

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

SEDE AMBATO

Unidad de Ingeniería de Sistemas

**TESIS DE GRADO PREVIA LA OBTENCION DEL TITULO
DE INGENIERO DE SISTEMAS**

**“Estudio de Interfaces Avanzadas Hombre – Máquina y su aplicación en
la creación de una Plantilla Generadora de Guías Virtuales”**

Lorena del Carmen Chiliquinga Véjar

Ricardo Patricio Medina Chicaiza

DIRECTOR DE TESIS: Ing. MsC. Roxana Meriño

Ambato, 1998



PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

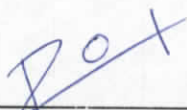
SEDE AMBATO

Unidad de Ingeniería de Sistemas

**TESIS DE GRADO PREVIA LA OBTENCION DEL TITULO
DE INGENIERO DE SISTEMAS**

**“Estudio de Interfaces Avanzadas Hombre – Máquina y su aplicación en
la creación de una Plantilla Generadora de Guías Virtuales”**

Director:

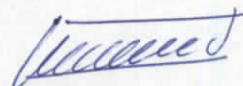


Ing. Roxana Meriño

Revisores:



Ing. Wigberto Sánchez



Ing. Rodrigo Marcial

Lorena del Carmen Chiliquinga Véjar

Ricardo Patricio Medina Chicaiza

Ambato, 1998

DEDICATORIA

Este trabajo lo dedico con todo cariño a mis queridos padres, cuyo esfuerzo y dedicación se sembraron en mí como verdaderos artífices de mi superación y cuyo amor solamente puedo corresponder de esta manera.

A mis familiares y amigos, por su constante apoyo a lo largo de mi vida y principalmente durante mi carrera universitaria.

A una persona muy especial, con quien he compartido hermosos momentos y me ha llenado con su amor.

Y sobre todo este trabajo está dedicado a Dios, quien con su infinito amor ha tomado mi mano y me ha guiado por el camino del bien.

Lorena

DEDICATORIA

A LA MEMORIA DE MI PADRE

La felicidad de una persona es cuando logramos llegar a objetivos propuestos y poder compartir dicho logro con personas que amamos; PADRE hoy que no estás aquí para compartir conmigo esta felicidad, quiero dedicarte este trabajo con todo mi cariño

También es el momento de expresar mi gratitud a mi Madre, a quien amo con todo mi ser.

De manera muy especial a mis SOBRINOS que de alguna manera se esfuerzan para poder ser personas útiles y finalmente para aquella persona que cree en mí demostrándome siempre fidelidad, lealtad, amor y respeto.

Patricio

AGRADECIMIENTO

Nuestra profunda gratitud a la Pontificia Universidad Católica del Ecuador Sede Ambato, por habernos acogido en sus aulas para formarnos como profesionales y como seres humanos.

A nuestros maestros por haber compartido con nosotros sus conocimientos y sobre todo por habernos inculcado el valor del saber.

De manera muy especial queremos reiterar nuestra gratitud a la Ing. Roxana Meriño, Directora de Tesis, por la ayuda prestada en la realización de este trabajo, así como también al Ing. Wigberto Sánchez y al Ing. Rodrigo Marcial, profesores revisores de la misma.

Lorena y Patricio ,

INDICE

CAPITULO I.- ESTUDIO MULTIMEDIA

1.1	Aspectos Generales.....	1
1.2	Campo de Aplicación.....	2
1.2.1	Enseñanza.....	2
1.2.2	Negocios.....	3
1.2.3	Hogar.....	3
1.2.4	Lugares Públicos.....	3
1.2.5	Realidad Virtual	4
1.3	Hardware Requerido.....	4
1.3.1	Dispositivos.....	5
1.3.1.1	Discos.....	5
1.3.1.2	Unidades de CD-ROM.....	6
1.3.1.3	Teclado.....	6
1.3.1.4	Ratón.....	6
1.3.1.5	Monitor.....	7
1.3.1.6	Pantallas Táctiles.....	7
1.3.1.7	Sistemas de reconocimiento de voz.....	8
1.3.1.8	Cámaras digitales.....	8
1.3.1.9	Dispositivos de audio.....	8
1.3.1.10	Dispositivos de video.....	8
1.4	Componentes de un Sistema Multimedia.....	9
1.4.1	Texto.....	9
1.4.2	Sonido.....	10
1.4.3	Gráficos e Imágenes.....	15

1.4.4	Animaciones.....	17
1.4.5	Video.....	19
1.5	Plataformas.....	20
1.5.1	Macintosh.....	21
1.5.2	Windows.....	21
1.6	Herramientas de Desarrollo Multimedia.....	22
1.6.1	Sistemas de Desarrollo.....	22
1.6.2	Software Multimedia.....	25

CAPITULO II.- PROGRAMACIÓN ORIENTADA A OBJETOS

2.1	Introducción.....	30
2.2	Aspectos Importantes.....	32
2.3	Orientación a Objetos: ¿Un asunto de moda o una revolución en la industria informática?	33
2.4	Desarrollo Incremental.....	33
2.5	La POO desde el punto de vista del usuario.....	34
2.6	Productividad.....	35
2.7	La Programacion Orientada a Objetos.....	36
2.7.1	Polimorfismo.....	38
2.7.2	Encapsulación de datos.....	40
2.7.3	Herencia.....	40
2.7.4	Otros conceptos importantes.....	41
2.8	Lenguaje Orientado a Objetos LOO.....	43
2.9	Orientación a Objetos en Multimedia.....	43
2.10	Hipermedia y Bases de Datos Orientadas a Objetos.....	48
2.11	Estándar ISO8613(Oda) de Representación Orientada A Objetos.....	49

2.12	Metodologías de Desarrollo de Sistemas Hipernmediales.....	49
2.13	Metodología EORM (Enhanced Object-Relationship Model).....	50
2.14	Metodología OOIIDM (Object Oriented Hypermedia Design Method).....	51
2.15	Conclusiones.....	54

CAPITULO III.- INTERFACES HOMBRE-MAQUINA

3.1	Aspectos Generales.....	55
3.2	Qué es una interface de usuario.....	56
3.3	Interfaz de usuario de un programa.....	57
3.3.1	Modelos de Interfaces.....	57
3.3.1.1	Modelo del Usuario.....	58
3.3.1.2	Modelo del Programador.....	59
3.3.1.3	Modelo del Diseñador.....	59
3.3.2	Aspectos de la Psicología Humana.....	61
3.3.3	Reglas para el Diseño de Interfaces de Usuario.....	61
3.3.3.1	Dar control al usuario.....	62
3.3.3.2	Ser directo.....	63
3.3.3.3	Ser indulgente.....	63
3.3.3.4	Reducir la carga de memoria del usuario, simplicidad.....	63
3.3.3.5	Buscar retroalimentación.....	66
3.3.3.6	Cuidar la estética.....	66
3.3.3.7	Ser Consistente.....	66
3.3.4	Guías De Diseño.....	67
3.4	Evolución de las Interfaces de Usuario.....	69
3.4.1	Interfaces de Líneas de mandatos.....	69
3.4.2	Interfaces de Menús.....	70



3.4.3	Interfaces Gráficas.....	74
3.4.4	Interfaces Orientadas a Objetos.....	76
3.5	Proceso de Diseño de Interfaces de Usuario.....	78
3.5.1	Reunión y análisis de la información del usuario.....	78
3.5.2	Diseño de la interface de usuario.....	78
3.5.3	Construcción de la interface de usuario.....	78
3.5.4	Validación de la interface de usuario.....	79
3.6	Técnicas avanzadas para el diseño de Interfaces de usuario.....	79
3.6.1	Presentación de la Información.....	79
3.6.2	Color.....	80
3.6.3	Audio.....	81
3.6.4	Animación	81
3.6.5	Diseño Internacional.....	81
3.6.6	Elección de controles.....	81
3.6.7	Diseño de una Interfaz Gráfica.....	83
3.7	Herramientas de desarrollo de Prototipos.....	83
3.8	Técnicas y herramientas para usar color en el diseño de la interfaz de computadora	85
3.8.1	Motivación.....	86
3.8.2	Fundamentos de color.....	87
3.8.3	Modelos mentales y el uso del color.....	90
3.8.4	Simplicidad.....	90
3.8.5	Consistencia.....	91
3.8.6	Claridad.....	92
3.8.7	Lenguaje de Color.....	92
3.9	Las Reglas de Murch.....	94
3.10	Conclusiones.....	96

CAPITULO IV.- ESTUDIO DE HERRAMIENTAS

4.1	Introducción.....	97
4.2	El nivel administrador.....	97
4.2.1	Alimentación de la información.....	98
4.2.1.1	Bases de Datos.....	98
4.2.1.2	Acces 97.....	99
	Arquitectura de Access.....	100
4.2.2	Construcción de la plantilla	103
4.2.2.1	Delphi 4.0.....	103
	Arquitectura de una aplicación Delphi.....	105
	Procedimientos creados por el usuario.....	107
	Ambiente Delphi.....	108
	Delphi y Multimedia.....	109
	Técnicas de animación.....	110
	Delphi y las Bases de Datos.....	110
	Modelos de Bases de datos.....	111
4.2.2.2	Visual Basic 5.0.....	112
	Ventajas.....	113
	Arquitectura de un proyecto de Visual Basic.....	116
	Entorno de desarrollo.....	117
	Mecánica de la escritura del código.....	121
	La Interface de Usuario en Visual Basic.....	122
	Visual Basic y las Bases de Datos.....	123
4.2.2.3	Análisis comparativo Delphi 4.0 vs Visual Basic 5.0.....	124

CAPITULO V.- EL GUIA VIRTUAL

5.1	Estructura	126
5.1.1	Nivel administrador.....	126
5.1.2	Nivel usuario.....	127
5.2	Interactividad.....	128
5.2.1	Antecedentes de la Interacción.....	129
5.2.2	Características.....	131
5.3	Forma de navegación.....	132
5.3.1	Lineal.....	132
5.3.2	Jerárquica.....	132
5.3.3	No lineal.....	133
5.3.4	Compuesta.....	133
5.4	Areas sensibles y botones.....	135
5.5	Adquisición del contenido.....	136
5.6	Consultas Virtuales.....	137
5.7	Proyecciones estadísticas de visitas.....	137
5.7.1	El método estadístico.....	138
5.8	Requerimientos y especificaciones.....	139
5.9	Portada de la aplicación.....	140
5.10	Aspectos técnicos.....	141
5.10.1	Interface de usuario.....	141
5.10.2	Interface entre Access y Visual Basic.....	142
5.11	Sistema de Ayuda.....	143

CAPITULO VI.- ANALISIS Y DISEÑO ORIENTADO A OBJETOS

6.1	Análisis Orientado a Objetos	145
6.1.1	Intereses del Usuario.....	145
6.1.2	Responsabilidades del sistema.....	145
6.1.3	Definición de clases.....	147
6.1.4	Definición del Guión.....	148
6.1.5	Grafo de Control.....	149
6.1.6	Definición de las colaboraciones.....	150
6.1.7	Documentación del análisis.....	151
6.1.7.1	Definición de clases y relaciones.....	151
6.2	Diseño Orientado a Objetos.....	155
6.2.1	Refinamiento de las clases.....	156
6.2.2	Diagrama de clases.....	159
6.2.3	Diagrama de transición de estados.....	160
6.2.3.1	Diagrama #1.....	161
6.2.3.2	Diagrama #2.....	162
6.2.4	Definición de contratos.....	164
6.2.5	Definición de las colaboraciones.....	165
6.2.6	Definición de subsistemas.....	166
6.2.7	Documentación del análisis.....	167
6.2.7.1	Diccionario de clases.....	167

CAPITULO VII.- CONCLUSIONES..... 171

BIBLIOGRAFÍA

ANEXOS

INDICE DE GRAFICOS

FIGURA 1.1	Diagrama de la interfaz MIDI.....	14
FIGURA 3.1	Ejemplo de una interface mal diseñada	56
FIGURA 3.2	Metáfora del escritorio en Windows 95.....	58
FIGURA 3.3	Metáfora del cuarto de juegos.....	59
FIGURA 3.4	Representación del modelo del diseñador. IBM (1992).....	60
FIGURA 3.5	Indicaciones visuales en Microsoft Word.....	64
FIGURA 3.6	Metáfora de la papelera de reciclaje en Windows.....	64
FIGURA 3.7	Ejemplo de mal (izquierda) y buen (derecha) diseño.....	65
FIGURA 3.8	Botones de las ventanas de Windows 3.1 y Windows 95.....	67
FIGURA 3.9	Pirámide de guías.....	68
FIGURA 3.10	Interface de Línea de mandato - Comando DIR.....	70
FIGURA 3.11	Menú de pantalla completa (Antivirus F-Prot).....	71
FIGURA 3.12	Menú de barra y menú desplegable.....	72
FIGURA 3.13	Menús en cascada de la barra de inicio de Windows 95.....	72
FIGURA 3.14	Paletas de herramientas en Corel Draw.....	73
FIGURA 3.15	Menú contextual de un icono en el escritorio de Windows 95.....	73
FIGURA 3.16	Elementos de diseño de pantalla y su percepción visual.....	80
FIGURA 3.17	Diferentes controles para los mismos datos.....	82
FIGURA 3.18	El Cubo RGB.....	88
FIGURA 3.19	Combinaciones de Colores para GUI con Despliegue Gráfico	94
FIGURA 4. 1	Objetos Principales de Access y sus relaciones.....	102
FIGURA 4. 2	Pantalla principal Borland Delphi 4.0.....	103
FIGURA 4. 3	Acceso a Bases de datos con Delphi.....	104

FIGURA 4. 4	Procedimientos en Delphi.....	107
FIGURA 4. 5	Componente Media Player – Delphi.....	109
FIGURA 4. 6	Cuadro Comparativo Paradox – dBASE.....	112
FIGURA 4. 7	Portada Visual Basic 5.0.....	114
FIGURA 4. 8	Proceso de desarrollo de software en Visual Basic.....	116
FIGURA 4.9	Barra de Menús de Visual Basic.....	118
FIGURA 4.10	Ejemplo Menú Contextual Visual Basic.....	118
FIGURA 4.11	Barra de herramientas Visual Basic.....	119
FIGURA 4.12	Project Explorer Visual Basic.....	119
FIGURA 4.13	Ventana de Propiedades Visual Basic.....	120
FIGURA 4.14	Examinador Objetos Visual Basic.....	120
FIGURA 4.15	Editor de Código Visual Basic.....	121
FIGURA 5. 1	Navegación Lineal.....	132
FIGURA 5. 2	Navegación Jerárquica.....	133
FIGURA 5. 3	Navegación No Lineal.....	133
FIGURA 5. 4	Navegación Compuesta.....	134
FIGURA 5. 5	Mapa de Navegación Guía Virtual.....	135
FIGURA 5.6	Botones utilizados en la Aplicación.....	136
FIGURA 5.7	Portada Guía Virtual Nivel administrador.....	141
FIGURA 5.7	Metáfora de la vida real utilizada en la construcción de la GUI.....	142
FIGURA 5.8	Subrutina de VB para grabar datos que serán recuperados en Access.....	143
FIGURA 5.9	Pantalla del Fichero de Ayuda de la aplicación nivel administrador.....	144
FIGURA 5.10	Sistema de Ayuda del Guía - Nivel Usuario.....	144

INTRODUCCION

Una de las actividades de mayor importancia dentro de una Institución de Servicio ya sea Pública o Privada, constituye la atención al público, de ahí la importancia de que este servicio sea óptimo y que alcance a transmitir información adecuada y oportuna al usuario.

La implementación de un guía virtual en este tipo de Instituciones supone un diseño personalizado, el cual en la mayoría de los casos toma mucho tiempo, de ahí la necesidad de contar con una herramienta que ahorre esfuerzo y optimice esta tarea

La tendencia actual en lo que a Informática respecta apunta a la utilización de nuevas tecnologías como la realidad virtual y al empleo de nuevas herramientas. Este proyecto propone la implementación de un paquete de atención al público completamente innovador, la creación de un guía virtual para una Institución, en donde el usuario dispondrá de información completa sobre toda la Institución, Departamentos, Secciones, Personal y Trámites, únicamente con la ayuda del computador la información requerida se obtendrá casi instantáneamente en el momento en que el usuario ordene su solicitud. Además se llevará un control estadístico de las visitas que el usuario realice a las que hemos denominado “consultas virtuales”, lo que marcará índices de importancia para la Institución e incluso será capaz de calcular proyecciones de visitas, en base a la información almacenada.

Nuestro objetivo principal consiste en realizar un estudio de los medios informáticos que permitan implementar una plantilla que genere guías virtuales para cualquier Institución Pública o Privada y demostrar su utilización mediante el desarrollo de un prototipo.

Básicamente el Guía Virtual es un software interactivo, tendrá elementos Multimedia y será implementado utilizando un lenguaje basado en Programación Orientada a Objetos. Por tal motivo en primer lugar se llevará a cabo un estudio Multimedia y un estudio de POO, la cual en los últimos años se ha convertido en una herramienta poderosa en el desarrollo e implementación de sistemas, ya que proporciona un entorno de trabajo agradable, confiable y rápido.

Por otro lado se llevará a cabo una profunda investigación sobre las interfaces Hombre – Máquina, para extraer de él las técnicas apropiadas en cuanto a su diseño y construcción.

Todo ello conllevará a la elaboración de un producto de calidad, que principalmente descargue el trabajo del personal humano encargado de la atención al público y sobre todo permita que el usuario conozca la Institución por sí solo.

CAPITULO I

ESTUDIO MULTIMEDIA

1.1 ASPECTOS GENERALES

Sería muy difícil encontrar una definición exacta para Multimedia, muchos la definen como una técnica, como magia e incluso como un arte.

“Multimedia es cualquier combinación de texto, arte gráfico, sonido, animación y video que llega al usuario por la computadora u otros medios electrónicos. Es un tema presentado con lujo de detalles”⁽¹⁾.

Para poder crear aplicaciones multimedia debemos emplear nuestra capacidad creativa y una tecnología muy bien seleccionada, considerando siempre que un proyecto de esta naturaleza puede en ocasiones resultar muy complicado y hacer que trabaje adecuadamente puede serlo mucho más aún. Es muy importante estudiar la utilización de las herramientas de computación y las tecnologías para lograr que en conjunto trabajen adecuadamente.

Un proyecto multimedia sea lineal o no-lineal, determinará la manera cómo el usuario navegará e interactuará con el sistema, para conseguir un buen resultado se debe prestar mucha atención al mensaje que se desea enviar, al guión que describe los parámetros del proyecto, a las ilustraciones y porsupuesto a la programación empleada.

¹ Todo el poder de la multimedia, Pg.4

Todos los elementos de multimedia se conjugan en un proyecto utilizando herramientas de desarrollo multimedia, las cuales han sido diseñadas para administrar los elementos individualmente y permitir que los usuarios interactúen con el proyecto, ofrecen facilidades para crear o editar imágenes y texto, y además tienen extensiones para controlar los reproductores de videodisco, entre otros.

El conjunto de los elementos que se reproduce y la forma en cómo se presenta al usuario es la interfaz humana, esta interface puede definirse tanto por las reglas de lo que debe suceder con los datos introducidos por el usuario, así como por los gráficos. El equipo y los programas que rigen los límites de lo que puede ocurrir es la plataforma o ambiente multimedia.

1.2 CAMPO DE APLICACION

Por todas las bondades ofrecidas por la multimedia, su ámbito de aplicación se hace cada vez más extenso y gana más espacio en todos los campos del saber que han sido asumidos por la computadora. La multimedia es muy útil cuando los usuarios necesitan acceder a información electrónica de cualquier tipo, ya que mejora las interfaces tradicionales basadas en texto únicamente logrando concentrar la atención del usuario y atrayendo su interés de una buena forma, otro punto a su favor es el hecho de que las personas que temen al computador al encontrarse con un ambiente amigable pierden el miedo y con sólo presionar un botón pueden encontrar información increíblemente interesante y útil.

Entre los campos de aplicación de la Multimedia más importantes, podemos citar los siguientes: Enseñanza, Negocios, Hogar, Lugares Públicos y Realidad Virtual.

1.2.1 EN LA ENSEÑANZA

Una de las aplicaciones mejores enfocadas de la multimedia se podría decir que es la educación, en donde es evidente la necesidad de implantar nuevas metodologías de aprendizaje que generarán enormes beneficios, con seguridad se puede decir que la Multimedia marcará la pauta en el proceso de

enseñanza, un proceso en donde los maestros son guías en vez de ser los proveedores del conocimiento.

Hoy en día los paquetes educativos se promocionan únicamente como enriquecedores del proceso, a pesar de existir miles y miles de títulos educativos para enseñanza de ciencias básicas y ciencias sociales.

1.2.2 EN LOS NEGOCIOS

En este campo son numerosos los usos que se le ha dado a Multimedia, se han desarrollado aplicaciones para capacitar personal, estudios de mercadotecnia, publicidad, catálogos, entre otras.

Es lógico pensar que esta técnica proporciona enormes ventajas operativas y económicas en su utilización, por ejemplo: tomar cursos completos de manejo de aviones, prevención contra asaltos, información sobre la venta de nuevos productos, etc.

1.2.3 EN EL HOGAR

La mayoría de usuarios que poseen un PC lo tienen con equipo multimedia, gracias a la cual pueden acceder a aplicaciones multimedia como: enciclopedias, juegos, música, turismo, entre otras. A pesar de que el número de usuarios caseros no es muy grande todavía, a medida que éste crezca, se requerirá una selección de títulos y material para satisfacer este mercado.

1.2.4 EN LUGARES PUBLICOS

Los sitios públicos son un punto importante dentro del campo de aplicación de la multimedia, centros comerciales, hoteles, museos, etc, disponen de terminales que proporcionan información y ayuda, este tipo de instalaciones reducen la demanda tradicional de personal y puestos de información, pueden trabajar las veinte y cuatro horas del día, cuando la ayuda humana está fuera de servicio.

El poder de la multimedia en estos lugares es parte de la experiencia de muchos años, increíbles efectos de sonido y movimiento proporcionan una respuesta efectiva, sorprendente y hasta divertida, en plataformas que cada vez van mejorando.

1.2.5 REALIDAD VIRTUAL

La realidad virtual se encuentra donde convergen la tecnología y la invención creativa. Los lentes, cascos, guantes especiales y extrañas interfaces humanas colocan al usuario dentro de una experiencia parecida a la vida misma.

La realidad virtual presenta una grave desventaja, requiere grandes recursos de computación para ser realista, computadoras sumamente rápidas para ofrecer efectos mucho más reales y movimientos fluidos. Recientemente se han construido videojuegos públicos especializados para ofrecer experiencias de vuelo por ejemplo. La realidad virtual es una extensión de multimedia que utiliza los elementos básicos de ésta, tales como: sonido, animación e imágenes, la realidad virtual es tal vez multimedia interactiva en su máxima expresión.

“Multimedia es revolucionaria, y cambiará nuestras vidas”⁽²⁾

1.3 HARDWARE REQUERIDO

Hay que considerar que el equipo necesario para desarrollar un proyecto depende de su contenido así como de su diseño, existe un sinnúmero de componentes de equipo para traducir las órdenes del usuario, despliegue de datos o comunicación entre colaboradores. Es importante conseguir una computadora rápida, que tenga buena memoria RAM así como también que disponga de un disco duro de buena capacidad, además se debe contar con material como efectos de sonido, música, animación y video.

² Sally Von Bargen, Rev. Century Media, Pg.21

Mientras más imágenes, sonido u otros elementos multimedia utilice, mayor será el requerimiento de memoria, además de asignar un porcentaje de ésta para guardar los archivos de trabajo que se emplean durante la proyección, el disco duro es indispensable para proporcionar un ambiente de almacenamiento adecuado.

1.3.1 DISPOSITIVOS

La cantidad y tipo de los dispositivos depende de la aplicación multimedia que se esté desarrollando, los más frecuentemente utilizados se describen a continuación.

1.3.1.1 DISCOS

Los discos pueden ser de dos clases, flexibles o magnéticos y fijos o duros. Los primeros se definen como soportes de almacenamiento de información de acceso directo, tienen una base magnética en donde la información es registrada sobre pistas concéntricas, después estos datos pueden leerse como una cadena de información binaria, los discos flexibles se hacen de diferentes tamaños y densidades para que puedan ser utilizados en diferentes unidades.

Los discos duros son medios de almacenamiento más comunes y se denominan medios de almacenamiento masivo, físicamente es una pila de platos de metal cubiertos con una material magnético sensible con una serie de cabezas grabadoras y/o sensores que flotan arriba de la superficie, se necesita de un disco duro de gran capacidad para desarrollar aplicaciones multimedia.

“Usted nunca tendrá suficiente memoria o espacio en disco”⁽³⁾

³ Todo el Poder de la Multimedia, Regla de Vaughan, Pg.35

1.3.1.2 UNIDADES DE CD-ROM

Este tipo de dispositivos se han popularizado enormemente en los últimos años, constituyen una parte fundamental del desarrollo de estaciones de trabajo multimedia y se han convertido en el medio de distribución más utilizado para proyectos grandes.

Tradicionalmente las unidades de CD han sido relativamente lentas para la transmisión de datos pero actualmente su velocidad se ha mejorado en forma considerable, además se han diseñado grabadoras de CD, gracias a las cuales podemos elaborar nuestros propios discos compactos utilizando discos vírgenes, cuya construcción es diferente a los CDs tradicionales pero pueden utilizarse de igual manera.

1.3.1.3 TECLADO

Es el dispositivo de entrada más común, permite la comunicación entre el usuario y el computador proporcionan respuestas táctiles, por lo general se conectan a la tarjeta madre del computador, los teclados estándares tienen 101 teclas y el tipo más común es el QWERTY, aunque los podemos encontrar con más o menos teclas.

1.3.1.4 RATON

Unidad de entrada constituida por una pequeña caja de aristas redondeadas de forma más o menos ergonómica, para su adaptación a la mano con una o varias teclas en su parte superior y una bola en su parte inferior. Es un medio para interactuar con el usuario en una interfaz gráfica (GUI) y su mayor ventaja es la de permitir el trabajo con ciertas aplicaciones sin que el usuario tenga que desviar la vista de la pantalla. Existen muchas tecnologías para generar localizaciones de cursor e información de órdenes

1.3.1.5 MONITOR

El tipo de monitor depende del proyecto que se vaya a desarrollar, a menudo los diseñadores de multimedia conectan más de un monitor utilizando tarjetas gráficas, varios sistemas de desarrollo le permiten trabajar con varias ventanas abiertas al mismo tiempo para que pueda dedicar su monitor a visualizar el trabajo mientras ejecuta varias tareas de edición en ventanas de otros monitores.

El número de colores máximo que se pueden desplegar en el monitor depende de la tarjeta gráfica y la cantidad de VRAM, en los PCs los monitores son básicamente de ocho bits (256 colores), pero esto puede mejorarse con tarjetas de dieciséis bits (más de 32000 colores) o tarjetas de veinticuatro bits (millones de colores), sin embargo hay que tener presente que mientras más colores se emplee más lento será el desempeño del sistema.

1.3.1.6 PANTALLAS TACTILES

Es una unidad de entrada/salida, es muy similar a la pantalla convencional en la que se ha incluido un dispositivo especial capaz de reconocer la zona misma donde se ha realizado un pequeño contacto con el dedo, son utilizadas para presentar información o realizar operaciones mediante un grupo de opciones localizadas en zonas a lo largo de la pantalla, de forma que una de ellas pueda ser reconocida por el tacto. Son monitores que generalmente tienen una cubierta de vidrio, sensible a la presión, el sistema TouchMate mide la presión aplicada, la dirección del movimiento y su desviación cuando se oprime con un dedo.

Las pantallas sensibles al tacto no se recomiendan para el uso diario, pero son excelentes para aplicaciones multimedia en lugares públicos en donde se requiere introducción de datos del público y tareas sencillas.

1.3.1.7 SISTEMAS DE RECONOCIMIENTO DE VOZ

Facilitan la interacción con el proyecto sin necesidad de utilizar las manos, estos sistemas tienen un diodo unidireccional especial, micrófonos para cancelación de ruido que automáticamente filtra los sonidos del fondo.

Los sistemas deben entrenarse para reconocer la voz de quien los utiliza y programarse con las respuestas apropiadas para reconocer ciertas frases o palabras.

1.3.1.8 CAMARAS DIGITALES

Para utilizar este tipo de dispositivos se requiere de un equipo con tecnología de video, las imágenes pueden reproducirse directamente desde la cámara a cualquier televisor estándar o por medio de un digitalizador para llevarlas a la computadora. El software controla las funciones de captura de imagen ajuste y grabación, luego ésta puede ser llevada fácilmente a cualquier aplicación.

1.3.1.9 DISPOSITIVOS DE AUDIO

Las computadoras personales no proporcionan audio de mucha calidad multimedia, para mejorarlo se pueden utilizar bocinas durante el desarrollo del proyecto. Existen muchos equipos de actualización que incluyen tarjeta de sonido y utilidades CD-ROM. WaveEdit es un sistema sencillo de producción y edición.

1.3.1.10 DISPOSITIVOS DE VIDEO

Se han constituido en el medio de comunicación de mayor impacto visual, utilizando una tarjeta de digitalización de video se puede desplegar la imagen de televisión en el computador, algunas de ellas le dan la oportunidad de capturar la imagen y convertirla en mapas de bits a color. En Windows las tarjetas de video se controlan a través de la interfaz de control de medios MCI.

Se debe tener presente que cuando empleamos video en nuestras aplicaciones se manejan enormes cantidades de datos. Un buen video mejorará considerablemente la presentación de su aplicación, es imprescindible contar con material de gran calidad, pues un video malo arruinará la presentación.

1.4 COMPONENTES DE UN SISTEMA MULTIMEDIA

La utilización y combinación de los componentes de un sistema multimedia determinará la eficacia del proyecto, por esto es muy importante conocer la información más importante relacionada con cada uno de ellos.

1.4.1 TEXTO

La forma básica de mostrar la información es a través del texto, el aprovechamiento de este recurso es decisivo en las aplicaciones que hacen un uso amplio de él, el texto puede ofrecer información con un significado poderoso, incluso una sola palabra puede encubrir varios significados, por ello es necesario cultivar la claridad y la exactitud en las palabras que escojamos. En las aplicaciones multimedia el texto aparecerá en títulos, menús y ayudas de navegación.

“ Es importante diseñar etiquetas para los títulos de las pantallas, menús y botones multimedia utilizando palabras que tengan el significado más preciso para expresar lo que se necesita decir “⁽⁴⁾

Hay que cuidar también de las propiedades que definen la apariencia del texto como son: tipo de letra, tamaño, interlineado y estilo. Se debe considerar también que existen estudios que han demostrado que frases y oraciones que incluyen mayúsculas y minúsculas son más fáciles de leer que aquellos que incluyen solamente mayúsculas.

⁴ Todo el poder de la Multimedia, Pg. 195

En multimedia, un proyecto que no utilice texto, a pesar de que su contenido no sea demasiado complejo, necesitará demasiadas imágenes y símbolos para guiar al usuario a través del proyecto, a pesar de que se puede utilizar voz y sonido para hacerlo éstos se cansarían demasiado pronto, ya que se requiere mayor esfuerzo para poner atención a las palabras que para leer el texto. Un solo elemento de texto de un menú acompañado por una sola acción requiere de poco entrenamiento y es fácil e inmediato.

Cuando se trabaja con aplicaciones en las que el usuario trabaja a tiempo real, se puede incluir gran cantidad de texto de información en la pantalla sin llegar al exceso, los usuarios podrán viajar a través del sistema de navegación, leer el texto y detenerse a analizar toda la pantalla en detalle, es importante encontrar el equilibrio entre muy poco texto que requiere de muchos cambios de página y actividad innecesaria del usuario o demasiado texto que hace que la pantalla se sobrecargue y sea desagradable, recuerde que hay que ser amable con el usuario.

Las pantallas de computadora brindan un espacio de trabajo muy pequeño para desarrollar ideas complejas, por lo que es mejor utilizar textos de alto impacto y concisos expresados en la forma más condensada posible.

Desde el punto de vista del diseño, la selección del tamaño, la fuente y la cantidad de encabezados que se coloque en la pantalla debe estar en relación con la complejidad del mensaje y su alcance.

Hay que tomar en cuenta las siguientes sugerencias cuando se va a escoger la fuente a utilizar:

- Utilizar el menor número de tipos de letra posible, variando el tamaño y peso, utilizando itálicas y negritas.
- Seleccionar fuentes que parezcan adecuadas, pida opiniones. Aprenda a aceptar la crítica
- Variar el tamaño de la fuente en proporción a la importancia del mensaje

- Ajustar el interlineado en los bloques de texto, líneas demasiado pegadas pueden dificultar la lectura
- Para resaltar el tipo de letra explorar efectos con diferentes colores y fondos
- Utilizar texto suavizado, lo cual da una apariencia más profesional
- Para resultados que llamen la atención tratar de alterar gráficamente y distorsionar el texto
- Experimentar con sombras
- Rodear los encabezados con mucho espacio en blanco

Nunca se debe suponer que las fuentes utilizadas para desarrollar la aplicación estarán en la computadora del usuario, hay que trabajar con un sistema de administración de fuentes de modo que no se corra el riesgo de que la fuente que se utilizó sea reemplazada por otra. Cuando el trabajo se va a distribuir en lugares donde no existen, o si no se tiene licencia para distribuir las, se deben construir mapas de bits de los textos con fuentes especiales, esto asegurará que la pantalla se verá bien siempre sin importar la plataforma de hardware que se utilice a las fuentes que se tienen instaladas.

La navegación a través del texto está estrechamente relacionada con la estructura general de la aplicación, desde un documento con texto el usuario puede acceder a otro relacionado con él, el uso de texto para la navegación proporciona gran libertad de diseño en el desarrollo de la aplicación, esta libertad de acceso debe estar restringida de acuerdo a los objetivos del proyecto.

En lugar de diseñar sistemas muy elaborados es aconsejable establecer vínculos directores entre las principales palabras de modo que al dar un clic sobre ellas, el usuario pueda ser transportado a un menú de temas relacionado con dichas palabras, a pesar de que esto limita la capacidad de movimiento del usuario a través del texto, el usuario no lo percibirá así y de esta manera podrá mantener un estricto control sobre los caminos de navegación y su diseño.

En caso de que se deba incorporar rutinas de búsqueda de información, se debe escoger un buen método de búsqueda, entre los principales tenemos:

- Categórica.- Selecciona o limita los campos o documentos donde el usuario va a buscar
- Relación de palabras.- Búsqueda de acuerdo a la cercanía y orden
- Colindancia.- Busca palabras que aparezcan cerca de la otra.
- Alternativa.- Aplica criterio OR
- Asociación.- Aplica criterio AND
- Negación.- Aplica criterio NOT
- Truncamiento.- Busca una palabra por cualquiera de sus sufijos posibles
- Palabras intermedias.- Busca palabras contiguas., como intermedios o iniciales

Dos términos de importancia utilizados en sistemas hipertexto son vínculos y nodos, los primeros son conexiones entre elementos conceptuales, los nodos contienen texto, gráficos, sonido e información relacionada en una base de conocimientos. Los vínculos son las guías de navegación y los menús, los nodos son temas disponibles, mensajes y elementos de contenido.

La forma más sencilla de navegar por estructuras hipertexto es con botones que permiten acceder a información vinculada que se encuentra contenida en nodos, cuando terminan de revisar la información regresa a su localización inicial.

1.4.2 SONIDO

El sonido es muy utilizado en las aplicaciones multimedia debido a que posibilita la salida de una gran cantidad de información que será captada en forma auditiva, además es quizás el elemento que más excita los sentidos, y la forma en que se lo utilice puede hacer la diferencia entre una presentación ordinaria y una espectacular y profesional.

En una aplicación se pueden utilizar dos tipos de sonido, el sonido digital y el sonido MIDI. La interface digital de instrumentos musicales MIDI es un estándar de comunicaciones desarrollado a principios de los años ochenta para instrumentos musicales electrónicos y computadores, permite que la música y los sintetizadores de sonido de diferentes fabricantes puedan comunicarse entre sí enviando mensajes a través de cables conectados a dispositivos,

MIDI proporciona un protocolo para pasar descripciones detalladas de una partitura musical, pero los datos MIDI no son digitalizados, son una representación "taquigráfica" de la música almacenada en forma numérica.

En cambio los datos de audio digital son la representación real de un sonido almacenado en forma de miles de números individuales llamados muestras, estos datos representan la amplitud instantánea de un sonido en periodos pequeños de tiempo. Debido a que dependen del dispositivo, los sonidos de audio digital suenan igual todas las veces que se tocan, pero esta consistencia tiene un precio, grandes archivos de almacenamiento.

MIDI tiene ciertas ventajas sobre los ficheros de audio digital, son más compactos y su tamaño es menor por lo tanto independiente de la calidad de reproducción, por su tamaño no emplean mucha memoria RAM, espacio de disco o recursos de la unidad central de proceso. Eventualmente los ficheros MIDI pueden sonar mucho mejor que los de audio digital si la fuente de sonido es de alta calidad.

También se puede cambiar el tamaño de un archivo MIDI variando su ritmo, sin que esto afecte el tono de la música ni baje la calidad de audio. Los datos de este tipo de ficheros son completamente editables hasta el nivel de una nota individual y se pueden manipular hasta sus detalles más pequeños. Sin embargo debido a que estos ficheros no son de sonido, su reproducción será precisa únicamente si

en medio de reproducción es idéntico al medio de producción, además no puede utilizarse para la reproducción de un diálogo hablado.

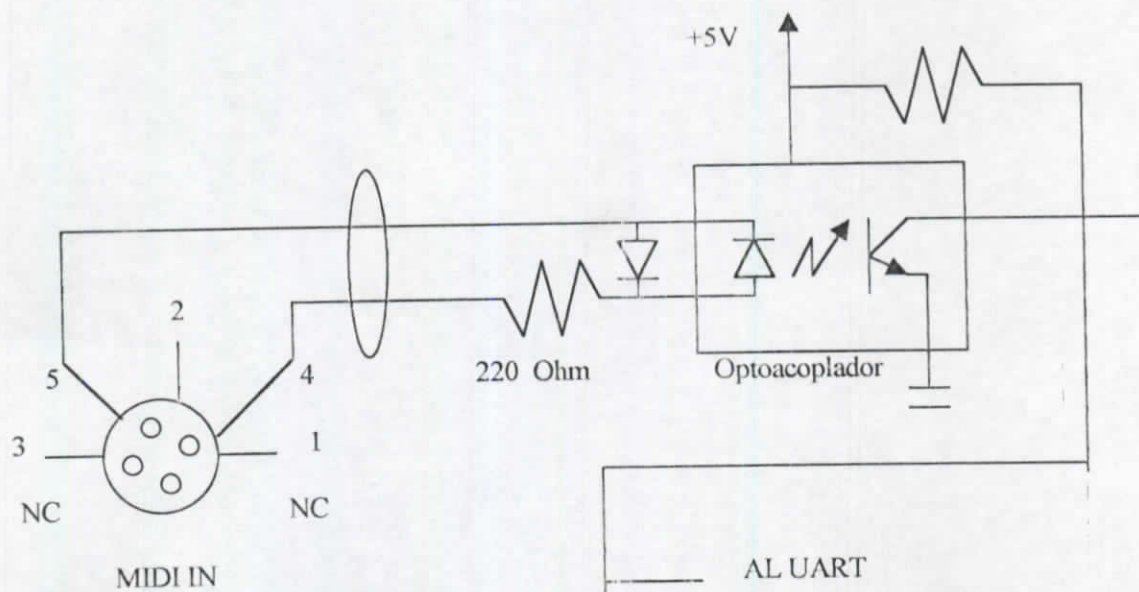


FIGURA 1.1 Diagrama de la interfaz MIDI

La ventaja más importante del fichero de sonido de audio digital es su consistente calidad de reproducción, siempre sonará tan bien al final como al principio, además está disponible una sección más amplia de programas de aplicación y su preparación no demanda de conocimientos musicales.

La mayoría de los desarrolladores de multimedia graban su material de sonido en casetes como primer paso en el proceso de digitalización, luego puede hacer varias tomas del mismo sonido o voz, escucharlas todas y decidir cual es la mejor para digitalizarla, grabar en medios económicos en lugar de hacerlo en el disco duro evitando llenarlo de basura, si la aplicación requiere sonido digitalizado de calidad entonces se debería alquilar un estudio de grabación, grabar un sonido de alta fidelidad es un arte.

También es importante contar con un buen software para edición de sonido como por ejemplo Stereo Studio de MediaVision o WaveEdit, además actualmente existen muchas herramientas de tipo shareware. El reto más importante es sincronizar los elementos de sonido con las imágenes visuales ya que no siempre la velocidad de la máquina a utilizar es la misma de la máquina en que fueron creadas.

1.4.3 GRAFICOS E IMAGENES

Estos elementos contribuyen significativamente en el impacto visual de las aplicaciones multimedia, los especialistas en el diseño gráfico de la aplicación tienen en sus manos la responsabilidad de una buena respuesta visual e incluso de la factibilidad de la interfaz usuario de la aplicación, hay que considerar a la pantalla como un medio poderoso y seductor para plasmar la creatividad.

No importa si se utiliza plantillas y pantallas predefinidas por el sistema de desarrollo o si se emplea recortes u objetos creados por terceros, aunque se copie la apariencia y manejo de otro proyecto siempre habrá un momento inicial en el que nos encontraremos con una pantalla en blanco.

En el mercado se puede encontrar muchos sistemas de desarrollo que permiten crear objetos gráficos de multimedia directamente en la pantalla, cuando se trabaja con animaciones, éstos ofrecen toda una utilería para activarlos, como un lenguaje de programación o funciones especiales.

Las imágenes fijas pueden ser grandes, pequeñas o inclusive ocupar toda la pantalla, tener cualquier forma y colores. Los mapas de bits se suelen ocupar para tener imágenes fotorealistas y dibujos complejos que requieren detalles finos, probablemente las imágenes fijas se constituyen en el elemento más importante del proyecto, las habilidades de desarrollo de arte gráfico y el diseño son vitales ya que el usuario juzga el trabajo sobre todo por el impacto visual de la aplicación.

Principalmente existen dos tipos de imágenes: las de mapas de bits y las vectoriales.

Las imágenes de mapas de bits se definen como una copia en memoria de los que se desea representar en pantalla, es una simple matriz de información que se describe como puntos individuales que son el punto de resolución más pequeño de la pantalla, se conocen como pels o pixels, los cuales pueden estar encendidos o apagados, ser blancos o negros o tener alguna tonalidad de color, para crear un archivo de este tipo existen tres formas:

- Crearlo desde cero con un programa de pintura
- Capturar una imagen desde la pantalla con algún software para el efecto y luego pegarlo en un programa de pintura o en su aplicación
- Capturar un mapa de bits desde una fotografía, arte gráfico o televisión utilizando un digitalizador o dispositivo de captura de video.

Una imagen de mapa de bits es guardada en memoria y actualizada cada sesentavo de segundo o más rápido dependiendo de la velocidad de barrido del monitor, la mejor forma de tener mayor poder creativo al manipular mapas de bits es utilizando un programa de edición de imágenes, que no solamente permiten retocar la imagen sino realizar interesantes trucos con ella como por ejemplo la morfosis que permitirá crear interesantes y a veces desconcertantes transformaciones animadas

Las imágenes vectoriales almacenan comandos para el procesamiento gráfico y contienen instrucciones para el trazo de figuras geométricas, la mayoría de sistemas de desarrollo multimedia proporcionan líneas, rectángulos, óvalos, polígonos y texto dibujados con vectores. Los programas de diseño asistido por computadora y animación en tres dimensiones utilizan ficheros vectoriales, éstos se dibujan en la pantalla de la computadora empleando una fracción del espacio de la memoria requerida para dibujar y almacenar el mismo objeto en un fichero de mapa de bits

El color es un componente vital de la multimedia, la administración de este elemento es por un lado subjetivo y por otro es un ejercicio técnico, la selección de los colores y sus combinaciones adecuadas

en el proyecto puede requerir de muchos intentos hasta llegar a obtener el deseado, pero la descripción técnica de un color puede expresarse en término de valores físicos conocidos. El color es la frecuencia de la onda de luz de la angosta banda del espectro electromagnético que puede ver el ojo humano.

El color en la computadora debe ser utilizado de una manera inteligente. Puesto que los receptores del ojo son sensibles a las luces de color rojo, verde y azul, haciendo combinaciones de estos tres se podrán interpolar combinaciones intermedias. Los monitores de las computadoras son fuentes de luz, en la parte trasera del espejo del monitor se encuentran miles de puntos de un compuesto químico de color fosforescente que son bombardeados por electrones que pintan la pantalla a grandes velocidades, cada punto tiene un diámetro de .3 mm o menos. Los puntos rojos, verdes y azules se encienden cuando el rayo de electrones choca contra ellos.

Las paletas de colores también son elementos importantes, son tablas matemáticas que definen el color del pixel desplegado en pantalla, cuando se cambia de colores de paleta y éstos se vuelven a mapear se produce un parpadeo molesto de colores extraños cuando se cambia de una imagen vieja a una nueva.

Para resolver este problema se debe tender a mapear todas las imágenes con una sola paleta compartida.

En cuanto a los formatos de los ficheros de imagen que se utilizan con mayor frecuencia tenemos los DIB, BMP, GIF, JPG, PCX, para mapas de bits, y CDR, CUR, CAD, WMF para vectoriales.

1.4.4 ANIMACIONES

La animación es posible gracias a un fenómeno biológico conocido como persistencia de la visión, se define como el proceso de obtener movimiento a partir de una secuencia de imágenes. Las aplicaciones multimedia hacen un extenso uso de las animaciones para conseguir ese indispensable

efecto de dinamismo y movimiento. Se puede animar un proyecto completo o una parte de él acentuando ciertas cosas y sobre todo dándole más vida. Efectos visuales como por ejemplo: transiciones, desvanecimientos, acercamientos y disolvencias se pueden lograr con la mayoría de los paquetes de desarrollo y algunos pueden usarse para lograr una animación rudimentaria.

Gracias al fenómeno conocido como persistencia de la visión, se hace posible que una serie de imágenes que cambian muy ligera y rápidamente parezcan mezclarse creando la ilusión de movimiento. La televisión utiliza este principio, construye treinta cuadros completos o fotografías por segundo, la velocidad con la que cada cuadro reemplaza al otro hace que se mezclen en un movimiento impecable.

Existen muchos principios de animación que se pueden emplear para incluir animaciones en un proyecto multimedia, por ejemplo para lograr que un objeto viaje a través de la pantalla solamente hay que moverla varios pixeles y reproducir los cuadros a una velocidad adecuada.

Otra técnica de animación muy famosa que ha sido empleado por muchos años en los estudio Disney, consiste en emplear una serie de gráficos progresivos diferentes de cada cuadro de película, así un minuto de animación puede requerir de hasta 1440 cuadros distintos. El trabajo artístico de una animación requiere empezar con cuadros clave, el primero y el último en acción, las series de cuadros entre los cuadros claves se dibujan en un proceso llamado tweening, primero requiere calcular el número de cuadros y luego se debe bosquejar a lápiz las series de figuras progresivamente en diferentes cuadros, mientras avanza el tweening la secuencia de la acción se verifica pasando las hojas a través de los cuadros, luego se juntan y se graban para crear una prueba, continuidad y temporización.

Existen además programas de computación que emplean la misma lógica de los procesos de animación a cuadros, empleando por ejemplo técnicas de capas. Se puede establecer velocidades propias por

cuadro de computadora, pero las velocidades a la que se efectúen los cambios y se refrescan las pantallas dependerá de la velocidad y poder del equipo.

Es importante recordar que mientras más pequeño sea el objeto más rápido se moverá. Un efecto interesante que se puede emplear para dar más vida a una aplicación es la morfosis, con la cual podemos hacer que una imagen se transforme en otra, entre las principales herramientas que permiten lograr este tipo de efectos tenemos: Grypo, Morph, Metafilo, entre otras.

Los formatos más frecuentemente utilizados para trabajar con ficheros de animación son MMM, FLI, FLC, PICS, AVI, etc.

1.4.5 VIDEO

El video es el elemento más nuevo que se ha integrado a la multimedia, constituye una de las herramientas más poderosas para acercar al usuario a la realidad, cuando las secuencias de video son

bien escogidas y por supuesto bien ejecutadas pueden contribuir sustancialmente al proyecto, el video digital es una de las facetas más prometedoras de la multimedia, cuando es utilizado se pueden presentar mensajes en forma efectiva, los usuarios tendrán a retener una mayor parte de la información que ven.

De todos los elementos de la multimedia, el video es el que exige más requerimientos de la computadora y especialmente de memoria, uno de los mayores esfuerzos de investigación multimedia consiste en la compresión de las imágenes digitales de video para llevarlas a dimensiones manejables. Para incluir video en un proyecto se pueden utilizar filmaciones caseras o también se pueden adquirir secuencias grabadas específicamente para el efecto por ciertas empresas, sin embargo esto último en ocasiones puede resultar demasiado costoso y además puede tomar un largo tiempo hasta obtener todas las licencias y permisos respectivos para utilizarlas.

La señal de video se entrega a través del conector de video de entrada (in) de la video grabadora y se guarda en una cinta magnética, el audio se graba en una pista separada en línea recta en la parte superior de la cinta de video, en la parte inferior de la cinta hay una pista de control que contiene los pulsos que regulan la velocidad. Cuando se produce video se deben tener en cuenta los estándares existentes: NTSC, PAL, SECAM, puesto que estos estándares no son intercambiables hay que tener cuidado dónde y cuando se los utiliza. Cada sistema se basa en un estándar distinto que determina la forma en como se codifica la información.

Para poder trabajar con video en nuestra computadora debemos disponer de una tarjeta capturadora de video, la cual permitirá capturarlo, editarlo o digitalizarlo en cuadros de video, así como también reproducirlos a través de fuentes analógicas, algunas tarjetas de video permiten incluso capturar audio y administrar sonido.

También es necesario hablar del color en un video, lo primero que hay que decir es que la reproducción y despliegue de color en un televisor es distinto en un monitor de computadora, puesto que se utiliza video con componentes RGB, sus colores son más puros y exactos que los de la televisión, ya que ésta emplea un conjunto de colores limitado y niveles de brillantez restringidos.

Para incluir video en toda la pantalla de la aplicación en tiempo real, se necesita invertir en buenos equipo y contar con programas especializados, en cuanto a los formatos utilizados para este tipo de archivos, estos formatos constituyen diferentes medios de poner una señal de video en una cinta, entre los principales tenemos VHS y YUV.

1.5 PLATAFORMAS

La plataforma que se escoja para desarrollar el proyecto de multimedia, depende de la preferencia personal, requerimientos de distribución, tipo de material y contenido del proyecto. Para decidir si se trabaja en plataforma Windows o Macintosh, se pueden considerar muchos criterios, la gran desventaja

de Macintosh es que un proyecto que haya sido desarrollado en esta plataforma está condenado a ejecutarse en ella, pero el trabajo es considerado más sencillo que en Windows. Sin embargo actualmente se han lanzado al mercado tantos paquetes de multimedia bajo Windows que todo apunta a que muy pronto será equiparada esta ventaja.

1.5.1 MACINTOSH

“Desde su lanzamiento, la Macintosh ha sido, por definición, una computadora multimedia”⁽⁵⁾

Todas las computadoras Macintosh pueden reproducir sonidos incluso algunas de ellas poseen equipo y programas de digitalización de sonido y video. Estas computadoras requieren de un ratón, su sistema operativo permite durante el proceso de producción, el almacenamiento y recuperación de archivos de información y gráficos. La serie PowerPC de Macintosh soporta prácticamente todas las impresoras, tarjetas de red y otros equipos accesorios y comparte datos.

La serie quadra es actualmente la tecnología más avanzada que ofrece Macintosh, tiene capacidades de video de ocho bits expandible a dieciséis, también poseen ranuras de expansión interna y para conexión directa con el procesador, también cuentan con tarjetas complementarias para las ranuras que permiten agregar un monitor y tener acceso en tiempo real y una producción de audio de alta calidad.

1.5.2 WINDOWS

La plataforma PC para multimedia es llamada MPC, no es una unidad de equipo en sí misma, sino más bien un estándar que incluye las especificaciones mínimas para hacer una computadora basada en microprocesadores, Intel es una computadora multimedia. Las especificaciones mínimas para una MPC son las siguientes:

- RAM de 8 MB

⁵ Todo el Poder de la Multimedia, Pg.57

- Procesador 486SX
- Almacenamiento magnético, unidad de disco flexible y disco duro
- Requerimientos de almacenamiento óptico, unidad de disco compacto con velocidad de 300K/seg
- Requerimientos de audio de CD-ROM con salidas CD/D/A y control de volumen
- Requerimientos de video, monitor de color con resolución de 640 x 480.
- Teclado tipo IBM
- Requerimientos de E/S, puerto serial asíncrono, puerto paralelo bidireccional.
- Software de sistema: compatibilidad binaria.

1.6 HERRAMIENTAS DE DESARROLLO MULTIMEDIA

Debido a las enormes posibilidades que ofrece un proyecto multimedia, existe un numeroso grupo de herramientas y sistemas de desarrollo, cada uno de los cuales tiene características bien definidas y cuya clasificación se presenta a continuación.

1.6.1 SISTEMAS DE DESARROLLO

Las herramientas multimedia brindan un marco esencial para organizar y editar los elementos de un proyecto incluyendo gráficos, sonidos, video y animaciones, estas herramientas se utilizan para diseñar interactividad y las interfaces de usuario. Los programas de desarrollo de multimedia brindan un ambiente integrado para unir el contenido y las funciones del proyecto, proporcionan facilidades para crear, editar e importar tipos específicos de datos, incorporar datos en la secuencia de reproducción y proporcionar un método estructurado o lenguaje para responder a las acciones del usuario, entre los principales sistemas de desarrollo para aplicaciones multimedia tenemos:

- Animaciones
- Presentaciones
- Producción de video

- Simulaciones
- Prototipos y visualizaciones técnicas
- Guías interactivas
- Discos de demostración

Existen muchos criterios de acuerdo a los cuales se podrían clasificar las herramientas de desarrollo multimedia, una de ellas es:

1. Herramientas basadas en tarjetas o páginas

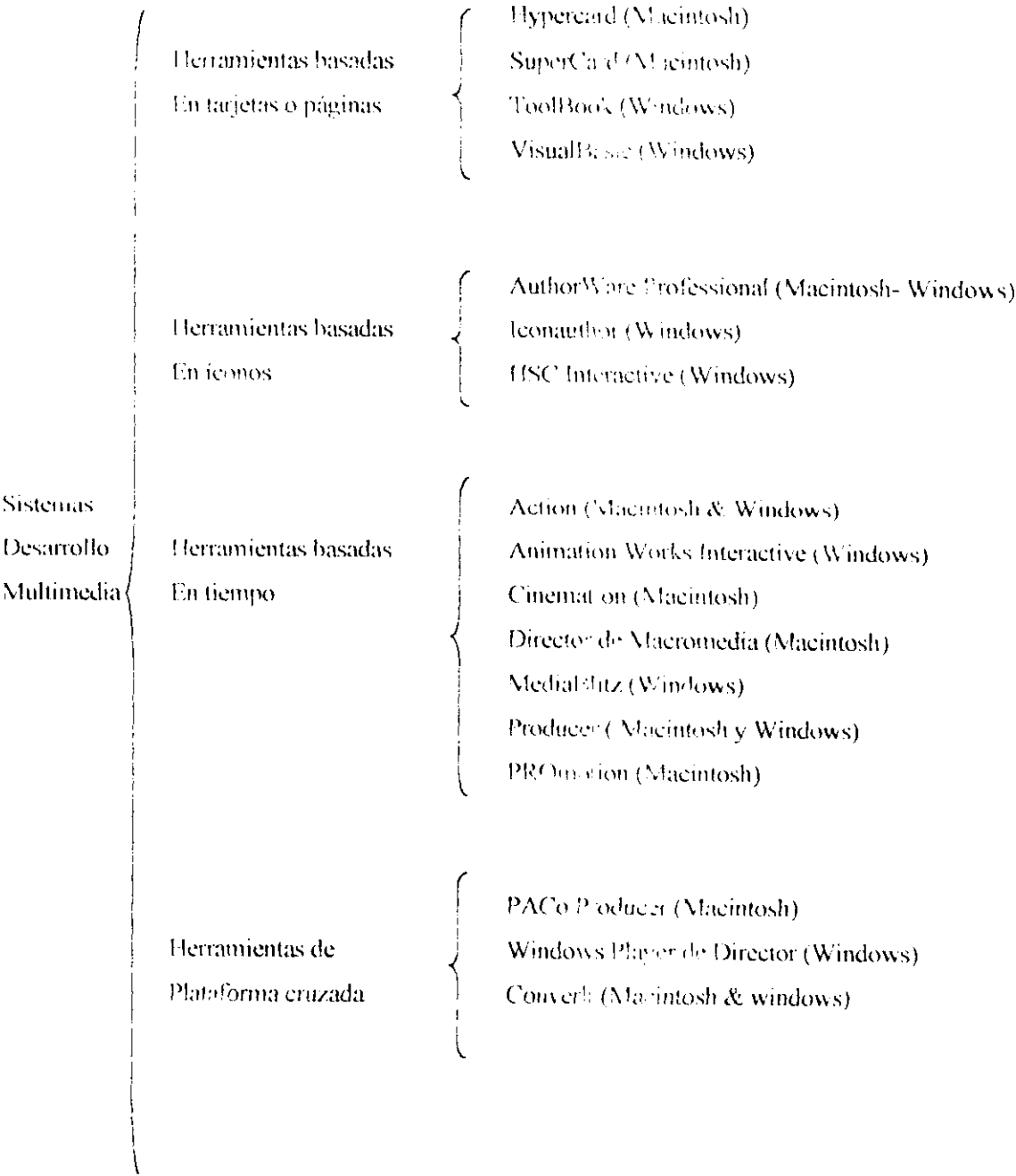
Los elementos se encuentran organizados en páginas, como lo están en un libro y en donde cada libro puede contener miles de páginas. Este tipo de herramientas son adecuadas cuando gran parte del contenido consiste en elementos que pueden verse individualmente, los sistemas de desarrollo permiten ligar estas páginas en secuencias organizadas, pudiendo saltar de un sitio a otro, además permiten reproducir elementos de sonido, ejecutar animaciones, etc.

2. Herramientas basadas en íconos controladas por eventos

Los elementos se organizan como objetos en un marco estructural o proceso, simplifican la organización del proyecto y siempre despliegan diagramas de flujo de actividades junto con vías de bifurcación.

3. Herramientas basadas en tiempo y de presentación

Los elementos y eventos están organizados en una línea de tiempo con resoluciones tan altas como un treintavo de segundo, son adecuadas cuando el mensaje tiene principio y fin, incluso permiten saltar a cualquier localización agregando de esta manera un mayor control de navegación interactivo.



1.6.2 SOFTWARE MULTIMEDIA.- HERRAMIENTAS BASICAS

Las herramientas usadas para crear y editar elementos multimedia tanto en plataforma Macintosh como Windows, soportan los sistemas de desarrollo descritos en el numeral anterior, estas herramientas se pueden clasificar de la siguiente forma:

1 Pintura y dibujo

Son uno de los elementos más importantes, ya que el impacto gráfico es siempre el de mayor influencia en el usuario. Algunas aplicaciones combinan capacidades de dibujo y pintura pero otros únicamente pueden importar imágenes de mapas de bits, para elegir un buen paquete de dibujo debemos asegurarnos de que tenga:

- Una interfaz gráfica intuitiva con menús desplegables, barras de estado, control de paleta y cuadros de diálogo.
- Dimensiones escalables
- Herramientas de pintura
- Soporte para fuentes de texto
- Acercamientos
- Buena administración de color y degradados
- Buena capacidad de importación y exportación de archivos para formatos de imágenes.

2 Cad y 3D

Permiten manipular matemáticamente imágenes de diseño asistido por computadora, dimensionarlas y si poseen medida de profundidad se puede darles vueltas en el espacio y aplicarles luz, sombra e incluso verlos desde diferentes ángulos.

3 Edición de Imágenes

Son herramientas especializadas y poderosas para realzar y retocar bitmaps. Pueden crear imágenes desde cero o digitalizarlas. Entre las principales características de estos paquetes tenemos:

- Controles de imagen, brillo, contraste
- Introducción directa de imagen desde digitalizadores y fuentes de video
- Capacidad de muestrear y redimensionar imagen
- Múltiples estilos de letra
- Transformaciones geométricas

4 OCR y Texto

Permiten incorporar material impreso transformándolos en material electrónico, es un software de reconocimiento óptico de caracteres que convierte caracteres de mapas de bits en texto ASCII reconocible electrónicamente.

5 Edición de sonido

Permiten editar sonido MIDI y digital, dibuja la representación de un sonido, en forma de partitura o de ondas y realizar las principales operaciones de edición con ellas.

6 Video y producción de películas

La mayoría de herramientas de desarrollo adoptan un enfoque por cuadro o una orientación de objetos para su animación, permiten crear, editar y presentar segmentos de video digitalizado en movimiento.

7 Accesorios

Estos accesorios tienen como misión hacer fácil y confortable su vida en la computadora, por ejemplo un capturador de pantallas, gracias a la cual podemos captar una parte de un mapa de bits colocarla en el portapapeles y moverla hacia otra aplicación sin necesidad de procesos complicados. Otro accesorio útil es el convertidor de formatos que trabaja con archivos de imágenes principalmente.

PINTURA Y DIBUJO	
Corel Draw	Illustrator
Color Studio	ImageStudio
Designer	Professional Studio
DeskDraw	SuperPaint
Hardvard Graphics	PixelPaintPro
CAD y 3D	
AutoCad	MiniCad
3-D Studio	Three-D
ClarisCad	StrataVision
Super3D	
MacroModel	
EDITORES DE IMAGENES	
Corel Studio	Picture Publisher
Gallery Effects	PhotoStyler
Composer	
Ofoto	
Photoshop	
OCR - TEXTO	
Perceive	
OmniPage	
TypeAlign	
Typestry	
TypeStyler	

FEDORES DE SONIDO	
AudioShop	Lincore
AudioTrax	WaveXtr
SoundDesigner II	
SoundEditPro	
MT ProStudio	
FEDORES DE VIDEO	
Animator	VideoShop
Morph	VideoVision
MotionMaker	Scene Motion
Premiere	VideoSignet
Super Video	
ACCESORIOS	
Touch	
Video Disc	
U-Jack	
FreezeFrame	
ZoomIt	
ShoeBox	
ImagePaint	
StopMovie	

CAPITULO II

PROGRAMACION ORIENTADA A OBJETOS

2.1 INTRODUCCION

Uno de los temas más comentados actualmente en el campo de la informática es sin duda la tecnología orientada a objetos, específicamente los Sistemas de Programación Orientados a Objetos (SPOO).

Los SPOO prometen grandes alcances y beneficios para el diseño, desarrollo y mantenimiento de software, ofreciendo una solución a largo plazo a problemas que han existido desde los inicios del desarrollo de software: falta de portabilidad, reutilización de código, complejidad para modificar y actualizar código, largos ciclos de desarrollo y técnicas de codificación no intuitivas.

Para facilitar el desarrollo de un SPOO se utilizan lenguajes de programación orientados a objetos (LOO), los cuales deben cumplir con tres características básicas: deben estar basados en objetos, en clases y ser capaces de heredar clases. Muchos lenguajes cumplen una o dos de estas características, pero en realidad son pocos los que cumplen con los tres criterios, tal es el caso de Smalltalk, C++, Actor, etc. El problema principal es que generalmente no soportan la herencia. El desarrollo de un SPOO implica un cambio total en la filosofía de analizar, diseñar y programar.

2.2 ASPECTOS IMPORTANTES

Muchas personas hoy en día hablan de la programación orientada a objetos como la salvación de la ingeniería del software moderno. Sin embargo la POO debería ser entendida como el único camino posible a seguir para la realización de programas en la actualidad, para poder llegar a tal conclusión

solamente basta con fijarse en los requisitos que exige la realización de un proyecto de software en la actualidad.

Por un lado están las ingentes cantidades de información a tratar en todos ellos, por otro lado, el equipo de personas que intervienen es bastante numeroso y todos deben de poder utilizar el código que el resto de miembros del equipo han realizado. En tercer lugar, los programas actuales ocupan miles de líneas, hay que seguir reglas muy estrictas que permitan construir programas robustos en cuanto al uso de sus estructuras de datos. Otro de los aspectos que hay que tener en cuenta es el de la reutilización, cualquier programa ha de estar lo suficientemente bien diseñado para que las partes básicas del mismo puedan servir en la realización y construcción de otros, ya que no nos podemos permitir el "lujo" de estar perdiendo tiempo construyendo una y otra vez las mismas cosas, hay que aprovechar el código que ya está hecho y que además pueda ser modificado o actualizado, sin que para ello haya que hacer modificaciones en las partes del programa que lo utilizan.

Estas son las principales características que exigen los proyectos de software de la actualidad, negar alguna de ellas no es más que engañarse, atrás quedaron los proyectos en los que un programador podía abarcarlo, el ejemplo más claro lo tenemos en el campo de la programación de juegos, antes bastaba con un programador, un grafista y ocasionalmente un músico para producir un sensacional juego, sin embargo, ahora para realizar un juego son necesario cuatro programadores como mínimo, los cuales deben estar colaborando continuamente en la creación de la obra. Es por ello que no se puede acudir a la programación tradicional para la construcción de programas en la actualidad, ya que esta, no se diseñó para dar soporte a grandes desarrollos de software con continuas actualizaciones y con multitud de equipos de desarrollo destinados a un campo específico del proyecto.

El propósito fundamental de la Programación Orientada a Objetos es el de permitir a los programadores gestionar y comprender programas más grandes y complejos, la clave para alcanzar esta meta es el objeto.

En esencia la Programación Orientada a Objetos implica la creación de objetos, los cuales se combinan y encapsulan tanto los datos como el código que opera sobre dichos datos. Los objetos pueden contener tanto elementos públicos como elementos privados, cuando un elemento de un objeto es privado, solamente los elementos de ese objeto pueden tener acceso a él, los elementos públicos son de libre acceso para cualquier otra parte del programa.

Al usar elementos privados es posible controlar estrictamente la forma en que se accede a un objeto, la ventaja de utilizar objetos es que cuando se aplican correctamente, un objeto es una entidad lógica que resulta más sencilla de entender y de manejar que los distintos elementos que forman el objeto.

Otra característica importante de la POO es que permite crear una jerarquía de objetos, pasando de los más general a lo más específico, en esta jerarquía cada objeto hereda características de los que la anteceden, la posibilidad de crear estructuras jerárquicas permite al programador organizar cuidadosamente las distintas partes del programa en unidades claras e independientes unas de otras.

2.3 ORIENTACIÓN A OBJETOS: ¿UN ASUNTO DE MODA O UNA REVOLUCIÓN EN LA INDUSTRIA INFORMÁTICA?

En algunas encuestas realizadas en el año 1995 se puede observar una disminución del interés de los departamentos de informática en la tecnología de objetos. Sin embargo, mientras los avances de la Orientación a Objetos en la informática de gestión se ven relativamente estancados, en otros sectores, tales como Diseño Asistido por Computadoras, Sistemas Expertos o desarrollos de Interfaces Gráficas de Usuario, la tecnología de objetos es algo totalmente asumido y los avances logrados en estos campos, serían imposibles sin ella.

Los resultados de su utilización han confirmado gradualmente las ventajas que ofrece la Orientación a Objetos, la facilidad en el diseño de sistemas de cierta complejidad y la alta productividad, sobre todo en la fase de mantenimiento de sistemas. La Orientación a Objetos permite desarrollar en tiempos muy cortos, aplicaciones a medida con un nivel de satisfacción de usuarios muy alto.

La POO fue desarrollada con la convicción absoluta de que la tecnología de Orientación a Objetos realmente podía proporcionar la productividad de desarrollo y la satisfacción de los usuarios deseadas y en consecuencia de ello, la decisión firme de desprenderse de todo el bagaje aprendido con anterioridad en el entorno de la programación procedural.

2.4 DESARROLLO INCREMENTAL

Al cabo de casi cuatro años creemos poder decir con convicción, que la Orientación a Objetos no ha fracasado. En primer lugar, en el desarrollo de aplicaciones permite entregar a los usuarios en un tiempo muy corto una aplicación utilizable aunque con una funcionalidad muy reducida. A partir de ese momento, en activa colaboración con ellos, este primer núcleo evoluciona rápidamente en aplicaciones de plena funcionalidad. Indudablemente, es virtualmente imposible captar en su totalidad, interpretar y plasmar en un sistema, de una sola vez requerimientos complejos.

"Un sistema grande que funciona, ha evolucionado inevitablemente a partir de un sistema pequeño, que había funcionado"⁽⁶⁾

⁶ Grady Booch, Diseño Orientado a Objetos

Antes de utilizar tecnología de Orientación a Objetos constantemente nos enfrentábamos al hecho de que apenas poníamos en marcha un sistema, cuyas especificaciones habían sido "extraídas de los usuarios" mediante intensos interrogatorios y cuyo desarrollo había sido largo y complejo, dichos usuarios solicitaban una serie de importantes extensiones y modificaciones.

Aún cuando se hubiera podido obtener, interpretar e incluir en el sistema las especificaciones absolutamente precisas y completas, y nuestro sistema se desarrollara tan rápido que en el momento de su puesta en marcha cumpliera con la totalidad de los requerimientos de este momento, estos requerimientos evolucionan muy rápidamente debido a la propia dinámica del negocio.

La Orientación a Objetos, debido a sus características básicas, permite un desarrollo incremental y ofrece la posibilidad de ubicar mucho más fácil y precisamente los "puntos a tocar", al realizarse el mantenimiento de un sistema.

2.5 LA POO DESDE EL PUNTO DE VISTA DEL USUARIO

Quizás el entusiasmo de los usuarios, producido por la posibilidad de responder rápidamente a sus peticiones de modificaciones, sea la experiencia más gratificante al utilizar tecnología de Orientación a Objetos. Dado que la característica más importante de la gestión actual de una empresa es su dinamismo la única constante es el cambio, la tecnología de Orientación a Objetos está predestinada para ser la herramienta idónea para el desarrollo de los sistemas de gestión. La complejidad de los sistemas requeridos y los cambios a que estos sistemas están siendo sometidos día tras día, no permitirán seguir utilizando la tecnología de desarrollo actual.

Los sistemas de gestión se han convertido en un arma competitiva de los negocios y la facilidad de modificarlos será una característica totalmente indispensable.

2.6 PRODUCTIVIDAD

La otra característica que predestina a la Orientación a Objetos a ser el enfoque de desarrollo de aplicaciones de gestión para el futuro es su productividad, proporcionada por la reutilización de componentes. En un grupo de desarrollo, una vez construido el framework necesario, los proyectos se aceleran substancialmente.

La perspectiva que ofrece la Orientación a Objetos para un futuro cercano es mucho más atractiva. Se trata de una reutilización amplia de "objetos de negocio" (business Objects), como pueden ser por ejemplo un contrato de seguro de vida, una letra de cambio, una declaración de IVA o una factura. Para esto es necesario, indudablemente, que se consoliden unos estándares de intercambio de objetos que en este momento se encuentran en proceso de su nacimiento.

Una vez establecidos y ampliamente adoptados los estándares necesarios, la Orientación a Objetos promete con toda certeza, convertir el desarrollo de sistemas en un proceso industrial, mucho más productivo que el actual enfoque "artesanal", donde se diferenciarán claramente los desarrolladores de objetos, los constructores de aplicaciones y los consumidores.

Es muy posible que la relativa pérdida de interés en los entornos de Orientación a Objetos que registran las encuestas, se deba a las múltiples confusiones que suelen producirse alrededor de cada innovación realmente importante. Sin embargo el impacto potencial de la Orientación a Objetos en la Informática es mucho más profundo que el de la programación estructurada, por tanto, también la transición será más fácil y compleja.

Actualmente muy pocas instalaciones se han decidido a asumir el difícil y costoso proceso de transición a la Orientación a Objetos, sólo un pequeño porcentaje de empresas de software ha realizado esta transición.

El desprenderse del criterio procedural, aprender a concebir con acierto la abstracción inicial del sistema, dominar nuevos lenguajes y técnicas, son algunos de los retos. Uno de los más importantes es el de implantar una nueva forma de división de trabajos en un equipo de desarrollo, que en desarrollos orientados a objetos son más pequeños, pero más preparados, "mejor entrenados".

La decisión para determinar la plataforma de desarrollo debe estar enfocada al estudio de cuál es la mejor manera de cambiar a Programación orientada a Objetos, porque cada vez son más claros sus beneficios, pero también implica altos costos, tanto en tecnologías como en recursos humanos.

Este cambio de filosofía de desarrollo, se centra en un modelo de desarrollo incremental fuertemente apoyado en prototipos, con una arquitectura cliente-servidor, tiempos relativamente cortos de desarrollo y en la búsqueda de una alta reutilización de código, con una necesidad crítica de un control de configuraciones y aseguramiento de calidad.

Por tanto, un cambio que puede ser drástico en la manera de trabajo, puede ser más efectivo y menos costoso si se hace paulatinamente y tratando de asegurar cada paso.

2.7 LA PROGRAMACION ORIENTADA A OBJETOS

La POO es una forma nueva de aproximarse a las tareas de programación, estas aproximaciones han ido cambiando de una forma dramática desde cuando se inventó la computadora, la razón principal de los cambios ha sido adaptarse a la creciente complejidad del programa.

Cuando las computadoras se inventaron la programación se la realizaba insertando mediante conmutadores las instrucciones binarias para la computadora a través del panel de control, esto funcionaba siempre y cuando se trataba de programas relativamente pequeños, sin embargo a medida que los programas fueron creciendo en extensión y complejidad, se fueron desarrollando

otro tipo de herramientas como por ejemplo los ensambladores con la finalidad de que los programadores tuvieran mejor opción de enfrentarse a programas complejos utilizando representaciones simbólicas del código de máquina.

A medida que los programas fueron creciendo, se fueron presentando los lenguajes de alto nivel para dar al programador herramientas que faciliten su tarea, el primer lenguaje de alto nivel que se utilizó fue el Fortran, el cual a pesar de constituirse en un paso importante no era el modelo de un lenguaje que propicie los programas claros y fáciles de entender.

En los años sesenta nace la programación estructurada con lenguajes como C y Pascal, con este tipo de lenguajes fue posible por primera vez escribir programas moderadamente complejos con bastante facilidad, sin embargo incluso estos lenguajes cuando un programa ha alcanzado ciertas dimensiones se vuelve incontrolable, la razón de ello se debe a que su complejidad sobrepasa lo que puede gestionar un programador que utilice programación estructurada. En cada hito del desarrollo de la programación se han creado métodos para permitir al programador enfrentarse a una complejidad cada vez mayor.

La POO ha tomado las mejores ideas de la programación estructurada y las ha combinado en varios conceptos nuevos y potentes que incitan a contemplar las tareas de programación desde un nuevo punto de vista, permite descomponer más fácilmente un problema en subgrupos de partes relacionadas del problema, entonces utilizando el lenguaje se pueden traducir estos subgrupos a unidades autocontenidas llamadas objetos.

Todos los lenguajes de programación orientada a objetos tienen tres cosas en común: Polimorfismo, Encapsulación y Herencia.

2.7.1 POLIMORFISMO

Entender el concepto de polimorfismo implica un buen número de horas de indagación y búsqueda de ejemplos. En la programación tradicional cuando declaramos una función tendemos que definir el tipo de datos que le enviamos, como por ejemplo dos números enteros, dos números reales, etc, en cambio en la programación Orientada a Objetos el tipo de dato no se conoce hasta que se ejecuta el programa.

El polimorfismo implica que el objeto será capaz de operar correctamente con cada tipo de dato y generar un resultado predecible. Al trabajar en la práctica con objetos aparecen otros conceptos que en un primer instante confunden a quienes están acostumbrados a la programación tradicional, por ejemplo en Visual Basic, las clases que normalmente se utilizan en la mayoría de programas ya están definidas, es decir, no se pueden crear nuevas clases. De esta manera se reduce el trabajo y el tiempo de desarrollo al no tener que empezar por la creación de clases o determinar si las que existen satisfacen las necesidades. La desventaja es que hay que adaptarse a los objetos disponibles, comprar más objetos creados por terceras partes o programarlos en C++. Las clases en la mayoría de lenguajes orientados a objetos presentan tres aspectos que considerar: las propiedades, los métodos y los eventos.

Para entender de mejor forma estos conceptos se puede realizar una analogía con algún objeto de la vida diaria: una televisión conectada a una videocreadora. Las propiedades representan los ajustes bajo los cuales opera el equipo: el volumen, el brillo de la imagen, la saturación de color, el tamaño vertical y horizontal de la imagen. Los eventos son las acciones o comandos que puedo ejecutar sobre el aparato: presionar el botón de encendido, presionar el botón de avance de cinta, presionar el control para el cambio de hora, mover el control de volumen. En el caso de VB el programa responde a los eventos por medio de procedimientos que se programan en forma tradicional. La principal dificultad consiste en seleccionar el evento correcto para la acción deseada.

Los métodos son los procesos internos que ocurren en el aparato (procedimientos) en respuesta a un comando: al presionar el botón de avance de la cinta, se activa un motor, se activa el contador, se “lee” y transmite la imagen de la cinta a la televisión, etc. El programador de VB no tiene acceso a la codificación de los métodos, sólo los usa. En VB tenemos como ejemplo el método FindFirst (parámetro) que al aplicarse a un objeto de base de datos localiza la primera ocurrencia del dato especificado en el parámetro.

En este momento del desarrollo de la POO, existen muchas opiniones encontradas sobre su porvenir. La mayoría de los expertos lo considera como la mejor o incluso la única opción para el futuro de la computación, pues promete resolver gran parte de los problemas actuales de desarrollo de software, incluyendo la programación para las futuras computadoras en paralelo que más adelante se utilizarán. Otros expertos tienen dudas sobre si la POO realmente proporcionará en verdad todo lo prometido.

Los lenguajes de programación Orientada a Objetos admiten el polimorfismo, que significa que un nombre se puede utilizar para especificar una clase numérica de acciones, sin embargo y dependiendo del tipo de datos con el que se esté tratando, se ejecuta una variante concreta del caso general. Por ejemplo en un programa que defina tres tipos de pilas, una que se utiliza para valores enteros, una para valores de coma flotante y otra para valores de tipo long, como consecuencia del polimorfismo se pueden crear tres grupos de funciones para estas pilas denominadas poner() y quitar(), el compilador seleccionará la rutina de datos correcta dependiendo del tipo de datos con el cual se invoque, en este ejemplo el concepto general es el de poner o quitar datos de una pila, las funciones definen el modo concreto en que se hace para cada tipo de dato.

Los primeros lenguajes de Programación Orientada a Objetos eran intérpretes así que en el polimorfismo se le permitía en el momento de la ejecución, pero hoy en día existen lenguajes que permiten el polimorfismo tanto en tiempo de compilación como en tiempo de ejecución.

2.7.2 ENCAPSULACION DE DATOS

La encapsulación de datos, no es más que la idea de mezclar las variables y las funciones que sirven para manipular dichas variables, en una misma estructura de datos.

La encapsulación de datos es fundamental en la POO, ya que permite construir objetos que no son más que instancias. Una de las ventajas que reporta la encapsulación de datos es la posibilidad de crear variables totalmente protegidas de posibles efectos no deseados, como cambio de valor por descuido o efectos laterales, gracias a tener unas funciones que se encargan de realizar todas las operaciones sobre ellas, es decir, si se construye un objeto con sus propias variables ya no es necesario darle valor a la variable directamente, como se ha hecho toda la vida, sino que lo que se hace es llamar a una función que lo único que hará será realizar dicha asignación. Se puede pensar que esto no es muy inteligente ya que se puede hacer directamente, pero precisamente en eso consiste la POO; al hacer eso se está ocultando información y evitando que las variables puedan tomar valores no deseados, limando en gran medida, fallos de paso de valores a las variables, tan comunes en la programación tradicional.

2.7.3 HERENCIA

La herencia es otro de los pilares sobre los que descansa la POO. Gracias a la herencia, podemos hacer que todas las clases que se tenga puedan estar conectadas entre sí, aprovechando unas de otras, diversas características básicas.

La idea de la herencia parte de poder tener un diseño en el que todo se vaya descomponiendo en módulos o problemas que van de lo más general a lo más específico y que a su vez, estén conectados entre sí.

Por ejemplo si tenemos dos clases que como tales tienen unas cosas en comunes y otras que no lo son, la solución consiste en crear una clase que contenga aquellas variables o atributos comunes para que

sean heredados por las dos clases, esta solución permite un ahorro de memoria y clarifica todo de una forma bastante aguda.

La herencia permitirá construir perfectas jerarquías de clases que nos ayudarán a partir de una idea general y llegar a algo concreto y de fácil tratamiento. Saber que es un objeto y que una clase, así como las características y ventajas que aporta una encapsulación de datos y funciones junto a la posibilidad de desglosar los problemas a modo de jerarquías mediante herencias, son cosas suficientes como para ir haciéndose una idea de los mecanismos de la POO.

La herencia es un proceso mediante el cual un objeto puede adquirir las propiedades de otro objeto. Esto es importante porque sirve de base para el concepto de clasificación. La mayor parte del conocimiento resulta tratable gracias a una clasificación jerárquica, sin embargo cuando se utilizan las clasificaciones los objetos necesitan definir sólo aquellas características que los hacen únicos dentro de su clase. El mecanismo de herencia es el que hace posible que un objeto sea un caso concreto más que un caso general.

2.7.4 OTROS CONCEPTOS IMPORTANTES

El tema de la Programación Orientada a Objetos involucra un sinnúmero de elementos cuya correcta interpretación es indispensable, su utilización está en dependencia del tipo de usuario y de aplicación, por esta razón a continuación se detallan los principales.

OBJETO

Cada objeto tiene una identidad única que es distinta e independiente de cualquiera de sus características. Cada objeto ofrece una o más operaciones.

La característica individual más importante de un lenguaje orientado a objetos es el objeto, definido en pocas palabras un objeto es una entidad lógica que contiene datos y un código que manipula

dichos datos. Dentro de un objeto, parte del código y/o los datos pueden ser privados para el objeto y no será accesible para nada que haya fuera del objeto. Un objeto proporciona un nivel significativo de protección contra alguna otra parte del programa que no está relacionada con él, y que pudiera modificar o utilizar correctamente las partes privadas del objeto. El enlazado del código y de datos de esta manera suele denominarse encapsulación.

A todos los efectos, un objeto es una variable de tipo definido por el usuario, al inicio talvés parezca extraño pensar que un objeto que enlaza código y datos es una variable, pero es precisamente esto lo que pasa e la Programación Orientada a Objetos, cuando se define un objeto se está creando implícitamente un nuevo tipo de datos.

CLASE

Conjunto de objetos con características similares. La clase considera una plantilla utilizada para crear nuevos objetos. Una clase incluye tanto datos como métodos.

INSTANCIA

Una implementación de la plantilla de una clase. Un objeto es una instancia de una clase, el cual contiene la implementación de la interface de datos.

MÉTODO

Una operación que efectúa acceso a datos.

MENSAJE

Es el mecanismo mediante el cual se comunican los objetos. Un mensaje permite a los métodos enviar o recibir datos. Los mensajes y objetos responden para formar las bases de sus protocolos.

PROTOCOLO

Los mensajes a los que responde un objeto.

SUBCLASE

Una clase que tiene todas las propiedades de su clase padre, así como también cambios localizados.

SUPERCLASE

Una clase padre de una subclase.

HERENCIA MÚLTIPLE

La habilidad de heredar de padres múltiples.

2.8 LENGUAJE ORIENTADO A OBJETOS (LOO)

Es un lenguaje de programación que se basa en objetos, clases y es capaz de tener herencia de clases. Los objetos, siendo unidades encapsuladas, son inherentemente portables, mientras la herencia permite la reutilización de código en los SPOO. Es más fácil modificar el código existente debido a que los objetos no interaccionan más que a través de mensajes, así, una modificación en el código de un objeto no afectará la operación de otro objeto, mientras los métodos respectivos permanezcan intactos. La reutilización del código reduce el tiempo empleado en el desarrollo, y hace al desarrollo de software más intuitivo debido a que la gente piensa en forma natural en términos de objetos en lugar de en términos de algoritmos de software.

2.9 ORIENTACION A OBJETOS EN MULTIMEDIA

Los documentos hipertextos son colecciones de nodos de información textual interconectados por enlaces. Si los nodos pueden contener información multimedia (como sonidos o secuencias de vídeo) se trataría entonces de documentos hipermedia. En esta comunicación se ofrece una visión general de las posibles aplicaciones del paradigma de orientación a objetos en el ámbito

multimedia y hablando con más propiedad en hipermedia: desde las posibilidades de almacenamiento multimedia que ofrecen los Sistemas de Bases de Datos Orientadas a Objetos, hasta las recientes metodologías de desarrollo de Sistemas Hipermediales, como OOHDM (Object Oriented Hypermedia Design Model) y EORM (Enhanced Object Relationship Model), que incorporan las ideas de orientación a objetos, como herencia y encapsulación, permitiendo modelar las especificaciones de la documentación hipermedia.

El término HIPERMEDIA, combinación de los conceptos HIPERtexto y multiMEDIA, hace referencia a una tecnología de construcción de (hiper)documentos que permite a los lectores encontrar fácilmente la información que realmente necesitan, de la manera que ellos decidan, a través de enlaces establecidos por el autor entre los diferentes elementos de información multimedia (texto, sonido, imagen, vídeo, etc.) que conforman el documento. Aunque en la literatura se suelen utilizar indistintamente ambos, el término Hipertexto se refiere a un documento hipermedia donde todos los elementos de información contienen únicamente texto. También suele confundirse con el término Multimedia, cuando éste, en realidad, hace referencia a sistemas que contienen y presentan texto, imágenes, sonido, video, etc, pero sin enlaces entre estos elementos de información.

Para ilustrar las posibilidades de esta tecnología, se podría imaginar como ejemplo de documento hipermedia o hiperdocumento, un posible informe policial sobre un accidente de tráfico que, además del texto describiendo el siniestro, incorpore fotografías digitalizadas tomadas por los agentes, la versión grabada de la propia voz o en vídeo de los testigos, imágenes "escaneadas" de los informes médicos, datos geoespaciales del lugar donde se produjo el siniestro, etc. El documento debería, por tanto, incluir enlaces entre estos elementos que establezcan las posibles vías de acceso o de navegación por la información. En este ejemplo de tráfico podría existir un enlace a una secuencia de video donde un testigo da explicaciones, desde el lugar del texto del informe en el que aparece el nombre de esa persona.

Un hiperdocumento se compone de dos tipos de elementos: los nodos o "contenedores" de la información multimedia y los enlaces, que interconectan los nodos permitiendo otras alternativas de navegación por la información diferente del clásico acceso secuencial "desde la primera a la última línea".

Sin embargo existe un tercer componente fundamental denominado ancla. Se trata de un mecanismo para señalar puntos incluidos en el interior de los nodos que sirven de origen o de destino a un determinado enlace entre nodos. Por ejemplo, en un nodo con contenido textual puede haber palabras o frases marcadas que al ser seleccionadas por el usuario activen un enlace originando el acceso al contenido de otro nodo. Este nodo destino podría contener información multimedia de tipo secuencia de vídeo, con lo que el ancla debería marcar el fotograma o el instante de la secuencia de vídeo que debería comenzar a visualizarse al activarse el enlace.

Los Sistemas Hipermediales son los entornos que ofrecen a los usuarios todos los mecanismos para la creación, manipulación y consulta de hiperdocumentos. Así, respecto a su interface con los usuarios, proporcionan browsers o visualizadores para la navegación y, opcionalmente, visiones de mapas generales del hiperespacio del documento para que los usuarios conozcan su situación en cada momento. También incluyen herramientas de autor para el mantenimiento de los contenidos de los nodos y de la estructura de enlaces. En este caso se deberán tener en cuenta todas las posibilidades multimedia, por lo que harán falta editores, no sólo de texto, sino también de sonido, de imágenes, etc. y mecanismos de conexión entre todos estos tipos de información.

Además de las facilidades anteriores que son percibidas por los usuarios del sistema y, teniendo en cuenta que la información de un documento hipermedia (tanto la de los nodos como la relativa a los enlaces) será registrada físicamente en un sistema de ficheros o en una base de datos, deberán existir también mecanismos que adapten dicha información a las características del sistema de

almacenamiento utilizado, así como los correspondientes procedimientos de acceso y recuperación de datos.

Debido a la complejidad evidente de los documentos hipertexto, desde los orígenes de esta tecnología se ha intentado establecer un modelo universal de hiperdocumento que permita su percepción desde diferentes niveles de abstracción para facilitar el desarrollo de estándares de interface entre niveles que garanticen la portabilidad de los documentos generados. Uno de los modelos más conocidos es el denominado Dexter Hypertext Reference Model, presentado en 1990 para Hipertextos y mejorado en 1993 por el Amsterdam Hypermedia Model para recoger aspectos específicos de la información multimedia. El modelo Dexter considera tres niveles para un hiperdocumento:

1 Nivel de ejecución (Run-time Layer)

Se trata del nivel de mayor abstracción, que se superpone a los restantes niveles. En él se describe el hiperdocumento tal y como lo perciben los usuarios, como una serie de nodos con contenido que aparecen en pantalla y desde los que se puede acceder a otros nodos a través de enlaces. Este nivel también incluye las características de la Interface Gráfica de Usuario (GUI) utilizada en la visualización del hiperdocumento.

2 Nivel de almacenamiento (Storage Layer)

En este nivel intermedio se veía el documento como una base de datos en la que se almacena toda la información en entidades o "componentes" relacionados entre sí.

3 Nivel de contenido (Within-Component Layer)

Este nivel se concentra en la estructura de la información en el interior de los componentes de la base de datos, que puede, a su vez modelarse a través de alguna método de estructuración de documentos como ODA o SGML.

De todo lo anterior se evidencian algunas consideraciones importantes a tener en cuenta respecto a la tecnología hipermmedia que pueden aconsejar la incorporación del paradigma de la orientación a objetos en los diferentes aspectos de dicha tecnología:

- La información que se maneja en los sistemas hipermmediales es muy heterogénea, lo cual es evidente debido al componente multimedia en la tecnología.
- Los hiperdocumentos se estructuran en bloques documentales (nodos) autónomos a los que se accede a través de enlaces.
- Se va a requerir un gran espacio de almacenamiento para la información.
- Se pueden crear hiperespacios de navegación muy complejos debido a las posibilidades de interconexión de nodos
- El usuario es el elemento más importante de un sistema hipermmedial, hay que cuidar especialmente la interface del sistema con el usuario para hacerlo atractivo.

Actualmente se habla de la posibilidad de aplicar el paradigma de la orientación a objetos con el fin de mejorar las prestaciones de los sistemas hipermmediales. En primer lugar se ha estudiado la posibilidad de utilizar Sistemas de Gestión de Bases de Datos Orientados a Objeto (SGBDOO) para el almacenamiento la información heterogénea y el estándar ISO 8613 como modelo orientado a objetos de estructuración del contenido de los componentes de un hiperdocumento. Finalmente se ha analizado las metodologías de desarrollo orientadas a objeto de sistemas hipermmediales que, al tomar conciencia de la dificultad del proceso de desarrollo, adoptan un enfoque de ingeniería de software, incidiendo sobre todo en la necesidad de un diseño previo que recoja en modelos abstractos las necesidades de los usuarios.

2.10 HIPERMEDIA Y BASES DE DATOS ORIENTADAS A OBJETOS

Los Sistemas de Gestión de Bases de Datos tradicionales, y en particular los relacionales, adolecen de falta de capacidad semántica y de soporte para la abstracción estructural necesarios para el almacenamiento de la información hipermmedia. El paradigma de orientación a objetos el que parece más adecuado para cubrir todas las necesidades del desarrollo de sistemas hipermmediales. Así, un SGBDOO sería el modelo ideal para el nivel de almacenamiento de hiperdocumentos, ya que podría soportar los requisitos de estos sistemas, como la necesidad de almacenamiento de información multimedia no estructurada y el establecimiento de enlaces complejos entre objetos documentales.

En este sentido, aunque algunos sistemas relacionales incorporan campos BLOBs (Binary Large Objects) para soportar información multimedia, esta solución es parcial y no siempre la más adecuada porque, a pesar de lograr que los objetos multimedia queden almacenados, no existe una integración real en el modelo de datos ni aporta soluciones.

Para poder lograr aplicaciones integradas es necesario poder definir operaciones sobre los tipos de datos convencionales y además poder definir nuevos tipos (texto, sonido, vídeo, series temporales, etc.) y nuevas estructuras para estos nuevos tipos, los cuales se podrían implementar junto con propiedades de composición con lo que se mejoraría la posible representación de la información multimedia y la estructura de enlaces de los hiperdocumentos, respecto a la utilización de las estructuras clásicas de los modelos relacionales.

El desarrollo de la tecnología hipermmedia está, por tanto, ligado a la de los SGBDOO, por lo que los fabricantes de entornos de producción de hiperdocumentos tendrán que prestar atención a la evolución de estas Bases de Datos y a los esfuerzos de normalización que actualmente se están realizando para la definición de estándares que permitan, por una parte el desarrollo de productos que ofrezcan mejores prestaciones, portabilidad y reutilización de estos productos.

2.11 ESTANDAR ISO8613(ODA) DE REPRESENTACION ORIENTADA A OBJETOS

Hay autores que distinguen en un hiperdocumento una estructura primaria y otra secundaria. La estructura primaria, normalmente jerárquica, es la que existe en cualquier documento tradicional en papel, con unos componentes de información que se combinan formando otros componentes de mayor nivel. La estructura secundaria estaría representada por otras relaciones adicionales entre estos componentes. Es precisamente esta segunda estructura la que se destaca en los sistemas hipermediales, por ser la que les confieren su carácter navegacional. Para evitar problemas de desorientación del usuario en el hiperespacio, es conveniente definir claramente una estructura primaria en los hiperdocumentos. En este sentido, la norma ODA (Office Document Architecture) de ISO puede resultar de utilidad ya que establece una estructura jerárquica y orientada a objetos para los documentos y ofrece mecanismos que pueden ser utilizados para especificar enlaces entre los componentes (objetos) de un documento.

2.12 METODOLOGIAS DE DESARROLLO DE SISTEMAS HIPERMEDIALES

Habitualmente el desarrollo de Sistemas Hipermediales suele hacerse utilizando directamente herramientas de autor a nivel de implementación, descuidándose el importante proceso previo de análisis y diseño abstracto de los aspectos estructurales, de navegación y de interface con el usuario que, de esta forma, no quedan recogidos en documentación alguna, haciendo complicado el mantenimiento futuro del sistema. Sin embargo, en los últimos años existe una tendencia a considerar el desarrollo hipermedial con un enfoque de proceso de ingeniería de software, por lo que ya se han propuesto diferentes metodologías, que establecen la necesidad de considerar un diseño previo a la construcción del sistema y ofrecen una serie de técnicas, más o menos formales, para recoger en diferentes modelos abstractos las especificaciones del sistema hipermedial a desarrollar.

Sólo recientemente han empezado a aparecer metodologías que asumen la orientación a objetos como paradigma de diseño al igual que ha ocurrido en el ámbito de la Ingeniería del Software. En el caso hipertextual, el enfoque OO en los modelos es muy útil debido al gran nivel de abstracción que ofrece y a sus mecanismos de composición (generalización, clasificación y agregación) que facilitan el modelado de la estructura hipertextual. Esto queda reflejado en las dos metodologías más conocidas de desarrollo orientado a objetos de sistemas hipertextuales como son EORM y OOHDM que se describen a continuación. En ambos casos el proceso de desarrollo comienza con una fase de análisis de la información del sistema en la que se aplica alguna de las metodologías OO conocidas en ingeniería del software, como OMT (Object Modeling Technique). La principal diferencia entre EORM y OOHDM radica en que la segunda separa claramente el diseño conceptual del sistema del diseño de las posibilidades de navegación por el hipertexto, ofreciendo técnicas diferentes para ambos casos.

2.13 METODOLOGIA EORM (Enhanced Object-Relationship Model)

En esta metodología, propuesta por D. Lange, el proceso de desarrollo de un Sistema de Información Hipertextual comprendería una primera fase de Análisis Orientado a Objetos del sistema, sin considerar los aspectos hipertextuales del mismo, obteniendo un Modelo de Objetos con la misma notación utilizada en OMT, que refleje la estructura de la información y el comportamiento del sistema.

La idea fundamental de esta metodología es considerar una segunda fase, de **Diseño**, durante la cual se proceda a modificar el modelo de objetos obtenido durante el análisis añadiendo la semántica apropiada a las relaciones entre clases de objetos para convertirlas en enlaces hipertextuales, obteniendo finalmente un modelo enriquecido, que su autor denomina EORM, en el que se refleje tanto la estructura de la información (modelo abstracto hipertextual compuesto de nodos y enlaces) como las posibilidades de navegación ofrecidas por el sistema. sobre dicha estructura, para lo cual

existirá un repositorio o librería de clases de enlaces, donde se especifican las posibles operaciones asociadas a cada enlace de un hiperdocumento.

El desarrollo del Sistema concluiría con una última fase de Construcción, durante la que se generaría el código fuente (por ejemplo en C++) correspondiente a cada clase, y se prepararía la Interface Gráfica de Usuario.

La adopción del enfoque orientado a objetos OMT garantiza todas las ventajas reconocidas para esta técnica de modelado, como la flexibilidad (posible existencia de múltiples formas de relaciones entre nodos) y la reutilización, por la existencia de una librería de clases de enlaces que pueden ser reutilizados en diferentes proyectos de desarrollo hipermedial.

Para automatizar la aplicación de la metodología EORM, su autor ha desarrollado, en los laboratorios de investigación de IBM, una herramienta denominada ODMTool que, junto a un generador comercial de Interfaces Gráficas de Usuario denominado ONTOS Studio y un Sistema de Gestión de Base de Datos Orientado a Objetos (SGBDOO), permite el diseño interactivo de esquemas EORM y la generación de código fuente, inicialmente en C++, de las clases incluidas en estos esquemas. El SGBDOO ofrece un repositorio de objetos que permite la compartición de la información de los esquemas entre las herramientas (ODMTool, ONTOS Studio) y las aplicaciones hipermediales desarrolladas.

2.14 METODOLOGIA OOHDM (Object Oriented Hypermedia Design Method)

La metodología OOHDM, propuesta por D. Schwabe y G. Rossi, establece que el desarrollo de un Hiperdocumento es un proceso de cuatro fases en el que se combinan diferentes estilos de desarrollo como el incremental, iterativo y prototipado. Las tres primeras fases son de diseño, en las que se obtiene un conjunto de modelos orientados a objeto que describen el documento que será construido en la última fase.

Durante la primera fase, denominada Diseño Conceptual o Análisis de Dominio, se realiza, como también ocurría en la metodología EORM, el modelado del dominio del hiperdocumento utilizando algún método análisis orientado a objetos de Sistemas de Información, por ejemplo OMT, obteniendo un esquema conceptual de clases en el que, además de clases abstractas y objetos, se representan las relaciones entre ellas, incluidas las de herencia y agregación, y los correspondientes atributos y métodos asociados a las clases.

Una vez obtenido el esquema conceptual la metodología OOHDM establece una segunda fase de Diseño Navegacional en la que se ha de definir la estructura de navegación a través del hiperdocumento mediante la realización de modelos navegacionales que representen diferentes vistas del esquema conceptual de la fase anterior. Se trata, en definitiva, de reorganizar la información para adaptarla a las necesidades de los usuarios del sistema. El Diseño Navegacional se expresa, también con un enfoque orientado a objetos, a través de dos tipos de esquemas o modelos: el denominado esquema de clases navegacionales, con las posibles vistas del hiperdocumento a través de unos tipos predefinidos de clases, llamadas navegacionales, como son los "nodos", los "enlaces", y otras clases que representan estructuras o formas alternativas de acceso a los nodos, como los "Indices" y los "recorridos guiados"; y el esquema de contexto navegacional, que permite la estructuración del hiperespacio de navegación en subespacios para los que se indica la información que será mostrada al usuario y los enlaces que estarán disponibles cuando se acceda a un objeto (nodo) en un contexto determinado.

La metodología OOHDM contempla una tercera fase de diseño, denominada Diseño de la Interface Abstracta, en la que se realiza un modelo, también orientado a objetos, para especificar la estructura y el comportamiento de la interface del sistema hipertexto con el usuario. Este modelo es abstracto y, por tanto, independiente de la implementación final del sistema. Sin embargo, se basa en las ideas que actualmente se aplican en las Interfaces Gráficas de Usuario (GUIs), por lo

que como la mayor parte de entornos hipertexto comerciales trabajan con GUIs, su implantación en un entorno de este tipo debe ser una tarea sencilla.

El modelo de la interface abstracta se expresa a través de tres tipos de diagramas que se complementan entre sí. En primer lugar se deben crear los denominados diagramas de Vistas de Datos Abstractos (ADV) que incluyen una vista (ADV) por cada clase navegacional (nodo, enlace o estructura de acceso) que fue establecida durante la fase de Diseño Navegacional. Un diagrama de este tipo se compone de una serie de cajas (una caja es un ADV) que representan las diferentes clases de objetos que aparecerán ante el usuario. Un segundo tipo de diagramas que componen el modelo de interface es el Diagrama de Configuración, donde se representan principalmente los eventos externos (provocados por el usuario, como ClicDeRatón o DobleClic) que maneja un ADV, los servicios que ofrece el ADV (como "visualización") y las relaciones estáticas entre las ADVs. Por último, el modelo se completa con los denominados Diagramas de Estado que, como ocurre en los modelos orientados a objetos de ingeniería del software, representan el comportamiento dinámico del sistema hipertextual mediante el establecimiento de un diagrama de transición de estados para cada ADVs, en el que se reflejan los posibles estados por los que puede pasar cada objeto de la interface (oculto, desactivado, ampliado, reducido, normal, etc.) y los eventos que originan los cambios de estado.

La última fase de la metodología OOHDM es la Implementación del hipertexto o sistema hipertextual diseñado, es decir, la concreción de los modelos navegacionales y de interface en objetos particulares con sus correspondientes contenidos (textuales, visuales sonoros, etc.) y sus posibilidades de navegación. Aunque, al utilizar un enfoque de orientación a objetos podría parecer conveniente que la implementación se hiciera en un entorno de construcción de hipertextos también orientado a objetos, como MacWeb, debido al carácter abstracto del diseño, ésta puede hacerse fácilmente en otros entornos hipertextuales como Hypercard, DIRECTOR, etc.

2.15 CONCLUSIONES

La tecnología hipertexto brinda muchas posibilidades en el ámbito documental para la producción de hiperdocumentos que incorporen elementos multimedia que los hagan especialmente atractivos.

Los avances en las tecnologías de almacenamiento de datos, como el CD-ROM, y de las telecomunicaciones, especialmente Internet, están facilitando la aparición de un gran número de productos hipertexto (enciclopedias, museos virtuales, periódicos en la red Internet, etc.) con una complejidad creciente. Por ello, es natural el interés que la comunidad dedicada al desarrollo hipertextual está prestando a las metodologías que recientemente han aparecido para racionalizar el proceso de construcción de estas aplicaciones mediante el desarrollo de modelos que faciliten su posterior mantenimiento y reutilización.

En este sentido, la consolidación de las metodologías orientadas a objetos en el campo de la Ingeniería del Software parece aconsejar este mismo enfoque en el campo hipertexto, que no deja de ser un caso particular de desarrollo de software. Como ha ocurrido con las herramientas CASE, la aplicación de las metodologías que se propongan será real si a los autores se les ofrecen herramientas potentes con las que diseñar y verificar los modelos y generar automáticamente los documentos hipertextuales a partir de esos modelos.

CAPITULO III

ESTUDIO DE INTERFACES HOMBRE - MAQUINA

3.1 ASPECTOS GENERALES

La interfaz con el usuario es la frontera entre un sistema de computadora, compuesto de hardware y software y un usuario humano. El software de interfaz con el usuario es un componente significativo de los sistemas computacionales contemporáneos y las interfaces gráficas con el usuario son hoy en día casi universales.

Una interface forma parte de un sistema computacional y determina como las personas operan y controlan el sistema, cuando una interface está bien construida es agradable, comprensible y controlable, lo que produce como resultado que los usuarios se sienten satisfechos.

La correctitud del software de una interfaz con el usuario es esencial para las aplicaciones críticas, tales como las que afectan la vida humana o la seguridad. La correctitud de la interfaz con el usuario es también importante para las aplicaciones que no son críticas porque la interfaz es el punto en donde el sistema es percibido por el usuario. El hacer pruebas es un método que es utilizado para determinar la correctitud del software, pero el hacer pruebas al software de una interfaz con el usuario es difícil.

Es fácil comprender la importancia de este asunto ya que todos somos usuarios de interfaces, cada vez que utilizamos un sistema utilizamos una interface. Una interface hombre-máquina es una preocupación de la industria en general y existió mucho antes de que se hablara de interfaces con un computador.

3.2 QUÉ ES UNA INTERFAZ DE USUARIO

Es un conjunto de elementos a través de los cuales un usuario interactúa con un objeto que realiza una tarea determinada. Ejemplos: televisor, teléfono, coche, despertador, puerta. El ser humano está continuamente interactuando con los objetos que le rodean, y tiene unas expectativas sobre cómo éstos deben comportarse, basada en pasadas experiencias con estos objetos u otros similares.

Si la interfaz 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. En muchas ocasiones se trata de objetos mal diseñados, donde suele primar la estética sobre la usabilidad, cuyas indicaciones de uso son confusas o no se considera la predisposición de los usuarios.

Cuando se diseña un objeto es preciso pensar en quiénes van a usar dicho objeto, y qué expectativas van a tener sobre su forma de uso, tanto si son objetos conocidos (el usuario espera que se comporte de una forma determinada) como si se trata de objetos novedosos (el usuario trata de asociarlos a los que conoce).

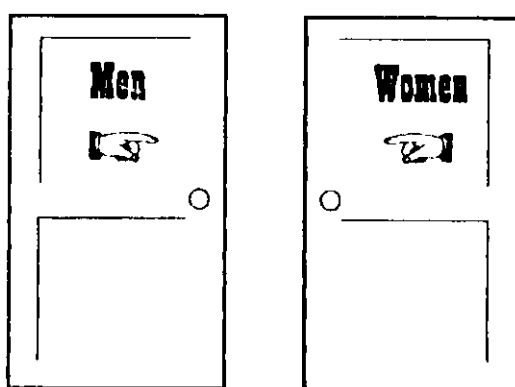


FIGURA 3.1.- Ejemplo de una interface mal diseñada

3.3 INTERFAZ DE USUARIO DE UN PROGRAMA

Es el 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. La interface 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. Los componentes software son los elementos que el usuario ve, oye, apunta o toca en la pantalla para interactuar con el ordenador, así como la información con la que trabaja.

También se puede considerar parte de la interfaz la documentación (manuales, ayuda, referencia, tutoriales) que acompaña al hardware y al software. El diseño de una interfaz es fundamental para el éxito de un programa. Un buen programa con una pobre interfaz tendrá una mala imagen, y al contrario, una buena interfaz puede realzar un programa mediocre.

Los objetivos de los usuarios son muy variables; incluso un usuario puede cambiar su objetivo cuando realiza una misma tarea. Los programas son usados por usuarios con distintos niveles de conocimientos desde novatos hasta expertos. Así pues, no existe una interfaz 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.

3.3.1 MODELOS DE INTERFACES

Existen tres puntos de vista distintos en una interfaz de usuario: el del usuario, el del programador y el del diseñador (analogía de la construcción de una casa). Cada uno tiene un modelo mental propio de la interfaz, que contiene los conceptos y expectativas acerca de la interfaz, desarrollados a través de su experiencia. El modelo permite, entre otras cosas, explicar o predecir comportamientos del sistema y tomar las decisiones adecuadas para modificar el mismo. Los modelos subyacen en la interacción con los ordenadores, de ahí su importancia.

3.3.1.1 MODELO DEL USUARIO

El usuario tiene su visión personal del sistema, y espera que éste se comporte de una cierta forma que se puede conocer estudiando al usuario (realizando tests de usabilidad, entrevistas o a través de una realimentación).

Una interfaz debe facilitar el proceso de crear un modelo mental efectivo. Para ello son de gran utilidad las metáforas, que asocian un dominio nuevo a uno ya conocido por el usuario; un ejemplo típico es la metáfora del escritorio, común a la mayoría de las interfaces gráficas actuales. Las metáforas constituyen una ayuda, relacionando el sistema con algo ya conocido por el usuario: escritorio (Figura 3.2), sala de juegos infantil (Figura 3.3). Son dos ejemplos de las ventajas del uso de gráficos.

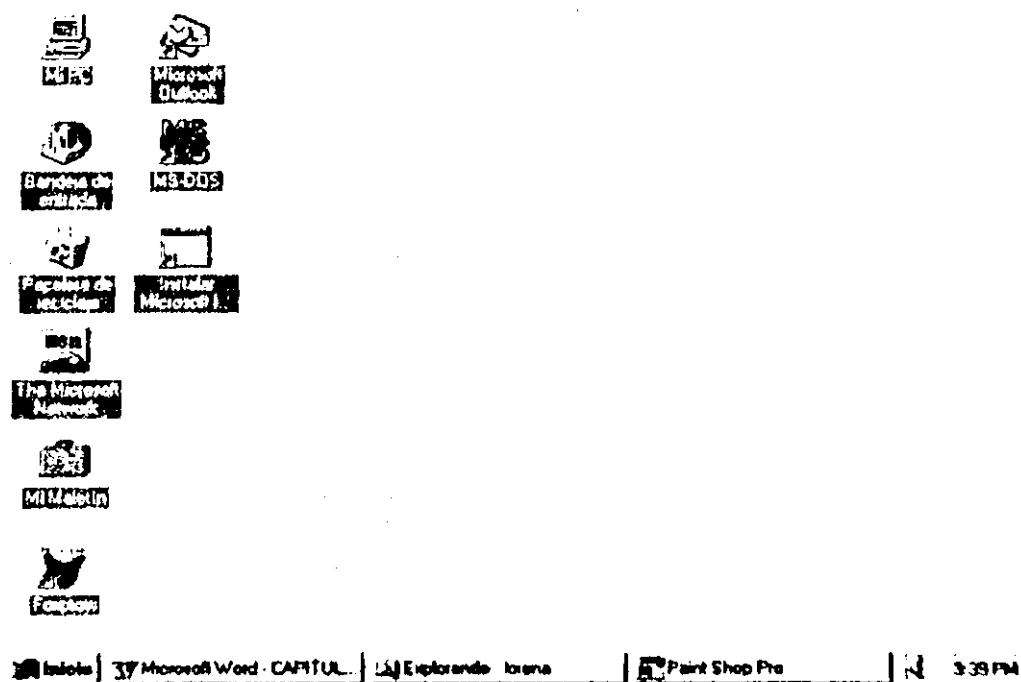


FIGURA 3.2 Metáfora del escritorio en Windows 95



FIGURA 3.3. Metáfora del cuarto de juegos

3.3.1.2 MODELO DEL PROGRAMADOR

Es el 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, especificaciones. Sin embargo, esto no significa necesariamente que tenga la habilidad de proporcionar al usuario los modelos y metáforas más adecuadas. Muchos no consideran el modelo del usuario del programa, y si sus propias expectativas acerca de cómo trabaja el ordenador.

3.3.1.3 MODELO DEL DISEÑADOR

El diseñador mezcla las necesidades, ideas, deseos del usuario y los materiales de que dispone el programador para diseñar un producto de software. Es por lo tanto, 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. El modelo consta de tres partes: presentación, interacción y relaciones entre los objetos (Figura 3.4). La presentación es lo que primero 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. La presentación no es lo más relevante, y un abuso en la misma (por ejemplo, en el color) puede ser contraproducente, distrayendo al usuario. La segunda parte define las técnicas de interacción del usuario, a través de diversos dispositivos. La tercera es la más importante, y es donde 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 del interfaz, los aspectos visuales saldrán de una manera lógica y fácil.

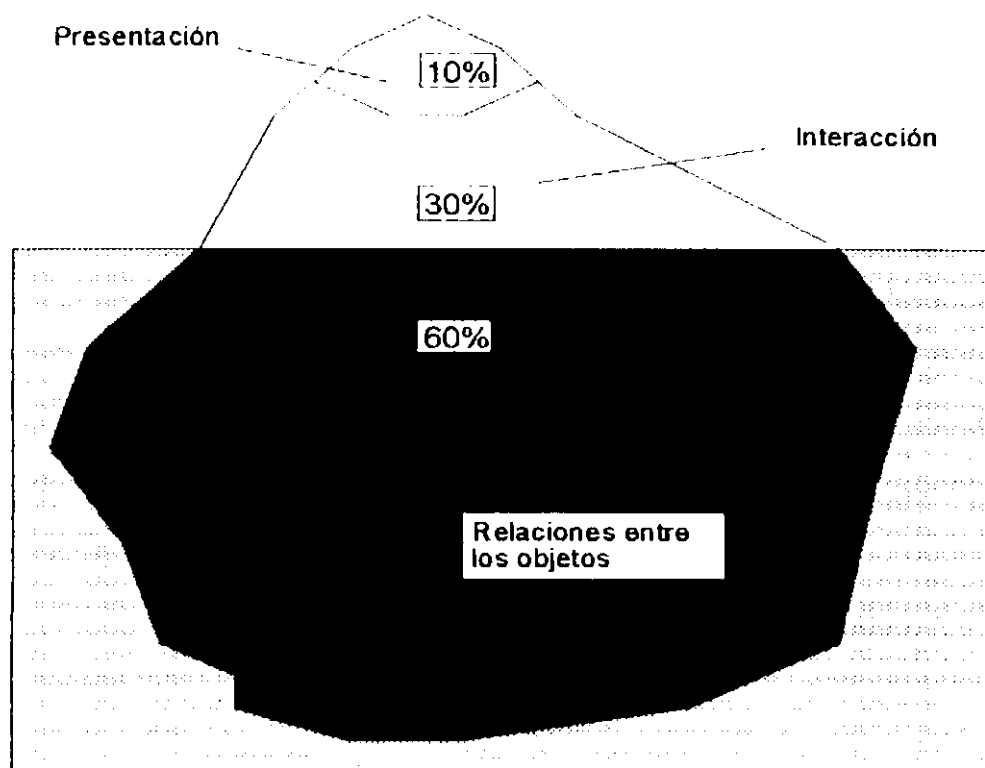


FIGURA 3.4 Representación del modelo del diseñador. IBM (1992)

Estos modelos deben estar claros para los participantes en el desarrollo de un producto, de forma que se consiga una interfaz atractiva y a la vez efectiva para el trabajo con el programa. Una interfaz no es simplemente una cara bonita; esto puede impresionar a primera vista, pero decepcionar a la larga. Lo realmente importante es que el programa se adapte bien al modelo del usuario, cosa que se puede comprobar utilizando el programa más allá de la primera impresión.

3.3.2 ASPECTOS DE LA PSICOLOGÍA HUMANA

Al diseñar interfaces de usuario deben tenerse en cuenta las habilidades cognitivas y de percepción de las personas, y adaptar el programa a ellas. Así, una de las cosas más importantes que una interfaz 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 a repetir operaciones ya realizadas (por ejemplo, introducir un mismo dato repetidas veces). La persona tiene unas habilidades distintas de la máquina, y ésta debe utilizar las suyas para soslayar las de aquella (como por ejemplo la escasa capacidad de la memoria de corto alcance).

Existen una serie de principios a seguir en el desarrollo de interfaces de usuario que son consideradas como reglas de oro del diseño de interfaces de usuario, estas son:

3.3.3 REGLAS PARA EL DISEÑO DE INTERFACES DE USUARIO

Cuando se diseña una aplicación se deben considerar los siete principios de diseño centrados en el usuario recomendados por Microsoft, los mismos que fueron utilizados en el diseño de Windows 95:

- Dar control al usuario.
- Ser directo
- Ser indulgente
- Reducir la carga de memoria del usuario, simplicidad.

- Buscar retroalimentación
- Cuidar la estética
- Ser Consistente

3.3.3.1 DAR CONTROL AL USUARIO.

El usuario debe sentir que está controlando lo que sucede en la pantalla, que ellos inician la acción en lugar de reaccionar a otra, sobre todo cuando la aplicación tiene un alto grado de automatización hay que asegurarse que el usuario tenga control sobre ella.

También es importante permitir que el usuario personalice su aplicación ya que puede tener distintas preferencias y necesidades. Una buena aplicación debe interactuar con el usuario, una buena aplicación le dice lo que está haciendo y en que modo está, la aplicación debe dar respuestas y ser tan interactiva como pueda.

El diseñador debe dar al usuario la posibilidad de hacer su trabajo, en lugar de suponer qué es lo que éste desea hacer. La interfaz debe ser suficientemente flexible para adaptarse a las exigencias de los distintos usuarios del programa. En concreto, se pueden enumerar los siguientes principios que permiten al usuario estar en posesión del control: -

- Usar adecuadamente los modos de trabajo.
- Permitir a los usuarios utilizar el teclado o el ratón.
- Permitir al usuario interrumpir su tarea y continuarla más tarde.
- Utilizar mensajes y textos descriptivos.
- Permitir deshacer las acciones, e informar de su resultado
- Permitir una cómoda navegación dentro del producto y una fácil salida del mismo.
- Permitir distintos niveles de uso del producto para usuarios con distintos niveles de experiencia.

- Hacer transparente la interfaz al usuario, que debe tener la impresión de manipular directamente los objetos con los que está trabajando.
- Permitir al usuario personalizar la interfaz (presentación, comportamiento e interacción).
- Permitir al usuario manipular directamente los objetos de la interfaz.

3.3.3.2 SER DIRECTO

Debe permitir que los usuarios manipulen directamente los objetos del ambiente, es mucho más sencillo recordar a que se parece algo que la sintaxis de un comando. Una de las formas más fáciles de interactuar con la computadora es a través de la utilización de metáforas las cuales se basan en que el usuario debe reconocer en lugar de recordar.

3.3.3.3 SER INDULGENTE

Este concepto se basa en dejar que el usuario explore la aplicación, todas las acciones deben ser reversibles o corregibles, se debe advertir al usuario que la acción que se va a ejecutar es destructiva, a menudo la gente comete errores y por lo tanto debemos considerar eso y darle la oportunidad de corregirlas en caso de que hayan sido iniciadas por error.

3.3.3.4 REDUCIR LA CARGA DE MEMORIA DEL USUARIO.

La interfaz debe evitar que el usuario tenga que almacenar y recordar información. Para ello, debe seguir los siguientes principios:

- Aliviar la carga de la memoria de corto alcance (permitir deshacer, copiar y pegar; mantener los últimos datos introducidos).
- Proporcionar indicaciones visuales de dónde está el usuario, qué está haciendo y qué puede hacer a continuación

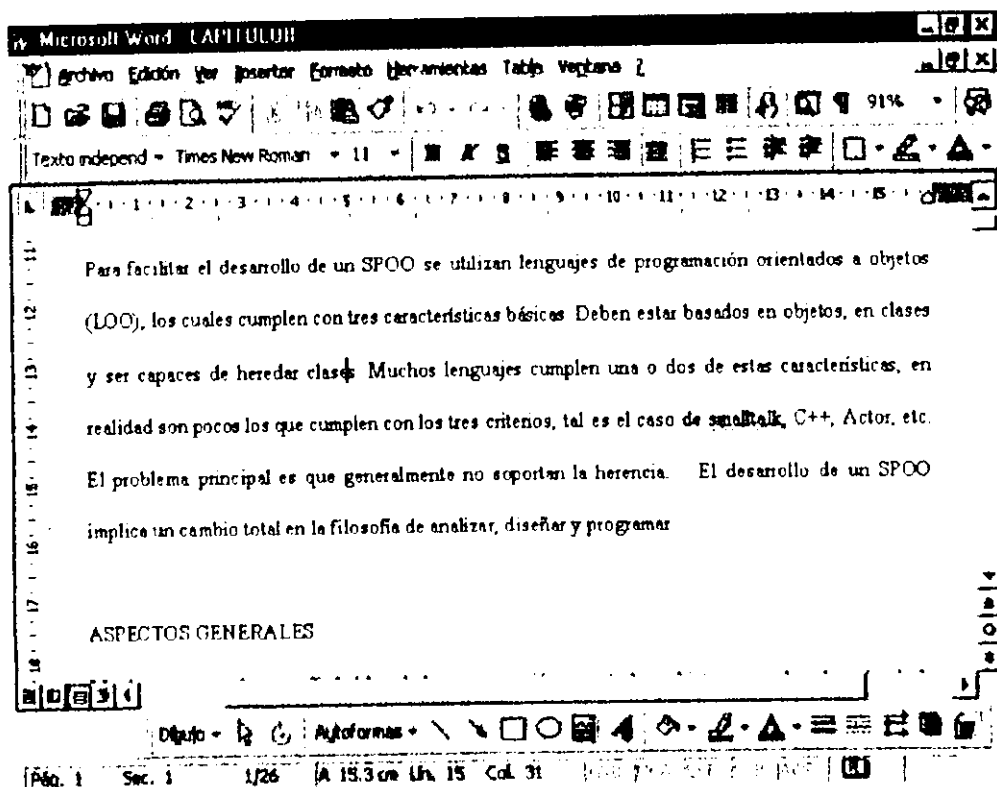


FIGURA 3.5 Indicaciones visuales en Microsoft Word

- Proporcionar atajos de teclado (iniciales en menús, teclas rápidas).
- Asociar acciones a los objetos (menú contextual).
- Utilizar metáforas del mundo real



FIGURA 3.6 Metáfora de la papelera de reciclaje en Windows 95

La primera vez que la comunicación con el ordenador se hizo con una metáfora de por medio fue en una interface gráfica desarrollada por la compañía Xerox, luego Apple Computer, primero con el ordenador Lisa pero sobre todo con el Macintosh, quien la popularizó. La metáfora de la interface del Macintosh consiste en un escritorio en el que la persona maneja, como en la realidad, documentos (textos, gráficos o partituras), carpetas donde pone estos documentos, una papelera en la que tira lo que no le interesa, etc. Gracias a esta metáfora, junto con una buena calidad gráfica, en informática personal hay un antes y un después del Macintosh, la prueba es que ha acabado siendo adoptada por todos los sistemas operativos. En la ideación de la metáfora, hoy en día, la libertad es tanta como en cualquier otro campo del diseño: hay campo para la imaginación. La metáfora ha de contar con los suficientes recursos audiovisuales para hacer atractiva la navegación, porque una metáfora pobre, conceptual y gráficamente empobrece el programa entero.

- Presentar al usuario sólo la información que necesita.
- Hacer clara la presentación visual (colocación/agrupación de objetos, evitar la presentación de excesiva información)

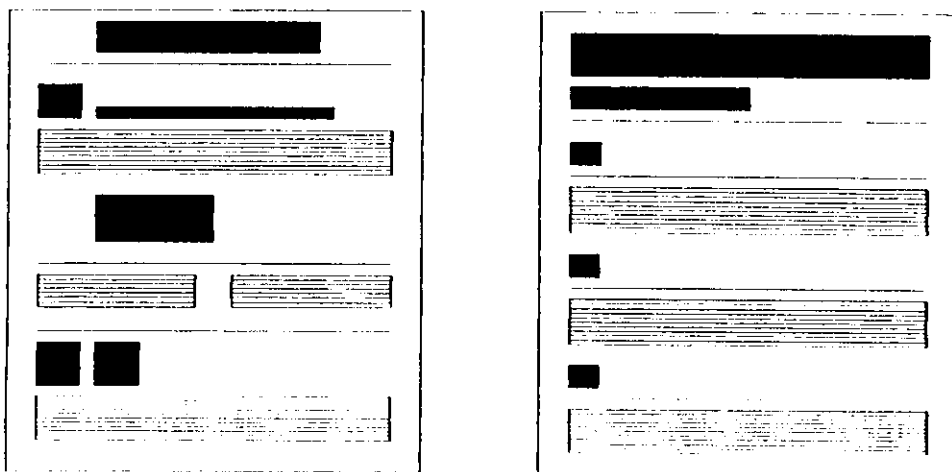


FIGURA 3.7 Ejemplo de mal (izquierda) y buen (derecha) diseño

3.3.3.5 BUSCAR RETROALIMENTACION

Es importante permitir que el usuario conozca que es lo que está pasando, se debe buscar una retroalimentación oportuna, para lograrlo se puede utilizar una combinación de pistas visuales y auditivas para que el usuario sienta que se está pendiente de ellos.

3.3.3.6 CUIDAR LA ESTETICA

La aplicación debe ser atractiva para el usuario, además de utilizar colores se debe cuidar el diseño, la ubicación de los elementos así como su número determina que tan usable es la pantalla. Varias investigaciones han determinado que lo más adecuado es darle al usuario de cinco a siete selecciones que es lo que puede retener en buena forma el cerebro humano al mismo tiempo, con más de nueve selecciones la gente tiende a confundirse y su cerebro a sobrecargarse.

3.3.3.7 SER CONSISTENTE

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

- Consistencia en la realización de las tareas: proporcionar al usuario indicaciones sobre el proceso que está siguiendo.
- Consistencia dentro del propio producto y de un producto a otro. La consistencia se aplica a la presentación (lo que es igual debe aparecer igual: color del texto estático), el comportamiento (un objeto se comporta igual en todas partes) y la interacción (los atajos y operaciones con el ratón se mantienen; el usuario espera los mismos resultados cuando interactúa de la misma forma con objetos distintos). Ejemplo de consistencia en la mejora de un interfaz: botones en las ventanas de Windows95



FIGURA 3.8 Botones de las ventanas de Windows 3.1 y Windows 95

- Consistencia en los resultados de las interacciones: misma respuesta ante la misma acción. Los elementos estándar del interfaz deben comportarse siempre de la misma forma (las barras de menús despliegan menús al seleccionarse).
- Consistencia de la apariencia estética (iconos, fuentes, colores, distribución de pantallas).
- Fomentar la libre exploración de la interfaz, sin miedo a consecuencias negativas.

3.3.4 GUIAS DE DISEÑO

Los estándares definen las características de los objetos y sistemas que usamos cada día. Ejemplo: disposición de las teclas en un teléfono. Existen estándares en muchas industrias, y también en la informática. Los principios anteriores no bastan para construir interfaces correctos. Las guías de diseño de interfaces afectan a la presentación, el comportamiento y la interacción de los elementos de la interfaz, y son reglas e indicaciones a seguir en cuanto a la apariencia y comportamiento de estos.

Las guías de diseño abarcan tres áreas del diseño de la interfaz: física (el hardware de la interfaz; por ejemplo, efecto de los botones de un ratón), sintáctica (presentación de la información y secuencia y orden de las acciones del usuario para realizar una tarea, como por ejemplo imprimir) y semántica (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: Apple (Macintosh), IBM (OS/2, DOS), Microsoft (Windows) y UNIX (OSF/Motif). Los desarrolladores deben seguirlas para conseguir la consistencia con el resto de productos. Ejemplo: logo de Windows95.

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

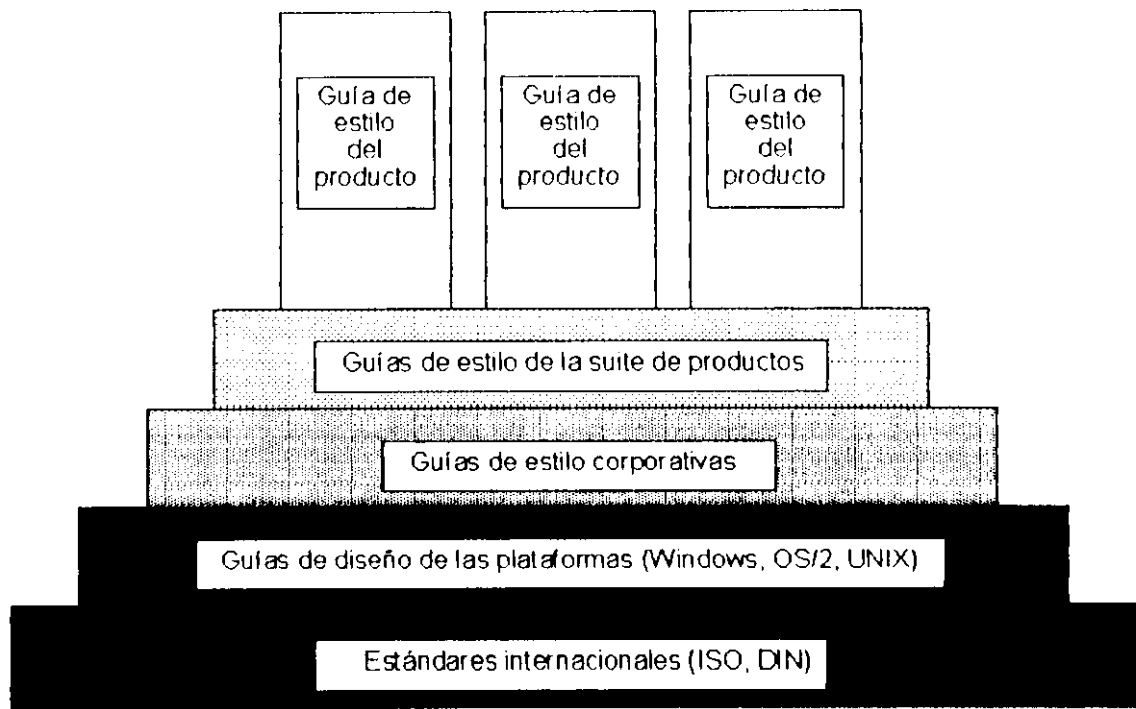


FIGURA 3.9 Pirámide de guías

En cualquier caso, aunque útiles, las guías de diseño son recomendaciones, y lo verdaderamente importante es que la usabilidad del programa sea adecuada, aunque ello suponga desviarse de las guías en un momento dado.

La usabilidad puede ser algo subjetivo, pero existen pruebas objetivas que permiten evaluar la usabilidad de un programa. Entre ellas se pueden citar la utilidad del programa, su facilidad de aprendizaje y de uso, su eficiencia y la satisfacción del usuario. Evaluar la usabilidad de un programa tiene un coste, pero a cambio tiene el beneficio de conseguir una mayor calidad del producto, por lo cual está plenamente justificada.

3.4 EVOLUCIÓN DE LAS INTERFACES DE USUARIO

La evolución de las interfaces de usuario corre en paralelo con la de los sistemas operativos; de hecho, la interfaz constituye actualmente uno de los principales elementos de un sistema operativo. A continuación se muestran las distintas interfaces que históricamente han ido apareciendo, ejemplificándolas con las sucesivas versiones de los sistemas operativos más populares.

3.4.1 INTERFACES DE LÍNEA DE MANDATOS (command-line user interfaces, CUIs).

Es el característico del DOS, el sistema operativo de los primeros PC, y es el estilo más antiguo de interacción hombre-máquina. El usuario escribe órdenes utilizando un lenguaje formal con un vocabulario y una sintaxis propia (los mandatos en el caso del DOS). Se usa un teclado, típicamente, y las órdenes están encaminadas a realizar una acción.

El usuario no suele recibir mucha información por parte del sistema (ejemplo: indicador del DOS), y debe conocer cómo funciona el ordenador y dónde están los programas (nada está oculto al usuario). El modelo de la interfaz es el del programador, no el del usuario. Ejemplo del DIR-DEL-DIR, por la falta de información de respuesta del DOS. Otras veces, en cambio, es excesiva, por ejemplo la etiqueta del volumen en el DIR. Los principales inconvenientes son: carga de memoria del usuario (debe memorizar los mandatos; incluso la ayuda es difícil de leer); nombres no siempre adecuados a las funciones, significado de los mandatos mal comprendido a veces (varios mandatos con el mismo o parecido significado, como DEL y ERASE); es inflexible en los nombres (DEL y no DELETE). Entre

las principales ventajas tenemos: potente, flexible y controlado por el usuario, aunque esto es una ventaja para usuarios experimentados. La sintaxis es estricta, y los errores pueden ser graves.

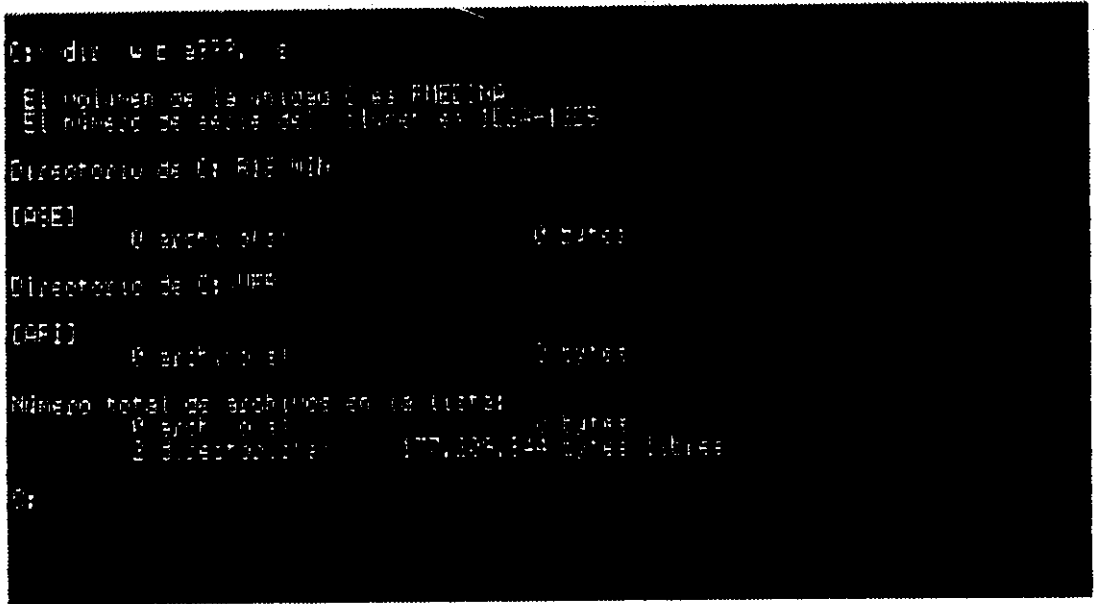


FIGURA 3.10 Interface de Línea de mandato - Comando DIR

En resumen, un CUI es adecuado para usuarios expertos, no para novatos. Para aquellos resultan más rápidos, por lo que se puede diseñar un CUI como parte de una interfaz, para que se pueda utilizar una vez que se tenga experiencia.

3.4.2 INTERFACES DE MENÚS

Un menú es una lista de opciones que se muestran en la pantalla o en una ventana de la pantalla para que los usuarios elijan la opción que deseen. Los menús permiten dos cosas: navegar dentro de un sistema, presentando rutas que llevan de un sitio a otro, y seleccionar elementos de una lista, que representan propiedades o acciones que los usuarios desean realizar sobre algún objeto.

Las interfaces de menús aparecen cuando el ordenador se vuelve una herramienta de usuario y no sólo de programadores. Las actuales interfaces gráficas u orientadas a objetos siguen utilizando este tipo de interfaces (los distintos estilos de interfaces no son mutuamente exclusivos).

Existen distintos tipos de menús. Los primeros fueron los menús de pantalla completa, estructurados jerárquicamente (Figura 3.11). Los menús de barra, situados en la parte superior de la pantalla, son profundamente utilizados en las aplicaciones actuales. Contienen una lista de acciones genéricas que dan paso a menús desplegables donde se concretan (Figura 3.12); estos menús pueden llevar a su vez a otros: son los menús en cascada (Figura 3.13). Pueden cambiar dinámicamente, y deshabilitar opciones que no estén disponibles en un momento dado (marcándolas habitualmente en gris). Las paletas o barras de herramientas (Figura 3.14) son menús gráficos con acciones, herramientas y opciones que se pueden colocar en la pantalla. Se utilizan mucho en programas gráficos. Los menús contextuales o menús pop-up (Figura 3.15) son los más recientes. Se llaman así porque el contenido del menú depende del contexto de trabajo del usuario. Contienen únicamente las opciones que son aplicables al objeto seleccionado, más algunas de uso frecuente que también son accesibles desde el menú de barra.

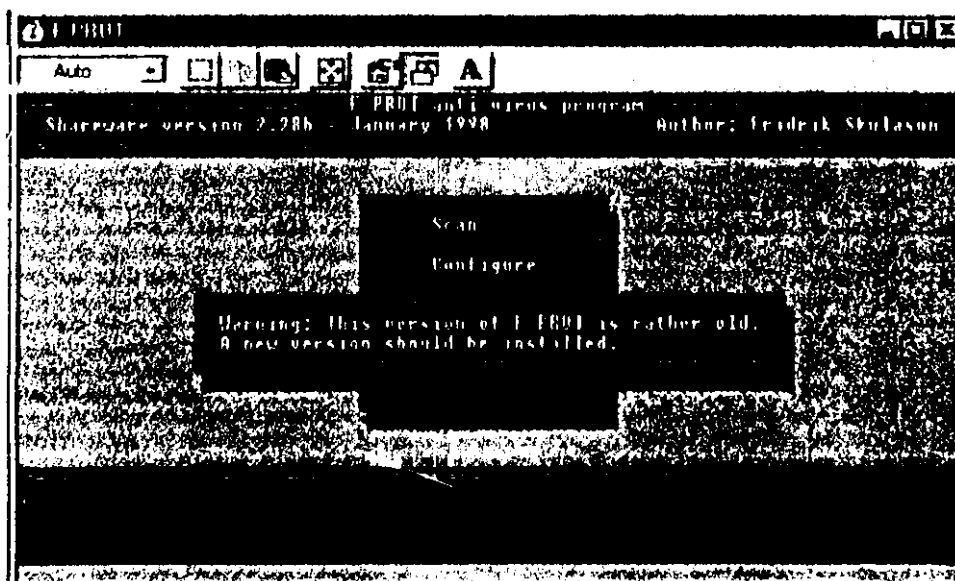


FIGURA 3.11 Menú de pantalla completa (Antivirus F-Prot)

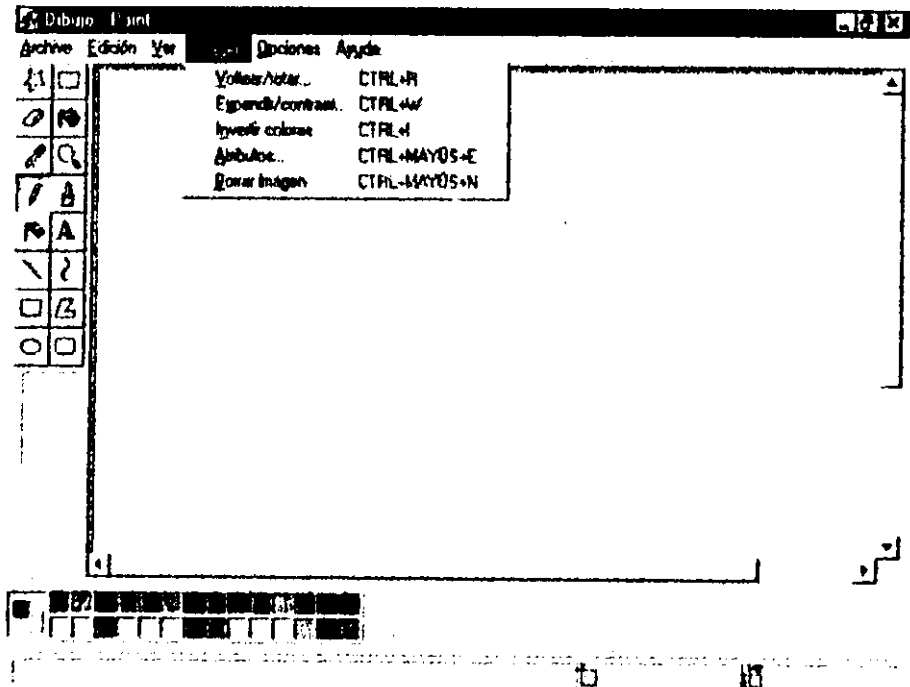


Figura 3.12 Menú de barra y menú desplegable

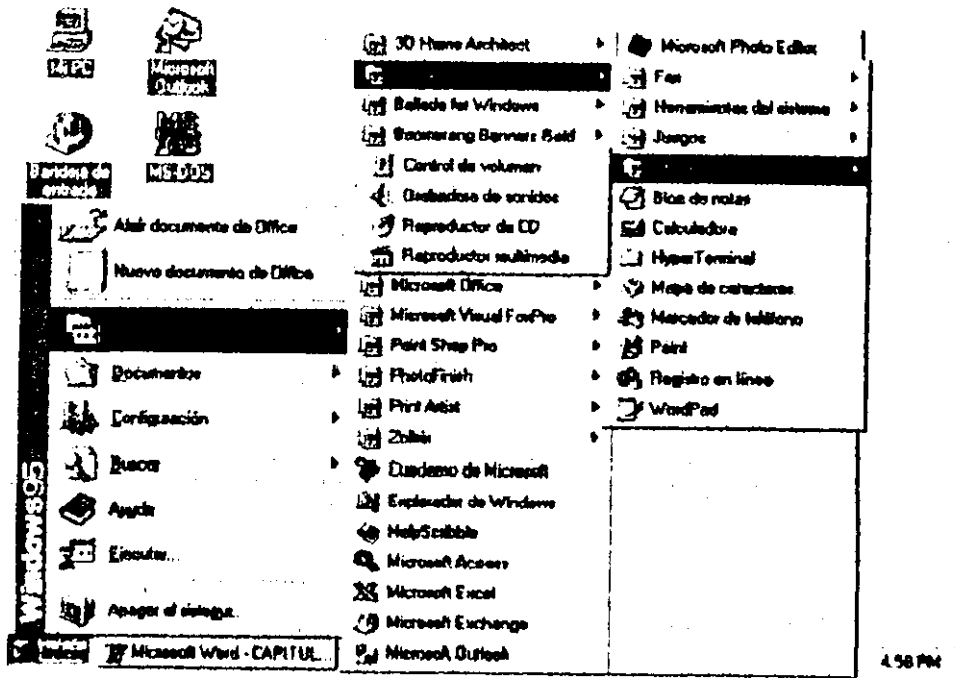


FIGURA 3.13 Menús en cascada de la barra de inicio de Windows 95

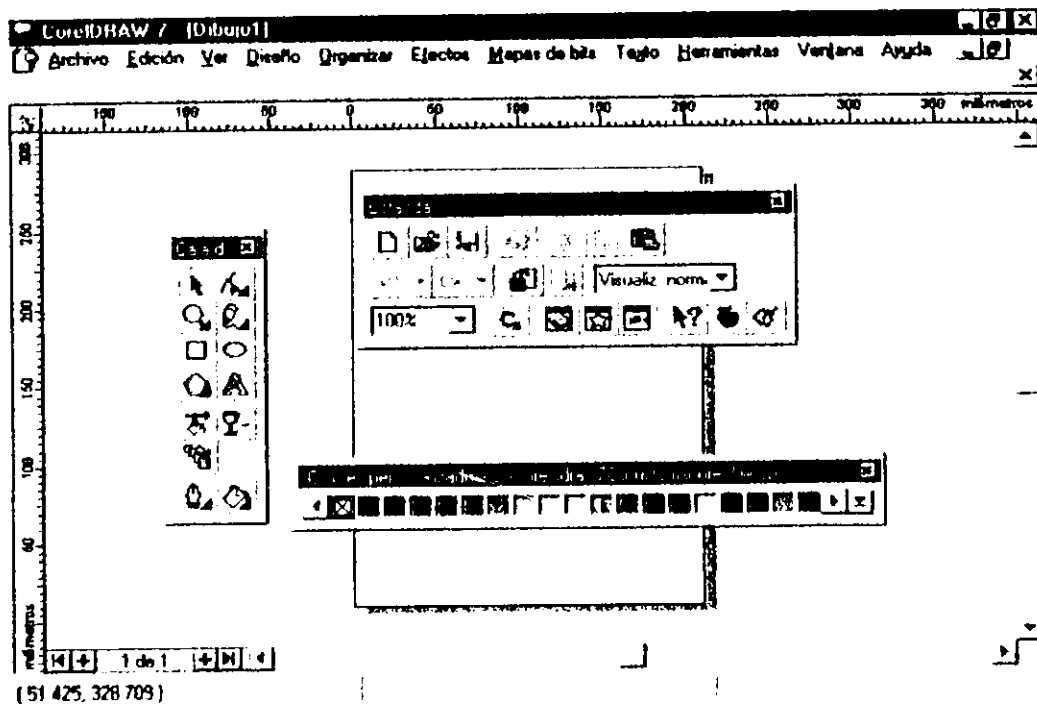


Figura 3.14 Paletas de herramientas en Corel Draw

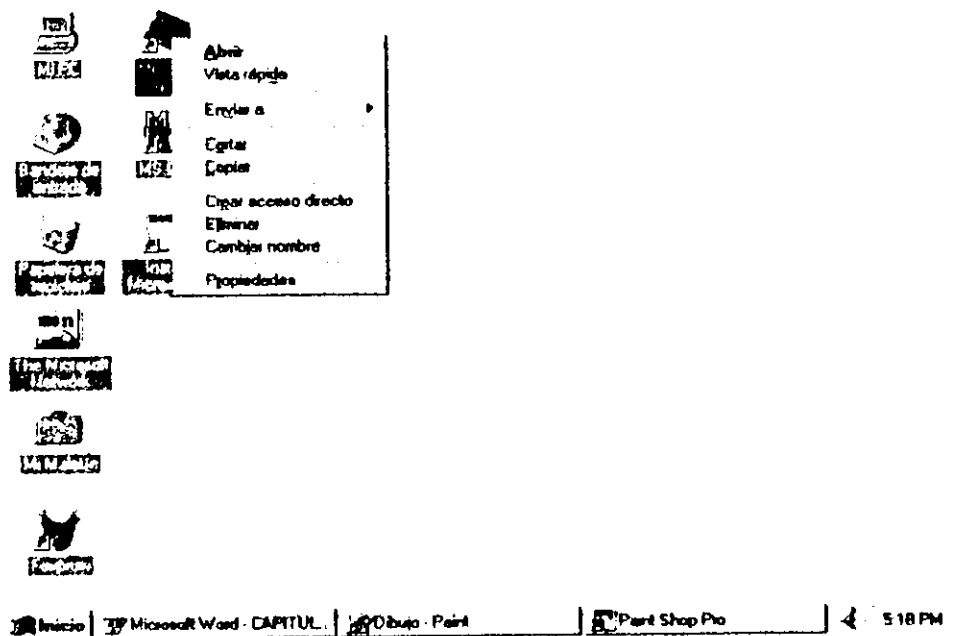


FIGURA 3.15 Menú contextual de un icono en el escritorio de Windows 95

Las interfaces de menús, bien estructuradas, son buenas para usuarios noveles o esporádicos. Son fáciles de aprender y de recordar. Pueden existir menús simples y avanzados, para adaptarse al tipo de usuario. Se debe tomar las siguientes precauciones: no ocupar demasiado espacio de la pantalla, recordar la información acumulada de menús precedentes, no colocar demasiados elementos en el menú, agruparlos de manera lógica, permitir la personalización por parte del usuario, usar una terminología adecuada y consistente dentro del programa y con otros programas (Exit, Quit, Escape, Close, Return, Back). Las interfaces de menús serán utilizadas normalmente en conjunción con los otros estilos de interfaces.

3.4.3 INTERFACES GRAFICAS (graphical user interfaces, GUIs).

Desarrolladas originalmente por XEROX (sistema Xerox Star, 1981, sin éxito comercial), luego fueron popularizadas por Apple (Steven Jobs se inspiró en los trabajos de Xerox y creó el Apple Lisa, 1983, sin éxito, y Apple Macintosh en 1984, con éxito debido en gran medida a su campaña publicitaria). Un GUI es una representación gráfica en la pantalla del ordenador de los programas, datos y objetos, así como de la interacción con ellos. Un 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.

Entre las principales características de una interface gráfica GUI, podemos mencionar:

- Posee un monitor gráfico de alta resolución.
- Posee un dispositivo apuntador (típicamente un ratón).
- Promueve la consistencia de la interfaz entre programas.
- Los usuarios pueden ver en la pantalla los gráficos y textos tal como se verán impresos.
- Sigue el paradigma de la interacción objeto-acción.
- Permite la transferencia de información entre programas.

- Se puede manipular en la pantalla directamente los objetos y la información.
- Provee elementos de interfaz estándar como menús y diálogos.
- Existe una muestra visual de la información y los objetos (iconos y ventanas).
- Proporciona respuesta visual a las acciones del usuario.
- Existe información visual de las acciones y modos del usuario/sistema (menús, paletas).
- Existen controles gráficos (widgets) para la selección e introducción de la información.
- Permite a los usuarios personalizar la interfaz y las interacciones.
- Proporciona flexibilidad en el uso de dispositivos de entrada (teclado/ratón).

Una característica importante es que el GUI permite manipular los objetos e información de la pantalla, no sólo presentarla.

Para usar un GUI, los usuarios deben conocer o aprender una serie de conceptos: organización del sistema (ficheros, directorios en Win95), diferentes tipos de iconos y efecto de las acciones sobre ellos, elementos básicos de una ventana, uso de los controles del GUI, uso del ratón.

Los GUI usan el estilo objeto-acción, en contraposición al acción-objeto de los CUI o las interfaces de menú. El usuario selecciona un objeto, y después la acción a realizar sobre dicho objeto. Los objetos son el principal foco de atención del usuario, lo cual resulta más natural y próximo a su modelo mental.

Su principal inconveniente es que son orientados a la aplicación. En la mayoría de los GUI, el usuario selecciona una aplicación y luego especifica los datos con los que va a trabajar. La orientación al objeto está implementada dentro de algunas aplicaciones. Las aplicaciones tienen un menú de barra con unas operaciones estándar (File, Edit, View, Help). Su principal ventaja es que usan metáforas de la vida real que se adaptan al modelo mental del usuario: escritorio, sala de juegos, agenda, cámara, etc.

3.4.4 INTERFACES ORIENTADAS A OBJETOS

Su aspecto es similar al de las GUIs. La diferencia estriba en el modelo subyacente: las GUIs son interfaces orientadas a la aplicación, mientras que las OOUIs están orientadas al objeto. La tabla siguiente muestra las principales diferencias entre ambos estilos de interfaz:

El objetivo de la OOUI es que el usuario se concentre en sus tareas en lugar de en el ordenador y cómo utilizar las aplicaciones y ficheros necesarios para cumplir sus objetivos. Por ello se esconde la organización del sistema al usuario (Ejemplo de los accesos directos en Windows95-OS/2).

El estilo de interacción de los OOUIs es el de objeto-acción (también se da en los GUIs, aunque mezclado con el estilo acción-objeto). La ventana es un objeto ventana, no una ventana de aplicación; desaparecen pues los menús de barra y ganan terreno los contextuales

INTERFACES ORIENTADAS A LA APLICACIÓN	INTERFACES ORIENTADAS A OBJETOS
La aplicación consiste en un icono, una ventana principal y varias secundarias	El producto consiste en una colección de objetos que cooperan y vistas de dichos objetos
Los iconos representan aplicaciones o ventanas abiertas	Los iconos representan objetos que se pueden manipular directamente
Los usuarios deben abrir una aplicación antes de trabajar con objetos	Los usuarios abren objetos como vistas en el escritorio
Proporciona al usuario las funciones necesarias para realizar las tareas	Proporciona al usuario los materiales necesarios para realizar las tareas
Se centra en la tarea principal determinada por la aplicación	Se centra en las entradas y salidas de los objetos y tareas
Las tareas relacionadas son soportadas por otras aplicaciones	Las tareas relacionadas son soportadas por el uso de otros objetos
Estructura rígida: función	Estructura flexible: objeto
Los usuarios pueden quedar atrapados en una tarea	Los usuarios no deben quedar atrapados en una tarea
Los usuarios deben seguir la estructura de la aplicación	Los usuarios pueden realizar tareas a su propio gusto
Se requieren muchas aplicaciones: una por tarea	Se requieren pocos objetos, que se reutilizan en muchas tareas

Los objetos se pueden clasificar en tres categorías: datos, contenedores y dispositivos. Sobre ellos se definen distintas vistas (por ejemplo, la ayuda constituye una vista del objeto). Definir los objetos y las vistas es lo más complicado del diseño de la interfaz. El objeto debe ser familiar al usuario (encajar con su modelo mental, apoyado en su vida diaria), y estar relacionado con el mundo real: uso de las metáforas.

Un ejemplo de lo que se pretende con una interfaz OOH es el considerar un documento como un objeto sobre el cual realizar tareas tales como incorporar gráficos y textos, sin necesidad de usar programas distintos para cada una de ellas. Estos programas suelen tener funciones que se solapan, con el consiguiente gasto extra en espacio y dinero. Actualmente existe una mezcla de productos orientados a la aplicación y al objeto, aunque se está produciendo una migración a estos últimos. Las aplicaciones están dejando paso a conjuntos de objetos.

3.5 PROCESO DE DISEÑO DE INTERFACES DE USUARIO

En el proceso de diseño de una interfaz de usuario se pueden distinguir cuatro fases:

- Reunir y analizar la información del usuario
- Diseñar la interfaz de usuario
- Construir la interfaz de usuario
- Validar la interfaz de usuario

3.5.1 REUNION Y ANALISIS DE LA INFORMACIÓN DEL USUARIO

Qué tipo de usuarios van a utilizar el programa, qué tareas van a realizar los usuarios y cómo las van a realizar, qué exigen los usuarios del programa, en qué entorno se desenvuelven los usuarios (físico, social, cultural).

3.5.2 DISEÑO DE LA INTERFAZ DE USUARIO

Es importante dedicar tiempo y recursos a esta fase, antes de entrar en la codificación. En esta fase se definen los objetivos de usabilidad del programa, las tareas del usuario, los objetos y acciones de la interfaz, los iconos, vistas y representaciones visuales de los objetos, los menús de los objetos y ventanas. Todos los elementos visuales se pueden hacer primero a mano y luego refinar con las herramientas adecuadas.

3.5.3 CONSTRUCCION DE LA INTERFAZ DE USUARIO

Es interesante realizar un prototipo previo, una primera versión del programa que se realice rápidamente y permita visualizar el producto para poderlo probar antes de codificarlo definitivamente.

3.5.4 VALIDACION DE LA INTERFAZ DE USUARIO

Se deben realizar pruebas de usabilidad del producto, a ser posible con los propios usuarios finales del mismo. Es importante, en suma, realizar un diseño que parta del usuario, y no del sistema.

3.6 TÉCNICAS AVANZADAS PARA EL DISEÑO DE INTERFACES DE USUARIO

Dada la importancia que tiene el diseño de la interface del usuario en la elaboración de un proyecto, varios investigadores han invertido mucho tiempo en el estudio y desarrollo de nuevas técnicas que facilitan a la construcción de una buena interface tanto a usuarios novatos como a usuarios especializados.

3.6.1 PRESENTACIÓN DE INFORMACIÓN

No se deben colocar demasiados objetos en la pantalla, y los que existen deben estar bien distribuidos. Cada elemento visual influye en el usuario no sólo por sí mismo, sino también por su combinación con el resto de elementos presentes en la pantalla.

En el ejemplo de la Figura 3.16, el número de elementos visuales que perciben son: a) 1 (el fondo); en b) 3 (la línea, lo que está encima y lo que está debajo); en c) son 5 (el espacio fuera del recuadro, el recuadro, la línea y el espacio encima y debajo de ésta); finalmente, en d) el número se eleva a 35, siguiendo el mismo criterio.

Como conclusión podemos decir que cada elemento nuevo que se añade influye más de lo que se piensa en el usuario.

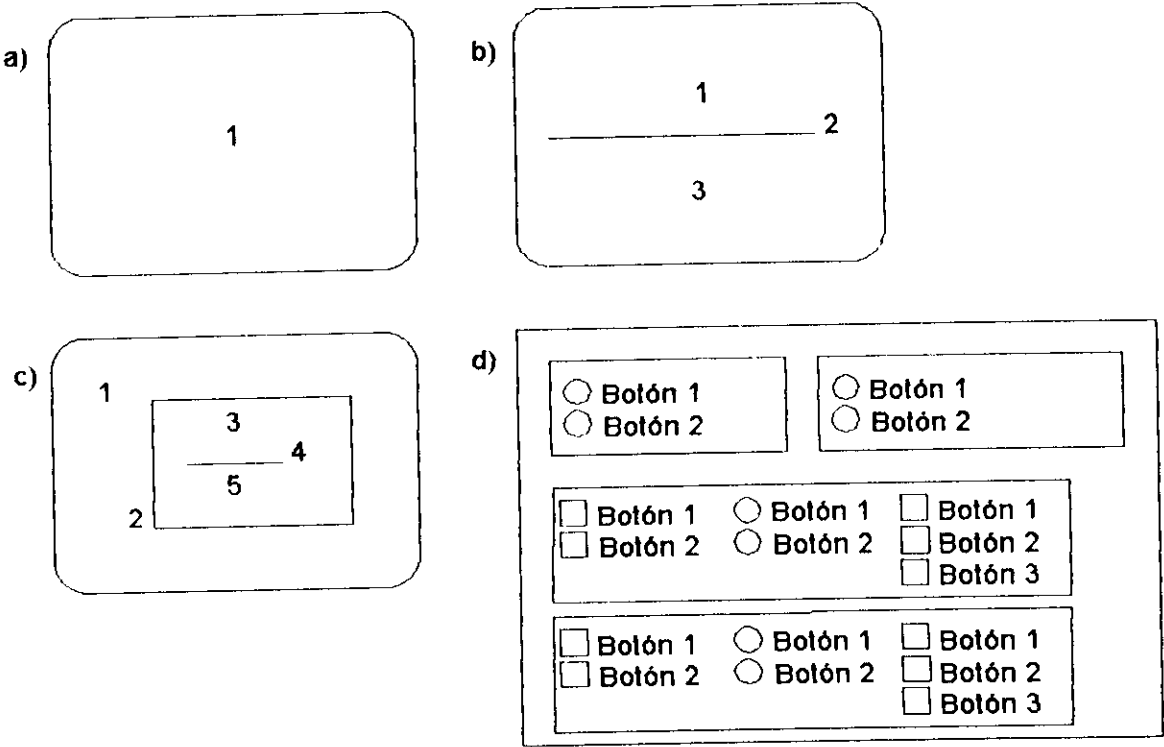


FIGURA 3.16 Elementos de diseño de pantalla y su percepción visual

3.6.2 COLOR

El color es probablemente el elemento de la interfaz que con más frecuencia es mal utilizado. El color comunica información, no es sólo decorativo (ejemplo: reforzar mensajes de error). Deben utilizarse combinaciones adecuadas (por ejemplo, las paletas proporcionadas por los sistemas operativos). El color debe atraer la atención, pero no cansar después de un rato de trabajo. Es especialmente importante seguir las líneas de diseño existentes. Principio básico: diseñar primero en blanco y negro, y luego añadir el color.

3.6.3 AUDIO

Primero es preciso ver cuándo es más apropiado que la información visual. Segundo, determinar el sonido adecuado. Tercero, permitir la personalización (volumen y desactivación). Como en el caso de los colores existen guías de uso. En lugares de trabajo abiertos, puede ser poco efectivo; además, puede ser embarazoso para algunas personas. El sonido debe usarse para informar, no cuando no añade nada nuevo (por ejemplo, un mensaje de aviso de correo o de bienvenida, respectivamente, al iniciar una sesión de trabajo).

3.6.4 ANIMACIÓN

Se define como un cambio en el tiempo de la apariencia visual de un elemento gráfico. Ejemplos de su uso: progreso de acciones (copia de ficheros en Windows 95, instalación de programas), estado de procesos (iconos de impresora), acciones posibles (cambios en el cursor al desplazar el ratón). La animación puede ayudar a subrayar iconos importantes, mostrar el estado de un objeto particular o explicar su comportamiento.

3.6.5 DISEÑO INTERNACIONAL

Debe hacerse un uso adecuado de la terminología. Hay mucho trabajo en este campo. Debe tenerse cuidado con las diferencias culturales (gestos, terminología, dibujos, formatos de teléfonos o calendarios, etc.).

3.6.6 ELECCIÓN DE CONTROLES

Muchas veces existe la duda de qué controles utilizar, en realidad no existe una única forma correcta. Un aspecto a considerar es la escalabilidad (menú de 10/1000 elementos; ejemplo: programas del menú inicio de Windows 95). Ejemplo de alternativas: usar un menú de barra o de paleta, permitir arrastrar objetos o no.

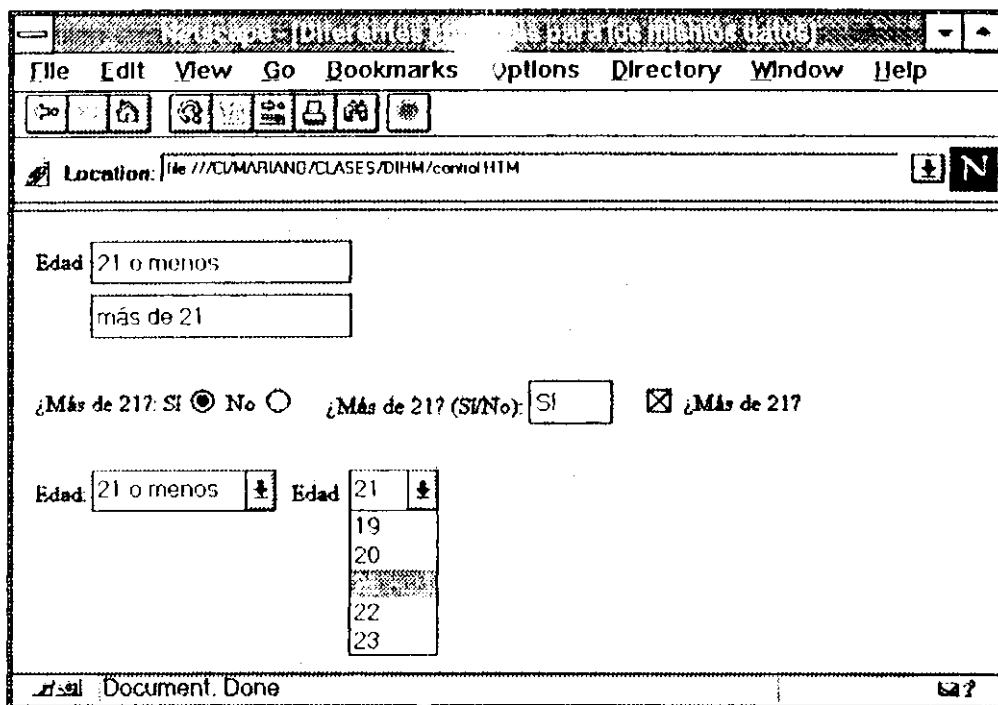


FIGURA 3.17 Diferentes controles para los mismos datos

Además de todas las reglas para diseño de interfaces de usuario estudiadas, existen diversas guías de diseño proporcionadas por expertos y comités, que las complementan, algunas de ellas son las siguientes:

- Demasiada simetría puede hacer las pantallas difíciles de leer.
- Si se ponen objetos sin alinear, hacerlo drásticamente.
- Asimetría = activo, simetría = sereno.
- Elementos de tamaño y color similares se perciben como pertenecientes a un grupo.
- Asumir errores en la entrada del usuario.
- Diseñar para el usuario, no para demostrar los propios conocimientos tecnológicos.
- Unos gráficos espectaculares no salvarán a una mala interfaz.

3.6.7 DISEÑO DE UNA INTERFAZ GRAFICA

Una interfaz gráfica es un tipo de formato de despliegue que permite al usuario elegir comandos, arrancar programas y ver las listas de archivos y otras opciones con sólo apuntar a representaciones pictóricas (íconos) y menús de elementos en la pantalla. Las selecciones pueden activarse con el teclado o con el ratón, un ejemplo de una GUI es Windows 95.

Los usuarios vivieron muchos años utilizando líneas de comandos y esta situación se hubiera prolongado por mucho tiempo más, con el apareamiento de las distintas versiones de Windows los usuarios pudieron apreciar todo el potencial que este tipo de interface ofrece para poder mejorar la productividad.

Hoy en día las GUIs se han convertido en una herramienta necesaria, nos permite acceder a muchas aplicaciones en un corto tiempo sin que sea necesario terminar una aplicación para acceder a otra. Si no tuviéramos las GUI gastaríamos un tiempo valioso arrancando y cerrando aplicaciones sin llevar a cabo un trabajo real.

3.7 HERRAMIENTAS DE DESARROLLO DE PROTOTIPOS

El término “ Desarrollo rápido de prototipos” se utiliza para describir la práctica de crear modelos de GUI desde los primeros pasos del proceso de diseño de una aplicación. Esto permite crear una interfaz a los clientes y obtener retroalimentación sobre la impresión que tienen de ella, la ubicación de los objetos y la inclusión de características, así como dar a los clientes un sentimiento de servicio debido a que los usuarios sienten que sus necesidades están siendo atendidas.

Hoy en día existen en el mercado varios productos que le permiten cumplir con esta tendencia cada vez más importante, estas herramientas rápidas para hacer prototipos construyen interfaces “falsas” que en el fondo no hacen ningún procesamiento de datos.

La nueva generación de herramientas de desarrollo rápido de aplicaciones RAD, como Delphi o Visual Basic, estas herramientas rápidas de prototipos ya no son necesarias, ahora se pueden utilizar las herramientas RAD para crear interfaces fantasma que pueden ser mostradas al cliente. Al usar un conjunto de herramientas de última generación se pueden desarrollar aplicaciones rápidas con un tiempo de respuesta muy corto.

Este tipo de herramientas permiten realizar grandes avances en funcionalidad con muy poco trabajo, como ejemplo típico del uso de estas herramientas tenemos los cuadros de diálogo comunes de Windows 95 para abrir y cerrar archivos en lugar de escribirlos desde cero o la utilización de un contenedor OLE para incrustar una hoja de cálculo.

La gran desventaja de la utilización de estas herramientas es la duplicación de esfuerzo, cuando un programador creaba una interfaz con alguna de ellas, luego de que el cliente la aprobara tenía que volver a codificar dicha interfaz en la herramienta para generar la aplicación en tiempo real. Con Delphi o Visual Basic se puede diseñar la interfaz y luego de que el usuario final la apruebe puede terminar la aplicación en el mismo ambiente.

Otro elemento fundamental en el diseño de una interfaz gráfica de usuario es el equipo de desarrollo, el cual debe estar formado por personas de diferentes áreas para que el diseño sea utilizable:

- Escritores
- Expertos en interfaces
- Especialistas en “usabilidad”
- Usuarios de computadoras de varios niveles de habilidad

Un consejo importante es probar el producto con una persona que no tenga nada que ver con computadores o mucho mejor con alguien que les tema. Además de todos estos aspectos que siempre se deben cuidar al momento de diseñar una interface gráfica se debe recordar: ser directo, indulgente, buscar la retroalimentación, estética y simplicidad

Se debe tratar de diseñar un software intuitivo, permitir que los usuarios vean como sus acciones afectan a los objetos en la pantalla. Una de las formas más directas en que los usuarios pueden interactuar con una computadora es mediante el uso de metáforas, por ejemplo el concepto de una carpeta tiene más sentido que un directorio, las metáforas se basan en el concepto de que los usuarios deben reconocer en lugar de recordar, hay que permitir que los usuarios manipulen directamente los objetos del ambiente, ellos pueden recordar mejor un significado asociado a un objeto en lugar de un comando.

Si todas las aplicaciones son consistentes en la forma en que presentan sus datos y la forma de interactuar con el usuario, el usuario podrá invertir su tiempo en otras tareas y no en entender la forma en que la aplicación interactúa con él. Se deben tomar en cuenta los siguientes aspectos:

- Hacer que la aplicación interactúe de forma similar a Windows, de esta forma los usuarios podrán emplear las habilidades que han aprendido de esta plataforma.
- El producto debe ser consistente consigo mismo, no se deben utilizar paradigmas diferentes en diferentes pantallas para una misma operación
- Utilizar metáforas consistentes.

3.8 TÉCNICAS Y HERRAMIENTAS PARA USAR COLOR EN EL DISEÑO DE LA INTERFAZ DE UNA COMPUTADORA

Las computadoras contemporáneas emplean predominantemente interfaces gráficas con el usuario (Graphical User Interfaces, GUIs) y el color es una componente principal de una GUI. Cada

interfaz hombre-máquina esta compuesta por dos partes principales: el hombre y la máquina. Las interfaces a color no son diferentes en ese sentido ya que ellas también están basadas en dos partes, el Sistema Humano Visual (Human Visual System, HVS) y un sistema de despliegue a color. Un examen teórico de estos dos componentes establece un fundamento para desarrollar guías prácticas para interfaces a color.

3.8.1 MOTIVACIÓN

El color tiene un impacto principal sobre la interacción humano-computadora: si no positivo, entonces negativo.

"El color puede ser una herramienta poderosa para mejorar la utilidad de un despliegue de información en una amplia variedad de áreas si el color es usado adecuadamente. Inversamente, el uso inapropiado del color puede seriamente reducir la funcionalidad de un sistema de despliegue"⁽⁷⁾.

El color es un componente principal en las GUIs. Debido a la proliferación de aplicaciones GUI en PCs, Macs, y Estaciones de trabajo Unix, el examen del color es pertinente a la interacción humano-computadora (human computer interaction, HCI).

El uso de color apropiado puede ayudar a la memoria del usuario y facilitar la formación de modelos mentales efectivos. Como Murch expresa, el uso efectivo del color puede ser una herramienta poderosa. Sin embargo, el uso inefectivo del color puede degradar el desempeño de una aplicación y disminuir la satisfacción del usuario. Debido a estos factores, el uso efectivo del color en interfaces de computadora es un importante tópico en HCI que requiere ser examinado cuando consideramos el diseño de una interfaz.

⁷ Murch, *Physiological Principles for the effective use of Color*

3.8.2 FUNDAMENTOS DEL COLOR

Para entender el potencial del color en las interfaces necesitamos examinar algunas características fundamentales de la percepción del color. Las características fundamentales primarias que discutimos aquí incluyen varios modelos de color, el sistema visual humano, principios fisiológicos del color y efectos del color tales como las ilusiones y las combinaciones del color.

Hay varios modelos de color que son organizados en dos divisiones básicas. Estas divisiones son los modelos basados en la percepción y los modelos basados en el despliegue. Como pudiera pensarse de estos nombres, el primero es organizado similarmente a la manera como percibimos el color y el segundo esta basado en las características de un dispositivo de despliegue. Los modelos basados en la percepción son conocidos como HSV (Matiz/Hue, Saturación/ Saturation y Valor/Value) y el HSL (Hue, Light/luz y Saturation).

El Matiz es la composición de la longitud de onda espectral de color que produce los colores que vemos tales como el anaranjado, azul, etc. La Saturación (chroma) es la pureza relativa del color sobre una escala del gris al tono más vibrante del color particular. El Valor es la fuerza u oscuridad del color. La Claridad también referida como brillantez, se refiere a la cantidad de energía luminosa que crea el color. El sistema HSV está basado en el sistema de color Munsell usado por artistas, diseñadores y fabricantes. El HSL fue desarrollado por Tektronix en 1978 para proporcionar un modelo más perceptual que el modelo de Tektronix que se estaba usando en ese tiempo basado en despliegues. Los modelos más comunes basados en despliegues son el RGB (Red-Green-Blue) y el YIQ/YUV. El RGB es usado para el despliegue en monitores de computadora y el YIQ/YUV es usado para transmisión de televisión.

La Comisión Internacional sobre Iluminación es una organización mundial que desarrolló la primera versión del modelo medido espectralmente conocido como CIE en 1931. El CIE es una medida precisa espectral usada para precisar colores y remover la ambigüedad de ellos. El modelo

CIE es basado físicamente, por lo tanto, no se adecua bien dentro de cualquiera el modelo basado perceptualmente o las categorías basadas en despliegue.

Todos los colores presentados en una computadora deben ser trasladados dentro del espacio del color RGB. Desafortunadamente, no hay un mapeo uno a uno de los modelos basados perceptualmente a los modelos basados en despliegue. Este hecho puede explicar algunas de las dificultades encontradas cuando tratamos de recrear justamente el color correcto para una interfaz de pantalla. No es siempre posible obtener la sombra exacta.

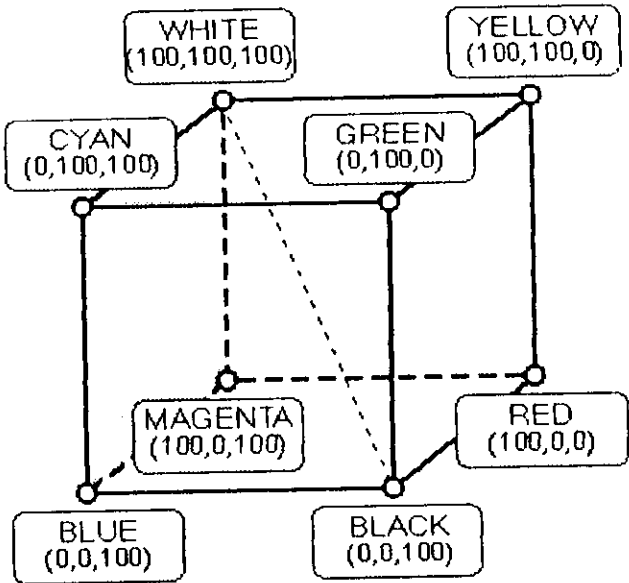


FIGURA 3.18 El Cubo RGB

El ojo humano contiene una lente y una retina. La retina contiene receptores sensitivos a la luz conocidos como bastones y conos. El propósito primario de los bastones es proporcionar visión de noche, mientras que los conos trabajan en niveles más altos de intensidad de la luz. Los conos contienen fotopigmentos, también conocidos como fotoreceptores, los cuales son sensitivos al rojo, al verde o al azul.

De acuerdo con Murch, aproximadamente 64% de los conos contienen fotopigmentos rojos, 32% contiene verdes y solamente alrededor de 2% contienen fotopigmentos azules. Las propiedades fisiológicas del sistema nervioso dictan la sensación del color. Los humanos son sensitivos a un rango de longitudes de onda. Las longitudes de onda no coloreadas, sin embargo el color es el resultado de la interacción de la luz y nuestro sistema nervioso. Las longitudes de onda que producen colores diferentes son enfocadas a distancias diferentes detrás de la lente.

La lente no transmite todas las longitudes de onda de la misma manera, exhibiendo menos sensibilidad a las longitudes de onda más cortas, lo cual tiene el efecto de absorber los azules. Inversamente, somos más sensitivos a las longitudes de onda más largas, lo cual es exhibido por una sensibilidad aumentada a los amarillos y anaranjados. Bastante raro, podemos ver los azules mejor en la periferia que en el frente debido a la distribución física de los fotoreceptores azules.

Consecuente a la organización física del ojo hay efectos interesantes o ilusiones causadas por ciertas organizaciones de color o combinaciones. Debido a la falta de fotoreceptores azules, las líneas azules delgadas (como el texto azul) tienden a verse borrosas, y pequeños objetos azules tienden a desaparecer cuando tratamos de enfocarlos. Los colores que difieren solamente por la cantidad de azul no producen bordes claros. Por ejemplo, los colores con la misma cantidad de verde y rojo que varían solamente en la cantidad del azul producen orillas borrosas. El contraste de colores adyacentes puede crear una ilusión observada fácilmente. Dos objetos del mismo color pueden aparecer marcadamente diferentes en color dependiendo del color del fondo

El uso ineffectivo de los colores puede causar vibraciones y sombras; imágenes que distraen al usuario y pueden forzar la vista.

3.8.3 MODELOS MENTALES Y EL USO EFECTIVO DEL COLOR

La gente interactúa con su mundo a través de modelos mentales que ellos han desarrollado. Específicamente, las ideas y las habilidades que traen a su trabajo están basadas en modelos mentales que ellos desarrollan acerca de ese trabajo. Como diseñadores de interfaces, necesitamos:

- Ayudar al usuario a desarrollar modelos mentales del sistema que le ayudarán a entender el trabajo, y
- Desarrollar las herramientas de la interfaz que le ayudarán a realizar el trabajo. El uso adecuado del color comunica hechos e ideas más rápidamente y más estéticamente al usuario. El color también puede ayudar a desarrollar modelos mentales eficientes y factibles si se siguen las siguientes pautas: simplicidad, consistencia, claridad y lenguaje del color.

3.8.4 SIMPLICIDAD

La simplicidad es importante en el diseño de interfaces a color. Existe una simplicidad inherente en el color la cual debería ser usada cuando se desarrolla el diseño. Los cuatro colores fisiológicamente primarios son el rojo, el verde, el amarillo y el azul. Estos colores son fáciles de aprender y recordar. Vinculando significados prácticos e intuitivos a estos colores simples cuando se diseña una pantalla, el diseñador de la interfaz enriquece el desarrollo del usuario con un modelo mental efectivo.

Mantener el esquema del color simple; según Miller el número mágico para la memoria de corto plazo es siete más menos dos. Al usar color en las interfaces el número no debe exceder cinco más menos dos. El número de colores en la pantalla deberá de ser limitado en conformidad con lo anterior. Si el usuario es agobiado o confundido por demasiados colores compitiendo por su atención, es muy poco probable que él desarrolle un modelo mental efectivo.

Mantener el mensaje sencillo. No sobrecargues el significado del color vinculando más de un concepto a un solo color. Los conceptos diferentes deberán usar colores diferentes. Mantener el mensaje simple se relaciona con la siguiente pauta, la cual es consistencia.

3.8.5 CONSISTENCIA

La consistencia es vital al asignar significados a los colores. El orden intuitivo de los colores puede ayudar a establecer consistencia intuitiva en el diseño. El orden espectral y perceptual rojo, verde, amarillo, azul puede guiar el orden de los conceptos vinculados a los colores. El rojo es primero en el orden espectral y se enfoca en el frente, el verde y el amarillo se enfocan en medio, mientras que el azul se enfoca en el fondo.

El color puede ser usado para codificar o agrupar piezas de información. Esto ayuda a incrementar el número de piezas de información que un usuario puede retener en la memoria de corto plazo. Es importante el ser consistente a la hora de "agrupar" información. Hay que evitar cambiar el significado de los colores para diferentes pantallas en la interfaz. Por ejemplo, una interfaz de base de datos que evaluamos cambia su color de fondo a azul cada vez que aparecen los campos en la pantalla que no pueden ser actualizados. Esto se hace para toda pantalla aplicable en el paquete y de esa manera proporciona continuidad y consistencia para el usuario.

Existen aspectos fisiológicos que impiden la consistencia en el uso del color. Varios matices del mismo color deberán de ser eliminados por diferentes conceptos e ideas. Esto es especialmente cierto para los azules. Los diferentes matices de azul son muy difíciles de distinguir y muchos pueden no ser reconocidos por el usuario. Si el concepto es diferente, use un color diferente. El diseñador deberá evitar el uso de colores que aparecen diferentes debido a la variación del color del fondo. Estos pueden ser percibidos por el usuario como colores diferentes y el significado se perderá.

COMBINATIONS FOR USER INTERFACES WITH GRAPHIC DISPLAYS		
BACKGROUND	BEST COLORS	WORST COLORS
WHITE	BLACK, BLUE	CYAN, YELLOW
BLACK	YELLOW, WHITE	BLUE
RED	BLACK	BLUE, MAGENTA
GREEN	BLACK, RED	CYAN
BLUE	RED, WHITE, YELLOW	BLACK
CYAN	BLUE, RED	GREEN, WHITE, YELLOW
MAGENTA	BLACK, BLUE	CYAN, GREEN
YELLOW	BLACK, BLUE, RED	CYAN, WHITE

Revised from T.A. Brown and S.C. Card, 1976. Reprinted from the book: *Principles of Computer User Interface Design*, Wiley & Sons, 1988. © John Wiley & Sons. Printed in Great Britain. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or by any information storage and retrieval system, without permission in writing from the publisher.

FIGURA 3.19 *Combinaciones de Colores para Interfaces con el Usuario con Despliegue Gráfico*

Además de usar estas combinaciones de color, también existen algunas reglas y sugerencias que son fáciles de seguir. Murch da diez reglas simples para crear buenas interfaces :

- Utilizar el color azul para el fondo
- Utilizar la secuencia de color espectral (rojo, anaranjado, amarillo, verde, azul, índigo y violeta)
- Mantener pequeño el número de colores
- Evitar usar colores adyacentes que difieren solamente en la cantidad de azules puros
- Utilizar colores brillantes para indicar peligro o para llamar la atención del usuario

Marcus también sugiere diseñar la interfaz primero en blanco y negro. Si se tiene una buena interfaz en blanco y negro y se le aplica color de una manera constructiva, se obtendrá una mejor interfaz.

3.9 LAS REGLAS DE MURCH

- Evitar el despliegue simultáneo de colores espectralmente extremos que estén altamente saturados.

- El color AZUL puro deberá de ser descartado para el texto, líneas delgadas y figuras pequeñas.
- Evitar colores adyacentes que se diferencien solo por la cantidad de azul que contienen.
- Los operadores de edad avanzada necesitan niveles más altos de brillo para distinguir los colores.
- Los colores cambian de apariencia a medida que el nivel de luz ambiental cambia.
- La magnitud de un cambio detectable en el color varía a través del espectro.
- Es difícil enfocar hacia las orillas creadas solamente por el color.
- Evitar utilizar el ROJO y el VERDE en la periferia de despliegues a gran escala.
- Los colores opuestos se ven bien juntos.
- Para los observadores que tienen deficiencias del color (ciegos al color), evita hacer distinciones de un solo color.

Uno de los elementos más importantes de usar el color efectivamente es conocer al usuario, el ambiente del usuario, y la tarea que el usuario está realizando. Esto es igualmente importante para la integración del color que para cualquier otra parte del diseño de la interfaz. Un ejemplo de una interfaz que utiliza efectivamente el color lo presenta Borland C++ 3.1 para DOS. En la pantalla del editor se utilizan seis colores básicos además del color azul de fondo. Estos colores son asignados a palabras (o cadenas) en el texto para categorizar el contenido textual:

- palabras reservadas - blanco
- identificadores - amarillo
- macros - verde
- letras - azul
- comentarios - gris
- errores de sintaxis - rojo

Este plan de color cuidadosamente escogido ayuda al usuario a reconocer los errores fácilmente (por ejemplo, palabras reservadas mal escritas), y lo alienta a desarrollar un buen modelo mental para el uso de la interfaz.

3.10 CONCLUSIONES

Algunos de los aspectos clave de usar efectivamente el color en el diseño de interfaces de computación conllevan una revisión. Hay que recordar la parte del hombre del sistema hombre-máquina cuando se diseñan interfaces de computadora, y observar los fundamentos del sistema visual del humano.

Se debe diseñar dentro de los principios fisiológicos señalados y las restricciones del despliegue. Utilizar el color con moderación, con consistencia, y con claridad para auxiliar en la formación de modelos mentales eficientes. Si el color es utilizado de esta manera, la interfaz resultante será más accesible para aprenderla y por consecuencia se utilizará más.

Finalmente, se debe utilizar las técnicas y las pautas desarrolladas por Murch, Marcus, Pancake y otros. Estas reglas, las cuales fueron establecidas de trabajos de investigación, describen como los colores encajan efectivamente. El ignorar estas reglas mientras establecemos un plan de color que nos agrada personalmente, puede ser perjudicial para el éxito de la interfaz resultante.

CAPITULO IV

ESTUDIO DE HERRAMIENTAS

4.1 INTRODUCCION

El proyecto de creación de una Plantilla Generadora de Guías Virtuales, considera dos niveles:

1 El nivel administrador

Desde donde el administrador alimentará la información requerida sobre la empresa o Institución de la cual se generará el guía, con los datos relativos a: su historia, departamentos, secciones, personal, etc.

Estos datos ingresados serán recuperados automáticamente utilizando un L.O.O a través de una plantilla diseñada especialmente para el efecto.

2 El nivel usuario

Que visualizará la información ingresada en el nivel administrador y desde donde utilizará el producto final previamente elaborado para conocer la empresa utilizando el computador

4.2 EL NIVEL ADMINISTRADOR

Para decidir que software se utilizaría, se realizó un estudio de los principales sistemas de gestión de base de datos en el mercado, analizando sus principales ventajas, herramientas y considerando sobre todo que permita un manejo óptimo de la información registrada.

4.2.1 ALIMENTACION DE LA INFORMACION

En los últimos años la gestión de bases de datos ha progresado notablemente pasando de ser una función propia de computadoras especializadas a ser el componente central del entorno informático del mundo moderno. La importancia que tiene la información en la mayoría de organizaciones y empresas, así como el tratamiento de la misma, ha llevado al desarrollo de una gran cantidad de conceptos y técnicas para la gestión eficiente de los datos.

En el nivel administrador el usuario debe en primer lugar introducir los datos que serán mostrados al usuario del Guía de la Institución, para eso se diseñará una base de datos utilizando como herramienta Microsoft ACCESS 97.

4.2.1.1 BASES DE DATOS

Una base de datos no es más que un conjunto de información relacionada que se encuentra agrupada o estructurada. Un archivo por sí mismo no constituye una base de datos, sino más bien la forma en que está organizada la información es la que da origen a la base de datos.

Desde el punto de vista informático una base de datos es un sistema formado por un conjunto de datos almacenados en memorias masivas que permiten un ingreso a ellos y un conjunto de programas que manipulan ese conjunto de datos.

Desde un punto de vista formal se puede definir a una base de datos como un conjunto exhaustivo de datos estructurados, fiables y homogéneos, organizados independientemente de su utilización y su implementación en máquina, accesibles en tiempo real, compartibles por usuarios concurrentes que tienen necesidades de información diferentes y no predecibles en el tiempo.

Al hablar de una base de datos, nos estamos refiriendo a una colección de datos:

- Estructurados independientemente de las aplicaciones y del soporte de almacenamiento que los contiene.
- Con la menor redundancia posible
- Que son compartidos por varios usuarios y aplicaciones
- Bajo control centralizado.

Entre la base de datos física y los usuarios del sistema existe una interfaz de software que recibe el nombre de SGBD, responsable de tratar todas las peticiones de los usuarios, es un conjunto de programas de propósito general que permite controlar el acceso y la utilización de las bases de datos, para incluir, modificar o recuperar información incluyendo prestaciones para conseguir la independencia, integridad y seguridad de datos y concurrencia de usuarios.

Actualmente todos los sistemas de gestión de base de datos modernos almacenan y tratan la información utilizando el modelo relacional. En una computadora existen diferentes formas de almacenar información, por lo tanto existen también distintos modelos de organización de bases de datos: Jerárquico, Red y Relacional

4.2.1.2 ACCESS 97

Microsoft Access es un sistema de gestión de base de datos totalmente funcional. Proporciona la definición, manipulación y control de los datos que se necesitan para gestionar grandes cantidades de los mismos.

Un SGBDR proporciona un control completo sobre la forma de definir los datos. Proporciona algunas características sofisticadas que permiten catalogar y gestionar cantidad de datos en muchas

tablas de forma sencilla. Access es muy funcional, presenta tres tipos de posibilidades: definición, manipulación y control de datos.

La siguiente figura muestra la potencia de Access como SGDB:

	Tablas	Formularios e Informes	Indagación De BD	Integ/Comp datos	Automatiz. Ayuda	Puntuación
Corel Paradox	7.1	7.5	7	9	4.5	7
Lotus Approach	8	8	7	6.6	6.5	7.2
Microsoft ACCESS	7.8	8.5	9	10	7.5	8.6

ARQUITECTURA DE ACCESS

Access da el nombre de objeto a todo lo que pueda poseer un nombre. Entre los principales objetos tenemos:

Tabla Objeto que se define y utiliza para almacenar datos. Una tabla contiene información sobre un tema o asunto en particular. Una tabla contiene campos que almacenan los diferentes tipos de datos y registros que recogen toda la información sobre una instancia determinada.

Consulta Este objeto proporciona la visión personal de los datos a partir de una o más tablas. Se pueden crear consultas para seleccionar, actualizar, insertar, eliminar datos o para crear nuevas tablas a partir de las existentes.

Formulario Este objeto fue diseñado para la introducción y visualización de los datos o control de la ejecución de la aplicación

Informe	Permite dar formato, calcular, imprimir y resumir los datos seleccionados.
Macro	Define de forma estructurada la acción o acciones que el usuario desea que Access realice en respuesta a un evento determinado.
Módulo	Objeto que contiene procedimientos personales que se codifican utilizando VBA

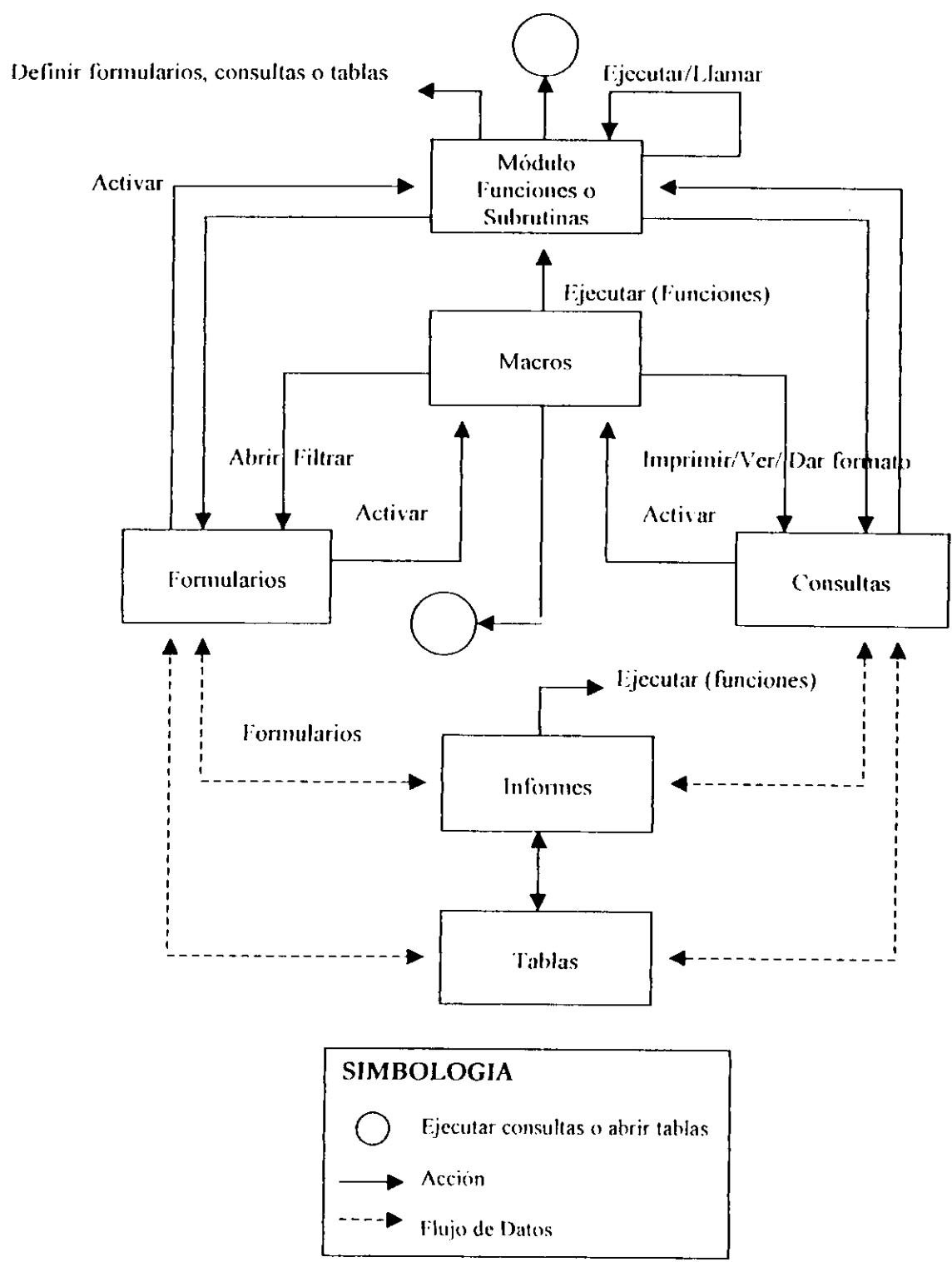


FIGURA 4.1 *Objetos Principales de Access y sus relaciones*

4.2.2 CONSTRUCCION DE LA PLANTILLA

Para esta etapa se han estudiado dos herramientas basadas en POO, Delphi versión 4.0 y Visual Basic versión 5.0. En cada caso se han realizado las pruebas necesarias para comparar los beneficios que ofrecen al usuario, la flexibilidad que proporcionan al desarrollador y principalmente la versatilidad de interface con bases de datos (Access).

4.2.2.1 DELPHI 4.0

En el Capítulo tres se detallan los fundamentos y componentes de la Programación Orientada a Objetos, por ello es lógico deducir que Delphi cumple y se acoge a todos estos conceptos.

Delphi es una herramienta de nueva generación de ambientes de desarrollo de software definidos actualmente como herramientas RAD (Desarrollo rápido de aplicaciones), en donde el programador dispone de herramientas más intuitivas y visuales.

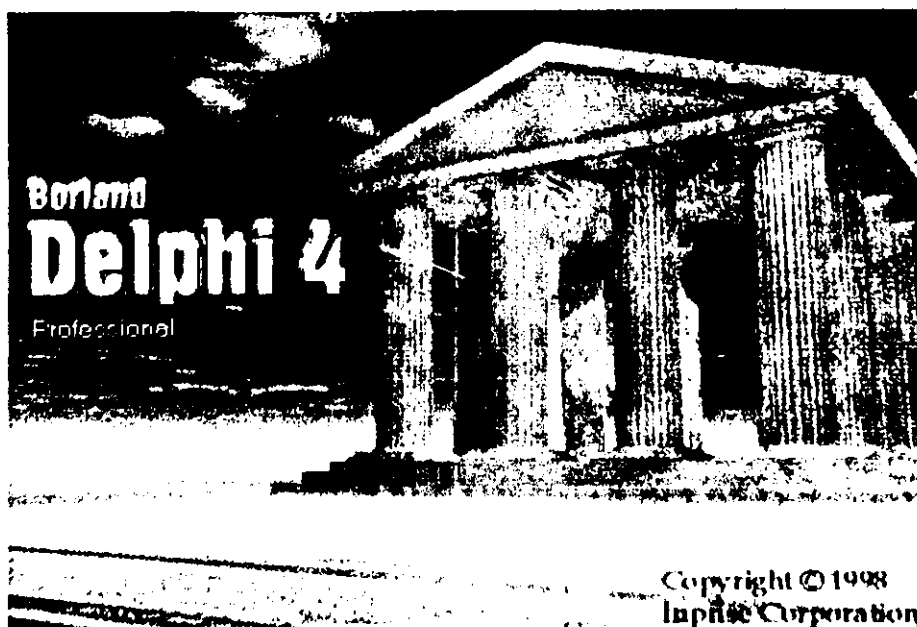


FIGURA 4.2 Pantalla principal Borland Delphi 4.0

Visual Basic fue el primer lenguaje que utilizó interfaces más visuales y que permitió que la programación pase de ser algo místico a algo manejable. Estas nuevas interfaces permiten al desarrollador construir la interfaz de usuario de una forma visual utilizando el mouse en lugar de texto.

Entre las principales características de Delphi podemos citar:

- Ofrece un buen diseño y construcción visual de la interfaz de usuario
- Está provisto de mecanismos que promueven un código altamente estructurado y compacto.
- Utiliza Object Pascal como lenguaje base
- Permite la utilización de herramientas OCX para 32 bits
- La arquitectura de las bases de datos y conectividad ofrecen un enfoque multinivel, que permite crear una aplicación cliente muy ligera y proporciona acceso nativo a las bases de datos, Access especialmente.

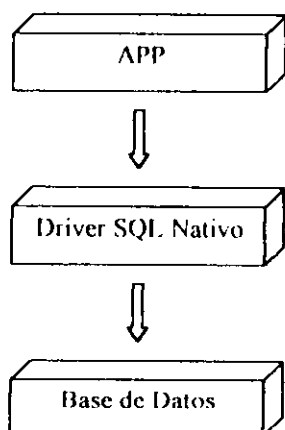


FIGURA 4.3 Acceso a Bases de Datos con Delphi

- Proporciona controles visuales ActiveX prefabricados o creados por el usuario
- Permite incluir bibliotecas de tiempo de ejecución en el ejecutable o trabajar con ellas como DLL separadas, lo cual permite incluir información en el archivo ejecutable para facilitar la distribución del producto.
- Las aplicaciones se pueden ejecutar como un control Activex
- El trabajo en Canvas y threads (hilos) en rutinas gráficas es seguro.
- Permite la reutilización de componentes.

ARQUITECTURA DE UNA APLICACIÓN DELPHI

Un programa sencillo puede ser considerado como si únicamente estuviera compuesto por un archivo de proyecto y una unidad, sin embargo existen otro tipo de ficheros que son creados por el lenguaje. A continuación se describen los componentes principales de una aplicación Delphi.

PROYECTO

Una proyecto Delphi es un conjunto formas, unidades, configuración de opciones, componentes y recursos. La información de estos elementos se almacena en archivos creados en forma automática por Delphi.

Un proyecto se crea con la extensión DPR y se puede considerar como la parte principal de la aplicación, contiene en resumen el código escrito en Object Pascal y es creado y modificado en forma automática y en muy pocas ocasiones el programador necesita realizar alguna modificación, excepto cuando desea incluir una rutina de programación avanzada o trabajar con librerías DLL.

FORMA

Una forma en Delphi es el equivalente de lo que en Windows se conoce como ventanas, las formas se utilizan para incluir controles, componentes, etc. Al igual que los archivos .dpr no requieren por lo general mantenimiento por parte del programador.

Es un archivo de tipo binario, se crea con la extensión DFM, almacena la información sobre el tamaño, forma, apariencia, ubicación, etc. Cada archivo de este tipo tiene asociado otro llamado unidad.

UNIDAD

Este tipo de archivos son utilizados para almacenar el código, tienen la extensión PAS. Existen tres tipos de unidades:

- Unidades asociadas con las formas
- Unidades de almacenamiento de procedimientos y funciones
- Unidades de construcción de componentes

La unidad más frecuentemente utilizada es la asociada con la forma, en primer lugar aparece el encabezado de la unidad, luego la sección de la interface y finalmente la sección de la implementación.

BIBLIOTECA DE COMPONENTES VISUALES

Se conoce como VCL, esta formado por un conjunto de objetos que serán utilizados como controles sobre las formas y que están escritos en Object Pascal, pueden ser:

- Botones
- Cuadros de edición
- Cuadros de listas desplegables
- Controladores de tiempo
- Cuadros de imagen
- Cuadros de selección, etc.

Se clasifican en dos grupos: visibles e invisibles. Los componentes visibles son colocados y programados por el usuario, mientras que los invisibles son utilizados como interfaz con el programador.

Cada uno de estos componentes tienen asociados un pequeño bloque de código que los hace funcionar y que evita que el programador tenga que darles mantenimiento, además cada uno de ellos tienen propiedades, eventos y métodos.

COMPONENTES OCX

Es un componente basado en OLE de 32 bits. Gracias a los objetos OLE, el usuario puede crear documentos complejos capaces de contener una serie de objetos de diversas fuentes.

PROCEDIMIENTOS CREADOS POR EL USUARIO

Delphi permite que el usuario incorpore a la aplicación procedimientos, funciones y manipuladores de eventos, los cuales permiten acceder a los componentes e indicarles qué deben hacer. Un procedimiento puede ser definido como un bloque de instrucciones o de código que puede ser activado mediante la ejecución de una llamada de procedimiento, Delphi ofrece de esta forma la posibilidad de reutilización del código.

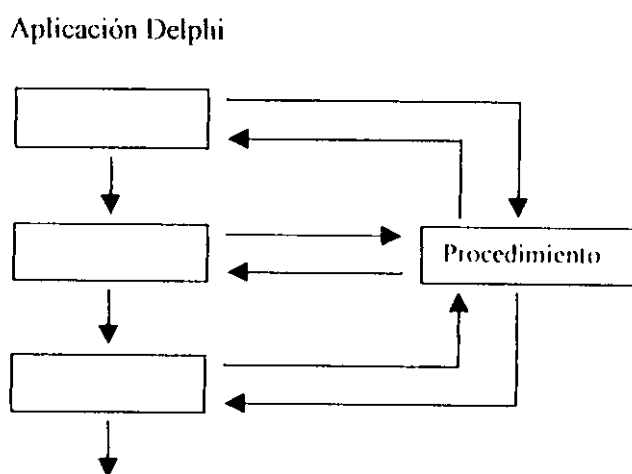


FIGURA 4.4 *Procedimientos en Delphi*

FUNCIONES

Una función se puede definir como un procedimiento que retorna un valor al programa desde el que fue invocado, además a diferencia de un procedimiento puede formar parte de una expresión.

EVENTOS

Actualmente la tendencia en lo que a programación respecta apunta a la utilización del modelo conducido por eventos, en el cual el compuesto de bloques de código se ejecutan cuando ocurre una acción o un suceso, por ejemplo:

- Clic del mouse
- Presión de teclas
- Ejecución de una aplicación
- Entrada de mouse
- Salida de mouse, etc

El modelo conducido por eventos es una forma efectiva de responder a las entradas e interacción del usuario en un ambiente gráfico.

AMBIENTE DELPHI

El entorno Delphi ha sido desarrollado de una manera muy funcional y organizada, se lo ha llamado IDE, que son las siglas de Ambiente de Desarrollo Integrado. Provee al usuario de todas las herramientas necesarias para diseñar una aplicación, consta de:

- Editor de códigos
- Barra de herramientas
- Depurador de errores
- Editor de imágenes

DELPHI Y MULTIMEDIA

Una de las características que hacen de este paquete una herramienta poderosa en el desarrollo de aplicaciones, es precisamente la posibilidad de explotar todo el poder de la multimedia en la construcción de sistemas audiovisuales, ofrece la posibilidad de integrar aplicaciones de audio y video o animaciones en tiempo real.

Delphi posee una llamada `APIPlaysound()`, que puede reproducir sonidos de diferentes fuentes, este sonido puede ser ejecutado en forma exclusiva o simultáneamente con otra tarea. Esta llamada se utiliza con sonidos de tipo WAVE, que es un archivo que almacena una representación digital del tono y el volumen sobre la longitud del sonido y no necesita gran cantidad de información para guardar el sonido.

Otro archivo muy utilizado en las aplicaciones multimedia es el fichero MIDI, realmente este archivo almacena las notas y el tiempo. Los ficheros de tipo MIDI pueden ser utilizados en una aplicación Delphi utilizando pocas instrucciones y sin mayor esfuerzo para el desarrollador.



FIGURA 4.5 *Componente Media Player - Delphi*

Delphi cuenta con un componente visual que encapsula la interface de comandos, que se denomina Media Player, éste puede utilizarse de dos formas diferentes: habilitado para que el usuario lo manipule a través de una serie de botones de avance, parada, etc, o escondiendo el componente para ejecutar archivos multimedia utilizando métodos desde un bloque de código. Este componente se utiliza con mayor frecuencia para reproducir archivos de tipo MIDI y AVI.

TECNICAS DE ANIMACION

Las aplicaciones multimedia de Delphi no podrían estar completas si no incluyen archivos de video, se puede incluir ficheros AVI utilizando el componente Media Player.

Cuando se utiliza animación o gráficos en movimiento, es muy importante que el movimiento sea lo más natural y fluido posible a los ojos del usuario final. La utilización de la técnica cuadro a cuadro, es decir dibujar una imagen, borrarla y poner otra, lamentablemente crea un efecto de parpadeo que es muy desagradable.

Delphi ofrece la posibilidad de utilizar un doble búffer, que es un conjunto de superficies de dibujo, se muestra una y la otra se utiliza como superficie de dibujo, cuando ésta se completa o se intercambian las superficies o se copia el búffer escondido se copia al que se muestra, esta técnica consiste en crear un mapa de bits en memoria y utilizarlo como búffer temporal.

DELPHI Y LAS BASES DE DATOS

Este software tiene un soporte poderoso para las bases de datos, su motor permite manipular y ver los datos en forma sencilla y rápida, liberando al programador de escribir rutinas para manejar la información, una de las ventajas más importantes del motor de bases de datos es que proporciona una capa de abstracción entre la base de datos subyacente y la funcionalidad de la base de datos.

Delphi incluye componentes visuales que posibilitan el acceso a las tablas y métodos para manipular sus registros, se pueden realizar rutinas complejas o muy simples, la interfaz de base de datos es un programa que facilita el acceso y la manipulación de los datos contenidos en una tabla, la herramienta Form Wizard permite crear rutinas completas sin que el usuario deba escribir el código, convierte los campos de la tabla en campos de entrada-salida de un formulario.

MODELOS DE BASES DE DATOS

Nuestro proyecto considera el trabajo en Delphi con bases de datos generadas en Access 97, aunque la tendencia actual apunta a la utilización de bases de datos relacionales, resulta muy necesario el analizar los distintos modelos que pueden ser utilizados en Delphi, éstos son:

- Bases de Datos Independientes
- Bases de Datos Compartidas
- Bases de Datos Cliente/Servidor

BASES INDEPENDIENTES

Este tipo de bases de datos por lo general tienen un sistema de archivo local y son las más fáciles de utilizar, no hay que preocuparse por la concurrencia, son utilizadas mayoritariamente para desarrollar aplicaciones que se distribuirán a muchos usuarios los cuales mantendrán las bases por separado y almacenarán sus datos individualmente.

BASES COMPARTIDAS

Su estructura es similar a la de una base de datos independiente, pero ésta será accesada por varios usuarios desde diferentes máquinas, no es recomendable utilizar este tipo de bases de datos en casos donde se requiera un elevado nivel de cómputo y accesos simultáneos.

BASES CLIENTE-SERVIDOR

Es la categoría más elevada de los modelos de bases de datos, en este caso una máquina que actúa como servidor, se encarga del acceso a la base de datos para dar servicio a un grupo de usuarios clientes. En este modelo el servidor debe realizar una tarea específica, ofrece muchas ventajas, pero una de sus principales desventajas es su costo y el hecho de necesitar un protocolo de comunicación.

Los manipuladores Paradox y dBASE están estrechamente ligados con Delphi y permiten crear y manipular en forma fácil las los datos desde el escritorio de bases de datos. El manejador de bases de datos también tiene la posibilidad de comunicarse con el ODBC, el cual a su vez puede comunicarse con la mayoría de las bases existentes en el mercado a través de una interfaz común utilizando un manejador que contiene el código que comprende aspectos específicos de una base de datos en particular y proporciona el acceso a ella mediante un conjunto estándar de llamadas API.

ATRIBUTO	PARADOX	DBASE
Máximo de Registros	Dos mil millones	Mil millones
Campos por tabla	255	1024
Caracteres por campo	No aplicable	256
Bytes por registro	32750	32767

FIGURA 4.6 Cuadro Comparativo Paradox - dBASE

4.2.2.2 VISUAL BASIC

Visual Basic es un lenguaje basado en POO, diseñado especialmente para crear aplicaciones de 32 bits, para los sistemas operativos de Microsoft de 32 bits, este tipo de aplicaciones aprovechan el espacio de direcciones largas y gestión de memoria, de forma que son rápidas y fáciles de utilizar en el entorno multitarea.

El lenguaje original fue el Basic , el cual muy rápidamente se convirtió en un lenguaje muy popular en todos los campos y en la enseñanza especialmente, fue adaptado por Bill Gates para su utilización en los computadores personales. A partir de esto se fueron desarrollando varias

versiones del lenguaje hasta llegar a implementarse el Visual Basic, cuya potencia y facilidad de uso son las razones principales por las que ha sido escogido como lenguaje de muchas aplicaciones de Windows, como por ejemplo Excel, Access, Project, etc.

Hace seis años, Microsoft Corporation lanzó una revolución en la manera de crear aplicaciones. Con el sistema de programación Visual Basic, los programadores sin experiencia pueden crear rápidamente aplicaciones para Windows. Actualmente millones de programadores están creando aplicaciones con una productividad sin precedentes y con una facilidad sin competencia con la tecnología Microsoft Visual Basic.

El lenguaje de programación Visual Basic no es exclusivo de Visual Basic. La edición para aplicaciones del sistema de programación de Visual Basic ha sido incluida en muchas aplicaciones Windows. El sistema de programación de Visual Basic, Scripting Edition (VBScript) para programar en Internet es un subconjunto del lenguaje de Visual Basic.

VENTAJAS

El proceso de desarrollo de aplicaciones tradicionales, se puede dividir en tres etapas: escritura, compilación y comprobación del código. A diferencia de los lenguajes tradicionales, Visual Basic utiliza una aproximación interactiva para el desarrollo, difuminando la distinción entre los tres pasos.

Visual Basic interpreta el código a medida que se escribe, además compila parcialmente el código mientras se escribe, por lo que cuando se ejecuta la aplicación la compilación tardará muy poco tiempo, por su naturaleza interactiva, el usuario se encontrará ejecutando la aplicación frecuentemente a medida que la desarrolle.

Todas las herramientas que necesita el usuario para aprender rápida y fácilmente el sistema de programación Visual Basic las proporciona su completo entorno de programación. Las características AutoComplete, Statement Builder y QuickTips, evita que el usuario tenga que memorizar comandos o el lenguaje.

Visual Basic 5.0 también ayuda a realizar una transición fácil a las fronteras de las nuevas tecnologías, como Internet, sin tener que abandonar el código existente o la técnica de programación.



FIGURA 4.7 Portada Visual Basic 5.0

Entre las múltiples ventajas de trabajar con Visual Basic y sus aplicaciones de 32 bits, podemos citar las siguientes:

- Manejo de múltiples proyectos en una misma instancia de Visual Basic. Útil para la depuración de controles ActiveX.

- Posibilidad de crear aplicaciones con interfaces de documento simple, múltiple o estilo Microsoft Explorer.
- Los objetos proporcionados por sus componentes pueden provocar eventos que se pueden utilizar en otras aplicaciones. Puede manipular eventos provocados por otras aplicaciones, o por su propio objeto, usando variables WithEvents.
- Las características de acceso a datos permiten crear bases de datos y aplicaciones cliente para los formatos más conocidos incluyen SQL Server.
- Con la edición de aprendizaje, puede definir constantes con nombres relacionados y agruparlos en un enumerador, o Enum.
- La mayoría de los controles permiten operaciones de arrastrar y colocar entre aplicaciones OLE, por ejemplo, arrastrar el contenido de un documento Word en un control TextBox.
- La característica Implements permite a sus clases compatibilidad con múltiples interfaces. Los usuarios de sus componentes estarán pronto ligados a estas interfaces, sin tener en cuenta los objetos que ellos implementen.
- Genera y manipula módulos de clases, propiedades, eventos y métodos. Puede definir visualmente interfaces de controles ActiveX, diseñar propiedades de página, y expulsar miembros de un control subclase.
- Permite conseguir un nuevo grado de productividad creando subclases y agregando controles ActiveX. Permite extender las clases existentes, propiedades, eventos y métodos creando interfaces de herencia desde otros módulos de clases.
- Acceso al espacio completo de los microprocesadores, puede utilizar hasta dos gigabytes de memoria de sistema.
- Acceso a Win32 API, con lo cual se podrán añadir características avanzadas a los programas.
- Mayor protección ante caídas del sistema provocadas por otros programas.
- Cada programa Basic se ejecuta en su propia región de memoria.
- Ayuda interactiva en la creación de sus proyectos, en ventanas, cuadros de herramienta, objetos de formulario, mensajes de error, etc.

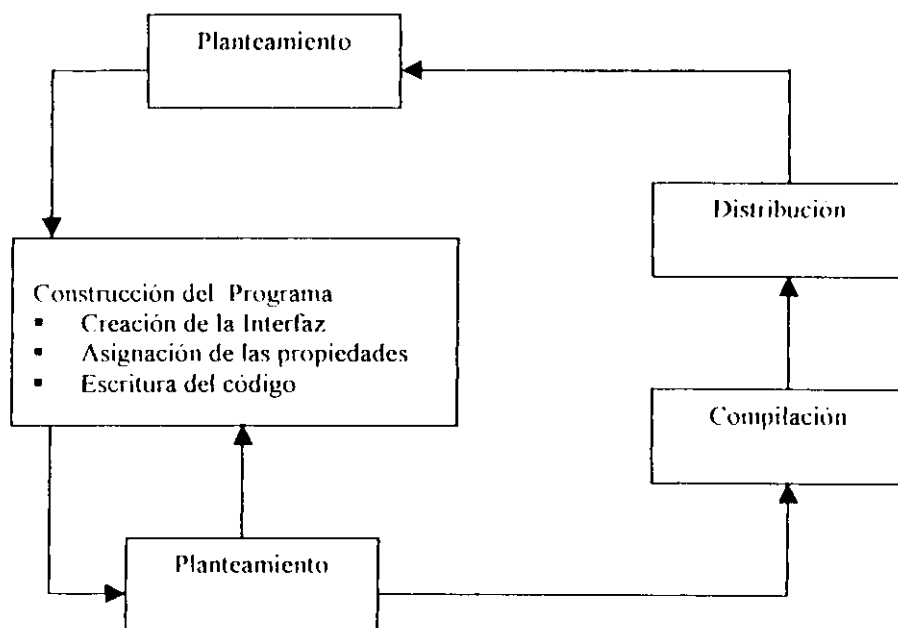


FIGURA 4.8 Proceso de desarrollo de software en Visual Basic

ARQUITECTURA DE UN PROYECTO DE VISUAL BASIC

Los elementos que componen un proyecto en Visual Basic se listan a continuación:

MODULOS DE FORMULARIO

Estos módulos pueden contener descripciones en forma de texto del formulario (.frm) y sus controles, incluyendo valores de sus propiedades, pueden además contener declaraciones a nivel de formulario de constantes, variables, procedimiento de eventos y procedimientos generales.

MODULOS DE CLASE

Los módulos de clase (.cls) son muy parecidos a los módulos de formulario, con la diferencia de que no tienen interface visible con el usuario, pueden ser utilizados para que éste cree sus propios objetos, incluyendo código para sus métodos y propiedades.

MODULOS ESTANDAR

Los módulos estándar (.bas) pueden contener declaraciones públicas a nivel de módulo de tipos, constantes, variables, procedimientos internos y procedimientos públicos.

ARCHIVOS DE RECURSOS

Los archivos de recursos (.res) contienen mapas de bits, cadenas de texto y otros datos que puede modificar sin volver a modificar el código.

DOCUMENTOS ACTIVEX

Tienen extensión (.dob) son similares a los de formularios, pero muestran un explorador de Internet.

ENTORNO DE DESARROLLO

Se lo conoce como entorno de desarrollo integrado IDE, ya que integra funciones diferentes como el desarrollo, modificación, compilación y depuración. Este entorno consta de los siguientes elementos.

BARRAS DE MENUS

Visualiza los comandos que se utilizan para trabajar con el lenguaje. Estos menús son: Archivo, Edición, Ver, Ventana, Ayuda, proyecto, Formato y Depuración.

BARRAS DE HERRAMIENTAS

Proporcionan un acceso rápido a los comandos usados normalmente en el entorno de programación.



FIGURA 4.11 Barra de herramientas Visual Basic

CUADRO DE HERRAMIENTAS

Contiene un conjunto de herramientas que se pueden usar durante el diseño para colocar controles en un formulario, además del diseño del cuadro de herramientas, el usuario puede crear su propio diseño personalizado.

EXPLORADOR DE PROYECTOS

Enumera los formularios y módulos del proyecto actual, el cual no es más que la colección de archivos que usa para generar una aplicación.

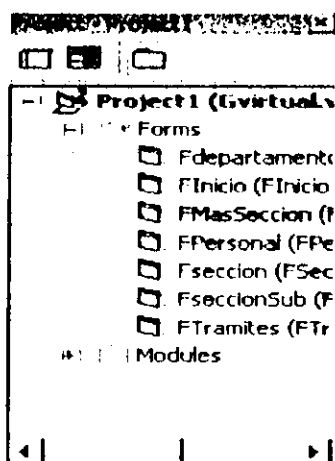


FIGURA 4.12 Project Explorer Visual Basic

VENTANA DE PROPIEDADES

Enumera los valores de las propiedades del control o formulario seleccionado, como por ejemplo color, tamaño, etc.

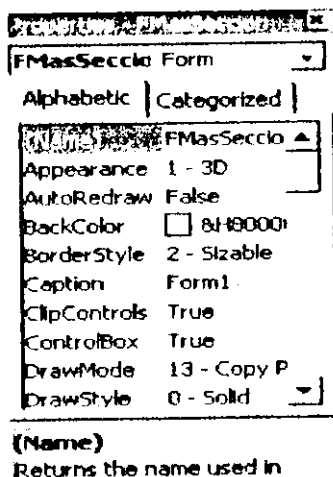


FIGURA 4.13 Ventana de Propiedades Visual Basic

EXAMINADOR DE OBJETOS

Enumera los objetos disponibles que se pueden utilizar en el proyecto y proporciona una manera rápida de desplazarse en el código

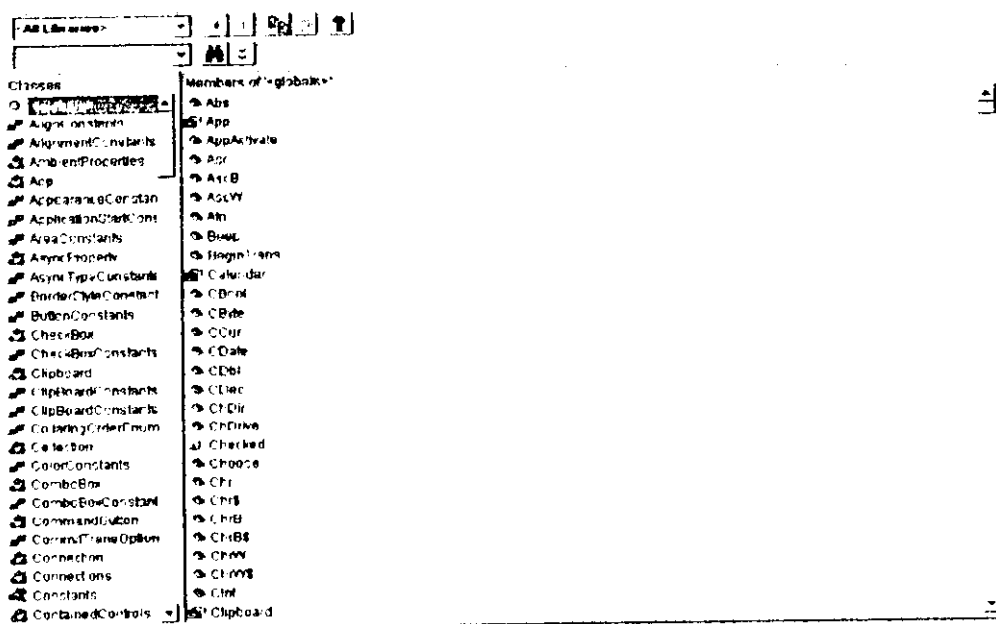


FIGURA 4.14 *Examinador Objetos Visual Basic*

DISEÑADOR DE FORMULARIOS

Funciona como una ventana en la que se personaliza el diseño de la interfaz de la aplicación, agregando controles gráficos, imágenes, etc. Cada ventana de la aplicación tiene su propio diseñador de formulario.

EDITOR DE CODIGO

Es una ventana desde la cual se puede escribir el código de la aplicación, por cada formulario o módulo de la aplicación, se crea una ventana de edición de código.

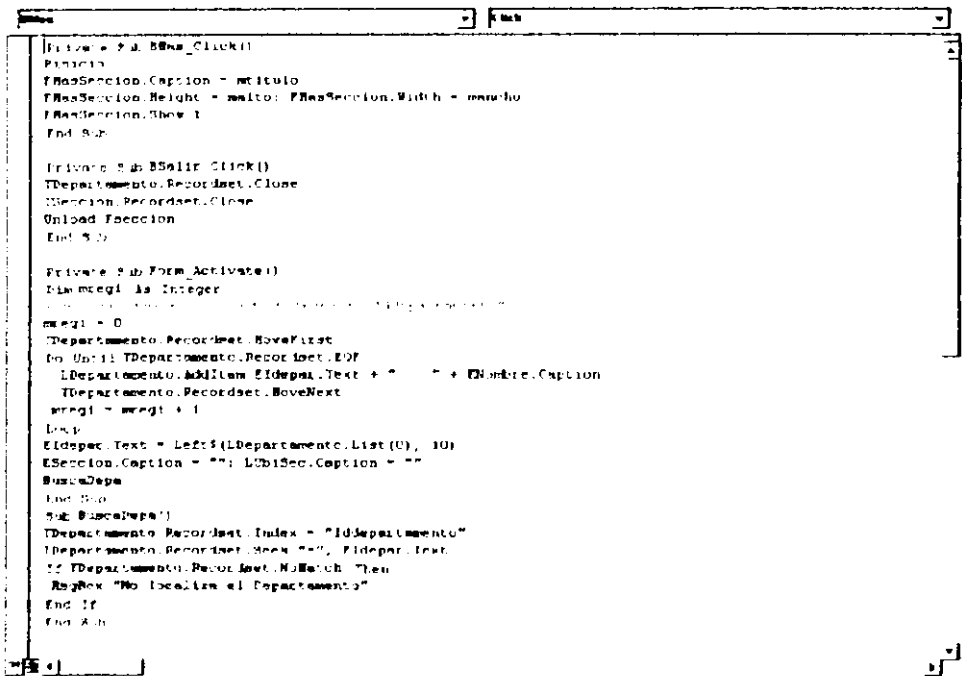


FIGURA 4.15 Editor de Código Visual Basic

MECANICA DE LA ESCRITURA DEL CODIGO

El código en Visual Basic se almacena en módulos, existen tres tipos de módulos: formulario, estándar y de clase.

Cada módulo se divide en secciones, una por cada objeto del módulo. Por cada módulo se abre una ventana del editor de código, cada sección de código puede contener varios procedimientos distintos para eventos del formulario o control. La lista de procedimientos de la sección general contiene una única selección, la sección de declaraciones donde se declaran las variables, constantes y D.L.L.

Visual Basic hace que la escritura del código sea mucho más sencilla mediante características que llenan automáticamente instrucciones, propiedades y argumentos, a medida que el usuario va introduciendo código el editor muestra listas de elecciones adecuadas, prototipos de instrucciones o funciones o valores.

LA INTERFACE DE USUARIO EN VISUAL BASIC

La interface de usuario es quizás la parte más importante de una aplicación o por lo menos es la más visible, para el usuario final la interfaz es la aplicación y con seguridad a ellos no les interesa el código que se ejecuta detrás. Independientemente del esfuerzo y el tiempo que se haya empleado en la escritura y optimización del código la facilidad de uso de la aplicación depende de la interface creada.

Antes de empezar a diseñar la interface de usuario se debe pensar en el propósito de la aplicación, el diseño de una aplicación principal de uso constante debe ser distinto al de una que se utilizará en forma ocasional. La mejor técnica de diseño de una interface de usuario es la del proceso interactivo.

El conjunto completo de pantallas e imágenes utilizadas en el programa se conoce como interface de usuario del programa. Esta interface incluye los menús, cuadros de diálogo, botones, objetos, objetos y dibujos que los usuario verán mientras utilizan el programa.

Visual Basic permite crear todos los componentes de una aplicación profesional basada en Windows de forma rápida y eficiente. Para aumentar la posibilidad de uso de una aplicación desarrollada en Visual Basic es conveniente utilizar los elementos de la interface de usuario de Windows de forma estándar y predecible.

VISUAL BASIC Y LAS BASES DE DATOS

Una de las ventajas de Visual Basic, es que permite utilizar el control de datos para crear aplicaciones que presentan, modifican y actualizan información de bases de datos de Microsoft Access, Btrieve, Dbase, Foxpro y Paradox.

También VB permite un acceso directo a Excel, Lotus y archivos de texto ASCII estándar como si fueran formas auténticas de bases de datos. Además permite acceso a bases de datos remotas de ODBC. El control de datos implementa el acceso a datos mediante un motor de Base de Datos de Microsoft Jet, el mismo que hace funcionar a Microsoft Access, esta tecnología permite un acceso transparente a muchos formatos estándar de base de datos y permite crear aplicaciones sin tener que escribir el código.

El control de datos puede llevar a cabo las siguientes tareas sin tener que programarlas

- Conectar con una base de datos local o remota
- Abrir una base de datos específica con un conjunto de registros, mediante una consulta SQL.
- Pasar campos de datos a controles enlazados
- Agregar registros o actualizar una base de datos
- Interceptar los errores que se produzcan cuando se realice el acceso a los datos
- Cerrar la base de datos

El control de datos proporciona una interface relacional con los archivos de las bases de datos, una base de datos relacional no es más que una tabla compuesta por filas y columnas, VB llama a las columnas campos y a las filas registros. El motor Microsoft Jet, hace funcionar a las bases de datos como un conjunto de tablas relacionales cualquiera que sea su formato físico de archivo. Los principales términos que se manejan a nivel de base de datos en VB, son los mismos que se utilizan con una base de datos externa: tabla, campo, registro, índice, SQL, etc.

CONTROLES ENLAZADOS PARA BASES DE DATOS

Los controles enlazados a datos son los controles de datos a través de los cuales se tiene acceso a la información de la base de datos, cuando un control está enlazado a un control de datos, VB aplica a ese control valores de campo del registro actual, a su vez el control muestra los datos y acepta los cambios, los cuales se pueden escribir en la base de datos en forma automática.

Visual Basic admite varios controles integrados que se pueden enlazar al control de datos, la mayoría de ellos se caracterizan por las propiedades: DataChanged, DataField y DataSource. Los controles integrados además se pueden enlazar a controles ActiveX (OLE).

4.2.2.3 ANALISIS COMPARATIVO.- VISUAL BASIC 5.0 vs DELPHI 4.0

Una vez estudiados los componentes principales de cada uno de estos paquetes, su estructura, ventajas y limitaciones, se realizó una prueba de compatibilidad tanto de Delphi como de Visual Basic con Microsoft Access, luego de la cual llegamos a conclusiones interesantes, que nos llevaron a elegir a Visual Basic como el lenguaje a utilizar para desarrollar la segunda fase del administrador, la construcción de la plantilla generadora del Guía Virtual.

- La interface entre Delphi y Access, no está construida adecuadamente, ya que los datos de las tablas de Access fueron recuperados en Delphi con errores en los campos de tipo OLE. Para

solucionar este problema se realizaron varias pruebas grabando los campos OLE con varias extensiones desde Access para recuperarlos en Delphi, sin lograr obtener un resultado positivo.

- La interface entre Visual Basic 5.0 y Access en lo que a recuperación de datos se refiere funcionó correctamente, los campos de las tablas de todos los tipos fueron recuperados en Visual Basic en perfecto estado.
- La velocidad de compilación en tiempo de diseño en Delphi 4.0 fue mayor con respecto a Visual Basic, el primero resultó demasiado lento.
- Visual Basic permite el trabajo y mantenimiento de módulos de programación, lo que contribuye enormemente a la facilidad del trabajo del programador.
- El entorno de programación de VB es más amigable y ofrece mayor comodidad
- Visual Basic contiene un conjunto de propiedades más grande para los objetos.
- Visual Basic incluye nuevos y variados componentes VBX y OCX, como por ejemplo IELLabel.

CAPÍTULO V

EL GUIA VIRTUAL

5.1 ESTRUCTURA

En el cuarto capítulo se habló en forma muy general de la estructura que tendrá la Plantilla Generadora de Guías Virtuales. El objetivo básico del paquete consiste en que el usuario ingrese en forma rápida y sencilla la información que él considere apropiada para ser mostrada al público que desee obtener información sobre la empresa o Institución, y luego de ello dicha información sea leída desde una plantilla (sin que esto deba ser procesado por el usuario) incorporándola en forma automática a un conjunto de pantallas diseñadas especialmente para el efecto, considerando e incluyendo elementos multimedia que le darán un toque de interactividad. Estos procesos están divididos en dos niveles: Administrador y Usuario

5.1.1 NIVEL ADMINISTRADOR

El administrador del Guía Virtual, desde este nivel podrá dar mantenimiento a la información de más importante de la Institución, tal como datos acerca de su historia, personal, departamentos, secciones y trámites, logrando de esta forma descongestionar el trabajo del personal encargado de la atención al público y a la vez proveyendo a sus clientes de una herramienta novedosa y útil a través de la cual conocerá a la Institución, y obtendrá información sin que ello implique largas colas o tratar con empleados que normalmente no están suficiente capacitados en una buena atención al público y lo que es más importante lo hará por sí solo. El administrador dará mantenimiento a la información, la misma que puede contener texto, imágenes, sonido y video, a través de las siguientes operaciones:

- Ingreso de datos
- Modificación
- Borrado
- Consulta

Estas operaciones permitirán que el Guía de la Institución no incluya la misma información siempre, sino que proporciona al usuario de ésta la posibilidad de actualizarla de acuerdo a las necesidades de la Empresa o Institución, es importante resaltar el hecho de que el usuario no requerirá en ningún momento de conocimientos técnicos para realizar este tipo de operaciones.

También en este nivel el administrador tendrá acceso a información estadística, generada en forma automática a través de una interface programada entre Access y Visual Basic, gracias a la cual podrá conocer el número de visitas realizadas al Guía Virtual, la misma que será presentada en descriptivos gráficos estadísticos.

5.1.2 NIVEL USUARIO

Una vez introducida la información, el Guía Virtual está listo, pues ni siquiera el administrador necesita realizar operaciones previas para que ésta pueda ser presentada al usuario final. Lo único que deberá hacer el usuario final del Guía es presionar botones, leer la información, escucharla, verla y sobre todo disfrutar el ambiente para él diseñado a través de la construcción de una plantilla.

Se han considerado varios aspectos técnicos para que el Guía Virtual consiga sus objetivos, llegar de la mejor manera posible con la información de la Empresa o Institución, entre estos tenemos:

- Interactividad

- Forma de Navegación.
- Areas Sensibles y botones

5.2 INTERACTIVIDAD

El multimedia interactivo abre nuevas posibilidades de creación audiovisual. La última oferta brindada por la informática a lo audiovisual es el multimedia interactivo, que permite presentar textos, sonidos e imágenes bajo la acción selectiva del usuario. El multimedia interactivo no es una tecnología para colaborar en una aplicación, como por ejemplo escribir un texto, editar un video, o componer una pieza musical. Tampoco es una tecnología en busca de aplicación, como pasa a veces con ciertos inventos. Efectivamente, en el ámbito educativo parece probado que el aprendizaje interactivo da mejores resultados que el pasivo. Stephen Wilson, experto en interactividad, dice al respecto: "Los interactivos multimedia permiten a las personas seguir los propios caminos asociacionistas; experimentar y construir sus propias estructuras cognitivas, y enlazar sus acciones con las necesidades emocionales y de identidad. En aplicaciones de aprendizaje y de información, las teorías sugieren que los materiales son recordados, utilizados e integrados mejor", "Extrapoladas estas teorías a los terrenos artístico o de entretenimiento, los espectáculos interactivos pueden ser más profundos y emocionantes que las experiencias no interactivas".

El multimedia interactivo es un conjunto de herramientas para diseñar, ensamblar y proporcionar aplicaciones, es decir, programas. Este conjunto de herramientas contribuye a facilitar una relación "hombre-máquina" cada vez más fluida y rica en medios. La evolución hace posible que esta relación se de en los dos sentidos, en consecuencia la relación "hombre-máquina-hombre", o sea, "hombre-hombre", puede también devenir más fluida y rica en medios. La interactividad es, pues, aplicable a la comunicación entre las personas, en un abanico de contenidos que va desde la

entrega de instrucciones hasta la expresión artística, pasando por el entretenimiento.

5.2.1 ANTECEDENTES DE LA INTERACCION

La cuestión de la máquina y en particular de la máquina en medio de las personas, no ha gustado nunca a los artistas. Antigüamente se creía que autor y máquina eran incompatibles. Ha habido obras expresamente abiertas a la participación del espectador, como las del teatro que serían precedentes de las obras interactivas por lo que se refiere a la voluntad de desafiar la pasividad y el estatismo del espectador. También lo serían las obras de arte llamado participativo (performances, happenings, instalaciones) de los años sesenta y setenta.

Pero los antecedentes más directos de los programas interactivos se encuentran en el mundo de la informática, desde donde se ha incubado la tecnología que los hace posibles. En este mundo, el conjunto de formas y gestos con los que se establece el diálogo entre una persona y un programa constituye la llamada "interface".

La interface entre hombre y máquina en los años cuarenta era solamente un concepto: el diálogo con el ordenador se hacía a base de complicadas operaciones de cambio de placas y circuitos electrónicos. En los años sesenta, la comunicación se hacía con cintas y fichas de papel perforado, una interface más práctica pero igualmente inhumana, porque el idioma del diálogo era el idioma del computador.

La interface más cercana a la persona que a la máquina comienza en el momento en que se hace posible digitalizar las imágenes. Como consecuencia aparecieron las primeras interfaces gráficas, en los años sesenta, a las que se debe la aceptación posterior del ordenador personal en los años

ochenta. Actualmente la interface es un aspecto clave de los programas informáticos; el éxito de un programa depende directamente de la calidad de su interface, hasta el punto de que cuando una persona dice no entender un programa, o bien no entender el computador, lo que está diciendo en la mayoría de los casos es que no entiende la interface. Por eso el diseño de interfaces moviliza a numerosos informáticos, psicólogos, diseñadores y expertos bajo la especialidad llamada "Interface Hombre-Máquina".

Las ideas seminales de los programas interactivos y su alcance se deben fundamentalmente a dos personajes. De un lado, Vannevar Bush, que en los años cuarenta, cuando los computadores solamente servían para calcular trayectorias de balas y bombas, imaginó un sistema digital, llamado "Memex", para almacenar grandes bases de conocimientos interconectados a las que acceder desde diversos y múltiples niveles. Por otro lado, Ivan Sutherland, en los años sesenta, inventó la digitalización de las imágenes, e imaginó unas gafas para pasearse dentro de las imágenes. Las ideas de Bush están en la base de los sistemas modernos de intercambio de información, más cercanas a la manera natural de relacionar la información de las personas, mientras que Sutherland es el padre de la tecnología que permite llevar a la práctica estas ideas y el primero en desvelar los horizontes de la comunicación interactiva.

Otro personaje de esta historia es Ted Nelson, creador en 1965 del término "hipertexto". Nelson ideó también otro utópico banco de datos, de nombre "Xanadú", a pesar de que "Xanadú" no ha acabado nunca de existir, las reflexiones paralelas de Nelson en el orden metodológico y lingüístico son esenciales en la evolución de los programas interactivos. La solución para una interactividad técnicamente factible, por fin, viene dado por las tecnologías inventadas por Douglas Engelbart (el ratón, las ventanas...) y por Alan Kay (las primeras interfaces gráficas), a principios de los setenta.

En el ámbito audiovisual, el precedente directo del programa interactivo es el video interactivo. Lo que hace pasar del video interactivo al multimedia interactivo son las tecnologías de la compresión y de la transmisión de las imágenes, una con el objetivo de reducir el volumen de la información que implican las imágenes en movimiento, y la otra para aumentar la velocidad de transferencia de información entre los dispositivos. Por cierto que en el proceso de conversión digital, textos, imágenes y sonidos acaban todos convertidos en bits de información.

5.2.2 CARACTERISTICAS DEL SOFTWARE INTERACTIVO

El multimedia interactivo propone una nueva lectura de las cosas. Además de la diversidad de soportes de la información, la relectura interactiva de un tema ha de tener sentido en sí misma para la interactividad. En definitiva, el medio ha de adaptarse al contenido y no al revés.

Debe permitir una navegación interesante tanto parcialmente como por entero, si el usuario se aburre, si no ve estimulada su interacción, no hay programa. Además de suministrar la información, un software interactivo debe ofrecer entretenimiento que procure satisfacer su interés, es decir, que los contenidos, además de permitir diversidad de enfoques, de puntos de mira y de vertientes en las que profundizar, debe disponer de recursos audiovisuales abundantes y atractivos.

Una audiencia infantil agradece más la interacción que una audiencia adulta, mientras que ésta última prefiere en general contenidos con hilo argumental. Un contenido debe dejar tiempo para la contemplación y no forzar la sobreactividad del espectador.

Una manera de distinguir los interactivos es según la capacidad de "control" que dan a la persona, el grado de "autonomía" que le permiten para decidir qué hacer, por dónde navegar, etcétera. Cuando el control posible es bajo, los caminos a seleccionar son restringidos y prefijados, como en la enseñanza programada. En otros, la persona tiene un cierto control pero sólo localizado en áreas determinadas del espacio de información.

5.3 FORMA DE NAVEGACION

El mapa de navegación de la estructura del proyecto es uno de los elementos más importantes, ya que bosqueja las conexiones o vínculos de las diferentes áreas de su contenido y le ayuda a organizar su contenido y mensajes, además proporciona una gráfica del flujo lógico de la interface interactiva, describe los objetos multimedia y muestra qué sucede cuando interactúa el usuario. Existen cuatro estructuras de organización fundamental utilizadas en los proyectos multimedia, a menudo en combinación.

5.3.1 LINEAL

En donde el usuario navega secuencialmente, de un cuadro o fragmento de la información a otro.

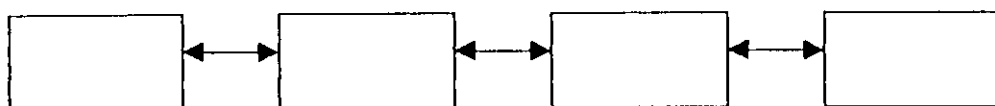


FIGURA 5.1 Navegación Lineal

5.3.2 JERARQUICA

El usuario navega a través de las ramas de la estructura de un árbol que se forma dada la lógica natural del contenido.

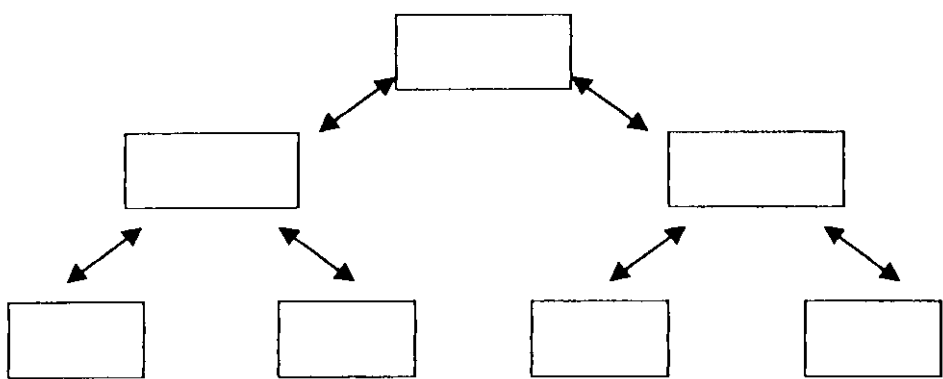


FIGURA 5.2 Navegación Jerárquica

5.3.3 NO LINEAL

El usuario navega libremente a través del contenido del proyecto sin limitarse a vías predeterminadas

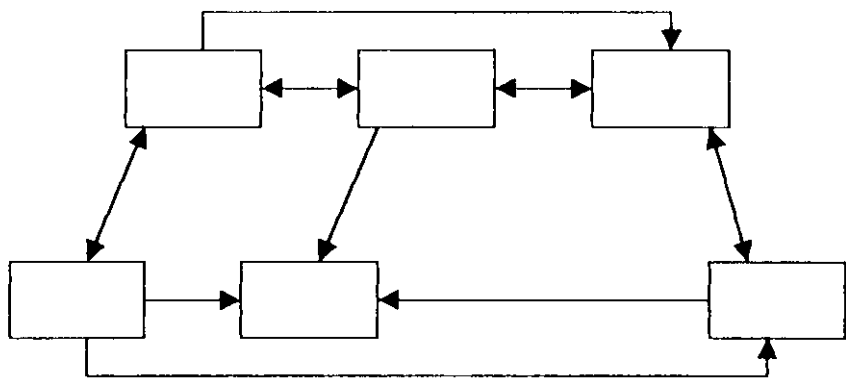


FIGURA 5.3 Navegación No Lineal

5.3.4 COMPUESTA

Los usuarios pueden navegar libremente (no linealmente) pero están limitados en ocasiones por presentaciones lineales de datos que se organizan con más lógica en una forma jerárquica.

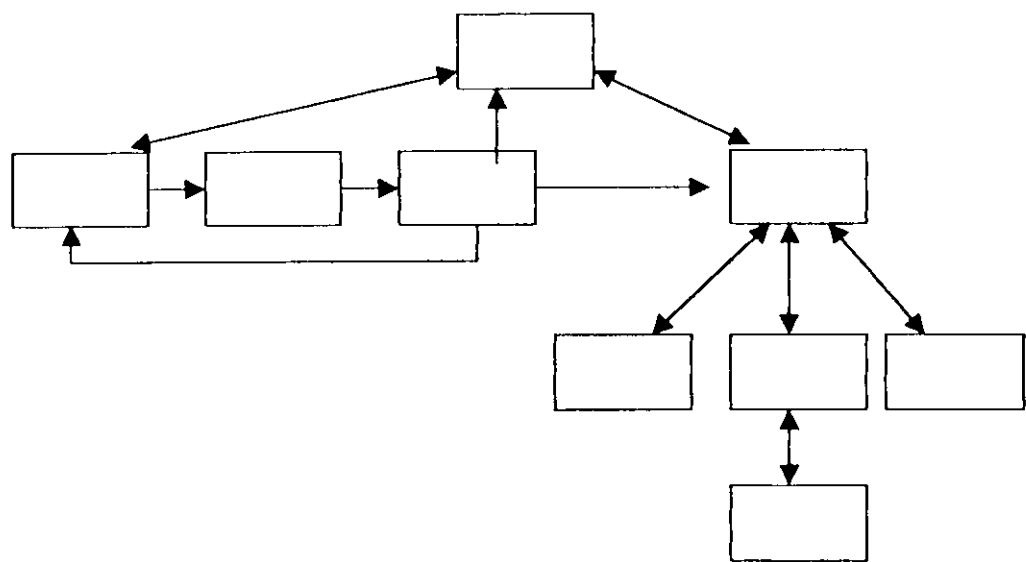


FIGURA 5.4 Navegación Compuesta

Para el proyecto de la plantilla generadora de Guías Virtuales, hemos escogido la navegación compuesta, considerando que el método de navegación es parte de la interface de usuario y que el éxito del proyecto no solamente depende del diseño general y del arte gráfico.

Es importante dar al usuario la noción de que tienen la libertad de escoger, ya que esto les da poder dentro del contexto de la materia que trata, pero es importante mantener un equilibrio ya que demasiada libertad puede llegar a desconcertarlo y hasta a perderlo. Es importante mantener los mensajes y contenidos organizados a través de un flujo constante entre los temas principales dejando que los usuarios hagan bifurcaciones para explorar los detalles.

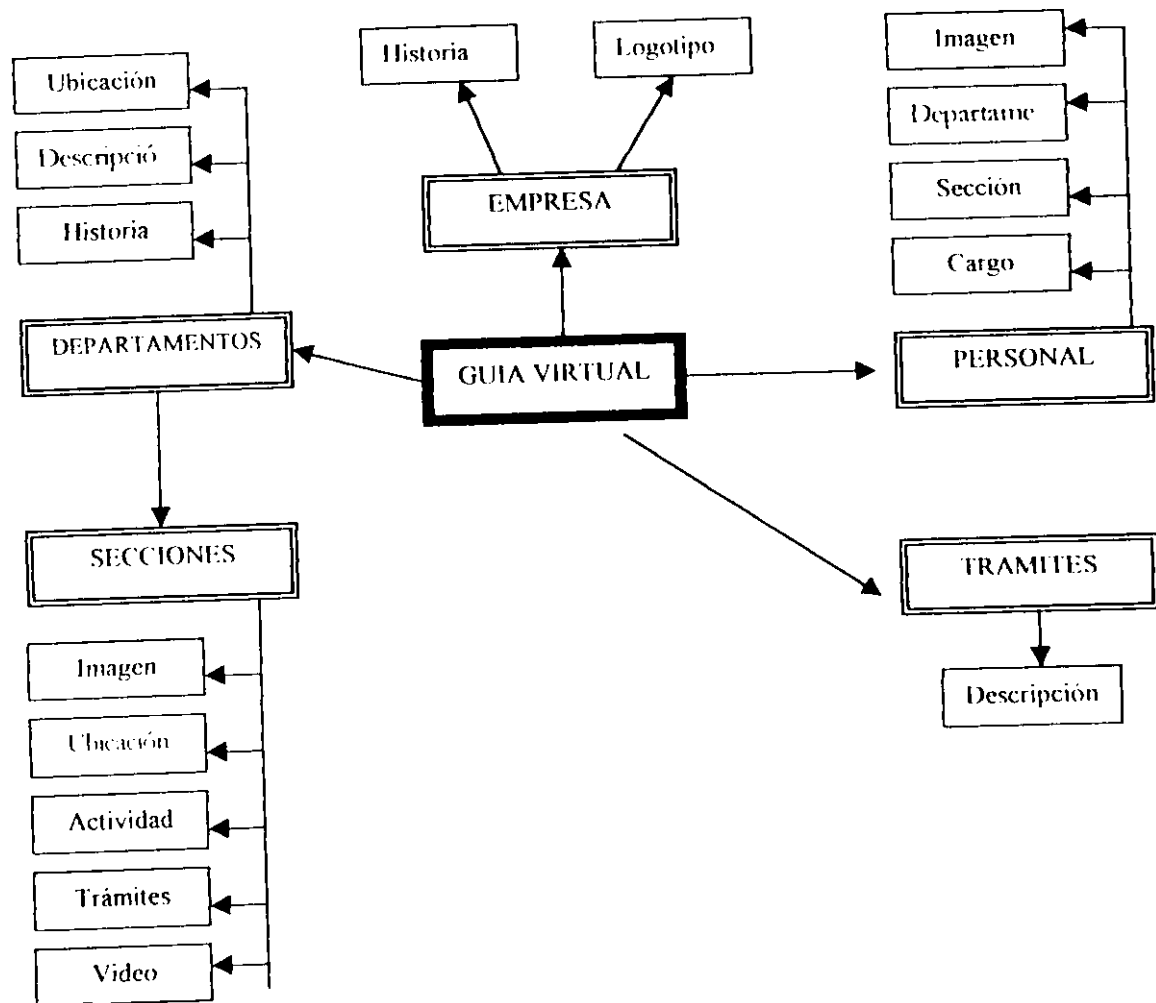


FIGURA 5.5 Mapa de Navegación Guía Virtual

5.4 AREAS SENSIBLES Y BOTONES

Una característica casi general de los sistemas multimedia es la de permitir que parte de la pantalla o cualquier objeto se convierta en un botón o hot spot, en las cuales mediante un clic algo sucede lo que hace que el programa sea interactivo y por ende emocionante.

Existen tres tipos de botones: los de texto, los gráficos y los de tipo icono. En la plantilla se han incluido los botones gráficos y los botones de iconos

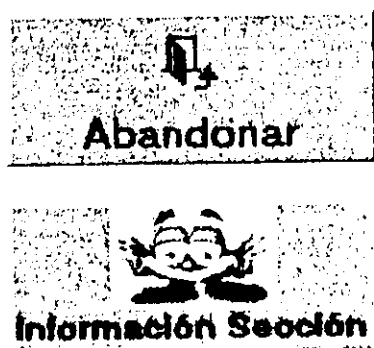


FIGURA 5.6 Botones utilizados en la Aplicación

5.5 ADQUISICION DEL CONTENIDO

El contenido es la información y el material que son la esencia del proyecto y por los mensajes que se utilicen, en la práctica el contenido puede ser cualquiera o todos los elementos multimedia y puede tener un alto o bajo contenido de producción, siempre se debe equilibrar el valor de la producción con el presupuesto y los resultados deseados.

“ Existe un nivel de calidad mínimo aceptable que satisface al público, incluso cuando ese nivel no sea el mejor que puede obtener con la tecnología, tiempo y esfuerzo”⁽⁸⁾

Para estructurar la calidad de la información que será solicitada al administrador del Guía, se realizó una investigación de la estructura de varias empresas e Instituciones, con la finalidad de obtener un modelo estándar que funcione para todas. La información base quedó formada por: empresa, departamentos, secciones, personal y trámites.

⁸ Todo el Poder de la Multimedia. Pg. 344

Para determinar con bases reales el contenido de nuestro proyecto se procedió a elaborar encuestas que fueron corridas en la Institución de la cual se personalizará el Guía Virtual, es decir de la Pontificia Universidad del Ecuador Sede Ambato, y en base a los resultados de dichas encuestas se conformó la estructura que tendrá el Guía y también se obtuvo la información que sería alimentada en el nivel administrador.

5.6 CONSULTAS VIRTUALES

El objetivo central del proyecto se basa en la posibilidad de utilizar el computador como lo que hemos denominado un “Guía Virtual”, gracias al cual el cliente podrá obtener información por si solo acerca de la Institución a través de “Consultas Virtuales”.

La consulta virtual será el proceso generado por la solicitud del usuario del paquete, es decir por un clic sobre un botón, icono o imagen, en respuesta la computadora le mostrará la información por él requerida en forma de texto, video o sonido, con ello se logrará que el usuario sienta que está siendo atendido por un guía computarizado.

5.7 PROYECCIONES ESTADISTICAS DE VISITAS

La estadística se constituye fundamentalmente en un método para examinar los procesos sociales y naturales a la vez que suministra bases a decisiones en la investigación y la administración, en detalle las técnicas estadísticas son numerosas y variadas pero en suma constituyen un enfoque unificado, sistemático y lógico para el estudio de las cuestiones humanas y del orden de la naturaleza, en sus aplicaciones cotidianas estas técnicas proporcionan recopilaciones descriptivas sucintas de masas de observaciones.

En su aspecto más significativo las técnicas estadísticas modernas son procedimientos para obtener lo que se ha denominado “aseveraciones fiables”, dichas aseveraciones cuando se basan en la estadística pueden ser estimaciones o generalizaciones que exceden a la muestra de observaciones estudiada.

En los trabajos estadísticos el investigador debe aprovechar principios orientadores y conceptos racionales ya que corre dos riesgos: sobrevalorar la acumulación de datos o que se interese excesivamente en las técnicas relativas a su manipulación.

La eficiencia del sistema se podrá medir en base al número de visitas virtuales que los usuarios hayan realizado a la empresa a través de la aplicación, por esta razón se ha incorporado un módulo estadístico en el nivel administrador desde el cual éste podrá obtener reportes estadísticos gráficos de las visitas realizadas, las cuales son registradas en forma automática por la aplicación las mismas que son registradas a través de una interface programada entre Visual Basic y Access.

Otro elemento importante del módulo estadístico de la aplicación es la posibilidad de obtener una proyección de visitas, la misma que está calculada en base a los datos de visitas registradas por la aplicación.

5.7.1 EL METODO ESTADISTICO

Para calcular la proyección de las visitas que se realizarán al Guía Virtual de la Institución se ha escogido el método más sencillo y eficaz utilizado en estos casos que es el empleo de una media geométrica la cual será aplicada a los valores recopilados durante varios meses de visitas.

La fórmula más comúnmente utilizada para el cálculo de una proyección es:

$$P_n = P_0 (1 + r)^n$$

En donde:

P_n	Proyección futura
P_0	Visitas existentes
r	Incremento probable
n	Tiempo considerado para la proyección

El valor del incremento será calculado de acuerdo a la siguiente fórmula:

$$r = \sqrt[n]{(P_2/P_1)} - 1$$

En donde:

P_2	Dato más reciente de visitas (último)
P_1	Dato menos reciente de visitas (primero)
n	Tiempo transcurrido

5.8 REQUERIMIENTOS Y ESPECIFICACIONES

Los requerimientos mínimos de hardware necesarios para la instalación y funcionamiento del Guía Virtual, son los siguientes:

- Computador 486 o superior
- Disco Duro de 1.2 GB
- Velocidad 200 Mhz o más

- Monitor Color SVGA -- resolución 1024 x 768
- 8 MB RAM
- CD- Rom 16x o más
- Kit multimedia
- Tarjeta de video
- Mouse
- Teclado

En cuanto a los requerimientos de software tenemos lo siguiente:

- Windows 95 o superior
- Access 97
- Software para capturar video
- Software para tratamiento de imágenes
- Software para tratamiento de sonidos

5.9 PORTADA DE LA APLICACION

También es de vital importancia la portada de la aplicación ya que constituye la primera impresión que se lleva el usuario de lo que a continuación verá.

Se construyeron dos portadas, una para cada nivel del sistema, es decir para el administrador y para el usuario, fueron trabajadas en Corel Draw 7.0 y Paint Shop Pro 4.0.

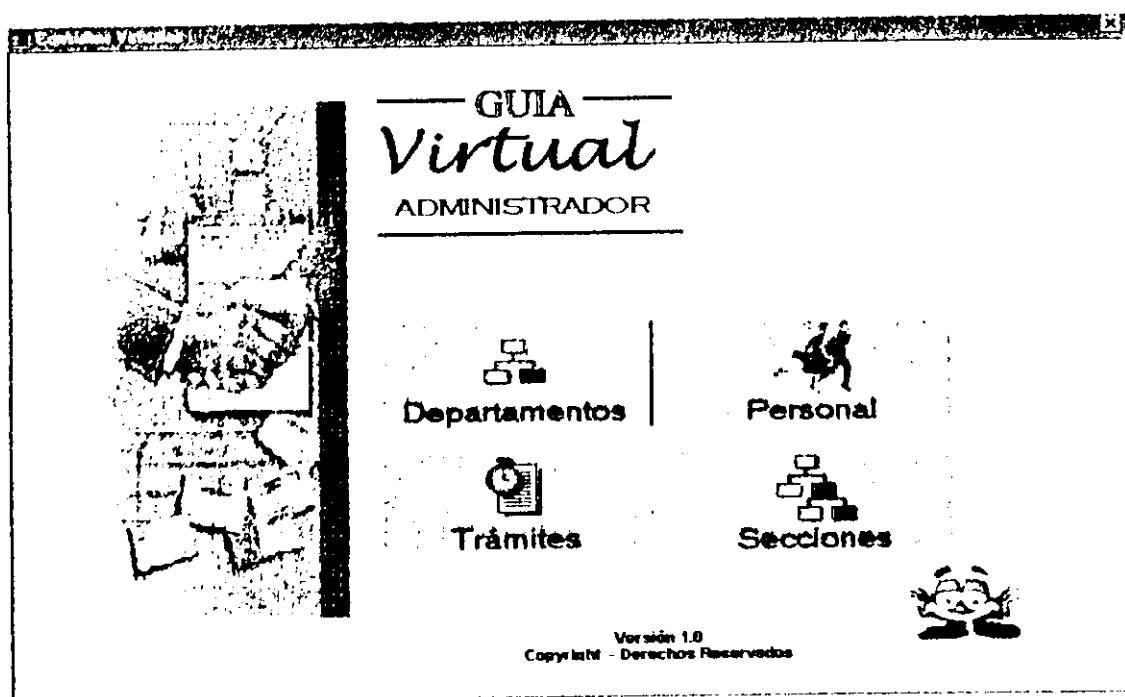


FIGURA 5.7 Portada Guía Virtual Nivel administrador

5.10 ASPECTOS TECNICOS

El diseño de la interface del usuario y la interface entre los lenguajes utilizados, fueron dos aspectos de vital importancia en la construcción y desarrollo de nuestro proyecto, a continuación se detallan las principales características de cada uno de ellos, así como los principales inconvenientes encontrados.

5.10.1 INTERFACE DE USUARIO

Para diseñar el ambiente del Guía se ha tomado especial cuidado de los aspectos técnicos que deben considerarse en la creación de la interface del usuario, los mismos que están ampliamente detallados en el tercer capítulo de este trabajo.

Sin embargo a continuación se presenta un resumen de los aspectos principales que han sido considerados.

- Tomando en cuenta que los posibles usuarios del Guía se clasifican en dos categorías novato y experto, la interface ha sido diseñada para tratar de satisfacer a estos dos tipos de usuario. Dotándolos de un gran poder de navegación y de un sistema de ayuda que les ofrezca tranquilidad y orientación.
- Las pantallas han sido diseñadas de forma similar para que el usuario se sienta rápidamente familiarizado con ellas y con conceptos fáciles de aprender logrando de esta manera una interface simple.
- Se han empleado metáforas de la vida real que serán comprendidas por la mayoría de los usuarios potenciales



FIGURA 5.7 Metáfora de la vida real utilizada en la construcción de la GUI

5.10.2 INTERFACE ENTRE ACCESS Y VISUAL BASIC

Para la lectura de la información introducida por el administrador en Access y su recuperación en las formas respectivas en Visual Basic se tuvieron que programar varias rutinas, ya que surgieron varios inconvenientes de orden técnico al realizar estas operaciones.

También se tuvo que considerar el proceso inverso es decir la escritura de registros desde Visual Basic y su recuperación en Access para el conteo de visitas y la graficación de las mismas.

En cuanto a la interface Access - Visual Basic no existió mayor problema, se incorporaron los objetos de manejo de tablas y se incorporaron las rutinas de lectura de datos. En cuanto a la interface Visual Basic - Access si se presentaron algunas dificultades, en primer lugar se intentó abrir una base de datos creada en Access, escribir la información y cerrarla, pero este procedimiento ocasionó problemas ya que varios objetos asociados a dicha tabla producían errores en el tiempo de ejecución. En un segundo intento se estudiaron las rutinas que permiten crear y manejar una estructura de tipo tabla desde Visual Basic, luego se escribían los datos y se la cerraba, con este procedimiento se eliminaron los errores antes mencionados.

```
Sub Graba()  
Dim x As Integer  
Dim dbsVisita As Database  
Dim rstVisita As Recordset  
  
Set dbsVisita = OpenDatabase("c:\Gvirtual\Virtual\Virtual.mdb")  
Set rstVisita = dbsVisita.OpenRecordset("TblVisita", dbOpenDynaset)  
rstVisita.MoveLast  
x = rstVisita.Fields(0) + 1  
AgregarNombre rstVisita, Str(x), Str(Date), Str(Time)  
rstVisita.Close  
dbsVisita.Close  
End Sub
```

FIGURA 5.8 Subrutina de VB para grabar datos que serán recuperados en Access

5.11 SISTEMA DE AYUDA

Para el nivel administrador se construyó un fichero de ayuda utilizando HelpSribble, la cual contiene en detalle todos los componentes de la aplicación Access con la finalidad de que el usuario pueda contar en todo momento con un medio que lo guíe en la introducción de la información que será manejada por el Guía Virtual.

También se realizó una investigación de la construcción de Hotspots en HelpScribble con la finalidad de incorporar un fichero de ayuda novedoso y agradable al usuario administrador.

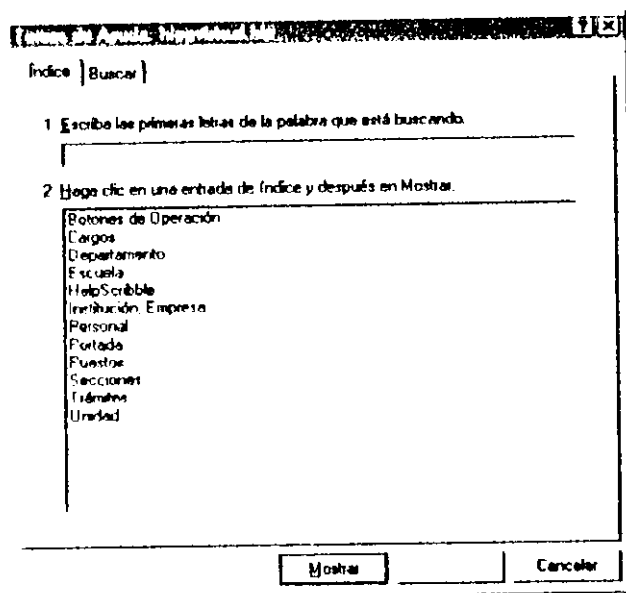


FIGURA 5.9 Pantalla del Fichero de Ayuda de la aplicación nivel administrador

Sin embargo quizás el nivel de ayuda que defina el éxito de la aplicación es aquella que se le proporcione al usuario final, es decir a aquel que no necesariamente tiene conocimientos de computación, ya que de ella dependerá que el usuario no se sienta desorientado o aburrido, por ello se ha incluido en el Guía Virtual un sistema de mensajes que serán transmitidos por la mascota de la aplicación, en forma hablada o escrita



FIGURA 5.10 Sistema de Ayuda del Guía - Nivel Usuario

CAPITULO VI

ANALISIS Y DISEÑO ORIENTADO A OBJETOS

6.1 ANALISIS ORIENTADO A OBJETOS

El análisis orientado a objetos AOO es el reto de comprender el dominio del problema y de esta forma reconocer las responsabilidades del sistema. En esta etapa lo más importante es comprender la profundidad y dominio del problema para poder determinar completamente todos los detalles de una determinada situación. Es importante comprender y modelar el dominio del problema especialmente para sistemas grandes y complejos.

6.1.1 INTERESES DEL USUARIO

- Incluya información concreta
- Sea de fácil manejo
- Incluya fotos
- Imprima los resultados
- Sea novedoso
- Incluya sonido
- De información en poco tiempo
- Informe sobre las principales áreas de la Institución
- Informe sobre la historia de la Institución.

6.1.2 RESPONSABILIDADES DEL SISTEMA

Las principales responsabilidades de nuestro sistema las hemos resumido en los siguientes puntos:

- Crear Empresa
- Crear Departamentos
- Crear Secciones
- Registrar información de la Empresa
- Ingresar información de Departamentos
- Ingresar información de Secciones
- Ingresar información de Personal
- Ingresar información de los principales trámites de la Institución
- Generar un ambiente de la empresa en base a la información introducida
- Generar un ambiente de departamentos en base a la información introducida
- Generar un ambiente de Secciones en base a la información introducida
- Generar un ambiente de Personal en base a la información introducida
- Generar un ambiente de Trámites en base a la información introducida
- Crear objetos para incorporar videos
- Crear objetos para incorporar fotografías
- Crear objetos para incorporar animaciones
- Registrar la visita del usuario
- Almacenar la cuenta de cada visita realizada
- Generar un informe estadístico de visitas realizadas
- Proporcionar información escrita de departamentos
- Proporcionar información escrita de secciones
- Proporcionar información escrita de personal
- Proporcionar información escrita de cargos
- Proporcionar información escrita de trámites.
- Proporcionar un mecanismo de ayuda al usuario

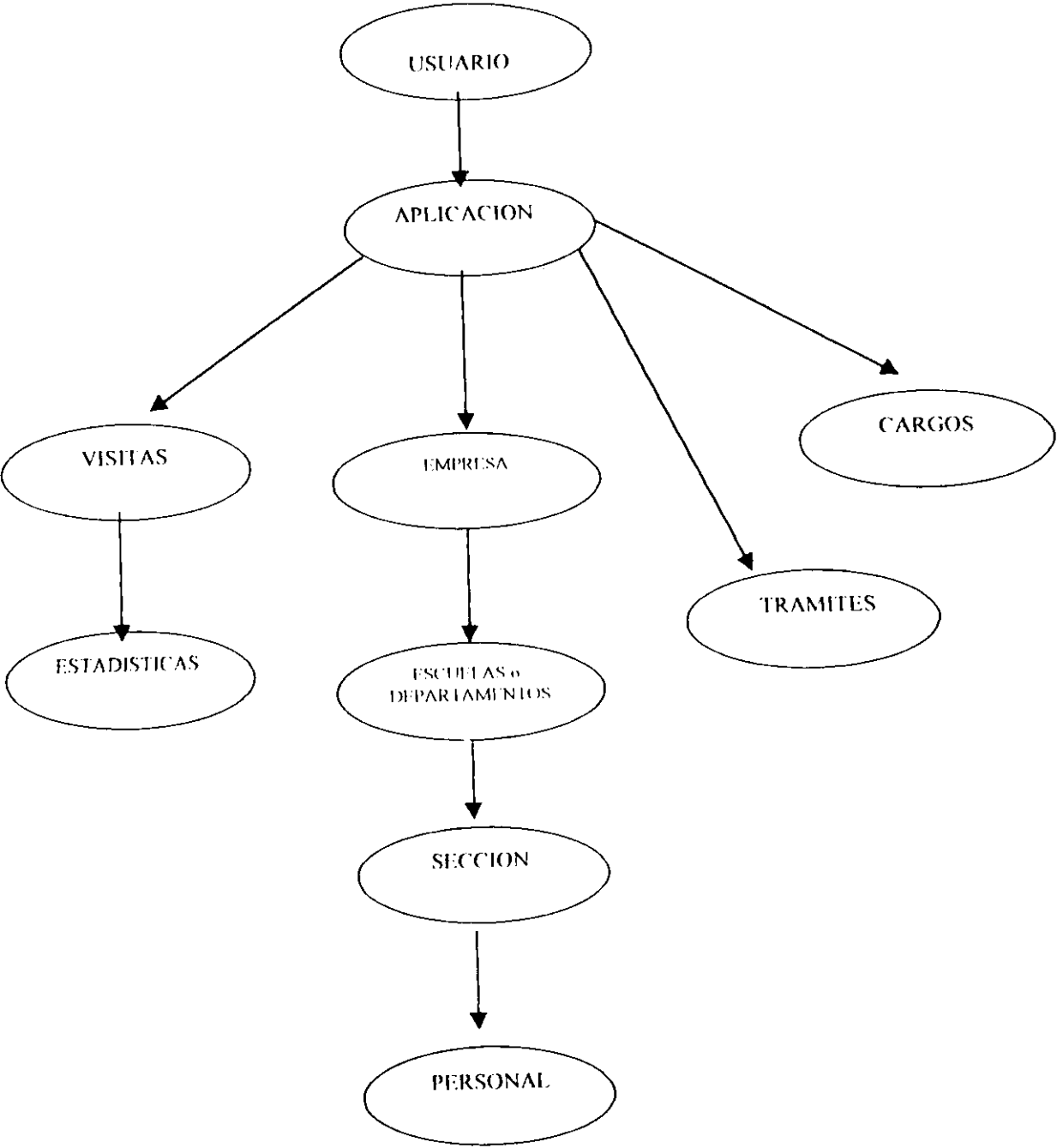
6.1.3 DEFINICION DE CLASES

- Usuario
- Aplicación
- Visitas
- Escuelas o Departamentos
- Secciones
- Personal
- Trámites
- Cargos
- Estadística

6.1.4 DEFINICION DE GUION

AGENTE	ACCION	RECEPTOR	RESULTADO
Aplicación	Generar Guía Virtual	Usuario	Usuario dotado guía
Empresa	Crear empresa	Aplicación	Empresa creada
Visitas	Informar visitas	Aplicación	Aplicación conoce Número Visitas
Estadísticas	Presentar estadísticas	Visitas	Visitas conoce estadísticas
Escuelas o Departam.	Crear Esc. O Dep.	Aplicación	Esc. Dep. creada
Secciones	Crear Secciones	Escuelas o Departam.	Sección creada en Esc. O Dep.
Trámites	Registrar Inf. Trámites	Aplicación	Aplicación registra Inf. Trámites
Personal	Registrar Inf. Personal	Sección	Sección registra Inf. Personal
Cargos	Registrar Información Cargos	Aplicación	Aplicación registra Inf. Cargos

6.1.5 GRAFO DE CONTROL



6.1.6 DEFINICION DE LAS COLABORACIONES

CLASE	SERVICIO SOLICITADO	COLABORADOR
Aplicación	Generar Guía Virtual	Visitas Escuelas o Departamentos Historia
Visitas	Informar Visitas	Estadística
Estadística	Presentar Estadística	
Empresa	Crear Empresa	Departamentos
Escuelas o Departamentos	Crear Esc. o Departam.	Secciones
Secciones	Crear Secciones	Personal
Trámites	Registrar información Trám.	
Personal	Registrar información Pers.	
Cargos	Registrar información Cargos	

6.1.7 DOCUMENTACION DEL ANALISIS

6.1.7.1 DEFINICIONES DE CLASES Y RELACIONES

NOMBRE: *Aplicación*

RESPONSABILIDADES:

Generar Guía Virtual

ATRIBUTOS

COLABORACIONES

SERVICIOS SOLICITADOS	COLABORADOR
Generar Guía Virtual	Visitas
	Empresa
	Trámites
	Cargo

CLASE	TIPO DE RELACION	RELACIONADA CON
Aplicación	Depende de	Usuario

NOMBRE: *Empresa*

RESPONSABILIDADES:

Registrar información de la empresa

ATRIBUTOS

COLABORACIONES

SERVICIOS SOLICITADOS	COLABORADOR
Empresa	Departamentos

CLASE	TIPO DE RELACION	RELACIONADA CON
Empresa	Depende de	Aplicación

NOMBRE: *Visitas*

RESPONSABILIDADES:

Informar Visitas Guía

ATRIBUTOS

COLABORACIONES

SERVICIOS SOLICITADOS	COLABORADOR
Informar # visitas	Estadísticas

CLASE	TIPO DE RELACION	RELACIONADA CON
Visitas	Depende de	Aplicación

NOMBRE: *Estadísticas*

RESPONSABILIDADES:

Presentar informe estadístico

ATRIBUTOS

COLABORACIONES

SERVICIOS SOLICITADOS	COLABORADOR
Usuario conozca estadística visitas	

CLASE	TIPO DE RELACION	RELACIONADA CON
Estadística	Depende de	Visitas

NOMBRE: *Departamentos*

RESPONSABILIDADES:

Crear Escuela o Departamento

ATRIBUTOS

COLABORACIONES

SERVICIOS SOLICITADOS	COLABORADOR
Crear departamento	Secciones

CLASE	TIPO DE RELACION	RELACIONADA CON
Departamentos	Depende de	Empresa

NOMBRE: *Secciones*

RESPONSABILIDADES:

Crear Secciones

ATRIBUTOS

COLABORACIONES

SERVICIOS SOLICITADOS	COLABORADOR
Sección creada	Personal

CLASE	TIPO DE RELACION	RELACIONADA CON
Secciones	Depende de	Escuelas

NOMBRE: Trámites

RESPONSABILIDADES:

Registrar información Trámites

ATRIBUTOS

COLABORACIONES

SERVICIOS SOLICITADOS	COLABORADOR
Información trámites registrada	

CLASE	TIPO DE RELACION	RELACIONADA CON
Trámites	Depende de	Secciones

NOMBRE: Personal

RESPONSABILIDADES:

Registrar información personal

ATRIBUTOS

COLABORACIONES

SERVICIOS SOLICITADOS	COLABORADOR
Personal	

CLASE	TIPO DE RELACION	RELACIONADA CON
Personal	Depende de	Secciones

NOMBRE: Cargos

RESPONSABILIDADES:

Registrar información de cargos

ATRIBUTOS

COLABORACIONES

SERVICIOS SOLICITADOS	COLABORADOR
Cargos	

CLASE	TIPO DE RELACION	RELACIONADA CON
Cargos	Depende de	Aplicación

6.2 DISEÑO ORIENTADO A OBJETOS

A diferencia de la fase de AOO en la que el analista del sistema está centrado en el mundo del usuario en la fase de diseño orientado a objetos DOO éste debe preocuparse de llevar los resultados del análisis hacia una implementación particular hardware software.

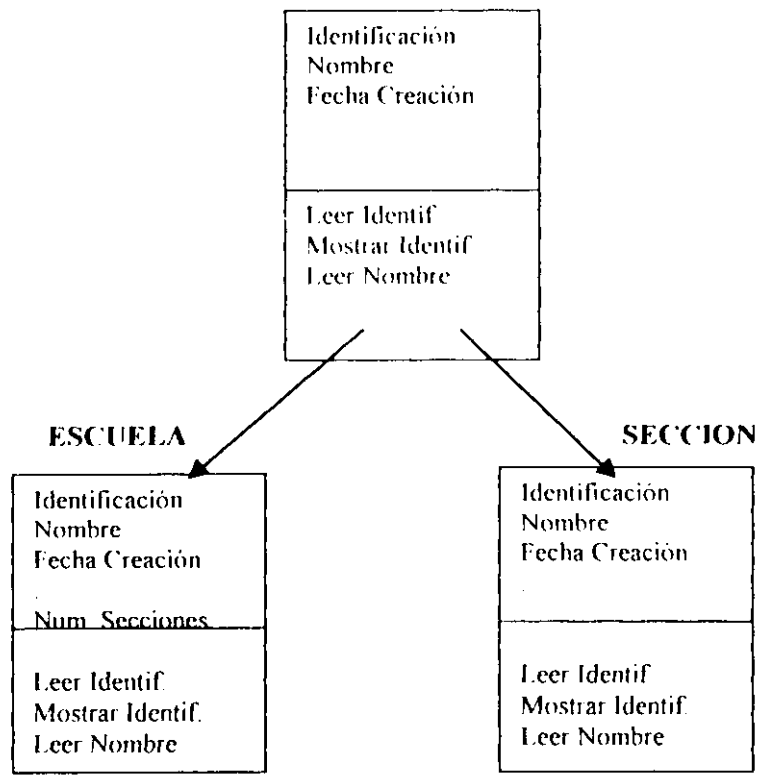
El diseño orientado a objetos enfoca su esfuerzo en la actividad más frontal del diseño del software. Actualmente el énfasis sobre la productividad ha restado importancia a la necesidad de mejoras de la calidad del software. El analista debe garantizar un diseño que permita incluir cambios que no pueden ser anticipados en el momento actual del diseño y para hacerlo deben las partes del sistema que son consideradas como volátiles de aquellas que se consideran estables.

El DOO se basa en los siguientes objetivos:

- Mejorar la productividad
- Incrementar la calidad
- Mejorar el mantenimiento

6.2.1 REFINAMIENTO DE LAS CLASES (Definición de clases abstractas)

ABSTRACCION ES-SE



CLASE:

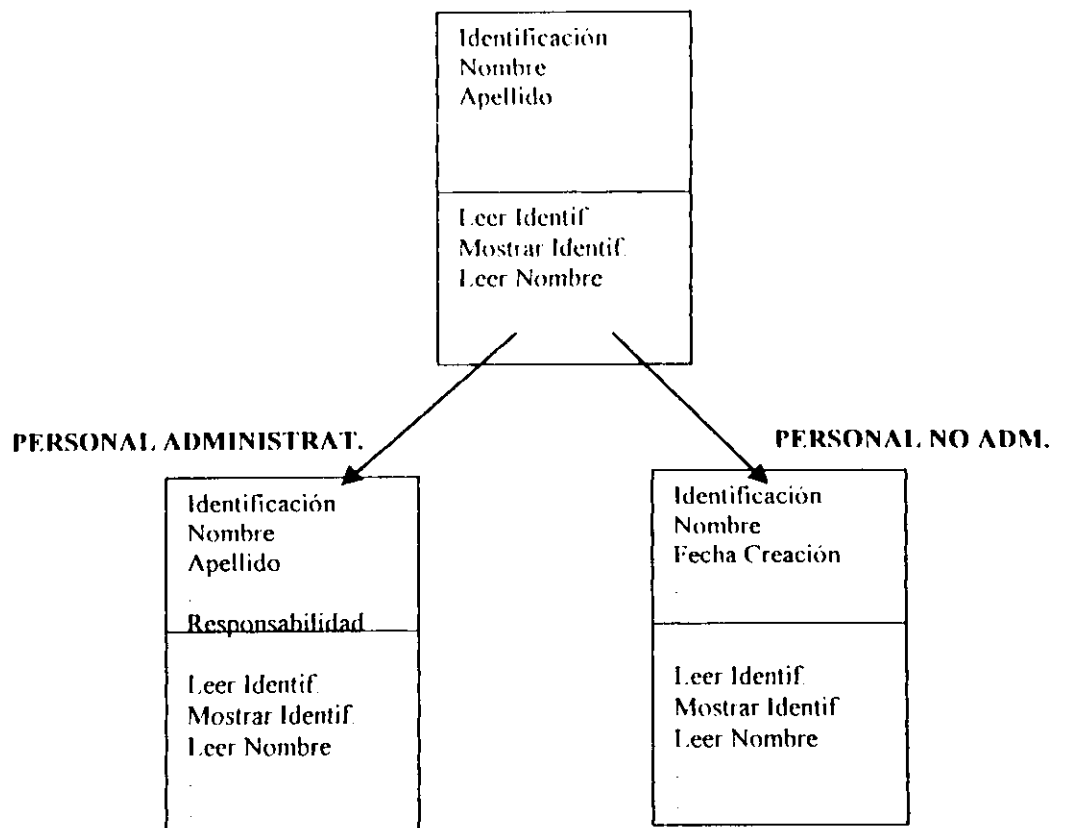
ABSTRACCION ES-SE

DESCRIPCION: Clase Abstracta

RESPONSABILIDADES: Datos comunes para clases especializadas Escuela y Sección

ATRIBUTOS: Identificación
Nombre
Fecha Creación
Imagen
Logotipo
Video
Servicios

ABSTRACCION PERSONAL

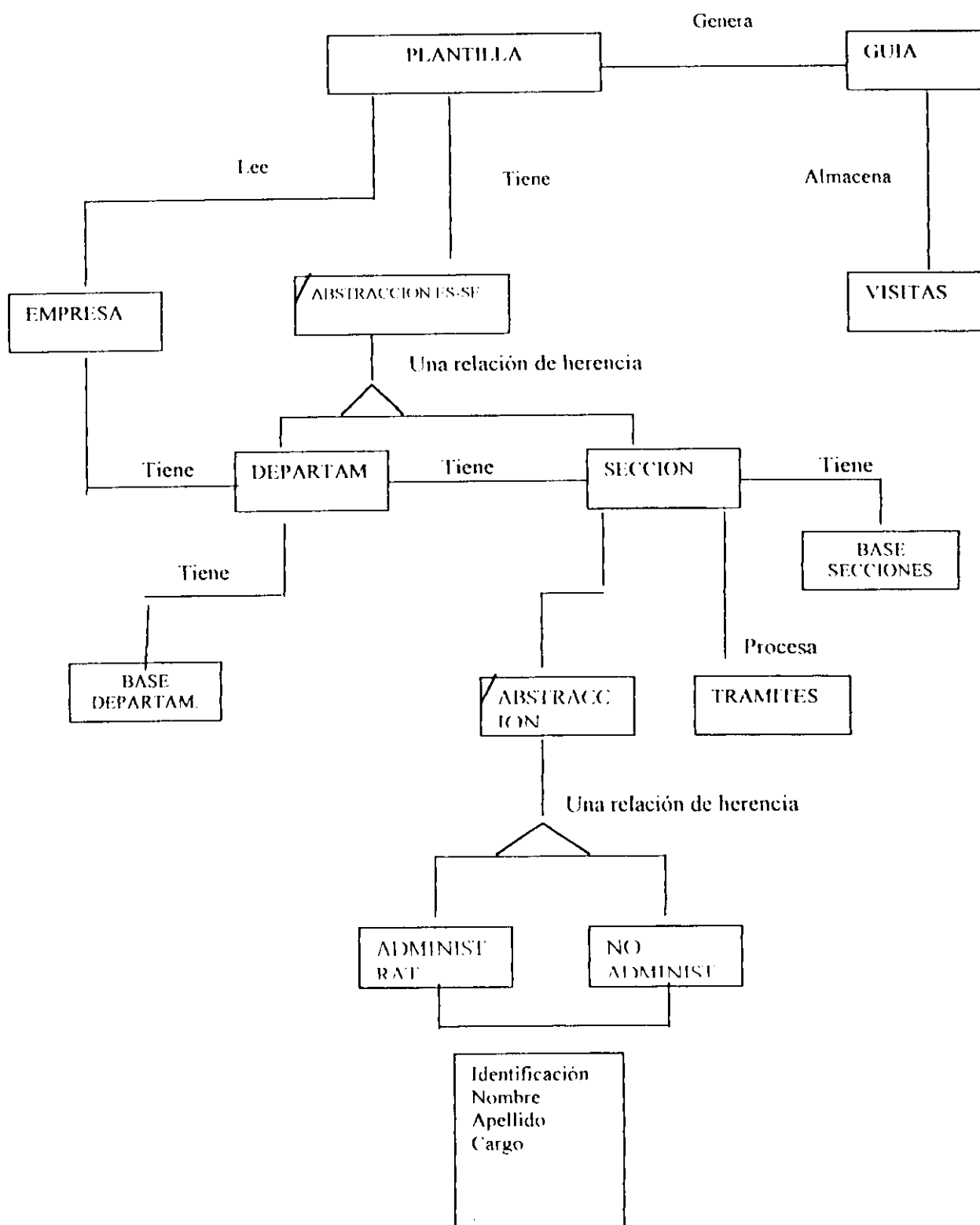


CLASE:

ABSTRACCION PERSONAL

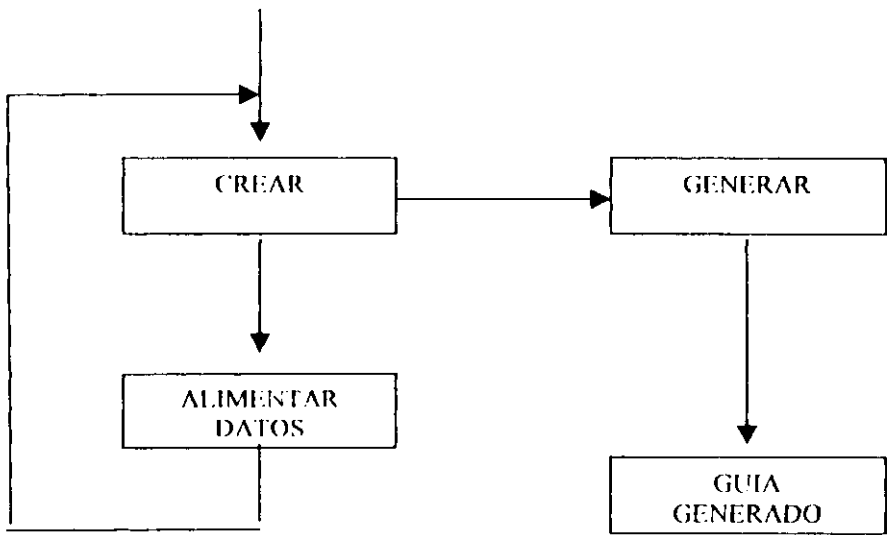
DESCRIPCION:	Clase Abstracta
RESPONSABILIDADES:	Datos comunes para clase especializada
ATRIBUTOS:	Identificación Nombre Apellido Cargo Foto Escuela Sección Funciones

6.2.2 DIAGRAMA DE CLASES



6.2.3 DIAGRAMA DE TRANSICION DE ESTADOS

ESTADO	ACCION
VISITANDO	Incrementar cuenta visitas
CREAR	Inicializar Plantilla
ALIMENTAR	Ingresar datos Institución
GENERAR	Localiza información y genera guia



6.2.3.1 DIAGRAMA #1

NODO:	CREAR
--------------	--------------

ALCANZADO DESDE : Plantilla

ACCION: Inicializa plantilla de datos

SALIDA: Tiene dos transiciones:

- Alimentar datos
- Generar

NODO:	ALIMENTAR DATOS
--------------	------------------------

ALCANZADO DESDE : Crear

ACCION: Ingresa y almacena datos base

SALIDA: Tiene una transición:

- Crear (Da la opción de continuar creando)

NODO:	GENERAR
--------------	----------------

ALCANZADO DESDE : Crear

ACCION: Localiza datos
Genera Guía

SALIDA: Tiene una transición:

- Guía Generado

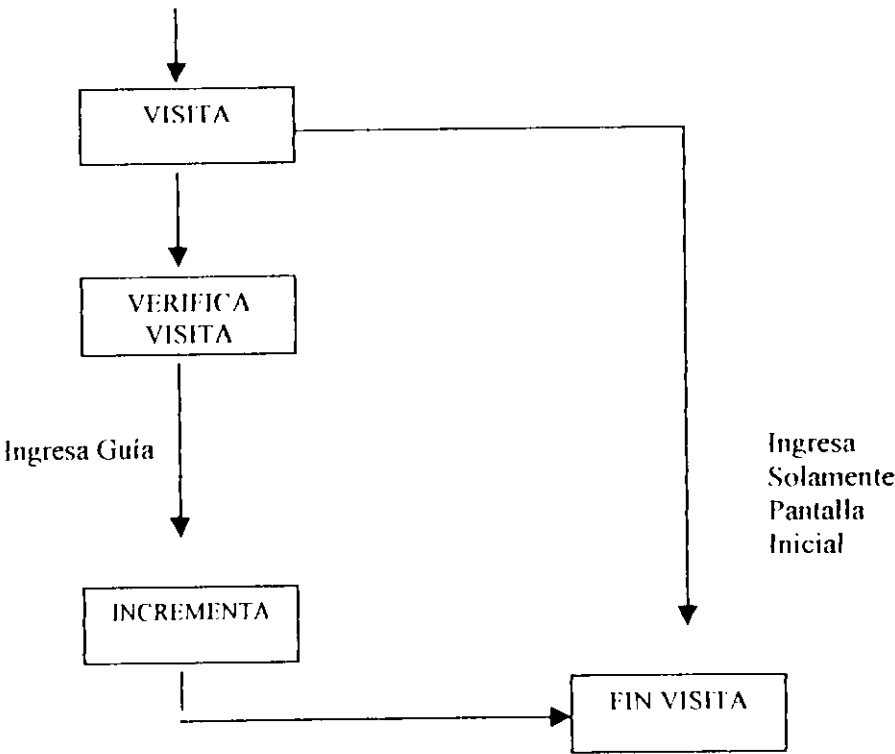
NODO:	GUIA GENERADO
--------------	---------------

ALCANZADO DESDE : Generar

ACCION: Obtener guía

SALIDA: Es el estado final

6.2.3.2 DIAGRAMA #2



NODO:	VISITA
--------------	---------------

ALCANZADO DESDE : Guía

ACCION: Accede a Guía

SALIDA: Tiene una transición:

- Verifica visita

NODO:	VERIFICA VISITA
--------------	------------------------

ALCANZADO DESDE : Visita

ACCION: Chequea acceso

SALIDA: Tiene dos transiciones:

- Incrementar cuenta
- Fin visita

NODO:	INCREMENTA CUENTA
--------------	--------------------------

ALCANZADO DESDE : Verifica visita

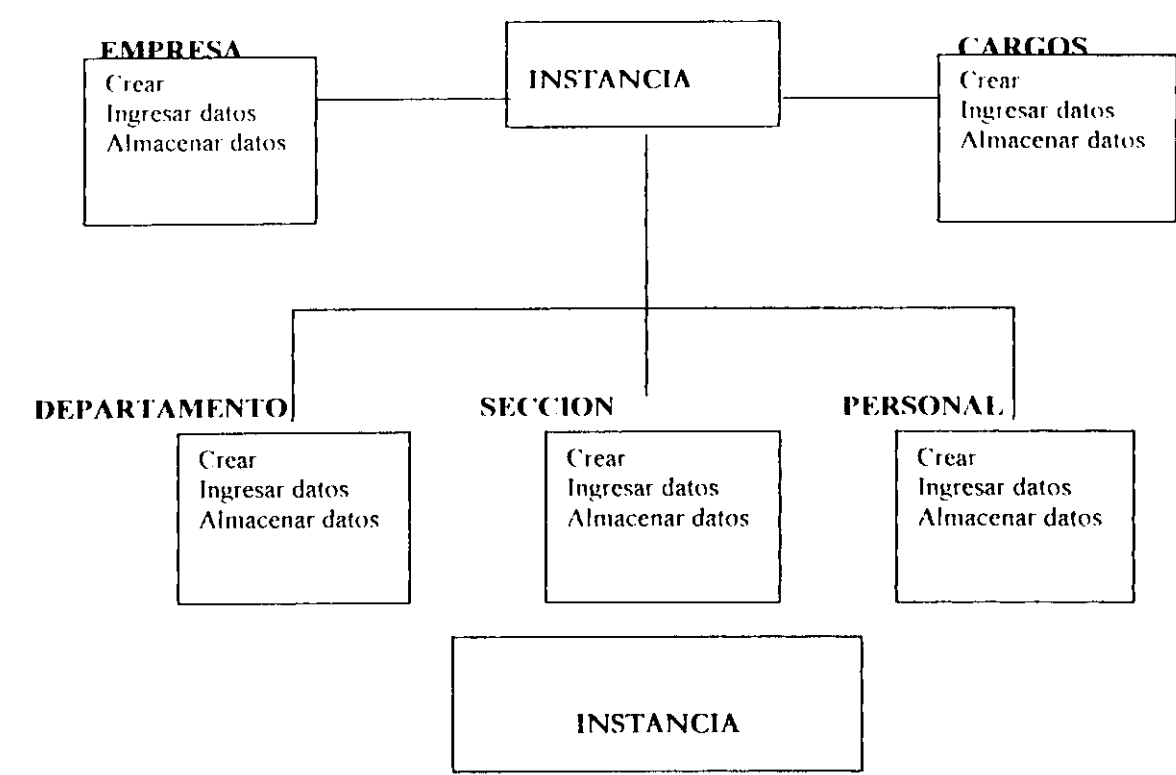
ACCION: Reasigna cuenta
Graba dato

SALIDA: Tiene una transición:

- Fin visita

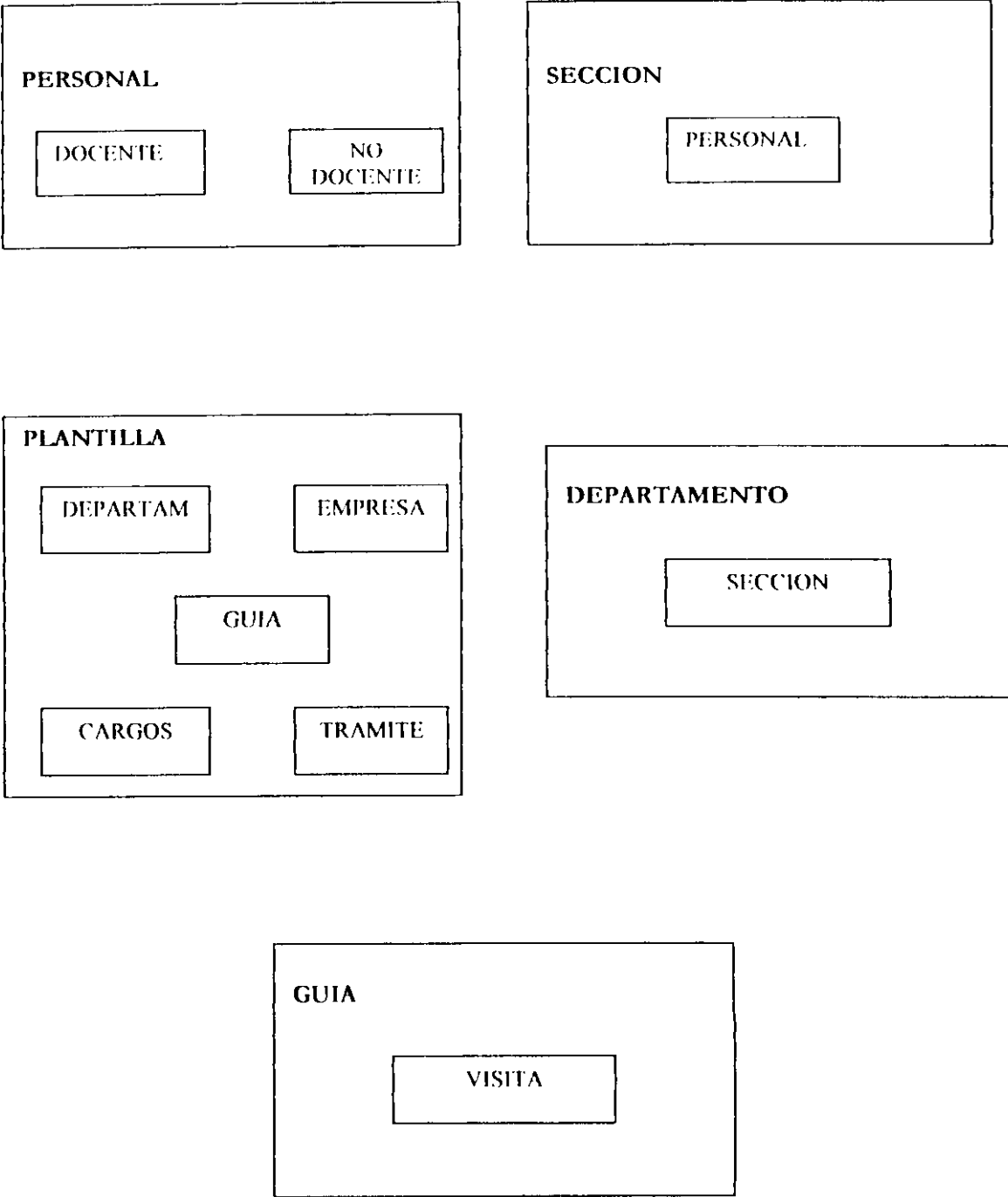
NODO:	FIN VISITA
ALCANZADO DESDE :	Verifica visita
ACCION:	Finalizar sesi3n
SALIDA:	Es el estado final

6.2.4 DEFINICION DE CONTRATOS

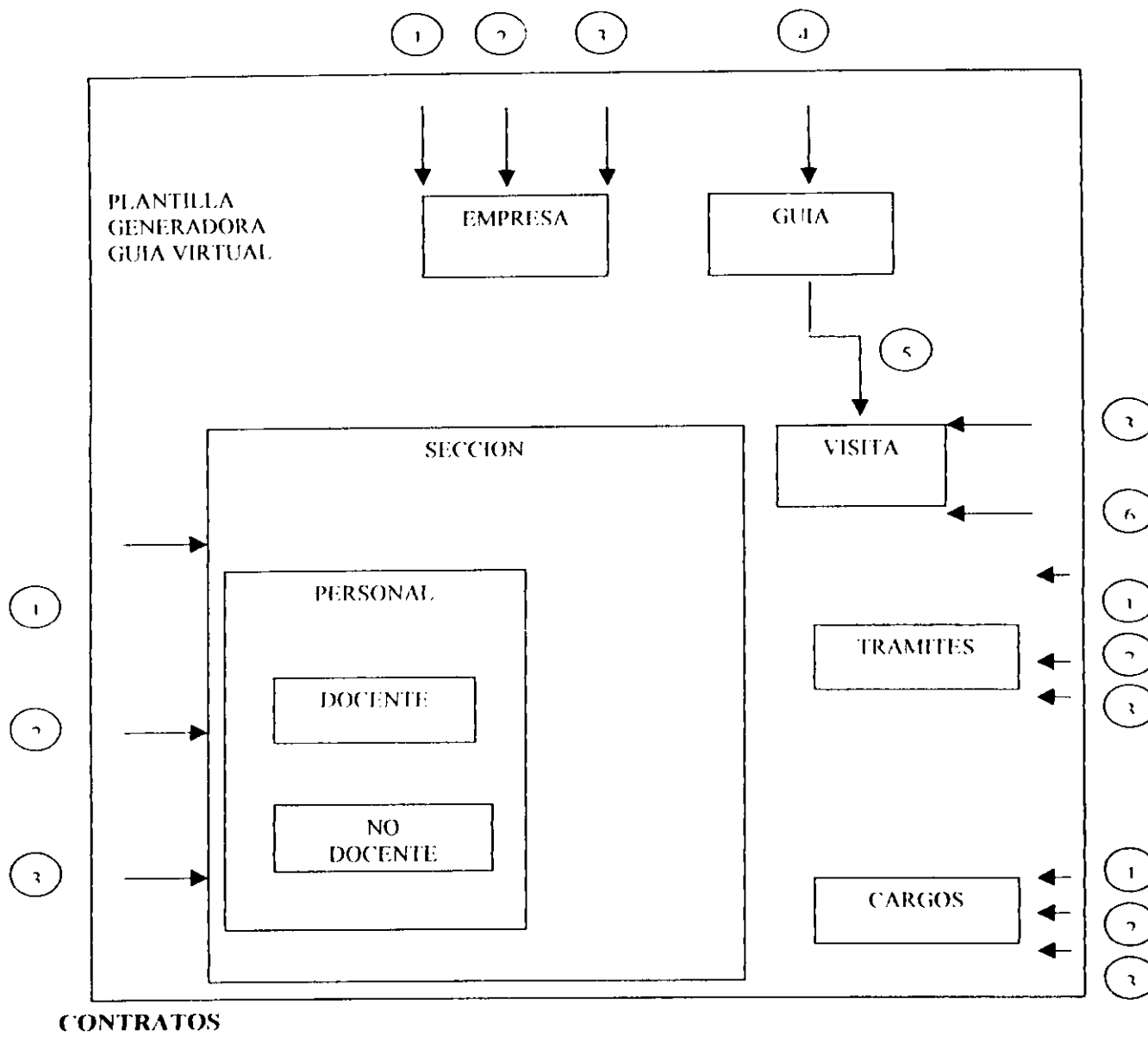


CREAR
INGRESAR DATOS
ALMACENAR DATOS

6.2.5 DEFINICION DE COLABORACIONES



6.2.5 DEFINICION DE SUBSISTEMAS



- 1 Crear
- 2 Ingresar Datos
- 3 Grabar Datos
- 4 Generar Guía
- 5 Verificar Visita
- 6 Incrementar cuenta visita

6.2.7 DOCUMENTACION DEL ANALISIS

6.2.7.1 DICCIONARIO DE CLASES

CLASE:

EMPRESA

DESCRIPCION:

Datos Institución

RESPONSABILIDADES:

Registrar información Empresa

ATRIBUTOS:

Nombre
Publicidad
Logotipo
Historia
Fondo

CLASE:

VISITAS

DESCRIPCION:

Datos número visitas

RESPONSABILIDADES:

Registrar información visitas

ATRIBUTOS:

Fecha
Hora
Cuenta

CLASE: ABSTRACCION ES-SE

DESCRIPCION:	Clase Abstracta
RESPONSABILIDADES:	Datos comunes para clases especializadas Departamento y Sección
ATRIBUTOS:	Identificación Nombre Fecha Creación Imagen Logotipo Video Servicios

CLASE: ABSTRACCION PERSONAL

DESCRIPCION:	Clase Abstracta
RESPONSABILIDADES:	Datos comunes para clase especializada
ATRIBUTOS:	Identificación Nombre Apellido Cargo Foto Escuela Sección Funciones

CLASE:	DEPARTAMENTOS
---------------	----------------------

DESCRIPCION: Datos referenciales de Departamentos

RESPONSABILIDADES: Crear Departamento y registrar su información

ATRIBUTOS: Hereda de Abstracción ES-SE
Numero Sección

CLASE:	SECCIONES
---------------	------------------

DESCRIPCION: Datos referenciales de Secciones

RESPONSABILIDADES: Crear sección y registrar su información

ATRIBUTOS: Hereda de Abstracción ESCUELAS

CLASE:	CARGOS
---------------	---------------

DESCRIPCION: Datos referenciales de Cargos

RESPONSABILIDADES: Crear sección y registrar su información

ATRIBUTOS: Identificador
Nombre
Descripción

CLASE:	PERSONAL ADMINISTRATIVO
---------------	--------------------------------

DESCRIPCION: Datos referenciales de Personal

RESPONSABILIDADES: Crear personal administrativo y registrar su información

ATRIBUTOS: Hereda de Abstracción Personal
Responsabilidad

CLASE:	PERSONAL NO ADMINISTRATIVO
---------------	-----------------------------------

DESCRIPCION: Datos referenciales de Personal

RESPONSABILIDADES: Crear personal administrativo y registrar su información

ATRIBUTOS: Hereda de Abstracción Personal

CLASE:	TRAMITES
---------------	-----------------

DESCRIPCION: Datos referenciales de Trámites

RESPONSABILIDADES: Registrar información trámites

ATRIBUTOS: Identificador
Nombre
Descripción

CAPITULO VII

CONCLUSIONES

El estudio llevado a cabo durante el desarrollo de este proyecto, nos han permitido llegar a varias conclusiones que nos han servido como punto de partida para la creación de la “Plantilla Generadora de Guías Virtuales” y que esperamos sean de utilidad en el desarrollo de nuevos proyectos en este campo.

- El mundo virtual puede volverse interesante si existen objetos en movimiento, al mismo tiempo que se permite que el usuario intervenga como un elemento activo en la aplicación.
- El computador puede ser utilizado como un medio efectivo para reemplazar en cierto ámbito a los seres humanos, en este caso en la atención al público llegando de una manera efectiva y novedosa.
- Las interfaces gráficas prometen futuro y nuevos vínculos que estrechen la interacción entre los seres humanos y las computadoras
- Una buena interface debe reducir la dependencia de las personas de su propia memoria, no debe forzarles a recordar cosas innecesariamente o a repetir operaciones ya realizadas .
- La mejor interface es aquella que exige el menor esfuerzo por parte del usuario final de la aplicación que se está desarrollando.
- Los estándares para las guías de diseño de una interface definen las características de los objetos y sistemas que usamos cada día. Por ello no podemos decir que haya una receta

perfecta para crear una interface perfecta sino que debemos considerar todos los elementos con los que el usuario está familiarizado.

- La interface entre Access y Visual Basic en cuanto al manejo de tablas es buena y no exige mayor esfuerzo en lo que a programación respecta.
- La interface entre Delphi y Access produce ciertos problemas de índole técnico cuando se trata de recuperar las tablas de datos, sobre todo en los campos que contienen imágenes o videos, es decir los de tipo OLE.
- La multimedia utilizada en combinación con la programación orientada a objetos constituye una herramienta poderosa en el desarrollo de aplicaciones interactivas.
- La tecnología de Orientación a Objetos está predestinada para ser la herramienta idónea para el desarrollo de los sistemas de gestión. La complejidad de los sistemas requeridos y los cambios a que estos sistemas están siendo sometidos día tras día, no permitirán seguir utilizando la tecnología de desarrollo actual.
- La Orientación a Objetos es el enfoque de desarrollo de aplicaciones de gestión para el futuro, sobre todo por su productividad proporcionada por la reutilización de componentes.
- La tecnología hipermedia brinda muchas posibilidades en el ámbito documental para la producción de hiperdocumentos que incorporen elementos multimedia que los hagan especialmente atractivos.
- Los avances en las tecnologías de almacenamiento de datos, como el CD-ROM, y de las telecomunicaciones, especialmente Internet, están facilitando la aparición de un gran número

de productos hipermedia (enciclopedias, museos virtuales, periódicos en la red Internet, etc.) con una complejidad creciente. Por ello, es natural el interés que la comunidad dedicada al desarrollo hipermedial está prestando a las metodologías que recientemente han aparecido para racionalizar el proceso de construcción de estas aplicaciones mediante el desarrollo de modelos que faciliten su posterior mantenimiento y reutilización.

BIBLIOGRAFIA

ANALISIS Y DISEÑO O. OBJETOS	Booch Grady
ANALISIS Y DISEÑO O. OBJETOS	Coad and Yourdon
BIBLIA DE MULTIMEDIA	Burger Jeff
COMPUTER GRAPHICS: Principles and Practice	Foley, J. D, Van Dam, A .
COREL DRAW 6 PARA WINDOWS 95	Matthews
DELPHI 3 EN CATORCE DIAS	Batson y Grobman
DESARROLLO DE APLICACIONES MULTIMEDIA	Meriño Roxana
DISEÑO DE INTERFACES GRAFICAS DE USUARIO	Marcus, A.
FUTURE VISIONS, NEW TECHNOLOGIES OF THE SCREEN	Hayward, Philip & Wollen
INTERACCION DEL COLOR	Albers, J.
INFORMATICA BASICA	Alcalde, Garcia
INGENIERIA SANITARIA	Irdenbergh W. A.
MACINTOSH HUMAN INTERFACE GUIDELINES	Apple Computer, Inc.

MICROSOFT ACCESS PARA WINDOWS 95

Viescas John

MICROSOFT ACCESS A SU ALCANCE

Sánchez navarro Daniel

MULTIMEDIA

Rickett, Frank (1993)

PCMAGAZINE, Revista

Volumen 7, Número 4

PCWORDL, Revista

Enero 1997

PCWORDL, Revista

Febrero 1998

PCWORDL, Revista

Marzo 1998

PCWORDL, Revista

Mayo 1998

PERCEPTUAL DISCREPANCIES IN COLOR.

Haferkamp, S., Slater, J

PHYSIOLOGICAL PRINCIPLES FOR THE
EFFECTIVE USE OF COLOR.

Murch, G. M.

PRINCIPLES OF COLOR USE FOR SOFTWARE
DEVELOPERS.

Pancake, C. M.

PROGRAMACION EN WINDOWS 95

Petzold Charles, Yao Paul

THE ART OF HUMAN-COMPUTER INTERFACE
DESIGN

Laurel, Brenda, ed. (1990)

THE RIGHT WAY TO THINK ABOUT
SOFTWARE DESIGN

Nelson, Theodor H. (1990)

TODO EL PODER DE LA MULTIMEDIA

UNA FILOSOFIA DE LAS INTERFACES
GRAFICAS DE USUARIO

Nakatani, L., and Rohrlieh, J

VISUAL BASIC 4 PARA WINDOWS 95

Halvorson

VISUAL BASIC 5.0

Mc GrawHill

http://dir.yahoo.com/Science/Computer_Science/Human_Computer_Interaction_HCI/

<http://www.lce.com.itesm.mx/~rtrejo/Interfaz/open-motif.html>

<http://www.uag.mx/servicios/tdi/comacad/pacopage/document/intgra.htm>

<http://www.uag.mx/servicios/tdi/comacad/pacopage/tutor/intgra/menu.htm>

<http://uazuay.edu.ec/web/web5.htm>

http://guille.costasol.net/vb_resumen.htm

www.eidos.es/trucos/Trucos.htm

www.lasalle.edu.co/~payalam/p_objeto.html

www.unicordoba.edu.co/informatica/poo.html

osiris.staff.udg.mx/Cursos/poo.html

www.danysoft.es/vb2delph.htm

www.losa.com.uy/html/infb/323875.htm

www.ramblainf.com/formacion/cursoVB5.htm

ANEXOS

LISTA DE ANEXOS

- 1 GLOSARIO DE TERMINOS TECNICOS
- 2 ENCUESTAS PERSONAL PUCESA
- 3 PANTALLAS GUIA VIRTUAL.- Nivel Administrador
- 4 INFORMES GUIA VIRTUAL.- Nivel Administrador
- 5 PANTALLAS GUIA VIRTUAL.- Nivel Usuario
- 6 DOCUMENTADOR ACCESS
- 7 DOCUMENTADOR VISUAL BASIC

GLOSARIO DE TERMINOS TECNICOS

ADV	Vistas de Datos Abstractos
BIT	Bit Map File
BLOB	Binary Large Objects
CD	Compact Disk
CUI	Interface de líneas de mandatos
EORM	Enhanced Object- Relationship Model
GUI	Graphical User Interface
HCI	Human computer interaction
HLS	Hue, Light and Saturation
HVS	Human Visual System
IDE	Integrated Development Environment
MCI	Media Control Interface
MIDI	Musical Instrument Digital Interface
ODA	Office Document Architecture
ODBC	Open Database Connectivity
ODL	Office Document Language
OOHDM	Object oriented Hypermedia Design Model
OLE	Object Linking and Embedding
OMT	Object Modeling Technique
OOUI	Object oriented user interfaces
PC	Personal Computer
POO	Programación Orientada a Objetos

RAID	Rapid Application Development
RAM	Random Access Memory
RGB	Red, Green and Black
SGDB	Sistema de Gestión de Bases de Datos
SGBDOO	Sistema de Gestión de Bases de Datos Orientadas a Objetos
VB	Visual Basic
VCL	Visual Component Library
VBA	Visual Basic for Application
VRAM	Video RAM

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
SEDE AMBATO
ESCUELA DE INGENIERIA DE SISTEMAS

***Encuesta para ser aplicada al personal administrativo
encargado de la atención al público***

OBJETIVO

Señor empleado, esta encuesta tiene como finalidad determinar los factores que determinan la calidad de servicio al público que proporciona esta Institución. Por favor responda el siguiente cuestionario en forma clara y concreta. La información que se registre será utilizada como base para diseñar un sistema de atención al público y mejorar este servicio.

DATOS INFORMATIVOS

NOMBRE _____

DEPARTAMENTO _____

CARGO _____ TIEMPO QUE LO DESEMPEÑA: _____

Años

FECHA _____

CUESTIONARIO

- 1 Qué tiempo como promedio se demora en atender a una persona que solicita información acerca de la Universidad?

_____ minutos

- 2 Indique el número aproximado de personas que acuden diariamente a su dependencia en busca de información?

_____ personas

- 3 Qué clase de información es solicitada con mayor frecuencia por los estudiantes que desean ingresar a la Universidad?

4 Considera usted que conoce toda la información requerida por los usuarios acerca de las diferentes carreras y trámites de la PUCESA?

☐ SI

☐ NO

5 Qué personas le proporcionan esta información?

NOMBRE

DEPARTAMENTO

CARGO

6 Que tipo de material es proporcionado al público?

☐ Folletos

☐ Otro

☐ Hojas Informativas

7 Cuáles son los trámites que se realizan con mayor frecuencia en su Departamento?

8 Marque dos aspectos que usted considera son los más importantes para brindar un buen servicio de atención al público?.

☐ Rapidez

☐ Calidad Humana

☐ Información Clara

☐ Otro

9 Le gustaría contar con un sistema computarizado de información al cliente.

☐ SI

☐ NO

Porqué?

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
SEDE AMBATO
ESCUELA DE INGENIERIA DE SISTEMAS

***Encuesta para ser aplicada al personal administrativo
encargado de la atención al público***

OBJETIVO

Señor empleado, esta encuesta tiene como finalidad recopilar la información más importante sobre su Unidad o Departamento, la misma que será introducido en un sistema computarizado cuya finalidad es:

- Descargar el trabajo del personal encargado de la atención al público
- Implantar un sistema novedoso de información al cliente

Por favor responda el siguiente cuestionario en forma clara y concreta. Incluya la información que le gustaría sea proporcionada al público por el nuevo sistema de información, de la calidad de la información que proporcione dependerán los resultados obtenidos para su dependencia.

DATOS INFORMATIVOS

ESCUELA _____

NOMBRE ENCUESTADO _____

CARGO _____ TIEMPO QUE LO DESEMPEÑA: _____

Años _____

FECHA _____

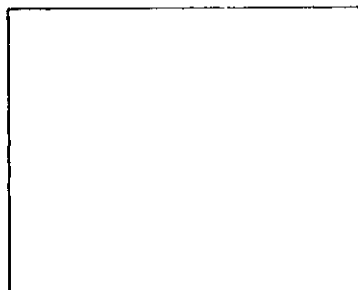
CUESTIONARIO

1. Escriba los siguientes datos referentes a su Unidad

Nombre de la Unidad _____

Ubicación _____

Logotipo o Sello: _____



2 Si en su Unidad existen varias secciones llene la siguiente información: (Una hoja por cada Sección)

Nombre de la Unidad:

Nombre de la Sección:

Ubicación

Síntesis de las funciones de la Sección:

Personal que trabaja en la Sección (Listado de Nombres)

-
-
-
-
-
-
-
-
-
-

Principales trámites que se realizan en esta sección (listado)

-
-
-
-
-
-
-
-
-
-

3 Llene la siguiente información por cada empleado que trabaja en la sección

Nombre de la Unidad

Nombre de la Sección:

DATOS DEL EMPLEADO

Nombres

Apellido

4 Indique la información solicitada por cada trámite que se realice en la Sección

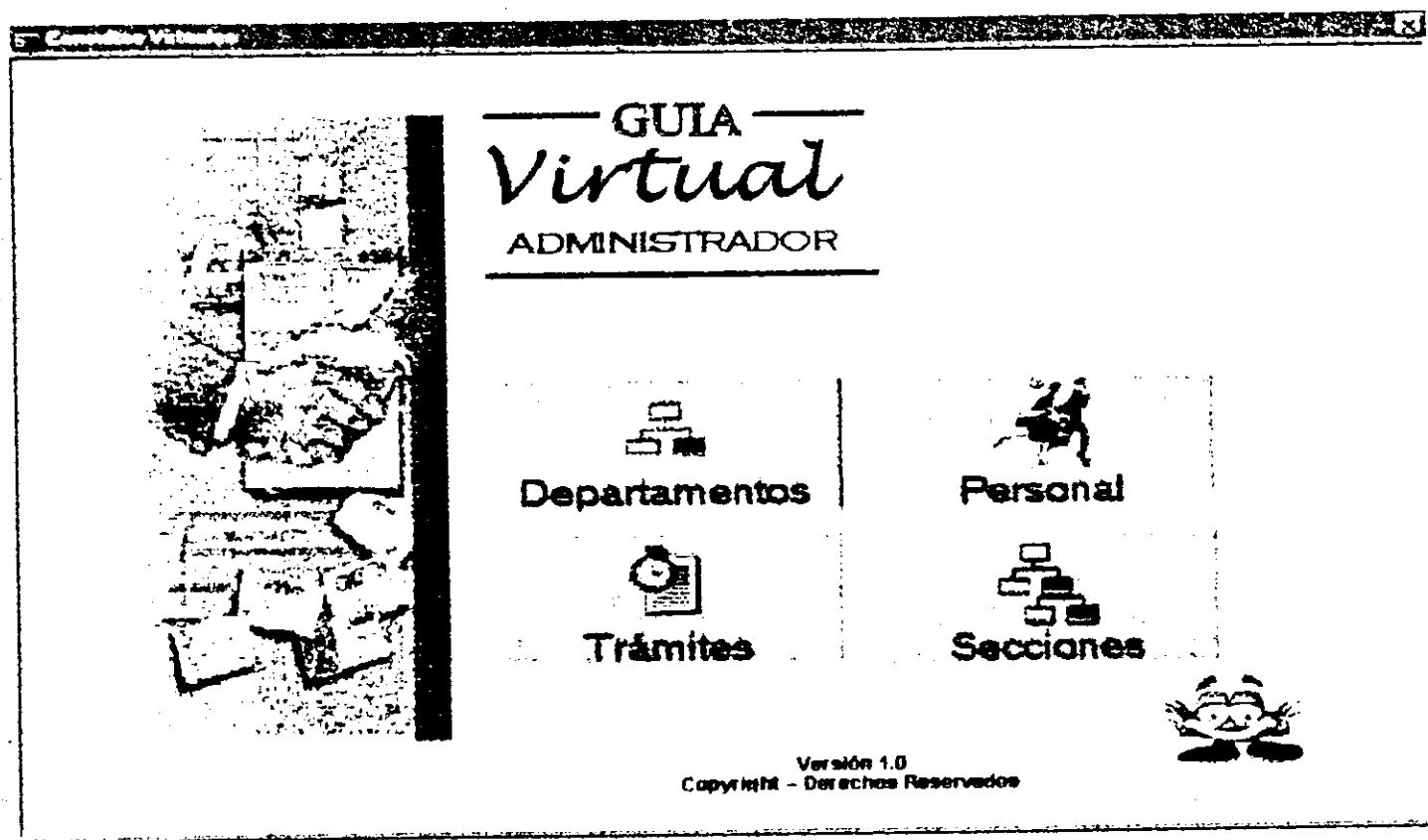
Nombre de la Unidad:

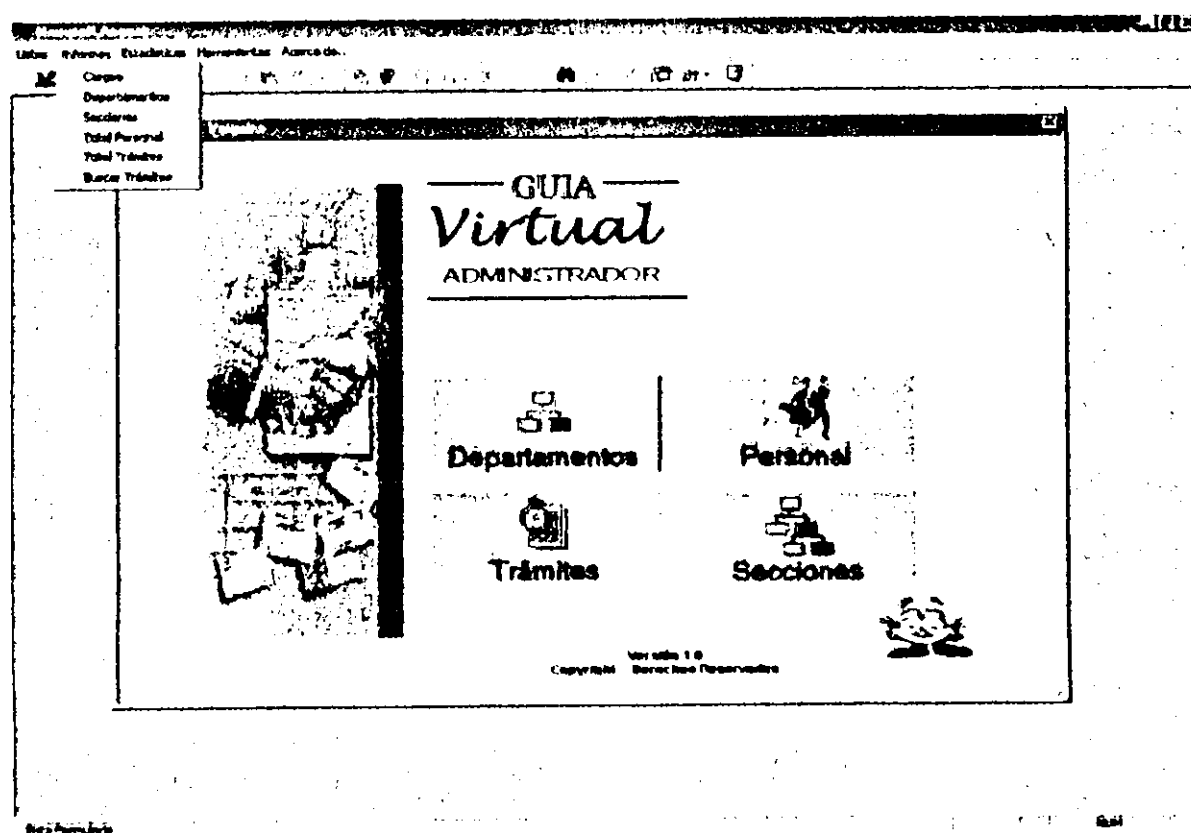
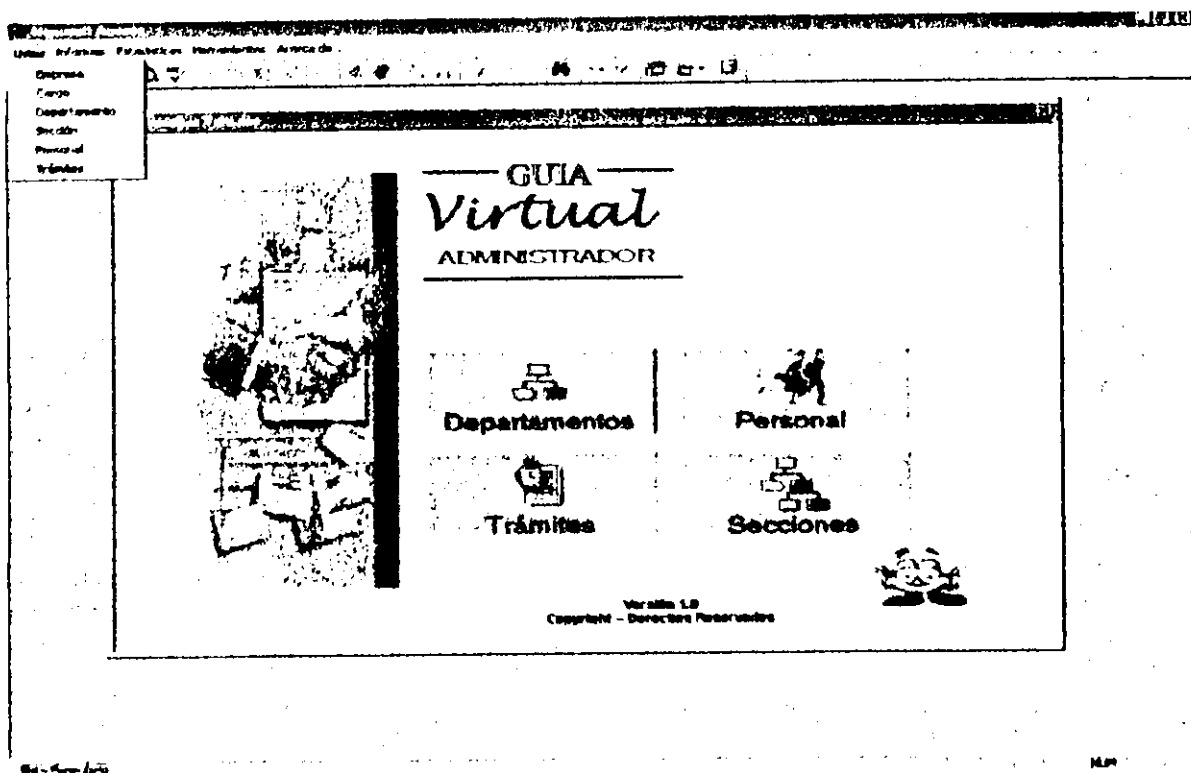
Nombre de la Sección:

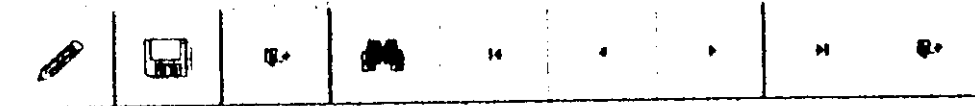
Nombre del Trámite

Descripción o razón del trámite (Opcional)

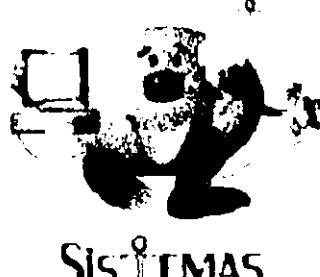
Procedimiento a seguir (Sea lo más claro y descriptivo posible)

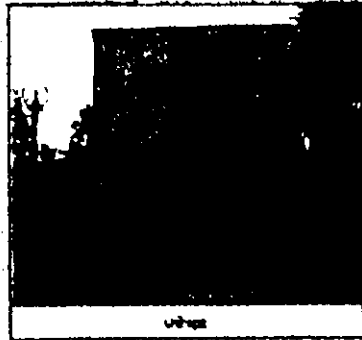


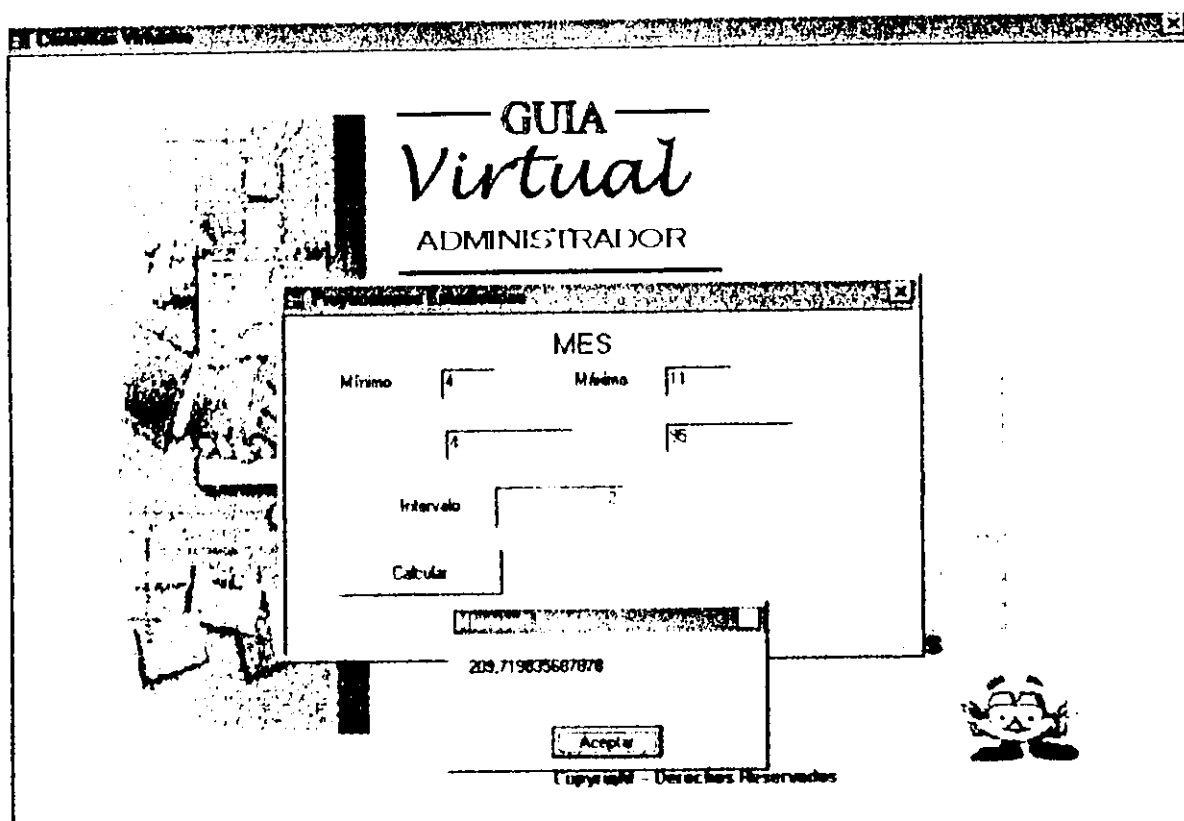


EMPRESA	
Identificador	Empresa
PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR AMBATO	
Slogan	Historia
Fondo	Logo
	
Monseñor Vicente Cisneros Durán, fue el mentalizador de la presencia de un núcleo universitario para la ciudad de Ambato, ya que ha visto siempre la necesidad e importancia de la educación católica. Este proyecto que ha sido inquietud y aspiración de todos los tungurahwenses, hoy es una gran realidad. De esta forma se hace pública, estable y universal la presencia del pensamiento cristiano, y ha formado hombres prestigiosos por su doctrina, preparados para el desempeño de las funciones más importantes en la sociedad y testigos de la fe.	
Abrir	Abrir
	

CARGO	
Identificador	C-001
Cargo	DIRECTOR DE ESCUELA
Descripción	Será el representante de la Escuela ante el H. Consejo Directivo, tiene la reponsabilidad de la administración de la Escuela a su cargo Entre sus funciones podemos citar las siguientes: - Cumplir y hacer cumplir los Estatutos y Reglamentos de la Universidad - Velar por la excelencia académica de la Escuela - Proponer al Consejo Directivo la contratación de profesores - Supervisar las labores de Secretaría y las demás dependencias de la
	

DEPARTAMENTO	
Identificador	PUCE-D0001
Departamento	SISTEMAS
Ubicación	El Tropezón - Bloque #1
Descripción	Formamos un profesional con una educación humanística y ética, altamente capacitado en el manejo de nuevas tecnologías, comprometido con el desarrollo social y tecnológico del país. Ingeniería de Sistemas es una carrera que entiende el área de la computación como una disciplina académica y como una profesión dentro del contexto de una sociedad más amplia, en la que el ser humano es su protagonista y por tanto su responsable
	
Abrir	
	

SECCION			
Identificador	Sección	DIRECCION SISTEMAS	Departamento
			SISTEMAS
Actividad	Ubicación		Sección
Controlar y propender al mejor desenvolvimiento de las labores académicas	Tropezón - Bloque#1 - 1er piso		
Trámites	Imagen	Visita	Abrir
Autorización de Matriculas Extensión de certificados			
	Abrir	Abrir	
			



GUIA VIRTUAL

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR AMBATO

CARGOS

<i>Identificador</i>	<i>Cargo</i>
C-001	DIRECTOR DE ESCUELA
C-002	PROFESOR
C-003	SECRETARIO(A)

GUIA VIRTUAL

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR AMBATO

DEPARTAMENTOS

Identificador	Departamento	Ubicación
PUCE-D0006	ADMINISTRATIVO CENTRAL	EL TROPEZÓN
PUCE-D0002	DISEÑO	EL TROPEZÓN - BLOQUE #2
PUCE-D0004	GERENCIA EN PYMES	EL TROPEZÓN - BLOQUE #2
PUCE-D0005	LINGÜÍSTICA	MEDALLA MILAGROSA - 4TO. PISO
PUCE-D0003	OPTOMETRIA	MEDALLA MILAGROSA - 1RO Y 5TO PISO
PUCE-D0001	SISTEMAS	EL TROPEZÓN - BLOQUE #1

GUIA VIRTUAL

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR AMBATO

DEPARTAMENTOS

Nombre *PUCE-D000 SISTEMAS*

<i>Código</i>	<i>Sección</i>	<i>Ubicación</i>
PUCE-S0	AREA ACADEMICA	EL TROPEZÓN - BLOQU
PUCE-S0	CENTRO DE COMPUTO	EL TROPEZÓN - BLOQU
PUCE-S0	DIRECCION SISTEMAS	TROPEZÓN - BLOQUE#
PUCE-S0	LABORATORIO NOVELL	EL TROPEZÓN - BLOQU
PUCE-S0	LABORATORIO NT	EL TROPEZÓN - BLOQU
PUCE-S0	LABORATORIO WINDOWS	EL TROPEZÓN - BLOQU
PUCE-S0	SECRETARIA SISTEMAS	TROPEZÓN - BLOQUE #

Nombre *PUCE-D000 DISEÑO*

<i>Código</i>	<i>Sección</i>	<i>Ubicación</i>
---------------	----------------	------------------

PUCE-S0 DIRECCION DISEÑO

EL TROPEZÓN - BLOQU

PUCE-S0 SUBDIRECCION DISEÑO

EL TROPEZÓN - BLOQU

PUCE-S0 TALLERES DISEÑO

Nombre *PUCE-D000 OPTOMETRIA*

<i>Código</i>	<i>Sección</i>	<i>Ubicación</i>
PUCE-S0	DIRECCION OPTOMETRIA	MEDALLA MILAGROSA -

PUCE-S0 OPTICA

PUCE-S0 SECRETARIA OPTOMETRIA

MEDALLA MILAGROSA -

Nombre *PUCE-D000 GERENCIA EN PYMES*

<i>Código</i>	<i>Sección</i>	<i>Ubicación</i>
PUCE-S0	DIRECCION PYMES	EL TROPEZÓN - BLOQU

PUCE-S0 SECRETARIA PYMES

EL TROPEZÓN - BLOQU

Nombre *PUCE-D000 LINGÜÍSTICA*

Código	Sección	Ubicación
PUCE-S0	CO-SIRECCION	MEDALLA MILAGROSA -
PUCE-S0	DIRECCION LINGÜÍSTICA	MEDALLA MILAGROSA -
PUCE-S0	LABORATORIO LINGÜÍSTICA	MEDALLA MILAGROSA -
PUCE-S0	SECRETARIA LINGÜÍSTICA	MEDALLA MILAGROSA -

Nombre *PUCE-D000 ADMINISTRATIVO CENTRAL*

Código	Sección	Ubicación
PUCE-S0	CONTABILIDAD	MEDALLA MILAGROSA -
PUCE-S0	DIRECCION FINANCIERA	MEDALLA MILAGROSA -
PUCE-S0	SECRETARIA (AUX) GENERAL	MEDALLA MILAGROSA -
PUCE-S0	SECRETARIA GENERAL	MEDALLA MILAGROSA -

GUIA VIRTUAL

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR AMBATO

TOTAL EMPLEADO

Código	PUCE-	Departamento	SISTEMAS		
Código	PUCE-S	Sección	DIRECCI	Cantidad	1
Apellido	Nombre		Cargo		
MEDINA CHICAIZA	RICARDO PATRICIO		PROFESOR		

Código	PUCE-S	Sección	SECRET	Cantidad	1
Apellido	Nombre		Cargo		
CHILQUINGA	LORENA		DIRECTOR DE ESCUELA		

GUIA VIRTUAL

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR AMBATO

TRAMITES

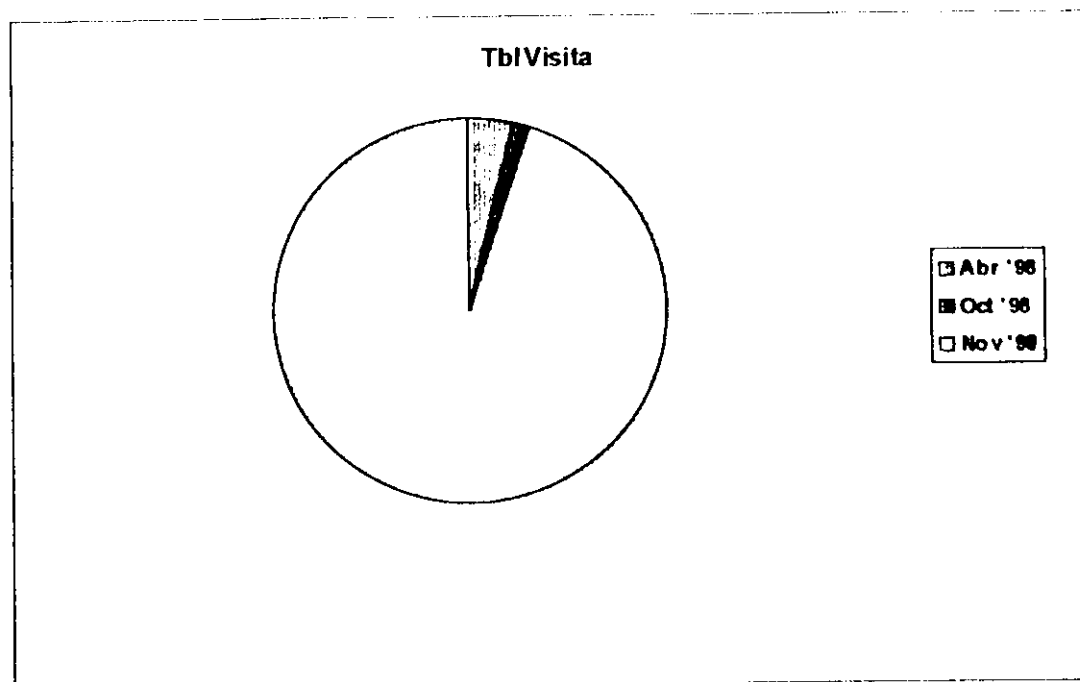
Identificador

Trámite

T-0003	CERTIFICACION DE ASISTENCIA
T-0004	CERTIFICACION DE CONDUCTA
T-0008	CERTIFICACION DE DOCUMENTOS
T-0006	CERTIFICACION DE EGRESAMIENTO
T-0002	CERTIFICACION DE MATRICULA
T-0005	CERTIFICACION DE NOTAS
T-0009	ENTREGA DE NOTAS (Profesores)
T-0001	MATRICULA
T-0007	PASE DE NOTAS
T-0011	PRESENTACION DE PLAN DE TESIS (INGENIERIA)
T-0010	PRESENTACION DE EXAMENES COMPLEXIVOS

Estadística de Visitas

Month	Visits
Abr '98	3
Oct '98	1
Nov '98	95





— GUIA — *Virtual* **INTERACTIVO**



Departamentos



Trámites



Personal



Secciones



Versión 1.0
Copyright – Derechos Reservados

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR AMBATO

En busca de una educación integral



PUCE SA



HISTORIA

Monseñor Vicente Cisneros Durán, fue el mentalizador de la presencia de un núcleo universitario para la ciudad de Ambato, ya que ha visto siempre la necesidad e importancia de la educación católica.

Este proyecto que ha sido inquietud y aspiración de todos los tungurahueses, hoy es una gran realidad.

De esta forma se hace pública, estable y universal la presencia del pensamiento cristiano, y ha formado hombres prestigiosos por su doctrina, preparados para el desempeño de las funciones más importantes en la sociedad y testigos de la fe en el mundo.

En las albas páginas de la historia de la Universidad, se inscribió con letras de oro la fecha lunes 6 de octubre 1986 cuando oficialmente se iniciaron las actividades universitarias, bajo la aprobación de la sede central dirigida por el Padre Doctor Julio Terán Dutari. Inicialmente funcionó en Izamba y posteriormente en el Centro Social

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR AMBATO

En busca de una educación integral



DEPARTAMENTOS

PUCE-D0001	SISTEMAS
PUCE-D0002	DISEÑO
PUCE-D0003	OPTOMETRIA
PUCE-D0004	GERENCIA EN PYMES
PUCE-D0005	LINGÜISTICA
PUCE-D0006	ADMINISTRATIVO CENTRAL

SECCIONES

PUCE-S0001	DIRECCION SISTEMAS
PUCE-S0002	SECRETARIA SISTEMAS
PUCE-S0004	AREA ACADEMICA
PUCE-S0005	CENTRO DE COMPUTO
PUCE-S0006	LABORATORIO NOVELL
PUCE-S0007	LABORATORIO WINDOWS
PUCE-S0008	LABORATORIO NT

SISTEMAS

Ubicación Departamento
El Tropezón - Bloque #1



Ubicación Sección

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR AMBATO

En busca de una educación integral



DEPARTAMENTOS

PUCE-D0001	SISTEMAS
PUCE-D0002	DISEÑO
PUCE-D0003	OPTOMETRIA
PUCE-D0004	GERENCIA EN PYMES
PUCE-D0005	LINGÜÍSTICA
PUCE-D0006	ADMINISTRATIVO CENTRAL

Logotipo



Secciones



Monitor

Ubicación

PUCE-D0001 SISTEMAS

El Tropezón - Bloque #1

Descripción

Formamos un profesional con una educación humanística y ética, altamente capacitado en el manejo de nuevas tecnologías, comprometido con el desarrollo social y

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR AMBATO

En busca de una educación integral



DEPARTAMENTO

PUCE-D0001	SISTEMAS
PUCE-D0002	DISEÑO
PUCE-D0003	OPTOMETRIA
PUCE-D0004	GERENCIA EN PYMES
PUCE-D0005	LINGÜISTICA
PUCE-D0006	ADMINISTRATIVO CENTRAL

DISEÑO

Ubicación Departamento

El Tropezón - Bloque #2



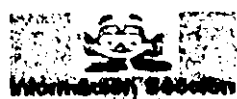
SECCIONES

PUCE-S0009	DIRECCION DISEÑO
PUCE-S0010	SUBDIRECCION DISEÑO
PUCE-S0011	TALLERES DISEÑO

DIRECCION DISEÑO

Ubicación Sección

El Tropezón - Bloque #2 - 2do piso



PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR AMBATO

En busca de una educación integral



DIRECCION DISEÑO

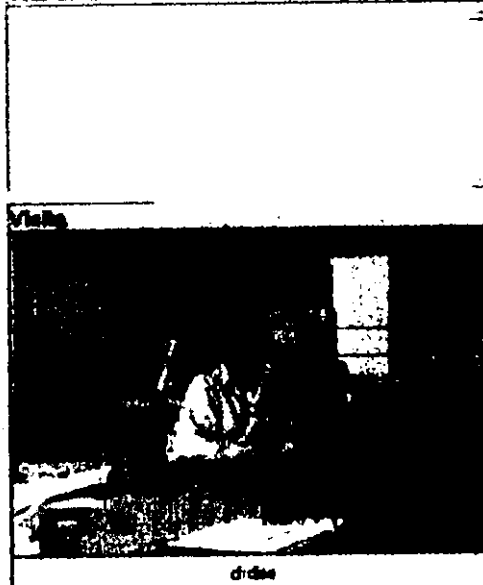


El Tropezón - Bloque #2 - 2do piso

Actividad

Sus actividades están encaminadas a la planificación, organización y control de la vida académica y administrativa de la Escuela.

Trámites



diseño



PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR AMBATO

En busca de una educación integral



EMPLEADOS

PUCE-P0001 MEDINA CHICALZA RICARDO PATRICIO
PUCE-P0002 CHILQUINGA LORENA

Foto



CHILQUINGA

LORENA

Departamento SISTEMAS

Sección SECRETARIA SISTEMAS

Carga DIRECTOR DE ESCUELA



Platón

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR AMBATO

En busca de una educación integral



TRAMITES

- T-0001 MATRICULA
- T-0002 CERTIFICACION DE MATRICULA
- T-0003 CERTIFICACION DE ASISTENCIA
- T-0004 CERTIFICACION DE CONDUCTA
- T-0005 CERTIFICACION DE NOTAS
- T-0006 CERTIFICACION DE EGRESAMIENTO
- T-0007 PASE DE NOTAS
- T-0008 CERTIFICACION DE DOCUMENTOS
- T-0009 ENTREGA DE NOTAS (Profesores)
- T-0010 PRESENTACION DE EXAMENES COM
- T-0011 PRESENTACION DE PLAN DE TESIS (I

Procedimiento

Llenar la hoja de créditos en la Secretaría de la Escuela, teniendo en cuenta el horario y los pre-requisitos académicos.

Dejar la hoja de créditos en la Secretaría de la Escuela

Pagar el valor de la matriculas más los créditos en la Administración Financiera

Legalizar la matricula en la Secretaría General



Platón

Propiedades

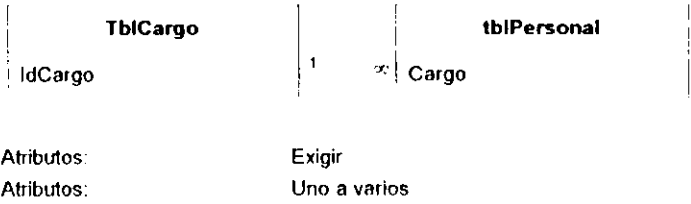
ActivarOrden:	Falso	Def. actualizable:	Verdadero
Fecha de creación:	15/05/98 07:34:17 p.m.	RecordCount:	3
Última actualización:	15/11/98 05:38:16 p.m.		

Columnas

Nombre	Tipo	Tamaño
IdCargo	Texto	6
Nombre	Texto	40
Descripción	Memo	-

Relaciones

TblCargotblPersonal



Índices de tabla

Nombre	Número de campos
IdCargo	1
Campos:	IdCargo, Ascendente
PrimaryKey	1
Campos:	IdCargo, Ascendente

Propiedades

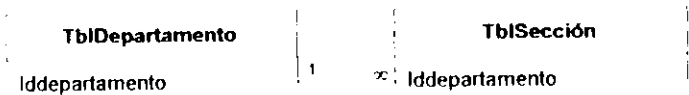
ActivarOrden:	Falso	Def. actualizable:	Verdadero
Fecha de creación:	03/04/98 09:49:52 a.m.	RecordCount:	6
Última actualización:	15/11/98 05:39:34 p.m.		

Columnas

Nombre	Tipo	Tamaño
Iddepartamento	Texto	10
Nombre	Texto	40
Ubicación	Texto	40
Antecedentes	Memo	-
Logo	Objeto OLE	-

Relaciones

tblDepartamentotblSección



Atributos: Exigir
Atributos: Uno a varios

Índices de tabla

Nombre	Número de campos
Iddepartamento	1
Campos:	Iddepartamento, Ascendente
PrimaryKey	1
Campos:	Iddepartamento, Ascendente

Propiedades

ActivarOrden:	Falso	Def. actualizable:	Verdadero
Fecha de creación:	22/08/98 03:23:20 p.m.	RecordCount:	1
Última actualización:	10/11/98 05:26:35 p.m.		

Columnas

Nombre	Tipo	Tamaño
IdEmpresa	Texto	5
Nombre	Texto	50
Fondo	Objeto OLE	-
Logo	Objeto OLE	-
Historia	Memo	-
Publicidad	Texto	40

Índices de tabla

Nombre	Número de campos
IdEmpresa	1
Campos:	IdEmpresa, Ascendente
PrimaryKey	1
Campos:	IdEmpresa, Ascendente

Propiedades

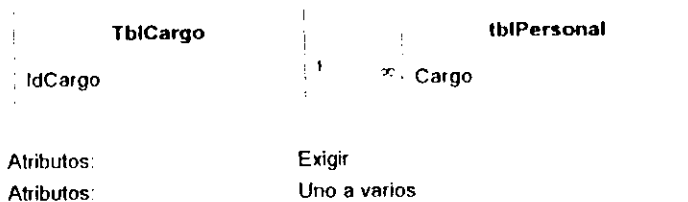
ActivarOrden:	Falso	Def. actualizable:	Verdadero
Fecha de creación:	03/04/98 09:58:59 a.m.	RecordCount:	2
Última actualización:	15/11/98 05:40:00 p.m.		

Columnas

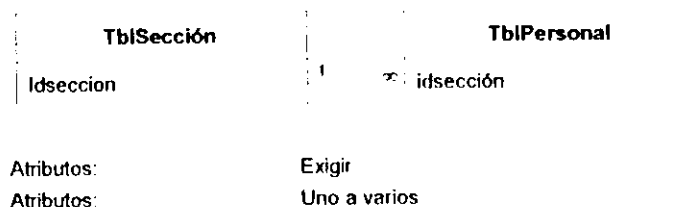
Nombre	Tipo	Tamaño
idPersonal	Texto	10
Nombre	Texto	25
Apellido	Texto	25
Cargo	Texto	6
Foto	Objeto OLE	-
idsección	Texto	10

Relaciones

TblCargotblPersonal



tblSeccióntblPersonal



Índices de tabla

Nombre	Número de campos
idPersonal	1
Cargos:	IdPersonal, Ascendente
idsección	1
Cargos:	idsección, Ascendente

PrimaryKey	1
Campos:	idPersonal, Ascendente
TblCargoTblPersonal	1
Campos:	Cargo, Ascendente
tblSecciónTblPersonal	1
Campos:	idsección, Ascendente

Propiedades

ActivarOrden:	Falso	Def. actualizable:	Verdadero
Fecha de creación:	03/04/98 09:51:42 a.m.	RecordCount:	13
Última actualización:	15/11/98 05:40:09 p.m.		

Columnas

Nombre	Tipo	Tamaño
Idseccion	Texto	10
Sección	Texto	40
Ubicación	Texto	50
Actividad	Memo	-
Funciones	Memo	-
Grafico	Objeto OLE	-
Visita	Objeto OLE	-
Sonido	Objeto OLE	-
Iddepartamento	Texto	10

Relaciones

tblDepartamentotblSección

TblDepartamento	TblSección
Iddepartamento	Iddepartamento

Atributos: Exigir
 Atributos: Uno a varios

tblSeccióntblPersonal

TblSección	TblPersonal
Idseccion	Idsección

Atributos: Exigir
 Atributos: Uno a varios

Índices de tabla

Nombre	Número de campos
Iddepartamento	1

Campos:	Iddepartamento, Ascendente
Idseccion	1
Campos:	Idseccion, Ascendente
PrimaryKey	1
Campos:	Idseccion, Ascendente
tblDepartamentotblSección	1
Campos:	Iddepartamento, Ascendente

Propiedades

ActivarOrden:	Falso	Def. actualizable:	Verdadero
Fecha de creación:	15/09/98 05:03:17 p.m.	RecordCount:	11
Última actualización:	15/11/98 05:40:40 p.m.		

Columnas

Nombre	Tipo	Tamaño
IdTrámite	Texto	6
Trámite	Texto	50
Procedimiento	Memo	-

Indices de tabla

Nombre	Número de campos
IdTrámite	1
Campos:	IdTrámite, Ascendente
PrimaryKey	1
Campos:	IdTrámite, Ascendente

Propiedades

ActivarOrden:	Falso	Def. actualizable:	Verdadero
Fecha de creación:	21/10/98 05:48:34 p.m.	RecordCount:	75
Última actualización:	21/10/98 06:06:41 p.m.		

Columnas

Nombre	Tipo	Tamaño
Numero	Número (largo)	4
FechaV	Fecha/Hora	8
HoraV	Fecha/Hora	8

Índices de tabla

Nombre	Número de campos
PrimaryKey	1
Campos:	Numero, Ascendente

Columnas

Nombre	Tipo	Tamaño
ElMes	Número (entero)	2
LaCuenta	Número (largo)	4
4	Número (largo)	4
10	Número (largo)	4
11	Número (largo)	4

Objetos

Sección: Detalle

Sección: EncabezadoDelFormulario

Sección: PieDelFormulario

Botón de comando: Btnborrar

Botón de comando: Btnbuscar

Botón de comando: Btnnuevo

Botón de comando: Btnprimero

Botón de comando: Btnsalir

Botón de comando: Btnsiguiente

Botón de comando: Btnultimo

Botón de comando: cmdgrabacargo

Botón de comando: Comando13

Cuadro de texto: Descripción

Etiqueta: Etiqueta17

Etiqueta: Etiqueta4

Cuadro de texto: IdCargo

Etiqueta: IdCargo Etiqueta

Cuadro de texto: Nombre

Etiqueta: Nombre Etiqueta

Objetos

Sección: Detalle

Sección: EncabezadoDelFormulario

Sección: PieDelFormulario

Cuadro de texto: Antecedentes

Botón de comando: cmdanteriordep

Botón de comando: Cmdborradepartamento

Botón de comando: cmdbuscadep

Botón de comando: cmdgrabadep

Botón de comando: CmdNuevoDepartamento

Botón de comando: cmdprimerodep

Botón de comando: CmdSalirDepartamento

Botón de comando: cmdsiguientedep

Botón de comando: cmdultimodep

Botón de comando: Comando8

Etiqueta: Etiqueta36

Etiqueta: Etiqueta4

Cuadro de texto: Iddepartamento

Etiqueta: Iddepartamento Etiqueta

Marco de objeto dependiente: Logo

Etiqueta: Logo Etiqueta

Cuadro de texto: Nombre

Etiqueta: Nombre Etiqueta

Cuadro de texto: Ubicación

Etiqueta: Ubicación Etiqueta

Objetos

Sección: Detalle

Sección: EncabezadoDelFormulario

Sección: PieDelFormulario

Botón de comando: cmdanterioremp

Botón de comando: CmdBorraemp

Botón de comando: cmdbuscaemp

Botón de comando: cmdgrabaemp

Botón de comando: cmdnuevoemp

Botón de comando: cmdprimeroemp

Botón de comando: CmdSaliremp

Botón de comando: cmdsiguienteemp

Botón de comando: cmdultimoemp

Botón de comando: Comando10

Botón de comando: Comando11

Etiqueta: Etiqueta14

Etiqueta: Etiqueta35

Etiqueta: Etiqueta4

Marco de objeto dependiente: Fondo

Etiqueta: Fondo Etiqueta

Cuadro de texto: Historia

Cuadro de texto: IdEmpresa

Etiqueta: IdEmpresa Etiqueta

Marco de objeto dependiente: Logo

Etiqueta: Logo Etiqueta

Cuadro de texto: Nombre

Etiqueta: Nombre Etiqueta

Objetos

Sección: Detalle

Sección: EncabezadoDelFormulario

Sección: PieDelFormulario

Cuadro de texto: Apellido

Etiqueta: Apellido Etiqueta

Etiqueta: CmbCargo

Cuadro combinado: CmbCargoValor

Botón de comando: cmdanteriorper

Botón de comando: cmdborraper

Botón de comando: cmdbuscaper

Botón de comando: cmdgrabaper

Botón de comando: CmdImprimePersonal

Botón de comando: cmdprimeroper

Botón de comando: CmdSalirPersonal

Botón de comando: cmdsiguienteper

Botón de comando: cmdultimoper

Botón de comando: Comando24

Botón de comando: Comando40

Cuadro combinado: Cuadro combinado49

Etiqueta: Departamento

Etiqueta: Etiqueta4

Marco de objeto dependiente: Foto

Etiqueta: Foto Etiqueta

Cuadro de texto: Iddepartamento

Cuadro de texto: IdPersonal

Etiqueta: IdPersonal Etiqueta

Etiqueta: SecciónEtiqueta_
Cuadro de texto: TblPersonal.Nombre
Etiqueta: TblPersonal.Nombre Etiqueta

Objetos

Sección: Detalle

Botón de comando: Comando0

Etiqueta: Etiqueta2

Etiqueta: Etiqueta4

Cuadro de texto: FechaF

Cuadro de texto: FInicio

Objetos

Sección: Detalle

Sección: EncabezadoDelFormulario

Sección: PieDelFormulario

Cuadro de texto: Actividad

Etiqueta: Actividad Etiqueta

Cuadro combinado: CmbDepar

Botón de comando: cmdanterior

Botón de comando: cmdborrar

Botón de comando: cmdbuscar

Botón de comando: cmdgrabasec

Botón de comando: cmdnuevo

Botón de comando: cmdprimero

Botón de comando: CmdSalir

Botón de comando: cmdsiguiente

Botón de comando: cmdultimo

Botón de comando: Comando26

Botón de comando: Comando27

Botón de comando: Comando28

Etiqueta: DepartamentoEti

Etiqueta: Etiqueta38

Etiqueta: Etiqueta39

Etiqueta: Etiqueta4

Marco de objeto dependiente: Grafico

Etiqueta: Grafico Etiqueta

Cuadro de texto: Idseccion

Etiqueta: Idseccion Etiqueta

Cuadro de texto: Sección
Etiqueta: Sección Etiqueta
Marco de objeto dependiente: Sonido
Cuadro de texto: Tramites
Etiqueta: Tramites Etiqueta
Cuadro de texto: Ubicación
Etiqueta: Ubicación Etiqueta
Marco de objeto dependiente: Visita

Objetos

Sección: Detalle

Sección: EncabezadoDelFormulario

Sección: PieDelFormulario

Cuadro de texto: Caso

Etiqueta: Caso Etiqueta

Botón de comando: cmdanteriorotra

Botón de comando: cmdborratra

Botón de comando: cmdbuscatra

Botón de comando: cmdgrabatra

Botón de comando: CmdNuevoTramite

Botón de comando: cmdprimerotra

Botón de comando: CmdSalirTramite

Botón de comando: cmdsiguientetra

Botón de comando: cmdultimotra

Etiqueta: Etiqueta4

Cuadro de texto: IdTrámite

Etiqueta: IdTrámite Etiqueta

Cuadro de texto: Trámite

Etiqueta: Trámite Etiqueta

Objetos

Sección: Detalle

Botón de comando: CmdSalirCasos

Botón de comando: Comando2

Etiqueta: Etiqueta1

Cuadro de texto: Texto0

Objetos

Sección: Detalle

Sección: EncabezadoDelFormulario

Sección: PieDelFormulario

Etiqueta: Etiqueta10

Etiqueta: Etiqueta15

Etiqueta: Etiqueta8

Etiqueta: Etiqueta9

Cuadro de texto: Texto14

Cuadro de texto: Texto4

Cuadro de texto: Texto6

Cuadro de texto: ValMax

Cuadro de texto: ValMin

Objetos

Nivel de grupo 0

AgruparEn:	Cada valor	GroupFooter:	Falso
GroupHeader:	Falso	IntervaloDelGrupo:	1
Mantener juntos:	No	Orden:	Falso
Origen del control:	Nombre		

Sección: Detalle

Sección: EncabezadoDelInforme

Sección: EncabezadoDePágina

Sección: PieDelInforme

Sección: PieDePágina

Etiqueta: Etiqueta4

Etiqueta: Etiqueta9

Cuadro de texto: IdCargo

Etiqueta: IdCargo Etiqueta

Línea: Línea7

Línea: Línea8

Cuadro de texto: Nombre

Etiqueta: Nombre Etiqueta

Cuadro de texto: Texto13

Cuadro de texto: Texto5

Cuadro de texto: Texto6

Objetos

Nivel de grupo 0

AgruparEn:	Cada valor	GroupFooter:	Falso
GroupHeader:	Falso	IntervaloDelGrupo:	1
Mantener juntos:	No	Orden:	Falso
Origen del control:	Nombre		

Sección: Detalle

Sección: EncabezadoDelInforme

Sección: EncabezadoDePágina

Sección: PieDelInforme

Sección: PieDePágina

Etiqueta: Etiqueta4

Etiqueta: Etiqueta9

Cuadro de texto: Iddepartamento

Etiqueta: Iddepartamento Etiqueta

Línea: Línea10

Línea: Línea9

Cuadro de texto: Nombre

Etiqueta: Nombre Etiqueta

Cuadro de texto: Texto13

Cuadro de texto: Texto7

Cuadro de texto: Texto8

Cuadro de texto: Ubicación

Etiqueta: Ubicación Etiqueta

Objetos

Sección: Detalle

Sección: EncabezadoDePágina

Sección: PieDePágina

Etiqueta: Etiqueta9

Marco de objeto independiente: OLEIndependiente0

Marco de objeto independiente: OLEIndependiente2

Cuadro de texto: Texto13

Objetos

Sección: Detalle

Sección: EncabezadoDelInforme

Sección: EncabezadoDePágina

Sección: PieDelInforme

Sección: PieDePágina

Cuadro de texto: Apellido

Etiqueta: Apellido Etiqueta

Etiqueta: Etiqueta4

Etiqueta: Etiqueta9

Marco de objeto dependiente: Foto

Etiqueta: Foto Etiqueta

Cuadro de texto: idPersonal

Etiqueta: idPersonal Etiqueta

Línea: Línea21

Línea: Línea22

Cuadro de texto: Sección

Etiqueta: Sección Etiqueta

Cuadro de texto: TblCargo.Nombre

Etiqueta: TblCargo.Nombre Etiqueta

Cuadro de texto: TblDepartamento.Nombre

Etiqueta: TblDepartamento.Nombre Etiqueta

Cuadro de texto: TblPersonal.Nombre

Etiqueta: TblPersonal.Nombre Etiqueta

Cuadro de texto: Texto13

Cuadro de texto: Texto19

Cuadro de texto: Texto20

Objetos

Nivel de grupo 0

AgruparEn:	Cada valor	GroupFooter:	Falso
GroupHeader:	Verdadero	IntervaloDelGrupo:	1
Mantener juntos:	No	Orden:	Falso
Origen del control:	TblDepartamento.Iddepartamento		

Nivel de grupo 1

AgruparEn:	Cada valor	GroupFooter:	Falso
GroupHeader:	Falso	IntervaloDelGrupo:	1
Mantener juntos:	No	Orden:	Falso
Origen del control:	Sección		

Sección: Detalle

Sección: EncabezadoDelGrupo0

Sección: EncabezadoDelInforme

Sección: EncabezadoDePágina

Sección: PieDelInforme

Sección: PieDePágina

Etiqueta: Etiqueta4

Etiqueta: Etiqueta9

Cuadro de texto: Idseccion

Etiqueta: Idseccion Etiqueta

Línea: Línea13

Línea: Línea14

Línea: Línea16

Línea: Línea17

Línea: Línea18

Línea: Línea19

Línea: Línea20

Línea: Línea21

Línea: Línea22

Cuadro de texto: Nombre

Etiqueta: Nombre Etiqueta

Cuadro de texto: Sección

Etiqueta: Sección Etiqueta

Cuadro de texto: TblSección.Iddepartamento

Cuadro de texto: Texto11

Cuadro de texto: Texto12

Cuadro de texto: Texto13

Cuadro de texto: Ubicación

Etiqueta: Ubicación Etiqueta

Objetos

Nivel de grupo 0

AgruparEn:	Cada valor	GroupFooter:	Falso
GroupHeader:	Verdadero	IntervaloDelGrupo:	1
Mantener juntos:	No	Orden:	Falso
Origen del control:	TblDepartamento.Iddepartamento		

Nivel de grupo 1

AgruparEn:	Cada valor	GroupFooter:	Verdadero
GroupHeader:	Verdadero	IntervaloDelGrupo:	1
Mantener juntos:	No	Orden:	Falso
Origen del control:	Idseccion		

Nivel de grupo 2

AgruparEn:	Cada valor	GroupFooter:	Falso
GroupHeader:	Falso	IntervaloDelGrupo:	1
Mantener juntos:	No	Orden:	Falso
Origen del control:	Apellido		

Nivel de grupo 3

AgruparEn:	Cada valor	GroupFooter:	Falso
GroupHeader:	Falso	IntervaloDelGrupo:	1
Mantener juntos:	No	Orden:	Falso
Origen del control:	TblPersonal.Nombre		

Sección: Detalle

Sección: EncabezadoDelGrupo0

Sección: EncabezadoDelGrupo1

Sección: EncabezadoDelInforme

Sección: EncabezadoDePágina

Sección: PieDelGrupo0

Sección: PieDelInforme

Sección: PieDePágina

Cuadro de texto: Apellido

Etiqueta: Apellido Etiqueta

Etiqueta: Etiqueta29

Etiqueta: Etiqueta31

Etiqueta: Etiqueta4

Etiqueta: Etiqueta9

Cuadro de texto: idsección

Línea: Línea19

Línea: Línea22

Línea: Línea23

Línea: Línea26

Línea: Línea27

Línea: Línea28

Cuadro de texto: Sección

Etiqueta: Sección Etiqueta

Cuadro de texto: TblCargo.Nombre

Etiqueta: TblCargo.Nombre Etiqueta

Cuadro de texto: TblDepartamento.Nombre

Etiqueta: TblDepartamento.Nombre Etiqueta

Cuadro de texto: TblPersonal.Nombre

Etiqueta: TblPersonal.Nombre Etiqueta

Cuadro de texto: TblSección.Iddepartamento

Etiqueta: TblSección.Iddepartamento Etiqueta

Cuadro de texto: Texto13

Cuadro de texto: Texto17

Cuadro de texto: Texto18

Cuadro de texto: Texto30

Objetos

Nivel de grupo 0

AgruparEn: Cada valor
GroupHeader: Falso
Mantener juntos: No
Origen del control: Trámite

GroupFooter: Falso
IntervaloDelGrupo: 1
Orden: Falso

Sección: Detalle

Sección: EncabezadoDelInforme

Sección: EncabezadoDePágina

Sección: PieDelInforme

Sección: PieDePágina

Etiqueta: Etiqueta4

Etiqueta: Etiqueta9

Cuadro de texto: IdTrámite

Etiqueta: IdTrámite Etiqueta

Línea: Línea11

Línea: Línea8

Cuadro de texto: Texto13

Cuadro de texto: Texto5

Cuadro de texto: Texto6

Cuadro de texto: Trámite

Etiqueta: Trámite Etiqueta

Objetos

Nivel de grupo 0

AgruparEn: Cada valor
GroupHeader: Falso
Mantener juntos: No
Origen del control: Trámite

GroupFooter: Falso
IntervaloDelGrupo: 1
Orden: Falso

Sección: Detalle

Sección: EncabezadoDelInforme

Sección: EncabezadoDePágina

Sección: PieDelInforme

Sección: PieDePágina

Etiqueta: Etiqueta4

Etiqueta: Etiqueta9

Cuadro de texto: IdTrámite

Etiqueta: IdTrámite Etiqueta

Línea: Línea11

Línea: Línea8

Cuadro de texto: Texto13

Cuadro de texto: Texto5

Cuadro de texto: Texto6

Cuadro de texto: Trámite

Etiqueta: Trámite Etiqueta

Acciones

Nombre	Condición	Acción	Argumento	Valor
		OpenForm	Nombre del formulario:	FrmDepartamento
			Vista:	Formulario
			Nombre del filtro:	
			Condición WHERE:	
			Modo de datos:	-1
			Modo de la ventana:	Normal

Acciones

Nombre	Condición	Acción	Argumento	Valor
Salir Administrador		Close	Tipo de objeto:	Formulario
			Nombre del objeto:	FinInicio
			Guardar:	Preguntar
Salir Access		Quit	Opciones:	Guardar todo

Acciones				
Nombre	Condición	Acción	Argumento	Valor
General		OpenReport	Nombre del informe:	InfEstadistica de Visitas
			Vista:	Vista preliminar
			Nombre de filtro:	
			Condición WHERE:	

Acciones

Nombre	Condición	Acción	Argumento	Valor
		OpenReport	Nombre del informe: Vista: Nombre de filtro: Condición WHERE:	na Vista preliminar [TbVisita]![FechaV] Between [Forms]![FrmRango de Fechas]![FInicio] And [Forms]![FrmRango de Fechas]![FechaF]

Acciones

Nombre	Condición	Acción	Argumento	Valor
		GoToControl	Nombre del control:	Fondo
		RunCommand	Comando:	33

Acciones

Nombre	Condición	Acción	Argumento	Valor
		GoToControl	Nombre del control:	Foto
		RunCommand	Comando:	33

Acciones

Nombre	Condición	Acción	Argumento	Valor
		GoToControl	Nombre del control:	Grafico
		RunCommand	Comando:	33

Acciones

Nombre	Condición	Acción	Argumento	Valor
Video		RunApp	Línea de comandos:	"C:\Archivos de programa\Zotrix\TVmax.exe"
Imagen		RunApp	Línea de comandos:	"C:\Graficos\Paint Shop Pro\Psp.exe"
Audio		RunApp	Línea de comandos:	"C:\Archivos de programa\Audio Applications\Rack\CMRACK.EXE"

Acciones

Nombre	Condición	Acción	Argumento	Valor
		GoToControl	Nombre del control:	logo
		RunCommand	Comando:	33

Acciones

Nombre	Condición	Acción	Argumento	Valor
		OpenReport	Nombre del informe:	InfPersona
			Vista:	Vista preliminar
			Nombre de filtro:	
			Condición WHERE:	[Forms]![FrmPersona]![idPersona]=[TblPersona]![idPersona]

Acciones

Nombre	Condición	Acción	Argumento	Valor
Cargos		OpenReport	Nombre del informe: Vista: Nombre de filtro: Condición WHERE:	InfCargos Vista preliminar
Departamentos		OpenReport	Nombre del informe: Vista: Nombre de filtro: Condición WHERE:	InfDepartamentos Vista preliminar
Secciones		OpenReport	Nombre del informe: Vista: Nombre de filtro: Condición WHERE:	InfSecciones Vista preliminar
Total Personal		OpenReport	Nombre del informe: Vista: Nombre de filtro: Condición WHERE:	InfTotal Personal Vista preliminar
Total Trámites		OpenReport	Nombre del informe: Vista: Nombre de filtro: Condición WHERE:	InfTrámites Vista preliminar
Buscar Trámites		OpenForm	Nombre del formulario: Vista: Nombre del filtro: Condición WHERE: Modo de datos: Modo de la ventana:	FrmTrámites Casos Formulario -1 Normal

Acciones

Nombre	Condición	Acción	Argumento	Valor
Empresa		OpenForm	Nombre del formulario:	FrmEmpresa
			Vista:	Formulario
			Nombre del filtro:	
			Condición WHERE:	
			Modo de datos:	-1
Cargo		OpenForm	Nombre del formulario:	FrmCargo
			Vista:	Formulario
			Nombre del filtro:	
			Condición WHERE:	
			Modo de datos:	-1
Departamento		OpenForm	Nombre del formulario:	FrmDepartamento
			Vista:	Formulario
			Nombre del filtro:	
			Condición WHERE:	
			Modo de datos:	-1
Sección		OpenForm	Nombre del formulario:	FrmSección
			Vista:	Formulario
			Nombre del filtro:	
			Condición WHERE:	
			Modo de datos:	-1
Personal		OpenForm	Nombre del formulario:	FrmPersonal
			Vista:	Formulario
			Nombre del filtro:	
			Condición WHERE:	
			Modo de datos:	-1
			Modo de la ventana:	Normal

Trámites

OpenForm	Nombre del formulario:	FrmTrámite
	Vista:	Formulario
	Nombre del filtro:	
	Condición WHERE:	
	Modo de datos:	-1
	Modo de la ventana:	Normal

Acciones

Nombre	Condición	Acción	Argumento	Valor
		AddMenu	Nombre del menú: Nombre de macro de menú: Texto de la barra de estado:	Listas mcrListaMenu
		AddMenu	Nombre del menú: Nombre de macro de menú: Texto de la barra de estado:	Informes McrInformeMenu
		AddMenu	Nombre del menú: Nombre de macro de menú: Texto de la barra de estado:	Estadísticas McrEstadistica
		AddMenu	Nombre del menú: Nombre de macro de menú: Texto de la barra de estado:	Herramientas Mcrherramientas
		AddMenu	Nombre del menú: Nombre de macro de menú: Texto de la barra de estado:	Acerca de.. McrAcerca

```

ivate Sub BDepartamento_Click()
n Forma
n Inicio
departamento.Caption = mtitulo
departamento.Height = malto: Fdepartamento.Width = mancho
n Empresa
departamento.ETitulo = mtituloE: Fdepartamento.ETitulo.Width = manchop
departamento.ETitulo.Height = maltoe: Fdepartamento.ETitulo.Top = Mtope
departamento.ETitulo.Left = mizqe
n Publicidad
departamento.EPublicidad.Caption = mpublicidad: Fdepartamento.EPublicidad.Width = manchop
departamento.EPublicidad.Height = maltop: Fdepartamento.EPublicidad.Top = Mtopep
departamento.EPublicidad.Left = mizqp
n Logo
departamento.OLElogo.dataField = FInicio.OLElogo.DataField
departamento.OLElogo.Data = FInicio.OLElogo.Data
Set Fdepartamento.OLElogo.Container = FInicio.OLElogo.Container
nidoBotones
departamento.Show
End Sub

ivate Sub BPersonal_Click(Index As Integer)
n Forma
n Inicio
personal.Caption = mtitulo
personal.Height = malto: FPersonal.Width = mancho
nidoBotones
personal.Show
End Sub

ivate Sub BSalir_Click(Index As Integer)
presa.Recordset.Close
oad FInicio
End Sub

ivate Sub BSecciones_Click(Index As Integer)
nidoBotones
ccion.Show
End Sub

ivate Sub BTramites_Click(Index As Integer)
n Forma
n Inicio
tramites.Caption = mtitulo
tramites.Height = malto: FTramites.Width = mancho
nidoBotones
tramites.Show
End Sub

ivate Sub Form_Initialize()
n RetVal
RetVal = Shell("C:\GVirtual\Imágenes\ojos.exe", 1)
nba
End Sub

o Graba()
x As Integer
dbsVisita As Database
Dim rstVisita As Recordset

Set dbsVisita = OpenDatabase("c:\Gvirtual\Virtual\Virtual.mdb")
Set rstVisita = dbsVisita.OpenRecordset("TblVisita", dbOpenDynaset)
rstVisita.MoveLast
x = rstVisita.Fields(0) + 1
AgregarNombre rstVisita, Str(x), Str(Date), Str(Time)
rstVisita.Close
dbsVisita.Close

```

```
Sub
Function AgregarNombre(rstTemp As Recordset, SNum As String, Sfecha As String, SHora As String)
With rstTemp
    .AddNew
    !Numero = SNum
    !FechaV = Sfecha
    !HoraV = SHora
    .Update
    .Bookmark = .LastModified
End With
Function
```

```

        Italic           - 0 'False
        Strikethrough    - 0 'False
    EndProperty
    Height              - 1400
    Left                - 240
    MouseIcon           - (Icon)
    MousePointer        - 99 'Custom
    Picture              - (Bitmap)
    Style                - 1 'Graphical
    TabIndex             - 5
    Top                  - 2760
    Width                - 3015
End
Begin VB.TextBox Ehistoria
    DataField            "Historia"
    DataSource            "Tempresa"
    BeginProperty Font
        Name                - "MS Sans Serif"
        Size                 - 13.5
        Charset              - 0
    EndProperty
End

```

```

ON 5.00
= "C:\A8AF28-1257-101B-8F80-0020AF039CA3\#1.1#0"; "MCI32.OCX"
VB.Form Finicio
OrderStyle = 3 'Fixed Dialog
Caption = "Guia Virtual"
ClientHeight = 11085
ClientLeft = 1170
ClientTop = 1230
ClientWidth = 15225
DefaultTopic = "Form1"
Enabled = 0 'False
Enabled = 0 'False
Enabled = 0 'False
ClientHeight = 11085
ClientWidth = 15225
ShowInTaskbar = 0 'False
StartUpPosition = 2 'CenterScreen
Begin MCI.MMControl MMControl1
Height = 735
Left = 840
TabIndex = 11
Top = 10080
Visible = 0 'False
Width = 3540
ExtentX = 6244
ExtentY = 1296
Version = 327680
DeviceType = ""
FileName = "C:\GVirtual\Sonidos\Arpequio.wav"
End
Begin VB.CommandButton BSalir
BackColor = &H00C0C000&
Caption = "Abandonar"
DownPicture = (Icon)
BeginProperty Font
Name = "MS Sans Serif"
Size = 18
Charset = 0
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 1400
Index = 0
Left = 240
MouseIcon = (Icon)
MousePointer = 99 'Custom
Picture = (Icon)
Style = 1 'Graphical
TabIndex = 9
Top = 8040
Width = 3015
End
Begin VB.CommandButton BTramites
BackColor = &H00C0C000&
Caption = "Trámites"
DownPicture = (Icon)
BeginProperty Font
Name = "MS Sans Serif"
Size = 18
Charset = 0
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 1400

```

```

Left = 240
MouseIcon = (Icon)
MousePointer = 99 'Custom
Picture = (Bitmap)
Style = 1 'Graphical
TabIndex = 8
Top = 6720
Width = 3015

```

End

Begin VB.CommandButton

BPersonal

BackColor = &H00C0C000&

Caption = "Personal"

DownPicture = (Icon)

BeginProperty Font

Name = "MS Sans Serif"

Size = 18

Charset = 0

Weight = 400

Underline = 0 'False

Italic = 0 'False

Strikethrough = 0 'False

EndProperty

Height = 1400

Index = 1

Left = 240

MouseIcon = (Icon)

MousePointer = 99 'Custom

Picture = (Bitmap)

Style = 1 'Graphical

TabIndex = 7

Top = 5400

Width = 3015

End

Begin VB.CommandButton

BSecciones

BackColor = &H00C0C000&

Caption = "Secciones"

DownPicture = (Icon)

BeginProperty Font

Name = "MS Sans Serif"

Size = 18

Charset = 0

Weight = 400

Underline = 0 'False

Italic = 0 'False

Strikethrough = 0 'False

EndProperty

Height = 1400

Index = 0

Left = 240

MouseIcon = (Icon)

MousePointer = 99 'Custom

Picture = (Bitmap)

Style = 1 'Graphical

TabIndex = 6

Top = 4080

Width = 3015

End

Begin VB.CommandButton

BDepartamento

BackColor = &H00C0C000&

Caption = "Departamentos"

DownPicture = (Icon)

BeginProperty Font

Name = "MS Sans Serif"

Size = 18

Charset = 0

Weight = 400

Underline = 0 'False

```

EndProperty
Height = 1400
Left = 240
MouseIcon = (Icon)
MousePointer = 99 'Custom
Picture = (Bitmap)
Style = 1 'Graphical
TabIndex = 5
Top = 2760
Width = 3015
d
gin VB.TextBox Ehistoria
DataField = "Historia"
DataSource = "Tempresa"
BeginProperty Font
    Name = "MS Sans Serif"
    Size = 13.5
    Charset = 0
    Weight = 400
    Underline = 0 'False
    Italic = 0 'False
    Strikethrough = 0 'False
EndProperty
Height = 6135
Left = 3960
Locked = -1 'True
Multiline = -1 'True
ScrollBars = 2 'Vertical
TabIndex = 3
Top = 3240
Width = 11175
d
gin VB.Data Tempresa
Connect = "Access"
DatabaseName = "C:\GVirtual\Virtual\Virtual.mdb"
DefaultCursorType = 0 'DefaultCursor
DefaultType = 2 'UseODBC
Exclusive = 0 'False
Height = 345
Left = 7560
Options = 0
ReadOnly = -1 'True
RecordsetType = 1 'Dynaset
RecordSource = "TblEmpresa"
Top = 4920
Visible = 0 'False
Width = 1380
d
gin VB.Label Label1
BackColor = &H80000000A&
BorderStyle = 1 'Fixed Single
Caption = "HISTORIA"
BeginProperty Font
    Name = "MS Sans Serif"
    Size = 18
    Charset = 0
    Weight = 700
    Underline = 0 'False
    Italic = 0 'False
    Strikethrough = 0 'False
EndProperty
Height = 615
Left = 3960
TabIndex = 10
Top = 2640
Width = 2055

```

```

Alignment = 2 'Center'
BackColor = &H00F0E0F0&
BackStyle = 0 'Transparent'
DataField = "Publicidad"
DataSource = "Tempresa"

```

```
BeginProperty Font
```

```

Name = "MS Sans Serif"
Size = 18
Charset = 0
Weight = 400
Underline = 0 'False'
Italic = -1 'True'
StrikeThrough = 0 'False'

```

```
EndProperty
```

```
Height = 615
```

```
Left = 0
```

```
TabIndex = 4
```

```
Top = 1320
```

```
Width = 12135
```

```
End
```

```
Begin VB.OLE OLELogo
```

```

Appearance = 0 'Flat'
BackStyle = 0 'Transparent'
BorderStyle = 0 'None'
DataField = "Logo"
DataSource = "Tempresa"

```

```
Height = 2895
```

```
Left = 12120
```

```
SizeMode = 1 'Stretch'
```

```
TabIndex = 2
```

```
Top = 0
```

```
Width = 3015
```

```
End
```

```
Begin VB.Label LNombreEmpresa
```

```

Alignment = 2 'Center'
BackColor = &H00C0C0C0&
BackStyle = 0 'Transparent'
DataField = "Nombre"
DataSource = "Tempresa"

```

```
EndProperty Font
```

```
Begin VB.Label LNombreEmpresa
```

```

Alignment = 2 'Center'
BackColor = &H00C0C0C0&
BackStyle = 0 'Transparent'
DataField = "Nombre"
DataSource = "Tempresa"

```

```
EndProperty Font
```

```
Begin VB.Label LNombreEmpresa
```

```

Alignment = 2 'Center'
BackColor = &H00C0C0C0&
BackStyle = 0 'Transparent'
DataField = "Nombre"
DataSource = "Tempresa"

```

```
EndProperty Font
```

```
Begin VB.Label LNombreEmpresa
```

```

Alignment = 2 'Center'
BackColor = &H00C0C0C0&
BackStyle = 0 'Transparent'
DataField = "Nombre"
DataSource = "Tempresa"

```

```
EndProperty Font
```

```
Begin VB.Label LNombreEmpresa
```

```

Alignment = 2 'Center'
BackColor = &H00C0C0C0&
BackStyle = 0 'Transparent'
DataField = "Nombre"
DataSource = "Tempresa"

```

```
EndProperty Font
```

```
Begin VB.Label LNombreEmpresa
```

```

Alignment = 2 'Center'
BackColor = &H00C0C0C0&
BackStyle = 0 'Transparent'
DataField = "Nombre"
DataSource = "Tempresa"

```

```
EndProperty Font
```

```
Begin VB.Label LNombreEmpresa
```

```

Alignment = 2 'Center'
BackColor = &H00C0C0C0&
BackStyle = 0 'Transparent'
DataField = "Nombre"
DataSource = "Tempresa"

```

```
EndProperty Font
```

```
Begin VB.Label LNombreEmpresa
```

```

Alignment = 2 'Center'
BackColor = &H00C0C0C0&
BackStyle = 0 'Transparent'
DataField = "Nombre"
DataSource = "Tempresa"

```

```
EndProperty Font
```

```
Begin VB.Label LNombreEmpresa
```

```

Alignment = 2 'Center'
BackColor = &H00C0C0C0&
BackStyle = 0 'Transparent'
DataField = "Nombre"
DataSource = "Tempresa"

```

```
EndProperty Font
```

```
Begin VB.Label LNombreEmpresa
```

```

Alignment = 2 'Center'
BackColor = &H00C0C0C0&
BackStyle = 0 'Transparent'
DataField = "Nombre"
DataSource = "Tempresa"

```

```
EndProperty Font
```

```
Begin VB.Label LNombreEmpresa
```

```

Alignment = 2 'Center'
BackColor = &H00C0C0C0&
BackStyle = 0 'Transparent'
DataField = "Nombre"
DataSource = "Tempresa"

```

```
EndProperty Font
```

```
Begin VB.Label LNombreEmpresa
```

```

Alignment = 2 'Center'
BackColor = &H00C0C0C0&
BackStyle = 0 'Transparent'
DataField = "Nombre"
DataSource = "Tempresa"

```

```
End
```

```

ate Sub BSatir_Click()
rtamento.Recordset.Close
d Edepartamento
ub

ate Sub BSecciones_Click(Index As Integer)
mpresa
ionSub.Caption = mtitulo
ionSub.Height = malto: EseccionSub.Width = mancho
ionSub.Show
ub

ate Sub Form_Activate()
regi As Integer
posi As Variant
= 0
rtamento.Recordset.MoveFirst
til Tdepartamento.Recordset.EOF
mposi = Tdepartamento.Recordset.Bookmark
departamento.AddItem Eldepar.Text + " " + ENombre.Caption
Tdepartamento.Recordset.Bookmark = mposi
departamento.Recordset.MoveNext
regi = mregi + 1

rtamento.Refresh
re.Caption = Ldepartamento.List(0)

Sub

ate Sub Form_Deactivate()
rtamento.Clear
Sub

ate Sub Ldepartamento_Click()
re.Caption = Ldepartamento.List(Ldepartamento.ListIndex)
par.Text = Left$(ENombre.Caption, 10)
rtamento.Recordset.Index = "Iddepartamento"
rtamento.Recordset.Seek "=", Eldepar.Text
departamento.Recordset.NoMatch Then
Box "No localiza el Registro"
If

Sub

```

```

Private Sub Bdepartamento_Click()
    Dim Forma
    Inicio
    departamento.Caption = mtitulo
    departamento.Height = malto: Fdepartamento.Width = mancho
    Dim Empresa
    departamento.ETitulo = mtituloE: Fdepartamento.ETitulo.Width = manchoE
    departamento.ETitulo.Height = maltoE: Fdepartamento.ETitulo.Top = Mtope
    departamento.ETitulo.Left = mizqE
    Dim Publicidad
    departamento.EPublicidad.Caption = mpublicidad: Fdepartamento.EPublicidad.Width = manchoP
    departamento.EPublicidad.Height = maltoP: Fdepartamento.EPublicidad.Top = MtopeP
    departamento.EPublicidad.Left = mizqP
    Dim Logo
    Fdepartamento.OLElogo.DataField = FInicio.OLElogo.DataField
    Fdepartamento.OLElogo.Data = FInicio.OLElogo.Data
    Set Fdepartamento.OLElogo.Container = FInicio.OLElogo.Container
    OnidoBotones
    departamento.Show
End Sub

Private Sub BPersonal_Click(Index As Integer)
    Dim Forma
    Inicio
    Personal.Caption = mtitulo
    Personal.Height = malto: FPersonal.Width = mancho
    OnidoBotones
    Personal.Show
End Sub

Private Sub BSalar_Click(Index As Integer)
    empresa.Recordset.Close
    Load FInicio
End Sub

Private Sub BSecciones_Click(Index As Integer)
    OnidoBotones
    seccion.Show
End Sub

Private Sub BTramites_Click(Index As Integer)
    Dim Forma
    Inicio
    Tramites.Caption = mtitulo
    Tramites.Height = malto: FTramites.Width = mancho
    OnidoBotones
    Tramites.Show
End Sub

Private Sub Form_Initialize()
    Dim RetVal
    RetVal = Shell("C:\GVirtual\Imágenes\ojos.exe", 1)
    Graba
End Sub

Sub Graba()
    Dim x As Integer
    Dim dbsVisita As Database
    Dim rstVisita As Recordset

    Set dbsVisita = OpenDatabase("c:\Gvirtual\Virtual\Virtual.mdb")
    Set rstVisita = dbsVisita.OpenRecordset("TblVisita", dbOpenDynaset)
    rstVisita.MoveLast
    x = rstVisita.Fields(0) + 1
    AgregarNombre rstVisita, Str(x), Str(Date), Str(Time)
    rstVisita.Close
    dbsVisita.Close

```

```
Sub
Function AgregarNombre(rstTemp As Recordset, SNum As String, SFecha As String, SHora As String)
With rstTemp
    .AddNew
    !Numero = SNum
    !FechaV = SFecha
    !HoraV = SHora
    .Update
    .Bookmark = .LastModified
End With
Function
```

```

Private Sub BPrograma_Click()
    Dim MasSeccion As Integer
    Dim Sub

Private Sub Form_Activate()
    IdSeccion0.Caption = FSeccion.T1bSec.Text
    blSeccion.Recordset.Index = "IdSeccion"
    blSeccion.Recordset.Seek "=", IdSeccion0.Caption
    If blSeccion.Recordset.NoMatch Then
        MsgBox "No localiza el Seccion"
    End If
End Sub

```

```

ate Sub BSalir_Click()
    Personal.Recordset.Close
    Seccion.Recordset.Close
    Cargo.Recordset.Close
    Departamento.Recordset.Close
    adFPersonal
Sub

ate Sub Form_Activate()
    mregi As Integer
    i = 0
    Personal.Recordset.MoveFirst
    Until TPersonal.Recordset.EOF
    Personal.AddItem TCodigoP.Text + " " + TApellido.Text + " " + TNombre.Text
    Personal.Recordset.MoveNext
    mregi = mregi + 1
    Personal.Refresh
    Nombre.Caption = LDepartamento.List(0)
    Busca
Sub

ate Sub Form_Deactivate()
    Personal.Clear
Sub

ate Sub LPersonal_Click()
    Busca
Sub

Busca()
    Nombre.Text = LPersonal.List(LPersonal.List.Index)
    CodigoP.Text = Left$(TNombre.Text, 10)
    Personal.Recordset.Index = "IdPersonal"
    Personal.Recordset.Seek "=", TCodigoP.Text
    If Personal.Recordset.NoMatch Then
        MsgBox "No localiza el Personal"
    End If
    Seccion
    Seccion.Recordset.Index = "IdSeccion"
    Seccion.Recordset.Seek "=", TSeccion.Text
    If Seccion.Recordset.NoMatch Then
        MsgBox "No localiza la Seccion"
    End If
    Seccion
    Departamento.Recordset.Index = "Iddepartamento"
    Departamento.Recordset.Seek "=", TIdepar.Text
    If Departamento.Recordset.NoMatch Then
        MsgBox "No localiza la Departamento"
    End If
    Cargo
    Cargo.Recordset.Index = "IdCargo"
    Cargo.Recordset.Seek "=", TCargo.Text
    If Cargo.Recordset.NoMatch Then
        MsgBox "No localiza el Cargo"
    End If
Sub

```

```

Private Sub BMas_Click()
Finicio
FMasSeccion.Caption = mtitulo
FMasSeccion.Height = mallo: FMasSeccion.Width = mancho
If (LSeccion.ListIndex <> -1) Then
    FMasSeccion.Show
Else
    MsgBox "No selecciona una Sección"
End If
End Sub

Private Sub BSalir_Click()
TDepartamento.Recordset.Close
TSeccion.Recordset.Close
Unload Fseccion
End Sub

Private Sub Form_Activate()
Dim mregi As Integer
Rem TDepartamento.Recordset.Index = "Iddepartamento"
mregi = 0
TDepartamento.Recordset.MoveFirst
Do Until TDepartamento.Recordset.EOF
    LDepartamento.AddItem Eldepar.Text + " " + ENombre.Caption
    TDepartamento.Recordset.MoveNext
    mregi = mregi + 1
Loop
Eldepar.Text = Left$(LDepartamento.List(0), 10)
ESeccion.Caption = "": LIdbiSec.Caption = ""
BuscaDepa
End Sub

Sub BuscaDepa()
TDepartamento.Recordset.Index = "Iddepartamento"
TDepartamento.Recordset.Seek "=", Eldepar.Text
If TDepartamento.Recordset.NoMatch Then
    MsgBox "No localiza el Departamento"
End If
End Sub

Private Sub Form_Deactivate()
LDepartamento.Clear
End Sub

Private Sub LDepartamento_Click()
Dim mregiS As Integer
ENombre.Caption = Mid$(LDepartamento.List(LDepartamento.ListIndex), 15)
Eldepar.Text = Left$(LDepartamento.List(LDepartamento.ListIndex), 10)
BuscaDepa
LSeccion.Clear
mregiS = 0
TSeccion.Recordset.MoveFirst
Do Until TSeccion.Recordset.EOF
    If TIdSeccion.Text = Eldepar.Text Then
        LSeccion.AddItem TIDSec.Text + " " + ESeccion.Caption
        mregiS = mregiS + 1
    End If
    TSeccion.Recordset.MoveNext
Loop
End Sub

Private Sub LSeccion_Click()
ESeccion.Caption = Mid$(LSeccion.List(LSeccion.ListIndex), 15)
TIDSec.Text = Left$(LSeccion.List(LSeccion.ListIndex), 10)
TSeccion.Recordset.Index = "IdSeccion"
TSeccion.Recordset.Seek " ", TIDSec.Text
If TSeccion.Recordset.NoMatch Then
    MsgBox "No localiza el Seccion"
End If

```



```

TDepartamento.Recordset.Close
TSeccion.Recordset.Close
Unload FSeccionSub
End Sub

```

```

Private Sub Form_Activate()
Dim mregi As Integer
Rem TDepartamento.Recordset.Index = "Iddepartamento"
mregi = 0
TDepartamento.Recordset.MoveFirst
Do Until TDepartamento.Recordset.EOF
    LDepartamento.AddItem EIdepar.Text + " " + ENombre.Caption
    TDepartamento.Recordset.MoveNext
    mregi = mregi + 1
Loop
EIdepar.Text = Left$(LDepartamento.List(0), 10)
ESeccion.Caption = " " & LIdbSec.Caption
BuscaDepa
End Sub
Sub Buscalpapa()
TDepartamento.Recordset.Index = "Iddepartamento"
TDepartamento.Recordset.Seek "=", EIdepar.Text
If TDepartamento.Recordset.NoMatch Then
    MsgBox "No localiza el Departamento"
End If
End Sub

```

```

Private Sub lDepartamento_Click()
Dim mregis As Integer
ENombre.Caption = Mid$(LDepartamento.List(LDepartamento.ListIndex), 15)
EIdepar.Text = Left$(Ldepartamento.List(Ldepartamento.ListIndex), 10)
BuscaDepa
LSeccion.Clear
mregis = 0
TSeccion.Recordset.MoveFirst
Do Until TSeccion.Recordset.EOF
    If TIdSeccion.Text = EIdepar.Text Then
        LSeccion.AddItem TIdSec.Text + " " + ESeccion.Caption
        mregis = mregis + 1
    End If
    TSeccion.Recordset.MoveNext
Loop
End Sub

```

```

Private Sub lSeccion_Click()
ESeccion.Caption = Mid$(LSeccion.List(LSeccion.ListIndex), 15)
TIdSec.Text = Left$(LSeccion.List(LSeccion.ListIndex), 10)
TSeccion.Recordset.Index = "IdSeccion"
TSeccion.Recordset.Seek "=", TIdSec.Text
If TSeccion.Recordset.NoMatch Then
    MsgBox "No localiza el Seccion"
End If

```

```

End Sub

```

```

Private Sub BSalir_Click()
    TTramites.Recordset.Close
    Unload FTramites
End Sub

Private Sub Form_Activate()
    Dim mregi As Integer
    mregi = 0
    TTramites.Recordset.MoveFirst
    Do Until TTramites.Recordset.EOF
        LTramites.AddItem EIdtramite.Text + "    " + ENombre.Caption
        TTramites.Recordset.MoveNext
        mregi = mregi + 1
    Loop
    TTramites.Refresh
    ENombre.Caption = LTramites.List(0)
End Sub

Private Sub Form_Deactivate()
    LTramites.Clear
End Sub

Private Sub Ltramites_Click()
    ENombre.Caption = LTramites.List(LTramites.ListIndex)
    EIdtramite.Text = Left$(ENombre.Caption, 10)
    TTramites.Recordset.Index = "IdTrámite"
    TTramites.Recordset.Seek "-", EIdtramite.Text
    If TTramites.Recordset.NoMatch Then
        MsgBox "No localiza el Trámite"
    End If
End Sub

```

Global mtitulo, mtituloE, mpublicidad As String
Global mancho, malto, Mtope, maltoe, manchoe, mizqE As Integer
Global Mtopep, maltop, manchop, mizqp, mtiempo As Integer
Global mlogo As Database

```
Sub Pinicio()  
Rem Forma  
mancho = FInicio.Width: malto = FInicio.Height  
mtitulo = FInicio.Caption  
Rem Empresa  
mtituloE = FInicio.LNombreEmpresa.Caption  
maltoe = FInicio.LNombreEmpresa.Height: Mtope = FInicio.LNombreEmpresa.Top  
manchoe = FInicio.LNombreEmpresa.Width: mizqE = FInicio.LNombreEmpresa.Left  
Rem Publicidad  
mpublicidad = FInicio.LPublicidad.Caption  
maltop = FInicio.LPublicidad.Height: Mtopep = FInicio.LPublicidad.Top  
manchop = FInicio.LPublicidad.Width: mizqp = FInicio.LPublicidad.Left  
Rem Logo  
mfile = 1  
mtiempo = 5  
End Sub  
Sub SonidoBotones()  
FInicio.MMControll.Command = "Close"  
FInicio.MMControll.Command = "Open"  
FInicio.MMControll.Command = "Play"  
End Sub
```

