
FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA  
UNIDAD DE ADMINISTRACION ESCOLAR  
PLAN DE ESTUDIOS DE LA CARRERA  
LIC. EN PSICOLOGIA

(HOJA 3)

SEPTIMO SEMESTRE		
1700	INTERV. Y EVAL. NIV. GRUP. (CT)	04
1701	INTERV. Y EVAL. NIV. GRUP. (SB)	04
1702	INTERV. Y EVAL. NIV. GRUP. SEM	08
1703	INTERV. Y EVAL. NIV. GRUP. SERV.	10
1704	PROGRAMACION (TEO)	04
1705	PROGRAMACION II (PRAC)	02
OCTAVO SEMESTRE		
1800	ANALISIS Y DISEÑO NIV. COM. (CT)	04
1801	ANALISIS Y DISEÑO NIV. COM. (SB)	04
1802	ANALISIS Y DISEÑO NIV. COM. SEM.	12
1803	ANALISIS Y DISEÑO NIV. COM. SERV.	10
1804	PROBLS. SOCIOECONOM. DE MEX. T.	08
NOVENO SEMESTRE		
1900	INTERV. Y EVAL. NIV. COMUN. (CT)	04
1901	INTERV. Y EVAL. NIV. COMUN. (SB)	04
1902	INTERV. Y EVAL. NIV. COMUN. SEM.	12
1903	INTERV. Y EVAL. NIV. COMUN. SERV.	10
1904	SEMINARIO SE TESIS (PRAC)	02

---

***APÉNDICE “B”***  
***REPORTES DEL***  
***SISTEMA***

---

---

# ***LISTAS DE GRUPO***

---

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
F.E.S.ZARAGOZA

LISTA DE ALUMNOS  
CARRERA (519) LIC. EN PSICOLOGIA

CLAVE: 1800 ASIGNATURA: ANA. Y DIS. NIV COM (CT)

GRUPO: 2801 PROFESOR: AAA

No	CAR	CLV	CUENTA	NOMBRE DEL ALUMNO
001	21	519	8828668-4	ABASCAL CAVAS MARTHA LAURA
002	21	519	9037236-0	ALAVEZ MEJIA ISABEL ESTHER
003	21	519	9016581-4	ALCANTARA CARDENAS LILIA
004	21	519	9017685-2	ALCANTARA MORENO HUGO
005	21	519	8918611-6	ALMANZA TREJO SANDRA AIDA
006	21	519	9018811-0	ALVAREZ AVILA INDIRA OLGA
007	21	519	9017707-9	APARICIO CRISOSTOMO GEORGINA
008	21	519	9023793-3	CALVO CARREON ROSARIO ANGELICA
009	21	519	9016703-8	CASTILLO CAMPOS SOCORRO ELENA
010	21	519	8639130-8	CASTRO PIMENTEL ANTONIO IVAN
011	21	519	8916573-5	CELIS ESPINOSA ROBERTO
012	21	519	9364139-5	CERON LOPEZ ROSA MARTHA
013	21	519	9016730-6	CHAVEZ SOTO BLANCA IVET
014	21	519	8819323-2	CORONA DAVILA ABEL
015	21	519	8517639-3	COSS DIAZ ANA BERTHA
016	21	519	9060332-9	CRUZ BADILLO MARIA TERESA
017	21	519	9019026-5	CRUZ RAMIREZ PATRICIA
018	21	519	9016786-3	DE JESUS PEREZ MARIA ARACELI
019	21	519	8917772-5	FERNANDEZ ORTIZ CLAUDIA IVETT
020	21	519	9022844-9	FIGUEROA PASTRANA BLANCA ESTELA
021	21	519	8819457-6	FLORES CRUZ ALFREDO
022	21	519	9016871-2	GARCIA AVALOS CARMEN NOELIA
023	21	519	9019185-3	GARCIA HERNANDEZ ADRIANA ISABEL
024	21	519	9251181-5	GONZALEZ GUTIERREZ JOSE IBAR
025	21	519	8818294-0	GONZALEZ MATA CAYETANO
026	21	519	9007923-4	GUTIERREZ ALVAREZ CAROLINA
027	21	519	9018100-1	HERNANDEZ VELAZQUEZ ANGELICA
028	21	519	9019363-7	JASSO REYES SUSANA DAYANARA

Apéndice "B"

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
F.E.S.ZARAGOZA

CLAVE: 1800 ASIGNATURA: ANA. Y DIS. NIV COM (CT)

GRUPO: 2801 PROFESOR: AAA

No	CAR	CLV	CUENTA	NOMBRE DEL ALUMNO
029	21	519	8828616-5	LOPEZ ARCINIEGA VERONICA
030	21	519	8918020-0	LOPEZ GARCIA RAUL
031	21	519	9018180-5	LOPEZ REYES MONSERRAT
032	21	519	9019471-3	MANJARREZ TREJO GISELA
033	21	519	9018209-1	MARIN VARGAS ELVIA
034	21	519	9354031-5	MARTINEZ ALVAREZ SILVIA IVETH
035	21	519	9259705-5	MELO LOPEZ WOOLWORTH FILADELPHIA
036	21	519	9017152-7	MENDOZA AGUILAR MA DEL CONSUELO
037	21	519	9020864-6	NAVARRO MONCADA GERARDO
038	21	519	8829199-8	NOGUEDA VALDES LILIANA
039	21	519	9020991-4	ORTEGA AREVALO MARISOL
040	21	519	8917087-4	ORTIZ GUTIERREZ BEATRIZ
041	21	519	9019707-5	PEREZ ALVARADO GLORIA GABRIELA
042	21	519	9354054-2	PIÑON GUTIERREZ LILIANA
043	21	519	9019734-3	PLATA MARIN MARIBEL
044	21	519	9019749-1	RAMIREZ BARRERA MARISOL
045	21	519	8715805-0	RAMIREZ SILVA MIRIAM
046	21	519	9354056-6	RAMIREZ VELAZQUEZ ITZEL VERONICA
047	21	519	9018515-7	RODRIGUEZ MARTINEZ MARLENE
048	21	519	9016284-7	RUBY MENDOZA MARIA YAZMIN
049	21	519	9259808-3	SALDAÑA VEGA MARIA ANGELICA
050	21	519	9020129-3	SANCHEZ JIMENEZ MARIA SELENE
051	21	519	8832837-1	SANCHEZ MANDUJANO SORAYA NAYELI
052	21	519	8821018-9	SANCHEZ POSADAS DIANA BEATRIZ
053	21	519	9027564-9	SANCHEZ ROSAS GRACIELA
054	21	519	8821846-3	TORRES ROJAS CRISTINA
055	21	519	9018655-6	TOVAR ALATRISTE EVELYN
056	21	519	9160903-2	VALDES MURILLO SALVADOR

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
F. E. S. ZARAGOZA

CLAVE: 1801 ASIGNATURA: ANA. Y DIS. NIV COM (SB)

GRUPO: 2801 PROFESOR: BBB

No	CAR	CLV	CUENTA	NOMBRE DEL ALUMNO
001	21	519	8828668-4	ABASCAL CAÑAS MARTHA LAURA
002	21	519	9037236-0	ALAVEZ MEJIA ISABEL ESTHER
003	21	519	9016581-4	ALCANTARA CARDENAS LILIA
004	21	519	9017685-2	ALCANTARA MORENO HUGO
005	21	519	8918611-6	ALMANZA TREJO SANDRA AIDA
006	21	519	9018811-0	ALVAREZ AVILA INDIRA OLGA
007	21	519	9017707-9	APARICIO CRISOSTOMO GEORGINA
008	21	519	9023793-3	CALVO CARREON ROSARIO ANGELICA
009	21	519	9016703-8	CASTILLO CAMPOS SOCORRO ELENA
010	21	519	8839130-8	CASTRO PIMENTEL ANTONIO IVAN
011	21	519	8916573-5	CELIS ESPINOSA ROBERTO
012	21	519	9364139-5	CERON LOPEZ ROSA MARTHA
013	21	519	9016730-6	CHAVEZ SOTO BLANCA IVET
014	21	519	8819323-2	CORONA DAVILA ABEL
015	21	519	8517639-3	COSS DIAZ ANA BERTHA
016	21	519	9060332-9	CRUZ BADILLO MARIA TERESA
017	21	519	9019026-5	CRUZ RAMIREZ PATRICIA
018	21	519	9016786-3	DE JESUS PEREZ MARIA ARACELI
019	21	519	8917772-5	FERNANDEZ ORTIZ CLAUDIA IVETT
020	21	519	9022844-9	FIGUEROA PASTRANA BLANCA ESTELA
021	21	519	8819457-6	FLORES CRUZ ALFREDO
022	21	519	9016871-2	GARCIA AVALOS CARMEN NOELIA
023	21	519	9019185-3	GARCIA HERNANDEZ ADRIANA ISABEL
024	21	519	9251181-5	GONZALEZ GUTIERREZ JOSE IBAR
025	21	519	8818294-0	GONZALEZ MATA CAYETANO
026	21	519	9007923-4	GUTIERREZ ALVAREZ CAROLINA
027	21	519	9018100-1	HERNANDEZ VELAZQUEZ ANGELICA
028	21	519	9019363-7	JASSO REYES SUSANA DAYANARA
029	21	519	8828616-5	LOPEZ ARCINIEGA VERONICA
030	21	519	8918020-0	LOPEZ GARCIA RAUL



Apéndice "B"

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
F. I. S. ZARAGOZA

CLAVE: 1801 ASIGNATURA: ANA. Y DIS. NIV COM (SB)

GRUPO: 2801 PROFESOR: BBB

No	CAR	CLV	CUENTA	NOMBRE DEL ALUMNO
031	21	519	9018180-5	LOPEZ REYES MONSERRAT
032	21	519	9019471-3	MANJARREZ TREJO GISELA
033	21	519	9018209-1	MARIN VARGAS ELVIA
034	21	519	9354031-5	MARTINEZ ALVAREZ SILVIA IVETH
035	21	519	9259705-5	MELO LOPEZ WOOLWORTH FILADELPHIA
036	21	519	9017152-7	MENDOZA AGUILAR MA DEL CONSUELO
037	21	519	9020964-6	NAVARRO MONCADA GERARDO
038	21	519	8829199-8	NOGUEDA VALDES LILIANA
039	21	519	9020991-4	ORTEGA AREVALO MARISOL
040	21	519	8917087-4	ORTIZ GUTIERREZ BEATRIZ
041	21	519	9019707-5	PEREZ ALVARADO GLORIA GABRIELA
042	21	519	9354054-2	PION GUTIERREZ LILIANA
043	21	519	9019734-3	PLATA MARIN MARIBEL
044	21	519	9019749-1	RAMIREZ BARRERA MARISOL
045	21	519	8715805-0	RAMIREZ SILVA MIRIAM
046	21	519	9354056-6	RAMIREZ VELAZQUEZ ITZEL VERONICA
047	21	519	9018515-7	RODRIGUEZ MARTINEZ MARLENE
048	21	519	9016294-7	RUBY MENDOZA MARIA YAZMIN
049	21	519	9259808-3	SALDAÑA VEGA MARIA ANGELICA
050	21	519	9020129-3	SANCHEZ JIMENEZ MARIA SELENE
051	21	519	8832837-1	SANCHEZ MANDUJANO SORAYA NAYELI
052	21	519	8921018-9	SANCHEZ POSADAS DIANA BEATRIZ
053	21	519	9027564-9	SANCHEZ ROSAS GRACIELA
054	21	519	8821646-3	TORRES ROJAS CRISTINA
055	21	519	9018655-6	TOVAR ALATRISTE EVELYN
056	21	519	9160903-2	VALDES MURILLO SALVADOR
057	21	519	8936665-7	VAZQUEZ FERNANDEZ MA DEL CARMEN
058	21	519	8821734-3	YAVEZ PERALTA RITA

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
F.E.S.ZARAGOZA

CLAVE: 1802 ASIGNATURA: ANA. Y DIS. NIV COM (SB)

GRUPO: 2801 PROFESOR: CCC

No	CAR	CLV	CUENTA	NOMBRE DEL ALUMNO
001	21	519	8828668-4	ABASCAL CAÑAS MARTHA LAURA
002	21	519	9037236-0	ALAVEZ MEJIA ISABEL ESTHER
003	21	519	9016581-4	ALCANTARA CARDENAS LILIA
004	21	519	9017685-2	ALCANTARA MORENO HUGO
005	21	519	8918611-6	ALMANZA TREJO SANDRA AIDA
006	21	519	9018811-0	ALVAREZ AVILA INDIRA OLGA
007	21	519	9017707-9	APARICIO CRISOSTOMO GEORGINA
008	21	519	9023793-3	CALVO CARREON ROSARIO ANGELICA
009	21	519	9016703-8	CASTILLO CAMPOS SOCORRO ELENA
010	21	519	8639130-8	CASTRO PIMENTEL ANTONIO IVAN
011	21	519	8916573-5	CELIS ESPINOSA ROBERTO
012	21	519	9364139-5	CERON LOPEZ ROSA MARTHA
013	21	519	9016730-6	CHAVEZ SOTO BLANCA IVET
014	21	519	8819323-2	CORONA DAVILA ABEL
015	21	519	8517639-3	COSS DIAZ ANA BERTHA
016	21	519	9060332-9	CRUZ BADILLO MARIA TERESA
017	21	519	9019026-5	CRUZ RAMIREZ PATRICIA
018	21	519	9016786-3	DE JESUS PEREZ MARIA ARACELI
019	21	519	8917772-5	FERNANDEZ ORTIZ CLAUDIA IVETT
020	21	519	9022844-9	FIGUEROA PASTRANA BLANCA ESTELA
021	21	519	9259796-9	FLORES CHAVEZ LETICIA ADRIANA
022	21	519	8819457-6	FLORES CRUZ ALFREDO
023	21	519	9016871-2	GARCIA AVALOS CARMEN NOELIA
024	21	519	9019185-3	GARCIA HERNANDEZ ADRIANA ISABEL
025	21	519	9251181-5	GONZALEZ GUTIERREZ JOSE IBAR
026	21	519	8818294-0	GONZALEZ MATA CAYETANO
027	21	519	9007923-4	GUTIERREZ ALVAREZ CAROLINA
028	21	519	9018100-1	HERNANDEZ VELAZQUEZ ANGELICA
029	21	519	9019363-7	JASSO REYES SUSANA DAYANARA
030	21	519	8828616-5	LOPEZ ARCINIEGA VERONICA

Apéndice "B"

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
F.E.S.ZARAGOZA

CLAVE: 1802 ASIGNATURA: ANA. Y DIS. NIV COM (SB)

GRUPO: 2801 PROFESOR: CCC

No	CAR	CLV	CUENTA	NOMBRE DEL ALUMNO
031	21	519	8918020-0	LOPEZ GARCIA RAUL
032	21	519	9018180-5	LOPEZ REYES MONSERRAT
033	21	519	9019471-3	MANJARREZ TREJO GISELA
034	21	519	9018209-1	MARIN VARGAS ELVIA
035	21	519	9354031-5	MARTINEZ ALVAREZ SILVIA IVETH
036	21	519	9259705-5	MELO LOPEZ WOOLWORTH FILADELPHIA
037	21	519	9017152-7	MENDOZA AGUILAR MA DEL CONSUELO
038	21	519	9020964-6	NAVARRO MONCADA GERARDO
039	21	519	8829199-8	NOGUEDA VALDES LILIANA
040	21	519	9020991-4	ORTEGA AREVALO MARISOL
041	21	519	8917087-4	ORTIZ GUTIERREZ BEATRIZ
042	21	519	9019707-5	PEREZ ALVARADO GLORIA GABRIELA
043	21	519	9354054-2	PIÑON GUTIERREZ LILIANA
044	21	519	9019734-3	PLATA MARIN MARIBEL
045	21	519	9019749-1	RAMIREZ BARRERA MARISOL
046	21	519	8715805-0	RAMIREZ SILVA MIRIAM
047	21	519	9354056-6	RAMIREZ VELAZQUEZ ITZEL VERONICA
048	21	519	9018515-7	RODRIGUEZ MARTINEZ MARLENE
049	21	519	9016294-7	RUBY MENDOZA MARIA YAZMIN
050	21	519	9259808-3	SALDAIA VEGA MARIA ANGELICA
051	21	519	9020129-3	SANCHEZ JIMENEZ MARIA SELENE
052	21	519	8832837-1	SANCHEZ MANDUJANO SORAYA NAYELI
053	21	519	8921018-9	SANCHEZ POSADAS DIANA BEATRIZ
054	21	519	9027584-9	SANCHEZ ROSAS GRACIELA
055	21	519	8821646-3	TORRES ROJAS CRISTINA
056	21	519	9018655-8	TOVAR ALATRISTE EVELYN
057	21	519	9160903-2	VALDES MURILLO SALVADOR
058	21	519	8936685-7	VAZQUEZ FERNANDEZ MA DEL CARMEN
059	21	519	8821734-3	YAÑEZ PERALTA RITA

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
F.E.S.ZARAGOZA

CLAVE: 1803 ASIGNATURA: ANA Y DIS. NIV COM SERV

GRUPO: 2801 PROFESOR: DDD

No	CAR	CLV	CUENTA	NOMBRE DEL ALUMNO
001	21	519	8828668-4	ABASCAL CAVAS MARTHA LAURA
002	21	519	9037236-0	ALAVEZ MEJIA ISABEL ESTHER
003	21	519	9016581-4	ALCANTARA CARDENAS LILIA
004	21	519	9017685-2	ALCANTARA MORENO HUGO
005	21	519	8918611-6	ALMANZA TREJO SANDRA AIDA
006	21	519	9018811-0	ALVAREZ AVILA INDIRA OLGA
007	21	519	9017707-9	APARICIO CRISOSTOMO GEORGINA
008	21	519	9023793-3	CALVO CARREON ROSARIO ANGELICA
009	21	519	9016703-8	CASTILLO CAMPOS SOCORRO ELENA
010	21	519	8639130-8	CASTRO PIMENTEL ANTONIO IVAN
011	21	519	8916573-5	CELIS ESPINOSA ROBERTO
012	21	519	9364139-5	CERON LOPEZ ROSA MARTHA
013	21	519	9016730-6	CHAVEZ SOTO BLANCA IVET
014	21	519	8819323-2	CORONA DAVILA ABEL
015	21	519	8517639-3	COSS DIAZ ANA BERTHA
016	21	519	9060332-9	CRUZ BADILLO MARIA TERESA
017	21	519	9019026-5	CRUZ RAMIREZ PATRICIA
018	21	519	9016786-3	DE JESUS PEREZ MARIA ARACELI
019	21	519	8917772-5	FERNANDEZ ORTIZ CLAUDIA IVETT
020	21	519	9022844-9	FIGUEROA PASTRANA BLANCA ESTELA
021	21	519	8819457-6	FLORES CRUZ ALFREDO
022	21	519	9016871-2	GARCIA AVALOS CARMEN NOELIA
023	21	519	9019185-3	GARCIA HERNANDEZ ADRIANA ISABEL
024	21	519	9251181-5	GONZALEZ GUTIERREZ JOSE IBAR
025	21	519	8818294-0	GONZALEZ MATA CAYETANO
026	21	519	9007923-4	GUTIERREZ ALVAREZ CAROLINA
027	21	519	9018100-1	HERNANDEZ VELAZQUEZ ANGELICA
028	21	519	9019363-7	JASSO REYES SUSANA DAYANARA
029	21	519	8828616-5	LOPEZ ARCINIEGA VERONICA
030	21	519	8918020-0	LOPEZ GARCIA RAUL
031	21	519	9018180-5	LOPEZ REYES MONSERRAT

Anexo "B"

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
F.E.S. ZARAGOZA

CLAVE: 1802 ASIGNATURA: ANA. Y DIS. NIV COM (SB)

GRUPO: 2801 PROFESOR: CCC

No	CAR	CLV	CUENTA	NOMBRE DEL ALUMNO
031	21	519	8918020-0	LOPEZ GARCIA RAUL
032	21	519	9018180-5	LOPEZ REYES MONSERRAT
033	21	519	9019471-3	MANJARREZ TREJO GISELA
034	21	519	9018209-1	MARIN VARGAS ELVIA
035	21	519	9354031-5	MARTINEZ ALVAREZ SILVIA IVETH
036	21	519	9259705-5	MELO LOPEZ WOOLWORTH FILADELPHIA
037	21	519	9017152-7	MENDOZA AGUILAR MA DEL CONSUELO
038	21	519	9020964-6	NAVARRO MONCADA GERARDO
039	21	519	8829199-8	NOGUEDA VALDES LILIANA
040	21	519	9020991-4	ORTEGA AREVALO MARISOL
041	21	519	8917087-4	ORTIZ GUTIERREZ BEATRIZ
042	21	519	9019707-5	PEREZ ALVARADO GLORIA GABRIELA
043	21	519	9354054-2	PIÑON GUTIERREZ LILIANA
044	21	519	9019734-3	PLATA MARIN MARIBEL
045	21	519	9019749-1	RAMIREZ BARRERA MARISOL
046	21	519	8715805-0	RAMIREZ SILVA MIRIAM
047	21	519	9354056-6	RAMIREZ VELAZQUEZ ITZEL VERONICA
048	21	519	9018515-7	RODRIGUEZ MARTINEZ MARLENE
049	21	519	9016294-7	RUBY MENDOZA MARIA YAZMIN
050	21	519	9259808-3	SALDAÑA VEGA MARIA ANGELICA
051	21	519	9020129-3	SANCHEZ JIMENEZ MARIA SELENE
052	21	519	8832837-1	SANCHEZ MANDUJANO SORAYA NAYELI
053	21	519	8921018-9	SANCHEZ POSADAS DIANA BEATRIZ
054	21	519	9027564-9	SANCHEZ ROSAS GRACIELA
055	21	519	8821646-3	TORRES ROJAS CRISTINA
056	21	519	9018655-6	TOVAR ALATRISTE EVELYN
057	21	519	9160903-2	VALDES MURILLO SALVADOR
058	21	519	8936665-7	VAZQUEZ FERNANDEZ MA DEL CARMEN
059	21	519	8821734-3	YAÑEZ PERALTA RITA

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
F. E. S. ZARAGOZA

CLAVE: 1804 ASIGNATURA: PROB. SOCIOECONO DE MEX T

GRUPO: 2801 PROFESOR: EEE

No	CAR	CLV	CUENTA	NOMBRE DEL ALUMNO
001	21	519	8828668-4	ABASCAL CAÑAS MARTHA LAURA
002	21	519	9037236-0	ALAVEZ MEJIA ISABEL ESTHER
003	21	519	9016581-4	ALCANTARA CARDENAS LILIA
004	21	519	9017685-2	ALCANTARA MORENO HUGO
005	21	519	8918611-6	ALMANZA TREJO SANDRA AIDA
006	21	519	9018811-0	ALVAREZ AVILA INDIRA OLGA
007	21	519	9017707-9	APARICIO CRISOSTOMO GEORGINA
008	21	519	9023793-3	CALVO CARREON ROSARIO ANGELICA
009	21	519	9016703-8	CASTILLO CAMPOS SOCORRO ELENA
010	21	519	8639130-8	CASTRO PIMENTEL ANTONIO IVAN
011	21	519	8916573-5	CELIS ESPINOSA ROBERTO
012	21	519	9364139-5	CERON LOPEZ ROSA MARTHA
013	21	519	9016730-6	CHAVEZ SOTO BLANCA IVET
014	21	519	8819323-2	CORONA DAVILA ABEL
015	21	519	8517639-3	COSS DIAZ ANA BERTHA
016	21	519	9060332-9	CRUZ BADILLO MARIA TERESA
017	21	519	9019026-5	CRUZ RAMIREZ PATRICIA
018	21	519	9016786-3	DE JESUS PEREZ MARIA ARACELI
019	21	519	8917772-5	FERNANDEZ ORTIZ CLAUDIA IVETT
020	21	519	9022844-9	FIGUEROA PASTRANA BLANCA ESTELA
021	21	519	8819457-6	FLORES CRUZ ALFREDO
022	21	519	9016871-2	GARCIA AVALOS CARMEN NOELIA
023	21	519	9019185-3	GARCIA HERNANDEZ ADRIANA ISABEL
024	21	519	9251181-5	GONZALEZ GUTIERREZ JOSE IBAR
025	21	519	8818294-0	GONZALEZ MATA CAYETANO
026	21	519	9007923-4	GUTIERREZ ALVAREZ CAROLINA
027	21	519	9018100-1	HERNANDEZ VELAZQUEZ ANGELICA
028	21	519	9019363-7	JASSO REYES SUSANA DAYANARA
029	21	519	8828616-5	LOPEZ ARCINIEGA VERONICA
030	21	519	8918020-0	LOPEZ GARCIA RAUL
031	21	519	9018180-5	LOPEZ REYES MONSERRAT

Apéndice "B"

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
F.E.S.ZARAGOZA

CLAVE: 1803 ASIGNATURA: ANA. Y DIS. NIV COM SERV

GRUPO: 2801 PROFESOR: DDD

No	CAR	CLV	CUENTA	NOMBRE DEL ALUMNO
032	21	519	9019471-3	MANJARREZ TREJO GISELA
033	21	519	9018209-1	MARIN VARGAS ELVIA
034	21	519	9354031-5	MARTINEZ ALVAREZ SILVIA IVETH
035	21	519	9259705-5	MELO LOPEZ WOOLWORTH FILADELPHIA
036	21	519	9017152-7	MENDOZA AGUILAR MA DEL CONSUELO
037	21	519	9020964-6	NAVARRO MONCADA GERARDO
038	21	519	8829199-8	NOGUEDA VALDES LILIANA
039	21	519	9020991-4	ORTEGA AREVALO MARISOL
040	21	519	8917087-4	ORTIZ GUTIERREZ BEATRIZ
041	21	519	9019707-5	PEREZ ALVARADO GLORIA GABRIELA
042	21	519	9354054-2	PIÑON GUTIERREZ LILIANA
043	21	519	9019734-3	PLATA MARIN MARIBEL
044	21	519	9019749-1	RAMIREZ BARRERA MARISOL
045	21	519	8715805-0	RAMIREZ SILVA MIRIAM
046	21	519	9354056-6	RAMIREZ VELAZQUEZ ITZEL VERONICA
047	21	519	9018515-7	RODRIGUEZ MARTINEZ MARLENE
048	21	519	9016294-7	RUBY MENDOZA MARIA YAZMIN
049	21	519	9259808-3	SALDAIA VEGA MARIA ANGELICA
050	21	519	9020129-3	SANCHEZ JIMENEZ MARIA SELENE
051	21	519	8832837-1	SANCHEZ MANDUJANO SORAYA NAYELI
052	21	519	8921018-9	SANCHEZ POSADAS DIANA BEATRIZ
053	21	519	9027564-9	SANCHEZ ROSAS GRACIELA
054	21	519	8821646-3	TORRES ROJAS CRISTINA
055	21	519	9018655-6	TOVAR ALATRISTE EVELYN
056	21	519	9160903-2	VALDES MURILLO SALVADOR
057	21	519	8936885-7	VAZQUEZ FERNANDEZ MA DEL CARMEN
058	21	519	8821734-3	YAÑEZ PERALTA RITA

Declaración Grupo/Profesor

CARR	GRUPO	ASIGN	NOMBRE DE LA ASIGNATURA	ALUMNOS
519	1101 1100	HIST DE CIENCIA Y PSI(CT)	57	21
		PROFESOR(ES): VIGUERAS ALVAREZ SILVIA		VIAS520627-XXX

CARR	GRUPO	ASIGN	NOMBRE DE LA ASIGNATURA	ALUMNOS
519	1101 1101	HIST DE CIENCIA Y PSI(SB)	56	21
		PROFESOR(ES): ANZURES AGUILAR MERCEDES		AUAM511203-XXX

CARR	GRUPO	ASIGN	NOMBRE DE LA ASIGNATURA	ALUMNOS
519	1101 1102	HIST DE CIENCIA Y PSI SEM	56	21
		PROFESOR(ES): ANZURES AGUILAR MERCEDES		AUAM511203-XXX

CARR	GRUPO	ASIGN	NOMBRE DE LA ASIGNATURA	ALUMNOS
519	1101 1103	PRACT DE TECNICAS DE EST.	56	21
		PROFESOR(ES): MORAL ESTUDILLO ERNESTO DEL		MOEE540116-XXX

CARR	GRUPO	ASIGN	NOMBRE DE LA ASIGNATURA	ALUMNOS
5191101	1104	MATEMATICAS (TEORIA)	57	21
		PROFESOR(ES): ORTIZ ORTIZ JOSE OCTAVIO		OIOO570720-XXX

CARR	GRUPO	ASIGN	NOMBRE DE LA ASIGNATURA	ALUMNOS
519	1101 1105	MATEMATICAS (PRACTICA)	56	21
		PROFESOR(ES): ORTIZ ORTIZ JOSE OCTAVIO		OIOO570720-XXX



*Apéndice "B"*

---

CARR	GRUPO	ASIGN	NOMBRE DE LA ASIGNATURA	ALUMNOS
519	1101 1106	MORFOL Y FIS SIS NERV (T)	PROFESOR(ES): GARCIA NAVA GONZALO	57 21 GANG520521-XXX

---

CARR	GRUPO	ASIGN	NOMBRE DE LA ASIGNATURA	ALUMNOS
519	1101 1107	MORFOL Y FIS SIS NERV (P)	PROFESOR(ES): GARCIA NAVA GONZALO	57 21 GANG520521-XXX

**DECLARACIÓN**  
**GRUPO/PROFESOR**

---

APÉNDICE "B"

---

# ***HISTORIA ACADÉMICA***

---

23/09/96

MONRROE JHONES MARILYN

9016574-2 1 1

F.E.S. ZARAGOZA  
INGENIERIA QUIMICA

505 96  
21 96-2

70	384	18.23	6
			2
0	0	0.00	8
70	384	18.23	9.33

PRIMER SEMESTRE

505 1101 6 SEM DE PROB SOCIO ECONOM	MB OR 96-1 1975668 1153 1
505 1102 18 MATEMATICAS I	S OR 96-1 1975691 1153 1
505 1103 14 QUIMICA I	MB OR 96-1 1975713 1153 1
505 1104 10 LAB DE CIENCIA BASICA I	MB OR 96-1 1975735 1153 1

SEGUNDO SEMESTRE

505 1200 10 MATEMATICAS II	NP OR 96-2 2042759 2204 1
505 1201 12 QUIMICA II	MB OR 96-2 2042774 2204 1
505 1202 14 FISICOQUIMICA I	NA OR 96-2 2042789 2204 1
505 1203 10 LAB DE CIENCIA BASICA II	MB OR 96-2 2042806 2204 1

510

# ***AVANCE ACADÉMICO***

---

23/09/96

MONRROE JHONES MARILYN

9016574-2 1 1

F.E.S. ZARAGOZA  
INGENIERIA QUIMICA

505 96  
21 96-2

70	384	18.23	6
			2
0	0	0.00	8
70	384	18.23	9.33

---

***APÉNDICE “C”***  
***MANEJO DE BASES DE***  
***DATOS EN VISUAL***  
***BASIC***

---

### **Función CreateDatabase ( Edición Profesional Solamente)**

**Descripción:**

Creando una base de datos en Microsoft Access, y retorna una base de datos objeto que es abierta en forma exclusiva de lectura y escritura.

**Sintaxis:**

**Set dbobject = CreateDatabase( dbname [, locale [, options ]])**

**Notas:**

La función CreateDatabase tiene las siguientes partes:

<b>Parte</b>	<b>Descripción</b>
<b>dbobject</b>	Variable objeto tipo Database
<b>dbname</b>	Una cadena que contenga el nombre de la nueva base de datos, así como su localización. Esta base puede ser llamada de la siguiente forma: C:\CONEFESZ.MDB. Si la red la soporta este tipo de llamada, usted puede también especificar un camino diferente para llamar a la base: \\MYSERVER\CONEFESZ\PROGRESS\CONEFESZ.MDB
<b>locale</b>	Una cadena especificando el lenguaje de la información para crear la base de datos. Este no es requerido pero si es incorrecta la información esta parte puede causar un error. Para esto mejor ver la tabla de abajo.



### Apéndice "C"

---

options Un valor numérico de tipo Long indicando una o más opciones. Ver la tabla de abajo

Constante(Local)	Descripción
DB_LANG_GENERAL	English, German, French.
DB_LANG_SPANISH	Spanish, Italian.
DB_LANG_DUTCH	Dutch.
DB_LANG_SWEDFIN	Sweden.
DB_LANG_NORWDAN	Norway.
DB_LANG_ICELANDIC	Iceland.
DB_LANG_NORDIC	Nordic countries (Microsoft Access 1.0 only).

Constante	Valor	Definición
DB_ENCRYPT	2	Desencriptar la base de datos.
DB_VERSION10	1	Crear en Microsoft Access 1.0 un base de datos.

**Tu puedes indicar más de una opción sumando los valores conjuntamente.**

**La función CreateDatabase abre la nueva base de datos de acceso exclusivo.**

**Si la base de datos ya existe ocasionará un error.**

**Ejemplo:**

```
Set DB = CreateDatabase("C:\CONEFESZ.MDB", DB_LANG_GENERAL,)
```

**Función Opendatabase ( Edición Profesional Solamente)**

**Descripción:**

Abre una base de datos existente y regresa un objeto de la Base de Datos.

**Sintaxis:**

```
Opendatabase ( dbname [, exclusive[, readonly [, connect]])
```

**Notas:**

La función Opendatabase tiene las siguientes partes:

**dbname** Es una cadena de caracteres que contiene el nombre de una base de datos existente. Esta base puede ser llamada de la siguiente forma: C:\CONEFESZ.MDB. Si la red la soporta este tipo de llamada, usted puede también especificar un camino diferente para llamar a la base: \\MYSERVER\CONEFESZ\PROGRESS\CONEFESZ.MDB

### **Apéndice "C"**

---

- exclusive** Es una expresión de tipo Booleana. Si es Verdadera, la base de datos se abre para acceso exclusivo ( un solo-usuario) y si es Falso, la base de datos se abre para acceso de multiusuario. El valor por omisión es Falso.
- readonly** Es una expresión de tipo Booleana. Si es Verdadero, la base de datos se abre para acceso de solo-lectura y si es Falso, la base de datos abre para leer y escribir. El valor por omisión es Falso.
- connect** Es una cadena que sirve para abrir una base de datos.

**Si dbname se refiere a una base de datos que ya esta abierta de forma exclusiva (solo-un-usuario) otro usuario, o dbname no existe, ocurre un error.**

**Si usted especificara uso exclusivo y otro usuario ya tiene abierta la base de datos, ocurre un error.**

**Si dbname es una cadena vacía y conecta un "ODBC"; entonces aparecerá un listado de todos los ODBC que existen en ese momento y podrá seleccionar alguno.**

El formato para conectar una base de datos con Visual es:

<b>Microsoft Access</b>	drive:\path\nombre_del_archivo	( none )
<b>dBASE III</b>	drive:\path	dBASE III;
<b>dBASE IV</b>	drive:\path	dBASE IV;
<b>Paradox</b>	drive:\path	Paradox;
<b>Btrieve</b>	drive:\path\file.DDF	Btrieve;
<b>FoxPro 2.0</b>	drive:\path	FoxPro 2.0;
<b>FoxPro 2.5</b>	drive:\path	FoxPro 2.5;
<b>ODBC</b>	El nombre fuente de la base o ODBC; DSN=server una cadena vacía("") DATABASE=defaultdatabase; UID=user; PWD=password; LoginTimeout=seconds	

**ejemplo:**

**Global db As DATABASE**

**Set db = OpenDatabase("C:\CONFESZ.MDB")**

**Método OpenTable ( Edición Profesional Solamente)**

**Descripción:**

**Abre una tabla existente y retorna una tabla objeto.**

## Apéndice "C"

---

### Sintaxis

**Set tableobject = dbject.OpenTable(tablename [, options] )**

### Notas:

El método de Opentable se conforma de las siguientes partes:

Partes	Descripción
<b>tableobject</b>	Una variable objeto tipo Tabla.
<b>dbject</b>	Una variable objeto identificando una base de datos abierta.
<b>tablename</b>	Una cadena que contenga el nombre de una tabla existente en el dbject.
<b>options</b>	Una expresión indicando uno o más opciones del set. Ver tabla inferior.

---

<i>Constante</i>	<i>Valor</i>	<i>Descripción</i>
<b>DB_DENYWRITE</b>	1	Para ambientes multitareas y multiusuarios, en otro tipo de ambientes no se pueden hacer cambios de los registros mientras este abierta la tabla. Usada para propósitos administrativos.
<b>DB_DENYREAD</b>	2	Para ambientes multitareas y multiusuarios, en otro tipo de ambientes no se pueden hacer cambios de los registros mientras este abierta la tabla. Usada para propósitos administrativos.
<b>DB_READONLY</b>	4	No se pueden hacer cambios en los registros de las tablas. Y se puede incrementar la velocidad de algunas operaciones

Si se desea tomar parte en una tabla. Se debe usar el método **CreateDynaset** que crea un subset de los datos. A través de registros en la tabla objeto pueden ser ordenados solo acordando de las claves índices. Tu puedes cortar un Dynaset en algunos campos. El Dynaset puede tener pocos registros que están debajo de la tabla. El método **Seek** es más rápido para encontrar que los métodos en Dynaset.

Gráfica 1. ? Viaja usted dentro de la República Mexicana por auto?  
**Set dynasetobject = { recordset | querydef | CreateDynaset [ options ] }**  
 .....130

Gráfica 2. ? Que tan seguido visita por carretera dentro de la República?  
**Set dynasetobject = database.CreateDynaset( source | options )**  
 .....131

Gráfica 3. ? La clase y diversidad de los servicios que se ofrecen a lo largo de estas carreteras, ha llegado a influir en su decisión de viajar por ella?  
**QueryDef** objeto o una declaración en SQL  
 .....130

Gráfica 4. ? Conoce las innovaciones y mejoras que se han hecho en las carreteras existentes, así como las nuevas carreteras que han sido construidas en los últimos años?  
**Descripción:**  
 .....131

Gráfica 5. ? Al contarse con servicios como mapas en donde se proporcione información general de las principales carreteras, así como de sus servicios, se siente más motivado a viajar por auto?  
**Método CreateDynaset ( Edición Profesional Solamente )**  
 .....132

Gráfica 6. ? Prefiere viajar en auto o por avión?  
**Set dsassign = db.OpenTable("assign")**  
 .....133

Gráfica 7. ? Porque?  
**Set dsgrupo = db.OpenTable("grupo")**  
 .....134

Gráfica 8. ? Considera que el precio de viajar por carretera es razonable, es decir, no es muy caro?  
**Set dsalumno = db.OpenTable("alumno")**  
 .....135

Gráfica 9. ? Alguna vez ha viajado por auto a la zona turística denominada como Mundo Maya?  
**Global dsgrupo** As table  
 .....136

Gráfica 10. ? Hace cuanto tiempo realizó un recorrido por auto a esta zona?  
**Global dsarrera** As table  
 .....137

Gráfica 11. ? Considera que las carreteras en esta zona, cuentan con servicios suficientes y de buena calidad?  
 .....138

Gráfica 12. ? Su principal motivación al viajar por auto es de negocios o de placer (recreación)  
**Ejemplo:**  
 .....139

INDICE DE GRAFICAS Y MAPAS

**Notas:**

El método CreateDynaset consta de las siguientes partes:

<b>Partes</b>	<b>Descripción</b>
<b>Dynasetobject</b>	Variable objeto de tipo Dynaset
<b>Database</b>	Variable objeto que identifica la base de datos abierta o las propiedades de los datos controlados de la base de datos
<b>Recordset</b>	Variable objeto que identifica la existencia de los registros objeto o las propiedades de los datos controlados de los registros
<b>Querydef</b>	Variable objeto que identifica la existencia de objeto QueryDef.
<b>Source</b>	Cadena que contiene el nombre de la existencia de un registro, QueryDef o declaración en SQL.
<b>Options</b>	Expresión numérica que indica uno o más opciones. Los valores se pueden combinar por una suma conjunta entre ellos Ver la tabla de abajo para los valores



Apéndice "C"

---

<i>Constante</i>	<i>Valor</i>	<i>Descripción</i>
<b>DB_DENYWRITE</b>	1	Para ambientes multitareas y multiusuarios. en otro tipo de ambientes no se pueden hacer cambios de los registros mientras este abierta la tabla en el Dynaset. Usada para propósitos administrativos.
<b>DB_READONLY</b>	4	No se pueden hacer cambios en los registros de las tablas. Y se puede incrementar la velocidad de algunas operaciones
<b>DB_APPENDONLY</b>	8	Solo se pueden meter nuevos registros en el Dynaset. No se puede leer o actualizar los registros existentes
<b>DB_INCONSISTENT</b>	16	Actualizar todos los campos del Dynaset, cada vez que estos sean afectados o algún renglón del Dynaset.
<b>DB_CONSISTENT</b>	32	(Default) Actualiza solo aquellos campos que no han sido afectados o renglones del Dynaset.

<i>Constante</i>	<i>Valor</i>	<i>Descripción</i>
DB_SQLPASSTHROUGH	64	Dynaset creado para rutinas de SQL, estos valores causan que las rutinas en SQL sean pasadas al ODBC de la base de datos para procesarlas, Así cuando se crea un Dynaset de las tablas, en SQL Server o Oracle, Ignorando los ODBC remotos de la bases de datos. El Dynaset no es actualizables.

Si el DB\_INCONSISTENT y DB\_CONSISTENT son incluidos, o ninguno de los dos son incluidos, el resultado es por default.

Cuando se aplica un Dynaset o Snapshot, CreateDynaset retorna un nuevo Dynaset conteniendo los registro ya filtrados y ordenadas dependiendo del filtro de los Dynaset's o Snapshot's y las propiedades de ordenamiento. Esto no afecta el contenido del primer Dynaset o Snapshot.

Al crear un Dynaset de un QueryDef que tiene parámetros, se abre el QueryDef, el conjunto de parámetros y entonces se usa el CreateDynaset. Para el conjunto de parámetros de un QueryDef se usa la siguientes sintaxis:

## Apéndice "C"

---

`querydef ! parameter = parametervalue`

Cuando solo se quiere leer un dato y no cambiar el dato, el objeto Snapshot es más rápido que el solo-lectura del objeto Dynaset. Sin embargo un Dynaset de solo-lectura podría reflejar cambios si se hicieran otros cambios, mientras que si se utiliza el Snapshot no tendría cambios.

**Nota:** Si el programa requiere de dos idénticos Dynasets, es más eficiente crear un solo Dynaset y entonces utilizar el método Clone o crear una segunda referencia.

### **Ejemplo:**

`Dim historia_dyn As dynaset`

`Dim historia_s As String`

`historia_s = "SELECT * FROM historia WHERE cuenta =" & com & cuenta_h &  
com`

`Set historia_dyn = db.CreateDynaset(historia_s)`

---

***APÉNDICE “D”***  
***EJEMPLO DE***  
***DESARROLLO EN***  
***VISUAL BASIC***

---

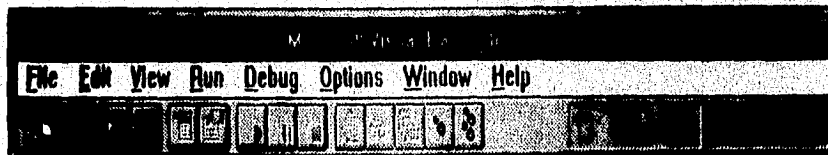
## CREACIÓN DE UNA INTERFACE DE USUARIO.

Para estructurar un interface de usuario vamos a crear un programa que mida el tiempo transcurrido, de manera muy similar a lo que hace un cronómetro. Un cronómetro tiene un botón que pulsa para que comience a contar el tiempo, otro que permite detener la cuenta y un lugar donde ver el tiempo transcurrido. En Visual Basic, un formulario servirá como lugar donde se vea el tiempo transcurrido y para situar los botones de inicio y parada.

Antes de comenzar a describir como se realiza una interface de usuario, se debe haber comprendido a detalle que significado y funciones tienen las siguientes ventanas que Visual Basic presenta al momento de crear un nuevo "programa" (forma):

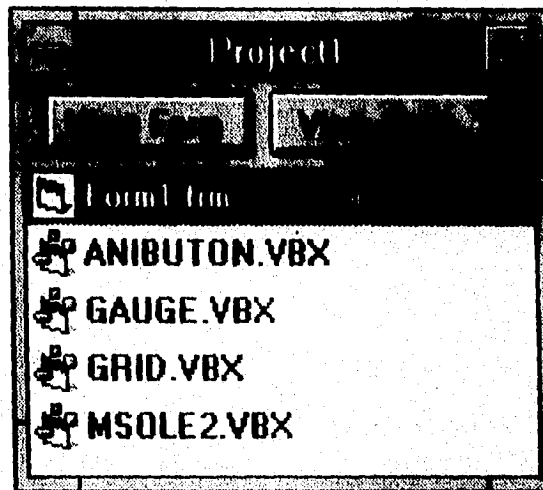
### 1.- La ventana principal.

La ventana principal contiene la barra de menús con ocho menús desplegables. La ventana principal también contiene la barra de herramientas. Los botones de la barra de herramientas son atajos para los comandos que se usan con más frecuencia.



**2.- La ventana de proyecto.**

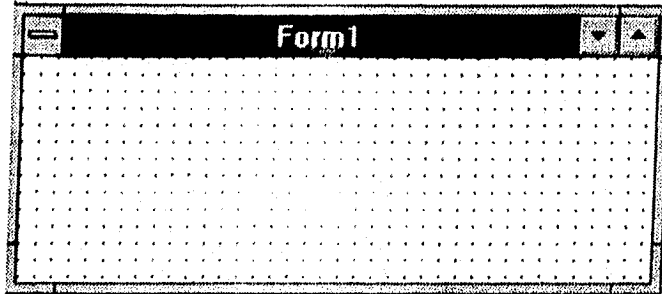
La ventana de proyecto (project) contiene una lista de todos los archivos necesarios para ejecutar el programa de Visual Basic que se esté creando. La ventana de proyecto también contiene dos botones, etiquetados View Form (ver forma) y View Code (Ver código). Por omisión Visual Basic muestra la forma correspondiente cuando se selecciona un archivo en la ventana de proyecto. Esta vista permite diseñar la interface de usuario para alguna aplicación.



**3.- La ventana de la forma y la caja de herramientas.**

Una forma es una zona de visualización que corresponde a una ventana que se verá cuando la aplicación esté funcionando.

Cuando se empieza un proyecto nuevo, Visual Basic crea una forma vacía y la nombra Form1.

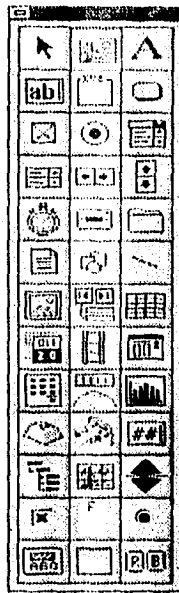


A medida que se diseña la aplicación, la forma sirve como lienzo en el que se puede dibujar diversas partes de la aplicación.

Los componentes de la aplicación que se colocan en la forma se denominan objetos o controles - Cuadros de dibujo, botones de opciones y barras de desplazamiento, por ejemplo.

De hecho, Visual Basic considera a la propia forma como un objeto.

Los controles se crean mediante la paleta de la caja de herramientas, la cual se muestra a continuación, en donde cada control está representado por un icono o herramienta de la caja de herramientas.



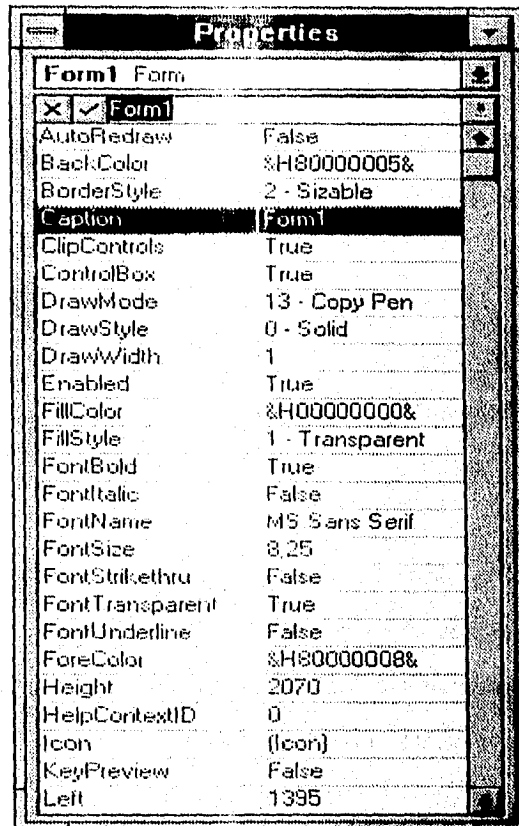
**4.- La ventana de propiedades.**

Las propiedades de Visual Basic son mecanismos formales que sirven para describir los atributos de un objeto.

Todo objeto de Visual Basic tiene propiedades específicas cuyos ajustes controlan la apariencia y comportamiento del objeto en una aplicación. Algunas propiedades están restringidas a ciertos valores. Aunque muchas propiedades pueden alterarse tanto en la fase de diseño como cuando la aplicación está funcionando, la ventana **Propiedades**, que se muestra a continuación, sólo está activa durante la fase de diseño. Se puede activar



ésta ventana de varias maneras: haciendo clic simplemente sobre ella, seleccionando el botón Properties del menú Window (ventana), pulsando F4 o haciendo clic en el botón Properties Window (Ventana propiedades) de la barra de herramientas.



El cuadro de lista desplegable que hay en la parte superior de la ventana de propiedades se denomina el cuadro objeto y muestra el nombre de todos los objetos de la aplicación, así como sus tipos.

## Apéndice "D"

---

Debajo del cuadro objeto se encuentra el cuadro de Ajustes (Settings) y la lista de propiedades. Si se quiere cambiar el valor de alguna propiedad se puede escribir el dato nuevo en el cuadro de ajustes o elegir un nuevo valor predefinido de una lista desplegable, dependiendo de la propiedad en concreto.

Una vez comprendidas la funciones de las diferentes ventanas, podremos comenzar con el diseño de la aplicación.

Cuando se ejecute la aplicación, la forma será presentado en una ventana de aplicación estándar. Si se quiere que esta ventana se parezca a las que muestran otras aplicaciones escritas para windows, debe tener ciertos atributos.

Por ejemplo, uno de los atributos compartidos con las aplicaciones escritas para windows consistiría en que el nombre de la aplicación aparezca en la barra de título. En Visual Basic muchos atributos de ese tipo se controlan mediante las propiedades de los objetos.

## **Apéndice "D"**

---

Debajo del cuadro objeto se encuentra el cuadro de Ajustes (Settings) y la lista de propiedades. Si se quiere cambiar el valor de alguna propiedad se puede escribir el dato nuevo en el cuadro de ajustes o elegir un nuevo valor predefinido de una lista desplegable, dependiendo de la propiedad en concreto.

Una vez comprendidas la funciones de las diferentes ventanas, podremos comenzar con el diseño de la aplicación.

Cuando se ejecute la aplicación, la forma será presentado en una ventana de aplicación estándar. Si se quiere que esta ventana se parezca a las que muestran otras aplicaciones escritas para windows, debe tener ciertos atributos.

Por ejemplo, uno de los atributos compartidos con las aplicaciones escritas para windows consistiría en que el nombre de la aplicación aparezca en la barra de título. En Visual Basic muchos atributos de ese tipo se controlan mediante las propiedades de los objetos.

---

### *Ejemplo de Desarrollo en Visual Basic*

Para comenzar a establecer las propiedades de la forma, en primer lugar, se debe seleccionar la forma como objeto actual haciendo clic en cualquier lugar de la ventana etiquetada como Form1. Ahora se mira la ventana de propiedades de Visual Basic, en la que Form1 aparece en el cuadro de objetos.

Para cambiar el título de la forma se debe modificar la propiedad de "CAPTION" en la ventana de propiedades. A la izquierda del cuadro de Ajustes hay dos botones que están marcados con una "X" y con una marca de confirmación. El botón de confirmación sirve para introducir el valor que se haya establecido en el cuadro de ajustes; hacer clic sobre este botón es equivalente a pulsar la tecla enter. Si se hace clic en el botón "X" se cancela la entrada actual y se restaura el valor anterior en el cuadro de ajustes.

Todos los objetos en Visual Basic tienen una propiedad llamada NAME (nombre). Cuando se establece la propiedad Name se le está dando al objeto una identidad que se puede usar dentro del programa para referirse a él.

## Apéndice "D"

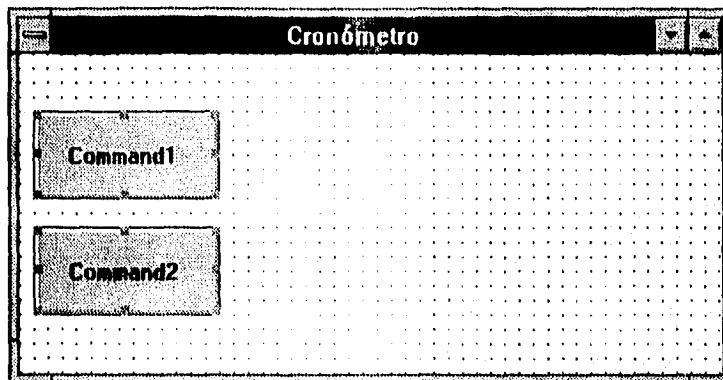
---

### \* ADICIÓN DE OBJETOS.

Volviendo al diseño de la forma, para crear los botones que inicien y detengan la cuenta del tiempo:

- 1.- Se hace clic en la herramienta Command Button de la caja de herramientas, y luego se mueve el puntero del ratón a la zona blanca de la ventana de la forma. El cursor se transformara en una cruz, lo que indica que se está en el modo de dibujo.
- 2.- Se sitúa el cursor en la esquina superior izquierda de la forma.
- 3.- Se pulsa y se mantiene pulsado el botón de ratón y arrastra el cursor hacia abajo y hacia la derecha.
- 4.- Se suelta el botón del ratón.
- 5.- Se deben repetir los pasos 1 a 4, para crear otro botón de comando.

De manera que al término de la realización de los pasos anteriores la forma se verá como la siguiente figura:



Para establecer las propiedades de cada uno de los botones de comando creados anteriormente, se siguen los siguientes pasos:

- 1.- Se selecciona el botón deseado.
- 2.- Se selecciona la propiedad Caption de la lista de propiedades de la ventana Properties.
- 3.- En el cuadro de Ajustes se cambia el título por el de Inicio.
- 4.- Se selecciona el segundo botón y se cambia el ajuste de su propiedad Caption para que ahora sea Parada.

Para referirse a los botones fácilmente, se les debe de cambiar el nombre mediante la modificación de la propiedad NAME. A manera didáctica les pondremos como nombres btnInicio y btnParada respectivamente en la ventana Properties.

## Apéndice "D"

### \* COMPLEMENTACIÓN DE LA VENTANA.

Ahora que ya hemos situado en la pantalla los controles del usuario de la aplicación cronómetro, necesitamos un lugar para mostrar los resultados cuando el programa se esté ejecutando. Para ello, vamos a añadir varias etiquetas a la forma:

- 1.- Se hace clic en la herramienta de etiqueta marcada con una letra "A".
- 2.- Se sitúa la etiqueta donde se desee.
- 3.- Haciendo clic en la ventana de propiedades se selecciona la propiedad de caption y se etiqueta como "INICIO".
- 4.- Usando el mismo método, se crean dos etiquetas más, etiquetándolas y organizándolos como sigue:



Ahora se reservarán los lugares en donde se mostrarán los resultados del programa.. Para ello, vamos a añadir algunas ventanas de texto a la forma:

- 1.- Se hace clic en la herramienta de Text Box marcada con las letras "ab".
- 2.- Se sitúa la caja de texto donde se desee.
- 3.- Haciendo clic en la ventana de propiedades se selecciona la propiedad de text y se etiqueta como cadena en blanco, es decir "".
- 4.- Usando el mismo método, se crean dos etiquetas más, cajas de texto y organizándolas como sigue:





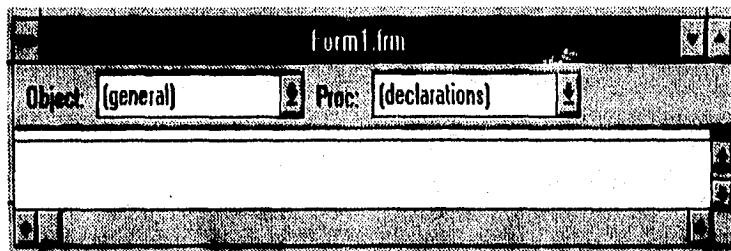
• **CODIFICACIÓN.**

Crear el código es el corazón de la programación. Como ya se han visto las propiedades de los objetos se pueden modificar para que lo afecten. De manera similar, cada objeto puede asociarse con un CONJUNTO de procedimientos que se ejecutan en momentos específicos. Un procedimiento es un grupo de instrucciones en el lenguaje de programación Visual Basic. Las instrucciones del procedimiento se llevan a cabo cuando se ejecuta el procedimiento. Todo el código ejecutable que se escriba se encapsulará en uno u otro tipo de procedimiento.

Todo procedimiento asociado con un objeto está ligado a un evento particular o acción y, por tanto, se le denomina "procedimiento para evento". Los eventos incluyen acciones como Clic, un evento que se produce cuando el usuario hace clic con botón del ratón, y como Resize, un evento que ocurre cuando el usuario cambia el tamaño de una ventana de forma. Los eventos ocurren sólo durante la ejecución del programa, y no durante el diseño. Antes de dar comienzo con la programación, es importante hacer notar que se tiene que tener un nivel aceptable de dominio de programación en Basic.

Como primer paso se deben declarar las variables que se van a utilizar, esto se hace haciendo doble clic en la ventana de la forma y seleccionando en el cuadro

de object general, como se muestra en la siguiente figura:



La sintaxis para declarar variables es la siguiente:

*DIM Nombre de la variable AS Tipo*

En nuestro caso utilizaremos tres variables que son:

**DIM HORAINICIO AS VARIANT**

**DIM HORAPARADA AS VARIANT**

**DIM LAPSUS AS VARIANT**

Una vez que se hayan declarado las variables, pasaremos a la codificación de los diferentes procedimientos que utilizaremos en nuestro programa.

Se selecciona el objeto btnInicio en el cuadro Object de la ventana de código. Primero el procedimiento btnInicio\_click introduciendo las siguientes instrucciones entre la línea Sub btnInicio\_click() y la línea End Sub:

## Apéndice "D"

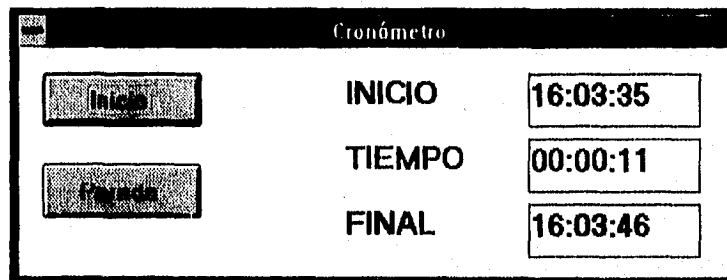
---

```
HORAINICIO = NOW
TXTINICIO.TEXT = FORMAT(HORAINICIO, "HH:MM:SS")
TXTPARADA.TEXT = ""
TXTLAPSUS.TEXT = ""
BTNPARADA.ENABLED = TRUE
BTNINICIO.ENABLED = FALSE
```

Ahora, se selecciona el objeto btnparada en el cuadro Object de la ventana de código, escribiendo entre la línea Sub btnparada\_click() y la línea End Sub las siguientes líneas:

```
HORAPARADA = NOW
LAPSUS = HORAPARADA - HORAINICIO
TXTPARADA.TEXT = FORMAT(HORAPARADA, "HH:MM:SS")
TXTLAPSUS.TEXT = FORMAT(LAPSUS, "HH:MM:SS")
BTNPARADA.ENABLED = FALSE
BTNINICIO.ENABLED = TRUE
```

Los procedimientos que acabamos de escribir se ejecutan cuando el usuario hace clic en el botón de Inicio y Parada respectivamente. Ahora resta solamente saber como se ejecuta el programa, sólo se requiere de que el usuario pulse la tecla de F5 para correr el programa. Para salvar el programa, en el menú de File existe la opción para realizar ésta acción, como en muchos otras utilerías de Microsoft las hay. Como punto final, veremos cómo Visual Basic, muestra los resultados, una vez que el usuario a pulsado la tecla de F5:



Para reforzar un poco más el entendimiento de cómo se comienza a estructurar un programa en Visual Basic, ilustraremos los pasos a seguir para la elaboración de un nuevo proyecto, con distintos componentes, el cual llamaremos "LUISMI AIR LINES".

Cabe hacer notar que algunos pasos, por su sencillez los omitiremos, asumiendo que se ha entendido completamente éstos con la realización del proyecto cronómetro.

El proyecto incluye los tres estilos de cuadro combinado uno al lado del otro. La aplicación simula un diálogo que interroga al usuario sobre sus preferencias personales para un próximo vuelo aéreo.

## Apéndice "D"

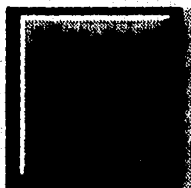
---

Comenzaremos estableciendo las propiedades de la forma seleccionando la forma como objeto actual haciendo clic en cualquier lugar de la ventana etiquetada como Form1. Ahora se mira la ventana de propiedades de Visual Basic, en la que Form1 aparece en el cuadro de objetos.

Tenemos que cambiar el título de la forma, entonces debemos modificar la propiedad de "CAPTION" en la ventana de propiedades.

\* **ADICIÓN DE OBJETOS.**

- 1.- Se crea un proyecto nuevo. Situamos tres etiquetas y dos botones de comando en la forma, estableciendo la propiedad "CAPTION" de cada objeto para que concuerde con los títulos que se muestran en el ejemplo siguiente.
- 2.- Se establece la propiedad default del botón Aceptar a True y la propiedad Cancel del botón cancelar a True.
- 3.- Debajo de la etiqueta destino se dibuja un cuadro combinado usando la herramienta etiquetada con el siguiente símbolo:



Se debe de establecer la propiedad Style a 1 y su propiedad Name como cboDest. Se debe tener cuidado de hacerlo lo suficientemente grande como para mostrar cuatro o cinco líneas de texto en la parte que representa el cuadro de lista.

- 4.- Se crea otro cuadro combinado de la etiqueta Preferencia de Asiento, y se establece la propiedad Style a 2 y la propiedad Name como cboAsiento.

## Apéndice "D"

- 5.- Se crea un cuadro combinado debajo de la etiqueta Preferencia de comida. Se debe también, establecer la propiedad Name del cuadro como cboComida y su propiedad Style a 0. La forma debe de verse como la siguiente:

The image shows a screenshot of a form with a grid background. The form is divided into three columns by vertical dotted lines. The columns are labeled at the top: "DESTINO", "PREFERENCIA DE ASIENTO", and "PREFERENCIA DE COMIDA". Under "DESTINO", there is a text box with the label "CBODEST". Under "PREFERENCIA DE ASIENTO", there is a text box with the label "CBOASIENTO". Under "PREFERENCIA DE COMIDA", there is a dropdown menu with the label "Combo1". At the bottom of the form, there are two dark rectangular buttons.

- 6.- Se necesitará de la escritura de dos procedimientos:
- Uno que informe de los resultados de las acciones del usuario cuando éste haga clic en el botón de comando Aceptar.
  - Y otro para inicializar los cuadros combinados.
- Esto último forma parte del procedimiento Form\_load.

\* **CODIFICACIÓN.**

Como ya dijimos, se necesitarán de dos procedimientos, los cuales listamos a continuación:

```
1.- Sub Command1_Click ()  
    Debug.Print CBODEST.Text  
    Debug.Print CBOASIENTO.Text  
    DEBIG.PRINT CBOCOMIDA.TEXT  
End Sub  
2.- Sub Command2_Click ()  
    End  
End Sub
```



Apéndice "D"

---

**3.- Sub Form\_Load ()**

```
CBODEST.TEXT = ""  
CBODEST.AddItem "OZUMBILLA"  
CBODEST.AddItem "MEXICO"  
CBODEST.AddItem "NUEVA YORK"  
CBODEST.AddItem "TOKYO"  
CBODEST.AddItem "INDIOS VERDES"  
CBODEST.AddItem "LINDAVISTA"  
CBODEST.AddItem "PACHUCA"  
CBODEST.AddItem "SAN PANCHO"  
CBOASIENTO.AddItem "PASILLO"  
CBOASIENTO.AddItem "PISO"  
CBOASIENTO.AddItem "POLISÓN"  
CBOASIENTO.AddItem "COLADO"  
CBOASIENTO.AddItem "COLGADO"  
CBOCOMIDA.AddItem "ENCHILADAS"  
CBOCOMIDA.AddItem "SOPES"  
CBOCOMIDA.AddItem "TACOS"  
CBOCOMIDA.AddItem "PAMBAZOS"  
CBOCOMIDA.AddItem "BARBACHA"  
End Sub
```

• EJECUCIÓN.

Antes de ejecutar el programa, es recomendable guardar el proyecto, usando la opción FILE del menú y luego escogiendo SALVAR COMO. Para poder ejecutar el programa, sólo es necesario pulsar la tecla F5, y el resultado será como el siguiente:

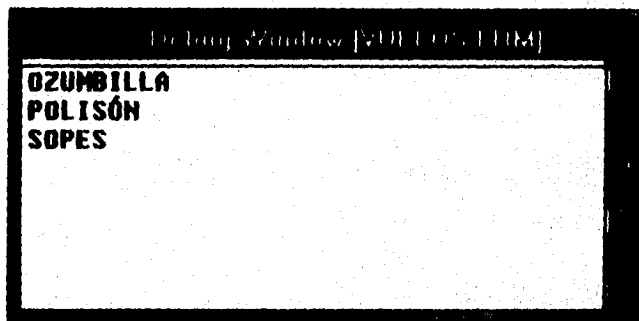
EJEMPLO APLICACION

DESTINO	PREFERENCIA DE ASIENTO	PREFERENCIA DE COMIDA
OZUMBILLA	POLISÓN	BARBACHA
INDIOS VERDES		ENCHILADAS
LINDAVISTA		PAMBAZOS
MEXICO		TACOS
NUEVA YORK		
OZUMBILLA		
PACHUCA		
SAN PANCHO		
TOKYO		

Apéndice "D"

---

Y si observamos la pantalla DEBUG nos mostrará el resultado de nuestra selección, después de haber seleccionado el botón de aceptar, como se muestra a continuación:



---

***APÉNDICE “E”***  
***CÓDIGO FUENTE***

---

```
FORMA: ALUMNOS.FRM
Sub actualiza_captura ()
On Error Resume Next
captura(0).Mask = "?"
captura(1).Mask = "##/##/##"
captura(3).Mask = "#####"
captura(4).Mask = "#####"
captura(0).SelText = DSALUMNO!sexo
captura(1).SelText = Format(DSALUMNO!nacim, "dd/mm/yy")
captura(2).SelText = DSALUMNO!direccion
captura(3).SelText = DSALUMNO!cp
captura(4).SelText = DSALUMNO!tel
captura(5).SelText = DSALUMNO!promedio
PROCEDENCIA.Text = DSALUMNO!esc_proc
End Sub
Sub ACTUALIZA_VALORES ()
BORRA_VALORES
On Error Resume Next
NUMCUENTA = DSALUMNO!CUENTA
DATOS(0).Caption = Format(DSALUMNO!CUENTA, "#####-#")
DATOS(1).Caption = DSALUMNO!ALUMNO
carrera = DSALUMNO!carrera
DATOS(2).Caption = checa_carrera(carrera)
DATOS(3).Caption = DSALUMNO!INGRESO
If DSALUMNO!NACION = 2 Then
    DATOS(4).Caption = "EXTRANJERA"
Else
    DATOS(4).Caption = "MEXICANA"
End If
DATOS(5).Caption = DSALUMNO!sexo
DATOS(6).Caption = Format(DSALUMNO!nacim, "dd/mm/yy")
DATOS(7).Caption = DSALUMNO!direccion
DATOS(8).Caption = DSALUMNO!cp
DATOS(9).Caption = DSALUMNO!tel
DATOS(10).Caption = DSALUMNO!esc_proc
DATOS(11).Caption = DSALUMNO!promedio
End Sub
Sub BORRA_VALORES ()
For t = 0 To 11
    DATOS(t).Caption = ""
Next
End Sub
Sub BOTONES_Click (Index As Integer)
```

## Apéndice "E"

---

```
MENSAJES.Caption = ""
Select Case Index
Case 0 ' SIGUIENTE
    BORRA_VALORES
    Call ESTADO_BOTONES(False, 1, 5)
    CUENTA.Visible = True
    CUENTA.Mask = ""
    CUENTA.Text = ""
    CUENTA.Mask = "#####-#"
    CUENTA.SetFocus
Case 1 ' MODIFICAR DATOS
    VERIFICA(0).Visible = True
    VERIFICA(1).Visible = True
    Call ESTADO_BOTONES(False, 0, 6)
    For t = 0 To 5
        captura(t).Visible = True
        captura(t).Mask = ""
        captura(t).Text = ""
    Next
    PROCEDENCIA.Visible = True
    actualiza_captura
    captura(0).SetFocus
Case 2 ' MATERIAS
    If DSALUMNO!INSC = 0 Then
        c = MsgBox("Alumno no esta inscrito!", 48, "AVISO")
    Else
        Load materias
        materias.Show
    End If
Case 3
    If VALIDA_ART19() Then
        c = MsgBox("Alumno en articulo 19", 48, "AVISO")
    Else
        If DSALUMNO!INSC <> 1 Then
            ALUMNOS.Enabled = False
            Load REINSCRIPCION
            REINSCRIPCION.Show
        Else
            c = MsgBox("Alumno inscrito!", 48, "AVISO")
        End If
    End If
Case 4:
    If DSALUMNO!INSC = 0 Then
        c = MsgBox("EL ALUMNO NO ESTA INSCRITO", 48, "AVISO")
    Else
```

```
Load CGRUPO
CGRUPO.Show
ALUMNOS.Enabled = False
End If
Case 5:
Load CCARRERA
CCARRERA.Show
ALUMNOS.Enabled = False
Case 6:
Call ESTADO_BOTONES(False, 1, 5)
BORRA_VALORES
For t = 0 To 5
captura(t).Visible = False
Next
ALUMNOS.Hide
INICIO.Show
End Select
End Sub
Sub BOTONES_GotFocus (Index As Integer)
Select Case Index
Case 0:
MENSAJES.Caption = "SELECCIONA ALUMNO "
Case 1:
MENSAJES.Caption = "MODIFICAR DATOS DEL ALUMNO"
Case 2:
MENSAJES.Caption = "MUESTRA INSCRIPCION"
Case 3:
MENSAJES.Caption = "INSCRIPCION ORDINARIA, EXTRAORDINARIA"
Case 4:
MENSAJES.Caption = "CAMBIOS Y BAJAS DE MATERIAS"
Case 5:
MENSAJES.Caption = "CAMBIOS DE CARRERA"
Case 6:
MENSAJES.Caption = "REGRESA A MENU PRINCIPAL"
End Select
End Sub
Sub captura_GotFocus (Index As Integer)
Select Case Index
Case 0:
MENSAJES.Caption = "CAPTURE SEXO M O F"
Case 1:
MENSAJES.Caption = "UTILICE EL FORMATO DD/MM/AA"
Case 2:
MENSAJES.Caption = ""
Case Else
```

Apéndice "E"

---

```
    MENSAJES.Caption = ""
End Select
End Sub
Sub cuenta_GotFocus ()
MENSAJES.Caption = "CAPTURE NUMERO DE CUENTA"
End Sub
Sub cuenta_LostFocus ()
If CUENTA.Text <> "" Then
    DSALUMNO.Index = "PRIMARYKEY"
    DSALUMNO.Seek "=", CUENTA.ClipText
    If DSALUMNO.NoMatch Then
        MsgBox ("NO EXISTE ALUMNO")
        CUENTA.Mask = ""
        CUENTA.Text = ""
        CUENTA.Mask = "#####-#"
        CUENTA.SetFocus
    Else
        MENSAJES.Caption = ""
        CUENTA.Visible = False
        BORRA_VALORES
        ACTUALIZA_VALORES
        Call ESTADO_BOTONES (True, 1, 5)
    End If
End If
End Sub
Sub ESTADO_BOTONES (v, i, f As Integer)
For t = i To f
    BOTONES(t).Visible = v
Next
End Sub
Sub Form_Activate ()
    'ACTUALIZA_VALORES
End Sub
Sub Form_Load ()
    PROCEDENCIA.AddItem "C.C.H."
    PROCEDENCIA.AddItem "E.N.P"
    PROCEDENCIA.AddItem "INC. UNAM"
    PROCEDENCIA.AddItem "INC. SEP"
    PROCEDENCIA.AddItem "NO INCOR."
    PROCEDENCIA.AddItem "OTRA"
    Call ESTADO_BOTONES(False, 1, 5)
End Sub
Sub GRABA_DATOS ()
    DSALUMNO.Edit
    DSALUMNO!sexo = captura(0).Text
```



```
DSALUMNO!nacim = captura(1).Text
DSALUMNO!direccion = captura(2).Text
DSALUMNO!cp = captura(3).Text
DSALUMNO!tel = captura(4).Text
DSALUMNO!promedio = Val(captura(5).Text)
DSALUMNO!esc_proc = PROCEDENCIA.Text
DSALUMNO.Update
End Sub
Sub Imagen_Click ()
End Sub
Sub PERIODO ()
End Sub
Sub PROCEDENCIA_GotFocus ()
MENSAJES.Caption = ""
End Sub
Function VALIDA_ART19 () As Integer
Dim fecha As Integer
Dim DURA As Integer
Dim INGRESO As Integer
dtcarrera.Index = "primarykey"
dtcarrera.Seek "=", DSALUMNO!carrera, DSALUMNO!PLANTEL
If dtcarrera.NoMatch Then
MsgBox ("Carrera no existe")
Else
DURA = dtcarrera!duracion
INGRESO = DSALUMNO!INGRESO
fecha = Val(Format(Now, "yy"))
If (INGRESO + (DURA / 2) + (DURA / 4)) >= fecha Then
VALIDA_ART19 = False
Else
VALIDA_ART19 = True
End If
End If
End Function
Sub VERIFICA_Click (Index As Integer)
Select Case Index
Case 0
GRABA_DATOS
VERIFICA(0).Visible = False
VERIFICA(1).Visible = False
Case 1
VERIFICA(0).Visible = False
VERIFICA(1).Visible = False
End Select
For t = 0 To 5
```

## Apéndice "E"

---

```
captura(t).Visible = False
Next
PROCEDENCIA.Visible = False
Call ESTADO_BOTONES(True, 0, 6)
ACTUALIZA_VALORES
End Sub
```

### FORMA: CCARRERA.FRM

```
Dim NCARRERA As Integer
Dim NPLAN As Integer
Sub CAMBIA_CARRERA ()
    dsalumno.Edit
    dsalumno!carrera = NCARRERA
    dsalumno!PLANTEL = NPLAN
    dsalumno.Update
    c = MsgBox("Cambio Realizado", 48, "Aviso")
End Sub
Sub CAP_LostFocus ()
    CADENA = Left(CAP.List(CAP.ListIndex), 1)
    Select Case CADENA
    Case Is = "1"
        NCARRERA = 503
        NPLAN = 22
    Case Is = "2"
        NCARRERA = 505
        NPLAN = 21
    Case Is = "3"
        NCARRERA = 505
        NPLAN = 24
    Case Is = "4"
        NCARRERA = 509
        NPLAN = 1
    Case Is = "5"
        NCARRERA = 512
        NPLAN = 6
    Case Is = "6"
        NCARRERA = 512
        NPLAN = 7
    Case Is = "7"
        NCARRERA = 512
        NPLAN = 8
    Case Is = "8"
        NCARRERA = 514
        NPLAN = 21
```

```
Case Is = "9"
    NCARRERA = 519
    NPLAN = 21
End Select
End Sub
Sub Command1_Click (Index As Integer)
Select Case Index
Case 0:
    If CAP.Text = "" Then
        c = MsgBox("Se debe elegir la carrera", 48, "AVISO")
    Else
        If NCARRERA = dsalumno!carrera Then
            c = MsgBox("Alumno ya esta en esa carrera", 48, "Aviso")
            Exit Sub
        Else
            CAMBIA_CARRERA
            ALUMNOS.DATOS(2).Caption = CHECA_CARRERA(NCARRERA)
            If dsalumno!INSC = 1 Then
                c = MsgBox("El alumno esta inscrito en este periodo.
                    Se eliminara inscripcion", 48, "AVISO")
                ELIMINA_INSCRIPCION

                End If
                ALUMNOS.Enabled = True
                Unload Me
            End If
        End If
    Case 1:
        ALUMNOS.Enabled = True
        Unload Me
    End Select
End Sub
Sub ELIMINA_INSCRIPCION ()
    DTINSCRIP.Index = "CUENTA"
    Do
        DTINSCRIP.Seek "=", NUMCUENTA
    If DTINSCRIP.NoMatch Then
        Exit Do
    Else
        DTINSCRIP.Delete
    End If
    Loop
    dsalumno.Edit
    dsalumno!INSC = 0
    dsalumno.Update
```

## Apéndice "E"

---

```
End Sub
Sub Form_Load ()
CAP.AddItem "1: BIOLOGO"
CAP.AddItem "2: INGENIERIA QUIMICA"
CAP.AddItem "3: QUIMICO FARMACEUTICO BIOLOGO"
CAP.AddItem "4: TECNICO EN ENFERMERIA"
CAP.AddItem "5: MEDICO CIRUJANO (PLAN 06)"
CAP.AddItem "6: MEDICO CIRUJANO (PLAN 07)"
CAP.AddItem "7: MEDICO CIRUJANO (PLAN 08)"
CAP.AddItem "8: CIRUJANO DENTISTA"
CAP.AddItem "9: LIC. EN PSICOLOGIA"
End Sub

FORMA: CGRUPO.FRM
Dim BANDERA3
Dim banlisto
Sub Actualiza_pantalla ()
    sqlq_materias
    For T = 0 To 4
        text1(T).Text = ""
    Next
    GRUPOM.Text = "_____"
    grid2.Visible = False
End Sub
Sub baja_materia ()
    grupoini = text1(0).Text
    asignini = Val(text1(1).Text)
    dtinscrip.Index = "Primarykey"
    dtinscrip.Seek "=", numcuenta, carrera, grupoini, asignini
    If dtinscrip.NoMatch Then
        MsgBox ("Registro no encontrado")
    Else
        dtinscrip.Delete
        C = MsgBox("Materia dada de baja", 64, "Aviso")
        banlisto = True
    End If
Actualiza_pantalla
End Sub
Sub bcambio_Click (Index As Integer)
Dim ASIGN
Select Case Index
Case 0:
    ASIGN = text1(4).Text
    If checa_aprobada(ASIGN) Then
```

```
        If Checa_Art27(ASIGN) Then
            If materia_repetida() Then
                cambia_materia
            End If
        End If
    End If
End If
Case 1
    baja_materia
Case 2
    If banlisto Then
        alumnos.Enabled = True
        Unload Me
        Exit Sub
    Else
        C = MsgBox("No se ha Realizado Ningún Cambio. Salir?", 68,
"AVISO")
        If C = 6 Then
            alumnos.Enabled = True
            Unload Me

            Exit Sub
        End If
    End If
End Select
End Sub
Sub bcambio_GotFocus (Index As Integer)
Select Case Index
Case 0:
    MENSAJES.CAPTION = "DAR DE BAJA MATERIA"
Case 1:
    MENSAJES.CAPTION = "CAMBIAR MATERIA"
Case 2:
    MENSAJES.CAPTION = "REGRESAR A ALUMNO"
Case Else
    MENSAJES.CAPTION = ""
End Select
End Sub
Sub cambia_materia ()
grupoini = text1(0).Text
asignini = Val(text1(1).Text)
dtinscrip.Index = "Primarykey"
dtinscrip.Seek "=", numcuenta, carrera, grupoini, asignini
If dtinscrip.NoMatch Then
    MsgBox ("Registro no encontrado")
Else
```

## Apéndice "E"

---

```
dtinscrip.Edit
dtinscrip!GRUPO = GRUPOM.Text
dtinscrip!ASIGN = text1(4).Text
dtinscrip.Update
DTCAMBIO.AddNew
DTCAMBIO!CUENTA = numcuenta
DTCAMBIO!cambio = 1
DTCAMBIO!fecha = Format(Now, "dd-mm-yy")
DTCAMBIO!origen = text1(0).Text
DTCAMBIO!destino = GRUPOM.Text
DTCAMBIO!carrera = carrera
DTCAMBIO!ASIGN = text1(1).Text
DTCAMBIO.Update
Actualiza_pantalla
C = MsgBox("Cambio Realizado", 64, "Aviso")
banlisto = True
End If

End Sub
Sub CAP_LostFocus ()
End Sub
Sub creaconsulta2 ()
Dim com
Dim sqlq As String
If SQLACTIVO = 1 Then
    dsgrp.Close
    SQLACTIVO = 0
End If
temp = Str(ncarrera)
com = Chr(34)
sqlq = "select grupo.ASIGN, Assign.desc_asign, ASIGN.CREDITOS from
grupo,assign where grupo.assign=assign.assign And grupo.grupo=" &
com & GRUPOM.Text & com & " AND GRUPO.CARRERA=" & carrera
Set dsgrp = DB.CreateDynaset(sqlq)
'If Err Then
' MsgBox "ERROR DE SINTAXIS EN CONSULTA"
'Else
If dsgrp.RecordCount <> 0 Then
    grid2.Visible = True
    SQLACTIVO = 1
    LOADFRAME3
End If
'End If
End Sub
Sub Form_Load ()
```

```
BANDERA2 = False
banlisto = False
sqlq_materias
End Sub
Sub Grid1_Click ()
bcambio(1).Enabled = True
grid1.Col = 1
text1(0).Text = grid1.Text
grid1.Col = 2
text1(1).Text = grid1.Text
grid1.Col = 3
text1(2).Text = grid1.Text
GRUPOM.Visible = True
labell(6).Visible = True
GRUPOM.SetFocus
End Sub
Sub Grid1_GotFocus ()
MENSAJES.CAPTION = "SELECCIONE MATERIA A CAMBIAR O BAJA"
End Sub
Sub Grid2_Click ()
grid2.Col = 1
text1(4).Text = grid2.Text
grid2.Col = 2
text1(3).Text = grid2.Text
bcambio(0).Enabled = True
End Sub
Sub Grid2_GotFocus ()
MENSAJES.CAPTION = "SELECCIONE MATERIA A CAMBIO"
End Sub
Sub grupom_GotFocus ()
MENSAJES.CAPTION = "TECLEE GRUPO"
End Sub
Sub GRUPOM_LostFocus ()
If GRUPOM.Text <> "" Then
    dtgrupo.Index = "INDEX1"
    dtgrupo.Seek "=", carrera, GRUPOM.Text
    If dtgrupo.NoMatch Then
        C = MsgBox("GRUPO NO EXISTE", 48, "AVISO")
        GRUPOM.Mask = ""
        GRUPOM.Text = ""
        GRUPOM.Mask = "AA###"
        GRUPOM.SetFocus
    End If
Else
```

Apéndice "E"

```
    creaconsulta2
End If
End If
End Sub
Sub LOADFRAME ()

    Dim r As Integer, C As Integer, temp As Integer
    Dim NUMCAMPOS As Integer
    Dim ciclo As Integer
    Dim Frame_size As Long

    On Error Resume Next
    DSINSCRIP.MoveLast
    Frame_size = DSINSCRIP.RecordCount
    DSINSCRIP.MoveFirst

    grid1.Rows = Frame_size + 1
    grid1.ColWidth(0) = 100

    grid1.Row = 0
    NUMCAMPOS = DSINSCRIP.Fields.Count
    grid1.Cols = NUMCAMPOS + 1
    For ciclo = 0 To NUMCAMPOS
        grid1.Col = ciclo + 1
        grid1.ColWidth(ciclo + 1) =
DSINSCRIP.Fields(ciclo).FieldSize() * 150
        grid1.Text = DSINSCRIP.Fields(ciclo).Name
    Next

    For r = 1 To Frame_size
        grid1.Row = r

        For ciclo = 0 To NUMCAMPOS
            grid1.Col = ciclo + 1
            grid1.Text = DSINSCRIP.Fields(ciclo).Value
        Next

        If r <> Frame_size Then
            DSINSCRIP.MoveNext
            If DSINSCRIP.EOF Then
                Exit For
            End If
        End If
    Next r
```



```
End Sub
Sub loadframe2 ()

    Dim r As Integer, C As Integer, temp As Integer
    Dim NUMCAMPOS As Integer
    Dim ciclo As Integer
    Dim Frame_size As Long

    On Error Resume Next
    dtgrupo.MoveLast
    Frame_size = dtgrupo.RecordCount
    dtgrupo.MoveFirst

    grid1.Rows = Frame_size + 1
    grid1.ColWidth(0) = 100

    grid1.Row = 0
    NUMCAMPOS = dtgrupo.Fields.Count
    grid1.Cols = NUMCAMPOS + 1
    For ciclo = 0 To NUMCAMPOS
        grid1.Col = ciclo + 1
        grid1.ColWidth(ciclo + 1) =
            dtgrupo.Fields(ciclo).FieldSize() * 150
        grid1.Text = dtgrupo.Fields(ciclo).Name
    Next

    For r = 1 To Frame_size
        grid1.Row = r

        For ciclo = 0 To NUMCAMPOS
            grid1.Col = ciclo + 1
            grid1.Text = dtgrupo.Fields(ciclo).Value
        Next

        If r <> Frame_size Then
            dtgrupo.MoveNext
            If dtgrupo.EOF Then
                Exit For
            End If
        End If
    Next r
End Sub
Sub LOADFRAME3 ()
```

Apéndice "E"

---

```
Dim r As Integer, C As Integer, temp As Integer
Dim NUMCAMPOS As Integer
Dim ciclo As Integer
Dim Frame_size As Long

On Error Resume Next
dsgrp.MoveLast
Frame_size = dsgrp.RecordCount
dsgrp.MoveFirst

grid2.Rows = Frame_size + 1
grid2.ColWidth(0) = 100

grid2.Row = 0
NUMCAMPOS = dsgrp.Fields.Count
grid2.Cols = NUMCAMPOS + 1
For ciclo = 0 To NUMCAMPOS
    grid2.Col = ciclo + 1
    grid2.ColWidth(ciclo + 1) = dsgrp.Fields(ciclo).FieldSize()
* 150
    grid2.Text = dsgrp.Fields(ciclo).Name
Next

For r = 1 To Frame_size
    grid2.Row = r

    For ciclo = 0 To NUMCAMPOS
        grid2.Col = ciclo + 1
        grid2.Text = dsgrp.Fields(ciclo).Value
    Next

    If r <> Frame_size Then
        dsgrp.MoveNext
        If dsgrp.EOF Then
            Exit For
        End If
    End If

Next r
End Sub
Function materia_repetida ()
grid2.Col = 2
materia_repetida = True
For T = 1 To 4
grid2.Row = T
```

```
If text1(4).Text = grid2.Text Then
  C = MsgBox("Materia ya esta Seleccionada", 48, "Aviso")
  materia_repetida = False
  Exit For
End If
Next
End Function
Sub sqlq_materias ()
  Dim sqlq As String
  com = Chr(34)
  If BANDERA2 Then
    DSINSCRIP.Close
  End If
  sqlq = "SELECT INSCRIP.GRUPO, inscrip.asign, ASIGN.DESC_ASIGN
FROM INSCRIP,ASIGN WHERE INSCRIP.ASIGN=ASIGN.ASIGN AND
INSCRIP.CUENTA=" & com & numcuenta & com & " and
inscrip.carrera=" & carrera
  Set DSINSCRIP = DB.CreateDynaset(sqlq)
  BANDERA2 = True
  grid1.Visible = True
  LOADFRAME
End Sub
```

**FORMA: FESZ.FRM**

```
Global cambio As Integer
Global DB As database
Global DSALUMNO As TABLE
Global DTGRUPO As TABLE
Global DTASIGN As TABLE
Global DTPROFESOR As TABLE
Global DTTALU As TABLE
Global dtcarrera As TABLE
Global dthistoria As TABLE
Global dtinscrip As TABLE
Global DSINSCRIP As DYNASET
Global dtcambio As TABLE
Global TEMP
Global carrera As Integer
Global NUMCUENTA As String
Global vasign As String
Sub ABREBASES1 ()
  Set DB = OpenDatabase("\VBPRG\fesz\fesz.mdb")
  Set DSALUMNO = DB.OpenTable("ALUMNO")
  Set DTASIGN = DB.OpenTable("ASIGN")
```

## Apéndice "E"

---

```
Set DTGRUPO = DB.OpenTable("GRUPO")
Set DTPROFESOR = DB.OpenTable("PROFESOR")
Set DSPRO = DB.CreateDynaset("PROFESOR")
Set dtcarrera = DB.OpenTable("carrera")
Set dthistoria = DB.OpenTable("historia")
Set dtinscrip = DB.OpenTable("inscrip")
Set dtcambio = DB.OpenTable("Cambio")
Set DTTALU = DB.OpenTable("TALU")
End Sub
Function checa_aprobada (asignatura)
dthistoria.Index = "primarykey"
dthistoria.Seek "=", NUMCUENTA, carrera, asignatura
If Not (dthistoria.NoMatch) Then
    If (dthistoria.calif = "MB") Or (dthistoria.calif = " B")
Or (dthistoria.calif = " S") Then
        c = MsgBox("Materia Aprobada", 48, "AVISO")
        checa_aprobada = False
    Else
        checa_aprobada = True
    End If
Else
    checa_aprobada = True
End If
End Function
Function Checa_Art27 (asignatura) As Integer
dthistoria.Index = "primarykey"
dthistoria.Seek "=", NUMCUENTA, carrera, asignatura
If Not (dthistoria.NoMatch) Then
    If dthistoria.num_ord = 2 Then
        c = MsgBox("Alumno en Art. 27", 48, "AVISO")
        Checa_Art27 = False
    Else
        Checa_Art27 = True
    End If
Else
    Checa_Art27 = True
End If
End Function
Function checa_carrera (dato As Integer) As String
Select Case dato
Case 503
    checa_carrera = "BIOLOGO"
```

```
Case 505
  If DSALUMNO!PLANTEL = 21 Then
    checa_carrera = "INGENIERIA QUIMICA"
  Else
    checa_carrera = "QUIMICO FARMACEUTICO BIOLOGO"
  End If
Case 509
  checa_carrera = "TECNICO EN ENFERMERIA"
Case 512
  checa_carrera = "MEDICO CIRUJANO"
Case 514
  checa_carrera = "CIRUJANO DENTISTA"
Case 519
  checa_carrera = "LIC. EN PSICOLOGIA"
End Select
End Function
```

**FORMA: GRUPOS.FRM**

```
Dim ncarrera As Integer
Dim NPLAN As Integer
Dim CADENA As String
Dim BANVALIDA(5) As Integer
Dim TEMP As String
Dim MENSAJE As String
Dim BAN As Integer
Sub ASIGNATURA_LostFocus ()

End Sub
Sub AYUDA_Click (Index As Integer)
  If Index = 1 Then BAN = 1
  PANEL3D2.Visible = True
  loadframe2
End Sub
Sub BORRA_PANTALLA ()
  DATOSG(0).Text = ""
  For T = 1 To 5
    DATOSG(T).Text = " "
  Next
  LABEL2.Caption = ""
  LABEL3.Caption = ""
  LABEL4.Caption = ""
  LABEL1(7).Caption = ""
  'verifica(0).Enabled = False
  'verifica(0).FontItalic = True
```

Apéndice "E"

---

```
'verifica(1).Enabled = False
'verifica(1).FontItalic = True
GRID1.Visible = False
CAP.Clear
End Sub
Sub CAP_LostFocus ()
CADENA = Left(CAP.List(CAP.ListIndex), 1)
Select Case CADENA
Case Is = "1"
    ncarrera = 503
    NPLAN = 22
Case Is = "2"
    ncarrera = 505
    NPLAN = 21
Case Is = "3"
    ncarrera = 505
    NPLAN = 24
Case Is = "4"
    ncarrera = 509
    NPLAN = 1
Case Is = "5"
    ncarrera = 512
    NPLAN = 6
Case Is = "6"
    ncarrera = 512
    NPLAN = 7
Case Is = "7"
    ncarrera = 512
    NPLAN = 8
Case Is = "8"
    ncarrera = 514
    NPLAN = 21
Case Is = "9"
    ncarrera = 519
    NPLAN = 21
End Select
If CAP.Text = "" Then
    BANVALIDA(0) = False
Else
    BANVALIDA(0) = True
End If
' 503 BIOLOGO                9  22    441    0|
' 505 INGENIERIA QUIMICA     9  21    384    0|
' 505 QUIMICO FARMACEUTICO BIOLOGO 9  24    384    0|
' 509 TECNICO EN ENFERMERIA  6  01    281    0|
```

Código Fuente

```

' 512 MEDICO CIRUJANO          0  06      0      0|
' 512 MEDICO CIRUJANO          8  07     446     0|
' 512 MEDICO CIRUJANO          9  08     446     0|
' 514 CIRUJANO DENTISTA        8  21     369     0|
' 519 LIC. EN PSICOLOGIA       9  21     320     0|
End Sub
Sub creaconsulta ()
Dim COM
Dim sqlq As String
If SQLACTIVO = 1 Then
    DSGRP.Close
    'GRID1.Visible = False
    SQLACTIVO = 0
End If
TEMP = Str(ncarrera)
COM = Chr(34)
sqlq = "select grupo.ASIGN, Asign.desc_asign, profesor.profesor,
grupo.cupo from grupo,asign,profesor where
grupo.asign=asign.asign and grupo.rfc=profesor.rfc And
grupo.grupo=" & COM & DATOSG(1).Text & COM & " and
grupo.CARRERA=" & TEMP
Set DSGRP = DB.CreateDynaset(sqlq)
' If Err Then
' MsgBox "ERROR DE SINTAXIS EN CONSULTA"
' Else
    If DSGRP.RecordCount <> 0 Then
        LABEL1(7).Caption = "GRUPO: " + DATOSG(1).Text
        GRID1.Visible = True
        SQLACTIVO = 1
        LOADFRAME
    End If

'End If
End Sub
Sub DATOSG_LostFocus (Index As Integer)
Select Case Index
Case 0:
    DTASIGN.Index = "PRIMARYKEY"
    DTASIGN.Seek "=", DATOSG(0).Text
    If DTASIGN.NoMatch Or DATOSG(0).Text = "" Then
        BANVALIDA(3) = False
    Else
        BANVALIDA(3) = True
        LABEL2.Caption = DTASIGN.DESC_ASIGN
        DTGRUPO.Index = "PRIMARYKEY"
    End If

```

Apéndice "E"

---

```
DTGRUPO.Seek "=", ncarrera, DATOSG(1).Text,
DATOSG(0).Text
If Not (DTGRUPO.NoMatch) Then
    LABEL2.Caption = ""
    BANVALIDA(2) = False
Else
    LABEL2.Caption = DTASIGN!DESC_ASSIGN
    BANVALIDA(2) = True
End If
End If
Case 1

If DATOSG(1).Text = "" Then
    BANVALIDA(1) = False
Else
    BANVALIDA(1) = True
    creaconsulta
End If
Case 2
If DATOSG(2).Text <> "" Then
    DTPROFESOR.Index = "PRIMARYKEY"
    DTPROFESOR.Seek "=", DATOSG(2).Text
    If DTPROFESOR.NoMatch Then
        BANVALIDA(4) = False

    Else
        BANVALIDA(4) = True
        LABEL3.Caption = DTPROFESOR!profesor
    End If
Else
    BANVALIDA(4) = False
End If
Case 3
If DATOSG(3).Text <> "" Then
    DTPROFESOR.Index = "PRIMARYKEY"
    DTPROFESOR.Seek "=", DATOSG(3).Text
    If DTPROFESOR.NoMatch Then
        BANVALIDA(5) = False
        DATOSG(3).Text = ""
    Else
        BANVALIDA(5) = True
        LABEL4.Caption = DTPROFESOR!profesor
    End If
End If
End Select
```



```
End Sub
Sub Form_Activate ()
SQLACTIVO = 0
BORRA_PANTALLA
CAP.AddItem "1: BIOLOGO"
CAP.AddItem "2: INGENIERIA QUIMICA"
CAP.AddItem "3: QUIMICO FARMACEUTICO BIOLOGO"
CAP.AddItem "4: TECNICO EN ENFERMERIA"
CAP.AddItem "5: MEDICO CIRUJANO (PLAN 06)"
CAP.AddItem "6: MEDICO CIRUJANO (PLAN 07)"
CAP.AddItem "7: MEDICO CIRUJANO (PLAN 08)"
CAP.AddItem "8: CIRUJANO DENTISTA"
CAP.AddItem "9: LIC. EN PSICOLOGIA"
For T = 0 To 5
  BANVALIDA(T) = True
Next
BAN = 0
End Sub
Sub GRABA_DATOS ()
On Error Resume Next
DTGRUPO.AddNew
DTGRUPO!GRUPO = DATOSG(1).Text
DTGRUPO!ASIGN = Str(DATOSG(0).Text)
DTGRUPO!RFC = DATOSG(2).Text
DTGRUPO!CUPO = Str(DATOSG(4).Text)
DTGRUPO!RFC_ASIS = DATOSG(3).Text
DTGRUPO!carrera = ncarrera
DTGRUPO!PLAN = NPLAN
DTGRUPO!CUPOI = Str(DATOSG(5).Text)
DTGRUPO.Update
End Sub
Sub Grid2_DblClick ()
GRID2.Col = 1
If BAN = 1 Then
  DATOSG(3).Text = GRID2.Text
  DATOSG(3).SetFocus
Else
  DATOSG(2).Text = GRID2.Text
  DATOSG(2).SetFocus
End If
PANEL3D2.Visible = False
End Sub
Sub Grid2_LostFocus ()
PANEL3D2.Visible = False
End Sub
```

Apéndice "E"

---

```
Sub inicia_tabla ()
    GRID1.Cols = 5
    GRID1.Row = 0
    GRID1.Col = 1
    GRID1.ColWidth(1) = 600
    GRID1.Text = "CLAVE"
    GRID1.Col = 2
    GRID1.ColWidth(2) = 3000
    GRID1.Text = "ASIGNATURA"
    GRID1.Col = 3
    GRID1.ColWidth(3) = 2500
    GRID1.Text = "PROFESOR"
    GRID1.Col = 4
    GRID1.ColWidth(4) = 500
    GRID1.Text = "CUPO"
End Sub
Sub INIPANTALLA ()

    DATOSG(0).Text = ""
    For T = 2 To 5
        DATOSG(T).Text = ""
    Next
    LABEL2.Caption = ""
    LABEL3.Caption = ""
    LABEL4.Caption = ""
    LABEL1(7).CAPTION = ""
    'verifica(0).Enabled = False
    'verifica(0).FontItalic = False
    'verifica(1).Enabled = False
    'verifica(1).FontItalic = False
    DATOSG(0).SetFocus
End Sub
Sub LOADFRAME ()

    Dim r As Integer, C As Integer, TEMP As Integer
    Dim NUMCAMPOS As Integer
    Dim ciclo As Integer
    Dim Frame_size As Long
    inicia_tabla
    On Error Resume Next
    DSGRP.MoveLast
    Frame_size = DSGRP.RecordCount
    DSGRP.MoveFirst

    GRID1.Rows = Frame_size + 1
```

```
GRID1.ColWidth(0) = 100

GRID1.Row = 0
NUMCAMPOS = DSGRP.Fields.Count
GRID1.Cols = NUMCAMPOS + 1
'For ciclo = 0 To NUMCAMPOS
' GRID1.COL = ciclo + 1
' GRID1.ColWidth(ciclo + 1) =
  DSGRP.Fields(ciclo).FieldSize() * 100
'GRID1.TEXT = DSGRP.Fields(ciclo).Name
'Next

For r = 1 To Frame_size
  GRID1.Row = r

  For ciclo = 0 To NUMCAMPOS
    GRID1.Col = ciclo + 1
    GRID1.Text = DSGRP.Fields(ciclo).Value
  Next

  If r <> Frame_size Then
    DSGRP.MoveNext
    If DSGRP.EOF Then
      Exit For
    End If
  End If

Next r
End Sub
Sub loadframe2 ()

Dim r As Integer, C As Integer, TEMP As Integer
Dim NUMCAMPOS As Integer
Dim ciclo As Integer
Dim Frame_size As Long

On Error Resume Next
DTPROFESOR.MoveLast
Frame_size = DTPROFESOR.RecordCount
DTPROFESOR.MoveFirst

GRID2.Rows = Frame_size + 1
GRID2.ColWidth(0) = 120

GRID2.Row = 0
```

## Apéndice "E"

---

```
NUMCAMPOS = DTPROFESOR.Fields.Count
GRID2.Cols = NUMCAMPOS + 1
For ciclo = 0 To NUMCAMPOS
    GRID2.Col = ciclo + 1
    GRID2.ColWidth(ciclo + 1) =
DTPROFESOR.Fields(ciclo).FieldSize() * 80
    GRID2.Text = DTPROFESOR.Fields(ciclo).Name
Next

For r = 1 To Frame_size
    GRID2.Row = r

    For ciclo = 0 To NUMCAMPOS
        GRID2.Col = ciclo + 1
        GRID2.Text = DTPROFESOR.Fields(ciclo).Value
    Next

    If r <> Frame_size Then
        DTPROFESOR.MoveNext
        If DTPROFESOR.EOF Then
            Exit For
        End If
    End If

Next r
End Sub
Function VALIDAPANTALLA () As Integer
VALIDAPANTALLA = False
For T = 0 To 5
    If BANVALIDA(T) = False Then
        Exit For
    End If
Next
Select Case T
Case 0
    MENSAJE = "SE REQUIERE SELECCIONAR CARRERA"
    C = MsgBox(MENSAJE, 48, "AVISO")
    CAP.SetFocus

Case 1
    MENSAJE = "SE REQUIERE GRUPO"
    C = MsgBox(MENSAJE, 48, "AVISO")
    DATOSG(1).SetFocus

Case 2
    MENSAJE = "GRUPO-MATERIA YA EXISTE!"
```

```
        C = MsgBox(MENSAJE, 48, "AVISO")
        DATOSG(0).SetFocus
Case 3
    MENSAJE = "MATERIA NO EXISTE!"
    C = MsgBox(MENSAJE, 48, "AVISO")
    DATOSG(0).SetFocus
Case 4
    MENSAJE = "PROFESOR NO EXISTE!"
    C = MsgBox(MENSAJE, 48, "AVISO")
    DATOSG(2).SetFocus
Case 5
    MENSAJE = "PROFESOR SUPLENTE NO EXISTE!"
    C = MsgBox(MENSAJE, 48, "AVISO")
    DATOSG(3).SetFocus

Case Else
    VALIDAPANTALLA = True
    Exit Function
End Select
End Function
Sub VERIFICA_Click (Index As Integer)
Select Case Index
    Case 0:
        If VALIDAPANTALLA() Then
            GRABA_DATOS
            creaconsulta
            INIPANTALLA

        End If
    Case 1:
        INIPANTALLA
        inicia_tabla
    Case 4:
        GRUPOS.Hide
        INICIO.Show

End Select
End Sub
```

```
FORMA: INICIO.FRM
Sub BOTONESP_Click (Index As Integer)
Select Case Index
Case 0
    alumnos.Show
```

Apéndice "E"

---

```
INICIO.Hide
Case 1
INICIO.Hide
GRUPOS.Show
Case 2
r = MsgBox(" Salir Sistema ?", 33, "AVISO")
If r = 1 Then
```

```
    Unload GRUPOS
    Unload alumnos
    Unload Me
End If
Case 3
Load PRIINGRESO
PRIINGRESO.Show
INICIO.Hide
Case 6:
    Load PROFESORES
    PROFESORES.Show
    INICIO.Hide
```

```
End Select
End Sub
Sub Form_Load ()
    Load alumnos
    Load GRUPOS
    ABREBASES1
End Sub
```

**FORMA: MATERIAS.FRM**

```
Sub Form_Load ()
    sqlq_materias
    PANEL3D1.Width = GRID1.Width
End Sub
Sub Grid1_LostFocus ()
    Unload Me
End Sub
Sub LOADFRAME ()
```

```
    Dim r As Integer, c As Integer, temp As Integer
    Dim NUMCAMPOS As Integer
    Dim ciclo As Integer
    Dim Frame_size As Long
```

```
    On Error Resume Next
```

```
DSINSCRIP.MoveLast
Frame_size = DSINSCRIP.RecordCount
DSINSCRIP.MoveFirst

GRID1.Rows = Frame_size + 1
GRID1.ColWidth(0) = 100

GRID1.Row = 0
NUMCAMPOS = DSINSCRIP.Fields.Count
GRID1.Cols = NUMCAMPOS + 1
For ciclo = 0 To NUMCAMPOS
  GRID1.Col = ciclo + 1
  GRID1.ColWidth(ciclo + 1) =
DSINSCRIP.Fields(ciclo).FieldSize() * 150
  GRID1.Text = DSINSCRIP.Fields(ciclo).Name
Next

For r = 1 To Frame_size
  GRID1.Row = r

  For ciclo = 0 To NUMCAMPOS
    GRID1.Col = ciclo + 1
    GRID1.Text = DSINSCRIP.Fields(ciclo).Value
  Next

  If r <> Frame_size Then
    DSINSCRIP.MoveNext
    If DSINSCRIP.EOF Then
      Exit For
    End If
  End If

Next r
End Sub
Sub sqlq_materias ()
  Dim sqlq As String
  com = Chr(34)
  sqlq = "SELECT INSCRIP.GRUPO, inscrip.ASIGN, ASIGN.DESC_ASIGN
FROM INSCRIP,ASIGN WHERE INSCRIP.ASIGN=ASIGN.ASIGN AND
INSCRIP.CUENTA=" & com & numcuenta & com & " and
inscrip.carrera=" & carrera
  Set DSINSCRIP = DB.CreateDynaset(sqlq)
  GRID1.Visible = True
  LOADFRAME
End Sub
```

## Apéndice "E"

---

```
FORMA: PRIINGRE.FRM
Sub ACTUALIZA_VALORES ()
On Error Resume Next
NUMCUENTA = dtalu|CUENTA
datos(0).Caption = Format(dtalu|CUENTA, "#####-#")
datos(1).Caption = dtalu|ALUMNO
carrera = dtalu|carrera
datos(2).Caption = checa_carrera(carrera)
datos(3).Caption = dtalu|INGRESO
If dtalu|NACION = 1 Then
    datos(4).Caption = "MEXICANA"
Else
    datos(4).Caption = "EXTRANJERO"
End If
datos(5).Caption = dtalu|sexo
datos(6).Caption = Format(dtalu|nacim, "dd/mm/yy")
datos(7).Caption = dtalu|direccion
datos(8).Caption = dtalu|cp
datos(9).Caption = dtalu|tel
datos(10).Caption = dtalu|esc_proc
datos(11).Caption = dtalu|promedio
GRUPO.Caption = dtalu|GRUPO
End Sub
Sub ALTA ()
dsalumno.Index = "PRIMARYKEY"
dsalumno.Seek "=", CAP(0).ClipText
If dsalumno.NoMatch Then
    dsalumno.AddNew
    dsalumno|CUENTA = dtalu|CUENTA
    dsalumno|ALUMNO = dtalu|ALUMNO
    dsalumno|carrera = dtalu|carrera
    dsalumno|PLANTEL = dtalu|PLAN
    dsalumno|INGRESO = dtalu|INGRESO
    dsalumno|NACION = dtalu|NACION
    dsalumno|sexo = dtalu|sexo
    dsalumno|nacim = dtalu|nacim
    dsalumno|direccion = dtalu|direccion
    dsalumno|cp = dtalu|cp
    dsalumno|tel = dtalu|tel
    dsalumno|esc_proc = dtalu|esc_proc
    dsalumno|promedio = dtalu|promedio
    dsalumno|INSC = 1
    dsalumno.Update
End Sub
```



```
dsalumno.MoveLast
' INSCRIPCION
DTGRUPO.Index = "INDEX1"
DTGRUPO.Seek "=", dttalulcarrera, dttalu!GRUPO
COM = Chr(34)
If DTGRUPO.NoMatch Then
    MsgBox ("NO EXISTE GRUPO")
Else
    SQLQ = "SELECT GRUPO,ASIGN,RFC,CARRERA,PLAN FROM GRUPO WHERE
GRUPO=" & COM & dttalu!GRUPO & COM & " AND CARRERA=" &
dttalulcarrera
    Set DSGRP = DB.CreateDynaset(SQLQ)
    Do Until DSGRP.EOF
        DTINSCRIP.AddNew
        DTINSCRIP!CUENTA = dttalu!CUENTA
        DTINSCRIP!GRUPO = dttalu!GRUPO
        DTINSCRIP!ASIGN = DSGRP!ASIGN
        DTINSCRIP!TIPO = 1
        DTINSCRIP!carrera = dttalulcarrera
        DTINSCRIP.Update
        DSGRP.MoveNext
    Loop
    dttalu.Edit
    dttalu!INSC = 1
    dttalu.Update
    C = MsgBox("ALUMNO INSCRITO EN GRUPO" + GRUPO.Caption, 48,
"AVISO")
    End If
Else
    MsgBox ("ALUMNO YA EXISTE")
End If
End Sub
Sub BORRA_VALORES ()
For t = 0 To 11
    datos(t).Caption = ""
Next
End Sub
Sub BOTONES_Click (Index As Integer)
Select Case Index
Case 0:
    CAP(0).Visible = True
    CAP(0).SetFocus
Case 1:
    For t = 1 To 7
        CAP(t).Visible = True
```

## Apéndice "E"

---

```
Next
PROCEDENCIA.Visible = True
PASA_VALORES
Call ESTADO_BOTONES(False, 0, 3)
VERIFICA(0).Visible = True
VERIFICA(1).Visible = True
CAP(1).SetFocus
Case 2:
If dtalulINSC = 1 Then
C = MsgBox("ALUMNO YA FUE INSCRITO", 48, "AVISO")
Else
ALTA
End If
Case 3:
Unload Me
INICIO.Show
End Select
End Sub
Sub BOTONES_GotFocus (Index As Integer)
Select Case Index
Case 0:
MENSAJES.Caption = "SELECCIONA NUMERO DE CUENTA"
Case 1:
MENSAJES.Caption = "MODIFICAR DATOS DEL ALUMNO"
Case 2:
MENSAJES.Caption = "INSCRIBIR AL ALUMNO"
Case 3:
MENSAJES.Caption = "REGRESA A MENU PRINCIPAL"
Case Else
MENSAJES.Caption = ""
End Select
End Sub
Sub CAP_GotFocus (Index As Integer)
Select Case Index
Case 0:
MENSAJES.Caption = "TECLEE NUMERO DE CUENTA"
Case 1:
MENSAJES.Caption = "MODIFICAR NOMBRE"
Case 2:
MENSAJES.Caption = "MODIFICAR DIRECCION"
Case 3:
MENSAJES.Caption = ""
Case 4:
MENSAJES.Caption = "TECLE SOLO NUMEROS"
Case Else
```

```
MENSAJES.Caption = ""
End Select
End Sub
Sub CAP_LostFocus (Index As Integer)
Select Case Index
Case 0:
    If CAP(0).ClipText <> "" Then
        dtalu.Index = "PRIMARYKEY"
        dtalu.Seek "=", CAP(0).ClipText
        If dtalu.NoMatch Then
            CAP(0).Mask = ""
            CAP(0).Text = ""
            CAP(0).Mask = "#####-#"
            MsgBox ("NO EXISTE ALUMNO")
        Else
            CAP(0).Visible = False
            Call ESTADO_BOTONES(True, 1, 2)
            BORRA_VALORES
            ACTUALIZA_VALORES
            BOTONES(1).Enabled = True
            BOTONES(2).Enabled = True
        End If
    End If
End Select
End Sub
Sub cuenta_LostFocus ()
End Sub
Sub ESTADO_BOTONES (EST, I, F As Integer)
For t = I To F
    BOTONES(t).Visible = EST
Next
End Sub
Sub fija_mascaras ()
CAP(2).Mask = "?"
CAP(3).Mask = "##/##/##"
CAP(5).Mask = "#####"
CAP(6).Mask = "#####"
End Sub
Sub Form_Activate ()
PROCEDENCIA.AddItem "C.C.H."
PROCEDENCIA.AddItem "E.N.P"
PROCEDENCIA.AddItem "INC. UNAM"
PROCEDENCIA.AddItem "INC. SEP"
PROCEDENCIA.AddItem "NO INCOR."
PROCEDENCIA.AddItem "OTRA"
```

Apéndice "E"

```
Call ESTADO_BOTONES(False, 1, 2)
End Sub
Sub GUARDA_CAMBIOS ()
dttalu.Edit
dttalu!ALUMNO = CAP(1).Text
dttalu!sexo = CAP(2).Text
dttalu!nacim = CAP(3).Text
dttalu!direccion = CAP(4).Text
dttalu!lcp = Val(CAP(5).Text)
dttalu!tel = CAP(6).Text
dttalu!esc_proc = PROCEDENCIA.Text
dttalu!promedio = Format(CAP(7).Text, "00.00")
dttalu.Update
End Sub
Sub PASA_VALORES ()
For t = 1 To 7
  CAP(t).Mask = ""
  CAP(t).Text = ""
Next
fija_mascaras
CAP(1).SelText = datos(1).Caption
CAP(2).SelText = datos(5).Caption
CAP(3).SelText = datos(6).Caption
CAP(4).SelText = datos(7).Caption
CAP(5).SelText = datos(8).Caption
CAP(6).SelText = datos(9).Caption
CAP(7).SelText = datos(11).Caption
PROCEDENCIA.Text = datos(10).Caption
End Sub
Sub PROCEDENCIA_GotFocus ()
MENSAJES.Caption = "SELECCIONE DE LISTA"
End Sub
Sub VERIFICA_Click (Index As Integer)
Select Case Index
Case 0:
  VERIFICA(0).Visible = False
  VERIFICA(1).Visible = False
  GUARDA_CAMBIOS
  Call ESTADO_BOTONES(True, 0, 3)
  ACTUALIZA_VALORES
  For t = 1 To 7
    CAP(t).Visible = False
  Next
  PROCEDENCIA.Visible = False
Case 1:
```

```
VERIFICA(0).Visible = False
VERIFICA(1).Visible = False
Call ESTADO_BOTONES(True, 0, 3)
For t = 1 To 7
  CAP(t).Visible = False
Next
PROCEDENCIA.Visible = False
End Select
End Sub
```

**FORMA: PROFESOR.FRM**

```
Dim cam As Integer
Sub actualiza ()
  DATOSM(0).CAPTION = DTPROFESOR|RFC
  DATOSM(1).CAPTION = DTPROFESOR!profesor
End Sub
Sub ALTA ()
  For T = 0 To 2
    BOTONES(T).Visible = False
  Next
  DATOSM(1).CAPTION = ""
  VERIFICA(0).Visible = True
  VERIFICA(1).Visible = True
  DATOSM(0).CAPTION = CAPRFC.ClipText
  CAPRFC.Mask = ""
  CAPRFC.Text = ""
  CAPRFC.Visible = False
  CAP.Visible = True
  CAP.SetFocus
End Sub
Sub BOTONES_Click (INDEX As Integer)
  Select Case INDEX
  Case 0
    cam = 1
    For T = 0 To 2
      BOTONES(T).Visible = False
    Next
    VERIFICA(0).Visible = True
    VERIFICA(1).Visible = True
    CAP.Text = DATOSM(1).CAPTION
    CAP.Visible = True
    CAP.SetFocus
  Case 1
    C = MsgBox("ESTA SEGURO?", 36, "AVISO")
```

## Apéndice "E"

---

```
    If C = 6 Then
        DTPROFESOR.Delete
        inicia
    End If
Case 2
    Unload Me
    inicio.Show

End Select
End Sub
Sub BOTONES_GotFocus (INDEX As Integer)
Select Case INDEX
Case 0:
    MENSAJES.CAPTION = "MODIFICAR DATOS PROFESOR"
Case 1:
    MENSAJES.CAPTION = "ELIMINAR PROFESOR"
Case 2:
    MENSAJES.CAPTION = "REGRESAR A MENU PRINCIPAL"
Case Else
    MENSAJES.CAPTION = ""
End Select
End Sub
Sub CAP_GotFocus ()
MENSAJES.CAPTION = "TECLEE NOMBRE DEL PROFESOR"
End Sub
Sub CAPRFC_GotFocus ()
MENSAJES.CAPTION = "TECLEE R. F. C"
End Sub
Sub CAPRFC_LOSTFocus ()
If CAPRFC.Text <> "" Then
    DTPROFESOR.INDEX = "PRIMARYKEY"
    DTPROFESOR.Seek "=", CAPRFC.Text
    If DTPROFESOR.NoMatch Then

        C = MsgBox("PROFESOR NO EXISTE. DESEA DAR DE ALTA?", 36,
"AVISO")
        If C = 6 Then
            ALTA
        Else
            CAPRFC.Mask = ""
            CAPRFC.Text = ""
            CAPRFC.Mask = "????-#####-AAA"
            CAPRFC.SetFocus
        End Sub
    End If
End If
```

```
Else
    CAPRFC.Visible = False
    BOTONES(0).Visible = True
    BOTONES(1).Visible = True
    On Error Resume Next
    DATOSM(0).CAPTION = DTPROFESOR!RFC
    DATOSM(1).CAPTION = DTPROFESOR!profesor
End If
End If
End Sub
Sub Form_Activate ()
    inicia
    cam = 0
End Sub
Sub inicia ()
    BOTONES(0).Visible = False
    BOTONES(1).Visible = False
    BOTONES(2).Visible = True
    DATOSM(0).CAPTION = ""
    DATOSM(1).CAPTION = ""
    VERIFICA(0).Visible = False
    VERIFICA(1).Visible = False
    CAP.Text = ""
    CAP.Visible = False
    CAPRFC.Visible = True
    CAPRFC.Mask = ""
    CAPRFC.Text = ""
    CAPRFC.Mask = "????-#####-AAA"
    CAPRFC.SetFocus
End Sub
Sub VERIFICA_Click (INDEX As Integer)
Select Case INDEX
Case 0:
    If cam = 1 Then
        DTPROFESOR.Edit
        DTPROFESOR!profesor = CAP.Text
        DTPROFESOR.Update
        inicia
        cam = 0
    Else
        DTPROFESOR.AddNew
        DTPROFESOR!RFC = DATOSM(0).CAPTION
        DTPROFESOR!profesor = CAP.Text
        DTPROFESOR.Update
        inicia
    End If
End Sub
```

**Apéndice "E"**

---

```
End If
Case 1:
  inicia
End Select
End Sub
```

**FORMA: REINSCRIP.FRM**

```
Dim VDESCRIP
Dim vcreditos
Dim FILA As Integer
Dim tcreditos As Integer 'total de creditos acumulados
Dim banderareg As Integer
Sub BOTONES_Click (Index As Integer)
Select Case Index
Case 1:
  REGISTRA_INSC
  c = MsgBox("Alumno inscrito", 64, "Aviso")
  Alumnos.Enabled = True
  Unload Me

  Exit Sub
Case 2:
  If banderareg Then
    banderareg = False
    Alumnos.Enabled = True
    Unload Me

  Else
    c = MsgBox("No se ha efectuado inscripción. Desea
continuar?", 36, "Aviso")
    If c = 6 Then
      Alumnos.Enabled = True
      Unload Me

    End If
  End If
End Select
End Sub
Sub bselecciona_Click ()
GRID1.Col = 1
vasign = GRID1.Text
GRID1.Col = 2
VDESCRIP = GRID1.Text
GRID1.Col = 3
```



```
vcreditos = GRID1.Text
If checa_aprobada(vassign) Then
  If checa_art27(vassign) Then
    If materia_repetida() Then
      If checa_creditos() Then
        verifica_cupo
      End If
    End If
  End If
End If

End Sub
Sub creaconsulta ()
Dim com
Dim sqlq As String
If SQLACTIVO = 1 Then
  DSGRP.Close
  SQLACTIVO = 0
End If
temp = Str(ncarrera)
com = Chr(34)
sqlq = "select grupo.ASIGN, Aassign.desc_assign, ASIGN.CREDITOS from
grupo,assign where grupo.assign=assign.assign And grupo.grupo=" &
com & GRUPO.Text & com & " AND GRUPO.CARRERA=" & carrera
Set DSGRP = DB.CreateDynaset(sqlq)
'If Err Then
' MsgBox "ERROR DE SINTAXIS EN CONSULTA"
'Else
  If DSGRP.RecordCount <> 0 Then
    GRID1.Visible = True
    SQLACTIVO = 1
    LOADFRAME
  End If
'End If
End Sub
Function checa_creditos () As Integer
Dim checa As Integer
checa = tcreditos + vcreditos
If checa >= 60 Then
  checa_creditos = False
  c = MsgBox("Excede Créditos", 48, "Aviso")
Else
  checa_creditos = True
End If
```

## Apéndice "E"

---

```
End Function
Sub Form_Activate ()
Dim c As String
Dim vperiodo As String
sqlq = 0
año = Val(Format(Now, "yy"))
mes = Val(Format(Now, "mm"))
If (mes >= 8) And (mes <= 12) Then
    vperiodo = Str((año + 1))
Else
    vperiodo = Str(año)
End If
If ((mes >= 8) And (mes <= 12)) Or (mes = 1) Then
    vperiodo = vperiodo + "1"
Else
    vperiodo = vperiodo + "2"
End If
inicia_tabla2
inicia_tabla
FILA = 1
tcreditos = 0
End Sub
Sub GRUPO_LostFocus ()
    dtgrupo.Index = "INDEX1"
    dtgrupo.Seek "=", carrera, GRUPO.Text
    If dtgrupo.NoMatch Then
        mensajes1.Caption = "GRUPO NO EXISTE"

    Else
        creaconsulta
    End If
End Sub
Sub inicia_tabla ()
    GRID1.Cols = 4
    GRID1.Row = 0
    GRID1.ColWidth(0) = 100
    GRID1.ColWidth(1) = 700
    GRID1.Text = "CLAVE"
    GRID1.Col = 2
    GRID1.ColWidth(2) = 3000
    GRID1.Text = "ASIGNATURA"
    GRID1.Col = 3
    GRID1.ColWidth(3) = 700
    GRID1.Text = "CRED."
End Sub
```

```
Sub inicia_tabla2 ()
  GRID2.Cols = 5
  GRID2.Rows = 5
  GRID2.Row = 0
  GRID2.ColWidth(0) = 100
  GRID2.Col = 1
  GRID2.ColWidth(1) = 900
  GRID2.Text = "Grupo"
  GRID2.Col = 2
  GRID2.ColWidth(2) = 700
  GRID2.Text = "CLAVE"
  GRID2.Col = 3
  GRID2.ColWidth(3) = 3000
  GRID2.Text = "ASIGNATURA"
  GRID2.Col = 4
  GRID2.ColWidth(4) = 700
  GRID2.Text = "CRED."
End Sub
Sub LOADFRAME ()

  Dim r As Integer, c As Integer, temp As Integer
  Dim NUMCAMPOS As Integer
  Dim ciclo As Integer
  Dim Frame_size As Long

  On Error Resume Next
  DSGRP.MoveLast
  Frame_size = DSGRP.RecordCount
  DSGRP.MoveFirst

  GRID1.Rows = Frame_size + 1
  GRID1.ColWidth(0) = 100

  GRID1.Row = 0
  NUMCAMPOS = DSGRP.Fields.Count
  GRID1.Cols = NUMCAMPOS + 1
  'For ciclo = 0 To NUMCAMPOS
  'GRID1.Col = ciclo + 1
  'GRID1.ColWidth(ciclo + 1) = DSGRP.Fields(ciclo).FieldSize()
* 500
  'GRID1.Text = DSGRP.Fields(ciclo).Name
  'Next

  For r = 1 To Frame_size
```

## Apéndice "E"

---

```
GRID1.Row = r

For ciclo = 0 To NUMCAMPOS
  GRID1.Col = ciclo + 1
  GRID1.Text = DSGRP.Fields(ciclo).Value
Next

If r <> Frame_size Then
  DSGRP.MoveNext
  If DSGRP.EOF Then
    Exit For
  End If
End If

Next r

End Sub

Function materia_repetida ()
  GRID2.Col = 2
  materia_repetida = True
  For t = 1 To 4
    GRID2.Row = t
    If vasign = GRID2.Text Then
      c = MsgBox("Materia ya esta Seleccionada", 48, "Aviso")
      materia_repetida = False
    Exit For
  End If
Next
End Function

Sub opcion_LostFocus (Index As Integer)
  If Index = 1 Then
    opcion(0).Enabled = False
    opcion(1).Enabled = False
    opcion(0).FontItalic = True
    opcion(1).FontItalic = True
  End If
End Sub

Sub pasa_materia ()
  GRID2.Row = FILA
  GRID2.Col = 1
  GRID2.Text = GRUPO.Text
  GRID2.Col = 2
  GRID2.Text = vasign
  GRID2.Col = 3
  GRID2.Text = VDESCRIP
  GRID2.Col = 4
```

```
GRID2.Text = vcreditos
FILA = FILA + 1
tcreditos = tcreditos + Val(vcreditos)
text1.Text = tcreditos
botones(1).Enabled = True
End Sub
Sub REGISTRA_INSC ()

For t = 1 To FILA - 1
    DTINSCRIP.AddNew
    GRID2.Row = t
    DTINSCRIP!CUENTA = NUMCUENTA
    GRID2.Col = 1
    DTINSCRIP!GRUPO = GRID2.Text
    GRID2.Col = 2
    DTINSCRIP!ASIGN = GRID2.Text
    DTINSCRIP!TIPO = 1
    DTINSCRIP!carrera = carrera
    DTINSCRIP.Update
Next
dsalumno.Edit
dsalumno!INSC = 1
banderareg = True
dsalumno.Update
End Sub
Sub verifica_cupo ()
    dtgrupo.Index = "primarykey"
    dtgrupo.Seek "=", carrera, GRUPO.Text, vassign
    If dtgrupo.NoMatch Then
        MsgBox ("no existe")
    End If

    If opcion(0).Value Then
        If dtgrupo!CUPO = 0 Then
            c = MsgBox(" Grupo saturado ", 48, "Aviso")
        Else
            dtgrupo.Edit
            dtgrupo!CUPO = dtgrupo!CUPO - 1
            dtgrupo.Update
            pasa_materia
        End If
    Else
        If dtgrupo!CUPOI = 0 Then
            c = MsgBox(" Grupo saturado!!!! ", 48, "Aviso")
        End If
    End If
End Sub
```

Apéndice "E"

---

```
Else
  dtgrupo.Edit
  dtgrupo|CUPOI = dtgrupo|CUPOI - 1
  dtgrupo.Update
  pasa_materia
End If
End Sub
```

# ***BIBLIOGRAFÍA***

- **Acceso Rápido Visual Basic**  
Stefan Dittrich  
CompuTec Editores, S.A. de C.V.  
1993.
- **Access Fácil**  
Paul McFedries  
Prentice-Hall Hispanoamericana, S.A.  
1995.
- **Introducción a la Programación de Visual Basic.**  
Kenyon Brown  
Grupo Noriega Editores.
- **Guía completa de Visual Basic para Windows.**  
Ross Nelson.  
McGraw - Hill.
- **Ingeniería del Software.**  
Roger S. Pressman.  
McGraw - Hill.
- **Organización de las Bases de Datos.**  
James Martín  
Prentice-Hall Hispanoamérica, S.A.
- **Sistema de Administración de Bancos de Datos.**  
Alfonso F. Cárdenas  
Limusa.



## Bibliografía

---

- **Introducción a los sistemas operativos.**  
**Harvey M. Deitel**  
**Addison-Wesley Iberoamericana.**
- **Automatización de la oficina de Servicios Escolares de la Facultad de Ingeniería por medio de PC.**  
**Tesis**
- **Diseño e Implementación de una Guía Gráfica de Ciudad Universitaria.**  
**Tesis**
- **Diseño e Implementación de un Sistema de Administración para el Centro de Cálculo del Instituto Tecnológico de Estudios Superiores de Monterrey. Campus Ciudad de México.**  
**Tesis.**
- **Implementación de un Sistema de Prestadores de Bienes y Servicios en Ixtapa Zihuatanejo.**  
**Tesis.**
- **Using Visual Basic 3.0**  
**Phil Feldman**  
**QUE**
- **Windows 3.1**  
**Tom Sheldon**  
**Osborne/McGraw-Hill.**

- Microsoft Visual Basic V3.0  
Programmer's Guide.
- Manual Novell Netware Versión 3.11 Concepts.
- Manual Novell Netware Versión 3.11 Administration guide.
- Revista DATA BASE ADVISOR  
Febrero 1993  
Artículo: LANGUAGE PRIMER, DB CONNECTIONS
- **WEB PAGES**
  - [WWW.PROGRESS.COM](http://WWW.PROGRESS.COM)
  - [WWW.MICROSOFT.COM](http://WWW.MICROSOFT.COM)
  - [WWW.MICROSOFT.COM\SUPPORT](http://WWW.MICROSOFT.COM\SUPPORT)
  - [WWW.MICROSOFT.COM\MSACCESS](http://WWW.MICROSOFT.COM\MSACCESS)
  - [WWW.MICROSOFT.COM\MSACCESS\PRODUCTS\OFFICE\ACCESS.HTM](http://WWW.MICROSOFT.COM\MSACCESS\PRODUCTS\OFFICE\ACCESS.HTM)
  - [WWW.MICROSOFT.COM\KB](http://WWW.MICROSOFT.COM\KB)

---

# ***GLOSARIO***

---

## GLOSARIO

**GIF:** Graphics Image File (archivo de imágenes gráficas); Formato de almacenamiento de gráficos que utiliza un mapa de bits comprimido.

**INTERNET:** Es la red de redes más grande del mundo. La componen miles de redes independientes que se comunican usando un protocolo.

**ITEM DE DATOS:** Es el grupo de datos más pequeño. Puede estar formado por cualquier número de bytes o bits.

**MULTIPROCESAMIENTO:** Ejecución de una aplicación por múltiples procesadores dando a cada uno de ellos un porcentaje de esta aplicación.

**ODBC:** (Open Databases Connectivity) Manejador que permite la utilización de bases de Datos Externas, esta categoría incluye las Bases de Datos en el modelo Cliente Servidor.

**SQL:** Structured Query Language (lenguaje de consultas estructurado). Es un lenguaje de base de datos relacional que utiliza un pequeño conjunto de ordenes para todas las operaciones.

## Glosario

---

**TIFF:** Tag Image File format Formato de almacenamiento de gráficos que utiliza un método combinado de mapa de Bits comprimido y sin comprimir.

**TUPLA:** Grupo de valores de campos relacionados específicamente por una relación. También una hilera de la relación, es decir una ocurrencia de un registro sin grupos repetidos.

**AGREGADO DE DATOS:** Es una colección de ítems de datos, dentro de un registro, al que se denomina como un todo.