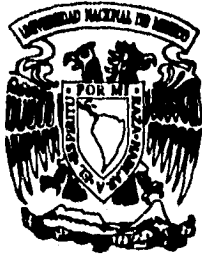


4
27



**UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO**

FACULTAD DE INGENIERIA

Diseño, desarrollo e implementación del sistema para el control de laboratorios del departamento de Física de la División de Ciencias Básicas de la Facultad de Ingeniería

T E S I S

Que para obtener el Título de
INGENIERO EN COMPUTACION
p r e s e n t a

MARIA CONCEPCION ALCANTARA MARTINEZ



DIRECTOR DE TESIS
ING. ROBERTO REYES CHALICO

México, D.F.

1996

**TESIS CON
FALLA DE ORIGEN**

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A mis padres, que siempre estuvieron conmigo,
A mis hermanos y amigos, que siempre me apoyaron,
Y a todos los que no he mencionado y que contribuyeron
a la realización del presente trabajo.

**DISEÑO, DESARROLLO E
IMPLEMENTACIÓN DEL SISTEMA
PARA EL CONTROL DE
LABORATORIOS DEL
DEPARTAMENTO DE FÍSICA DE LA
DIVISIÓN DE CIENCIAS BÁSICAS
DE LA FACULTAD DE INGENIERÍA.**

	Página
INTRODUCCION	3
I. DEFINICIÓN DEL SISTEMA.	
I.1. Definición del Sistema	6
I.1.1. Objetivo del Sistema.	7
I.2. Justificación del Sistema	7
I.3. Alcances del Sistema.	7
I.4. Restricciones del Sistema.	8
I.5. Propuestas y Selección de una estrategia de solución	8
I.5.1 Diagrama General del Sistema.	8
I.5.2. Manejadores de Bases de Datos.	9
I.5.3. Manejadores de archivos.	10
I.5.4. Análisis de Manejadores de Bases de Datos	12
I.5.4.1. DBASE IV	12
I.5.4.2. FoxPro	12
I.5.4.3. Paradox	13
I.5.4.4. Acces	15
I.5.4.5. Clipper	16
I.5.4.6. Visual Basic	17
I.5.4.7. Btrieve	18
I.5.5. Selección de una estrategia de una solución.	20

II. DEFINICIÓN DEL SISTEMA

II.1. Diseño del Sistema	28
II.1.1. Modularidad.	29
II.2. Diseño modular del sistema	30
II.2.1 Inscripciones a laboratorios.	31
II.2.2. Personal Académico.	32
II.2.3. Análisis Estadístico.	32
II.2.4. Impresión de Reportes.	33
II.2.5. Bitácora.	34
II.3. Definición de las entidades.	37
II.3.1. Definición de las relaciones entre entidades.	40
II.3.2. Diseño de menús y pantallas del sistema.	47

	Página
III. DESARROLLO DEL SISTEMA	
III.1. Desarrollo del Sistema.	57
III.2. Objetivos para el desarrollo del Sistema.	58
III.3. Codificación del Sistema	58
III.3.1. Codificación del programa principal del sistema.	59
III.3.2. Codificación del módulo de inscripciones	61
IV. PRUEBAS DE VERIFICACIÓN Y VALIDACIÓN DEL SISTEMA	
IV.1. Pruebas de Verificación y Validación del Sistema	74
IV.2. Pruebas de Unidad	75
IV.3. Pruebas de Integración	76
IV.4. Pruebas de Aceptación	77
IV.5. Creación de casos de prueba.	78
IV.6. Arreo de prueba.	78
IV.7. Prueba de módulos	79
IV.8. Casos de prueba	81
IV.9. Pruebas de integración y aceptación	84
APENDICE A	86
APENDICE B	103
CONCLUSIONES	106
BIBLIOGRAFIA	108

INTRODUCCIÓN

En el Departamento de Física de la Facultad de Ingeniería los procesos que se llevan acabo tales como: inscripciones, análisis estadísticos, captura de datos personales de los profesores que imparten los laboratorio de este Departamento, etc., se elaboran manualmente.

El Departamento se vio en la necesidad de automatizar estas actividades para dejar los procesos manuales, debido a sus ineficiencias y así aprovechar el equipo de cómputo con el que se cuenta. Con el desarrollo de este sistema, se pretende resolver los problemas que se suscitan la realización de cada una de estas actividades, y lo más importante, se dejarán los procesos manuales y estos serán procesados a través de el sistema de computo.

Para el desarrollo de esta Tesis se diseñarán y desarrollarán los programas de software, tomando en cuenta el concepto de módulos de programación, con lo que se pretenderá tener unidades básicas de programación que contengan tanto interfaces de control como Interfaces de datos por medio de procedimientos o funciones que serán llamadas entre los módulos. con el fin de que si en algún momento se requiere de hacer alguna modificación a alguna parte del sistema, este no deje de funcionar en su totalidad, si no parcialmente, ya que la unidad a modificar no estará funcionando hasta que haya concluido su modificación. Una vez que se tengan terminados los módulos se les deberán aplicar las pruebas correspondientes para determinar si cumplen con los requisitos que pide el Departamento de Física, y así liberar cada uno de estos módulos. Los módulos se unificarán para formar el sistemas como tal, para posteriormente proceder a su instalación, y a la entrega del dicho Sistema así como el Manual de Usuario para su operación, al Departamento de Física.

CAPITULO I
DEFINICIÓN DEL SISTEMA

I. DEFINICIÓN DEL SISTEMA

	Página
I.1. Definición del Sistema	6
I.1.1. Objetivo del Sistema.	7
I.2. Justificación del Sistema	7
I.3. Alcances del Sistema.	7
I.4 Restricciones del Sistema.	8
I.5 Propuestas y Selección de una estrategia de solución	8
I.5.1 Diagrama General del Sistema.	8
I.5.2. Manejadores de Bases de Datos.	9
I.5.3. Manejadores de archivos.	10
I.5.4. Análisis de Manejadores de Bases de Datos	12
I.5.4.1. DBASE IV	12
I.5.4.2. FoxPro	12
I.5.4.3. Paradox	13
I.5.4.4. Acces	15
I.5.4.5. Clipper	16
I.5.4.6. Visual Basic	17
I.5.4.7. Btrieve	18
I.5.5. Selección de una estrategia de solución.	20

CAPITULO I

I.1 DEFINICIÓN DEL SISTEMA

En el Departamento de Física de la Facultad de Ingeniería, los procesos que se llevan a cabo se realizan manualmente, en los diferentes laboratorios que forman parte de este Departamento, algunos de estos procesos son:

- Inscripción de los alumnos a los diferentes laboratorios
- Entrega de constancias de calificaciones
- Control de asistencias de profesores
- Recopilación semestral de datos del personal académico
- Análisis estadístico del Índice de aprobación y reprobación que se tienen en los diferentes laboratorios

entre otros.

Los laboratorios que coordina éste departamento son:

- Física Experimental
- Química
- Termodinámica
- Electricidad y Magnetismo
- Óptica.

Debido a que estas actividades se encuentran relacionadas entre sí y que se llevan a cabo semestre tras semestre se tomó la decisión de desarrollar un sistema de captura y procesamiento estadístico de datos, para poder tener así el control de inscripciones y el resultado que refleje el desempeño académico, para poder actuar en beneficio de un mejoramiento del mismo.

Por lo tanto un sistema automático de computadora, es la solución que permitirá cubrir las necesidades de este departamento. Este sistema proporcionará los mismos servicios e información que se obtienen trabajando manualmente, utilizando menos tiempo y personal ofreciendo las ventajas de realizar las correcciones, actualizaciones, consultas y reportes varios según sean las necesidades del departamento con un alto grado de confiabilidad

I.1.1 OBJETIVO DEL SISTEMA

Desarrollo de un Sistema para inscripción a los laboratorios, así como el control estadístico de Alumnos y Profesores de los laboratorios que se imparten en el Departamento de Física de la División de Ciencias Básicas de la Facultad de Ingeniería.

I.2 JUSTIFICACIÓN DEL SISTEMA

Con la implementación de este sistema se pretende automatizar la información que se maneja manualmente en el Departamento de Física, para que en el momento en que se requiera información de dicho Departamento, se tenga de manera inmediata y que no sea necesario tener que esperar para obtenerla.

Otro de los puntos a cubrir es evitar la duplicidad de la inscripción a un laboratorio ya que como no existe un control sobre la inscripción a los laboratorios, un alumno puede estar inscrito de dos a tres veces a un mismo laboratorio con diferentes Grupos. Con la implementación de este sistema se tendrá un control en el proceso de inscripción a laboratorios detectando a aquellos alumnos que pretendan llevar el mismo laboratorio en diferentes Grupos impidiéndole la inscripción a estos y sólo otorgándole una inscripción por laboratorio. Además, todas las actividades que se llevan a cabo en este Departamento se realicen en el menor tiempo del que se está requiriendo actualmente para realizar dichas actividades.

I.3 ALCANCES DEL SISTEMA

El Sistema de Automatización que se desarrollará para el Departamento de Física presentará las siguientes ventajas:

- a) Eficiencia en los procesos que se realicen.
- b) Rapidez en el tiempo de respuesta.
- c) Información oportuna.
- d) Reporte estadístico.
- e) Control de inscripción a los laboratorios evitando así la duplicidad de esta.
- f) Información histórica para el análisis comparativo de los índices de aprobación y reprobación que se tenga en sus diferentes laboratorios.
- g) Recopilación y Actualización de la información del personal académico.

Estas ventajas permitirán al Departamento tener control en las actividades que se realicen, y contar con la información en el momento que se requiera, además se permitirá el acceso a las diferentes opciones del Sistema sólo al personal autorizado, para poder así detectar las fallas que se tengan, el día, la hora y la persona que pudiera haber cometido la falta.

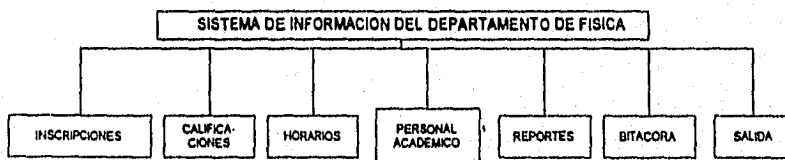
1.4 RESTRICCIONES DEL SISTEMA

Las restricciones que se han fijado para este sistema son:

- Se tendrá que fijar una fecha de duración de la información histórica, debido a que el espacio del disco duro puede presentar una limitante ya que al ir respaldando la información, disminuirá el espacio libre en el disco duro.
- Acceso a opciones del sistema sólo al personal autorizado para su uso.

1.5 PROPUESTAS Y SELECCIÓN DE UNA ESTRATEGIA DE SOLUCIÓN

1.5.1 DIAGRAMA GENERAL DEL SISTEMA



Las Bases de Datos (BD), fueron concebidas como una solución a los problemas causados por el uso de archivos "planos" cuyo diseño depende de una buena forma de los programas que van a manipular la información, haciendo difíciles los cambios y la explotación de datos por otros programas.

El objetivo original de las Bases de Datos fue simplificar la administración de la información de grandes organizaciones, para lo cual se requería equipos con alta capacidad de procesamiento y almacenamiento, y también se observó que este mismo objetivo podía aplicarse a las computadoras personales (PC), surgiendo así diferentes softwares que son ofrecidos como Sistemas de Administración de Bases de Datos (DBMS).

Un sistema manejador de bases de datos consiste en un conjunto de datos relacionados entre sí y un grupo de programas para tener acceso a los datos, con el objetivo de crear un ambiente en el que sea posible guardar y recuperar la información de la base de datos en forma conveniente y eficiente.

Las propuestas que se tienen son:

- Manejadores de bases de datos
- Manejadores de archivos.

I.5.2. MANEJADORES DE BASES DE DATOS

Un *manejador de bases de datos* (DBMS), es un módulo de programa que constituye la interfaz entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicaciones y las consultas hechas al sistema, un manejador de bases de datos es un software por medio del cual se puede :

- almacenar datos,
- obtener datos,
- modificar datos,
- mantener la consistencia de los datos,
- resolver problemas de concurrencia,
- controlar el acceso a los datos,
- permitir hacer respaldos y recuperación de la información

Otro aspecto importante en el diseño de sistemas, es la elección del software, para el desarrollo del Proyecto. En todo sistema para la elección de un lenguaje de programación se deberá tomar en cuenta las ventajas y desventajas que presentan los diversos softwares con los que se puede desarrollar el sistema.

1.5.3. MANEJADORES DE ARCHIVOS

En un *manejador de archivos*, los programas de aplicaciones los elaboran programadores de sistemas en respuesta a las necesidades del usuario. Y cada vez que se necesitan nuevas aplicaciones al sistema se agregan más archivos y programas, y como es de esperarse, el hecho de que los elaboran distintos programadores, implica que tanto los lenguajes de programación como los formatos de los archivos sean diferentes. El ambiente que se acaba de describir es un sistema de *manejador de archivos*, que se apoya en un sistema operativo convencional. Los registros se guardan permanentemente en diversos archivos y se escriben varios programas de aplicación para sacar registros y agregarlos a los archivos apropiados.

A continuación se realiza un análisis comparativo entre los manejadores de archivos y manejadores de bases de datos.

	Ventajas	Desventajas
Manejador de Bases de Datos	<ul style="list-style-type: none"> • Redundancia controlada • Seguridad a nivel de campos • Integridad • Independencia datos, programas • Recuperación de información • Control de concurrencia • Respaldos en línea 	<ul style="list-style-type: none"> • Alto costo • Bajo rendimiento
Manejador de Archivos	<ul style="list-style-type: none"> • Menor Costo • Alto rendimiento(Alta velocidad) 	<ul style="list-style-type: none"> • Redundancia • Incongruencia • Programas dependientes de los datos • Seguridad solo a nivel de archivos

Para poder tomar la decisión para elegir alguno de ellos, además de tomar en cuenta los aspectos técnicos debe evaluarse el soporte y la solidez de cada proveedor que garantice apoyo técnico y evolución en los productos.

Del cuadro anterior se decide que se tomará un *manejador de bases de datos*, como la elección de la primera propuesta, en lugar de elegir un *manejador de archivos*, esto debido a la tabla anterior. Ahora se analizará cual de los manejadores de Bases de Datos nos dará mejores resultados, tomando en cuenta sus características y las del proyecto a realizar.

En la elección de una DB deben tomarse en cuenta las necesidades de información que se desean satisfacer, el número de usuarios y el alcance del proyecto. Es importante determinar que tan necesario es por ejemplo, adquirir un paquete que corra en Windows, el cual requiera más recursos, aunado al costo del hardware está la consideración de si se necesitará la conectividad con otros equipos, ya que en las aplicaciones cliente-servidor es más importante la velocidad del servidor de BD que la de su contraparte el cliente.

Cuando la BD se instala en un servidor de red, el desempeño cambia ya que el comportamiento de estas será diferente como *front-end*.

El software que se puede usar para la realización de este sistema son:

- Btrieve,
- Visual BASIC,
- Acces,
- Paradox
- Clipper,
- FoxPro
- dBASE IV.

Una vez que se tiene la lista de diferentes opciones de programación se procede a hacer un análisis de tipo cuantitativo para realizar la primera eliminación del software empleando los criterios de disponibilidad y madurez del compilador entre otros. Para posteriormente proceder a realizar un análisis intermedio empleando criterios cuantificables como:

- Tiempos de ejecución de la programación
- Diversidad de proveedores
- Versatilidad
- Tiempo y costo de desarrollo
- Mantenimiento durante el ciclo de vida

Con el advenimiento de las Redes de microcomputadoras, apareció la necesidad de incluir en estos paquetes la capacidad de compartir la información entre varios usuarios de manera simultánea. Además, hay que considerar la aparición de la arquitectura *cliente-servidor* en el que no solamente se comparten datos en la red sino que existe un servidor para los procesos de DB.

Cabe hacer mención que en los últimos 4 o 5 años, virtualmente la mayoría de los paquetes que habrán sido diseñados para su ejecución en el sistema operativo DOS se han modificado para trabajar bajo el ambiente Windows, con las implicaciones de manejos gráficos que éste conlleva.

1.5.4. ANÁLISIS DE MANEJADORES DE BASES DE DATOS

1.5.4.1. dBASE IV

En sus diferentes versiones dBASE es probablemente el DBMS que más éxito ha tenido debido a que fue uno de los primeros en aparecer, además de que introdujo el estándar para los datos y gracias a su sencillez tuvieron éxito sus lenguajes de definición de datos y programas. De entre las facilidades que ofrece, aunque no son del todo versátiles ni rápidas, se encuentra la creación de consultas, formas, reportes y etiquetas. dBASE IV incluye la opción de utilizar SQL para BD relacionales.

Una de sus limitaciones es que su velocidad de procesamiento es baja, debido a que los programas se ejecutan por medio de un intérprete, es decir, que no genera un código binario, sino que cada instrucción se ejecuta según se van encontrando. Para evitar esto es necesario que se tenga el paquete completo o adquirir el compilador por separado para poder obtener el código binario. A pesar de que dBASE IV genera un código ejecutable, por el lado de la velocidad no puede decirse que haya mucho avance, además de que se requiere de muchísimos más recursos de procesador, memoria y disco.

Requerimientos:

- Computadora con procesador 286 o superior.
- 2 MB en RAM.
- Versión de sistema operativo MS-DOS 3.3 o superior.

1.5.4.2. FoxPro

FoxPro es una opción que surgió para los desarrolladores en dBASE; en un principio se llamó FoxBase y con el tiempo se llamó FoxPro. Entre sus características están la emigración que tuvo a plataformas Unix y Macintosh. Este ambiente se creó para el manejo de base de datos (BD) en forma muy semejante a como lo hace dBASE, pero con la característica de permitir una velocidad de ejecución de las aplicaciones mucho mayor gracias a que, además de tener un modo de operación como intérprete, permite generar una versión ejecutable que se puede distribuir junto con un programa runtime. FoxPro extendió el lenguaje original de dBASE para darle mayor funcionalidad y manejo de multiusuarios, además de incluir un generador automático de pantallas y reportes.

La versión que apareció para Windows, además de permitir el diseño con la interfaz gráfica y el uso de SQL para hacer consultas, integra la capacidad de usar BD residentes en servidores tales como Oracle y Sybase. Esto último aumenta la posibilidad de crear aplicaciones cliente-servidor en las que se optimiza el acceso a

los datos siempre que éstos no se tengan localmente. Sin embargo, una consecuencia del cambio de plataforma (de DOS a Windows) es que FoxPro para Windows disminuye su rendimiento en cuanto a la velocidad de ejecución (para aplicaciones locales) e incrementó en forma considerable sus requerimientos de espacio en disco y memoria.

Consideraciones: Es parecido a Paradox sin la robustez de éste aplica al desarrollo de alto volumen.

FoxPro 2.5 para Windows:

- Tiene flexibilidad de manejar hasta 255 áreas de trabajo.
- La ayuda que ofrece es muy amigable y fácil de utilizar.
- La creación de reportes de varias columnas se torna más sencillo.
- Se puede lograr un fácil acceso a las etiquetas más utilizadas.
- Se pueden diseñar pantallas gráficas, utilizar todas las fuentes y colores de Windows, figuras, gráficas, botones.
- Con el administrador de proyecto incluido se facilitan las decisiones y el diseño del proyecto a realizar.
- Las consultas carecen de enlace dinámico y no se pueden crear pantallas demasiado complejas.

Requerimientos de equipo:

- Microsoft Windows 3.0 o superior
- PC con INTEL 386 o mayor
- 4 MB de memoria
- 14.5 MB de espacio en disco
- Monitor EGA o de mayor resolución

Plataformas en las que Trabaja:

- Novell NetWare
- Windows for Workgroups
- Windows NT
- LAN Manager

I.5.4.3. Paradox

Paradox tiene su origen en una filosofía distinta a la de los manejadores tipo Xbase. Desde su inicio ha mostrado un esquema de funcionamiento que da una idea un poco más cercana a un DBMS: en lugar de archivos maneja tablas como ORACLE, contiene el concepto de vistas lógicas, realiza validación de datos, integridad referencial, maneja índices primarios y secundarios y seguridad de archivos. Además posee una tendencia a administrar en forma coherente todos los elementos de un proyecto.

Como todo software, tiene sus ventajas y desventajas. Entre las desventajas tenemos las siguientes a diferencia de dBASE, Paradox presenta una mayor dificultad para aprenderlo y operarlo, y un mayor requerimiento de recursos, a la vez se obtienen tiempos de ejecución un poco menores en modo local. Las versiones actuales de Paradox están hechas para Windows y para DOS. Para Windows incluye el modelado de datos en forma visual para formas y reportes, un ambiente de desarrollo orientado a objetos, capacidad de utilizar simultáneamente datos de Paradox y dBASE, y la conectividad a servidores de BD como Oracle, Sybase e Informix, mediante IDAPI-tecnología propietaria de Borland y ODBC-estándar ampliamente aceptado. Además, sus lenguajes de programación PAL (para DOS) y Object PAL (para Windows) son muy poderosos.

Consideraciones: Como BD es un producto más robusto que puede utilizarse como *Front-end*, como BD en redes locales y en modo local independiente de servidores, tanto como para desarrollo de aplicaciones de alto volumen como de menor escala.

Paradox 4.5 y 5.0 para Windows

- Ofrece aplicaciones gráficas en corto tiempo.
- Se maneja por medio de botones para crear de forma sencilla tablas, reportes, formas, gráficas.
- Se considera como el manejador más dependiente de un dispositivo de navegación en pantalla como lo es el ratón.
- Su lenguaje de programación con debugger, editor, ayuda y orientación a objetos se ha convertido en un ambiente agradable y amigable para la tarea de crear sistemas.
- Creación de reportes de calidad, tiene diseñadores de formas y reportes.
- Versatilidad para visualizar relaciones existentes entre tablas.
- Existe integridad y seguridad de la información.
- En una opción existente de trabajo en grupo, se ofrece la posibilidad de expresar quien puede modificar o consultar nuestra información, con quien compartir objetos, etc.
- Diferentes alternativas para refrescar la pantalla después de que otro usuario hace algún cambio a los datos que estamos consultando.
- Facilidad de crear gráficas dentro del mismo ambiente.
- No tiene la facilidad de generar un programa ejecutable de la aplicación realizada.

Plataformas en las que Trabaja

- Novell Netware
- Windows for Workgroups 3.11
- Windows NT 3.11
- LANtastic

Requerimientos de equipo (para la versión 5.0)

- Microsoft Windows 3.1
- PC con procesador 80386 o mayor.
- 6 MB de memoria RAM (se recomiendan 8 MB)
- 20 MB de espacio en disco para instalación completa, 15 para instalación mínima.
- Monitor EGA o de mayor resolución.
- Mouse.

1.5.4.4. Acces

Es uno de los productos de más reciente aparición para manejo de BD en PC. En su diseño, a semejanza de Paradox, contiene muchos de los elementos de un DBMS formal: manejo de tablas, índices, llaves primarias y foráneas (también llamadas secundarias), integridad referencial, soporte de transacciones, seguridad de datos, y consultas mediante SQL. Existen varias formas en las que se puede utilizar Acces: generación automática de declaraciones SQL mediante una herramienta gráfica de consultas, el uso de macros para controlar las pantallas y reportes generados automáticamente y para validar la información, además del uso del lenguaje de programación Acces Basic. Cabe mencionar que este lenguaje tiene mucha semejanza con Visual BASIC para Windows, y que éste último a su vez contiene un motor (la parte básica del funcionamiento) de Acces, por lo que es posible realizar programas en Visual Basic que exploten los datos de Acces. Dentro de Acces es posible utilizar en forma transparente datos de Btrieve, dBASE y FoxPro, además del acceso a datos en servidores de BD mediante el estándar ODBC (Open Database Connectivity). De manera similar a Paradox, las demandas de espacio en disco, memoria y procesador son altas, y la ejecución de aplicaciones en forma local es muy lenta.

Consideraciones: Es un producto dual sirve como *Front-end* en cliente-servidor y como base de datos local, esta orientado a usuario final, no especialista en computación.

Microsoft Acces 2.0

- Ofrece un tutorial para familiarizarse con el ambiente, ayuda rápida mediante iconos y ayuda disponible para cada paso que se realice.
- Ayuda a crear pantallas de forma gráfica paso por paso.
- Permite crear gráficas de pastel, barras, líneas.
- La generación de reportes ofrece múltiples opciones de forma elemental y guiada.
- Incluye las búsquedas, tablas, reportes, formas y macros en un sólo archivo.
- Tiene requerimientos muy altos.
- No genera ejecutable.

Requerimientos de equipo

- Microsoft Windows 3.1
- PC con INTEL 386 o posterior.
- 6 MB de memoria (se recomiendan 8 MB).
- 17.8 MB de espacio en disco.
- Monitor EGA o de mayor resolución.

Plataformas en las que Trabaja

- Novell NetWare
- Windows for Workgroups
- Windows NT
- LAN Manager

1.5.4.5. CLIPPER

Es un lenguaje de programación de bases de datos completo, con todas las herramientas necesarias para desarrollar aplicaciones de bases de datos totalmente independientes. Clipper, nació como un compilador del lenguaje dBASE III para generar el código binario para PC. Clipper básicamente permite manipular el lenguaje de programación de dBASE III, con algunas pocas restricciones, sobre todo en cuanto a la consulta de datos. Sin embargo las continuas extensiones al paquete le han agregado mucha más instrucciones que las que tiene dBASE y lo hacen más versátil. Clipper permite interactuar con otros lenguajes como C, ensamblador, etc., y generar aplicaciones para red debido a su concurrencia. Además no se requiere tener dBASE ya que con Clipper se pueden generar los archivos como índices.

En la actualidad Clipper no tiene la posibilidad de conectarse a servidores de DB y tampoco existen una versión para Windows.

Consideraciones: Es una excelente opción para el desarrollo de aplicaciones de bases de datos por tres razones principales:

- Velocidad, debido a que Clipper compila su propio dialecto de dBASE produciendo aplicaciones muy rápidas.
- Flexibilidad, como Clipper es un lenguaje de programación de alto nivel, tiene un gran número de funciones y comandos (cerca de 300 en la versión 5.01) que están diseñados específicamente para la creación de aplicaciones que se deseen hacer, evitando así la generación de cientos de líneas de código.
- Seguridad, se refiere a que las aplicaciones desarrolladas producen un programa ejecutable .EXE, totalmente independiente, protegiendo sus programas una vez que son compilados, nadie puede ver ni copiar parte de ellos.

Requerimientos:

- Una computadora IBM-PC.
- Versión del sistema operativo MS-DOS 3.1 o superior
- Por lo menos una unidad de disco flexible de 5¼ o 3½ pulgadas.
- Un disco duro con al menos 6 MB de espacio libre para la instalación.
- Por lo menos 400K de RAM libre.

1.5.4.6. Visual BASIC

Hace más de dos años Microsoft Lanzo al mercado un sistema de Programación de Visual BASIC 1.0 como alternativa para desarrollar aplicaciones que corrieran bajo el ambiente de Windows. Esta primera versión estaba Integrada por un editor, un depurador y un compilador, con un modelo de programación manejado por eventos y un conjunto de herramientas visuales que permitían desarrollo de programas. Esa nueva propuesta aligeraba la tediosa tarea de aprender a usar lenguajes que como entonces existían para programas enfocados para Windows, tales como C++.

Microsoft asegura que la documentación que acompaña a Visual BASIC es la manera más rápida y más sencilla de crear aplicaciones para el ambiente Windows, facilitando el diseño de aplicaciones atractivas y útiles que explotan de manera total la interfaz gráfica del usuario. Visual BASIC, es un ambiente de programación orientada a eventos que han avanzado muchísimo y con cambios significativos desde su versión 1.0 hasta la versión 4.0 en menos de cinco años. A diferencia de lo que ocurre en los lenguajes procedurales (como PASCAL o Clipper). El desarrollo de Visual BASIC se efectúa por medio de una programación por eventos, los cuales requieren de la existencia de los objetos, es decir, del las formas o de los controles.

Un evento es una acción que es reconocida por una forma o un control. Los controles son elementos principales de visual BASIC y entre ellos se encuentran las cajas de textos, los botones de comandos y las cajas con listas de selección. Cada tipo de control tiene su conjunto de propiedades y eventos a los que puede responder. Por ser un lenguaje de componentes, Visual BASIC soporta su poder en lo que se denomina Controles y que son añadidos del lenguaje creados en C para realizar tareas que en código se llevarían varias líneas de programación. Este lenguaje no esta orientado a objetos, lo que pudiese suponer una desventaja. No obstante, el tiempo de desarrollo es mucho más reducido. Visual BASIC carece, además de estructuras de datos (como apuntadores y listas ligadas) pero emplea controles en la mayoría de las funciones donde la aplicación de estas estructuras es vital.

La versión 4.0 ofrece un motor de acceso de datos que elimina los errores que surgían en las versiones anteriores. Es completamente compatible con Windows 95 y Windows NT. Añade clases de ambiente de programación y pese a no ser un

lenguaje orientado a objetos , ofrece tiempos de desarrollo muy reducidos. No obstante es mejor olvidarse de esta versión si no se piensa actualizar a Windows 95 o NT, de inmediato. La sugerencia es continuar con la versión 3.0 de Windows 3x y olvidarse de la 4.0 hasta actualizarse a Windows 95 o NT.

Para resolver este problema Microsoft ofrece también la versión de MS-DOS de Visual BASIC que, por desgracia, solo se concretó a la versión 1.0. Sin embargo, si no le interesa Windows, pero le agrada el desarrollo basado en ventanas, menús y componentes. Visual BASIC es la mejor opción

Requerimientos de Hardware y Software:

- computadora compatible con IBM con un procesador 386 para las versiones anteriores a la 4.0 y para la versión 4.0 procesador 486 o superior
- Memoria RAM de 8 MB.
- Windows 3.1 o superior para las versiones anteriores a la 4.0.. Y Windows 95 o NT para la versión 4.0
- Disco duro con espacio de 20 MB ó más para la instalación

1.5.4.7. Btrieve

Es un manejador de bases de datos que es un producto de Novell, el cual permite que la base de datos que se genera sea tanto compartida como para manejo personal en la PC, una de las ventajas que ofrece este manejador de bases de datos, es la utilización de un lenguaje de programación como BASIC, PASCAL, C, C++, FORTRAN. Btrieve tiene una cierta estructura para ser manejado con cualquiera de estos lenguajes de programación. Cabe mencionar que estas estructuras ahora también pueden ser vistas desde Visual BASIC.

Actualmente, se considera a Btrieve como una de las bases de datos mas confiables del mundo. Permite trabajar como cliente/servidor, sin que se pierda el control por parte del programador. Este puede navegar hacia arriba, abajo, y hacia los lados de la base de datos a nivel de registro, lo cual facilita el desarrollo de aplicaciones y estructuras de bases de datos, pudiendo incluso agregarse aplicaciones de SQL. Btrieve permite el desarrollo de estructuras relacionales y seguirlas con precisos controles direccionales para actualizar, recuperar, insertar y borrar datos distribuidos. Estos controles direccionales se integran con el código de la aplicación que se esté utilizando, lo cual permite que gradualmente se actualicen las aplicaciones normales, a cliente/servidor. Como este cambio en el manejo de los datos es transparente, el usuario final no requiere de ningún reentrenamiento para seguir trabajando normalmente.

Una vez que se ha creado alguna aplicación, esta puede ser instalada en cualquier configuración sin necesidad de adaptaciones (desde una laptop, hasta un servidor bajo Netware, Windows NT u OS/2), y pudiendo soportar cientos de

usuarios. El grado de confiabilidad de Btrieve, lo ha hecho digno de ser utilizado en aplicaciones muy delicadas como en el caso de los sistemas de contabilidad. Puede manejar bases de datos multiservidor, que soportan a cientos de usuarios con tiempos de respuesta de fracciones de segundo. Garantiza la integridad de los datos a través de un rico soporte de procesamiento de transacciones, controles de integridad referencial, y recuperación automática de archivos. En caso de que ocurriera un falla en el servidor, se cuenta con utilerías que permiten la recuperación de la última transacción

Otra característica importante incluye a los algoritmos de indexación interna, que usan un caché de pagina extenso, para una recuperación y actualización muy rápidas, además de un balanceo de índice automático para asegurarse de que la velocidad de acceso a los datos no se deteriore conforme van creciendo los archivos. Btrieve también ofrece "operación continua", al mantener siempre los datos en línea, sin interrupción para hacer respaldos o procesamiento por lotes.

Btrieve está construido sobre la denominada *MicroKernel Database Engine*, la cual almacena todos los datos en un formato común, independiente del modelo, por lo cual es posible trabajar con otras aplicaciones teniendo todas ellas acceso concurrente a todos los datos. No se requiere de mucho mantenimiento ni de optimizar recursos, pues Btrieve incrementa las bases dinámicamente, de modo tal que al borrar datos, por ejemplo, este espacio es reutilizado automáticamente, y además es optimizado todo el espacio usado en disco mediante compresión automática de datos, truncamiento de espacios en blanco, y administración de registros de longitud variable.

En resumen, las características de Btrieve son las siguientes:

- La *MicroKernel Database Engine* da acceso concurrente a todos los datos.
- Función de registro de cambios hechos a un archivo designado.
- Mecanismo de recuperación ante fallas del servidor.
- Algoritmos automáticos de caché y métodos de búsqueda para mejorar el acceso y la compartición de caches
- Tamaño de 4 GB con operaciones en una porción del registro
- Soporte mejorado para registros de longitud variable (mayores a 64K)
- Balanceo automático de índices
- Respaldo en línea
- Encriptamiento y desencriptamiento dinámicos
- Aseguramiento de archivos de datos, con nombres asignados del propietario
- Detección de "deadlocks"
- Bloqueo sencillo o múltiple de registros
- Capacidad de hacer lecturas mientras se reindexa

Requerimientos del Sistema de hardware y software:

- Computadora 386 o superior
- Sistema operativo DOS 3.x o superior, OS/2 o superior, Windows 3.1 o NT

Especificaciones

- Tamaño máximo de archivo 4 GB
- Tamaño máximo de DBMS: ilimitado
- Tamaño de registro: ilimitado
- No. máximo de registros ilimitado
- Longitud máxima de llave: 255 byte
- Máximo de llaves por archivo: 119

1.5.5. SELECCIÓN DE UNA ESTRATEGIA DE SOLUCIÓN.

A continuación se resumen las características de los Manejadores de Bases de Datos analizados, para obtener una mejor evaluación de estos:

Acces de Microsoft, es un producto dual; sirve como *front-end* en *cliente-servidor* y como *BD local* para aplicaciones de cierto calibre. Está orientado a un usuario final no especialista en computación, un profesionalista libre que quiere administrar su pequeño negocio con un sistema, o para departamentos pequeños que desean utilizarlo como *front-end* (conectividad).

Paradox, de Borland, se parece más a una BD profesional, puede ser usado tanto por profesionalistas libres como por pequeñas y medianas empresas o departamentos. Como BD es un producto más robusto que puede utilizarse como *front-end*, como BD en redes locales y en modo local independiente de servidores, tanto para desarrollo de aplicaciones de alto volumen como de menor escala. Desde luego guardando las proporciones debidas con respecto a BD profesionales.

FoxPro, de Microsoft, es parecido a Paradox sin la robustez que este aplica al desarrollo de aplicaciones de alto volumen, para pequeñas y medianas empresas o departamentos. En aspectos de conectividad es mejor Acces. La versión de Windows puede (aunque no se recomienda) usarse con cierta facilidad por profesionalistas con pocos pero que se tengan conocimientos de computación.

Visual BASIC, de Microsoft apesar de que existen tanto la versión para DOS como para Windows, requiere de un muy buen equipo de cómputo, además de estar a la vanguardia en cuanto al software de Windows. Esta orientado a un usuario final especialista en computación.

BTRIEVE, de Novell es de fácil manejo y considerada como una de las más confiables bases de datos del mundo, permite trabajar como cliente/servidor sin que se pierda el control de los programas que se generan en lenguajes como BASIC, PASCAL, FORTAN, y/o C++. además de soportar a cientos de usuarios garantizando la integridad de su información.

dBASE IV, de Microsoft es una de los iniciadores de bases de datos, presenta ciertas desventajas ya que su velocidad de procesamiento es baja y requiere de muchísimo espacio en disco, asico memoria y recursos del procesador. Esta orientado, si no a un profesional de computación a aquel usuario que tenga las nociones de programación.

Clipper, de Computer Associates, difícilmente será elegido por profesionistas libres, que no tienen conocimientos de programación. Generalmente lo utilizan empresas y departamentos pequeños que cuentan con programadores.

En base a las características descritas anteriormente, se generó una tabla comparativa considerando los siguientes aspectos:

- **Integridad Referencial.** Se refiere al vínculo entre llaves primarias y secundarias (también llamadas foráneas) de dos o más tablas distintas.
- **Manejo de Llaves.** Significa que permite definir y manipular llaves primarias y secundarias.
- **Seguridad de Datos.** Es la capacidad de un DBMS para evitar tentativas de acceso no permitidas.
- **Transacciones.** Significa que el DBMS permite garantizar operaciones de inicio (begin transaction), de compromiso (commit) y de nulificación (roll back), para propósitos de cómputo consistente y confiable.

Producto	Maneja llaves	Seguridad de Datos	Corre sobre Windows Dos		Cliente/servidor	Soporte de transacciones*
FoxPro			•	•	•	
Paradox Win	•	•			•	•
Paradox Dos	•	•	•	•	•	
Acces	•	•	•	•	•	•
DBASE IV		•	•	•	•	•
Visual Basic	•	•	•	•	•	•
Btrieve	•	•	•	•	•	•
Clipper			•	•		
Visual Basic			•	•	•	

Análisis Comparativo de BD

*No existe Windows 4 y NT van a soportar transacciones por sí mismos y apoyarán a las BD's que corran en ellos.

Producto	Observaciones finales
dBASE	No tan actualizado tecnológicamente y limitaciones como DBMS
Clipper	No tan actualizado tecnológicamente, no es un DBMS, pero es más rápido y requiere un mínimo de recursos
FoxPro para DOS	Las mismas limitaciones de DBASE aunque corre mucho más rápido.
FoxPro para Windows	Tecnología actual, rápido, aunque carece de seguridad y requiere de mayores recursos que la versión para DOS.
Paradox para DOS	Tecnología actual, buen nivel de seguridad, pero es lento en forma local y requiere mucho más recursos.
Paradox para Windows	Tecnología actual, rápido, buen nivel de seguridad, pero sólo trabajo archivos en formato Paradox DOS.
Acces	Tecnología actual, buen nivel de seguridad, pero es muy lento y requiere de muchos recursos.
Btrieve	Tecnología actual, no requiere de muchos recursos, rápido al acceso de la información, con un alto nivel de seguridad.
Visual BASIC Windows	Tecnología actual, rápido si se tiene actualizada la versión de Windows a la versión 95, requiere de muchos recursos.

En la elección de una Bases de Datos deben tomarse en cuenta las necesidades de información que se desean satisfacer, el numero de usuarios y el alcance del proyecto. A pesar de que en general el costo del hardware ha ido disminuyendo, es importante determinar qué tan necesario es, por ejemplo, adquirir un paquete que corra en Windows, el cual requerirá mas recursos. Aunado al costo del hardware está la consideración de si se necesitará o no conectividad con otros equipos, pues como ya se dijo, en las aplicaciones *cliente-servidor* es mas importante la velocidad de proceso del servidor de BD que la de su contraparte el cliente.

Cuando la BD misma se instala en el servidor de una red local (lo que no se aconseja con algunos productos, por su diseño y mercado) el desempeño cambia porque, por citar un ejemplo, Paradox maneja tablas (más parecido a una BD profesional) y con una consulta SQL Incrementa su desempeño ya que la consulta se ejecuta en el servidor. Otras BD's manejan archivos y una consulta al servidor hace que se traigan los archivos al cliente para ejecutarla.

En la arquitectura cliente-servidor y contando en el servidor con DBMS's como Oracle, Informix o Sybase, entre otras, y en el cliente con alguna de las BD's descritas, la situación cambia. Ahora el comportamiento de estas como front-end será diferente y unas tendrán ventajas sobre otras.

Para efectos de aterrizar los resultados mostrados es preciso ubicar el mercado de cada producto. Para tomar la decisión por uno de ellos, además de aspectos técnicos debe evaluar el soporte y la solidez de cada proveedor que le garantice apoyo técnico y evolución en los productos. La mayoría de las veces, al menos en corporativos y grandes empresas, la decisión se inclina mas por el tipo, seriedad y tamaño del proveedor que por los productos mismos, se trata al fin y al cabo de decisiones estratégicas más que tecnológicas. Por lo que se debe considerar ambos aspectos.

Una vez descritas las características de los posibles manejadores de Bases de Datos que se tienen, las características que debe cubrir este manejador de bases de datos para su selección en la elaboración de este proyecto son:

- Que pueda ejecutarse en cualquier equipo de computo con procesador 386 o superior.
- Que la velocidad de respuesta sea rápida.
- Que sea capaz de ser transportadas sus bases de datos a un sistema de red.
- Que sea compatible con otras bases de datos.
- Que tenga seguridad e integridad en cuanto al manejo de la información.

De entre los manejadores de bases de datos que se analizaron anteriormente las dos opciones finales con las que se puede contar para la elaboración de este sistema son:

- Visual BASIC que por las características antes mencionadas, y por el poco tiempo que se lleva para la elaboración del sistema.
- Btrieve que al igual que Visual BASIC, cumple con las características para la elaboración de este sistema y por el gran soporte que este tiene.

Considerando las características que estos presentan y la experiencia que se tiene en el manejo de estos dos Manejadores de Bases de Datos, se concluye que el software que es más viable para la elaboración de este sistema es **BTRIEVE**.

Btrieve para su ejecución requiere de un lenguaje de programación por lo que se utilizará PASCAL V6.0. Además hay que considerar que para la ejecución de los programas tanto en PC's como en estaciones de trabajo no cambia ni la estructura, ni los programas. Esto para los usuarios es una ventaja ya que si en este momento no se cuenta con alguna red de comunicaciones y si se piensa instalar alguna a mediano o largo plazo, sólo basta con trasladar la información al servidor de la red y antes de correr BTRIEVE.EXE se ejecuta BREQUEST.EXE que en este caso Brequest.exe nos permite compartir la información de los archivos, incluso se puede monitorear cuantos archivos y que usuarios están accediendo los archivos tipo Btrieve que existen en el servidor.

Además de las características ya mencionadas Btrieve permite:

- Manejar grandes cantidades de información
- Rapidez en la consulta de información.
- Seguridad en el almacenamiento de datos

Otros de los criterios que intervinieron en la elección de este manejador de bases de datos fueron:

- los técnicos y
- los administrativos.

De los criterios de tipo administrativo que pueden destacarse:

- Tiempo y el costo de desarrollo. En cuanto al tiempo de desarrollo este es mínimo por la experiencia que se tiene en cuanto a la programación con el lenguaje PASCAL y el manejador de bases de datos Btrieve. Además el costo de el manejador Btrieve y PASCAL no son tan elevados si tomamos en cuenta el costo del software, incluso puede reprogramarse este sistema con otro lenguaje
- Mantenimiento durante el ciclo de vida. Aquí se midió la facilidad con lo que un programa puede modificarse. En este caso PASCAL nos permite modificar sus programas con facilidad para obtener los resultados que el usuario requiere, siempre y cuando la persona que modifique el programa sea un programador.

De los criterios de tipo técnico pueden destacarse:

- La transportabilidad. Btrieve consta de una utilidad llamada BUTIL.EXE la cual nos permite producir un código ACSII que puede ser utilizado en diferentes procesadores, incluso Btrieve puede ser manejado con SQL y utilizado en manejadores de bases de datos como Visual BASIC.
- La eficiencia del tiempo: Los programas generados tienen un tiempo de ejecución inmediato, sin importar la cantidad de información que se tenga en su base de datos, cabe destacar que apesar de las grandes cantidades de información la respuesta es relativamente rápida.

CAPITULO II
DISEÑO DEL SISTEMA

II. DEFINICIÓN DEL SISTEMA

	Página
II.1. Diseño del Sistema	28
II.1.1. Modularidad.	29
II.2. Diseño modular del sistema	30
II.2.1 Inscripciones a laboratorios.	31
II.2.2. Personal Académico.	32
II.2.3. Análisis Estadístico.	32
II.2.4. Impresión de Reportes.	33
II.2.5. Bitácora.	34
II.3. Definición de las entidades.	37
II.3.1. Definición de las relaciones entre entidades.	40
II.3.2. Diseño de menús y pantallas del sistema.	47

CAPITULO II

II.1. DISEÑO DEL SISTEMA

En esta etapa de diseño se definen los siguientes elementos:

- Diseño Arquitectónico: En la cual se definen las estructuras de datos.
- Diseño de datos: En la cual se definen las relaciones entre los principales elementos de los programas.

En la etapa de diseño del sistema se deben cubrir todos los requerimientos del usuario, además de que la información generada para el usuario sea de calidad. En el Departamento de Física de la Facultad de Ingeniería, los procesos que se deben cubrir con el sistema son:

- inscripción de los alumnos a los diferentes laboratorios
- entrega de constancias de calificaciones
- recopilación semestral de datos del personal académico
- análisis estadístico del índice de aprobación y reprobación que se tienen en los diferentes laboratorios que coordina a este Departamento (Física Experimental, Química, Termodinámica, Electricidad y Magnetismo y Óptica).

Como estas actividades se encuentran relacionadas entre, sí y se llevan a cabo semestre tras semestre, se tomó la decisión de desarrollar un sistema de captura y procesamiento de información para satisfacer los requerimientos del Departamento, proporcionando los mismos servicios e información que se obtienen trabajando manualmente, pero utilizando menos tiempo y personal, y ofreciendo las ventajas de realizar las correcciones, actualizaciones, consultas y reportes varios según sean las necesidades del Departamento con un alto grado de confiabilidad.

Un buen diseño, dependiendo de la aplicación y de los registros del proyecto en particular, puede ser:

- Uno que permita una codificación muy eficiente
- Uno donde la aplicación sea lo más compacta posible
- Uno de fácil mantenimiento.

Este último implica minimizar el costo de los cambios del sistema es decir que sea comprensible y que las modificaciones deben tener un efecto local. Estas dos características se logran si el diseño es muy coherente y poco acoplado. Para esto es preciso manejar el concepto de Modularidad.

II.1.1. MODULARIDAD

Es un concepto de gran importancia dentro del diseño. Para aplicar este concepto es necesario dividir en elementos con nombres y direcciones separadas llamadas módulos, los cuales al integrarlos deberán de satisfacer las necesidades del sistema. La Modularidad es un requisito indispensable al establecer el software, ya que los módulos que se generen deben ser capaces de desarrollar una y solo una tarea. La independencia de estos módulos se da en base a los siguientes criterios:

- **Cohesión:** Cuando el módulo ejecuta una tarea sencilla y requiere poca interacción con procedimientos que se ejecutan en otras partes del programa, es decir, que un programa es coherente si cada instrucción de la unidad sea esencial para que alcance su propósito.
- **Acoplamiento:** Es la interconexión que existe entre módulos. Entre menos interconexiones existan entre los módulos, el sistema será más eficiente. Este concepto se relaciona con la coherencia ya que es un indicador de las conexiones entre los programas, ya que mientras el sistema sea más débilmente acoplado, se contará con un sistema cuyas unidades sean lo más independientes posibles, garantizando que el sistema sea más eficiente..

Cuando los módulos de un sistema presentan mucha coherencia y poco acoplamiento, implica que cualquier unidad de programa se puede remplazar por una unidad equivalente con poco o ningún cambio en las otras unidades del sistema.

II.2. DISEÑO MODULAR DEL SISTEMA

Un buen diseño se logra utilizando una metodología consistente, una de estas metodologías es el *diseño funcional descendente* que se utiliza para proyectos de pequeña y gran escala. Las notaciones que se pueden utilizar para describir un diseño de software son:

- Diagramas de flujo que se utilizan para describir un diseño de sistema de alto nivel, mostrando como se van transformando los datos al pasar por un componente del sistema.
- Diagramas de estructura que son las gráficas de jerarquía que muestran la relación estructural de los componentes de un sistema.
- Lenguaje para la descripción del diseño en la notación con algunos atributos de los lenguajes de programación para describir las operaciones de control y diseño detallado.

Cabe mencionar que las notaciones utilizadas solo son complementarias y que no hay notación única para describir los diseños de un sistema. Para esto es preciso que en cualquier proyecto de sistema esté bien documentado mostrando las notaciones y gráficas para ilustrar la estructura del mismo.

Empleando el concepto de modularidad en el diseño del sistema se desarrolló el siguiente modelo conceptual de información considerando los requerimientos del usuario.

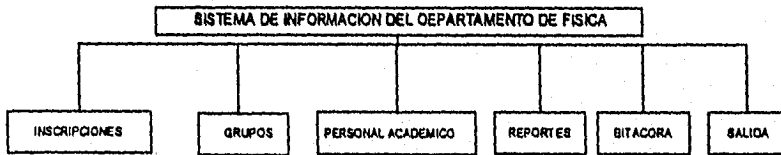


Diagrama general del sistema.

Es en este momento donde se procede a la especificación detallada del proceso o función que cada módulo realiza, indicando todos los datos con que cada módulo interactúa; para esto son necesarias las especificaciones de los datos internos y externos.

II.2.1. INSCRIPCIONES A LABORATORIOS

Este módulo obtendrá los datos de un archivo "ALUMNO.DAT" y transferirá la información al archivo "LAB.DAT" presentando su información en pantalla para determinar a que laboratorio se va a inscribir el alumno. Una vez capturado la clave y el grupo del laboratorio se diagnosticará si procede la inscripción en los casos donde el alumno no se encuentre en el sistema, la modificación para el cambio de laboratorio y/o grupo que el alumno solicite, deberá tomarse en cuenta el cupo disponible que haya en el grupo que se este solicitando o en su caso la cancelación al laboratorio para que el alumno pueda hacer algún movimiento, impidiendo en estas opciones el ingreso de números de cuenta inválidos.

Requisitos de operación:

- Cargar la información que se pida a SECAD o bien pretender hacer una petición directamente a DGAE (Dirección General de Administración Escolar) para que se envíe la información de los alumnos con derecho a inscripción a los laboratorios que pertenecen a este departamento.
- Cargar la información de los grupos, cupos, periodo actual, nombre del profesor de cada uno de los laboratorios que se van a impartir en el semestre en curso.
- Creación de las estructuras de datos para el almacenamiento de los grupos, los alumnos, y para el almacenamiento de la información a la inscripción a los laboratorios.

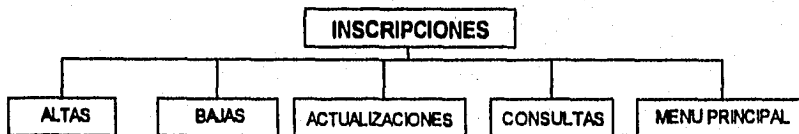


Diagrama General del Módulo de INSCRIPCIONES.

II.2.2. PERSONAL ACADÉMICO

En este módulo la información que se visualizará en la pantalla se obtendrá del archivo "PROFESOR.DAT", que permitirá al usuario agregar información de los profesores que no se encuentren en este archivo, así como la modificación de ésta para su actualización.

Requisitos de operación:

- Cargar la Información que se pida a ÚNICA o bien empezar a capturar la información para ser almacenada en la DB
- Generación de la estructura de datos para el almacenamiento de la información del personal académico

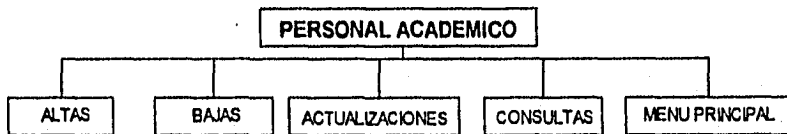


Diagrama General de PERSONAL ACADÉMICO,

II.2.3. ANÁLISIS ESTADÍSTICO

Módulo que se implementará para obtener un informe estadístico para su análisis, esta información se obtendrá del archivo "CALIFICA.DAT". Este informe consistirá en presentar la cantidad de alumnos que obtuvieron las diferentes calificaciones que se asientan en las actas para cada uno de los laboratorios para hacer un estudio minucioso y encontrar las fallas para mejorar el desempeño académico en los semestres subsecuentes. Por tanto, debe tenerse información histórica para que pueda hacerse un estudio comparativo, y lograr ver los avances que se hayan obtenido de semestres anteriores con el que acabe de concluir.

Requisitos de Operación.

- Carga de las calificaciones que los profesores asientan en las actas,
- Respaldo de la información de las calificaciones de semestres anteriores.
- Generación de las estructuras de los archivos que contendrán la información de las calificaciones por laboratorio.

II.2.4. IMPRESIÓN DE REPORTES.

En este módulo se podrán obtener los listados para verificación y validación de la información que se capture, permitiendo al usuario la corrección de esta información en el caso que así lo requiera. También nos permitirá expedir constancias de acreditación para aquellos alumnos que la soliciten. Requisitos de Operación:

- Se requiere la información generada por los tres puntos anteriores.
 - I. Para checar la información capturada
 - II. Para checar el Cupo de los grupos de los diferentes laboratorios.
 - III. El estudio de los índices de aprobación y reprobación de los alumnos que cursaron algunos de los diferentes laboratorios que imparte este departamento
 - IV. Para la expedición de constancias de acreditación del laboratorio.

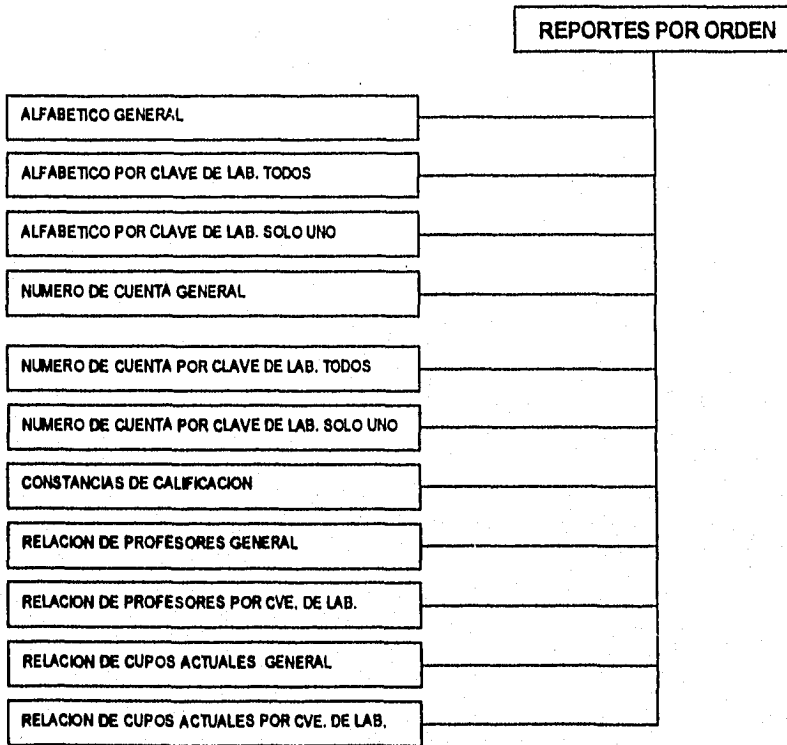


Diagrama General del Módulo de REPORTES.

II.2.5. BITÁCORA

En este módulo el usuario con la clave de acceso correcta tendrá derecho a realizar los respaldos de la información del semestre que concluya e inicializar los archivos para el nuevo semestre. También podrá monitorear la bitácora para saber todas y cada una de las actividades que se realizaron y en su caso conocer los errores que hayan cometido y saber por quien, a que hora, el día y en que actividad se realizó.

Requisito de operación:

- Clave de acceso a esta opción
- Estructuras de los archivos limpias para la inicialización del sistema
- Permiso para modificar la clave de acceso a las diferentes opciones que se generen en el sistema

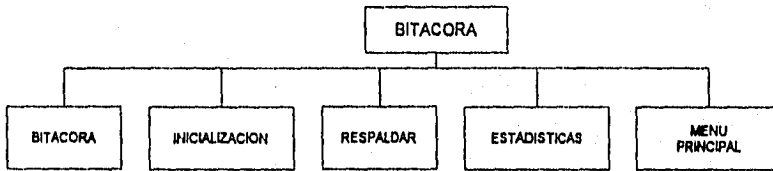


Diagrama General del módulo de BITÁCORA.

En este módulo existen dos opciones las cuales son INICIALIZACIÓN y RESPALDAR, opciones en las que se presentan los siguientes diagramas:

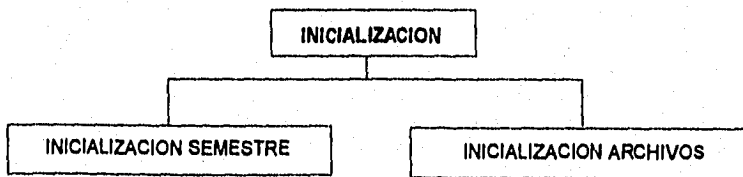


Diagrama General del módulo de INICIALIZACIÓN

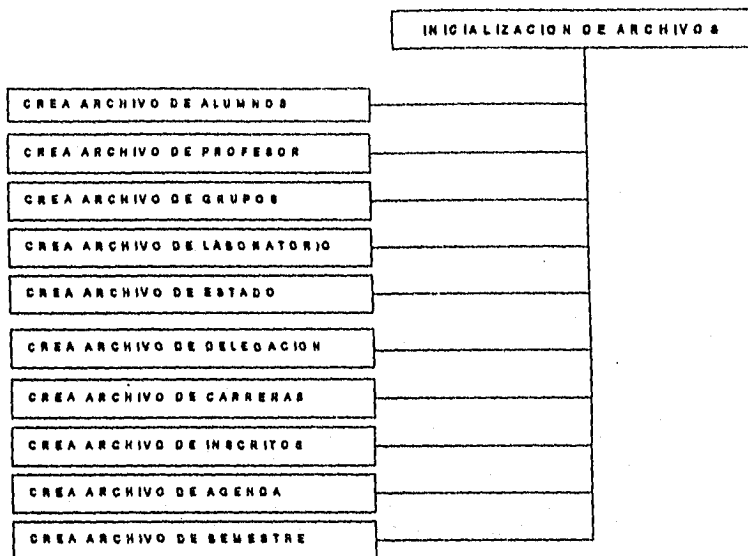


Diagrama General del modulo de INICIALIZACIÓN DE ARCHIVOS

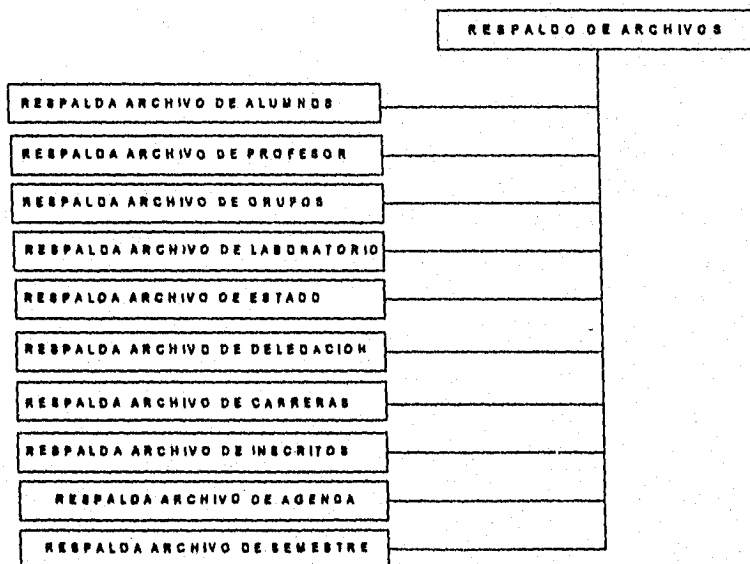


Diagrama General del módulo de RESPALDO.

Cabe mencionar que en cada uno de los tres primeros módulos se contará con las siguientes tres opciones:

- **Altas:** nos permitirá capturar todos los datos del alumno y/o del personal académico que no se encuentre en el archivo.
- **Bajas:** nos permitirá borrar la información de algún alumno y/o algún profesor que ya no tenga porque estar almacenada su información.
- **Actualizaciones:** nos permitirá hacer los cambios necesarios a la información que ya se tenga para que esta se encuentre actualizada.

Una vez definidos cada uno de los módulos del sistema de acuerdo a las actividades que realiza el usuario, los requisitos de operación de el sistema para su diseño son:

- Presentación de un menú de acceso a cada una de las opciones del sistema.
- Petición de una llave de acceso para limitar las operaciones del sistema a usuarios específicos.
- Presentación de un menú para cada proceso que especifique las diferentes opciones de éste.
- Manejo de menús en forma de cascada para visualizar la opción elegida y que este nos permita regresar al menú anterior.
- Fácil manejo para la impresión de reportes, para el caso en que se requiera cierta información específica o toda la información general, según las necesidades del usuario.
- Presentación de una barra de ayuda en la parte inferior de la pantalla.

Es necesario identificar las partes funcionales y exponer las formas para poder satisfacer los requerimientos del usuario, para lo cual se deben de incluir los siguientes elementos:

- Bases de datos,
- Control de procesos,
- Generador de estadísticas y reportes,
- Monitoreo de actividades.

II.3. DEFINICIÓN DE LAS ENTIDADES.

De acuerdo al modelo y a la descripción detallada de los módulos que el usuario generó se identificaron las siguientes entidades:

- ALUMNOS
- PROFESORES
- GRUPOS
- ASIGNATURAS
- ESTADOS
- DELEGACIÓN
- INSCRITOS
- CARRERAS

entidades que en base al modelo de bases de datos Btrieve se genera una estructura para cada una de las entidades mencionadas con las siguientes características:

ALUMNOS

Datos de alumnos del semestre en curso.

Nombre	Tipo	Longitud
CUENTA	CADENA	08
NOMBRE_ALUMNO	CADENA	60
CLAVE_CARRERA	CADENA	02

PROFESORES

Datos personales de los profesores que imparten laboratorios que pertenecen al Departamento de Física.

Nombre	Tipo	Longitud
RFC	CADENA	13
NOMBRE_PROFESOR	CADENA	60
CALLE	CADENA	50
COLONIA	CADENA	50
CLAVE_DEL	CADENA	02
CODIGO_POSTAL	CADENA	05
TELEFONO_CASA	CADENA	10
TELEFONO_OFICINA	CADENA	10
CIUDAD	CADENA	02
CLAVE_CARRERA	CADENA	04
ESPECIALIDAD	CADENA	50

GRUPOS

Grupos de los laboratorios que se imparten en el semestre en curso.

Nombre	Tipo	Longitud
CLAVE_LAB	CADENA	04
GRUPO	CADENA	04
CUPO	CADENA	02
INSCRITOS	CADENA	02
RFC	CADENA	13

ASIGNATURAS

Nombres de los diferentes laboratorios que se imparten en este Departamento.

Nombre	Tipo	Longitud
CLAVE_LAB	CADENA	04
NOMBRE_LAB	CADENA	50

ESTADOS

Nombres de los diferentes estados.

Nombre	Tipo	Longitud
CLAVE_ESTADO	CADENA	02
NOMBRE_ESTADO	CADENA	50

DELEGACION

Nombres de los diferentes municipios y delegaciones.

Nombre	Tipo	Longitud
CLAVE_DEL	CADENA	02
NOMBRE_DEL	CADENA	50

CARRERAS

Nombres de las diferentes carreras que tienen los profesores.

Nombre	Tipo	Longitud
CLAVE_CARRERA	CADENA	02
NOMBRE_CARRERA	CADENA	50

INSCRITOS

Laboratorio y grupo en donde un alumno se encuentra inscrito.

Nombre	Tipo	Longitud
CUENTA	CADENA	08
CLAVE_LAB	CADENA	04
GRUPO	CADENA	04
SEMESTRE	CADENA	04
CALIFICACION	CADENA	02

II.3.1. DEFINICIÓN DE LAS RELACIONES ENTRE ENTIDADES.

La estructura lógica general de una base de datos puede expresarse gráficamente por medio de *diagramas de entidad-relación*, consta de los siguientes componentes:

- **Rectángulos**, que representan conjuntos de entidades.
- **Elipses**, que representan los atributos de las entidades.
- **Rombos**, que representan las relaciones entre conjuntos de entidades.
- **Líneas**, que conectan los atributos a los conjuntos de entidades y los conjuntos de entidades a las relaciones.

Cada componente se etiqueta con el nombre correspondiente. El diagrama de E-R de las relaciones descritas en las tablas anteriores son:

ENTIDAD A	ENTIDAD B	RELACIÓN
INSCRITO	GRUPOS	M-M
	ALUMNO	1-1

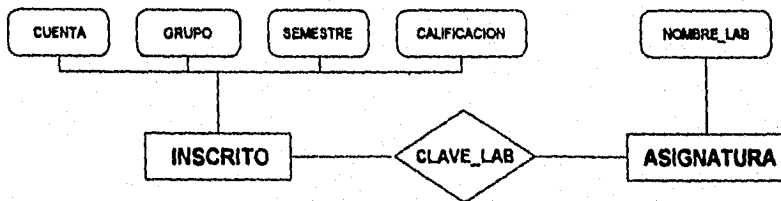


Diagrama de relación INSCRITO-ASIGNATURA

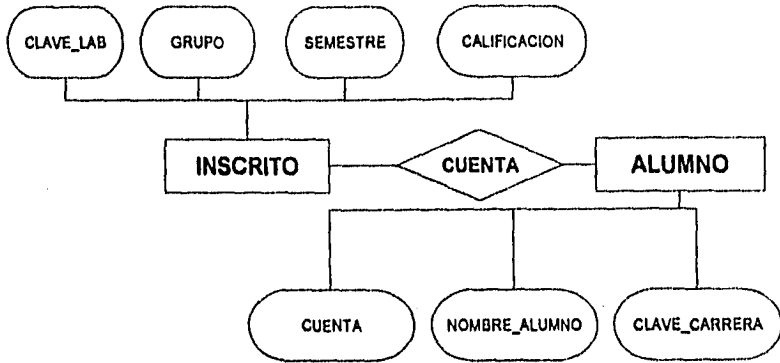


Diagrama de relación INSCRITO-ALUMNO

ENTIDAD A	ENTIDAD B	RELACIÓN
ASIGNATURA	GRUPOS	1-M

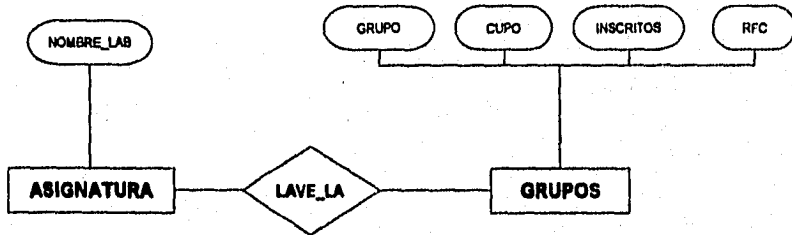


Diagrama de relación ASIGNATURA-GRUPOS

ENTIDAD A	ENTIDAD B	RELACIÓN
CARRERAS	PROFESOR	M-M
	ALUMNO	1-M

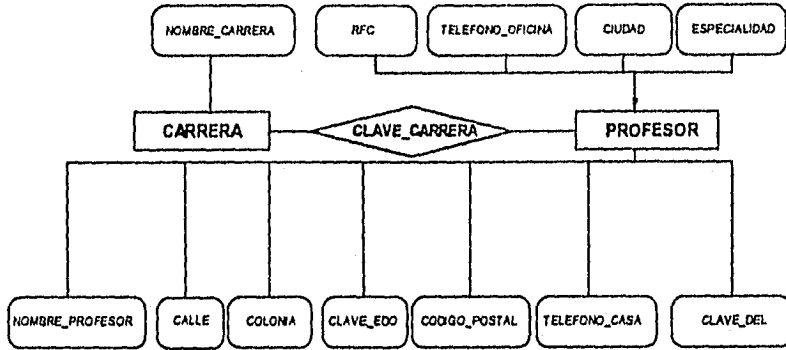


Diagrama de relación CARRERA-PROFESOR

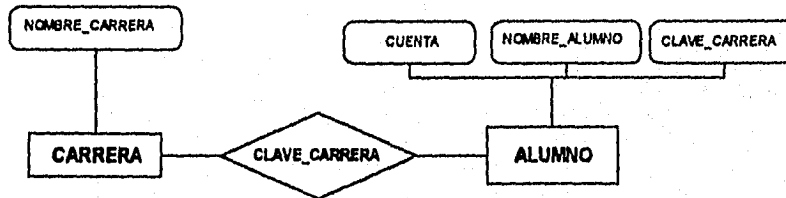


Diagrama de relación CARRERA-ALUMNO

ENTIDAD A	ENTIDAD B	RELACIÓN
DELEGACION	PROFESOR	1-M

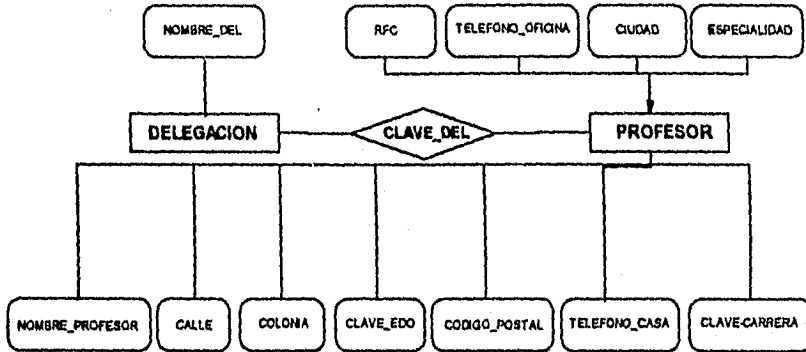


Diagrama de relación DELEGACION-PROFESOR

ENTIDAD A	ENTIDAD B	RELACIÓN
GRUPOS	INSCRITO	M-M
	PROFESOR	M-M
	ASIGNATURA	M-1

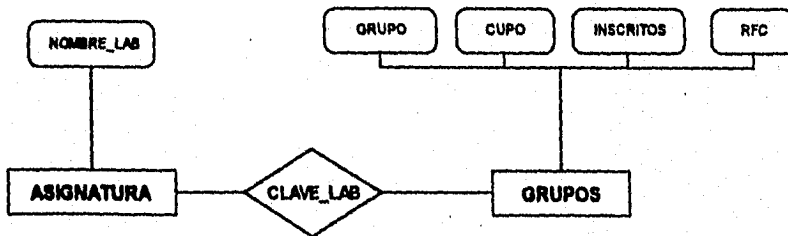


Diagrama de relación ASIGNATURA-GRUPOS

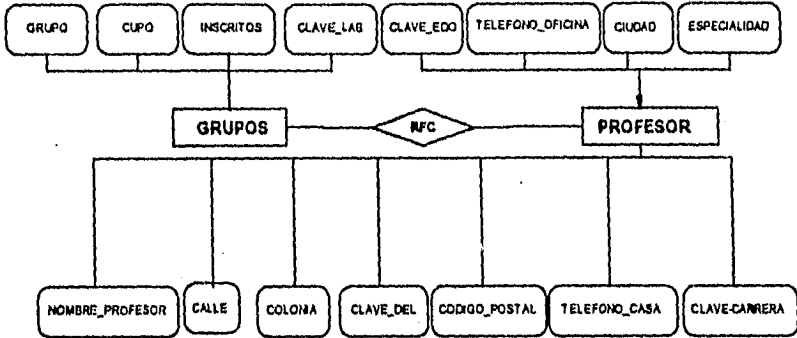


Diagrama de relación GRUPOS-PROFESOR

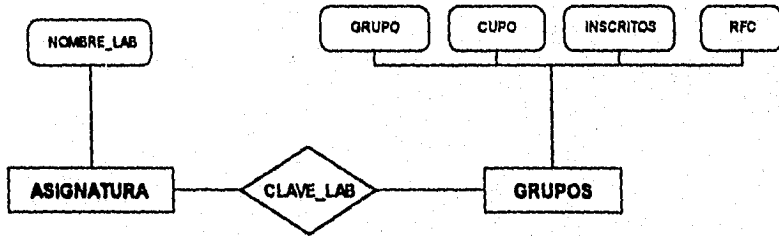


Diagrama de relación ASIGNATURA-GRUPOS

ENTIDAD A	ENTIDAD B	RELACIÓN
PROFESOR	GRUPOS	1-M

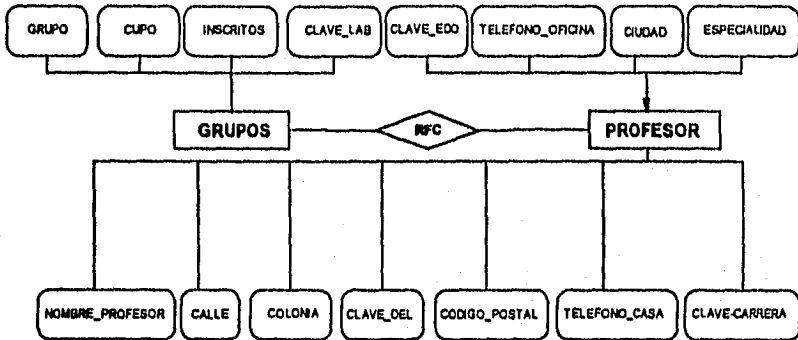


Diagrama de relación GRUPOS-PROFESOR

ENTIDAD A	ENTIDAD B	RELACIÓN
ESTADOS	PROFESOR	1-M

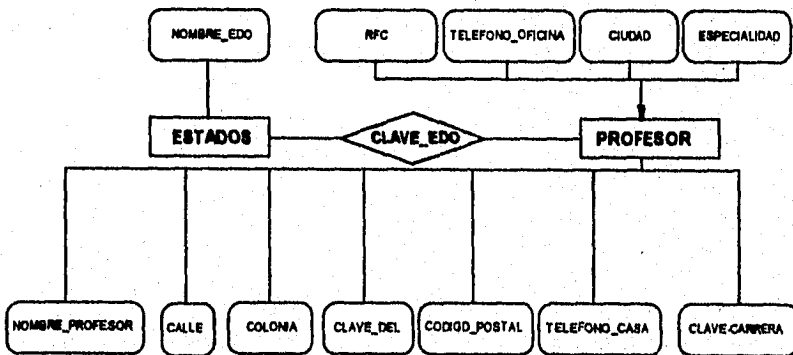


Diagrama de relación ESTADOS-PROFESOR

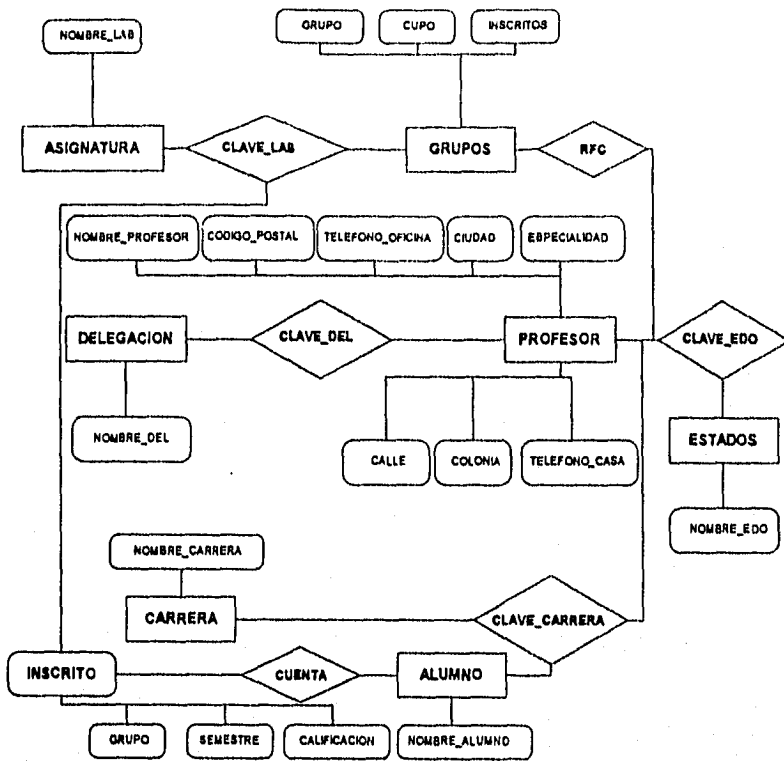


Diagrama de entidad-relación general.

II.3.2. DISEÑO DE MENÚS Y PANTALLAS DEL SISTEMA.

Con la información anterior se procede al diseño de menús y pantallas para posteriormente elaborar los programas que ejecutarán cada uno de los procesos planteados en el análisis anterior.

Las siguientes pantallas que se muestran pertenecen a los menús del sistema:

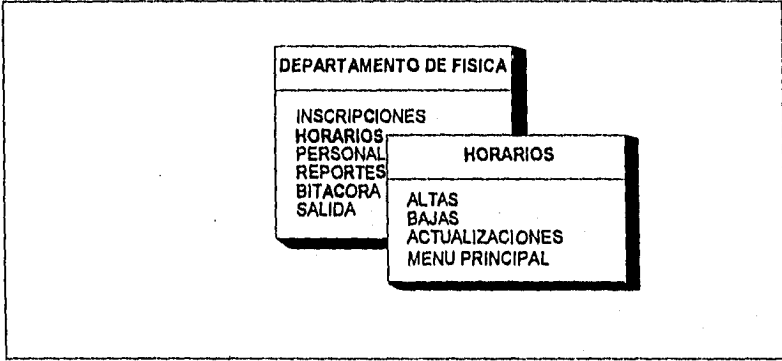
U.N.A.M.	DEPARTAMENTO DE FISICA	F.I.
LABORATORIOS		

DEPARTAMENTO DE FISICA	
INSCRIPCIONES	
HORARIOS	
PERSONAL ACADEMICO	
REPORTES	
BITACORA	
SALIDA	

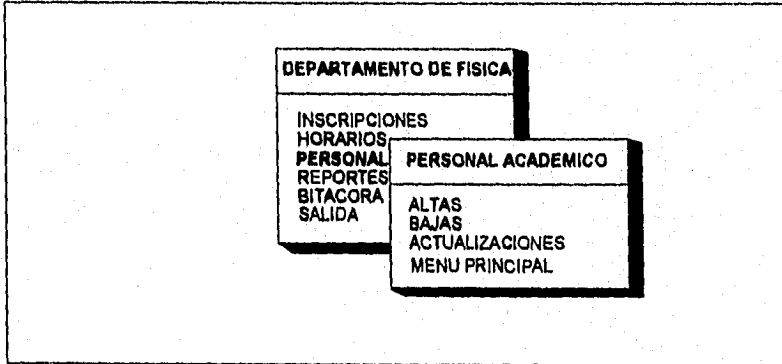
U.N.A.M.	DEPARTAMENTO DE FISICA	F.I.
LABORATORIOS		

DEPARTAMENTO DE FISICA	
INSCRIPCIONES	
HORARIOS	
PERSONAL	
REPORTES	
BITACORA	
SALIDA	
INSCRIPCIONES	
ALTAS	
BAJAS	
ACTUALIZACIONES	
MENU PRINCIPAL	

U.N.A.M. DEPARTAMENTO DE FISICA F.I.
LABORATORIOS



U.N.A.M. DEPARTAMENTO DE FISICA F.I.
LABORATORIOS



**U.N.A.M. DEPARTAMENTO DE FISICA
LABORATORIOS**

DEPARTAM	REPORTES
INSCRIPCI HORARIOS PERSONAL REPORTES BITACORA SALIDA	ALFABETICO GENERAL ALFABETICO POR CLAVE DE LAB. TODOS ALFABETICO POR CLAVE DE LAB. SOLO UNO NUMERO DE CUENTA GENERAL NUMERO DE CUENTA POR CLAVE DE LAB. SOLO UNO CONSTANCIA DE CREDITOS RELACION DE PROFESORES GENERAL RELACION DE PROFESORES POR CLAVE DE LAB. RELACION DE CUPOS ACTUALES GENERAL RELACION DE CUPOS ACTUALES POR CLAVE DE LAB. MENU PRINCIPAL

**U.N.A.M. DEPARTAMENTO DE FISICA F.I.
LABORATORIOS**

DEPARTAMENTO DE FISICA	
INSCRIPCIONES HORARIOS PERSONAL REPORTES BITACORA SALIDA	BITACORA BITACORA INICIALIZACION RESPALDAR ESTADISTICAS MENU PRINCIPAL

U.N.A.M.

DEPARTAMENTO DE FISICA
LABORATORIOS

INICIALIZACION

INICIALIZACION DE SEMESTRE
INICIALIZACION DE ARCHIVOS

PERSONA
REPORTE
BITACOR
SALIDA

BITACORA

BITACORA
INICIALIZACION
RESPALDAR
ESTADISTICAS
MENU PRINCIPAL

U.N.A.M.

DEPARTAMENTO DE FISICA
LABORATORIOS

INICIALIZACION

INICIALIZACION DE SEMESTRE
INICIALIZACION DE ARCHIVOS

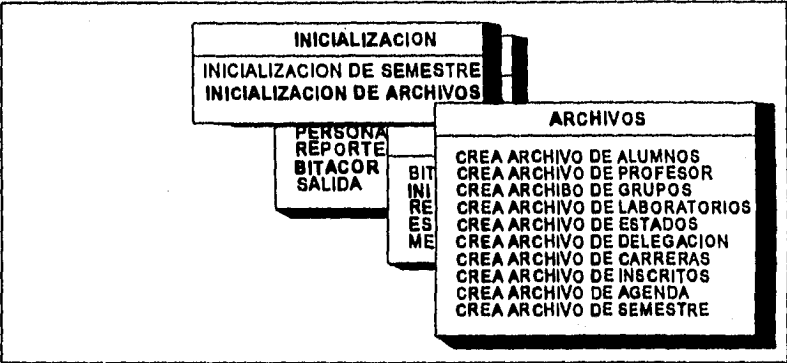
PERSONA
REPORTE
BITACOR
SALIDA

BITA

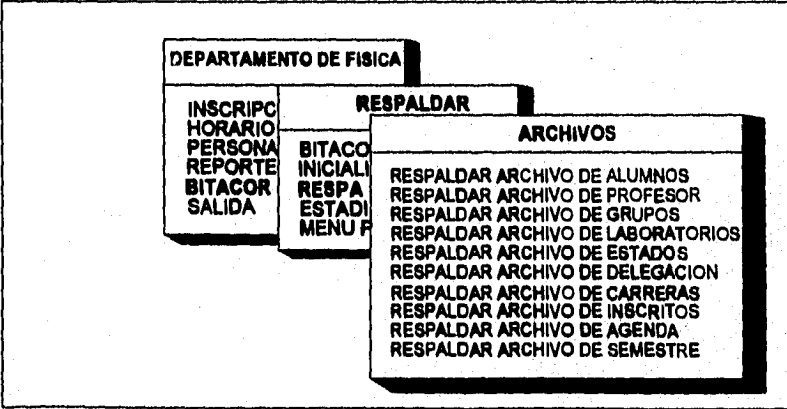
SEMESTRE:

BITACORA
INICIALIZACION
RESPALDAR
ESTADISTICAS
MENU PRINCIPAL

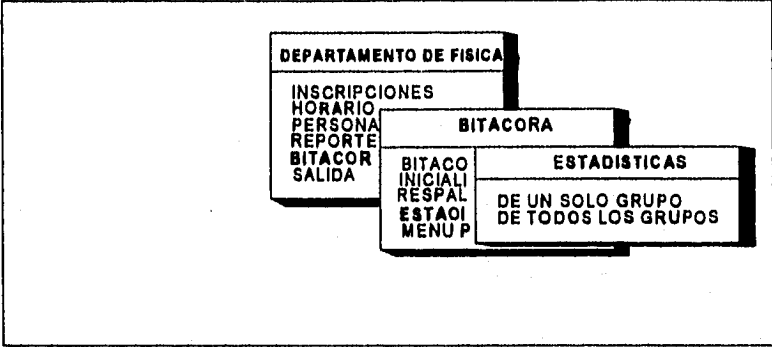
U.N.A.M. DEPARTAMENTO DE FISICA LABORATORIOS



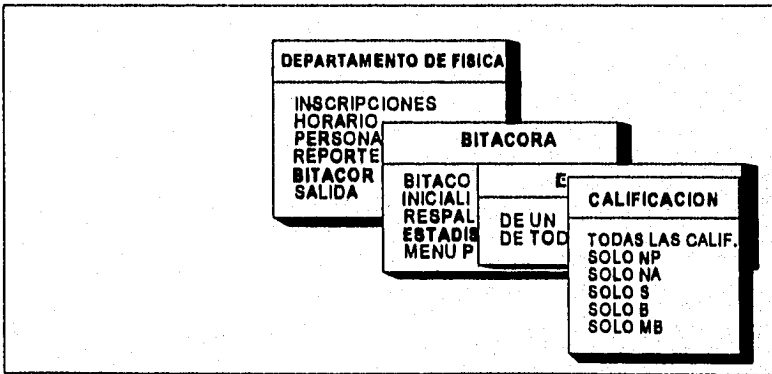
U.N.A.M. DEPARTAMENTO DE FISICA LABORATORIOS



**U.N.A.M. DEPARTAMENTO DE FISICA
LABORATORIOS**



**U.N.A.M. DEPARTAMENTO DE FISICA
LABORATORIOS**



A continuación se muestran las pantallas de captura de datos:

U.N.A.M.	DEPARTAMENTO DE FISICA	F.I.
LABORATORIOS		

No. DE CUENTA:	<input type="text"/>	
NOMBRE:	<input type="text"/>	
CLAVE DEL LAB.:	<input type="text"/>	<input type="text"/>
GRUPO DEL LAB.:	<input type="text"/>	
LABORATORIOS A LOS QUE ESTA INSCRITO:		
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

Esta pantalla pertenece a la captura de inscripciones, en la cual se inscribirá, se actualizará, se dará de baja alguna materia o se consultará la inscripción del alumno.

U.N.A.M.	DEPARTAMENTO DE FISICA	F.I.
LABORATORIOS		

NOMBRE PROF.:	<input type="text"/>				
RFC:	<input type="text"/>	CARRERA:	<input type="text"/>		
		ESPECIALIDAD:	<input type="text"/>		
CALLE:	<input type="text"/>				
COLONA:	<input type="text"/>				
DELEGACION:	<input type="text"/>	C.P.:	<input type="text"/>	TELEFONO:	<input type="text"/>

En esta pantalla se presentan los campos a capturar, modificar o consultar de los datos del profesor solicitado.

U.N.A.M.

DEPARTAMENTO DE FISICA

F.I.

LABORATORIOS

CLAVE MATERIA:

NOMBRE MATERIA:

GRUPO MATERIA:

RFC DEL PROFESOR:

CUPO MATERIA:

INSCRITOS:

En esta pantalla se presentan los datos a capturar, actualizar, modificar o dar de baja de los grupos que se imparten en el semestre en curso.

CAPITULO III

DESARROLLO DEL SISTEMA

III. DESARROLLO DEL SISTEMA

	Página
III.1. Desarrollo del Sistema.	57
III.2. Objetivos para el desarrollo del Sistema.	58
III.3. Codificación del Sistema	58
III.3.1. Codificación del programa principal del sistema.	59
III.3.2. Codificación del modulo de inscripciones	61

III.2 OBJETIVOS PARA EL DESARROLLO DEL SISTEMA.

Los objetivos que se tienen en la fase de desarrollo son:

- Codificar los módulos,
- Verificar el correcto funcionamiento de los cada uno de los módulos.
- Integración de los módulos para formar los programas.
- Integración de los programas para formar el Sistema.

Esta etapa queda concluida cuando el usuario acepta el sistema, un vez que se hayan tomado en cuenta las modificaciones y discrepancias generadas durante el proceso de prueba del sistema.

III.3 CODIFICACIÓN DEL SISTEMA

Como se vio en el capítulo anterior, de los procesos, que se llevan a cabo en el Departamento de Física, la mayoría de estos están relacionadas entre sí y se realizan de manera manual. Como se cuenta con un diseño modular, esto se traduce en instrucciones de máquina (Computadora), para lo cual se utilizará el lenguaje PASCAL, utilizando la programación estructurada que nos permitirá escribir los programas más claramente y legibles, y por lo tanto con el menor número de errores. La claridad de un programa depende en gran medida del estilo del programador y la mejor documentación de un programa de computadora la constituyen: la claridad de su estructura y el código mismo.

El siguiente ejemplo nos ilustra la codificación del programa "TESIS.PAS". Los módulos del programa se codifican a partir del detallado de los módulos descritos en el capítulo anterior.

III.3.1. CODIFICACIÓN DEL PROGRAMA PRINCIPAL DEL SISTEMA.

Este es el programa principal, que ejecuta los módulos seleccionados en cada una de las opciones que se presentan en los menús desplegables.

```
{
Programa fuente      : SIDF.PAS
Programa ejecutable: SIDF.EXE
Propósito           : Obtener la información adecuada para el acceso a cada una
                    : de las opciones.
Nombre y Fecha de implantación:
                    : Concepción Alcántara.      Mayo 1996
Nombre y Fecha de revisiones
                    :
}
PROGRAM Departamento_Física;

($A+,+B-,D+.E+,F-,1+,L+,N+,O-,R-,S+,V-) ($M $8192,0,0) ($M $4000,0,0)
($M 65000,0,65360)
USES
RUTINAS, BASES, BTRIEVE, DOS, CRT, PRINTER;
VAR
laboratorio, opcion, reporte, bitacora: Interger;
st0, st1, st3, st4: char;
password, Pass, Passs1, Passs2, Passs3: String;
Begin
Carga Pantalla('scr000.scr'.0);
Clrscr,
LeePassword(password);
Repeat
Corta_Pantalla(1,1,1,80,25,1,1);
Lee Menu Laboratorios(Laboratorio, St0);
If (laboratorio <> 6) AND (st0 <> Esc) Then
Begin
Push_Screen;
Form(Laboratorio, 'Z', Passs1);
Repeat
LeeMenuOpcion(Opcion, Laboratorio, st1);
If (opcion <> 6) AND (st1 <> Esc) Then
Begin
Push_Screen;
Form(Opcion, 'Z', Passs2);
Pass := '';
Pass := Password + Passs1 + Passs2;
CASE opcion OF
```

```

1,2,3: Begin
    Push_Screen;
    CASE opcion OF
        1: exec('ins2.exe',Pass);
        2: exec('Horarios.ece',Pass);
        3: exec('Aca2.exe',Pass);
    End;
    Pop_Screen;
End;
4: exec('Rep2.exe',Pass);
5: Begin
    Push_Screen;
    Repeat
        LeeMenuBitacora(Bitacora,Opcion, St4);
        If (bitacora<>6)AND(st4<>Esc) Then
            Begin
                Form(Bitacora,'Z',Passs3);
                Pass:="";
                Pass:=Password+Passs1+Passs2+Passs3;
                CASE Bitacora OF
                    1: exec('bita2.exe',Pass);
                    2: exec('iniciali.exe',Pass);
                    3: exec('respaldo.exe',Pass);
                    4: exec('Estadist.exe',Pass);
                end,
                Until (bitacora=2)OR(st4=Esc);
            Pop_Screen;
        End;
    End;
    Pop_Screen;
    Until (opcion=6)OR(st1=Esc);
    End;
    Until (laboratorio=6)OR(st0=Esc)
    clrscr;
End.

```


III.3.2. CODIFICACIÓN DEL MODULO DE INSCRIPCIONES.

El siguiente programa analiza la situación del alumno que se este procesando para indicarle al usuario si procede su inscripción, baja o actualización al laboratorio que el alumno necesite inscribirse.

```
{
Programa fuente      : INSCRITO.PAS
Programa ejecutable: INSCRITO.EXE
Propósito           : Obtener la información del alumno con el fin de determinar si
                    : procede la opción elegida por el usuario en caso de que sea
                    : inválida su elección el programa le desplegara un mensaje
                    : de error.
Nombre y Fecha de implantación:
                    : Concepción Alcántara.      Mayo 1996
Nombre y Fecha de revisiones
                    :
}
```

```
Program Incripciones_A_Los_Laboratorios;
Uses
  Rutinas,Bases,Btrieve,Dos,Crt;
Const
  MaximoArreglo=5;
Type
  arreglo=array[1..5]of string[75];
Var
  ActBaj,actividad,i: Integer;
  St2: Char;
  GrupoTemp,ClaveTemp,RespAlum,Llave,Nombre: String;
  Eligio,Vector: Arreglo;
Procedure CapturaAlumno(Var RegAlumno:Alumnos; Var
  codecapAlum:byte;NumCta:string);
const
  MovAlum:Array[1..4,1..4]of byte=((1,1,2,2),(1,2,3,3),(1,2,3,3),(4,4,4,4));
var
  op,Clave_Carreraint,AgnoIngreso,AgnoEnCurso:integer;
  Carrera,Clave_Carrerastr,agno,fecha,anioactual:string;
  anioingvalido:boolean;
Begin
  Corta_Pantalla(2,1,8,80,25,1,8);
  LimpiaAlumnos(RegAlumno);
  RegAlumno.nocta:=NumCta;
  Escribe(21,10,RegAlumno.NoCta,#0);
  resp:='N';
```

```

Carrera:= ' ';
CodecapAlum:=0;
op:=1;
st:=#0;
repeat
  With RegAlumno do
    Begin
      Case op of
        1 : Begin
          Repeat
            Nombre_Alumno:=Valida(21,12,0,1,1,Nombre_Alumno,['A'..'Z','&',' '],st);
            Nombre_Alumno:=justifica(compacta(Nombre_Alumno));
            if (Nombre_Alumno="" and(st<>#27)) then
              desp_error('ERROR, FALTA ESCRIBIR EL NOMBRE DEL ALUMNO');
            Until (Nombre_Alumno<>"") or (st=#27);
            If st<>#27 then
              Begin
                Nombre_Alumno:=completa(Nombre_Alumno,32);
                escribe(21,12,Nombre_Alumno,#0);
              end;
            end;
          2 : Begin
            Repeat
              Clave_Carrera:=Valida(21,14,0,1,1,Clave_Carrera,['0'..'9'],st);
              Clave_Carrera:=justifica(compacta(Clave_Carrera));
              if (Clave_Carrera="" and(st<>#27)) then
                Begin
                  desp_error('ERROR, FALTA ESCRIBIR LA CARRERA DEL'+
                    ' ALUMNO');
                escribe(21,14,' ',#0)
                end
              else
                val(Clave_Carrera,Clave_Carreraint,code);
              Until (Clave_Carreraint In [21..29,31..33]) or(st=#27);
            end;
          3 : Begin
            repeat
              AnioIngreso:=Valida(21,16,0,1,1,anioIngreso,['0'..'9'],st);
              AnioIngreso:=completa(AnioIngreso,2);
              agno:="";
              agno:=AnioIngreso;
              val(agno,AgnoIngreso,code);
              Anioingvalido:=false;
              fecha:="";
              fecha:=fecha_sistema;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

anioactual:=copy(fecha,5,2); {6,2}
val(anioactual,AgnoEnCurso,code);
AgnoEnCurso:=AgnoEnCurso+1;
if (AgnoIngreso>50)and(st<>#27)and(AgnoEnCurso>AgnoIngreso) then
  Begin
  Anioingvalido:=true;
  end
else
  Begin
  If st<>#27 then
    desp_error3('Año de ingreso inválido');
    Anioingvalido:=false;
    AnioIngreso:="";
    end;
  until (st=#27)or(Anioingvalido=true);
  end;
4 : Begin
  push_screen;
  resp:= 'N';
  desp_error3(" Están correctos los dalos <S/N> ?");
  validarespuesta(67,25,resp,st,inv_att);
  if (st<>#27)and(resp='S') then
    Begin
    temp:=RegAlumno.NoCta;
    lral:=sizeof(Alumnos);
    StatusAl:=Btrv(2,ArchAlumnos,RegAlumno,lral,temp,0);
    If StatusAl=0 then
      codecapalum:=1;
    end
  else
    if (st<>#27)and(resp='N') then
      Begin
      op:=0;
      st:=#0;
      end;
    pop_screen;
    END;
  End;
Case St of
  #00 : Inc (op);
  #01 : op := 1;
  #02 : op := 3;
  #04 : op := MovAlum [op,1];
  #05 : op := MovAlum [op,2];
  #06 : op := MovAlum [op,3];

```

```

    #07 : op := MovAlum [op,4];
  End;
end;
until (op>4)or(st=#27)or(respt='S');
end;

```

```

Procedure DesplegaInscritos(RegInscritos:Inscritos;Var vector:arreglo;
                          Var i:integer; actividad,ActBaj:Integer);

```

```

Var
  Key: String;
  ren: Integer;
Begin
  With RegInscritos DO
  Begin
    Escribe(21,10,cuenta,#0);
    Escribe(21,12,nombre_alumno,#0);
    llave:="";
    llave:=Cuenta;
    Escribe(21,14,clave_lab,#0);
    nombre:=SPACE(32);
    Nombre:=TraeNombreLaboratorio(clave_lab,status);
    Escribe(26,14,nombre,#0);
    Escribe(21,16,grupo,#0);
    key:="";
    Key:=llave;
    Lr := sizeof(RegInscritos1);
    StatusIn:=BTRV(5,ArchInscritos,RegInscritos1,Lr,llave,1);
    ren:=20;
    i:=0;
    If StatusIn=0 then
      Begin
        while (StatusIn=0)and(Key=RegInscritos.Cuenta) do
          Begin
            inc(i);
            nombre:=SPACE(32);
            Nombre:=TraeNombreLaboratorio(RegInscritos1.clave_lab,status);
            Escribe(10,ren,RegInscritos1.clave_lab,#0);
            Escribe(20,ren,nombre,#0);
            Escribe(15,ren,RegInscritos1.grupo,#0);
            If ((Actividad=3)or(Actividad=2)) And (Eligio[ActBaj] = RegInscritos.Cuenta +
              RegInscritos1.clave_lab) then
              begin
                Escribe(21,14,RegInscritos1.clave_lab,#0);
                Escribe(26,14,Nombre,#0);
                Escribe(21,16,RegInscritos1.grupo,#0);
              end
            end
          end
        end
      end
    end
  end

```

```

Escribe(10,ren,RegInscritos1.clave_lab,#126);
Escribe(20,ren,nombre,#126);
Escribe(15,ren,RegInscritos1.grupo,#126);
End;
if compacta(justifica(RegInscritos1.clave_lab))<>" then
  Begin
    vector[i]:=' '+RegInscritos1.clave_lab+' | '+RegInscritos1.grupo+' |
              '+nombre;
    ellgio[i]:=RegInscritos1.Cuenta+RegInscritos1.clave_lab;
    End;
  Lr := sizeof(RegInscritos1);
  StatusIn:=BTRV(6,ArchInscritos,RegInscritos1,Lr,Key,1);
  inc(ren);
  End;
End;
End;
End;

Procedure CapturaInscripcion(Var RegInscritos:Inscritos;actividad,ActBaj:Integer);
Const
  Moves:Array[1..3,1..4]of byte=((1,1,1,2),(1,2,2,3),(2,3,3,3));
Var
  op: byte;
  numerocve: longint;
  llave,llave1,NumeroStr:string;
Begin
  st:=#0;
  op:=1;
  llave1:= "";
  Llave1:=RegInscritos.Cuenta+RegInscritos.clave_lab;
  DespliegaInscritos(RegInscritos,vector,i,actividad,ActBaj);
  With RegInscritos DO
  Repeat
  CASE op OF
  1 : Begin
    Repeat
    ClaveTemp:= "";
    ClaveTemp:=Clave_lab;
    ClaveTemp:=Valida(21,14,0,1,1,ClaveTemp,['0'..'9',#27],st);
    Until (st=#27)OR(ClaveTemp<>"");
    IF (st<>#27)AND(ClaveTemp<>"") THEN
    Begin
    VAL(ClaveTemp,numerocve,code);
    form(numerocve,'ZZZ',NumeroStr);
    IF (numerocve<>0) THEN

```

```

Begin
nombre:=SPACE(32);
Nombre:=TraeNombreLaboratorio(NumeroStr,status);
If Status=0 then
  Begin
  Escribe(26,14,nombre,#0);
  Escribe(21,14,NumeroStr,#0);
  Clave_lab:=NumeroStr;
  llave:="";
  llave:=Cuenta+clave_lab;
  BuscaEnInscritos(llave,Status);
  IF Status=0 THEN
    Begin
    Desp_Error('No se puede volver a inscribir al mismo laboratorio');
    st:=#00;
    op:=0;
    End;
  End
ELSE
  Begin
  Desp_error('El Grupo no se encuentra en el Archivo');
  op:=0;
  st:=#0;
  End;
End
ELSE
  Begin
  Desp_error('Grupo Inválido');
  op:=0;
  st:=#0;
  End;
End;
End;
2: Begin
Repeat
  GrupoTemp:="";
  GrupoTemp:=Grupo;
  GrupoTemp := Valida (21,16,0,1,1,GrupoTemp,['0'..'9',#27],st);
Until (st=#27)OR(GrupoTemp<>"");
IF (st<>#27)AND(GrupoTemp<>"") THEN
  Begin
  VAL(GrupoTemp,numerocve,code);
  form(numeroCve,'ZZZZ',NumeroStr);
  IF (numerocve<>0) THEN
    Begin

```

```

llave:="";
llave:=NumeroStr+Clave_Lab;
TraeGrupoClave(llave,status);
If Status=0 then
  Begin
    grupo:=NumeroStr;
    escribe(21,16,Grupo,#0);
  End
ELSE
  Begin
    Desp_error('El Grupo no se encuentra en el Archivo');
    op:=1;
    st:=#0;
    End;
  End
End
ELSE
  Begin
    If st<>#27 then
      Begin
        op:=1;
        st:=#0;
      End;
    End;
  End;
3: Begin
  PUSH_SCREEN;
  resp:="";
  desp_error3(" Están correctos los datos <S/N> ?");
  validarespuesta(67,25,resp,st,inv_att);
  IF (st<>#27)AND(resp='S') THEN
    Begin
      llave:="";
      llave:=Cuenta+clave_lab;
      RegInscritos.cuentaclave_lab:=llave;
      CASE actividad OF
        1: Begin (Alta de registro)
          LrIn:=sizeof(RegInscritos);
          StatusIn:=BTRV_msg(2,ArchInscritos,RegInscritos,LrIn,llave,0);
          End;
        3: Begin (Actualización del registro)
          LrIn:=sizeof(RegInscritos);
          StatusIn:=BTRV_msg(5,ArchInscritos,RegInscritos1,LrIn,llave1,0);
          If StatusIn=0 then
            begin

```

```

    Lrn:=sizeof(RegInscritos);
    StatusIn:=BTRV_msg(4,ArchInscritos,RegInscritos1,Lrn,Ilave1,0);
    If StatusIn=0 then
        begin
            Lrn:=sizeof(RegInscritos);
            StatusIn:=BTRV_msg(2,ArchInscritos,RegInscritos,Lrn,Ilave,0);
        End;
    end;
End;
End;
End
ELSE
    IF (st<>#27)AND(respt='N') THEN
        Begin
            op:=0;
            st:=#0;
            End;
        POP_SCREEN;
        End;
End;

Case St of
    #00 : Inc (op);
    #01 : op := 1;
    #02 : op := 3;
    #04 : op := moves [op,1];
    #05 : op := moves [op,2];
    #06 : op := moves [op,3];
    #07 : op := moves [op,4];
End; (Case)
Until (op > 3)or(st=#27);
End;

Begin
Carga_Pantalla('scr111.scr',0);
Carga_Pantalla('scr1u.scr',0);
passx:=paramstr(1);
if passx<>' then
    Begin
        RegresaNumEncabezado(Passx,ActMenu,cod1);
        Pop_Screen;
        Repeat
            LeeMenuActividad(Actividad,ActMenu,St2);
            Form(Cod1+(Actividad*10),'ZZZZ',Cod2);
            Form(ActMenu,'ZZ',Cod3);
        Until (st=#27);
    End;
End;

```



```

IF (actividad<>6)AND(st2<>#27) THEN
Begin
Push_Screen;
LlaveAgenda:= "";
E_S_Agenda:=agenda(Cod2,Cod3,Passx,'E',LlaveAgenda);
if E_S_Agenda=true then
begin
Repeat
Corta_Pantalla(1,1,1,80,25,1,1); {2}
NumCta:=SPACE(08);
ValidaNumeroCta(21,10,NumCta,8,st);
IF (st<>#15)AND(st<>#27)AND(LENGTH(NumCta)=8) THEN
Begin
limpiaAlumnos(regalumnos);
limpiaInscritos(RegInscritos);
limpiaInscritos(RegInscritos1);
limpiaInscritos(RegInscritos2);
For i:=1 to MaximoArreglo do
Begin
vector[i]:= "";
eligio[i]:= "";
End;
i:=0;
AbreInscritos(StatusIn);
IF StatusIn=0 THEN
Begin
AbreAlumnos(StatusAl);
If StatusAl=0 then
Begin
CASE actividad of
1: Begin {ALTAS}
RegInscritos.cuenta:=NumCta;
Have:= "";
Nombre:=TraeNombreAlumno(NumCta,StatusAl);
If StatusAl<>0 then
begin
push_screen;
RespAlum:='N';
desp_error3(" Se Desea capturar datos del alumno <S/N> ?");
validarespuesta(63,25,RespAlum,st,inv_att);
pop_screen;
codecapalum:=0;
If RespAlum='S' then
Begin
Push_Screen;

```

```

CapturaAlumno(RegAlumnos,Codecapalum,NumCta);
Pop_Screen;
End;
If codecapalum=1 then
Begin
RegInscritos.Nombre_Alumno:=Copy(RegAlumnos.Nombre_alumno,1,32);
CapturaInscripcion(RegInscritos,actividad,ActBaj);
End;
end
else
Begin
RegInscritos.Nombre_Alumno:=Copy(Nombre,1,32);
CapturaInscripcion(RegInscritos,actividad,ActBaj);
End;
End;
2,3: Begin
RegInscritos.cuenta:=NumCta;
DespliegaInscritos(reginscritos,vector,i,actividad,ActBaj);
If i<>0 then
Begin
EligeTitulo(Actividad,Titulo);
Repeat
ActBaj:=1;
ActBaj:=Barra2(18,13,i,i,75+1,1,1,ActBaj,@Vector,'ELIGE EL '+
'LABORATORIO A DAR DE BAJA',st2);
Until (ActBaj IN [1..i]);
If (ActBaj in [1..i])and(st2<>#27) then
Begin
llave:="";
Llave:=Eligio[ActBaj];
Case Actividad of
2: Begin {BAJAS}
DespliegaInscritos(reginscritos,vector,i,actividad,ActBaj);
BajaInscripcion(llave);
End;
3: Begin {ACTUALIZACION}
Lr:=sizeof(RegInscritos1);
StatusIn:=BTRV(5,ArchInscritos,RegInscritos,Lr,llave,0);
CapturaInscripcion(RegInscritos,actividad,ActBaj);
End;
End;
End
else
If (i<>0)and(st2<>#27) then
Desp_Error('Este alumno no tiene ningún registro para '+
'proceder la baja');

```

```
End;
End;
4: Begin {CONSULTAS}
GOTOXY(57,25);
RegInscritos.cuenta:=NumCta;
DespliegaInscritos(reginscritos,vector,i,actividad,ActBaj);
Consultas;
End;
End;
CierraAlumnos(StatusAl);
End;
CierraInscritos(StatusIn);
End
ELSE
status:=-1;
End;
Until ST=#27;
E_S_Agenda:=agenda(Cod2,Cod3,Passx,'S',LlaveAgenda);
End;
End;
Pop_Screen;
Until (actividad=6)OR(st2=#27);
End;
End.
```

CAPITULO IV

PRUEBAS DE VERIFICACIÓN Y VALIDACIÓN DEL SISTEMA

IV. PRUEBAS DE VERIFICACIÓN Y VALIDACIÓN DEL SISTEMA

	Página
IV.1. Pruebas de Verificación y Validación del Sistema	74
IV.2. Pruebas de Unidad	75
IV.3. Pruebas de Integración	76
IV.4. Pruebas de Aceptación	77
IV.5 Creación de casos de prueba.	78
IV.6 Arreo de prueba.	78
IV.7 Prueba de módulos	79
IV.8 Casos de prueba	81
IV.9 Pruebas de integración y aceptación	84

CAPITULO IV

IV.1 PRUEBAS DE VERIFICACIÓN Y VALIDACIÓN DEL SISTEMA

Los planes de prueba son importantes, y a menudo se pasan por alto, esta clase de pruebas prescriben varias clases de actividades que se efectúan con el objetivo de que el sistema cumple con los requisitos pedidos. El objetivo de estas pruebas es lograr una operación libre de errores bajo las condiciones que se especificaron durante la realización de este proyecto. así como el control de calidad del sistema realizado, en la cual se toman en cuenta la validación y verificación del sistema.

Las pruebas implican dos clases de actividades: pruebas de integración y las pruebas de aceptación. Antes de realizar la pruebas de integración se deben realizar las pruebas de unidad.

- Las *pruebas de Unidad* consiste en que el programador individual realiza pruebas a cada una de las unidades de programa, las cuales comprenden: pruebas funcionales, pruebas de desempeño, pruebas de tensión y pruebas de estructura, para detectar los errores de codificación, es decir, las fallas no detectadas en la etapa de análisis lo que impide satisfacer las necesidades del usuario. Hay ocasiones en las que también se tienen fallas de especificación, causadas por especificaciones incompletas o imprecisas dando interpretación equivocada de las actividades a programar. Estos errores pueden ser eliminados revisando cada una de los módulos en la etapa de desarrollo del sistema.
- Las *pruebas de integración* dicta el orden en que los módulos deben estar disponibles, ejerciendo gran influencia en el orden en el que se escriben, depuran y se hacen pruebas de unidad a cada uno de los módulos programados.
- Las *pruebas de aceptación* deben tomarse en cuenta la planeación y la ejecución de pruebas funcionales de desempeño para verificar que el sistema que se ha realizado satisfaga las necesidades del usuario final.

IV.2 PRUEBAS DE UNIDAD

Los aspectos principales de las pruebas de unidad son el desarrollo de casos de pruebas efectivos y los criterios de determinación contra los cuales se conducirán las pruebas.

Las pruebas de función y desempeño se desarrollan durante la fase de diseño, basándose en las especificaciones de los requisitos. Las pruebas de tensión se proyectan de manera intencional a sobrecargar o romper el sistema. Las pruebas de estructura se relacionan con el examen de la estructura lógica del código fuente. Las pruebas de unidad son la principal oportunidad para probar cada una de las unidades ya que una vez están integradas se vuelve más complejo el detalle de las pruebas que se les realicen.

Hay cuatro categorías de prueba, que debería de efectuar un programador a una unidad de programa; éstas son:

- Pruebas funcionales
- Pruebas de desempeño
- Pruebas de tensión
- Pruebas de estructura

Las *pruebas funcionales* y de las *pruebas de desempeño* se basan en las especificaciones de requisitos, se diseñan para demostrar que el sistema satisface los requisitos del usuario. Esto implica que los resultados que se obtengan de estas pruebas deben de ser tan buenos como los requisitos que se proporcionaron para el desarrollo de este sistema, además deben de especificar la configuración de la máquina, las suposiciones concernientes al estado del sistema para el caso de la prueba.

Las *pruebas funcionales* especifican las condiciones operativas normales, los valores comunes de entrada y los resultados de salida normales esperados.

Las *pruebas de tensión* son aquellas diseñadas para romper, en forma intencional la unidad en prueba, sobrecargándolo de varias maneras, con el propósito de determinar sus limitaciones y en el momento en el que falle determinar la manera en que se manifiesta la falla.

Las *pruebas de desempeño* deben diseñarse para verificar el tiempo de respuesta, tiempo de ejecución, capacidad de procesamiento, utilización de memoria primaria y secundaria.

Las *pruebas estructurales* están relacionadas con el examen de la lógica del sistema, las rutinas particulares que invoca la unidad en prueba, con el objetivo de recorrer un número especificados de caminos a través de cada una de las rutinas que la unidad en examen llama estableciendo así la profundidad de la prueba, esta prueba no puede ser diseñada hasta el momento en el que sistema ha sido instrumentado

Las pruebas funcionales, de desempeño y de tensión se denominan pruebas de caja negra, mientras que las de estructura se denominan pruebas de caja blanca o de cristal.

IV.3 PRUEBAS DE INTEGRACIÓN

El objetivo esencial de esta prueba consiste en integrar los componentes de un sistema de programación en una unidad funcional lo que requiere de una planeación cuidadosa de modo en que se disponga de los módulos cuando estos se necesiten y el diseño de casos de prueba que la viabilidad del sistema en desarrollo. Las estrategias para integrar los componentes de software en una unidad que funciones son

- Ascendente
- Descendente
- Emparedado

Y para esto es necesario que se tenga orden en que los módulos del sistema deben de estar disponibles.

La *integración ascendente* consiste en pruebas de unidad seguidas por pruebas de subsistemas y luego por pruebas del sistema completo. Las *pruebas de unidad* tienen el objetivo de descubrir los errores en los módulos individuales, probándose aisladamente uno de otro modulo, en un ambiente de prueba llamado "arreo de prueba", que esta formado por los programas en análisis y los datos necesarios para su ejecución. Las *pruebas de subsistema* verifican las interfaces entre los módulos y deblendo de probar las interfaces de control como de datos. Las *pruebas del sistema* se relacionan con, la lógica de decisión, el flujo de control, los procedimientos de recuperación, la eficiencia, y las características de tiempo del sistema completo. Este tipo de integración ascendente tiene la desventaja de que necesita escribir y depurar arreos de prueba para los módulos y subsistemas.

La *Integración descendente*, empieza con la rutina principal y una o dos unidades subordinadas en la estructura del sistema, convirtiéndose en una prueba de arreo para las subrutinas inmediatamente subordinadas del sistema, esta estrategia de integración presenta las siguientes ventajas:

1. La integración del sistema se distribuye en toda la fase de implantación, integrando los módulos a medida que se van desarrollando.
2. Las interfaces de nivel más alto se prueban primero y con más frecuencia.
3. Los errores se localizan en los nuevos módulos e interfaces que se están añadiendo.

La *Integración emparedado* es predominantemente descendente y la técnica ascendente se utiliza en algunos módulos y subsistemas. Mitigando muchos de los problemas encontrados.

IV.4 PRUEBA DE ACEPTACIÓN.

Incluye casos de prueba, resultados esperados y capacidades para demostrar que el sistema satisface las necesidades establecidas en los requisitos del usuario. Las herramientas de especial importancia durante esta prueba son:

1. Analizador de cobertura de la prueba: registra las rutas de control seguidas por cada caso de prueba.
2. Verificador de los estándares de codificación: pueden estar relacionados con el proyecto, con el cliente, con el responsable del desarrollo o con el lenguaje. La revisión manual no suele ser un mecanismo adecuado para detectar las violaciones a los estándares de codificación.

IV.5 CREACIÓN DE CASOS DE PRUEBA

Los casos de prueba deberán de simular fielmente los datos reales del sistema, debiendo crear casos de prueba para probar completamente el sistema, tomando en cuenta las entradas válidas e inválidas de datos. Los casos de prueba contiene los siguientes secciones:

1. Objetivo de la prueba
2. Resultados esperados
3. Caso de prueba
4. Resultados reales.

RESULTADOS ESPERADOS				
Objetivo de la prueba	rechazar	desplegar mensaje de error	diseño de casos de prueba	Resultados reales
Determinar si se calcula el dígito verificador y rechaza el número de cuenta	X	X	Introducir número de cuenta incorrectos	Rechazado, y despliega mensaje de error.
Verificación de claves de laboratorios	X	X	Introducir claves de laboratorios inválidas	Rechaza y despliega mensaje de error.
Verificación de RFC de profesores	X	X	Introducir RFC's inválidos	Rechaza y despliega mensaje de error.

IV.6 ARREO DE PRUEBA

El objetivo de este proceso consiste en introducir intencionalmente errores para probar la eficiencia y confiabilidad del sistema. Si los datos introducidos al sistema no detectan errores estamos hablando de la calidad y confiabilidad del sistema.

Las pruebas de unidad analizarán cada uno de los módulos que integran al sistema hasta que lleguemos a la integración total de los módulos involucrados. Se iniciará con una breve descripción de lo que cada uno de los módulos hace y de los errores que se deben de prever.

IV.7 PRUEBA DE MÓDULOS.

Módulo de verificación de números de cuenta. En este módulo la clave con la que el alumno accesa al sistema es el número de cuenta que está formado por 8 dígitos de los cuales el último es el dígito verificador, validando de entrada que los caracteres sean numéricos y después valida que los caracteres numéricos correspondan a un número de cuenta. Si el número de cuenta esta correcto el alumno seguirá el proceso, en caso contrario se desplegará el mensaje de 'Número de cuenta inválido' y se permitirá corregir el numero de cuenta.

Módulo de Inscripciones: Módulo en el cual se verifica que el alumno no este inscrito más de una vez en un mismo laboratorio y solo permitirá la inscripción a 4 laboratorios por semestre. Permitiendo la actualización de datos en el caso de que el alumno cambie de opinión o en su caso extremo darlo de baja en el laboratorio que el alumno lo solicite.

Módulo de validación de RFC: Módulo en el cual se verifica que la cadena de caracteres incluyan en los primeros cuatro caracteres 'Letras', los siguientes seis caracteres que sean números y que correspondan a una fecha, y los siguientes tres caracteres que sea el homónimo del RFC del profesor, que consta de una letra y dos numeros. En el caso de ser incorrecto el RFC se procederá a desplegar el mensaje de 'RFC inválido' y permitirá la corrección de este dato, y en caso contrario se seguirá el proceso.

Módulo de personal Académico: Módulo en el cual se verifica que los datos de un profesor no aparezcan más de una vez. Si no está se procederá a la Alta de datos personales del profesor. En caso de estar en el archivo se procederá a la actualización de datos y se podrá prescindir de estos mismos, en el caso de que así se requiera con la opción de Baja.

Módulo de Horarios: Módulo en el cual se permite almacenar los grupos, claves, profesores, que impartirán clases de laboratorios en el semestre que se encuentre en curso, validando las claves de los laboratorios dando pase a la captura, modificación o eliminación de un determinado grupo según sea el caso. En el caso de que la clave no corresponda a los laboratorios que se imparten se desplegará un mensaje de error, permitiendo al usuario corregir las claves de laboratorios.

Módulo de Reportes: En este módulo se hace la validación de datos según el reporte que se requiera. Para el caso de reportes de alumnos, se validan la clave y el grupo del laboratorio, para el caso de constancias de calificación se validan los números de cuenta y además de que el alumno no pertenezca al laboratorio de 'Física Experimental' ya que para obtener la calificación de laboratorio hay que volver a recurrir el laboratorio.

ESTA TESIS NO DEBE
VALER DE LA BIBLIOTECA

Módulo de bitácora: Dentro de cada uno de los módulos anteriormente descritos cada movimiento que el usuario realice se van registrando dentro de una bitácora que incluye: la opción del menú que eligió, la clave y nombre del departamento, la hora de entrada y salida de esta opción y el usuario que la accede, esto se hace con la finalidad de tener un histórico para poder detectar, las fallas y por quienes fueron ocasionadas en caso de que éstas sucedan, o bien para estar informado sobre el personal que está accediendo las diferentes opciones del sistema.

Módulo de inicialización: Dentro de este módulo existen dos tipos de inicialización: de semestre y de archivos, esto se hace con la finalidad de que los archivos durante cada proceso de reinscripción estén limpios y listos para iniciar el evento. así como la actualización del semestre en curso.

Módulo de respaldo: Este módulo permite al usuario tener protecciones de los archivos de datos, por diferentes motivos, que pueden ser para prever desastres, o simplemente para tener una serie de archivos históricos para el caso de que se lleguen a ocupar.

Módulo de estadísticas: En este módulo se presentarán en la pantalla las gráficas de unos o todas las calificaciones que existan en uno o varios grupos con una clave de laboratorio específica.

IV.8 CASOS DE PRUEBA.

Módulo a analizar	Objetivo de la prueba	Resultados Esperados en la introducción de datos			Descripción de casos de prueba	Resultados Reales
		Rechazar	Exhibe mensaje	Permite Corrección		
Verificación de números de cuenta	Determinar si se verifica el número de cuenta	X	X	X	Números de cuenta incorrectos	Rechaza el número de cuenta, exhibe el letrero de error y permite hacer la corrección.
Inscripción	Determina si los datos del alumno existen en el archivo		X		Números de cuenta correctos	Si los datos existen se despliegan en pantalla, o bien si los datos no existen permite la captura de estos y sigue el proceso.
	Determina si esta correcta la clave del laboratorio	X	X	X	Claves de laboratorios incorrectas Claves de Laboratorio inexistentes	Rechaza la clave del laboratorio, exhibe el letrero de error y permite hacer la corrección.
	Determinar si el grupo corresponde al laboratorio	X	X	X	Grupo incorrecto Grupo no corresponde al laboratorio	Rechaza el grupo, exhibe el letrero de error y permite hacer la corrección.

Módulo a analizar	Objetivo de la prueba	Resultados Esperados en la introducción de datos			Descripción de casos de prueba	Resultados Reales
		Rechazar	Exhibe mensaje	Permite Corrección		
Inscripción	Determinar si el alumno tiene derecho a la inscripción	X	X	X	Introducción de la clave y grupo de laboratorio donde el alumno ya se encuentra inscrito.	Rechaza la inscripción o actualización a ese laboratorio, exhibe el letrero de error y permite hacer la corrección.
Verificación de RFC.	Verificar que los datos capturados correspondan a un RFC	X	X	X	Introducción de RFC incorrectos	Rechaza el RFC, exhibe el letrero de error y permite hacer la corrección.
Horarios	Determina si esta correcta la clave del laboratorio	X	X	X	Claves de laboratorios incorrectas Claves de Laboratorio inexistentes	Rechaza la clave del laboratorio, exhibe el letrero de error y permite hacer la corrección.
Reportes	Determina si esta correcta la clave del laboratorio	X	X	X	Claves de laboratorios incorrectas Claves de Laboratorio inexistentes	Rechaza la clave del laboratorio, exhibe el letrero de error y permite hacer la corrección.
	Determinar si el grupo corresponde al laboratorio	X	X	X	Grupo incorrecto Grupo no corresponde al laboratorio	Rechaza el grupo, exhibe el letrero de error y permite hacer la corrección.

Módulo a analizar	Objetivo de la prueba	Resultados Esperados en la introducción de datos			Descripción de casos de prueba	Resultados Reales
		Rechazar	Exhibe mensaje	Permite Corrección		
Reportes	Determinar si el alumno tiene derecho a la constancia de calificación	X	X		Introducción de la clave y grupo de laboratorio que pertenecen al Física Experimental.	Rechaza la emisión de la constancia del laboratorio de física Experimental, exhibe el letrero de error y permite hacer la corrección.
Bitácora	Determinar si el usuario tiene acceso a este módulo	X	X		Elección de esta opción sin permiso a usaria.	Rechaza la clave de acceso a esta opción, exhibe el letrero de error
Inicialización	Determinar si el usuario tiene acceso a este módulo	X	X		Elección de esta opción sin permiso a usaria.	Rechaza la clave de acceso a esta opción, exhibe el letrero de error
Respaldo	Determinar si el usuario tiene acceso a este módulo	X	X		Elección de esta opción sin permiso a usaria.	Rechaza la clave de acceso a esta opción, exhibe el letrero de error
	Determinar que la ruta de respaldo exista	X	X	X	Introducción de ruta de respaldo inexistente.	Rechaza la ruta de acceso para respaldar la información, exhibe el letrero de error

Módulo a analizar	Objetivo de la prueba	Resultados Esperados en la Introducción de datos			Descripción de casos de prueba	Resultados Reales
		Rechazar	Exhibe mensaje	Permite Corrección		
Estadísticas	Determinar si el usuario tiene acceso a este módulo	X	X		Elección de esta opción sin permiso a usarla.	Rechaza la clave de acceso a esta opción, exhibe el letrero de error

IV.9 PRUEBAS DE INTEGRACIÓN Y ACEPTACIÓN.

Una vez probado cada uno de los módulos que compondrán el sistema se procedió a integrar los módulos, con el fin de poder verificar que los requerimientos del sistema se cumplen, para esto se llevo acabo una prueba general al sistema denominado pruebas de caja blanca, esta prueba consistió en probar que cada uno de los campos claves, cumpliera con las disposiciones pedidas en los requerimientos del sistema. Los campos claves son:

- Número de cuenta,
- RFC,
- Clave del laboratorio
- Grupo

Se realizaron las siguientes pruebas para los siguientes casos:

- de números de cuenta, claves de laboratorio y grupos:
 - Caracteres letras y números,
 - Espacios en blanco,
 - Caracteres letras, números y espacios en blanco; combinados,
 - Números que no corresponden a ninguno de los campos estudiados.
- de RFC
 - Fechas de nacimiento invalidas,
 - Caracteres mezclados de números y letras

Para cada uno de los casos anteriores se verifico que mandara el mensaje de error correcto, y en el caso de haber introducido los datos correctos se verifico que se siguiera con el proceso que se había determinado.

Para la aceptación del sistema, se sugiere que el usuario final (Personal que labora en el Departamento de Física), tenga a prueba el sistema para verificar que cumple con todas las especificaciones establecidas para el mismo, y así poder liberar el sistema.

Las actividades de mantenimiento implican realizar mejoras y/o actualizaciones en los módulos que conforman el sistema, adaptandolos a nuevos ambientes y corregir los problemas que este conlleva.

Como el sistema SIDF ha sido liberado, entregado y además entregada la documentación al personal que labora en el Departamento de Física, la fase de manteneimiento deberá de llevarse acabo él personal de éste Departamento, para así ajustar el sistema a las necesidades del semestre que va a iniciar.

APÉNDICE A
(MANUAL DE USUARIO)

MANUAL DE USUARIO

Este manual tiene por objetivo explicar el proceso de instalación del sistema, así como el manejo de cada una de las opciones del sistema SIDF (Sistema de Inscripciones del Departamento de Física).

Como primer punto se instalarán los archivos SIDF en el disco duro. Esto se llevará a cabo con un programa que se encuentra en el disco de instalación llamado **INSTALAR.BAT**. El Sistema SIDF requiere de:

- Una computadora IBM PC o compatible.
- Sistema operativo MS-DOS 5.0 o superior,
- 4 MB de memoria en RAM.
- Por lo menos una unidad de disco flexible,
- Un disco duro de por lo menos 5 MB de espacio libre para su instalación. Cabe mencionar que conforme usted almacena información el espacio requerido llega hasta 50 MB y deberá de tomar en cuenta este punto una vez instalado el sistema SIDF.

MANEJO DEL PROGRAMA DE INSTALACIÓN

La instalación del sistema se efectúa de manera automática, **INSTALA** creará la estructura del directorio SIDF para copiar los archivos del sistema. Siga los siguientes pasos para la instalación del sistema SIDF.

1. Inserte el disco de instalación en la unidad A:\
2. Teclee A: y presione ENTER para cambiar a la unidad A:\
3. Teclee **INSTALA** y presione ENTER.
4. Lea el mensaje y presione cualquier tecla para continuar.
5. Especifique el directorio destino o presione ENTER para respetar el directorio destino

Después de crear la aplicación, se deberá incluir en el archivo **AUTOEXEC.BAT** la siguiente línea para la actualización de la ruta. Por ejemplo si la ruta actual es:

PATH=C:\;C:\DOS; Usted deberá cambiarla a. **PATH=C:\;C:\DOS;C:\SIDF;**

LO QUE USTED NECESITA SABER

Poner en marcha el sistema SIDF: Una vez encendido su ordenador y esté en el prompt de dos por ejemplo C:\ teclee SIDF y presione ENTER para su ejecución. Una vez hecho esto visualizará la siguiente pantalla:

CLAVE DE USUARIO:

Aquí deberá de introducir la clave de acceso al sistema, dicha clave que consta de 5 caracteres, y a su vez le otorgara ciertos privilegios que le permitirán tener acceso a las diferentes opciones del sistema. El sistema le brinda tres oportunidades para acceder al sistema, si al termino de la tercera oportunidad usted vuelve a fallar el sistema saldrá automáticamente y se encontrara en el prompt de DOS. En caso contrario, es decir, que usted haya introducido la clave de acceso correcta se desplegará la siguiente pantalla:

U.N.A.M.	DEPARTAMENTO DE FISICA	F.I.
LABORATORIOS		

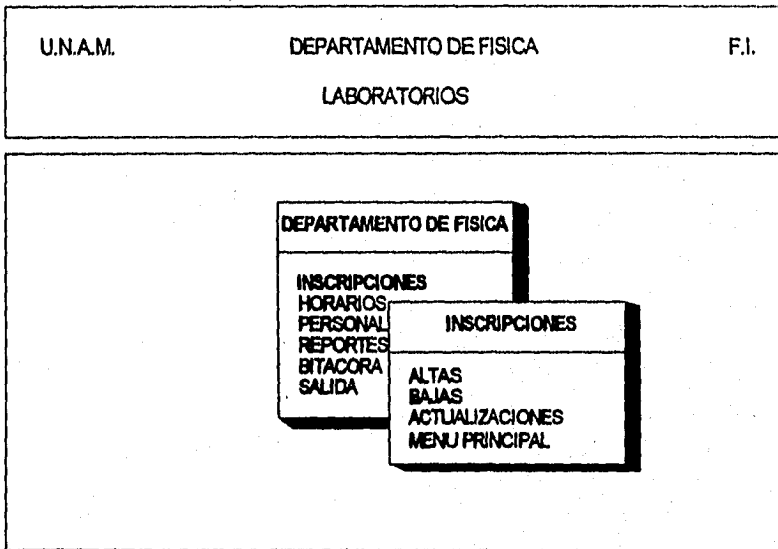
<table border="1"><tr><td style="text-align: center;">DEPARTAMENTO DE FISICA</td></tr><tr><td style="text-align: center;">INSCRIPCIONES HORARIOS PERSONAL ACADEMCO REPORTES BITACORA SALIDA</td></tr></table>	DEPARTAMENTO DE FISICA	INSCRIPCIONES HORARIOS PERSONAL ACADEMCO REPORTES BITACORA SALIDA
DEPARTAMENTO DE FISICA		
INSCRIPCIONES HORARIOS PERSONAL ACADEMCO REPORTES BITACORA SALIDA		

Este es el menú principal del sistema, de estas opciones solo podrá acceder aquellas a las que su clave de acceso le permita entrar. Los privilegios que tenga su clave de acceso serán aquellas opciones a las que usted tiene derecho a utilizar. Para seleccionar alguna opción basta con mover la flecha hacia arriba o hacia abajo y presionar <ENTER> para seleccionar la opción elegida.

A continuación se explicará cada una de las opciones para el Laboratorio de Física Experimental. Dado que en cada uno de los laboratorios se llevan a cabo las mismas actividades.

I INSCRIPCIONES

La opción de Inscripciones permitirá capturar los datos correspondientes a la inscripción, baja o actualización del alumno al laboratorio que este haya elegido. El menú que presenta la opción de inscripciones se muestra a continuación:



para cada una de las opciones que se presentan en este menú se presentará la siguiente pantalla de datos:

U.N.A.M.	DEPARTAMENTO DE FISICA	F.I.
LABORATORIOS		

No. DE CUENTA:	<input type="text"/>	
NOMBRE:	<input type="text"/>	
CLAVE DEL LAB.:	<input type="text"/> <input type="text"/>	
GRUPO DEL LAB.:	<input type="text"/>	
LABORATORIOS A LOS QUE ESTA INSCRITO:		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Esta pantalla permitirá al usuario dar de alta la inscripción del alumno al laboratorio que solicite.

La captura de información se hará de la siguiente manera:

Al momento de entrar a la pantalla el cursor estará posicionado en el campo número de cuenta, donde usted deberá introducir el número de cuenta del alumno a inscribir, en el caso de que el número de cuenta sea incorrecto el sistema se lo hará saber, en el caso contrario, el sistema se encargará de hacer las validaciones correspondientes y determinar si procede la inscripción del alumno, o de la contrario desplegará un mensaje indicando la causa. En caso de proceder la inscripción del alumno, deberá de hacer la captura del los campos.

CLAVE LAB: que consta de cuatro dígitos, si la clave de laboratorio que usted tecleo es correcta se desplegara el nombre del laboratorio y pasará al siguiente campo de captura, en caso de haber tecleado una clave invalida se desplegará un mensaje de error y podrá usted volver a teclear la clave del laboratorio.

GRUPO LAB: consta de cuatro dígitos, una vez tecleados los dígitos correspondientes, se verifica si existe el grupo y que además corresponda al laboratorio en cuestión, en caso de no concordar cualquiera de estas validaciones se desplegará un mensaje de error y usted podrá volver a capturar el grupo del laboratorio.

Una vez terminada la captura de datos se desplegará en la parte inferior de la pantalla el mensaje: ¿Están correctos los datos <S/N> ? y usted solo podrá responder S o N. Para el caso de respuesta 'S' el sistema almacenará los datos que están en pantalla, para el caso de 'N' el sistema permitirá al usuario hacer las correcciones que este crea pertinente hasta que estén correctos los datos.

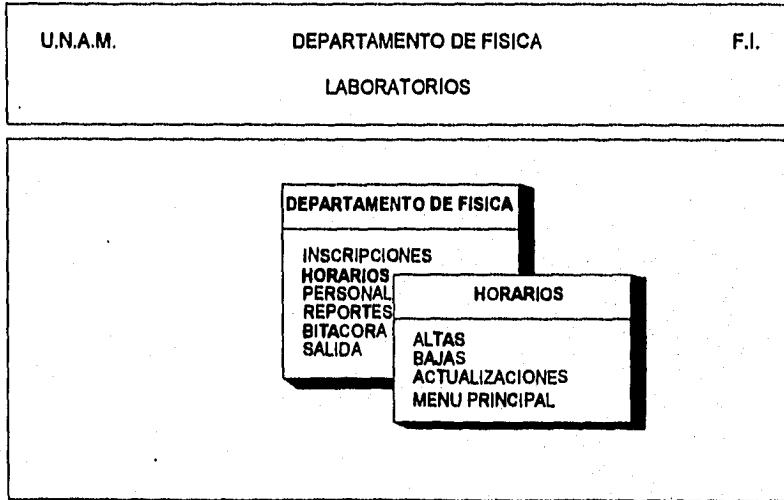
Para la opción de actualización y bajas de laboratorio se visualizará un menú en el cual usted podrá elegir el laboratorio a modificar; una vez hecho esto para el caso de actualización se procederá a la captura de datos que se acaba de mencionar. En el caso de baja de datos se preguntará si se desea dar de baja a este alumno en el laboratorio que se haya elegido.

Los tipos de respuesta del sistema son 'S' para almacenar, actualizar o dar de baja los datos. La Respuesta 'N' cancela todos los cambios hechos, impide dar de baja al registro del alumno y no almacena datos. Cabe hacer mención que presionando la tecla <ESC> todos los cambios o ajustes hechos se cancelan de inmediato en el alumno que se este visualizando en pantalla.

Para salir de la pantalla de captura basta con presionar <ESC> y saldrá al menú de inscripciones, y lo mismo para salir de este menú puede presionar <ESC> o posicionarse en la opción 'MENÚ PRINCIPAL' y presionar <ENTER>.

II HORARIOS

Al elegir esta opción en el menú principal se visualizará la siguiente pantalla:



Pantalla en la cual usted podrá seleccionar la opción que desee según la actividad que tenga que realizar. Para cualquiera de la opciones visualizara la siguiente pantalla de captura de datos:

U.N.A.M.	DEPARTAMENTO DE FISICA	F.I.
LABORATORIOS		

CLAVE MATERIA:	<input type="text"/>
NOMBRE MATERIA:	<input type="text"/>
GRUPO MATERIA:	<input type="text"/>
RFC DEL PROFESOR:	<input type="text"/>
CUPO MATERIA:	<input type="text"/>
INSCRITOS:	<input type="text"/>

El cursor esta posicionado en el campo CLAVE MATERIA que consta de cuatro dígitos, si la clave de laboratorio que usted tecleo es correcta se desplegará el nombre del laboratorio y pasará al siguiente campo de captura, en caso de haber tecleado una clave invalida se desplegará un mensaje de error y podrá usted volver a teclear la clave del laboratorio.

GRUPO MATERIA: consta de cuatro dígitos, una vez tecleados los dígitos correspondientes, se pasa al siguiente campo de captura .

RFC DEL PROFESOR: que consta de los cuatro primeros caracteres como letras los siguientes seis caracteres que corresponden a una fecha y que son dígitos y los últimos tres caracteres que corresponden al homónimo, de los cuales el primer caracter es una letra y los dos siguientes corresponden a números. En el caso de introducir el RFC correcto se pasará al siguiente campo de captura, en caso contrario se desplegará un mensaje de error para hacer la corrección del RFC.

CUPO MATERIA: que consta de dos dígitos ya que el cupo de un laboratorio no podrá ser nunca superior de 100 alumnos.

Una vez terminada la captura de datos se desplegará en la parte inferior de la pantalla el mensaje: ¿Están correctos los datos <S/N> ? y usted solo podrá responder S o N. Para el caso de respuesta 'S' el sistema almacenará los datos que están en pantalla, para el caso de 'N' el sistema permitirá al usuario hacer las correcciones que éste crea pertinente hasta que estén correctos los datos.

Para la opción de actualización y bajas de laboratorio deberá de capturar la CLAVE MATERIA y el GRUPO MATERIA para visualizar los datos correspondientes al laboratorio con un grupo definido. Una vez hecha esta captura de información se visualizarán, y realizará la actividad que usted eligió. En el caso de baja de datos se preguntará si se desea dar de baja el horario de la clave del laboratorio que se haya elegido.

Los tipos de respuesta del sistema son 'S' para almacenar, actualizar o dar de baja los datos. la Respuesta 'N' cancela todos los cambios hechos, impide dar de baja al registro del grupo en cuestión y no almacena datos. Cabe hacer mención que presionando la tecla <ESC> todos los cambios o ajustes hechos se cancelan de inmediato en el horario que se este visualizando en pantalla.

Para salir de la pantalla de captura basta con presionar <ESC> y salida al menú de horarios, y lo mismo para salir de este menú puede presionar <ESC> o posicionarse en la opción 'MENÚ PRINCIPAL' y presionar <ENTER>.

III. PERSONAL ACADÉMICO

La elección de esta opción visualizará el siguiente menú de opciones:

U.N.A.M.	DEPARTAMENTO DE FISICA	F.I.
LABORATORIOS		

DEPARTAMENTO DE FISICA
INSCRIPCIONES
HORARIOS
PERSONAL
REPORTES
BITACORA
SALIOA
PERSONAL ACADEMICO
ALTAS
BAJAS
ACTUALIZACIONES
MENU PRINCIPAL

independientemente de la opción a elegir se visualizará la siguiente pantalla de captura de datos del personal académico.

U.N.A.M.	DEPARTAMENTO DE FISICA	F.I.
LABORATORIOS		

NOMBRE PROF.:	<input type="text"/>				
RFC:	<input type="text"/>	CARRERA:	<input type="text"/>		
		ESPECIALIDAD:	<input type="text"/>		
CALLE.:	<input type="text"/>				
COLONIA:	<input type="text"/>				
DELEGACION:	<input type="text"/>	C.P.:	<input type="text"/>	TELÉFONO:	<input type="text"/>

En la pantalla de captura el cursor estará posicionado en el campo de RFC donde nosotros teclearemos el RFC del profesor., en el caso de no existir datos se procederá a la alta de información; en caso de existir datos en la base se procederá a la actualización o baja de datos según sea la actividad que se haya elegido. Para la captura de datos:

RFC DEL PROFESOR: que consta de los cuatro primeros caracteres como letras los siguientes seis caracteres que corresponden a una fecha y que son dígitos y los últimos tres caracteres que corresponden al homónimo, de los cuales el primer caracter pertenece a una letra y los siguientes dos caracteres son numéricos. En el caso de introducir el RFC correcto se pasara al siguiente campo de captura en caso contrario se desplegará un mensaje de error para hacer la corrección del RFC.

NOMBRE DEL PROFESOR: Consta de 32 caracteres tipo letras y espacios se puede capturar la 'Ñ', el sistema no permitirá el acceso de campos vacíos y justificara el nombre del profesor si al principio el usuario introduce espacios en blanco el sistema se encargara de quitarlos. Una vez cubiertos los 32 o menos caracteres se procederá a la captura del siguiente campo.

CARRERA y ESPECIALIDAD: los dos campos constan de 32 caracteres de entre los cuales se pueden capturar letras y espacios en blanco, el sistema no permitirá el acceso de campos vacíos al inicio de la cadena de caracteres y justificará

la CARRERA o ESPECIALIDAD, según el campo que se este capturando. Una vez cubiertos los 32 o menos caracteres se procederá a la captura del siguiente campo.

CALLE y COLONIA constan de 35 caracteres de entre los cuales consta de letras, números y algunos símbolos como punto, punto y coma y # así como el espacio en blanco; el sistema no permitirá el acceso de campos vacíos y justificará la CALLE o COLONIA, si al principio de cada campo el usuario introduce espacios en blanco el sistema se encargará de quitarlos. Una vez cubiertos los 32 o menos caracteres se procederá a la captura del siguiente campo.

DELEGACIÓN consta de dos dígitos que son las claves con las que se localizan los nombres de las delegaciones en un archivo de datos.

CÓDIGO POSTAL: consta de cinco dígitos, el sistema no permitirá el acceso de campos vacíos, a vez cubiertos los 5 caracteres que corresponden al código postal, se procederá a la captura del siguiente campo.

TELÉFONO consta de 13 o menos dígitos y es el único campo que puede ir en blanco.

Una vez terminada la captura de datos se desplegará en la parte inferior de la pantalla el mensaje: ¿'Están correctos los datos <S/N> ? y usted solo podrá responder S o N. Para el caso de respuesta 'S' el sistema almacenará los datos que están en pantalla, para el caso de 'N' el sistema permitirá al usuario hacer las correcciones que éste crea pertinente hasta que estén correctos los datos.

Para la opción de actualización y bajas de laboratorio deberá de capturar el RFC del profesor para visualizar los datos correspondientes al personal académico y se realizara la actividad que usted eligió. En el caso de baja de datos se preguntará si se desea dar de baja los datos de este profesor que usted haya elegido.

Los tipos de respuesta del sistema son 'S' para almacenar, actualizar o dar de baja los datos. La Respuesta 'N' cancela todos los cambios hechos, Impide dar de baja o almacenar los datos del profesor que se tenga en pantalla. Cabe hacer mención que presionando la tecla <ESC> todos los cambios o ajustes hechos se cancelan de inmediato en el horario que se este visualizando en pantalla.

Para salir de la pantalla de captura basta con presionar <ESC> y salida al menú del Personal Académico, y lo mismo para salir de este menú puede presionar <ESC> o posicionarse en la opción 'MENÚ PRINCIPAL' y presionar <ENTER>.

IV REPORTEES

En la elección de este menú se visualizarán las siguientes opciones de menú:

U.N.A.M.	DEPARTAMENTO DE FISICA LABORATORIOS
----------	--

DEPARTAM	REPORTES
INSCRIPCI HORARIOS PERSONAL REPORTES BITACORA SALIDA	ALFABETICO GENERAL ALFABETICO POR CLAVE DE LAB. TODOS ALFABETICO POR CLAVE DE LAB. SOLO UNO NUMERO DE CUENTA GENERAL NUMERO DE CUENTA POR CLAVE DE LAB. SOLO UNO CONSTANCIA DE CREDITOS RELACION DE PROFESORES GENERAL RELACION DE PROFESORES POR CLAVE DE LAB. RELACION DE CUPOS ACTUALES GENERAL RELACION DE CUPOS ACTUALES POR CLAVE DE LAB. MENU PRINCIPAL

Al elegir algunas de las opciones que indiquen CLAVE DE LAB. se visualizará la siguiente pantalla de captura:

CLAVE DE LAB.:	<input type="text"/>	<input type="text"/>
GRUPO DE LAB.:	<input type="text"/>	

donde la captura de estos dos campos se lleva acabo de la siguiente manera:

CLAVE LAB: que consta de cuatro dígitos, si la clave de laboratorio que usted tecleo es correcta se desplegará el nombre del laboratorio y pasará al siguiente campo de captura, en caso de haber tecleado una clave inválida se desplegará un mensaje de error y podrá usted volver a teclear la clave del laboratorio.

GRUPO LAB: consta de cuatro dígitos, una vez tecleados los dígitos correspondientes, se verifica si existe el grupo y que además corresponda al

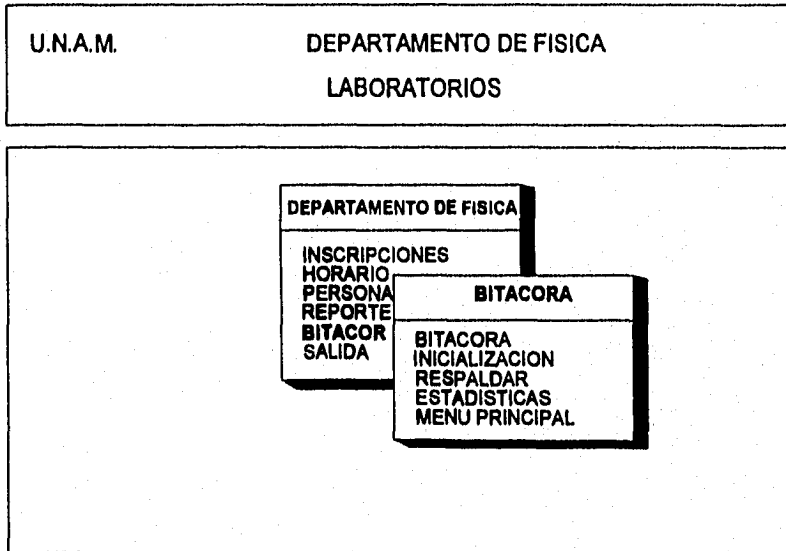
laboratorio en cuestión, en caso de no concordar cualquiera de estas validaciones se desplegará un mensaje de error y usted podrá volver a capturar el grupo del laboratorio.

En cualquiera de estas opciones de reportes se preguntará si se direccionan los datos a pantalla o a la impresora, para esto usted deberá de contesta a la pregunta 'S' o 'N', según sus indicaciones.

Cabe hacer mención de que para la emisión de constancias no se le dará derecho al alumno que la pida y este cursando el Laboratorio de Física experimental. Para esta opción deberá de teclearse el numero de cuenta del alumno que solicita la constancia de calificación, una vez introducido el número de cuenta se validará para determinar si es correcto, de lo contrario el sistema se lo hará saber, si es correcto, el sistema se encargará de hacer las validaciones correspondientes y determinar si procede la emisión de la constancia de calificación.

V BITÁCORA

En la opción de bitácora se visualizarán las siguientes opciones del submenu, de las que usted podrá elegir a las que tenga usted acceso:



el menú de bitácora solo tendrá acceso a esta opción personal autorizado, con la finalidad de saber cada uno de los movimientos durante la ejecución del sistema. La pantalla de información es:

U.N.A.M.	DEPARTAMENTO DE FISICA	F.I.
OPCION	CLAVE DE LABORATORIO	ENTRADA SALIDA USUARIO
FECHA SELECCIONADA:		

En la opción de inicialización se podrá inicializar el semestre o los archivos para la nueva captura de información. el menú que se presenta en el sistema es el siguiente:

U.N.A.M.	DEPARTAMENTO DE FISICA
LABORATORIOS	

INICIALIZACION
INICIALIZACION DE SEMESTRE
INICIALIZACION DE ARCHIVOS

PERSONA	BITACORA
REPORTE	BITACORA
BITACOR	INICIALIZACION
SALIDA	RESPALDAR
	ESTADISTICAS
	MENU PRINCIPAL

en el caso de haber elegido la primera opción se visualizará la siguiente pantalla:

U.N.A.M.	DEPARTAMENTO DE FISICA LABORATORIOS
----------	--

INICIALIZACION		
INICIALIZACION DE SEMESTRE INICIALIZACION DE ARCHIVOS		
PERSONA REPORTE BITACOR SALIDA	BITA	SEMESTRE: <input type="text"/>
	BITACORA INICIALIZACION RESPALDAR ESTADISTICAS MENU PRINCIPAL	

donde deberá capturarse el nuevo semestre, el cual consta de tres dígitos, los dos primeros corresponden al año y el tercer dígito corresponde al semestre 1 o 2. Para el caso de haber elegido la segunda opción ésta presentara la siguiente pantalla:

U.N.A.M.	DEPARTAMENTO DE FISICA LABORATORIOS
----------	--

INICIALIZACION		
INICIALIZACION DE SEMESTRE INICIALIZACION DE ARCHIVOS		
PERSONA REPORTE BITACOR SALIDA	BIT INI RE ES ME	ARCHIVOS CREA ARCHIVO DE ALUMNOS CREA ARCHIVO DE PROFESOR CREA ARCHIVO DE GRUPOS CREA ARCHIVO DE LABORATORIOS CREA ARCHIVO DE ESTADOS CREA ARCHIVO DE DELEGACION CREA ARCHIVO DE CARRERAS CREA ARCHIVO DE INSCRITOS CREA ARCHIVO DE AGENOA CREA ARCHIVO DE SEMESTRE

y usted deberá elegir cual de las diferentes opciones es la que usted debe elegir. El sistema borrará el archivo existente y dejara en su lugar un archivo limpio y listo para empezar a usarse.

En el caso de que usted haya decidido respaldar su información se presentará la siguiente pantalla, para que usted pueda elegir el archivo a respaldar:

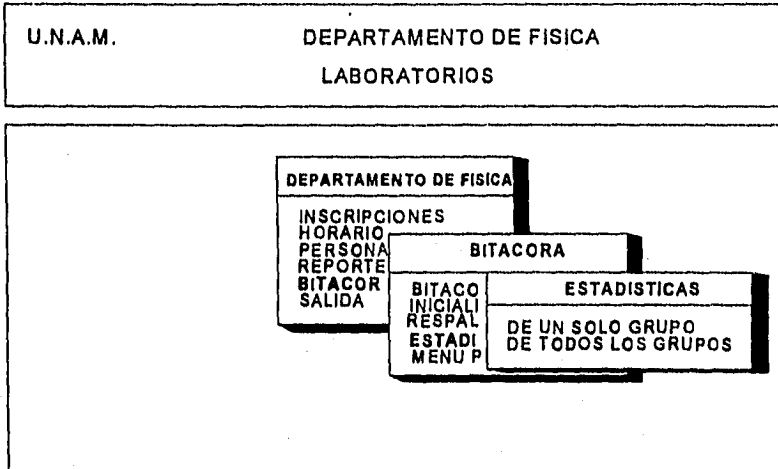
U.N.A.M.		DEPARTAMENTO DE FISICA	
		LABORATORIOS	

DEPARTAMENTO DE FISICA		RESPALDAR	
INSCRIPC HORARIO PERSONA REPORTE BITACOR SALIDA	BITACO INICIALI RESPA ESTADI MENU P	ARCHIVOS	
		RESPALDAR ARCHIVO DE ALUMNOS RESPALDAR ARCHIVO DE PROFESOR RESPALDAR ARCHIVO DE GRUPOS RESPALDAR ARCHIVO DE LABORATORIOS RESPALDAR ARCHIVO DE ESTADOS RESPALDAR ARCHIVO DE DELEGACION RESPALDAR ARCHIVO DE CARRERAS RESPALDAR ARCHIVO DE INSCRITOS RESPALDAR ARCHIVO DE AGENDA RESPALDAR ARCHIVO DE SEMESTRE	

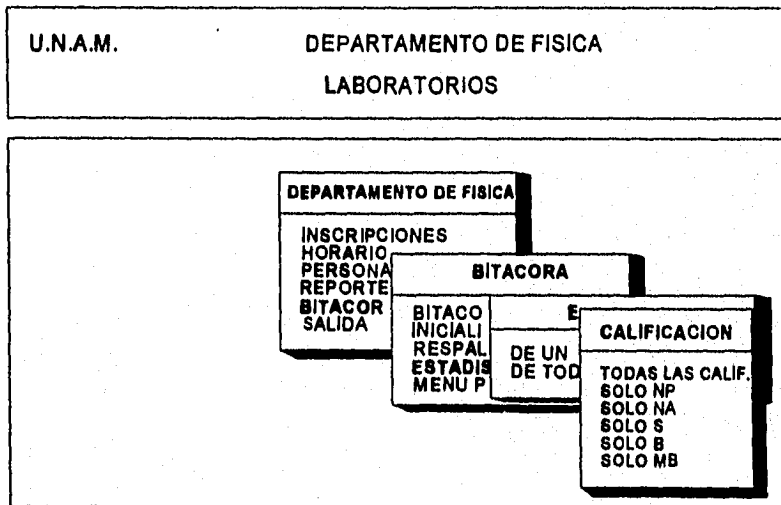
una vez elegido su archivo a respaldar usted deberá de introducir la ruta de respaldo y el nombre del archivo como usted desee llamarle al archivo de respaldo; el sistema se encargará de investigar si la ruta que usted eligió existe y el mismo le dará a conocer su resultado en caso de error. Dichos datos usted deberá introducirlos en la siguiente pantalla que se visualizará en el momento de que usted elija una de las opciones a respaldar.

ARCHIVO A RESPALDAR:	<input type="text"/>
RUTA:	<input type="text"/>
NOMBRE:	<input type="text"/>

En el caso de que usted haya elegido la opción de obtener las estadísticas de calificaciones el sistema le presentará el siguiente menú:



usted deberá de elegir una opción, una vez elegida esta opción, se visualizará la siguiente pantalla:



una vez elegida la opción de este menú se visualizará la siguiente pantalla para saber de que laboratorio se han de obtener las calificaciones y en su caso de que grupo en particular. Visualizándose la siguiente pantalla inmediatamente después de haber elegida la opción de este menú:

CLAVE DE LAB.:	<input type="text"/>	<input type="text"/>
GRUPO DE LAB.:	<input type="text"/>	

El cursor esta posicionado en el campo CLAVE LAB. que consta de cuatro dígitos, si la clave de laboratorio que usted tecleo es correcta se desplegará el nombre del laboratorio y pasará al siguiente campo de captura, en caso de haber tecleado una clave invalida se desplegará un mensaje de error y podrá usted volver a teclear la clave del laboratorio. GRUPO LAB.: consta de cuatro dígitos, una vez tecleados los dígitos correspondientes, se pasa al proceso de estadísticas que se pidió.

Para salir de cualquiera de estas pantalla de menú o captura basta con presionar <ESC> y saldrá al menú de anterior, y lo mismo para salir de este menú puede presionar <ESC> o posicionarse en la opción 'MENÚ PRINCIPAL' y presionar <ENTER>.

Observaciones:

- Para quita los mensajes de la pantalla basta con teclear <ENTER>
- Para cancelar los cambios hechos presionar <ESC>
- Para responder a una pregunta sólo con 'S' o 'N'

APÉNDICE B
(DICCIONARIO DE DATOS)

DICCIONARIO DE DATOS

El Diccionario de Datos es un documento que define y registra los elementos de entidades relevantes para el sistema que se describe. Son de gran importancia ya que implica el control de las bases de datos, porque en el caso de modificar algún programa se sabe con exactitud que otro u otros programas pueden llegarse a afectar con dicha modificación. A continuación se presenta el diccionario de datos de el sistema SIDF.

Nombre del campo	Entidad a la que pertenece	Nombre de la base de datos	Programas en los que se usa	Propósito
Cuenta	Inscritos	Inscrit.dat	Inscrito	<ul style="list-style-type: none"> • dar de alta la inscripción del alumno • dar de baja la inscripción de algún laboratorio • actualizar los datos de inscripción
	Alumnos	Alumnos.dat	Inscritos	<ul style="list-style-type: none"> • Para obtener los datos del alumno al que pertenece este número de cuenta • Dar de alta aquel numero de cuenta que no existe en archivo y que además esta correcto.
Rfc	Profesor	Profeso.dat	Academic	<ul style="list-style-type: none"> • dar de alta los datos de algún profesor • dar de baja los datos del profesor que ya no impartirá clase en este departamento. • actualizar la información
	Grupos	Gruposs.dat	Grupos	<ul style="list-style-type: none"> • Obtener el nombre del profesor que impartirá alguna de las clases de laboratorio.

Nombre del campo	Entidad a la que pertenece	Nombre de la base de datos	Programas en los que se usa	Propósito
Clave_Lab	Grupos	Gruposs.dat	Grupos	<ul style="list-style-type: none"> dar de alta grupo que se impartirá en el semestre en curso dar de baja al registro de grupos que se encuentre incorrecto o que así se disponga actualizar los datos de los grupos
	Laboratorio	Asignat.dat	Grupos	Desplegar el nombre del laboratorio.
Clave_Lab	Inscritos	Inscrit.dat	Inscrito	desplegar el nombre del laboratorio.
Clave_Del	Delegacion	Delegac.dat	Academic	Desplegar el nombre de la Delegacion
Clave_Estado	Estado	Estados.dat	Academic	desplegar el nombre de la Delegacion o municipio de donde radique el profesor
Clave_Carrera	Carreras	Carrera.dat	Academic	Desplegar el nombre de la carrera del profesor
			Inscrito	Desplegar el nombre de la carrera del alumno

CONCLUSIONES

Aunque existe un sin número de softwares en el mercado para la realización del sistema, se hizo un análisis exhaustivo de algunos que se tienen al alcance considerando el no depender de un experto en un software especial para hacer modificaciones al sistema. El software elegido fue Btrieve que es un manejador de bases de datos y se apoya para su manejo de lenguajes de programación de alto nivel, en nuestro caso se utilizó Turbo PASCAL V6.0, para la realización de los módulos de programación que conformaron el sistema, teniendo como ventaja que cualquier alumno de la Facultad de Ingeniería pueda darle mantenimiento al sistema, con los conocimientos de programación que este tenga y no tener la necesidad de contratar a un experto en software para que descifre la programación del sistema y desperdiciar recursos económicos en su contratación, teniendo a algunos alumnos de la facultad dispuestos a colaborar, ya que están capacitados para darle mantenimiento a este sistema de computación.

Durante el desarrollo de la presente tesis se programó de forma modular y se unificaron cada uno de estos módulos para darle forma al sistema SIDF, una vez que el sistema fue aprobado por el Departamento, se instaló, con lo cual quedaron resueltos los siguientes problemas que se presentaban:

- Para la expedición de constancias de calificación: el alumno no tendrá que regresar al día siguiente para que se le extienda su constancia, ya que en el momento en que se teclee su número de cuenta para la emisión de constancias el sistema imprimirá su constancia de calificación.
- Así mismo se hace la captura de datos personales de los profesores así como de los Grupos que se imparten en el semestre en curso para que en el momento de las inscripciones a los laboratorios esta información se tenga lista para su utilización.
- Durante las inscripciones a los laboratorios el sistema detectará si algún alumno trata de cometer alguna anomalía y será detectado por el sistema y se le hará saber al alumno.
- Una vez terminado el semestre se podrán capturar las calificaciones obtenidas por los alumnos para así poder obtener las estadísticas y saber cual es el índice de reprobación que existe en un grupo, y así poderlo comparar con la gráficas que se hayan obtenido de semestres anteriores.

estos eran unos de los tantos problemas a los que se enfrentaba el personal del Departamento de Física

La metodología usada para la realización de la presente tesis, consistió en primera instancia a hacer un análisis detallado de lo que el Departamento de Física requería para la realización del proyecto, en segunda instancia se hizo un análisis detallado de los diferentes softwares que existen en el mercado y dado que requieren de un equipo de computo a la vanguardia se tomo el software que se ajusto a lo que el departamento en cuanto a materia de computo tiene. En tercera instancia consistió en hacer una programación modular, para llevar lo que se tenía en papel a la programación de alto nivel, diseñando e implementando cada uno de los módulos que se necesitaron para cubrir los requerimientos del Departamento y probando cada uno de estos en cuanto se tenían terminados. Una de las ventajas más sobresalientes de este tipo de programación radica en el momento de querer hacer alguna actualización en alguno de los módulos que conforman el sistema, ya que el sistema seguirá funcionando de forma parcial y sólo será suspendido el módulo a modificar el tiempo que lleve su actualización; misma ventaja que se tendrá en el momento en que se le de mantenimiento al sistema, es decir, que funcionarán todas aquellas opciones excepto aquellas que requieran del mantenimiento.

Para la instalación de dicho sistema los problemas que fueron planteados, en la etapa de análisis de requerimientos de sistema, han quedado cubiertos. Y dado que se han cumplido con los lineamientos que el Departamento de Física fijó durante el desarrollo de esta Tesis, y estando de acuerdo con la implementación y puesta en marcha del sistema SIDF se da por concluida la presente Tesis.

BIBLIOGRAFÍA

A. Ramalho José

Clipper 5.01 Avanzado

Editorial McGraw-Hill

España 1992.

375 pp.

Adad, Rubén

Fundamentos de las Estructuras de Datos Relacionales

Editorial Megabyte

México, D.F., 1993

226 PP.

Barba, Atilano Jorge

Sql con Oracle.

Facultad de Ingeniería U.N.A.M. Unidad de Servicios de Cómputo Académico,

México, D.F., 1995.

18 pp.

Barba, Atilano Jorge

Modelado de Bases de Datos

Facultad de Ingeniería U.N.A.M. División de Educación continua

México, D.F., 1995,

18 pp.

Borland Paradox For Windows

Borland Internacional Inc.

Scotts Valley, USA. 1992

521 pp.

Date, C.J

Introducción a los Sistemas de Bases de Datos.

Editorial Addison-Wesley Iberoamericana.

México, D.F., 1986.

648 pp.

Dittrich Stefan

Visual Basic Acceso Rápido

Editorial Computec, Marcombo

México, D.F., 1993

138 pp.

F., Korth Henry
Fundamentos de Bases de Datos
McGraw-Hill
México, D.F., 1990
528 pp.

Fairley, Richard
Ingeniería de Software
Editorial McGraw-Hill
Edo. de México, 1990
390 pp.

Gerez, Victor
Desarrollo y Administración de programas de computadora (Software).
Editorial Continental.
México, D.F., 1985
300 pp.

Koch, George
Oracle7 Manual de referencia
Editorial McGraw-Hill
México 1995.

Liskin, Miriam
Dbase III Plus Avanzado
Editorial McGraw-Hill
México, D.F., 1988
748 pp.

Microsoft, Visual Basic
México, D.F., 1993

Professional Devopment Series
Btrieve for DOS Installation y Operation
Novell, Incorporated
Austin, Texas USA 1991.

Professional Devopment Series
Btrieve Programmer's manual
Novell, Incorporated
Austin, Texas USA 1991.

Sommerville, Ian
Ingeniería de Software
Editorial Addison-Wesley Iberoamericana.
Edo. México, 1988
364 pp.

Alan, Freedman
Diccionario de Computación
© 1981-1993 The Computer Language Co. Inc.

Red Internet

Direcciones:

Info BtrvTech.com

TechSupport BtrvTech.com

Se estudiaron los siguientes artículos:

How To Work with Btrieve Technologies.

Who is Btrieve Technologies?

Getting Started

Btrieve Technologies Solutions Directory

Btrieve Technologies Solution Network

Don't miss this opportunity to be listed on the Btrieve Technologies.

Join BTI's Solution Network

What is the Btrieve Technologies solution Network.

Btrieve 6 Navigational Client/Server Database

Take the safe, Direct Route to Client/Server.

Scale from standalone to client/server with no recoding

seaworthy reliability, State-of-the-art performance

the painless course to SQL

anchor your operations in btrieve 6 and relax

Upgrade to the btrieve 6.15. server.

additional product features

product platforms and contents

32-Bits system recommendations

client/server database engines

Revistas

Byte México
Año 9 No. 90 Julio 1995
74 pp.

PC Computing En español.
Año 02, Número 11
98 pp.

PC Magazine
Mayo 11, 1993 Volume 12 Number 9
458 pp.

PC Magazine
Mayo 31, 1994 Volume 13 Number 10
438 pp.

PC Memo
Año 5/N°14 Agosto 1991
40 pp.

Personal Computing México.
Octubre 1995.
80 pp.