17
20

# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN

## DISEÑO, CREACION Y DESARROLLO DE UN SOFTWARE INTERACTIVO PARA EL MANEJO ADMINISTRATIVO DE LABORATORIOS DE QUIMICA

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO QUIMICO

P R E S E N T A

**DOMINGO SERGIO SANCHEZ SANCHEZ**

ASESOR: M. EN C. ENRIQUE ANGELES ANGUIANO



CUATITLAN IZCALLI EDO: DE MEXICO 1994


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

UNIVERSIDAD NACIONAL AVENOMA DE MEXICO

# FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN
## UNIDAD DE LA ADMINISTRACION ESCOLAR
## DEPARTAMENTO DE EXAMENES PROFESIONALES

U. N. A. M.
FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN

Departamento de Exámenes Profesionales

**ASUNTO:** VOTOS APROBATORIOS

**DR. JAIME KELLER TORRES**
**DIRECTOR DE LA FES-CUAUTITLAN**
**P R E S E N T E .**

**AT'N:** Ing. Rafael Rodríguez Ceballos
Jefe del Departamento de Exámenes
Profesionales de la F.E.S. - C.

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la TESIS TITULADA:
Diseño, Creación y Desarrollo de un Software interactivo para el manejo administrativo de laboratorios de química.

que presenta el pasante: Domingo Sergio Sánchez Sánchez
con número de cuenta: 7240021-6 para obtener el TITULO de:
Ingeniero Químico .

Considerando que dicha tesis reúne los requisitos necesarios para ser discutida en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

A T E N T A M E N T E .
"POR MI RAZA HABLARA EL ESPIRITU"
Cuautitlán Izcalli, Edo. de Méx., a 9 de marzo de 1994

| | | |
|---|---|---|
| **PRESIDENTE** | I.Q.I. Alvaro Leo Ramírez | 9/03/94 |
| **VOCAL** | I.P. Moises Cobos Butrón | 17/03/ |
| **SECRETARIO** | M.enC. Enrique Angeles Anguiano | 14/02/9 |
| **PRIMER SUPLENTE** | M.enC. Ricardo P. Hernández García | |
| **SEGUNDO SUPLENTE** | I.Q. Rogelio Ramos Carranza | 9/FEB./94 |

UAE/DEP/VAP/01

## AGRADECIMIENTOS.

Agradezco profundamente al M.C. Enrique Angeles Anguiano y al Q. Ismael Salas Butrón, el haber dirigido la elaboración del presente trabajo y el tiempo dedicado a la revisión del mismo.

Un especial agradecimiento a los ingenieros Alvaro Leo y Fernando Beristain, por el apoyo y esfuerzo que me brindaron para el inicio y culminación de este trabajo.

A la Facultad de Estudios Superiores Cuautitlán.
U.N.A.M.

**DEDICATORIAS.**

A **Dios**, por todas las bendiciones que me ha dado y permitirme alcanzar otra meta en mi vida.

A la memoria de mis padres, **Severino** y **Juanita**, por sus consejos y ejemplos que me dieron.

A mis **Hermanos**: Francisco
Socorro
Irma
Vicente
Lourdes
y Pedro.
Por su gran apoyo en todo.

A todos mis **jefes y compañeros** de trabajo, por su ayuda y amistad que me han ofrecido.

A todos mis **amigos** por la gran amistad y momentos buenos y malos que hemos compartido juntos.

## TEMA.

Diseño, creación y desarrollo de un software interactivo para el manejo administrativo de laboratorios de Química.

INDICE.

pag

**CAPITULO 3**

MANIPULACION DE LOS DATOS POR MEDIO DEL LENGUAJE DE DBIII.

## INTRODUCION.

Actualmente las computadoras han cobrado una gran importancia de tal forma que cada día se hace necesario tener conocimientos de ellas. Muchos de nosotros tenemos contacto directo o indirecto con una computadora dentro de nuestras actividades diarias que realizamos, como es el caso de ir a un banco, o hacer una reservación en un hotel, en la medicina, en la educación, en los cálculos de la ingeniería, en control de inventarios, etc.

El presente trabajo brinda la oportunidad de que cualquier persona de un laboratorio de química, como el responsable del laboratorio, el profesor que imparte el laboratorio, el laboratorista o el alumno, y que esté en contacto con una computadora, pueda aprender los conceptos básicos de las bases de datos, la metodología para el diseño de una base de datos eficiente y sus usos y aplicaciones.

Además, de conocer los principios básicos de un manejador de base de datos, como es el lenguaje de DBaseIII plus, y su programación, mismo que será una de las herramientas más utiles en el desarrollo de un sistema administrativo, como ayuda auxiliar del

manejo administrativo de algún laboratorio de Química.

En el presente trabajo se tomó como ejemplo al laboratorio de Química Orgánica de la F.E.S. Cuautitlán, debido a la diversidad de cursos que imparte, población estudiantil y organización, se eligió dicho laboratorio por considerarse uno de los más completos y así tener la seguridad de que el material presentado aquí sea digno de tomarse en cuenta para la administración general de cualquier otro laboratorio de química.

## OBJETIVOS.

**OBJETIVO GENERAL.** Proporcionar al administrador, responsable o estudiante de las carreras de Química los conocimientos para poder diseñar una base de datos en DBASEIII, por medio de un ejemplo de las prácticas realizadas en un laboratorio de Química.

**OBJETIVOS PARTICULARES.**

- Conocer que es una base de datos, su estructura y almacenamiento.
- Conocer como diseñar una base de datos relacional.
- Conocimiento básico del DBIII.
- Ejemplo aplicativo en una base de datos relacional de las prácticas realizadas en un laboratorio de las carreras de Química.
- Conocer la importancia de la documentación de un sistema aplicativo.

# C A P I T U L O 1

## CONCEPTOS BASICOS DE BASE DE DATOS

En la actualidad las computadoras se estan revolucionando más, de tal forma que han simplificado y facilitado mucho los trabajos manuales tradicionales, automatizándolos y haciéndolos más rápidos, por consiguiente, es necesario adquirir los conocimientos de computación en álgun lenguaje (programador) o saber manejar una computadora (usuario). Por lo que es necesario conocer algunas definiciones conceptuales sobre lo que es una base de datos y elementos que la componen para poder entender su diseño y manejo.

### 1.1 ¿QUE ES UN SISTEMA DE BASE DE DATOS?

Digamos que un sistema de base de datos no es más que "un sistema de mantenimiento de registros basado en computadores, es decir, un sistema cuyo propósito general es registrar y mantener información o datos"[1], de tal forma, que contiene los datos necesarios de una organización para los procesos de toma de decisiones. La base de datos debe ser integrada y compartida, pudiéndose considerar como una unificación de varios archivos de datos independientes donde se elimina parcial o totalmente

cualquier redundancia entre los mismos, accesándola por varios usuarios al mismo tiempo.

**DATOS.** Los datos tienen las características de propiedad, atributos y valor, es decir que la propiedad es a lo que pertenece, que podemos llamar 'Entidad', y esta puede ser una persona, un lugar, una cosa, un evento o un concepto del cual se registra la información, el atributo es lo que lo caracteriza y puede describir por su tamaño, color, antigüedad, etc., y el valor que es el contenido y puede ser cuantitativo, cualitativo o descriptivo.

**REGISTROS.** Un registro es una colección de datos relacionados que se almacenan en algún medio, éste puede ser una hoja de papel, la memoria de la computadora, o un dispositivo auxiliar de almacenamiento como una cinta, un disco o tambor, un diskette, etc.

**ARCHIVO.** "Un archivo es un conjunto de registros"[2] de la misma especie o tipo, el cual puede ser homogéneo (registros con el mismo número de campos y longitud) y heterogéneo (registros de diferente número de campos de datos y por tanto de longitud).

**DICCIONARIO DE DATOS.** Un diccionario de datos es un depósito central de información acerca de las entidades, de los campos de datos que representan estas entidades, sus orígenes, significados, usos, formatos de representación en los diseños lógicos y físicos y sus relaciones entre sí mismas.

**BASE DE DATOS.** "Una base de datos es un conjunto de datos de operación almacenados y utilizados por los sistemas de aplicación de una empresa específica"[3].

Las ventajas de usar una base de datos son:

- Reducir la redundancia y así disminuir espacio de almacenamiento por repetición de datos.

- Evitar inconsistencia y asegurar que cualquier cambio hecho a una o más entradas, se efectué de manera automática la actualización. Entendiendo como actualización la creación supresión y modificación.

- Que los datos puedan compartirse y así nuevas aplicaciones puedan usar los datos existentes.

- Centralizar los datos y unificarlos como ayuda para el intercambio o migración de datos entre sistemas.

- La seguridad sobre los datos o controles de acceso como recuperar, suprimir, crear, modificar, etc.

- La integridad que es garantizar que los datos de la base de datos sean exactos y verídicos.

- La independencia de los datos, esto es, permitir que la base de datos sea capaz de crecer sin que se afecten las aplicaciones existentes.

¿QUIENES ACCESAN LA BASE DE DATOS?.

Los usuarios que accesan la base de datos son tres:

1.- Los programadores que se encargan de escribir programas de aplicación.

2.- Los usuarios finales que accesan la base de datos desde una terminal.

3.- Los administradores de base de datos o DBA's que se encargan de organizar el sistema y hacer los ajustes adecuados en la medida que cambian los requerimientos a la base de datos.

ADMINISTRACION DE LA BASE DE DATOS.

El proceso administrativo de toma de decisiones depende de la calidad y cantidad de la información existente. Por lo que la base de datos se debe diseñar, procesar y mantener adecuadamente para proporcionar información correcta en el momento oportuno a las personas autorizadas a modificar la

base de datos. La función de administración de la base de datos esta compuesta de personas (administrador de base de datos llamado DBA) y procedimiento (manejador del sistema de base de datos llamado DBMS). La diferencia entre el DBA y el DBMS es que el DBMS en su mayoría es un paquete que se compra y el DBA tiene la tarea de resolver las diferencias entre varias funciones de la organización, con el fin de desarrollar una estructura conceptual y la lógica del modelo de base de datos, el DBA es el negociador entre diferentes departamentos para llegar a un acuerdo 'correcto' sobre las entidades de la empresa.

El DBA al inicio del diseño de la base de datos deberá concentrarse en:

- La definición de los campos de datos y sus entidades.

- La determinación de los distintos nombres que se usarán para referirse a los mismos elementos.

- La definición de las relaciones entre los campos de datos.

- El establecimiento de la descripción textual de los campos de datos

- La determinación del uso de los campos de datos con propósitos de control y planeación.

**CICLO DE VIDA DEL SISTEMA DE BASE DE DATOS.**

Las fases principales de vida del sistema de base de datos son:

1.- Diseño de la base de datos. Este dependerá de toda la información levantada para definir campos de datos, sus entidades y las relaciones intrínsecas. Es necesario hacer un diccionario de datos, con el fin de tener documentado el sistema.

2.- Creación física de la base de datos. Este define como será la estructura física de la base de datos y presentarla al DBMS reflejando satisfactoriamente su funcionamiento, y cargando la base de datos. Aquí no existe algún procedimiento matemático que juzgue si el modelo es o no correcto.

3.- Conversión del conjunto de datos. Aquí se definen reglas de conversión del conjunto de datos existentes y cuales serán o no convertidos.

4.- Integración de aplicaciones en la nueva base de datos. Se debe contemplar la facilidad de poder expander la estructura física para integrar nuevas aplicaciones a la nueva base de datos es decir la flexibilidad de la base de datos.

5.- Fase de operaciones. En esta fase se establecen los procedimientos de control, seguridad,

acceso, recuperación y los criterios de funcionamiento.

6.- Fase de crecimiento y mantenimiento. Esta fase esta ligada a la fase 4 ya que se enfrenta a los cambios que muchas veces el diseño de la base de datos es relativamente flexible y su mantenimiento sea cada vez más difícil.

## 1.2 TIPOS DE ESTRUCTRUCTURAS DE BASE DE DATOS.

Los sistemas de base de datos se clasifican de acuerdo al enfoque que adopten y estos pueden ser de tres tipos de estructuras:

- Estructura jerárquica o de árbol.
- Estructura de red o plex.
- Estructura relacional.

Para poder explicar las diferentes estructuras lo haremos por medio de un ejemplo. Veamos en la figura 1.1 un archivo de ventas, en el cual suponemos que tenemos una compañía que maneja diferentes vendedores (Villagrana,Corominas y Kuri), y cada uno de ellos vende cierto tipo de productos (tuercas, tornillos y pernos) en una determinada cantidad, que se encuentran clasificadas por sus propiedades como

el color y peso, que sus ventas se hacen en diferentes ciudades. Supongamos que cada vendedor y producto tienen un número y un nombre únicos, a los que llamaremos V1, V2 y V3 (vendedores) y P1, P2 y P3 (productos).

Archivo de Ventas

| Vendedor | Edo | Cd. Vend. | Cant. | Producto | Color | Peso | Cd. Prod. |
|---|---|---|---|---|---|---|---|
| Villagrana | 80 | Londres | 300 | Tuercas | Rojo | 12 | Londres |
| Villagrana | 20 | Londres | 200 | Perno | Verde | 17 | Paris |
| Villagrana | 20 | Londres | 400 | Tornillo | Azul | 17 | Roma |
| Corominas | 10 | Paris | 300 | Tuercas | Rojo | 12 | Londres |
| Corominas | 10 | Paris | 400 | Perno | Verde | 17 | Paris |
| Kuri | 30 | Paris | 200 | Perno | Verde | 17 | Paris |

*Fig. 1.1 Datos de muestra.*

**ESTRUCTURA JERARQUICA O DE ARBOL.**

Este tipo de estructura define relaciones tipo árbol entre sus registros sobre la base de datos, es decir, las unidades de datos se estructuran en niveles multiples que representan gráficamente un árbol y el nivel superior se le llama **'raíz'** o **'padre'**.

Así en nuestro, ejemplo, el usuario ve solo cuatro árboles individuales u ocurrencias jerárquicas, una para cada producto. El tipo de registro en la cima es el registro raíz o padre, en este caso sería el registro de producto, y los registros de abajo son

los hijos, en este caso serían los registros de vendedores. En general, el padre puede tener cualquier número de hijos y éstos a su vez cualquier número de hijos, y así sucesivamente, hasta cualquier número de niveles.

La figura 1.2 muestra la estructura jerárquica de nuestro ejemplo, el cual solo contiene dos niveles.

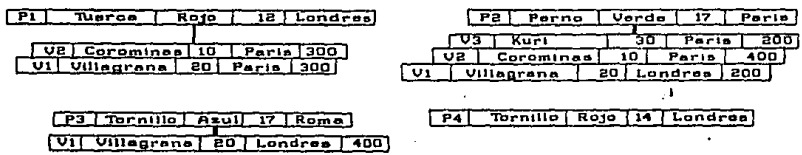



*Fig. 1.2 Estructura jerárquica (datos de muestra).*

Las operaciones de actualización de esta estructura contienen ciertas anomalías como:

**Insertar.** No es posible, insertar un vendedor nuevo V4 sin haber introducido un producto ficticio o insertárlo hasta que éste haya vendido un producto.

**Suprimir.** Aquí será costoso si suprimimos la remesa que conecta al vendedor V3 y el producto P2, ya que hay que suprimir la ocurrencia del vendedor V3 lo cual causa que se pierda la información del

vendedor V3, pues es la única ocurrencia de éste vendedor.

**Actualizar.** Si necesitamos cambiar la ciudad del vendedor V1, es necesario buscar en todos los registros del vendedor V1 y cambiar la ciudad y si nos faltará cambiar algún registro, tendríamos la posibilidad de tener inconsistencia en la base de datos.

**ESTRUCTURA DE RED O PLEX.**

Una estructura de red o plex, es una relación donde un hijo puede tener más de un padre, lo que permite una correspondencia múltiple entre los elementos de datos de forma más directa, pero se complican aún más, cuando las relaciones son también entre los mismos elementos, por ejemplo la relación de vendedor y producto, aquí el vendedor puede tener más de un producto que venda, pero a su vez el producto puede tener varios vendedores. La figura 1.3 muestra la representación esquemática de esta estructura de red con el ejemplo que manejamos desde un principio.

En las operaciones de actualización no se representan anomalías como en la estructura

jerárquica, sin embargo la programación no es tan directa como podría serlo.

**Insertar.** Para insertar datos relativos a un nuevo vendedor S4, sencillamente se crea una ocurrencia del registro de vendedores, al principio no habrá un conector hacia un producto, y su apuntador es a sí mismo.

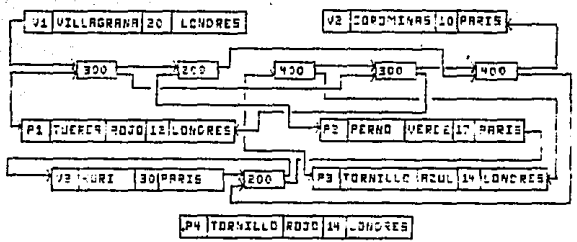



*Fig. 1.3 Estructura de red (datos de muestra).*

**Suprimir.** Para suprimir la remesa que conecta al vendedor V3 y producto P2 se suprime la ocurrencia del conector que liga a este vendedor y el producto, y las dos cadenas que intervienen necesitarán ajustarse de manera adecuada.

**Actualizar.** Se puede cambiar la ciudad del vendedor V1 sin ningún problema de busqueda y sin la posibilidad de inconsistencia porque la ciudad del

vendedor V1 aparece precisamente en un solo sitio de la estructura.

ESTRUCTURA RELACIONAL.

Esta estructura "se basa en la suposición de que los datos estan potencialmente relacionados entre sí"[4]. Estas relaciones se expresan en forma de tablas, en nuestro ejemplo las organizaremos en tres tablas: tabla de Vendedores (V), tabla de productos (P) y tabla de remesas (VP). La tabla V contiene para cada vendedor, su nombre, el código de estado, la localización del vendedor; la tabla P contiene para cada producto, su número, su nombre, su color, su peso y la localización donde se almacenan; y por último la tabla VP contiene para cada remesas, un número de vendedor, un número de producto y cantidad vendida ver figura 1.4.

En las operaciones de actualización ya no existen los problemas que se presentaron con las dos estructuras anteriores, ya que el manejo se hace más fácilmente.

**Insertar.** Dar de alta al vendedor nuevo V4 en la tabla V.

**Suprimir.** Suprimir la remesas que conecta al vendedor 3 y el producto 2 se realiza solamente

eliminando el renglón de la tabla VP, donde el número de vendedor es V3 y número de producto es P2.

TABLA DE REMESAS (VP)

| #V | #P | CPT |
|---|---|---|
| V1 | P1 | 300 |
| V1 | P2 | 200 |
| V1 | P3 | 400 |
| V2 | P1 | 300 |
| V2 | P2 | 400 |
| V3 | P2 | 200 |

TABLA DE VENDEDORES (V)

| #V | NombV | Edo | Ciudad |
|---|---|---|---|
| V1 | Villagrana | 20 | Londres |
| V2 | Corominas | 10 | Paris |
| V3 | Kuri | 30 | Paris |

TABLA DE PRODUCTOS (P)

| #P | NombP | Color | Peso | Ciudad |
|---|---|---|---|---|
| P1 | Tuerca | Rojo | 12 | Londres |
| P2 | Perno | Verde | 17 | Paris |
| P3 | Tornillo | Azul | 17 | Roma |
| P4 | Tornillo | Rojo | 14 | Londres |

*Fig. 1.4 Estructura relacional (datos de muestra).*

**Actualizar.** El vendedor 1 se translada a otra cuidad, entonces, solo hay que cambiar en la tabla V la ciudad donde el número de vendedor es V1.

## 1.3 ESTRUCTURA DE DATOS RELACIONADOS.

Muchos de los trabajos desarrollados en el mundo de DBMS están mejorando los productos relacionales. Algunos de los productos del mercado como "relacionales", son extremadamente limitados.

Ejemplos de la buena tecnología de mainframe relacional son DB2 de IBM, Oracle de Oracle Co.,

DATACOM de Aplied Data Research y FOCUS de Information Builder. Esta es una pequeña lista y ninguno de estos paquetes pudiera conocer una definición pura de un sistema relacional "verdadero". Sin embargo, son las mejores que hay hoy en día.
Es interesante notar que virtualmente todos los productos de DBMS de computadores personales están basados en un sistema relacional. Esto es, porque es conceptualmente el más fácil de manejar, el más flexible y suministra a los usuarios inexpertos en desarrollo de trabajos aplicados en menor tiempo.

**TABLAS.** Una base de datos relacional consiste en uno o más archivos planos bidimensionales llamados tablas o relaciones.

Cada tabla contiene renglones y columnas análogas a un registro de campos de datos no relacional, ver la figura 1.5.

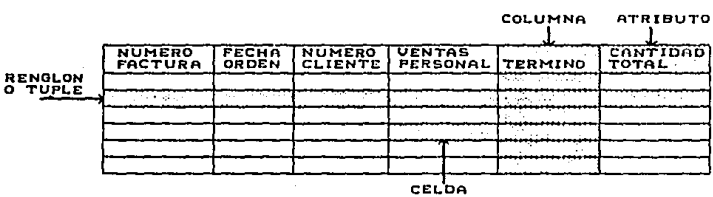



*Fig. 1.5 Terminología Relacional.*

A cada renglón se le llama **tuple** que significa "un grupo de". Si se tiene en el tuple dos columnas, entonces, se dice que tenemos una relación de grado binario, si es de tres es de grado terciario y si es de n entonces es de grado enesímo.

Un tuple es un conjunto de valores ordenados análogos a una entidad que contiene todas o parte de la información almacenada de una persona, lugar o cosa de interés a nuestra aplicación.

Las columnas son llamadas **columnas** y representan el valor de un atributo específico, ver figura 1.5.

Un **dominio** es el conjunto de todos los valores posibles para un atributo, y no necesariamente representan alguna cualidad inherente de los atributos. El dominio cambia conforme cambian las condiciones del negocio.

Una **celda** representa al dato situado en la intersección de una columna y un renglón, ver figura 1.5.

La **cardinalidad** es el número de renglones dados en una tabla a un punto y tiempo específico. Esto es equivalente al número de registros contenidos en un archivo en un tiempo. La cardinalidad es muy importante para el diseño de nuestra base de datos,

así conocer que tan grande llegará a ser nuestra base de datos.

Las siguientes reglas deben satisfacer a una tabla.

- Cada columna representa un atributo.

- La secuencia de atributos debe ser idéntica en todos los renglones.

- Cada renglón debe ser único , esto es, que dos renglones de la misma tabla deben diferir al menos en el valor de una celda.

- Todos los renglones deben de estar completos. Así, seis valores deben estar presentes en cada renglón para una relación de grado 6. Cada valor debe de ser parte del dominio del atributo al que pertenece.

- No se permite la repetición de grupos. Esto es, solo un valor permitido para una celda. Por ejemplo, si dos fechas son necesarias dos atributos deben ser definidos.

- La secuencia de renglones no puede ser importante para la aplicación. Por ejemplo no podemos definir y depender de un total de renglones que siempre siguen renglones de transacciones detalladas que nos dan una cantidad total de información.

**LLAVES.** Una llave es un atributo o colección de atributos que identifican a un tuple en una tabla específica, se le conoce como **llave primaria**. Existe otro tipo de llaves llamadas **llave extranjera** y tiene la función de llave primaria en una tabla pero también aparece en otra tabla donde no es una llave primaria. En la figura 1.6 el número de cliente es una llave primaria en la tabla de clientes y una llave extranjera en la tabla de ordenes. Así también el atributo vendedor es ambas llaves, primaria y secundaria.

Las llaves extranjeras son importantes porque con ellas se establecen las relaciones entre registros, enlazándolos para tomar datos juntos desde una o más tablas. Las llaves extranjeras tienen la misma función en una base de datos como los apuntadores la tienen en los modelos no relacionales.

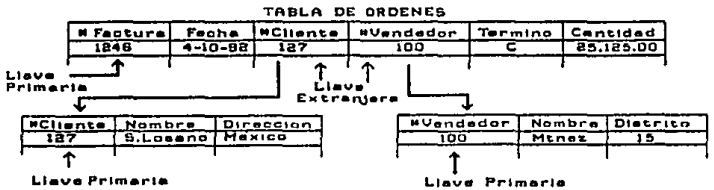



*Fig. 1.6 Llaves Primarias y Extranjeras.*

Una llave que consiste de multiples atributos es llamada llave compuesta concatenada o combinada. El número de factura y número de parte forman la llave de la tabla Cantidad de la figura 1.7. Una llave individual de atributo es algunas veces llamada una llave simple ordenada para distinguirla de una llave compuesta.

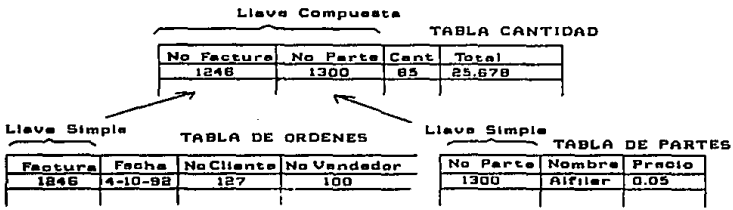



*Fig. 1.7 Llaves Compuestas y Simples.*

**OPERADORES RELACIONALES.**

Si queremos examinar los datos en una base de datos relacional, usamos tres operaciones genéricas que son ilustradas en la figura 1.8 y son:

- **Selección.** Se usa para extraer los valores específicos de uno u varios atributos de varios renglones de una tabla.

- **Proyección.** Se usa para extraer los valores específicos de una o varios atributos removiendo renglones con redundancia.

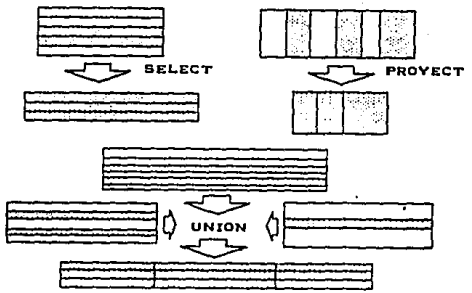



*Fig. 1.8 Operadores Relacionales.*

- **Unión.** Esta operación combina datos de dos o más datos uniendo renglones de ambas tablas.

## 1.4 ARQUITECTURA DEL SISTEMA RELACIONAL.

Un sistema relacional soporta una base de datos relacional donde todos los usuarios perciben todos los datos en forma de tablas. y todo acceso a esta base de datos se hace por medio de un sublenguaje de datos. Los diferentes sublenguajes son por ejemplo

MAGNUM de TYMSHARE, MRDM de HONEYWELL con MULTICS y NOMAD de NATIONAL CSS., e IBM que utiliza el sublenguaje de datos llamado SQL (Structure Query Language) este lenguaje no sólo ofrece funciones de recuperación sino también una diversidad de operaciones de actualización. El SQL se puede utilizar desde una terminal en línea o desde un programa de aplicación en línea o en batch, escrito en cobol o en PL/1. En general sus operadores funcionan casi siempre en términos de conjuntos (comparable al algebra relacional) y no de registros independientes.

La figura 1.9 nos muestra como un usuario individual percibe un sistema relacional. Aquí el usuario puede ver una tabla o una vista.

"Una vista es una tabla que no tiene existencia por derecho propio sino que se deriva de una o varias tablas de base"[5].

En el nivel interno cada tabla se representa por medio de un archivo de almacenamiento distinto. Cada archivo del almacenamiento puede tener un número de indices asociados a él.

El SQL influye el DDL (Data Definition Language) que como su nombre lo indica es un lenguaje de

definición de datos, y el DML (Data Manangement Language) que es el lenguaje de manipulación de los

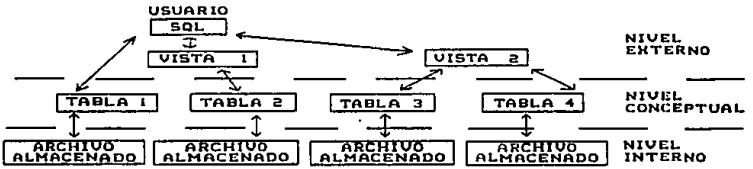



*Fig. 1.9 El Sistema Relacional visto por un usuario individual*

datos. El DDL define los objetos en el nivel externo las vistas en el nivel conceptual las tablas, y en el nivel interno los índices. El DML es el que opera las tablas y vistas,, como operaciones de recuperación y operaciones de actualización.

# C A P I T U L O 2.

## 2.1 DISEÑO LOGICO DE LA BASE DE DATOS.

Para desarrollar una base de datos que satisfaga las necesidades de información actuales y futuras, se debe diseñar un modelo conceptual, y después el modelo lógico. El modelo conceptual refleja las entidades y sus relaciones y está basado en nuestro caso, en las necesidades del Laboratorio de Química.

El primer paso para el diseño conceptual es necesario realizar el análisis de los datos, por lo que se procedió al levantamiento de la información para recolectar datos.

Se entrevistó al responsable del Laboratorio de Química Orgánica como muestra representativa, para conocer las actividades que se desarrollan en la administración del laboratorio, y se obtuvo que la administración de los laboratorios depende de tres controles principales, los cuales son; los horarios, los alumnos y las prácticas.

Explicaremos estos tres controles como se manejan en el laboratorio de Química, comenzando por el horario, y que este lo asigna la Coordinación o Sección de Química el cual lo envía al responsable

del Laboratorio, el horario se realiza en función de los laboratorios disponibles y el número de grupos que se abran.

Respecto a los alumnos, diremos que al comienzo de cada semestre acuden a inscribirse al laboratorio según el horario que más les convenga. Esta inscripción se realiza por una tarjeta de archivo y que es única por cada alumno, y donde se lleva el control de las calificaciones de los laboratorios cursados. Por otro lado, se llevan unas listas del material adeudado por los alumnos, y si algún alumno aparece en esta lista, no podrá inscribirse al siguiente laboratorio hasta que reponga dicho material. Estos reportes sirven también para dar a los pasantes o alumnos activos comprobantes o constancias de NO ADEUDO de material cuando lo soliciten para trámites escolares. Los reportes de adeudo se van controlando en cada práctica que se realiza por medio de papelería como:

- Solicitud de material por equipo o grupo
- Vales de adeudo de material.

Esta parte es representada en el diagrama de burbuja que muestra la figura 2.1 y que se intrerpreta siguiendo la numeración de las flechas.

Por último el control de las prácticas se lleva por medio de un documento llamado programa tradicional de prácticas, en el que consta de una lista de las prácticas que se deben de llevar en cada semestre y por cada carrera. Además de tener un libro de control de prácticas, donde se lleva el control de las prácticas realizadas por cada grupo.

Y por último una carpeta de prácticas que contiene, todo el material y sustancias necesarias para realizar una práctica determinada. De igual forma que en los alumnos, en la figura 2.2, se muestra el flujo

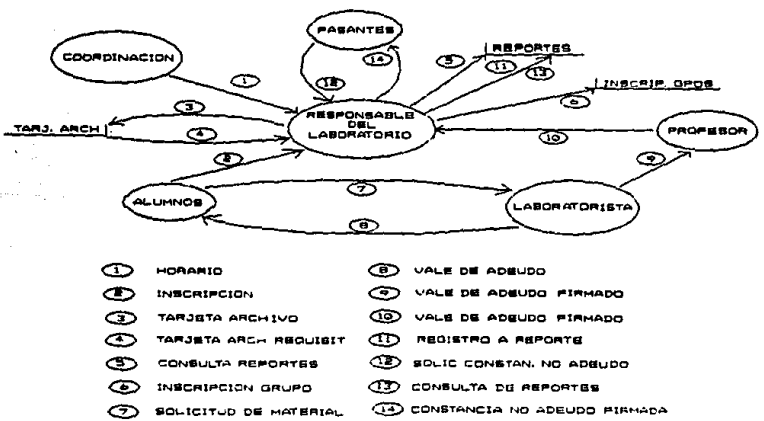



*Fig. 2.1. Representación de la inscripciones de los alumnos al laboratorio.*

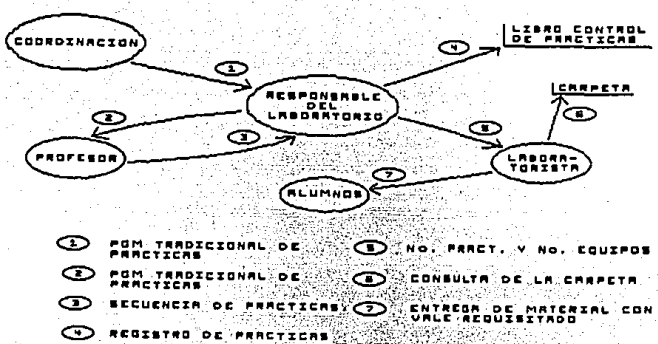



*Fig. 2.2. Representación del control de las prácticas en el laboratorio de Química.*

administrativo que se lleva a cabo en el laboratorio de química.

Como segundo paso para el diseño conceptual es necesario seguir el proceso de normalización el cual consiste en "agrupar a los campos de datos en tablas que representen a las entidades y sus relaciones", de tal forma que nos permita asegurar que el modelo conceptual de la base de datos funcionará. El proceso de normalización, se basa en el concepto de formas normales, dicho proceso se lleva a cabo en tres etapas llamadas formas normalizadas y que a continuación describimos.

## 2.2 NORMALIZACION DE LAS ENTIDADES DE DATOS.

**PRIMERA FORMA NORMALIZADA.**

Consiste en transformar los campos de datos a una tabla de dos dimensiones, eliminando así las ocurrencias repetidas de los campos de datos y obtener un archivo fijo. Y la tabla debe tener las siguientes características:

- Cada columna represente un atributo
- La secuencia de atributos es identica
- Cada renglón es único
- El valor de cada llave esta presente
- No hay grupos repetitivos
- La secuencia de renglones no es importante
- A una llave unica es identificada para cada renglón
- Todos los atributos son funcionalmente dependientes en todo o parte de la llave.

Por tanto de la información levantada se conjuntaron todos los documentos que se utilizan en el proceso administrativo del laboratorio, elaborándose una tabla como muestran las figuras 2.3a y 2.3b, donde las columnas son los documentos y los renglones son los campos de datos, y las celdas marcadas con una 'X' son los datos que tiene cada

| DOCTO / CAMPO | HORARIO | TARJETA ARCHIVO | CNTL PRACT TRADICIONAL | LIBRO PRACT | CARPETA | SOLIC MATER EQUIP | SOLIC MATER GRUPO | VALE ADEUDO MATER | REPTE DE ADEUDO | REPTE SEMES ADEUDO | CONSTAN NO ADEUDO | LISTA DE CALIF | COMPRO BANTE ALUMNO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SEMESTRE | X | X | | | | | | | X | X | | X | X |
| DIA DE SEMANA | X | | | | | | | | | | | X | |
| NO LABORAT | X | | | | | | | | | | | | |
| CVE GRUPO | X | | | | | | X | X | | | | | |
| HORA INICIO | X | | | X | | | | | | | | | |
| HORA TERMINA | X | | | X | | | | | | | | | |
| NOMBRE PROF | X | | | X | | X | X | X | X | | | | |
| NO CUENTA | | X | | | | | | | | X | X | | X |
| CARRERA | X | X | X | | | | | X | | | | X | |
| NO CREDENCIAL | | X | | | | | | | | | | | |
| DIRECC ALUM | | X | | | | | | | | | | | |
| TEL ALUM | | X | | | | | | | | | | | |
| T SANGRE ALUM | | X | | | | | | | | | | | |
| MATERIA LAB | X | X | X | | | | | X | | | | X | X |
| CALIF ALUM | | X | | | | | | | | | | X | X |
| OBSERV ALUM | | X | | | | X | | | | | | | |
| NOMBRE ALUM | | X | | | | X | X | X | X | X | X | X | X |
| NO SECUENCIAL | | | X | | | | | | | | | | |
| NOMBRE PRACT | | | X | X | X | X | X | | | | | | |

*Fig 2.3a Tabla de documentos del laboratorio de Q.O.*

| DOCTO / CAMPO | HORARIO | TARJETA ARCHIVO | CNTL PRACT TRADICIONAL | LIBRO PRACT | CARPETA | SOLIC MATER EQUIP | SOLIC MATER GRUPO | VALE ADEUDO MATER | REPTE DE ADUEDO | REPTE SEHES ADUEDO | CONSTAN NO ADEUDO | LISTA DE CALIF | COMPRO BANTE ALUMNO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CVE PRACTICA | | | X | | X | | | | | | | | |
| CVE MATERIA | | | | X | | | | | | | | | |
| FECHA REALIZ | | | | X | | | X | | | | | | |
| NO PRACTICA | | | | X | | X | X | | | | | | |
| RESULT PRACT | | | | X | | | | | | | | | |
| OBSERV PRACT | | | | X | | | | | | | | | |
| CANT MAT EQPO | | | | | X | | | | | | | | |
| NOMBRE MAT | | | | | X | | X | X | X | X | X | | X |
| CANT MAT GPO | | | | | X | | X | | | | | | |
| NOMBRE REACT | | | | | X | | | | | | | | |
| CONC REACTIVO | | | | | X | | | | | | | | |
| FECHA PROMESA | | | | | | | | X | | | | | |
| NOMBRE SECC | | | | | | | | | | | X | | |
| FECHA VENCIM | | | | | | | | | | | X | | |
| FECHA EMISION | | | | | | | | | | | X | | |
| INSCRIPCION | | | | | | | | | | | | X | |
| % APROBADO | | | | | | | | | | | | X | |
| % REPROBADO | | | | | | | | | | | | X | |
| % NO ACREDIT | | | | | | | | | | | | X | |

Fig 2.3b Tabla de documentos del laboratorio de Q.O (continuación).

documento. Claro está, que ésta tabla es una tabla no normalizada, por lo que comenzaremos por crear varias tablas de tal forma de llevar a cabo la primera forma normalizada.

Por lo tanto, si los datos que tenemos en las figuras 2.3a y 2.3b, los tomamos como los atributos formamos un archivo fijo sin ocurrencias repetidas, y así transformamos la tabla no normalizada en una tabla normalizada, a la que llamaremos tabla general, esta tabla la muestra la figura 2.4: en donde los campos fecha de promesa, nombre de la sección, fecha de vencimiento, fecha de emisión, no los tomaremos como los atributos, ya que estos son campos de datos que se obtendrán del usuario, cuando este quiera obtener un reporte.

**SEGUNDA FORMA NORMALIZADA.**

Se dice que una tabla está en la segunda forma normalizada cuando todo atributo que no es llave es dependiente de la llave primaria.

Entendamos por dependencia a la relación entre los atributos de como el valor de uno fija o determina el valor del otro. La dependencia es un concepto importante porque nos hace capaces de descomponer datos en sus formas más eficientes.

| SEMESTRE | DIA | NoLAB | CVE GPO | HR INI | HR TER | NOMBRE PROFESOR | No CTA | CARRERA |
|---|---|---|---|---|---|---|---|---|
| 92-I | LUN | L123 | 1204 | 09:00 | 13:00 | GABRIELA. ZENTENO | 92047514 | QFB |

| No CRED | DIRECCION ALUM | TELEFONO | TPO S | MAT LAB | CALIF | OBSERVAC | NOMBRE ALUM |
|---|---|---|---|---|---|---|---|
| 000838 | LERDO 175 ED A-302 | 5971254 | O | Q ORG I | MB | | RAQUEL CRUZ |

| No SEC | NOMBRE PRACTICA | CVE PRACT | CVE MATE | FEC REAL | No PRAC | RESULTADO |
|---|---|---|---|---|---|---|
| 2 | OBTENCION DE ALQUINOS | 1 | QOI | 92-10-20 | 3 | |

| OBSERV PRACT | CANT MATE | NOMBRE MATE | CONCENT | NOMBRE REACTIVO | INSCRIP |
|---|---|---|---|---|---|
| | 2 | PIPETA 10ml | 0.1M | AC. CLORHIDRICO | S |

*Fig 2.4 Tabla General.*

Existen diferentes tipos de dependencias las cuales son:

**Dependencia Funcional.** Esta existe cuando un valor único un atributo puede ser siempre determinado si conocemos el valor de otro, de nuestros atributos si conocemos la clave del grupo podemos conocer la hora de inicio o la hora de terminación, en cambio si vemos la relación contraria si conocemos la hora de inicio no podemos conocer la clave del grupo, ya que varios grupos pueden comenzar a la misma hora. La dependencia funcional se ilustra por medio de una

flecha en sentido de la variable independiente (X) a la variable dependiente (Y).

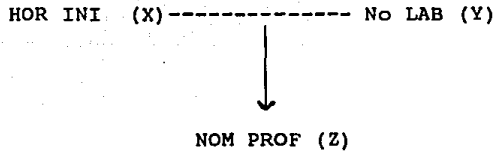
CVE GPO (X) -------> HOR INI

**Dependencia Total.** Esta dependencia es de tipo especial. Supongamos que un individuo solamente debe tener un número de cuenta y un número de credencial de laboratorio, entonces para encontrar a ésta persona, podemos buscarla por su número de credencial o su número de cuenta, por lo tanto estas dos variables tienen una dependencial total. Esta dependencia se ilustra con una flecha en los dos sentidos. Así el número de cuenta es totalmente dependiente del número de credencial y viceversa.

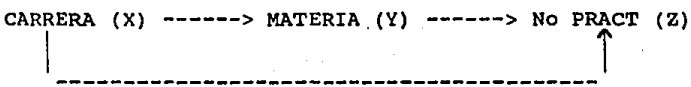
No CTA <-------> No CRED

**Dependencia Completa.** Esta dependencia ocurre cuando un atributo es siempre dependiente de al menos de otros dos atributos. Así por la relación entre la hora de inicio, el número del laboratorio y nombre del profesor, nos damos cuenta como es esta dependencia, ya que si sabemos la hora de inicio no podemos conocer el nombre del profesor a menos que

conozcamos el número del laboratorio. Similarmente no podemos conocer el nombre del profesor si solo sabemos el número del laboratorio. La dependencia completa la ilustramos abajo, donde **X** y **Y** son variables independientes y **Z** es la variable dependiente.

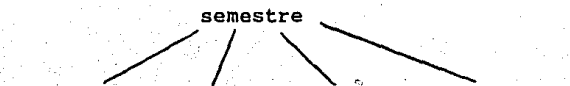



**Dependencia Transitiva.** Esta dependencia ocurre cuando la variable **Y** depende de **X** y la variable **Z** depende de **Y**, por consiguiente **Z** depende de **X**. Como ejemplo decimos que, la materia depende de la carrera y el número de prácticas dependen de la materia, por lo tanto si conocemos la materia y el número de prácticas conocemos la carrera. Abajo ilustramos nuestra dependencia transitiva.

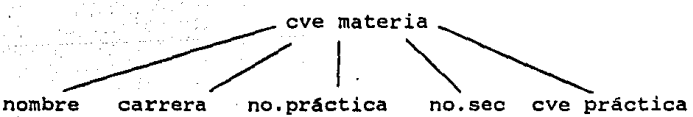


Conociendo como son las diferentes dependencias que existen entre los atributos, comenzaremos a analizar nuestra tabla General y las relaciones posibles;

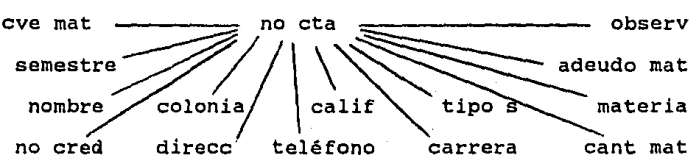
De esta manera el semestre tiene las siguientes relaciones:



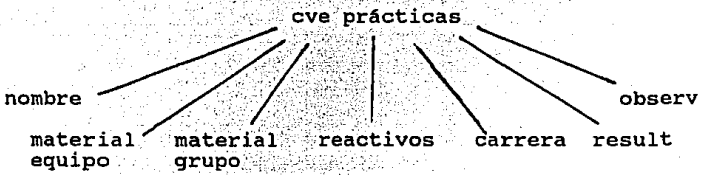

La clave de la materia tiene las siguientes relaciones:



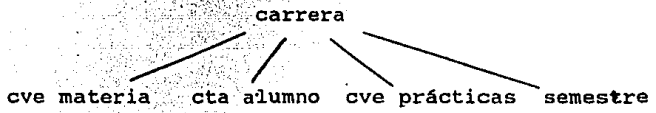

El alumno tiene un número de cuenta único y sus relaciones son:



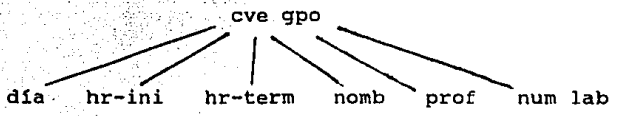
Las prácticas se relacionan con:




La carrera está relacionada con:




La clave del grupo tiene las relaciones:




Así de esta segunda forma normalizada se obtuvieron las siguientes tablas que se muestran en la figura 2.5a y 2.5b.

- Tabla horario
- Tabla prácticas
- Tabla reactivos
- Tabla control de prácticas
- Tabla material
- Tabla libro de prácticas
- Tabla alumnos

TABLA HORARIO

| LLAVE | | ATRIBUTOS | | | | |
|---|---|---|---|---|---|---|
| SEM | CVEGPO | DIA | NoLAB | HORIN | HORTE | NOMPROF |
| 92-1 | 1204 | LUN | L123 | 09:00 | 13:00 | ROCIO GARCIA |
| 92-1 | 1205 | LUN | L123 | 15:00 | 19:00 | RAQUEL CRUZ |
| | | | | | | |

TABLA CONTROL DE PRACTICAS

| LLAVE | | | | ATRIBUTOS |
|---|---|---|---|---|
| SEM | CARRERA | CVEMATE | CVEPRAC | NOPRCT |
| 92-1 | IQ | QOIII | 1 | 3 |
| 92-1 | IQ | QOIII | 2 | 6 |
| | | | | |

TABLA MATERIAL-PRACTICAS

| LLAVE | | ATRIBUTOS | |
|---|---|---|---|
| CVE PRACT | CVETML | CVEEQ | CANT |
| 1 | MA1 | E | 1 |
| 1 | FI | G | 2 |
| 1 | AS1 | R | 3 |
| | | | |

TABLA DE PRODUCTOS

| LLAVE | ATRIBUTOS |
|---|---|
| CVE PROD | NOMB PROD |
| ALQI | ALQUINOS |
| BENC | BENCENO |
| AN1 | AC. NITROSO |
| | |

*Fig 2.5a Tablas Resultantes de la 2a.forma normalizada.*

**TERCERA FORMA NORMALIZADA.**

Se dice que una tabla está en la tercera forma normalizada si la tabla está en una segunda forma normalizada, y todas las dependencias transitivas

TABLA DE MATERIAL

| LLAVE | ATRIBUTOS |
|---|---|
| CVEMTL | NOMBRE MATERIAL |
| MT1 | MATRAZ ERLENM 100ml |
| PI1 | PIPETA 10ml |

TABLA DE REACTIVOS

| LLAVE | ATRIBUTOS |
|---|---|
| CVEREA | NOMBRE MATERIAL |
| AC1 | AC. CLORHIDRICO |
| AS1 | AC. SULFURICO |

TABLA LIBRO DE PRACTICAS

| LLAVE | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| CVE SEM | CVE MAT | CVEPRCT | NO SEC | CVE GPO | NO PRACT | FEC REAL | RESULT | OBSERVACIONES |
| 92-I | 00111 | 1 | 3 | 1204 | 6 | 92-10-23 | | |
| 92-I | 00111 | 2 | 4 | 1204 | 1 | 92-10-30 | NEGAT | FALTO AGUA |

TABLA DE ALUMNOS

| LLAVE | ATRIBUTOS | | | |
|---|---|---|---|---|
| NO CTA | NO CREDE | NOMB ALUM | DIRECC | COL |
| 91400216 | 000021 | ANA BELEN MANZO SAIZ | CDA ZARAGOZA 213-4 | BALBUENA |
| 91456781 | 000078 | CARLOS DEL PRADO A. | DEPORTES 34 | ARBOLEDAS |

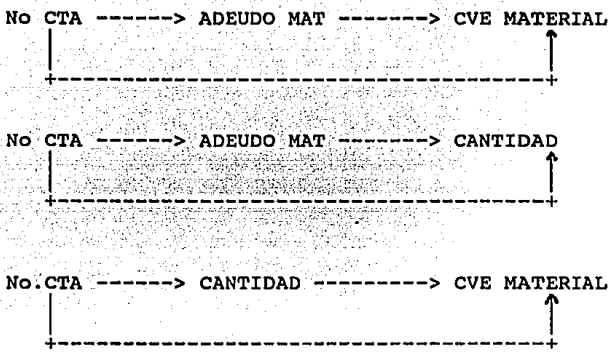
TABLA DE ALUMNOS (continuación)

| ATRIBUTOS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| TEL | TIP S | CVEMATE | CALIF | CALIF2 | CARRERA | ADEUDA | CANT1 | FEC1 | CVEMTL1 |
| 5234567 | O | 00111 | B | | QFB | SI | 1 | 92-03-23 | MA1 |
| 3249219 | O | COIV | MB | | Q | | | | |

*Fig 2.5b Tablas Resultantes de la 2a.forma normalizada (continuación).*

entre los atributos que no son llaves han sido eliminadas.

Así, tenemos una dependencia transitiva entre el adeudo de material y la clave de material con el número de cuenta.




De igual forma sucede con los atributos nombre del alumno, dirección y número de cuenta, por lo que nos vemos en la necesidad de separar los datos propios del alumno (dirección, colonia, teléfono, etc.) y los datos que pertenecen a el adeudo de material, de ésta manera creamos una tabla a la que llamaremos ADEUDOS, con el número de cuenta, la materia y el semestre entre ellas existe una dependencia transitoria, por lo que creamos una tabla que llamaremos tabla de CALIFICACIONES, en la figura 2.6 mostramos como la tabla ALUMNOS finalmente queda, junto con las dos nuevas tablas.

Todas nuestras relaciones las podemos representar gráficamente como sigue:

1.- NO.CTA ---> NO.CRED,NOMBAL,DIRECC,COL,TEL,TIPS,
CARR

TABLA DE ALUMNOS

| LLAVE | ATRIBUTOS | | | |
|---|---|---|---|---|
| NO CTA | NO CREDE | NOMB ALUM | DIRECC | COL |
| 91400216 | 000021 | ANA BELEN MANZO SAIZ | CDA ZARAGOZA 213-4 | BALBUENA |
| 91456781 | 000078 | CARLOS DEL PRADO A. | DEPORTES 34 | ARBOLEDAS |

TABLA DE ALUMNOS (continuación)

| ATRIBUTOS | | |
|---|---|---|
| TEL | TIP S | CARRERA |
| 5234567 | O | QFB |
| 3249219 | O | Q |

TABLA DE CALIFICACIONES

| LLAVE | | | ATRIBUTOS | |
|---|---|---|---|---|
| NO CTA | CVEMATE | CALIF | INSCRIP | OBSERV |
| 91400216 | QOIII | B | S | |
| 90254681 | QOII | MB | S | OYENTE |

TABLA ADEUDO

| LLAVE | | | ATRIBUTOS | |
|---|---|---|---|---|
| SEM | NO CTA | CVEMTL | CANT | FECADEUDA |
| 92-1 | 90542878 | MA1 | 1 | 24-05-91 |

*Fig 2.6. Tabla alumnos,adeudos y calificaciones de la 3a Forma normalizada.*

2.- NO.CTA,CVEMAT,CALIF ------> INSCRIP,SEM,OBSERV, CVEGPO

3.- SEM,NO.CTA,CVEMAT ------> CANT,FECADEUDO

4.- CVEPRCT,CVEMTL -----> CVEEQ,CANT

5.- CVEMTL -----> NOMBMTL, COSTMTL

6.- CVEREA -----> NOMBREA,COSTREA

7.- CVEPROD -----> NOMBPRO,COSTPRO

8.- SEM,CARR,CVEMTE,CVEMTL -----> NOSEC

9.- SEM,CVEMATE,CVEPRCT -----> NOSEC,CVEGPO,NOPRCT, FECREAL,RESULT,OBSERV

10.- SEM,CVEGPO -----> DIA,NOLAB,HRINI,HRTERM,
NOMBPROF

11.- CVEPRCT -----> NOMBPRCT

12.- CVEMATE -----> NOMBMATE

13 .-SEM -----> INISEM,FINSEM

Cada relación para la cual la clave primaria consta de un campo representa una entidad. Las relaciones 1, 5, 6, 7, 11, 12 y 13, representan a ALUMNOS, MATERIAL, REACTIVOS, PRODUCTOS, PRACTICAS, MATERIAS, SEMESTRE respectivamenté. Todas las entidades de este tipo estan situadas en el nivel 1. En la figura 2.7 nos muestra como esta el nivel 1, Los campos de datos estan escritos dentro del cuadro y las llaves estan subrayadas.

Las relaciones con una llave primaria de dos elementos representan a las relaciones entre dos entidades. Las relaciones de este tipo están situadas en el segundo nivel. Si una parte de la llave no esta representada como una entidad se generará una nueva relación de entidad. Como sucede en la relación 10, SEM+CVEGPO, que nos representan las entidades SEMESTRE y GRUPOS, entonces se crea al nivel 1 la entidad GRUPOS. De ésta manera obtenemos nuestro modelo lógico que en la figura 2.7 cada cuadro es una

entidad y nos representa una tabla. Y decimos por ejemplo, en nuestra relación SEMESTRE, tiene los

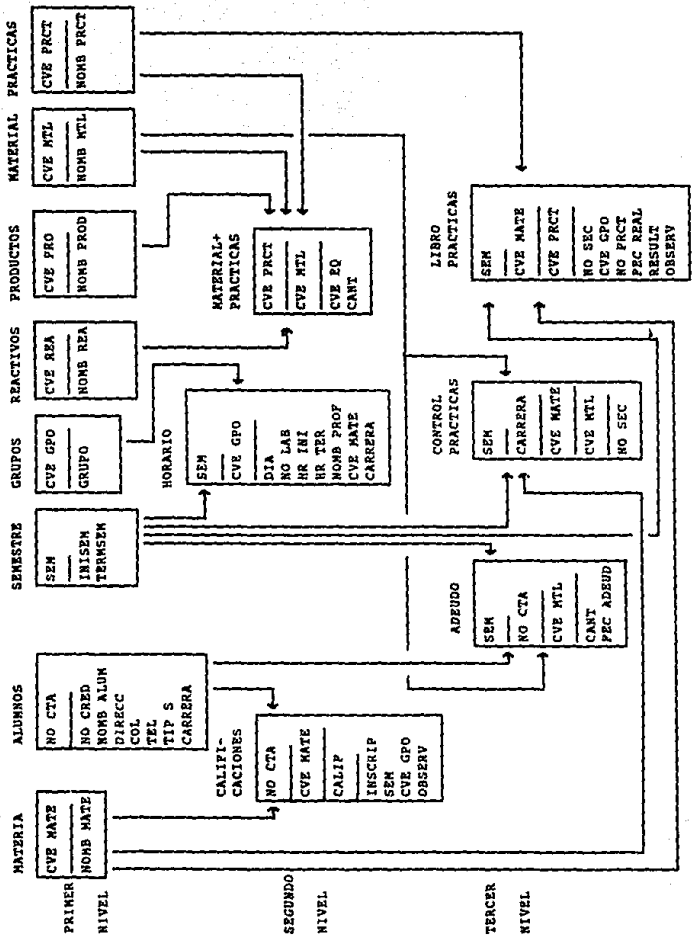



*Fig 2.7.Diagrama lógico.*

atributos SEM (semestre), INISEM (fecha de inicio del semestre) y TERMSEM (fecha de terminación del semestre), siendo SEM la llave primaria. Y dado el valor de SEM sólo existe una cadena con ese valor.

## 2.3 MODELO DE DATOS FISICOS.

En esta sección veremos como el programador de aplicaciones distribuye los datos en las unidades de almacenamiento. Nombraremos a continuación algunos de los factores más importantes que afectan al modelo físico, y estos son:

**Economía del espacio.** El objetivo de este factor es maximizar la cantidad de espacio de datos que se almacenan en diversos volumenes. Se hace de dos maneras una es agrupando registros lógicos en un registro físico creando bloques de registros, y la otra forma es compactando datos por ejemplo el dato de fecha, que se escribe así; 12 de Enero de 1993, al compactarse sería 12-ENE-1993 y aún más 12-01-93.

**Minimizar redundancia.** El buscar la manera de no tener duplicidad de datos pudiendolos manejar en un solo archivo, ya sea por medio de referencias o por apuntadores.

**Procesamiento al azar o procesamiento secuencial.** Muchos trabajos de procesamiento, se

hacen, por su naturaleza en forma secuencial como por ejemplo, el obtener un reporte de la nómina. Algunos otros se realizan por procesamientos al azar como las actualizaciones, y estos procesos implican tener un menor tiempo en el proceso y su respuesta es más rápida. Por lo que hay que considerar el proceso que se va a llevar a cabo y escoger si se va a realizar secuencialmente o random.

**Razón de la actividad del archivo.** En muchos sistemas de procesos secuenciales no se explotan todos los registros y en algunos casos son solamente una pequeña porción del total de registros. Por lo que definiremos la actividad del archivo como la razón de los registros leídos y usados en la pasada y los registro explotados.

$$\text{Razón Actividad} = \frac{\text{No Reg Leídos y usados}}{\text{No Reg Explotados}}$$

Sí esta razón es muy grande, se puede utilizar una cinta o un dispositivo de acceso directo, ya que practicamente se ejecutará en el mismo tiempo, y si la razón es muy pequeña digamos un 2%, usaremos un acceso directo, estos dispositivos nos saltan los registros inecesarios y nos baja el tiempo de ejecución, además que un dispositivo de acceso directo es más rápido que una cinta.

**Frecuencia de las referencias.** A ciertos registros se hacen frecuentes referencias, mientras otros se mencionan muy pocas veces, por ejemplo en un banco los registros de saldos son más llamados frecuentemente que los registros de tipo administrativo donde llevan la fecha de nacimiento o su dirección.

**El tiempo de respuesta.** Esto es, si un dato se utiliza con mucha frecuencia y se desea obtener rápidamente, todo dependerá del modo conversacional y la formulación de búsqueda en canales donde la cola de espera es larga y no deben tomarse demasiado tiempo en la respuesta. esto se dá muy frecuentemente en las terminales de los sistemas en línea,

**Producción.** En este factor se deben de tomar en cuenta el crecimiento de la producción que va a tener en un determinado tiempo, para preveer el espacio y la organización física para minimizar los tiempos de búsqueda.

**Volatilidad de los datos.** Se llama un archivo vólatil a aquel que tiene mucha actividad de inserción y eliminación de registros, lo cual es un factor determinante en el diseño físico, ya que periódicamente hay que realizar una distribución y reorganización de los datos.

**Complejidad de las estructuras.** Las estructuras lógicas simples, tales como las de las bases de datos normalizadas, se prestan para su nítida conversión a las formas físicas.

**Disponibilidad.** La importancia de que el sistema de tiempo real continue trabajando todo el tiempo posible, y muchas veces es necesario tener un archivo duplicado en otro dispositivo, con el objeto de que si existe alguna falla en el archivo se pueda recurrir al archivo duplicado sin tener pérdida de tiempo en la recuperación

**Recuperación de datos.** un accidente puede destruir los archivos. Por lo que es necesario realizar porcedimientos de respaldo, copiando periódicamente los archivo y sus cambios en dispositivos como cintas o diskettes según el computador que se este utilizando.

Para la organización física, se desea economizar espacio de almacenamiento y estar al mismo tiempo en condiciones de localizar y leer los registros con bastante rapidez.

Algunas veces los registros físicos pueden ser de longitud fija, en la que conviene tener un número óptimo de registros lógicos, para tener mejor aprovechamiento del espacio, ya que entre cada

registro físico hay un área o espacio ocupado que nos indica la longitud del registro físico y lógico, como se muestra en la figura 2.8, y por lo tanto entre menos registros lógicos en el registro físico, mayor espacio desperdiciado.

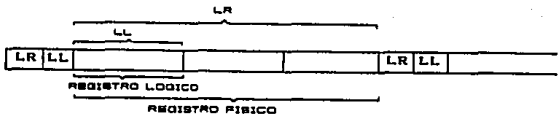



*Fig 2.8 Representación gráfica del registro físico.*

Cuando existen varias unidades de almacenamiento, es necesario organizar los datos en partes de tal manera que el acceso a la lectura y búsqueda de los datos sea más rápida y el tiempo real y el de respuesta sean más breves, porque si estan organizados en forma serial, nos eleva el tiempo en la búsqueda de los datos debido a que tiene que recorrer todo el archivo para encontrar el ultimo registro. Estas distribuciones las podemos ver en la figura 2.9.

En nuestro caso usaremos un microcomputador que tiene una sola unidad de disco de acceso directo, y un manejador de base de datos llamado **DBASEIII**, por lo que nuestra base de datos permanecerá físicamente en solo una unidad de disco de acceso directo.

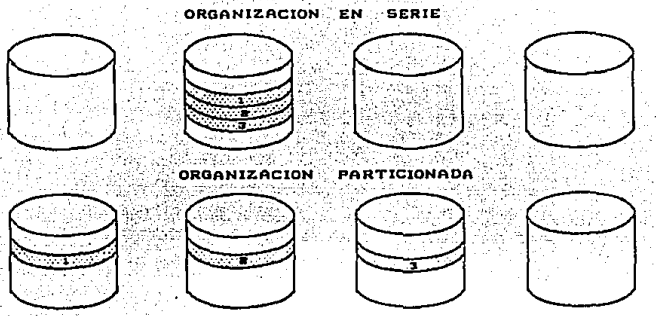



*Fig 2.9 Organización en serie y particionada en discos de acceso directo.*

De nuestro diseño lógico obtenemos una lista de todos los campos y sus longitudes, de tal manera que después obtengamos la longitud de nuestras tablas o base de datos que vamos a utilizar, y entonces calcular el espacio que se necesitamos para la información que vamos a cargar. A continuación damos el nombre de cada campo (atributo) su longitud, el tipo de campo, si es número (N), alfanúmerico (A) o de formato fecha (D) con su nombre nemonico y longitud total de la tabla en bytes, le llamaremos a esto el **LAYOUT** de la tabla. Donde el atributo subrayado será la llave de la tabla:

**TABLA MATERIA.**

| Atributos | long | tipo | nemonico |
|---|---|---|---|
| Clave de materia | 5 | A | CVEMAT |
| Nombre de materia | 20 | A | NOMMAT |
| Longitud Total | 25 | | |

**TABLA ALUMNOS.**

| Atributos | long | tipo | nemonico |
|---|---|---|---|
| Número de cuenta | 8 | N | CTAALU |
| Número de credencial | 5 | N | CREDALU |
| Nombre del alumno | 60 | A | NOMALU |
| Dirección del alumno | 40 | A | DIRALU |
| Colonia del alumno | 30 | A | COLALU |
| Télefono del alumno | 7 | A | TELALU |
| Tipo de sangre | 3 | A | TIPALU |
| Carrera | 3 | A | CARALU |
| Longitud Total | 156 | | |

**TABLA MATERIAL.**

| Atributos | long | tipo | nemonico |
|---|---|---|---|
| Clave del material | 8 | A | CVEMTL |
| Nombre del material | 30 | A | NOMMTL |
| Longitud Total | 38 | | |

**TABLA REACTIVO.**

| Atributos | long | tipo | nemonico |
|---|---|---|---|
| Clave del reactivo | 8 | A | CVEREA |
| Nombre del reactivo | 30 | A | NOMREA |
| Longitud Total | 38 | | |

**TABLA PRODUCTOS.**

| Atributos | long | tipo | nemonico |
|---|---|---|---|
| Clave del producto | 8 | A | CVEPRO |
| Nombre del producto | 30 | A | NOMPRO |
| Longitud Total | 38 | | |

**TABLA PRACTICAS.**

| Atributos | long | tipo | nemonico |
|---|---|---|---|
| Clave de la práctica | 3 | N | CVEPRT |
| Nombre de la práctica | 120 | A | NOMPRT |
| Longitud Total | 123 | | |

**TABLA CALIFICACIONES.**

| Atributos | long | tipo | nemonico |
|---|---|---|---|
| Número de cuenta | 8 | N | CATCAL |
| Clave de la materia | 5 | A | MATCAL |
| Calificación | 4 | A | CALCAL |
| Semestre | 5 | A | SEMCAL |
| Clave del grupo | 5 | A | GPOCAL |

| | | | |
|---|---|---|---|
| Observaciones | 20 | A | OBSCAL |
| Longitud Total | 47 | | |

**TABLA HORARIOS.**

| Atributos | long | tipo | nemonico |
|---|---|---|---|
| Semestre | 5 | A | SEMHOR |
| Clave del grupo | 5 | A | GPOHOR |
| Día de la semana | 3 | A | DIAHOR |
| Número del laboratorio | 5 | A | LABHOR |
| Hora de inicio | 5 | A | INIHOR |
| Hora de terminación | 5 | A | TERHOR |
| Nombre del profesor | 25 | A | PROFHOR |
| Carrera impartida | 3 | A | CARHOR |
| Longitud Total | 56 | | |

**TABLA SEMESTRE.**

| Atributos | long | tipo | nemonico |
|---|---|---|---|
| Semestre | 5 | A | CVESEM |
| Fecha de inicio | 10 | D | INISEM |
| Fecha de terminación | 10 | D | TERSEM |
| Longitud Total | 25 | | |

**TABLA MATERIAL DE PRACTICAS.**

| Atributos | long | tipo | nemonico |
|---|---|---|---|
| Clave de la práctica | 2 | N | CVEMYP |

| | | | |
|---|---|---|---|
| Clave del material | 8 | A | MTLMYP |
| Clave del equipo | 1 | A | EQUMYP |
| Cantidad o concentración | 8 | A | CANMYP |
| Longitud Total | 19 | | |

**TABLA ADEUDO DE MATERIAL.**

| Atributos | long | tipo | nemonico |
|---|---|---|---|
| Semestre | 5 | A | SEMADE |
| Número de cuenta | 9 | N | CTAADE |
| Clave del material | 8 | A | MTLADE |
| Cantidad | 2 | N | CANADE |
| Fecha de adeudo | 10 | D | FECADE |
| Longitud Total | 24 | | |

**TABLA CONTROL DE PRACTICAS.**

| Atributos | long | tipo | nemonico |
|---|---|---|---|
| Semestre | 5 | A | SEMCTL |
| Carrera | 3 | a | CARCTL |
| Clave de la materia | 5 | A | MATCTL |
| Clave del material | 8 | A | MTLCTL |
| Número de secuencia | 2 | N | SECCTL |
| Longitud Total | 23 | | |

**TABLA LIBRO DE PRACTICAS.**

| Atributos | long | tipo | nemonico |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Semestre | 5 | A | SEMLIB |
| Clave de la materia | 5 | A | MATLIB |
| Clave de la práctica | 2 | N | PRTLIB |
| Número de secuencia | 2 | N | SECLIB |
| Clave del grupo | 5 | A | GPOLIB |
| Número de práctica | 2 | N | NPRTLIB |
| Fecha realizada | 10 | D | FECLIB |
| Resultado | 20 | A | RESLIB |
| Observaciones | 20 | A | OBSLIB |
| Longitud Total | 71 | | |

Ahora nos ocuparemos de estimar el tamaño físico de nuestra base de datos, lo haremos como lo mostramos a continuación con el número de renglones estimados por cada tabla:

| Nombre de Tabla | long.(bytes) | No renglones por semestre | Total (bytes) |
|---|---|---|---|
| 1) MATERIA | 25 | 5 | 125 |
| 2) MATERIAL | 38 | 400 | 15,200 |
| 3) PRODUCTOS | 38 | 500 | 19,000 |
| 4) REACTIVOS | 38 | 500 | 19,000 |
| 5) PRACTICAS | 123 | 200 | 24,600 |
| 6) MATERIAL PRACT | 19 | 1000 | 19,000 |
| 7) CALIFICACIONES | 47 | 380 | 17,860 |
| 8) HORARIOS | 56 | 30 | 1,680 |
| 9) SEMESTRE | 25 | 1 | 25 |

| | | | |
|---|---|---|---|
| 10) ALUMNOS | 156 | 350 | 54,600 |
| 11) ADEUDO MATERIAL | 24 | 20 | 480 |
| 12) CONTROL PRACT | 23 | 1000 | 23,000 |
| 13) LIBRO PRACTICAS | 71 | 1000 | 71,000 |
| | | TOTAL | 265,570 |

Nuestra base de datos contendra 265,570 bytes cada semestre, proyectandolo a 10 semestres nuestro espacio crecerá de la siguiente manera; las primeras seis tablas prácticamente permanecen constantes a menos que, se tengan nuevas prácticas, por lo que su cambio de espacio es mínimo, y las demás tablas por cada semestre se insertan nuevos renglones, éstas tablas ocupan un espacio de 168,645 bytes, por lo que el espacio calculado para los diez semestres es aproximadamente de 1'731,875 bytes, equivalen a tres diskettes con densidad de 720 Kb o dos diskettes de 1.4 Mb.

# C A P I T U L O  3

## MANIPULACION DE LOS DATOS POR MEDIO DEL LENGUAJE DE DBIII

El dBASE III es un lenguaje sencillo de usar, podemos decir que se puede manejar de dos maneras, una es por medio del asistente (assist) y la otra es por programación. El assist es una herramienta que nos ayuda a crear, editar, desplegar, copiar, eliminar bases de datos, pantallas etc., también podemos eliminar, modificar, añadir, localizar registros de una base de datos; además de muchas otras funciones. Sin embargo, solamente nos ocuparemos de los conceptos básicos de la programación.

### 3.1 CONCEPTOS BASICOS DE DBASE III.

En computación, un **programa** es una serie de instrucciones o comandos, que le indican a la computadora lo que debe de hacer. Y al conjunto de programas se le conoce como **Sistema de Aplicación.**

En el diseño de los programas deben aparecer los comandos en una forma lógica y progresiva, tomando en cuenta los mecanismos de control de los programas, para dirigir el flujo de los mismos de acuerdo a lo que se desea.

Para el diseño del sistema de aplicación es muy importante involucrar al usuario, con el fin de saber qué desea que haga el programa, y saber sus necesidades respecto a los contenidos de datos que se le dan ya sea por reportes o pantallas, sin perder de vista la integridad de los datos, filtrando los datos de entrada para la obtención de datos de salida correctos.

De esta manera identificamos las necesidades básicas del programa, que dividiéndolas en actividades o pequeños módulos, y éstos a su vez en tareas específicas, obtenemos un diagrama de enfoque de nuestro sistema aplicativo, como lo muestra la figura 3.1. A toda esta división en módulos lógicos, se le conoce como programación estructurada.

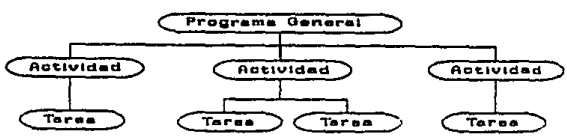



*Fig 3.1.- Enfoque del diseño de programas.*

Teniendo nuestro módulos lógicos procederíamos a codificarlos, ya que cada módulo podría ser un programa o subprograma. Es muy importante que los

programas se documenten, por medio de líneas de comentarios en las funciones que se realicen.

Los mecanismos de control para el flujo de programas en dBASE III, se utilizan cuatro construcciones principales que son:

- Bifurcaciones con la instrucción **DO**.
- Procesos repetitivos con la instrucción **DO WHILE... ENDDO**.
- Condiciones con la instrucción **IF ... ENDIF**.
- Multiples condiciones con la instrucción **DO CASE ... ENDCASE**.

Existen dos tipos de variables de memoria llamadas **PUBLICAS** y **PRIVADAS**. Cuando se declara una variable pública esta se utiliza en todos los módulos del programa, y cuando se declara una variable privada esta solo se utiliza en el módulo donde fue declarada.

Para definir el ambiente de trabajo se usan los comandos **SET**, los más importantes son:

- **SET TALK OFF** es para no tener comunicación con el ASSIST.
- **SET ESCAPE OFF** no permite que usuario usar la tecla de escape para interrumpir el programa.
- **SET BELL OFF** es la que controla el sonido de la campana.

- **SET STATUS OFF** elimina las líneas de mensajes y de barra de estado.

Exiten dos tipos de funciones las NUMERICAS y las de CADENAS. Las númericas son; **ABS()** es el valor absoluto, **EXP()** es el valor exponencial, **INT()** es el valor entero, **LOG()** es el logaritmo natural, **MAX()** y **MIN()** son los valores máximo y mínimo, **MOD()** es el residuo, **ROUND()** redondeo de un número y **SQRT()** es la raíz cuadrada de un número. Las de cadena son; **LEN()** nos da la longitud de una cadena, **SUBSTR()** tomamos una parte de la cadena, **LEFT()** y **RIGHT()** tomar una parte de de la cadena comenzando desde la izquierda o derecha. También hay funciones que nos convierten carácteres a números (**VAL()**) y viceversa y carácteres a fechas (**CTOD()**) y viceversa.

Para la comunicación con el usuario se realiza por medio de la pantalla, la cual esta dividida en 25 renglones por 80 carácteres, y se hace por medio de las instrucciónes como **@...GET** unida a un **READ** que deja que usuario suministre uno o más datos, y con las instrucciones **INPUT**, **ACCEPT** y **WAIT** donde el usuario solo responde a un dato. De esta manera se hace fácil y amigable la comunicación con el usuario. A esta comunicación se le puede colocar máscaras, con el fin de que la información que dé el usuario sea lo

más depurada posible, y no introducir basura al sistema, así la instrucción como **PICTURE** puede ser "9" solo puede aceptar números, con "A" solo aceptar alfabéticos, con "X" acepta cualquier carácter, con "99/99/99" acepta fechas, con "Y" acepta desiciones lógicas (Si o No), y **RANGE** para rangos entre números o fechas.

Cuando trabajamos con una base de datos, es necesario abrirla y se utiliza el comando **USE**, que también sirve para abrir archivos indexados creados por la instrucción **INDEX ON**.

Para buscar un campo específico se usa **LOCATE FOR** o irse a un registro dado con **GOTO**. otras instrucciones son **FIND** y **SEEK** que solo se usan en archivos indexados. Para saber si encontro el registro se pregunta con la instrucción **FOUND()**. Para irse al primer registro es con **TOP** y para el último registro con **BOTTOM**, para saber si es fin de archivo se pregunta con **EOF()** y para saber si existe un archivo se pregunta con **FILE(nombre archivo)**

Para la lectura secuencial se usa la instrucción **SKIP**, si deseamos actualizar la base de datos, usamos los comandos, **APPEND BLANK** que nos agrega un registro en blanco y con **REPLACE...WITH**, remplaza los campos que le indiquemos,o quizás deseamos borrarlo y usamos

el comando **DELETE** que solo marca el registro, para recuperar se realiza con **RECALL**, y borrarlo definitivamente con **PACK**, que empaca la base de datos, o eliminar todos los registros de la base de datos con **ZAP**.

Para las salidas como impresión, usamos **SET DEVICE TO PRINT** y con la instrucción **@...SAY**, para saltar de hoja con **EJECT**, y los margenes se manejan con **SET MARGEN TO**.

Y por último del programa, debemos cerrar archivos con **USE** o **CLOSE**, y limpiar la pantalla con **CLEAR**.

Resumiendo un programa tiene 3 pasos a seguir y son: La creación del ambiente de trabajo, el control del proceso y regresar al ambiente original.

## 3.2 DESARROLLO DE LAS PANTALLAS PARA EL SISTEMA.

Para el desarrollo de estas pantallas, es necesario que hagamos una lista de las actividades o funciones que se van a relizar en el sistema, como son:

Inicio de semestre. Que nos indica la fecha de inicio y terminación del semestre, así como los horarios de clases del laboratorio.

Inscripción de alumnos. Dar de alta al alumno en el laboratorio que va a cursar.

Adeudo de material. Llevar el control de todos aquellos alumnos que deben material de laboratorio.

Programa tradicional de prácticas. Este es el documento oficial de las prácticas realizadas del laboratorio para las diferentes carreras.

El libro de prácticas. Aquí el responsable anota el resultado de las prácticas que el profesor llevo a cabo.

La carpeta de prácticas. Es donde se anota el material mínimo necesario para llevar a cabo la práctica.

Reportes. Es para obtener las listas de los alumnos que adeudan material y los comprobantes de laboratorio y comprobantes de no adeudo.

Al conocer las funciones que se realizan procedemos a crear un diagrama de las actividades que se llevaran a cabo en el sistema, como lo muestra la figura 3.2. En este diagrama señalamos como primera opción un menu principal, el cual nos indica un indice de las actividades que realizará el sistema, que a su vez esa actividad tiene tareas de alta, bajas, cambios y consultas.

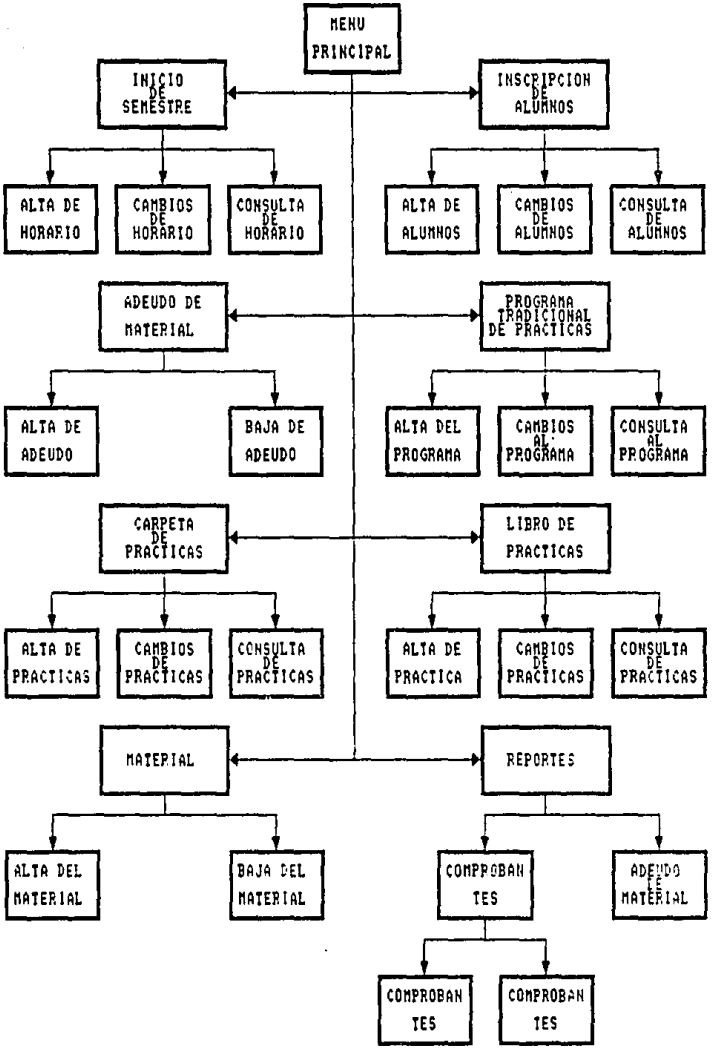



*Fig 3.2.- Diagrama de actividades o funciones.*

Cada cuadrito representaria una pantalla, que sería manejado por un programa, de esta manera comenzamos a diseñar las pantallas por medio de una formato especial,como el que se ve en la figura 3.3. Se mencionó en el capítulo anterior que la pantalla consta de 25 renglones por 80 caracteres, en donde utilizaremos los cuatro primeros renglones para información general como, fecha, hora, descripción de la actividad, identificación del programa, los siguientes 19 renglones se usarán para el contenido de los datos que cada actividad debe tener,

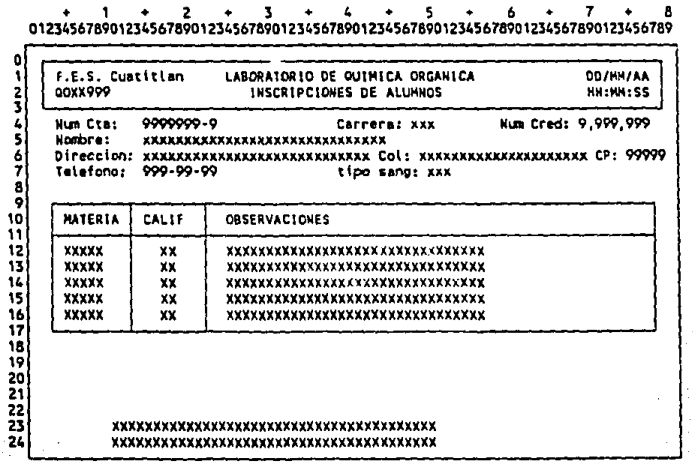



*Fig 3.3.- Formato para diseño de pantallas.*

y por último los dos renglones restantes se usan para manejar mensajes al usuario, ejemplo: "NO SE ENCONTRO AL ALUMNO" , "ERROR EN LA OPCION", "MODIFICACION REALIZADA", "TECLEE LA OPCION", etc.

La siguientes figuras (figura 3.4a, 3.4b, 3.4c ) son solo algunos ejemplos de como quedan las pantallas diseñadas para el sistema y con esto poder después empezar a programar, ya que estas pantallas es parte de las especificaciones que se le dan al programador para que pueda comenzar a codificar las instrucciones del programa para el sistema.

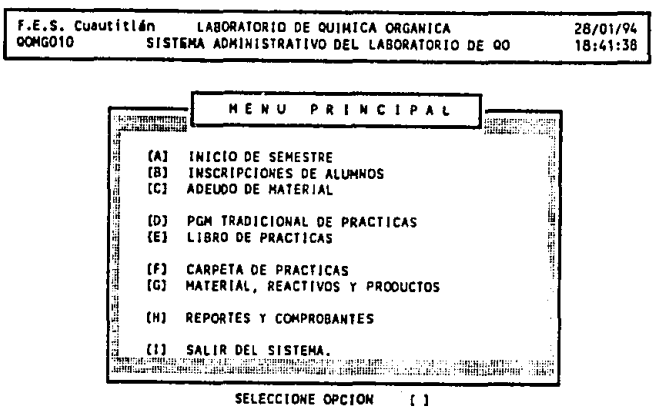



*Fig 3.4a.- Pantalla Menú principal.*

```
F.E.S. Cuautitlán      LABORATORIO DE QUIMICA ORGANICA          28/01/94
OOSA010                       ALTA DE SEMESTRE                18:41:43

                        ALTA DEL SEMESTRE

   Teclee los siguientes datos:

   Cve Semestre (año-sem):        93-II
   Fecha de inicio:               20/09/93
   Fecha de terminación:          20/01/94

   Presione alguna tecla para continuar
```

*Fig 3.4b.- Pantalla Alta de semestre.*

```
F.E.S. Cuautitlán      LABORATORIO DE QUIMICA ORGANICA          28/01/94
OOSA020                       ALTA DE HORARIOS                18:43:21

   Semestre: 93-II   Fecha Inicio: 20/09/93   Fecha Terminación: 20/01/94

     Día:              No.Lab:        Día:              No.Lab:
Grupo:                              Grupo:
Materia:                            Materia:
Carreras:                           Carreras:
Profesor:                           Profesor:
Hora inic:                          Hora inic:
Hora term:                          Hora term:

     Día:              No.Lab:        Día:              No.Lab:
Grupo:                              Grupo:
Materia:                            Materia:
Carreras:                           Carreras:
Profesor:                           Profesor:
Hora inic:                          Hora inic:
Hora term:                          Hora term:

   Presione '←Enter←' para regresar al menu anterior
```

*Fig 3.4c.- Pantalla Alta de horario.*

Así de esta manera, se diseñaron un total de 34 pantallas para todo el sistema.

## 3.3 PROGRAMAS APLICATIVOS DE LA BASE DE DATOS PARA LOS LABORATORIOS DE QUIMICA.

Ya que se diseñaron las pantallas, ahora lo que sigue es dar las especificaciones para el programador, éstas especificaciones son las pantallas y lo que el programa debe de hacer, apoyandonos en el diagrama de funciones de la figura 3.2. Para la programación debemos de tener una nomenclatura la cual nos facilitará la identificación de programas en el sistema y que se ha diseñado de la siguiente manera: El nombre del programa constará de siete carácteres, QOXXNNN donde QO es constante, XX nos indica si es un menú o un programa, si es menú el primer carácter es M y el segundo carácter el tipo del menú, una G si es general, una S si es de semestre, I si es de inscripción, T si es del programa tradicional de prácticas, R para reportes etc, y si es programa el primer carácter de que menú proviene, si es del semestre S, si es de reportes R etc.,el segundo carácter es A, B, C, o Q, que significan altas, bajas, cambios y consultas, y por

último NNN nos indica el número secuencial del programa, que van de 10 en 10. Así podemos decir que, el primer programa será el menú general llamándolo QOMG010, y que tendrá 10 opciones y si se teclea una diferente de éstas se obtendrá un mensaje que dirá "OPCION INVALIDA". Las opciones son con letras desde la A a la I y la X para salir del sistema. Así de esta manera se le dan al programador las especificaciones de cada programa y el empezará a codificar para después seguir con la fase de pruebas, creando datos, para probar el funcionamiento del sistema.

A continuación veremos la codificación de algunos programas ya terminados y probados, como es el menú general, alta de semestre, alta de horarios, de tal forma que nos sirva de ejemplo para darnos cuenta de como es la estructura de los programas y como estan hechos:

```
*************************************************************
*
* SISTEMA        :    PROCESO ADMINISTRATIVO DEL LAB DE QO
* PROGRAMA       :    QOMG010
* FUNCION        :    MENU PRINCIPAL
* ELABORADO POR  :    DOMINGO SERGIO SANCHEZ SANCHEZ
* FECHA          :    SEPTIEMBRE 1993.
*
*************************************************************
*
CLEAR ALL
SET SCOREBOARD OFF
SET ESCAPE     OFF
SET CONSOLE    OFF
SET TALK       OFF
SET STATUS     OFF
SET BELL       OFF
SET DATE       BRIT

SET PROCEDURE TO BIBLOS

*-------------------------------------------------------------
*                          ENTRADA AL SISTEMA
*-------------------------------------------------------------

do GRL_MKEYS      && rutina de definiciÂn de variables del sist

public TIME
TIME = 500

   SIST_NAME = "SISTEMA PARA LABORATORIOS DE QUIMICA"
   FUNC_ID   = "ENTRADA"
   FUNC_NAME = "ELABORADO POR DOMINGO S. SANCHEZ SANCHEZ"
   clear

do GRLTITUL     &&  titulos generales

chance = 0
do while chance <> 3
   mpsw = space(7)
   @ 11, 20 say "TECLEE LA PASSWORD PARA ENTAR AL SISTEMA"
   set color to ,w/w
   @ 12, 37 get mpsw picture "!!!!!!!"
   read
   if mpsw = "SISTEMA"
     set color to
     exit
   else
     do GRLERROR with "ERROR EN LA PASSWORD INTENTE NUEVAMENTE"
     chance = chance + 1
   endif
   set color to
enddo
```

```
if chance = 3
   do GRLERROR with "LO SIENTO AGOTO SUS OPORTUNIDADES"
   clear
   do TERMINO
   SET SCOREBOARD ON
   SET ESCAPE     ON
   SET CONSOLE    ON
   SET TALK       ON
   SET STATUS     ON
   SET BELL       ON
   quit
endif

*-------------------------------------------------------------
*                      OPERACION DEL SISTEMA
*-------------------------------------------------------------

DO WHILE .T.
   SIST_NAME = "LABORATORIO DE QUIMICA ORGANICA"
   FUNC_ID   = "QOMG010"
   FUNC_NAME = "SISTEMA ADMINISTRATIVO DEL LABORATORIO DE QO"
   clear

do GRLTITUL     && titulos generales

*-------------------------------------------------------------
*                  PINTANDO LA PANTALLA DEL MENU
*-------------------------------------------------------------

@  6, 13  to 23, 68 double
@  7, 14  say REPLICATE(chr(176),9)
@  7, 59  say REPLICATE(chr(176),9)
*
@  8, 14  say chr(176)
@  9, 14  say chr(176)
@ 10, 14  say chr(176)
@ 11, 14  say chr(176)
@ 12, 14  say chr(176)
@ 13, 14  say chr(176)
@ 14, 14  say chr(176)
@ 15, 14  say chr(176)
@ 16, 14  say chr(176)
@ 17, 14  say chr(176)
@ 18, 14  say chr(176)
@ 19, 14  say chr(176)
@ 20, 14  say chr(176)
@ 21, 14  say chr(176)
@ 22, 14  say REPLICATE(chr(176),54)
```

```
@  8, 67  say chr(176)
@  9, 67  say chr(176)
@ 10, 67  say chr(176)
@ 11, 67  say chr(176)
@ 12, 67  say chr(176)
@ 13, 67  say chr(176)
@ 14, 67  say chr(176)
@ 15, 67  say chr(176)
@ 16, 67  say chr(176)
@ 17, 67  say chr(176)
@ 18, 67  say chr(176)
@ 19, 67  say chr(176)
@ 20, 67  say chr(176)
@ 21, 67  say chr(176)
*
@  5, 23  to  7, 58 double
*
@ 09, 18  say "[A]"
@ 10, 18  say "[B]"
@ 11, 18  say "[C]"
@ 13, 18  say "[D]"
@ 14, 18  say "[E]"
@ 16, 18  say "[F]"
@ 17, 18  say "[G]"
@ 19, 18  say "[H]"
*
@  6, 24  say "    M E N U   P R I N C I P A L   "
@ 09, 23  say "INICIO DE SEMESTRE"
@ 10, 23  say "INSCRIPCIONES DE ALUMNOS"
@ 11, 23  say "ADEUDO DE MATERIAL"
@ 13, 23  say "PGM TRADICIONAL DE PRACTICAS"
@ 14, 23  say "LIBRO DE PRACTICAS"
@ 16, 23  say "CARPETA DE PRACTICAS"
@ 17, 23  say "MATERIAL, REACTIVOS Y PRODUCTOS"
@ 19, 23  say "REPORTES Y COMPROBANTES"
*
@ 21, 18  say "[I]"
@ 21, 23  say "SALIR DEL SISTEMA."
@ 24, 50  say "["
@ 24, 52  say "]"
@ 24, 29  say "SELECCIONE OPCION"

*-------------------------------------------------------------
*                  VALIDACION DE LA SELECCION
*-------------------------------------------------------------

  I = 0
  do while I = 0
    I = INKEY()
    @  2, 70  say TIME()
    @ 24, 51 say ""
```

```
    if UPPER(CHR(I))$"ABCDEFGHI"
       exit
    endif
    I = 0
  enddo

*-------------------------------------------------------------
*                  VALIDACION DE LA SELECCION
*-------------------------------------------------------------

  DO CASE
   case CHR(I) $ "Aa"
       clear
       do QOMS010
   case CHR(I) $ "Bb"
       clear
       do QOMI010
   case CHR(I) $ "Cc"
       clear
       do QOMA010
   case CHR(I) $ "DdEeFfGgHh"
       clear
       @  5, 23  to  7, 58 double
       @  6, 24  SAY " NO DISPONIBLE "

       I = 0
       do while I = 0
          I = INKEY()
          @  2, 70  say TIME()
          @ 24, 51 say ""
          @ 15, 24  SAY "TECLEA UNA "X" PARA SALIR"
          if UPPER(CHR(I))$"X"
          exit
          endif
          I = 0
       enddo
   case CHR(I) $ "Ii"
       clear
       do TERMINO
       SET SCOREBOARD ON
       SET ESCAPE     ON
       SET CONSOLE    ON
       SET TALK       ON
       SET STATUS     ON
       SET BELL       ON
       return
  ENDCASE
ENDDO
****************** fin del programa QOMG010 ********************
```

```
*****************************************************************
*
* SISTEMA         :     PROCESO ADMINISTRATIVO DEL LAB DE QO
* PROGRAMA        :     QOMS010
* FUNCION         :     MENU INICIO DE SEMESTRE
* ELABORADO POR :     DOMINGO SERGIO SANCHEZ SANCHEZ
* FECHA           :     SEPTIEMBRE 1993.
*
*****************************************************************

*---------------------------------------------------------------
*                          OPERACION DEL SISTEMA
*---------------------------------------------------------------
DO WHILE .T.

SIST_NAME = "LABORATORIO DE QUIMICA ORGANICA"
FUNC_ID   = "QOMS010"
FUNC_NAME = "INICIO DE SEMESTRE"

do GRLTITUL     &&  titulos generales

*---------------------------------------------------------------
*                    PINTANDO LA PANTALLA DEL MENU
*---------------------------------------------------------------

@  6, 13  to 23,68 double
@  7, 14  say REPLICATE(chr(176),9)
@  7, 59  say REPLICATE(chr(176),9)

@  8, 14  say chr(176)
@  9, 14  say chr(176)
@ 10, 14  say chr(176)
@ 11, 14  say chr(176)
@ 12, 14  say chr(176)
@ 13, 14  say chr(176)
@ 14, 14  say chr(176)
@ 15, 14  say chr(176)
@ 16, 14  say chr(176)
@ 17, 14  say chr(176)
@ 18, 14  say chr(176)
@ 19, 14  say chr(176)
@ 20, 14  say chr(176)
@ 21, 14  say chr(176)
@ 22, 14  say REPLICATE(chr(176),54)

@  8, 67  say chr(176)
@  9, 67  say chr(176)
@ 10, 67  say chr(176)
@ 11, 67  say chr(176)
@ 12, 67  say chr(176)
```

```
@ 13, 67  say chr(176)
@ 14, 67  say chr(176)
@ 15, 67  say chr(176)
@ 16, 67  say chr(176)
@ 17, 67  say chr(176)
@ 18, 67  say chr(176)
@ 19, 67  say chr(176)
@ 20, 67  say chr(176)
@ 21, 67  say chr(176)

@  5, 23  to  7, 58 double

@ 11, 18  say "[A]"
@ 13, 18  say "[B]"
@ 15, 18  say "[C]"
@ 17, 18  say "[X]"
*
@  6, 24  say "      HORARIOS DE LABORATORIO      "
@ 11, 23  say "ALTA DE FECHA DEL SEMESTRE Y HORARIO"
@ 13, 23  say "CAMBIO DE FECHA DE SEMESTRE Y HORARIO"
@ 15, 23  say "CONSULTA DE HORARIOS POR LABORATORIO"
@ 17, 23  say "REGRESAR AL MENU PRINCIPAL"
*
@ 24, 50  say "["
@ 24, 52  say "]"
@ 24, 29  say "SELECCIONE OPCION"

*-------------------------------------------------------------
*                  VALIDACION DE LA SELECCION
*-------------------------------------------------------------

  I = 0
  do while I = 0
    I = INKEY()
    @  2, 70  say TIME()
    @ 24, 51 say ""
    if UPPER(CHR(I))$"XxAaBbCc"
       exit
    endif
    I = 0
  enddo
  DO CASE
   case CHR(I)$"Aa"
   clear
   do QOSA010
   case CHR(I)$"Bb"
   clear
   do QOSC010
   case CHR(I)$"Cc"
   clear
```

```
    do QOSQ010
    case CHR(I) $ "Xx"
    exit
   ENDCASE
   loop
ENDDO
return
****************** fin del programa QOMS010 ******************
```

```
*************************************************************
*
* SISTEMA        :    PROCESO ADMINISTRATIVO DEL LAB DE QO
* PROGRAMA       :    QOSA010
* FUNCION        :    ALTA DE SEMESTRE
* ELABORADO POR  :    DOMINGO SERGIO SANCHEZ SANCHEZ
* FECHA          :    SEPTIEMBRE 1993.
*
*************************************************************

*-------------------------------------------------------------
*         VERIFICA SI EXISTE LA BASE DE DATOS
*-------------------------------------------------------------

if .not. file ("SEMESTRE.DBF")
   @ 22,  0  clear
   do GRLERROR with "NO EXISTE BASE DE DATOS SEMESTRE.DBF"
   do ANY
   clear all
   clear
   return
else
   if .not. file ("SEM_IDX.NDX")
      select 1
      use SEMESTRE
      do ESPERA
      INDEX ON cvesem  TO  sem_idx
      @ 24,  0  clear
      use
   endif
endif

*-------------------------------------------------------------
*                       OPERACION DEL SISTEMA
*-------------------------------------------------------------
DO WHILE .T.

  do while .t.
   SIST_NAME = "LABORATORIO DE QUIMICA ORGANICA"
   FUNC_ID   = "QOSA010"
   FUNC_NAME = "ALTA DE SEMESTRE"
   do GRLTITUL     &&  titulos generales

*-------------------------------------------------------------
*                PINTANDO LA PANTALLA DEL MENU
*-------------------------------------------------------------

   @  6, 10  to 23,71 double
   @  7, 11  say REPLICATE(chr(176),12)
   @  7, 59  say REPLICATE(chr(176),12)
```

```
@  8, 11  say chr(176)
@  9, 11  say chr(176)
@ 10, 11  say chr(176)
@ 11, 11  say chr(176)
@ 12, 11  say chr(176)
@ 13, 11  say chr(176)
@ 14, 11  say chr(176)
@ 15, 11  say chr(176)
@ 16, 11  say chr(176)
@ 17, 11  say chr(176)
@ 18, 11  say chr(176)
@ 19, 11  say chr(176)
@ 20, 11  say chr(176)
@ 21, 11  say chr(176)
@ 22, 11  say REPLICATE(chr(176),60)

@  8, 70  say chr(176)
@  9, 70  say chr(176)
@ 10, 70  say chr(176)
@ 11, 70  say chr(176)
@ 12, 70  say chr(176)
@ 13, 70  say chr(176)
@ 14, 70  say chr(176)
@ 15, 70  say chr(176)
@ 16, 70  say chr(176)
@ 17, 70  say chr(176)
@ 18, 70  say chr(176)
@ 19, 70  say chr(176)
@ 20, 70  say chr(176)
@ 21, 70  say chr(176)

@  5, 23  to  7, 58 double
@  6, 24  say "        ALTA DEL SEMESTRE          "

select 1
USE SEMESTRE INDEX sem_idx

go top
msem = space(4)
set color to gr+
@ 10, 15  say "Teclee los siguientes datos:"
@ 12, 15  say "Cve Semestre (aÊo-sem):"
@ 24, 17  say "Presione '+Enter+' para retornar al menu prin

*-------------------------------------------------------------
*                ENTRA EL DATO DEL SEMESTRE
*-------------------------------------------------------------

set color to gr+
@ 12, 48  get msem  picture  '@R 99-!!'
```

```
   read

   if msem = " "
      close databases
      close index
      clear
      return
   endif

*-------------------------------------------------------------
*        BUSCA SI EXISTE NUM DE SEM EN LA BASE DE DATOS
*-------------------------------------------------------------

   if eof()
      exit
   else
    find &msem
    if found()
       @ 13, 15 say "Fecha de inicio:"
       @ 14, 15 say "Fecha de terminaciÁn:"
       @ 13, 48 say inisem   picture "@D"
       @ 14, 48 say tersem   picture "@D"
       do GRLERROR with "YA EXISTE ESTE SEMESTRE"
       @ 24, 17 say "Presione alguna tecla para continuar"
       wait
       clear
       close databases
       close index
       do QOSA020 with msem
       loop
    else
       exit
    endif
   endif
  enddo

*-------------------------------------------------------------
*         VALIDACION DE LAS FECHAS DE INICIO Y TERMINACION
*-------------------------------------------------------------

  do while .t.
     do while .t.
        mfini = space(8)
        set color to gr
        @ 13, 15  say "Fecha de Inicio (DD/MM/AA):"
        set color to gr+
        @ 13, 48  get mfini picture  '@D'
        read
        do VALIDFEC with mfini
        if valida = ' '
```

```
          exit
       else
         loop
       endif
    enddo

    do while .t.
       mfter = space(8)
       set color to gr
       @ 14, 15  say "Fecha de TerminaciÂn (DD/MM/AA):"
       set color to gr+
       @ 14, 48  get mfter picture  '@D'
       read
       do VALIDFEC with mfter
       if valida = ' '
          exit
       else
         loop
       endif
    enddo
    if substr(mfter,7,2) < substr(mfini,7,2)
      do GRLERROR with "FECHA INICIO MAYOR QUE FECHA DE TERMIN
      @ 13, 48 say "  /  /  "
      @ 14, 48 say "  /  /  "
      loop
    else
     if substr(mfter,7,2) = substr(mfini,7,2)
       if substr(mfter,4,2) < substr(mfini,4,2)
         do GRLERROR with "ERROR MES DE INICIO MENOR AL DE TER
         @ 13, 48 say "  /  /  "
         @ 14, 48 say "  /  /  "
         loop
       else
         if substr(mfter,4,2) = substr(mfini,4,2)
            do GRLERROR with "ERROR MISMO AËO Y MES EN FECHAS"
            @ 13, 48 say "  /  /  "
            @ 14, 48 say "  /  /  "
            loop
         else
            exit
         endif
       endif
     else
       exit
     endif
    endif
enddo
```

```
*-------------------------------------------------------------
*          CONFIRMACION DE LA ALTA DEL SEMESTRE
*-------------------------------------------------------------

  @ 24,  1 clear
  @ 24, 60 say "["
  @ 24, 62 say "]"
  @ 24, 25 say "Teclee 'S' Confirmar 'N' Cancelar"
  I = 0
  do while I = 0
   I = INKEY()
   @  2, 70 say TIME()
   @ 24, 61 say ""
   if UPPER(CHR(I))$"SsNn"
     exit
   endif
  enddo

  if UPPER(CHR(I))$"Ss"
    APPEND BLANK
    REPLACE cvesem WITH msem
    REPLACE inisem WITH CTOD(mfini)
    REPLACE tersem WITH CTOD(mfter)
    do GRLERROR with "ALTA REALIZADA"
    go top
    clear
    close databases
    close index
    do QOSA020 with msem
  endif
    loop
ENDDO
return
******************** fin de programa QOSA010 ******************
```

```
*****************************************************************
*
* SISTEMA        :    PROCESO ADMINISTRATIVO DEL LAB DE QO
* PROGRAMA       :    QOSA020
* FUNCION        :    ALTA DE HORARIOS
* ELABORADO POR  :    DOMINGO SERGIO SANCHEZ SANCHEZ
* FECHA          :    SEPTIEMBRE 1993.
*
*****************************************************************

parameters MSG

msem = space(4)
store MSG to msem

SIST_NAME = "LABORATORIO DE QUIMICA ORGANICA"
FUNC_ID   = "QOSA020"
FUNC_NAME = "ALTA DE HORARIOS"

do GRLTITUL     && titulos generales

*---------------------------------------------------------------
*         VERIFICA SI EXISTE LA BASE DE DATOS
*---------------------------------------------------------------

if .not. file ("HORARIOS.DBF")
   @ 22,  0  clear
   do GRLERROR with "NO EXISTE BASE DE DATOS HORARIOS.DBF"
   do ANY
   clear all
   clear
   return
else
   if .not. file ("HORA_IDX.NDX")
      select 1
      use HORARIOS
      do ESPERA
      INDEX ON semhor + gpohor + labhor TO hora_idx
      @ 24,  0  clear
      use
   endif
endif

*---------------------------------------------------------------
*                       OPERACION DEL SISTEMA
*---------------------------------------------------------------

select 1
USE SEMESTRE INDEX sem_idx
seek msem
```

```
if .not. found()
   do GRLERROR with "NO ENCONTRO SEMESTRE"
   return
endif

*-------------------------------------------------------------
*                  PINTANDO LA PANTALLA DEL MENU
*-------------------------------------------------------------

@  5,  7  say "Semestre:"
@  5, 25  say "Fecha Inicio:"
@  5, 51  say "Fecha TerminaciÂn:"

@  5, 17  say cvesem   picture  "@R XX-XX"
@  5, 39  say inisem   picture  "@D"
@  5, 70  say tersem   picture  "@D"

select 2
USE HORARIOS INDEX hora_idx

DO WHILE .T.
  mnum =  1
  mren =  0
  mcol =  0
   @  7,  9  say "DÅa:"
   @  7, 28  say "No.Lab:"
   @  8,  5  say "Grupo:"
   @  9,  5  say "Materia:"
   @ 10,  5  say "Carreras:"
   @ 11,  5  say "Profesor:"
   @ 12,  5  say "Hora inic:"
   @ 13,  5  say "Hora term:"

   @  7, 46  say "DÅa:"
   @  7, 65  say "No.Lab:"
   @  8, 46  say "Grupo:"
   @  9, 46  say "Materia:"
   @ 10, 46  say "Carreras:"
   @ 11, 46  say "Profesor:"
   @ 12, 46  say "Hora inic:"
   @ 13, 46  say "Hora term:"

   @ 15,  9  say "DÅa:"
   @ 15, 28  say "No.Lab:"
   @ 16,  5  say "Grupo:"
   @ 17,  5  say "Materia:"
   @ 18,  5  say "Carreras:"
   @ 19,  5  say "Profesor:"
   @ 20,  5  say "Hora inic:"
   @ 21,  5  say "Hora term:"
```

```
  @ 15, 46  say "DÀa:"
  @ 15, 65  say "No.Lab:"
  @ 16, 46  say "Grupo:"
  @ 17, 46  say "Materia:"
  @ 18, 46  say "Carreras:"
  @ 19, 46  say "Profesor:"
  @ 20, 46  say "Hora inic:"
  @ 21, 46  say "Hora term:"

*-------------------------------------------------------------
*   CONTROL DE LAS CUATRO PARTES DE LA PANTALLA CON MNUM
*-------------------------------------------------------------

  do while .not. mnum = 5
    do case
       case mnum = 1
         mren = 7
         mcol = 20
       case mnum = 2
         mren = 7
         mcol = 57
       case mnum = 3
         mren = 15
         mcol = 20
       case mnum = 4
         mren = 15
         mcol = 57
       otherwise
         do GRLERROR with "ERROR EN EL PROGRAMA"
         close databases
         close index
         clear
         return
    endcase
    mdia = space(3)
    @ mren, mcol  get  mdia  picture '!!!'
    @ 24, 17  say "Presione '+Enter+' para regresar al menu ant
    read
    if mdia = " "
       close databases
       close index
       clear
       return
    else
       if mdia = "LUN" .or. mdia = "MAR" .or. mdia = "MIE";
        .or. mdia = "JUE" .or. mdia = "VIE" .or. mdia = "SAB"
          mlab = space(4)
          mgpo = space(5)
          @ mren, mcol+16 get mlab  picture '@R !-999'
          @ mren-1, mcol  get mgpo  picture '9999!'
```

```
      read
    else
      do GRLERROR with "DIA ERRONEO INTENTE DE NUEVO"
      loop
    endif
endif
@ 24,  1  clear

if eof()
  do GRLERROR with "REGISTRO NUEVO"
else
  locate for semhor=msem .and. gpohor=mgpo .and. labhor=mla
  if .not. eof()
     do while .not. eof()
       msw = 0
       if mdia = diahor
         msw = 1
         @ 24, 1  say semhor
         @ 24, 6  say gpohor
         @ 24, 15 say labhor
         @ 24, 22 say diahor
         @ 24, 28 say recno()
         exit
       else
         @ 24, 1 clear
         @ 24, 28 say recno()
         continue
       endif
     enddo
     if msw = 1
       do GRLERROR with "YA EXISTE ESTE GRUPO EL MISMO DIA"
       @ mren, mcol clear to mren+6, mcol+20
       loop
     endif
  endif
endif

mmat = space(5)
mcar = space(6)
mprof = space(18)
mhin = space(4)
mhte = space(4)
@ mren+2, mcol get mmat   picture '!!!!!'
@ mren+3, mcol  get mcar   picture '@R !!!-!!!'
@ mren+4, mcol  get mprof  picture '!!!!!!!!!!!!!!!!!!'
@ mren+5, mcol  get mhin   picture '@R 99:99'
@ mren+6, mcol  get mhte   picture '@R 99:99'
read
```

```
    @ 24,  1  clear
    @ 24, 61  say "["
    @ 24, 63  say "]"
    @ 24, 25  say "Teclee 'S' Confirmar y 'N' Cancelar"

    I = 0
    do while I = 0
      I = INKEY()
      @  2, 70  say TIME()
      @ 24, 62 say ""
      if UPPER(CHR(I))$"SsNn"
        exit
      endif
      I = 0
    enddo
    if UPPER(CHR(I))$"Ss"
      APPEND BLANK
      REPLACE semhor  WITH msem
      REPLACE gpohor  WITH mgpo
      REPLACE diahor  WITH mdia
      REPLACE labhor  WITH mlab
      REPLACE inihor  WITH mhin
      REPLACE terhor  WITH mhte
      REPLACE profhor WITH mprof
      REPLACE carhor  WITH mcar
      REPLACE mathor  WITH mmat
      mnum = mnum + 1
      @ 24,  1  clear
      do GRLERROR with "ALTA REALIZADA"
    else
      @ mren, mcol clear to mren+6, mcol+20
    endif
    goto top
  enddo

*-------------------------------------------------------------
*                 VALIDACION DE LA SELECCION
*-------------------------------------------------------------

  @ 24,  1  clear
  @ 24, 50  say "["
  @ 24, 52  say "]"
  @ 24, 25  say "Desea otra ALTA (S/N)"

  I = 0
  do while I = 0
    I = INKEY()
    @  2, 70  say TIME()
    @ 24, 51 say ""
```

```
    if UPPER(CHR(I))$"SsNn"
       exit
    endif
    I = 0
  enddo
  if UPPER(CHR(I))$"Ss"
    @  7, 1 clear to 21, 79
    loop
  endif
  clear
  exit
ENDDO
clear
return

****************** fin del programa QOSA020 ******************
```

## 3.4 MANUAL DE OPERACION DEL SISTEMA INTERACTIVO PARA EL LABORATORIO DE QUIMICA.

El manual de la operación es un punto muy tedioso para muchos usuarios y profesionales, ya que muchas veces no lo contemplan como algo de primera necesidad. Este manual es también llamado documentación de usuario, y nos describe la operación correcta del sistema. En muchos casos los sistemas son desarrollados pero la documentación no esta bien o ni siquiera existe, comunmente los que desarrollan el sistema, piensan que tienen cosas más importantes que sentarse a escribir el manual de operación para el usuario, y ésta postura es contraproducente para quienes le dan mantenimiento al sistema, y los usuarios finales inclusive dicen que el sistema no sirve o que no esta bien hecho, pero es, por que no lo conocen o no lo saben manejar.

El manual de operación para el usuario como mínimo debe cumplir con los siguientes puntos:

- Una explicación del arranque del sistema.
- Una breve descripción de lo que hace la aplicación.
- Una descripción del menú general de cada opción.
- Una explicación detallada de todos los submenus.
- Ilustración de las pantallas con datos válidos.

- Una explicación del los reportes que se tienen en el sistema y descripción de los datos del mismo.
- Y por último una explicación de como generar los respaldos (backup's) del sistema.

Ya terminado el manual de operación, debe ser validado por el usuario final, para que esté convencido del sistema y conozca su funcionamiento, de esta manera el trabajo será más profesional.

A continuación describiremos parte del manual de la operación para el sistema interactivo para el manejo administrativo del laboratorio de química.

**INTRODUCCION.** El sistema interactivo para el manejo administrativo del laboratorio de química es un sistema hecho en un administrador de base de datos, diseñado para tener el control de los alumnos inscritos en el laboratorio de química, y materias que han cursado, así como los diferentes programas tradicionales de prácticas impartidas de cada carrera que se dan en el laboratorio, y la emisión de reportes y comprobantes de adeudo y no adeudo de material.

**ARRANQUE DEL SISTEMA.** El sistema arranca de la siguiente manera:

- Encienda su computadora IBM o compatible.

- Cuando el sistema operativo nos avise que esta lista para usarse, que aparezca el C> o A>. Debemos de irnos al subdirectorio donde esté instalado el Dbase III plus, tecleando cd dbiii.

- Teclear dbase syslab aparecerá la pantalla donde nos pide una password, le tecleamos sistema, y ya entramos al menú general del sistema.

**CONTENIDO DEL SISTEMA.** El menú general le permite elegir una de las siguientes posibilidades, tecleando la letra correspondiente:

A) INICIO DE SEMESTRE.

B) INSCRIPCION DE ALUMNOS.

C) ADEUDO DE MATERIAL.

D) PGM TRADIONAL DE PRACTICAS.

E) LIBRO DE PRACTICAS.

F) CARPETA DE PRACTICAS.

G) MATERIAL, REACTIVOS Y PRODUCTOS.

H) REPROTES Y COMPROBANTES.

I) SALIDA DEL SISTEMA.

INICIO DE SEMESTRE. Es la opción A del menú general nos envía a un submenú (QOMS010), el cual contiene 4 opciones más que son: A ALTA DE FECHAS DEL SEMESTRE Y HORARIOS (QOSA010), en donde podemos dar de alta la clave del semestre, con un formato 99-XX donde 99 es el año escolar en número cardinal y XX es primer o

segundo semestre en número romano (I o II), ejemplo, primer semestre de 1994, se escribe 94-I. A continuación se teclean las fechas de inicio y terminación del semestre en formato DD/MM/AA, donde DD es el número de día, MM es el número de mes y AA es el año. Al terminar aparecerá un mensaje para confirmar si los datos estan correctos o no, entonces se teclea N para cancelar, y nos regresa nuevamente al inicio de la clave del semestre, o S para confirmar y se da de alta el semestre enviadonos a otra pantalla para dar de alta los horarios que corresponden a este semestre (QOSA020), y aquí llenaremos la pantalla con los datos; Día de la semana que son las primeras 3 letras del día, número de laboratorio en formato L-999, donde L es constante y 999 es el número del laboratorio, el número de grupo en formato 9999X donde 9999 es el número de grupo y X si el grupo se divide en dos partes A o B, o espacio si no se divide, las siglas de la materia, ejemplo Química Orgánica II será QOII, la siglas de la carrera a la que se le imparte, ejemplo, Ingeniería Química IQ, Químico Q, o Químico Farmacobiólogo QFB, el nombre del profesor de 25 caracteres, y por último la hora de inicio y terminación del laboratorio, en formato HH:MM, donde

HH es la hora y MM los minutos. Cuando terminamos de dar de alta los horarios, y nos regresa al menú anterior, solamente tecleamos enter y nos regresa al submenú y elegimos la segunda opción B.CAMBIO DE FECHA DE SEMESTRE Y HORARIOS (QOSC010), aquí damos un cambio a las fechas de inicio y terminación del semestre en caso de que ésten erróneas las fechas, tecleandolas igualmente que en la alta del semestre, también podemos cambiar el horario (QOSC020), en caso de que nos hallamos equivocado, tecleando el semestre y el laboratorio al que deseamos cambiar el horario, de igual forma para regresar al menú anterior, solamente tecleamos enter, y nos regresa al menú anterior. En la opción C CONSULTA DE HORARIOS POR LABORATORIO (QOSQ010), en donde nos pide el semestre y laboratorio que deseamos consultar como horarios, dandonos una pantalla por cada día de la semana, Nuevamente para regresar al menú anterior tecleamos enter. Y la opción X es para regresar al menú principal.

Así, de esta manera, nosotros vamos describiendo cada una de las opciones que tenemos en el menú principal y que valores va tomando cada dato para que sea válido, también debemos de ilustrar las pantallas y no olvidar de que consta el sistema, esta es,

cuantos programas pantallas, archivo como bases de datos e indices, dando los nombres de cada uno. Que en realidad estos datos se encuentran en la documentación del diseño detallado del sistema.

Nuestro sistema consta de 50 programas en total que equivalen a 50 pantallas. con 13 bases de datos y 20 indices.

## CONCLUSIONES.

El presente trabajo se estructuró, de tal manera que cualquier persona que tenga contacto con un laboratorio de química, sea capaz de tener los conocimientos básicos de lo que es una base de datos y de que elementos se compone, de llevar a cabo una metodología para el diseño de la misma y de como desarrollar un sistema en lenguaje DBIII plus para microcomputadora IBM o compatibles, qúe le facilite llevar un mejor control administrativo del laboratorio de química.

Con esto, nos damos cuenta de las ventajas tan importantes de tener un sistema interactivo en una microcomputadora como son:

- Ahorro de tiempo.
- Mayor facilidad de manejo.
- Ahorro de papel.
- Ahorro de espacio de archiveros.
- Facilidad de control de sus diferentes funciones.
- Tener el manual de operación del sistema.
- Y facilidad de instalarlo.

Espero que este trabajo, sea de gran ayuda para estudiantes y maestros, que tengan interés en conocer una metodología del diseño y desarrollo de un sistema automatizado hecho en base de datos.

## BIBLIOGRAFIA.


1.- Date.
Introducción a los Sistemas de Base de Datos.
Adisson. 1988, pp5.

2.- Shakuntala Atre.
Técnicas de Base de Datos.
1a. Ed. Trillas. 1988, pp17.

3.- Date.
Introducción a los Sistemas de Base de Datos.
Adisson. 1988, pp5.

4.- Villagrana R. Jorge.
Introdución a los Sistemas de Cómputo.
Tésis. ENEP Aragón 1990, pp97.

5.- Date.
Introducción a los Sistemas de Base de Datos.
Adisson. 1988, pp5.

6.- Gio Wiederhold.
Diseño de base de Datos.
2a. Ed. (1a en español). 1985.

7.- Joseph-David Carrabis.
DBase III plus Manual de Referencia.
McGraw-Hill. 1990.

8.- N. T. Dinerstein.
DBase III Como diseñar y realizar un programa de aplicaciones.
Gustavo Gili. 1990.

9.- Edward Jones.
DBase III plus Guía para Usuarios Expertos.
McGraw-Hill. 1989.

10.- Ashton Tate.
Aprendiendo y Usando DBase III plus.
Vol. 1. 1989.

11.- Ashton Tate.
Programando el DBase III plus.
Vol. 2. 1989.

12.- DBMS (Developing Corporate Application).
November 1991.
Vol. 4 number 12.

13.- Adad Rubén, Saez Margarita.
Curso de Base de Datos DB2.
IBM Corp. 1991.

14.- Alexander Gaydasch,Jr.
Efective Database Management.
Prentice-Hall, Inc. 1988.