



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ARAGÓN**

**“SISTEMA DE CAPTURA Y CONTROL DE
DATOS PARA ESCUELA PRIMARIA CON
INTERFASE GRÁFICA”**

T E S I S

QUE PARA OBTENER EL TÍTULO DE :
INGENIERO EN COMPUTACIÓN
P R E S E N T A :
RAÚL MEDINA GONZÁLEZ

ASESOR:

M. en C. DAVID MOISÉS TERÁN PÉREZ

MÉXICO

2005

m. 341599



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Dedicatoria

Dedico este trabajo a Dios, ya que el hizo posible llegar a este momento.

A mis hijas, Darita, por ser la más hermosa bebé y Andy, por ser una niña tan ejemplar; por ser ambas un aliciente y hacer posible este proyecto, esto es para ustedes y por ustedes, las amo.

A mi esposa, la Enf. Pediatra Gabriela Romero Hurtado, gracias por todo el apoyo, por las veces que me comprendiste y las veces que no lo hiciste, por las noches que dejaste a un lado tu descanso para ayudarme a recopilar información. En buena medida esto no hubiera sido posible sin ti, recuérdalo.

A mis padres, Raúl Medina Bravo y Yolanda González Rocha, por que ahora comprendo que todo lo que hicieron, fue para que yo pudiera llegar a esta instancia, por toda su confianza y por hacer de mi una persona honesta y responsable.

A mis hermanos, Alejandra y Alán, por ayudarme y estar conmigo en las buenas y en las malas, con nada pago todo lo que han hecho por mi.

A mis sobrinos: Pris, Miguelito y Maney, por que también son parte de esto.

A mi abuela "Carmelita", por que a pesar de todo este tiempo sin ti, estoy seguro que desde donde estas, te sentirás orgullosa de mi.

A mi abuela "Magda", por que Dios ha permitido que hasta ahora estés con nosotros y seas parte de esto.

A mis suegros y cuñados, por todo el apoyo que también me brindaron en este proyecto.

A ti Toño (q.e.p.d.), por que a pesar de solo haber convivido un semestre contigo, llegaste a ser un gran amigo y compañero.

A toda mi familia, amigos y aquellos de los que ahora no me acuerdo, pero que se acuerdan de mi.

Agradecimientos

Quiero agradecer muy en especial a la Lic. Profra. Rosalva Malagón Salazar, por darme la confianza, respaldo, apoyo, etcétera, para que yo pudiera utilizar el nombre, logotipo y la matrícula escolar de la escuela primaria "Francisco Díaz de León", ya que sin esto, no hubiera sido posible materializar este trabajo. Muchas gracias "Miss", con nada podré pagarle esto, además de darme la oportunidad de laborar en esta "escuelita" como cariñosamente usted la llama. Dios la bendiga, para que continúe haciendo esta encomiable labor de dirigir y educar tanto a alumnos como a profesores. Es usted un ejemplo a seguir.

Al Profr. David Carlos Moscoso Orozco, por ser una persona tan responsable, humana y honesta, además de un gran amigo.

A mi asesor, el M. en C. David Moisés Terán Pérez, por conducirme de la manera en que lo hizo en la realización de este trabajo. Muchas gracias por todo Maestro.

Al Ing. Jaime Fernando Carrillo Ramírez, gracias ca, por el enorme apoyo al asesorarme y ayudarme con los "atorones" que tuve.

Al Ing. Oscar Estrada García, por todo el apoyo y asesoría que me brindo durante la realización de los trámites de mi titulación, gracias.

Al Dr. Salvador Díaz Rodríguez, por enseñarme a trabajar con responsabilidad y honestidad, y por darme mi primer oportunidad en el ámbito profesional, por la paciencia que me tuvo cuando cometía errores y trataba de corregirlos, por todos los consejos que me dio y que siempre he tomado, gracias.

A mi escuela, la UNAM ENEP Aragón, por brindarme la oportunidad de ser parte de ella, por haberme formado como profesionista en sus aulas y llegar a amarla, por que mi corazón azul es y mi piel dorada, siempre te querré y por que cada que escucho un "Goya", se me eriza la piel.

A todos y cada uno de mis profesores, por que de ellos aprendí y me quede con algo, que ha sido de gran ayuda en mi vida profesional, muchas gracias por darme parte de su tiempo y de su vida.

A mis cuates: Roberto "Caco", Ricardo "Pizzas", Eduardo "Abuela", Ernesto "Neto", Ramón "Vaca", Salvador "Gallo", Marco "Pato", Julio "Shagui" y los que me faltaron, por compartir conmigo angustias, desveladas, mal pasadas, proyectos, exámenes, cotorreos, fiestas y hasta borracheras, pero sobre todo por haber sido mis compañeros durante toda la carrera, gracias.

Índice

Justificación

Plan Propuesto

Objetivo General

Objetivos Particulares

Introducción

CAPITULO I. Generalidades de Sistemas de Información.....	1
1.1.-Introducción.....	1
1.2.-Análisis de sistemas.....	1
1.2.1.-Requerimientos de sistemas.....	3
1.3.-Diseño de sistemas de información.....	4
1.4.-Desarrollo de sistemas de información.....	5
1.5.-Función de los sistemas de información.....	7
1.6 Futuro de los sistemas de información.....	9
CAPITULO II. Generalidades de Diseño de Software de Aplicación.....	12
2.1.-Introducción al diseño de software de aplicación.....	12
2.2.-Software orientado a objetos.....	13
2.3.-Proceso de desarrollo de software.....	14
2.3.1.-Análisis de Requerimientos.....	14
2.3.2.-Principios del Análisis.....	17
2.3.3.-El dominio de la Información.....	17
2.3.4.-Partición.....	18
2.3.5.-Construcción de Prototipos de Software.....	19

2.3.6.-Un escenario para la construcción de prototipos.....	19
2.3.7.-Especificación.....	20
2.3.8.-Metodologías de Análisis de Requerimientos.....	23
2.3.9.-Métodos de Análisis Orientados al Flujo de Datos.....	24
2.3.10.-Diagramas de Flujos de Datos.....	25
CAPÍTULO III. Fundamentos y Herramientas para la Elaboración de Bases de Datos e Interfases Gráficas.....	26
3.1.-Introducción a las interfases gráficas (GUIs).....	26
3.2.-Características de una interfase.....	27
3.3.-Interfase de usuario. (UI).....	27
3.3.1.-Cómo usan una GUI los usuarios.....	28
3.3.2.-Evolución de las interfases de usuarios.....	28
3.4.-Diseño de una interfase gráfica.....	29
3.5.-Factores de diseño.....	30
3.5.1.-Modelos.....	31
3.5.1.1.-Partes de que consta un modelo.....	33
3.5.2.-Lenguaje de comando.....	33
3.5.3.-El diseño del menú.....	35
3.5.3.1.-Interfases de menús.....	35
3.5.4.-Realimentación.....	38
3.5.5.-Formatos de salida.....	39
3.6.-Estructura de los fundamentos del diseño.....	40

3.7.-Principios para el diseño de interfases de usuario.....	41
3.8.-Programación orientada a objetos.....	44
3.8.1.-El objeto.....	44
3.8.1.1.-Estructura de un objeto.....	44
3.8.1.2.-Organización de los objetos.....	45
3.8.1.3.-Relaciones entre objetos.....	45
3.8.1.4.-Propiedades de los objetos.....	46
3.9.-Características de la POO.....	46
3.10.-Aplicaciones de software.....	47
3.10.1.-Historia de los sistemas operativos.....	47
3.10.2.-Software para elaborar interfases gráficas.....	49
3.10.2.1.-X Windows.....	49
3.10.2.2.-Xforms.....	50
3.10.2.3.-Lenguaje C++ Builder.....	50
3.10.3.-Lenguajes de programación utilizados en la elaboración de interfases gráficas.....	51
3.10.3.1.-Lenguaje C++.....	51
3.10.3.2.-Conociendo a Visual Basic.....	52
3.10.3.3.-Como escribir una aplicación en Visual Basic.....	53
3.10.3.4.-Cómo se crea un proyecto.....	54
3.10.3.5.-Bibliotecas de vínculos dinámicos.....	56
3.10.3.6.-Términos importantes utilizados.....	57
3.10.4.-Lenguaje Fortran.....	57
3.11.-Introducción a las Bases de Datos.....	57

3.12.-Sistema Manejador de Base de Datos. (DBMS).....	58
3.13.-Administrador de Base de Datos. (DBA).....	58
3.14.- Objetivos de los sistemas de Bases de Datos.....	59
3.15.- Base de Datos Orientadas a Objetos.....	60
3.16.-Lenguaje de manipulación de datos.....	61
3.17.- Lenguajes de Programación para Bases de Datos. (OO).....	61
3.18.-Herramientas para elaborar Bases de Datos.....	62
CAPÍTULO IV. Aplicación de un Sistema de Captura de Datos para Escuela Primaria con Interfase Gráfica.....	65
4.1.-Metodología de diseño para el proyecto Sistema de Captura de Datos para Escuela Primaria con Interfase Gráfica.....	65
4.1.1.-Planteamiento del problema.....	65
4.1.2.-Pregunta de Investigación.....	66
4.1.3.-Objetivo General.....	67
4.1.4.-Objetivos particulares.....	67
4.1.5.-Justificación.....	67
4.1.6.-Contextualización.....	68
4.2.-Entendiendo al Sistema de Captura para Primaria.....	68
4.2.1.-Sistema de Captura.....	68
4.3.-Requerimientos para el sistema.....	69
4.3.1.-Requerimientos de hardware y software.....	69
4.3.2.-Información fuente.....	70
4.4.-Estructura del sistema.....	71

4.4.1.-Estructura principal.....	71
4.4.2.-Estructura de la creación de archivos.....	71
4.4.3.-Diseño de la interfase.....	72
4.4.3.1.-Ventana de presentación.....	72
4.4.3.2.-Ventana del Menú Principal.....	74
4.4.3.2.1.-El Menú Captura.....	75
4.4.3.2.2.-El Menú Reportes.....	95
4.4.3.2.3.-El Menú Utilerías.....	96
4.4.3.2.4.-El Menú Salir.....	98
4.4.3.2.5.-El Menú Ayuda.....	99
4.4.4.-La conexión Interfase / Base de Datos.....	99

Conclusiones

Bibliografía

Anexo A

Anexo B

INTRODUCCIÓN

Cuando hablamos de una mejor sociedad, de una óptima productividad o de una tecnología, no debe dejarse a un lado el tema de desarrollo de aplicaciones, el avance que se da de esto, así como del hardware para trabajarlo y el software para desarrollarlo, da como resultado el tener mejores aplicaciones y mejores herramientas para su elaboración.

Uno de los ámbitos más necesitados de aplicaciones más eficientes, y también uno de los más descuidados, es el de la educación, sobre todo en el sector oficial.

Existe un gran retraso en su elaboración de sistemas de captura y control escolar aunado a esto la falta de apoyo e interés por parte de Autoridades Educativas, por lo menos en el Estado de México, por averiguar cuales son las necesidades reales de las Instituciones Educativas para que estas puedan cumplir con los requisitos que las mismas autoridades establecen para obtener información de la situación educativa del país, sobre todo a nivel primaria.

El propósito de este trabajo es proporcionar una alternativa para las Instituciones de Educación Básica en el Estado de México para que logren un mejor trabajo para su control escolar, que cada año presenta serios problemas y preocupación en Directivos por estar concientes de lo que significa trabajar sin apoyo y con un sistema que presenta varias carencias. Además de recibir evasivas por parte de las Autoridades Educativas al momento de solicitar apoyo.

Se espera que así sea por lo menos en la Escuela Primaria Francisco Díaz de León, y por que no en toda la zona escolar número 6 de Nezahualcóyotl.

Justificación.

Para que una Institución Educativa funcione de manera eficiente, es necesario que todos sus elementos trabajen en conjunto para obtener los resultados que se planean dentro de la misma Institución y que son avalados por las autoridades Educativas. Dentro de estos elementos se encuentra también el del control escolar, el cual debe ofrecer un respaldo y confianza enormes. Para esto debe contarse con herramientas capaces de solventar todos los requerimientos establecidos. Es por eso que los sistemas de control y captura de datos escolares deben dejar de ser una carga más en las actividades y procesos del control escolar. Es por ello que se echa mano de la tecnología, más de software que de hardware en este caso, para elaborar una herramienta que sea capaz de llenar las expectativas de las instituciones educativas, tanto particulares como oficiales y se terminen ya los problemas que desde hace algunos años son causa de conflictos de diversa índole por no cumplir con la información para los diversos trámites en beneficio de lo que hoy es la causa de la planeación del presente proyecto: LA MATRICULA ESTUDIANTIL.

Plan Propuesto.

Cada capítulo se enfoca al diseño de software como son: Sistemas de Información, Bases de Datos, y lo que es más importante en el desarrollo de este trabajo: el diseño y elaboración de una Interfase Gráfica.

En el capítulo I, se hace referencia a algunos aspectos de los Sistemas de Información. De manera general se tocan puntos como: análisis, requerimientos, desarrollo, función, etcétera.

En el capítulo II, se comienza con el tema de diseño de software, también de manera general, dándose una introducción de esto, tocando un tema importante como es el de software orientado a objetos. Después se entra ya de lleno a los aspectos de diseño: proceso, análisis de requerimientos, construcción de prototipos y el escenario para construirlos. Se presentan también una serie de propuestas para el diseño del software.

El capítulo III habla de fundamentos para el diseño de Bases de Datos e Interfases Gráficas, desde los sistemas operativos existentes y el software que puede ser utilizado para su elaboración, hasta los lenguajes utilizados en el desarrollo de la Interfase Gráfica.

Por último, el capítulo IV, se enfoca ya de lleno en lo que es el proyecto: la metodología aplicada para llevar a cabo su diseño, la presentación de las herramientas de software utilizadas, así como la explicación del diseño de las Bases de Datos y de la Interfase Gráfica y sus códigos respectivos para que pueda funcionar de la manera que se planteo. Para cerrar, dentro de este capítulo, se tienen las conclusiones del proyecto.

Objetivo General.

Aplicar una metodología de diseño para ser aplicada a el problema que se esta planteando utilizando el software disponible y desarrollar una herramienta que sea capaz de brindar el apoyo y respaldo al departamento de captura y control escolar con Interfase Gráfica de la Escuela Primaria Particular Incorporada "Francisco Díaz de León", y que este pueda dar resultados de manera inmediata.

Objetivos particulares.

a).-Proporcionar una herramienta de trabajo eficaz y segura para la captura y control de datos escolares lo que propiciara un ahorro en materia de recursos y tiempo para el trabajo en otros ámbitos de la Institución Educativa.

b).-En base a los resultados obtenidos, comenzar de manera inmediata el trabajo de desarrollo de otros programas de control y estadística escolar.

c).-Proponer en otra Institución Educativa el uso de este sistema y observar el desempeño de este fuera del entorno donde fue creado.

Generalidades de Sistemas de Información

1.1.-Introducción.

En el vertiginoso afán de construir sistemas de información, se ha olvidado, casi por completo, la necesidad de un plan para comprenderlos, las quejas ya se han hecho sentir, habiéndose destacado la urgencia de incorporar sustancia al nombre "sistema"; para hacer de su popularidad algo más que una moda pasajera.

La conceptualización, como segundo paso hacia la sistematización del conocimiento tropieza con la dificultad de tener que inventar sistemas para describir sistemas (metalenguajes para describir lenguajes). Se corre por ello el riesgo de confundir el sistema a describir con los objetos a denotar.

Para llegar a la formalización, basta elegir o crear un lenguaje convencional, con vocabulario fijo y sintaxis inviolable para describir los sistemas y sus componentes. (Frischknecht, 1990).

En la actualidad, para muchas organizaciones, los sistemas de información basados en computadoras, son el corazón de las actividades cotidianas y objeto de gran consideración en la toma de decisiones, las empresas consideran con mucho cuidado las capacidades de sus sistemas de información cuando deciden ingresar o no en nuevos mercados o cuando planean la respuesta que darán a la competencia. (Seen 1990 - 2000)

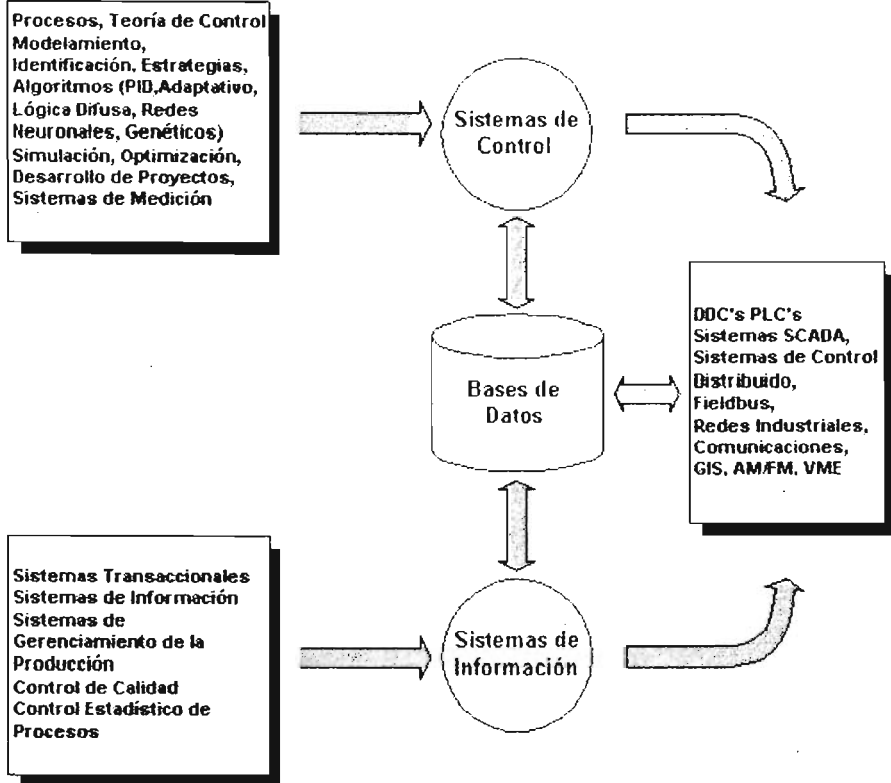
1.2.-Análisis de sistemas.

El análisis de sistemas utiliza métodos cuantitativos, cuando son aplicables, además de factores cualitativos, como el juicio, la creatividad, la heurística, el sentido común y la experiencia.

Cuándo y dónde se debe empezar un análisis de sistemas es hasta cierto punto arbitrario. Se pueden requerir un gran número de sesiones de discusión y varios estudios y experiencias empíricas antes de efectuar el análisis de sistemas.

Dentro del contexto del enfoque a la resolución de problemas, hay seis etapas básicas para la aplicación del análisis de sistemas:

1. Definición y formulación del problema;
2. Desarrollo de soluciones alternativas;
3. Construcción de modelos que formalicen las alternativas;
4. Determinación del costo / eficacia de las alternativas;
5. Presentación de las recomendaciones;
6. Implantación de la alternativa seleccionada.



1.1 Conceptualización en esquema de un sistema de información a ser implementado

Uno de los objetivos principales de la realización de un análisis de sistemas consiste en maximizar la eficacia de la solución con un costo mínimo. Para lograr esta razón óptima entre costo y eficacia es necesario obtener alternativas de cursos de acción y efectuar comparaciones. Cualquier curso de acción requiere de una asignación de recursos y produce una salida con un nivel determinado de eficacia. Una organización puede optar por uno de los criterios excluyendo el otro, por ejemplo:

1. Criterio de la eficacia: para obtener un nivel específico de eficacia en el logro de algún objetivo el analista ensaya las alternativas que alcanzarán ese nivel.
2. Criterio del costo: para un nivel específico de recursos el analista intenta determinar las alternativas que producirán el mayor nivel posible de eficacia.

De manera ideal, debe haber un balance óptimo entre los dos criterios. (Bursh, 1996).

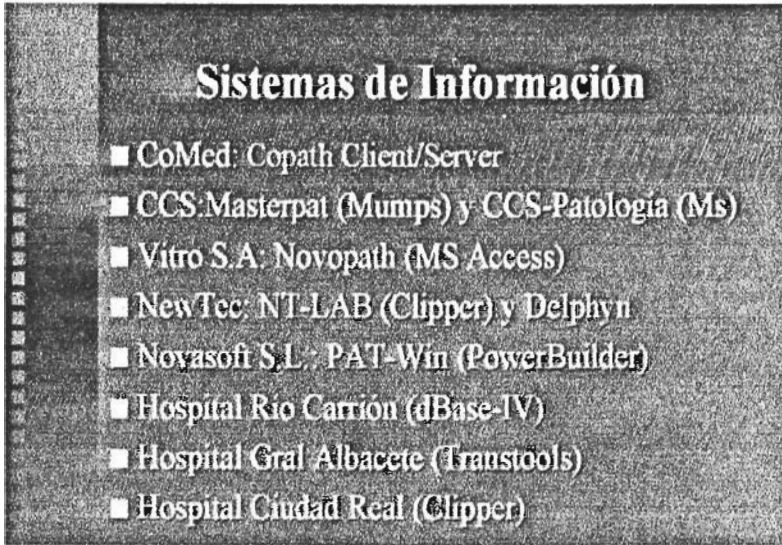
1.2.1.-Requerimientos de sistemas.

Los requerimientos del sistema son los requerimientos operacionales inherentes al sistema de información mismo y surgen de o están influenciados por otras fuerzas del diseño o requerimientos, especialmente la producción de información de calidad. Estos requerimientos son: 1) confiabilidad, 2) disponibilidad, 3) flexibilidad, 4) programa de instalación, 5) expectativa de vida y potencial de crecimiento, 6) capacidad para recibir mantenimiento.

1. **Confiabilidad:** ésta se refiere al grado de seguridad con que un recurso realiza su función, produciendo los mismos resultados en procesos sucesivos.
2. **Disponibilidad:** significa que el sistema es accesible a los usuarios. Un sistema puede ser confiable, pero no estar disponible cuando por ejemplo, está siendo probado o cuando se le está agregando un componente. Por otra parte, un sistema puede estar disponible pero no ser confiable.
3. **Flexibilidad:** este requerimiento se refiere a la habilidad del sistema para cambiar o adaptarse para satisfacer los requerimientos cambiantes de los usuarios.
4. **Programa de instalación:** el programa de instalación comprende el espacio de tiempo existente entre el momento en que una organización reconoce una necesidad y el momento en que implementa la solución. Supuestamente, aunque no siempre, entre más tiempo se requiere para diseñar un sistema, mejor será el diseño. Normalmente, no se puede dar este lujo, tiene que diseñarse el mejor sistema que se pueda, dentro del tiempo asignado.
5. **Expectativa de vida y potencial de crecimiento:** algunos sistemas no cuentan con una expectativa de vida debido a que ya son obsoletos en el momento en que se implementan. O bien, un sistema puede instalarse y trabajar muy bien durante cierto tiempo, pero debido a que es un sistema que tiene una sola salida sin la capacidad de crecer queda "patas para arriba" cuando aumentan las necesidades de los usuarios. Por lo tanto, los sistemas deben diseñarse para satisfacer requerimientos durante un tiempo razonable y ser también capaces de crecer si las necesidades cambian de manera significativa.
6. **Capacidad para recibir mantenimiento:** una vez que un sistema se implementa, debe recibir mantenimiento, debido a que se deben corregir fallas, a que se deben satisfacer solicitudes especiales y a que deben efectuarse mejoras generales a los sistemas. La pregunta no es si se tendrá o no que dar mantenimiento al sistema. Se le tendrá que hacer.

La meta por lo tanto, deberá ser diseñar sistemas que sean capaces de recibir mantenimiento mediante el empleo de nombres de Datos estándar y lenguajes de programación estructurada y modular, configuraciones estándar y procedimientos estándar de documentación. (Burch, 1992).

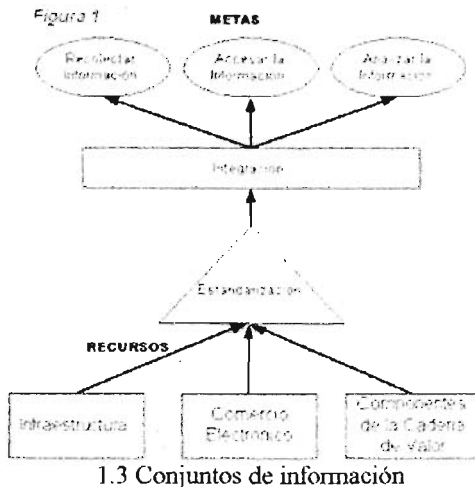
El siguiente cuadro muestra un conjunto de empresas con los sistemas de información que cumplen con los puntos descritos anteriormente.



1.3.-Diseño de sistemas de información.

Por un sistema de información se entiende: un sistema de conjuntos de información para la decisión y el señalamiento en un sistema más amplio (del cual es un subsistema) que contiene subsistemas para: (Frischknecht, 1990)

1. Almacenar
2. Procesar
3. Distribuir
4. Recolectar

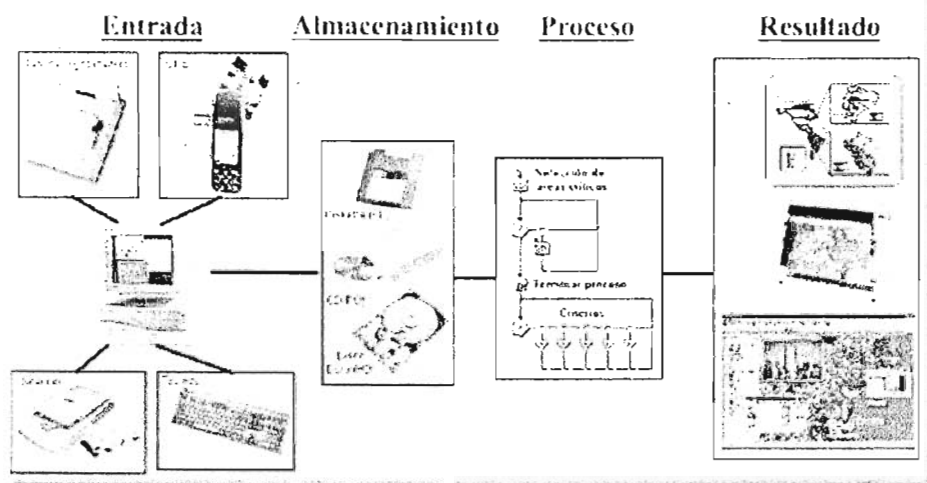


Formalización del diseño de sistemas de información:

1. Se requiere no sólo de una teoría formal, que suministre procedimientos de análisis y diseño, sino también un análisis y exposición sistemáticos del amplio volumen de evidencia empírica existente.
2. La técnica de sistemas se utiliza así mismo en las fases de diseño y puesta en práctica. Por lo tanto, normalmente se ejecutan tareas en varios subsistemas en paralelo. Esto es posible mediante la partición apropiada en subsistemas y la definición formal de interfases de subsistemas.
3. Además, se destaca una fase por vez para cada equipo de subsistemas. Esto es consecuencia de la formalización del trabajo de sistemas.
4. Así mismo, los diferentes problemas de los componentes por separado se aíslan y definen de modo que puedan resolverlos en forma individual distintos grupos, principalmente fuera del equipo central de sistemas. (Frischknecht, 1990).

1.4.-Desarrollo de sistemas de información.

El desarrollo de un sistema de información se puede denominar como la solución a un problema: la necesidad de saber. Sin embargo, para describir mejor las actividades que se llevan a cabo durante el desarrollo de un sistema es posible identificar tres etapas: 1) análisis de sistemas; 2) diseño de sistemas y; 3) implantación de sistemas.



1.4 Etapas del desarrollo de sistemas

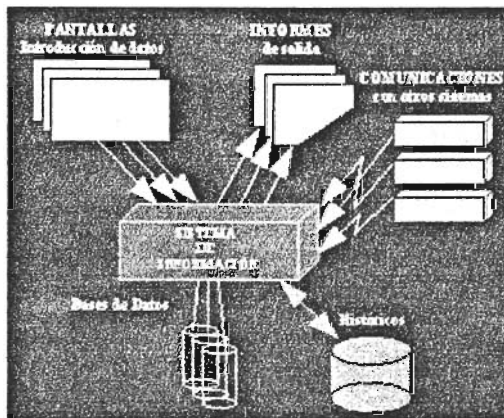
Las actividades que se realizan en estas tres etapas son muy semejantes a las seis etapas que se describieron en los requerimientos de sistemas. Por ejemplo, el análisis de sistemas es análogo a la etapa uno; la definición y formulación del problema. Para ser más específicos, abarca la definición y descripción de las metas, objetivos y requerimientos del sistema. La etapa de diseño de sistemas incluye actividades análogas a las etapas 2 a la 5. Estas actividades se pueden redefinir de la siguiente manera:

1. Desarrollo de diseños alternativos
2. Construcción de modelos que formalicen los diseños alternativos
3. Determinación del cociente costo/eficacia de los diseños alternativos;
4. Presentación de recomendaciones.

Las actividades de que se compone la etapa denominada implantación de sistemas se podrán igualar a la etapa 6, implantación de la alternativa seleccionada.

Como sucede con toda entidad dinámica, la utilidad de la salida del sistema de información, la eficiencia de sus operaciones y la confiabilidad en la operación global del sistema pueden variar de manera considerable con el tiempo. Por consiguiente, el sistema de información está expuesto a la deterioración, obsolescencia y, por último, a la sustitución. No obstante, rara vez acontece que en un momento dado, se le hagan mejoras al sistema de información completo. En consecuencia, es evidente que parte del sistema de información, o un subsistema de la organización, necesitará continuamente reparación o reconstrucción.

A primera vista se podría afirmar que la duración del ciclo equivale a la razón con que el ciclo vital del sistema se repite una y otra vez. Muchos expertos en sistemas de información calculan que un ciclo de vida normal para un sistema es de dos a diez años antes de que sea necesario un rediseño importante o una reparación total. Este cálculo acerca del ciclo vital de un sistema de información está sujeto a muchas condiciones y es necesario realizarlo por separado para cada sistema o subsistema de información. (Fig. 1.5) (Bursh, 1996).



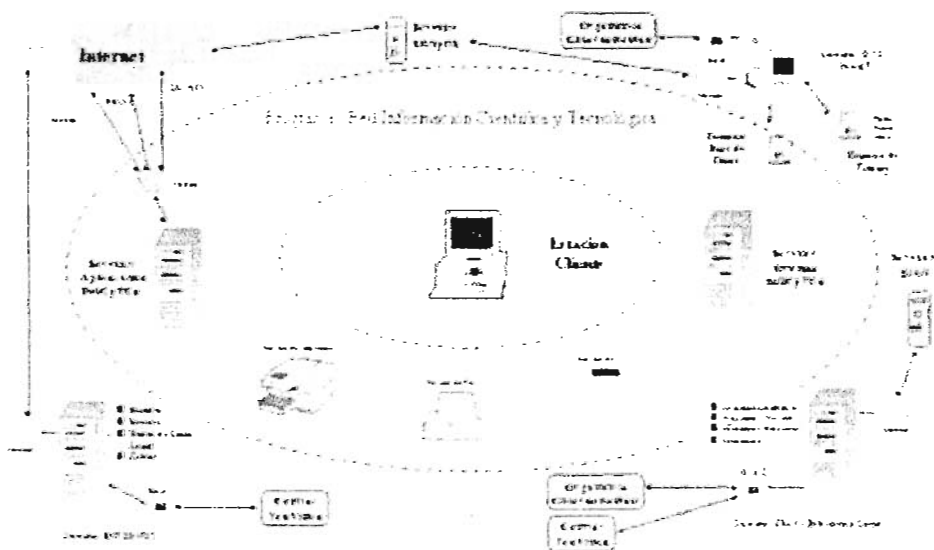
1.5 Separación del ciclo vital de un sistema de información

1.5.-Función de los sistemas de información.

La función de un sistema de información:

1. Un sistema de información se necesita con el propósito de usarlo como auxiliar de otro sistema, el sistema de objetos o sistema administrado. El sistema de información ha de suministrar la información necesaria en cualquier punto y en cualquier momento, en un sistema de objetos. Este último será a menudo una organización, es decir, una empresa o cuerpo administrativo.
2. Es usual comprobar que cierto tipo de información se requiere con frecuencia, pero, en realidad, solo en una parte limitada de objetos; en cambio, otra información reclama Datos originados en muchos puntos del sistema de objetos, pero puede realizarse un muestreo en puntos más distantes del tiempo.
3. En el diseño de un sistema de información es importante por supuesto, ser capaz de identificar situaciones de esta clase, por que al descuidar estas propiedades del sistema uno puede realizar una labor excesiva de transporte y procesamiento de Datos o tal vez menos de lo necesario. Ambos extremos originarán pérdidas.

4. A fin de determinar la cantidad exacta de información que se debe suministrar se tiene que ser capaz de descubrir por qué se necesita la información en un sistema. Se comprobará que es conveniente invertir la pregunta e inquirir, en cambio: ¿Cómo se puede usar la información?
5. Precisamente cuando un sistema cambia su estado en el tiempo, se necesita un sistema de información que actúe continuamente y, por lo tanto, que afronte el problema de diseñar el sistema de información que permita obtener eficientes flujos de información y recolectarla.
6. Si el sistema de objetos -servido (o controlado) por el sistema de información – es estático, el sistema de información adquiere un carácter meramente computacional, por ejemplo, en la creación de un sistema de ecuaciones, el descubrimiento de su solución, y su evaluación.
7. Una consecuencia del punto 5 es que el diseño del sistema de información se referirá, muy a menudo, a información vinculada con el control de un sistema. La figura 1.6 muestra un diagrama con la descripción de la función de un sistema de red de información Científica y Tecnológica. (Frischknecht, 1990).



1.6 Descripción de la función de un sistema de red de información científica y tecnológica

1.6 Futuro de los sistemas de información.

Ha sido aceptado que la confluencia de la Ingeniería de Software y de la Inteligencia Artificial que ha convertido la Teoría del Conocimiento, una vez más la más especulativa de las ciencias, en una ciencia empírica en que se puede aprender mediante experiencias vivenciales, generando nuevos modelos que posibilitan, nuevas formas de aprendizaje, el desarrollo de habilidades mentales superiores y potenciando la creatividad para la percepción y solución de problemas de la empresa en general. La ingeniería de software, unida a su propia matemática y a sus formas de legitimación y validación, ha demostrado su madurez que la llevó a independizarse de las tradicionales ciencias de la computación, presentando un cuerpo coherente de conocimientos generando grandes posibilidades industriales en todos los saberes disciplinarios, que caracterizan las sociedades modernas. El problema radica no tanto en la ponderación que pueda tener la ingeniería de software como rama especializada, sino en cuanto a su capacidad de contribuir al desarrollo de todos los demás sectores económicos, sociales y actividades de toda índole. En la nueva sociedad del conocimiento bien es conocido que a diferencia de la información, el conocimiento no se compra ni se copia: *se cultiva*, lo cual requiere nuevos modelos gerenciales y nuevos servicios de inferencia y descubrimiento lo cual se traduce indudablemente en nuevas oportunidades para la ingeniería de software cuyo aporte a la socio-economía y en particular a la industria del software se puede visualizar entre otros, en los siguientes aspectos:

- Como tecnología habilitadora y potenciadora de la competitividad y en general de la modernización industrial;
- En el mejoramiento cuantitativo y cualitativo del software para el aprendizaje de las nuevas habilidades gerenciales apoyada por las telecomunicaciones y los impresionantes avances de la realidad virtual,
- Las nuevas oportunidades laborales y el trabajo ubicuo que ha generado el comercio electrónico y las empresas virtuales;
- La eficiencia y eficacia de los nuevos sistemas de información en la empresa en todos sus niveles, potenciado por Internet y las tecnologías del conocimiento.

Como ejemplo se presenta la siguiente ilustración aplicada en sistemas de salud:



1.7 Aplicación de un sistema de salud

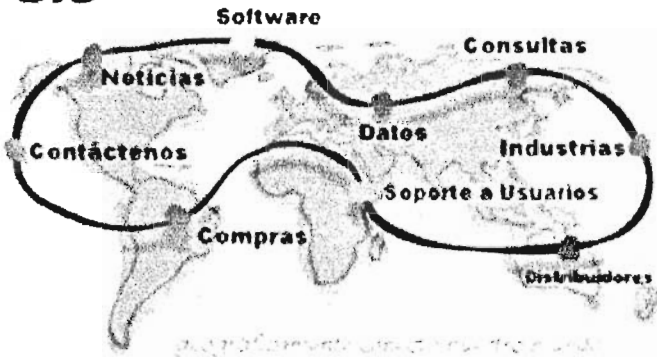
La demanda cada vez mayor de actividades científicas y de innovación tecnológica para la industria de software como nuevo reto para los jóvenes talentos.

La posición nacional de los sistemas de información gerencial y de Datos geográficos a la hora de compartir ventajas en el proceso de integración;

La pronta inserción del país en la era del conocimiento y en particularidad en la gerenciabilidad del conocimiento. (SISISSC, 2001).

El principal reto que enfrentan los analistas de sistemas de hoy en día es construir sistemas de información que mejoren la habilidad de la organización para competir, ayudando a los trabajadores de la información y de operaciones a realizar un mejor trabajo. Para hacer frente a este reto se requiere que cambie la actitud mecanicista de los analistas de sistemas por una actitud con espíritu emprendedor e innovador. La figura 1.8, muestra una conceptualización artística de cómo estaría conectado el mundo con un sistema de información "gigantesco" (Burch, 1992).

GIS



1.8 Conceptualización del mundo interconectado por un sistema de información

Generalidades de Diseño de Software de Aplicación

2.1.-Introducción al diseño de software de aplicación.

El software de aplicación está diseñado y escrito para realizar tareas específicas personales, empresariales o científicas como el procesamiento de nóminas, la administración de los recursos humanos o el control de inventarios. Todas estas aplicaciones procesan Datos (recepción de materiales) y generan información (registros de nómina), para el usuario.

Sistemas Operativos: Un sistema Operativo (SO) es en sí mismo un programa de computadora. Sin embargo, es un programa muy especial, quizá el más complejo e importante en una computadora. El SO despierta a la computadora y hace que reconozca a la CPU, la memoria, el teclado, el sistema de vídeo y las unidades de disco. Además, proporciona la facilidad para que los usuarios se comuniquen con la computadora y sirve de plataforma a partir de la cual se corran programas de aplicación. Cuando se enciende una computadora, lo primero que ésta hace es llevar a cabo un autodiagnóstico llamado autoprueba de encendido (Power On Self Test, POST). Durante la POST, la computadora indentifica su memoria, sus discos, su teclado, su sistema de vídeo y cualquier otro dispositivo conectado a ella. Lo siguiente que la computadora hace es buscar un SO para arrancar (boot).

Una vez que la computadora ha puesto en marcha su SO, mantiene al menos parte de éste en su memoria en todo momento. Mientras la computadora esté encendida, el SO tiene 4 tareas principales.

1. Proporcionar ya sea una interfase de línea de comando o una interfase gráfica al usuario, para que este último se pueda comunicar con la computadora. Interfase de línea de comando: se introducen palabras y símbolos desde el teclado de la computadora, ejemplo, el MS-DOS. Interfase gráfica del Usuario (GUI), seleccionas las acciones mediante el uso de un Ratón para pulsar sobre figuras llamadas iconos o seleccionar opciones de los menús.

2. Administrar los dispositivos de hardware en la computadora. Cuando corren los programas, necesitan utilizar la memoria, el monitor, las unidades de disco, los puertos de Entrada/Salida (impresoras, módem, etc.). El SO sirve de intermediano entre los programas y el hardware.

3.Administrar y mantener los sistemas de archivo de disco. Los SO agrupan la información dentro de compartimentos lógicos para almacenarlos en el disco. Estos grupos de información son llamados archivos. Los archivos pueden contener instrucciones de programas o información creada por el usuario. El SO mantiene una lista de los archivos en un disco, y proporciona las herramientas necesarias para organizar y manipular estos archivos.

4.Apoyar a otros programas. Otra de las funciones importantes del SO es proporcionar servicios a otros programas. Estos servicios son similares a aquellos que el SO proporciona directamente a los usuarios. Por ejemplo, listar los archivos, grabarlos a disco, eliminar archivos, revisar espacio disponible, etcétera.

Cuando los programadores escriben programas de computadora, incluyen en sus programas instrucciones que solicitan los servicios del SO. Estas instrucciones son conocidas como "llamadas del sistema". (Dir. [http://www.salohogar.com/ciencias/tecnologias/computadoras/software de aplicaciones.htm](http://www.salohogar.com/ciencias/tecnologias/computadoras/software_de_aplicaciones.htm))

2.2.-Software orientado a objetos.

Significa que el software se organiza como una colección de objetos discretos que contiene tanto estructura de Datos como también un comportamiento, sus características son:

Identidad: Los Datos están cuantificados en entidades discretas y distinguibles denominadas objetos, por ejemplo una persona, los objetos pueden ser tangibles (como un archivo) o intangibles.

Clasificación: Los objetos con la misma estructura de Datos (atributos) y comportamiento (operaciones) se agrupan para formar una misma clase, se dice que cada objeto es una instancia de su propia clase, y una clase es una abstracción que describe propiedades importantes para una aplicación y se olvida del resto.

Polimorfismo: Significa que una misma operación puede comportarse de modos distintos en distintas clases, una operación es una acción o transformación que se aplica a un objeto

Herencia: Comparte atributos y operaciones entre clases tomando como Base una relación jerárquica, es decir que puedo definir una clase que después producirá subclases, sabiendo que todas las subclases adquirirán todas y cada una de las propiedades de su super-clase y le agrega además sus propiedades exclusivas.(UTNFRT, Cd. San Miguel de Tucumán – Argentina)

2.3.-Proceso de desarrollo de software.

Definición: Es el conjunto de técnicas y procedimientos que permiten conocer los elementos necesarios para definir un proyecto de software.

Es la etapa más crucial del desarrollo de un proyecto de software.

La IEEE los divide en funcionales y no funcionales:

Funcionales: Condición o capacidad de un sistema requerida por el usuario para resolver un problema o alcanzar un objetivo.

No Funcionales: Condición o capacidad que debe poseer un sistema para satisfacer un contrato, un estándar, una especificación u otro documento formalmente impuesto.

Para realizar bien el desarrollo de software es esencial realizar una especificación completa de los requerimientos de los mismos. Independientemente de lo bien diseñado o codificado que esté, un programa pobremente especificado decepcionará al usuario y hará fracasar el desarrollo.

La tarea de análisis de los requerimientos es un proceso de descubrimiento y refinamiento. El ámbito del programa, establecido inicialmente durante la ingeniería del sistema, es refinado en detalle. Se analizan y asignan a los distintos elementos de los programas las soluciones alternativas.

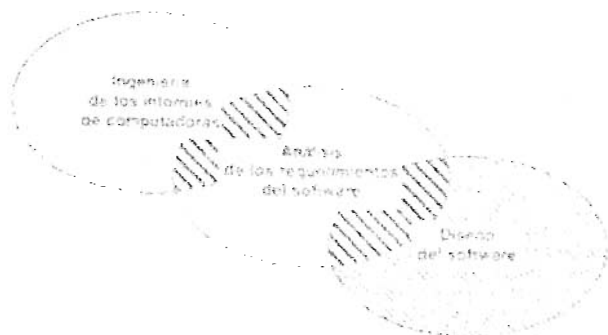
Tanto el que desarrolla el software como el cliente tienen un papel activo en la especificación de requerimientos. El cliente intenta reformular su concepto, algo nebuloso, de la función y comportamiento de los programas en detalles concretos. El que desarrolla el software actúa como interrogador, consultor y el que resuelve los problemas.

El análisis y especificación de requerimientos puede parecer una tarea relativamente sencilla, pero las apariencias engañan. Puesto que el contenido de comunicación es muy alto, abundan los cambios por mala interpretación o falta de información. El dilema con el que se enfrenta un ingeniero de software puede ser comprendido repitiendo la sentencia de un cliente anónimo: "Sé que crees que comprendes lo que piensas que he dicho, pero no estoy seguro de que lo que creíste oír sea lo que yo quise decir".

2.3.1.-Análisis de Requerimientos.

El análisis de requerimientos es la tarea que plantea la asignación de software a nivel de sistema y el diseño de programas (Figura 2.1). El análisis de requerimientos facilita al ingeniero de sistemas especificar la función y comportamiento de los programas, indicar la interfase con otros elementos del sistema y establecer las ligaduras de diseño que debe cumplir el programa.

El análisis de requerimientos permite al ingeniero refinar la asignación de software y representar el dominio de la información que será tratada por el programa. El análisis de requerimientos de al diseñador la representación de la información y las funciones que pueden ser traducidas en Datos, arquitectura y diseño procedimental. Finalmente, la especificación de requerimientos suministra al técnico y al cliente, los medios para valorar la calidad de los programas, una vez que se haya construido.



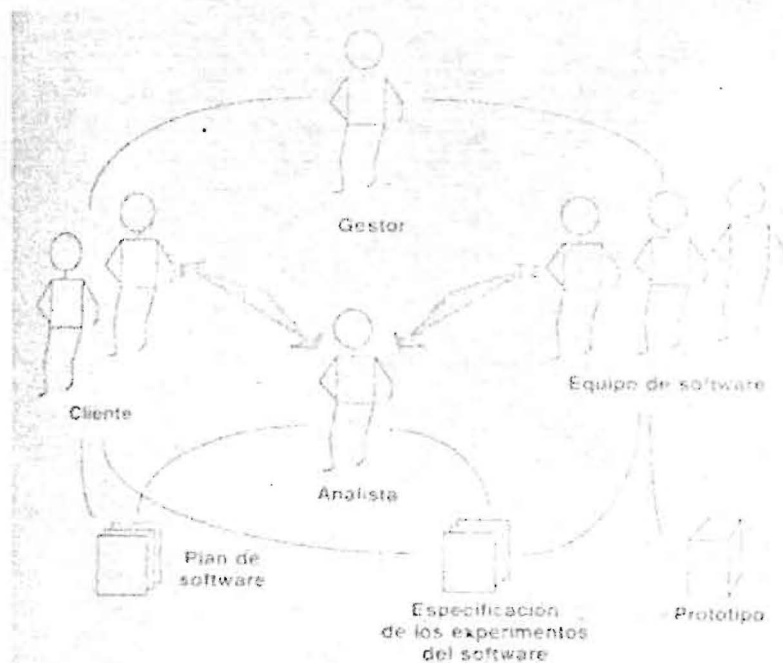
2.1 Tareas del Análisis

El análisis de requerimientos puede dividirse en cuatro áreas:

- 1.- Reconocimiento del problema
- 2.- Evaluación y síntesis
- 3.- Especificación
- 4.- Revisión.

Inicialmente, el analista estudia la especificación del sistema (si existe) y el plan de proyecto. Es importante comprender el contexto del sistema y revisar el ámbito de los programas que se usó para generar las estimaciones de la planificación. A continuación, debe establecerse la comunicación necesaria para el análisis, de forma que se asegure el reconocimiento del problema.

Las formas de comunicación requeridas para el análisis se ilustran en la Figura 2.2. El analista debe establecer contacto con el equipo técnico y de gestión del usuario / cliente y con la empresa que vaya a desarrollar el software. El gestor del programa puede servir como coordinador para facilitar el establecimiento de los caminos de comunicación. El objetivo del analista es reconocer los elementos básicos del programa tal como lo percibe el usuario / cliente.



2.2 Formas de comunicación requeridas para el análisis

La evaluación del problema y la síntesis de la solución es la siguiente área principal de trabajo del análisis. El analista debe evaluar el flujo y estructura de la información, refinar en detalle todas las funciones del programa, establecer las características de la interfase del sistema y descubrir las ligaduras del diseño. Cada una de las tareas sirven para descubrir el problema de forma que pueda sintetizarse un enfoque o solución global.

Las tareas asociadas con el análisis y especificación existen para dar una representación del programa que pueda ser revisada y aprobada por el cliente. En un mundo ideal el cliente desarrolla una especificación de requerimientos del software completamente por sí mismo. Esto se presenta raramente en el mundo real. En el mejor de los casos, la especificación se desarrolla conjuntamente entre el cliente y el técnico.

Una vez que se hayan descrito las funcionalidades básicas, comportamiento, interfase e información, se especifican los criterios de validación para demostrar una comprensión de una correcta implantación de los programas. Estos criterios sirven como Base para hacer una prueba durante el desarrollo de los programas. Para definir las características y atributos del software se escribe una especificación de requerimientos formal. Además, para los casos en los que se desarrolle un prototipo se realiza un manual de usuario preliminar.

Puede parecer innecesario realizar un manual de usuario en una etapa tan temprana del proceso de desarrollo, pero de hecho, este borrador del manual de usuario fuerza al analista a tomar el punto de vista del usuario del software. El manual permite al usuario / cliente revisar el software desde una perspectiva de ingeniería humana y frecuentemente produce el comentario: "La idea es correcta pero ésta no es la forma en que pensé que se podría hacer esto". Es mejor descubrir tales comentarios lo mas tempranamente posible en el proceso.

Los documentos del análisis de requerimiento (especificación y manual de usuario) sirven como Base para una revisión conducida por el cliente y el técnico. La revisión de los requerimientos casi siempre produce modificaciones en la función, comportamiento, representación de la información, ligaduras o criterios de validación. Además, se realiza una nueva apreciación del plan del proyecto de software para determinar si las primeras estimaciones siguen siendo validas después del conocimiento adicional obtenido durante el análisis.

2.3.2.-Principios del Análisis.

En la pasada década, se desarrollaron varios métodos de análisis y especificación del software. Los investigadores han identificado los problemas y sus causas y están desarrollando reglas y procedimientos para resolverlos. Cada método de análisis tiene una única notación y punto de vista. Sin embargo, todos los métodos de análisis están relacionados por un conjunto de principios fundamentales:

- El dominio de la información, así como el dominio funcional de un problema debe ser representado y comprendido.
- El problema debe subdividirse de forma que se descubran los detalles de una manera progresiva (o jerárquica)
- Deben desarrollarse las representaciones lógicas y físicas del sistema.

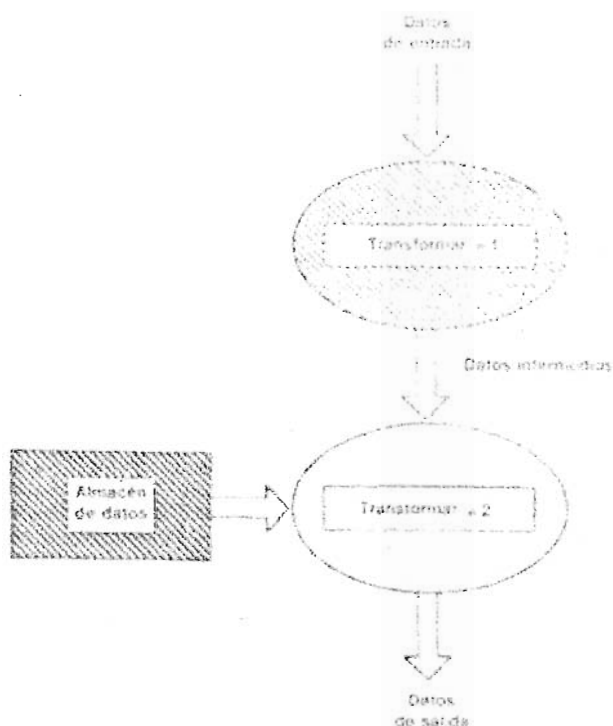
Aplicando estos principios, el analista enfoca el problema sistemáticamente. Se examina el dominio de la información de forma que pueda comprenderse su función más completamente. La partición se aplica para reducir la complejidad.

La visión lógica y física del software, es necesaria para acomodar las ligaduras lógicas impuestas por los requerimientos de procesamiento, y las ligaduras físicas impuestas por otros elementos del sistema.

2.3.3.-El dominio de la Información.

Todas las aplicaciones del software pueden colectivamente llamarse procesamiento de Datos. Este término contiene la clave de lo que se entiende por requerimientos del software. El software se construye para procesar Datos; para transformar Datos de una forma a otra; esto es, para aceptar entrada, manipularla de alguna forma y producir una salida.

El flujo de la información representa la manera en la que los Datos cambian conforme pasan a través de un sistema. Refiriéndose a la Figura 2.3, la entrada se transforma en Datos intermedios y más adelante se transforma en la salida.



2.3 Transformación de los Datos de entrada

2.3.4.-Partición.

Normalmente, los problemas son demasiado grandes y complejos para ser comprendidos como un todo. Por esta razón, se tiende a particionar (dividir) tales problemas en partes que puedan ser fácilmente comprendidas, y establecer interfases entre las partes, de forma que se realice la función global.

Durante el análisis de requerimientos, el dominio funcional y el dominio de la información del software pueden ser particionados.

En esencia la partición descompone un problema en sus partes constituyentes. Conceptualmente, se establece una representación jerárquica de la función o información y luego se parte el elemento superior mediante: 1) el incremento de los detalles, moviéndose verticalmente en la jerarquía, o 2) descomponiendo funcionalmente el problema, moviéndose horizontalmente en la jerarquía.

2.3.5.-Construcción de Prototipos de Software.

En análisis debe ser conducido independientemente del paradigma de ingeniería de software aplicado. Sin embargo, la forma que ese análisis tomara puede variar. En algunos casos es posible aplicar los principios de análisis fundamental y derivar a una especificación en papel del software desde el cual pueda desarrollarse un diseño. En otras situaciones, se va a una recolección de los requerimientos; se aplican los principios de análisis y se construye un modelo de software, llamado un prototipo, según las apreciaciones del cliente y de quien lo desarrolla. Finalmente, hay circunstancias que requieren la construcción de un prototipo al comienzo del análisis, puesto que el modelo es el único mediante el que los requerimientos pueden ser derivados efectivamente.

2.3.6.-Un escenario para la construcción de prototipos.

Todos los proyectos de ingeniería de software comienzan con una petición del cliente. La petición puede estar en la forma de una memoria que describe un problema, un informe que define un conjunto de objetivos comerciales o del producto, una petición de propuesta formal de una agencia o compañía exterior, o una especificación del sistema que ha asignado una función y comportamiento al software, como un elemento de un sistema mayor basado en computadora. Suponiendo que existe una petición para un programa de una de las formas dichas anteriormente, para construir un prototipo del software se aplican los siguientes pasos:

PASO 1. Evaluar la petición del software y determinar si el programa a desarrollar es un buen candidato para construir un prototipo.

Debido a que el cliente debe interactuar con el prototipo en los últimos pasos, es esencial que: 1) el cliente participe en la evaluación y refinamiento del prototipo, y 2) el cliente sea capaz de tomar decisiones de requerimientos de una forma oportuna. Finalmente, la naturaleza del proyecto de desarrollo tendrá una fuerte influencia en la eficacia del prototipo.

PASO 2. Dado un proyecto candidato aceptable, el analista desarrolla una representación abreviada de los requerimientos.

Antes de que pueda comenzar la construcción de un prototipo, el analista debe representar los dominios funcionales y de información del programa y desarrollar un método razonable de partición. La aplicación de estos principios de análisis fundamentales, pueden realizarse mediante los métodos de análisis de requerimientos.

PASO 3. Después de que se haya revisado la representación de los requerimientos, se crea un conjunto de especificaciones de diseño abreviadas para el prototipo.

El diseño debe ocurrir antes de que comience la construcción del prototipo. Sin embargo, el diseño de un prototipo se enfoca normalmente hacia la arquitectura a nivel superior y a los aspectos de diseño de Datos, en vez de hacia el diseño procedimental detallado.

PASO 4. El software del prototipo se crea, prueba y refina

Idealmente, los bloques de construcción de software preexistentes se utilizan para crear el prototipo de una forma rápida. Desafortunadamente, tales bloques contruidos raramente existen.

Incluso si la implantación de un prototipo que funcione es impracticable, es escenario de construcción de prototipos puede aún aplicarse. Para las aplicaciones interactivas con el hombre, es posible frecuentemente crear un prototipo en papel que describa la interacción hombre-máquina usando una serie de hojas de historia.

PASO 5. Una vez que el prototipo ha sido probado, se presenta al cliente, el cual "conduce la prueba" de la aplicación y sugiere modificaciones.

Este paso es el núcleo del método de construcción de prototipo. Es aquí donde el cliente puede examinar una representación implantada de los requerimientos del programa, sugerir modificaciones que harán al programa cumplir mejor las necesidades reales.

PASO 6. Los pasos 4 y 5 se repiten interactivamente hasta que todos los requerimientos estén formalizados o hasta que el prototipo haya evolucionado hacia un sistema de producción.

El paradigma de construcción del prototipo puede ser conducido con uno o dos objetivos en mente: 1) el propósito del prototipo es establecer un conjunto de requerimientos formales que pueden luego ser traducidos en la producción de programas mediante el uso de métodos y técnicas de ingeniería de programación, o 2) el propósito de la construcción del prototipo es suministrar un continuo que pueda conducir al desarrollo evolutivo de la producción del software. Ambos métodos tienen sus méritos y ambos crean problemas.

2.3.7.-Especificación.

No hay duda de que la forma de especificar tiene mucho que ver con la calidad de la solución. Los ingenieros de software que se han esforzado en trabajar con especificaciones incompletas, inconsistentes o mal establecidas han experimentado la frustración y confusión que invariablemente se produce. Las consecuencias se padecen en la calidad, oportunidad y completitud del software resultante.

La especificación, independientemente del modo en que se realice, puede ser vista como un proceso de representación. Los requerimientos se representan de forma que conduzcan finalmente a una correcta implantación del software.

Baltzer y Goldman (1979) proponen ocho principios para una buena especificación:

PRINCIPIO #1. Separar funcionalidad de implantación.

Primero, por definición, una especificación es una descripción de lo que se desea, en vez de cómo se realiza (implementa). Las especificaciones pueden adoptar dos formas muy diferentes. La primera forma es la de funciones matemáticas: dado algún conjunto de entrada, producir un conjunto particular de salida. La forma general de tales especificaciones es encontrar [un/el/todos] resultado tal que P (entrada), donde P representa un predicado arbitrario. En tales especificaciones, el resultado a ser obtenido ha sido expresado enteramente en una forma sobre el que (en vez de cómo). En parte, esto es debido a que el resultado es una función matemática de la entrada (la operación tiene puntos de comienzo y parada bien definidos) y no está afectado por el entorno que le rodea.

PRINCIPIO #2. Se necesita un lenguaje de especificación de sistemas orientado al proceso.

Considerar una situación en la que el entorno sea dinámico y sus cambios afecten al comportamiento de alguna entidad que interactúe con dicho entorno. Su comportamiento no puede ser expresado como una función matemática de su entrada. En vez de ello, debe emplearse una descripción orientada al proceso, en la cual la especificación del que se consigue mediante la especificación de un modelo del comportamiento deseado en términos de respuestas funcionales, a distintos estímulos del entorno.

PRINCIPIO #3. Una especificación debe abarcar el sistema del cual el software es una componente.

Un sistema está compuesto de componentes que interactúan. Solo dentro del contexto del sistema completo y de la interacción entre sus partes puede ser definido el comportamiento de una componente específica. En general, un sistema puede ser modelado como una colección de objetos pasivos y activos. Estos objetos están interrelacionados y dichas relaciones entre los objetos cambian con el tiempo. Estas relaciones dinámicas suministran los estímulos a los cuales los objetos activos, llamados agentes, responden. Las respuestas pueden causar posteriormente cambios y, por tanto, estímulos adicionales a los cuales los agentes deben responder.

PRINCIPIO #4. Una especificación debe abarcar el entorno en el que el sistema opera.

Similarmente, el entorno en el que opera el sistema y con el que interactúa debe ser especificado.

Afortunadamente, esto tan solo necesita reconocer que el propio entorno es un sistema compuesto de objetos que interactúan, pasivos y activos, de los cuales el sistema especificado es una agente. Los otros agentes, los cuales son por definición inalterables debido a que son parte del entorno, limitan el ámbito del diseño subsecuente y de la implantación. De hecho, la única diferencia entre el sistema y su entorno es que el esfuerzo de diseño e implantación subsecuente opera exclusivamente sobre la especificación del sistema. La especificación del entorno facilita que se especifique la interfase del sistema de la misma forma que el propio sistema, en vez de introducir otro formalismo.

PRINCIPIO #5. Una especificación de sistema debe ser un modelo cognitivo.

La especificación de un sistema debe ser un modelo cognitivo, en vez de un modelo de diseño o implantación. Debe describir un sistema tal como es percibido por su comunidad de usuario. Los objetivos que manipula deben corresponderse con objetos reales de dicho dominio; los agentes deben modelar los individuos, organizaciones y equipo de ese dominio; y las acciones que ejecutan deben modelar lo que realmente ocurre en el dominio. Además, debe ser posible incorporar en la especificación las reglas o leyes que gobiernan los objetos del dominio. Algunas de estas leyes proscriben ciertos estados del sistema (tal como "dos objetos no pueden estar en el mismo lugar al mismo tiempo"), y por tanto limitan el comportamiento de los agentes o indican la necesidad de una posterior elaboración para prevenir que surjan estos estados.

PRINCIPIO #6. Una especificación debe ser operacional.

La especificación debe ser completa y lo bastante formal para que pueda usarse para determinar si una implantación propuesta satisface la especificación de pruebas elegidas arbitrariamente. Esto es, dado el resultado de una implantación sobre algún conjunto arbitrario de Datos elegibles, debe ser posible usar la especificación para validar estos resultados. Esto implica que la especificación, aunque no sea una especificación completa del como, pueda actuar como un generador de posibles comportamientos, entre los que debe estar la implantación propuesta. Por tanto, en un sentido extenso, la especificación debe ser operacional.

PRINCIPIO #7. La especificación del sistema debe ser tolerante con la incompletitud y aumentable.

Ninguna especificación puede ser siempre totalmente completa. El entorno en el que existe es demasiado complejo para ello. Una especificación es siempre un modelo, una abstracción, de alguna situación real (o imaginada).

Por tanto, será incompleta. Además, al ser formulada existirán muchos niveles de detalle. La operacionalidad requerida anteriormente no necesita ser completa. Las herramientas de análisis empleadas para ayudar a los especificadores y para probar las especificaciones, deben ser capaces de tratar con la incompletitud.

Naturalmente esto debilita el análisis, el cual puede ser ejecutado ampliando el rango de comportamiento aceptables, los cuales satisfacen la especificación, pero tal degradación debe reflejar los restantes niveles de incertidumbre.

PRINCIPIO #8. Una especificación debe ser localizada y débilmente acoplada.

Los principios anteriores tratan con la especificación como una entidad estática. Esta surge de la dinámica de la especificación. Debe ser reconocido que aunque el principal propósito de una especificación sea servir como Base para el diseño e implantación de algún sistema, no es un objeto estático precompuesto, sino un objeto dinámico que sufre considerables modificaciones. Tales modificaciones se presentan en tres actividades principales: formulación, cuando se está creando una especificación inicial; desarrollo, cuando la especificación se está elaborando durante el proceso iterativo de diseño e implantación; y mantenimiento, cuando la especificación se cambia para reflejar un entorno modificado y/o requerimientos funcionales adicionales.

2.3.8.-Metodologías de Análisis de Requerimientos.

Las metodologías de análisis de requerimientos facilitan al analista la aplicación de los principios fundamentales del análisis de una manera sistemática.

Características Comunes

Aunque cada método introduce nueva notación y heurística de análisis, todos los métodos pueden ser evaluados en el contexto de las siguientes características comunes:

1. Mecanismos para el análisis del dominio de la información
2. Método de representación funcional
3. Definición de interfaces
4. Mecanismos para subdividir el problema
5. Soporte de la abstracción
6. Representación de visiones físicas y lógicas

Aunque el análisis del dominio de la información se conduce de forma diferente en cada metodología, pueden reconocerse algunas guías comunes. Todos los métodos se enfocan (directa o indirectamente) al flujo de Datos y al contenido o estructura de Datos.

En la mayoría de los casos el flujo se caracteriza en el contexto de las transformaciones (funciones) que se aplican para cambiar la entrada en la salida.

El contenido de los Datos puede representarse explícitamente usando un mecanismo de diccionario o, implícitamente, enfocando primero la estructura jerárquica de los Datos.

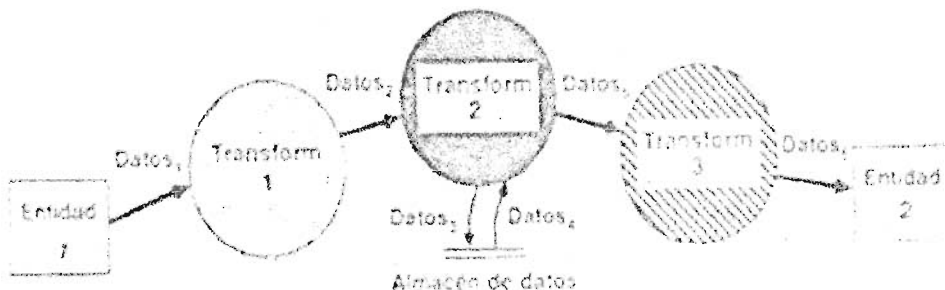
Las funciones se describen normalmente como transformaciones o procesos de la información. Cada función puede ser representada usando una notación específica. Una descripción de la función puede desarrollarse usando el lenguaje natural, un lenguaje procedimental con reglas sintácticas informales o un lenguaje de especificación de forma.

2.3.9.-Métodos de Análisis Orientados al Flujo de Datos.

La información se transforma como un flujo a través de un sistema basado en ordenador. El sistema acepta entrada de distintas formas; aplica un hardware, software y elementos humanos para transformar la entrada en salida; y produce una salida en distintas formas. La entrada puede ser una señal de control transmitida por un transductor, una serie de números escritos por un operador humano, un paquete de información transmitido por un enlace a red, o un voluminoso archivo de Datos almacenado en memoria secundaria. La transformación puede comprender una sencilla comparación lógica, un complejo algoritmo numérico, o un método de inferencia basado en regla de un sistema experto. La salida puede encender un sencillo led o producir un informe de 200 paginas. En efecto, un modelo de flujo de Datos puede aplicarse a cualquier sistema basado en ordenador independientemente del tamaño o complejidad.

Una técnica para representar el flujo de información a través del sistema basado en ordenador se ilustra en la figura 4. La función global del sistema se representa como una transformación sencilla de la información, representada en la figura como una burbuja. Una o más entradas. Representadas como flechas con etiqueta, conducen la transformación para producir la información de salida.

Puede observarse que el modelo puede aplicarse a todo el sistema o solo a un elemento de software. La clave es representar la información dada y producida por la transformación.



2.4 Modelo de aplicación a un sistema o solo a un elemento de software

2.3.10.-Diagramas de Flujos de Datos.

Conforme como la información se mueve a través del software, se modifica mediante una serie de transformaciones. Un diagrama de flujos de Datos (DFD), es una técnica gráfica que describe el flujo de información y las transformaciones que se aplican a los Datos, conforme se mueven de la entrada a la salida. La forma básica de un DFD se ilustra en la figura 2.5. El diagrama es similar en la forma a otros diagramas de flujo de actividades, y ha sido incorporado en técnicas de análisis y diseños propuesto por Yourdon y Constantine, DeMarco y Gane y Sarson. También se le conoce como un grafo de flujo de Datos o un diagrama de burbujas.



2.5 Forma básica de un DFD

Como los métodos orientados al flujo de Datos, los métodos de análisis orientados a la estructura de Datos proporcionan la Base para el diseño de software. Siempre puede extenderse un método de análisis para que abarque el diseño arquitectural y procedimental del software. (Roger, 1990)

Fundamentos y Herramientas para la Elaboración de Bases de Datos e Interfases Gráficas

3.1.-Introducción a las interfases gráficas (GUIs).

Las interfases gráficas son una representación gráfica en la pantalla del ordenador de los programas, Datos, objetos, así como la interacción con ellos. Una GUI proporciona al usuario las herramientas para realizar sus operaciones, más que una lista de las posibles operaciones que el ordenador es capaz de hacer. Las interfases tienen una mejor comunicación con el usuario ya que éstas serán creadas para ellos, de la manera que sea lo mejor adaptable posible para su utilización.

El ser humano, está continuamente interactuando con los objetos que le rodean, y tiene unas expectativas sobre cómo éstos deben comportarse, basadas en pasadas experiencias con estos objetos u otros similares. Si la interfase está bien diseñada, el usuario encontrará la respuesta que espera a su acción; si no es así, puede ser frustrante para el usuario, que habitualmente tiende a culparse a sí mismo por no saber usar el objeto. Es el conjunto de elementos a través de los cuales un usuario interactúa con un objeto que realiza una tarea determinada. Ejemplos de esto son: televisor, automóvil, despertador, etcétera.

La interfase incluye el hardware que forma el sistema, como el teclado, un dispositivo apuntador tal como un ratón, joystick o trackball, la UCP y el monitor y los componentes de software son los elementos que el usuario ve, oye, a los que apunta o toca en la pantalla para interactuar con el ordenador, además también se puede considerar parte de la interfase la documentación, como manuales, ayuda, referencia y tutoriales que acompañan al hardware y al software, para lo cual será necesario tomar en cuenta todo lo que se menciona en este capítulo.

Las GUIs proporcionan un entorno sencillo para introducir y modificar Datos, deben proporcionar una interfase entre la máquina y el usuario que sea agradable y cómodo, tan sencillo para que un usuario, sin conocimientos de computación, pueda usar el sistema. Para poder proporcionar un entorno sencillo las GUIs, se menciona lo siguiente:

- Utilizan elementos de control visuales: botones, menús, íconos, barras, etc.
- Los Datos pueden manipularse en pantalla de manera intuitiva.
- Si se utiliza una GUI como estándar, se consigue una gran consistencia a través de distintas plataformas y aplicaciones. El tiempo de aprendizaje de las nuevas aplicaciones se ve minimizado, sobre todo, si el usuario está familiarizado con Microsoft Windows o Apple Macintosh.

- Al poder ejecutarse distintos procesos en la misma máquina, las GUI permiten un mejor manejo, al nivel de ventanas, de dichos procesos, pudiendo asignarse una ventana (o más) para cada aplicación. (Eran / Baker, 1999)

3.2.-Características de una interfase.

Una interfase deberá de tener como características lo siguiente:

1. Posee un modo gráfico de alta resolución.
2. Posee un dispositivo apuntador (Ratón).
3. Promueve la consistencia de la interfase entre programas.
4. Los usuarios pueden ver en la pantalla los gráficos y textos tal como se verán impresos.
5. Permite la transferencia de información entre programas.
6. Se puede manipular en la pantalla directamente los objetos y la información.
7. Provee elementos de interfase estándar como menús y diálogos.
8. Existe una muestra visual de la información y los objetos (íconos y ventanas).
9. Proporciona respuesta visual a las acciones del usuario.
10. Existen controles gráficos (widgets) para la selección e introducción de la información.
11. Permite a los usuarios personalizar la interfase y las interacciones.
12. Proporciona flexibilidad en el uso de dispositivos de entrada (teclado, ratón). (Eran / Baker, 1999)

3.3.-Interfase de usuario. (UI)

La Interfase de Usuario, de un programa es un conjunto de elementos *hardware* y *software* de un ordenador que presentan información al usuario y le permiten interactuar con la información y con el ordenador. También se puede considerar parte de la IU la documentación (manuales, ayuda, referencia, tutoriales) que acompaña al *hardware* y al *software*.

Si la IU está bien diseñada, el usuario encontrará la respuesta que espera a su acción. Si no es así, puede ser frustrante su operación, ya que el usuario habitualmente tiende a culparse a sí mismo por no saber usar el objeto.

Los programas son usados por usuarios con distintos niveles de conocimientos, desde principiantes hasta expertos. Es por ello que no existe una interfase válida para todos los usuarios y todas las tareas. Debe permitirse libertad al usuario para que elija el modo de interacción que más se adecue a sus objetivos en cada momento. La mayoría de los programas y sistemas operativos ofrecen varias formas de interacción al usuario.

La interfase de usuario es quizá la parte más importante de la aplicación, dado que es el componente que define la percepción del comportamiento del programa en el usuario final. La interfase gráfica de usuario es una forma de comunicarse entre la aplicación y el usuario o se puede decir que es el conjunto de componentes empleados por los usuarios para comunicarse con los ordenadores. (Eran / Baker, 1999, MCSS, M en I S, EP, ITBA. Molich, / Nielsen, 1990)

3.3.1.-Cómo usan una GUI los usuarios.

Para usar una GUI los usuarios deben conocer o aprender una serie de conceptos:

1. Conocer la organización del sistema (ficheros, directorio en Windows)
2. Conocer los diferentes tipos de realimentación y efectos de las acciones sobre ellos.
3. Conocer los elementos básicos de una ventana.
4. Aprender el uso de los controles de una GUI.
5. Aprender el uso del ratón.

(Eran / Baker, 1999)

3.3.2.-Evolución de las interfases de usuarios.

La evolución de las interfases de usuario corre en paralelo con el de los sistemas operativos; de hecho la interfase constituye actualmente uno de los principales elementos de un sistema operativo.

Algunos inconvenientes y ventajas de las interfases son:

Inconveniente de las interfases:

- Carga de memoria del usuario.
- Debe memorizar los manDatos.
- La ayuda es difícil de leer.
- El significado de los manDatos es mal comprendido a veces (varios manDatos con el mismo o parecido significado, como DEL o ERASE).
- Inflexibilidad en los nombres (DEL y DELETE).

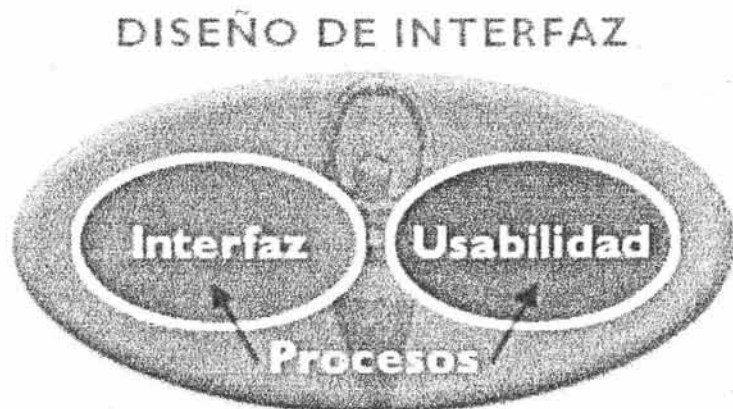
Ventajas de las interfases:

- Potente.
 - Flexible y controlado por el usuario, aunque esto sea una ventaja para usuarios experimentados.
 - La sintaxis es estricta, y los errores pueden ser graves como el comando COPY, si el destino es un directorio, y se teclea mal es un error de sintaxis.
- (Eran / Baker, 1999)

3.4.-Diseño de una interfase gráfica.

El diseño de interfases de usuario es una tarea que ha adquirido relevancia en el desarrollo de un sistema. La calidad de la interfase de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso. Los principios que se presentan son de utilidad para creación de interfases funcionales y de fácil operación. A pesar de no ser capaces de resolver todos los aspectos propios del contexto con el que se esté trabajando, pueden ser combinados con el prototipo y la aplicación de heurísticas de evaluación para facilitar el proceso de diseño. El presente artículo se centra en los componentes de software de las interfases de usuario, quedando fuera del alcance de mismo otros aspectos, como hardware y documentación. Lo anteriormente expuesto se complementa con un caso práctico de diseño de interfases de usuario, producto de realizar la actividad de "Definición de Interfases de Usuario" (EFS 4) de la metodología Métrica Versión 2. (Key words, 1990)

Una interfase se debe diseñar de manera que proporcione un medio adecuado y efectivo para que el usuario acceda a funciones de gráfica básicas, como el despliegue de objetos, establecimiento de atributos o realización de transformaciones. Ver figura 3.1.



3.1 Establecimiento de atributos o realización de transformaciones

Es muy importante conocer a fondo a los usuarios, ya que ellos podrían responder a muchas preguntas que probablemente se tendrán:

- ¿Qué nombres usar en cada elemento del menú?
- ¿Qué deben incluir los paquetes de entrenamiento y ayuda?
- ¿Qué características debe proveer el sistema?

Por ejemplo, un sistema para usuarios de Windows debe proveer las características generales con los cuales los usuarios de este sistema operativo están acostumbrados a trabajar, esto probablemente incluya la forma de cortar y pegar información de la misma manera, aunque estas dos acciones no jueguen un papel importante en la funcionalidad principal del sistema. Ver figura 3.2. (Eran / Baker, 1999)



3.2 Pantalla de un sistema Windows

3.5.-Factores de diseño.

Existen muchos factores que se incluyen en el diseño de una interfase de usuario, pero hay tres puntos de vista distintos en una interfase del usuario que son:

- 1) El del usuario.
- 2) El del programador.
- 3) El del diseñador.

Cada uno tiene un modelo mental propio de la interfase, que contiene los conceptos y expectativas acerca de la interfase, desarrollados a través de su experiencia. (Eran / Baker, 1999)

3.5.1.-Modelos.

El modelo permite, entre otras cosas, explicar o predecir comportamientos del sistema y tomar las decisiones adecuadas para modificar el mismo.

- **Modelo del usuario.**-El usuario tiene su visión personal del sistema, y espera que éste se comporte de una cierta forma, que se pueda conocer estudiando al usuario realizando pruebas de uso, entrevistas, o a través de una realimentación.

La persona tiene algunas habilidades distintas a las de la máquina, y ésta debe utilizar las suyas para soslayar la de aquella. Al diseñar interfases de usuario deben tenerse en cuenta las habilidades cognitivas y de percepción de las personas, y adaptar el programa a ellas.

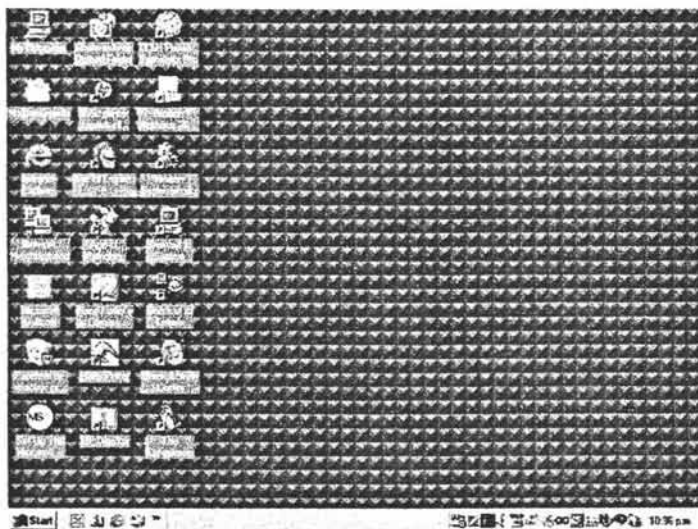
- **Modelo del programador.**-Es la más fácil de visualizar, al poderse especificar formalmente. Está constituido por los objetos que manipula el programador, distintos de los que trata el usuario (ejemplo: Base de Datos, agenda telefónica). Estos objetos deben esconderse del usuario.

Los conocimientos del programador incluyen la plataforma de desarrollo, el sistema operativo, las herramientas de desarrollo y especificaciones. Sin embargo, esto no significa necesariamente que tenga la habilidad de proporcionar al usuario los modelos y metáforas más adecuadas.

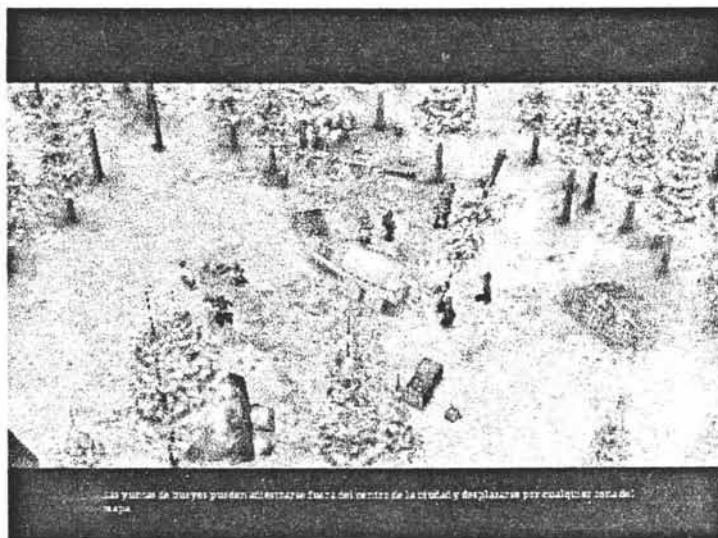
- **Modelo del diseñador.**-*"Most software is run by confused users acting on incorrect and incomplete information, doing things the designer never expected"* (Paul Heckel). El diseñador mezcla las necesidades, ideas, deseos del usuario y los materiales de que dispone el programador para diseñar un producto de software. Es pues, un intermediario entre ambos.

El modelo del diseñador describe los objetos que utiliza el usuario, su presentación al mismo y las técnicas de interacción para su manipulación.

Una interfase debe facilitar el proceso de crear un modelo mental efectivo. Para ello son de gran utilidad las metáforas¹, un ejemplo típico es la metáfora del escritorio común a la mayoría de las interfases gráficas actuales. Las metáforas constituyen una ayuda, relacionando el sistema con algo ya conocido por el usuario: escritorio y escenario de juegos son dos ejemplos de las ventajas del uso de gráficos. Ver figuras 3.3 y 3.4.



3.3 Metáfora del escritorio de Windows 98



3.4 Metáfora del escenario de juego

¹Metáfora. - desde el enfoque de computación es la muestra de un diseño a seguir

3.5.1.1.-Partes de que consta un modelo.

Número de partes	Nombre de la parte	Función
Primera	La presentación	Es la parte donde se capta la atención del usuario, pero más tarde pasa a un segundo plano, y adquiere más importancia cómo el producto cumple las expectativas del usuario.
Segunda	La técnica	Es la parte que define técnicas de interacción a través de diversos dispositivos.
Tercera	Determinación de la metáfora	Aquí el diseñador determina la metáfora adecuada que encaja con el modelo mental del usuario. El modelo debe comenzar por esta parte e ir hacia arriba. Una vez definida la metáfora y los objetos de la interfase, los aspectos visuales saldrán de una manera lógica y fácil.

3.5 Cuadro que representa las partes del modelo

Así, una de las cosas más importantes que una interfase puede hacer, es reducir la dependencia de las personas de su propia memoria, no forzándoles a recordar cosas innecesariamente, por ejemplo; información que apareció en una pantalla anterior o repetir operaciones ya realizadas, por ejemplo, introducir un mismo dato en varias ocasiones. (Eran / Baker, 1999)

3.5.2.-Lenguaje de comando.

El lenguaje elegido debe ser lo más natural posible para que el usuario lo aprenda, con todas las operaciones especificadas en términos relativos al área de aplicación, también debe efectuarse de modo que no distraiga la atención del usuario constantemente de un dispositivo de entrada a otro. Algunos puntos a considerar serían:

Minimización de la memorización: Aquí, se debe considerar que un conjunto de comandos simplificados es más fácil de aprender y recordar, por lo que el usuario puede concentrarse en la aplicación en vez del lenguaje, deben evitarse formatos oscuros, complicados, inconsistentes y abreviados.

El lenguaje de comando puede diseñarse en varios niveles, los cuales podrían ser para el primer nivel, usuarios principiantes y los demás niveles; usuarios avanzados, considerando que durante la práctica los usuarios principiantes podrán llegar a utilizar los siguientes niveles ampliando el lenguaje de comando un poco cada vez.

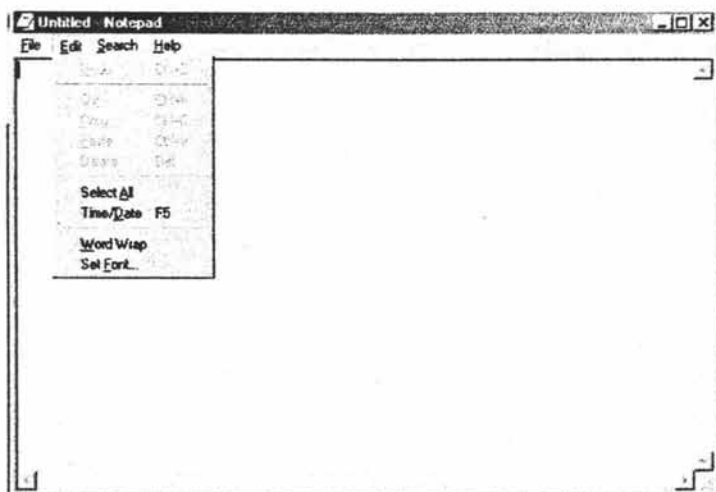
Facilidades de ayuda para el usuario: Es muy importante que se incluyan secciones de ayuda pudiendo incluir una sesión de "tutorial" que ofrezca instrucción de cómo se utiliza el sistema, además se puede incluir varias aplicaciones de ejemplo de manera que el usuario pueda apreciar la manera en que las operaciones trabajan en realidad con aplicaciones comunes. Las solicitudes de entrada pueden darse en forma de menús o mensajes desplegados, las solicitudes de entrada pueden indicar exactamente qué hacer a continuación.

Respaldo y manejo de errores: Durante cualquier secuencia de operaciones, debe disponerse de algún mecanismo sencillo de respaldo o aborto, también una operación puede cancelarse antes de que se complete una ejecución. Con la capacidad de respaldo, cualquier usuario puede explorar con confianza las capacidades del sistema, sabiendo que pueden eliminarse los efectos de cualquier error. Los respaldos se pueden ofrecer en muchas formas: por ejemplo se pueden hacer respaldos y preguntas como estas: ¿Usted está seguro de que desea hacer esto?, que podría ser como una seguridad ofrecida al usuario durante una ejecución. El mensaje de error debe ofrecer una explicación clara de lo que anda mal y de lo que se necesita hacer para corregir un error. Los errores confusos no pueden ayudar al usuario a corregir un error.

Tiempo de respuesta: Este depende de la complejidad de la tarea solicitada, para muchas solicitudes de entrada de rutinas, el sistema puede responder en forma inmediata, pero cuando es una solicitud de procesamiento complicada, puede demorar, sin embargo; sin importar la complejidad de la solicitud de la entrada, los usuarios pueden esperar que los sistemas den algún tipo de respuesta inmediata; de lo contrario, no podrán estar seguros de que la entrada fue recibida y que el sistema realiza el procesamiento. Debe diseñarse un paquete de gráficas que dé respuesta "instantánea" a la entrada del usuario en cerca de una décima de segundo. Por ejemplo podría mandarse el mensaje "SE ESTÁ LLEVANDO A CABO EL PROCESO", para lo cual el usuario entenderá que el tiempo que está transcurriendo es el de proceso de la información, algunos sistemas están diseñados de manera que la variabilidad en tiempos de respuesta tengan diferentes tipos de procesamiento que no sean demasiado grandes.

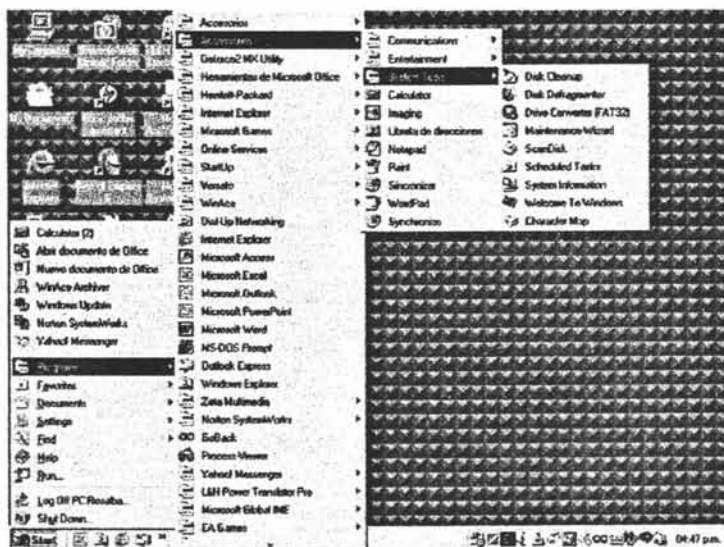
Estilos de lenguaje de comando: Existen varios estilos posibles y la elección depende de varios factores, entre los cuales se incluyen:

- Los objetivos del paquete.
- El tipo de dispositivo de entrada.
- El tipo de usuario.(Deitel,1999)



3.7 Menú de barra con menú desplegable

3. Menús de cascadas. Pueden cambiar dinámicamente y deshabilitar opciones que no están disponibles en un momento dado.



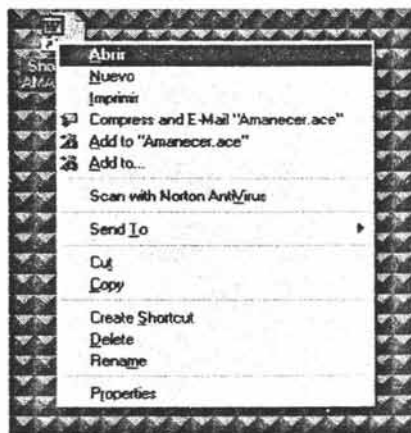
3.8 Menú en cascada de la barra de inicio de Windows 98

4. Menús de barras de herramientas. También llamados paletas, son menús gráficos con acciones, herramientas y opciones que se pueden colocar en la pantalla, se utilizan mucho en programas gráficos.



3.9 Paletas de barras de herramientas de Power Point

5. Los menús contextuales o menús "pop-up" son los más recientes, se llaman así porque el contenido del menú depende del contexto del trabajo del usuario. Contienen únicamente las opciones que son aplicables al objeto seleccionado y son accesibles desde el menú de barra.



3.10 Menú contextual en el escritorio de Windows 98 (Deitel, 1999, Cevallos, 1999)

3.5.4.-Realimentación.

Una parte importante de cualquier sistema de gráficas es la cantidad de realimentación suministrada por el usuario, el sistema necesita realizar un dialogo interactivo e informar al usuario qué está haciendo el sistema en cada etapa.

Conforme cada entrada de usuario es recibida, debe aparecer una respuesta inmediata en la pantalla, el mensaje debe ser breve e indicar con claridad el tipo de procesamiento en progreso, los mensajes de realimentación deben darse con la claridad suficiente para que tenga poca oportunidad de ser pasados por alto. Si no tuviese una realimentación un usuario puede empezar a preguntarse qué está haciendo el sistema y si la entrada debe volver a darse, algunos métodos recomendados son:

Audio: Tiene la ventaja de que no utiliza más espacio de la pantalla y el usuario no tiene que distraer su atención del área de trabajo con el fin de recibir mensajes.

Eco: Los caracteres tecleados pueden desplegarse en la pantalla conforme se introduce de modo que el usuario puede detectar y corregir errores de inmediato, la entrada con botón y cuadrante puede reflejarse en la misma forma.

Símbolos especiales: Estos se pueden hacer para diferentes tipos de realimentación, por ejemplo, un símbolo de una cruz, puede usarse para indicar un error, un signo centelleante "en trabajo" podría servir para indicar al usuario que el sistema está procesando una entrada.

Realce, cambio de color y centelleo del elemento seleccionado: Es otro método que se podría utilizar para la realimentación de la selección de menú.

Despliegue rastreador: Las intensidades de los pixeles se invierten con facilidad, de modo que este método puede emplearse para ofrecer una rápida realimentación para selecciones de menú.

En algunos casos puede ser útil colocar mensajes de realimentación en el área de trabajo del usuario a la proximidad del cursor, pueden usarse diferentes colores para distinguir la realimentación de otros objetos del despliegue. (Deitel,1999)

3.5.5.-Formatos de salida.

La información que se presenta al usuario de un paquete de gráficas incluye una combinación de imágenes, menús, mensajes de salida y otras formas de diálogo generadas por el sistema, existen muchas posibilidades para presentar la información de salida de usuario, el diseñador de un paquete de gráficas debe considerar la mejor manera de diseñar los formatos de salida para lograr la mayor efectividad visual.

Las consideraciones en el diseño de formatos de salida incluyen:

Estructura de menús y mensajes: Se verá influenciada por el proyecto elegido para la salida de la pantalla.

Formas de iconos y símbolos: Estos se escogen de modo que ofrezcan una imagen todavía más clara y simple del objeto u operación que se supone debe representar, otros símbolos como los cursores o apuntadores de menú deben diseñarse para que sean claramente diferentes de los otros iconos. En algunos casos un paquete puede permitir al usuario especificar la forma de algunos símbolos.

Proyecto de la pantalla: Tres componentes básicos del proyecto de pantalla son:

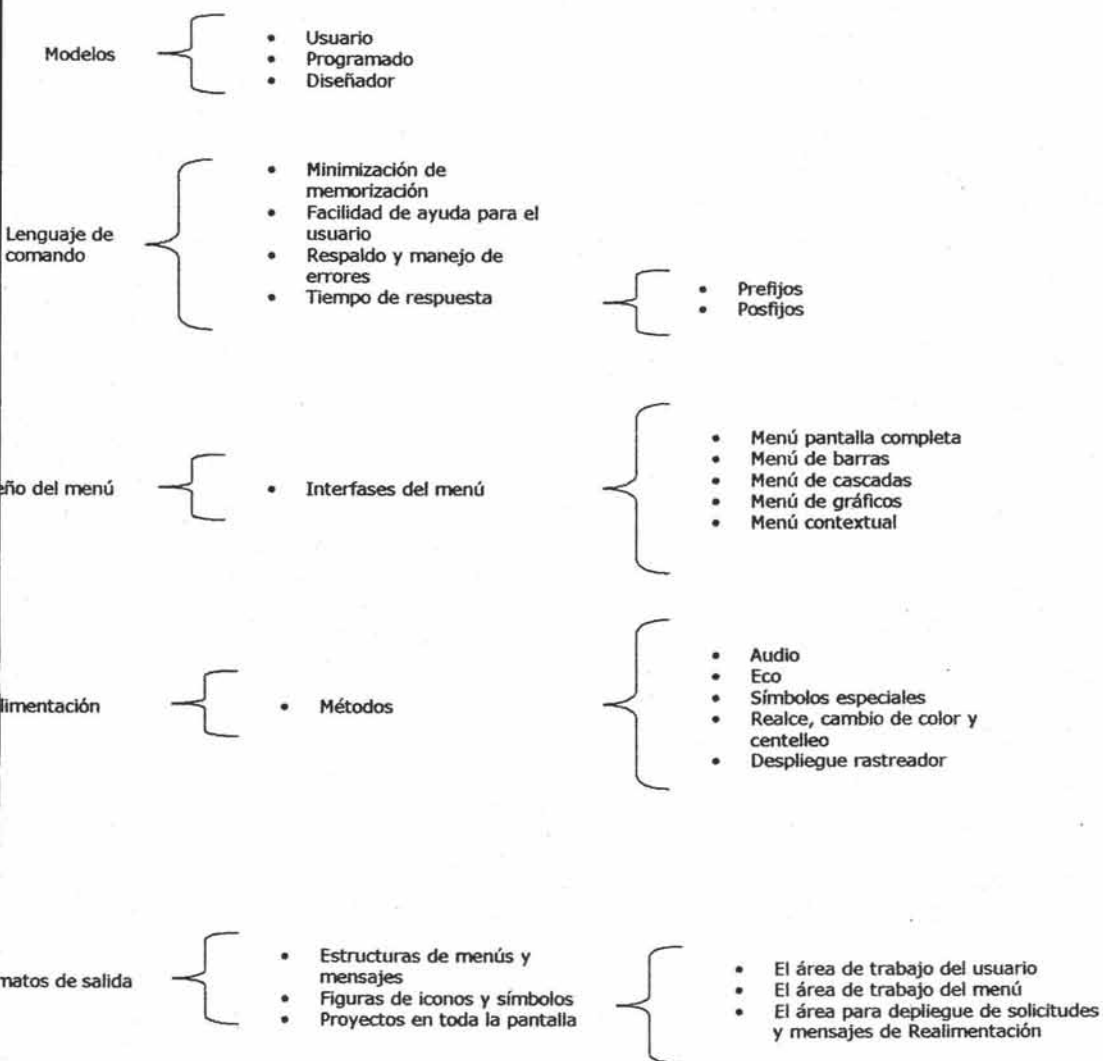
- El área de trabajo del usuario.
- El área de trabajo del menú.
- El área para desplegar solicitudes de entrada y mensajes de realimentación.

La explicación para trabajar con estas tres áreas se dice que en conjunto los proyectos de la pantalla escogidos, deben evitar un aspecto amontonado, los menús y las otras áreas deben conservarse simples y fáciles de entender y al usuario se le deben presentar patrones conocidos en forma consistente.

En conclusión, se puede mencionar que al realizar el diseño de una interfase gráfica debemos considerar:

- Uso extensivo de elementos de control visual estándar como los iconos y las barras de desplazamiento.
- Manipulación directa de los elementos que se localizan en la pantalla.
- Consistencia a través de aplicaciones y plataformas.
- Un sistema atractivo y fácil de usar. (Deitel,1999)

3.6.-Estructura de los fundamentos del diseño.



3.11 Esquema de estructura de los fundamentos del diseño

(Deitel,1999)

3.7.-Principios para el diseño de interfases de usuario.

Existe una serie de principios a seguir en el desarrollo de interfases de usuario y son las siguientes:

Dar un control al usuario: Se enumeran los siguientes principios que permiten al usuario estar en posesión del control:

1. Usar adecuadamente los modos de trabajo.
2. Permitir a los usuarios utilizar el teclado o el ratón.
3. Permitir al usuario interrumpir su tarea y continuarla más tarde.
4. Utilizar mensajes y textos descriptivos.
5. Permitir deshacer las acciones e informar de su resultado.
6. Permitir una cómoda navegación dentro del producto y una fácil salida del mismo.
7. Permitir distintos niveles de uso del producto para usuarios con distintos niveles de experiencia.
8. Hacer transparente la interfase de usuario, que debe tener la impresión de manipular directamente los objetos con los que ésta trabajando.
9. Permitir al usuario manipular directamente los objetos de la interfase.

Reducir la carga de memoria del usuario: La interfase debe evitar que el usuario tenga que almacenar y recordar información. Para ello debe seguir los siguientes principios:

1. Aliviar la carga de la memoria de corto alcance o sea que se permita deshacer, copiar y pegar.
2. Basarse en el reconocimiento antes que en el recuerdo por ejemplo: elegir de entre una lista en lugar de teclear de nuevo.
3. Proporcionar indicaciones visuales de dónde está el usuario, qué está haciendo y qué puede hacer.
4. Proporcionar funciones deshacer, rehacer y acciones por defecto.
5. Proporcionar atajos de teclado como inicios en menús y teclas rápidas.
6. Asociar acciones a los objetos como en un menú contextual.
7. Utilizar metáforas del mundo real.
8. Presentar al usuario sólo la información que necesita como menús simples, avanzados, wizards (auxiliares) y asistentes.
9. Hacer clara la presentación visual.

Consistencia: Permite al usuario utilizar el conocimiento adquirido en otros programas consistentes con el nuevo programa por ejemplo: mostrar siempre el mismo mensaje ante un mismo tipo de situación, aunque se produzca en distintos lugares.

1. Consistencia en la realización de las tareas: proporcionar al usuario indicaciones sobre el proceso que está siguiendo.

2. Consistencia dentro del mismo producto y de un producto a otro. La consistencia se aplica a la presentación, el comportamiento y la interacción.
3. Consistencia de la apariencia estética iconos, fuentes, colores, distribución de pantallas.
4. Fomentar la libre exploración de la interfase, sin miedo a consecuencias negativas.

Guías de diseño: Los estándares definen las características de los objetos y sistemas que se usan cada día, por ejemplo; disposición de las teclas en un teléfono.

Las guías de diseño de interfases afectan a la presentación, el comportamiento y la interacción de los elementos de la interfase, y son reglas e indicaciones a seguir en cuanto a la apariencia y comportamiento de éstos, las guías de diseño abarcan tres áreas del diseño de la interfase:

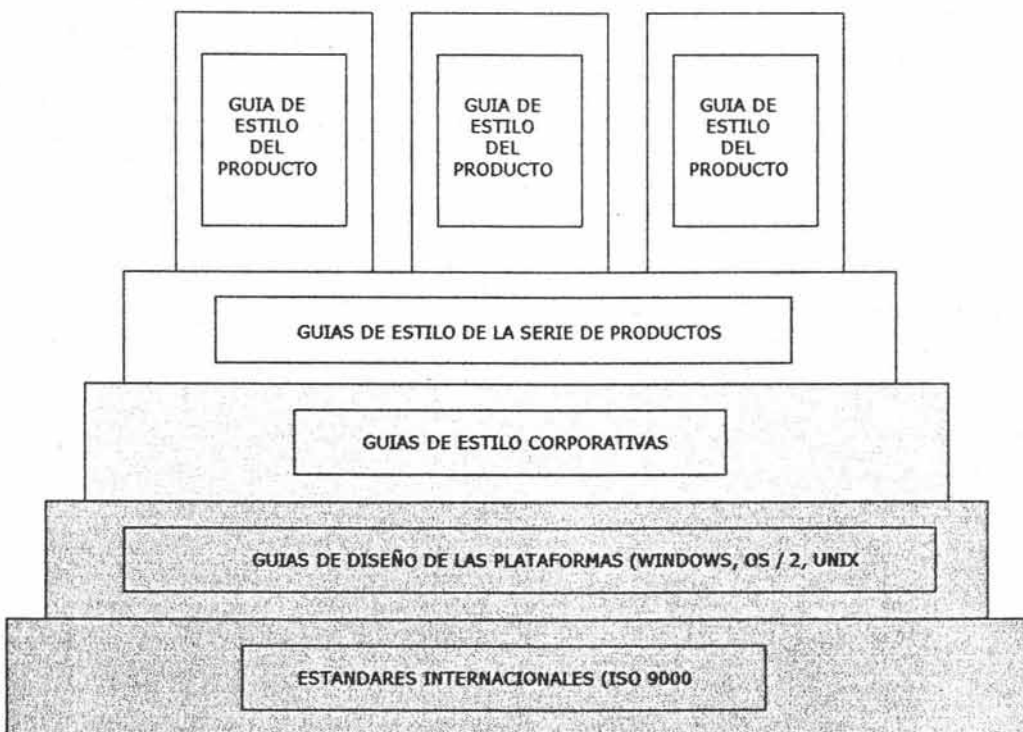
- Física: Es el hardware de la interfase; que puede ser el efecto de los botones de un ratón.
- Sintáctica: Es la presentación de la información, secuencia y orden de las acciones del usuario para realizar una tarea, como por ejemplo imprimir.
- Semántica: Es el significado de los objetos y acciones, como por ejemplo el de las palabras Exit y Cancel.

Un aspecto que deben cuidar las guías de diseño es el soporte para diversos lenguajes, con vistas al uso del producto a nivel internacional. Cada entorno de desarrollo tiene sus guías de diseño:

ENTORNO	GUIAS DE DISEÑO
Apple	Macintosh
IBM	OS / 2
Microsoft	Windows
UNIX	OSF / Motif

3.12 Tabla de entorno con guías de diseño

Las guías de diseño se aplican a una plataforma. Además, cada Empresa suele tener sus propias guías de estilo, que se aplican a todos sus productos con el fin de darles un aspecto uniforme y mantener una imagen corporativa, la Figura 3.13 muestra la denominada pirámide de guías, donde se observa cómo unas guías se construyen sobre las otras.



3.13 Pirámide de Guías

Usabilidad: La usabilidad es una de las partes más importantes en un sistema, tiene cinco atributos, los cuales son:

- **Aprender:** el sistema debe ser fácil de aprender.
- **Eficiencia:** el sistema debe ser eficiente en su uso.
- **Memorizar:** el sistema debe ser fácil de recordar.
- **Errores:** el sistema debe tener un bajo porcentaje de error.
- **Satisfacción:** el sistema debe ser agradable para los usuarios que lo utilicen.

En cualquier caso, aunque útiles, las guías de diseño son recomendaciones, y lo verdaderamente importante es que el uso del programa sea adecuada, aunque ello suponga desviarse de las guías en un momento dado. (Eran, Baker 1999)

3.8.-Programación orientada a objetos.

La ingeniería de software comprende las técnicas de desarrollo formal para el diseño de software. Problemas frecuentes del software tales como: aumento de la complejidad, cambios continuos, no confiable, dificultad para verificarlo, dificultad para especificar los requerimientos, éstas son algunas de las razones por lo cual la programación orientada a objetos sea tan popular.

La orientación a objetos promete mejoras de amplio alcance en la forma de diseño, desarrollo y mantenimiento del software, ofreciendo una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo en el desarrollo del software, la falta de portabilidad del código y reusabilidad, código que es difícil de modificar, ciclos de desarrollo largos y técnicas de codificación no intuitivas. Tiene tres características básicas: debe estar basado en objetos, basado en clases y capaz de tener herencias de clases. Muchos lenguajes cumplen uno o dos de estos puntos. (Deitel,1999, Reisdorph 2000)

3.8.1.-El objeto.

El elemento fundamental de la POO es, como su nombre lo indica, el objeto. Podemos definir un objeto como un conjunto complejo de Datos y programas que poseen estructura y forman parte de una organización. Esta definición especifica varias propiedades importantes de los objetos. En primer lugar, un objeto no es un dato simple, sino que contiene en su interior cierto número de componentes bien estructurados. En segundo lugar, cada objeto no es un ente aislado, sino que forma parte de una organización jerárquica o de otro tipo. (Deitel,1999, Reisdorph 2000)

3.8.1.1.-Estructura de un objeto.

Un objeto puede considerarse como una especie de cápsula dividida en tres partes:

1. Relaciones: Las relaciones permiten que el objeto se inserte en la organización y están formadas esencialmente por punteros a otros objetos.
2. Propiedades: Las propiedades distinguen un objeto determinado de los restantes que forman parte de la misma organización y tiene valores que dependen de la propiedad de que se trate. Las propiedades de un objeto pueden ser heredadas a sus descendientes en la organización.
3. Métodos: Los métodos son las operaciones que pueden realizarse sobre el objeto, que normalmente estarán incorporados en forma de programas (código) que el objeto es capaz de ejecutar y que también pone a disposición de sus descendientes a través de la herencia.

Como se ha visto, cada objeto es una estructura compleja en cuyo interior hay Datos y programas, todos ellos relacionados entre sí, como si estuvieran encerrados conjuntamente en una cápsula. Esta propiedad (encapsulamiento), es una de las características fundamentales de la POO. (Deitel,1999, Reisdorph 2000)

3.8.1.2.-Organización de los objetos.

En principio, los objetos forman siempre una organización jerárquica, en el sentido de que ciertos objetos son superiores a otros, de cierto modo existen varios tipos de jerarquías: serán simples cuando su estructura pueda ser representada por medio de un "árbol". En cualquier caso, sea la estructura simple o compleja, podrán distinguirse en ella tres niveles de objetos:

- La raíz de la jerarquía: Se trata de un objeto único y especial. Éste se caracteriza por estar en el nivel más alto de la estructura y suele recibir un nombre muy genérico, que indica su categoría especial, como por ejemplo: objeto madre, Raíz o Entidad.
- Los objetos intermedios: Son aquellos que descienden directamente de la raíz y que a su vez tienen descendientes. Representan conjuntos o clases de objetos, que pueden ser muy generales o muy especializados, según la aplicación. Normalmente, reciben nombres genéricos que denotan al conjunto de objetos que representan, por ejemplo, ventana, cuenta, fichero. En un conjunto reciben el nombre de clases o tipos si descienden de otra clase o subclase.
- Los objetos terminales: Son todos aquellos que descienden de una clase o subclase y no tienen descendientes. Suelen llamarse casos particulares, instancias o ítems porque representan los elementos del conjunto representado por la clase o subclase a la que pertenecen. (Deitel,1999, Reisdorph 2000)

3.8.1.3.-Relaciones entre objetos.

Las relaciones entre objetos son, precisamente, los enlaces que permiten a un objeto relacionarse con aquellos que forman parte de la misma organización. Hay de dos tipos fundamentales:

1.-Relaciones jerárquicas: son esenciales para la existencia misma de la aplicación porque la construyen. Son bidireccionales, es decir, un objeto es padre de otro cuando el primer objeto se encuentra situado inmediatamente encima del segundo en la organización de la que ambos forman parte; asimismo, si un objeto es padre de otro, el segundo es hijo del primero.

Una organización jerárquica simple puede definirse como aquella en la que un objeto puede tener un solo padre, mientras que una organización jerárquica compleja, un hijo puede tener varios padres.

2.-Relaciones semánticas: se refieren a las relaciones que no tienen nada que ver con la organización de la que forman parte los objetos que las establecen. Sus propiedades y consecuencia solo dependen de los objetos en sí mismos y no de su posición en la organización. (Deitel,1999, Reisdorph 2000)

3.8.1.4.-Propiedades de los objetos.

Todo objeto puede tener cierto número de propiedades, cada una de las cuales tendrá a su vez, uno o varios valores. En la POO, las propiedades corresponden a las clásicas "variables" de la programación estructurada. Son, por lo tanto, Datos encapsulados dentro del objeto, junto con los métodos (programas) y las relaciones (punteros a otros objetos).

Las propiedades de un objeto pueden tener un valor único o pueden contener un conjunto de valores más o menos estructurados (matrices, vectores, listas, etcétera), además los valores pueden ser de cualquier tipo (numérico, alfabético, etcétera) si el sistema de programación lo permite. Pero existe una diferencia con las "variables", y es que las propiedades se pueden heredar de unos objetos a otros. En consecuencia, un objeto puede tener una propiedad de maneras diferentes:

- Propiedades propias: Están formadas dentro de la cápsula del objeto.
- Propiedades heredadas: Están definidas en un objeto diferente, antepasado de éste (padre, abuelo, bisabuelo, tatarabuelo). A veces estas propiedades se llaman propiedades miembro porque el objeto las posee por el mero hecho de ser miembro de una clase.
- Procedimental: Escrito en cualquier lenguaje, que está asociado a un objeto determinado y cuya ejecución, sólo puede desencadenarse a través de un mensaje recibido por éste o por sus descendientes. (Deitel,1999, Reisdorph 2000)

3.9.-Características de la POO.

Polimorfismo: Una de las características fundamentales de la POO es el polimorfismo, que no es otra cosa que la posibilidad de construir varios métodos con el mismo nombre, pero con relación a la clase a la que pertenece cada uno, con comportamientos diferentes. Esto conlleva la habilidad de enviar un mismo mensaje a objetos de clases diferentes. Estos objetos recibirán el mismo mensaje global pero responderían a él de formas diferentes; por ejemplo, un mensaje " + " a un objeto ENTERO significaría suma, mientras que para un objeto STRING, significaría concatenación.

Dominios: Es un tipo especial de métodos, relativamente poco frecuentes en los sistemas de POO, que se activa automáticamente cuando sucede algo especial.

Es decir, es un programa como los métodos ordinarios, pero se diferencia de éstos porque su ejecución no se activa con un mensaje, sino que se desencadena automáticamente cuando ocurre un suceso determinado; la asignación de un valor a una propiedad de un objeto, la lectura de un valor determinado y demás. Los dominios, cuando existen, se diferencian de otros métodos porque no son heredables y porque a veces están ligados a una de las propiedades de un objeto, más que al objeto entero. (Deitel,1999, Reisdorph 2000)

3.10.-Aplicaciones de software.

Tanto los usuarios, como los especialistas, encontraban difícil trabajar con los primeros sistemas, algunas personas aún recuerdan los años de las tarjetas perforadas, años cincuenta, sesenta y principios de los setenta. En aquella época, la mayor parte de las personas no calificadas se veían forzadas a elegir entre dos opciones poco satisfactorias: usar solamente unos pocos programas de ordenador y depender de especialistas que les ayudarán a acceder a los Datos que necesitaban.

Los programadores se consideraban afortunados si podían probar un programa tres o cuatro veces al día, ya que les eran difíciles de escribir y depurar. Después de todo esto, el software fue evolucionando notablemente, pero no tan rápido como el hardware, por lo que al tener mejores recursos de software se llevan a cabo programas cada vez con más precisión y rapidez, lo que origina que se pueda escoger el software más útil para la aplicación que se requiera. (Deitel,1999, Reisdorph 2000)

3.10.1.-Historia de los sistemas operativos.

La historia del sistema operativo comenzó con el DOS, que probó ser exitoso pero no amigable para los novatos, si la persona desconocía los comandos, estaba desamparada. En 1984, Apple cambió la faz del mundo computarizado para siempre con la introducción de la Macintosh. La Mac fue la primer interfase gráfica de usuario que hizo que la computación personal fuera más amigable con el usuario y por lo tanto, más popular. Fueron apareciendo más sistemas operativos los cuales son:

1.-Windows

En 1985 se introdujo Windows 1.0, aunque empleaba un ratón e íconos. Microsoft lanzó al mercado más versiones de Windows, las más notables fueron Windows 3.1 en 1992 y Windows 95 en 1995.

En la actualidad Windows esta disponible en varias formas: Windows 98, para uso personal y para pequeños negocios así como el Windows 2000, Windows NT para el uso en grupos de ordenadores en red y Windows CE para ordenadores portátiles.

2.-MacOS

Sigue siendo un estándar de comparación para todas las versiones de Windows, la línea Mac sostenía desde hace muchos años que los MacOS, actualmente en su versión 8.6 era muy superior a Windows en simplicidad. Muchos creen que MacOS es superior a Win98, pero deben considerarse ciertos hechos: en primer término que las ordenadores Mac apenas constituyen el 4 % del mercado de las ordenadores personales; y en segundo lugar, que los desarrolladores de software y accesorios producen casi sin excepción productos compatibles con los formatos más vendidos, lo que hace que en ocasiones sea difícil encontrar el software y los periféricos para la MAC.

3.-Os/2

La IBM decide contratar una compañía externa para el desarrollo de un sistema operativo. Las ventas de los ordenadores IBM fueron buenas. Las primeras versiones de OS/2 era una mezcla de los sistemas operativos basados con texto como DOS y de GUIs de Windows, corre en ordenadores con apenas cuatro megabytes de RAM, apenas la mitad de lo que se requiere para correr Win95 y soporta una gran cantidad de tareas, incluyendo al acceso a Internet.

4.-BeOS

Be Inc, ofrece otra alternativa llamada BeOS. La nueva versión 4.0, es una GUI desarrollada teniendo a Intel, el líder de los mercados de procesadores. Aunque BeOS se distingue a primera vista y con facilidad de Win95 o Win98, no parecerá completamente extraño a los usuarios de Windows. El BeOS utiliza un Tracbar en la esquina superior derecha de la pantalla para informar a los usuarios cuales son las aplicaciones que están activas, para navegar por los directorios del ordenador y para realizar búsquedas (se podría estar hablando de una barra de tareas de Win95), la apariencia se asemeja con Windows. BeOS pretende modificar la estructura del sistema operativo, pero no la manera en que se usa. También usa "soft links, que son iconos que pueden ser colocados en la pantalla para un acceso rápido a los programas o archivos que se utilizan con mayor frecuencia, utiliza ventanas para desplegar pantalla a las aplicaciones actuales llamadas ventanas.

El BeOS pretende ser un sistema operativo nuevo construido a Base de hardware y las aplicaciones actuales, como multimedia, sin embargo basar un nuevo sistema operativo en la arquitectura de un sistema operativo viejo, limita severamente el desarrollo de tecnologías de acuerdo al equipo BeOS.

Este es ideal para usuarios que necesitan de la máxima funcionalidad para crear y usar aplicaciones de audio y video.

5.-UNIX

Es un sistema operativo multiusuario y multitarea, preparado para operar ordenadores aislados como integrados a la red, éste requiere de una administración centralizada para tareas como: registrar nuevos usuarios, asignar cuotas de recursos a los diferentes usuarios y definir dispositivos. La primera versión de UNIX disponible fuera de Bell Laboratories fue la versión 6, en el año de 1976. En 1978 se distribuyó la versión 7. Tras esta distribución UNIX se convirtió en un producto y no sólo en una herramienta de investigación sino también educacional. El período comprendido entre 1977 y 1982, Bell Laboratories cambió varios sistemas UNIX desarrollado dentro de AT&T, con características de la versión 7.

UNIX es un sistema operativo que actúa como interfase entre el hardware y los usuarios, no es difícil de utilizar, pero muy conciso. Además de ser un sistema operativo que admite muchas aplicaciones también admite muchos lenguajes de programación. Cada lenguaje tiene su propio conjunto de herramientas para procesar modelos de códigos ASCII y convertidos en programas ejecutables, tales herramientas incluyen procesadores, editores de enlaces (linker) y depuradores. (Sánchez,1998, Ashley, 1991, R PC M Año V No.10)

3.10.2.-Software para elaborar interfases gráficas.

3.10.2.1.-X Windows.

X es el nombre que se la ha dado a los sistemas de ventanas X, permite la implantación de aplicaciones gráficas de alta resolución, una característica importante es que las aplicaciones X son prácticamente transportables a cualquier plataforma de hardware y sistema operativo que soporta un servidor X.

El sistema X Window se desarrolló en el Instituto de Tecnología de Massachusetts (MIT) en 1984 por Robert W. Scheinfler y James Getty, es conocido como "X". Éste comenzó como un proyecto de ventanas en los laboratorios de Ciencias de la Computación del MIT. X Window es para los sistemas UNIX lo que MS Windows es para los sistemas MS-DOS, pero con la diferencia importante, que los sistemas X Window son un estándar aceptado por la industria para los sistemas de ventanas UNIX, esta estandarización tiene consecuencias de gran alcance:

Las interfases gráficas de usuario desarrolladas para un ordenador. Se puede ejecutar en ordenadores completamente diferentes. Los sistemas X Windows proporcionan una salida común para estaciones de trabajo, grandes ordenadores, miniordenadores y microordenadores conectados en red. Una cosa interesante sobre X es que es de dominio público y se puede bajar de la red.

Dentro de X Windows se maneja un Gestor de ventanas conocido también como manejador de ventanas, es el componente que determina la apariencia de las ventanas y proporciona los medios necesarios para que el usuario pueda interactuar con ella. Cada usuario puede determinar el aspecto que tendrán sus ventanas y no verse obligado a soportar una interfase rígida y no modificable, algunos gestores se describen a continuación:

En el gestor twm, podemos definir aspectos como el color de los bordes de las ventanas, anchura de los mismos y el menú de la ventana raíz.

En el gestor de ventanas fvwm, su concepción inicial es aprovechar mejor la memoria, la manipulación es similar a otros gestores, ya que pueden redimensionar, mover, minimizar, maximizar, etc., también proporciona entornos de trabajos virtuales.

Entorno xfce es un conjunto de aplicaciones para el sistema X Window que incluye barra de herramientas, un manejador de ventanas y un manejador de fondos, cuyo objetivo es facilitar el manejo de sistema. (Tutorial X Window www.tutoriales)

3.10.2.2.-Xforms.

Es una herramienta de ayuda para la elaboración de GUI basada sobre Xlib para sistema X Window, es característico por la utilización de objetos como botones, barras, menús, etc., permite un variado y fácil construcción de aplicaciones X, los objetos utilizados en Xforms son los siguientes:

- | | |
|----------------------|-------------|
| ➤ Botones | ➤ Cajas |
| ➤ Barras | ➤ Símbolos |
| ➤ Menús | ➤ Atributos |
| ➤ Archivo de entrada | ➤ Marcos |
| ➤ Etiquetas | ➤ Otras más |

(Tutorial Xforms www.tutoriales)

3.10.2.3.-Lenguaje C++ Builder.

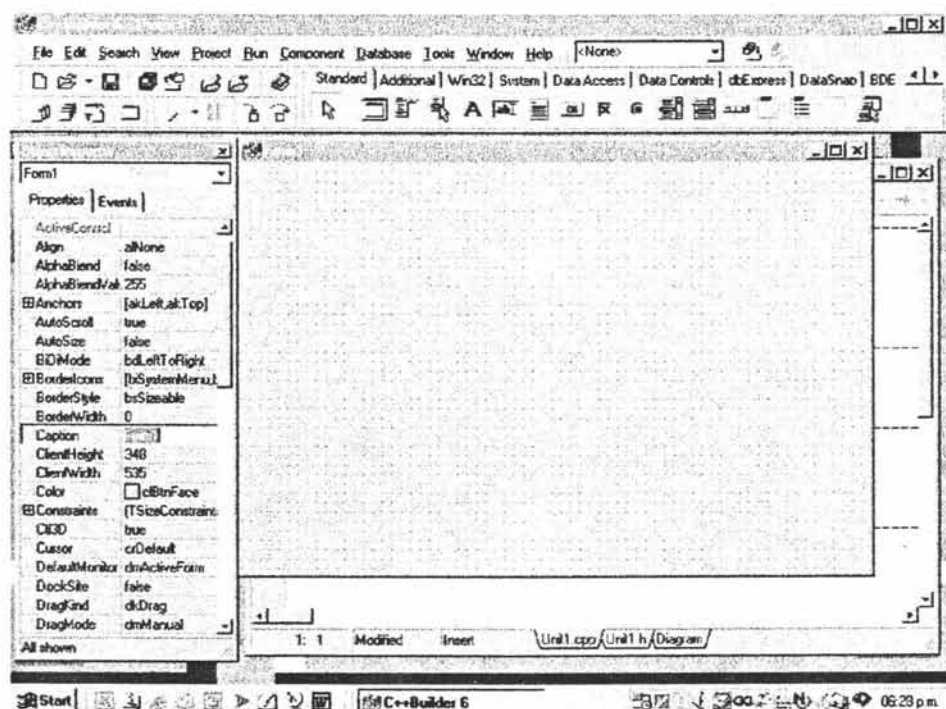
C++ Builder es un producto de RAD (Desarrollo Rápido de Aplicaciones) de Borland, esto quiere decir que es posible escribir programas para Windows de manera rápida y sencilla.

El IDE de C++ Builder consta de los siguientes elementos:

- Menú principal y barra de herramientas
- Component palette
- Form design
- Code editor

- > Object inspector
- > Project manager

C++ Builder tiene un entorno de desarrollo integrado (IDE), cuando inicia el programa, aparece un formulario en blanco como se muestra a continuación:



3.14 Pantalla principal de Visual C++ Builder

Se mencionaron algunas herramientas para la elaboración de interfases, sin mencionar que en el ámbito computacional hay más herramientas, pero se hizo una recolección de las más utilizadas. (CHARTE, 1999)

3.10.3.-Lenguajes de programación utilizados en la elaboración de interfases gráficas.

3.10.3.1.-Lenguaje C++.

Por años, el hardware ha estado mejorando a pasos agigantados, pero el software por alguna razón a parecido resistir a todos los intentos por hacerlo más rápido y mejor. Hoy en día estamos a la mitad de una evolución en la forma de escribir el software.

El lenguaje de programación orientado a objetos más utilizado es C++ desarrollado.

C++ fue una evolución de C, el cual surgió de dos lenguajes de programación previos, BCPL y B. BCPL fue desarrollado en 1967 por Martín Richards como lenguaje para escribir software de sistemas operativos y compiladores. Tanto BCPL como B eran lenguajes "sin tipo" cada elemento de información ocupaba una "palabra" en memoria y la tarea de tratar cada elemento de Datos como número entero o número real. El lenguaje C fue una evolución de B llevada a cabo por Dennis Ritchie en los laboratorios Bell y originalmente se implantó en 1972 en una ordenador DEC PDP-11.

Actualmente, muchos sistemas operativos se escriben con C o C++, durante las últimas dos décadas, C ha quedado disponible en casi todos los ordenadores, este es independiente del hardware con diseños cuidadosos, además de que es posible escribir programas en C que pueden ser portables para utilizarse en la mayoría de los ordenadores.

Hacia finales de los 70, C evolucionó en lo que se conoce como "C tradicional", "C clásico" o "C de Kernighan y Ritchie", la amplia difusión de C a varios tipos de ordenadores por desgracia, tuvo como consecuencia que hubiese demasiadas variantes, ya que eran parecidas pero frecuentemente incompatibles.

A principios de la década de los 80, Bjarne Stroustrup desarrolló C++, una extensión de C en los laboratorios Bell. C++ ofrece una serie de características que "engalanan" al lenguaje C, sin embargo, lo más importante es que proporcionan capacidades para la programación orientada a objetos. (Deitel, 1999)

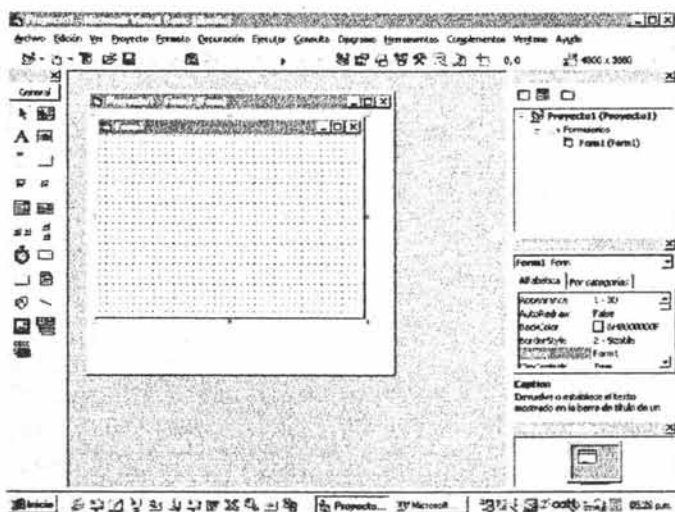
3.10.3.2.-Conociendo a Visual Basic.

La palabra Visual hace referencia al lenguaje Basic, un lenguaje utilizado por más programadores que ningún otro lenguaje utilizado en informática, este lenguaje también es utilizado por Microsoft Excel, Microsoft Access y otras aplicaciones en Windows. Permite crear programas para su uso personal, para su grupo de trabajo, para una empresa, aplicaciones distribuidas a través de Internet, aplicaciones de Bases de Datos y otras más.

Como todos sabemos, Visual Basic es un lenguaje de programación Visual, que si bien no cumple con todos los requisitos de un lenguaje propiamente dicho: encapsulado, herencia y polimorfismo. Nos hace la programación más fácil, ya que la mayoría de sus funciones y procedimientos están automatizados, y sus controles tienen una cierta cantidad de propiedades, fáciles de modificar. Además su entorno de desarrollo totalmente visual, facilita la creación de aplicaciones, sumando a esto la compatibilidad que tiene con Windows.

Hoy en día a cualquier aplicación se le exige un entorno multimedia, que resulte atrayente al alumno o usuario, y con Visual Basic es posible lograr esto con distintos controles que permiten activar: sonidos, videos, animaciones, desplegar imágenes y texto, y conectarse con otras aplicaciones del entorno Windows.

En Visual Basic se tiene manejo de botones, barras, menús, cajas y todas las aplicaciones necesarias para realizar una interfase gráfica. A continuación se muestra la pantalla principal de Visual Basic.



3.15 Pantalla principal de Visual Basic

3.10.3.3.-Como escribir una aplicación en Visual Basic.

Para realizar el software, es necesario emplear un grupo de proyectos y formularios, donde el proyecto es un conjunto de archivos que trabajan en equipo para crear un archivo ejecutable independiente y el formulario es la forma o pantalla donde se desarrolla el diseño.

Al crear un nuevo proyecto genera los siguientes tipos de ficheros:

- Módulos de formularios (.frm)
- Módulos estándar (.bas)
- Módulos de clases (.cls)
- Ficheros binarios (.frx)
- Ficheros de recursos (.res)

3.10.3.4.-Cómo se crea un proyecto.

A continuación se explicará como se crea un proyecto con un menú y submenú explicándolo por pasos:

Primer paso: Crear un proyecto

1. Al entrar a Visual Basic, mostrará un cuadro de dialogo del cual se elige el icono EXE Estándar (Estándar EXE).
2. Seleccionar Archivo / Nuevo Proyecto (New Project) para desplegar el cuadro de dialogo Nuevo Proyecto.
3. Hacer clic en el botón para maximizar la ventana Formulario y ampliar el área de visión a su tamaño máximo (se ampliará el fondo punteado, no el formulario).
4. Arrastrar el controlador de tamaño de la parte inferior derecha del formulario hacia abajo y a la derecha. Mientras se arrastra el formulario, se observará cómo cambian las medidas de anchura y altura a la derecha de la barra de herramientas. El tamaño del formulario se ajustará para que mida aproximadamente 6195 por 4935 twips (un twip es una medida de pantalla. Es equivalente a la vigésima parte de un punto. Un punto es = 1/72 de pulgada, un twip es equivalente a 1/1440 de pulgada)
5. Buscar en la ventana Propiedades aquella que dice StartupPosition, seleccionar la opción 2, CenterScreen de la lista desplegable. Esto hará que el formulario se centre en la pantalla cuando inicie la aplicación
6. Asignar un nombre al formulario (en este caso se llamará Ejemplo), para ello se debe trabajar nuevamente con la ventana Propiedades. Existe una propiedad llamada Name, dar clic en esta y escribir el nombre que llevara precedido de *frm* (*frmEjemplo*).
7. Cambiar el valor original de la barra de título del formulario por Aplicación de Menús. Esto se hace al escribir Aplicación de Menús como valor en la propiedad Caption.
8. Antes de seguir adelante, y por motivos de seguridad, guardar el formulario. Seleccionar Archivo/Guardar (File/SaveProject). Luego Visual Basic pide un nombre de proyecto para el archivo de descripción de proyecto. Responder NO en caso de que Visual Basic pregunte si se agrega el proyecto a la biblioteca SourceSafe.

Segundo paso: Agregar menús

1. En la ventana de formulario, puede oprimirse Ctrl+E que es el método abreviado de Herramientas/Editor de menús (Tools / Menú Editor para mostrar en pantalla el editor de menús.
2. Escribir &Archivo en el campo Caption. El ampersand indica que la A será la letra aceleradora para la opción del menú.

A medida que se escribe el título, Visual Basic lo desplegará en el cuadro de lista de menús, que se encuentra en la mitad inferior del editor de menús (Menú Editor).

3. Oprimir Tab para ir al campo Name.
4. Escribir mnuArchivo como nombre de la primera opción del menú.
5. Dejar en blanco los demás campos y dar clic en el botón Siguiente (Next), para las opciones restantes. El botón siguiente le indica al editor de menús que ha terminado con la primera opción y que desea incluir otra.
6. Escribir &Editar en el campo Caption de la siguiente opción de la barra de menús y escribir mnuEditar como nombre.
7. Hacer clic en siguiente.
8. Escribir &Ayuda en el campo Caption de la tercera y última opción de la barra de menús y escriba mnuAyuda como nombre.

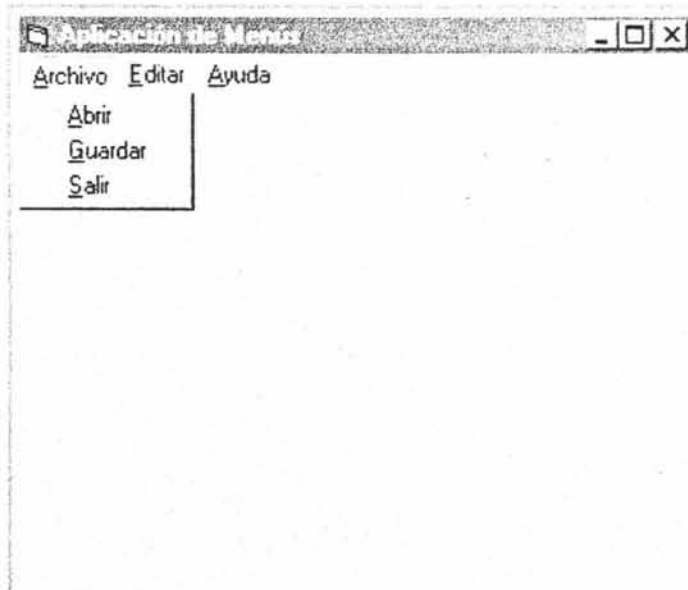
Tercer paso: Agregar submenú

Mientras se construye la barra de menús, pueden crearse menús desplegables, o agregarlos más adelante. El botón Insertar (Insert) es el que permite hacer esto.

NOTA: Antes de hacer clic en Insertar, seleccionar la opción de menú que aparece después de donde se desea insertar una opción.

1. Hacer clic en la opción &Editar del cuadro de lista de control de menús para resaltar.
2. Hacer clic tres veces en el botón Insertar y, luego en la flecha a la derecha, para agregar tres filas vacías para las opciones del menú Archivo.
3. Resaltar la primera fila en blanco, donde se agregará la opción Abrir.
4. Escribir &Abrir en el campo Caption y mnuAbrir en el campo Name. Cuando se ejecute el programa por primera vez, la opción Abrir aparecerá sin marcar.
5. Hacer clic en Siguiente (Next) para pasar a la opción que sigue.
6. Escribir &Guardar en el campo Caption y mnuGuardar como nombre de opción.
7. Hacer clic en Siguiente (Next) para pasar a la opción siguiente.
8. Escribir &Salir en el campo Caption y mnuSalir como nombre de opción.
9. La opción Salir estará marcada cuando el usuario inicie el programa. Por lo tanto, hacer clic en la casilla Checked para colocar la marca de verificación junto a Salir.

Al terminar de realizar lo anterior, seleccionar del menú Ejecutar "Iniciar" y se mostrará lo siguiente:



3.16 Ejecución del proyecto Aplicación de menús

3.10.3.5.-Bibliotecas de vínculos dinámicos.

Se les conoce como DLLs y son muy útiles, éstas constan de uno o más fragmentos de código almacenado en un archivo con extensión .dll, este archivo se puede llamar desde programas ejecutables, para la realización del programa fue necesario utilizarla, las DLLs ofrecen los siguientes beneficios:

- Reutilización efectiva del código
- La posibilidad de compartir fácilmente código entre muchas aplicaciones
- La posibilidad de seleccionar código
- Internacionalización de la aplicación
- Uso efectivo de los recursos de Windows

Lo que significa que al tener una DLLs, se podrán crear formularios a los que se puedan llamar desde casi cualquier aplicación Windows, sin importar en que este escrita como Visual Basic, FORTRAN, Delphi, etc.

El uso de recursos en DLLs es eficaz si se tienen muchos en una aplicación y se pretende controlar cuándo y dónde se cargan dichos recursos.

3.10.3.6.-Términos importantes utilizados.

- **Componente:** Es una parte de software binario auto contenida que ejecuta alguna función predefinida, como una etiqueta de texto, un control de edición o un cuadro de lista.
- **Propiedad:** Es la que determina la operación y características de un componente.
- **Evento:** Es lo que ocurre como resultado de la interacción de un componente con el usuario o con Windows.
- **Manejador de eventos:** Es un método que se activa en la aplicación como respuesta a un evento.
- **Clases:** Al igual que una estructura, es un conjunto de Datos y funciones miembro que realizan una tarea de programación específica.
- **Conjunto:** Serie de objetos similares.
- **Unidad:** Este término se emplea para hacer referencia a los archivos de código fuente.

En conclusión, se puede decir que Visual Basic es una herramienta de gran utilidad para la creación de interfases gráficas y su aplicación será de gran ayuda en la elaboración de este trabajo. (Ceballos,1999, Smart Cad,2001)

3.10.4.-Lenguaje Fortran.

Fortran fue el primer lenguaje de programación de ordenadores, utilizado por la comunidad científica, en 1954, un grupo de trabajo encabezado por John Backus genera un lenguaje de programación flexible, del que su nombre significa "Traducción de fórmulas", pero no fue hasta 1957 que es liberado el primer compilador Fortran. Al igual que todos los lenguajes este fue combinado codificándose hasta llegar al Fortran 90 en 1991. Este lenguaje fue adoptado como estándar a nivel mundial y como estándar ANSI en Estados Unidos en 1992. (Tutorial Fortran, www.tutoriales)

3.11.-Introducción a las Bases de Datos.

Es una colección de archivos interrelacionados, son creados con un DBMS. El contenido de una Base de Datos engloba a la información concerniente (almacenadas en archivos) de una organización, de tal manera que los Datos estén disponibles para los usuarios. Una finalidad de la Base de Datos es eliminar la redundancia o al menos minimizarla. Los tres componentes principales de un sistema de Base de Datos son el hardware, el software DBMS y los Datos a manejar, así como el personal encargado del manejo del sistema.

3.12.-Sistema Manejador de Base de Datos. (DBMS)

Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de una tarea específica.

El objetivo primordial de un sistema manejador Base de Datos es proporcionar un contorno que sea a la vez conveniente y eficiente para ser utilizado al extraer, almacenar y manipular información de la Base de Datos.

Todas las peticiones de acceso a la Base, se manejan centralizadamente por medio del DBMS, por lo que este paquete funciona como interfase entre los usuarios y la Base de Datos.

Esquema de Base de Datos.

Es la estructura por la que esta formada la Base de Datos, se especifica por medio de un conjunto de definiciones que se expresa mediante un lenguaje especial llamado lenguaje de definición de Datos. (DDL)

3.13.-Administrador de Base de Datos. (DBA)

Denominado por sus siglas como: DBA, DataBase Administrator. Es la persona encargada y que tiene el control total sobre el sistema de Base de Datos. Es la persona o equipo de personas profesionales responsables del control y manejo del sistema de Base de Datos, generalmente tiene (n) experiencia en DBMS, diseño de Bases de Datos, Sistemas operativos, comunicación de Datos, hardware y programación, sus funciones principales son:

➤ Definición de esquema.

Es el esquema original de la Base de Datos se crea escribiendo un conjunto de definiciones que son traducidas por el compilador de DDL a un conjunto de tablas que son almacenadas permanentemente en el diccionario de Datos.

➤ Definición de la estructura de almacenamiento del método de acceso.

Estructuras de almacenamiento y de acceso adecuados se crean escribiendo un conjunto de definiciones que son traducidas por el compilador del lenguaje de almacenamiento y definición de Datos.

➤ Concesión de autorización para el acceso a los Datos.

Permite al administrador de la Base de Datos regular las partes de las Bases de Datos que van a ser accedidas por varios usuarios.

➤ Especificación de limitantes de integridad.

Es una serie de restricciones que se encuentran almacenados en una estructura especial del sistema que es consultada por el gestor de Base de Datos cada vez que se realice una actualización al sistema.

Los sistemas de Base de Datos se diseñan para manejar grandes cantidades de información, la manipulación de los Datos involucra tanto la definición de estructuras para el almacenamiento de la información como la provisión de mecanismos para la manipulación de la información, además un sistema de Base de Datos debe de tener implementados mecanismos de seguridad que garanticen la integridad de la información, a pesar de caídas del sistema o intentos de accesos no autorizados.

Un objetivo principal de un sistema de Base de Datos es proporcionar a los usuarios finales una visión abstracta de los Datos, esto se logra escondiendo ciertos detalles de como se almacenan y mantienen los Datos.

3.14.- Objetivos de los sistemas de Bases de Datos.

Los objetivos principales de un sistema de Base de Datos es disminuir los siguientes aspectos:

- Redundancia e inconsistencia de Datos.

Puesto que los archivos que mantienen almacenada la información son creados por diferentes tipos de programas de aplicación existe la posibilidad de que si no se controla detalladamente el almacenamiento, se pueda originar un duplicado de información, es decir que la misma información sea más de una vez en un dispositivo de almacenamiento. Esto aumenta los costos de almacenamiento y acceso a los Datos, además de que puede originar la inconsistencia de los Datos - es decir diversas copias de un mismo dato no concuerdan entre si -, por ejemplo: que se actualiza la dirección de un cliente en un archivo y que en otros archivos permanezca la anterior.

- Dificultad para tener acceso a los Datos.

Un sistema de Base de Datos debe contemplar un entorno de Datos que le facilite al usuario el manejo de los mismos. Supóngase un banco, y que uno de los gerentes necesita averiguar los nombres de todos los clientes que viven dentro del código postal 78733 de la ciudad. El gerente pide al departamento de procesamiento de Datos que genere la lista correspondiente. Puesto que esta situación no fue prevista en el diseño del sistema, no existe ninguna aplicación de consulta que permita este tipo de solicitud, esto ocasiona una deficiencia del sistema.

➤ Aislamiento de los Datos.

Puesto que los Datos están repartidos en varios archivos, y estos no pueden tener diferentes formatos, es difícil escribir nuevos programas de aplicación para obtener los Datos apropiados.

➤ Anomalías del acceso concurrente.

Para mejorar el funcionamiento global del sistema y obtener un tiempo de respuesta más rápido, muchos sistemas permiten que múltiples usuarios actualicen los Datos simultáneamente. En un entorno así la interacción de actualizaciones concurrentes puede dar por resultado Datos inconsistentes. Para prevenir esta posibilidad debe mantenerse alguna forma de supervisión en el sistema.

➤ Problemas de seguridad.

La información de toda empresa es importante, aunque unos Datos lo son más que otros, por tal motivo se debe considerar el control de acceso a los mismos, no todos los usuarios pueden visualizar alguna información, por tal motivo para que un sistema de Base de Datos sea confiable debe mantener un grado de seguridad que garantice la autenticación y protección de los Datos. En un banco por ejemplo, el personal de nóminas sólo necesita ver la parte de la Base de Datos que tiene información acerca de los distintos empleados del banco y no a otro tipo de información.

➤ Problemas de integridad.

Los valores de Datos almacenados en la Base de Datos deben satisfacer cierto tipo de restricciones de consistencia. Estas restricciones se hacen cumplir en el sistema añadiendo códigos apropiados en los diversos programas de aplicación.

3.15.- Base de Datos Orientadas a Objetos.

Las aplicaciones de las Bases de Datos en áreas como el diseño asistido por computadora, la ingeniería de software y el procesamiento de documentos no se ajustan al conjunto de suposiciones que se hacen para aplicaciones del estilo de procesamiento de Datos. El modelo de Datos orientado a objetos se ha propuesto para tratar algunos de estos nuevos tipos de aplicaciones.

El modelo de Bases de Datos orientado a objetos es una adaptación a los sistemas de Bases de Datos. Se basa en el concepto de encapsulamiento de Datos y código que opera sobre estos en un objeto. Los objetos estructurados se agrupan en clases. El conjunto de clases esta estructurado en sub y superclases basado en una extensión del concepto ISA del modelo Entidad - Relación. Puesto que el valor de un dato en un objeto también es un objeto, es posible representar el contenido del objeto dando como resultado un objeto compuesto.

El propósito de los sistemas de Bases de Datos es la gestión de grandes cantidades de información. Las primeras Bases de Datos surgieron del desarrollo de los sistemas de gestión de archivos. Estos sistemas primero evolucionaron en Bases de Datos de red o en Bases de Datos jerárquicas y, más tarde, en Bases de Datos relacionales.

3.16.-Lenguaje de manipulación de datos.

La manipulación de Datos se refiere a las operaciones de insertar, recuperar, eliminar ó modificar Datos; dichas operaciones son realizadas a través del lenguaje de manipulación de Datos (DML, Data Manipulation Language) que es a quien permite el acceso de los usuarios a los Datos.

Existen básicamente 2 tipos de lenguajes de manipulación de Datos:

➤ Procedimentales:

Los LMD requieren que el usuario especifique que Datos se necesitan y cómo obtenerlos.

➤ No Procedimentales:

Los LMD requieren que el usuario especifique que Datos se necesitan y sin especificar como obtenerlos.

3.17.- Lenguajes de Programación para Bases de Datos. (OO)

La principal diferencia que existe entre los lenguajes de programación orientados a objetos y los lenguajes de un sistema manejador de Bases de Datos es que:

- Los lenguajes utilizan el mismo modelo para la manipulación completa de las Bases de Datos: MANIPULACION REGISTRO A REGISTRO (PROCEDURAL).
 - Los Sistemas Manejadores de Base de Datos utilizan MEZCLADOS(INMERSOS) los dos paradigmas: PROCEDURAL y DECLARATIVO. Adicionalmente deben soportar un mecanismo para acoplar las diferencias de impedancia entre los dos paradigmas.
 - En este sentido existen dos tendencias importantes para soportar BDOO:
 - Extender a los lenguajes de programación orientados a objetos (como el C++) con aspectos de Base de Datos como: PERSISTENCIA.
 - Extender a los manejadores de Bases de Datos comerciales(que soportan SQL) con la facilidad del manejo de clases, herencia, polimorfismo, etc.
- (Kroenke,1996 – Tsai,1990 – Marlyn, Hartley, 1991 – Korth, Silberschatz, 1991)

3.18.-Herramientas para elaborar Bases de Datos.

A continuación se dará una breve reseña de algunas de las herramientas para elaborar Bases de Datos, con lo que también se recorrerá la historia y evolución de estas, hasta llegar a algunas de las más utilizadas en nuestros días.

- dBASE.

Cuando aparecieron los primeros microordenadores, Wayne Ratliff, un diseñador de software del JPL, impresionado por la capacidad y las posibilidades del gestor de Base de Datos del ordenador central, se puso a redactar un sistema de Base de Datos para su micro, usando el sistema del JPL como modelo. Una vez terminado, Ratliff decidió comercializar su gestor de Base de Datos y le puso el nombre de Vulcan (el nombre fue en honor del planeta natal de Mr. Spock, personaje de Star Trek. El Vulcan constituyó un atisbo del dBASE III Plus, o, por tanto, del dBASE II. Le faltaban algunas de las potentes órdenes de indexación y ordenación presentes en el dBASE II y el dBASE III Plus, pero a pesar de sus limitaciones, el Vulcan era un potente lenguaje de Base de Datos para aquella época. El programa encontró pocos pero dedicados seguidores.

Uno de esos seguidores fue George Tate, un distribuidor de software, quien puso en marcha unas cuantas técnicas de mercadeo para incrementar las ventas del Vulcan, en primer lugar cambió el nombre por el de dBASE II (no hubo dBASE I).

Años más tarde, y debido al auge de las redes de ordenadores y al deseo de los compradores de aumentar la cooperación entre usuarios, se desarrolló el dBASE III Plus. El programa ofrece un número significativo de mejoras respecto al dBASE II y el dBASE III. (Jones, 1990)

- Access.

Desde la introducción de Microsoft Access al mercado de las Bases de Datos en octubre de 1992, millones de usuarios en todo el mundo han adoptado esta flexibilidad y facilidad de uso y lo han convertido en uno de los primeros sistemas de manejo de Base de Datos.

Access 95.

Microsoft Access para Windows 95 es una potente herramienta de gestión de datos que se puede utilizar para ordenar, organizar o presentar la información que necesitamos en nuestros días.

Access 97.

Microsoft Access 97, se construye sobre el éxito de versiones anteriores: creación de tablas, relaciones, consultas, formularios y macros y la posibilidad de utilizar sus tablas con otras aplicaciones como Visual Basic (como se hará en este trabajo). (Microsoft, 1996 – O'Brien, Pogge, White, 1997)

- Visual Fox Pro.

En los "viejos tiempos" (finales de la década de los ochenta) se pretendió que FoxBASE fuera un clon de dBASE. Si dBASE era bueno, FoxBASE sería mejor y más rápido (o al menos esta era la idea).

Fox Pro 1.0 fue la primera desviación respecto a la compatibilidad con dBASE. Comenzó introduciendo algunos conceptos nuevos en el diseño GUI y formas para el desarrollo de software que lo pusieron por encima de dBASE. Fox Pro realmente tomó las riendas con la versión 2.0. cuando se presentó Fox Pro 2.0, se incluyeron varias tecnologías importantes que revolucionaron al mercado de desarrollo de Bases de Datos.

Visual Fox Pro es el mejor producto de desarrollo para la creación de aplicaciones de misión crítica, en el ámbito empresarial, orientadas a objetos y cliente / servidor, diseñadas para su distribución global o local. (Bazian, 2000)

- SQL.

El SQL Programmer 32 es una herramienta orientada para equipo de desarrollo, en particular para programadores en Sistema Manejadores de Bases de Datos Relacionales (RDBMS) multiusuario. Sin embargo, este puede ser de utilidad para administradores de sistema, administradores de Bases de Datos y líderes de proyecto. De hecho, cualquiera que se encuentre involucrado en el desarrollo, la administración o el mantenimiento de RDBMS que use servidores Oracle, SyBase y Microsoft SQL Server. (Tutorial SQL, www.tutoriales)

- Oracle.

Oracle 9i, producto que se lanzó a inicios del año 2000 y que ofrece funciones para e-commerce, transacciones mercantiles y otras mediante Oracle9i Application Server y el manejador de Bases de Datos propio. Permite compartir archivos en línea, desplegar información recopilada o utilizar aplicaciones predefinidas por terceros, controlando todo mediante la autenticación de usuarios y más.

Sus principales características son:

- Registro en línea y definición de todos los parámetros del portal desde Internet.
- Integración de aplicaciones y recursos desde el Web (páginas, aplicaciones y otros componentes reutilizables).
- Opción para que usuarios finales publiquen y compartan archivos en línea, de forma pública y con password.
- Interface personalizable por usuario pero estandarizada dentro de la idea general.
- Herramientas de administración basadas en Web, tanto monitores de uso como tráfico.

Ofrecen también valores agregados como servicios de noticias, herramientas de productividad (mapas, rastreo de paquetes, traductores, etc.) y herramientas de desarrollo. (Portal.Oracle.com)

Aplicación de un Sistema de Captura de Datos para Escuela Primaria con Interfase Gráfica.

El presente capítulo tiene como objetivo, describir brevemente lo que es un sistema de captura de datos para una escuela primaria y la descripción de las ventanas utilizadas dentro de la interfase gráfica.

4.1.-Metodología de diseño para el proyecto: Sistema de Captura de Datos para Escuela Primaria con Interfase Gráfica.

El motivo del presente, es definir el planteamiento del problema de investigación, entendiéndolo que es sobre éste el objeto de estudio para determinar la factibilidad del proyecto que se va a presentar. Una vez presentado el problema; como siguiente paso se planteará la pregunta de investigación en la que debe reflejarse el cómo establecer la solución del objeto en estudio. A continuación, se definirá el objetivo general, dentro del cual se debe describir cuál es la razón por la que se desarrollará el proyecto, es decir, cuál es el problema al que se le quiere dar solución con la implementación; cabe mencionar que a la par del objetivo general, se definirán los objetivos particulares, dentro de los cuales deben entrar la justificación de la importancia que el proyecto tiene para la institución para la cual es elaborado y su población (alumnos, directivos, docentes).

Posteriormente, se visualiza el contexto de estudio, en este punto se especifica el tipo de población al que se aplica, así como su entorno.

4.1.1.-Planteamiento del problema.

Uno de los grandes desafíos dentro del departamento de control escolar de toda institución educativa, es lograr el mejoramiento de la calidad en los servicios que se ofrece a los alumnos e institución en general.

El personal encargado de llevar toda esta tarea, requiere que los sistemas de captura y control de información propios (elaborados dentro de la misma institución) u oficiales (enviados por la Secretaría de Educación Pública –SEP-), dejen de ser una “pesadilla”, ya que es bien sabido por todos los que se dedican al rubro educativo, que muy pocos son los que se arriesgan a trabajar con un sistema de captura y control escolar, ya que según dicen, el problema empieza desde la instalación del sistema y continúa durante los procesos de captura (fallas del sistema al hacer algún proceso), impresión (conflictos de comunicación sistema/impresora), por citar otros, y finaliza con la no menos difícil o a veces imposible tarea de tratar de reparar el daño en el sistema.

Otro problema conocido, es que la mayoría de los sistemas de control están diseñados en lenguajes de nivel de sistema operativo, lo cual hace que el capturista, secretaria, etcétera, tengan poco interés en entrar o utilizar el sistema, ya que como no hay "botoncitos" para dar clic, comienzan a hacerse a la idea de que el sistema es aburrido y difícil. Dentro de éste, también se menciona que existen conflictos a la hora de la captura en el sentido de que una palabra se teclea de diferentes formas, por lo que se da la situación de no saber "cuál es la buena", ya que no existen posibles alternativas de las que el usuario pueda elegir por ejemplo en ocasiones: Netzahualcóyotl, se captura como "Nezahualcóyotl" o "Nezahualcoyotl".

Todos estos conflictos y fallas del sistema generan una gran cantidad de problemas: retraso en la captura y entrega de los reportes requeridos por las autoridades educativas, conflictos por no entregar dichos reportes como debe ser, evasivas por parte del personal de control para no trabajar en el sistema, confusión por no existir apoyo por parte del departamento de sistemas de la Secretaría de Educación Pública (SEP), etcétera.

Con este antecedente, la dirección de la Escuela Primaria Particular Incorporada "Francisco Díaz de León" ha solicitado sea desarrollado un sistema de captura y control que acabe con todos los problemas y conflictos descritos. Se pide que el sistema sea elaborado de manera que pueda ser instalado y utilizado de manera fácil, segura y por cualquier persona, aunque no tenga amplios conocimientos en manejo de un ordenador, además de que se tenga la seguridad de que todo funcionará de manera correcta y sin contratiempos para todos los procesos que deban hacerse para lograr un control óptimo de calidad. El sistema deberá soportar una matrícula de 247 alumnos (aproximadamente), 11 docentes y 3 directivos; además de que sea desarrollado con la tecnología que se tiene disponible en este momento.

4.1.2.-Pregunta de Investigación.

El funcionamiento óptimo de una institución educativa, se logra cuando todos sus elementos funcionan coordinadamente para lograr los objetivos como se establece y en el tiempo planeado.

A nivel de zona escolar, tanto escuelas particulares, como públicas (más aún éstas) durante el ciclo escolar 2002-2003, presentaron un alto porcentaje en las variaciones de tiempo de entrega de la documentación y estadísticas de inscripción, cambios, bajas, etcétera, a causa de conflictos con los sistemas de control enviados por las autoridades educativas, ya que mientras algunas instituciones aún no hallaban cómo instalar el sistema, otras aún no entendían como manejarlo (no existen manuales de apoyo y los que hay no son claros y concisos). En el ciclo actual se está presentando una situación similar, ya que el "nuevo" sistema simplemente no funciona, por lo que cada escuela debe sacar como puede todo este trabajo.

Es aquí donde las instituciones públicas quedan en desventaja, ya que al no contar con personal capacitado en el área, no pueden continuar y por lo tanto no pueden llevar al día su control de información de alumnos y personal (Dirección de la escuela Primaria Francisco Díaz de León y Supervisión Escolar 06 de Netzahualcóyotl).

Toda la problemática descrita anteriormente conlleva a un problema mayor al final del ciclo escolar: error en los datos = faltante de documentos oficiales (boletas, certificados, etcétera).

Considerando todas estas situaciones, el problema a ser estudiado se basa en la siguiente pregunta de investigación:

¿Cómo consideran los directivos de las instituciones de la zona escolar 06 de Educación Primaria los sistemas de Captura y Control que reciben de parte de autoridades de la SEP?

Y con base en ello:

¿Al utilizar un sistema con Interfase Gráfica se resolverán todos los conflictos en la Captura y Control de Datos de Alumnos, Directivos y Docentes?

4.1.3.-Objetivo General.

Desarrollar un sistema de captura y control de datos escolares con Interfase Gráfica que mejore la calidad de los servicios escolares en la Escuela Primaria Particular Incorporada "Francisco Díaz de León" a partir del conocimiento de la problemática planteada para obtener soluciones con la tecnología que se tiene disponible.

4.1.4.-Objetivos particulares.

a).-Determinar a partir de la opinión de los usuarios del sistema, los avances, ventajas y desventajas de utilizar un sistema con Interfase Gráfica.

b).-Determinar en base a los resultados obtenidos, las medidas pertinentes que ayuden a mejorar la calidad del sistema de la Escuela.

4.1.5.-Justificación.

La importancia del presente trabajo radica en que permitirá a la Institución tener un control más seguro de los datos e información de los directivos y docentes, pero lo más importante, de la matrícula escolar. La opinión de los encargados del control escolar será muy importante, ya que permitirá conocer la problemática que esta actividad implica.

Esto es importante, ya que si los directivos conocen bien las necesidades que el departamento de Control Escolar tiene, podrá tomar medidas para adoptar y adaptar decisiones que permitan satisfacer la demanda que se especifica.

Todo esto permitirá que el prestigio de la Escuela esté en excelente nivel y la satisfacción de los directivos, padres de familia y alumnos sea de tal manera, que determinen inscribir a otros miembros de la familia o recomendar la Escuela a personas cercanas a ellos.

4.1.6.-Contextualización.

Dentro del marco de la automatización de sistemas y de las transformaciones que éstos sufren día a día, la Educación como parte del sistema social es la Institución en la que se perciben de manera más clara las condiciones por las que atraviesa esta sociedad.

De esta manera, la organización y desarrollo de la educación deben ser consideradas a la luz de las determinaciones del orden social. Las características particulares de cada institución no solo de nivel primaria, deben hacer un énfasis a la utilización de técnicas de la Institución y a la atención de los requerimientos académicos mediante cuestionarios orientados a los recursos humanos para atacar los problemas existentes que cada nivel enfrenta.

El desafío consiste en que a mediano plazo, la Escuela Primaria "Francisco Díaz de León", tenga la posibilidad de lograr un equilibrio entre sus condiciones internas de eficacia y productividad, y las exigencias de calidad que demanda el contexto de la Educación Básica en México.

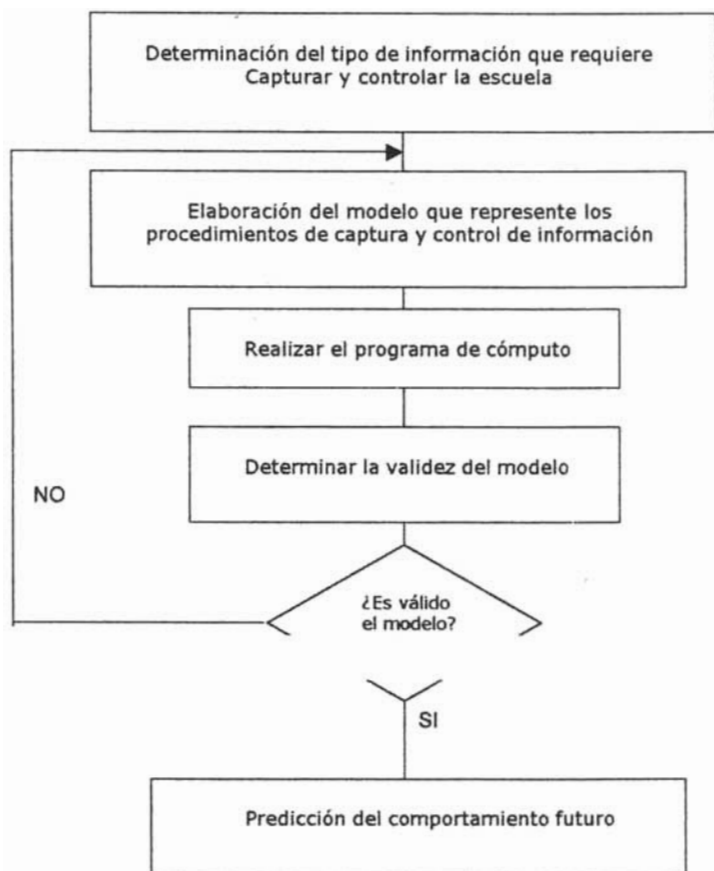
4.2.-Entendiendo al Sistema de Captura para Primaria.

4.2.1.-Sistema de Captura.

De acuerdo a lo expuesto anteriormente, el sistema de Captura y Control quedaría integrado de la siguiente manera:

El software con el que cuenta la Escuela, y que de acuerdo a las exigencias del caso puede utilizarse es el siguiente:

- ❖ Visual Basic para la interfase gráfica
- ❖ Access para el manejador de bases de datos



4.1 Diagrama donde se muestran las etapas para desarrollar un modelo

4.3.-Requerimientos para el sistema.

En esta sección se describen los requerimientos de hardware y software para la elaboración de la interfase gráfica y la base de datos. Se tiene como objetivo desarrollar la interfase gráfica con la tecnología disponible. El ordenador es el elemento central de la interfase y la base de datos que será desarrollada, esto es, el ordenador interviene en el inicio de la operación, en la captura de los datos por el operador, en el envío, recepción y representación de datos.

4.3.1.-Requerimientos de hardware y software.

A continuación se enlistan los requerimientos de hardware y software utilizados en el desarrollo del sistema.

CONCEPTO	ESPECIFICACION
HARDWARE	
Tipo de computadora	PC x86 Family 15 Model 1 Stepping 2
Microprocesador	GenuineIntel Pentium 4
Monitor	GeForce2 MX-100 / 200
Memoria	128,0 Mb
Cd ROM	CDRW 24x 10x 40x
Disco Duro 1	26.0 Gb
Disco Duro 2	11.1 Gb
Modem	HSP56 MR
Teclado	Estándar 101 / 102 teclas Ortek
Ratón	Logitech 2 botones
Unidad de disco flexible	3 ½ "
SOFTWARE	
Sistema operativo	Windows 98
Software utilizado	Visual Basic
	Access

4.2 Cuadro que describe el hardware y software utilizado

4.3.2.-Información fuente.

Aquí se menciona la información que se utilizó en la elaboración de la interfase gráfica, los datos que se utilizaron son los siguientes:

DATOS GENERALES
Presentación
Nombre del sistema
Objetivo
Objetivo del sistema
Sugerencias y ayuda
Ingreso del grado a capturar
Menú Principal
Captura
Reportes
Utilerías
Salir

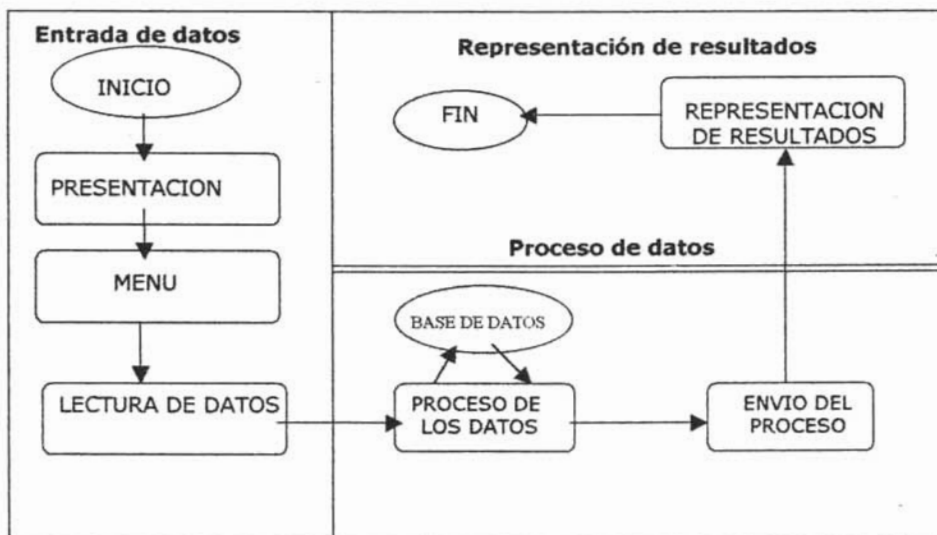
4.3 Cuadro que muestra la información general de la interfase gráfica

4.4.-Estructura del sistema.

El sistema está diseñado en el lenguaje de programación Visual Basic, en el cual se desarrolla el ambiente gráfico (GUI) ya que su fácil utilización en la elaboración de ventanas, botones, barras y menús hacen de ella una herramienta amigable para desarrollar dicho ambiente gráfico para un mejor manejo del sistema de captura.

4.4.1.-Estructura principal.

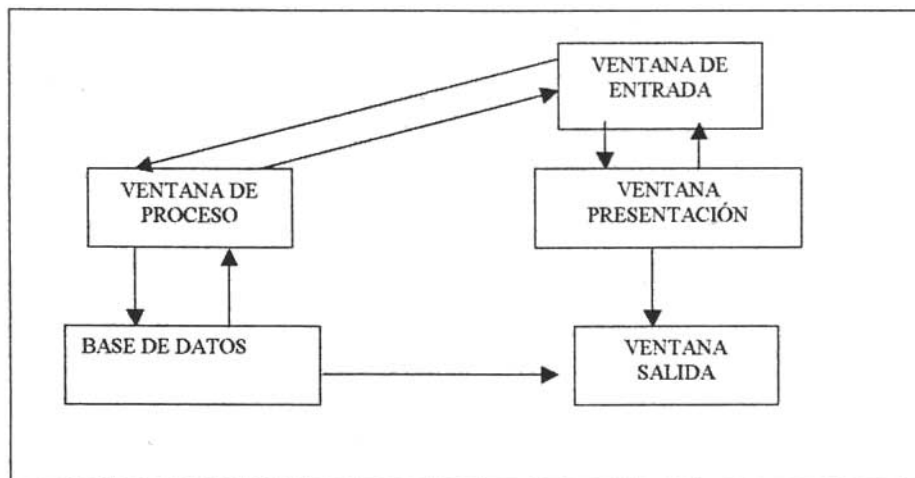
La estructura principal para interfase está basada en un proceso que incluye una serie de ventanas, este proceso comienza con la entrada de datos, sigue con el proceso en el que el sistema guarda los datos y termina con la presentación de resultados. La siguiente figura muestra el diagrama funcional de la interfase.



4.4 Diagrama funcional del programa de interfase gráfica

4.4.2.-Estructura de la creación de archivos.

Esta estructura muestra como trabajan interactuando las pantallas, desde la ventana principal, donde el usuario puede viajar por el sistema como se muestra a continuación:



Para realizar este trabajo, será necesario tomar en cuenta el contenido de cada uno de los capítulos anteriores, tomando un punto importante para el desarrollo de la interfase: los principios de diseño para una interfase de usuarios.

Cabe mencionar que la manera en que se presentará el Sistema en este trabajo, será explicando uno a uno la manera en que fueron diseñados los formularios que lo integran.

4.4.3.1.-Ventana de presentación.

Será la primera que verá el usuario al cargar al Sistema, será la encargada de explicar cual es el objetivo de dicho Sistema, además de que pedirá la clave de acceso para poder acceder a la siguiente ventana. La figura 4.6, muestra la Ventana de presentación.



ESCUELA PRIMARIA PARTICULAR INCORPORADA
"FRANCISCO DIAZ DE LEON"

ABEDUL 1.0

Objetivo:

- 1.-Capturar toda la información de los alumnos y maestros de todos los grados y grupos.
- 2.-Corregir los errores que puedan surgir durante o después de la captura (error en actas, CURP, etc.)

CLAVE DE ACCESO:

X Cancelar

✓ Aceptar

4.6 Ventana del Objetivo y Acceso

Como puede observarse, existe una caja de texto que es donde el usuario tendrá que ingresar la clave de acceso; además de dos botones que servirán para Cancelar o Aceptar el ingreso a la siguiente ventana. Su función y código se describe a continuación.

Una vez que el usuario ha ingresado la clave correcta, podrá dar clic en el botón "Aceptar" para que el sistema verifique si la clave es válida o no, en caso de ser correcta, se mostrará la siguiente pantalla para continuar trabajando, en caso contrario, el sistema mandará un mensaje de error, (figura 4.7) informando al usuario de esto y para que nuevamente se tenga la oportunidad de ingresar la clave.

Error



ERROR. CLAVE INCORRECTA

Aceptar

4.7 Mensaje de error en la clave de acceso al Sistema

El botón "Cancelar" servirá para detener el trabajo del sistema y salir de éste.

A continuación se muestra el código necesario para el funcionamiento de los botones.

```
*** Botón "Aceptar"  
Private Sub Command1_Click()  
If Text1.Text = "" Then  
BIENVENIDA.Hide  
MENUPRINCIPAL.Show , Me  
Else  
MsgBox "ERROR, CLAVE INCORRECTA",  
vbCritical, "Error"  
Text1.Text = ""  
Text1.SetFocus  
End If  
End Sub  
  
***Botón "Cancelar"  
Private Sub Command2_Click()  
End  
End Sub
```

El siguiente código será para que el usuario no tenga problema si da la clave con el teclado configurado en letras minúsculas, automáticamente se conviertan en mayúsculas. Cabe mencionar que este mismo código será utilizado en todas las cajas de texto del programa ya que todo se manejará con este tipo de letra.

```
Private Sub Text1_KeyPress(KeyAscii As Integer)  
If (KeyAscii >= 97) And (KeyAscii <= 122) Then  
KeyAscii = KeyAscii - 32  
End If  
End Sub
```

4.4.3.2.-Ventana del Menú Principal.

Es la que mostrará al usuario los diferentes menús que contendrán las opciones para poder realizar la Captura, Imprimir Reportes, Realizar RespalDOS, tanto de alumnos y maestros, además de los menús Salida y Ayuda. La siguiente figura, muestra la ventana del Menú Principal.



De acuerdo a la figura anterior, se comenzarán a explicar cada una de las opciones que presentan los menús mencionados anteriormente.

4.4.3.2.1.-El Menú Captura.

A través de este menú, se llevará a cabo toda la captura de datos y evaluaciones finales y promedio general de alumnos, además de los datos del personal directivo y docente.

El Submenú Alumnos.

Dentro de este submenú estarán las dos opciones para capturar lo concerniente a alumnos: Datos y Calificaciones.

Datos.- Servirá para llevar a cabo la captura de los datos generales de los alumnos por grado. El proceso de captura será el mismo para los seis grados, así como los botones que ayudarán a trabajar dentro de esta pantalla. La pantalla Datos es similar a la que se muestra a continuación:

Primer Grado [X]

Nombre del alumno: Sexo:
 Apellido paterno Apellido Materno Nombre (s) M o F

Fecha de Nacimiento: C.U.R.P.: Grupo:
 dd/mm/aa

Dirección:
 Calle/Número Colonia Delegación o Municipio

Entidad: Teléfono:

REGISTRO:

 Nuevo	 Anterior	 Siguiente	 Borrar
			

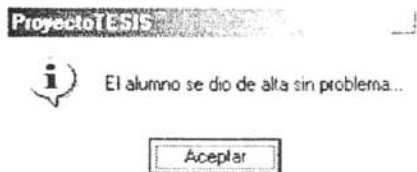
Como se observa, existen ocho botones, los cuales servirán para trabajar con los registros de los alumnos. A continuación se muestra el código para que cada botón haga su función específica.

Botón Nuevo.-Es el que pondrá una nueva pantalla cuando se quiera dar de alta un nuevo registro. El código utilizado es el siguiente:

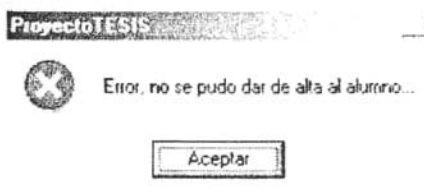
```
Private Sub Command4_Click()
sMovimiento = "N"
***NUEVO REGISTRO
Command1.Enabled = True
Text1.Enabled = True
Text1.Text = ""
Text1.SetFocus
Text2.Enabled = True
Text2.Text = ""
Text3.Enabled = True
Text3.Text = ""
Text4.Enabled = False
Text6.Enabled = True
Text6.Text = ""
Text7.Enabled = True
Text7.Text = ""
Text8.Enabled = True
Text8.Text = ""
Text9.Enabled = True
Text9.Text = ""
Text10.Enabled = True
Text10.Text = ""
```

Exit Sub
End Sub

Botón Guardar.-Será el encargado de guardar en la Base de Datos los registros que son capturados. Si el registro se guarda correctamente, se mostrará el siguiente mensaje.



En caso contrario, mostrará el siguiente mensaje.



El siguiente código es el que ayudará a realizar esto.

```
Private Sub Command1_Click()  
Dim CommAlumno As New ADODB.Command  
***GUARDAR REGISTRO hacer el insert en tabla_alumnos  
Command1.Enabled = False  
sNombre = Me.Text1.Text  
sSexo = Me.Text2.Text  
sFecha = Me.Text3.Text  
sCURP = Me.Text6.Text  
sGRUPO = Me.Text10.Text  
sDireccion = Me.Text7.Text  
sEntidad = Me.Text8.Text  
sTelefono = Me.Text9.Text  
sId = Me.Text4.Text
```

```
If sMovimiento = "N" Then  
SQL = "insert into  
alumnos(Nombre,Grado,Sexo,Fecha,CURP,Grupo,Direccion,Entidad," & _  
"Telefono) values (" & sNombre & "," & sGrado & "," & sSexo & "," &  
sFecha & _
```

```
"", "" & sCURP & "", "" & sGRUPO & "", "" & sDireccion & "", "" & sEntidad & "", ""  
& Val(sTelefono) & "")"
```

```
SQL1 = "insert into Calificaciones(Nombre,Grupo,Grado" & _  
") values (" & sNombre & "", "" & sGRUPO & "", "" & sGradog & "")"
```

```
Else
```

```
SQL = "update alumnos set Nombre =" & sNombre & "", Sexo= "" & sSexo  
& "", Fecha = "" & sFecha & _  
", CURP =" & sCURP & "",Grupo = "" & sGRUPO & "",Direccion="" &  
sDireccion & "",Entidad="" & sEntidad & "",Telefono= " & Val(sTelefono) & "  
where Id =" & sId & ""
```

```
SQL1 = "update Calificaciones set Nombre =" & sNombre & "", Grupo = "" &  
sGRUPO & "" where Id =" & sId & ""
```

```
End If
```

```
CommAlumno.ActiveConnection = BaseAlumnos
```

```
CommAlumno.CommandType = adCmdText
```

```
CommAlumno.CommandText = SQL
```

```
CommAlumno.Execute regsaff
```

```
CommAlumno.ActiveConnection = BaseAlumnos
```

```
CommAlumno.CommandType = adCmdText
```

```
CommAlumno.CommandText = SQL1
```

```
CommAlumno.Execute regsaff1
```

```
If regsaff > 0 Then
```

```
MsgBox "El alumno se dio de alta sin problema...", vbInformation
```

```
Text1.Enabled = False
```

```
Text2.Enabled = False
```

```
Text3.Enabled = False
```

```
Text4.Enabled = True
```

```
Text6.Enabled = False
```

```
Text7.Enabled = False
```

```
Text8.Enabled = False
```

```
Text9.Enabled = False
```

```
Text10.Enabled = False
```

```
Else
```

```
MsgBox "Error, no se pudo dar de alta al alumno...", vbCritical
```

```
Text1.Enabled = False
```

```
Text2.Enabled = False
```

```
Text3.Enabled = False
```

```
Text4.Enabled = True
```

```
Text6.Enabled = False
```

```
Text7.Enabled = False
```

```
Text8.Enabled = False
```

```
Text9.Enabled = False
```

```
Text10.Enabled = False
```

```
End If
```

```
End Sub
```

Por otro lado, puede darse la situación de que el usuario por diferentes causas, capture un registro dos o más veces, en este caso antes de guardar, el sistema verifica en toda la Base de Datos de Alumnos que no esté dado de alta el registro, si no es así, procederá a guardarlo, en caso contrario, mandará el siguiente mensaje de advertencia, cancelando el proceso de guardar.

ProyectoTESIS



El Alumno ya Existe en la Base de Datos...

Aceptar

El código que hace posible esto es el siguiente.

```
**** BUSCAR SI YA EXISTE EL ALUMNO ***
```

```
SQL = "select * from alumnos where Nombre =" & sNombre & ""
```

```
Tabla_Alumnos.Open      SQL,      BaseAlumnos,      adOpenKeyset,  
adLockReadOnly, adCmdText
```

```
NumRegs = Tabla_Alumnos.RecordCount
```

```
If NumRegs > 0 Then
```

```
MsgBox "El Alumno ya Existe en la Base de Datos...", vbInformation
```

```
Tabla_Alumnos.Close
```

```
Exit Sub
```

```
End If
```

```
Tabla_Alumnos.Close
```

Botón Modificar.-Es el encargado de modificar algún dato, datos o registro completo. El código para modificar un registro se muestra a continuación.

```
Private Sub Command5_Click()  
**** MODIFICAR REGISTRO  
sMovimiento = "M"
```

**ESTA TESIS NO SALE
DE LA BIBLIOTECA**

```

Text1.Enabled = True
Text2.Enabled = True
Text3.Enabled = True
Text4.Enabled = False
Text6.Enabled = True
Text7.Enabled = True
Text8.Enabled = True
Text9.Enabled = True
Text10.Enabled = True
Text1.SetFocus
Command1.Enabled = True
End Sub

```

Botón Buscar.-Este botón se encarga de buscar un registro a partir del nombre completo del alumno, esto lo hace a través del siguiente cuadro de texto.

A screenshot of a dialog box titled "Proyecto TESIS" with a close button (X) in the top right corner. The main text inside the dialog is "¿Que alumno busca?". To the right of the text are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel). Below the dialog box is a horizontal line.

Si lo encuentra, pondrá en la pantalla los demás datos del registro, en caso contrario mandará un mensaje avisando que no se encontró el alumno y es el que se muestra a continuación.

A screenshot of a dialog box titled "Proyecto TESIS" with a close button (X) in the top right corner. On the left side, there is a circular icon containing a white 'X' on a dark background. To the right of this icon is the text "No se encontró al alumno...". Below the text is a single button labeled "Aceptar". Below the dialog box is a horizontal line.

El código completo se muestra a continuación.

```

Private Sub Command2_Click()
***BUSCAR REGISTRO
Dim Tabla_Alumnos As New ADODB.Recordset
sNombre = InputBox("¿Que alumno busca?")
SQL = "select * from alumnos where Nombre ='" & sNombre & "'"

Tabla_Alumnos.Open      SQL,      BaseAlumnos,      adOpenKeyset,
adLockReadOnly, adCmdText

```



```
NumRegs = Tabla_Alumnos.RecordCount
```

```
If NumRegs > 0 Then
```

```
    MsgBox "El alumno fue localizado...", vbInformation  
    Me.Text1.Text = Tabla_Alumnos.Fields("Nombre").Value  
    Me.Text2.Text = Tabla_Alumnos.Fields("Sexo").Value  
    Me.Text3.Text = Tabla_Alumnos.Fields("Fecha").Value  
    Me.Text6.Text = Tabla_Alumnos.Fields("CURP").Value  
    Me.Text10.Text = Tabla_Alumnos.Fields("Grupo").Value  
    Me.Text7.Text = Tabla_Alumnos.Fields("Direccion").Value  
    Me.Text8.Text = Tabla_Alumnos.Fields("Entidad").Value  
    Me.Text9.Text = Tabla_Alumnos.Fields("Telefono").Value  
    Me.Text4.Text = Tabla_Alumnos.Fields("Id").Value  
    Command1.Enabled = True  
    Command3.Enabled = True  
    Command5.Enabled = True  
    Command6.Enabled = True  
    Command4.Enabled = True
```

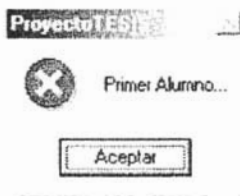
```
Else
```

```
    MsgBox "No se encontró al alumno...", vbCritical
```

```
End If
```

```
End Sub
```

Botón Anterior.-Su función es ayudar a navegar retrocediendo por los registros existentes, lo cual servirá para revisar que todos los alumnos inscritos estén dados de alta. Al llegar al primer alumno dado de alta mostrará el siguiente mensaje:



Este es el código para efectuar lo anterior.

```
Private Sub Command8_Click()  
****RETROCEDE UN REGISTRO  
Dim iNumAlm As Integer  
  
On Error Resume Next  
  
If iban = 0 Then  
    Tabla_AlumnosCons.MoveFirst  
    iban = 1
```

```

Else
    Tabla_AlumnosCons.MovePrevious
End If

If Tabla_AlumnosCons.BOF = False Then

    Me.Text1.Text = Tabla_AlumnosCons.Fields("Nombre").Value
    Me.Text2.Text = Tabla_AlumnosCons.Fields("Sexo").Value
    Me.Text3.Text = Tabla_AlumnosCons.Fields("Fecha").Value
    Me.Text6.Text = Tabla_AlumnosCons.Fields("CURP").Value
    Me.Text7.Text = Tabla_AlumnosCons.Fields("Direccion").Value
    Me.Text8.Text = Tabla_AlumnosCons.Fields("Entidad").Value
    Me.Text9.Text = Tabla_AlumnosCons.Fields("Telefono").Value
    Me.Text10.Text = Tabla_AlumnosCons.Fields("Grupo").Value
    Me.Text4.Text = Tabla_AlumnosCons.Fields("Id").Value
Else
    MsgBox "Primer Alumno...", vbCritical
End If

End Sub

```

Botón Siguiente.-Su función es ayudar a navegar avanzando por los registros existentes, lo cual servirá para revisar que todos los alumnos inscritos estén dados de alta. Al llegar al último alumno dado de lata mostrará el siguiente mensaje:



El código para avanzar por los registros se muestra a continuación.

```

Private Sub Command7_Click()
    ****AVANZA UN REGISTRO
    Dim iNumAlm As Integer

    On Error Resume Next
    iNumAlm = Tabla_AlumnosCons.RecordCount

    If iban = 0 Then
        Tabla_AlumnosCons.MoveFirst
        iban = 1
    Else

```

```

Tabla_AlumnosCons.MoveNext
End If

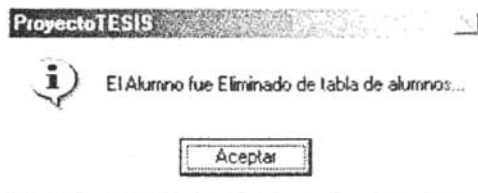
If Tabla_AlumnosCons.EOF = False Then

    Me.Text1.Text = Tabla_AlumnosCons.Fields("Nombre").Value
    Me.Text2.Text = Tabla_AlumnosCons.Fields("Sexo").Value
    Me.Text3.Text = Tabla_AlumnosCons.Fields("Fecha").Value
    Me.Text6.Text = Tabla_AlumnosCons.Fields("CURP").Value
    Me.Text7.Text = Tabla_AlumnosCons.Fields("Direccion").Value
    Me.Text8.Text = Tabla_AlumnosCons.Fields("Entidad").Value
    Me.Text9.Text = Tabla_AlumnosCons.Fields("Telefono").Value
    Me.Text10.Text = Tabla_AlumnosCons.Fields("Grupo").Value
    Me.Text4.Text = Tabla_AlumnosCons.Fields("Id").Value
Else
    MsgBox "Ultimo Alumno...", vbCritical
End If

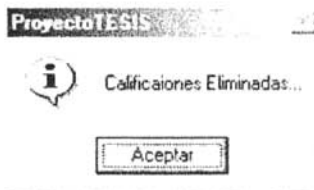
End Sub

```

Botón Eliminar.-Se encarga de eliminar el alumno que es buscado o el que se tiene en la pantalla *Datos* en ese momento. Esto lo hace de forma definitiva, por lo que debe tenerse cuidado al usar este botón. Al eliminar el alumno, mostrará el siguiente mensaje confirmando la eliminación.



Cabe mencionar, que al eliminar un registro de *Datos*, también se eliminarán las evaluaciones que pudiera haber tenido del registro *Calificaciones* que será explicado más adelante, por lo que el sistema mandará el mensaje de eliminación de evaluaciones.



El código para eliminar un registro se muestra a continuación.

```

Private Sub Command3_Click()
****ELIMINAR REGISTRO
Dim CommAlumno As New ADODB.Command
Command1.Enabled = False

If Trim(Me.Text1.Text) = "" Then
    sNombre = InputBox("¿Que alumno desea eliminar?")
Else
    sNombre = Me.Text1.Text
End If

'BORRAR ALUMNO DE DATOS DE ALUMNOS
SQL = "delete from alumnos where Nombre = " & sNombre & ""
CommAlumno.ActiveConnection = BaseAlumnos
CommAlumno.CommandType = adCmdText
CommAlumno.CommandText = SQL
CommAlumno.Execute regsaff

If regsaff > 0 Then
    MsgBox "El Alumno fue Eliminado de tabla de alumnos...", vbInformation
    Text1.Text = ""
    Text2.Text = ""
    Text3.Text = ""
    Text4.Text = ""
    Text6.Text = ""
    Text7.Text = ""
    Text8.Text = ""
    Text9.Text = ""
    Text10.Text = ""
Else
    MsgBox "No se pudo eliminar el alumno...", vbCritical
End If

'BORRAR ALUMNO DE CALIFICACIONES DE ALUMNOS
SQL = "delete from Calificaciones where Nombre = " & sNombre & ""
CommAlumno.ActiveConnection = BaseAlumnos
CommAlumno.CommandType = adCmdText
CommAlumno.CommandText = SQL
CommAlumno.Execute regsaff
If regsaff > 0 Then
    MsgBox "Calificaciones Eliminadas...", vbInformation
Else
    MsgBox "No se pudo eliminar el alumno...", vbCritical
End If
End Sub

```

Botón Salida.-Su función únicamente consistirá en salir de la pantalla datos y volver al Menú Principal. El código es el siguiente.

```
Private Sub Command6_Click()  
Unload Me  
End Sub
```

Calificaciones.- Servirá para llevar a cabo la captura de las evaluaciones finales así como del promedio general de los alumnos por grado. El proceso de captura será el mismo para los seis grados, lo único diferente será que el número de materias varía en algunos grados. A continuación se presentan las pantallas de Calificaciones de los diferentes grados.

Primero y Segundo Grados.

Calificaciones Primer Grado

Alumno: Grupo:

Promedios finales por materia

ESPAÑOL	MATEMATICAS	CONOCIMIENTO DEL MEDIO	EDUCACION ARTISTICA	EDUCACION FISICA	PROMEDIO GENERAL ANUAL
0	0	0	0	0	






Anterior **Siguiente** Guardar Modificar Buscar CALCULAR PROMEDIO

Tercer Grado.

Calificaciones Tercer Grado

Alumno: _____ Grupo: _____

Promedios finales por materia





ESPAÑOL	MATEMATICAS	CIENCIAS NATURALES	HISTORIA, GEOGRAFIA Y EDUCACION CIVICA	EDUCACION ARTISTICA	EDUCACION FISICA	PROMEDIO GENERAL ANUAL
						
Anterior		Siguiente				

Cuarto, Quinto y Sexto Grados.

Calificaciones Cuarto Grado

Alumno: _____ Grupo: _____

Promedios finales por materia

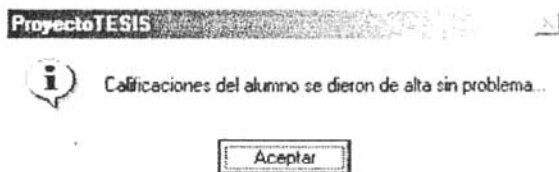
ESPAÑOL	MATEMATICAS	CIENCIAS NATURALES	HISTORIA	GEOGRAFIA	EDUCACION CIVICA	EDUCACION ARTISTICA	EDUCACION FISICA	PROMEDIO GENERAL ANUAL
								
Anterior		Siguiente						

Los botones que ayudarán a trabajar dentro de esta pantalla son los mismos para los seis grados, por lo que se explicará una sola ocasión la función de los siete botones de estas pantallas.

IMPORTANTE. Antes de comenzar con la explicación de los botones, es necesario mencionar que para poder trabajar con las pantallas de calificaciones, deben tenerse capturados los Datos de los alumnos, ya que de ellos se toman los campos "Nombre del Alumno" y "Grupo", que son los que ayudaran al usuario a llevar el orden de cual es el alumno que se va a capturar en cuestión, y así eliminar cualquier problema de coordinación de los alumnos capturados en Datos con los que se capturan en Calificaciones. Estos campos se ilustran en la siguiente figura.

Alumno: Grupo:

Botón Guardar.-Es el encargado de guardar las evaluaciones y promedio del alumno en cuestión cuando se ha verificado que estas sean correctas. Al hacerlo mostrará el siguiente mensaje.



El código que realiza todo esto se muestra a continuación.

```
Private Sub Command6_Click()
Dim CommAlumno As New ADODB.Command
Dim sC1, sC2, sC3, sC4, sC5, sPG, sNombre, SQL As String
Dim regsaff As Integer
****GUARDAR REGISTRO hacer el insert en tabla_alumnos
sC1 = Me.Text1.Text
sC2 = Me.Text2.Text
sC3 = Me.Text3.Text
sC4 = Me.Text4.Text
sC5 = Me.Text5.Text
sPG = Me.Text6.Text
sNombre = Me.Text7.Text
If sMovimiento = "N" Then
Else
SQL = "update calificaciones set Espanol = " & sC1 & ", Matematicas = "
& sC2 & ", Conocimiento = " & sC3 & _
", Artistica = " & sC4 & ",Fisica = " & sC5 & ",PG= " & sPG & " where
Nombre =" & sNombre & ""
End If

CommAlumno.ActiveConnection = BaseAlumnos
CommAlumno.CommandType = adCmdText
```

```
CommAlumno.CommandText = SQL
CommAlumno.Execute regsaff
```

```
If regsaff > 0 Then
    MsgBox "Calificaciones del alumno se dieron de alta sin problema...",
vbInformation
Else
    MsgBox "Error, no se pudo dar de alta las calificaciones del alumno...",
vbCritical
End If

End Sub
```

Botón Modificar.-Este botón tiene dos funciones, una de ellas es que es el botón que se necesita oprimir para poder capturar las calificaciones de un alumno, ya que mientras no se oprima, las casillas de evaluaciones, permanecerán bloqueadas. La otra función es la de modificar las evaluaciones de un determinado alumno. El código que permite hacer esto es el siguiente.

```
Private Sub Command10_Click()
    *** MODIFICAR REGISTRO
    sMovimiento = "M"
    Text1.Enabled = True
    Text2.Enabled = True
    Text3.Enabled = True
    Text4.Enabled = True
    Text5.Enabled = True
    Text6.Enabled = False
    Text7.Enabled = False
    Text8.Enabled = False
    Text1.SetFocus
End Sub
```

Botón Buscar.-Su función es buscar a un alumno para verificar que ya tiene evaluaciones asignadas o modificar si fuera necesario. Este botón funciona de manera similar al botón "Buscar" de la Pantalla "Datos", es decir, coloca un cuadro de diálogo en pantalla preguntando cuál es el alumno que se busca, y mandará los mismos mensajes de la Pantalla "Datos" en caso de encontrarlo y en caso de no hacerlo. El código para este botón se muestra a continuación.

```
Private Sub Command7_Click()
    '===BUSCAR REGISTRO
    Dim Tabla_Alumnos As New ADODB.Recordset
    Dim sNombre As String
    Dim SQL As String
    Dim NumRegs As Integer
```



```
sNombre = InputBox("¿De qué alumno busca calificaciones?")  
SQL = "select * from Calificaciones where Nombre ='" & sNombre & "' and  
grado between '1' and '2'"
```

```
Tabla_Alumnos.Open SQL, BaseAlumnos, adOpenKeyset,  
adLockReadOnly, adCmdText  
NumRegs = Tabla_Alumnos.RecordCount
```

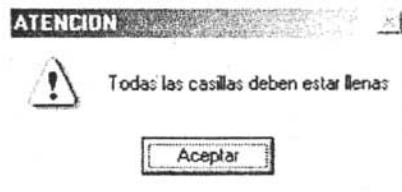
```
If NumRegs > 0 Then  
    Tabla_Alumnos.MoveFirst  
    MsgBox "El alumno fué localizado...", vbInformation  
    Me.Text7.Text = Tabla_Alumnos.Fields("Nombre").Value  
    Me.Text8.Text = Tabla_Alumnos.Fields("Grupo").Value  
    Me.Text1.Text = Tabla_Alumnos.Fields("Español").Value  
    Me.Text2.Text = Tabla_Alumnos.Fields("Matematicas").Value  
    Me.Text3.Text = Tabla_Alumnos.Fields("Conocimiento").Value  
    Me.Text4.Text = Tabla_Alumnos.Fields("Artistica").Value  
    Me.Text5.Text = Tabla_Alumnos.Fields("Fisica").Value  
    Me.Text6.Text = Tabla_Alumnos.Fields("PG").Value  
    Command5.Enabled = True  
    Command6.Enabled = True  
    Command10.Enabled = True
```

```
Else  
    MsgBox "No se encontro al alumno en este grado...", vbCritical  
End If
```

```
End Sub
```

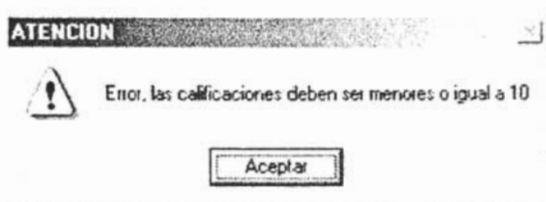
Botón Calcular Promedio.-Aparte de la función de calcular el promedio general de un alumno, este botón se encarga de verificar:

- Que todas las casillas estén llenas, ya que en ningún caso se permitirá una casilla o casillas vacías. Si esto sucede, al oprimir este botón se pondrá en pantalla el mensaje:



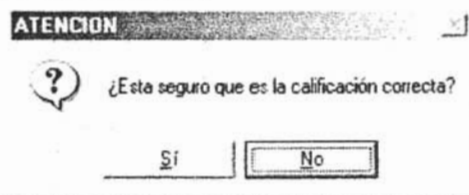
con lo que se advierte al usuario que debe llenar todas las casillas.

- Que las evaluaciones estén en el rango menor a 10 (no existe 11, 12, 34, como evaluación) o menores a 5, ya que no puede colocarse un 4, por ejemplo, como evaluación final (aunque a veces nos lo merecemos). En caso de que esto suceda, el sistema mandará el mensaje:



con lo que se advierte al usuario que debe corregir la evaluación que esté mal.

- Otro caso que puede presentarse y que es válido para evaluaciones finales, es que exista una evaluación de 5, en este caso el sistema preguntará si es correcta esta calificación, en caso afirmativo, se calculará el promedio tomando en cuenta este 5, en caso contrario, se permitirá corregirlo. El mensaje se muestra a continuación:



El código para este botón es el siguiente.

```
Private Sub Command5_Click()
```

```
***CALCULAR PROMEDIO
```

```
Dim CAL1, CAL2, CAL3, CAL4, CAL5 As Double
```

```
Dim mensaje, botones, título, respuesta
```

```
On Error GoTo Manejador_error
```

```
CAL1 = CDbI(Text1.Text)
```

```
CAL2 = CDbI(Text2.Text)
```

```
CAL3 = CDbI(Text3.Text)
```

```
CAL4 = CDbI(Text4.Text)
```

```

CAL5 = CDbI(Text5.Text)

If CAL1 > 10 Or CAL2 > 10 Or CAL3 > 10 Or CAL4 > 10 Or CAL5 > 10 Then
    MsgBox "Error, las calificaciones deben ser menores o igual a 10",
        vbExclamation, "ATENCIÓN"
Text6.Enabled = False
Text1.SetFocus
ElseIf CAL1 < 6 Or CAL2 < 6 Or CAL3 < 6 Or CAL4 < 6 Or CAL5 < 6 Then
    mensaje = "¿Esta seguro que es la calificación correcta?"
    botones = vbYesNo + vbQuestion + vbDefaultButton2
    título = "ATENCIÓN"
    respuesta = MsgBox(mensaje, botones, título)
    If respuesta = vbYes Then
        Text6.Text = CStr((CAL1 + CAL2 + CAL3 + CAL4 + CAL5) / 5)
        Text6.Enabled = False
        Command6.Enabled = True
        Command7.Enabled = True
        Command10.Enabled = True
    Else
        Text1.SetFocus
        Text6.Text = ""
        Text6.Enabled = False
        Command6.Enabled = False
        Command7.Enabled = False
        Command10.Enabled = False
    End If
Else
    Text6.Text = CStr((CAL1 + CAL2 + CAL3 + CAL4 + CAL5) / 5)
    Text6.Enabled = False

```

```
Command6.Enabled = True
Command7.Enabled = True
Command10.Enabled = True
```

```
End If
```

```
Exit Sub
```

```
Manejador_error:
```

```
MsgBox "Todas las casillas deben estar llenas", vbExclamation, "ATENCIÓN"
```

```
Text1.SetFocus
```

```
End Sub
```

Botón Anterior.-Tiene la misma función que su similar de la pantalla "Datos", es decir, retrocede uno a uno entre los registros existentes, y al llegar al primero mandará el mensaje de "Primer Alumno". Su código es el mostrado a continuación.

```
Private Sub Command2_Click()
```

```
'RETROCEDER UN REGISTRO
```

```
Dim iNumAlm As Integer
```

```
On Error Resume Next
```

```
iNumAlm = Tabla_AlumnosCons.RecordCount
```

```
If iban = 0 Then
```

```
    Tabla_AlumnosCons.MoveFirst
```

```
    iban = 1
```

```
Else
```

```
    Tabla_AlumnosCons.MovePrevious
```

```
End If
```

```
If Tabla_AlumnosCons.BOF = False Then
```

```
    Me.Text7.Text = Tabla_AlumnosCons.Fields("Nombre").Value
```

```
    Me.Text8.Text = Tabla_AlumnosCons.Fields("Grupo").Value
```

```
    Me.Text1.Text = Tabla_AlumnosCons.Fields("Español").Value
```

```
    Me.Text2.Text = Tabla_AlumnosCons.Fields("Matematicas").Value
```

```
Me.Text3.Text = Tabla_AlumnosCons.Fields("Conocimiento").Value
```

```
Me.Text4.Text = Tabla_AlumnosCons.Fields("Artistica").Value
```

```
Me.Text5.Text = Tabla_AlumnosCons.Fields("Fisica").Value
```

```
Me.Text6.Text = Tabla_AlumnosCons.Fields("PG").Value
```

```
Else
```

```
MsgBox "Primer Alumno...", vbCritical
```

```
End If
```

```
End Sub
```

Botón Siguiente.-También tiene la función que el botón "Siguiente" de la pantalla "Datos": avanzar un registro cada vez que sea oprimido. Al llegar al último registro, mandará el mensaje "Ultimo Alumno" y su código es el siguiente.

```
Private Sub Command1_Click()
```

```
***AVANZAR UN REGISTRO
```

```
Dim iNumAlm As Integer
```

```
On Error Resume Next
```

```
iNumAlm = Tabla_AlumnosCons.RecordCount
```

```
If iban = 0 Then
```

```
Tabla_AlumnosCons.MoveFirst
```

```
iban = 1
```

```
Else
```

```
Tabla_AlumnosCons.MoveNext
```

```
End If
```

```
If Tabla_AlumnosCons.EOF = False Then
```

```
Me.Text7.Text = Tabla_AlumnosCons.Fields("Nombre").Value
```

```
Me.Text8.Text = Tabla_AlumnosCons.Fields("Grupo").Value
```

```
Me.Text1.Text = Tabla_AlumnosCons.Fields("Español").Value
```

```
Me.Text2.Text = Tabla_AlumnosCons.Fields("Matematicas").Value
```

```
Me.Text3.Text = Tabla_AlumnosCons.Fields("Conocimiento").Value
```

```
Me.Text4.Text = Tabla_AlumnosCons.Fields("Artística").Value
```

```
Me.Text5.Text = Tabla_AlumnosCons.Fields("Física").Value
```

```
Me.Text6.Text = Tabla_AlumnosCons.Fields("PG").Value
```

```
Else
```

```
MsgBox "Ultimo Alumno...", vbCritical
```

```
End If
```

```
End Sub
```

Botón Salida.-Es el botón con el que el usuario podrá salir de la pantalla "Calificaciones" y regresar al Menú Principal. Su código se muestra a continuación.

```
Private Sub Command11_Click()
```

```
****SALIR
```

```
Unload Me
```

```
On Error Resume Next
```

```
If Tabla_AlumnosCons.Status = 1 Then
```

```
Tabla_AlumnosCons.Close
```

```
End If
```

```
End Sub
```

```
El Submenú Maestros.
```

Dentro de este submenú se van a capturar los datos del personal Directivo y Docente. Al entrar a este submenú, se mostrará la siguiente pantalla.

Nombre del docente: MALAGON SALAZAR ROSALVA 25
 Apellido Paterno Apellido materno Nombre (s)

C.U.R.P.: MASR570909MDFLLS Cargo: Director sin Grupo Grado: Grupo:







Preparación Profesional: LIC. EN EDUCACION PRIMARIA

Grado Académico: Titulado Estado Civil: Soltero

Dirección: PRADO ABEDUL # 12 PRADOS DE ARAGON NEZAHUALCOYOTL
 Calle / Número Colonia Delegación o Municipio

Código Postal: 57170 Entidad: ESTADO DE MEXICO Teléfono: 57949832

Registro:

 Nuevo	 Guardar	 Modificar	 Buscar
Anterior	Siguiente	 Eliminar	

Como puede observarse, en esta pantalla se manejan los mismos botones que en la pantalla "Datos" del submenú *Alumnos*, por lo que su función es exactamente la misma. La única diferencia que existe es que los datos de los Directivos y Docentes, se guardarán en la tabla designada "Maestros".

4.4.3.2.2.-El Menú Reportes.

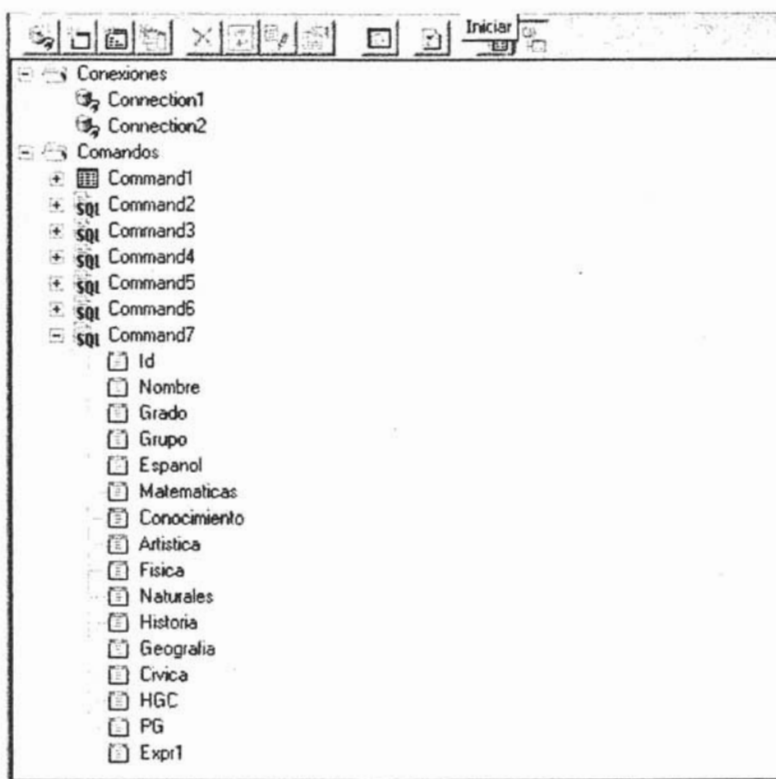
Este Menú es el encargado de generar los reportes para que puedan ser impresos en un formato similar al reporte oficial que debe entregarse a la Supervisión Escolar. Dentro de este Menú, existen dos submenús: *Alumnos* y *Maestros*.

El submenú *Alumnos*.-Es utilizado para imprimir los reportes de las evaluaciones y promedio general de los alumnos por grado. Al seleccionar este submenú, se puede observar que existen los seis grados para que se pueda imprimir su reporte. Para que puedan generarse estos reportes, se procedió de la forma siguiente.

Para los reportes de *Alumnos* se utilizaron seis diseñadores (Data Report), uno para su respectivo grado. En el anexo **A** de este trabajo se pueden observar estos reportes.

El submenú Maestros.-Con el se podrá imprimir un reporte similar al que solicita la Supervisión Escolar, y el cual contiene algunos de los datos capturados de personal Directivo y Docente. En el anexo de este trabajo se puede observar este reporte.

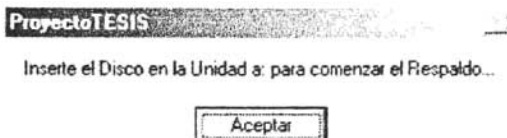
Ahora bien, para conectar los reportes tanto de Alumnos como de Maestros con sus respectivas bases de datos, se utilizó un Data Environment. A continuación se muestra la estructura con la que se llevo a cabo la conexión.



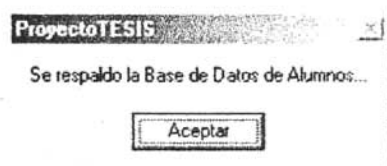
4.4.3.2.3.-El Menú Utilerías.

La tarea que lleva a cabo es respaldar las Bases de Datos en un disco de 3 ½. Esto lo hace a través de los submenús Alumnos y Maestros.

El submenú Alumnos.-Se encarga de hacer el respaldo de todos los alumnos existentes de los seis grados, basta con dar un clic en este submenú. Antes de comenzar el proceso, mandará un mensaje pidiendo sea introducido un disco en la unidad destinada a ello, este mensaje es el siguiente:



y comenzará con el proceso de respaldo. Al terminar de hacer el respaldo, mostrará el siguiente mensaje:



El código para respaldo de Alumnos, se muestra a continuación.

```
Private Sub mnuAlumnosres_Click()  
Dim Ruta As String  
  
MsgBox "Inserte el Disco en la Unidad a: para comenzar el Respaldo..."  
  
archorigen = App.Path + "\datalumnos.mdb"  
Ruta = "a:\datalumnos.mdb"  
n = FreeFile  
Open archorigen For Binary As #n  
byt = LOF(n)  
BB$ = String$(byt, " ")  
BB$ = Input(byt, n)  
Close #n  
m = FreeFile  
Open Ruta For Binary As #m  
Put #m, , BB$  
Close #m  
  
MsgBox "Se respaldo la Base de Datos de Alumnos..."  
End Sub
```

El submenú Maestros.-Tiene la función de respaldar los datos capturados del personal Directivo y Docente, de la misma manera como lo hace con los Alumnos, es decir muestra el mismo mensaje, alertando al usuario que debe introducir un disco, y al terminar, muestra el mensaje:

Se respaldo la Base de Datos de Maestros...

Aceptar

El código para respaldar la Base de Datos de Maestros se muestra a continuación.

```
Private Sub mnuMaestrosres_Click()
Dim Ruta As String
```

```
MsgBox "Inserte el Disco en la Unidad a: para comenzar el Respaldo..."
```

```
archorigen = App.Path + "\maestros.mdb"
```

```
Ruta = "a:\maestros.mdb"
```

```
n = FreeFile
```

```
Open archorigen For Binary As #n
```

```
byt = LOF(n)
```

```
BB$ = String$(byt, " ")
```

```
BB$ = Input(byt, n)
```

```
Close #n
```

```
m = FreeFile
```

```
Open Ruta For Binary As #m
```

```
Put #m, , BB$
```

```
Close #m
```

```
MsgBox "Se respaldo la Base de Datos de Maestros..."
```

```
End Sub
```

4.4.3.2.4.-El Menú Salir.

La función que cumple este menú es muy sencilla, ya que se encarga de salir del sistema de forma definitiva. Es el que el usuario debe utilizar cuando termina una sesión de trabajo. Esto lo lleva a cabo dando clic en el submenú "Salir del Sistema", y su código es el mostrado a continuación:

```
Private Sub mnuSalirsis_Click()
```

```
End
```

```
End Sub
```

4.4.3.2.5.-El Menú Ayuda.

Su función es muy importante, ya que es el que dará apoyo al usuario en tiempo de ejecución del sistema, en caso de que surja alguna duda en alguno de los procedimientos que se estén llevando a cabo. El funcionamiento de este menú, consiste en "jalar" el archivo del manual de usuario creado y el cual es mostrado en el anexo **B** de este trabajo. El código utilizado para hacer este proceso es el siguiente.

```
Private Sub mnuAyuda_Click()  
    Comando = App.Path & "MANUAL DE USUARIO"  
    Shell "c:\Archivos de programa\Microsoft Office\Office10\WINWORD" & "  
" & Comando, vbNormalFocus  
End Sub
```

4.4.4.-La conexión Interfase / Base de Datos.

Es un proceso muy importante, ya que sin ella, no podríamos trabajar con los registros de Alumnos y Maestros que son capturados.

Para poder hacer la conexión, se recurrió a un Módulo (elemento que sirve como el proveedor de los datos de una Base, y al cual se puede acceder desde cualquier parte de la aplicación que lo requiera). El código utilizado dentro del módulo para la conexión de las Bases de Datos de Alumnos y Maestros es el que se muestra a continuación.

```
*** Base de Datos  
Public BaseAlumnos As New ADODB.Connection  
Public BaseMaestros As New ADODB.Connection  
Public sMovimiento, sGradog As String  
Public Tabla_AlumnosCons As New ADODB.Recordset  
Public Tabla_MaestrosCons As New ADODB.Recordset  
Public Tabla_Calificaciones As New ADODB.Recordset  
Public NumRegsCons As Integer  
Public iban As Integer  
  
Public Sub Main()  
Dim sCadena1, sCadena2 As String  
  
sCadena1 = "Provider=Microsoft.Jet.OLEDB.4.0;Data  
Source=datalumnos.mdb;Persist Security Info=False"  
sCadena2 = "Provider=Microsoft.Jet.OLEDB.4.0;Data  
Source=maestros.mdb;Persist Security Info=False"  
  
On Error GoTo ErrAbrir  
BaseAlumnos.Open sCadena1: MsgBox "Se conecto...sBase ok"  
BaseMaestros.Open sCadena2: MsgBox "Se conecto...sServidor ok"
```

BIENVENIDA.Show

Exit Sub

ErrAbrir:

MsgBox "Hubo un Error al Conectarse a las Bases de Datos de Alumnos y Maestros...", vbCritical

End Sub

BIBLIOGRAFÍA

- 📖 Ashley Ruth (1991). **Aprende Unix**. México: Salvat.
- 📖 Balena Francesco. (2000). **Programación Avanzada con Microsoft Visual Basic 6.0**. México: Mc Graw Hill.
- 📖 Balzer, R. and N. Goodman. (1979, pp.58-67). **Principles of Good Software Specification, Proc, on Specifications of Reliable Software** IEEE.
- 📖 Bazian. (2000). **Programación con Visual Fox Pro**. México: McGraw Hill.
- 📖 Burch John G. (1992). **Diseño de Sistemas de Información**. México: Mc Graw Hill, Segunda Edición.
- 📖 Cevallos Francisco Javier (1999). **Visual Basic 6.0, Curso de Programación**. México: Alfaomega.
- 📖 Charre Francisco. (1999). **Programación con C++ Builder**. México: Anaya Multimedia
- 📖 Deitel. (1999). **C++ como programar**. México: Prentice May. 2ª Edición.
- 📖 Eran Donal / Baker M. Pauline (1999). **Gráficas por computadora**. México: Prentice May.
- 📖 Frischknecht. (1990). **Filosofía de la información. (System Teory)**. México: Mc Graw Hill.
- 📖 Hartley Tim, Martín Tim (1991). **DB2/SQL, Manual para programadores** España: Editorial Mc.Graw Hill.
- 📖 Jones Edward. (1990). **Aplique el dBase III Plus**. México: Osborne / Mc Graw Hill.
- 📖 Kroenke David M. (1996). **Procesamiento de Bases de Datos Fundamentos, diseño e instrumentación**. México: Ed. Prentice May Quinta Edición.
- 📖 Microsoft (1996). **Access Para Windows 95 paso a paso**. México. Mc Graw Hill / Microsoft Press

- 📖 Molich, R., Nielsen, J. (1990). Improving a human computer dialogue Communications of the ACM 33. 3 (March), pp 338-348.
- 📖 Molich, R., y Nielsen, J. (1990). "Heuristic evaluation of user interfaces Proceedings of the ACM CHI'90 Conference", pp. 249-256.
- 📖 Molich, R., y Nielsen, J. (1994). "Enhancing the explanatory power of usability heuristics", Proceedings of the ACM CHI'94 Conference, pp. 152-158.
- 📖 O'Brien M. Timothy, Pogge Steven J., White Geoffrey E. (1997). Microsoft Access 97. Desarrollo de soluciones. México: Mc Graw Hill / Microsoft Press
- 📖 PC Media Revista. Año V No. 10 México. Ness.
- 📖 Pressman Roger S. (1990). Ingeniería del Software: Un enfoque práctico. México: Editorial McGraw Hill. Segunda edición.
- 📖 Reisdorph Kent (2000). Aprendiendo Borland C++ Builder 3. México: Prentice Hall.
- 📖 Sánchez Sebastián (1998). Unix y Linux Guía práctica. México: Alfa Omega.
- 📖 Senn James A. (1990). Análisis y Diseño de Sistemas de Información
- 📖 Senn James A. (1990). Análisis y Diseño de Sistemas de Información
- 📖 Silberschatz Abraham, Korth Henry F. Fundamentos de Bases de Datos México: Editorial Mc.Graw Hill. Segunda edición.
- 📖 Smart Cad. (2001). Apuntes del curso de programación Visual Basic Módulos Básico y Avanzado.
- 📖 Tsai Alice Y. H. (1990). Sistemas de Bases de Datos. Administración y uso. México: Editorial Prentice May.

Consultas en Internet

(Dir. [http://www.salonhogar.com/ciencias/tecnologias/computadoras/software de aplicaciones.htm](http://www.salonhogar.com/ciencias/tecnologias/computadoras/software_de_aplicaciones.htm))

(Portal.Oracle.com)

(Tutorial Fortran, [www.tutoriales](http://www.tutoriales.com))

(Tutorial SQL, www.tutoriales)

(Tutorial X Window www.tutoriales)

(Tutorial Xforms www.tutoriales)

📖 Este es un resumen de la materia homónima dictada en el año 1997 en la Universidad Tecnológica Nacional - Facultad Regional Tucumán, ciudad de San Miguel de Tucumán – Argentina.

📖 Freeman, P. Requirements Analysis and Specification, Proc. Intl. Computer Technology Conf. ASME, San Francisco, August, 1980.

📖 Guía de Estudio del Módulo III "Metodología de Construcción de Sistemas de Software" de la Maestría en Ingeniería del Software, Escuela de Posgrado, Instituto Tecnológico de Buenos Aires.

<http://www.geocities.com/lsialer/NotasInteresantes.htm>

<http://www.monografias.com>

<http://www.monografias.com/trabajos/auditoinfo/auditoinfo.shtml>

<http://www.monografias.com/trabajos/maudisist/maudisist.shtml>

📖 Key words, 1990.

📖 Simposio Iberoamericano de Sistemas de Información e Ingeniería de Software en la Sociedad del Conocimiento. Bogota , agosto 29,30 y 31 de 2001.

ANEXO A
REPORTES DE ALUMNOS Y MAESTROS

ESCUELA PRIMARIA FRANCISCO DIAZ DE LEON



ACREDITACION ESCOLAR PARA EDUCACION PRIMARIA CICLO ESCOLAR 2003 -

DOMICILIO: BOULEVARD PRADOS DE ARAGON S/N, FRACC. PRADOS DE

CLAVE DEL CENTRO DE TRABAJO: 15PPR2731D ZONA ESCOLAR: 06

MUNICIPIO: NEZAHUALCOYOTL

Nombre:	CURP:	Grupo:	Esp:	Mat:	Con	Artística:	Física:	PG:
ALARCON GARCIA ERIKA ELIAM	AAGE970304MDFLRR00	A	9.4	9	10	9.2	9.8	9.4
CARDOSO REYES KEVIN ALEXIS	CARK970803HDFRYV01	A	9.6	9.6	10	10	10	9.8
CASTRO PEÑA CRISTINA SARAHÍ	CAPC970616MDFSXR09	A	10	9.6	10	10	10	9.9
CASTRO SANDOVAL JAIME ARTURO	CASJ970204HDFSNM00	A	9.2	9	9.6	10	10	9.5
DEL CORRAL MUÑOZ KAREN	COMK971008MDFRXR09	A	0	0	0	0	0	0
DIAZ AVALOS FERNANDO	DIAF970203HDFZVR00	A	8.2	8.2	9.4	10	10	9.1
FLORES CORTES ARTURO	FOCA970803HMCLLR08	A	10	9.6	9.8	10	10	9.8
GAMERO ESPINOZA VALERIA	GAEV971011MDFMSL04	A	10	9.6	10	10	10	9.9
GARCIA IBAÑEZ ALFONSO	GAIA970112MDFRBL04	A	10	9.8	10	10	10	9.9
HUERTA RIVAS ANA KAREN	HURA970407MDFRVN01	A	9.4	9.4	10	10	10	9.7
LLANO OLIVA MARIT ELISA	LAOM970809MDFLLR06	A	10	10	10	10	10	10
LOMELI VENTURA ROBERTO	LOVR970716HDFMNB05	A	10	9.6	9.8	10	10	9.8
MELO PLAZA SOFIA JIMENA	MEPS971002MDFLLF07	A	10	9.8	10	10	10	9.9
MORALES ARIAS DIANA	MOAD970815MDFRRN06	A	9.6	9.4	9.8	10	10	9.7
MORALES BALDERAS JORGE	MOBJ970901HDFRLR08	A	10	10	10	10	10	10
OLMOS MALDONADO ALISON YAEL	OOMA970912MDFLLL00	A	8.8	8.4	9.6	10	9.8	9.3
PECERO PACHECO BRENDA	PEPB970927MDFCCR05	A	10	10	9.6	10	9.6	9.8
SANCHEZ MUÑOZ JOSE ERIK	SAME970210HDFNXR05	A	9.4	9.6	9.6	10	10	9.7
SANDOVAL FERNANDEZ MAURICIO	SAFM970508HDFNRR06	A	8.4	8.6	9.4	10	10	9.2
SEGOVIA ALATORRE ANDREA	SEAA970901MDFGLN	A	9.2	9.4	10	10	9.8	9.6
SERRANO HERNANDEZ EDMUNDO	SEHE971111HDFRRD01	A	10	9.8	10	10	10	9.9
SOTO OROZCO KAREN ELIZABETH	SOOK970603MDFTRR00	A	10	9.8	9.8	10	10	9.9
TREJO LARA YANINE YURITSI	TELY970808MDFRRN06	A	9.8	10	10	10	10	9.9
TREJO MUÑOZ BERENICE	TEMB970616MDFRXR01	A	9.6	9.6	10	10	10	9.8
TRUJILLO SAMPERIO KARINA	TUSK970411MDFRMR00	A	9.6	9	9.8	10	10	9.6
VALENCIA GRANADOS VERONICA	VAGV970723MDFLRR09	A	9.4	8.4	9.4	10	9.8	9.4
VALENCIA LEON BETSAIDA MADAI	VALB970528MDFLNT05	A	10	9.4	10	10	10	9.8
VILLALOBOS GALLARDO VICTOR	VIGV970430HDFLLC09	A	9.8	9.6	10	10	9.8	9.8
YAÑEZ FERNANDEZ MARIA	YAFF970311MDFXRR03	A	9.8	9.8	10	10	10	9.9
ZEPEDA TORRES ANGEL DE JESUS	ZETA970805HDFPRN06	A	10	9.8	9.8	10	10	9.9



ESCUELA PRIMARIA FRANCISCO DIAZ DE LEON

ACREDITACION ESCOLAR PARA EDUCACION PRIMARIA CICLO ESCOLAR 2003 - 2004

DOMICILIO: BOULEVARD PRADOS DE ARAGON S/N, FRACC. PRADOS DE ARAGON

CLAVE DEL CENTRO DE TRABAJO: 15PPR2731ZONA ESCOLAR: 06

MUNICIPIO:NEZAHUALCOYOTL

Nombre:	CURP:	Grupo:	Espa	Mate	Conocimiento:	Arbética:	Física:	PG:
ALMANZA CHACON ROGELIO	AACR960802HDFHGO00	A	10	10	10	10	10	10
BARAJAS RAMOS KATHERINE DANAE	BARK960914MMCRMT05	A	9.4	9.4	9	9.6	10	9.4
CORDERO NEPOMUCENO MARTHA	CONM960212MDFRPR02	A	10	10	10	10	10	10
CORREA SANCHEZ DIANA TZAZIL	COSD960214MDFRNN00	A	8.8	8	9.2	9.4	10	9.4
DIAZ DE LA VEGA AGUIR EDUARDO	DIV960809HMCZGG	A	9.2	8.8	9.2	10	10	9.4
FERNANDEZ OLIVARES MICHELLE	FEOM961227MDFRLC04	A	10	10	10	10	10	10
GALLEGOS ANAYA NESTOR ALEJANDRO	GAAN960226HDFLNS06	A	9.8	10	10	9.8	10	9.9
GONZALEZ ESPEJEL OSCAR AARON	GOEO951117HDFNSS06	A	9.6	9.6	9.4	10	10	9.7
GUERRERO HERNANDEZ JAVIER	GUHJ961015HDFRRV02	A	8.8	8.2	9	10	10	9.2
HERNENDEZ GONZALEZ ERUVIEL ALEXIS	HEGE960803HDFRNR08	A	8	8.2	8.4	10	10	8.9
MERIDA AGUILAR CESAR	MEAC690516HMCRCGS00	A	10	9.8	9.8	10	10	9.9
MONTAÑO MORALES PATRICIA CORINA	MOMP960921MDFNRT01	A	9.2	9.2	9.4	10	10	9.5
MONTAÑO SOSA MARCELINA IVETTE	MOSM960627MMNNSR	A	8.8	8.8	9	10	10	9.3
ORDUÑO AVENDAÑO LIZET	OUAL960312MDFRVZ00	A	0	0	0	0	0	0
PEREZ BAZAN RAFAEL	PERB960709HDFRZF00	A	9.8	10	9.6	10	10	9.8
RIVERA RAMIREZ DANIEL	RIRD960804HDFVMN05	A	9.6	9.8	9.6	10	10	9.8
RODRIGUEZ SANTIAGO AEJANDRO	ROSA960424HMCDNL01	A	9.8	9.8	10	10	10	9.9
SANTOS MENDEZ MARIA FERNANDA	SAMF961022MDFNRR09	A	0	0	0	0	0	0
TELLEZ AVIÑA YARENY ITZANY	TEAY961219MMNLVR	A	9.8	9.6	9.6	10	10	9.8
VAZQUEZ BARRERA JESSICA	VABJ961107MDFZRS07	A	10	10	10	10	10	10
VILLASEÑOR HUERTA MIRIAM	VIHM960921MDFLRR06	A	10	10	10	10	10	10
BARRAGAN CONTRERAS KARINA	BACK960509MDFRNR07	B	8.8	9.2	9	9.6	10	9.3
BASILIO NIEMBRO ALAN ANTONIO	BANA960630HDFSML01	B	9	9	9	10	10	9.4
BAUTISTA PAZ RODRIGO	BAPR961104HDFTZD03	B	9	9.2	9.4	10	10	9.5
BERMEO ELIAS VALERIA PATRICIA	BEEV961005MMCRLL00	B	9.8	10	10	10	10	9.9
CASTAÑEDA ELIAS MARIA FERNANDA	CAEF961011MMCSLR05	B	8.8	8.2	8.8	10	10	9.1
CONTRERAS OLIVARES ALFREDO	COOA960323HDFNLL06	B	9.4	9.6	10	10	10	9.8
COVARRUBIAS OCHOA ROBERTO	COOR960426HDFVCB09	B	9	9.2	9.4	10	10	9.5
CRUCES MENDOZA KARINA SARAHÍ	CUMK960218MDFRNR02	B	10	10	9.8	10	10	9.9
GASPAR JARAMILLO JAIME OMAR	GAJJ960915HDFSMR05	B	9	9.4	9.6	10	10	9.6
LEDESMA VAZQUEZ MAURICIO	LEVMM960315HMCDZR05	B	9.2	9.2	9.8	10	10	9.6
MALAGON GARCIA URIEL MAURICIO	MAGU960318HMCLRR01	B	8.2	7.8	8.8	9.6	10	9.8
MARTINEZ GARCIA ANA CAROLINA	MAGA960422MMCCRNO9	B	9.8	9.2	10	10	9.8	9.7
OCAMPO GOMEZ NARDA PAOLA	OAGN960616MMCCMR	B	0	0	0	0	0	0
OSCOY GONZALEZ VANESA	OOVG960828MDFSNNO2	B	8.8	9	9.2	10	10	9.4
PEREZ PADILLA IVETTE ANAHI	PEPI960927MDFRDOV07	B	10	10	10	10	10	0
ROMANO MARTINEZ ISRAEL	ROMI960705HMCMR02	B	7.4	6.6	8	9.8	10	8.3
ROMANO ZAVALA ELISA	ROZE960816MDFMYL01	B	9.2	10	9.4	10	9.8	9.6
SANDOVAL BOLAÑOS SAMANTHA	SABS961124MDFNLM	B	9.2	9	9.6	10	10	9.5
SANTIAGO QUINTERO ANA KAREN	SAQA960223MDFNNO00	B	9	9.2	9.4	10	10	9.5
UGALDE VELAZQUEZ NNCY MONSERRAT	UAVN960924MDFGLN	B	7.8	7.6	8	9.8	10	9.6
VALENCIA LEON HANIEL	VALH960417HDFLNN09	B	8.2	8.4	9	9.8	10	9



ESCUELA PRIMARIA FRANCISCO DIAZ DE LEON

ACREDITACION ESCOLAR PARA EDUCACION PRIMARIA CICLO ESCOLAR 2003 - 2004

DOMICILIO: BOULEVARD PRADOS DE ARAGON S/N, FRACC. PRADOS DE ARAGON

CLAVE DEL CENTRO DE TRABAJO: 15PPR2731D

ZONA ESCOLAR: 06

MUNICIPIO: NEZAHUALCOYOTL

Nombre:	CURP:	Grupo:	Espa	Mate	Naturales:	HGC:	Artística:	Física:	PG:
REYES GARCIA JAQUELINE	REGJ950206MDFYRQ05	A	0	0	0	0	0	0	0
RODRIGUEZ BECERRA ARELY ARLET	RXBA951106MMCDOR05	A	0	0	0	0	0	0	0
RODRIGUEZ CANO PALMIRA	ROCP941123MDFDNL04	A	0	0	0	0	0	0	0
VAZQUEZ BARRERA JOSE ALBERTO	VABA950113HDFZRL	A	0	0	0	0	0	0	0
ZAMORA RANGEL EDGAR ENRIQUE	ZARE941126HDFMND01	A	0	0	0	0	0	0	0
ZAMORA VEGA MARIA FERNANDA	ZAVF950530MDFMGR04	A	0	0	0	0	0	0	0
ZAMUDIO MARTINEZ MIGUEL ANGEL	ZAMM950908HDFMRG01	A	0	0	0	0	0	0	0
ZEPEDA TORRES LEZLYE MAIRANY	ZETL950818MDFPRZ01	A	0	0	0	0	0	0	0
ALARCON IBAÑEZ LUIS EDUARDO	AAIL951003HDFLBS04	B	9.4	9.2	10	10	10	10	9.7
ARENAS LARREA LAURA REBECA	AELL950731MDFRRR01	B	9.4	9.4	9.4	9.2	9.6	10	9.5
AVILA VICENTE LILIAN ARANTZA	AVIL950423MDFVCL07	B	9.8	9.6	9.6	10	10	10	9.8
BAEZ TORES CHRISTIAN EMILIO	BATC950903HDFZRH06	B	9.6	9.4	10	10	10	10	9.8
BECERRA LELLENQUIEN PABLO ANTONIO	BELP950222HDFCLB06	B	9.8	10	9.8	10	10	10	9.9
CABRERA ROJAS ADAN HAIR	CARA950403HDFBJD07	B	8.2	8.8	9	9.2	9.6	10	9.1
CHAVARRIA VALDES LUZ MARIA PAULINA	CAVL950709MMCHLZ09	B	8.4	8	9	8.6	8.6	10	8.7
GONZALEZ ROJAS STEVENN ROBERTO	GORS950203HMCNJT01	B	9	8.2	9.6	9.4	9.6	10	9.3
HERNANDEZ LOPEZ RAUL URIEL	HELRL950619HDFRPL08	B	8.8	8.8	9.2	9	9.6	9.8	9.2
HERNANDEZ RIOS CARLOS ARTURO	HERC950903HDFRSR07	B	8.8	9	9.4	9	9.4	9.8	9.2
IBARRA SALAS MARIA JOSE	IASJ950429MDFBLS08	B	9.2	9.2	10	10	10	10	9.7
MARTINES GUTIERREZ MARIA FERNANDA	MAGF950624MDFRTR09	B	9.6	9.8	9.8	10	9.8	9.8	9.8
PEÑA VELAZQUEZ ALEJANDRA	OISN951218MDFRNN00	B	9.2	9	9.8	9.6	10	10	9.6
RAMIREZ SANCHEZ ALBA DIANA	PEVA951028MDFXLL03	B	9.8	9.4	10	10	10	10	9.8
RIZO SANCHEZPAOLA	RASA950912MDFMNL02	B	8.6	8.8	9.6	9	10	10	9.3
RODRIGUEZ AGUIRRE ERICK MANUEL	RISP950728MDFZNL04	B	9.8	9.4	10	9.8	10	10	9.8
SANCHEZ ROMERO EDGAR ALEJANDRO	ROAE951206HMCDCGR18	B	9	9.2	9.8	9.2	10	9.8	9.5
SANDOVAL FERNANDEZ NADIA IVONNE	SARE950826HMCNMD03	B	8.8	8.8	9	9	10	10	9.2
TURQUIE VEGA EMILIO SAMUEL	SAFN950601MDFRDO04	B	7.6	6.8	9	8.4	9.8	10	8.6
VALLE MUÑOZ CUPER DANIEL	TUVE950718HDFRGM04	B	9.2	8.8	9.6	9	9.6	9.8	9.3
VAZQUEZ HUERTA DIANA CAROLINA	VAMC950329HZSLXP06	B	8	9.4	9.8	10	10	10	9.7
AGUILAR JAUREGUI FERNANDA DESIREE	VAHD950601MDFZRN03	C	9.8	9.4	10	9.2	10	9.8	9.7
ALATAMIRANO CUELLAR RODRIGO	AUJF950715MDFGRR08	C	9	8.4	9.6	9	10	10	9.3
AMADOR LOPEZ GILBERTO	AACR950804NDFLLD08	C	9.6	9.6	10	10	10	10	9.8
AZCONA HINOJOSA BRENDA VERONICA	AALG950819HDFMPL00	C	9.6	9.6	10	9.8	9.8	9.8	9.7
BARRIOS ALANIS VANNYA	AOHB9509016MCMCZNR06	C	8	7.2	8.4	8.6	9.4	10	8.6
DE LA CRUZ MORAN KEILA DAYANA	BAAV950703MDFRLN07	C	9.8	9.8	9.8	9.8	10	10	9.8
GALLEGOS COVARRUVIAS ANA KAREN	CUMK950627MDFRRL03	C	9.6	9	9.6	9.6	10	10	9.6
GARCIA MENDOZA TRYCIA MERCEDES	GACA950815MDFLVN07	C	10	9.6	10	9.8	10	10	9.9
GONZALEZ GUTIERREZ MARTIN YAIR	GAMT950404MDFRNR06	C	10	9.8	9.8	9.8	9.8	10	9.8
GUTIERREZ MIGUEL MARIA FERNANDA	GOMG951218HMCNTR03	C	9.6	9.2	9.8	9.6	9.6	10	9.6
GUTIERREZ SALINAS ALEJANDRO	GUMF950209MMCTGR04	C	9.2	9	9.8	9.8	10	10	9.6
HURTADO BALBUENA URIEL RAUL	GUSA950821HMCTLL04	C	10	10	10	10	9.8	10	9.9
JIMENEZ DELGADO GERARDO	HUBU950503HDFRLR08	C	9	9.2	10	9.6	9.2	10	9.5
MARQUEZ REYES JOCELIN	JIDG950825HDFMLR05	C	8.8	8.8	9.2	9.2	10	10	9.3
MUÑOZ PEREZ MONICA MONSERRAT	MARJ950802MDFRYC01	C	8.8	7.8	8.8	8.8	9.6	10	8.9

Nombre:	CURP:	Grupo:	Espa	Mate	Naturales:	HGC:	Artística:	Física:	PG:
OCAMPO GOMEZ KAREN MICHELLE	MUPM951113MDFXRN02	C	9.4	9.6	9.8	10	9.8	10	9.7
SOTENO ECHAZARRAETA ARMANDO	OAGK950728MDFCMR05	C	0	0	0	0	0	0	0
SOTENO GUERRERO JUAN ALBERTO	SOEA951001HDFTCR08	C	8.6	7.2	9.6	8.6	9.4	9.8	8.7
SUAREZ GUERRERO KEVIN	SOGJ951025HDFTRN06	C	8.2	8.8	9.2	9.4	9.2	10	9.1
CARDONA LARRAZOLO ANA KAREN	CALA950810MDFRRN01	A	0	0	0	0	0	0	0
ESPARZA CRUZ ANDREA ITZEL	EACA951115MHGSRN00	A	0	0	0	0	0	0	0
FERRUSCA SERVIN CLAUDIA	FESC941204MDFRRL08	A	0	0	0	0	0	0	0
FLORES OROZCO CYNTHIA IVETTE	FOOC950203MDFLRY06	A	0	0	0	0	0	0	0
MALAGON ROJAS FRANCIOS JAVIER	MARF950117HDFLJR04	A	0	0	0	0	0	0	0
MANRIQUE AYALA JUAN DANIEL	MAAJ951211HDFNYN04	A	0	0	0	0	0	0	0
MORA NORIEGA HELENA ESTEPHANIA	MONH950706MDFRRL02	A	0	0	0	0	0	0	0
MORALES ARIAS ERIKA	MOAE950113MMCRRR02	A	0	0	0	0	0	0	0
ORTIZ GARCIA VALERIA	OIGV950330MDFRRL02	A	0	0	0	0	0	0	0
PEREZ VELAZQUEZ KARINA	PEVK950310MDFRLR03	A	0	0	0	0	0	0	0
RAMIREZ ORTIZ SONIA	RAOS951204MDFMRN02	A	0	0	0	0	0	0	0
RENTERIA SANPEDRO ERICK EDUARDO	RESE950225HDFNRR08	A	0	0	0	0	0	0	0



ESCUELA PRIMARIA FRANCISCO DIAZ DE LEON

ACREDITACION ESCOLAR PARA EDUCACION PRIMARIA CICLO ESCOLAR 2003 - 2004

DOMICILIO: BOULEVARD PRADOS DE ARAGON S/N, FRACC. PRADOS DE ARAGON

CLAVE DEL CENTRO DE TRABAJO: 15PPR2731D

ZONA ESCOLAR: 06

MUNICIPIO: NEZAHUALCOYOTL

Nombre:	CURP:	Grupo:	Espe	Mate	Natur	Histori	Geogr	Civica:	Artistic	Fisica:	PG:
AGUILAR LOPEZ EDGAR ALAIN	SUGK950329HDFRRV05	A	10	9.8	9.8	0	0	0	9.8	10	9.8
ALDANA GONZALEZ INGRID ANDREA	AULE940629HMC GPD07	A	9.4	8.8	9.6	10	9.6	9.6	9.6	10	9.5
AMADOR LOPEZ BRENDA PAOLA	AAGI940920MDFLNN04	A	9.6	8.6	9.8	9.6	9.6	10	9.6	10	9.6
BELMONTE PORTILLO JESSICA DANIELA	AALB940810MDFMPR08	A	9.4	9.4	10	9.6	9.4	9.8	9.8	10	9.6
DIAZ DE LEON CONSTANTINO MARCOS	BEPJ940928MDFLRS03	A	9.4	9	10	9.8	10	9.8	9.8	10	9.7
DIAZ LOZANO FERNANDO EMMANUEL	DICM941116HMCZNR05	A	8.6	8.2	9.2	9	8.8	9.2	9.4	10	9
DIAZ MARTINEZ MIGUEL ANGEL	DILF941112HDFZZR05	A	9.6	9.2	10	10	10	10	10	10	9.8
GALVAN MEJIA YOSELIM RAQUEL	DIIM940907HDFZRG19	A	9.2	9.2	10	10	10	9.8	9.6	10	9.7
GONZALEZ FABILA CESAR ALEJANDRO	GAMY930714MMC LJS07	A	8.6	7.4	9	8.2	9.2	9.2	9.4	10	8.8
GUERRERO HERNANDEZ LEONARDO	GOFC940807HDFNBS05	A	8.8	8.2	9.8	9.4	9.6	9.8	9.8	10	9.4
GUERRERO MENDOZA DAFNE DENISSE	GUHL940916HDFRRN09	A	8.8	9.4	9.6	9.6	9.4	9.6	10	10	9.5
GUZMAN HERNANDEZ BRENDA SARAJ	GUMD940418MDFRNF00	A	8.2	6.2	8.6	6.2	7.6	8.6	9.2	10	6.3
HERNANDEZ ESCAMILLA ABRIL PAULINA	GUHB941129MDFZRR02	A	9.4	9	9.4	9.4	9.4	9.8	9.8	10	9.5
LAGUNAS ORNELAS KEVIN AUGUSTO	HEEA940320MDFRSB05	A	8.8	8.8	8.2	8.6	7.8	9.2	8.8	10	8.7
LOPEZ LOPEZ ITZEL STEPHANIE	LAOK940527HJCGRV07	A	10	10	10	10	9.8	10	10	10	9.9
MENDEZ MORENO LENIKA DARINKA	LOLI940728MHGPPD09	A	8.6	8.2	9.8	9.6	9.6	9.4	9.6	10	9.3
MENDOZA HERRERA MARIA FERNANDA	MEML940721MDFNRN02	A	8	6.4	8.2	7.2	7.6	7.6	9.4	10	8
MIRELES GONZALEZ MARIA CLARISSA	MEHF940326MDFNRN06	A	8	6.4	7.8	7.4	6.8	8.8	9.4	9.8	8
MORENO VALDES GEMMA ATEPHANY	MIGC940314MDFRNL03	A	10	9.8	10	10	10	10	10	10	9.9
ODRIOZOLA LOPEZ OSCAR MISAEL	MOVG940901MDFRLM04	A	7.6	7	7.4	7	7.8	8.2	9.6	10	8
PARDO LOPEZ ARTURO FEDERICO	OILO940428HDFDPS06	A	8.2	7.2	9	9.2	8.2	9.4	8.6	10	8.7
PEREZ ARRIAGA STEPHANIE DEL ROCIO	PALA940701HMCRPR02	A	9	8.6	9.4	8.8	9.2	9.6	9.8	10	9.3
PEREZ PADILLA DIEGO OMAR	PEAS940804MDFRRT08	A	8.4	7.6	9	9	8.2	9.4	9.4	10	8.8
ROJAS BARRALES JOSE EDUARDO	PEPD940723HDFRDG04	A	8.8	8	9.2	8	9	9.8	9.6	10	9
ROMERO DEL RIO ERIK RAYMUNDO	RXBE9411230HDFJRD07	A	8.2	8	8.8	9	9.4	9	10	10	9
SANCHEZ CIRIACO MIGUEL ANGEL	RORE940307HMC MXR07	A	8.8	8.6	9.2	9.4	9.6	9.8	10	10	9.4
SOSA HIDALGO BRENDA JAZMIN	SACM940531HDFNRG02	A	10	10	10	9.8	10	10	10	10	9.9
VELAZQUEZ VILLAREAL LUIS ENRIQUE	SOHB940109MDFSDR08	A	9.8	8.6	9.8	9.4	8.8	9.8	10	10	9.5



ESCUELA PRIMARIA FRANCISCO DIAZ DE LEÓN

ACREDITACION ESCOLAR PARA EDUCACION PRIMARIA CICLO ESCOLAR 2003 - 2004

DOMICILIO: BOULEVARD PRADOS DE ARAGON S/N, FRACC. PRADOS DE ARAGON
 CLAVE DEL CENTRO DE TRABAJO: 15PPR2731D ZONA ESCOLAR: 06

MUNICIPIO: NEZAHUALCOYOTL

Nombre:	CURP:	Grupo:	Espa	Mate	Natur	Histori	Geogra	Cívica:	Artistic	Fisica:	PG:
AGUILAR GONZALEZ GUADALUPE	VEVL940617HMCLLS04	A	8	7.2	8.2	8	7.4	7.4	9.8	9.6	8.2
ARELLANO FLORES KEVIN	AUGG921212MDFGND05	A	9	8	9	8.4	8.6	9.2	9.6	10	8.9
CARRERA PEREZ DUARTE JANATHAN	AEFK930312HMCRLV04	A	9	8.6	8.8	9.2	9.2	9.2	9.4	10	9.1
CASTRO ARTEAGA ESLI	CAPJ930414HDFRRN00	A	8.8	9.2	9.2	9	9.6	9	9.8	9.6	9.2
CORDOVA LOPEZ GRECIA CRISTINA	CAAE930921MDFSRS00	A	9.6	8.8	9.6	8.6	8.4	9	10	10	9.2
DEL PINO MUÑOZ DIANA PAMELA	COLG921230MDFRPR02	A	0	0	0	0	0	0	0	0	0
DOMINGUEZ SALMERON JAVIER	PIMD930730MDFNXX00	A	8.8	7.2	7.8	7.8	7.6	7	9.6	10	8.2
FLORES RODRIGUEZ ANGEL	DOSJ921210HDFMLV09	A	8.8	8.6	7.4	8	8.6	7.6	9.6	9.6	8.5
FUENTES SERVIN MARTHA BEATRIZ	FORA930821HDFLDN02	A	10	9.2	10	9.8	9.8	9.6	10	10	9.8
GARRIDO TORREZ KARINA	FUSM930718MDFNRR07	A	9.8	9.2	9	9	9.2	9	9.8	10	9.3
GONGORA ALVARADO JORGE	GATK930920MCSRRR06	A	8.8	6.4	8.4	7.6	8	8.4	8.4	10	8.6
GONZALEZ GONZALEZ GABRIEL	GOAJ931110HDFNLR04	A	9.4	9	9.8	9.2	9.4	9.4	10	10	9.5
GUERRERO HERNANDEZ DANIEL	GOGG930818HDFNBN09	A	9	7.8	8	8.2	8.6	8.6	9.4	9.6	8.6
JUAREZ CORONA MARIA FERNANDA	GUHD930520HDFRRN00	A	8.8	7.8	8.4	8	8.8	7.2	9.6	10	8.5
LOPEZ GONZALEZ RAUL ALFREDO	JUCF930814MDFRRR02	A	9.6	9.4	9	8.2	8.6	9	9.8	10	9.2
MENDEZ ARELLANO HUGO ALBERTO	LOGRS30925HMCPLN05	A	9.2	9	8.8	8.6	8.6	8.8	10	10	9.1
ORTIZ RAMIREZ SARAI	MEAH930417HDFNRG07	A	9.4	9.4	8.4	7.8	8	9.2	10	10	9
RAZO LOPEZ LUIS ANGEL	OIRS930615MMCRMRO1	A	8.2	6.8	7.8	7.8	7	8.4	9.2	9.8	8.1
RODRIGUEZ CONTRERAS JORGE	RALL921223HDFZPS04	A	8.8	8.6	8.2	8.2	8.8	7.8	9.2	10	8.7
RODRIGUEZ PUNZO PAUL DE JESUS	ROCJ931015HDFDNRO4	A	9.4	8.6	8.8	8.4	8.4	9	9.8	9.6	9
ROMAN DOMINGUEZ JOSE ANTONIO	ROPP930117HDFDNL09	A	8	7.8	6.2	7	8.4	6.6	8.2	9.6	7.4
SUAREZ GUERRERO PAMELA	RODAS31004HDFMMN06	A	9.4	9.4	9.2	9.2	9.8	9.4	9.2	10	9.4
TELLEZ AVIÑA ARITZI ELIZABETH	SUGP930614MDFRRM03	A	9.8	9.8	9.2	9.4	9.8	9.6	10	10	9.7
TERRONES CELIS EDUARDO DANIEL	TEAA931102MMLNVR01	A	9	8.4	8	7.2	8	8.6	9.8	10	8.6
VAZQUEZ TRUJILLO CHRISTYAN	TECE930413HMCRLD09	A	10	9.8	10	10	10	10	10	10	9.9
ALDANA CENTENO HUGO RUBEN	VATC930404HDFZRH09	B	9.2	9	8.8	8.8	9	9.8	10	10	9.3
AMEZCUA RINQUILLO ANNA JESSICA	AACH930703HDFLNG02	B	9.8	9.8	9.8	9.6	9.4	8.8	9.8	10	9.6
ANDRADE CADENA JESSICA	AERA930125MMCMNN00	B	8	6.2	6	5.6	5.6	7.8	8.8	9.8	7.2
BRETON ARANA SILVIA MARIA JOSE	AACJ930412MDFNDS05	B	9.4	8.6	8.4	8.4	8.4	8.6	9.8	10	8.9
BRETON SANTOS DILAN DAVID	BEAS931026MDFRRL06	B	8.6	8.4	8.4	8.2	7.8	7.8	9.2	10	8.5
CUEVAS GONZALEZ DANIEL	BESD931123HDFRNL08	B	8.6	8.8	8.2	8.6	8	9	9.6	10	8.8
GALINDO GONZALEZ OCTAVIO	CUGO930403HDFVNN02	B	8.8	8.4	8.6	9.2	8.6	9	9.8	10	9
GALLEGOS OCHOA DIEGO ARMANDO	GAGO931018HDFLNC02	B	7.6	7	6.8	7.8	6.4	8	8.8	10	7.8
HERNANDEZ LARRAZOLO HECTOR	GAOD931027HDFLCG09	B	9.2	8.6	8.2	8.4	8.69	9	9.6	10	8.9
HORTA PINTOR JIMENA SAYARET	HELH930609HDFRRC05	B	9.2	8.2	9.4	9.2	8.6	8.6	9.6	10	9.1
HURTADO BALBUENA MAHETZI	HOPJ930825MDFRNM05	B	9.4	8.8	9.2	9.4	9	9.2	9.8	9.8	9.3
LARADE LA ROSA CARLOS	HUBM930612MDFRHL00	B	8.8	7.2	8.6	8.2	8	9.2	9.6	10	8.7
LOPEZ ESTRELLA JUAN PABLO	LARC940128HDFRSR06	B	8.8	8.8	8.4	8.6	8.4	8.6	10	10	8.9
MANRIQUEZ MARTINEZ CLAUDIA	LOEJ931220HDFPSN07	B	7.8	6.2	6.2	8	6.8	7.4	9.4	10	7.7
MÉLO BENÉTEZ DANIELA MICHEL	MAMC930612MDFNRL06	B	7.8	6.8	6.2	6.8	6.8	7.8	9.4	10	7.7
MORENO MEDINA IVONNE	MEBD9306307MDFLHN01	B	7	6	7	7.4	6.8	7.4	9	10	7.5
PÉREZ VELÁZQUEZ MAGDALENA	MOMI930518MDFRVD00	B	9.8	9.4	9.8	10	9.6	10	10	10	9.8
QUILES MARTINEZ BETZABE	PEVM930722MDFRGL03	B	9.6	8.8	9.2	8.8	8.8	9.4	9.4	10	9.2
RAMIREZ BELTRAN PEDRO ANTONIO	QUMB930524MDFLRT00	B	9.6	9	9.4	9.6	9.4	9.4	10	10	9.5

Nombre:	CURP:	Grupo:	Espa	Mate	Natur	Histori	Geogra	Civica:	Artistic	Fisica:	PG:
RODRIGUEZ ABRIZ VICTOR MANUEL	RABP930728HDFMLD06	B	8.8	8.4	8.4	8	7.4	8.4	9.4	10	8.6
SOLANO GARCIA STEPHANY	ROAV921112HDFDBC06	B	6	6.2	6.6	5.8	5.8	5.6	8.6	10	6.8
TORRES CHAVEZ ARGEL	SOGS930312MDFLRT02	B	9.8	9.2	10	10	9.8	10	10	10	9.8
ZAMORA VEGA LILIAN	TOCA830619HDFRHR02	B	8.2	7.2	7.2	7.8	7.4	7.8	9.4	10	8.1
ZAMUDIO CARRILLO ANGELO	ZAVL93012MDFMGL07	B	8.8	7.6	7.8	8.2	7.4	8.2	9	10	8.3
ZAMUDIO MICHAUD ALEJANDRO	ZACA930518HDFMRN00	B	8.4	6.8	7.4	6.6	6.2	7.6	9.4	10	7.8



ESCUELA PRIMARIA FRANCISCO DIAZ DE LEON

ACREDITACION ESCOLAR PARA EDUCACION PRIMARIA CICLO ESCOLAR 2003 - 2004

DOMICILIO: BOULEVARD PRADOS DE ARAGON S/N, FRACC. PRADOS DE ARAGON

CLAVE DEL CENTRO DE TRABAJO: 15PPR2731D

ZONA ESCOLAR: 06

MUNICIPIO: NEZAHUALCOYOTL

Nombre:	CURP:	Grupo:	Espe	Mate	Natur	Histori	Geogra	Civica:	Artistic	Fisica:	PG:
ACEVEDO MARQUEZ LAURA TERESA	ZAMA931009HDFMCL04	A	10	9.2	9.6	9.8	9.8	9.4	9	10	9.7
CASTRO GUILLEN ABIGAIL	AEML920504MDFCRR09	A	9.4	9	9.6	9.2	9.4	9.2	10	10	9.4
DE LA ROSA LOPEZ DIANA PATRICIA	CXGA920807MDFSLB08	A	8	7.2	8.6	8.2	8.2	8.2	9.4	10	8.4
DIAZ DE LEON CONSTANTINO	ROLD920921MDFSPN05	A	9.2	7.8	9	8	8.4	8.8	10	10	8.9
ENRIQUEZ ZAMORA DAVID MANUEL	DICM920221MMCZNR07	A	8.6	7.6	8.6	8.6	9.2	9	9.6	10	8.9
GARCIA SANCHEZ CESAR DANIEL	EIZD920122HDFNMV04	A	8.2	6.8	8.8	8.4	8.6	8.8	8.8	10	8.5
GIL DIAZ SHARON MARLENNE	GASC901020HDFRNS04	A	6.6	6.2	8	8.4	8	7.8	9	10	7.7
GUTIERREZ GARCIA MAYRA	GIDS911028MMCLZH18	A	8.8	8	9	9.6	9.2	9.6	9.6	10	9.2
GUTIERREZ LEMUS CAROLINA	GUGM920424MDFTRY03	A	8.8	7.8	9.2	8.8	9.2	8.8	10	10	9
IBARRA SANCHEZ FERNANDO	GULC920526MDFTRM09	A	9.2	8.4	8.8	6.8	8.8	8.6	9.6	10	9
MANZANO LEON JANSEL	IASF920821HDFBNR18	A	8.4	7.8	8.2	8.2	8	8.4	9.4	10	8.5
MARTINEZ RIOS JORGE CARLOS	MALJ920605HDFNND07	A	9.8	9.6	9.8	9.6	9.4	9.4	9.8	10	9.6
MIRANDA GUTIERREZ YAMILE SAHAD	MARJ920625HMCRSR06	A	8	7.8	8.4	6.8	7.2	7.2	9.2	10	8
OSCOY GONZALEZ JOSE ALBERTO	MIGY920914MCLRTM08	A	8.4	6.6	7.8	7.4	7.8	8.6	9.8	10	8.3
PEÑALOZA CRUCES PAULINA GISELA	OOGA921126HDFSNL03	A	7.8	7.4	8.4	8.4	9.2	8.4	9	9.8	8.5
RAMIREZ GARCIA DIANA ESMERALDA	PEC920508MDFXRL01	A	9.6	8.6	9.2	8.6	9.6	9.2	9.8	10	9.3
RENTERIA SAN PEDRO IRWING	RAGD920815MDFMRN02	A	8.4	8	8.8	8.2	7.2	8.6	9.4	10	8.5
SANTILLAN LEGARIA ARTURO	RESI920924HDFNRR09	A	7.8	6.4	7.8	7.4	6.8	8.8	9.6	10	7.9
VAZQUEZ BARRERA MONSERRAT	SALAS920307HDFNGR04	A	7.2	8.4	7.6	7.8	7.2	7.8	8.8	10	7.8
ARAJA ROMERO CARLOS JACK	VABM930124MDFZRN08	B	9.2	8.6	9.4	9.4	9	9	9.8	10	9.3
COVARRUBIAS OCHOA FABIOLA	AARC920330HDFMRM08	B	6.8	6.2	7	6	7	6.8	9.2	10	7.3
ESPARZA CRUZ EMMANUEL	COOF911211MDFVCB04	B	9.8	9	9.6	9.6	9.4	9.4	10	10	9.6
GARCIA FIERROS DANIEL	EACS921115HHGSRR17	B	8.4	8.8	7.6	7.6	8.2	8.4	9.4	10	8.5
GUTIERREZ CERVANTES ANGELICA	GAFD920622HDFRRN03	B	9.4	8.6	9	8.6	8.8	8.6	10	10	9.1
IBARRA SALAS MARIA DEL ROSARIO	GUCA920903MDFTRN07	B	8.8	8	8	7.2	8.2	7.6	9.8	10	8.4
LUNA RODRIGUEZ KAREN ITZEL	IASR920125MDFBLS01	B	9.8	9.6	9.4	9.4	9.4	9.4	10	10	9.6
MENDOZA HERRERA JESUS AARON	LURK920503MDFNDR05	B	9.8	8.6	9.4	9	10	9.2	10	10	9.5
MORENO GONZALEZ ALEJANDRA	MEHJ920513HDFNRS04	B	8	6.6	8	6.8	8	7.4	9.2	10	8
NAPOLES SANCHEZ ANA LILIA	MOGA920429MDFRNL04	B	9.6	8.4	8.8	8.4	9.2	8.4	9.8	10	9
NIETO SERNA JONATHAN MIGUEL	NASAS920406MDFPNN04	B	9.4	9	9.2	9	9	9.4	10	10	9.3
PACHECO HERRERA CARLOS	NISJ920810HMCTRNO8	B	9.2	8.6	8.8	8.8	9.2	8.8	9.6	10	9.1
PEREZ CARRASCO ANGELICA	PAHC921122HDFCRR04	B	9.2	8.6	9	9	9.6	9	8.8	9.6	9.1
RAMIREZ GOMEZ MARIANA	PECA921115MDFRRN04	B	9.2	8.8	9	8.6	8.6	9.2	10	10	9.1
ROCHA ESPARZA NAYELI	RAGM920114MDFMMR02	B	8.8	8	8	7.8	7.8	7.6	9.6	10	8.4
RODRIGUEZ ROMERO CARLOS	ROEN921118MDFCSY07	B	8.2	7.6	8	7.6	7.6	8.2	9.4	10	8.3
ROSAS MARTINEZ MARLENE	RORC921003HDFDMR08	B	9.6	9	9.2	9.2	9.2	9	10	10	9.4
TORRES BARRAGAN MARIA DEL	ROMM920429MDFSRR04	B	9.8	8.6	8.8	8.6	9.2	9	10	10	9.2
ZEPEDA TORRES ANGELES	TOBC920109MDFRRR02	B	9.4	9.2	9.6	9.2	9.4	9.2	10	10	9.4



ESCUELA PRIMARIA FRANCISCO DIAZ DE LEON

ACREDITACION ESCOLAR PARA PRIMARIA CICLO ESCOLAR 2003 - 2004

DOMICILIO: BOULEVARD PRADOS DE ARAGON S/N, FRACC. PRADOS DE ARAGON

CLAVE DEL CENTRO DE TRABAJO: 15PPR2731D ZONA ESCOLAR: 06

MUNICIPIO: NEZAHUALCOYOTL

Nombre:	CURP:	Cargo:	Preparacion:	Grado	Grup
MARTINEZ LOPEZ NORMA	MALN730414MDFRPR	PROFESOR PASANTE	LIC. EN PEDAGOGIA	3	C
ORDUÑO BENAVIDES PATRICIA	OUBP700817MDF RNT	PROFESOR PASANTE	LIC. EN PEDAGOGIA	5	A
TAPIA GARCIA MARIA GUADALUPE	TAGG761019MDFPRD	PROFESOR PASANTE	LIC. EN PEDAGOGIA	5	B
VARELA HERNANDEZ JOVITA	VAHJ73022MDFRRV	PROFESOR TITULADO	LIC. EN PEDAGOGIA	4	A
MALAGON SALAZAR ROSALVA	MARS570909MDFLLS	DIRECTOR SIN GRUPO	LIC. EN EDUCACION		
MÓSCOSO OROSCO DAVID	MOOD570903HCSSRV	SECRETARIO	LIC. MATEMATICAS		
AGUILAR SANCHEZ LUCIA	AUSL670714MDFGNC	SUBDIRECTOR	NORMAL ELEMENTAL		
BARRIOS ALANIS ADA LARISSA	BAAA721113MDFRLD	PROFESOR PASANTE	LIC. EN PEDAGOGIA	6	A
BUENOS AIRES MARTINEZ	BUMS720718MDFNRN	PREFESOR PASANTE	LIC. EN PADAGOGIA	2	A
CORRO DURAN DORELY	CODD780507	PROFESOR PASANTE	LIC. EN PEDAGOGIA	3	B
GUZMAN MEDINA GABRIELA	GUMG760612MDFZDB	PROFESOR PASANTE	LIC. EN PEDAGOGIA	3	A
LOPEZ GONZALESZ GUILLERMINA	LOGG690424MMCPNL	PROFESOR PASANTE	LIC. EN PEDAGOGIA	6	B
MARTINEZ ANGELES INES	MAAI760302MHGRNN	PROFESOR PASANTE	LIC. EN PEDAGOGIA	1	A

ANEXO B
MANUAL DE USUARIO DEL SISTEMA
ABEDUL 1.0

MANUAL DE USUARIO.

SISTEMA DE CAPTURA Y CONTROL DE DATOS ABEDUL 1.0.

Requerimientos del sistema.

El sistema funciona en ambiente Windows, por lo que puede ser instalado en cualquier ordenador que cuente con este Sistema Operativo en sus versiones 95 y posteriores. debe contarse también con un microprocesador 486 en adelante, 16 Mb de memoria RAM y un disco duro de 256 Mb.

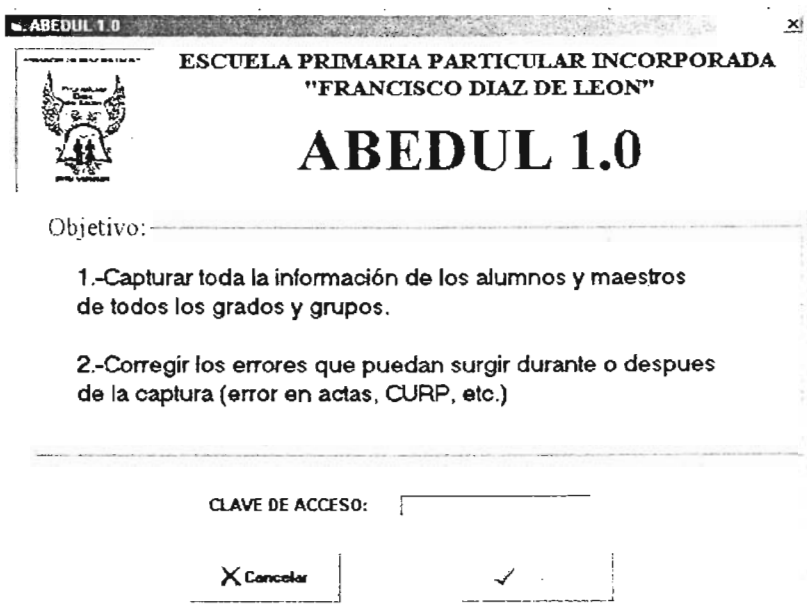
Instalación del programa.

Este programa puede copiarse como cualquier otro archivo Windows. A continuación se describe este procedimiento:

- 1.-Ingresar al Explorador de Windows
- 2.-Seleccionar la opción de Disco de 3 ½
- 3.-Dentro de esta opción, seleccionar el archivo "ABEDUL" y arrastrarlo hacia la opción de Disco duro (C:)

Acceso al programa.

Puede ingresarse al programa dando doble clic en el archivo "ABEDUL" o bien dando un clic y oprimiendo la tecla "ENTER", al hacerlo, podrá observarse la siguiente pantalla:



Esta es la pantalla de bienvenida al programa y la que nos pedirá la clave de acceso, la cual deberá teclearse en el cuadro de texto destinado a ello. La palabra clave es: "PRIMARIA", después de escribirla, oprimir el botón "ACEPTAR". Si se escribió la palabra correctamente, se mostrará la siguiente pantalla de trabajo, en caso contrario, mandará un mensaje de error y se tendrá que repetir el proceso.

Menú Principal.

Este menú será el que mostrará las diferentes herramientas para trabajar con el programa, y su pantalla será como la que se muestra a continuación:



Como puede observarse, los menús para trabajar son los siguientes:

- CAPTURA
- REPORTE
- UTILERIAS
- SALIDA
- AYUDA

Menú Captura.

Es el encargado de capturar los datos que van a ser introducidos y guardados en el sistema. La manera de trabajar entre los distintos campos será utilizando la tecla "Tabulador" (Tab), o bien situarse en los campos con la ayuda del mouse, dando clic en el campo del dato que vaya a capturarse. Existen dos submenús: Alumnos y Maestros.

Alumnos.-Dentro de este submenú, existen dos tipos de registros que se manejan para los alumnos: Datos, Calificaciones.

Datos.-Servirá para llevar a cabo la captura de los datos generales de los alumnos por grado. La pantalla Datos es similar a la que se muestra a continuación:

1º. Primer Grado

Nombre del alumno: Sexo: M o F

Apellido paterno Apellido Materno Nombre (s)

Fecha de Nacimiento: C.U.R.P.: Grupo:

dd/mm/aa

Dirección:

Calle/Número Colonia Delegación o Municipio

Entidad: Teléfono:

REGISTRO:

Los campos que conforman esta pantalla son los siguientes:

- Nombre del Alumno.-Aquí debe introducirse el nombre completo del alumno, sin abreviaciones y en el orden: *Apellido Paterno, Apellido Materno y Nombre o Nombres*.
- Sexo.-Género Masculino o Femenino, aunque basta con teclear *M* o *F*, en caso de teclear otro carácter distinto, mandará un mensaje de error para corregir.
- Fecha de Nacimiento.-Debe teclearse con el formato *dd/mm/aa*, esto es día, mes y año, separados por diagonales.
- CURP.-Se refiere a la Clave Única de Registro de Población, y debe capturarse tal y como viene en el formato oficial sin separaciones.
- Grupo.-Se refiere al grupo en donde quedo inscrito el alumno, A, B, C, etcétera.
- Dirección.-Es el domicilio del alumno, debe capturarse con el formato: *Calle número, manzana, lote, interior, departamento, etcétera; Colonia, Delegación o Municipio* al que pertenece.
- Entidad.-Estado de la república al que pertenece el dato anterior.
- Teléfono.-Número telefónico del alumno, incluyendo clave si es del interior de la República, también puede ser número celular.

El proceso será el mismo en los seis grados.

Los botones de esta pantalla de trabajo y sus funciones son:

- Nuevo.-Sirve para dar de alta a un nuevo registro en el sistema.
- Guardar.-Sirve para guardar los registros que son capturados en la Base de Datos.
- Modificar.-Sirve para modificar un dato de un registro o un registro completo.
- Buscar.-Sirve para buscar un registro ya sea para Modificarlo, Eliminarlo o simplemente para confirmar que existe.

Eliminar.-Sirve para eliminar un registro del sistema en forma definitiva, por lo que se debe estar seguro de querer hacerlo, de lo contrario se tendrá que volver a capturar el registro.

Salida.-Sirve para salir de esta pantalla de trabajo y volver al menú principal.

Anterior.-Sirve para retroceder al registro anterior al que se esta mostrando. Al llegar al primer registro, se mostrará el mensaje "Primer Alumno", con lo que se indica que ya no hay más registros anteriores en el grado en el que se este trabajando.

Siguiente.-Sirve para avanzar al siguiente registro del que se esta mostrando, al llegar al último registro, se mostrará el mensaje "Ultimo Alumno" lo que indica que ya no hay más registros en el grado en cuestión.

Calificaciones.-Será utilizado para capturar las evaluaciones finales de cada materia de cada alumno y calcular su promedio general.

Cabe mencionar que para llevar a cabo la captura de las evaluaciones, debe contarse ya con los datos de los alumnos, ya que de los datos se obtendrán el nombre y grupo en cada grado, lo que marcará la pauta para llevar el orden de qué alumno es al que se le asignaran evaluaciones y promedio para no caer en incertidumbre. La pantalla de calificaciones será de acuerdo a las materias de cada grado. A continuación se muestran estas pantallas:

Para Primero y Segundo Grados:

Calificaciones Primer Grado

Alumno: [CARLOS SANDOZ PEREZ LEJIA] Grupo: []

Promedios finales por materia

ESPAÑOL	MATEMATICAS	CONOCIMIENTO DEL MEDIO	EDUCACION ARTISTICA	EDUCACION FISICA	PROMEDIO GENERAL ANUAL
[0]	[0]	[0]	[0]	[0]	[]

Guardar Modificar Buscar CALCULAR PROMEDIO





Anterior **Siguiente**

Para Tercer Grado:

Calificaciones Tercer Grado

Alumno: Grupo:

Promedios finales por materia





ESPAÑOL	MATEMATICAS	CIENCIAS NATURALES	HISTORIA Y EDUCACION CIVICA	GEOGRAFIA	EDUCACION ARTISTICA	EDUCACION FISICA	PROMEDIO GENERAL ANUAL
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
		 Guardar	 Modificar	 Buscar	CALCULAR PROMEDIO		
Anterior		Siguiente					

Para Cuarto, Quinto y Sexto Grados:

Calificaciones Cuarto Grado

Alumno: Grupo:

Promedios finales por materia

ESPAÑOL	MATEMATICAS	CIENCIAS NATURALES	HISTORIA	GEOGRAFIA	EDUCACION CIVICA	EDUCACION ARTISTICA	EDUCACION FISICA	PROMEDIO GENERAL ANUAL
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
		 Guardar	 Modificar	 Buscar	CALCULAR PROMEDIO			
Anterior		Siguiente						

Como se observa, los campos *Nombre* y *Grupo*, son los que indicaran cual es el alumno al que deben capturarse evaluaciones finales y calcular promedio. En esta pantalla no se permite corregir Nombre o Grupo, ya que para esto existe la pantalla *Datos*. Este proceso será el mismo para los seis grados.

Los botones de estas pantallas de trabajo y sus funciones son:

Guardar.-Servirá para guardar las evaluaciones y promedio obtenido de cada alumno.

Modificar.-Este botón tiene dos funciones. Es el que permite capturar las evaluaciones de cada alumno, ya que cuando aparece la pantalla de Calificaciones, las casillas están bloqueadas y solo oprimiendo este botón, será posible hacerlo, ya que es el que las activa. También tiene la función de Modificar calificaciones y por lo tanto, promedios que ya hallan sido capturados.

Buscar.-Tiene la función de buscar a algún alumno de un determinado grado para verificar sus evaluaciones o modificarlas, pero no eliminarlas, ya que esto únicamente podrá hacerse desde la pantalla "Datos".

Calcular promedio.-Como su nombre lo indica, este botón será el encargado de obtener el promedio general del alumno.









Nota.-En caso de dejar una casilla vacía, el sistema mandará un mensaje de error, indicando que no es posible hacer esto, por lo que todas las casillas deben estar llenas para calcular el promedio. En caso de poner una calificación menor a cinco o mayor a diez, también se mandará un mensaje de error advirtiendo que no existen estas calificaciones. En caso de poner un cinco en una casilla, el sistema preguntará si es correcta esta calificación (puede darse que un alumno obtenga esta calificación en una o más materias), en caso de ser correcta, la tomará en cuenta para calcular el promedio, en caso contrario, permitirá hacer la corrección. En la captura de calificaciones será válido teclear estas hasta sus décimas, por ejemplo 8.54, 9.34, etcétera.

Anterior.-Sirve para retroceder entre los registros y observar sus evaluaciones. Esto será bueno para verificar que todos los alumnos tienen sus calificaciones y promedios de forma correcta.

Siguiente.-Tiene la misma función del botón anterior, solo que en lugar de retroceder, avanzará entre los registros existentes.

Salida.-Sirve para salir de la pantalla Calificaciones y regresar al Menú Principal.

Maestros.-Este submenú será el que llevará la captura de los datos generales del personal Directivo y Docente. Al ingresar se podrá observar la siguiente pantalla:

Captura de datos de Maestros			
Nombre del docente:		[MARTIN ADRIAN FERRER VA]	
		[25]	
		Apellido Paterno Apellido materno Nombre (s)	
C.U.R.P.:	Cargo:	Grado:	Grupo:
[MARTIN ADRIAN FERRER VA]	[Directivo de Grado]	[]	[]
Preparación Profesional: [] EN EDUCACIÓN PRIMARIA			
Grado Académico:	Estado Civil:		
[]	[]		
Dirección: []			
		Calle / Número Colonia Delegación o Municipio	
Código Postal:	Entidad:	Teléfono:	
[]	[]	[]	
Registro:			
			
Nuevo	Guardar	Modificar	Buscar
			
Anterior	Siguiente	Eliminar	Eliminar

Los campos que conforman esta pantalla son los siguientes:

- Nombre del Docente.-Aquí debe introducirse el nombre completo del Directivo o Docente, sin abreviaciones y en el orden: *Apellido Paterno, Apellido Materno y Nombre o Nombres*.
- CURP.-*Clave Única de Registro de Población* del Directivo o Docente, tal como viene en su formato oficial.
- Cargo.-Función que desempeña la persona en cuestión. En este campo debe seleccionarse uno de los 6 cargos que existen, y que son los que oficialmente reconoce la Secretaría de Educación Pública, ya que si se intenta introducir uno distinto, el sistema lo rechazará.
- Preparación Profesional.-Carrera, curso, seminario, etcétera que curso el Directivo o Docente.
- Grado Académico.-Debe seleccionarse el grado académico que ostenta el Directivo o Docente de los tres que son posible elegir en este campo y que son los que se reconocen, ya que el sistema rechazará cualquier otro que se quiera meter.
- Estado Civil.-Debe elegirse alguno de los cinco que contiene este campo, ya que no será posible introducir ningún otro.
- Dirección.-Es el domicilio del Directivo o Docente, debe capturarse con el formato: *Calle número, manzana, lote, interior, departamento, etcétera; Colonia, Delegación o Municipio* al que pertenece.
- Código Postal.-Número de código postal del domicilio.
- Entidad.-Estado de la república al que pertenecen los dos datos anteriores.
- Teléfono.-Número telefónico del Directivo o Docente, incluyendo clave si es del interior de la República, también puede ser número celular.

Los botones de esta pantalla de trabajo y sus funciones son:

Nuevo.-Sirve para dar de alta a un nuevo registro en el sistema.

Guardar.-Sirve para guardar los registros que son capturados en la Base de Datos.

Modificar.-Sirve para modificar un dato de un registro o un registro completo.

Buscar.-Sirve para buscar un registro ya sea para Modificarlo, Eliminarlo o simplemente para confirmar que existe.

Eliminar.-Sirve para eliminar un registro del sistema en forma definitiva, por lo que se debe estar seguro de querer hacerlo, de lo contrario se tendrá que volver a capturar el registro.

Salida.-Sirve para salir de esta pantalla de trabajo y volver al menú principal.

Anterior.-Sirve para retroceder al registro anterior al que se esta mostrando. Al llegar al primer registro, se mostrará el mensaje "*Primer Maestro*", con lo que se indica que ya no hay más registros anteriores en el grado en el que se este trabajando.

Siguiente.-Sirve para avanzar al siguiente registro del que se esta mostrando, al llegar al último registro, se mostrará el mensaje "*Ultimo Maestro*" lo que indica que ya no hay más registros en el grado en cuestión.

Menú Reportes.

Es el encargado de generar los reportes para poder imprimirlos. Puede utilizarse la impresora que esté conectada al equipo en el que se trabaja o en cualquier otra que sea instalada. Bastará con seleccionar qué es lo que se quiere imprimir a través de los submenús Alumnos o Maestros. Dentro del submenú Alumnos, se podrá elegir el grado que se desea imprimir. El formato que se podrá imprimir será de las evaluaciones de los alumnos. Dentro del submenú Maestros, se podrá imprimir un formato de los datos de los maestros. Ambos formatos son similares a los que pide la SEP (Secretaría de Educación Pública).

Menú Utilerías.

A través de este menú, se podrá respaldar en disco de 3 ½ la información tal como lo pide la SEP (Secretaría de Educación Pública), ya sea de Alumnos o de Maestros de acuerdo a la opción que se seleccione. Puede utilizarse un solo disco para respaldar ambas Bases de Datos, siempre y cuando el archivo no sea demasiado grande, o bien un disco para respaldo de Alumnos y otro para respaldo de Maestros.

Menú Salir.

Este Menú sirve para salir del sistema "ABEDUL 1.0".

Menú Ayuda.

Como su nombre lo indica, este menú es el que proporcionará ayuda al usuario durante el trabajo con el programa en caso de que surja alguna duda. Al entrar a este menú se podrá observar una explicación del uso del programa y de los datos que deben estar en los diferentes campos y la manera en que deben ir escritos o de la función de los botones de cada pantalla de trabajo. En otras palabras, lo mismo que usted leyó, pero en tiempo de ejecución (funcionamiento) del programa.