



**UNIVERSIDAD LASALLISTA
BENAVENTE**



ESCUELA DE INGENIERÍA EN COMPUTACIÓN

Con los estudios incorporados a la Universidad
Nacional Autónoma de México
CLAVE: 8793-16

**“DISEÑO E IMPLEMENTACIÓN PARA EL
CONTROL Y REGISTRO DE PRÉSTAMO DE
EQUIPO DEL LABORATORIO DE
ELECTRÓNICA DEL INSTITUTO
TECNOLÓGICO DE CELAYA”**

TESIS QUE PARA OBTENER EL TÍTULO DE:
INGENIERA EN COMPUTACIÓN

Presenta:

LUISA JOSETT BELMONTE JARAMILLO

Asesor: Ing. Anselmo Ramírez González

Celaya, Gto. Diciembre de 2005



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Agradecimientos

Dios:

Por haber puesto en mí camino a la gente que me ha ayudado incondicionalmente y ha estado conmigo en las buenas y en las malas. Y por estar siempre junto a mí.

Padres:

Por haberme dado una educación, su cariño y un buen ejemplo, por haber hecho de mí la persona que soy ahora. A ti, Papá † un beso y un abrazo, gracias por seguir conmigo.

Hermano:

Gracias por estar siempre apoyándome en los proyectos que me he trazado.

Lic. Héctor Aguilar Tamayo †:

Mil gracias por el apoyo incondicional y el cariño que siempre mostró para conmigo y mi familia; por estar siempre en las buenas y en las malas. Doy gracias a Dios por darme la oportunidad de conocerlo, un abrazo y un beso donde quiera que se encuentre.

Félix López Z.:

Gracias por estar conmigo siempre en las buenas y en las malas y apoyarme en este gran esfuerzo incondicionalmente.

Ing. Yolanda Guirette Zavala:

Gracias por haberme brindado tú apoyo y dedicarme tiempo aun cuando no nos conocíamos; sobre todo por darme lo más importante que tiene el ser humano que es tú amistad. Mil gracias.

Ing. Anselmo Ramírez González:

Mil gracias por la preocupación que mostró para poder ver terminado este proyecto, por preocuparse por mí y estar siempre recordándome ese "último" paso.

Ing. Miguel Ángel Jamaica Arreguin:

Gracias por ser una persona que aparte de darme conocimientos me brindo apoyo y amistad.

ÍNDICE

INTRODUCCIÓN

CAPÍTULO I DESARROLLO DE SISTEMAS

1.1	¿Qué es un sistema?	2
1.2	Características de los sistemas	3
1.3	¿Qué es un sistema de información?	5
1.4	Elementos de un sistema de información	5
1.5	Objetivos de un sistema de información	6
1.6	Pasos del ciclo de vida de un sistema	7

CAPÍTULO II BASES DE DATOS

2.1	Introducción a base de datos	11
2.1.1	Objetivos de los sistemas de base de datos	11
2.1.2	Modelos de datos	12
2.1.3	Instancias y Esquemas	13
2.1.4	Independencia de los Datos	14
2.1.5	Lenguaje de Definición de Datos	15
2.1.6	Lenguaje de Manejo de Datos	15
2.1.7	Manejador de Base de Datos	16
2.1.8	Administrador de la Base de Datos	16
2.1.9	Usuario de la Base de Datos	17
2.1.10	Estructura General del Sistema	18
2.2	Modelo Entidad Relación	19
2.2.1	Entidad y Conjunto de Entidades	19
2.2.2	Relación y Conjunto de Relaciones	20
2.2.3	Limitantes de Mapeo	20
2.2.4	Llaves Primarias	23
2.2.5	Diagramas de Entidad Relación	24

2.2.6	Generalización y Especialización	25
2.3	Modelo Relacional	26
2.3.1	Estructura de las Bases de Datos Relacionales	27
2.3.2	Lenguajes Formales de Consulta	27
2.3.3	Lenguajes Comerciales de consulta	29
2.3.4	Modificaciones de las Bases de Datos	32
2.3.5	Vistas	32
CAPÍTULO III ¿QUÉ ES VISUAL BASIC 6.0?		
3.1	Introducción	34
3.2	El Entorno de Visual Basic 6.0	36
3.2.1	La Barra de Menú y las Barras de Herramientas	36
3.2.2	Las Herramientas (Toolbox)	38
3.3	Formularios y Módulos	41
3.4	La ventana de Proyecto	42
3.5	La ventana de Propiedades	43
3.6	Creación de Programas Ejecutables	44
3.7	Utilización del Code Editor	45
3.8	Utilización del Debugger	49
3.8.1	Ejecución Controlada de un Programa	49
3.8.2	Ventanas immediate ,Locals y Watches	50
CAPÍTULO IV DISEÑO E IMPLEMENTACIÓN PARA EL CONTROL, REGISTRO Y PRÉSTAMO DEL LABORATORIO DE ELECTRÓNICA.		
4.1	Análisis Preliminar del Sistema	56
4.2	Determinación de Requerimientos	65
4.3	Desarrollo del Sistema	73

4.4 Prueba del Sistema	82
4.5 Instalación del Sistema	83

CONCLUSIONES

ANEXO PROGRAMA FUENTE

BIBLIOGRAFÍA

INTRODUCCIÓN

La propuesta aquí hecha nace de la problemática que se presenta en los Laboratorios de Electrónica, en cuanto a tiempo se refiere, ya que el alumno llega al Laboratorio a solicitar en préstamo de equipo electrónico, al pedir este, el encargado tiene que revisar un vale donde se toman los datos del alumno, nombre del equipo que solicita, materia, fecha, firma, etc. Lo cual es tardado, debido a que este proceso se hace a mano utilizando vales. Otro punto, la pérdida de tiempo se presenta cuando llega un grupo de alumnos a hacer uso del Equipo electrónico, pues el personal tiene que atender a cada uno de los alumnos uno por uno para poder entregar el equipo solicitado, como anteriormente se mencionó la entrega de equipo se hace por vale, por lo tanto la persona encargada en ese momento del laboratorio tiene que hacer un vale por cada alumno, en este caso los afectados son los alumnos pues pierden hasta media hora en la fila para poder pedir el préstamo del equipo que en ese momento necesitan.

Otra situación que se presenta es que los encargados no se están enterando del estado en que se encuentra el equipo esto es, si está trabajando al 100% o si le falta una palanca o un botón, detalles que no toman en cuenta o cuando les regresan el equipo no lo revisan si lo entregaron en las mismas condiciones.

Otro punto importante es que si por algún motivo uno de los encargados ya no va a estar en el laboratorio de electrónica, la persona que llegue en su lugar esté al tanto del manejo del laboratorio de electrónica esto es, que sepa qué equipos se manejan, en qué estado se encuentran, si alguno de los alumnos tiene que entregar algún equipo, etc., en pocas palabras que no importe si la persona anterior no lo capacitó o no lo puso al tanto de la situación del laboratorio.

Estas son las situaciones por la cuales se llegó a la conclusión de que había que automatizar dicho proceso, la propuesta es realizar un sistema que utilice un lector óptico para facilitar el préstamo de equipo electrónico, ya

que con solo deslizar el lector óptico por el código de barras automáticamente se den las características del equipo electrónico, y así se ahorra tiempo, y el personal esta al tanto de lo que pasa con los aparatos electrónicos, la base de datos está actualizada y lo importante es que sería un sistema amigable, esto es, que el usuario no tenga ningún problema al utilizarlo. Y los alumnos podrán tener más tiempo en su práctica en lugar de estar esperando turno en la fila.

Estas son las razones que dieron origen a la aplicación de las computadoras y recursos informativos, pues gracias a este medio la información se puede procesar y almacenar de manera organizada según las necesidades de cada usuario, ofreciendo a la vez confiabilidad y rapidez, con la ventaja de que en el momento deseado se puede tener acceso a la información.

CAPÍTULO I

“Desarrollo de sistemas”

1.1 ¿Qué es un sistema?

Para poder comprender mejor el funcionamiento de cualquier sistema es necesario definir o explicar el significado de dicha palabra; "sistema es un conjunto de elementos que interactúan entre si para lograr un objetivo común".¹

Nosotros mismos manejamos sistemas en nuestra vida diaria ya sea un sistema escolar, sistema nervioso ó circulatorio, sistema político, sistemas de información, etc. Por lo tanto nos podemos dar cuenta que los sistemas ocupan un lugar muy importante en nuestra vida.

Los sistemas de información ocupan un lugar muy importante ya que son los que hacen posible el funcionamiento de una oficina, escuela, reservaciones aéreas, departamentos de registros hospitalarios, contabilidad y funciones de nómina, operaciones bancarias electrónicas, sistemas de intercomunicación telefónica, entre otras incontables aplicaciones ya sea en empresas grandes o pequeñas. Anteriormente cuando no existían los sistemas de información las actividades mencionadas anteriormente eran elaboradas con demasiado tiempo de trabajo y dificultad y algunas imposibles de llevar a cabo.

La tecnología avanza a una velocidad impresionante, pero en realidad lo que realmente es importante en cualquier sistema es el conocimiento humano y el uso de las ideas para poder utilizar la computadora para poder llegar a una respuesta satisfactoria. Esto es la esencia para poder desarrollar un sistema. Para que un sistema de información basado en una computadora sea de gran utilidad debe tener un excelente funcionamiento y así poder facilitarle al personal el trabajo para que esto suceda dicho sistema debe de ser amigable o fácil de manejar por los usuarios; si el sistema utilizado no tuviera las

¹ SENN, James, *Análisis y Diseño de Sistemas de Información*, Editorial McGraw-Hill/Interamericana de México S.A. de C.V., Colombia 1990, p.5

características ya mencionadas lo que hace el personal o usuarios es evitarlo y por lo tanto no cumpliría con los objetivos por el cual fue creado.

Los sistemas se clasifican en: Sistemas Abiertos y Sistemas Cerrados. Los sistemas abiertos tienen entradas, procesan los datos y emiten salidas. Esto es, sistemas que interactúan con su medio ambiente; por ejemplo: un sistema educativo, político, eléctrico o de información. Los sistemas cerrados tienen vida mientras tengan material para subsistir, esto es, sistemas que no interactúan con sus alrededores.



fig. 1.1 Diagrama a bloques de un sistema abierto

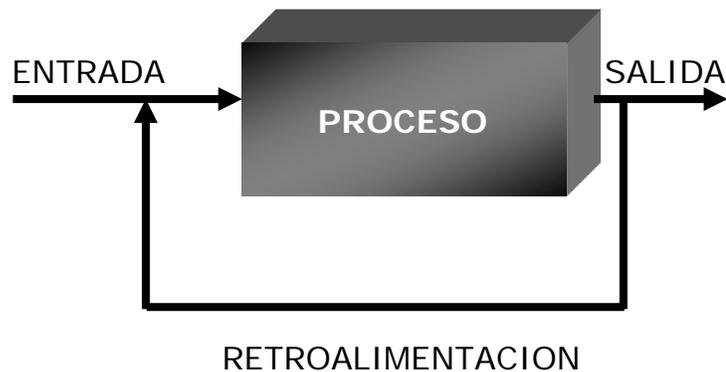


Fig. 1.2 Diagrama a bloques de un sistema cerrado.

1.2 Características de los sistemas

Se debe tomar en cuenta que todo sistema se caracteriza por diferentes puntos tales como son:

1. Los sistemas forman un todo.
2. Cuando algún elemento del sistema falla se comienzan a afectar entre si.
3. Deben de contar con políticas y reglas; Elementos de un sistema de información.
4. Los sistemas deben contar con fronteras o limites.
5. Los sistemas están compuestos de subsistemas y cada subsistema debe tener un líder o un tomador de decisiones y un líder del sistema o tomador de decisiones global.
6. Tienen un objetivo común.
7. Deben de tener entradas, procesos y salidas.

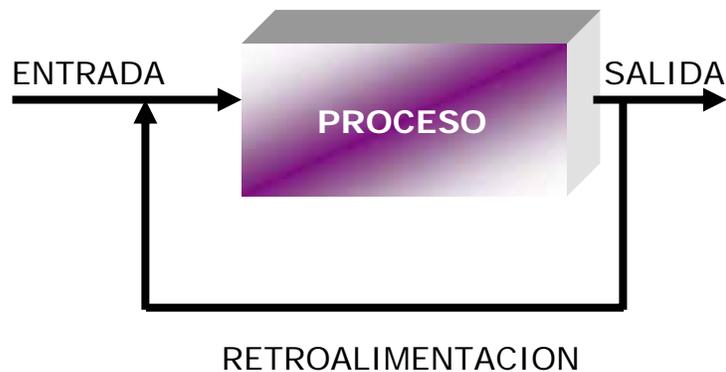


Fig. 1.3 Diagrama a bloques de un sistema cerrado

8. Deben contar con una medida de control que nos permita saber que estamos haciendo y si lo estamos haciendo correctamente.
9. Los sistemas deben contar con planeación, organización, dirección y control.

10. Los sistemas son continuos por lo tanto deben tener seguimiento.

Los sistemas que pueden hacer modificaciones según sus necesidades seguirán trabajando los que no logren este aspecto se detienen.

1.3 ¿Qué es un sistema de información?

Como hemos mencionado hay varios tipos de sistemas por lo tanto todo esta hecho o basado en un sistema, es donde nos damos cuenta lo abundante que son los sistemas, por ejemplo cada sistema de un negocio depende de una o mas entidades abstractas las cuales llamamos sistemas de información. Gracias a estos sistemas pueden estar en constante comunicación el personal ya sea por vía telefónica o por computadoras donde se informen datos periódicamente gracias a un sistema de información, aunque estos se encuentren en departamentos diferentes u oficinas diferentes. Dichos sistemas son el lazo que mantiene unidos a dichos componentes los cuales persiguen un mismo objetivo.

Podemos definir a los "Sistemas de Información" como conjunto de componentes o elementos relacionados entre si para dar apoyo a las actividades de un negocio o empresa, realizando sus funciones con mayor eficiencia y haciendo de esta forma que se incremente la productividad".²

Por lo tanto la importancia de la información en los negocios; a través del tiempo, los empleados y gerentes de las empresas o negocios se han dado cuenta que contar con información confiable, exacta y oportuna no es por casualidad, sino porque cuentan con sistemas de información que les permite mantener un mejor control del negocio, así como una buena toma de decisiones, un negocio mas productivo y competitivo que les permita permanecer en el mercado.

² KENDALL, & Kendall, *Análisis y Diseño de Sistemas*, Editorial McGraw-Hill, Colombia 1990, p.33

1.4 Elementos de un sistema de información

Un sistema debe de tener los siguientes elementos para tener un buen funcionamiento:

- Datos de entrada
- Computadora
- Software o sistema de información
- Personal capacitado para operar el sistema
- Capacitación de la utilización del sistema
- Instrumentos necesarios para laborar

1.5 Objetivos de un sistema de información

Es preciso marcar para un buen seguimiento de un sistema de información los objetivos a los cuales se debe de llegar como son:

1. Reducción del tiempo, tanto en la entrada, proceso y obtención de la información.
2. Permite tener un mejor control administrativo.
3. Reducción de errores.
4. Cubrir las necesidades de la empresa.
5. Permite contar con información confiable, exacta y oportuna.
6. Permite la integración de las diferentes áreas de un negocio.
7. Reducción de costos.
8. Ofrecer un mejor servicio al cliente.
9. Mejorar la comunicación entre otras empresas.

10. Contar con un nivel mas competitivo y esto ayuda a permanecer en el mercado.
11. Dar apoyo a la toma de decisiones.
12. Se reduce el reproceso.

1.6 Pasos del ciclo de vida de un sistema

El desarrollo de sistemas es un proceso que consiste en dos etapas: Análisis y Diseño de Sistemas; estas dos etapas se refieren al proceso de examinar una situación de la empresa teniendo como objetivo el mejoramiento de ésta mediante nuevos procedimientos y métodos.

El análisis es cuando llevamos acabo el proceso de recopilar e interpretar los hechos, diagnosticar problemas y utilizar estos hechos a fin de mejorar el sistema. El diseño cuando se empieza realiza el proceso de planeación de un nuevo sistema dentro de la empresa para reemplazar o completar al existente.

El análisis y diseño de sistemas para los negocios es el proceso que consiste en estudiar una situación de negocios para ver como opera y si es necesaria alguna mejora. Las personas que llevan a cabo estas mejoras se les conoce como analistas de sistemas. La información recopilada a través del estudio forma la base para crear las estrategias alternativas de diseño y la gerencia selecciona la estrategia por utilizar. El ciclo de vida del desarrollo de sistemas es el conjunto de actividades de los analistas, diseñadores y usuarios.

Algunos componentes pueden estar dentro del análisis y otros en etapas avanzadas del diseño. El desarrollo de vida de un sistema se lleva a cabo en las siguientes actividades.

Pasos del Ciclo de Vida

1. Análisis preliminar.
2. Determinación de requerimientos.

3. Desarrollo del sistema prototipo.
4. Diseño del sistema.
5. Desarrollo del software.
6. Prueba del sistema.
7. Instalación del sistema.

1. El **análisis preliminar** se inicia cuando un gerente o empleado o especialista de sistemas inicia un requerimiento para recibir ayuda de un sistema de información. Esta actividad consiste en 3 partes:

a) *Clarificación del requerimiento.* Se debe de entender las necesidades que tiene el usuario, el requerimiento del proyecto debe estar claramente establecido y entender lo que el usuario o la empresa desea.

b) *Factibilidad,* el proyecto es factible si se cuenta con el tiempo disponible para el desarrollo, el costo, el personal y la prioridad. Con respecto a la Factibilidad se tienen que estudiar los siguientes aspectos:

- La factibilidad técnica se refiere a que para poder realizar el trabajo se debe contar con el equipo necesario así como el software y personal disponible.
- Factibilidad operativa, es común que el usuario se resista a los cambios dentro de su trabajo que tiene muchas veces ya mecanizado, por lo tanto se debe de tener la disponibilidad del usuario para llevar a cabo el uso del nuevo sistema.
- Factibilidad económica, si el sistema cumple con los requisitos o necesidades de la empresa debe de tener como resultado a través del tiempo que los beneficios sean mayores que el costo inicial.

2. **La determinación de los requerimientos,** es la recopilación de datos en cuanto a las actividades actuales en la empresa, esto es necesario ya que el

analista debe de tener en claro como es el funcionamiento de la empresa para así tener en claro el objetivo de la misma y poder llegar a un sistema que sea factible.

3. **El desarrollo del sistema prototipo**, cuando se ha llegado a la conclusión de cómo debe de quedar el sistema se empiezan a crear los prototipos de este, que no son mas que pruebas de cómo realmente debe de quedar el diseño del sistema. De manera que el desarrollo del sistema prototipo no tiene otra función mas que de proporcionar información preliminar del sistema.

4. **El diseño del Sistema** se realiza partiendo de los resultados arrojados de la investigación de datos hecha al personal, así pudiendo identificar entradas, salidas y el proceso.

5.- El **desarrollo del software** lo llevan a cabo los programadores, los analistas hacen entrega del diseño a los programadores los cuales lo pasan en forma de software además de especificar el lenguaje de programación que utilizaran, así los formatos tanto de entrada como de salida guiaron a los programadores cuando hacen el cambio de un diseño lógico a un sistema.

6. **La prueba del sistema** como ya lo mencionamos la llevan a cabo los programadores, estos hacen pruebas por partes según lo avanzado del sistema, antes de poner en marcha el sistema en la empresa, los programadores ponen a funcionar el sistema en forma experimental en su totalidad para poderse dar cuenta de su funcionamiento, utilizando personal de la empresa, si este es satisfactorio y cumple el objetivo de la empresa en ese momento el sistema se instala y se hecha andar.

7. **La instalación del sistema** se realiza cuando el personal de sistemas lleva a cabo la instalación del sistema en las computadoras, las oficinas indicadas con el personal indicado de la empresa.

CAPÍTULO II

“Bases de datos”

2.1 Introducción a la base de datos

Una base de datos es un conjunto de información que tiene un significado encontrándose de una manera organizada en entidades logrando tener menos redundancia, manteniendo la integridad y seguridad de la información manejada.

Un Sistema de Manejador de Base de Datos (DBMS), tiene como objetivo crear un ambiente en el que sea posible manejar información ya sea recuperar y almacenarla de una manera eficiente y conveniente.

Existen formas de organizar los datos en la base de datos, aunque las bases de datos relacionales es una de las maneras más efectivas. Los sistemas de bases de datos relacionales son una aplicación de la teoría matemática de los conjuntos al problema de organización de datos. En una base de datos relacional, los datos se organizan en tablas; este sistema es el que se encarga de forzar la estructura de la base de datos, de la siguiente manera:

- Mantiene las relaciones entre los datos en la base de datos
- Checa que estos datos se hayan almacenado correctamente que no sean cambiadas las reglas que definen las relaciones entre los datos.
- Si el sistema llegara a fallar éste recuperará los datos.

2.1.2 Objetivos de los sistemas de base de datos

Las Bases de Datos tienen como objetivo crear un ambiente en el cual sea posible almacenar y la recuperación de información de una manera eficiente. Otro de los modelos que se utilizan para la manipulación de la base de datos es el sistema de procesamiento de archivos, el cual cuenta con un conjunto de programas que permiten tener acceso a nuestra base de datos con el problema de no optimizan los métodos ya conocidos y utilizados, así presentándose algunos otros problemas como:

- Redundancia, es la repetición de datos en los archivos de nuestra base de datos. Llevando al aumento de los costos de acceso y almacenamiento e inconsistencia de datos.
- Dificultad en el acceso a los datos, en este punto el formato de la información no es uniforme debido a que los sistemas de procesamiento de archivos se conforman en distintos tiempos y por lo regular por distintos programadores por lo tanto hay que establecer métodos de enlace y conversión para combinar los datos que ya existen en diferentes archivos.
- Aislamiento de los datos, cuando se presenta la dificultad de extender las aplicaciones que nos permitan controlar a la base de datos. Por ejemplo: nuevas utilerías, reportes, etc., debido a la diferencia de formatos en los archivos almacenados.
- Anomalías en el acceso concurrente, esto se da cuando nuestro sistema es multiusuario y no son establecidos los controles para sincronizar los procesos que afectan a la base de datos. En cuanto a la poca o nula efectividad de los procedimientos de bloqueo.
- Problemas de Seguridad, sino se estableció una contraseña para el acceso para el sistema, fácilmente podrá acceder cualquier persona a el sistema.
- Problemas de integridad, si por alguna razón no tenemos en todo el sistema procedimientos uniformes de validación para los datos.

2.1.3 Modelos de datos

Estos modelos de datos son las herramientas conceptuales para dar una descripción de los datos, su relación entre ellos, sus límites y semántica. Se clasifican de la siguiente manera:

- 1) Modelos lógicos basados en objetos, nos permiten una definición clara y concisa de los esquemas conceptual y de visión. Lo más importante de este

modelo es que permite definir de forma detallada las Limitantes de los datos, a continuación mencionaremos algunos ejemplos:

- Modelo entidad relación
- Modelo binario
- Modelo semántico de los datos
- Modelo infológico

2) Modelos lógicos basados en registros, estos modelos trabajan en niveles conceptual y de visión. Estos permiten una descripción amplia de la implantación, pero no son capaces de especificar con claridad las limitantes de los datos, por ejemplo:

- Modelo relacional, los datos se presentan mediante tablas al igual que sus relaciones. Cada una con diferentes columnas y nombres únicos.
- Modelo de red, los datos son presentados mediante nombres de registros y sus relaciones por medio de un conjunto de ligas.
- Modelo jerárquico, este modelo utiliza una estructura arbolada, presenta algunas similitudes con el modelo de red.

3) Modelos físicos de datos, estos modelos nos describen los datos a un nivel muy bajo y permiten identificar algunos detalles de implantación para el manejo del hardware de almacenamiento. Ejemplos de este modelo son:

- Modelo unificador
- Modelo memoria de cuadros

2.1.4 Instancias y esquemas

Una base de datos tiene la característica de ser dinámica encontrándose sujeta o constantes modificaciones ya sea en alterar datos, eliminar datos, agregar datos o como altas, bajas y cambios de los datos.

Para explicar las etapas que tiene una base de datos existe el concepto de instancias de la base de datos consiste en el estado que esta guardada en un momento determinado. Hay que tener presente que uno o varios archivos pudieron haber sido restaurados o reorganizados.

Una reestructuración es donde se cambia la estructura ya sea quitando, poniendo un campo, modificando longitud o tipo. Y una reorganización es cuando se cambia el modelo con el cual se controla el acceso a los datos por lo tanto la reorganización tiene como consecuencia una reestructuración.

Para poder solucionar el problema al estado que guarda la estructura de la base de datos, se define el concepto de esquema de la base de datos, hace referencia al estado que guarda la organización conceptual por ejemplo: ligas, relaciones, estructura, etc., de la base de datos en un momento específico existiendo varios esquemas para cada uno de los niveles de la base de datos y un esquema conceptual.

2.1.5 Independencia de los datos

Aquí se nos presenta la libertad que tenemos para modificar esquemas sin tener que volver a reescribir los programas de aplicación. Tenemos dos tipos de independencia: Independencia Física e Independencia Lógica.

La independencia física se presenta cuando es posible la modificación del esquema físico sin que afecte a los otros sistemas. Lo principal para hacer una modificación del esquema físico será un ajuste en el hardware de almacenamiento o hacer una redistribución de los datos en el.

La independencia lógica se da cuando se modifica el esquema conceptual sin afectar a los demás esquemas esto es que el sistema conceptual se modifica cuando las características de los datos a almacenar cambian.

Por lo tanto es más sencillo lograr la independencia física ya que una modificación del esquema conceptual requerirá de cambios en el código par su manejo.

2.1.6 Lenguaje de definición de datos (DDL)

El esquema de base de datos es especificado por un conjunto de definiciones que se expresan por medio de un lenguaje especial. El resultado de la combinación de sentencias de Data Definition Language (DDL) es un conjunto de tablas las cuales son almacenadas en un archivo llamada diccionario de datos. Siendo el DDL el que permita describir un esquema de base de datos, y las definiciones resultantes conformaran al diccionario de datos, el cual es un archivo que contiene metadatos que se consulta antes de modificar o leer datos reales en el sistema de base de datos.

2.1.7 Lenguaje de manejo de datos

El lenguaje de manipulación de datos (LMD), es el que maneja la información contenida en la base de datos, este manejo consiste principalmente en la inserción, recuperación, eliminación y modificación de la información.

El lenguaje de manipulación de datos utilizado a nivel físico realizará procesos que permitan tener acceso eficiente a la información, en cuanto al nivel de visión mostrara al usuario destino los datos de una manera clara y sencilla.

Tenemos dos tipos de DML , de Procedimientos y sin procedimientos los cuales definiremos:

- a) Procedimientos, éste especifica que datos deberán de ser manipulados y el método utilizado para ello.
- b) Sin procedimientos, especifican solamente los datos que deberán manejar.

El de procedimientos es el más eficiente en cuanto a sus capacidades de manejo y control de la información aunque su complejidad es mayor.

Existe también un lenguaje de consulta este procesa la recuperación de la información, el cual es parte del DML.

2.1.8 Manejador de base de datos

Es una interfase que hay entre los datos de bajo nivel y los programas de aplicación así como los módulos de consulta que son utilizados a nivel usuarios.

El manejador de base de datos tiene las siguientes funciones:¹

- a) La interacción con el manejador de archivos, esto se realiza traduciendo proposiciones con el DML a instrucciones de bajo nivel par ala manipulación de los datos.
- b) Implantación de integridad, se encarga de verificar que durante las actualizaciones no se viole ninguna limitante de consistencia.
- c) Mejoramiento de nivel de seguridad, se encarga de restringir el acceso mediante una serie de password u otros medios de identificación y validación.
- d) Respaldo y recuperación, proporciona medios automáticos o semiautomáticos para el respaldo de la información. Permite también la recuperación del sistema en caso de caídas, restablecimiento el estado original de la base de datos hasta antes de la falla.
- e) Control de concurrencia, supervisa los accesos en un ambiente multiusuario, determinando a que parte del código y de los datos pueden acceder los usuarios en un momento determinado. El objetivo primordial es mantener la consistencia de la base de datos.

2.1.9 Administrador de la base de datos (DBA)

Como su nombre lo indica es el que tiene el control centralizado de la base de datos, el fin de esto es contar con un control para que solo el personal indicado sea el que tenga acceso a los detalles técnicos y de diseño para la operación de sistema de manejo de la base de datos (DBMS).

¹ KROENKE, David, *Procesamiento de Base de Datos*, Editorial Prentice Hall, México, 1996, p.52

El administrador de la base de datos cuenta con las principales soluciones que son:

- Definición del esquema, crea el esquema original de la base de datos y genera el diccionario de datos por medio de proposiciones en DDL.
- Definición de estructuras de almacenamiento y métodos de acceso, se encarga de generar a seleccionar estructuras para el medio secundario y definir los métodos de acceso a la información, por medio de proposiciones en DML.
- Modificación de esquemas y organización, este es poco frecuente que consiste en rediseñar el esquema de la base de datos. Esto se haría necesario ante la modificación abrupta de las condiciones originales que dieron pie al diseño del esquema primario. Las proposiciones para llevar a cabo esta tarea se realizan en DDL.
- Concesión de autorizaciones de acceso, se encarga del registro a los usuarios para permitir su acceso al sistema de manejo de la base de datos (DBMS), asigna a cada uno de ellos una serie de atributos que le permiten gozar de privilegios como el acceso a determinadas áreas de aplicación, de los datos o del uso de recursos en el sistema.
- Especificación de las limitantes de integridad, crea una serie de tablas donde se especifica el conjunto de restricciones que serán aplicables durante los procesos de actualización.

2.1.10 Usuarios de la base de datos

Entendemos como usuarios de la base de datos a las personas que van a hacer uso y tener acceso a la base de datos. El cual les proporcionará un entorno para obtener información y guardar nueva información en la base de datos. Estas personas que tienen acceso a la base de datos las clasificaremos de la siguiente manera:

- a) Usuarios ingenuos, estas personas interactúan con el sistema con aplicaciones permanentes.
- b) Usuarios sofisticados, ellos acceden a la información por medio de los lenguajes de consulta.
- c) Programadores de aplicación, ellos cuentan con el conocimiento necesario del lenguaje de manipulación de datos creando módulos y utilerías que puedan manejar nuevos datos en el sistema.
- d) Usuarios especiales, es cuando se crean módulos que no manejan datos sino aplicaciones avanzadas como sistemas expertos, reconocimiento de imágenes, audio, etc.

2.1.11 Estructura general del sistema

Un sistema de base de datos esta dividido en módulos los cuales tratan las responsabilidades del sistema general. El diseño de un sistema de base de datos debe de tomar en cuenta la consideración de interfaz entre la base de datos y el sistema operativo.

El sistema global puede verse como la agrupación e interacción de los siguientes elementos: ²

- Manejador de archivos, asigna espacio en el medio de almacenamiento para las estructuras que habrán de almacenar la información.
- Manejador de base de datos, es la interfase entre los datos de bajo nivel y los programas de aplicaciones.
- Procesador de consultas, se encarga de traducir las proposiciones de un lenguaje de consultas a instrucciones de bajo nivel.
- Precompilador de DML, se encarga de traducir las proposiciones en DML al lenguaje de diseño del manejador (Pascal, C, Ensamblador, etc.).

² Idem p.63

- Compilador de DDL, se encarga de convertir las proposiciones en DDL a tablas que contienen metadatos.

Para poder manejar un sistema de manejo de base de datos tenemos la siguiente estructura de datos:

1. Archivo de datos, almacenan a la base de datos.
2. Diccionario de datos, almacenan información referente a la estructura de la base de datos.
3. Índices, permiten un acceso eficiente (rápido y confiable) a la información almacenada en la base de datos.

2.2 Modelo entidad relación

El modelo entidad relación este se basa en objetos por tal motivo se va a los niveles conceptual y de visión. Tiene como característica permitir con claridad las limitantes de los datos, este modelo es una herramienta para la representación del mundo real utilizando símbolos y expresiones determinadas.

2.2.1 Entidad y conjunto de entidades

Cuando hablamos de entidad nos referimos a un objeto el cual puede ser distinto a otro objeto, una entidad puede ser concreta por ejemplo: una silla, un libro, un cuaderno, etc., o abstracta (edad, fecha, etc.). Una misma entidad puede pertenecer a más de un solo conjunto de entidades a la vez. Por ejemplo: la entidad nombre puede ser parte de los conjuntos de entidades alumnos, empleados, maestros; por lo tanto un conjunto de entidades es un grupo de entidades del mismo tipo.

Para distinguir una entidad de otra es de acuerdo a sus características que la hacen única. Estas características son los atributos. El rango de los valores válidos para un atributo determinado será conocido como dominio del atributo.

Ejemplo:



Una entidad es un conjunto de parejas de la siguiente manera: (atributo, valor del dato), se especifica una pareja por cada atributo de la entidad.

Ejemplo:

{(Nombre, Pedro), (Carrera, Ingeniería), (Domicilio, Ulsab)}

2.2.2 Relación y conjunto de relaciones

Entendamos que una relación es una asociación entre varias entidades. Un conjunto de relaciones es un grupo de relaciones del mismo tipo.

Si asociamos dos conjuntos de entidades la relación tendrá una función determinada; a esto se le llama papel. Los papeles se utilizan como etiquetas y así se conocen las relaciones establecidas.

Las relaciones cuentan con atributos descriptivos, la relación se describe indicando la pareja, (atributo, último valor del atributo) sobre la relación.

2.2.3 Limitantes de mapeo

En los modelos entidad-relación se pueden especificar limitantes aplicadas a la información contenida en la base de datos.

Existen dos tipos de Limitantes:

1. Cardinalidad del mapeo
2. Dependencia de existencia

La Cardinalidad del Mapeo es donde especificamos el número de entidades que se asociarán por medio de una relación. Esta se aplica cuando manejamos dos conjuntos de entidades. Como pueden ser:

Una a Una, Cuando solamente se asocia una entidad x con una entidad y

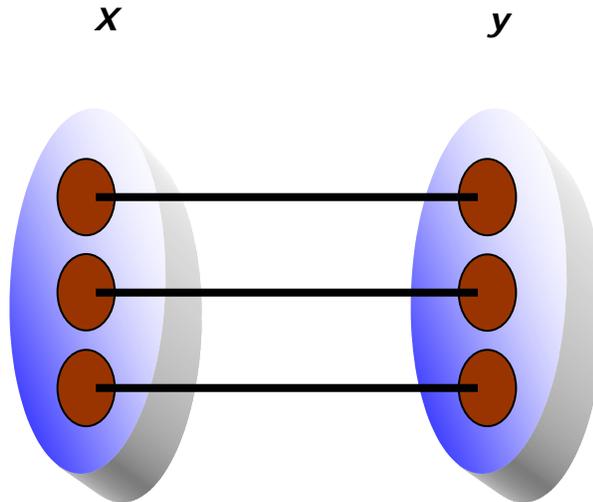


Fig. 2.1 Alumno Tesis

Una a Muchas, Una entidad x se puede asociar con cualquier entidad de entidades de y .

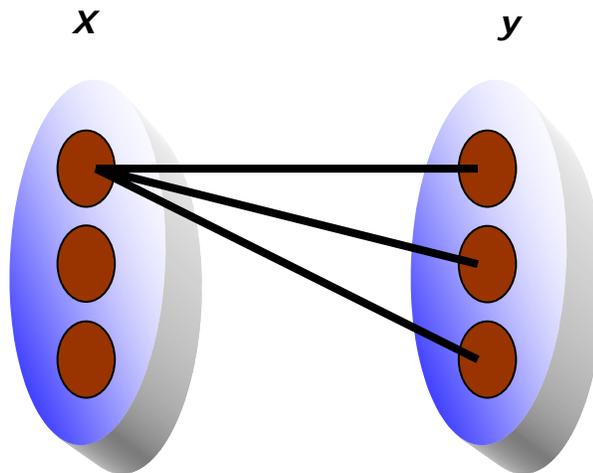


Fig. 2.2 Carrera Alumnos

Muchas a Una, varias entidades de x pueden asociarse con una entidad de y .

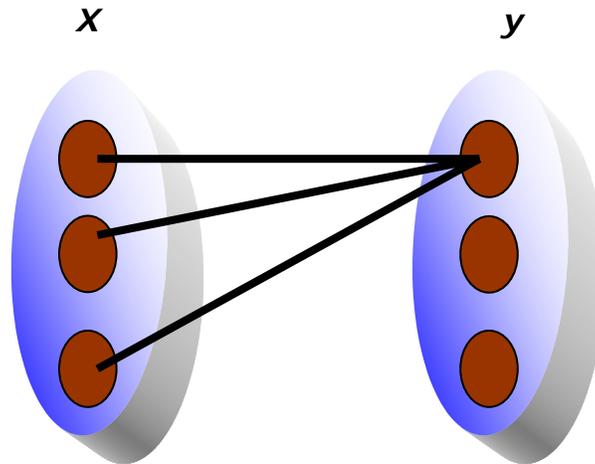


Fig. 2.3 Equipo Alumno

Muchas a Muchas, varias entidades de x pueden asociarse con varias entidades en y .

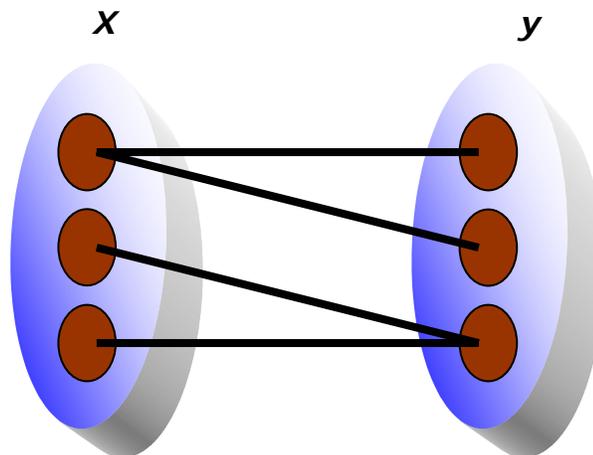


Fig. 2.4 Materia Alumno

La dependencia de existencia es donde un conjunto de entidades esta condicionado ala existencia de otro. Un ejemplo sería Una entidad alumno y la entidad carrera.

A esta limitante se le denomina dependencia por existencia. Si una entidad B requiere de una entidad A para existir entonces quiere decir que B es dependiente por existencia de A, y si eliminamos la entidad A se tendría que eliminar B.

2.2.4 Llaves primarias

En una base de datos hay que distinguir entre las entidades y relaciones que son manipuladas. Una llave es quien nos permite identificar en forma única a una entidad dentro de un conjunto de entidades.

Existen varios tipos de llaves:

- a) Super Llave, es donde el conjunto de atributos nos permite reconocer a la entidad, Este tipo de llaves contiene atributos ajenos, que no son indispensables para el reconocimiento del registro.
- b) Llave candidato, a diferencia de la super llave éstas no contienen atributos ajenos, esto es no tienen un subconjunto menor que pueda considerarse como una super llave.
- c) Llave primaria, es la cual el diseñador de base de datos le da un nombre no repetitivo. Es la seleccionada entre las llaves candidatos encontradas.

Se dice que una entidad es débil cuando el conjunto de entidades no posee los atributos necesarios para formar una llave primaria. Una entidad es fuerte cuando existen atributos para formar una llave primaria.

Las llaves primarias de los conjuntos de relación están formadas por llaves primarias de los conjuntos de entidades que se asocian en la relación y todos los atributos descriptivos de la relación.

2.2.5 Diagramas de entidad relación

En estos diagramas representamos conjuntos de entidades y sus relaciones mediante los siguientes símbolos.

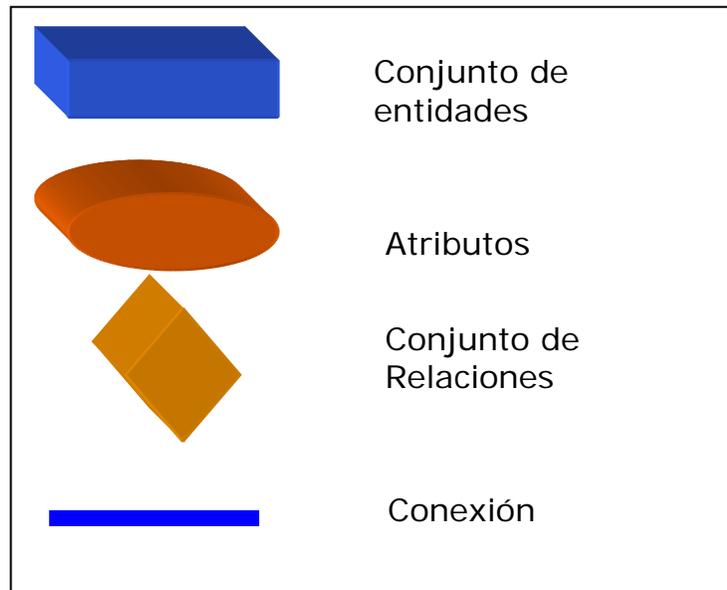


Fig. 2.5 Diagramas de Entidad

Cardinalidad de las Relaciones

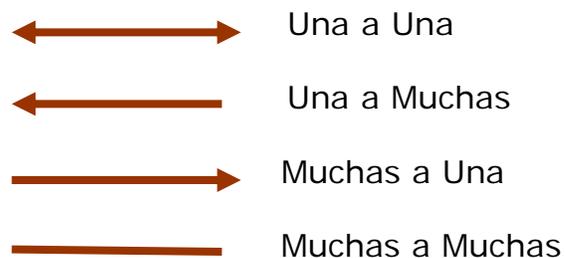


Fig. 2.6 Cardinalidades de la Relación

2.2.6 Generalización y Especialización

La generalización consiste en hacer mínimas la redundancia en la base de datos de manera que se puedan ocultar las diferencias entre entidades formando entidades comunes.

La especialización consiste en reducir el espacio de almacenamiento necesario por la base de datos en el medio físico, esto tiene como consecuencia una redundancia necesaria.

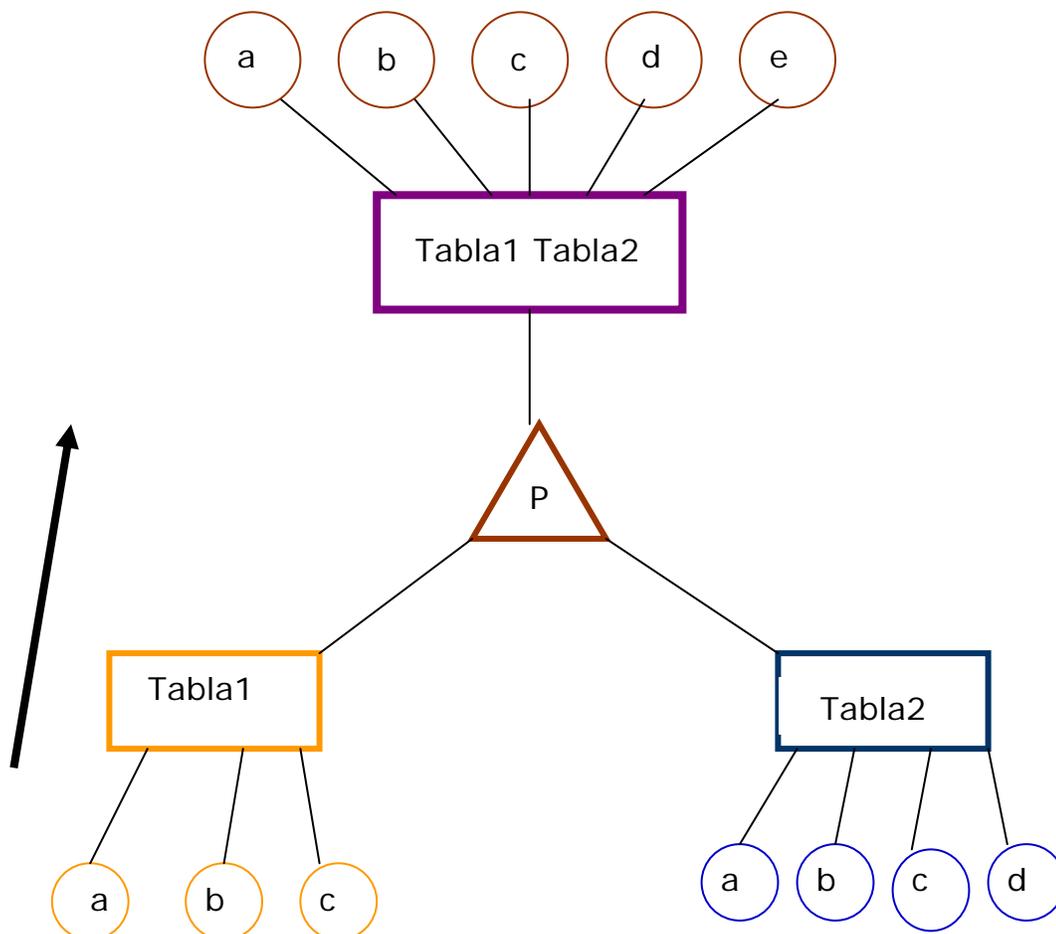


Fig. 2.7 Generalización

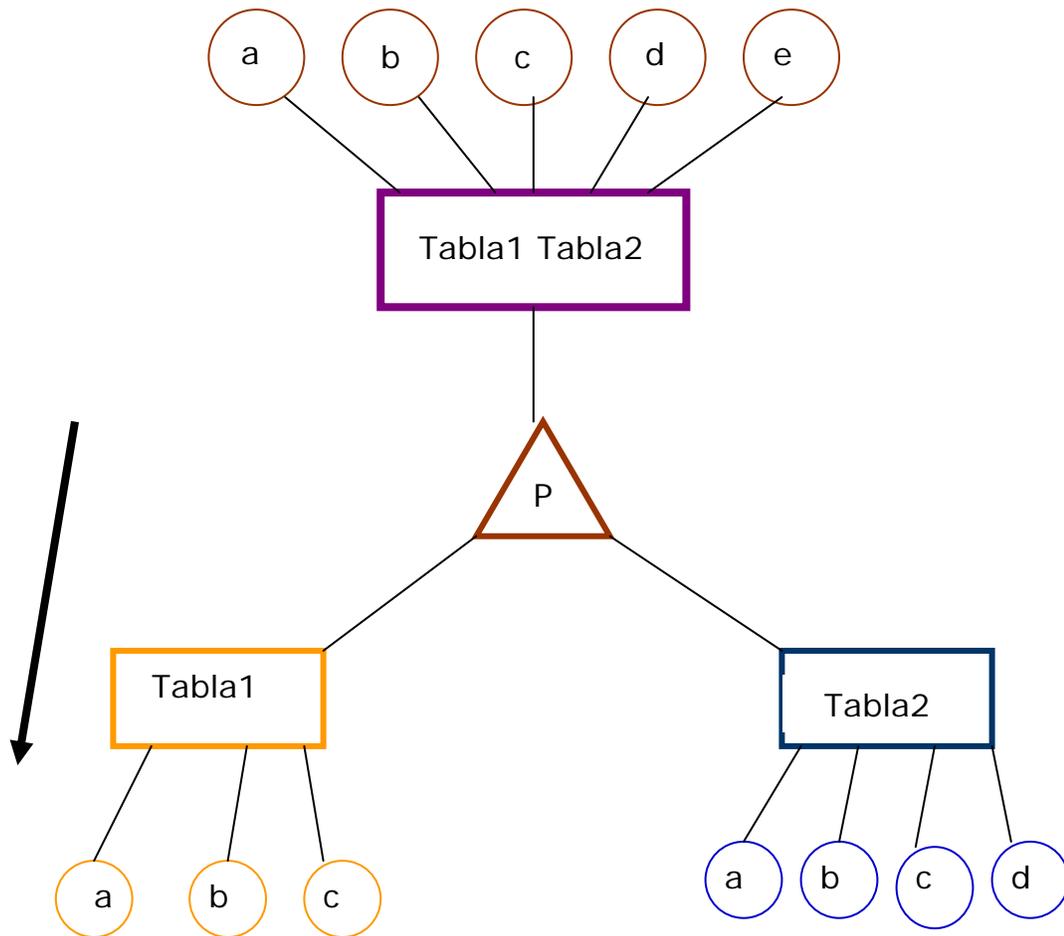


Fig. 2.8 La Especialización

2.3 Modelo relacional

Las estructuras de datos que se manejan en el modelo relacional corresponden a los conceptos de relación, entidad, atributo y dominio, los cuales se introducen aquí intencionalmente:

Relación.

Por una relación se entiende una colección o grupo de objetos que tienen en común un conjunto de características o atributos.

Entidad.

Es una unidad de datos en una relación con un conjunto finito de atributos. Es también conocido como *n-ada*, a raíz de que consiste de *n-valores*, uno por cada atributo.

Atributo.

También llamado *característica*, cada atributo de una relación tiene asociado un dominio en el cual toma sus valores.

Dominio.

Es un conjunto de valores que puede tomar un atributo en una relación.

2.3.1 Estructuras de las bases de relacionales

Una base de datos relacional es un conjunto de relaciones normalizadas. Para representar el esquema de una base de datos relacional se debe dar el nombre de sus relaciones, los atributos de éstas, los dominios sobre los que se definen estos atributos, las claves primarias y las claves ajenas.

2.3.2 Lenguajes formales de consulta

Al sistema DML se le conoce como álgebra relacional ya que permite especificar operaciones de consulta por medio de pasos intermedios de generación de tablas usando formatos especiales. En el sistema DML o álgebra relacional no se pueden utilizar los operadores lógicos.

En el álgebra relacional hay dos clasificaciones de procesos en donde en cada uno de ellos se toman uno o dos tablas como entrada y obtenemos una tabla de salida. Estas clasificaciones son:

- Operaciones tradicionales:
 - Unión (union)
 - Intersección (Intersect)
 - Diferencia (Minus)

Producto Cartesiano (Times)

Estos operadores necesitan que las tablas a operar tengan igual cantidad de atributos y sus dominios correspondientes sean semejantes o congruentes.

- a) Unión, constituye una tabla que contiene a todas las tablas que aparecen en una o ambas tablas. $\langle \text{tabla1} \rangle \text{ unión } \langle \text{tabla2} \rangle$.
- b) Intersección, produce una tabla que contiene las tablas que aparecen en las dos tablas. $\langle \text{tabla1} \rangle \text{ intersect } \langle \text{tabla2} \rangle$.
- c) Diferencia, produce una tabla que contiene todas las tablas de la primer tabla operando que no aparecen en la segunda. $\langle \text{tabla1} \rangle \text{ minus } \langle \text{tabla2} \rangle$.
- d) Producto cartesiano, hace una tabla que tiene todas las posibles concatenaciones entre los elementos de las tablas utilizadas. $\langle \text{tabla1} \rangle \text{ times } \langle \text{tabla2} \rangle$.

- Operadores Especiales

Select

Project

Join

Divide

- a) Select, es un formato el cual se encarga de extraer tablas que satisfacen una condición. $\langle \text{tabla1} \rangle \text{ where } \langle \text{condición} \rangle$.
- b) Project, nos permite filtrar atributos en la tabla resultante, especificando a los que se quiere obtener. $\langle \text{tabla1} \rangle (\langle \text{Lista de atributos} \rangle)$.
- c) Join, se construye una tabla partiendo de dos tablas específicas, así obtendremos todas las combinaciones posibles con los elementos de estas y mostrando a las que satisfacen una condición determinada. $\langle \text{tabla1} \rangle \text{ join } \langle \text{tabla2} \rangle \text{ where } \langle \text{condición} \rangle$. Podemos ver que es

semejante a una consulta sobre una tabla global cuando tienen que ver dos tablas y una condición.

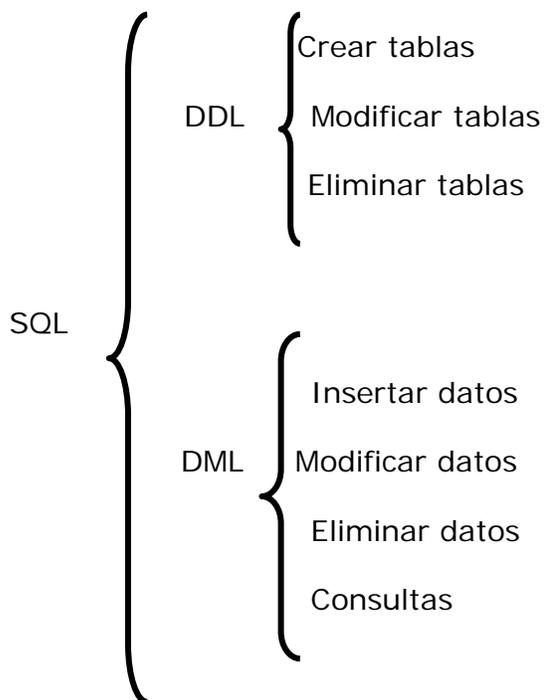
- d) Divide, toma dos tablas una de ellas con grado $(M+N)$ y otra de (N) , realiza una tabla de grado M que cuenta con todos los valores m de la relación $(M+N)$ donde su complemento es igual a todos los valores de la relación de orden N . $\langle \text{tabla1} \rangle \text{divide} \langle \text{tabla2} \rangle$.

2.3.3 Lenguajes comerciales de consulta

Uno de los lenguajes con el cual se pueden realizar un modelo relacional, es el lenguaje considerado estándar es el Structured Query Lenguaje conocido como SQL, este es un lenguaje de consulta estructurado el cual proporciona formatos y sintaxis para el uso y definición de datos.

El SQL es una base de datos relacional cliente-servidor basada en el lenguaje de consultas SQL.

Los sistemas de base de datos relacionales son una aplicación de la teoría matemática de los conjuntos al problema de la organización de datos. Estos datos son organizados en tablas.



DDL.- Cuando vamos a crear una tabla debemos de especificar el nombre de esta al igual que el nombre de las columnas y el tipo de datos de las columnas. Cada tabla debe de tener sus propios nombres en las columnas. No deberá de existir un campo con el mismo nombre en dos tablas y tenga significado diferente.

Los tipos de datos validos son:

- Char (<LONG>)(VAR)
- FLOAT
- INTEGER
- SMALLINT

La sintaxis para crear una tabla es:

```
Create Table nombre de la tabla  
Nombre columna tipo dato /Null/Not null/
```

Un ejemplo sería:

```
Create Table Alumnos  
Pk_NoControl      int  
Nom_Alumno        var char (40)  
Id_Carrera        var char (20)  
Id_Semestre       var char (15)
```

Si por alguna razón se necesita borrar una tabla los procedimientos guardados que hagan referencia a la tabla eliminada se deben eliminar con la instrucción DROP VIEW o DROP. PROCEDURE su sintaxis es:

```
DROP TABLE Nombre de la Tabla
```

Con ALTER se adiciona y borra columnas en una tabla.

ALTER TABLE nombre de la tabla **ADD** nom columna var char (20)

/null/not null

DML.- Los datos que va a utilizar el usuario se basan en los tipos de datos del sistema SQL, se pueden utilizar cuando varias tablas deben almacenar el mismo tipo de datos en una columna y desea estar seguro de que tales columnas tienen exactamente el mismo tipo de datos, longitud y condiciones que la aceptación de valores NULL.

- Para insertar datos a la tabla:

```
INSERT INTO <nombre tabla>
```

```
(<campo1>,<campo2>..):
```

```
<<valor1>,<valor2>...>
```

- Para Modificar datos

```
UPDATE <nombre tabla>
```

```
Set<campo1>=<campo1>
```

```
<campo2>=<campo2>,...(where<condición>)
```

- Para eliminar datos

```
DELETE<nombre tabla>
```

```
(where<condición>)
```

- Para consultar datos en una tabla

```
SELECT(UNIQUE) <lista de campos/*>
```

```
FROM <nombre>tabla>
```

```
(where <condición>)
```

```
(ORDER BY <campo>(ascendente/descendente))
```

2.3.4 Modificación de la base de datos

Tal modificación se hace usando el operador asignación, estas asignaciones se hacen a relaciones que ya existen en la base de datos.

- Eliminación en este caso se quitan las tablas seleccionadas de la base de datos, solo se eliminan las tablas completas; no se eliminan solamente sus valores de determinados atributos.
- Inserción, para poder insertar datos en una relación se tiene que especificar la tabla que se va a insertar o escribimos una consulta la cual arrojará como resultado un conjunto de tablas que se van a insertar.
- Actualización, hay ocasiones que por alguna razón tenemos la necesidad de cambiar un valor en la tabla sin que se modifiquen todos los valores, es cuando utilizamos el operador actualización.

2.3.5 Vistas

Una vista es una especie de tabla virtual o una consulta almacenada. Los datos que tiene una vista no están almacenados en un objeto diferente a la base de datos, lo que está almacenado en la base de datos es un SELECT, el resultado de dicha instrucción es la que forma la tabla virtual que la vista devuelve.

Las vistas se utilizan para algunas de los siguientes ejemplos o para todos:

- ◆ Restringir el acceso del usuario a filas concretas de una tabla
- ◆ Restringir el acceso del usuario a columnas específicas.
- ◆ Combinar columnas de varias tablas de forma que parezcan una sola.
- ◆ Agregar información en lugar de presentar los detalles.

Su formato es:

```
DEFINE VIEW <nombre vista>  
((identif_campo1, identif_campo2,...))  
AS<operación consulta>
```

CAPÍTULO III

¿Qué es Visual Basic 6.0?

3.1 Introducción ¿qué es visual basic 6.0?

Visual basic es un lenguaje de programación, el cual sirve para crear programas o aplicaciones.

Un lenguaje de programación está formado por un conjunto de sentencias que representan ordenes dadas a la computadora, cada sentencia es igual a muchas ordenes o instrucciones que debe llevar a cabo la máquina.

Este lenguaje es perfecto para crear aplicaciones de uso general como:

- ◆ Aplicaciones y utilitarios para Windows de cualquier índole.
- ◆ Aplicaciones que manejen base de datos (Access) pequeñas y medianas.
- ◆ Aplicaciones multimedia o publicaciones electrónicas en CD.
- ◆ Juegos sencillos.

Visual basic tiene algunas limitaciones como son:

- ◆ Crear juegos o multimedia de alto vuelo, (juegos de tres dimensiones).
- ◆ Manejar base de datos gigantes.

Con Visual Basic podemos crear desde una simple calculadora hasta una hoja de cálculo.

Este programa nos permite crear ventanas, botones, menús, etc. Cualquier otro elemento de Windows de una manera fácil

BASIC se compone de las siglas en ingles Beginner's All-purpose Symbolic Instruction Code.

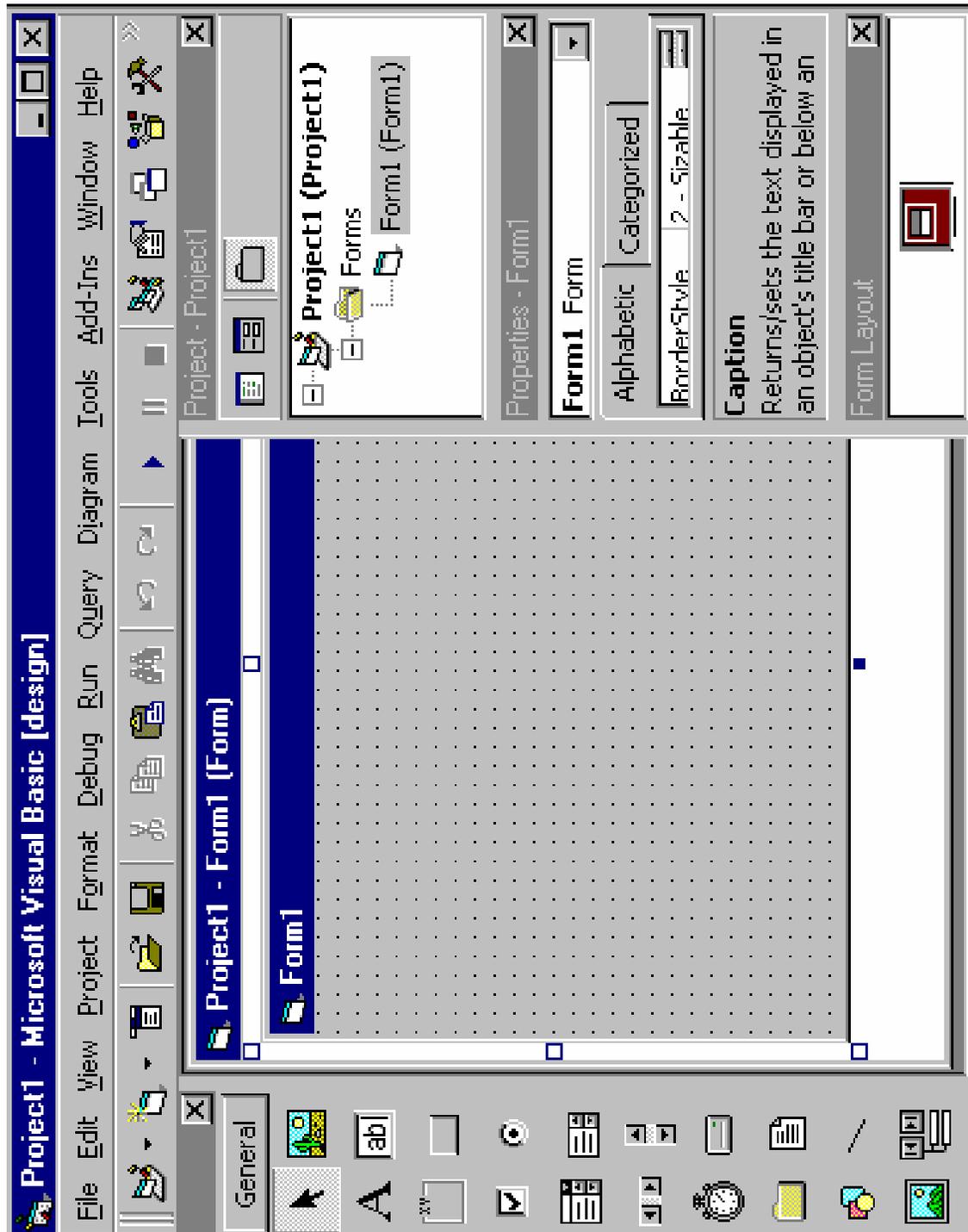


Fig. 3.1 Ventana principal de Visual Basic 6.0

3.2 El entorno de visual basic 6.0

Tiene todos los elementos que caracterizan a los programas de windows, el entorno de visual basic 6.0 es muy lógico y natural, podemos tener una descripción de la mayoría de los elementos. Integra funciones de diseño, edición, compilación, y depuración de proyectos mediante una interfase gráfica.

3.2.1 La barra de menús y las barras de herramientas

Mediante esta de barra podemos tener acceso a todas las funciones incorporadas en visual basic. Esta barra es similar a las otras aplicaciones de windows. Existen otras barras a las cuales se hace acceso más rápido y se derivan de la barra de menú, estas barras son las barras de herramientas las cuales tienen una serie de botones con los que se hace acceso de una forma más rápida.

Existen cuatro barras de herramientas las cuales son:

- A) Barra de herramientas estándar, esta barra es la que aparece por default, ésta cuenta con 21 botones. Los cuales corresponden a las funciones más importantes.

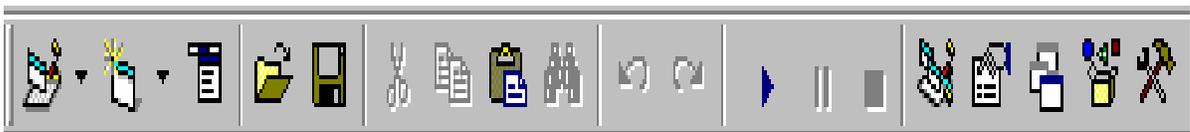


Fig. 3.2 Barra de Herramientas Estándar

A continuación mencionaremos el nombre de cada icono según se muestran:

1. Agrega un proyecto nuevo
2. Incorpora un nuevo formulario
3. Editor de menús
4. Abre un Proyecto

5. Guarda un proyecto
6. Corta
7. Copiar
8. Pegar
9. Buscar
10. Deshacer
11. Rehacer
12. Ejecutar
13. Pausa
14. Detener
15. Explorador de proyectos
16. Ventana de propiedades
17. Posición del formulario
18. Explorador de objetos
19. Caja de herramientas
20. Ventana de datos
21. Administrador visual de componentes

B) Barra de Herramientas Debug (depuración), esta tiene botones para la depuración de programas.

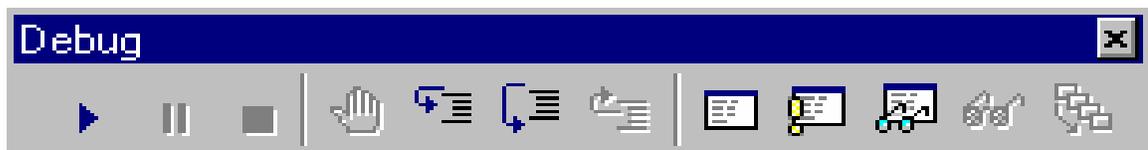


Fig. 3.3 Barra de Herramientas Debug

C) Barra de herramientas edición (Edit), incluye funciones de edición que son necesarias al momento de escribir el código fuente.



Fig. 3.4 Barra de Herramientas Edit

D) Barra de herramientas editor de formularios (Form Edit), los botones que tiene permiten retocar los controles insertados en los formularios.

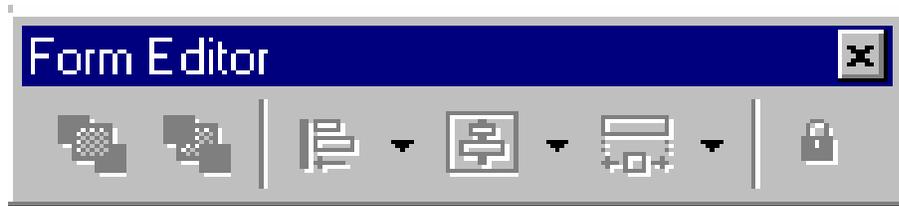


Fig. 3.5 Barra de Herramientas Editor de Formularios

3.2.2 Las herramientas toolbox

La caja de herramientas contienen la mayoría de los objetos o controles que se pueden hacer en los formularios o pantalla de aplicación como pueden ser botones, etiquetas, cajas de texto, etc.

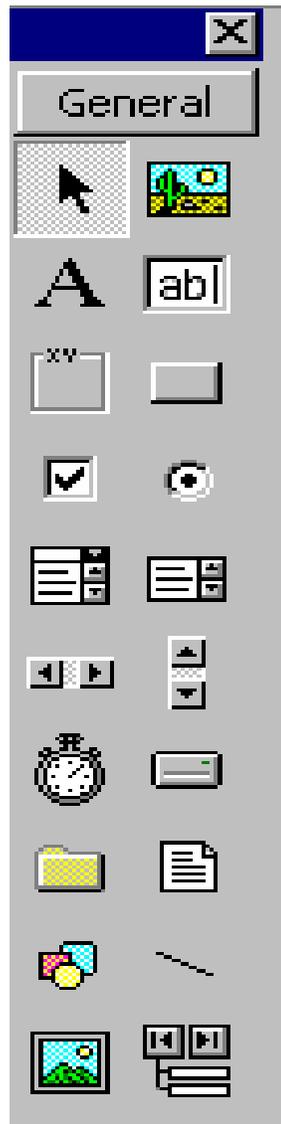


Fig. 3.6 Barra de Herramientas ToolBox

A continuación nombraremos a los iconos de la barra antes expuesta:

1. Puntero
2. Caja de imagen
3. Etiqueta

4. Caja de texto
5. Marco
6. Botón de comando
7. Caja de verificación
8. Botón de opción
9. Caja combinada
10. Caja de listado
11. Barra de desplazamiento horizontal
12. Barra de desplazamiento vertical
13. Temporizador
14. Caja de listado de unidades
15. Caja de listado de directorios
16. Caja de listado de archivos
17. Figura
18. Línea
19. Control de imagen
20. Dato
21. OLE

El número de controles que puede aparecer en la ventana varía con la configuración del sistema, visual basic, nos permite tener controles personalizados y utilizar otros de terceros para agregar funcionalidad a las aplicaciones. Para esto utilizamos el menú project y elegimos la opción components (ctrl++) y obtenemos la siguiente ventana de dialogo.

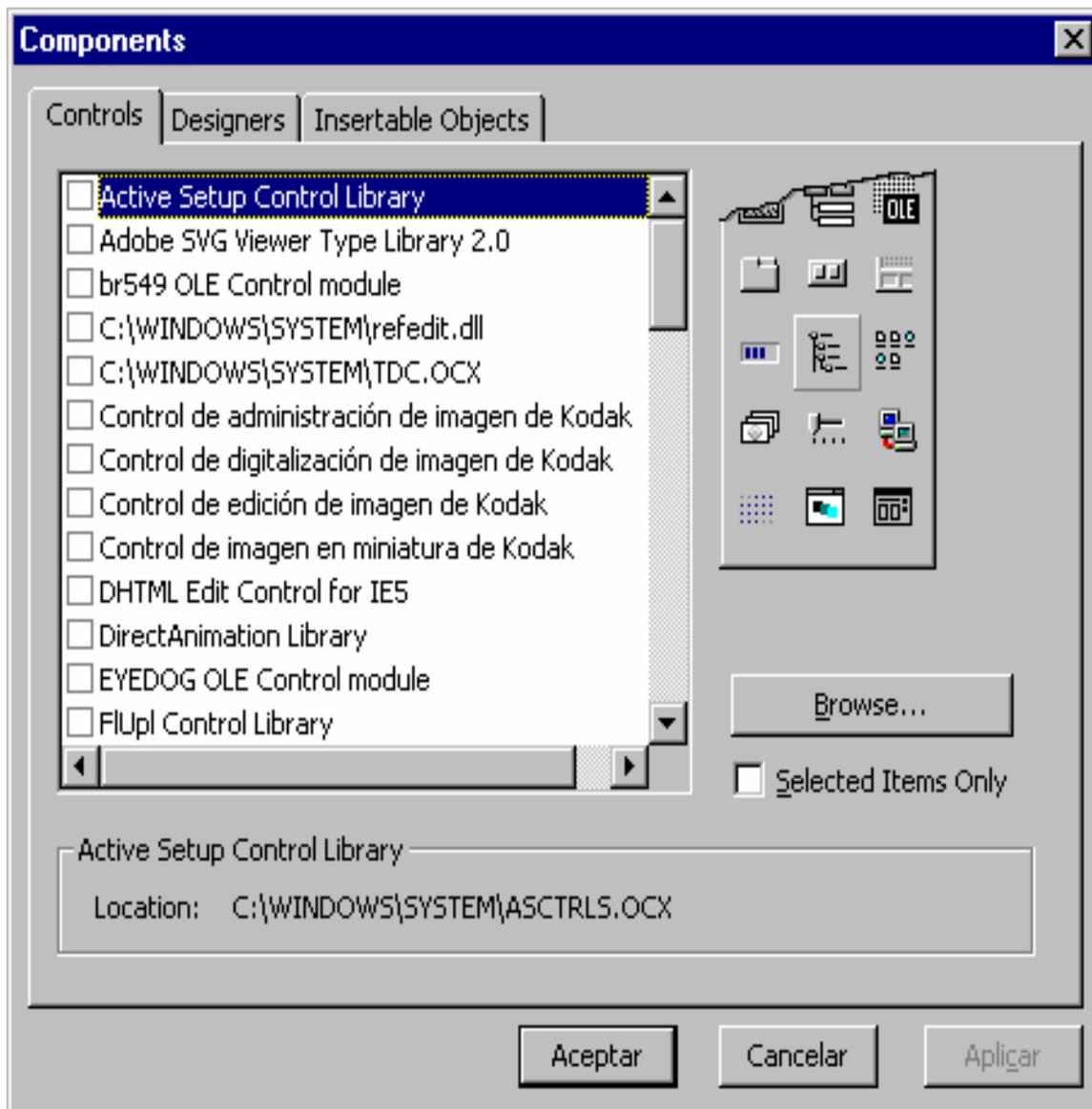


Fig. 3.7 Ventana de Componentes

3.3 Formularios (forms) y modulos

Estos formularios son las ventanas sobre las que se diseña el programa y donde se ubican los controles de la toolbox.

Cuando se ejecuta el programa el **form** se convierte en la ventana de aplicación, donde aparecerán los botones, el texto y gráficos. Cuando se está haciendo un formulario este aparece compuesto de forma cuadrículada de puntos, llamada *grid*, la cual nos ayuda a alinear los controles manualmente de una manera precisa, sin tener que poner coordenadas en cada momento. Cuando se termine el diseño del programa esta malla ya no aparecerá; al momento de ejecutar el programa la *grid* desaparece. Esta se puede modificar haciendo acceso al menú tools, Options, poniendo el valor deseado.

El formulario y los controles en él situados serán el esqueleto o la base del programa. Una aplicación puede tener varios formularios pero solo habrá uno con el que va a arrancar la aplicación, este formulario se determina a partir del menú Project/Properties, es startup Objects.

Cuando se crea un programa en visual basic 6.0 se diseña y se prepara la parte gráfica como son formularios, botones, menús, etc., y por último se debe realizar la programación que gestione la respuesta del programa ante los diferentes eventos.

3.4 La ventana de proyecto (project)

Es una ventana por medio de la cual accedemos al código de un módulo, formulario, o al diseño del mismo formulario. Para cada uno de los componentes el explorador de proyectos aparece del lado izquierdo el nombre del objeto y del lado derecho entre paréntesis es el nombre con el que se guardó el objeto en disco.

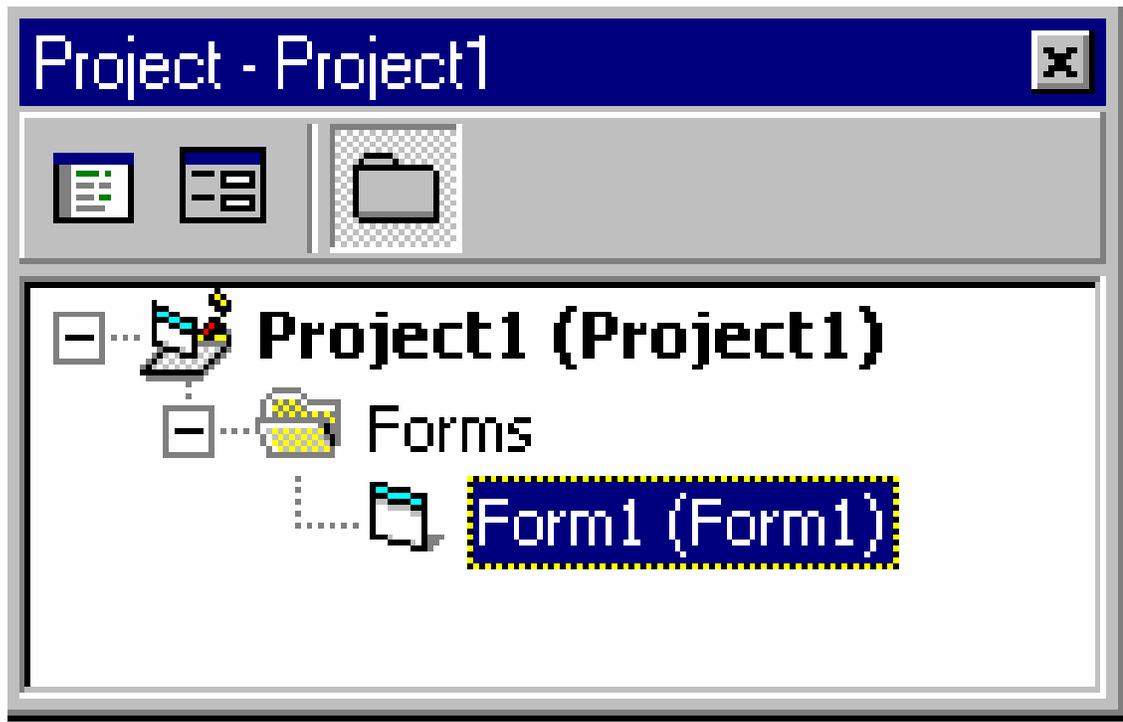


Fig. 3.8 Ventana de Proyecto

3.5 La ventana de propiedades

Con esta ventana se pueden cambiar las propiedades de los objetos cuando se está llevando a cabo el diseño. Estas propiedades con las que cuentan los objetos se definen: su nombre (name), su etiqueta (caption), el texto que contiene (text), tamaño, posición, si esta activado o desactivado (enable), etc. Estas propiedades son almacenadas dentro de cada formulario como una estructura.

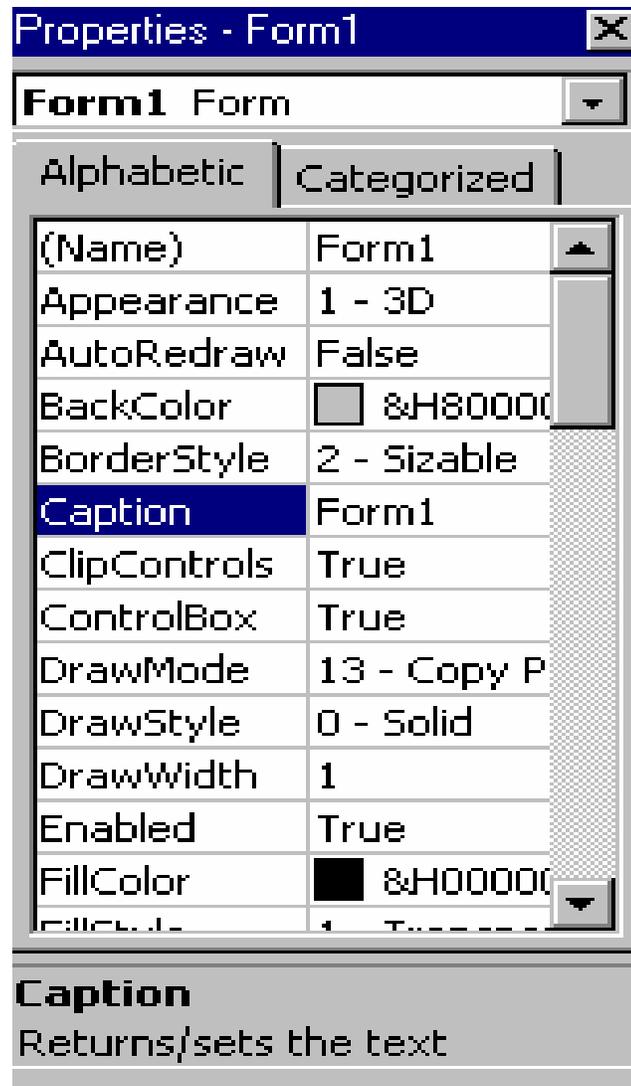


Fig. 3.9 Ventana de Propiedades

3.6 Creación de programas ejecutables

Para poder hacer de nuestra programación de la nueva aplicación la creación de un programa ejecutable se utiliza el comando make nombre Proyecto.exe dentro del menú file, aquí se genera un fichero con la extensión de ejecutable (.exe). Para tener una mejor información sobre cuales son los ficheros necesarios se consulta el fichero .vbp ya que contiene los datos completos del proyecto. Para que el programa corra se necesita el fichero MSVBVM60.DDL esta instalado en el directorio c:/Windows/system o c:/winnt/sustem32. (.exe).

Al instalar el compilador los ficheros que se necesitan se instalan automáticamente. Si se utiliza el programa en una computadora donde no está instalado visual basic 6.0 se pueden construir unos disquetes de instalación que hagan menos pesada la instalación del programa en cualquier computadora sin tener que ver en cada caso cuales son los ficheros que faltan.

Visual basic 6.0 cuenta con un asistente (*wizard*) el cual simplifica la creación de estos disquetes, este se encuentra en el mismo grupo de programas que visual basic y se llama *Package and Deployment Wizard*.

3.7 Utilización del code editor

Este editor de código es la ventana donde se describen las sentencias del programa. Para abrir esta ventana nos posicionamos en el menú View y tomamos la opción Code.

Esta ventana code editor tiene o cuenta con algunos colores para facilitarnos el manejo de ella, maneja el color azul con las palabras claves de basic, maneja el color rojo para mostrarnos los errores y el color verde para los comentarios; con el uso de estos colores se corrige con más facilidad.

Para poder ver los procedimientos del formulario en la misma ventana al igual que sus controles o ver solo un procedimiento se utilizan los dos botones que están ubicados en la parte inferior izquierda de la siguiente ventana.

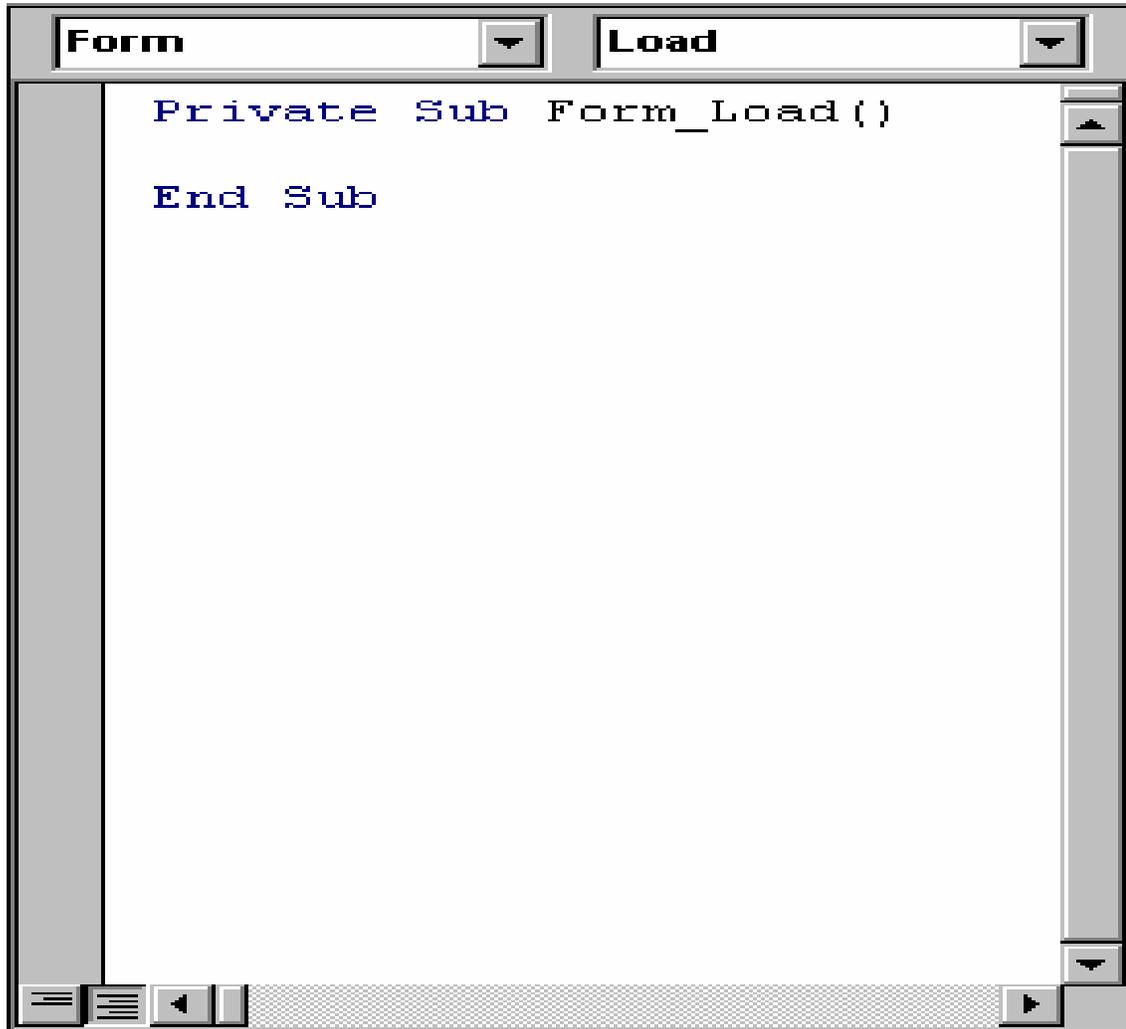


Fig. 3.10 Ventana Code Editor

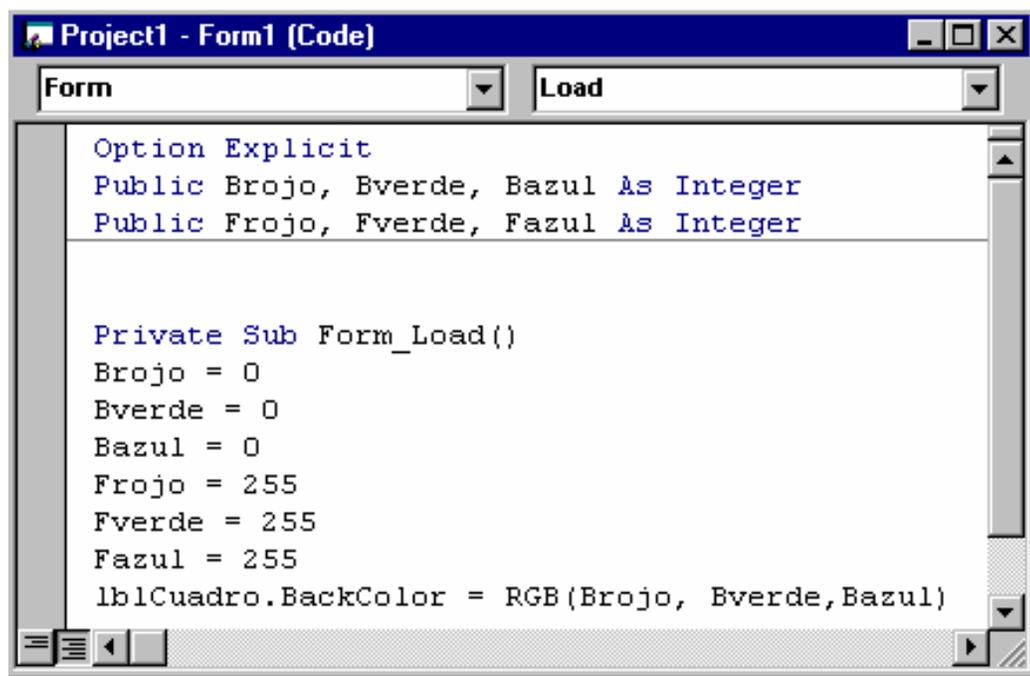


Fig. 3.11 Ventana del Project1

Donde el primero es el *procedure view* y el segundo el *full module view*. También se puede completar automáticamente el código (*automatic completion code*), en la figura siguiente se ve como cuando se teclea el punto o la letra inicial de una propiedad después del punto detrás del nombre del objeto en ese momento se abre una lista con las propiedades del objeto. Con la tecla tab se introduce el nombre completo de la propiedad seleccionada. A esa acción se le conoce como *auto list members*.

La opción *autoquickinfo*, cuando se empieza a poner el nombre de una función aparece información sobre esta función: nombre, argumento y valor de retorno.

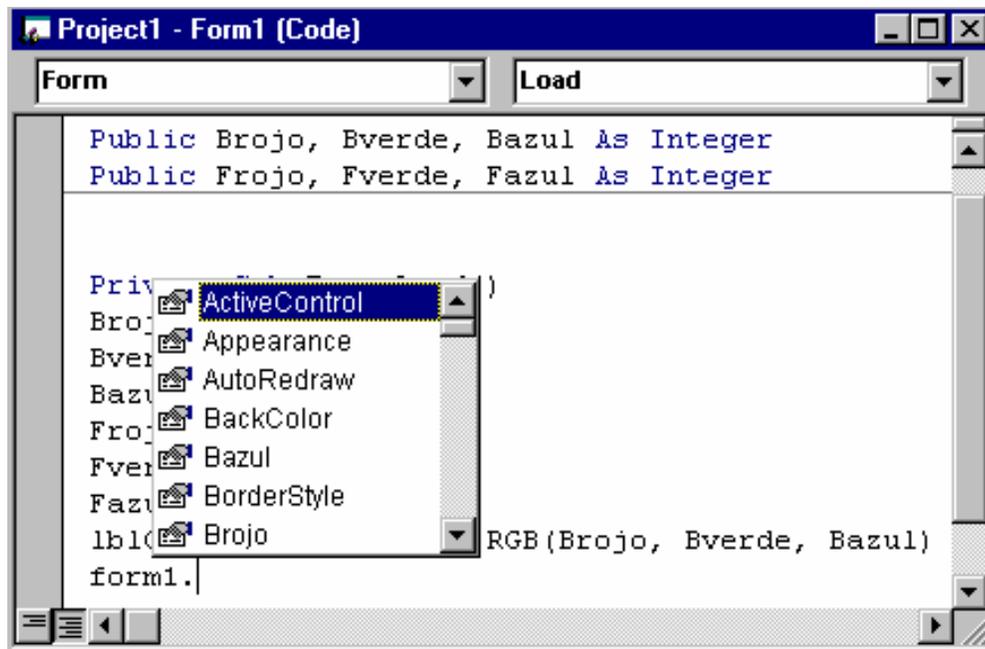


Fig. 3.12 Inserción automática de propiedades

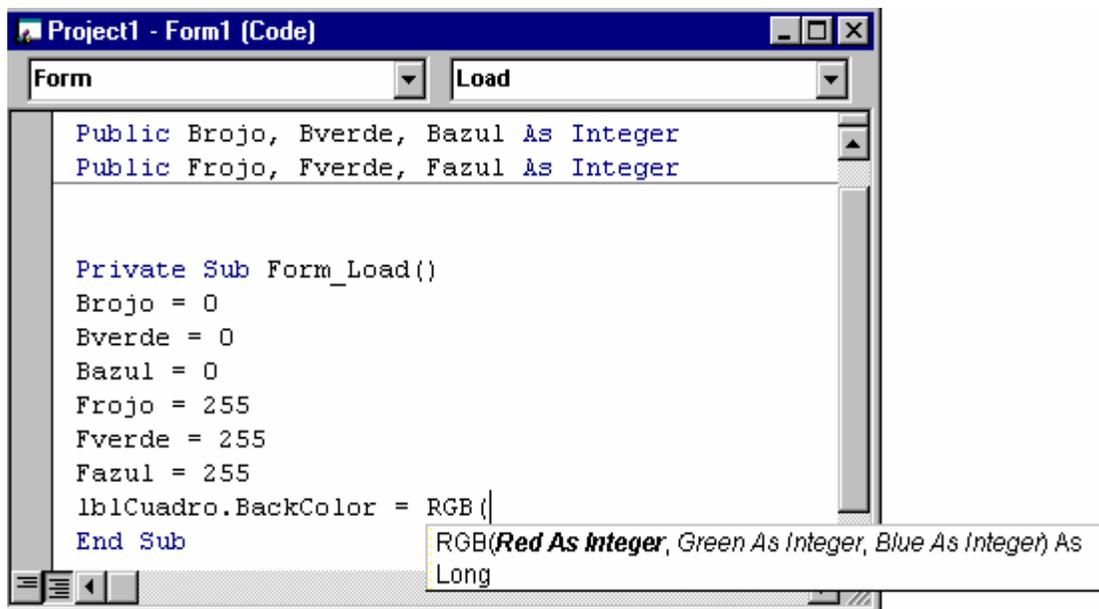


Fig. 3.13 Ayuda para la inserción de Funciones

Estas dos opciones se activan en el cuadro de dialogo que obtenemos de *Tools/Options/Editor*.

3.8 Utilización del debugger

La utilización del debugger es para la detección de errores y corrección de errores, este permite ejecutar parcialmente el programa, deteniéndose dicha ejecución en un punto deseado y viendo en cada momento el valor de cada una de las variables. Así es como es más sencillo descubrir los errores.

El Debugger a parte de que ya sabe que sentencia es la que sigue para ejecutarse con *debug/show next statement*, sino que también va a decidir cuál va a ser dicha sentencia, con *debugger /set next statement*, con esto puede estar cambiando el curso habitual de la ejecución, saltando sentencias, volviendo a una sentencia ya ejecutada.

Visual basic da información sobre las llamadas a funciones y procedimientos, esto lo podemos obtener por medio del comando *view/call stack* o con el botón *debug* y así sabremos que función ha llamado a que función, hasta la sentencia donde la ejecución esta detenida.

3.8.1 Ejecución controlada de un programa

Si de alguna manera necesitáramos ejecutar parcialmente el programa se pueden incluir *break points* que son puntos de parada en la ejecución, en las líneas de código necesarias.

Estos *breakpoints* se indican con un punto grueso en el margen seguido de un cambio de color en la línea.

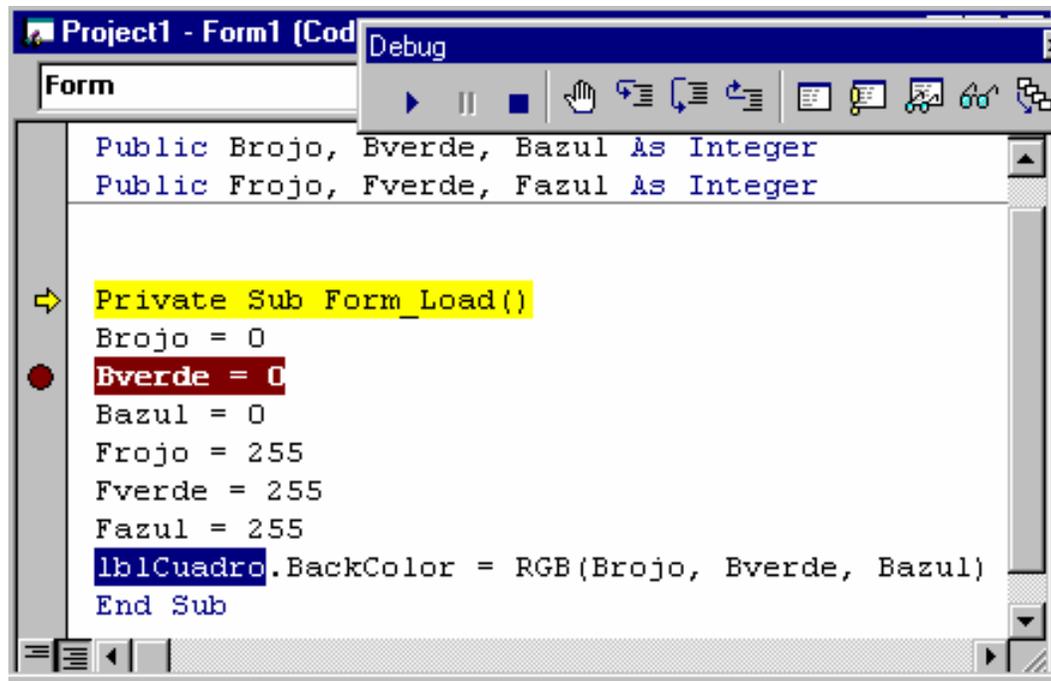


Fig. 3.14 Utilización del Debugger

Al colocar el *breakpoint* en una línea determinada cuando el programa se esté ejecutando se va a detener en esa línea. Para esto se utiliza la opción *toggle breakpoint* del menú *debug*.

También podemos ejecutar el programa línea por línea o paso a paso, esto es posible con F8 o haciendo acceso al menú *RUN* con la opción *step into*.

3.8.2 Ventanas immediate, locals y watches

Para poder consultar el valor de las variables y propiedades el debugger de visual Basic 6.0 tiene tres ventanas, así como para probar su funcionamiento. Estas ventanas son:

- a) La ventana immediate, esta ventana realiza diversas acciones como son:

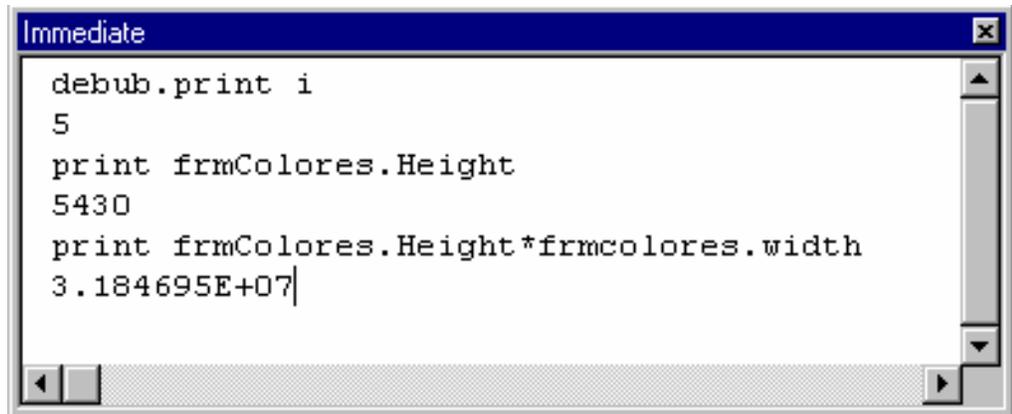


Fig. 3.15 Ventana Immediate

1. Imprime el valor de cualquier variable y/o propiedad accesible la función o procedimiento que se esta ejecutando, se hace con *print var name*, directamente en la ventana o poniendo en el código del programa sentencias del tipo *debug.print var name*. Esto se escribe en la ventana immediate sin tener que parar el programa durante su ejecución, estas sentencias son guardados en el formulario por lo tanto no hay que volver a escribirlos cuando se haga otra ejecución.
 2. Asigna valores a variables y propiedades cuando se está detenida la ejecución y sigue la ejecución con valores nuevos sin poder crearse nuevas variables.
 3. Ejecutar expresiones y probar funciones y procedimientos incluyendo la ventana immediate la llamada correspondiente.
- b) La ventana Locals, aquí se ven los valores de las variables visibles en el procedimiento en el que esta detenida la ejecución.

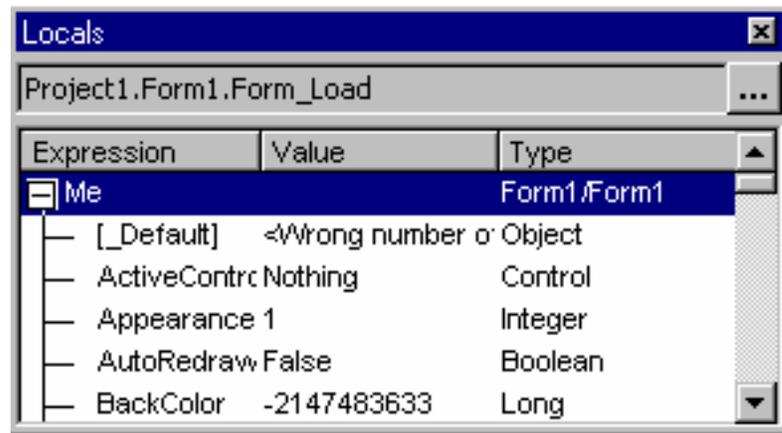


Fig. 3.16 Ventana Locals

- c) Para saber el valor de una variable rápidamente se utiliza la ventana *quick watches*, para observar continuamente el valor de una expresión o variable hay que añadirla a la ventana *Watches*, lo podemos hacer con la opción *ADD watch* del menú *debug*, el valor de las variables de esta ventana se actualizan automáticamente.

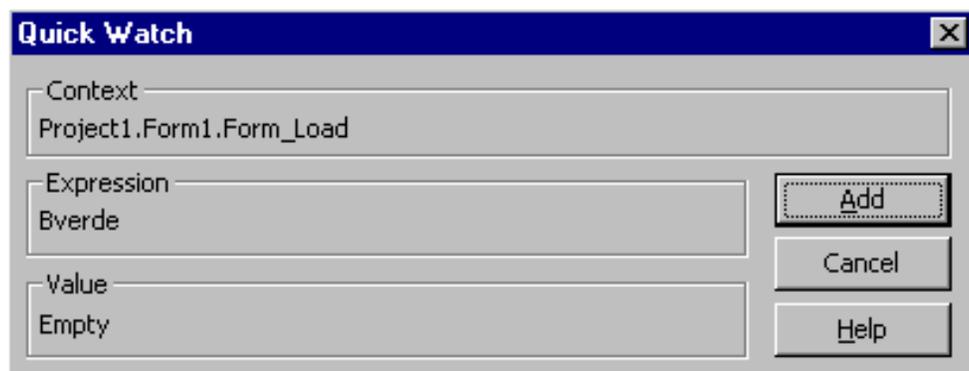


Fig. 3.17 Ventana Quick Watch

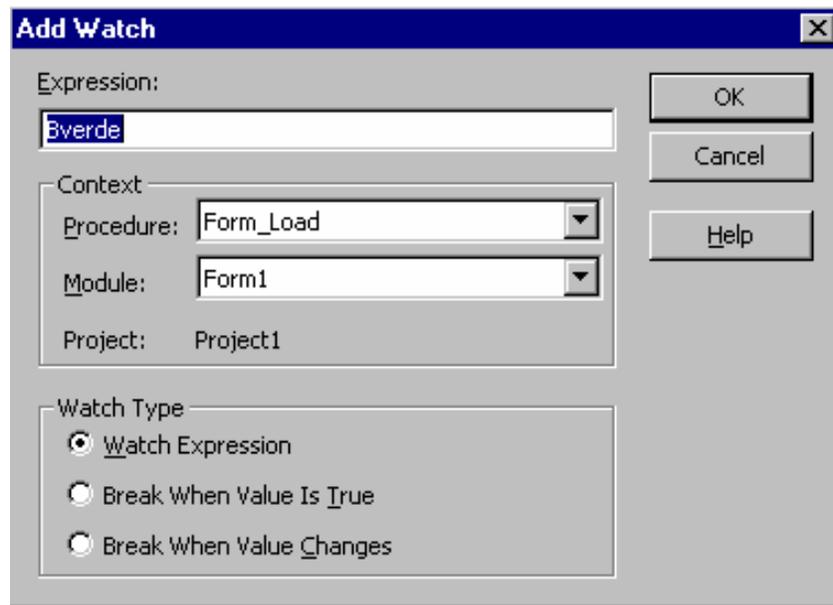


Fig. 3.18 Ventana Add Watch

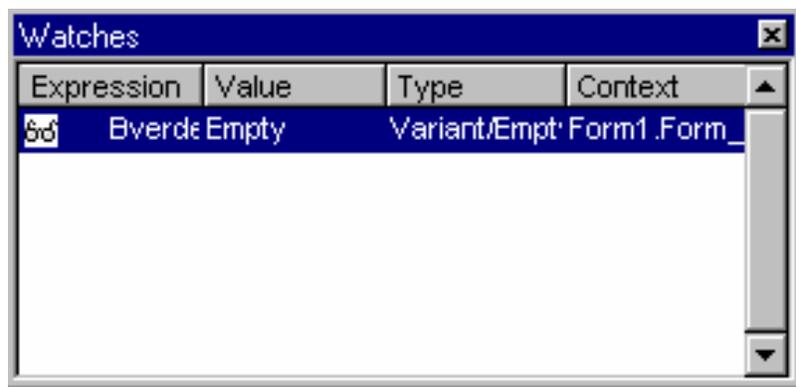


Fig. 3.19 Ventana Watches

Indicando también cuando no son accesibles desde el procedimiento que se esta ejecutando en la ventana *add watches* se pueden poner *breaks* o paradas del programa condicionales, cuando se cumple una condición o cuando cambia el valor de la variable.

Para vigilar el valor de las variables esta se activa del menú *debug* o con la barra de herramientas.



Fig. 3.20 Iconos de Watches

CAPÍTULO IV

"Diseño e Implementación del Sistema"

4.1 Análisis preliminar del sistema

Actualmente las organizaciones necesitan de un manejo rápido, seguro y claro de la información que generan en el desarrollo de sus operaciones, debido a las normas internas de la organización, como sería la optimización de los recursos propios de los negocios, y de las normas externas las cuales serían los requisitos de las autoridades.

Estas son algunas de las razones que dan origen a la aplicación de las computadoras y recursos informáticos en las diferentes organizaciones (negocios, escuelas, bancos, etc.); es así como la utilización de este medio la información puede procesar y almacenar de una manera organizada según las necesidades de cada usuario, ofreciendo a la vez confiabilidad y rapidez, ya que en cualquier momento se puede obtener la información generada, siempre y cuando el sistema esté desarrollado correctamente.

Este proyecto consiste en el estudio y desarrollo a través de un análisis del manejo del laboratorio de electrónica del Instituto Tecnológico de Celaya, en este caso es un laboratorio de electrónica donde el equipo con el que cuenta se presta a los alumnos de una manera manual, esto lo realizan utilizando unos vales los cuales llenan a mano. La forma de realizarlo es la siguiente:

Los alumnos llegan y tiene que formarse para poder pedir el equipo que necesitan en ese momento según su práctica y materia, tomando en cuenta que son varios alumnos la fila de espera es larga, lógicamente se pierde tiempo para los alumno y maestros; el tiempo de los alumnos es pérdida ya que tienen que esperar su turno y cuando este llega todavía tienen que llenar un vale que con los datos de ellos como son: nombre, carrera, semestre, equipo que necesita, etc.; en lo que el personal revisa este vale el alumno tiene que esperar hasta que se lo den para poder sacar el equipo solicitado. El alumno tendrá menos tiempo para hacer su práctica. El maestro pierde tiempo puesto que no puede empezar a explicar lo de la práctica ya que no esta todo el grupo con el material necesario. También se tiene el problema de que el alumno por alguna razón extravíe el vale.

Con ese método el personal pues no se da cuenta de cuantas veces a prestado determinado equipo y no es posible dar una fecha aproximada para que tal equipo tenga un mantenimiento preventivo, no se conoce el estado al 100% del equipo, por lo tanto si se tuviera un chequeo permanente de cada equipo y sus características como: si es nuevo o usado, que le falta alguna perilla, marca, modelo, etc., se tendría un mejor control del Laboratorio.

Descripción general del proyecto

Este proyecto consiste en el análisis y diseño del sistema de la “Automatización del préstamo de equipo e inventario del laboratorio de Ingeniería Electrónica del Instituto Tecnológico de Celaya”; con los procesos que implica como son el control de alumnos, control de equipo, control de los empleados o personal del laboratorio, control de las materias y horarios.

Todo esto con la finalidad de incrementar la precisión en el manejo de la información antes mencionada. Pues actualmente se incurre en la pérdida de tiempo tanto para el alumno como para el maestro, puesto que el trámite lo realizan manualmente provocando inconvenientes como:

- Llana de vales a mano para obtener datos del alumno y del equipo solicitado.
- El principal problema es la pérdida de tiempo que el alumno tiene, pues hay que esperar turno para poder sacar el equipo electrónico necesitado.
- Los alumnos que se encuentran al final de la fila les queda menos tiempo para realizar su práctica.
- Los maestros no pueden empezar su práctica hasta que su grupo se encuentre completo dentro del laboratorio con el equipo electrónico necesario.
- Es necesario saber el estado en el que se encuentra el equipo electrónico.
- Tener un control sobre el personal que atiende a los alumnos para la realización del préstamo de equipo de electrónica, como se maneja a más

de un encargado es importante mantener información restringida por si hubiera algún problema con el equipo, alumno o personal.

Estos problemas tenemos que minimizarlos o eliminarlos para lograr esto, se ha aceptado aportar información necesaria para posteriormente implementar un sistema controlado por una computadora que se encargue de minimizar los problemas antes mencionados.

El Instituto Tecnológico de Celaya, esta en la mayor disposición de adquirir una computadora que reúna los requisitos sugeridos, para poder llevar a cabo el sistema.

Objetivo General

Diseñar un sistema que mediante tecnología de punta permita a los alumnos y profesores hacer mejor uso de los instrumentos e instalaciones del laboratorio. El sistema consiste en que el alumno muestre su credencial del Instituto Tecnológico de Celaya, dicha credencial cuenta con un código de barras, haciendo uso de un lector óptico, el personal del laboratorio de electrónica del Instituto Tecnológico de Celaya lo deslizara por el código de la credencial de cada alumno, tendiendo así los datos de este y el equipo también contara con un código de barras para cuando el alumno solicite un equipo determinado se pase al equipo el lector óptico por el código de barras y obtener los datos de dicho equipo, el cual se le cargará al alumno que lo solicito, haciendo mas rápida la fila de espera de los alumnos para poder solicitar un equipo electrónico.

El sistema también contará con medidas de seguridad, para tener control con el personal del laboratorio de electrónica, contando con la fecha y hora en que se realizo el préstamo.

Se contará también con un control del estado del equipo, se llevara la cuenta de cuantas veces se ha usado el equipo, con esto se puede tener una visión mas

clara de que equipo necesita algún mantenimiento o la sustitución en que se encuentra determinado equipo y tomar la decisión de cambiarlo por uno nuevo.

Este sistema automatizado permite tener un mejor control para los siguientes puntos:

- Agiliza la entrega-recepción de los instrumentos.
- Confiabilidad en la entrega-recepción.
- Incrementa el número de hora-práctica.
- Identificación oportuna de deudores.
- Ahorro en vales de almacén.
- Lleva estadísticas de uso de los instrumentos o equipo electrónico
- Permite el mantenimiento preventivo y predictivo.
- Permite actualizar y valorar el inventario físico del laboratorio de electrónica.
- Justifica la adquisición y caducación de los equipos.

Objetivos particulares

Atención a alumnos, agilizar y hacer mas precisa y clara la captura y el procesamiento de la información de los alumnos de nuevo ingreso, así como los ya existentes, facilitando el acceso a su información personal y permitiendo una atención rápida en el momento de solicitar el equipo , además de realizar la información exacta tanto del alumno como del equipo, abarcando situaciones como: si el equipo fue prestado, si tiene algún defecto, si el equipo prestado fue devuelto en las mismas condiciones, quien lo tiene, a que hora solicito el préstamo, etc.

Con respecto al equipo del laboratorio de electrónica, se tiene un mayor control del mismo, en existencia según el estatus de cada uno, como puede ser que se

encuentre o no se encuentre en existencia, en que estado físico se encuentra, marca, modelo, número de serie, nombre, o si está prestado, número de veces que ha sido utilizado ese equipo electrónico, etc., Además de ofrecer la información propia de cada equipo con la finalidad de identificarlo para tomar decisiones pertinentes de acuerdo a su estado físico.

En cuanto a la administración del laboratorio de electrónica del Instituto Tecnológico de Celaya, permite tener un acceso rápido y seguro a la información generada en el ejercicio del laboratorio, teniendo la posibilidad de obtener información sobre equipo electrónico, usuarios, personal que realiza los préstamos; lo necesario para poder dar un informe mensual y semestral al quien lo solicite. Con esto tenemos una información más confiable y detallada de todo el manejo del laboratorio de electrónica del Instituto Tecnológico de Celaya.

Técnicas de recopilación de información

Las técnicas empleadas para recopilar la información requerida para el análisis y desarrollo del sistema serán las siguientes:

- Entrevista.- Consultar directamente a las personas involucradas en el control y manejo del laboratorio de electrónica. En este caso el encargado del laboratorio de electrónica para conocer sus puntos de vista y sus necesidades que serán cubiertas por el sistema.
- Observación.- La finalidad de ésta es enriquecer la información obtenida a través de la entrevista, ya que de esta manera se obtendrá un punto de vista mas preciso sobre la manera en que se desarrollará el sistema y de cómo éste debe comportarse, puesto que se ve la manera de realizar el préstamo de equipo electrónico o el manejo del laboratorio de electrónica.

Flujo de la información

Una vez realizada la entrevista y la observación obtuvimos los datos que necesitábamos para poder entender mejor el manejo que realizan en el laboratorio de electrónica, una vez esto realizado llegamos al siguiente flujo de información:

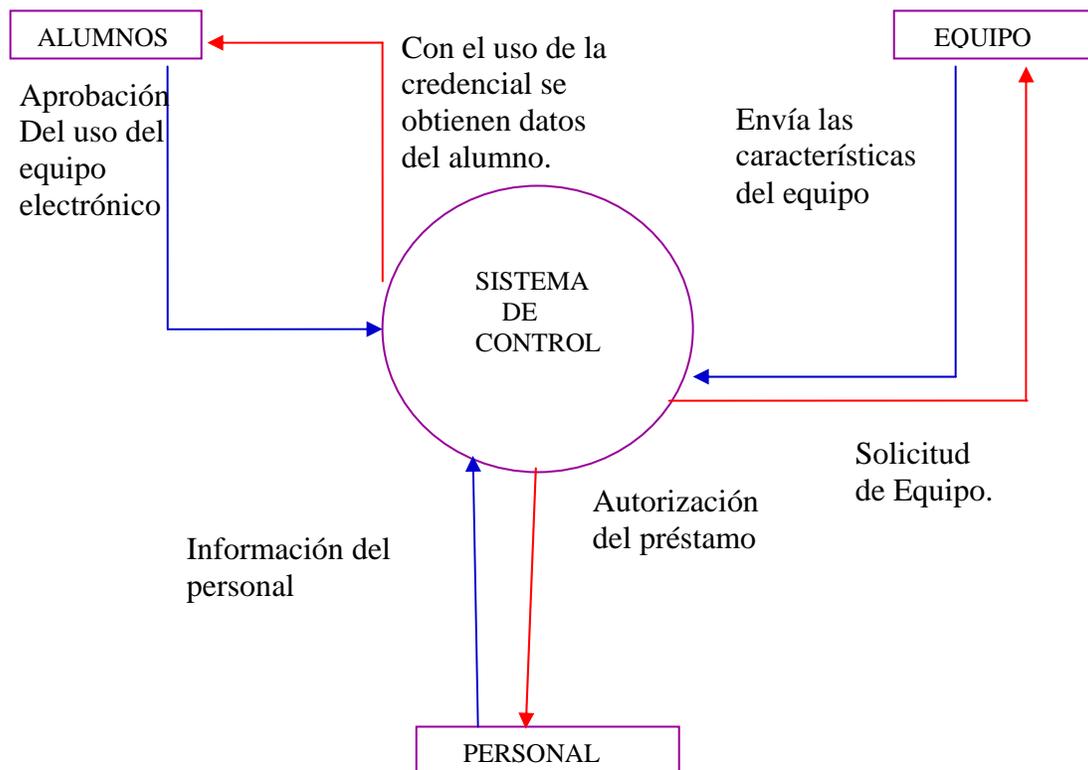


Fig. 4.1 Diagrama que muestra el Flujo de Información

El control de equipo o sistema de control del diagrama anterior, es una gráfica de todo lo que representa al laboratorio de electrónica en el sistema.

Alumno, equipo y personal.- Representan los factores externos con lo que interactúa el laboratorio de electrónica. También se consideran como "Las fronteras del sistema".

El equipo.- El control de existencia de equipo de electrónica en el laboratorio de electrónica, su estado físico, se está prestado o no, marca, modelo, número de serie, el alumno que lo solicitó.

El personal.- La persona encargada en ese momento del laboratorio de electrónica toma los datos de los alumnos por medio de su credencial del Instituto Tecnológico de Celaya y mediante la barrida con un lector óptico en la credencial que cuenta con código de barras se obtiene los datos del alumno que solicita el equipo, de igual manera cuando se presta el equipo se da una barrida con el lector óptico tomando los datos del equipo solicitado ya que el equipo electrónico también cuenta con un código de barras. Se tienen dos horarios para el personal, cada horario tiene su propio encargado, turno matutino, vespertino, por tal motivo cada una de estas personas cuenta con un password para tener un mejor control de la información y no tengan acceso a cierta información; y se protegen los préstamos de cada uno.

Alumno.- Solicita el préstamo del equipo del laboratorio de electrónica haciendo uso de su credencial la cual cuenta con un código de barras para que por medio de ella tomen sus datos con la barrida del lector óptico, estos datos son: nombre del alumno, materia, semestre, qué equipo es el que se le va a prestar. Cuando el alumno ya no necesite el equipo se le descarga de su registro que se tiene en el sistema, tomando nota de alguna observación que se diera, por ejemplo si lo está devolviendo con algún desperfecto el cual no tuviera el equipo electrónico.

De una manera mas detallada tenemos el siguiente flujo de información:

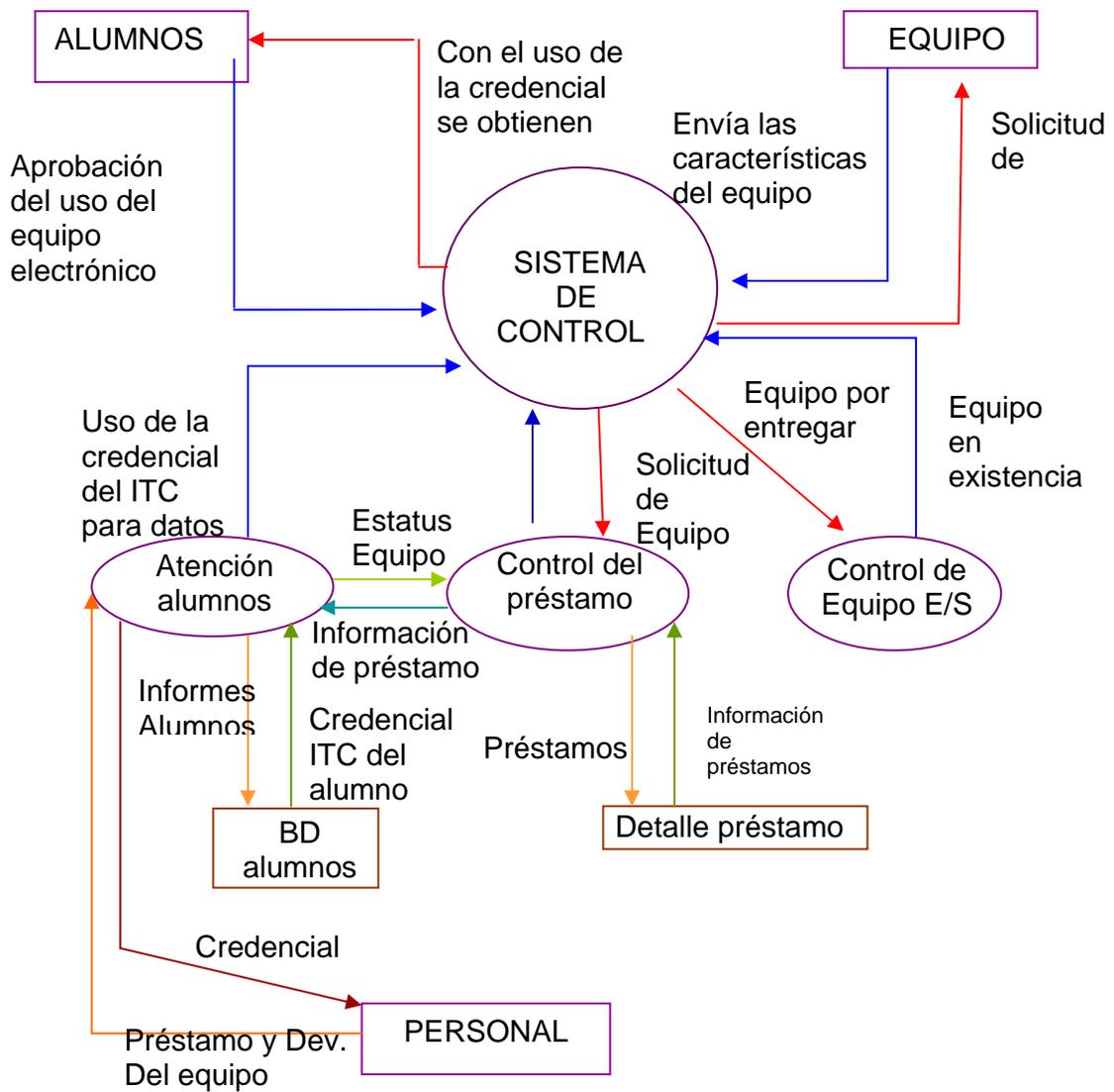


Fig. 4.2 Diagrama que muestra el Flujo de Información a 2do. Nivel.

Análisis de factibilidad operativo

Con base en el análisis que he desarrollado hasta este punto del proyecto y a través de la revisión del mismo mediante la implantación de los flujos de información antes expuestos se ha obtenido lo siguiente:

- El flujo de la información se mejora en rapidez y claridad mediante la aplicación de las funciones y procesos que se describen en los diagramas flujo de información.
- El análisis realizado otorga al sistema actual una flexibilidad para aplicar el mismo sistema a algún laboratorio más.
- Permite identificar las funciones de cada persona del laboratorio, lo cual es útil para reducir errores a través de la especialización de cada uno del personal.
- Se tendrá un control detallado del estado físico del equipo electrónico, para su mantenimiento o reposición.
- La espera de los alumnos para el préstamo del equipo electrónico se reducirá bastante.
- Se tendrá un conocimiento de los detalles del equipo electrónico mensual y semestral.
- Los objetivos del proyecto contemplan las necesidades reales del negocio.

Por este análisis se concluye que el proyecto hasta este punto *sí es operativo*, por acuerdo del Jefe del laboratorio. Además que permitió al personal del laboratorio conocer mas detalles del estado físico del equipo de electrónica.

4.2 Determinación de requerimientos.

En el laboratorio de electrónica del Instituto Tecnológico de Celaya, como ya lo explicamos su funcionamiento de préstamo de equipo lo realizan manualmente con el llenado de unos vales los cuales contienen los datos del alumno y de qué equipo es el que está solicitando. Al terminar el día el personal no le queda dato alguno de cuantas veces prestó determinado equipo, no deja registro de que si hay algún equipo dañado en ese día, qué equipo electrónico empezó a fallar, quién usó cierto equipo y en qué estado lo recibió y como lo regresa; así tendremos un reporte registrado del equipo y al llegar otro encargado cuando quiera prestar determinado equipo puede ver si hay alguna nota de que ese equipo tiene algún daño y así ya no lo presta, y se den cuenta de que hay que mandarlo a mantenimiento o hacer la solicitud de un equipo nuevo, en caso de que el mantenimiento ya no sea efectivo para el daño del equipo.

Al término del mes pueden tener un reporte de cómo está el laboratorio de electrónica en cuanto al equipo electrónico, cuál está fallando, cuál hay que reponer, etc., al igual que al término del semestre se tendrá un conocimiento más a fondo y así tendrán conocimiento de cómo se está comportando el laboratorio de electrónica.

Diccionario de datos

Cla..	Id.	Nombre	Tipo de d..	Tama..	Valor..
		nocontrol	int	4	<input type="checkbox"/>
		nomalumno	char	40	<input checked="" type="checkbox"/>
		domalumno	char	50	<input checked="" type="checkbox"/>
		idcarrera	int	4	<input checked="" type="checkbox"/>
		idsemestre	int	4	<input checked="" type="checkbox"/>

Fig. 4.3 Tabla que muestra los campos del alumno

Cla...	Id.	Nombre	Tipo de d...	Tama...	Valor...
		idcarrera	int	4	<input type="checkbox"/>
		nomcarrera	char	20	<input checked="" type="checkbox"/>

Fig. 4.4 Tabla que muestra los campos de la carrera

Cla...	Id.	Nombre	Tipo de d...	Tama...	Valor...
		idmateria	int	4	<input type="checkbox"/>
		nocontrol	int	4	<input type="checkbox"/>

Fig. 4.5 Tabla que muestra los campos de la materia

Cla...	Id.	Nombre	Tipo de d...	Tama...	Valor...	Predetermi
		idmateria	int	4	<input type="checkbox"/>	
		iddia	int	4	<input type="checkbox"/>	
		idhorario	int	4	<input type="checkbox"/>	

Fig. 4.6 Tabla que muestra los campos de Detalle Horario

Cla...	Id.	Nombre	Tipo de d...	Tama...	Valor...	Predetermi
		nocontrol	int	4	<input type="checkbox"/>	
		noserie	int	4	<input type="checkbox"/>	
		noempleado	int	4	<input type="checkbox"/>	
		fecha	datetime	8	<input type="checkbox"/>	
		observaciones	char	60	<input checked="" type="checkbox"/>	

Fig. 4.7 Tabla que muestra los campos de Detalle Préstamo

Cla...	Id.	Nombre	Tipo de d...	Tama...	Valor...	Predetermi
		iddia	int	4	<input type="checkbox"/>	
		nomdia	char	10	<input checked="" type="checkbox"/>	

Fig. 4.8 Tabla que muestra los campos del Día

Cla...	Id.	Nombre	Tipo de d...	Tama...	Valor...	Predete
		noserie	int	4	<input type="checkbox"/>	
		nomequipo	char	20	<input checked="" type="checkbox"/>	
		descripcion	char	60	<input checked="" type="checkbox"/>	
		marca	char	15	<input checked="" type="checkbox"/>	
		modelo	char	15	<input checked="" type="checkbox"/>	
		estado	char	2	<input checked="" type="checkbox"/>	
		uso	int	4	<input checked="" type="checkbox"/>	

Fig. 4.9 Tabla que muestra los campos del Equipo

Cla...	Id.	Nombre	Tipo de d...	Tama...	Valor...	Predetermi
		idhorario	int	4	<input type="checkbox"/>	
		horas	char	10	<input checked="" type="checkbox"/>	

Fig. 4.10 Tabla que muestra los campos del Horario

Cla...	Id.	Nombre	Tipo de d...	Tama...	Valor...	Predetermi
		idmateria	int	4	<input type="checkbox"/>	
		nommateria	char	20	<input checked="" type="checkbox"/>	

Fig. 4.11 Tabla que muestra los campos de la materia

Cla...	Id.	Nombre	Tipo de d...	Tama...	Valor...	Predetermi
		noempleado	int	4	<input type="checkbox"/>	
		nomempleado	char	40	<input checked="" type="checkbox"/>	
		horario	char	10	<input checked="" type="checkbox"/>	
		contraseña	char	10	<input checked="" type="checkbox"/>	

Fig. 4.12 Tabla que muestra los campos del Empleado

Cla...	Id.	Nombre	Tipo de d...	Tama...	Valor...	Predetermi
		idsemestre	int	4	<input type="checkbox"/>	
		nomsemestre	char	15	<input checked="" type="checkbox"/>	

Fig. 4.13 Tabla que muestra los campos del Semestre

Identificación de entidades

Las entidades identificadas son:

Alumno, detalle alumno, equipo, personal, día, detalle préstamo, detalle horario, materia, carrera, semestre, horario.

RELACION ENTRE ENTIDADES

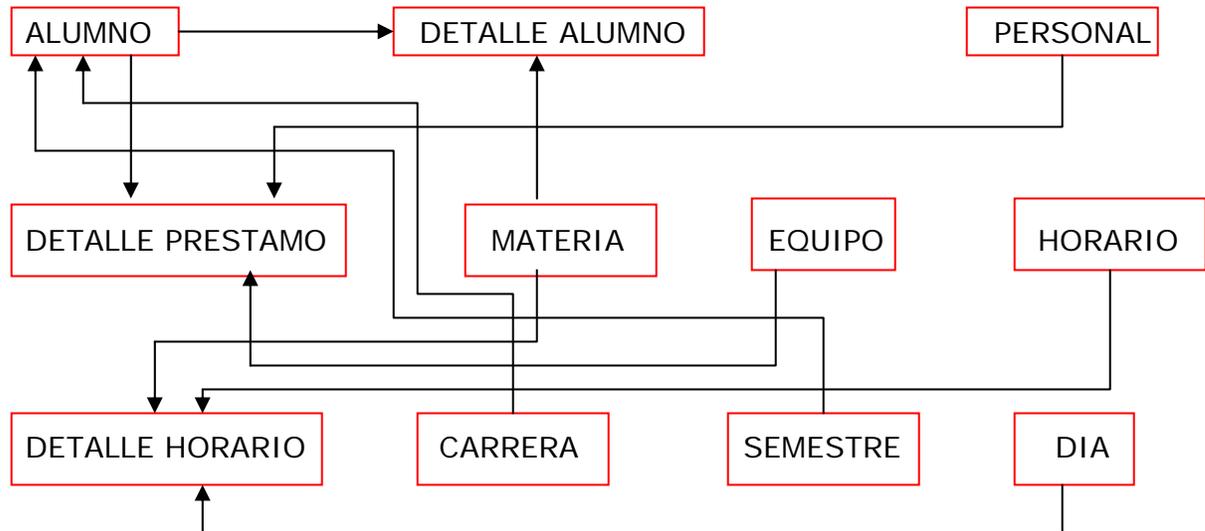


Fig. 4.14 Relación entre las entidades del sistema.

Modelo Entidad-Relación.- Relaciones de entidades de muchos a muchos, se muestra como van a ir relacionadas nuestras entidades que he identificado para el sistema del laboratorio de electrónica del Instituto Tecnológico de Celaya.

Atributos de las entidades

Entidades alumno:

Nocontrolpk nomalumno domalumno idcarrera idsemestre

Entidades equipo:

Noseriepk nomequipo descripción marca modelo estado uso

Entidades personal:

Noempleadopk nompleado horario contraseña

Entidad horario:

Idhorariopk horas

Entidad carrera:

Idcarrerapk nomcarrera

Entidad semestre:

Idsemestrepk nomsemestre

Entidad detalle prestamo:

Nocontrolpk noseriepk noempleadopk fechapk descripción

Entidad detalle alumno:

Idmateriapk nocontrolpk

Entidad materia:

Idmateriapk nommateria

Entidad detalle horario:

Idhorario idmateria iddiapk

Entidad día:

Iddiapk nomdía

De acuerdo al estudio practicado del manejo del laboratorio de electrónica del Instituto Tecnológico de Celaya, se llegaron a los atributos mencionados anteriormente, señalando cuales son llaves primarias e id.

Modelo extendido entidad-relación

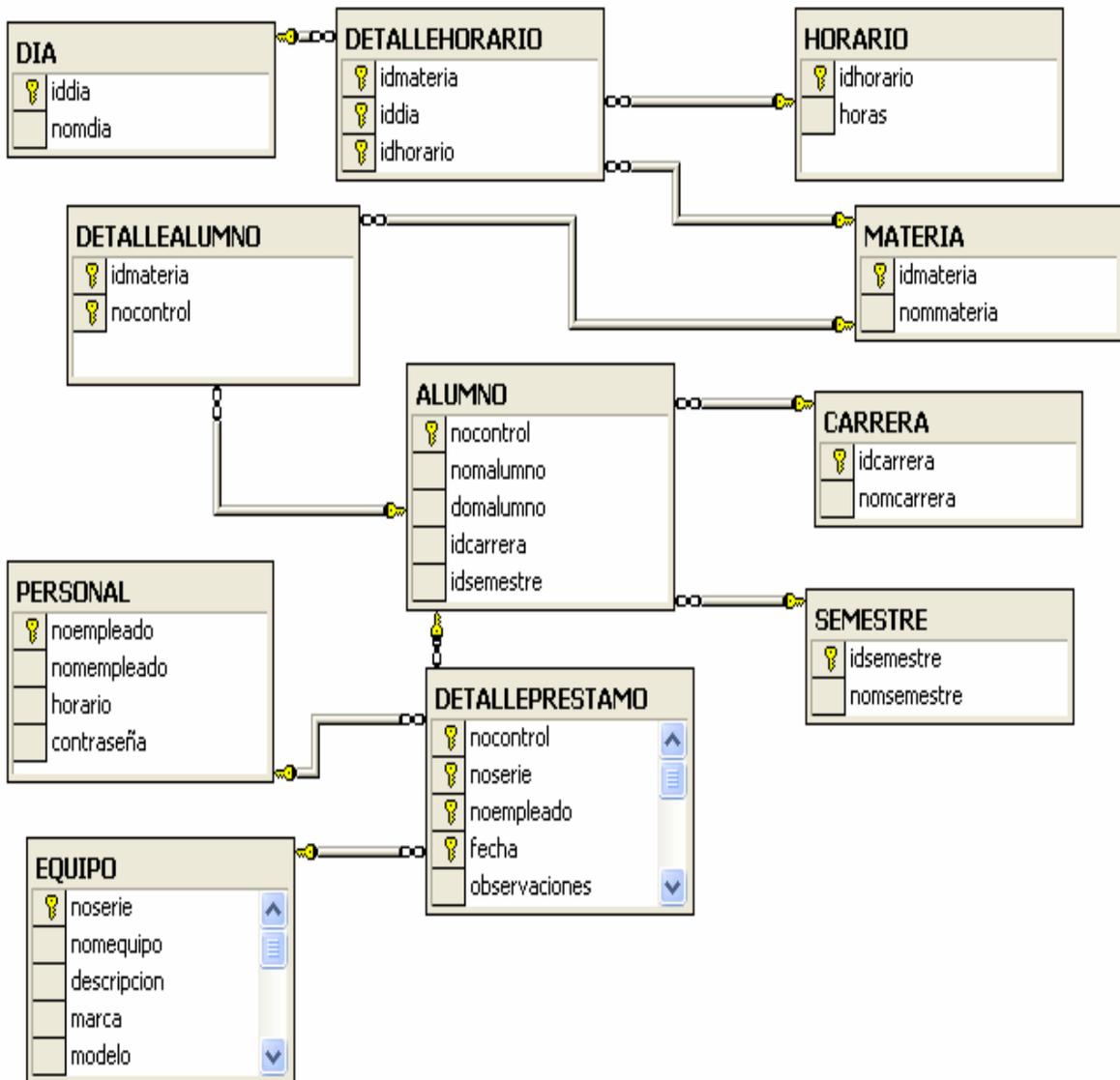


Fig. 4.15 Modelo Extendido Entidad Relación

Requerimientos de hardware del sistema

Para la instalación e implementación del sistema es necesario el hardware que cumpla con los siguientes requisitos:

Tres pantallas planas para computadora

1 CPU con un disco duro de 40 GB, memoria RAM de 256 MB, un quemador de CD a una velocidad mínimo de 32x, tarjeta de Red y puertos USB.

3 Lectores ópticos

1 Etiquetadora de barras

1 Computadora portátil

Software, mínimo Windows 2000 professional, SQL 2000 server, Visual Basic 6.0 y Crystal Reports 8.5.

Todo este equipo debe ser instalado por una persona capacitada y teniendo en cuenta que es de gran importancia mantener este equipo en buen estado al igual que el sistema puesto que ambos tanto software como hardware requieren de mantenimiento preventivo y correctivo.

El mantenimiento al software (el sistema) siempre podrá ser realizado por cualquier persona que esté capacitada en el ambiente de visual Basic 6.0 y SQL 2000 server.

El sistema queda abierto a opciones de crecer según las necesidades que vaya teniendo el Laboratorio de Electrónica del Instituto Tecnológico de Celaya.

4.3 Desarrollo del sistema

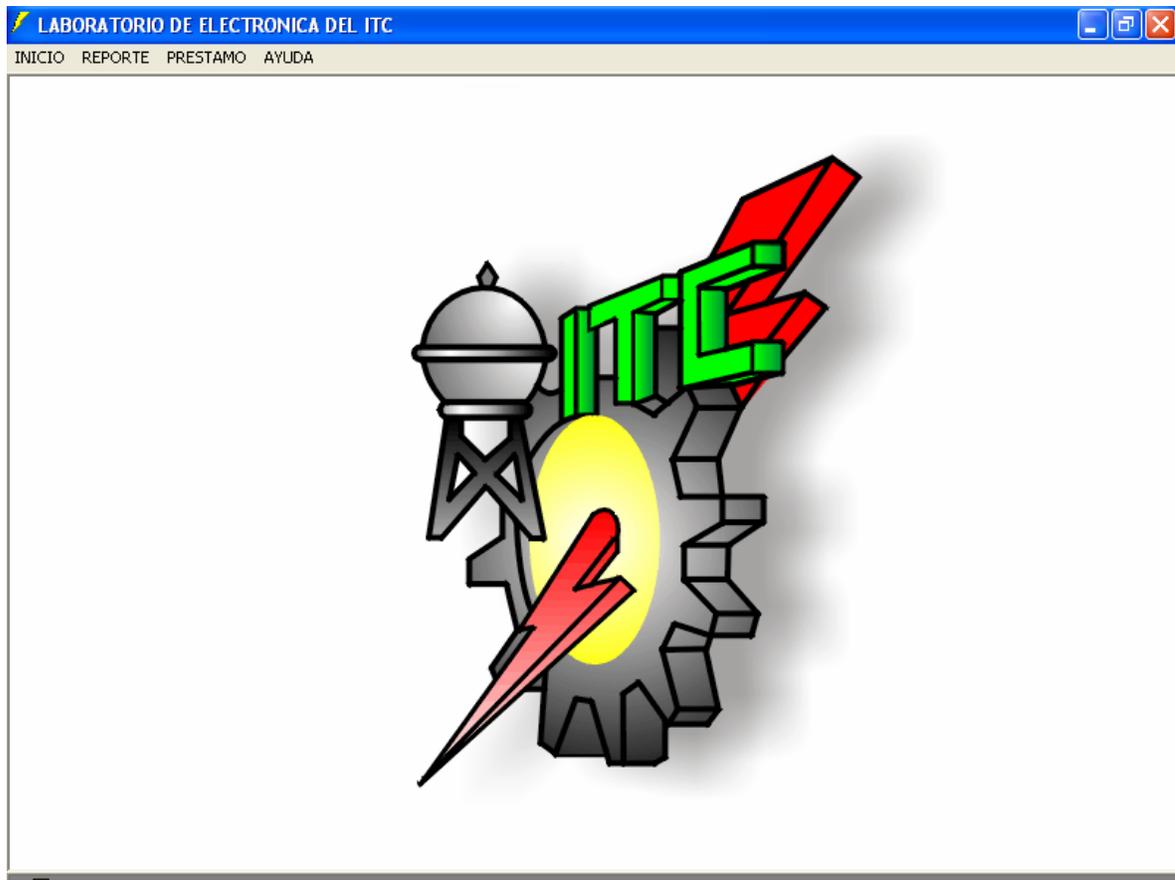


Fig. 4.16 Ventana Principal

Como se muestra en la ventana anterior los menús utilizados, a su vez cuentan con submenús como se describe cada uno a continuación:

El menú ***Inicio*** cuenta con los submenús ***Contraseña, Fin de sesión y Salir.***

El menú ***Reporte*** cuenta con los submenús ***Personal, Equipo, Horario y Alumno.***

El menú ***Préstamo*** cuenta con los submenús ***Entrega de equipo y Detalle Préstamo.***

El menú ***Ayuda*** cuenta con el submenú ***Índice.***

A continuación mostraré cada una de las ventanas que corresponden a los submenús:

En el menú inicio encontramos los siguientes submenús:

En la ventana *contraseña* pedirá al empleado su nombre y password, si estos son correctos podrá tener acceso al sistema en caso de que no fueran los datos correctos el sistema no le permitirá el acceso. Esto es una medida de seguridad.



Fig. 4.17 Ventana Contraseña

El submenú *fin de sesión* permite salir a la ventana principal, enviando un mensaje el cual dice "por hoy terminaste", en ese momento quedan deshabilitados todos los submenús excepto contraseña, salir e índice; si el empleado se va del laboratorio al dar fin de sesión nadie podrá tener acceso al sistema, solamente volviendo al menú contraseña. Esto es otra medida de seguridad para nuestra base de datos, para cuando el empleado cambia de turno.

El submenú *salir* como su nombre lo indica es para salir completamente del sistema.

En el menú reporte encontramos los siguientes submenús:

En la ventana de alumno podemos obtener los datos como el número de control, nombre del alumno, domicilio, carrera y semestre.

Para obtener el número de control es necesario que el alumno muestre su credencial del ITC para así poder tomar los datos con la ayuda de un lector óptico el cual se barre por el código de barras de la credencial y así los datos siguientes el sistema no los arroja automáticamente, sin necesidad de que el personal esté llenando datos, esto se logra dando clic en el botón de buscar.

Ahora si quisiera el empleado buscar a otro alumno solo da clic en el botón limpiar y todos los campos quedarán en blanco para la siguiente búsqueda; para poder actualizar la información de algún alumno se hace el cambio en el campo necesario y se da clic en el botón actualizar y en ese momento la información ha quedado actualizada; o si llega un alumno nuevo el personal tendrá que darlo de alta en la base de datos del sistema lo único que hace es utilizar el código de barras de la credencial del ITC del alumno y pasarle el lector óptico registrar sus datos, en la carrera y semestre solo tendrá el empleado que elegir el datos correcto sin necesidad de escribirlo y dar clic en el botón agregar.

También se tendrá un reporte de cada alumno que ha hecho uso del laboratorio de electrónica del Instituto Tecnológico de Celaya, que dará los siguientes datos número de control del alumno, nombre, fecha, observaciones, número de serie del equipo que solicitó, nombre del equipo, marca, número de veces que se ha utilizado ese equipo, número de empleado que lo prestó y el nombre; y por último el botón salir nos regresa a la ventana principal, se sale del menú reporte.

<i>No. Control</i>	<input type="text"/>	<i>Limpiar</i>
<i>Nombre del alumno</i>	<input type="text"/>	<i>Agregar</i>
<i>Dom. del Alumno</i>	<input type="text"/>	<i>Actualizar</i>
<i>Carrera</i>	<input type="text"/>	<i>Buscar</i>
<i>Semestre</i>	<input type="text"/>	<i>Salir</i>
		<i>Reporte</i>

Fig. 4.18 Ventana Alumno

En la ventana de horario tenemos el nombre de la materia, la hora y el día, esta información ya la tenemos en los combos solo hay que hacer clic en la flecha del combo de cada uno para poder elegir el dato que sea necesario, si por alguna razón se tiene la necesidad de agregar alguna materia entonces se utilizará el botón que tiene los puntos suspensivos ahí se da clic y el empleado le dará el nombre de la materia que sea necesaria para poder elegir la hora y el día se usan los combos que tienen estos datos se elige la necesaria y se da clic en el botón agregar y la materia estará agregada con su horario; para actualizar el horario de alguna materia se cambian los datos que se tienen en los campos una vez ya modificados estos datos se dará clic en el botón actualizar; el botón limpiar se utiliza para que los campos estén vacíos; el botón de reporte nos arroja el horario completo; en la ventana también podemos ver nuestro horario, y si se ha hecho alguna modificación ahí se podrá ver si esa modificación fue hecha correctamente ya que se muestra día, hora y materia y

el botón cerrar como su nombre lo indica cierra la ventana y regresa a la ventana principal.

The screenshot shows a window titled "Horario" with a blue title bar. It contains three dropdown menus labeled "Materia", "Hora", and "Dia". To the right of these fields are five buttons: "Agregar", "Actualizar", "Limpiar", "Reporte", and "Cerrar". Below the form is a table with the following data:

Dia	Hora	Materia
Lunes	12-13	CIRCUITOS II
Martes	9-10	SIST. DIGITALES I
Martes	10-11	SIST. DIGITALES I
Miercoles	9-10	SIST. LINEALES I
Lunes	12-13	MICRO II
Jueves	14-15	MICRO II
Martes	14-15	SIST. DIGITALES
Lunes	9-10	quimica

Fig. 4.19 Ventana Horario

En la ventana equipo se ingresa el número de serie del equipo por medio de un lector óptico ya que el equipo cuenta con un código de barras, se hace el barrido y en ese momento aparece el número de serie del equipo si se necesitaran los datos del equipo se da clic en el botón de buscar y en ese momento se obtienen los datos de los campos con los que cuenta la ventana como son nombre del equipo, descripción, marca y modelo; el botón limpiar, como su nombre lo indica limpia los datos que hemos consultado y deja limpios los campos; el botón agregar es utilizado para dar de alta a un nuevo equipo esto se da el barrido del lector óptico al nuevo equipo y se escriben los datos del equipo y se da clic en el botón agregar; el botón actualizar se utiliza cuando vamos a actualizar un dato de algún equipo ya en existencia solo se hace el cambio del dato y se da clic en actualizar; el botón reporte nos arroja una lista

de todo el equipo en existencia como nombre, número de serie, descripción, marca y modelo; el botón salir regresa a la ventana principal.



The image shows a software window titled "EQUIPO" with a blue header bar containing a yellow lightning bolt icon and standard window control buttons (minimize, maximize, close). The main area has a light beige background and contains five text input fields on the left, each with a label in italics: "No. Serie", "Nom. Equipo", "Descripción", "Marca", and "Modelo". To the right of these fields is a vertical column of seven buttons, also with labels in italics: "Limpiar", "Agregar", "Actualizar", "Buscar", "Borrar", "Reporte", and "Salir".

Fig. 4.20 Ventana de Equipo

En la ventana del personal se pueden ver los botones limpiar este se utiliza para limpiar los datos del personal que se habían consultado anteriormente; si se tuviera un nuevo empleado se tendría que dar de alta en la base de datos, se capturarían los datos del nuevo empleado y se da clic en el botón agregar; el botón actualizar se utiliza cuando se ve la necesidad de actualizar algún dato del empleado se hace el cambio del dato y se da clic en actualizar mientras no se haga ningún cambio el botón estará deshabilitado; el botón buscar es para obtener los datos del empleado solo se ingresa el número del empleado se da clic en buscar y los datos aparecen; el botón borrar es para dar de baja a alguien del personal, ya que por alguna razón no pueda estar trabajando en el laboratorio de electrónico del Instituto Tecnológico de Celaya; el botón reporte dará un listado de los empleados que se tienen en el sistema dados de alta; y por último el botón salir manda a la ventana principal. A esta ventana solo

podrá tener acceso el jefe del laboratorio de electrónica del Instituto Tecnológico de Celaya.



Fig. 4.21 Ventana del Personal

En el menú préstamo encontramos los siguientes submenús:

En la ventana detalle préstamo cuando se de clic en este submenú va a mandar nuevamente la ventana de contraseña esto es como una medida de seguridad ya que estará siempre la ventana principal y si por alguna razón el empleado salió un segundo y llegara alguien a querer ver esa información no se lo va a permitir el sistema ya que tiene que dar los datos correctos para poder ingresar a dicho submenú, en esta ventana se verán los datos del empleado que esté en ese momento trabajando con el sistema, también se tendrá la fecha, solo con hacer el barrido del lector óptico en el código de barras de la credencial del ITC del alumno se tendrán los datos del él solo el empleado tendrá que buscar en el combo la materia y listo, aparecerá un mensaje preguntando que si los datos son correctos y al contestar "si" pasará al siguiente recuadro, ahora hay que hacer el barrido con el equipo para obtener

los datos del equipo que en ese momento se va a prestar, si se va a realizar un nuevo préstamo se da clic en el botón nuevo préstamo y se limpiarán los datos pero solo del alumno y del equipo, los datos del empleado serán igual ya que el es el que sigue prestando equipo; también esta ventana cuenta con un listado de lo que se ha prestado como el nombre del equipo y su descripción; el botón salir regresa a la ventana principal.

DETALLE DE PRESTAMO

Fecha 07/07/2005

No. Empleado 0 *Empleado* ADMINISTRADOR

No. Control *Alumno*

Semestre *Carrera*

Materia

No. Serie *Equipo* *Descripción*

Equipo	Descripción

Nuevo Préstamo

Salir

Fig. 4.22 Ventana Detalle Préstamo

En la ventana entrega de equipo como su nombre lo dice se utiliza en el momento que el alumno entrega el equipo que se le prestó, el empleado le solicita una vez más la credencial del ITC para hacer el barrido y realiza el barrido también del equipo que el alumno está regresando y el sistema en ese momento toma los datos del alumno y le descarga el equipo que ya entregó.

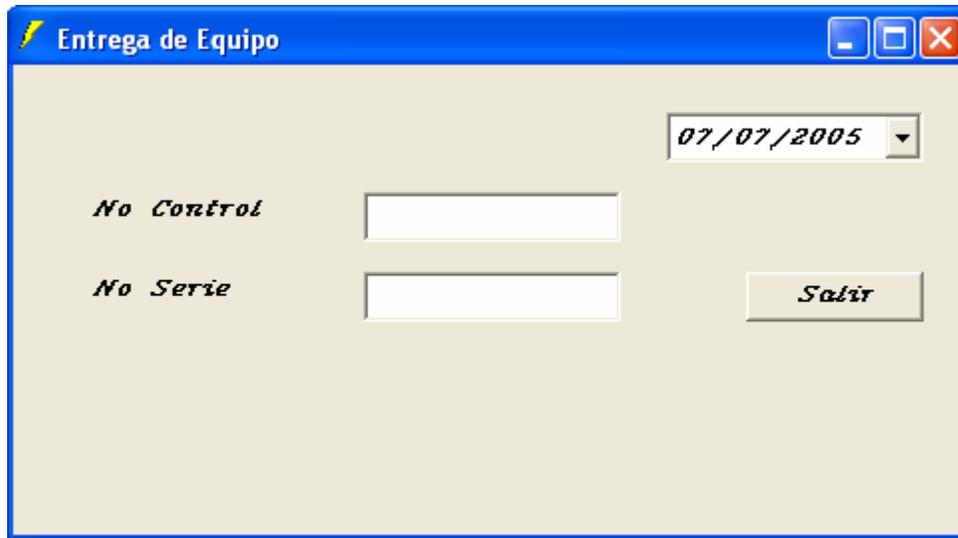


Fig. 4.23 Ventana Entrega de Equipo

Y por último el sistema cuenta con una ventana de ayuda, la cual presenta una explicación de cómo es el funcionamiento del sistema del laboratorio de electrónica del Instituto Tecnológico de Celaya, tratando de auxiliar al usuario del sistema a continuación se mostrará su ventana.



Fig. 4.24 Ventana de Ayuda

4.4 Prueba del Sistema

Una vez que tenemos el sistema terminado se le hacen las pruebas pertinentes con datos reales utilizados dentro del Laboratorio de Electrónica del Instituto Tecnológico de Celaya. Con estas pruebas me pude dar cuenta de algunas deficiencias que existían en el sistema, no tenía previsto la contraseña del administrador por lo cual ya se corrigió. Otro problema encontrado fue que no arrojaba información acerca del tiempo de uso de cada equipo, por lo tanto se hicieron los ajustes necesarios para resolver dicho problema, otro problema que se encontró fue que en una de nuestras ventanas se requería de que el empleado tecleara algunos datos lo cual se cambió a un simple pase del lector óptico ya que el objetivo del sistema es hacer más rápido y en menor tiempo

posible el préstamo del equipo del Laboratorio de Electrónica del Instituto Tecnológico de Celaya.

Una vez corregidos los problemas detectados en nuestro sistema podemos concluir que dicho sistema esta funcional y listo para su instalación.

4.5 Instalación del Sistema

La instalación del sistema en este momento quedará pendiente ya que el Instituto Tecnológico de Celaya está evaluando los requerimientos necesarios para poder implementar dicho sistema en el Laboratorio de Electrónica.

Conclusiones

Recordando tiempo atrás, cuando las computadoras no ocupaban todavía un lugar tan importante como ahora en la vida del ser humano, las actividades de éste eran realizadas por él mismo, por el hombre. No necesitaba la ayuda de ninguna máquina, entonces el trabajo se volvía pesado y requería demasiado tiempo y dinero por si tenía que contratar los servicios de alguien más para ayudarlo.

En la actualidad contamos con las computadoras que son las que nos hacen el trabajo menos pesado, y más rápido; en mi caso me puse a investigar cómo es el manejo del Laboratorio de Electrónico del Instituto Tecnológico de Celaya, cuál era realmente la problemática de ese lugar, puesto que los resultados que obtenían tanto alumnos como maestros no era el deseado, la manera de operar del Laboratorio de Electrónica del Instituto Tecnológico de Celaya es la siguiente: El maestro lleva a su grupo a dicho Laboratorio con el conocimiento de la práctica que se va a realizar, por lo tanto el alumno ya tiene una idea de qué equipo va a solicitar; y aquí viene el problema todos quieren ganar el primer lugar en la fila puesto que el encargado del laboratorio en ese momento tiene que hacer uso de los vales para obtener los datos del alumno que va a solicitar el equipo del Laboratorio, este llenado de vales el empleado lo hace a mano pidiendo el folio, fecha, nombre del equipo que va a solicitar, materia, nombre del alumno, nombre del maestro y firma; y ese es el método que hace el empleado con cada uno de los alumnos que solicita equipo del Laboratorio; por lo tanto la pérdida de tiempo es bastante y el alumno no podrá terminar su práctica en el tiempo establecido por la clase.

Otro problema que encontré fue que son bastantes vales los que se llenan lo cual se tiene el riesgo de que se pierdan, tampoco se tiene un registro o reporte del equipo que ya está descompuesto o que necesita mantenimiento, no se tiene el conocimiento exacto de la cantidad de equipo que se tiene en el laboratorio, no hay un control tampoco de quién es el empleado que presta determinado equipo por si ocurriera un detalle, todo el personal se da cuenta de lo que el empleado anterior realizó y talvez se pudieran alterar los vales, no se tiene un conocimiento detallado del estado actual del equipo

como cuántas veces se ha utilizado para así poder dar un diagnóstico de que mantenimiento se necesita para cada uno de los equipos.

En el sistema ya antes explicado llegué a la conclusión de que sí es factible puesto que arroja todos los datos que se requieren o necesita el encargado del Laboratorio de Electrónica del Instituto Tecnológico de Celaya, puesto que el sistema ofrece seguridad en los datos o sea no cualquier persona puede tener acceso a toda la información, pues se tiene un control de esta, además de que se cumple el objetivo puesto que el empleado ya no hace perder tanto tiempo a alumnos como a maestros debido al llenado de los vales, ya que ahora hará uso del lector óptico en credenciales del alumno para poder tomar sus datos personales así como del equipo que solicite y el sistema se encargará de tener actualizada la información de cada equipo y persona que haga uso del Laboratorio de Electrónica del Instituto Tecnológico de Celaya.

ANEXO PROGRAMA FUENTE

CÓDIGO DE LA VENTANA DETALLE PRESTAMO

```
Dim pres As String
```

```
Private Sub Combo1_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
If KeyCode = 13 Then
```

```
    If MsgBox("Son correctos los datos?", vbYesNo, "Aviso") = vbYes Then
```

```
        Frame3.Enabled = True
```

```
        Frame2.Enabled = False
```

```
        Text2(0).SetFocus
```

```
        Command1.Enabled = True
```

```
    End If
```

```
End If
```

```
End Sub
```

```
Private Sub Command1_Click() 'Botón de Nuevo Préstamo
```

```
    Frame2.Enabled = True
```

```
    Frame3.Enabled = False
```

```
    Text1(0) = ""
```

```
    Text1(1) = ""
```

```
    Text1(4) = ""
```

```
    Text1(5) = ""
```

```
    Text2(0) = ""
```

```
    Text2(1) = ""
```

```
    Text2(2) = ""
```

```
    Combo1.Text = ""
```

```
    Text1(0).SetFocus
```

```
    llenaListView ListView1, "Select B.nomequipo, B.Descripcion, A.noserie
```

```
    From detalleprestamo A inner join equipo B on (A.noserie=B.Noserie)
```

```
    Where A.nocontrol ="" & Text1(0) & "" and A.fecha = "" &
```

```
    DTPicker2.Value & """"
```

```

End Sub

Private Sub Command2_Click() 'Botón de Salir
    Unload Me
End Sub

Private Sub DTPicker2_KeyDown(KeyCode As Integer, Shift As Integer)
'Cambia el cursor del Data Picker
    If KeyCode = 13 Then Text1(0).SetFocus
End Sub

Private Sub Form_Load() 'Eventos al cargar la forma
    Frame2.Enabled = True
    Frame3.Enabled = False
    DTPicker2.Value = Format(Now, "dd/mm/yyyy")
    Call LlenaMateria
End Sub

Private Sub Text2_LostFocus(Index As Integer) 'Hace la consulta del equipo y sus características
    If Index = 0 Then
        If Text2(0) = "" Then Exit Sub
        Set rs = cn.Execute("Select * from equipo where noserie = '" & Text2(0) & "'")
        If Not rs.EOF Then
            Text2(1).Text = rs!Nomequipo
            idequi = rs!noserie
            Text2(2).Text = rs!descripcion
            pres = rs!estado
            uso = rs!uso
            hrs = rs!tiempo
            If pres = "Si" Then
                uso = uso + 1
                pres = "No"
            End If
            t1 = Time
        End If
    End If
End Sub

```

```

        cn.Execute "insert into Detalleprestamo
(nocontrol,noserie,noempleado,fecha,hrprestamo) values (" &
Text1(0) & "," & Text2(0) & "," & Text1(2) & "," & DTPicker2.Value &
"," & t1 & ")"

        cn.Execute "Update equipo set estado="" & pres & ", uso="" &
uso & "" where noserie = "" & Text2(0) & ""

        llenaListView ListView1, "Select B.nomequipo, B.Descripcion,
A.noserie From detalleprestamo A inner join equipo B on
(A.noserie=B.Noserie) Where A.nocontrol ="" & Text1(0) & "" and
A.fecha = "" & DTPicker2.Value & ""

        Else

            MsgBox "El Equipo se encuentra prestado", vbOKOnly, "ITC"

            Text2(0) = ""
            Text2(1) = ""
            Text2(2) = ""

            Text2(0).SetFocus

        End If

    End If

End If

End Sub

Private Sub Text2_KeyPress(Index As Integer, KeyAscii As Integer) Cambia
el cursor de las cajas de texto

If KeyAscii = 13 Then

    If Index = 2 Then

        Text2(1) = ""
        Text2(2) = ""
        Text2(0) = ""

        Text2(0).SetFocus

    Else

        Text2(Index + 1).SetFocus

    End If

End If

```

End If

End Sub

Private Sub LlenaMateria() **'Llena el combo de materia**

Dim rs As New ADODB.Recordset

Set rs = cn.Execute("select idMateria,NomMateria from Materia")

Combo1.Clear

While Not rs.EOF

Combo1.AddItem rs!nommateria

Combo1.ItemData(Combo1.NewIndex) = rs!idmateria

rs.MoveNext

Wend

Set rs = Nothing

End Sub

Private Sub Text1_KeyPress(Index As Integer, KeyAscii As Integer) **'Cambia el cursor de las cajas de texto**

If KeyAscii = 13 Then

Combo1.SetFocus

End If

End Sub

Private Sub Text1_LostFocus(Index As Integer) **'Busca los datos del alumno**

If Index = 0 Then

If Text1(0) = "" Then Exit Sub

Set rs = cn.Execute("select

A.Nomalumno,c.nomsemestre,B.nomcarrera from alumno A inner join
carrera B on (A.idcarrera=B.idcarrera) inner join Semestre C on
(A.idSemestre=C.idSemestre) where NoControl="" & Text1(0) & """)

If rs.EOF Then

MsgBox "El alumno no existe", vbCritical + vbOKOnly, "CHECAR"

Text1(0) = ""

Text1(1) = ""

```

Text1(4) = ""
Text1(5) = ""
Text1(0).SetFocus

Else

Text1(1) = rs!nomalumno
Text1(4) = rs!nomsemestre
Text1(5) = rs!nomcarrera
End If

End If
End Sub

```

CÓDIGO DE LA VENTANA EQUIPO

```

Private Sub Command1_Click() 'Busca las características equipo
    Dim rs As New ADODB.Recordset
    If Text1(0) = "" Then Exit Sub
    Set rs = cn.Execute("select Nomequipo,descripcion,marca,modelo
    from equipo where Noserie="" & Text1(0) & """)
    If rs.EOF Then
        MsgBox "El equipo no existe", vbCritical + vbOKOnly, "CHECAR"
        Call Command6_Click
    Else
        Text1(1) = rs!Nomequipo
        Text1(2) = rs!descripcion
        Text1(3) = rs!marca
        Text1(4) = rs!modelo
        Command4.Enabled = True
        Text1(0).SetFocus
    End If
End Sub

```

Private Sub Command2_Click() **'Insertar nuevo equipo**

 If Text1(0) = "" Then Exit Sub

 cn.Execute "insert into

Equipo(Noserie,nomequipo,descripcion,marca,modelo) values ('" & Text1(0)
& "','" & Text1(1) & "','" & Text1(2) & "','" & Text1(3) & "','" & Text1(4) & "'"")

 Call Command6_Click

 Text1(0).SetFocus

End Sub

Private Sub Command3_Click() **'Borrar equipo**

 If Text1(0) = "" Then Exit Sub

 cn.Execute "Delete from equipo where noserie='" & Text1(0) & "'"

 MsgBox "Registro Borrado", vbOKOnly, "BAJA"

 Call Command6_Click

 Text1(0).SetFocus

End Sub

Private Sub Command4_Click() **'Actualizar los datos del equipo**

 cn.Execute "update equipo set nomequipo='" & Text1(1) &
 "',descripcion='" & Text1(2) & "',marca='" & Text1(3) & "',modelo='" &
Text1(4) & "'" where noserie= '" & Text1(0) & "'"

 MsgBox "Actualizado", vbOKOnly, "ITC"

 Call Command6_Click

 Text1(0).SetFocus

End Sub

Private Sub Command5_Click() **'Salir de la ventana**

 Unload Me

End Sub

Private Sub Command6_Click() **'Limpiar las cajas de texto**

 Text1(0) = ""

 Text1(1) = ""

 Text1(2) = ""

 Text1(3) = ""

```

Text1(4) = ""
Text1(0).SetFocus
End Sub

Private Sub Command7_Click() 'Realiza el reporte del equipo
    CrystalReport1.ReportFileName = "D:\laboratori\Requi.rpt"
    CrystalReport1.WindowTitle = "Equipo"
    CrystalReport1.ReportTitle = "Equipo"
    CrystalReport1.Action = 1
    CrystalReport1.WindowState = crptMaximized
End Sub

Private Sub Text1_KeyPress(Index As Integer, KeyAscii As Integer) 'Cambia el cursor de posición a otra caja de texto
    If KeyAscii = 13 Then
        If Index = 4 Then
            Text1(0).SetFocus
        Else
            Text1(Index + 1).SetFocus
        End If
    End If
End Sub

```

CÓDIGO DE LA VENTANA DE ENTREGA DE EQUIPO

```

Private Sub Command1_Click() 'Busca el equipo prestado
    Dim rs As New ADODB.Recordset
    Dim rs2 As New ADODB.Recordset
    If Text1(0) = "" Then Exit Sub
    Set rs = cn.Execute("select * from Detalleprestamo where NoControl="" & Text1(0) & ""AND NoSerie="" & Text1(1) & """)
    Set rs2 = cn.Execute("select * from equipo where NoSerie="" & Text1(1) & """)
    hrs = rs2!tiempo

```

```

t1 = rs!hrprestamo
t2 = Time
t3 = t2 - t1
t4 = Format(t3, "hh:mm:ss")
totalh = hrs + t4
If rs.EOF Then
    MsgBox "El prestamo no existe", vbCritical + vbOKOnly, "CHECAR"
Else
    llenaListView ListView1, "Select B.nomequipo, B.Descripcion,
A.noserie, B.estado From detalleprestamo A inner join equipo B on
(A.noserie=B.Noserie) Where A.nocontrol ='" & Text1(0) & "' and
A.fecha = '" & DTPicker2.Value & "'"
End If
    Command2.Enabled = True
End Sub

```

```

Private Sub Command2_Click()Realiza la entrega del equipo prestado
    pres = "Si"
    cn.Execute "update equipo set estado='" & pres & "',tiempo='" &
totalh & "' where noserie='" & Text1(1) & "'"
    Command2.Enabled = False
    Text1(0) = ""
    Text1(1) = ""
    Text1(2) = ""
    Text1(0).SetFocus
    llenaListView ListView1, "Select B.nomequipo, B.Descripcion, A.noserie
From detalleprestamo A inner join equipo B on (A.noserie=B.Noserie)
Where A.nocontrol ='" & Text1(0) & "' and A.fecha = '" &
DTPicker2.Value & "'"
    MsgBox "Equipo Entregado", vbInformation, ITC
End Sub

```

```
Private Sub Command3_Click() 'Salir
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub Form_Load() 'Da la fecha actual del sistema al cargar la  
forma
```

```
    DTPicker2.Value = Format(Now, "dd/mm/yyyy")
```

```
End Sub
```

```
Private Sub Text1_KeyPress(Index As Integer, KeyAscii As Integer) 'Cambia  
el cursor de caja de texto actual
```

```
If KeyAscii = 13 Then
```

```
    If Index = 1 Then
```

```
        Call Command1_Click
```

```
        Call Command2_Click
```

```
        Text1(0).SetFocus
```

```
    Else
```

```
        Text1(1).SetFocus
```

```
    End If
```

```
End If
```

```
End Sub
```

BIBLIOGRAFÍA

- 1) BIRNIOS, Baltazar, y otros, **Microsoft Visual Basic 6.0 Manual de Referencia**, 2ª ed., Editorial MP, México, 2000, p.p617.
- 2) CEBALLOS, Francisco Javier, **Visual Basic Curso de Programación**, Editorial AlfaOmega, S.A. de C.V., México, 1997, p.p 478 .
- 3) KENDALL, & Kendall, **Análisis y Diseño de Sistemas**, McGraw-Hill, Colombia, 1190, p.p 998.
- 4) SENN, James, **Análisis y Diseño de Sistemas de Información**, Editorial McGraw-Hill/Interamericana de México S.A. de C.V., Colombia 1990, p.p.643.

Otras fuentes

www.pesarrolloweb.com/manuales/1/

www.programatium.com/vb/libros/index.htm

www.maxitrucos.com/manuales/mprogramacion

www.visualbasic.htm

www.desarrolloweb.com/manuales/9/

www.unav.es/cti/manuales/introsql/indice.htm

www.personal.lobocom.es/claudio/sq1001.htm