



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE INGENIERIA

**SISTEMA PARA LA ADMINISTRACION DE EVENTOS
EN UNA INSTITUCION EDUCATIVA.**

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A N :
BORJA MARTINEZ, FERNANDO
DE LA CRUZ LUCAS JACINTO
RANGEL RODRIGUEZ CARLOS ALBERTO

DIRECTOR DE TESIS: M.I. JUAN CARLOS ROA BEIZA





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A mis padres porque con nada podría pagarles el gran apoyo y ayuda que me brindaron, principalmente a mi madre que a base de su sacrificio contribuyó a culminar una de mis mas grandes metas.

A mi esposa Yolanda por la infinita paciencia y apoyo que me brindó en todo momento | te quiero |

A mi hijo Luis Fernando por permitirme robarle mucho del tiempo en el que merecía estar con él.

Fernando

A mis padres y a mis hermanos de quienes siempre he contado con su apoyo.

Al maestro Juan Carlos Roa Beiza de quién con sus comentarios y recomendaciones fue posible la realización de esta tesis.

Y de manera muy especial a mi amiga Amalia Gúzman Martínez, por su amistad y por que en ella he encontrado la motivación por ser mejor cada día.

Jacinto De la Cruz Lucas

A mis queridos padres Teresa y Antonio:
Por todo el cariño y apoyo que me han dado...
Mamita, quiero que te sientas orgullosa de mí
y que siempre tengas presente lo mucho que
te admiro...

Papá, gracias por enseñarme a ser responsable,
a no depender de nadie para salir adelante y
sobre todo a valerme por mi mismo...

Los amo y siempre los llevo en mi corazón y
en mis pensamientos...

A mis hermanos:

Por todo su apoyo y sobre todo por la paciencia
que han tenido conmigo... de todos y cada uno
de ustedes he aprendido algo, quiero que sepan
que siempre van a contar conmigo...

Mariana, Daniel, Angélica, Juan, María, Bárbara,
los quiero mucho...

Ernesto, que te puedo decir... más que mi hermano,
tú has sido como un padre para mí y siempre serás
mi mayor ejemplo en la vida, gracias a ti esto es
una realidad... te quiero...

Yeni:

Mi único y gran amor, la mujer con la que voy a
compartir toda mi vida... gracias por estar conmigo...
te amo...

Carlos Alberto

ÍNDICE

Introducción 1

Capítulo 1 Antecedentes

- 1.1 Panorama general de las instituciones educativas 4
- 1.2 Entorno administrativo de las instituciones educativas 10
 - 1.2.1 Marco de referencia: planeación estratégica 11
 - 1.2.2 Análisis y síntesis de propuestas existentes 13

Capítulo 2 Teoría Básica

- 2.1 Características, ventajas y desventajas de la programación objeto-relacional 18
 - 2.1.1 Fase de creación e instrumentación de una Base de Datos 20
 - 2.1.2 Una metodología para el diseño de la Base de Datos 22
 - 2.1.3 Características de una Metodología de Diseño 23
 - 2.1.4 Metodología de MERISE 24
 - 2.1.5 Metodología de SSADM 26
 - 2.1.6 Metodología de YOURDON 29
- 2.2 Características, ventajas y desventajas de Interbase 6.01 34
 - 2.2.1 ¿Qué es Interbase? 34
 - 2.2.2 Especificaciones técnicas 35
 - 2.2.3 Requerimientos del sistema 36
 - 2.2.4 Características 38
 - 2.2.4.1 Servidor activo 38
 - 2.2.4.2 Escalabilidad y portabilidad 38
 - 2.2.4.3 Cliente/Servidor de dos capas 39
 - 2.2.4.4 Correcciones para el año 2000 40
 - 2.2.4.5 Tecnología 40

2.2.4.6	Extensiones SQL	41
2.3	Características, ventajas y desventajas de DELPHI 5 Enterprise	42
2.3.1	Herramientas de desarrollo y características	42
2.3.2	Usando la Librería de Componentes Visuales	43
2.3.3	Manejo de errores	44
2.3.4	Conectividad a base de datos y utilidades	44
2.3.4.1	Administrador BDE	45
2.3.4.2	Explorador SQL	45
2.3.4.3	Database Desktop	46
2.3.4.4	Diccionario de datos	46
2.3.5	Tipos de proyectos de desarrollo	46
2.3.6	Aplicaciones y servidores	47
2.3.7	DLLs	47
2.3.8	Componentes personalizados y paquetes	47
2.3.9	Marcos	47
2.3.10	Objetos COM y ActiveX	48
2.3.11	Características del desarrollo para internet	48
2.3.12	Multicapas: MIDAS	49
2.3.13	Internet Express	50
2.4	Características, ventajas y desventajas de Apache 1.3.20	51
2.4.1	Características	51
2.4.2	Configuración	51
2.4.2.1	El archivo de configuración httpd.conf	52
2.4.3	Proyectos asociados	56
2.4.3.1	PHP	57
2.4.3.2	Apache-SSL	57
2.5	Características, ventajas y desventajas de los CGI's	58
2.6	Características, ventajas y desventajas de HTML	61
2.6.1	HTML describe la estructura de la página	61
2.6.2	HTML no describe el diseño de la página	62
2.6.3	Por qué funciona así	63

- 2.6.4 HTML es un lenguaje de marcado 63
- 2.6.5 Cómo son los archivos HTML 64
- 2.6.6 Programas que ayudan a escribir HTML 66

Capítulo 3 Planteamiento del problema y propuesta de solución

- 3.1 Problemática actual 69
- 3.2 Necesidades de la dependencia 73
- 3.3 Identificación del problema 78
 - 3.3.1 Fallas en el hardware del sistema 79
 - 3.3.1.1 Falla en el almacenamiento del sistema 80
 - 3.3.1.2 Falla de energía eléctrica 80
 - 3.3.1.3 Falla en la comunicación 80
 - 3.3.2 Fallas en el software del sistema 81
 - 3.3.2.1 Fallas en la información almacenada en el sistema 81
 - 3.3.3 Modificaciones y mantenimiento del sistema 81
- 3.4 Opciones de solución y elección de la óptima 82
 - 3.4.1 Manejadores de Base de Datos 82
 - 3.4.1.1 Informix 83
 - 3.4.1.2 Oracle 85
 - 3.4.2 Tablas comparativas de los manejadores de Base de Datos 86
 - 3.4.3 Herramienta visuales de desarrollo 88
 - 3.4.3.1 Visual C++ 88
 - 3.4.3.2 Visual Basic 6 89
 - 3.4.4 Tablas comparativas de las herramienta visuales de desarrollo 90
 - 3.4.5 Elección de la solución 91
 - 3.4.5.1 Elección de la Base de Datos 92
 - 3.4.5.2 Elección de la herramienta visual de desarrollo 94

Capítulo 4 Desarrollo e implementación del sistema

- 4.1 Diagrama de contexto 99
- 4.2 Diagramas 101
 - 4.2.1 Diagrama de flujo de datos 101
 - 4.2.2 Diccionario de datos 123
 - 4.2.3 Diagrama Entidad Relación 126
 - 4.2.4 Normalización 132
- 4.3 Diseño y desarrollo de la Base de Datos 137
- 4.4 Diseño y desarrollo de las interfaces gráficas 153
 - 4.4.1 Procedimiento para la creación de las interfaces gráficas 155
- 4.5 Diseño y desarrollo de los CGI's 172
 - 4.5.1 Procedimiento para la creación de los CGI's 172
- 4.6 Pruebas, depuración y liberación del sistema 183
 - 4.6.1 Pruebas 183
 - 4.6.1.1 El proceso de prueba 184
 - 4.6.1.2 Tipos de prueba 185
 - 4.6.1.3 Pruebas al Sistema para la Administración de Eventos en una Institución Educativa 186
 - 4.6.2 Depuración 187
 - 4.6.3 Liberación 188
- 4.7 Factibilidad técnica y operativa 188
 - 4.7.1 Requerimientos del servidor 189
 - 4.7.2 Requerimientos del cliente 190
 - 4.7.3 Impresoras 191
 - 4.7.4 Puesta a punto 191
 - 4.7.5 Nombre del servidor y rol asignado 191
 - 4.7.6 Instalación de Interbase 191
 - 4.7.7 Administración del servidor 192
 - 4.7.8 Mantenimiento 192
 - 4.7.9 Mecanismos de respaldo y protección 193

4.7.10 Factibilidad operativa 194

4.7.11 Capacitación a usuarios y administrador del sistema 194

Conclusiones 196

Manual Técnico 198

Manual de Usuario 210

Bibliografía 227

INTRODUCCIÓN

El presente documento describe el proceso que se siguió en el desarrollo del Sistema para la Administración de Eventos en una Institución Educativa. Para su análisis se encuentra estructurado en cuatro capítulos, los cuales comentaremos a continuación.

En el capítulo 1 se presenta el panorama general y el entorno administrativo de las instituciones educativas en nuestro país.

En el capítulo 2 se abordan las características, ventajas y desventajas de la programación objeto relacional, del servidor de bases de datos relacionales Interbase 6.01, de la herramienta de programación Delphi 5.0 Enterprise, del servidor web Apache 1.3.20, de los CGI's y del lenguaje HTML; herramientas que fueron utilizadas para el desarrollo del sistema.

En el capítulo 3 se presenta la problemática actual de las instituciones educativas así como las posibles soluciones y la elección de la óptima.

El capítulo 4 es sin duda el más extenso ya que abarca todo el proceso de desarrollo de la base de datos, de las interfaces gráficas y de los CGI's.

CAPÍTULO I

1.1. PANORAMA GENERAL DE LAS INSTITUCIONES EDUCATIVAS

La educación en un contexto general tiene la finalidad de dirigir las relaciones de las personas consigo mismas, con los demás y con su entorno. Una educación así concebida, asume un carácter eminentemente social y va más allá de los muros escolares para convertirse en un proceso permanente que se da en todo momento y en todas partes; así pues, se educa en la calle, con los amigos, en el trabajo, etc. La educación es entonces un proceso permanente que se inicia con la vida y termina con la muerte.

Lo que es una realidad es que la educación y la manera de impartir ésta en nuestro país en los últimos años no ha logrado consolidar una reestructuración seria que sea capaz de confrontar la problemática tanto educativa como administrativa en las instituciones educativas (ver figura 1.1.1.) ya que por el contrario sigue apegada, desde nuestro personal punto de vista, a una enseñanza y una forma de trabajo muy tradicionalista.

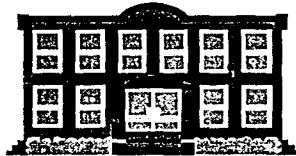


Figura 1.1.1. Institución educativa

Esta realidad se contrapone a la de los discursos oficiales que ponderan la educación como la columna vertebral del desarrollo y factor clave para el bienestar económico. Nuestra realidad requiere de una nueva estructura real crítica y tangible que considere todos los aspectos en torno a la evolución de la educación y las instituciones educativas.

Debemos ver la educación y a las instituciones educativas desde otra perspectiva, es decir, es necesario buscar nuevos enfoques, nuevos modelos y sistemas educativos que nos permitan enfrentar los nuevos tiempos.

El desarrollo de un sistema educativo que cumpla con todas las perspectivas presentes y futuras debe sustentarse en la convicción de que la educación es un proceso permanente, responsabilidad de toda la sociedad. Por ello debemos concebir la escuela no como una entidad aislada a la que se acude sólo para adquirir conocimientos, sino como el organismo que ha recibido de la sociedad la responsabilidad de fomentar ambientes educativos y de promover la sistematización permanente de aprendizajes realizados en cualquier circunstancia y por cualquier medio.

Pero es claro que cuando hablamos de educación lo primero que nos llega a la mente son las escuelas donde desde pequeños comenzamos todo un proceso de aprendizaje. Es entonces en las instituciones educativas donde queremos enfocar nuestra atención una vez que ya quedo claro que no es sólo es en un aula escolar donde adquirimos educación.

Pero no sólo la educación evoluciona, también las instituciones educativas tanto en sus métodos como en sus procesos, no sólo educativos sino además administrativos han logrado avances significativos. Es en este punto donde queremos llamar la atención pues la aportación de nuestro presente trabajo va enfocado en este sentido.

Al proponer a una institución educativa nuevos métodos y procesos que de alguna manera significan una transformación de formas y costumbres, nos estamos enfrentando a un gran problema pues al menos en lo personal consideramos que son más fuertes las costumbres que la aceptación o adaptación a nuevas formas de trabajo que sistematicen los procesos. Es importante este punto pues es aquí donde podemos darnos cuenta del porque muchos de los trámites y procesos administrativos en general realizados en las instituciones educativas son bastante lentos y poco eficientes ya que no se cuenta con un proceso sistematizado para el control de los eventos (ver figura 1.1.2.). Debemos entender por eventos todos aquellos servicios y actividades que ofrece una institución educativa.

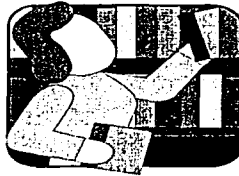


Figura 1.1.2. Proceso administrativo lento

Anteriormente comentamos que la educación no debe concebirse como una entidad aislada, sino como un medio que fomente ambientes educativos y promueva la sistematización permanente de aprendizajes realizados en cualquier circunstancia y por cualquier medio. Así mismo, las instituciones educativas deben tener la flexibilidad de adaptarse a nuevos ambientes de servicio que faciliten su trabajo, en cualquier circunstancia y por cualquier medio.

En este sentido la incorporación de la computación y el acceso a internet es fundamental (ver figura 1.1.3). Es importante incorporar las tecnologías modernas de la informática para la formación y desarrollo de un proceso educativo más integral, entendiendo por proceso educativo no sólo a la educación que se imparte, sino además al medio por el cual se imparte.

En un estudio publicado por la Comisión de Cultura del Senado de Estados Unidos, sobre el poder del Internet como herramienta en el proceso del aprendizaje, se estima que el número de personas en todo el mundo que tienen acceso a la red de Internet, asciende aproximadamente a 377 millones y que la mitad de esa cifra vive en los Estados Unidos.

En otro estudio, pero éste relativo a México y de fuente mexicana, se menciona que en nuestro país en el que habitan más de 100 millones de habitantes, se estima que menos de tres millones de computadoras se encuentran conectadas al Internet, pero menos de la mitad de ellas tiene un acceso de alta velocidad. Según señala este estudio, los equipos de baja velocidad de conexión al Internet son insuficientes para la

adecuada explotación de las herramientas, debido a que el Internet es un medio de alto contenido gráfico.

Si no hubiera muchas otras razones, la disparidad en la velocidad del acceso al Internet bastaría para excluir la pretensión de adoptar un modelo de cualquier país desarrollado, en el uso del Internet para la enseñanza.

Con lo anterior no tratamos de eliminar las pretensiones de utilizar estas herramientas como lo son la computación y el acceso a internet en la transformación de nuevos métodos y procesos que faciliten la labor de las instituciones educativas, por el contrario nuestro trabajo esta enfocado en este sentido, más sin embargo debemos ser claros en los terrenos que vamos y estamos pisando para ser congruentes y realistas con lo que estamos haciendo.



Figura 1.1.3 Importancia de la computación e internet

De todo lo anterior se desprenden dos puntos muy importantes:

- Las instituciones educativas deben adaptarse a nuevas formas y métodos de trabajo que les permitan brindar un mejor servicio, lo cual indudablemente se verá reflejado en el desempeño de todas las actividades que llevan a cabo (académicas, culturales, deportivas, etc.).
- La importancia de la computación y el internet tanto en el desempeño académico como en el proceso de automatización de las actividades administrativas que se llevan a cabo en toda institución educativa.

En el punto uno es de suma importancia considerar al personal que labora en las instituciones educativas, ya que de ellos depende el que se cumplan o no los objetivos propuestos. El recurso humano es entonces uno de los aspectos primordiales y que no podemos pasar por alto para que las instituciones educativas de nuestro país puedan concebir un sistema educativo de calidad; así entonces, es necesario hacer una integración eficiente del recurso humano con los recursos físicos y materiales para poder así alcanzar un nivel de competitividad que permita satisfacer los requerimientos que demandan los nuevos tiempos. Una de las premisas a considerar en el campo de los recursos humanos es lograr la integración del elemento humano para el máximo aprovechamiento de su eficacia y capacidad, a través de nuevas formas y métodos de trabajo que hagan más claras y transparentes sus actividades. La calidad en el servicio de cualquier institución educativa debe estar sustentada en la capacidad de mantener operando establemente sus sistemas de trabajo, y para poder lograrlo, se necesita contar con personas motivadas y capacitadas para hacer un buen trabajo.

El segundo punto es claro en los objetivos que pretende, pues es con el aporte de nuevas herramientas como se puede mejorar de manera sustancial los eventos que desarrollan en las instituciones educativas para de esta manera poder brindar un servicio eficiente y de calidad. El aporte de la computación y el internet debe abrir un nuevo panorama en la forma en como se debe resolver gran parte de las actividades que llevan a cabo en una institución educativa, ya que por un lado se van a sistematizar muchos de los procesos que en la actualidad son lentos y poco eficientes y por otra parte las instituciones educativas se incorporarán a los cambios que exigen los nuevos tiempos.

En el México de hoy las condiciones que faltan parecen residir en el clima de trabajo y en la cultura escolar, que gobiernan el funcionamiento cotidiano de las instituciones educativas y que influyen en el desarrollo del trabajo docente y administrativo. Estas condiciones tienen relación con el conjunto de valores, de formas de entender la tarea de las instituciones educativas, de normas explícitas e implícitas y de costumbres.

A pesar de que se han visto algunos intentos por cambiar la cultura educativa y administrativa dentro de las instituciones es evidente que se requiere de un esfuerzo que va más allá de los deseos por lograrlo; las instituciones educativas deben considerar aspectos fundamentales como lo son: el financiamiento, la organización general de la institución, planes y programas de trabajo, materiales educativos y de apoyo para la enseñanza, etc.

Aunque nuestra intención no va orientada a tocar aspectos como los mencionados anteriormente, es importante tomar en consideración aspectos como el financiero ya que el hecho de sistematizar métodos y formas de trabajo dentro de las instituciones educativas requiere en muchos casos de inversiones importantes de capital; sabemos que las condiciones de nuestras instituciones educativas en términos generales no son las más idóneas para alcanzar los objetivos que se desean, más sin embargo se debe pensar en mecanismos alternos, en este caso de financiamiento, que permitan ir armando una infraestructura que facilite la automatización de las actividades tanto académicas como administrativas dentro de una institución educativa.

Las instituciones educativas hoy en día se encuentran aún apegadas a procesos poco eficientes para poder brindar un servicio de calidad que se vea reflejado en todos los ámbitos de sus actividades diarias. Los servicios deben ser más ágiles y confiables evitando burocratizar procesos tan simples como lo sería, por ejemplo el préstamo de material de apoyo para dar una clase. Ejemplos como el anterior hay muchos, que como ya se mencionó anteriormente abarcan aspectos educativos, culturales, deportivos y recreativos.

La incorporación de sistemas que permitan llevar a cabo de manera eficaz los procesos tanto educativos como administrativos debe ser uno de los objetivos de toda institución. Con la incorporación de nuevos sistemas es posible reducir los tiempos de espera entre procesos (sean estos de cualquier índole: educativos, culturales, deportivos, recreativos), dado que se contará con mecanismos automatizados.

Lo que es una realidad es que en nuestro país existen muchas instituciones educativas que no cuentan actualmente con un mecanismo automatizado para la administración de eventos, por el contrario, este proceso se realiza de forma manual lo que lo hace poco eficiente, tardado y burocrático.

Toda institución educativa que desee optimizar sus recursos debe implementar sistemas para que tanto los usuarios como el personal que los opere reduzca los tiempos en que se realiza un proceso y el servicio sea mucho más eficiente.

Para que los recursos de una institución educativa puedan ser administrados de forma óptima es necesario contar con medios que nos permitan programar las actividades. Herramientas como lo son la computación y el internet son un claro ejemplo de mecanismos que nos permiten hacer frente a las necesidades derivadas tanto de los procesos educativos como administrativos dentro de las dependencias.

Finalmente, debemos de tener en cuenta que la sistematización de todo tipo de eventos, sean educativos, administrativos, etc., debe ser generadora de nuevas formas y métodos de trabajo para las instituciones educativas, lo que les va a permitir brindar un mejor servicio, el cual se verá reflejado en su desempeño diario. El aporte de herramientas computacionales será fundamental para lograr que las dependencias educativas automaticen sus procesos de manera segura y eficiente.

1.2. ENTORNO ADMINISTRATIVO DE LAS INSTITUCIONES EDUCATIVAS

La administración dentro de las instituciones educativas en nuestro país tiene como principal reto la modernización. En mayor o menor medida las instituciones requieren cambios que garanticen la continuidad de sus funciones. Son pocas las instituciones que buscan un mejoramiento continuo en sus servicios, la mayor parte de ellas no toman acciones que permitan mejorar la eficiencia de las mismas, en la actualidad un gran número de éstas cuentan con una administración tradicional que se caracterizan

por estar formada por órganos rígidos, y que, consecuentemente, no sufren demasiadas alteraciones en su existencia. Debido a esta situación la aparición de nuevos ámbitos de acción de la administración se traducen en la creación sucesiva de nuevos órganos. En estas circunstancias, se elude la reflexión sobre los criterios organizativos de las administraciones públicas o sobre las formas de integrar los servicios que se prestan, los órganos que los realizan y las personas que han de trabajar en dichos órganos.

El entorno es el contexto social, económico, político y cultural en el que actúan las dependencias y entidades educativas. El análisis permanente de las relaciones de los distintos elementos de la estructura social, es de vital importancia para asegurar que las estrategias de la administración educativa, sean congruentes con las expectativas y necesidades de la población. Las variantes resultantes del análisis de la organización social contemporánea, de su cultura, de lo económico, lo político, de la interrelación entre lo público y lo privado, afectan a los procesos y la calidad de los servicios que cada institución tiene como objetivo. Por ello, las instituciones educativas buscan establecer una disciplina de continua observación de las modificaciones de estos elementos, para permitir identificar las amenazas y oportunidades que ofrece la realidad en términos de temporalidad, para así exponer y definir los principios de una gestión eficiente, e identificar los elementos de fortaleza y debilidad de cada sector. Lo anterior permite de manera permanente conducir la planeación estratégica y los procesos sustantivos de la administración educativa, hacia el servicio eficaz y oportuno que satisfaga a los usuarios.

1.2.1 Marco de referencia: planeación estratégica

La planeación con frecuencia es la etapa que marca el punto de partida del proceso administrativo, anticipa los fines y cómo se deben de alcanzar. Significa que en la planeación deben de fijarse los objetivos y jerarquizarlos, considerando la forma y recursos necesarios para su logro.

Los objetivos son los resultados futuros a los que se dirige una organización: Los hay globales para toda la organización o particulares para partes específicas de la misma. En la medida que se avance en el desarrollo de la planeación, se establecen políticas,

directrices, metas, programas, procedimientos, métodos y normas que vinculan en su ejecución las acciones para alcanzar las metas.

En la medida en que se realiza la planeación de acuerdo con la jerarquía otorgada a los objetivos pueden identificarse tres niveles diferentes de planeación: la estratégica, la táctica y la operacional.

La planeación estratégica es la más amplia y subordina a los otros dos niveles. Se caracteriza por que:

- Se proyecta en el largo plazo
- Engloba a la organización completa y sus recursos
- Es desarrollada por el nivel más alto de la organización.

La planeación táctica se desarrolla en el nivel de departamento y se caracteriza por que:

- Se proyecta para el mediano plazo (frecuentemente un año).
- Abarca cada sección o departamento y sus objetivos particulares.
- Se define por cada departamento.

La planeación operacional se efectúa para cada tarea o labor y está constituida generalmente por metas, programas, procedimientos, métodos y normas, se caracteriza por que:

- Se proyecta para el plazo inmediato.
- Se dirige a tareas específicas y sus metas.
- Se define por cada tarea o actividad.

De esta manera, puede identificarse la planeación estratégica como un modelo particular de la etapa con la que conceptualmente se inicia el proceso administrativo,

ésta puede verse como una normalización del proceso de planeación que enfatiza el logro de objetivos de la organización más que los medios para obtenerlos, ante un ambiente dinámico circundante a la organización.

La forma más general de este proceso podría esquematizarse con los siguientes pasos:

- Se establecen y jerarquizan los objetivos generales de una organización para el largo plazo.
- Se hace el análisis interno de la organización.
- Se hace el análisis externo de la organización.
- Se formulan alternativas estratégicas.
- La última etapa se complementa estableciendo los planes tácticos y operativos vinculados como requisitos del plan estratégico.

1.2.2. Análisis y síntesis de propuestas existentes

Cabe mencionar que respecto a investigaciones y artículos sobre educación se consideró el acervo referente a México, por considerar que en un tema como éste, las particularidades del sistema de educación superior son fundamentales.

- a) Los trabajos existentes sobre México que analizan la planeación administrativa de las instituciones de educación superior son en su mayoría ensayos y opiniones referidas a un marco global de la educación superior pública.
- b) Hay más o menos un consenso en los trabajos que abordan el tema, respecto a la necesidad de profesionalizar la administración de las instituciones de educación superior en el país.
- c) Los trabajos sobre planeación educativa en México se refieren generalmente a las instituciones públicas y a una visión nacional de la educación; lo cual podría definirse como macroplaneación.

Una propuesta señala que la planeación existente para las instituciones educativas es, por lo general, de tipo táctico, es decir, de muy corto plazo y atribuye esta situación a factores inherentes a las organizaciones educativas, como carencia de recursos humanos especializados, falta de formalización en los procesos (improvisación); actividad directiva ausente de visión estratégica; nulo convencimiento de la importancia de la función de planeación (no se considera redituable); divorcio entre contenidos curriculares y necesidades del entorno, etc.

La propuesta menciona que la planeación le da a la educación la posibilidad de precisar el sentido de su actividad y su incidencia en el desarrollo del país, lo cual resulta fundamental para una institución de carácter privado. Enumera tres funciones básicas que debe realizar la persona o grupo encargado de la planeación:

1. Función de información, que describe como la recopilación de información ayuda a conocer la realidad educativa.
2. Función de coordinación, se refiere básicamente a la elaboración de diferentes propuestas de acción para armonizar la planeación con el estilo de la organización.
3. Función de exploración, que consiste en realizar estudios de los problemas de la institución, su medio ambiente y el establecimiento de acciones concretas.

Finalmente, el modelo se base en mostrar sistemáticamente los pasos que el dirigente de las instituciones educativas a nivel superior debe seguir, con el fin de desarrollar acciones de planeación estratégica diferenciándola de la planeación táctica.

El modelo propuesto inicia con el análisis de los demandantes del servicio, a los que denomina como "directos" (es decir, en nuestro caso, los alumnos representarían lo más típico) e "indirectos", que demandan el producto final, o sea los profesionales egresados. Este concepto resulta especialmente importante en la fase de planeación estratégica para el análisis del mercado y del cliente.

La siguiente etapa del modelo es la formalización de un sistema de creencias y valores (filosofía de la institución), éste indica que debe contrastarse con el diagnóstico del entorno para disminuir la subjetividad con la que suele impregnarse.

Siguiendo el modelo, continúa el diagnóstico que describirá los riesgos y oportunidades que acechan a la institución, para así prever las posibles líneas de acción. Basado en lo anterior se determina el futuro deseado, lo cual esboza la estrategia a seguir.

Simultáneamente, se elabora un análisis integral de la organización, con lo que se detectan sus fuerzas y debilidades, y se pasa a la determinación del futuro deseado y posible de alcanzar.

Con todo lo anterior, puede formularse objetivamente la estrategia, así como el punto de origen de la planeación táctica, mediante los planes de acción. Así se procede a la elaboración de los programas y a la asignación de los presupuestos correspondientes.

Finaliza el modelo con la idea de definir la situación de las instituciones y destaca la necesidad del análisis ante factores fundamentales como: las fases del sector y el valor del sector.

Se expone la posición competitiva como aquella que se guarda frente a otras instituciones similares respecto a su participación, en lo que se podría denominar como "mercado" (este término utilizado con las características que le correspondería en el ámbito educativo, no precisamente equivalente al de un servicio mercantil, pero que guarda ciertas semejanzas y valida su utilización, sobre todo desde el punto de vista de la educación privada). Esta participación podría ser dominante, fuerte, mediana o débil.

En cuanto a las fases, el modelo advierte que son denotativas del grado de madurez de cada institución dentro del sector, y se basan en cuestiones tales como experiencia, imagen, tasa de crecimiento, etc.

Como valor de sector apunta a las cualidades que valorizan el servicio prestado, y generaliza para la educación tres "diadas" en ese sentido, aclarando que se trata del punto de vista del demandante directo:

- 1) diada: posición económica-estatus. Movilidad en la escala social por el grado de educación adquirido.
- 2) diada: elitismo-calidad. Reservar la educación para un conjunto reducido de demandantes.
- 3) diada: cultura-agente del cambio. La educación puede producir agentes del cambio en la sociedad.

Esta última "diada" podría colocarse más bien en el valor del sector, denomina demanda "indirecta", es decir, el demandante de profesionales, lo que desde el enfoque de este trabajo es muy importante y se podría agregar una más:

- 4) Calidad institucional-calidad profesional. A mayor calidad de la institución educativa se obtendrán mejores profesionistas, factor de gran importancia para el demandante de profesionales.

El acervo de la planeación estratégica en instituciones de educación superior en países como Estados Unidos es bastante más extenso y rebasa la propuesta teórica. De hecho, a partir de 1994 puede rastrearse un interés creciente por la aplicación de estrategias de negocios en la administración de la educación superior, cuyo enfoque resulta de la necesidad urgente de las instituciones de aquel país por garantizar los recursos financieros provenientes de la sociedad, en un momento en el que fuertes reducciones presupuestales se imponen para escuelas de esta naturaleza, debido a una pérdida de confianza del público estadounidense en el uso de los recursos en la educación superior.

CAPÍTULO II

2.1. CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE LA PROGRAMACIÓN OBJETO-RELACIONAL

El proceso de diseñar una base de datos implica identificar los objetos a ser modelados. Un objeto en este sentido es cualquier elemento distinguible relevante en torno a un diseño en específico. Estos objetos poseen un número indeterminado de propiedades, que son fragmentos de información que describen a los objetos de alguna manera.

El modelo objeto-relacional es una herramienta de modelado para representar los objetos, sus propiedades y las relaciones entre ellos, a su vez el modelo objeto-relacional cuenta con una representación gráfica que facilita una visión global de lo que se modela (ver figura 2.1.1).

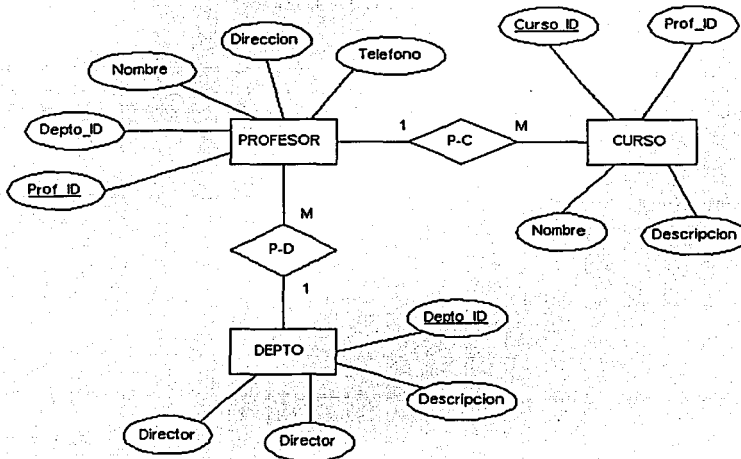


Figura 2.1.1 Modelo objeto-relacional

Las relaciones entre los objetos tienen diferentes grados de cardinalidad; la cardinalidad de una relación es el número de elementos de los objetos que se relacionan entre sí, a

través de la relación en cuestión. Los tres casos de mayor interés son 1:1 (uno a uno), 1:M (uno a muchos) y M:M (muchos a muchos).

La normalización dentro de un modelo objeto-relacional implica eliminar redundancia del proceso de almacenamiento de la base de datos. Al final del proceso de normalización los objetos se convierten en tablas y los atributos en clases; los objetos utilizan un OID (identificador del objeto) para identificarse de manera única en la base de datos.

El álgebra relacional es un sistema que permite especificar operaciones de consulta a través de pasos intermedios de generación de tablas utilizando formatos especiales. Existen dos clasificaciones de procesos en álgebra relacional:

Operadores tradicionales

- Union.
- Intersect.
- Minus.
- Times.

Operadores especiales

- Select
- Project
- Join
- Divide

El modelo objeto-relacional ha evolucionado desde sus inicios y hoy en día nos encontramos con nuevos conceptos y enfoques en torno a este tipo de modelado. Es conveniente entonces entender algunos conceptos de manera más formal para poder abordar la metodología que será utilizada a lo largo del proyecto.

El diseño de sistemas basados en bases de datos ha evolucionado de forma bastante acelerada en los últimos tiempos, por lo que los profesionales en el tema están cada vez más interesados en buscar nuevas y más sencillas técnicas para el desarrollo de estos sistemas; de la misma forma, los productos comerciales, es decir, los sistemas administradores de bases de datos que se nos ofrecen en el mercado, han mejorado sensiblemente la calidad de sus presentaciones, tanto en sus aspectos funcionales como de eficiencia, existiendo en la actualidad una amplísima y variada oferta de estos sistemas.

2.1.1 Fases de creación e instrumentación de una base de datos.

Debido a que la creación de una base de datos suele ser en la mayoría de los casos difícil, larga y costosa no puede improvisarse. El desarrollo de una base de datos no es sólo un tema técnico en el que sólo intervienen los desarrolladores, es necesario que el personal que labora en las áreas en las que se va a implementar el sistema se involucren desde un principio en la elaboración del sistema.

Las fases que comprende la puesta en marcha de un sistema de información orientado hacia las bases de datos, se resumen en la figura 2.1.2.

La decisión política y fijación de objetivos, debe preceder obligatoriamente a cualquier operación de concepción o diseño de una base de datos; en ella se ha de concretar la voluntad de los directivos de abordar el proyecto, definiendo unos objetivos claros y concretos que sirvan de pauta en todo el desarrollo.

Obtenida la conformidad activa por parte de la dirección para emprender el proyecto, será preciso hacer un plan de trabajo detallado en el que se especifiquen las distintas fases, con los plazos y medios que requerirán cada una de ellas.



Figura 2.1.2 Fases para la puesta en marcha de una base de datos

El plan de trabajo detallado ha de ser aprobado por la dirección antes de pasar a la siguiente etapa, y su rechazo puede obligar bien a la reelaboración o incluso a un replanteamiento de la etapa inicial de estudio de oportunidad, reconsiderando los objetivos, medios y plazos.

La fase llamada de concepción, en la cual se concretará el mundo real en un esquema, que es lo que nosotros percibimos (esquema conceptual) de la fase de diseño donde se plasmará dicho esquema en una estructura lógica y en una estructura física.

Para su puesta en marcha, será preciso evaluar las exigencias en cuanto a equipo, en especial respecto al sistema administrador de base de datos y a la dimensión del equipo (memoria principal y secundaria, capacidad de proceso, terminales, etc.).

El diseñador deberá aplicar el modelo propio del sistema administrador de la base de datos (por ejemplo, relacional) al esquema conceptual; sin embargo, un modelo aplicado a un determinado esquema conceptual no tiene porqué llevar a un único esquema sino a un conjunto de ellos, debiéndose buscar el mejor, de acuerdo con determinados criterios (mejor representación de la información conservando el máximo de semántica, poca redundancia –normalización-, facilidad de comprensión por los

usuarios, adaptabilidad a los cambios, protección de datos, máxima eficiencia de ciertas aplicaciones críticas, etc.).

Definida la estructura física de la base de datos es preciso cargar los datos en la misma. En general, muchos de estos datos proceden de aplicaciones anteriormente automatizadas, en cuyo caso lo único que habrá que hacer es proceder a la carga de estos archivos; muchos sistemas de administración de base de datos dan facilidades en este aspecto evitando escribir los correspondientes programas.

Cargados en la base de datos algunos archivos se deben comenzar inmediatamente las pruebas de la base de datos y medir su rendimiento, con objeto de poder ir ajustando la estructura física e incluso, a veces, la estructura lógica con fines de optimización.

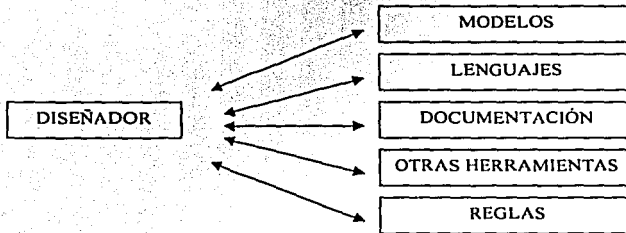
2.1.2 Una metodología para el diseño de la base de datos

Las dificultades inherentes al diseño de una base de datos han de afrontarse con procedimientos ordenados y metódicos. A veces, el diseño de una base de datos relacional se ha limitado simple y llanamente a la teoría de la normalización, cuando en realidad debe abarcar muchas otras etapas que van desde la concepción hasta la instrumentación. Una metodología es un conjunto de modelos y herramientas que nos permiten pasar de una etapa a otra en el proceso de diseño de la base de datos.

Teniendo en cuenta que una metodología es un conjunto de modelos, lenguajes y otras herramientas que nos facilitan la representación de los datos en cada fase del proceso de diseño de una base de datos, junto con las reglas que permiten el paso de una fase a la siguiente, el análisis de todos estos elementos es fundamental para poder comprender y aplicar correctamente una metodología de diseño.

La documentación nos permitirá describir de forma normalizada los resultados de cada etapa, facilitando así la labor del diseñador y ayudando al mantenimiento de la base de datos. Las reglas actuarán sobre los elementos de entrada de cada fase para conseguir las salidas de cada una de ellas, permitiendo en algunos casos elaborar distintas alternativas de diseño.

Estos cinco conceptos (modelos, lenguajes, documentación, otras herramientas y reglas), que se presentan en la figura 2.1.3, están estrechamente ligados: un lenguaje permite la expresión organizada de los conceptos del modelo, los modelos no pueden aplicarse de forma satisfactoria sin una metodología, y una metodología será más eficaz con el apoyo de herramientas que faciliten su aplicación y con reglas que permitan pasar de una etapa a otra, ayudando a resolver los problemas que van apareciendo en el proceso de diseño, el cual debe estar perfectamente documentado para que puedan llevarse a cabo las revisiones y el mantenimiento.



2.1.3 Componentes básicos de la metodología.

2.1.3 Características de una metodología de diseño

Las características que se consideran deseables en una buena metodología de diseño son las siguientes:

- Claridad y comprensibilidad.
- Capacidad de soportar la evolución de los sistemas.
- Facilitar la portabilidad.
- Versatilidad respecto a tipos de aplicaciones.
- Flexibilidad.
- Rigurosidad.
- Adoptar estándares.
- Automatización.

2.1.4 Metodología de MERISE

Es un método de enfoque sistemático, el cual considera las etapas siguientes:

- Estudio previo. Tiene por misión definir de manera global las soluciones conceptuales, organizativas y técnicas del futuro de la base de datos.
- Estudio Detallado. Permite definir explícitamente las especificaciones funcionales de la aplicación.
- Estudio técnico. Realiza la distribución de datos en archivos físicos, y el tratamiento en módulos de programas.
- Realización y puesta en marcha. Producción de los módulos de programación e implantación de los medios técnicos y organizativos necesarios, así como formación del personal, lanzamiento de la aplicación y recepción definitiva por parte del usuario.
- Mantenimiento. Etapa que abarca el resto de la vida de la base de datos.

Quedando un cuerpo metodológico completo agrupado bajo la denominación de metodología MERISE, ver figura 2.1.4.

A continuación se describen las etapas y fases consideradas por la metodología MERISE:

Esquema director. Tiene por misión prever y planificar el desarrollo de la base de datos, definiendo a medio y largo plazo los recursos a utilizar y la política presupuestaria correspondiente. Consta de las siguientes fases:

- Preparación.
- Estudio de la realidad.
- Balance de necesidades.
- Construcción de escenarios.
- Planes de acción.

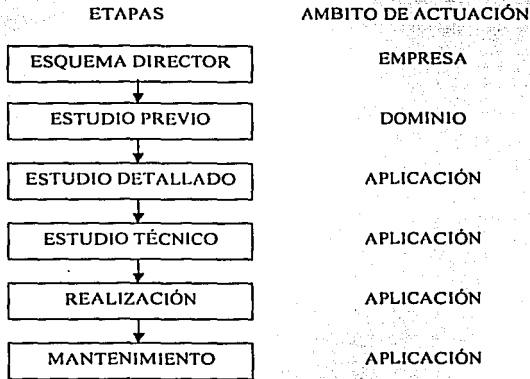


Figura 2.1.4 Etapas del ciclo de vida en la Metodología MERISE

Estudio previo: Persigue conseguir una visión del dominio estudiado, definido en un esquema director, evaluar riesgos y ventajas, planificar la puesta en marcha y determinar costos y recursos. Consta de las siguientes fases:

- Estudio de lo existente.
- Concepción de la base de datos.
- Conclusión del estudio previo.

Estudio detallado: Profundiza en la información obtenida en el estudio previo, centrándose en la solución diseñada, define los datos a utilizar, así como sus interrelaciones, procesos y flujos de información, además planifica la realización e implantación de la base de datos. Consta de las siguientes fases:

- Diseño funcional.
- Diseño detallado.
- Conclusión del estudio detallado.

Estudio técnico: En él se definen la estructura y los datos de la base de datos, así como su organización física, la arquitectura técnica y los procedimientos de seguridad. Consta de las siguientes fases:

- Diseño técnico.
- Modelo físico de datos.
- Modelo operativo de tratamientos.
- Estrategia de producción/entrega del software.

Realización y puesta en marcha: Su objetivo es la obtención de la base de datos conforme a las reglas especificadas y aceptación por los usuarios, y el establecimiento de un plan de lanzamiento basado en los plazos definidos al final del estudio previo. Consta de las siguientes fases:

- Producción.
- Pruebas individuales.
- Pruebas de integración.
- Preparación de los recursos de puesta en marcha.
- Entrega y lanzamiento.

Mantenimiento: Consiste en evolucionar las aplicaciones en función de las necesidades de los usuarios, del entorno y de los progresos tecnológicos. Consta de las fases:

- Estudio del impacto.
- Análisis de las modificaciones.
- Realización de las modificaciones.
- Puesta en marcha.

2.1.5 Metodología de SSADM

SSADM proporciona un conjunto de especificaciones y procedimientos para llevar a cabo las tareas de análisis y diseño de las bases de datos. No cubre la planificación estratégica ni el control de proyectos, y el diseño físico llega hasta el máximo grado de

detalle en especificaciones, pero sin entrar en la construcción del código. La estructura de la metodología se presenta en la figura 2.1.5.

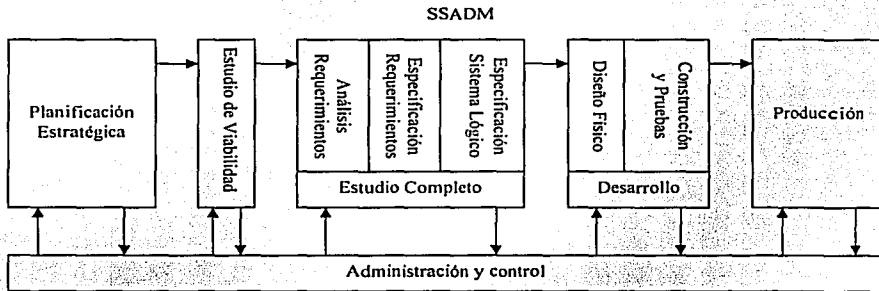


Figura 2.1.5 Etapas del ciclo de vida en la Metodología SSADM

La estructura esta formada de los siguientes módulos:

Módulo de estudio de viabilidad: Este módulo llega a establecer cuál puede ser la base de datos que responda satisfactoriamente a los requerimientos de la organización.

- Preparar el estudio de viabilidad.
- Definición del problema.
- Selección de opciones.
- Composición del informe de viabilidad.

Módulo de análisis de requerimientos: Define el alcance del proyecto, establece la integración de la tecnología con las necesidades de la organización. Según los requerimientos, y consigue una visión global de costos y beneficios.

- Establecimiento del marco de trabajo.
- Investigación y definición de requerimientos.
- Investigación de procesos actuales y datos.
- Obtención de una visión lógica actual.
- Composición de los resultados de la investigación.
- Definición de opciones de la base de datos.

- Selección de una de las opciones.

Módulo de especificación de requerimientos: Este módulo proporciona una descripción detallada del futuro de la base de datos, estableciendo criterios de aceptación medibles que permitan llegar al diseño lógico con la garantía del acuerdo entre las partes implicadas. También especifica las descripciones de datos, procesos y requerimientos con el máximo grado de detalle. A partir del modelo de flujo de datos se describe la funcionalidad de la base de datos mediante las técnicas de definición de funciones y estructuras de entradas y salidas. Se completa el estudio de los datos con el análisis relacional de datos y la historia de vida de las entidades. Se especifican con detalle los eventos y se definen con la correspondencia de efectos. Con el estudio de los requerimientos de acceso a la base de datos se completa el modelo lógico de datos.

- Definición de procesos y datos de la base de datos requeridos.
- Obtención de las funciones de la base de datos.
- Refinación del modelo de datos requerido.
- Desarrollo de prototipos.
- Desarrollo de especificaciones de procesos.
- Confirmación de objetivos de la base de datos.
- Composición de la especificación de los requerimientos.

Módulo de especificaciones lógicas: El sistema administrador de la base de datos seleccionado y la especificación de requerimientos se traducen en un conjunto de opciones de implementación técnica que serán evaluadas por la dirección.

- Definición de opciones técnicas.
- Selección de la opción técnica.
- Definición de diálogos de usuarios.
- Definición de procesos de actualización.
- Definición de procesos de consulta.
- Composición del diseño lógico.

Módulo de diseño físico: Este módulo especifica datos, procesos, entradas y salidas, de acuerdo al entorno físico elegido, incorporando los estándares de la instalación.

- Preparación del diseño físico.
- Creación del diseño físico de datos.
- Creación de funciones.
- Optimización del diseño físico de datos.
- Complementación de funciones.
- Consolidación de interfaces de proceso.
- Composición del diseño físico.

2.1.6 Metodología de YOURDON

La metodología de análisis y diseño estructurado de YOURDON, esta clasificada como perteneciente a la escuela de la ingeniería de software, además de estar orientada a procedimientos y contar con soporte para sistemas de información, tal y como se muestra en la figura 2.1.6.

Esta metodología se basa en un análisis de diseño estructural, usa los métodos de arriba-abajo y descomposición funcional para definir los requerimientos de sistema y diseño de bases de datos. Las especificaciones de la base de datos por el proceso de análisis estructurado, es un modelo descendente particionado. La descripción de los requerimientos se convierte en la liga entre el análisis y el diseño.

Esta metodología utiliza las siguientes herramientas gráficas de modelado: diagrama de flujo de datos (DFD), diagrama entidad-relación (DER), diagrama de transición de estados y diagrama de estructuras; la figura 2.1.7 muestra lo anterior, donde las diagramas ayudarán a concentrarse en las propiedades más sobresalientes de la base de datos, aclarar cambios y correcciones de requerimientos del usuario y respaldar con documentación el ambiente del usuario.

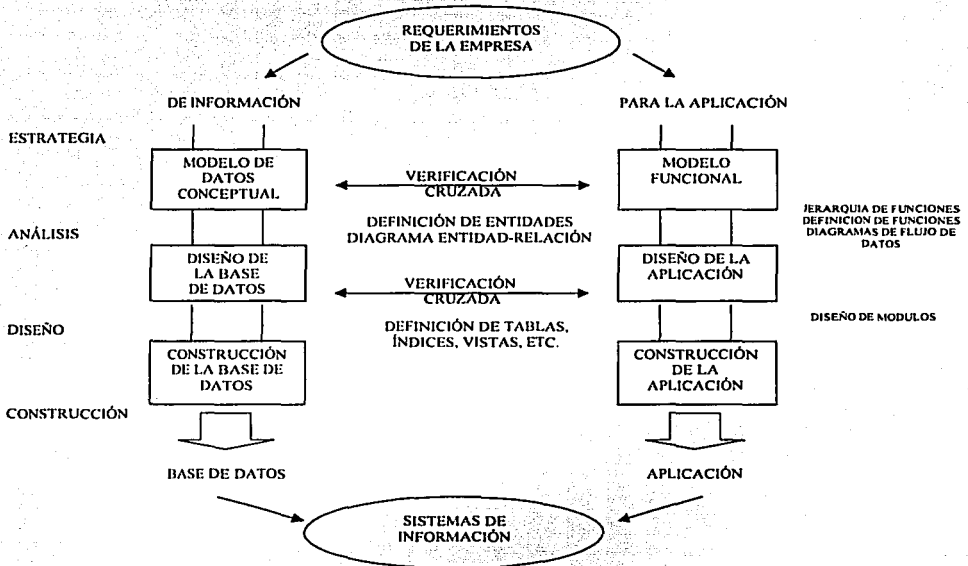


Figura 2.1.6 Diseño estructural de la Metodología de YOURDON

El diagrama de flujo de datos describe los procesos de la base de datos, es decir, la transformación de entradas y salidas. Esta herramienta permite visualizar una base de datos como una red de procesos funcionales, conectados entre sí por líneas de flujo y almacenamiento de datos. Los diagramas de flujo de datos se componen de los siguientes elementos:

- **Procesos.** Los cuales representan las diferentes funciones que la base de datos lleva a cabo.
- **Flujos.** Son conexiones entre los procesos e indican la información que se requiere como entrada y la información que genera como salida de cada proceso.
- **Archivos de datos.** Son colecciones de datos que la base de datos guardará para ser llamados en un periodo de tiempo.

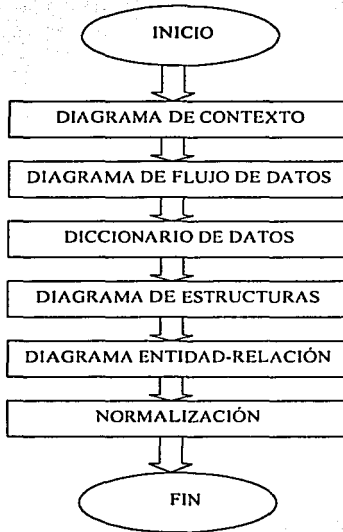


Figura 2.1.7 Diagrama de la Metodología de YOURDON

Las siguientes reglas ayudan a elaborar un diagrama de flujo de datos con el mínimo de errores y cambios:

- Escoger nombres adecuados para los procesos, flujos y archivos.
- Enumerar los procesos.
- Redibujar el diagrama de flujo de datos las veces que sea necesario.
- Evitar los diagramas de flujo de datos complejos
- Asegurar que el diagrama de flujo de datos sea internamente consistente.

Además, para mostrar con más detalle el diagrama de flujo de datos se utiliza el diccionario de datos, el cual define el significado de los datos almacenados en la base de datos.

Los diagramas de flujo de datos sólo muestran un aspecto principal de la base de datos. Sin embargo, es necesario conocer más a detalle la relación que existe entre los datos.

Este aspecto es elaborado con la herramienta llamada diagrama de entidad-relación, el cual enfatiza las relaciones entre los archivos de datos en el diagrama de flujo de datos.

El diagrama entidad-relación se utiliza para mostrar el tipo de relación existente entre las diferentes entidades de una base de datos, mismas que pueden ser "uno a uno", "uno a muchos", "muchos a uno" o "muchos a muchos". El diagrama entidad relación esta compuesto de los siguientes elementos:

- Entidades. Se caracterizan porque cada una puede identificarse de manera única por algún medio, cada una juega un papel importante en la base de datos que se desarrolla y cada una puede describirse por uno o más datos.
- Relaciones. Son asociaciones entre las entidades, cada instancia de la relación representa una asociación entre cero o más ocurrencias de un objeto y cero o más ocurrencias del otro.

De igual forma que el diagrama de flujo de datos, es necesario acompañar el diagrama de entidad relación con información textual detallada. De esta manera, también podrá usarse el diccionario de datos para mantener información apropiada acerca de objetos y relaciones.

Un diagrama de transición de estados, es la secuencia con la cual se tendrá acceso a los datos y se ejecutarán las funciones. Esto, se refiere a que en algunos sistemas computacionales, la función N no puede llevar a cabo su labor hasta que recibe la entrada que requiere, y esta entrada se produce como salida de una función N-1, y así sucesivamente.

Una herramienta gráfica de modelado que desarrollan los diseñadores de sistemas y no los analistas, son los diagramas de estructuras que representan la jerarquía de software. Los diagramas de estructuras están compuestos de invocaciones a módulos, además de los parámetros de entrada que se le dan a cada módulo invocado, y los

parámetros de salida devueltos por cada módulo cuando terminan su labor y le devuelven el control al que lo llama.

2.2. CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE INTERBASE 6.01

2.2.1 ¿Qué es InterBase ?

Interbase es un sistema de administración de bases de datos relacionales completo, implementado como una arquitectura cliente/servidor de dos capas: una aplicación cliente y una tecnología servidor que ofrece soporte transparente a través de redes heterogéneas. Esta arquitectura es flexible, permitiéndole incluir fácilmente la ingeniería de bases de datos. En la figura 2.2.1. se muestra la consola de administración de Interbase.

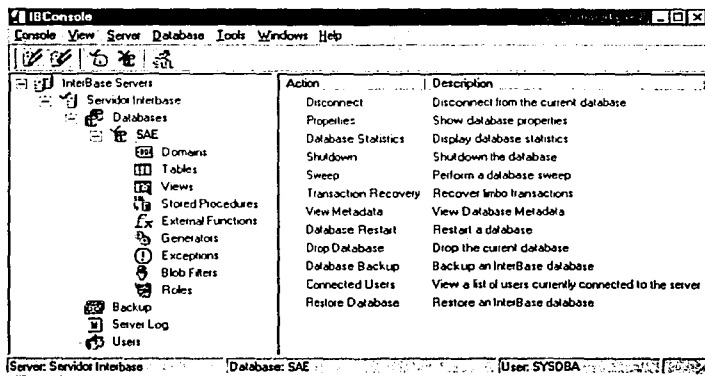


Figura 2.2.1 Consola de Interbase

Las características de servidor activo de Interbase le permiten trabajar autónomamente por lo cual no es necesario tener un administrador de bases de datos dedicado. Interbase es por mucho el servidor de bases de datos más fácil de usar.

La integración de Interbase con las herramientas de desarrollo cliente/servidor líderes y la implementación de estándares de la industria lo hacen la elección ideal de los desarrolladores que deben incluir una base de datos en sus aplicaciones.

2.2.2 Especificaciones técnicas

Integridad

- Llaves primarias.
- Llaves foráneas.
- Integridad referencial en cascada.
- Verificación de valores en dominios y columnas.
- Triggers (disparadores) con las siguientes características:
 - Número ilimitado de triggers por actualización/inserción/eliminación en un registro de una tabla.
 - Triggers múltiples por acción (agregar/modificar/eliminar) con opción a ordenarlos.
 - Triggers en cascada.

Control de Concurrencia

- Bloqueo optimista.
- Niveles de aislamiento de datos.
- Bloqueos compartidos y protegidos para cuando se bloquea una tabla explícitamente.

Disponibilidad

- Respaldos en línea (no hay que dar de baja el servicio).
- Recuperación inmediata en caso de una falla en el servicio.

Base de datos distribuida

- Conexiones ilimitadas de clientes (únicamente limitadas por el hardware).
- Proceso de transacciones distribuidas automáticas mediante commits de dos fases.

Tipos de datos

- Caracteres (de longitud fija y variable) de hasta 64kb por campo.
- Enteros (8, 16 y 32 bits).
- Punto flotante: de precisión sencilla y doble.
- Fecha y hora: desde el 1 de enero del 100 hasta el 11 de diciembre del 5491.
- Cumple con el año 2000.
- Arreglos multidimensionales: hasta 16 dimensiones por columna.
- BLOBS (memos, campos binarios) de tamaño ilimitado.
- Importa y exporta datos ASCII de tamaño fijo.

Estándares

- Cumple con ANSI SQL-92.
- ODBC rev 2.0 (16 bits).
- ODBC rev 3.0 (16 bits).
- UNICODE.

2.2.3 Requerimientos del Sistema

- Requiere un mínimo de RAM y de espacio en disco, dependiendo del sistema operativo sobre el cual trabaje.

Número máximo de aplicaciones clientes conectadas a un servidor	No hay límite, aunque depende mucho de la capacidad de la computadora y el sistema operativo sobre el cual corra Interbase. Un sistema común puede manejar sin problema unos 150 clientes concurrentes.
---	--

Tamaño máximo de la base de datos	Interbase puede manejar archivos de hasta 2Gb en plataforma Windows 95 y de 4Gb en Windows NT y Unix. Dado que Interbase puede manejar muchos archivos por base de datos, este limite no se aplica para las bases de datos, así que puede manejar hasta varios Terabytes en una base de datos.
Número máximo de archivos por base de datos	Interbase puede manejar hasta 65536 archivos, pero este número es limitado por las capacidades de los sistemas operativos sobre los que corre.
Número máximo de bases de datos abiertas en una transacción	No hay restricción, aunque depende de los recursos del sistema.
Número máximo de tablas por base de datos	65536
Tamaño máximo por registro	64kb
Número máximo de registros y columnas (campos) por tabla	Número de registros: 4294967296 Número de columnas: depende del tipo de las columnas. El tamaño total de la suma de los tamaños en bytes de todas las columnas no debe pasar los 64Kb.
Número máximo de índices por base de datos	4294967296
Número máximo de eventos por procedimiento	No hay limite, sólo el impuesto por el tamaño del código de un procedimiento o un disparador (trigger).
Tamaño máximo de un procedimiento o un trigger	48Kb. de BLR, que es el código al cual son compilados los triggers y los procedimientos

2.2.4 Características

2.2.4.1 Servidor Activo

Interbase fue pionero en el concepto de servidor activo, el cual habilita a la base de datos para contener reglas de negocios y funcionar como un administrador central de accesos concurrentes de varios usuarios.

Las características de Interbase que soportan el concepto de servidor activo son:

- El analizador de SQL esta en el servidor.
- Los triggers del lado del servidor automatizan las reglas del negocio.
- Los stored procedures implementan en el servidor operaciones de bases de datos compartidas, modulares y seguras.
- Las funciones definidas por el usuario extienden el lenguaje SQL.
- Los eventos de alerta notifican a los clientes de cambios en la base de datos.
- El servidor mantiene la integridad de los datos de acuerdo a reglas de integridad referencial declarativas con operaciones en cascada.
- Los dominios y validaciones extienden los tipos SQL.
- El commit de dos fases estabiliza las transacciones distribuidas de múltiples bases de datos.

2.2.4.2 Escalabilidad y Portabilidad

Interbase implementa su interface con los desarrolladores y su tecnología a través de una variedad de plataformas manteniendo idénticas características. Provee interoperabilidad transparente en plataformas heterogéneas y pocas modificaciones cuando es necesario migrar las bases de datos.

Las plataformas soportadas por Interbase son las siguientes:

- Windows 95/98/NT/2000
- HP/UX

- Solaris
- SCO 5.0
- Red Hat Linux
- Novel Netware
- AT&T NCR
- IBM AIX
- DG-UX Avlion
- DEC- Unix
- Open/VMS

Interbase tiene pocos requerimientos de hardware lo que lo habilita para trabajar en plataformas modestas así como también en servidores de rango medio. Una instalación mínima requiere menos de 3 Mb de espacio en disco y una instalación completa, incluyendo ejemplos y documentación en línea, menos de 20 Mb. El servidor de Interbase administra correctamente hasta 150 clientes simultáneos.

2.2.4.3 Cliente/Servidor de dos capas

Los productos de Interbase incluyen una instalación cliente/servidor y una biblioteca cliente. El cliente y el servidor pueden comunicarse a través de los siguientes protocolos de red:

- TCP/IP (todas las plataformas del servidor).
- IPX/SPX (solamente el servidor para Novell NetWare).
- NetBEUI/Named pipes (solamente el servidor para Windows NT).

Cualquier plataforma cliente puede comunicarse con cualquier plataforma servidor en forma transparente. La biblioteca cliente provee un API para que las aplicaciones clientes generen los requerimientos y reciban los datos desde el servidor Interbase ejecutándose en NT, UNIX, o NETWARE. La biblioteca cliente implementa un protocolo de red especial, optimizado para comunicarse con el servicio Interbase remoto. Las

herramientas de Borland proveen acceso transparente a la biblioteca cliente a través del BDE.

Interbase para Windows puede ser ejecutado también como una ingeniería de base de datos local incluida dentro de una aplicación. Esta configuración es conocida como Local InterBase Server (LIBS). No utiliza protocolos de red sino que la biblioteca cliente se comunica con el servidor a través de procesos locales.

2.2.4.4 Correcciones para el año 2000

Interbase siempre ha almacenado e interpretado fechas correctamente. Interbase interpreta años de dos dígitos como siglo XX o siglo XXI dependiendo de la cercanía con la fecha actual. Por ejemplo, 1/9/32 es interpretado como 1/9/2032, porque este año esta a menos de 50 años de la fecha actual. Años de cuatro dígitos reemplazan esta interpretación. Una vez que es interpretado, Interbase siempre almacena e informa las fechas utilizando 4 dígitos para el año. Interbase almacena fechas desde el 1 de enero del año 100 DC hasta el 28 de febrero del año 32767 DC.

2.2.4.5 Tecnología

Interbase implementa el nivel de entrada del estándar SQL-92, así como también muchas características del nivel intermedio y características selectas del nivel completo. Interbase es miembro con voto en el comité para el estándar ANSI SQL, X3H2. Las características SQL más notables incluyen:

- Integridad referencial declarativa con reglas de operación en cascada.
- Roles SQL para proveer privilegios de seguridad para grupos de usuarios.
- Sintaxis de SQL-92 para cláusulas JOIN inner y outer.
- Vistas para tablas y JOINS.

2.2.4.6 Extensiones SQL

Características avanzadas de Interbase, tales como stored procedures, han propuesto el modelo para el estándar SQL-3 para este tipo de objetos de bases de datos.

Extensiones notables al SQL que fueron implementadas en Interbase incluyen:

- Stored procedures, modularizan operaciones de bases de datos complejas.
- Select procedures, retornan un conjunto de datos.
- Definiciones de domain, extienden los tipos básicos del SQL.
- Triggers, automatizan las reglas de negocio.
- Generators, implementan valores enteros auto-incrementales.

2.3 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE DELPHI 5 ENTERPRISE

Desde hace tiempo, Inprise/Borland ha gozado de una excelente reputación por sus poderosas herramientas de programación. Una de sus herramientas más innovadoras, Delphi, ha sido enriquecida para proporcionar un valioso soporte para la programación Web.

2.3.1 Herramientas de desarrollo

El ambiente de desarrollo integrado (IDE — Integrated Development Environment) incluye: el diseñador de formas, el inspector de objetos, la paleta de componentes, el explorador de código, el editor de código, el diseñador de módulos de datos, etc. En la figura 2.3.1 se observa la primera pantalla de un proyecto Delphi, al lado izquierdo se encuentra el inspector de objetos, en la parte superior la paleta de componentes y en el centro el diseñador de formas.

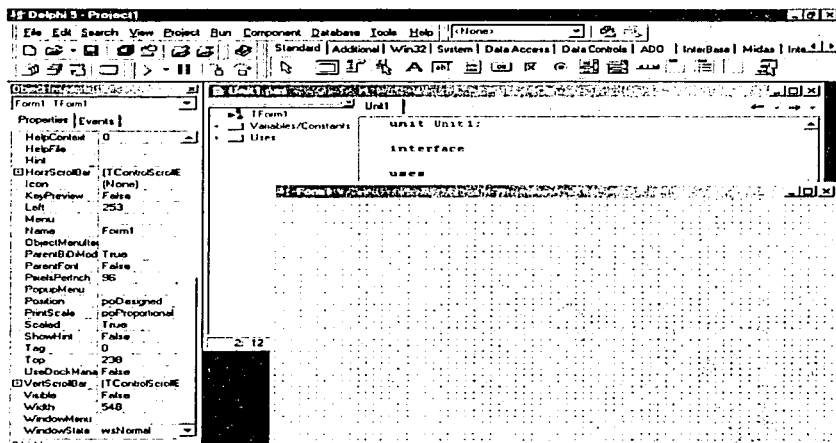


Figura 2.3.1 Pantalla principal de Delphi 5

TESIS CON
FALLA DE ORIGEN

2.3.2 Usando la Librería de Componentes Visuales

Delphi incluye componentes que forman parte de una jerarquía de clases llamada Biblioteca de Componentes Visuales (VCL — Visual Component Library). La VCL incluye objetos que son visibles en tiempo de ejecución (runtime) — tales como, controles de edición, botones y otros elementos de interfaz de usuario — así como también controles no visuales como datasets y cronómetros. En la figura 2.3.2 se muestra parte de la estructura de un componente VCL.

Los objetos que descienden de TComponent poseen propiedades y métodos que les permiten ser instalados en la paleta de componentes y ser agregados a las formas de Delphi.

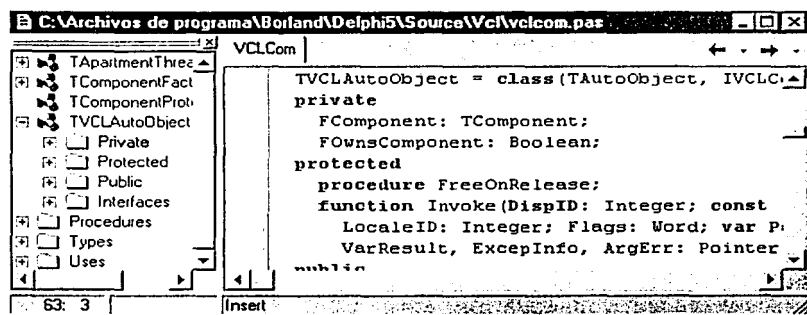


Figura 2.3.2 Parte de un componente VCL

Los componentes se encuentran altamente encapsulados. Por ejemplo, los botones están preprogramados para responder a los click's del mouse, al activar el evento *OnClick*. Si se utiliza un control de botones VCL, no se necesita escribir el código para manejar los mensajes de Windows cuando un botón es presionado; si no que solamente se tiene que programar la lógica de la aplicación que se ejecuta en respuesta al evento.

La mayoría de las versiones de Delphi incluyen el código completo de la VCL. A parte de suplementar la documentación en línea, el código de la VCL proporciona valiosos ejemplos de las técnicas de programación en Object Pascal.

2.3.3 Manejo de errores

El manejo de errores en Delphi está basado en excepciones, objetos especiales generados en respuesta a una entrada inesperada o a la ejecución de un programa con errores. Las excepciones pueden generarse tanto en tiempo de diseño como en tiempo de ejecución, y la VCL contiene muchas clases de excepciones que están asociadas con condiciones de error específicas. Las excepciones también son valiosas como herramienta de depuración pues, generalmente, la clase de una excepción entrega una idea acerca de qué la causó.

2.3.4 Conectividad a bases de datos y utilidades

Delphi y la VCL ofrecen una variedad de herramientas de conectividad que simplifican el desarrollo de aplicaciones de bases de datos. El Borland Database Engine (BDE) es un conjunto de controladores que soporta muchos formatos populares de bases de datos como se observa en la figura 2.3.3 entre los que se encuentran dBASE, Paradox, FoxPro, Access y cualquier fuente de datos ODBC. Los controladores Links SQL, disponibles en algunas versiones de Delphi, soportan servidores tales como Oracle, Sybase, Informix, DB2, SQL Server e InterBase.

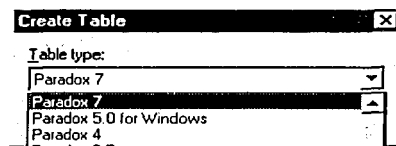


Figura 2.3.3 Formatos de base de datos soportados por Delphi

Delphi incluye los componentes InterBaseExpress (IBX) que pueden ser utilizados para acceder a las bases de datos Interbase. Las aplicaciones IBX accesan a características avanzadas de InterBase además de ofrecer la interfaz de más alto rendimiento.

El IBX se basa en la conocida arquitectura de componentes de Delphi para acceso a datos, y está integrado con el Data Module Designer. El IBX es compatible con los componentes data-aware de la librería de Delphi y no requiere del Borland Database Engine (BDE).

Algunas versiones de Delphi incluyen componentes para conectarse a bases de datos usando ActiveX Data Objects (ADO). ADO es la interfaz de alto nivel de Microsoft para cualquier fuente de datos, incluyendo bases de datos relacionales y no relacionales.

Además, Delphi incluye las siguientes herramientas para los desarrolladores de bases de datos:

2.3.4.1 Administrador BDE

Se usa el Administrador BDE para configurar los drivers BDE y establecer los alias usados por los controles VCL data-aware para conectarse a las bases de datos.

2.3.4.2 Explorador SQL

El explorador SQL permite consultar y editar bases de datos. Puede usarse para crear los alias de la base de datos, visualizar el diseño de la información, ejecutar las consultas SQL y mantener diccionarios de datos. En la figura 2.3.4 se observa la forma en que el explorador SQL muestra el contenido de una base de datos.

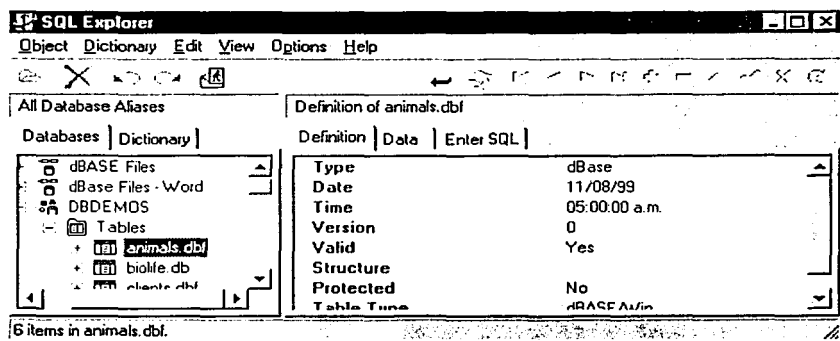


Figura 2.3.4 Visualización de una base de datos desde el explorador SQL

2.3.4.3 Database Desktop

El Database Desktop (figura 2.3.5) permite crear, visualizar y editar tablas de las bases de datos Paradox y dBase.

items	OrderNo	ItemNo	PartNo	Qty	Discount
1	1,003.00	1.00	1,313.00	5	0.00
2	1,004.00	1.00	1,313.00	10	50.00
3	1,004.00	2.00	12,310.00	10	0.00
4	1,004.00	3.00	3,316.00	8	0.00

Record 1 of 945

Figura 2.3.5 Edición de una base de datos desde el Database Desktop

2.3.4.4 Diccionario de datos

El diccionario de datos proporciona un área configurable de almacenaje, independiente de las aplicaciones, donde se pueden crear atributos de campos extendidos, que describen el contenido y la apariencia de los datos. El diccionario de datos puede localizarse en un servidor remoto para compartir la información.

2.3.5 Tipos de proyectos de desarrollo

Delphi puede ser utilizado para escribir:

- Aplicaciones GUI para Windows.
- Aplicaciones de consola.
- Aplicaciones de servicio.
- Dynamic-Link Libraries.
- Packages.

2.3.6 Aplicaciones y servidores

Delphi tiene características que facilitan la escritura de aplicaciones distribuidas, incluyendo cliente/servidor, multi-capas y sistemas basados en el Web. Además del soporte para estándares como COM algunas versiones de Delphi proporcionan herramientas para el desarrollo CORBA.

2.3.7 DLLs

Las DLLs (Dynamic-Link Libraries) son módulos compilados que contienen rutinas que pueden ser llamadas por aplicaciones u otras DLLs.

2.3.8 Componentes personalizados y paquetes

Un paquete es una DLL especial usada por las aplicaciones Delphi, el IDE o ambos. Aunque los paquetes pueden ser usados en una variedad de formas, su propósito más común es la encapsulación de componentes Delphi. De hecho, todos los componentes instalados en el IDE fueron compilados como paquetes.

Los componentes que vienen con Delphi son preinstalados en el IDE y ofrecen un rango de funcionalidad que debería ser suficiente para la mayoría de las necesidades de distribución. Los componentes configurables complementan la VCL, promoviendo la reutilización del código y la consistencia entre aplicaciones. Delphi provee un asistente que facilita la creación e instalación de los componentes.

2.3.9 Marcos

Un marco, al igual que una forma, es un contenedor para otros componentes. De algún modo, un marco es más parecido a un componente personalizado que a una forma. Los marcos pueden ser guardados en la paleta de componentes para facilitar su reutilización, y pueden ser anidados dentro de formas, marcos u otros objetos contenedores.

Luego de crear y guardar un marco, éste sigue funcionando como una unidad y heredando los cambios de los componentes (incluyendo otros marcos) que contiene.

Cuando un marco está dentro de otro marco o forma, sigue heredando los cambios realizados sobre el marco o la forma de la que deriva.

2.3.10 Objetos COM y ActiveX

Delphi soporta el estándar COM (Component Object Model) de Microsoft y proporciona asistentes para la fácil creación de controles ActiveX. En la carpeta ActiveX de la paleta de componentes, figura 2.3.6, se encuentran instalados una serie de controles ActiveX de muestra.

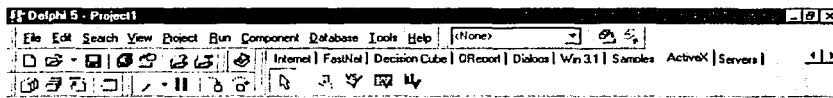


Figura 2.3.6 Componentes ActiveX.

2.3.11 Características del desarrollo para internet

Lado Cliente:

Aunque los controladores ActiveX son muy populares en algunos ambientes de desarrollo, no tiene sentido envolver un pequeño componente (o un componente de poca funcionalidad) en un control ActiveX. Pero sí hace sentido en Delphi envolver un componente de la Librería de Componentes Visuales (VCL) en un control ActiveX, en vez de un componente VCL nativo. A parte de esto, algunos componentes VCL no pueden ser usados como un componente padre base para ser derivados a un control ActiveX. Por ejemplo, los controles data-aware necesitan conectarse a un componente VCL TDataSource y no a una versión ActiveX de ese componente.

Un componente VCL de Delphi es un componente nativo escrito en ObjectPascal, que puede ser usado solamente en Delphi (y en C++Builder). Un control ActiveX es un componente VCL de Delphi. Una ActiveForm de Delphi puede contener muchos controles VCL; esto significa que es un control ActiveX que contiene muchos componentes-hijos VCL. Como tal, una ActiveForm puede contener mucha más

funcionalidad que un único control ActiveX. No todos los componentes VCL son aptos para transformarse en controles ActiveX.

Una ActiveForm es igual a un control ActiveX, la única diferencia radica en que luce como una forma de Delphi, en la cual podemos posicionar casi cualquier componente y por lo tanto construir una aplicación completa en un control ActiveX.

Lado Servidor:

Delphi 5 soporta varias soluciones del lado servidor. Tradicionalmente, éstas consisten en dos aproximaciones principales: CGI o ISAPI. Un script CGI -un ejecutable en Delphi- es cargado, ejecutado y descargado nuevamente de la memoria. Mientras que esto es conveniente para aplicaciones CGI pequeñas, puede transformarse en un peso para sistemas mayores, especialmente si una conexión a una base de datos debe establecerse repetidamente. por esta razón, tanto Microsoft como Netscape inventaron las soluciones llamadas ISAPI y NSAPI respectivamente. Y esta DLL es cargada sólo una vez en el espacio de memoria del servidor Web.

Desafortunadamente, la parte API de ISAPI y NSAPI, no es directamente compatible con un simple CGI ejecutable, y las conversiones de uno a otro pueden fácilmente introducir nuevos errores y problemas potenciales. La solución de Delphi, la tecnología WebBroker, integra el CGI, WinCGI y las soluciones ISAPI/NSAPI en un único módulo Web

Delphi 5 también incluye soporte para ASP. Este consiste en un asistente para objetos ASP especiales y la generación de plantillas. El resultado, como se observa en un navegador, puede ser bastante similar a lo que podemos lograr usando el WebBroker.

2.3.12 Multicapas: MIDAS

Las ActiveForms tienden a ser realmente pesadas. No sólo por el tamaño de los archivos, sino por las DLLs y drivers requeridos para mostrar las tablas de las bases de datos. Esto puede solucionarse con MIDAS de Borland (Middle Tier Database Architecture System) -la capa de middleware para Delphi y C++Builder (y JBuilder con JMIDAS).

Al usar MIDAS, se puede dividir una aplicación en dos partes: un cliente (GUI) y un servidor middleware, el que resulta estar frecuente y directamente conectado al manejador de bases de datos (como Oracle, Interbase o ADO). En esta arquitectura, el cliente no tiene nada de código relacionado con bases de datos, y se conecta al servidor utilizando un dataset cliente y un componente de conexión. Éste último viene en distintas opciones: CORBA, DCOM, sockets (TCP/IP) y HTTP, que permite poner el servidor detrás de un proxy o un firewall.

Visualmente no hay ningún cambio, pues los datasets pesados son reemplazados por datasets cliente y conexiones al servidor. Los módulos de datos pueden ser convertidos en módulos de datos remotos para operar en el lado servidor. Las restricciones y los parámetros pueden ser pasados desde el cliente al servidor, y viceversa. Al desconectarse del servidor, los datos del cliente pueden ser almacenados localmente. Esto permite que el usuario final trabaje localmente.

2.3.13 InternetExpress

Esta nueva funcionalidad combina el poder del WebBroker con la escalabilidad de MIDAS, para producir páginas Web dinámicas usando Dynamic HTML con JavaScript y paquetes de datos XML. Los paquetes de datos XML están conectados a los controles HTML a través del código JavaScript, lo que le da a estas páginas Web una apariencia muy interactiva. (Hay navegadores activos, por ejemplo, los datos en los controles se actualizan sin la necesidad de actualizar la página Web completa, así que, no más parpadeos ni demoras al visualizar tablas dentro de un navegador).

InternetExpress extiende el WebBroker con un nuevo componente: MidasPageProducer, y a MIDAS con un nuevo componente: XMLBroker, combinando estas técnicas en una. El hecho de que InternetExpress use XML como su formato de paquete de datos, permite generar aplicaciones distribuidas multiplataforma (Windows, Linux) y multilenguaje (Delphi, C++Builder).

2.4 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE APACHE 1.3.20.

Apache era inicialmente una serie de parches al servidor WWW de NCSA conocido como httpd (principios de 1995). Fue un proyecto que atrajo a mucha gente por el gran interés de su objetivo: lograr el servidor web más rápido, más eficiente y con mayor funcionalidad desde el enfoque del software libre.

Con un enorme equipo de voluntarios a lo largo y ancho de toda la red, se han logrado estos objetivos, logrando batir a compañías comerciales de la talla de Microsoft y Netscape. Y no sólo se ha logrado batir a los servidores web de grandes compañías, se ha logrado atraer la atención de IBM que está apoyando tanto a nivel de soporte como de desarrollo.

2.4.1 Características

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Entre sus características destacan:

- Multiplataforma.
- Es un servidor de web conforme al protocolo HTTP/1.1.
- Modular: puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

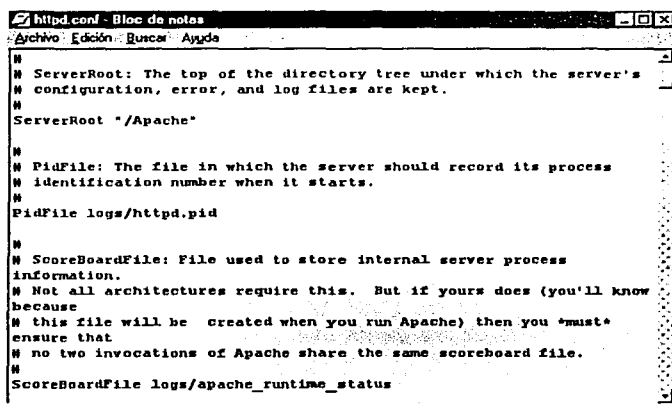
2.4.2 Configuración

Los archivos de configuración son los siguientes:

- **httpd.conf:** Archivo principal de configuración de Apache.

- **srm.conf**: Archivo de definición del espacio de nombres. En este archivo también se especifica donde se encuentran los cgi-bin, los iconos, el tipo de documento por defecto, como se responde ante los errores, que fichero es el índice dentro de un directorio, donde está la página personal de los usuarios del sistema.
- **access.conf**: Archivo de control de acceso global a los datos del servidor de web. En el se especifican los permisos de accesos a directorios, archivos y URLs dentro del servidor, así como diferentes configuraciones.
- **Mime.types**: archivo de control de los tipos MIME que son enviados al cliente en función de la extensión del archivo.

El archivo principal es `httpd.conf`, ver figura 2.4.1, por lo que a continuación se detalla su configuración. Los demás archivos son de apoyo y su comprensión es más sencilla.



```
httpd.conf - Bloc de notes
Archivo: Edición: Buscar: Ayuda

#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
ServerRoot "/Apache"

#
# PidFile: The file in which the server should record its process
# identification number when it starts.
#
PidFile logs/httpd.pid

#
# ScoreBoardFile: File used to store internal server process
# information.
# Not all architectures require this.  But if yours does (you'll know
# because
# this file will be  created when you run Apache) then you *must*
# ensure that
# no two invocations of Apache share the same scoreboard file.
#
ScoreBoardFile logs/apache_runtime_status
```

Figura 2.4.1. Fragmento del archivo de configuración `httpd.conf`

2.4.2.1 El archivo de configuración `httpd.conf`

Es el archivo de configuración principal de Apache. Junto con la distribución, viene un ejemplo completo de las características que se controlan desde este archivo. Se

muestran a continuación las principales opciones de dicho archivo, que el usuario debe editar para adaptarlas a su situación concreta:

- **ServerType: standalone o inetd:** El servidor de web puede ser arrancado, bien como un demonio individual, o bien como un demonio más dentro del conjunto de demonios que arranca inetd. Dadas las características de un servidor de web, que suele recibir peticiones de forma constante, es mejor arrancarlo de forma independiente, y no desde el inetd.
- **Port 80:** Puerto desde el que escucha el servidor de web. Un usuario del sistema, sin derechos de superusuario, solo puede utilizar puertos por encima del 1024.
- **HostnameLookups on u off:** Dentro de los logs del servidor, se intenta traducir la dirección IP del cliente que ha hecho la petición a un nombre utilizando DNS o no. El intento de traducción lleva asociada unas comunicaciones que pueden reducir el rendimiento del servidor.
- **User www-data Group #-1:** Identidad del usuario bajo el que se ejecuta el servidor. Si es diferente a la del usuario que lanza el demonio, el usuario que lanza el demonio debe de ser root.
- **ServerAdmin acs@ieeesb.etsit.upm.es:** Dirección de correo del usuario responsable de la administración del servidor de web, y al que se enviarán mensajes en caso de errores.
- **ServerRoot /usr/local/apache:** Directorio donde se encuentra la configuración del servidor, los archivos de bitácora y los archivos de error.
- **#BindAddress *:** Con Apache se puede hacer que el servidor escuche en varias direcciones IP. Con esta opción se le dice al servidor que direcciones IP debe escuchar.
- **ErrorLog logs/error_log:** archivo donde se almacenan los errores, como peticiones de usuarios de archivos inexistentes, o peticiones que no se pueden reconocer. El directorio es relativo a ServerRoot.

- **TransferLog logs/access_log:** Archivo donde se almacenan las transferencias de datos con los clientes. El directorio es relativo a ServerRoot. Este archivo es fundamental para sacar estadísticas de acceso.
- **PidFile logs/httpd.pid:** Archivo donde se almacena el pid (identificador del proceso del servidor). El directorio es relativo a ServerRoot.
- **ScoreBoardFile logs/apache_status:** Archivo que almacena información acerca del estado del servidor de web, estado que se actualiza en tiempo real, y que permite analizar a los administradores autorizados lo que está ocurriendo en el interior del servidor, como las peticiones actuales en curso y los recursos que está consumiendo.
- **ServerName www.ieeesb.etsit.upm.es:** Este es el nombre que el servidor envía a los clientes, en caso de que los clientes hayan accedido a la máquina con un nombre diferente. Una máquina puede tener varios nombres, y por cualquiera de ellos se puede acceder al puerto 80, donde suele estar el servidor de web. Pero normalmente, hay un nombre preferido para el web, algo como www.dominio. El servidor puede indicar al cliente que en las peticiones futuras, debe utilizar ese nombre. El nombre ha de ser un nombre real registrado dentro del sistema DNS.
- **Timeout 300:** El número de segundos tras el cual se envía o se recibe el fin de plazo de una petición.
- **KeepAlive On:** Para permitir la existencia de conexiones persistentes (HTTP 1.1), es decir, de que por una misma conexión se puedan enviar varias peticiones HTTP.
- **MaxKeepAliveRequests 100:** Número máximo de peticiones que se pueden cursar por una misma conexión. En caso de poner 0, el número será ilimitado. Cuantas más peticiones se puedan cursar por una misma conexión, mejor será el rendimiento de la comunicación, al evitar establecer nuevas conexiones TCP, que son muy costosas.
- **KeepAliveTimeout 15:** Número de segundos máximos en los que se esperará la siguiente petición.

- **MinSpareServers 5 - MaxSpareServers 10:** Número mínimo y máximo de procesos que serán lanzados en la máquina. Apache de forma dinámica lanzará un número entre los dos límites, dependiendo de la carga de las peticiones recibidas. Esta parte va a sufrir un importante cambio con la aparición de threads en la versión 2.0 de Apache.
- **StartServers 5:** Número de procesos que se lanzan inicialmente.
- **MaxClients 150:** Número máximo de clientes que pueden conectarse al servidor de web. Debe ser un número relativamente alto, ya que una vez superado, ningún cliente más podrá acceder a la información. Es un método de protección, que evita que el sistema pueda quedar totalmente bloqueado.
- **MaxRequestsPerChild 30:** número máximo de peticiones que puede servir un proceso hijo del proceso servidor principal, antes de que sea eliminado. Esto es necesario en sistemas con agujeros en la gestión de memoria, como por ejemplo en algunas librerías de Solaris.
- **#ProxyRequests On:** Si queremos que nuestro servidor de web haga también la función de servidor proxy, es decir, que guarde un caché con las páginas más accedidas por los clientes, podemos activar esta funcionalidad con dicha opción.
 - #CacheRoot /usr/local/apache/proxy
 - #CacheSize 5
 - #CacheGcInterval 4
 - #CacheMaxExpire 24
 - #CacheLastModifiedFactor 0.1
 - #CacheDefaultExpire 1
 - #NoCache www.sun.com

En caso de que se haya habilitado esta opción se deben especificar las siguientes opciones sobre el caché: tamaño del caché, periodo en el que expirarán los documentos y de que URL's no se debe hacer caché.

- #Listen 3000

- o #Listen 12.34.56.78:80

Permite obligar al servidor de web a que escuche en una dirección IP determinada, junto con el puerto TCP. También escuchará el servidor en la dirección y puerto por defecto.

- o #<VirtualHost host.some_domain.com>
- o #ServerAdmin webmaster@host.some_domain.com
- o #DocumentRoot /www/docs/host.some_domain.com
- o #ServerName host.some_domain.com
- o #ErrorLog logs/host.some_domain.com-error_log
- o #TransferLog logs/host.some_domain.com-access_log #<VirtualHost>

Una de las características más destacadas en Apache es que puede escuchar las peticiones para diferentes direcciones IP. Con ello, se puede centralizar el servicio de diferentes dominios web con direcciones IP diferentes, en un único servidor de web (hosting de varios dominios). En Linux esta característica se puede implementar compilando el núcleo con soporte para alias IP, lo que permite definir en una única tarjeta de red, varias direcciones IP. De este modo, nuestra tarjeta de red aceptará paquetes con diferentes direcciones IP de destino, peticiones que se gestionarán de forma diferente por el servidor de web, según a que dirección IP o nombre DNS, vayan dirigidas. Cada servidor virtual tiene su propia estructura de direcciones y su configuración.

2.4.3 Proyectos asociados

Hay muchos proyectos asociados a Apache cuyo objetivo es aumentar su funcionalidad. De los cuales destacan dos de ellos: uno muy útil para los desarrolladores, conocido como PHP y otro orientado a la privacidad de las comunicaciones Apache-SSL.

2.4.3.1 PHP

En la programación de aplicaciones en Internet, es importante las herramientas de programación que se utilizan tanto en el lado del cliente, como del servidor.

En el servidor de Web Apache, un lenguaje de programación que se puede utilizar en las páginas HTML se conoce como PHP. Es un lenguaje similar a Perl y muy sencillo de utilizar. Es un lenguaje en constante desarrollo y cuya principal característica es que proporciona una librería de funciones que permite acceder a las principales bases de datos del mercado (Adabas, Ilustra de Informix, Oracle, MySQL, PostgreSQL, Interbase entre muchas otras), lo que facilita mucho la integración del Web con el mundo de las bases de datos.

2.4.3.2 Apache-SSL

El comercio electrónico es el campo que más fuerza esta tomando y es uno de los sectores con mayores perspectivas de futuro. En la actualidad, el único freno al comercio electrónico en Internet es la seguridad o la falta de ella.

Debido al gran interés que hay, se están desarrollando rápidamente estándares que brinden la seguridad a Internet, en especial, dentro del Web. Apache-SSL es el servidor Apache pero con nuevas características de seguridad, como son la encriptación y la autenticación. Lo que más puede sorprender al desarrollador, es que dichos parches son también de libre distribución, utilizándose una librería de encriptación (RSA, DES, MD5) de libre distribución.

El mundo de la encriptación está lleno de connotaciones militares, y depende mucho del país en el que nos encontremos, de las utilidades de encriptación que podamos utilizar. En México no hay ningún problema en utilizar por ejemplo, la encriptación RSA de 128 bits, pero en el caso de utilizar dicha encriptación en comunicaciones internacionales, las cosas se complican mucho.

2.5 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE LOS CGI'S

Cuando el World Wide Web inició su funcionamiento como lo conocemos, empezando a tomar popularidad aproximadamente en 1993, solo se podía apreciar texto, imágenes y enlaces. La introducción de Plugins en los navegadores permitió mayor interactividad entre el usuario y el cliente, aunque estaba limitado por la velocidad y la necesidad de tener que bajar e instalar cada plugin que se necesitara, por lo que estos se desarrollaron mayormente en áreas de vídeo, audio y realidad virtual.

El CGI (Por sus siglas en inglés "Common Gateway Interface") cambió la forma de manipular información en el web. En sí, es un método para la transmisión de información hacia un compilador instalado en el servidor. Su función principal es la de añadir una mayor interacción a los documentos web que por medio del HTML se presentan de forma estática.

El CGI es utilizado comúnmente para contadores, bases de datos, motores de búsqueda, formularios, generadores de email automático, foros de discusión, chats, comercio electrónico, juegos en línea y otros.

Esta tecnología tiene la ventaja de correr en el lado del servidor cuando el usuario lo solicita por lo que es dependiente del servidor y no de la computadora del usuario.

De acuerdo a la traducción de la NCSA: "Un documento HTML es estático, lo que significa que existe en un estado constante; es un archivo de texto que no cambia. Un CGI por otro lado, es ejecutado en tiempo real, lo que permite que regrese información dinámica. Por ejemplo, digamos que se quiere conectar una base de datos al World Wide Web para permitir que las personas de todo el mundo la manipulen. Básicamente, lo que se debe hacer es crear un CGI que será ejecutado por el servidor para transmitir información al motor de la base de datos, recibir los resultados y mostrárselos al cliente. Este es un ejemplo que muestra los orígenes del CGI.

Los CGI's pueden estar compilados en diferentes lenguajes de programación. El más popular para el desarrollo de contenidos Web es el lenguaje Perl de distribución gratuita, aunque también podemos mencionar a: C, C++, Delphi y Java. El funcionamiento de esta tecnología es muy sencillo. Como lo muestra la figura 2.5.1 los CGI's residen en el servidor, donde son llamados, ejecutados y regresan información de vuelta al usuario.

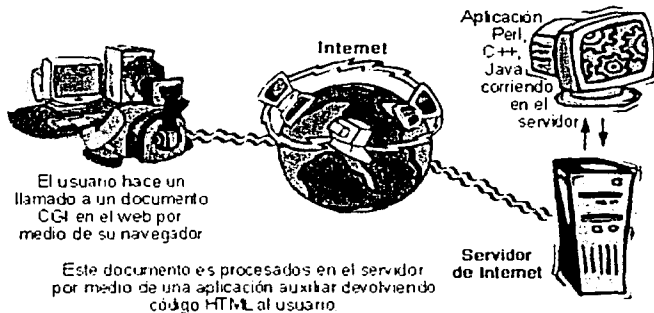


Figura 2.5.1 Ciclo de vida de los CGI's

Un ejemplo de un llamado a un CGI puede ser:

<http://www.cecafi.unam.mx/cgi-bin/ejemplo.cgi>

Hay que tomar muy en cuenta que existen dos formas de enviar información; siendo estas por medio de GET y POST.

La primera envía la información dentro del URL separando el URL de la información por medio de un signo de interrogación, por ejemplo:

<http://www.cecafi.unam.mx/cgi-bin/ejemplo.cgi?nombre=Fernando&apellido=Borja>

Esta forma presenta varias desventajas, ya que no permite manejar grandes cantidades de información y por ser una dirección, los caches de los navegadores pueden guardar

los resultados, lo que dificulta el poder acceder nuevamente a la aplicación, pero simplifica el uso de los CGIs con solo utilizar una dirección URL.

La otra forma de enviar la información es por medio de POST, el cual envía la información directamente al servidor, haciéndolo invisible en el URL y permitiendo cualquier cantidad de información, como el envío de archivos.

2.6 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE HTML

HTML significa Lenguaje de Marcado de Hipertexto. HTML se basa en el SGML (Lenguaje Estándar de Marcado Generalizado), un sistema mucho más grande de procesamiento de documentos. Para escribir páginas HTML no se necesita conocer mucho de SGML, pero sí ayudará saber que una de las principales características de este lenguaje es que describe la estructura general del contenido de los documentos y no el aspecto en sí de la página.

2.6.1 HTML describe la estructura de la página

En virtud de su herencia de SGML, HTML es un lenguaje que describe la estructura del documento, no su presentación real. La idea de esto es que la mayoría de los documentos tengan elementos en común; por ejemplo títulos, párrafos y listas (ver figura 2.6.1).

HTML define una serie de estilos comunes en las páginas Web: encabezados, párrafos, listas y tablas. También define formatos de carácter, como negritas y ejemplos de código. Cada elemento tiene nombre y está contenido en lo que se llama etiqueta. Cuando se escribe una página en HTML se marca cada elemento de la página con una etiqueta que dice "esto es un encabezado" o "esto es un elemento de una lista".

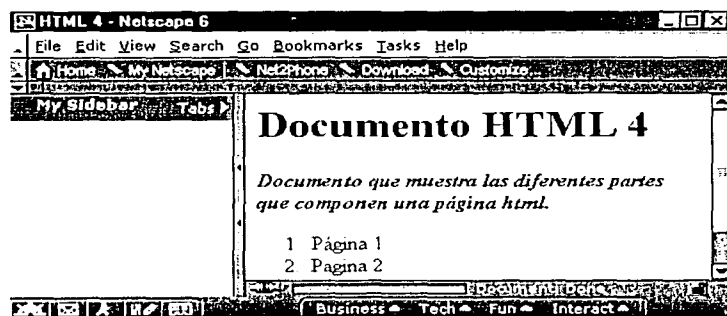


Figura 2.6.1 Documento HTML visualizado en Netscape 6.

2.6.2 HTML no describe el diseño de la página

Cuando se trabaja con un procesador de texto o un programa de diseño de páginas, los estilos no son sólo elementos de la página, sino que también cuentan con información del formato, como el tamaño de letra y estilo, la sangría, el subrayado, etcétera. Así, cuando se escribe un texto que va a ser el encabezado, se le aplica el estilo de encabezado y el programa automáticamente lo formatea con el estilo correcto.

HTML no llega a tanto, en general, HTML no dice nada acerca del aspecto de la página. Las etiquetas HTML sólo indican que determinado elemento es encabezado o lista; no indican nada acerca de cómo deberán formarse. Lo único que tiene que hacer el programador es marcar la sección que su supone será el encabezado.

Los navegadores funcionan como formateadores HTML. Cuando se lee una página HTML dentro de un navegador, como Netscape o Internet Explorer, éste lee o analiza las etiquetas HTML y formatea el texto e imágenes en la pantalla. El navegador puede hacer mapeos entre el nombre de los elementos de página y el estilo real en la pantalla; por ejemplo, los encabezados irán en un tipo de letra más grande que el resto de la página. El navegador también ajusta todo el texto para que quepa en el ancho de la ventana.

Cada navegador, en diferentes plataformas, tiene sus propios mapeos para cada elemento de página. Algunos navegadores usan diferentes estilos de letra. Así, por ejemplo, un navegador puede desplegar las cursivas como tales, en tanto que otro quizá las despliegue como texto inverso o subrayado por que el sistema carece de cursivas. La información real y los vínculos dentro de las páginas seguirán estando allí; pero variará el aspecto en la pantalla. La figura 2.6.2 muestra la misma página de la figura 2.6.1 pero visualizada en Internet Explorer.

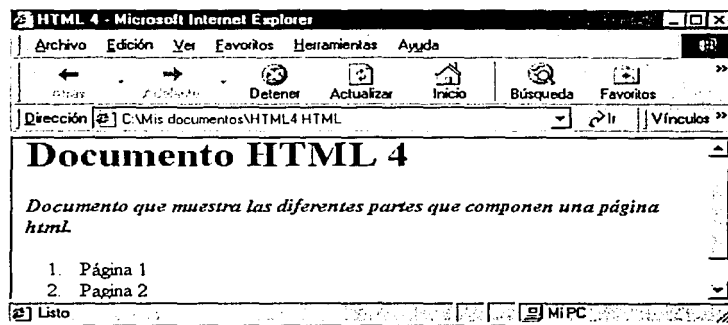


Figura 2.6.2 Documento HTML visualizado en Internet Explorer 5.5.

2.6.3 Por qué funciona así

Si el objetivo es que las páginas sean legibles para cualquier usuario se debe tomar en cuenta que los lectores no tendrán el mismo sistema operativo, las pantallas del mismo tamaño, el mismo número de colores ni los mismo tamaños de letra. HTML toma en cuenta todas estas diferencias y permite a todos los navegadores estar al mismo nivel.

Las reglas más importantes del diseño de páginas HTML son las siguientes:

- No diseñar páginas según como se vean en su computadora y en su navegador.
- Diseñar páginas para que funcionen en la mayoría de los navegadores.
- Definir un contenido claro y bien estructurado, que sea fácil de leer y de entender.

2.6.4 HTML es un lenguaje de marcado

Escribir en un lenguaje de marcado significa colocar etiquetas en ciertas palabras y párrafos.

Las etiquetas señalan partes específicas de la página y producen efectos diferentes en el navegador.

HTML tiene un conjunto definido de etiquetas, por lo que no se pueden inventar las propias para crear nuevas apariencias o características; cada navegador tiene soporte para diferentes conjuntos de etiquetas.

El conjunto básico de etiquetas se llama HTML 2.0, el cual es la norma antigua de HTML y constituye el conjunto de etiquetas que debe soportar todo navegador.

La versión HTML 4.0 tiene muchas funciones nuevas que dan mayor flexibilidad para el diseño de páginas con respecto a la versión HTML 2.0.

Actualmente, HTML 4.0 incluye algunos de los elementos que han surgido de la guerra de características entre los dos principales navegadores, Netscape Navigator y Microsoft Internet Explorer; tiene soporte para hojas de estilo en cascada, ubicación absoluta, elaboración de scripts y marcos, así como muchas otras características nuevas. Incluso algunas características de HTML 4.0 aún no cuentan con soporte de ningún navegador.

HTML es un lenguaje de marcado especialmente pequeño y fácil de aprender; mucho más pequeño que otros, como el PostScript y el troff de UNIX. Esos lenguajes son tan grandes y complejos, que aprender lo suficiente para escribir los documentos más sencillos tomaría años, en cambio con HTML se puede empezar de inmediato.

2.6.5 Cómo son los archivos HTML

Las páginas escritas en HTML son archivos de texto simple (ASCII), lo cual significa que no contienen información específica de ningún programa o plataforma. Puede leerlos cualquier editor que soporte texto (que es lo mínimo que debe hacer cualquier editor). Los archivos HTML contienen lo siguiente:

- El texto de la página en sí.
- Las etiquetas HTML que indican los elementos de la página, la estructura, el formato y los vínculos de hipertexto con otras páginas.

La mayoría de las etiquetas HTML se ven del siguiente modo:

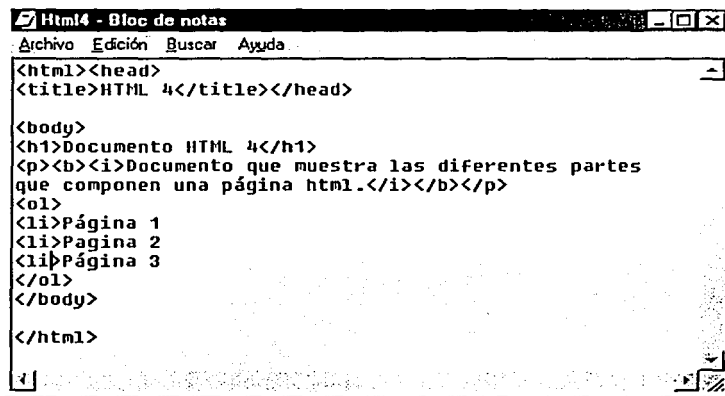
<NombreDeLaEtiqueta> texto afectado </NombreDeLaEtiqueta>

El nombre de la etiqueta está encerrado entre paréntesis angulares (<>).

Las etiquetas HTML encierran el texto afectado, la etiqueta inicial activa la característica (como encabezado, negrita, etcétera) y la de cierre la desactiva. En la etiqueta de cierre, el nombre va precedido de una diagonal (/).

No todas las etiquetas tienen inicio y cierre. Algunas son de un solo lado e incluso otras son contenedores que tienen información adicional y texto dentro de los paréntesis angulares.

Las etiquetas HTML no hacen diferencia entre mayúsculas y minúsculas; pueden especificarse en mayúsculas, minúsculas o cualquier combinación. Se recomienda utilizar mayúsculas para diferenciarlas mejor del resto del texto. En la figura 2.6.3. se presenta el documento HTML que se utilizó en la pantallas de las figuras 2.6.1. y 2.6.2.



```
<html><head>
<title>HTML 4</title></head>

<body>
<h1>Documento HTML 4</h1>
<p><b><i>Documento que muestra las diferentes partes
que componen una página html.</i></b></p>
<ol>
<li>Página 1
<li>Pagina 2
<li>Página 3
</ol>
</body>

</html>
```

Figura 2.6.3 Documento HTML con etiquetas en minúsculas.

2.6.6 Programas que ayudan a escribir HTML

Existen muchos programas freeware y shareware para editar archivos HTML. La mayoría de estos programas son, en esencia, editores de texto con menús y botones adicionales que insertan las etiquetas HTML correspondientes en el texto. Los editores de texto HTML son buenos por dos razones: no se tiene que recordar el nombre de todas las etiquetas y no se tiene que dedicar tiempo para teclearlas.

Además de los editores HTML, también pueden usarse convertidores, los cuales toman archivos de los procesadores de palabras más utilizados y los convierten a HTML. Con un sencillo juego de plantillas podrá escribir todas las páginas en el procesador de su preferencia y convertir el resultado cuando haya terminado.

En muchos casos, los convertidores llegan a ser útiles, en especial cuando se necesita publicar documentos ya hechos de la manera más rápida posible. Sin embargo, los convertidores sufren del mismo problema que los editores: el resultado puede variar de un navegador a otro, y no cuentan con muchas de las características nuevas o avanzadas. Asimismo, la mayoría de los programas de conversión son bastante

limitados, no necesariamente por sus propias características, sino por las limitaciones del mismo HTML.

CAPÍTULO III

3.1 PROBLEMÁTICA ACTUAL

Una institución educativa tiene como tarea primordial ofrecer educación a sus usuarios, que en este caso son toda la matrícula de alumnos inscritos. Pero también es cierto que para cumplir con dicha tarea las instituciones educativas deben contar con mecanismos alternos que le permitan de manera más fácil y eficiente brindar sus servicios.

Lo que se percibe hoy en día es que las instituciones educativas, en su mayoría, no han logrado consolidar una reestructuración seria que sea capaz de confrontar la problemática tanto de la educación que brindan como la de los procesos administrativos que controlan su funcionamiento. En este sentido las instituciones educativas aún se encuentran apegadas a procesos poco eficientes para poder brindar un servicio de calidad que se vea reflejado en todos los ámbitos de sus actividades diarias.

Actualmente nos encontramos con que muchos de los trámites y procesos administrativos, para cualquier tipo de evento son realizados de manera lenta y poco eficiente ya que no se cuenta con procesos sistematizados que los lleven a cabo.

Las instituciones educativas deben tener la flexibilidad de adaptarse a nuevos ambientes de servicio que faciliten su trabajo, no sólo en el campo educativo sino también en lo que respecta al campo administrativo, en cualquier circunstancia y por cualquier medio.

Existen muchas instituciones educativas que no cuentan actualmente con un mecanismo automatizado para la administración de eventos, por el contrario, este proceso se realiza de forma manual y con tramites demasiado tardados lo que lo hace poco confiable e ineficiente (ver figura 3.1.1). Al ser procesos realizados en

forma manual siempre hay riesgos tales como: traspapeleo, recopilación de datos erróneo, etc.

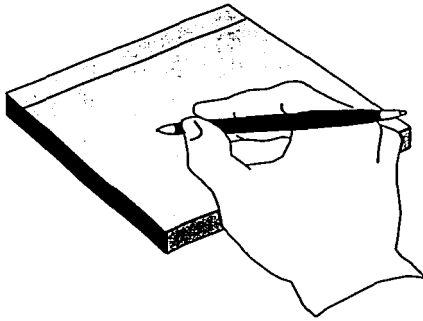


Figura 3.1.1 Proceso manual.

La falta de un sistema que contenga información automatizada y actualizada es causa directa de que el proceso sea lento, problemático y poco eficiente. Para que los recursos de una institución educativa puedan ser administrados de forma eficaz es necesario que la programación de eventos se controle y administre plenamente.

Conforme crece la demanda en los servicios que presta una institución educativa crece el trabajo administrativo y por lo tanto la cantidad de información la cual por lo general se encuentra plasmada en hojas, carpetas y relaciones que de alguna manera ocupan espacio y pueden ocasionar como se mencionó traspapeleo y por consiguiente pérdida de información.

Lo que se pretende es desarrollar un sistema que sea capaz de llevar a cabo una adecuada administración de los eventos que se realizan dentro de una institución educativa, para lo cual se propone:

- El desarrollo de un sistema que permita eficazmente la administración de eventos.

- Un sistema que cuente con una base de datos de gran capacidad de almacenamiento para el manejo de la información de cada uno de los eventos.
- Una aplicación que tenga interfaces de captura, búsqueda, reportes y catálogos de eventos; las cuales deberán ser amigables y de fácil manejo.
- Reducir el tiempo que dura actualmente la programación de eventos debido a que no se cuenta con un mecanismo automatizado.

Con la implantación de un sistema de estas características se espera optimizar la asignación de los recursos que los usuarios requieren para la realización de un evento dentro de las instalaciones de cualquier institución educativa y así evitar en lo posible las inconsistencias del proceso manual (ver figura 3.1.2).

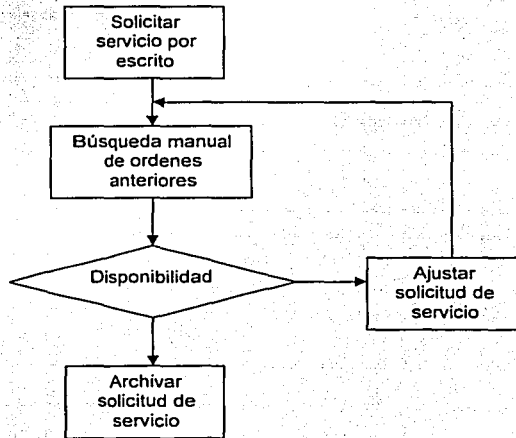


Figura 3.1.2 Diagrama a bloques del proceso manual.

Una de las ventajas que va a ofrecer el sistema es que todos los usuarios podrán a través de internet dar de alta un nuevo evento, consultar los existentes y generar reportes de los mismos (ver figura 3.1.3).



Figura 3.1.3 Acceso vía internet.

Esto reducirá en forma efectiva el tiempo en que se realiza el proceso de asignación, mejorando de manera sustancial el control de eventos.

3.2 NECESIDADES DE LA DEPENDENCIA

La manera como se lleva a cabo la administración de eventos es la siguiente:

- El organizador elige fecha y sala para el evento.
- Manda un e-mail, habla por teléfono, acude personalmente o manda fax solicitando el registro del evento.
- Se guarda la información recibida.
- Se pasa la información a una hoja de Excel.
- El administrador avisa al organizador que su evento ha sido registrado.
- El día del evento el organizador llena una forma para el control administrativo.
- Los datos obtenidos el día del evento se pasan a una hoja de Excel.
- Se realiza un informe del evento.

Principales problemas con la forma de trabajar antes descrita:

- El proceso es lento ya que se tiene que pasar la información a una hoja de Excel.
- Al llenar formatos administrativos mas de una vez se realiza doble trabajo dando como resultado que la información se tenga por duplicado.
- Los diferentes trámites que existen durante el proceso administrativo se ven afectados por el exceso de normas y de papeleo los cuales complican o retrasan la organización de los eventos.

Con el fin de detectar las necesidades del personal involucrado en la administración de eventos, se llevo a cabo una encuesta la cual generó los siguientes requerimientos:

1. Contar con tres tipos de usuario:

- Usuario de Lectura
- Usuario de Escritura

- Usuario de Lectura/Escritura (Administrador)

El usuario de lectura solo podrá consultar información de los eventos, el usuario de escritura podrá agregar, borrar, consultar o modificar eventos y por último el usuario administrador tendrá los privilegios de lectura/escritura.

2. Contar con una pantalla de acceso al sistema, ésta deberá solicitar tanto el nombre de usuario como su clave de acceso, en caso de que los datos sean válidos se mostrará el menú correspondiente al perfil de usuario (lectura, escritura y lectura/escritura), en caso contrario se negará el acceso a la aplicación.

3. Contar con un menú que muestre las siguientes categorías y subcategorías:

- Eventos
 - Nuevo
 - Listado
- Consultas
 - General
 - Disponibles
 - Eventos
- Reportes
 - General
 - Alimentos
 - Por Lugar
 - Número de Subevento
 - Uso de Salas
 - Uso de Equipo
- Catálogos
 - Salas
 - Listado

- Agregar
- Equipo
 - Listado
 - Agregar
- Alimentos
 - Listado
 - Agregar
- Accesorios
 - Listado
 - Agregar
- Montajes
 - Listado
 - Agregar
- Usuarios
 - Listado
 - Agregar
- Divisiones
 - Listado
 - Agregar
- Niveles
 - Listado
 - Agregar

4. La pantalla de eventos deberá permitir agregar, borrar, consultar o modificar eventos. Los datos con que debe contar son:

EVENTO

- Responsable del Evento
 - Nombre
 - Número de Cuenta
 - Área

- E-mail
- Teléfono
- Evento
 - Semestre
 - Nombre
 - Tipo
 - Número de personas
 - Comentarios

FECHA

- Sala
- Fecha
- Hora de Inicio
- Hora de Fin
- Montaje
- Descripción
- Seguridad

EQUIPO

- Tipo
- Cantidad

ALIMENTOS

- Tipo
- Hora de Inicio
- Hora de Fin
- Comentarios

ACCESORIOS

- Tipo
- Comentarios

5. La pantalla de consultas deberá incluir opciones de:

- Búsqueda de eventos por sala, fecha, número de folio y nombre.
- Búsqueda de eventos disponibles por sala, fecha y hora.

Esta interface mostrará los resultados de la consulta y permitirá la impresión de los mismos.

6. La pantalla de reportes generará informes de:

- Eventos por fecha.
- Alimentos por fecha.
- Eventos por sala y fecha.
- Eventos por semestre y folio.
- Uso de salas por semestre, sala y fecha.
- Uso de equipo por semestre, equipo y fecha.

Esta interface también mostrará los resultados del reporte y permitirá la impresión de los mismos.

7. La pantalla de catálogos deberá permitir agregar, borrar, consultar o modificar:

- Salas.
- Equipo.
- Alimentos.
- Accesorios.
- Montajes.
- Usuarios.
- Divisiones.
- Niveles.

3.3 IDENTIFICACIÓN DEL PROBLEMA

El principal problema de la administración de eventos es la lentitud en los procesos involucrados en esta actividad. Actualmente existen dos procesos para la inscripción de eventos: registro vía Internet (e-mail) y registro por algún otro medio (teléfono, fax), ambas formas de inscripción son capturadas en una hoja de Excel, la cual después es procesada para la obtención de consultas, reportes, etc.

Otro problema en la organización de eventos es la duplicación de información, lo que implica doble trabajo. No existe hoy en día ningún proceso que automatice la mayor parte de las actividades administrativas de los eventos.

Por ejemplo, cuando se recibe un correo electrónico o fax para el registro de un evento se tiene que copiar dato por dato (nombre del evento, responsable del evento, etc.) a la hoja de Excel.

Para solucionar estos problemas se diseñará, desarrollará e implementará un sistema basado en los requerimientos del personal encargado de organizar los eventos.

El sistema consistirá de una serie de bloques (módulos) con funcionalidades e interfaces entre ellos claramente definidas, ver figura 3.3.1.

Los CGIs serán ejecutados por los usuarios a través de las interfaces gráficas (Front End) para enviar y recibir información de la base de datos (Back End).

El sistema se diseñará como una aplicación web (Figura 3.3.2) ya que este tipo de aplicaciones nos proporcionan varias ventajas con respecto a las aplicaciones tradicionales, entre las principales se encuentran:

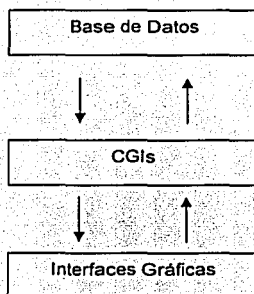


Figura 3.3.1 Módulos del Sistema

- La información clave se mantiene actualizada, centralizada y disponible en todo momento.
- No es necesario instalar el mismo software en diversos equipos.
- Es posible acceder a la aplicación desde cualquiera computadora que tenga acceso a la Intranet o Internet.
- Las modificaciones y actualizaciones que se le hagan a las páginas, repercutirán de manera instantánea a cualquier cliente que se conecte a partir de ese momento, facilitando de esta manera el mantenimiento de la aplicación.

Por ejemplo, los nuevos eventos se agregarán por medio de páginas HTML a una base de datos cuya información podrá ser explotada en todo momento, los reportes como las consultas se generarán rápidamente con tan sólo seleccionar el criterio deseado logrando con esto un proceso más rápido y eficiente.

3.3.1 Fallas en el hardware del sistema

Previendo posibles fallas en el hardware se adoptarán las siguientes estrategias para minimizar sus efectos sobre el correcto funcionamiento del sistema:

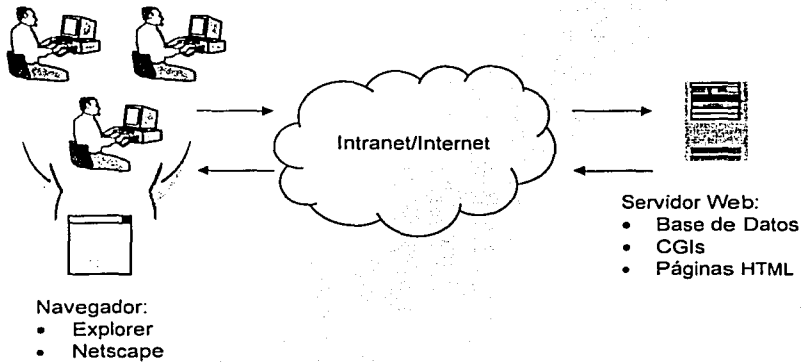


Figura 3.3.2 Modelo Operacional

3.3.1.1 Falla en almacenamiento del sistema

Se tomó la decisión de adoptar una arquitectura de base de datos centralizada de manera que si se llega a presentar una falla en la computadora del usuario, bastará con arreglar el problema localmente no afectando la información almacenada.

3.3.1.2 Falla de energía eléctrica

Se recomienda tener tanto el servidor de base de datos, como el servidor web conectados a un sistema ininterrumpido de energía que permita terminar todos los procesos correctamente en caso de falla de energía eléctrica.

3.3.1.3 Falla en la comunicación

Debido a la arquitectura de base de datos, es posible que se presente un problema de comunicación, existen dos posibles casos en los que se puede presentar este problema: que el sistema no pueda establecer una conexión con el servidor de base de datos, en cuyo caso será necesario esperar un momento y

reintentar o bien avisar al administrador de la base de datos para corregir el problema, el otro caso es que una vez que se ha establecido la conexión y se está trabajando con el sistema, esta se pierda, para lo cual se maneja un almacenamiento temporal de datos en la computadora del usuario (cache), por lo que en caso de presentarse este problema, el usuario tendrá que reintentar o en su caso dirigirse al DBA (DataBase Administrator).

3.3.2 Fallas en el software del sistema

Debido al diseño de la base de datos, se pueden presentar dos tipos de fallas generales en el software; en primer lugar una falla del propio sistema, para lo cual se cuenta con una pantalla que mostrará el código de excepción generado para que pueda ser corregido posteriormente.

La otra posible falla de software que puede presentarse es en el programa manejador de la base de datos, en caso de una falla de este tipo el sistema reportará que se ha producido un error en la conexión con la base de datos y se tendrá que avisar al DBA.

3.3.2.1 Fallas en la información almacenada en el sistema

Para proteger la integridad de los datos el DBA otorgará a los usuarios un nombre de usuario y contraseña para que puedan conectarse y tener acceso a la base de datos.

El sistema deberá validar toda entrada de datos donde sea factible que el usuario cometa errores de tecléo, como por ejemplo las fechas.

3.3.3 Modificaciones y mantenimiento del sistema

Las modificaciones al sistema serán la ampliación de características de acuerdo a las necesidades de los usuarios, también se integrará esta aplicación al sistema global de la institución educativa en caso de que cuente con el, obteniendo de esta

manera un sistema integral que contenga todos los servicios que presta la institución.

El mantenimiento del sistema se realizará principalmente cuando se informe de fallas en el software o cuando se adquiera nuevo hardware, la finalidad del mantenimiento es de que exista una perfecta interacción entre el software y el hardware involucrado en el funcionamiento del sistema.

3.4 OPCIONES DE SOLUCIÓN Y ELECCIÓN DE LA ÓPTIMA

Uno de los temas más importantes en el desarrollo de todo proyecto es sin duda elegir la herramienta con la cual se ha de resolver el problema, la evolución de la tecnología a traído como beneficio a los desarrolladores de sistemas una amplia variedad de nuevas y sofisticadas herramientas tanto de diseño como de desarrollo.

3.4.1 Manejadores de bases de datos

Los sistemas manejadores de bases de datos relacionales han sido altamente aceptados por la forma en que manejan los datos. Los sistemas relacionales ofrecen los siguiente beneficios:

- Acceso sencillo a los datos.
- Flexibilidad en el modelado de los datos.
- Disminución de la redundancia de los datos y el nivel de almacenamiento.
- Existe independencia entre el almacenamiento físico y el diseño lógico de los datos.
- Alto nivel de manipulación de datos.

El manejador de bases de datos es la interfaz entre los datos de bajo nivel almacenados en la base de datos y las aplicaciones.

3.4.1.1 Informix

Informix es un lenguaje de cuarta generación, considerado como uno de los manejadores de bases de datos más potentes y flexibles. Tiene como base para el desarrollo de aplicaciones, el manejo de base de datos relacionales.

Informix esta construido sobre RDSQL, que es una extensión del Structured Query Lenguaje (SQL) para la explotación de información.

Informix cuenta con un servidor dinámico el cual proporciona las siguientes ventajas:

- Arquitectura Cliente/Servidor.
- Escalable.
- Alto desempeño.
- Alta disponibilidad.
- Administración de sistema dinámico.
- Consulta de datos distribuida.
- Seguridad en el servicio de la base de datos.

Características

- Transacciones simples que implican pequeñas cantidades de datos.
- Acceso indexado a datos.
- Concurrencia.
- Tiempo de respuesta rápido.
- Aplicaciones de apoyo para la toma de decisiones

Cuenta con aplicaciones que proveen información para contabilizar, planear y tomar decisiones.

Escalabilidad

Se basa en el concepto de Arquitectura Dinámica Escalable cuya característica principal es la de contar con un proceso virtual que maneja procesamiento central, disco I/O, trabajo de red y funciones ópticas paralelas.

Alto desempeño

Proporciona un alto desempeño a través de los siguientes mecanismos:

- Administración de disco duros.
- Administración de memoria compartida dinámica.
- Fragmentación.
- Paralelización.

Alta disponibilidad

Utiliza los siguientes mecanismos para proteger la integridad y consistencia de los datos.

- Resaldos de bitácoras y bases de datos.
- Replicación de datos.
- Recuperación rápida

Consulta de datos distribuidos

Permite hacer consultas a más de una base de datos a través de múltiples servidores.

Seguridad

Valida que los usuarios que entren a las bases de datos cuenten con los privilegios de acceso necesarios a través del uso de los comandos de SQL GRANT y REVOKE.

3.4.1.2 ORACLE

Es un manejador de bases de datos relacionales que entre sus principales ventajas se encuentran la de soportar múltiples sistemas operativos y la de manejar aplicaciones de gran tamaño, esto último debido a su ejecución en memoria extendida.

Integridad

Garantiza la integridad de los datos, llevando un control de los registros que han sido modificados dentro de una transacción a fin de poder deshacer todos los cambios en caso de que por algún motivo el proceso no se termine correctamente.

Seguridad

Oracle provee seguridad tanto para el acceso a los datos como contra fallos. Estas características hacen fácil el manejo del diseño más complejo.

Concurrencia

Soporta un gran número de usuarios concurrentes, los cuales pueden estar ejecutando varias aplicaciones de bases de datos.

Disponibilidad

Puede controlar selectivamente la disponibilidad de los datos, según el nivel y subnivel elegido para la base de datos.

Ambiente Cliente/Servidor

Toda la responsabilidad de compartir el manejo de datos puede ser procesada por el servidor, mientras los clientes corren su aplicación, concentrándose en la interpretación y despliegue de los datos.

Este producto esta diseñado para proveer todas las facilidades que sean necesarias para el almacenamiento y recuperación de la información.

Soporta los esquemas de disco espejo para poder continuar con los procesos en caso de que un disco se dañe.

3.4.2 Tablas comparativas de los manejadores de bases de datos

En las siguientes tablas se muestran las comparaciones de los diferentes manejadores de bases de datos que se han presentado hasta el momento.

Concurrencia	Oracle	Informix	Interbase
Bueno	X		
Regular		X	X
Malo			

Tabla 3.3.4.1 Cantidad de clientes que pueden atender.

Multiplataforma	Oracle	Informix	Interbase
Bueno			
Regular	X	X	X
Malo			

Tabla 3.3.4.2 Variedad de plataformas en las que opera.

Índices	Oracle	Informix	Interbase
Bueno	X		X
Regular			
Malo		X	

Tabla 3.3.4.3 Tiempo de importación y generación de índices.

Cantidad de Archivos Abiertos	Oracle	Informix	Interbase
Bueno	X	X	X
Regular			
Malo			

Tabla 3.3.4.4 Número máximo de archivos abiertos.

Facilidad de Uso	Oracle	Informix	Interbase
Bueno			
Regular	X		X
Malo		X	

Tabla 3.3.4.5 Facilidad de uso.

Seguridad	Oracle	Informix	Interbase
Bueno			
Regular	X		X
Malo		X	

Tabla 3.3.4.6 Seguridad en la información.

Integridad	Oracle	Informix	Interbase
Bueno	X		
Regular		X	X
Malo			

Tabla 3.3.4.7 Integridad de los datos.

Reportes	Oracle	Informix	Interbase
Bueno	X		
Regular		X	X
Malo			

Tabla 3.3.4.8 Calidad en los reportes.

Programación	Oracle	Informix	Interbase
Bueno			X
Regular	X	X	
Malo			

Tabla 3.3.4.9 Facilidad de programación.

Compartición de Datos	Oracle	Informix	Interbase
Bueno	X		X
Regular		X	
Malo			

Tabla 3.3.4.10 Compartir los datos con otras bases de datos.

Tipos de Datos	Oracle	Informix	Interbase
Bueno			
Regular	X		X
Malo		X	

Tabla 3.3.4.11 Diversidad de tipos de datos.

Código Fuente	Oracle	Informix	Interbase
Sí			X
No	X	X	

Tabla 3.3.4.12 Código abierto.

3.4.3 Herramientas visuales de desarrollo

La elección de la herramienta de desarrollo es un aspecto que debe revisarse con especial cuidado ya que puede llevar al éxito o fracaso.

3.4.3.1 Visual C++

Visual C++ cuenta con las herramientas de programación visuales App Wizzard, Class Wizzard y el editor de recursos App Studio, las cuales están disponibles

dentro del WWB (Wizzard Workbench). Todas las herramientas visuales son para proyectos en Windows.

Puede generar aplicaciones para DOS si se tiene la edición profesional, pero no es en eso en lo que Visual C++ se destaca. Las herramientas App Wizzard y Class Wizzard dependen de la biblioteca Microsoft Foundation Class (MFC), que incluye la mayor de la programación de Windows (SDK) en una clase de C++. Si lo que se requiere es programar solamente en C o escribir directamente a la API (Interfaz de Programación de Aplicaciones) de Windows, los asistentes no serán útiles.

El AppWizzard permite seleccionar funciones básicas como "Printing = impresión", "Previewing = Vista Preliminar" y "Context Sensitivity Help = Ayuda Sensible al Contexto" y luego generar código basado en MFC.

Es fácil generar la estructura de una aplicación, personalizarla requiere un amplio conocimiento del lenguaje C++ y de MFC, Los asistentes son herramientas buenas para el aprendizaje y pueden aumentar considerablemente la productividad de aquellos que tienen una buena base de C++ y MFC.

La Herramienta más importante del C++ es App Studio, que es un editor de recursos el cual puede manejar cualquier recurso y es completamente visual.

3.4.3.2 Visual Basic 6

Con esta versión concluye toda una generación en este lenguaje. Las principales novedades son la creación de nuevos componentes y la creación de controles ActiveX, estas características fueron incluidas debido a las presiones comerciales recibidas por Delphi (versión visual de Pascal) de Borland, que captó muchos desarrolladores desilusionados con el atrasó de Visual Basic. También se mejoró el performance, debido a la posibilidad de generar código compilado en vez de código interpretado como en las versiones anteriores. Pero a pesar de que se trata de código compilado, es necesario seguir incluyendo el "runtime".

Denominado "Dream Release" por el equipo de desarrolladores, incluye muchas novedades, como la posibilidad de crear componentes.

Este lenguaje ha crecido conforme aparece una nueva versión. Se orienta sobre todo al desarrollo empresarial y a las redes (Intranet, Extranet e Internet).

Cuenta con tres ediciones que se adaptan a los requerimientos del usuario.

- Working Model, que es gratuita y puede obtenerse desde internet. Esta edición es más apta para conocer el entorno y ver algunas de sus características, que para utilizarse en un ambiente de producción.
- Estándar es aquella que se consigue junto con una serie de libros y que sirve para aprender programación en Visual Basic. Incluye varias funciones completas, aunque carece de suficientes características para generar aplicaciones de tipo profesional, ésta es la más utilizada en el medio de los desarrolladores. Incluye gran parte de herramientas necesarias para generar aplicaciones comerciales aptas para cubrir casi cualquier necesidad.
- Empresarial, es la más completa. Incluye herramientas que se utilizan en el entorno empresarial, como acceso a datos en redes LAN. También cuenta con la capacidad de crear grupos de desarrollo controlados y algunos otros aspectos del desarrollo empresarial, como la conexión directa a bases de datos Oracle y Microsoft SQL Server.

Todas estas versiones están disponibles en el idioma español.

3.4.4 Tablas comparativas de las herramientas visuales de desarrollo

En las siguientes tablas se muestran las características que son fundamentales al momento de hacer una elección.

Aprendizaje	Visual Basic 5.0	Visual C++	Delphi 5.0
Bueno	X		
Regular		X	X
Malo			

Tabla 3.4.6.1 Facilidad de aprendizaje.

Componentes Visuales	Visual Basic 5.0	Visual C++	Delphi 5.0
Bueno			X
Regular	X	X	
Malo			

Tabla 3.4.6.2 Componentes visuales para el desarrollo de sistemas.

Servicios Web	Visual Basic 5.0	Visual C++	Delphi 5.0
Bueno			X
Regular	X	X	
Malo			

Tabla 3.4.6.3 Herramientas para el desarrollo de servicios web.

Base de Datos	Visual Basic 5.0	Visual C++	Delphi 5.0
Bueno		X	X
Regular	X		
Malo			

Tabla 3.4.6.4 Conectividad a las diferentes bases de datos.

3.4.5 Elección de la solución

La base de datos y la herramienta visual de desarrollo que se ha decidido utilizar para la elaboración del proyecto son Interbase (Back-End) y Delphi 5 Enterprise (Front-End) debido a sus características y a la gran relación que existe entre ellos, al ser estos productos de Borland.

3.4.5.1. Elección de la base de datos

Uno de los temas más importantes en el desarrollo de una aplicación, es la elección de la base de datos que se va utilizar.

La decisión por la que se tiene que optar, debe ser una decisión fundamentada en cual de las diferentes opciones de las que disponemos se adapta mejor a nuestras necesidades.

A continuación se enumeran las razones por las cuales se eligió Interbase.

1. Madurez del producto.
2. Gratuito.
3. Mantenimiento nulo.
4. Costo de desarrollo.
5. Velocidad de desarrollo.
6. Multiplataforma.
7. Internet.
- 8 Fácil integración en las herramientas de desarrollo.

Madurez del producto

Interbase es un producto con más de quince años de existencia en el mercado, siendo utilizado por firmas como Motorola, la NASA y el Gobierno de los Estados Unidos, lo cual nos garantiza que es un producto fiable y robusto, probado exhaustivamente y que ofrece buenos niveles de seguridad.

Gratuito

Interbase a pasado a ser Open Source, significando que los fuentes del mismo pueden ser editados y modificados por cualquier usuario.

Además se puede redistribuir sin tener que pagar por ello y se puede hacer uso comercial del mismo. En otras palabras, se ha convertido en gratuito y cualquier persona lo puede utilizar.

Esto significa en términos reales una disminución importante en el precio final del proyecto.

Mantenimiento nulo

La instalación es muy sencilla, ya que se limita a ejecutar un instalador. Una vez instalado, prácticamente no hay que hacer nada y nos podemos olvidar de términos como table spaces, clusters así como de los dolores de cabeza que nos producen sistemas como Oracle e Informix

Costo de Desarrollo

El costo baja considerablemente debido a la potencia que nos otorga SQL en las consultas, evitando usar ciclos, instrucciones de salto, instrucciones de búsqueda, lo que también permite una mayor legibilidad del código; además de poder trabajar con más de una base de datos a la vez.

Velocidad de Desarrollo

Si las cosas se hacen bien, el tiempo de desarrollo se disminuye, además de facilitar el mantenimiento de la base de datos.

Multiplataforma

Si un día se decide cambiar de sistema operativo y optar por Unix o Novell, se puede hacer ya que Interbase soporta las siguientes plataformas:

- Windows 95/98/NT/2000
- HP/UX

- SUN
- SCO 5.0
- Red Hat Linux
- Novell Netware
- AT&T NCR
- IBM AIX
- DG-UX Aviiion
- DEC-Uinx
- Open/VMS

Internet

Uno de los mayores requisitos para una base de datos es que esté preparada para ser usada en Internet y este es el caso de Interbase.

Fácil integración en las herramientas de desarrollo

Interbase perteneció a Borland y por lo tanto era una herramienta comercial de Borland. Como tal, Borland se preocupó por la integración de Interbase a sus herramientas de desarrollo Delphi y C++Builder.

Con ese espíritu existen los componentes IBX, que permiten acceder directamente desde las herramientas de desarrollo a las bases de datos, sin necesidad de capas intermedias, obteniéndose con esto un incremento en el rendimiento.

3.4.5.2 Elección de la herramienta visual de desarrollo

A la hora de que se plantea el desarrollo de una aplicación, una de las primeras preguntas que surge es el tipo de conectividad que se va utilizar para acceder la base de datos.

Hoy en día, Delphi, nos proporciona una amplia variedad de opciones:

BDE

ADO

IBX

IBO

FreeIB

DBE

ODBC

Seguramente se pueden encontrar más opciones (por ejemplo COM) para conectar la aplicación de Delphi con Interbase, pero las opciones citadas anteriormente son las más utilizadas..

Ante un abanico tan amplio de posibilidades, debemos tener los suficientes criterios de decisión para elegir entre un tipo de conectividad u otro.

La primera gran decisión deberá ser si se opta por el rendimiento bruto o por la compatibilidad con diferentes manejadores de bases de datos.

Si se decide por la compatibilidad con diferentes bases de datos, las posibles soluciones son utilizar BDE, ADO o ODBC.

En el caso de que se decida utilizar Interbase como el servidor de bases de datos, se pueden utilizar los componentes IBX, IBO o FreeIB, en cuyo caso se asume la pérdida de compatibilidad a favor de un mayor rendimiento.

En cambio, las otras opciones (BDE, ADO, ODBC) proporcionan compatibilidad entre diferentes bases de datos, y son estas las que traducen las peticiones de nuestra aplicación a instrucciones que son capaces de entender las bases de datos. Dicha interpretación de las instrucciones consume una determinada

cantidad de tiempo, lo cual provoca que estos sistemas sean más lentos que los anteriores.

El BDE (Borland Database Engine) es la solución proporcionada por Borland para este tipo de sistemas.

El inconveniente principal del BDE es que inicialmente fue ideado como un sistema para proporcionar accesibilidad a las bases de datos de escritorio tipo Paradox y Dbase, pero posteriormente fue ampliado para permitir la conectividad a bases de datos relacionales como Interbase, Oracle o Informix.

Algo similar sucede con ODBC y ADO. ODBC es el sistema planteado inicialmente por Microsoft en sus sistemas operativos Windows para el acceso a bases de datos. Una de las quejas más frecuentes era su lentitud, y con ese fin nació ADO, además de tener una mayor orientación hacia las bases de datos relacionales. Aún así, ADO es relativamente más lento que las opciones presentadas por Borland.

Un pregunta a tener en cuenta a la hora de desarrollar aplicaciones, sería sobre que plataforma se desea ejecutar la aplicación cliente. Si se decide que sobre una plataforma Windows, ambas soluciones (ADO y DBE) son perfectamente correctas. En cambio si se opta por migrar la aplicación a Linux a través de Kylix, se deben usar irremediamente los componentes IBX, ya que no existe de momento forma de portar una aplicación desarrollada en Delphi usando ADO / BDE a Kylix.

Componentes de acceso directo a la base de datos

Son aquellos que nos permiten acceder desde la aplicación a la base de datos sin necesidad de capas intermedias.

Si bien se logra un importante incremento de rendimiento a la hora de realizar operaciones sobre la base de datos, se pierde la posibilidad de migrar la base de datos a otra diferente. En Delphi 5 se puede hablar de dos grandes grupos de componentes que nos proporcionan dicha facilidad: IBO y FreeIB.

Los componentes IBO (Interbase Objects) han sido desarrollados por Jason Wharton, siendo un conjunto de componentes, tanto de acceso a datos, como de presentación de los mismos. Disponibles en todas las versiones de Delphi, a su favor tienen que es un producto con una vida relativamente larga, habiendo sido probados con profundidad. En su contra estaría el hecho de que se tiene que pagar por ellos. A pesar de ello son altamente recomendables.

Por otra parte nos encontramos con FreeIB, un movimiento Open Source desde sus comienzos, llevado a cabo por Greg Deatz, aunque actualmente ha sido adoptado por Second Dream. También funcionan para todas las versiones de Delphi y su rendimiento es más que satisfactorio. Además de ser gratuitos.

Finalmente, Borland decidió sacar sus propios componentes de acceso nativo a Interbase bajo la denominación de IBX basados en el código fuente de los FreeIB.

CAPÍTULO IV

En el presente capítulo haremos la descripción detallada de la construcción del sistema que simplificará el proceso de administración de eventos. Para esto haremos uso de pantallas que hemos capturado en el transcurso de la elaboración del proyecto, así como gráficas y todo aquello que se ha considerado necesario para una adecuada comprensión del sistema.

Con la finalidad de mantener congruencia con lo que hasta este momento se ha planteado, se usarán las herramientas y metodologías presentadas en el capítulo 2. Tanto para la elaboración de la base de datos como de las interfaces gráficas se hará uso de la metodología objeto-relacional.

Algunos de los beneficios con que cuenta el sistema es que está elaborado con una metodología estándar, facilitando de esta manera el mantenimiento del mismo.

Finalmente el propósito de este proyecto es simplificar los procesos administrativos que se llevan a cabo en la organización de eventos dentro de las instituciones educativas.

4.1 Diagrama de contexto

El diagrama de contexto es un caso especial del diagrama de flujo de datos, en donde un sólo proceso (representado por un círculo) representa todo el sistema. El nombre del proceso suele ser el nombre completo del sistema o bien un acrónimo convenido.

A partir de un diagrama de contexto es posible recalcar varias características importantes del sistema, tales como:

- Las personas, organizaciones y aplicaciones con las que se comunica el sistema. Estos se conocen como terminales.

- Los datos que el sistema recibe del mundo exterior y que deben procesarse de alguna forma.
- Los datos que el sistema produce y que se envían al mundo exterior.
- Los almacenes de datos que el sistema comparte con los terminales. Tales almacenes de datos se crean fuera del sistema para su uso, o bien, son creados en él y usados en el exterior.

Los terminales se representan por medio de rectángulos en el diagrama de contexto. Se comunican con el sistema a través de flujos de datos o de control. Estos no deben comunicarse directamente entre sí, ya que es incorrecto dentro de un diagrama de contexto.

A continuación se muestra en la figura 4.1.1 el diagrama de contexto del Sistema de Administración de Eventos. Como se puede observar el acceso al sistema se llevará a cabo de tres formas:

- La primera como Usuario Administrador, el cual contará con privilegios para crear, borrar, consultar y modificar eventos así como para administrar los catálogos de salas, equipo, alimentos, accesorios, montajes, usuarios, divisiones y niveles.
- La segunda forma de acceso al sistema es con los privilegios de Usuario Lectura/Escritura, el cual podrá crear, borrar, consultar y modificar eventos.
- La tercera forma es como Usuario Lectura y sus privilegios estarán limitados sólo a la consulta de la información de los eventos.

De acuerdo al perfil de usuario se tendrá acceso a las siguientes operaciones:

- Eventos, este módulo permite agregar, borrar, consultar y modificar eventos.
- Búsquedas, este módulo realiza consultas de eventos por sala, fecha, nombre y número de folio.

- Reportes, este módulo genera informes de eventos por fecha, de alimentos por fecha, de eventos por sala y fecha, de eventos por semestre y número de folio, de uso de salas por semestre, sala y fecha.
- Catálogos, este módulo permite agregar, borrar, consultar y modificar los catálogos de: salas, equipo, alimentos, accesorios, montajes, usuarios, divisiones y niveles

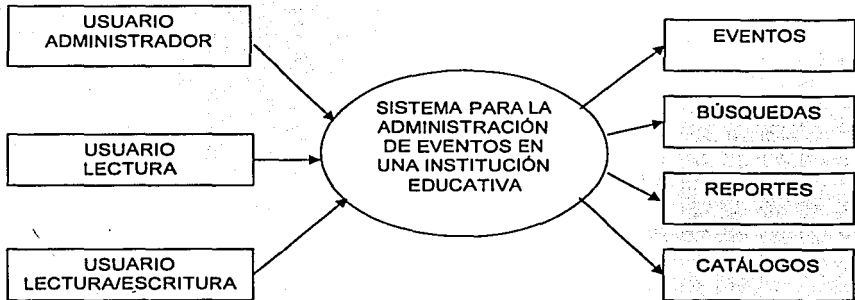


Figura 4.1.1 Diagrama de Contexto.

4.2 Diagramas

Actualmente existen diferentes técnicas o herramientas que nos ayudan a manejar los datos de una forma eficiente. Una de las herramientas mas frecuentemente utilizadas para el análisis estructurado son los diagramas de flujo de datos.

4.2.1 Diagrama de flujo de datos

El diagrama de flujo de datos es una técnica gráfica que representa el flujo de la información y de las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida.

Un diagrama de flujo de datos es un modelo que describe los flujos de datos y los procesos que cambian o transforman los datos de un sistema.

Algunas de las características más importantes de los diagramas de flujo de datos son:

- Es gráfico.
- Particionado.
- Multidimensional.

Esta herramienta permite visualizar el sistema desde el punto de vista de los datos y no de quien trabaja con ellos. Los elementos de un diagrama de flujo de datos son:

- Flujo de datos.- Representado por vectores.
- Procesos.- Representados por círculos o burbujas.
- Archivos.- Representados por rectángulos.
- Fuentes o depósitos de datos.- Representados por rectángulos con barras a los costados.

Un diagrama de flujo de datos preliminar consta de un solo nivel. Por ello es necesario una nivelación ascendente del diagrama de flujo de datos preliminar. Esto significa que se deben agrupar procesos relacionados, cada uno de los cuales representará un proceso (círculo) de un diagrama de nivel superior.

A continuación se presenta el diagrama de flujo de datos del módulo evento (ver figura 4.2.1.1). En este diagrama se muestran dos campos, uno para dar de alta un nuevo evento y otro para ver el listado de los eventos existentes.



Figura 4.2.1.1 Diagrama nivel uno de eventos.

Los campos que dan de alta un nuevo evento dentro del módulo eventos se presentan en la siguiente figura (ver figura 4.2.1.2).

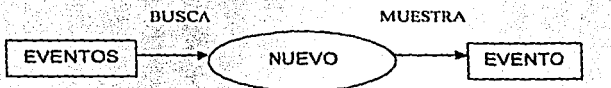


Figura 4.2.1.2 Diagrama nivel dos de nuevo evento.

1
Clave del evento
Nombre del evento
Nivel
Responsable
Teléfono
E-mail
Número de cuenta
Dirección/División
Comentarios

Una vez agregados los datos para un nuevo evento, es posible dar de alta de manera opcional equipo, alimentos y accesorios (ver figura 4.2.1.3).

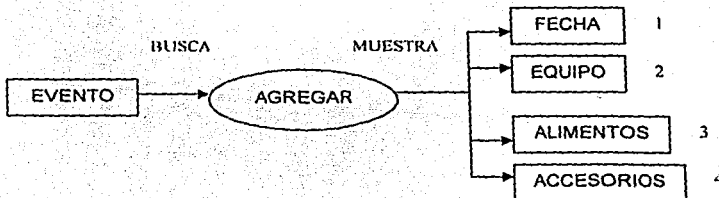


Figura 4.2.1.3 Diagrama nivel tres de nuevo evento.

1	2	3	4
Sala	Nombre del equipo	Sala	Accesorio
Fecha	Cantidad solicitada	Alimento	Comentarios
Inicio		Inicio	
Fin		Fin	
Montaje		Número de personas	
Descripción		Costo	
Seguridad		Total	
		Comentarios	

El siguiente es un diagrama de nivel dos que se encarga de buscar el listado de los eventos que se han dado de alta, realizando la búsqueda por semestre y número de folio (ver figura 4.2.1.4).



Figura 4.2.1.4 Diagrama nivel dos de listado eventos.

1
Semestre
Del folio
Al folio

La figura 4.2.1.5 presenta el resultado de la búsqueda de eventos en forma de listado, mostrando datos tales como nombre del evento, responsable del evento, etc.



Figura 4.2.1.5 Diagrama nivel tres eventos.

1
Clave del evento
Nombre del evento
Nombre del responsable
Teléfono
E-mail

El siguiente es el módulo de búsqueda (ver figura 4.2.1.6). Aquí se cuenta con tres tipos de búsqueda: general, disponibles y evento.

El primer campo del módulo de búsqueda es el de general. La búsqueda se realiza por sala, fecha y nombre del evento (ver figura 4.2.1.7).

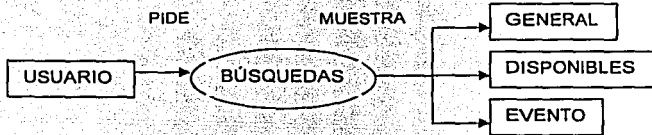


Figura 4.2.1.6 Diagrama nivel uno de búsquedas.



Figura 4.2.1.7 Diagrama nivel dos búsquedas-general.

1
Sala
Fecha
Nombre del evento

El figura 4.2.1.8 es un diagrama de nivel tres y muestra los campos que resultan de realizar una búsqueda general.



Figura 4.2.1.8 Diagrama nivel tres resultados búsqueda general.

1
Número
Evento
Nombre del evento
Sala
Fecha
Inicio
Fin

A continuación se presenta el diagrama de flujo de datos de nivel dos para la búsqueda de salas disponibles (ver figura 4.2.1.9).



Figura 4.2.1.9 Diagrama nivel dos búsquedas-disponibles.

1
Salas
Fecha
Inicio
Fin

En la figura 4.2.1.10 se presenta lo que resulta de realizar la búsqueda por salas disponibles. El diagrama mostrado es de nivel tres y muestra el nombre de la sala requerida y su disponibilidad.

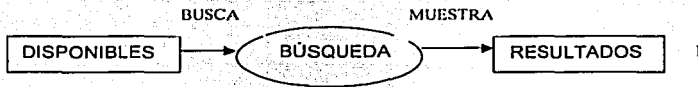


Figura 4.2.1.10 Diagrama nivel tres de los resultados búsqueda-disponibles.

1
Sala
Status

El diagrama de flujo de datos que a continuación se presenta es el que corresponde a la búsqueda por evento (ver figura 4.2.1.11). El diagrama es de nivel dos y la búsqueda aquí se realiza tan sólo por el número de folio que le es asignado al evento cuando éste es dado de alta.



Figura 4.2.1.11 Diagrama nivel dos búsquedas-evento.

1
Folio

La figura 4.2.1.12 muestra el diagrama de nivel tres con los datos completos del evento buscado.

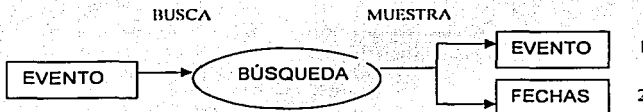


Figura 4.2.1.12 Diagrama nivel tres resultados búsqueda-evento.

1	2
Clave del evento	Número
Nombre del evento	Sala
Nivel	Fecha
Responsable	Inicio
Teléfono	Fin
E-mail	Montaje
Número de personas	Equipo
Número de cuenta	Alimentos
Dirección/División	Accesorios
Comentarios	

El siguiente es el diagrama de flujo de datos de nivel uno para la generación de reportes (ver figura 4.2.1.13). El reporte puede ser: general, por alimentos, por lugar, por número de subevento, por uso de sala y por uso de equipo.

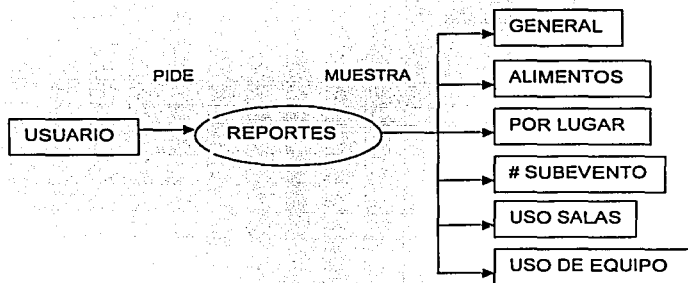


Figura 4.2.1.13 Diagrama nivel uno de reportes.

El primero de los reportes que se puede mostrar es el general. En la figura 4.2.1.14 se muestra el diagrama de nivel dos con los campos que se requieren para iniciar nuestra consulta.

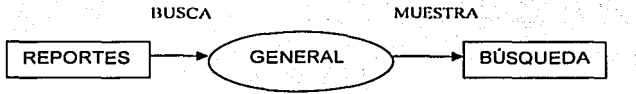


Figura 4.2.1.14 Diagrama nivel dos reportes-general.

1
Fecha inicial
Fecha final

En la figura 4.2.1.15 se presenta el diagrama de flujo de datos de nivel tres en el que podemos observar el resultado de la búsqueda que realizamos.. Entre los datos que aquí se muestran están el nombre del evento, el folio del evento, la descripción del evento, etc.

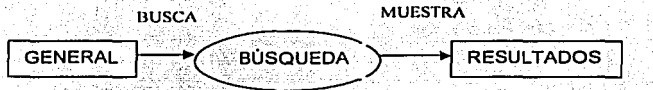


Figura 4.2.1.15 Diagrama nivel tres resultados búsqueda-general.

1
Número
Folio
Evento
Horario
Sala
Descripción
Equipo
Alimentos

Otro de los reportes que se puede generar es por lugar (ver figura 4.2.1.16). Los datos que aquí se piden son el nombre de la sala y un rango de fechas, y de manera opcional se pueden solicitar los detalles de equipos y alimentos.

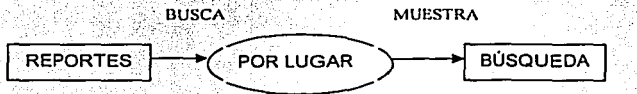


Figura 4.2.1.16 Diagrama nivel dos reportes-por lugar.

1
Salas
Fecha inicial
Fecha final
Mostrar detalles de equipo y alimentos

El diagrama de nivel tres que a continuación se presenta muestra los datos obtenidos después de generar un reporte por lugar (ver figura 4.2.1.17).



Figura 4.2.1.17 Diagrama nivel tres resultados búsqueda-por lugar.

1
Número
Evento
Responsable
Fecha
Inicio
Fin
Descripción
Equipo
Alimentos

Otro de los reportes que se puede generar es por subevento (ver figura 4.2.1.18). Para realizar el reporte por subevento se deben proporcionar tres datos: semestre, folio inicial y folio final.



Figura 4.2.1.18 Diagrama nivel dos reportes-# subevento.

1
Semestre
Del folio
Al folio

El diagrama de nivel tres que a continuación se presenta muestra el resultado de un reporte por número de subevento (ver figura 4.2.1.19). Los datos que contiene el reporte son: número, folio, nombre del evento, nivel, dirección-división y subeventos.

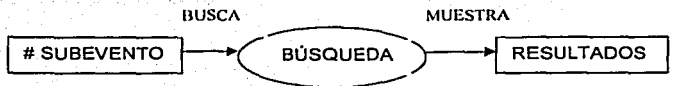


Figura 4.2.1.19 Diagrama nivel tres resultados búsqueda-#subevento.

1
Número
Folio
Nombre del evento
Nivel
Dirección/División
Subeventos

En la figura 4.2.1.20 se muestra el diagrama de nivel dos con los datos que se requieren para generar un reporte por uso de salas.



Figura 4.2.1.20 Diagrama nivel dos reportes-uso salas.

1
Semestre
Sala
Fecha inicial
Fecha final

El diagrama de nivel tres que a continuación se presenta, muestra los datos obtenidos de generar un reporte por uso de salas (ver figura 4.2.1.21). Los datos que contiene el reporte generado son: número, sala, #subeventos y horas.

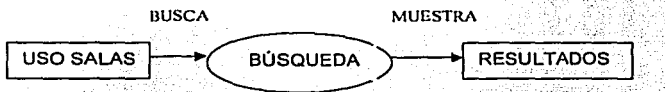


Figura 4.2.1.21 Diagrama nivel tres resultados búsqueda-uso-salas.

1
Número
Sala
Subeventos
Horas

Para realizar un reporte por uso de equipo se deben proporcionar los siguientes datos: semestre, equipo, fecha inicial y fecha final (ver figura 4.2.1.22).



Figura 4.2.1.22 Diagrama nivel dos reportes-uso-equipo.

1
Semestre
Equipo
Fecha inicial
Fecha final

El diagrama de nivel tres que a continuación se presenta muestra los datos obtenidos de generar un reporte por uso de equipo (ver figura 4.2.1.23).

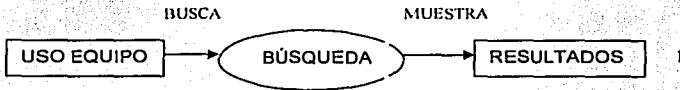


Figura 4.2.1.23 Diagrama nivel tres resultados búsqueda-uso-equipo.

1
Número
Equipo
Subeventos
Horas

La figura 4.2.1.24 muestra el diagrama de flujo de nivel uno para la administración de catálogos. Existen catálogos de salas, equipo, alimentos, accesorios, montajes, usuarios, divisiones y niveles.

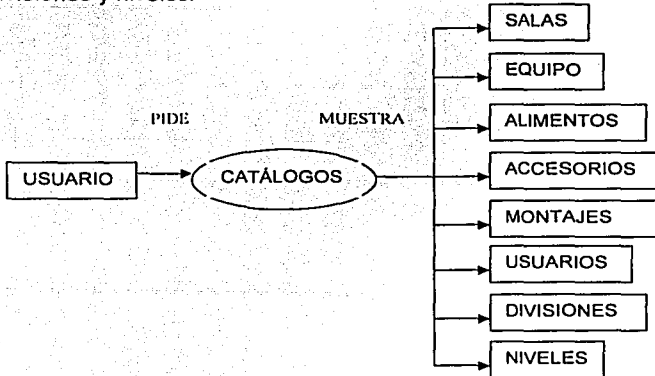


Figura 4.2.1.24 Diagrama nivel uno de catálogos.

El diagrama de nivel dos que se presenta en la figura 4.2.1.25 es el que corresponde al catálogo de salas. Aquí tenemos dos opciones, la primera opción nos permite consultar las salas dadas de alta y la segunda nos permite agregar una nueva sala.

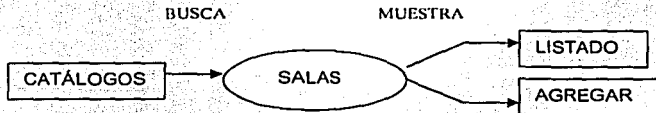


Figura 4.2.1.25 Diagrama nivel dos de salas.

La figura 4.2.1.26 muestra en un diagrama de nivel tres el listado de las salas que se encuentran dadas de alta en el catálogo. Los datos que se presentan son: número, sala y capacidad.

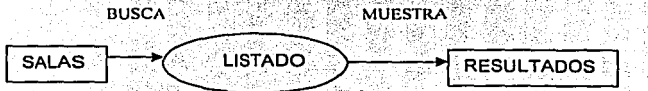


Figura 4.2.1.26 Diagrama nivel tres listado-salas.

1
Número
Sala
Capacidad

El siguiente diagrama de nivel tres (ver figura 4.2.1.27) muestra los datos que debemos proporcionar para poder dar de alta una nueva sala dentro de nuestro catálogo de salas.



Figura 4.2.1.27 Diagrama nivel tres agregar-salas.

1
Sala
Capacidad

El diagrama de nivel dos que se presenta en la figura 4.2.1.28 es el que corresponde al catálogo de equipo. Aquí tenemos dos opciones, la primera opción nos permite consultar los equipos existentes y la segunda nos permite agregar un nuevo equipo.

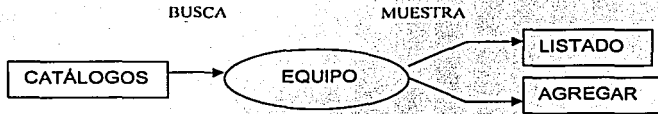


Figura 4.2.1.28 Diagrama nivel dos de equipo.

La figura 4.2.1.29 muestra en un diagrama de nivel tres el listado de los equipos que se encuentran dados de alta en el catálogo. Los datos que se presentan son: número, equipo y capacidad.



Figura 4.2.1.29 Diagrama nivel tres listado-equipos.

1
Número
Equipo
Capacidad

Para agregar un nuevo equipo al catálogo debemos proporcionar el nombre del equipo y la cantidad de éste, ver figura 4.2.1.30.



Figura 4.2.1.30 Diagrama nivel tres agregar-equipo.

1
Equipo
Cantidad

El diagrama de nivel dos que se presenta en la figura 4.2.1.31 es el que corresponde al catálogo de alimentos. Aquí tenemos dos opciones a elegir, la primera nos permite ver el listado de los alimentos dados de alta y la segunda nos permite dar de alta un nuevo alimento.

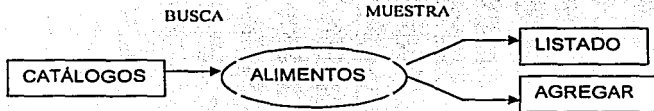


Figura 4.2.1.31 Diagrama nivel dos de alimentos.

La figura 4.2.1.32 muestra en un diagrama de nivel tres el listado de los alimentos que se encuentran dados de alta en el catálogo. Los datos que se presentan son: número, alimento y costo.



Figura 4.2.1.32 Diagrama nivel tres listado-alimentos.

1
Número
Alimento
Costo

Para agregar un nuevo alimento al catálogo debemos proporcionar el nombre del alimento y el costo de éste, ver figura 4.2.1.33.



Figura 4.2.1.33 Diagrama nivel tres agregar-alimentos.

1
Alimento
Costo

El diagrama de nivel dos que se presenta en la figura 4.2.1.34 es el que corresponde al catálogo de accesorios. Contamos con dos opciones, la primera nos permite ver el listado de los accesorios existentes y la segunda nos permite dar de alta un nuevo accesorio.



Figura 4.2.1.34 Diagrama nivel dos de accesorios.

La figura 4.2.1.35 muestra en un diagrama de nivel tres el listado de los accesorios que se encuentran dados de alta en el catálogo. Los datos que se presentan son: número y accesorio.



Figura 4.2.1.35 Diagrama nivel tres listado-accesorios.

1
Número
Accesorio

Para agregar un nuevo accesorio al catálogo debemos proporcionar el nombre de éste, ver figura 4.2.1.36.



Figura 4.2.1.36 Diagrama nivel tres agregar-accesorios.

1
Accesorio

El diagrama de nivel dos que se presenta en la figura 4.2.1.37 es el que corresponde al catálogo de montajes. Existen dos opciones, la primera nos permite ver el listado de montajes existentes y la segunda nos permite dar de alta un nuevo montaje.

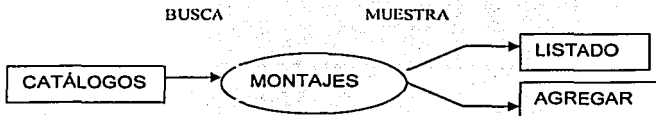


Figura 4.2.1.37 Diagrama nivel dos de montajes.

La figura 4.2.1.38 muestra en un diagrama de nivel tres el listado de los montajes que se encuentran dados de alta en el catálogo. Los datos que se presentan son: número y montaje.



Figura 4.2.1.38 Diagrama nivel tres listado-montajes.

1
Número
Montaje

Para agregar un nuevo montaje al catálogo debemos proporcionar el nombre de éste, ver figura 4.2.1.39.



Figura 4.2.1.39 Diagrama nivel tres agregar-montajes.

1
Montaje

El diagrama de nivel dos que se presenta en la figura 4.2.1.40 es el que corresponde al catálogo de usuarios. Contamos con dos opciones, la primera nos permite ver el listado de usuarios existentes y la segunda nos permite dar de alta un nuevo usuario.

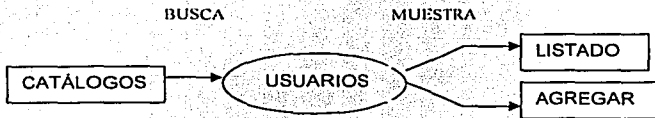


Figura 4.2.1.40 Diagrama nivel dos de usuarios.

La figura 4.2.1.41 muestra en un diagrama de nivel tres el listado de los usuarios que se encuentran dados de alta en el catálogo. Los datos que se presentan son: número, usuario y tipo de usuario.



Figura 4.2.1.41 Diagrama nivel tres listado-usuarios

1
Número
Usuario
Tipo de usuarios

Para agregar un nuevo usuario al catálogo debemos proporcionar el nombre de éste así como su clave de acceso y tipo, ver figura 4.2.1.42.

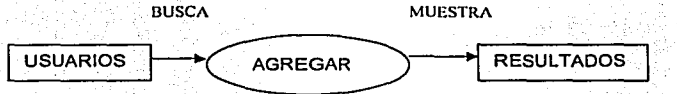


Figura 4.2.1.42 Diagrama nivel tres agregar-usuarios.

1
Usuario
Clave de Acceso
Tipo de usuario

El diagrama de nivel dos que se presenta en la figura 4.2.1.43 es el que corresponde al catálogo de dirección/división. Aquí tenemos dos opciones, la primera nos permite ver el listado de direcciones/divisiones existentes y la segunda nos permite dar de alta una nueva dirección/división.

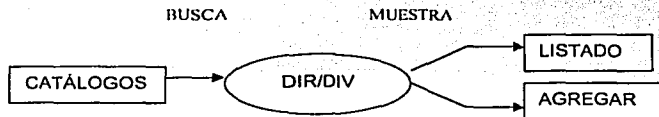


Figura 4.2.1.43 Diagrama nivel dos de direcciones/divisiones.

La figura 4.2.1.44 muestra en un diagrama de nivel tres el listado de las direcciones/divisiones que se encuentran dadas de alta en el catálogo. Los datos que se presentan son: número y nombre de la dirección/división.

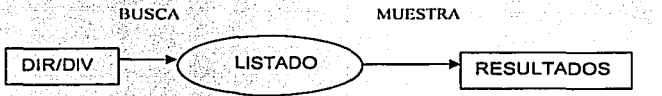


Figura 4.2.1.44 Diagrama nivel tres listado-direcciones/divisiones.

1
Número
Dirección/División

Para agregar una nueva dirección/división al catálogo debemos proporcionar el nombre de ésta, ver figura 4.2.1.45.

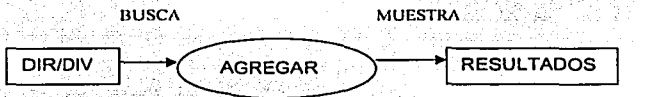


Figura 4.2.1.45 Diagrama nivel tres agregar-direcciones/divisiones.

1
Dirección/División

El diagrama de nivel dos que se presenta en la figura 4.2.1.46 es el que corresponde al catálogo de niveles. Existen dos opciones, la primera nos permite ver el listado de los niveles dados de alta y la segunda nos permite dar de alta un nuevo nivel.

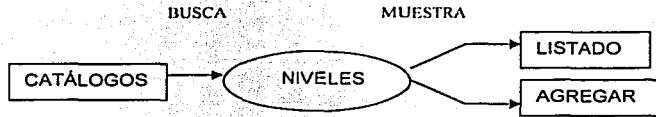


Figura 4.2.1.46 Diagrama nivel dos de niveles.

La figura 4.2.1.47 muestra en un diagrama de nivel tres el listado de los niveles que se encuentran dados de alta en el catálogo. Los datos que se presentan son: número y nombre del nivel.

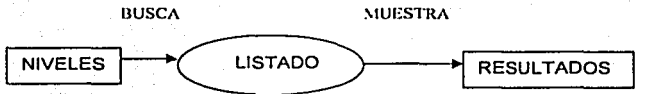


Figura 4.2.1.47 Diagrama nivel tres listado-niveles.

1
Número
Nivel

Para agregar un nuevo nivel al catálogo debemos proporcionar el nombre de éste, ver figura 4.2.1.48.

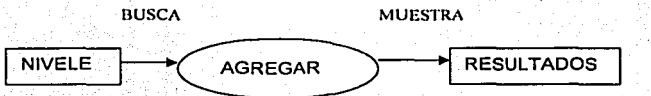


Figura 4.2.1.48 Diagrama nivel tres agregar-niveles.

2
Nivel

4.2.2 Diccionario de datos

El diccionario de datos contiene las características de las tablas que definen la estructura de la base de datos, en la lista que a continuación se presenta se muestra el nombre de la tabla, el nombre del campo, el tipo de dato, indicador de llave primaria (PK), indicador de llave foránea (FK) y por último una pequeña descripción del campo.

Accesorios

Catálogo de accesorios.

Campo	Tipo de Dato	PK	FK	Descripción
Cve_accesorio	varchar(5)	X		Clave del accesorio
Nombre	varchar(255)			Nombre del accesorio

Alimentos

Catálogo de alimentos.

Campo	Tipo de Dato	PK	FK	Descripción
Cve_alimento	varchar(5)	X		Clave del alimento
Nombre	varchar(255)			Nombre del alimento
Costo	numeric(9, 2)			Precio del alimento

Dirección_División

Catálogo de divisiones.

Campo	Tipo de Dato	PK	FK	Descripción
Cve_direcc_div	integer	X		Clave de la división
Nombre	varchar(255)			Nombre de la división

Equipo

Catálogo de equipo.

Campo	Tipo de Dato	PK	FK	Descripción
Cve_equipo	varchar(5)	X		Clave del equipo
Nombre	varchar(255)			Nombre del equipo
Cantidad	integer			Cantidad del equipo

Eventos

Información de los eventos.

Campo	Tipo de dato	PK	FK	Descripción
Cve_evento	varchar(8)	X		Clave del evento
Cve_semestre	char(4)		X	Clave del semestre
Folio	Integer			Folio del evento
Cve_nivel	Integer		X	Clave del nivel
Nombre_evento	varchar(255)			Nombre del evento
Nombre_responsable	varchar(255)			Nombre del responsable
Telefono	varchar(10)			Teléfono del responsable
E-mail	varchar(255)			E-mail del responsable
Num_personas	integer			Número de personas
Num_cuenta	varchar(255)			No. de cuenta del responsable
Dirección_división	varchar(255)			Nombre de la división
Cve_direcc_div	integer		X	Clave de la división
Comentarios	blob			Comentarios del evento

Eventos_Acesorios

Accesorios elegidos para el evento.

Campo	Tipo de Dato	PK	FK	Descripción
Cve_fecha	integer	X		Clave de la fecha
Cve_accesorio	varchar(5)	X		Clave del accesorio
Descripción	varchar(255)			Comentarios

Eventos_Alimentos

Alimentos elegidos para el evento.

Campo	Tipo de Dato	PK	FK	Descripción
Cve_fecha	integer	X		Clave de la fecha
Cve_Alimento	varchar(5)	X		Clave del alimento
Cve_sala	varchar(5)		X	Clave de la sala
Hora_inicio	date			Hora de inicio
Hora_Fin	date			Hora de terminación
Num_personas	integer			Número de personas
Descripción	varchar(255)			Comentarios

Eventos_Equipo

Equipos elegidos para el evento.

Campo	Tipo de Dato	PK	FK	Descripción
Cve_fecha	integer	X		Clave de la fecha
Cve_equipo	varchar(5)	X		Clave del equipo

Cantidad	integer			Cantidad del equipo
----------	---------	--	--	---------------------

Eventos_Fechas

Fechas elegidas para el evento

Campo	Tipo de Dato	PK	FK	Descripción
Cve_fecha	integer	X		Clave de la fecha
Cve_evento	varchar(8)		X	Clave del evento
Cve_sala	varchar(5)		X	Clave de la sala
Fecha	date			Fecha del evento
Hora_Inicio	date			Hora de inicio
Hora_Fin	date			Hora de terminación
Cve_montaje	varchar(5)		X	Clave de montaje
Descripción	blob			Comentarios
Seguridad	blob			Comentarios sobre seguridad

Montaje

Catálogo de montajes.

Campo	Tipo de Dato	PK	FK	Descripción
Cve_montaje	varchar(5)	X		Clave del montaje
Nombre	varchar(255)			Nombre del montaje

Niveles

Catálogo de niveles.

Campo	Tipo de Dato	PK	FK	Descripción
Cve_nivel	integer	X		Clave del nivel
Nombre	varchar(255)			Nombre del nivel

Salas

Catálogo de salas.

Campo	Tipo de Dato	PK	FK	Descripción
Cve_sala	varchar(5)	X		Clave de la sala
Nombre	varchar(255)			Nombre de la sala
Capacidad	integer			Capacidad de la sala

Semestres

Catálogo de semestres.

Campo	Tipo de Dato	PK	FK	Descripción
Cve_semestre	char(4)	X		Clave del semestre

Usuarios

Catálogo de usuarios.

Campo	Tipo de Dato	PK	FK	Descripción
Usuario	char(20)	X		Nombre del usuario
Pwd	char(50)			Clave de acceso del usuario
Administrador	char(1)			Perfil del usuario
Permiso	char(1)			Privilegios del usuario

4.2.3 Diagrama Entidad Relación

El diagrama de entidad relación es una parte fundamental en el diseño de bases de datos relacionales, ya que por medio de éste podemos definir cada una de las entidades que van a formar parte de nuestra base de datos así como los atributos que contendrán.

Una base de datos que se ajusta a un diagrama entidad relación puede representarse por medio de una colección de tablas. Por cada conjunto de entidades y por cada conjunto de relaciones existe una tabla única. Cada tabla tiene un número de columnas con nombres únicos.

El diagrama entidad relación para el sistema de administración de eventos se muestra en la figura 4.2.3.1, en él se describen las entidades que forman parte de la aplicación así como sus atributos y relaciones.

El sistema constará de catorce entidades como se puede apreciar en el diagrama, las cuales se describen a continuación:

Entidad Usuarios

Función

Validar el acceso al sistema.

Relacionada con la Entidad

Ninguna.

Tipo de Relación

Ninguna.

Descripción de la Relación

Ninguna.

Entidad Accesorios

Función

Almacenar el catálogo de accesorios.

Relacionada con la Entidad

Eventos Accesorios.

Tipo de Relación

Muchos a Uno.

Descripción de la Relación

Cada accesorio debe estar asignado a uno y solo un evento.

Entidad Alimentos

Función

Almacenar el catálogo de alimentos.

Relacionada con la Entidad

Eventos Alimentos.

Tipo de Relación

Muchos a Uno.

Descripción de la Relación

Cada alimento debe estar asignado a uno y solo un evento.

Entidad Dirección División

Función

Almacenar el catálogo de divisiones.

Relacionada con la Entidad

Eventos.

Tipo de Relación

Uno a Muchos.

Descripción de la Relación

Cada división puede tener uno o mas eventos.

Entidad Equipo

Función

Almacenar el catálogo de equipos.

Relacionada con la Entidad

Eventos Equipo.

Tipo de Relación

Muchos a Uno.

Descripción de la Relación

Cada equipo debe estar asignado a uno y solo un evento.

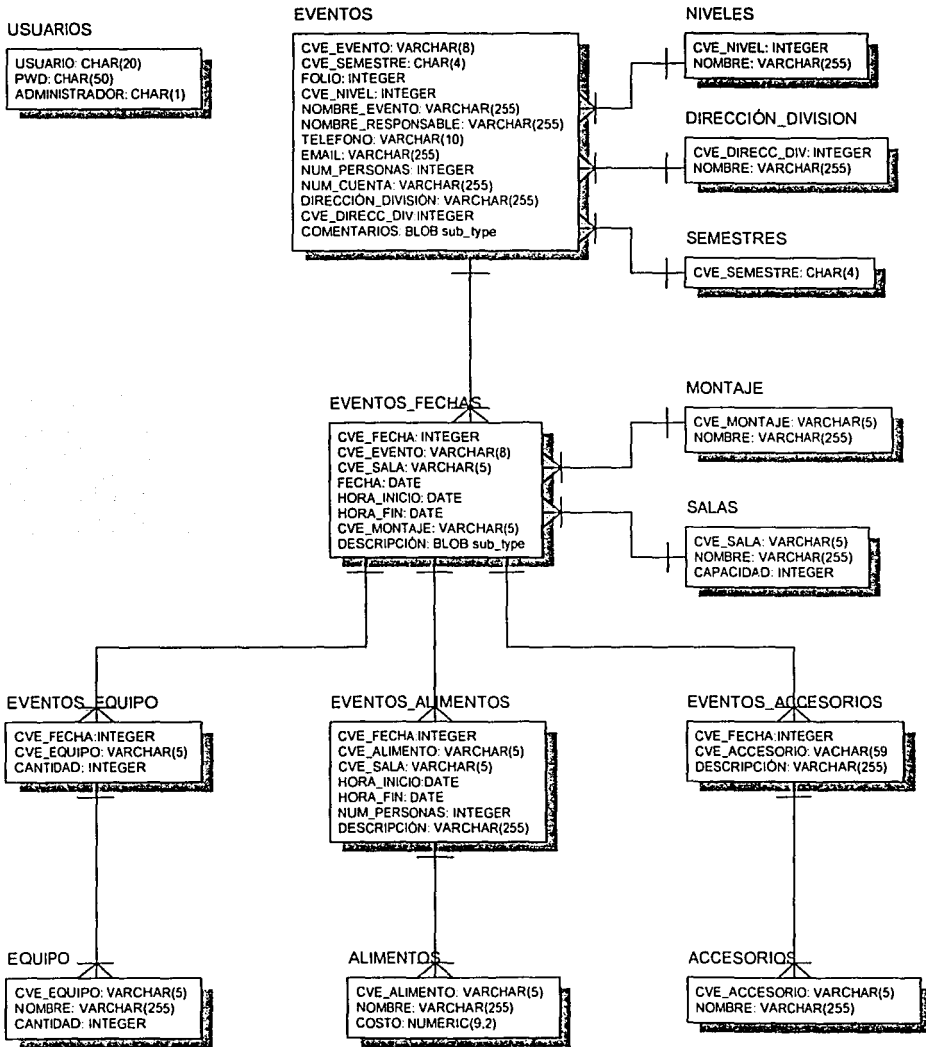


Figura 4.2.3.1 Diagrama Entidad Relación.

Entidad Montaje

Función

Almacenar el catálogo de montajes.

Relacionada con la Entidad

Eventos Fechas.

Tipo de Relación

Uno a Muchos.

Descripción de la Relación

Cada montaje puede ser seleccionado por uno o mas eventos.

Entidad Niveles

Función

Almacenar el catálogo de niveles.

Relacionada con la Entidad

Eventos.

Tipo de Relación

Uno a Muchos.

Descripción de la Relación

Cada nivel puede ser seleccionado por uno o mas eventos.

Entidad Salas

Función

Almacenar el catálogo de salas.

Relacionada con la Entidad

Eventos Fechas.

Tipo de Relación

Uno a Muchos.

Descripción de la Relación

Cada sala puede ser seleccionada por uno o mas eventos.

Entidad Semestres

Función

Almacenar el catálogo de semestres.

Relacionada con la Entidad

Eventos.

Tipo de Relación

Uno a Muchos.

Descripción de la Relación

Cada semestre puede tener uno o mas eventos.

Entidad Eventos Accesorios

Función

Almacenar los accesorios seleccionados para los eventos.

Relacionada con la Entidad

Accesorios.

Tipo de Relación

Uno a Muchos.

Descripción de la Relación

Cada evento puede tener uno o mas accesorios.

Relacionada con la Entidad

Eventos Fechas.

Tipo de Relación

Muchos a Uno.

Descripción de la Relación

Los accesorios deben estar asignados a uno y solo un evento.

Entidad Eventos Alimentos

Función

Almacenar los alimentos seleccionados para los eventos.

Relacionada con la Entidad

Alimentos.

Tipo de Relación

Uno a Muchos.

Descripción de la Relación

Cada evento puede tener uno o mas alimentos.

Relacionada con la Entidad

Eventos Fechas.

Tipo de Relación

Muchos a Uno.

Descripción de la Relación

Los alimentos deben estar asignados a uno y solo un evento.

Entidad Eventos Equipo

Función

Almacenar los equipos seleccionados para los eventos.

Relacionada con la Entidad

Equipo.

Tipo de Relación

Uno a Muchos.

Descripción de la Relación

Cada evento puede tener uno o mas equipos.

Relacionada con la Entidad

Eventos Fechas.

Tipo de Relación

Muchos a Uno.

Descripción de la Relación

Los equipos deben estar asignados a uno y solo un evento.

Entidad Eventos Fechas

Función

Almacenar las fechas seleccionadas para los eventos.

Relacionada con la Entidad

Montaje.

Tipo de Relación

Muchos a Uno.

Descripción de la Relación

Cada fecha del evento debe tener asignado uno y solo un montaje.

Relacionada con la Entidad

Salas.

Tipo de Relación

Muchos a Uno.

Descripción de la Relación

Cada fecha del evento debe tener asignada una y solo una sala.

Relacionada con la Entidad

Eventos Equipo.

Tipo de Relación

Uno a Muchos.

Descripción de la Relación

Cada fecha del evento puede tener uno o mas equipos.

Relacionada con la Entidad

Eventos Alimentos.

Tipo de Relación

Uno a Muchos.

Descripción de la Relación

Cada fecha del evento puede tener uno o mas alimentos.

Relacionada con la Entidad
Eventos Accesorios.

Tipo de Relación
Uno a Muchos.

Descripción de la Relación
Cada fecha del evento puede tener uno o mas accesorios.

Entidad Eventos

Función
Almacenar la información de los eventos.

Relacionada con la Entidad
Niveles.

Tipo de Relación
Muchos a Uno.

Descripción de la Relación
Cada evento debe estar asignado a uno y solo un nivel.

Relacionada con la Entidad
Dirección División.

Tipo de Relación
Muchos a Uno.

Descripción de la Relación
Cada evento debe estar asignado a una y solo una división.

Relacionada con la Entidad
Semestres.

Tipo de Relación
Muchos a Uno.

Descripción de la Relación
Cada evento debe estar asignado a uno y solo un semestre.

Relacionada con la Entidad
Eventos Fechas.

Tipo de Relación
Uno a Muchos.

Descripción de la Relación
Cada evento puede tener una o mas fechas.

4.2.4 Normalización

Uno de los factores mas importantes en la creación de sistemas es el diseño de las bases de datos. Si las tablas no están correctamente diseñadas, pueden causar

problemas cuando se tengan que realizar complicadas llamadas SQL en el código de la aplicación para extraer los datos que se necesitan.

Básicamente, las reglas de normalización están encaminadas a eliminar redundancias e inconsistencias de dependencia en el diseño de las tablas.

Primera Forma Normal:

1. Eliminar los grupos repetitivos de la tablas individuales.
2. Crear una tabla separada por cada grupo de datos relacionados.
3. Identificar cada grupo de datos relacionados con una llave primaria.

Segunda Forma Normal:

1. Crear tablas separadas para aquellos grupos de datos que se aplican a varios registros.
2. Relacionar estas tablas mediante una llave externa.

Tercera Forma Normal:

1. Eliminar aquellos campos que no dependan de la llave.

Para normalizar los datos que intervienen en el sistema se aplicaron las tres primeras formas normales de la siguiente manera.

En la figura 4.2.4.1 se muestra la tabla de eventos, esta es una entidad que requiere de normalización, con el fin de poder utilizar su información en forma eficiente.

Eventos

cve_evento
folio
nivel
nombre_evento
nombre_responsable
telefono
e-mail
num_personas
num_cuenta
dirección_división
fecha
sala
hora_inicio
hora_fin

Figura 4.2.4.1 Tabla Eventos.

Todos los campos de los eventos tienen su propio dominio, por lo que para cada registro sólo existe un valor para cada campo, se puede concluir que esta tabla ya se encuentra en la primera forma normal por lo que cumple con la definición de esta forma, que dice, "Una tabla se encuentra en la primera forma normal cuando todas las relaciones de la base de datos cumplen con la propiedad de tener un valor en cada uno de sus atributos".

Ahora considerando la definición de la segunda forma normal, que dice, "Una relación está en segunda forma normal si, y sólo si, está en primera forma normal y todos los atributos que no son llave, dependen completamente de la llave primaria".

Como podemos observar el teléfono, e-mail y número de cuenta no dependen de la llave primaria sino más bien del responsable del evento; así como la sala, hora de inicio y hora de terminación dependen de la fecha, por lo que crearemos dos tablas nuevas, una que contenga los datos de los responsables de los eventos y otra que contenga las fechas en las que se llevarán a cabo los eventos, como se muestra en las figuras 4.2.4.2, 4.2.4.3 y 4.2.4.4.

Eventos

cve_evento
folio
nivel
nombre
num_personas
dirección_división

Figura 4.2.4.2. Tabla Eventos.

Responsables_Eventos

cve_respondable_evento
cve_evento
nombre
num_cuenta
e-mail
telefono

Figura 4.2.4.3. Tabla Responsables_Eventos.

Eventos_Fechas

cve_fecha
cve_evento
sala
fecha
hora_inicio
hora_fin

Figura 4.2.4.4. Tabla Eventos_Fechas.

Ahora si, todos los campos, como se especifica en la definición, ya dependen de la llave primaria y con esto se puede concluir que cumplen con la segunda forma normal.

Continuando con la normalización de las tablas, el siguiente paso es llevarlas a la tercera forma normal, que dice, "Una relación esta en tercera forma normal si, y sólo si, está en segunda forma normal y todos los atributos que no son llave principal, son mutuamente independientes".

Como podemos ver nivel y dirección-división en la tabla eventos no son mutuamente independientes por lo que crearemos una tabla nueva para cada uno de estos campos, como se muestra en la figura 4.2.4.5, 4.2.4.6 y 4.2.4.7.

Eventos

cve_evento
cve_nivel
cve_direcc_div
folio
nombre
num_personas

Figura 4.2.4.5. Tabla Eventos.

Niveles

cve_nivel
nombre

Figura 4.2.4.6. Tabla Niveles.

Dirección_Division

cve_direcc_div
nombre

Figura 4.2.4.7. Tabla Dirección_División.

Después de esta descomposición, las tablas se encuentran en la tercera forma normal por no tener dependencia funcional entre los campos que no son llave principal.

De esta forma se normalizaron cada una de las tablas que conforman el sistema, dando como resultado que nuestro esquema de datos pueda manejar fácilmente los datos de la aplicación.

4.3 Diseño y desarrollo de la base de datos

En esta sección vamos a presentar el procedimiento que nos llevó a la elaboración de la base de datos del sistema que aquí se presenta. Las herramientas utilizadas son Interbase, la cual básicamente podemos decir que es el servidor de bases de datos que utilizamos en el desarrollo de nuestro sistema; y por otro lado, la segunda herramienta utilizada es EMS QuickDesk con la cual hicimos toda la estructura de nuestra base de datos: definición de tablas, definición de llaves primarias, definición de generadores, definición de funciones externas, definición de índices y definición de procedimientos de almacenamiento.

Así entonces, vamos a iniciar utilizando EMS QuickDesk para crear la estructura de nuestra base de datos. El primer paso es crear una nueva base de datos (ver figura 4.3.1).

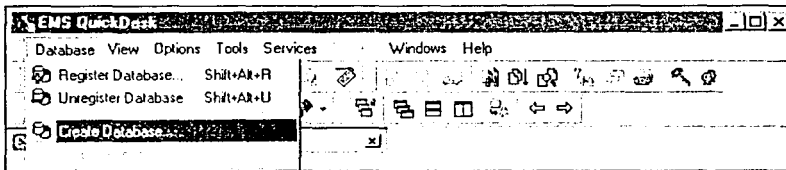


Figura 4.3.1 Nueva base de datos.

La pantalla que se muestra para la creación de una nueva base de datos en EMS QuickDesk tiene tres secciones: opciones generales, archivos secundarios y crear base de datos.

En la primera sección que es la de opciones generales debemos especificar el nombre con el que vamos a identificar a nuestra base de datos así como la ubicación física donde la vamos a guardar además de un nombre de usuario con el que vamos acceder nuestra información por lo que estos datos deben ser significativos del trabajo que se va a desarrollar (ver figura 4.3.2).

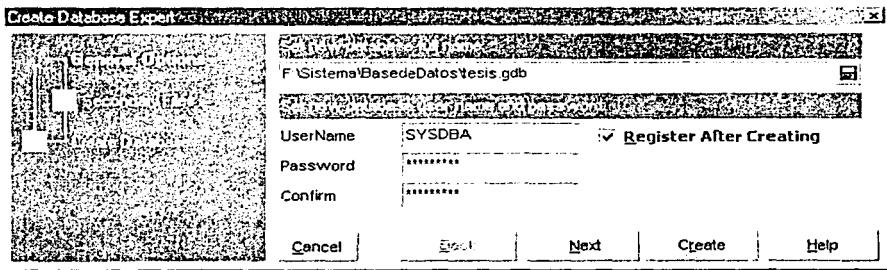


Figura 4.3.2 Opciones Generales.

En la sección de archivos secundarios se determina la manera en como va a interactuar nuestra base de datos, es decir si va a tener relación con otros archivos de base de datos o va a trabajar de manera independiente (ver figura 4.3.3). Para nuestro caso en particular le indicaremos que va a ser una base de datos independiente.

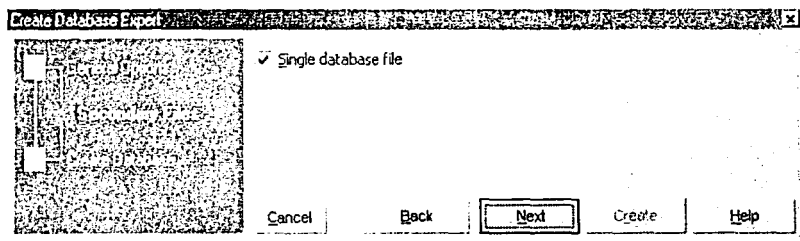


Figura 4.3.3 Archivos secundarios.

En la tercera y última sección se muestran los datos que se configuraron en las secciones anteriores: nombre de la base de datos, usuario, tamaño de página y grupo de caracteres (ver figura 4.3.4).

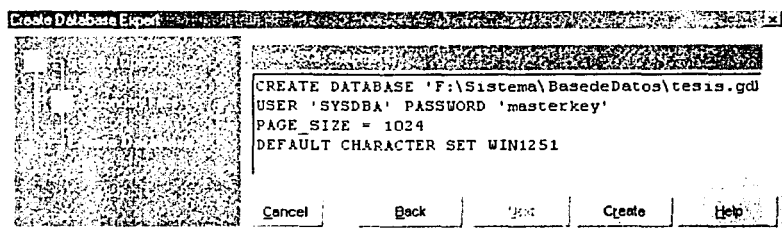


Figura 4.3.4 Crear base de datos.

El siguiente paso es registrar la base de datos que acabamos de crear, para ello contamos con cinco secciones: general, adicional, opciones de salvado, bitácoras y opciones adicionales.

En la sección general debemos indicar la localización de la base de datos (local o remota), la ubicación física de la base de datos, el alias de la base de datos, el nombre de usuario, la clave de acceso del usuario y el conjunto de caracteres que se utilizarán como predeterminados (ver figura 4.3.5).

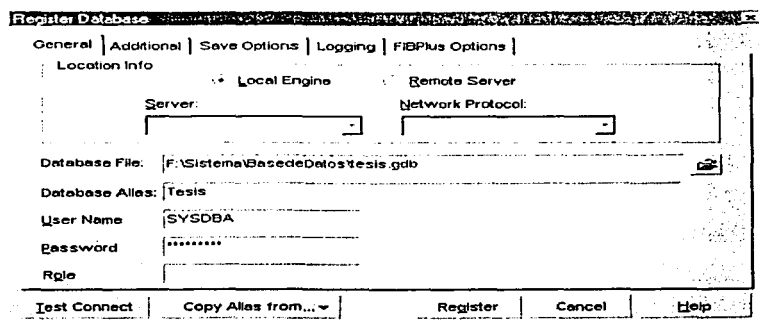


Figura 4.3.5 Registro de la base de datos.

En la sección adicional se deben configurar las opciones del explorador de la base de datos, del editor SQL y de Interbase (ver figura 4.3.6).

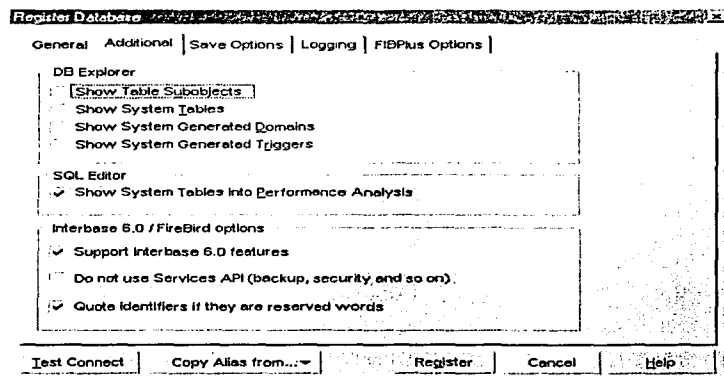


Figura 4.3.6 Sección adicional.

La sección de opciones adicionales nos permite configurar el soporte de campos booleanos, los campos de sólo lectura, la obtención de valores predeterminados, etc. (ver figura 4.3.7).

Una vez configurados los datos correspondientes para cada sección procederemos al registro de la base de datos.

El programa EMS QuickDesk cuenta con un editor de SQL con el que podremos generar toda la estructura de nuestra base de datos: tablas, llaves primarias, generadores, funciones externas, índices y procedimientos de almacenamiento. El primer paso para esto es identificar el icono que nos permite ejecutar el editor de SQL, el cual como podemos ver en la figura 4.3.8 se encuentra en la parte superior de la barra de herramientas.

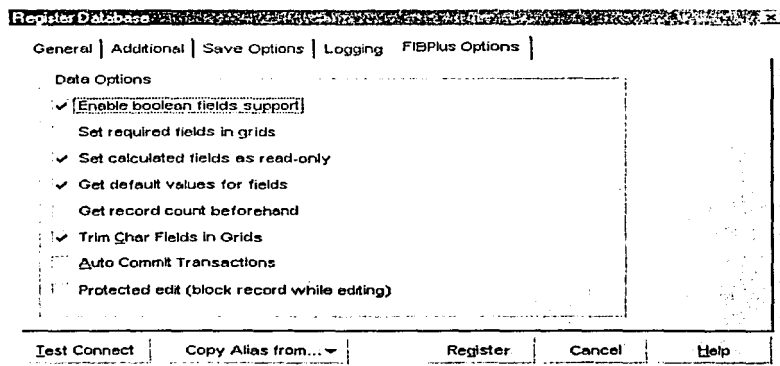


Figura 4.3.7 Opciones adicionales.

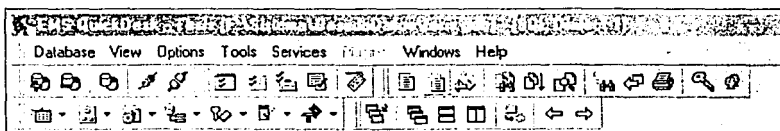
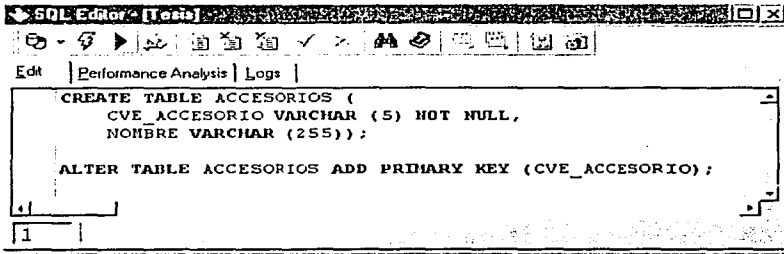


Figura 4.3.8 Ejecutar editor de SQL.

Una vez ejecutado el editor de SQL procederemos a la creación de las tablas que formarán parte de nuestra base de datos.

La primera tabla que crearemos será la de accesorios, la cual esta formada por los siguientes campos: `cve_accesorio` (clave del accesorio) y `nombre` (nombre del accesorio) (ver figura 4.3.9).



```

SQL Editor - [tesis]
Edit | Performance Analysis | Logs |
CREATE TABLE ACCESORIOS (
  CVE_ACCESORIO VARCHAR (5) NOT NULL,
  NOMBRE VARCHAR (255));

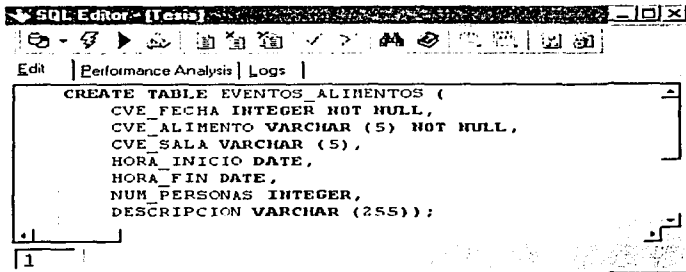
ALTER TABLE ACCESORIOS ADD PRIMARY KEY (CVE_ACCESORIO);

```

Figura 4.3.9 Creación de la tabla accesorios.

La segunda tabla que crearemos será la de alimentos, la cual esta formada por los siguientes campos: `cve_alimento` (clave del alimento), `nombre` (nombre del alimento) y `costo` (precio del alimento) (ver figura 4.3.10).

La siguiente tabla que crearemos será la de `direccion_división`, la cual esta formada por los siguientes campos: `cve_direcc_div` (clave de la división) y `nombre` (nombre de la división) (ver figura 4.3.11). De esta manera se crearon cada una de las tablas que conforman el sistema.



```

SQL Editor - [tesis]
Edit | Performance Analysis | Logs |
CREATE TABLE EVENTOS_ALIMENTOS (
  CVE_FECHA INTEGER NOT NULL,
  CVE_ALIMENTO VARCHAR (5) NOT NULL,
  CVE_SALA VARCHAR (5),
  HORA_INICIO DATE,
  HORA_FIN DATE,
  NUM_PERSONAS INTEGER,
  DESCRIPCION VARCHAR (255));

```

Figura 4.3.10 Creación de la tabla alimentos.

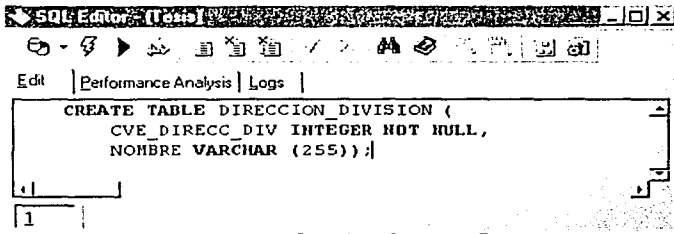


Figura 4.3.11 Creación de la tabla dirección_división.

Una vez creadas todas las tablas, el siguiente paso es definir las llaves primarias. El campo que es usado para identificar de manera única cada registro de la tabla se llama llave primaria.

En la tabla de accesorios vamos a definir como llave primaria el campo `cve_accesorio` (ver figura 4.3.12).

Para la tabla de alimentos vamos a definir como llave primaria el campo `cve_alimento` (ver figura 4.3.13).

La siguiente llave primaria que vamos a definir es la que corresponde al campo `cve_direcc_div` de la tabla `direccion_division` (ver figura 4.3.14). De este forma se definieron las llaves primarias restantes.

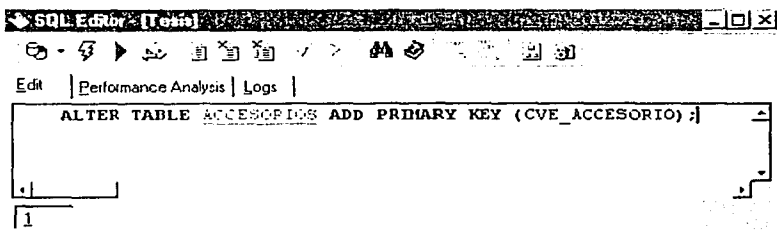


Figura 4.3.12 Definición de la llave primaria para la tabla accesorios.

Continuando con la creación de la base de datos, el siguiente paso es crear los índices. El empleo adecuado de índices acelera el acceso a la información, pero consume espacio considerable, es por esto que vale la pena hacer un análisis cuidadoso de cuáles campos requieren ser indexados.

El primer campo que definiremos como índice será el de `cve_equipo` de la tabla `eventos_equipo` (ver figura 4.3.15).

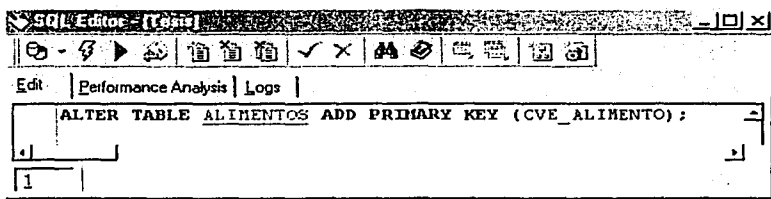


Figura 4.3.13 Definición de la llave primaria para la tabla `alimentos`.

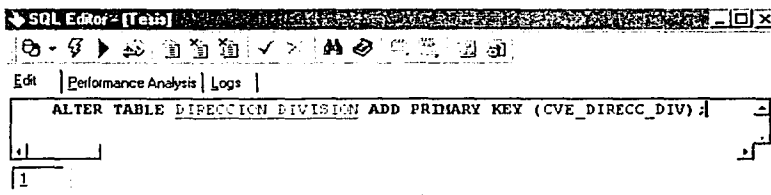


Figura 4.3.14 Definición de la llave primaria para la tabla `dirección_división`.

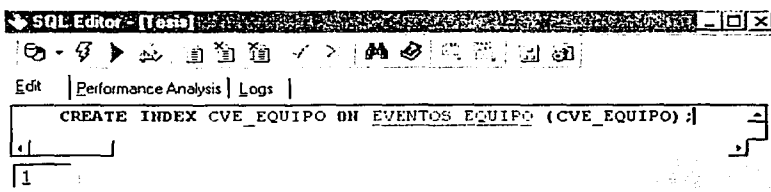


Figura 4.3.15 Creación del índice para la tabla `eventos_equipo`.

El segundo campo que definiremos como índice será el de `cve_evento` de la tabla `eventos_fecha` (ver figura 4.3.16).

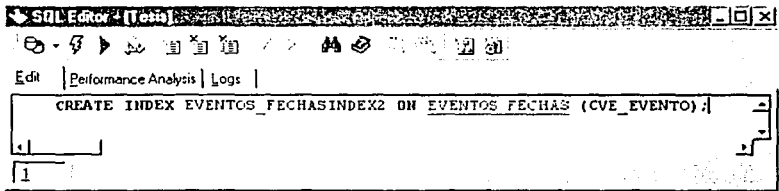


Figura 4.3.16 Creación del índice para la tabla `eventos_fecha`.

El siguiente campo que definiremos como índice será el de `cve_direcc_div` de la tabla `eventos` (ver figura 4.3.17). De esta se definieron cada uno de los índices que forman parte de nuestra base de datos.

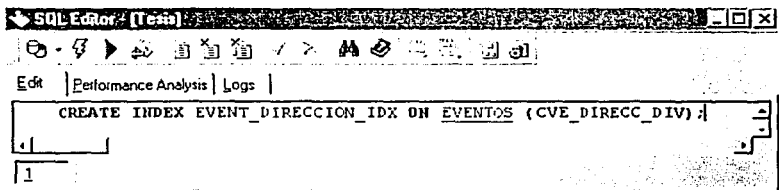


Figura 4.3.17 Creación del índice para la tabla `eventos`.

El siguiente paso en el proceso de desarrollo de la base de datos es crear los generadores. Un generador es un número secuencial que garantiza un valor único para una llave primaria.

El primer generador que crearemos será `GEN_ACCESORIOS` (ver figura 4.3.18), el cual utilizaremos para agregar registros a la tabla de accesorios.

El segundo generador que crearemos será `GEN_ALIMENTOS` (ver figura 4.3.19), el cual utilizaremos para agregar registros a la tabla de alimentos.

El siguiente generador que crearemos será GEN_EQUIPO (ver figura 4.3.20), el cual utilizaremos para agregar registros a la tabla de equipo.

De esta forma se crearon cada uno de los generadores que forman parte de nuestra base de datos.

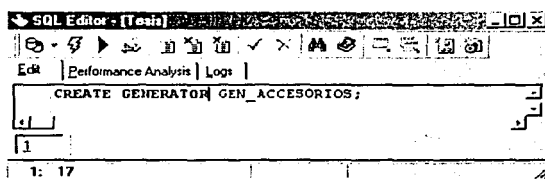


Figura 4.3.18 Creación del generador para la tabla accesorios.

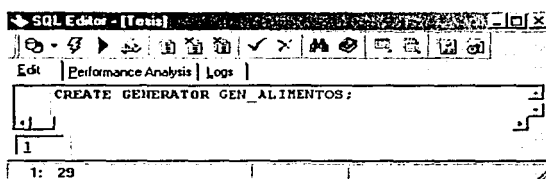


Figura 4.3.19 Creación del generador para la tabla alimentos.

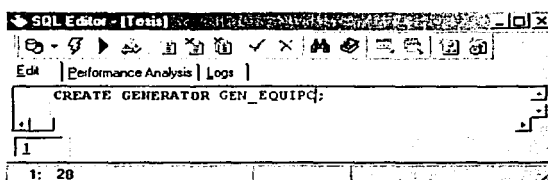
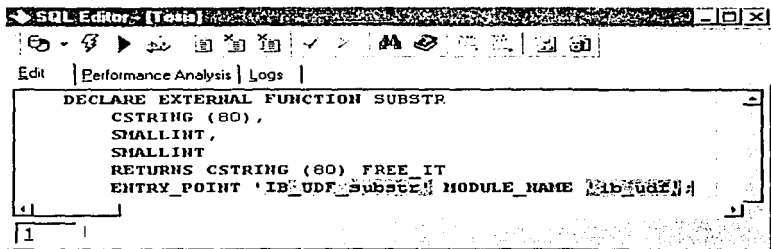


Figura 4.3.20 Creación del generador para la tabla equipo.

A continuación crearemos la función externa SUBSTR (ver figura 4.3.21), la cual utilizaremos para obtener la clave del semestre y el número de folio a partir de la clave del evento.

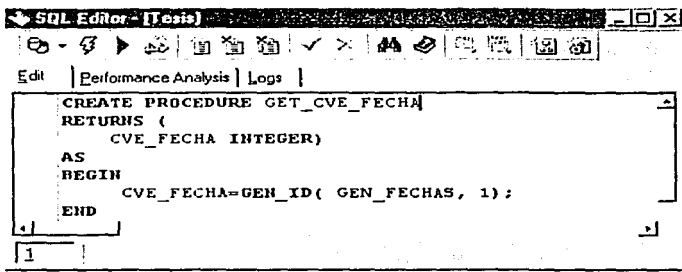


```

SQL Editor - [Tesis]
Edit | Performance Analysis | Logs |
DECLARE EXTERNAL FUNCTION SUBSTR
  CSTRING (80),
  SMALLINT,
  SMALLINT
  RETURNS CSTRING (80) FREE IT
  ENTRY_POINT 'IB_UDF_SUBSTR' MODULE_NAME 'IB_UDF';
  
```

Figura 4.3.21 Creación de la función externa SUBSTR.

Por último crearemos el procedimiento almacenado GET_CVE_FECHA (ver figura 4.3.22). Este procedimiento nos servirá para incrementar el generador GEN_FECHAS antes de insertar un nuevo registro en la tabla eventos_fecha.



```

SQL Editor - [Tesis]
Edit | Performance Analysis | Logs |
CREATE PROCEDURE GET_CVE_FECHA
  RETURNS (
    CVE_FECHA INTEGER)
  AS
  BEGIN
    CVE_FECHA=GEN_ID( GEN_FECHAS, 1);
  END
  
```

Figura 4.3.22 Creación del procedimiento almacenado GET_CVE_FECHA.

A continuación se muestra el código completo para la creación de la base de datos del sistema.

```
SET SQL DIALECT 1;
```

```
/* CREATE DATABASE 'localhost:C:\Tesis\BasedeDatos\Csi_db.gdb' PAGE_SIZE 1024 DEFAULT CHARACTER SET */
```

/* External Function declarations */

```
DECLARE EXTERNAL FUNCTION SUBSTR
CSTRING(80) CHARACTER SET NONE, SMALLINT, SMALLINT
RETURNS CSTRING(80) CHARACTER SET NONE FREE_IT
ENTRY_POINT 'IB_UDF_substr' MODULE_NAME 'ib_udf';
```

/* Table: ACCESORIOS */

```
CREATE TABLE ACCESORIOS
(
  CVE_ACCESORIO  VARCHAR(5) NOT NULL,
  NOMBRE        VARCHAR(255),
  PRIMARY KEY (CVE_ACCESORIO)
);
```

/* Table: ALIMENTOS */

```
CREATE TABLE ALIMENTOS
(
  CVE_ALIMENTO  VARCHAR(5) NOT NULL,
  NOMBRE        VARCHAR(255),
  COSTO         NUMERIC(9, 2) DEFAULT 0 NOT NULL,
  PRIMARY KEY (CVE_ALIMENTO)
);
```

/* Table: DIRECCION_DIVISION */

```
CREATE TABLE DIRECCION_DIVISION
(
  CVE_DIRECC_DIV  INTEGER NOT NULL,
  NOMBRE        VARCHAR(255),
  PRIMARY KEY (CVE_DIRECC_DIV)
);
```

/* Table: EQUIPO */

```
CREATE TABLE EQUIPO
(
  CVE_EQUIPO     VARCHAR(5) NOT NULL,
  NOMBRE        VARCHAR(255),
  CANTIDAD      INTEGER,
  PRIMARY KEY (CVE_EQUIPO)
);
```

/* Table: EVENTOS */

```
CREATE TABLE EVENTOS
(
  CVE_EVENTO     VARCHAR(8) NOT NULL,
  CVE_SEMESTRE  CHAR(4) NOT NULL,
```

```

FOLIO                INTEGER NOT NULL,
CVE_NIVEL            INTEGER,
NOMBRE_EVENTO       VARCHAR(255),
NOMBRE_RESPONSABLE  VARCHAR(255),
TELEFONO            VARCHAR(10),
EMAIL               VARCHAR(255),
NUM_PERSONAS        INTEGER,
NUM_CUENTA          VARCHAR(255),
DIRECCION_DIVISION  VARCHAR(255),
CVE_DIRECC_DIV      INTEGER,
COMENTARIOS         BLOB SUB_TYPE TEXT SEGMENT SIZE 80,
PRIMARY KEY (CVE_EVENTO)
);

```

/* Table: EVENTOS_ACCESORIOS */

```

CREATE TABLE EVENTOS_ACCESORIOS
(
  CVE_FECHA INTEGER NOT NULL,
  CVE_ACCESORIO VARCHAR(5) NOT NULL,
  DESCRIPCION VARCHAR(255),
  CONSTRAINT EVENTOS_ACCESORIOSPRIMARYKEY1 PRIMARY KEY (CVE_FECHA,
  CVE_ACCESORIO)
);

```

/* Table: EVENTOS_ALIMENTOS */

```

CREATE TABLE EVENTOS_ALIMENTOS
(
  CVE_FECHA INTEGER NOT NULL,
  CVE_ALIMENTO VARCHAR(5) NOT NULL,
  CVE_SALA VARCHAR(5),
  HORA_INICIO TIMESTAMP,
  HORA_FIN TIMESTAMP,
  NUM_PERSONAS INTEGER,
  DESCRIPCION VARCHAR(255),
  CONSTRAINT E_AL_PRIMARYKEY1 PRIMARY KEY (CVE_FECHA, CVE_ALIMENTO)
);

```

/* Table: EVENTOS_EQUIPO */

```

CREATE TABLE EVENTOS_EQUIPO
(
  CVE_FECHA INTEGER NOT NULL,
  CVE_EQUIPO VARCHAR(5) NOT NULL,
  CANTIDAD INTEGER,
  CONSTRAINT EVENTOS_EQUIPOPRIMARYKEY1 PRIMARY KEY (CVE_FECHA, CVE_EQUIPO)
);

```

/* Table: EVENTOS_FECHAS */

```
CREATE TABLE EVENTOS_FECHAS
(
  CVE_FECHA      INTEGER NOT NULL,
  CVE_EVENTO     VARCHAR(8),
  CVE_SALA       VARCHAR(5),
  FECHA          TIMESTAMP,
  HORA_INICIO    TIMESTAMP,
  HORA_FIN       TIMESTAMP,
  CVE_MONTAJE    VARCHAR(5),
  DESCRIPCION    BLOB SUB_TYPE TEXT SEGMENT SIZE 80,
  SEGURIDAD      BLOB SUB_TYPE TEXT SEGMENT SIZE 80,
  CONSTRAINT EVENTOS_FECHASPRIMARYKEY1 PRIMARY KEY (CVE_FECHA)
);
```

/* Table: MONTAJE */

```
CREATE TABLE MONTAJE
(
  CVE_MONTAJE    VARCHAR(5) NOT NULL,
  NOMBRE         VARCHAR(255),
  PRIMARY KEY (CVE_MONTAJE)
);
```

/* Table: NIVELES */

```
CREATE TABLE NIVELES
(
  CVE_NIVEL     INTEGER NOT NULL,
  NOMBRE        VARCHAR(255),
  PRIMARY KEY (CVE_NIVEL)
);
```

/* Table: SALAS */

```
CREATE TABLE SALAS
(
  CVE_SALA      VARCHAR(5) NOT NULL,
  NOMBRE        VARCHAR(255),
  CAPACIDAD     INTEGER,
  PRIMARY KEY (CVE_SALA)
);
```

/* Table: SEMESTRES */

```
CREATE TABLE SEMESTRES
(
  SEMESTRE      CHAR(4) NOT NULL,
  CONSTRAINT SEMESTRES_NDX PRIMARY KEY (SEMESTRE)
);
```

/* Table: USUARIOS */

```
CREATE TABLE USUARIOS
(
  USUARIO          CHAR(20) NOT NULL,
  PWD              CHAR(50),
  ADMINISTRADOR    CHAR(1) DEFAULT 'N' NOT NULL,
  PERMISO          CHAR(1) DEFAULT 'R' NOT NULL,
  CONSTRAINT USUARIOS_NDX PRIMARY KEY (USUARIO)
);
```

/* Index definitions for all user tables */

```
CREATE INDEX EVENT_DIRECCION_IDX ON EVENTOS(CVE_DIRECC_DIV);
CREATE INDEX EVENT_FOL_IDX ON EVENTOS(FOLIO);
CREATE INDEX EVENT_NIVEL_IDX ON EVENTOS(CVE_NIVEL);
CREATE INDEX EVENT_SEM_IDX ON EVENTOS(CVE_SEMESTRE);
CREATE INDEX E_A_CVE_ACCESORIO_NDX ON EVENTOS_ACCESORIOS(CVE_ACCESORIO);
CREATE INDEX E_A_CVE_FECHA_NDX ON EVENTOS_ACCESORIOS(CVE_FECHA);
CREATE INDEX E_AL_CVE_ALIMENTO ON EVENTOS_ALIMENTOS(CVE_ALIMENTO);
CREATE INDEX E_AL_CVE_FECHA ON EVENTOS_ALIMENTOS(CVE_FECHA);
CREATE INDEX E_AL_CVE_SALA ON EVENTOS_ALIMENTOS(CVE_SALA);
CREATE INDEX CVE_EQUIPO ON EVENTOS_EQUIPO(CVE_EQUIPO);
CREATE INDEX E_EQ_CVE_FECHA ON EVENTOS_EQUIPO(CVE_FECHA);
CREATE INDEX EVENTOS_FECHASINDEX2 ON EVENTOS_FECHAS(CVE_EVENTO);
CREATE GENERATOR GEN_ACCESORIOS;
CREATE GENERATOR GEN_ALIMENTOS;
CREATE GENERATOR GEN_DIRECCION_DIVISION;
CREATE GENERATOR GEN_EQUIPO;
CREATE GENERATOR GEN_FECHAS;
CREATE GENERATOR GEN_MONTAJES;
CREATE GENERATOR GEN_NIVELES;
CREATE GENERATOR GEN_SALAS;
CREATE GENERATOR G_0002;
CREATE GENERATOR G_0008;
CREATE GENERATOR G_0102;
CREATE GENERATOR G_0108;
CREATE GENERATOR G_0202;
CREATE GENERATOR G_0208;
COMMIT WORK;
SET AUTODDL OFF;
SET TERM ^;
```

/* Stored procedures */

```
CREATE PROCEDURE GET_CVE_FECHA
RETURNS
(
  CVE_FECHA INTEGER
)
AS
BEGIN EXIT; END ^
```

```
ALTER PROCEDURE GET_CVE_FECHA
RETURNS
(
  CVE_FECHA INTEGER
)
AS
BEGIN
  CVE_FECHA=GEN_ID( GEN_FECHAS, 1);
END
^

SET TERM ; ^
COMMIT WORK;
SET AUTODDL ON;
```

4.4 Diseño y desarrollo de las interfaces gráficas

El diseño de las interfaces gráficas se tiene que hacer de acuerdo a los requerimientos del usuario para facilitar la operación del sistema.

Para la construcción de las interfaces gráficas del sistema de administración de eventos se utilizó FrontPage versión 2000.



Figura 4.4.1 Vistas de Frontpage.

En la figura 4.4.1 se muestran las vistas de FrontPage las cuales nos ayudaron a organizar y administrar nuestro sitio web. A continuación se describe cada una de ellas.

Vista Exploración

Es tal vez donde mejor podemos visualizar el sitio, ya que nos muestra su estructura en forma de organigrama, en la parte superior tenemos la página principal (con un icono de una casita) y debajo de esta, todas las páginas secundarias ramificadas. Podemos colocar nuestras páginas, o no, en la vista exploración (arrastrándolas de la lista de

carpetas o seleccionando nueva página con el menú del botón derecho del mouse) pero si alguna no se coloca, esta no aparecerá en las barras de exploración que coloquemos en las páginas (una barra de exploración es una lista automática con hipervínculos a las demás páginas).

Vista Página

Esta es la vista donde se arma y se da formato a la página, el proceso es similar a editar un documento en Word, ya que las herramientas son parecidas.

Dentro de esta vista existen tres carpetas: Normal, HTML y Vista Previa.

La vista Normal es en donde editaremos nuestra página, es decir, donde podremos agregar texto, gráficos, etc.

La vista HTML, nos será útil para cuando tengamos que hacer alguna modificación en el código de la página.

Por último la Vista Previa nos da una idea de como se va a ver la página en el explorador.

Vista Carpetas

En esta vista podemos ver las diferentes carpetas en donde guardamos los archivos. Generalmente nos conviene organizarlos para trabajar más ordenados, por ejemplo, las imágenes las guardamos todas juntas en una carpeta; también podemos guardar una sección entera de nuestro sitio en una carpeta diferente. Es importante no confundir la estructura de exploración con la estructura de las carpetas ya que no tienen que ser iguales.

Vista Informes

En esta vista podremos obtener una serie de datos muy útiles para nuestro sitio como páginas lentas, archivos no vinculados, archivos nuevos, hipervínculos rotos, etc.

Vista Hipervínculos

Esta vista nos muestra cada página con sus hipervínculos entrantes y salientes.

Vista Tareas

En esta vista podremos organizar las próximas tareas que realizaremos en nuestro sitio o en algún archivo determinado. Es una especie de agenda.

4.4.1 Procedimiento para la creación de las interfaces gráficas

Seleccionar de la barra de menú Archivo / Nuevo / Web y elegir Web Vacío, como se muestra en la figura 4.4.2.

Una vez hecho lo anterior se crearán las carpetas images y _private, y aparecerá una página en blanco, la cual cerraremos e iremos a la vista de exploración.

Una vez que estemos en la vista de exploración hacemos click derecho y seleccionamos Nueva Página, como se observa en la figura 4.4.3. Una vez creada, hacemos doble click, automáticamente pasaremos a la vista de página en donde podremos comenzar a colocarle todos los controles, ver figura 4.4.4. Si hacemos click derecho sobre la página y elegimos Cambiar Nombre podremos colocar un título a la página (que es diferente al nombre de archivo).

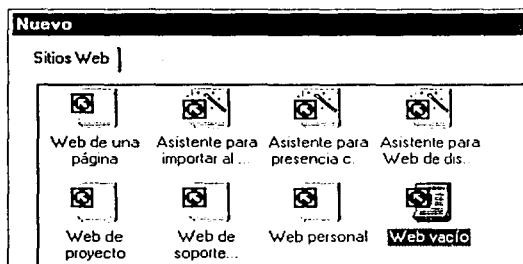


Figura 4.4.2 Nuevo web.

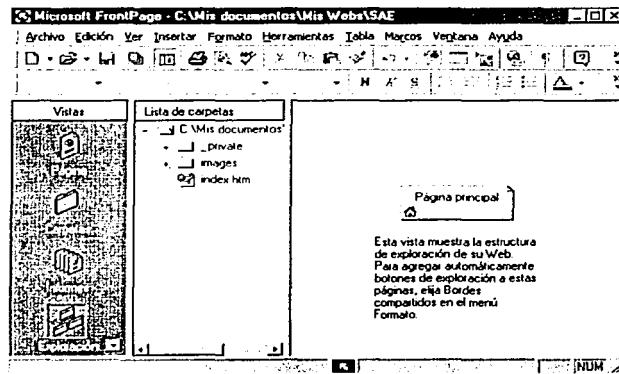


Figura 4.4.3 Nueva página.

Podemos, de este modo, definir toda la estructura de nuestro sitio para luego tener listas todas las páginas para trabajar.

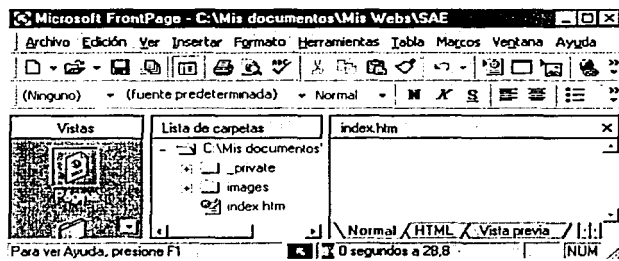


Figura 4.4.4 Vista de página.

Sobre los formularios HTML se fueron colocando los objetos necesarios para crear las interfaces gráficas, como son: etiquetas, cuadros de texto, casillas de verificación, botones de opción, menús despegables, botones de comando, imágenes, etc; con la finalidad de ingresar o visualizar los datos.

Una vez seleccionados y colocados los objetos necesarios para la creación de los formularios HTML, se configuraron las propiedades de cada uno de ellos, cada objeto lleva asociado un conjunto de propiedades, entre las que se encuentran tamaño, color, tipo de letra, etc.

Los pasos que se siguieron para la creación de la interface gráfica de inicio fueron:

1. Crear nueva página (Figura 4.4.5).

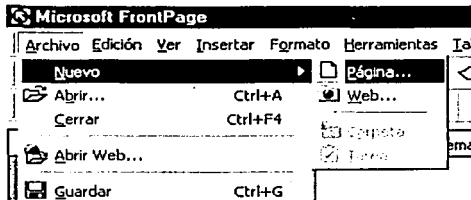


Figura 4.4.5 Crear página.

2. Editar las propiedades de fondo y título de la página (Figuras 4.4.6 y 4.4.7).

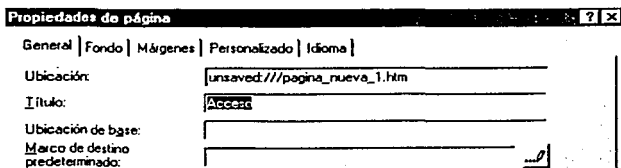


Figura 4.4.6 Propiedad de título.

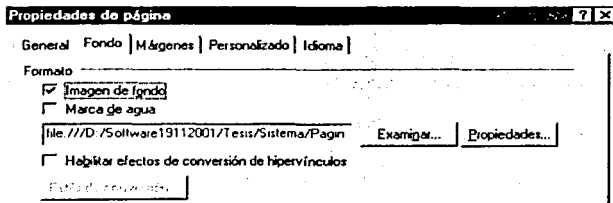


Figura 4.4.7 Propiedad de fondo.

3. Insertar imagen de centro de servicios (Figura 4.4.8).

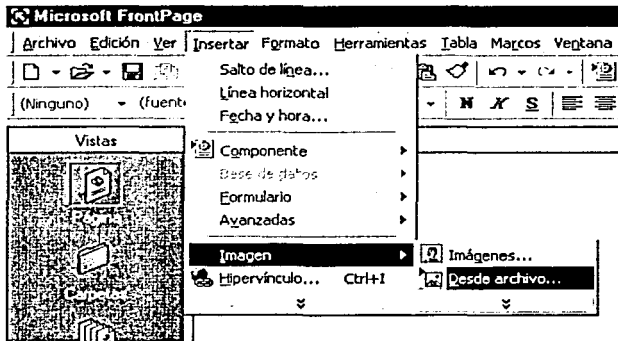


Figura 4.4.8 Insertar imagen.

4. Insertar formulario (Figura 4.4.9).

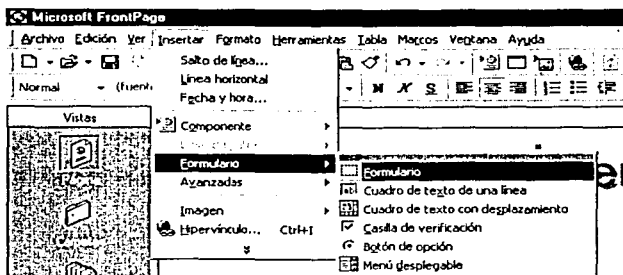


Figura 4.4.9 Insertar formulario.

5. Editar las propiedades nombre, acción y método del formulario (Figuras 4.4.10 y 4.4.11).
6. Insertar tabla de dos por dos dentro del formulario (Figura 4.4.12).
7. Escribir en la celda 1-1 usuario y en la celda 2-1 contraseña (Figura 4.4.13).
8. Insertar un cuadro de texto en la celda 1-2 y otro en la celda 2-2 (Figura 4.4.14).
9. Editar las propiedades nombre, ancho de caracteres y campo de contraseña; de los cuadros de texto para el usuario y contraseña (Figuras 4.4.15 y 4.4.16).
10. Editar las propiedades nombre, valor/etiqueta y tipo de botón; de los botones de comando del formulario (Figuras 4.4.17 y 4.4.18).

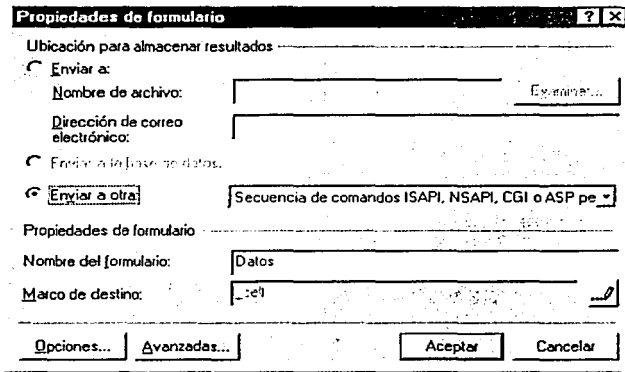


Figura 4.4.10 Propiedad nombre del formulario.

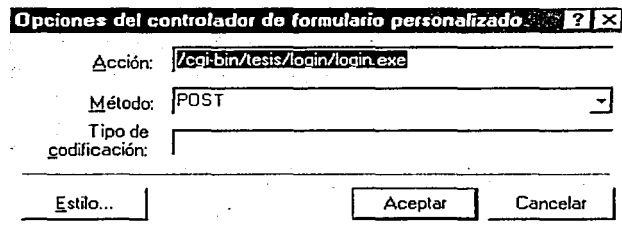


Figura 4.4.11 Propiedades acción y método del formulario.

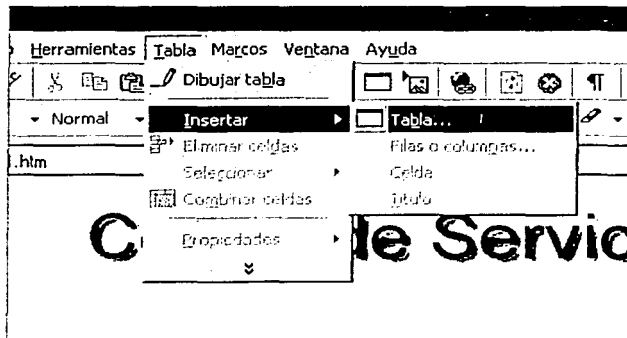


Figura 4.4.12 Insertar tabla.

TESIS CON
 FALLA DE ORIGEN

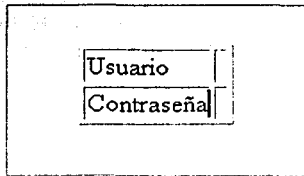


Figura 4.4.13 Celdas de la tabla.

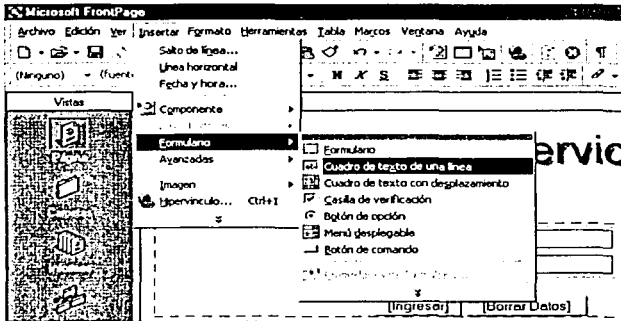


Figura 4.4.14 Insertar cuadro de texto.

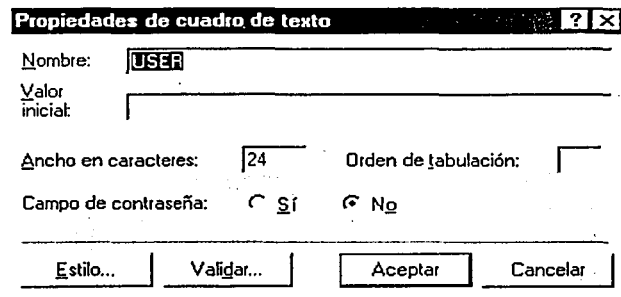


Figura 4.4.15 Propiedades del cuadro de texto usuario.

Propiedades de cuadro de texto [?] [X]

Nombre: PWD

Valor inicial:

Ancho en caracteres: 24 Orden de tabulación:

Campo de contraseña: Sí No

[Estilo...] [Validar...] [Aceptar] [Cancelar]

Figura 4.4.16 Propiedades del cuadro de texto contraseña.

Propiedades de botón de comando [?] [X]

Nombre: name

Valor/Etiqueta: [Ingresar]

Tipo de botón: Normal Enviar Restablecer

Orden de tabulación:

[Estilo...] [Aceptar] [Cancelar]

Figura 4.4.17 Propiedades del botón de comando ingresar.

Propiedades de botón de comando [?] [X]

Nombre: name

Valor/Etiqueta: [Borrar Datos]

Tipo de botón: Normal Enviar Restablecer

Orden de tabulación:

[Estilo...] [Aceptar] [Cancelar]

Figura 4.4.18 Propiedades del botón de comando borrar datos.

A continuación se muestran las interfaces graficas del Sistema de Administración de Eventos , ya terminadas.

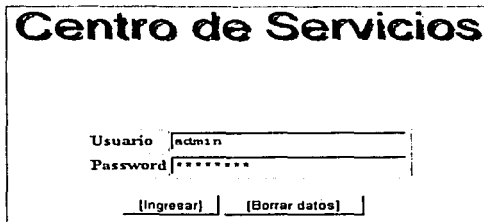
Inicio

La figura 4.4.19 muestra la interface de inicio del sistema, en la cual se debe proporcionar el nombre del usuario y su contraseña respectiva.

La segunda interface del sistema contiene un menú con las siguientes opciones:

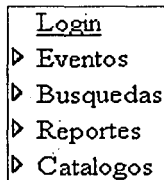
- Eventos
- Búsquedas
- Reportes
- Catálogos

Véase figura 4.4.20.



The screenshot shows a login window titled "Centro de Servicios". It contains two input fields: "Usuario" with the text "admin" and "Password" with asterisks. Below the fields are two buttons: "(Ingresar)" and "(Borrar datos)".

Figura 4.4.19 Interface de inicio.



The screenshot shows a menu window titled "Login". It contains four options, each preceded by a right-pointing triangle symbol: "Eventos", "Busquedas", "Reportes", and "Catalogos".

Figura 4.4.20 Menú.

Eventos

Si se selecciona la opción eventos del menú principal, mostrará dos submenús: nuevo y listado. Al elegir la opción deseada se mostrará la interface gráfica respectiva.

- Nuevo: Agregar un nuevo evento. (Figura 4.4.21).

The screenshot shows a web form titled "EVENTO". It contains the following fields and values:

- Semestre:** 0100
- Nombre del Evento:** (empty)
- Nivel:** FECHAS CANCELADAS
- Responsable:** (empty)
- Teléfono:** (empty)
- Email:** (empty)
- Número de Fax:** (empty)
- Número de Cuenta:** (empty)
- Dirección/División:** BASICO

Figura 4.4.21 Nuevo evento.

- Listado: Lista los eventos existentes. (Figura 4.4.22).

The screenshot shows a web form titled "LISTADO DE EVENTOS". It contains the following fields and values:

- Semestre:** 0100
- Dirección/División:** (empty)
- Alcance:** (empty)

Figura 4.4.22 Listado de eventos.

Búsquedas

Si se selecciona la opción búsquedas del menú principal, mostrará tres submenús: general, disponibles y evento. Al elegir la opción deseada se mostrará la interface gráfica respectiva.

- General: Busca eventos por sala, fecha y nombre. (Figura 4.4.23).

Figura 4.4.23 Búsqueda general.

- Disponibles: Busca salas disponibles por fecha y hora. (Figura 4.4.24).

Figura 4.4.24 Búsqueda de salas disponibles.

- Evento: Busca eventos por folio. (Figura 4.4.25).

Figura 4.4.25 Búsqueda por evento.

Reportes

Si se selecciona la opción reportes del menú principal, mostrará seis submenús: general, alimentos, por lugar, número de subevento, uso de salas y uso de equipo. Al elegir la opción deseada se mostrará la interface gráfica respectiva.

- General: Genera reporte de eventos por fecha. (Figura 4.4.26).

Figura 4.4.26 Reporte general.

- Alimentos: Genera reporte de alimentos por fecha. (Figura 4.4.27).

Figura 4.4.27 Reporte de alimentos.

- Por lugar: Genera reporte de eventos por sala y fecha. (Figura 4.4.28).

Figura 4.4.28 Reporte por lugar.

- Número de subevento: Genera reporte de eventos y subeventos, por semestre y folio. (Figura 4.4.29).

Reporte Eventos y Subeventos

Semestre: [0108 ▾]
Del folio: _____
Al: _____

*Al dejar en blanco el campo el reporte se realizará sobre todos los eventos del semestre seleccionado

Figura 4.4.29 Reporte por eventos y subeventos.

- Uso de salas: Genera reporte de uso de salas por semestre, sala y fecha. (Figura 4.4.30).

Reporte de uso de Salas

Semestre: [0108 ▾]
Salas: ALBERCA - ALBERCA
 AUDIOVISUAL LANGUAGE LAB - 8302
 AULA MAGNA - 6104
 AULA MAGNA - 6105
 AULA MAGNA - 6106
 AULA MAGNA - 6207
 AULA MAGNA - 8402
 AULAS VI - EXPLANADA
 AUDITORIO - AUDITORIO
 BIBLIOTECA - JARDIN LATERAL

Fecha Inicial: [dd/mm/yyyy] _____
Fecha Final: [dd/mm/yyyy] _____

*Al dejar en blanco el campo el reporte se realiza sobre todos los eventos del semestre seleccionado

Figura 4.4.30 Reporte de uso de salas.

- Uso de equipo: Genera reporte de uso de equipo por semestre, equipo y fecha. (Figura 4.4.31).

Reporte de uso de Equipo

<input type="checkbox"/>	AUDIOVISUAL 1 Equipo Fijo (TV27-Vhs-Retro)
<input type="checkbox"/>	AUDIOVISUAL 2 Equipo Fijo (TV31-Vhs-Retro)
<input type="checkbox"/>	AUDIOVISUAL 3 Equipo Fijo (TV31-Vhs-Retro)
<input type="checkbox"/>	AUDIOVISUAL 4 Equipo Fijo (TV31-Vhs-Retro)
<input type="checkbox"/>	AUDIOVISUAL 5 Equipo Fijo (TV20-Vhs-Retro)
<input type="checkbox"/>	AUDIOVISUAL MULTIMEDIA 1 Equipo Fijo (TV31-Vhs-Retro-Modular-CompuMAC)
<input type="checkbox"/>	AUDIOVISUAL MULTIMEDIA 2 Equipo Fijo (TV31-Vhs-Retro-Modular-CompuFC)
<input type="checkbox"/>	AUDITORIO Equipo Fijo (Retro-Canon-TV14-Vhs-Beta-Micro-Alamb)
<input type="checkbox"/>	AULAS II Cámaras pr.stamo
<input type="checkbox"/>	AULAS II Cámaras VB pr.stamo

Fecha Inicial:

Fecha Final:

*Atender en tiempo el campo el reporte se realizar 4 sobre todos los eventos del semestre seleccionado

Figura 4.4.31 Reporte de uso de equipo.

Catálogos

Si se selecciona la opción catálogos del menú principal, mostrará ocho submenús: salas, equipo, alimentos, accesorios, montajes, usuarios, divisiones y niveles; cada uno a su vez cuenta con otros dos submenús: listado y agregar. Al elegir la opción deseada se mostrará la interface gráfica respectiva.

- Salas – Listado: Lista las salas existentes. (Figura 4.4.32).

SALAS		
#	Sala	Capacidad
1	ALBERCA - ALBERCA	1
2	AUDIOVISUAL LANGUAGE LAB - 8302	42
3	AULA MAGNA - 6104	130
4	AULA MAGNA - 6105	133
5	AULA MAGNA - 6106	123
6	AULA MAGNA - 6207	125
7	AULA MAGNA - 8402	90

Figura 4.4.32 Listado de salas.

- Salas – Agregar: Agrega una nueva sala. (Figura 4.4.33).

SALA

Sala:

Capacidad:

Figura 4.4.33 Nueva sala.

- Equipo – Listado: Lista los equipos existentes. (Figura 4.4.34).

EQUIPO		
#	Equipo	CANTIDAD
1	AUDIOVISUAL 1 Equipo Fijo (TV27-Vhs-Petrol)	1
2	AUDIOVISUAL 2 Equipo Fijo (TV31-Vhs-Petrol)	1
3	AUDIOVISUAL 3 Equipo Fijo (TV31-Vhs-Petrol)	1
4	AUDIOVISUAL 4 Equipo Fijo (TV31-Vhs-Petrol)	1
5	AUDIOVISUAL 5 Equipo Fijo (TV20-Vhs-Petrol)	1
6	AUDIOVISUAL MULTIMEDIA 1 Equipo Fijo (TV31-Vhs-Petrol-Modular-CompuMAC)	1
7	AUDIOVISUAL MULTIMEDIA 2 Equipo Fijo (TV31-Vhs-Petrol-Modular-CompuPC)	1

Figura 4.4.34 Listado de equipos.

- Equipo – Agregar: Agrega un nuevo equipo. (Figura 4.4.35).

EQUIPO		
#	Equipo	Cantidad

Figura 4.4.35 Nuevo equipo.

- Alimentos – Listado: Lista los alimentos existentes. (Figura 4.4.36).

ALIMENTOS		
#	Alimento	Costo
1	Agua bonafont (chica)	5
2	Bebida especial (cocktail)	16
3	Bocadillos mexicanos con agua de sabor	40
4	Box lunch con sandwich	24
5	Box lunch con torta	30
6	Break (45 minutos)	18
7	Break día completo (8 horas)	51

Figura 4.4.36 Listado de alimentos.

- Alimentos – Agregar: Agrega un nuevo alimento. (Figura 4.4.37).

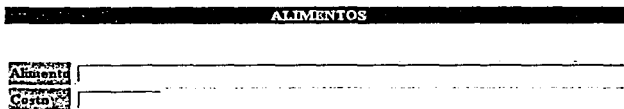


Figura 4.4.37 Nuevo alimento.

- Accesorios – Listado: Lista los accesorios existentes. (Figura 4.4.38).



Figura 4.4.38 Listado de accesorios.

- Accesorios – Agregar: Agrega un nuevo accesorio. (Figura 4.4.39).



Figura 4.4.39 Nuevo accesorio.

- Montajes – Listado: Lista los montajes existentes. (Figura 4.4.40).

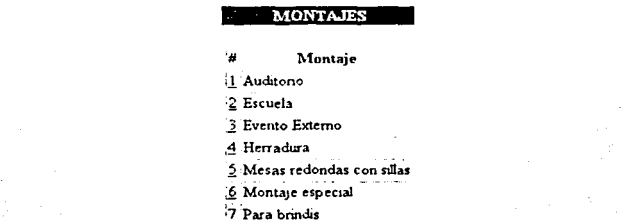


Figura 4.4.40 Listado de montajes.

- Montajes – Agregar: Agrega un nuevo montaje. (Figura 4.4.41).

MONTAJES



Figura 4.4.41 Nuevo montaje.

- Usuarios – Listado: Lista los usuarios existentes. (Figura 4.4.42).

USUARIOS

#	Usuario	Tipo de Usuario
1	admin	
2	cservici1	
3	ROGELIO	
4	cservici2	
5	cservici3	
6	cservici4	
7	cservici5	

Figura 4.4.42 Listado de usuarios.

- Usuarios – Agregar: Agrega un nuevo usuario. (Figura 4.4.43).

USUARIOS



Solo Lectura
 Lectura y Escritura
 Administrador

Figura 4.4.43 Nuevo usuario.

- Divisiones – Listado: Lista las divisiones existentes. (Figura 4.4.44).

DIRECCION-DIVISION

#	Direccion/División
1	BASICO
2	DACS-COMPOTAMIENTO ORGANIZACIONAL
3	DACS-COMUNICACION
4	DACS-CONTABILIDAD
5	DACS-DERECHO
6	DACS-DIRECCION
7	DACS-ECONOMIA Y FINANZAS

Figura 4.4.44 Listado de divisiones.

- Divisiones – Agregar: Agrega una nueva división. (Figura 4.4.45).



Figura 4.4.45 Nueva división.

- Niveles – Listado: Lista los niveles existentes. (Figura 4.4.46).

CATEGORIAS DE EVENTOS

#	Nivel
1	FECHAS CANCELADAS
2	FOLIO CANCELADO
3	NIVEL 1 BASICO
4	NIVEL 2 INTERMEDIO
5	NIVEL 3 MAGNO
6	NO VALIDO
7	NO VALIDO

Figura 4.4.46 Listado de niveles.

- Niveles – Agregar: Agrega un nuevo nivel. (Figura 4.4.47).



Figura 4.4.47 Nuevo nivel.

4.5 Diseño y desarrollo de los CGIs

Para el diseño e implementación de los CGIs del sistema de administración de eventos se utilizó Delphi 5.0 Enterprise véase capítulo 2 inciso 5.

4.5.1 Procedimiento para la creación de los CGIs

1. Seleccionar de la barra de herramientas File / New Application, como se muestra en la figura 4.5.1.

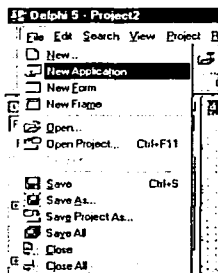


Figura 4.5.1 Nueva aplicación.

2. Seleccionar la ventana de formulario y comenzar a colocar todos los componentes visuales necesarios (TGeneralHttpEngine, THttpDBGrid, THttpMemoFilter, THttpFileFilter, THttpInput, THttpCookie, etc), ver figura 4.5.2.

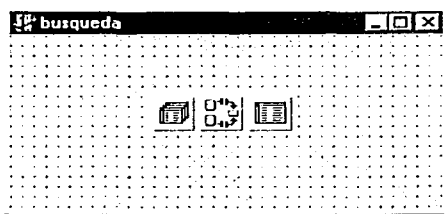


Figura 4.5.2 Colocación de componentes visuales sobre el formulario.

3. Configurar las propiedades de cada uno de los componentes mediante el inspector de objetos.
4. Escribir el código necesario para el funcionamiento del CGI mediante el editor de código fuente.

A continuación se muestra parte del código fuente de los CGIs con los que cuenta el Sistema de Administración de Eventos.

CGI Login

Función: Validar el acceso al sistema.

```

procedure TDataModule2.GeneralHttpEngine1ExecRequest(Sender: TObject);
var
  PathInfo:String;
begin
  PathInfo:=Trim(UpperCase(GeneralHttpEngine1.ServerVar('PATH_INFO')));
  if (PathInfo='/DEFAULT') or (PathInfo='') or (PathInfo='/') then
    Begin
      Do_Login;
    End
  else
    if (PathInfo='/FRAME_MENU') then
      Begin
        Do_Frame_Menu;
      End;
    end;
end;

procedure TDataModule2.GeneralHttpEngine1LastExecRequest(Sender: TObject);
begin
  session.close;
end;

```

CGI Eventos

Función: Administrar eventos.

```

procedure TFrm_Main_Eventos.GeneralHttpEngine1FirstExecRequest(
  Sender: TObject);
begin
  ShortDateFormat:='dd/mm/yyyy';
  ShortTimeFormat:='hh:nn';
  Nombre_CGI:=ScriptName;
end;

procedure TFrm_Main_Eventos.GeneralHttpEngine1LastExecRequest(
  Sender: TObject);
begin
  Session.Close;
end;

procedure TFrm_Main_Eventos.GeneralHttpEngine1ExecRequest(Sender: TObject);
var
  PathInfo:String;
  USER:String;
begin
  USER:=HttpCookie1.ReadCookie("");
  frm_Data.Query_Usuarios.Params[0].AsString:=USER;
  frm_Data.Query_Usuarios.Open;
  if frm_Data.Query_Usuarios.Eof and frm_Data.Query_Usuarios.Bof then
    Begin
      frm_Data.Query_Usuarios.Close;
      exit;
    End;
end;

```

```

EditMode:=(frm_Data.Query_Usuarios.FieldByName('PERMISO').AsString='W');
EditAdmin:=(frm_Data.Query_Usuarios.FieldByName('ADMINISTRADOR').AsString='S');
frm_Data.Query_Usuarios.Close;

PathInfo:=Trim(UpperCase(GeneralHttpEngine1.ServerVar('PATH_INFO')));
if (PathInfo='/DEFAULT') or (PathInfo='') or (PathInfo='/') then
Begin
    Do_Show_List_Eventos;
End
else
if PathInfo='/EVENTO/SELEC_LIST' then
Begin
    Do_Show_List_Eventos2;
End
else
if PathInfo='/EVENTO' then
Begin
    Do_ShowEvento(URLDecode(GeneralHttpEngine1.FormVar('cve_evento','ERROR')));
End
else
if PathInfo='/EVENTO/EDIT' then
Begin
    Do_Edit_Evento(URLDecode(GeneralHttpEngine1.FormVar('cve_evento','ERROR')));
End
else
if PathInfo='/EVENTO/UPDATE' then
Begin
    Do_Evento_Update;
End
else
if PathInfo='/EVENTO/DELETE' then
Begin
    Do_Evento_Delete(URLDecode(GeneralHttpEngine1.FormVar('cve_evento','ERROR')));
End
else
if PathInfo='/EVENTO/ADD' then
Begin
    Do_Evento_Add;
End
else
if PathInfo='/EVENTO/INSERT' then
Begin
    Do_Evento_Insert;
End
else
if PathInfo='/FECHA/SHOW' then
Begin
    Do_Show_Fecha(GeneralHttpEngine1.FormVar('cve_fecha','ERROR'));
End
else
if PathInfo='/FECHA/EDIT' then
Begin
    Do_Fecha_Edit(GeneralHttpEngine1.FormVar('cve_fecha','ERROR'));
End
else
if PathInfo='/FECHA/UPDATE' then

```

```

Begin
  Do_Fecha_Update;
End
else
if PathInfo='/FECHA/DELETE' then
Begin
  Do_Fecha_Delete;
End
else
if PathInfo='/FECHA/ADD' then
Begin
  Do_Fecha_Add;
End
else
if PathInfo='/FECHA/INSERT' then
Begin
  Do_Fecha_Insert;
End
else
if PathInfo='/EQUIPO/EDIT' then
Begin
  Do_Equipo_Edit;
End
else
if PathInfo='/EQUIPO/UPDATE' then
Begin
  Do_Equipo_Update;
End
else
if PathInfo='/EQUIPO/DELETE' then
Begin
  Do_Equipo_Delete;
End
else
if PathInfo='/EQUIPO/ADD' then
Begin
  Do_Equipo_Add;
End
else
if PathInfo='/EQUIPO/INSERT' then
Begin
  Do_Equipo_Insert;
End
else
if PathInfo='/ALIMENTOS/EDIT' then
Begin
  Do_Alimentos_Edit;
End
else
if PathInfo='/ALIMENTOS/UPDATE' then
Begin
  Do_Alimentos_Update;
End
else
if PathInfo='/ALIMENTOS/DELETE' then
Begin

```

```
    Do_Alimentos_Delete;
End
else
if PathInfo='/ALIMENTOS/ADD' then
Begin
    Do_Alimentos_Add;
End
else
if PathInfo='/ALIMENTOS/INSERT' then
Begin
    Do_Alimentos_Insert;
End
else
if PathInfo='/ACCESORIOS/EDIT' then
Begin
    Do_Accesorios_Edit;
End
else
if PathInfo='/ACCESORIOS/UPDATE' then
Begin
    Do_Accesorios_Update;
End
else
if PathInfo='/ACCESORIOS/DELETE' then
Begin
    Do_Accesorios_Delete;
End
else
if PathInfo='/ACCESORIOS/ADD' then
Begin
    Do_Accesorios_Add;
End
else
if PathInfo='/ACCESORIOS/INSERT' then
Begin
    Do_Accesorios_Insert;
End
else
//if PathInfo='/EDIT/FECHAS' then
Begin
    Do_ShowVariables;
End;
end;
```

CGI Búsqueda

Función: Consultar eventos.

```
procedure TDataModule1.GeneralHttpEngine1FirstExecRequest(Sender: TObject);
begin
    ShortDateFormat:='dd/mm/yyyy';
    ShortTimeFormat:='hh:nn';
    Nombre_CGI:=ScriptName;
end;
```

```

procedure TDataModule1.GeneralHttpEngine1LastExecRequest(Sender: TObject);
begin
    session.close;
end;

```

```

procedure TDataModule1.GeneralHttpEngine1ExecRequest(Sender: TObject);
var
    PathInfo:String;
    USER:String;
begin
    USER:=HttpCookie1.ReadCookie("");
    Query_Usuarios.Params[0].AsString:=USER;
    Query_Usuarios.Open;
    if Query_Usuarios.Eof and Query_Usuarios.Bof then
        Begin
            Query_Usuarios.Close;
            exit;
        End;
    EditMode:=(Query_Usuarios.FieldByName('PERMISO').AsString='W');
    AdminMode:=(Query_Usuarios.FieldByName('ADMINISTRADOR').AsString='S');
    Query_Usuarios.Close;

    PathInfo:=Trim(UpperCase(GeneralHttpEngine1.ServerVar('PATH_INFO')));
    if (PathInfo='/DISPONIBLES/SHOW') then
        Do_Disponibles_Show
    else
    if (PathInfo='/GENERAL/SHOW') then
        Do_General_Show
    else
    if (PathInfo='/EVENTO/SHOW') then
        Do_Evento_Show
    else
    if (PathInfo='/EVENTO/DO') then
        Do_Evento_Do
    else
        Do_ShowVariables;
end;

```

CGI Reportes

Función: Generar reportes.

```

procedure TForm1.GeneralHttpEngine1FirstExecRequest(Sender: TObject);
begin
    TimeSeparator:=':';
    DateSeparator:= '/';
    ShortDateFormat:='dd/mm/yyyy';
    ShortTimeFormat:='hh:nn';
    Nombre_CGI:=ScriptName;
end;

```

```

procedure TForm1.GeneralHttpEngine1LastExecRequest(Sender: TObject);

```



```

begin
    session.close;
end;

procedure TForm1.GeneralHttpEngine1ExecRequest(Sender: TObject);
var
    PathInfo:String;
    USER:String;
begin
    USER:=HttpCookie1.ReadCookie("");
    frm_Data.Query_Usuarios.Params[0].AsString:=USER;
    frm_Data.Query_Usuarios.Open;
    if frm_Data.Query_Usuarios.Eof and frm_Data.Query_Usuarios.Bof then
        Begin
            frm_Data.Query_Usuarios.Close;
            exit;
        End;
    EditMode:=(frm_Data.Query_Usuarios.FieldByName('PERMISO').AsString='W');
    AdminMode:=(frm_Data.Query_Usuarios.FieldByName('ADMINISTRADOR').AsString='S');
    frm_Data.Query_Usuarios.Close;

    PathInfo:=Trim(UpperCase(GeneralHttpEngine1.ServerVar('PATH_INFO')));
    if (PathInfo='/SALAS/SHOW') then
        Do_Salas_Show
    else
        if (PathInfo='/SALAS/DO') then
            Do_Salas_Do
        else
            if (PathInfo='/GENERAL/SHOW') then
                Do_General_Show
            else
                if (PathInfo='/GENERAL/DO') then
                    Do_General_do
                else
                    if (PathInfo='/ALIMENTOS/SHOW') then
                        Do_Alimentos_Show
                    else
                        if (PathInfo='/ALIMENTOS/DO') then
                            Do_Alimentos_Do
                        else
                            if (PathInfo='/GENERAL_SUBEVENTOS/SHOW') then
                                Do_GeneralSubEventos_Show
                            else
                                if (PathInfo='/GENERAL_SUBEVENTOS/DO') then
                                    Do_GeneralSubEventos_Do
                                else
                                    if (PathInfo='/USO_SALAS/SHOW') then
                                        Do_Usos_Salas_Show
                                    else
                                        if (PathInfo='/USO_SALAS/DO') then
                                            Do_Usos_Salas_Do
                                        else
                                            if (PathInfo='/USO_EQUIPO/SHOW') then
                                                Do_Usos_Equipo_Show
                                            else
                                                if (PathInfo='/USO_EQUIPO/DO') then

```

```

Do_Usa_Equipo_Do
else
Do_ShowVariables;
end;

```

CGI Catálogos

Función: Administrar catálogos.

```

procedure TForm1.GeneralHttpEngine1FirstExecRequest(Sender: TObject);
begin
    ShortDateFormat:='dd/mm/yyyy';
    ShortTimeFormat:='hh:nn';
    Nombre_CGI:=ScriptName;
end;

```

```

procedure TForm1.GeneralHttpEngine1LastExecRequest(Sender: TObject);
begin
    Session.Close;
end;

```

```

procedure TForm1.GeneralHttpEngine1ExecRequest(Sender: TObject);
var
    PathInfo:String;
    USER:String;
begin
    USER:=HttpCookie1.ReadCookie("");
    frm_Data.Query_Usuarios.Params[0].AsString:=USER;
    frm_Data.Query_Usuarios.Open;
    if frm_Data.Query_Usuarios.Eof and frm_Data.Query_Usuarios.Bof then
    Begin
        frm_Data.Query_Usuarios.Close;
        HttpFileFilter_ERROR_ADMIN.Put ;
        exit;
    End;
    EditMode:=(frm_Data.Query_Usuarios.FieldByName('PERMISO').AsString='W');
    AdminMode:=(frm_Data.Query_Usuarios.FieldByName('ADMINISTRADOR').AsString='S');
    frm_Data.Query_Usuarios.Close;
    if Not AdminMode then
    Begin
        HttpFileFilter_ERROR_ADMIN.Put;
        Exit;
    End;
    PathInfo:=Trim(UpperCase(GeneralHttpEngine1.ServerVar('PATH_INFO')));
    if (PathInfo='/DEFAULT') or (PathInfo='') or (PathInfo='/') then
    Begin
        //Do_Show_List_Catalogos;
    End
    else
    if PathInfo='/SALAS/SHOW' then
        Do_Salas_Show
    else

```

```
if PathInfo='/SALAS/EDIT' then
  Do_Salas_Edit
else
if PathInfo='/SALAS/UPDATE' then
  Do_Salas_Update
else
if PathInfo='/SALAS/ADD' then
  Do_Salas_ADD
else
if PathInfo='/SALAS/INSERT' then
  Do_Salas_Insert
else
if PathInfo='/SALAS/DELETE' then
  Do_Salas_Delete
else
if PathInfo='/EQUIPO/SHOW' then
  Do_Equipo_Show
else
if PathInfo='/EQUIPO/EDIT' then
  Do_Equipo_Edit
else
if PathInfo='/EQUIPO/UPDATE' then
  Do_Equipo_Update
else
if PathInfo='/EQUIPO/ADD' then
  Do_Equipo_ADD
else
if PathInfo='/EQUIPO/INSERT' then
  Do_Equipo_Insert
else
if PathInfo='/EQUIPO/DELETE' then
  Do_Equipo_Delete
else
if PathInfo='/ALIMENTOS/SHOW' then
  Do_Alimentos_Show
else
if PathInfo='/ALIMENTOS/EDIT' then
  Do_Alimentos_Edit
else
if PathInfo='/ALIMENTOS/UPDATE' then
  Do_Alimentos_Update
else
if PathInfo='/ALIMENTOS/ADD' then
  Do_Alimentos_Add
else
if PathInfo='/ALIMENTOS/INSERT' then
  Do_Alimentos_Insert
else
if PathInfo='/ALIMENTOS/DELETE' then
  Do_Alimentos_Delete
else
if PathInfo='/ACCESORIOS/SHOW' then
  Do_Accesorios_Show
else
if PathInfo='/ACCESORIOS/EDIT' then
  Do_Accesorios_Edit
```

```

else
if PathInfo='/ACCESORIOS/UPDATE' then
    Do_Accesorios_Update
else
if PathInfo='/ACCESORIOS/ADD' then
    Do_Accesorios_Add
else
if PathInfo='/ACCESORIOS/INSERT' then
    Do_Accesorios_Insert
else
if PathInfo='/ACCESORIOS/DELETE' then
    Do_Accesorios_Delete
else
if PathInfo='/USUARIOS/SHOW' then
    Do_Usuarios_Show
else
if PathInfo='/USUARIOS/EDIT' then
    Do_Usuarios_Edit
else
if PathInfo='/USUARIOS/UPDATE' then
    Do_Usuarios_Update
else
if PathInfo='/USUARIOS/ADD' then
    Do_Usuarios_Add
else
if PathInfo='/USUARIOS/INSERT' then
    Do_Usuarios_Insert
else
if PathInfo='/USUARIOS/DELETE' then
    Do_Usuarios_Delete
else
if PathInfo='/MONTAJES/SHOW' then
    Do_Montajes_Show
else
if PathInfo='/MONTAJES/EDIT' then
    Do_Montajes_Edit
else
if PathInfo='/MONTAJES/UPDATE' then
    Do_Montajes_Update
else
if PathInfo='/MONTAJES/ADD' then
    Do_Montajes_Add
else
if PathInfo='/MONTAJES/INSERT' then
    Do_Montajes_Insert
else
if PathInfo='/MONTAJES/DELETE' then
    Do_Montajes_Delete
else
if PathInfo='/DIRECCION_DIVISION/SHOW' then
    Do_Direccion_Division_Show
else
if PathInfo='/DIRECCION_DIVISION/EDIT' then
    Do_Direccion_Division_Edit
else
if PathInfo='/DIRECCION_DIVISION/UPDATE' then

```

```

        Do_Direccion_Division_Update
    else
    if PathInfo='/DIRECCION_DIVISION/ADD' then
        Do_Direccion_Division_Add
    else
    if PathInfo='/DIRECCION_DIVISION/INSERT' then
        Do_Direccion_Division_Insert
    else
    if PathInfo='/DIRECCION_DIVISION/DELETE' then
        Do_Direccion_Division_Delete
    else
    if PathInfo='/NIVEL/SHOW' then
        Do_Nivel_Show
    else
    if PathInfo='/NIVEL/EDIT' then
        Do_Nivel_Edit
    else
    if PathInfo='/NIVEL/UPDATE' then
        Do_Nivel_Update
    else
    if PathInfo='/NIVEL/ADD' then
        Do_Nivel_Add
    else
    if PathInfo='/NIVEL/INSERT' then
        Do_Nivel_Insert
    else
    if PathInfo='/NIVEL/DELETE' then
        Do_Nivel_Delete
    else
        Do_ShowVariables;
end;
```

4.6 PRUEBAS, DEPURACIÓN Y LIBERACIÓN DEL SISTEMA

Una de las etapas fundamentales en el desarrollo de todo sistema, es el punto que ahora nos ocupa: las pruebas, depuración y liberación de nuestro sistema. Consideramos de gran importancia este punto debido a que es la etapa posterior a la conclusión del proyecto y en la cual se van a detectar los errores o anomalías en que pudimos haber incurrido a lo largo de todo nuestro trabajo.

4.6.1 Pruebas

Entendemos por prueba a la acción que se realiza cuando queremos demostrar el funcionamiento adecuado de algo que hemos creado o elaborado.

Las pruebas de los programas son parte del proceso de confirmación que suele realizarse durante el desarrollo de la aplicación y también, en forma distinta, cuando éste ha terminado. Las pruebas consisten en ejercitar el programa utilizando datos similares a los datos reales que habrán de ser procesados por el programa, observar los resultados y deducir la existencia de errores o insuficiencias del programa a partir de los resultados obtenidos.

En ocasiones, se piensa que las pruebas y la depuración de los programas son una misma cosa. Aunque están muy relacionadas, en realidad son procesos distintos. La prueba es el proceso de establecer la existencia de errores en el programa. Depuración es el proceso de localizar dónde se produjeron esos errores y corregir el código en caso de ser necesario.

Es muy importante comprender que las pruebas nunca demuestran que un programa este libre de errores. Siempre es posible que existan errores aún después de la prueba más completa. La prueba de programas sólo puede demostrar la presencia de errores, no su ausencia, por lo tanto, se considera prueba acertada aquella que establece la presencia de uno o más errores en el sistema objeto de prueba.

Las pruebas de sistemas son sumamente importantes dentro de su ciclo de vida, debido a que de estas depende que se puedan corregir los posibles errores antes de la implantación del sistema y que se evite incurrir en costos de mantenimiento, cuando lleguen a surgir errores no descubiertos, en la vida útil del sistema.

4.6.1.1 El proceso de prueba

El proceso de prueba, al igual que el de programación, debe avanzar en etapas, siendo cada una de ellas la continuación lógica de la etapa anterior.

En el proceso de prueba se pueden identificar cinco etapas:

- Prueba de funciones: La prueba de funciones o de unidad es el nivel básico en donde se prueban las funciones que componen un módulo para garantizar que operan de manera correcta.
- Prueba de módulos: Un módulo se compone de varias funciones que pueden cooperar entre sí. Después de haber probado cada función individual, es necesario probar la cooperación de estas funciones cuando componen un módulo. Debe ser posible probar un módulo como una entidad aislada, sin la presencia de otros módulos del sistema.
- Prueba de subsistemas: Esta prueba es el siguiente paso del proceso, en el cual los módulos se agrupan para formar subsistemas. Puesto que los módulos cooperan y se comunican, la prueba de subsistemas se debe centrar en las interfaces de éstos.
- Pruebas del sistema: La prueba del sistema (a veces llamada prueba de integración) se lleva a cabo cuando se integran los subsistemas para conformar el sistema completo. En esta etapa, el proceso de prueba tiene que ver con el descubrimiento de errores en el diseño y la codificación.
- Prueba de aceptación: Hasta esta etapa, todas las pruebas se realizan mediante el empleo de datos generados por nosotros mismos. La prueba de aceptación del sistema se efectúa con datos reales (información con la que el sistema deberá operar). El proceso en la prueba de aceptación a menudo descubre errores en la definición de requisitos del sistema.

4.6.1.2 Tipos de prueba

Existen distintas estrategias de prueba, las dos más comunes se conocen como prueba ascendente y prueba descendente. El enfoque ascendente empieza por probar módulos individuales; esto a menudo se conoce como prueba de unidades, prueba de módulos o prueba de programas.

Luego, los módulos individuales se combinan para formar unidades cada vez más grandes que se probaran en masa; esto se conoce como prueba del sistema, y suele estar seguido de las pruebas de aceptación, donde se permite al usuario usar sus

propios casos de prueba para verificar que el sistema esté trabajando de manera correcta.

El enfoque de prueba descendente empieza con un esqueleto del sistema, es decir, la estrategia de prueba supone que se han desarrollado módulos de alto nivel, pero que los de bajo nivel existen sólo como módulos vacíos (módulos que no procesan nada, sino que simplemente terminan luego de ser llamados). Dado que muchas de las funciones detalladas del sistema no se han implantado, las pruebas iniciales están muy limitadas; el propósito es simplemente comenzar a ejercitar las interfaces entre los subsistemas principales.

4.6.1.3 Pruebas al sistema para la administración de eventos en una institución educativa

Fue necesario aplicar diversas pruebas al sistema para poder asegurar que su funcionamiento cumpliera con los objetivos planteados. Dichas pruebas fueron realizadas con la finalidad de detectar posibles errores o anomalías en los procesos que lleva a cabo el sistema, además de tener como objetivo primordial el liberar un sistema de alta calidad que garantice el funcionamiento óptimo de todos los procesos que en él se realicen, además de brindar un alto grado de confiabilidad y un porcentaje nulo de errores.

Las pruebas realizadas al sistema para la administración de eventos en una institución educativa se llevaron a cabo en forma ascendente, por lo que se procedió en primera instancia a verificar el funcionamiento correcto de cada uno de los módulos (prueba de unidad), para tratar de localizar posibles errores; posteriormente se probó la integración de cada módulo en el subsistema (prueba de integración). Y por último se probaron todos los componentes que integran el sistema (prueba de sistema), para verificar la interacción de los subsistemas.

En la figura 4.6.1 se observan algunas de las principales operaciones de prueba que se aplican durante el desarrollo de un sistema.

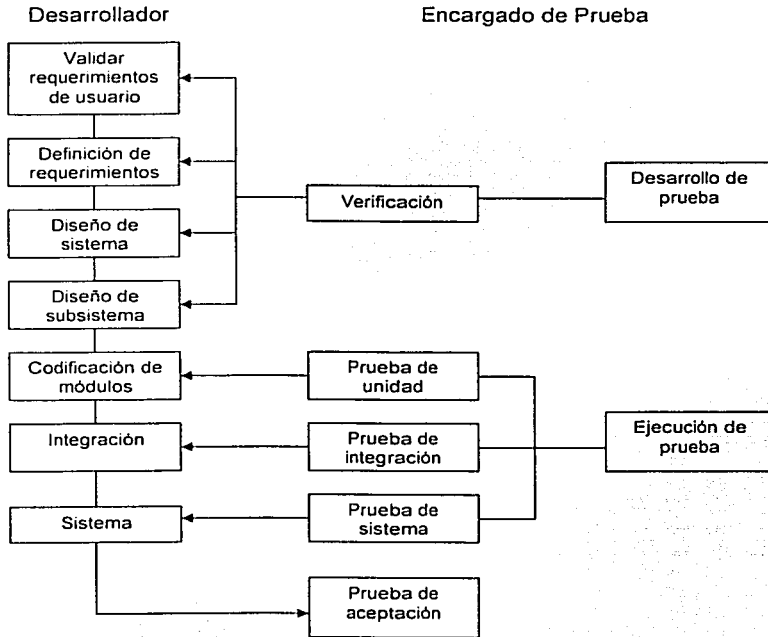


Figura 4.6.1 Operaciones de prueba durante el desarrollo de un sistema.

4.6.2 Depuración

Después de haber desarrollado el sistema y haber efectuado las pruebas pertinentes en forma exhaustiva, viene la fase de depuración. Entendemos por depuración a la acción de eliminar o quitar imperfecciones.

El proceso de pruebas fue de gran utilidad por que nos permitió conocer las condiciones tanto favorables como desfavorables que pudieron haber ocurrido y que no se tenían contempladas dentro del sistema. Gracias a las pruebas se pudo poner en marcha la fase de depuración, la cual se encargó de canalizar las condiciones de nuestro sistema, con la finalidad de que todos y cada uno los elementos de la aplicación realizaran su trabajo de manera adecuada.

Consideramos que el sistema, en base a las pruebas que se realizaron, se depuró lo máximo posible y quedo listo para la fase de liberación.

4.6.3 Liberación

En este punto se tienen ya todas las condiciones para poder poner en marcha el sistema y que cumpla con la finalidad para la cual fue creado.

En esta fase una vez que se ha verificado que el sistema está libre de errores, se procede a la instalación del mismo dentro del ambiente real en donde operará desde ese momento en adelante. Junto con la instalación se realizan otras actividades paralelas, de las cuales las más importantes son: la creación de archivos para el sistema, entrega de manuales de usuario y capacitación del personal que operará el sistema.

4.7 FACTIBILIDAD TÉCNICA Y OPERATIVA

Este punto es fundamental para la puesta en marcha del Sistema de Administración de Eventos dado que es aquí donde se describen los requerimientos técnicos y operativos tanto del hardware como del software. Debemos separar en dos puntos nuestros requerimientos:

- Requerimientos del servidor.

- Requerimientos del cliente.

4.7.1 Requerimientos del servidor

Una de las partes importantes para la puesta en marcha del sistema la constituye el servidor debido a que es éste el que va a dar alojamiento a todos los recursos con los que se trabajará. Es por esto que se hace necesario que la computadora que va a desempeñar la labor de servidor cumpla con requisitos específicos para realizar de manera eficiente todas las funciones que el sistema demanda. Estos requisitos son básicamente: requerimientos mínimos de hardware y software, requerimientos eléctricos y su ubicación física.

Los requerimientos mínimos de hardware para el servidor son los siguientes:

- Procesador Pentium III a 800 Mhz.
- 256 MB en RAM.
- Unidad de CD-ROM 44X.
- Unidad de respaldo de 24 GB SCSI interna.
- Disco duro de 20 GB.
- Tarjeta de red compatible.
- Bahías disponibles para crecimiento futuro.
- 2 puertos seriales y 1 puerto paralelo.
- Un ratón u otro dispositivo para apuntar.
- Monitor SVGA a color 15".

Los requerimientos mínimos de software para el servidor son los siguientes:

- Windows NT Server v4.x o superior
- InterBase 6.01
- Apache 1.3.20

Para los requerimientos eléctricos del servidor, se hace necesario un UPS con interface al servidor con capacidad de 1200 watts. La interface es para protegerlo de forma automática en caso de falla en el suministro eléctrico, además de que es posible apagar de forma segura el servidor, evitando así una posible pérdida de información y daño al equipo.

En lo que respecta a la ubicación física del servidor, éste debe ser colocado en una área exclusiva y libre de paso (de preferencia en un lugar cerrado y con aire acondicionado), con el fin de que pueda ser operado sin dificultad cuando sea necesario.

4.7.2 Requerimientos del cliente

Del lado del cliente los requerimientos son más flexibles, ya que sólo se preocupará por enviar y recibir información.

Los requerimientos mínimos de hardware para el cliente son los siguientes:

- Computadora Pentium (100 MHz).
- 32 MB en RAM.
- Un ratón u otro dispositivo para apuntar.
- Monitor SVGA a color.
- Tarjeta de red compatible o MODEM de 33 K.
- Impresora de matriz de puntos.

Los requerimientos mínimos de software para el cliente son los siguientes:

- Windows 95
- Netscape Navigator 4.7

4.7.3 Impresoras

Se deberá de contar con el acceso a una impresora ya sea en forma local o remota, para realizar las impresiones de los documentos generados por el sistema.

4.7.4 Puesta a punto

Para la instalación del servidor que alojará la base de datos del sistema es necesario definir la contraseña del administrador, configurar los protocolos y elegir el tipo de licencia para los accesos.

4.7.5 Nombre del servidor y rol asignado

El servidor debe de contar con un nombre único que lo identifique y evite problemas de comunicación hacia otras redes.

Un servidor Windows NT/2000 puede configurarse bajo tres esquemas diferentes de seguridad como PDC(Primary Domain Controller o Controlador Primario de Dominio), BDC (Backup Domain Controller o Controlador de Dominio de Respaldo) y como Server (Stand-Alone Server o Servidor Único). Este esquema de seguridad es conocido como Dominio. El nombre del servidor así como el rol asignado deberán registrarse y solicitarse en el departamento de informática.

4.7.6 Instalación de Interbase

La instalación de Interbase es muy sencilla ya se realiza a través de una interface gráfica. Dándonos a elegir la ubicación, el tipo de instalación y los componentes a instalar. Esta instalación toma poco tiempo aun eligiendo todas sus aplicaciones.

4.7.7 Administración del Servidor

Es necesario nombrar a un administrador que cumpla con ciertas tareas y responsabilidades para la administración y mantenimiento del servidor.

El administrador deberá asumir las siguientes responsabilidades:

- Administrar los usuarios.
- Administrar la seguridad del servidor (asignación de privilegios y claves de acceso).
- Administrar los servicios de archivos e impresión.
- Respalidar la información.
- Instalar y dar soporte del sistema.
- Asegurar el cumplimiento de las normatividades del sistema.

4.7.8 Mantenimiento

Un proceso importante para el continuo y adecuado funcionamiento del sistema es el mantenimiento que se le brinde. Entre los más comunes que se presentan en la vida útil de un sistema de información se encuentran los siguientes:

- **Mantenimiento Perfectivo.** El objetivo de este tipo de mantenimiento es mejorar el desempeño de las funciones existentes.
- **Mantenimiento Preventivo.** Este mantenimiento se lleva a cabo cuando se realizan cambios al sistema con el fin de mejorar algún proceso, previniendo los errores que se pudieran presentar.
- **Mantenimiento Adaptativo.** Se debe a cambios en el ambiente del programa y a la adaptación de nuevas unidades o módulos. De este mantenimiento pueden derivarse los siguientes mantenimientos.

- **Mantenimiento Aumentativo.** Este tipo de mantenimiento se da cuando se incluyen nuevas funciones que no se tenían contempladas al inicio del desarrollo del sistema.
- **Mantenimiento Tecnológico.** Se da debido a la evolución del hardware y software.
- **Mantenimiento Correctivo.** El proceso que incluye el diagnóstico y corrección de uno o más errores en el sistema se denomina mantenimiento correctivo.

Por la misma naturaleza inherente a todo sistema de información se requerirá de un periodo de adaptabilidad para la posterior aceptación del usuario, el mantenimiento en un principio básicamente será de tipo perfectivo, es decir, de acuerdo a las recomendaciones de los usuarios se tendrán que hacer modificaciones para la plena operación del sistema.

Una vez que el sistema esté trabajando en plenitud, es decir, que tenga el éxito deseado, se realizará el mantenimiento preventivo para garantizar el buen funcionamiento del sistema y para prevenir posibles errores.

Debido a los cambios constantes en las necesidades de las instituciones, el sistema tendrá que ajustarse a las nuevas condiciones que los cambios presenten y esto se hará a través del mantenimiento adaptativo.

El mantenimiento correctivo, será realizado en cuanto se presente un problema o error en el sistema, en este punto la participación de los usuarios es de vital importancia debido a que son ellos quienes muchas veces detectan las fallas.

4.7.9 Mecanismos de respaldo y protección

Contar con mecanismos que garanticen la seguridad de la información y que se pueda disponer de ella en todo momento es sin duda el eje de todo sistema. Para los casos de contingencia se deberán de tener en cuenta las siguientes recomendaciones:

- De ser posible, es necesario tener mecanismos de replicación en el servidor, esto es, tener un servidor espejo el cual, en caso de falla del hardware, la sustitución de este equipo pueda hacerse en forma casi inmediata.
- Se deben realizar respaldos en cinta de la base de datos y de los programas utilizados por la aplicación. Estos respaldos pueden realizarse cuando exista poca actividad en el sistema.

4.7.10 Factibilidad operativa

Para que los usuarios puedan operar el sistema de una manera adecuada es necesario capacitarlos.

Existen varias técnicas para capacitar al personal, entre las cuales figuran las siguientes:

- Seminarios e instrucción de grupo.
- Capacitación con base en procedimientos.
- Capacitación personalizada.
- Simulación.
- Entrenamiento directo.
- Uso de metodologías modernas (video, Internet, multimedia).

4.7.11 Capacitación a usuarios y administrador del sistema

En la capacitación del usuario se contemplan los siguientes puntos:

- Problemática en los procesos que se desarrollan actualmente.
- Solución y/o sistema propuesto.
- Ventajas del sistema.
- Secuencia y funcionamiento del sistema.

Para la capacitación del administrador se contemplan los siguientes puntos:

- Capacitación de usuario.
- Manejo y administración de Interbase.
- Manejo y administración de Apache.
- Administración y mantenimiento del sistema.

CONCLUSIONES

Conclusiones

Gran parte del mérito para el desarrollo del presente trabajo fue gracias a la experiencias que hemos adquirido en nuestros lugares de trabajo, ya que de esa manera pudimos avanzar de manera más sencilla en los problemas que se fueron presentando en el transcurso de toda la tesis.

La Facultad de Ingeniería nos proporciona las herramientas necesarias para la toma de buenas decisiones en cualquier proyecto, por muy difícil que este resulte, con estas herramientas se pueden realizar aplicaciones profesionales de una manera sencilla y con un rendimiento aceptable.

Las materias que se imparten en la carrera de Ingeniería en Computación nos proporcionan las bases para realizar el análisis y desarrollo de cualquier sistema.

Los conocimientos adquiridos en la Facultad de Ingeniería fueron de vital importancia para la realización de la presente tesis, ya que sin ellos no hubiera sido posible llevarla a cabo.

El Programa de Apoyo a la Titulación (PAT) es una buena opción para realizar el trabajo de tesis ya que en base a los conocimientos de quienes participamos y la aportación del asesor, se puede realizar ésta en un tiempo relativamente corto.

Una de las experiencias más importantes que se obtiene al participar en el grupo de personas que colaboran en la tesis, es la coordinación para dividirse los trabajos de investigación y desarrollo para poder avanzar de manera constante en su estructuración.

El trabajo en grupo es bueno por que se intercambian ideas, se observan diferentes opiniones y se aprende de esas observaciones, para completar las tareas asignadas.

El Sistema de Administración de Eventos (SAE) cumplió con los objetivos de automatizar y agilizar los procesos administrativos en la organización de eventos, ayudando a mantener la información más ordenada, proporcionado con esto una atención más rápida y eficiente a los usuarios, además de facilitar el flujo de información entre las diferentes áreas de las instituciones educativas.

La incorporación de la tecnología a un proceso como lo es la administración de las instituciones educativas siempre viene acompañado con las mejoras en los servicios que éstas prestan a sus usuarios. En el caso del Sistema de Administración de Eventos permitirá consultar información vía Internet de los eventos que realiza una institución, con lo que repercutirá en una mayor comodidad para las personas interesadas en tal información.

La aplicación esta diseñada como un sistema de arquitectura modular, es decir, que la aplicación consiste en una serie de bloques (módulos) con funcionalidad e interfaces entre ellos claramente definidas, facilitando de esta manera su mantenimiento.

La evolución de los sistemas de computo (lenguajes de programación, software, arquitecturas, tecnologías, etc) nos obligan a estar al día, porque de no ser así nuestros conocimientos se volverían obsoletos.

La modernización en una Institución Educativa es un proceso continuo que determina la calidad de sus servicios; contar con información confiable y disponible en todo momento representa un enorme beneficio para los usuarios ya que pueden tomar decisiones mejor fundamentadas y en ese mismo instante.

Se pudo observar la gran importancia que tienen hoy en día los sistemas de computo en las instituciones educativas por muy pequeñas o grandes que estas sean.

MANUAL TÉCNICO

MANUAL TÉCNICO

Requerimientos del Sistema

Se asume que la persona que realizará la instalación es el administrador de la red, o cuenta con privilegios de acceso equivalentes y con conocimientos acerca de:

- El uso de programas bajo Windows
- Instalación de aplicaciones de Windows
- Configuración de servicios de red bajo Windows

Los requerimientos mínimos del sistema son:

- PC con procesador Pentium III a 800 MHz o superior
- 256 MB en RAM
- 100 MB libres en disco duro
- Unidad de CD ROM
- Conexión a una red
- Sistema Operativo Windows NT 4.x con Service Pack 6 o superior, o bien, Windows 2000 Professional, Server o Advance Server.

Instalación de la aplicación

El sistema para la Administración de Eventos se divide en cuatro etapas que deben ejecutarse en el siguiente orden:

1. Instalación y configuración de Interbase 6.01
2. Instalación y configuración de Borland Database Engine (BDE)
3. Instalación y configuración del Servidor Web Apache 1.3.20.
4. Instalación y configuración de la aplicación

Instalación de Interbase 6.01

Inserte el CD de instalación en el lector de CD ROM y con ayuda del explorador de Windows localice la carpeta Interbase 6 y ejecute el programa de instalación: setup.exe haciendo doble click en el nombre del archivo. El programa comienza mostrando una pantalla indicando el inicio de instalación de Interbase, ver figura 1.

Haga click en el botón Next> para continuar el proceso. Las siguientes pantallas muestran información sobre el proceso de instalación y la licencia del producto. Para continuar el proceso en cada una de ella haga click en los botones con el texto Next>.

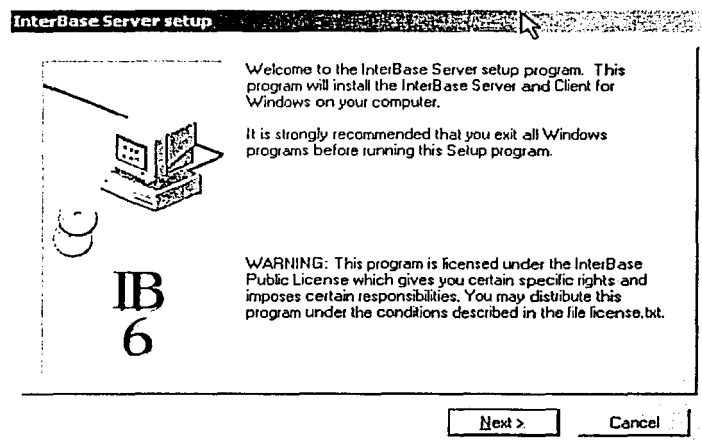


Figura 1 Inicio de la instalación.

En la siguiente pantalla se seleccionan los componentes que se instalarán, ver figura 2. Para contar con los servicios que el sistema para la administración de eventos proporciona, se requiere seleccionar los cuatro primeros componentes: Interbase Server, Interbase Client, Command-Line Tools y Graphical Tools, siendo opcionales los restantes.

También se debe elegir la ubicación donde se instalará Interbase. El programa de Instalación sugiere una ubicación predeterminada pero es posible cambiarla con ayuda del botón Browse... y utilizando el selector de archivos de Windows; al seleccionar la ubicación, el programa regresará a esta pantalla.

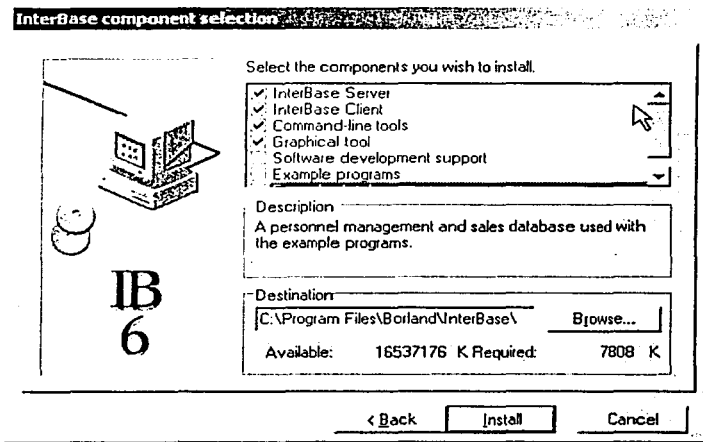


Figura 2 Selección de componentes.

Una vez hecho esto, se hará click en el botón Install, con lo que se iniciará el copiado de los archivos al disco duro.

Al finalizar, aparecerá una pantalla indicando que el proceso ha concluido y para salir de la aplicación se hará click en el botón Finish.

Puede suceder que alguna otra aplicación ya esté utilizando Interbase, en cuyo caso no será posible instalar una nueva copia, sin embargo la versión instalada se puede utilizar sin problema alguno siempre y cuando sea la versión 6.01 o superior.

Configuración de Interbase 6.01

Para la configuración de Interbase, se accederá la consola de control por medio de la barra de tarea de Windows

Inicio > Programas > Interbase > Interbase Server Manager

Se abrirá el administrador de Interbase, mostrando el control de los servicios, ver figura 3.

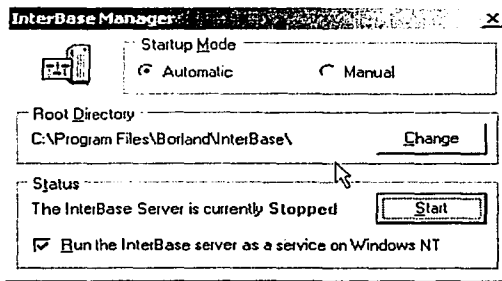


Figura 3 Administrador de Interbase.

Para arrancar el servicio, se hará click en el botón Start y el indicador de estado Stopped cambiará a un estado de ejecución.

Una vez iniciado el servicio, se debe cerrar la ventana con el botón superior derecho.

No es necesario repetir este proceso en lo sucesivo, ya que se iniciará cada vez que Windows arranque.

Instalación de Borland Database Engine (BDE)

Inserte el CD de instalación en el lector de CD ROM y con ayuda del explorador de Windows localice la carpeta BDE y ejecute el programa de instalación: setup.exe haciendo doble click en el nombre del archivo, como se muestra en la figura 4.

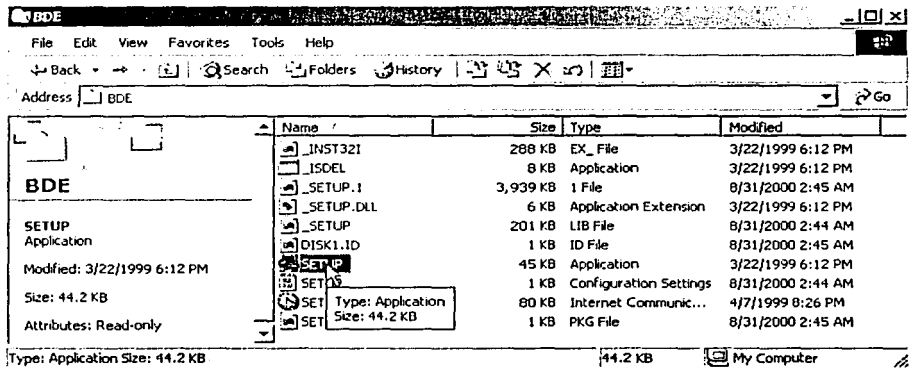


Figura 4 Instalación del BDE.

Configuración de Borland Database Engine (BDE)

Para configurar el BDE, deberá abrir su panel de control, al cual se accesa por medio de la barra de tareas:

Inicio > Configuración > Panel de Control

y se deberá localizar el icono BDE Administrator, ver figura 5.

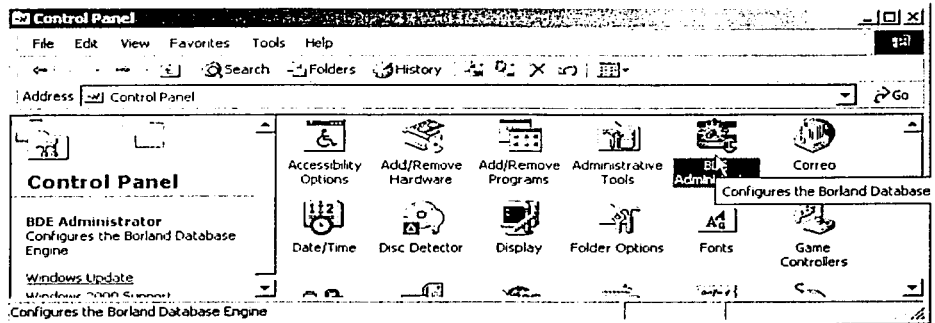


Figura 5 Administrador del BDE.

En la ventana que aparecerá, localice Databases, como se muestra en la figura 6, de click derecho y seleccione New... para crear el alias hacia la base de datos de la aplicación.

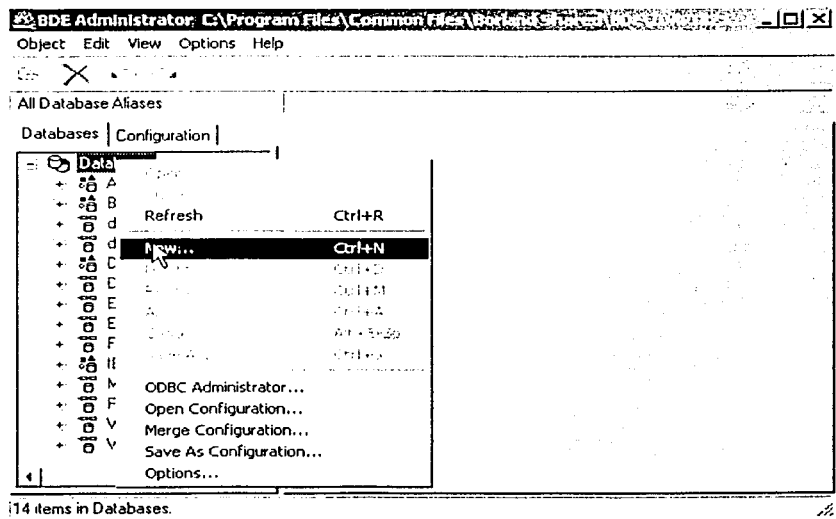


Figura 6 Creación del alias.

Se abrirá la pantalla de nuevo alias de base de datos, en la cual se deberá seleccionar INTRBASE, ver figura 7.

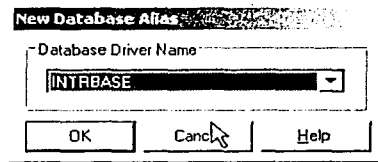


Figura 7 Driver de la base de datos.

En la siguiente pantalla se deberá cambiar el nombre del alias INTERBASE1 por CSI_SQL, dado que algunos elementos del sistema hacen referencia a nuestra base de datos por medio de este nombre, ver figura 8.

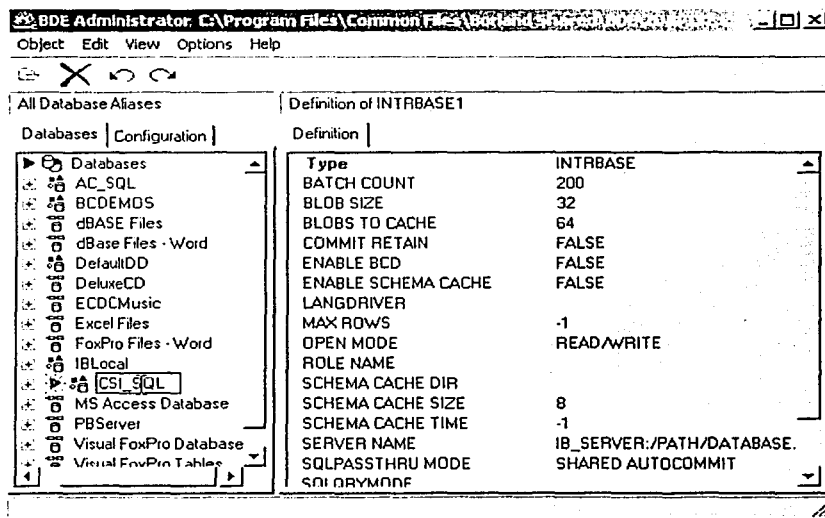


Figura 8 Modificación del alias.

Para que el BDE pueda proporcionar acceso a la base de datos del sistema, se deberá hacer referencia a ésta última, modificando los siguientes valores:

1. Nombre del Servidor. Se deberá escribir el nombre de la base de datos en el campo SERVER NAME, de acuerdo a la siguiente sintaxis, ver figura 9.

<unidad de disco><directorio><base de datos>

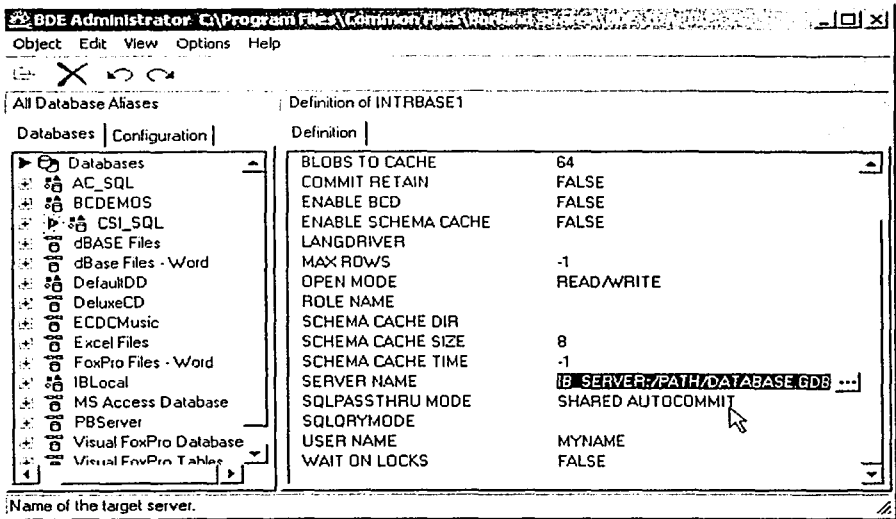


Figura 9 Selección de la base de datos.

Alternativamente, se puede seleccionar la base de datos con el icono de elipsis al extremo derecho de ese mismo renglón y por medio del selector de archivos se puede buscar el archivo csi_db.gdb, como se muestra en la figura 9.

2. Nombre del Usuario. Se sustituirá en el campo USER NAME, el valor que aparece (MYNAME) por el de SYSDBA, ver figura 10.

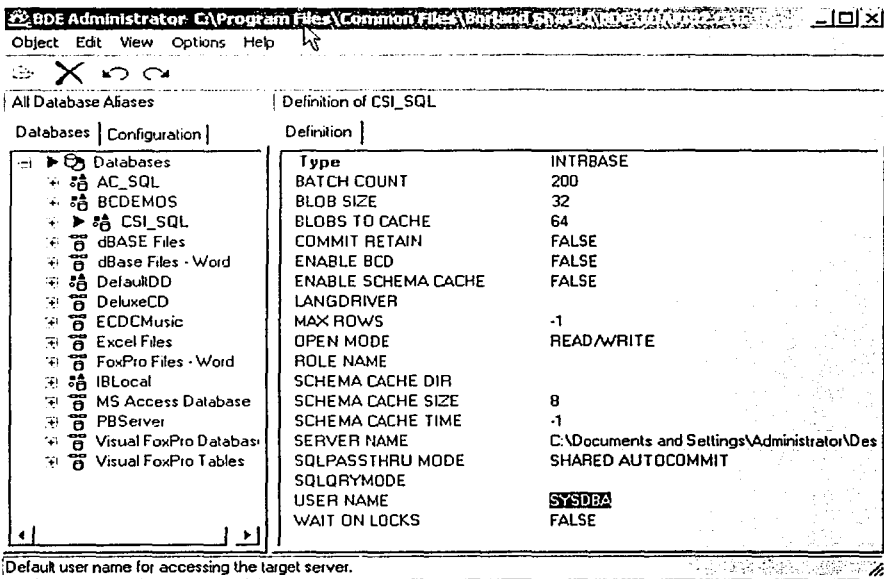


Figura 10 Nombre de usuario.

- Oprimir el botón de actualización para grabar los cambios que se han hecho, ver figura 11.

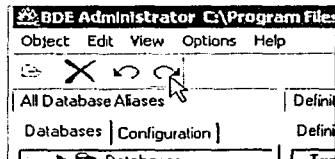


Figura 11 Grabar cambios.

- El programa desplegará una ventana para confirmar los cambios, para lo cual se selecciona OK. En caso que los datos sean correctos.

Una vez que se ha configurado la base de datos, es posible probar la configuración haciendo nuevamente doble click en el alias CSI_SQL, activando una ventana que solicita el usuario y la contraseña.

En este caso el nombre de usuario (USER NAME) de la base de datos será SYSDBA y la contraseña (PASSWORD) será proporcionado por el administrador del sistema, como se muestra en la figura 12.

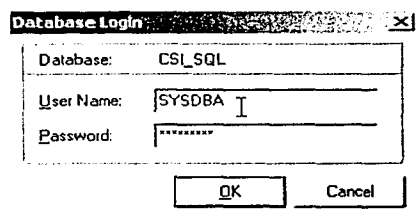


Figura 12 Pantalla de acceso.

Si la configuración fue correcta después de proporcionar los datos y hacer click en el botón OK, reaparecerá la ventana que se muestra en la figura 10 con un recuadro verde alrededor del alias de la base de datos CSI_SQL indicando que la configuración es correcta.

Una vez que se ha verificado esto, es necesario cerrar la base de datos haciendo click con el botón derecho en el nombre CSI_SQL y seleccionando la opción Close.

Instalación del Servidor Web Apache 1.3.20.

Inserte el CD de instalación en el lector de CD ROM y con ayuda del explorador de Windows localice la carpeta Apache y descompacte el archivo Apache_1.3.20-Mod_SSL_2.8.4-OpenSSL_0.9.6a-WIN32.zip.

Configuración del Servidor Web Apache 1.3.20.

Para la configuración del Servidor Web Apache, localice la carpeta conf y edite el archivo httpd.conf véase capítulo 2 inciso 4, lo cual será suficiente para que el servidor funcione de manera adecuada (ver figura 13).

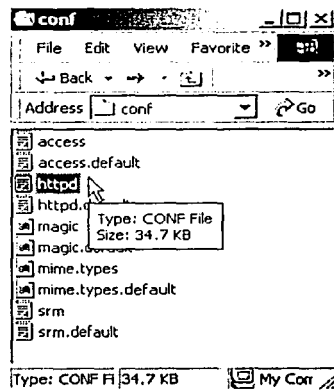


Figura 13 Archivo de configuración.

Finalmente para iniciar el servidor web localice dentro del directorio bin el archivo Apache.exe y ejecútelo (ver figura 14).

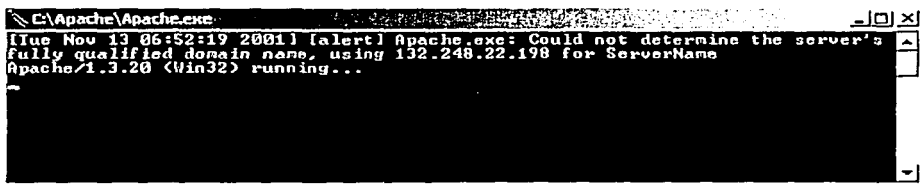


Figura 14 Puesta en marcha del servidor web.

Instalación y configuración de la aplicación

Inserte el CD de instalación en el lector de CD ROM y con ayuda del explorador de windows localice y abra el directorio sistema, copie el contenido del directorio paginas (tesis e imágenes) al directorio htdocs del servidor web, ver figura 15. Por último copie el contenido del directorio cgis (tesis) al directorio cgi-bin del servidor web. La figura 16 muestra los CGIs del sistema.

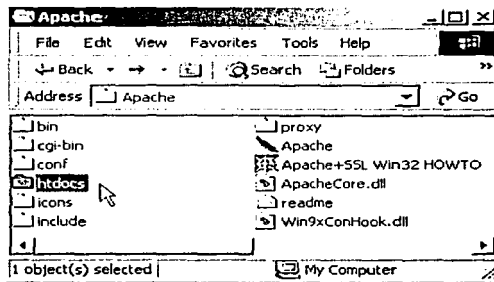


Figura 15 Ubicación física de las páginas e imágenes.

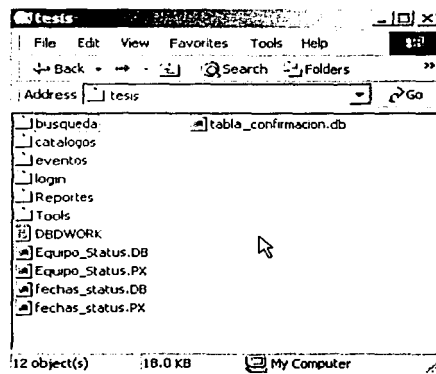


Figura 16 CGIs del sistema.

MANUAL DE USUARIO

MANUAL DE USUARIO

Introducción

El sistema de administración de eventos surge de la necesidad de automatizar los procesos involucrados en la organización de los mismos. Para llevar a cabo esta tarea el sistema cuenta con características que nos permiten manejar la información de manera sencilla y amigable.

Objetivos:

1. Permitir al usuario contar con un documento que especifique el funcionamiento y operación del sistema.
2. Describir todas y cada una de las pantallas con que cuenta la aplicación:
 - Acceso
 - Eventos
 - Consultas
 - Reportes
 - Catálogos
3. Explicar los diferentes perfiles de usuario que existen.

Inicio de la aplicación

Para dar inicio a la aplicación debe ejecutar el navegador Netscape Navigator. La dirección a la cual se debe de conectar para poder acceder al sistema será proporcionada por el administrador, ver figura 1

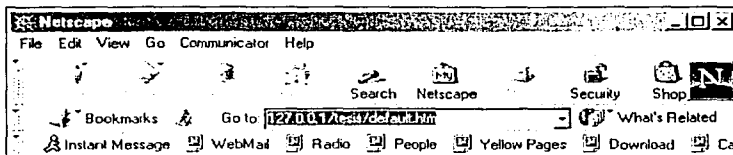


Figura 1 Netscape Navigator.

Pantalla de Inicio

La página de inicio solicitará el nombre de usuario y contraseña, los cuales deben obtenerse con el administrador del sistema, una vez ingresados estos datos se debe de hacer click con el ratón sobre el botón de Ingresar.

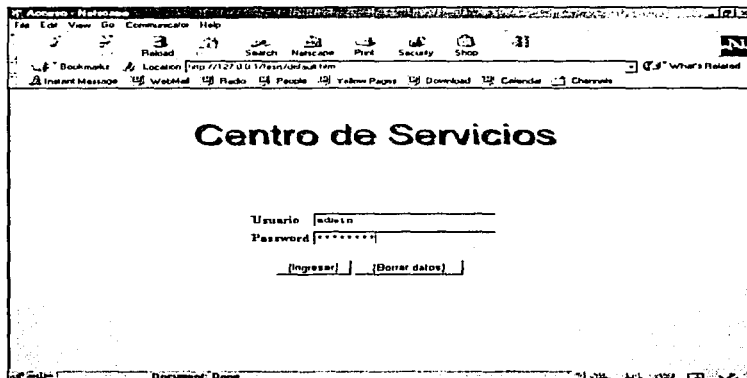


Figura 2 Página de inicio.

Menú Principal

Una vez que la aplicación reconoce que el nombre y la contraseña son los de un usuario válido, inmediatamente el sistema mostrará la pagina del menú principal (ver figura 3). El menú principal variará de acuerdo al perfil de usuario (Lectura, Lectura/Escritura, Administrador).

La página del menú principal contiene cuatro módulos: Eventos, Búsquedas, Reportes y Catálogos. Sólo el usuario con privilegios de administrador tendrá acceso a todos los módulos.

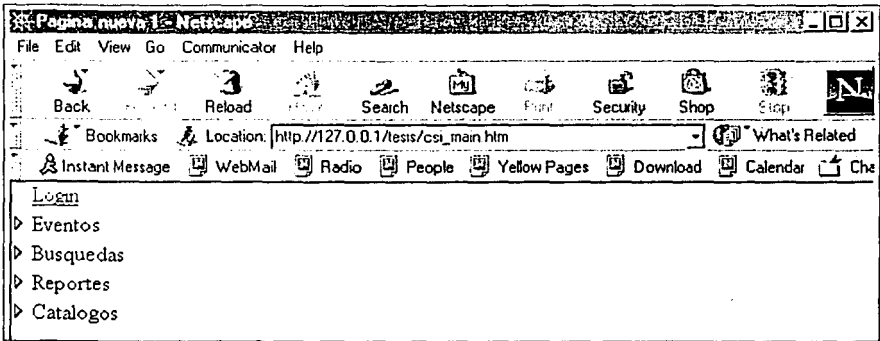
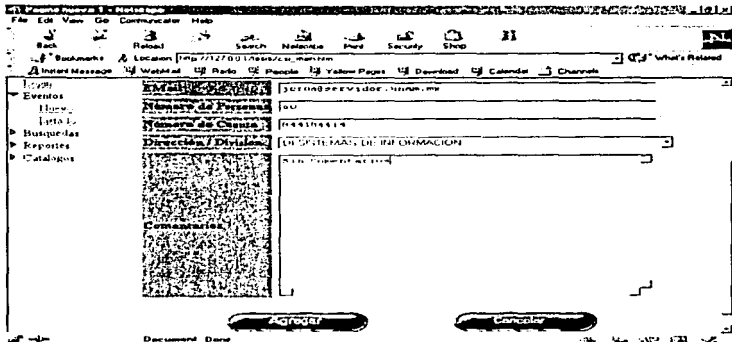


Figura 3 Menú principal.

Módulo Eventos

Este módulo cuenta con dos opciones: Nuevo y Listado, la primera de ellas nos permite agregar un nuevo evento (figura 4).



**TESIS CON
FALLA DE ORIGEN**

Figura 4 Registro de un nuevo evento.

Una vez ingresados los datos para el nuevo evento debe hacer click sobre el botón agregar, respondiendo el sistema con la página de la figura 5. En esta página aparecerán 7 nuevas opciones, las dos primeras le permitirán modificar y borrar el evento.

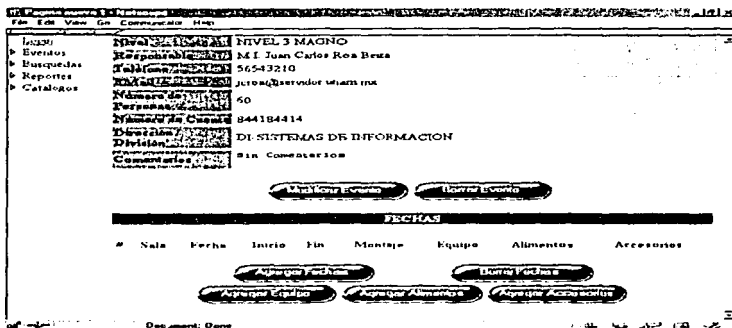


Figura 5 Página para agregar y editar información del evento.

En la opción de agregar fechas podrá definir las fechas y horario en que se llevará a cabo el evento así como el tipo de montaje que tendrá, ver figura 6.

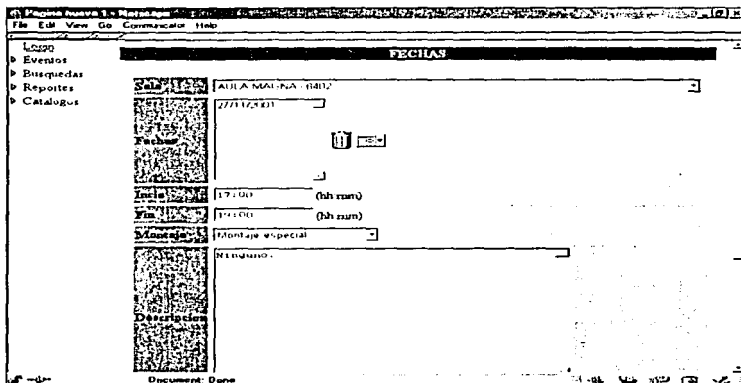


Figura 6 Página para definir las fechas y el horario del evento.

Una vez que haya concluido de ingresar los datos debe hacer click sobre el botón agregar, el sistema mostrará la página de la figura 7.

Evento nueva 1 - Netstage

File Edit View Go Comunicador Help

Logueo

Eventos **EVENTO**

Busquedas

Reportes

Catalogos

Clave del Evento: 02020005

Nombre del Evento: Examen Profesional

Responsable: MI Juan Carlos Ros Briza

FECHAS

FECHA	HORA_INICIO	HORA_FIN	SALA	CVE_EVENTO
27/11/2001	05:00:00 p.m.	07:00:00 p.m.	AULA MAGNA - 8402	Reservado

Figura 7 Página que muestra las fechas dadas de alta.

Para continuar agregando fechas, equipo o alimentos debe hacer clic en la clave del evento.

Una vez que haya agregado las fechas, equipo, alimentos y accesorios del evento, el sistema mostrará la información de la figura 8.

Evento nueva 1 - Netstage

File Edit View Go Comunicador Help

Logueo

Eventos

Busquedas

Reportes

Catalogos

Teléfono: 56543210

E-mail: jcroa@servidor.unam.mx

Número de Personas: 60

Número de Cuenta: 844184414

Dirección: DI-SISTEMAS DE INFORMACION

División:

Comentarios: Sin Comentarios

Modificar Evento

Borrar Evento

FECHAS

#	Sala	Fecha	Inicio	Fin	Montaje	Equipo	Alimentos	Accesorios
1	AULA MAGNA - 8402	27/11/2001	17:00	19:00	Montaje especial	✓	✓	✓

Agregar Fechas

Borrar Fechas

Agregar Equipo

Agregar Alimentos

Agregar Accesorios

http://127.0.0.1/cgi-bin/tesis/eventos/eventos.asp/fecha/show_fecha=30

Figura 8 Página que indica que se han elegido fechas, equipo, alimentos y accesorios.

La opción listado de eventos le permitirá ver los eventos que se encuentran dados de alta. Los datos que se requieren para realizar un listado son el número de semestre, un número de folio inicial y otro final; aunque puede omitir alguno de estos datos, es recomendable definir alguna restricción, y así obtener un lista reducida. La figura 9 muestra la pantalla para realizar una listado de eventos.

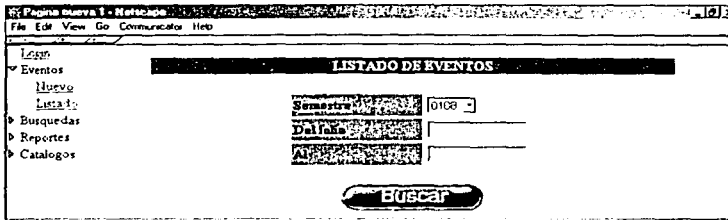


Figura 9 Página de listado de eventos.

En la figura 10 se muestran los eventos registrados en el semestre 0108.

CVE_EVENTO	NOMBRE_EVENTO	NOMBRE_RESPONSABLE	TELEFONO	EMA
01080001	Diplomado "Administración de Manufactura"	Regina Alans	5757	ralans@campus.c
01080002	Diplomado "Administración de los Recursos Humanos"	Cristina Gonzalez	5772	c.boera@campus.c
01080003	Diplomado "Administración Financiera"	Cristina Gonzalez	5772	c.boera@campus.c
01080004	Diplomado "Administración por Calidad Total"	Thalia Gomez	3363	tgomez@campus.c
01080005	Diplomado "Control Estadístico de Calidad"	Regina Alans	5757	ralans@campus.c
01080006	Diplomado "Desarrollo de Habilidades Gerenciales"	Regina Alans	5757	ralans@campus.c
01080007	Diplomado "Direccion y Administración de Proyectos"	Regina Alans	5757	ralans@campus.c

Figura 10 Eventos registrados.

Para editar la información de un evento en particular deberá hacer clic sobre la clave de éste.

Módulo de Búsquedas

Este módulo estará disponible para todos los usuarios que ingresen al sistema. Existen tres tipos de búsquedas:

- General
- Por Salas Disponibles
- Por Folio del Evento

Búsqueda General. Para realizar una consulta de este tipo debe proporcionar al menos uno de los siguientes datos: sala, fecha o nombre del evento, ver figura 11.

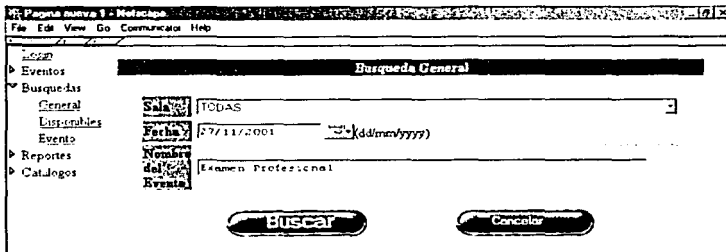


Figura 11 Página de búsqueda general.

El resultado de la búsqueda anterior se muestra en la figura 12.

Busqueda General

Sala: TODAS
 Fecha: 27/11/2001 (dd/mm/yyyy)
 Nombre del Evento: Examen Profesional

Resultados

#	Evento	Nombre del Evento	Sala	Fecha	Inicio	Fin
1	20020095	Examen Profesional	AULA MAGNA - 8402	27/11/2001	17:00	19:00

Figura 12 Resultado de la búsqueda general.

Para editar la información de un evento en particular deberá hacer click sobre la clave de éste.

Búsqueda por Salas Disponibles. En este tipo de búsqueda los datos de sala, fecha, hora de inicio y hora final son obligatorios para formar el criterio de consulta, ver figura 13.

Busqueda de Salas Disponibles

Sala: ALBERCA - ALBERCA
 AUDIOVISUAL LANGUAGE LAB - 6302
 AULA MAGNA - 6104
 AULA MAGNA - 6105
 AULA MAGNA - 6106
 AULA MAGNA - 6207
AULA MAGNA - 8402
 ANILAS VI - EXPLANADA
 Auditorio - AUDITORIO
 BIBLIOTECA - JARDIN LATEHAL

Fecha: 27/11/2001 (dd/mm/yyyy)
 Inicio: 17:00 (hh:mm)
 Fin: 20:00 (hh:mm)

Figura 13 Página de búsqueda de salas disponibles.

La figura 14 muestra el resultado de la búsqueda de salas disponibles.

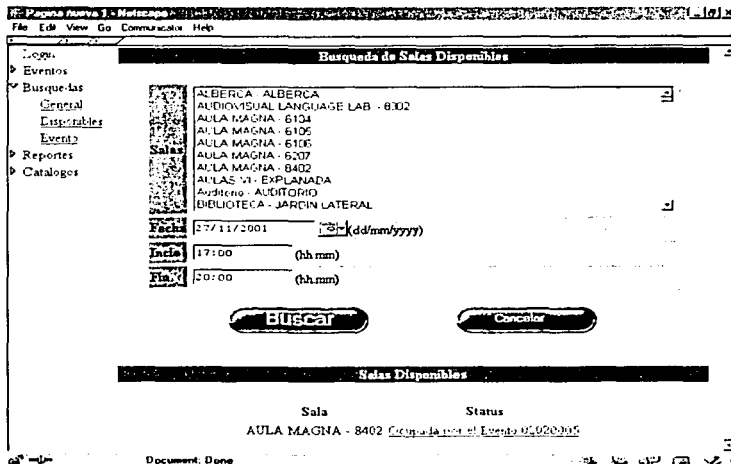


Figura 14 Resultado de la búsqueda de salas disponibles.

Búsqueda por Folio del Evento. Para este tipo de búsqueda debe proporcionar el número de folio del evento que desea consultar, ver figura 15.

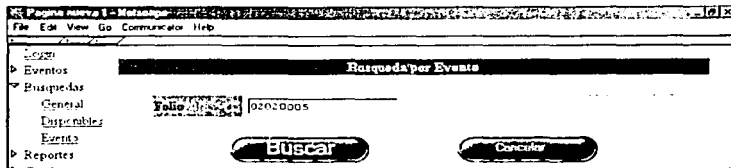


figura 15 Búsqueda por folio del evento.

El resultado de este tipo de búsqueda se muestra en la figura 16.

The screenshot shows a web browser window with a menu on the left and a main content area. The menu includes 'Inicio', 'Eventos', 'Busquedas', 'General', 'Disponibles', 'Eventos', 'Reportes', and 'Catalogos'. The main content area is titled 'EVENTO' and displays the following details:

Clave del Evento	02020005
Nombre del Evento	Examen Profesional
Nivel	NIVEL 3 MAGNO
Responsable	M I. Juan Carlos Roa Beiza
Teléfono	56543210
E-Mail	jeroa@servidor.unam.mx
Número de Personas	60
Número de Cuenta	844184414
Dirección	DI-SISTEMAS DE INFORMACION
División	
Comentarios	Sin Comentarios

Below the table are two buttons: 'Modificar Evento' and 'Borrar Evento'. At the bottom of the main content area is a section titled 'FECHAS'.

Figura 16 Resultado de la búsqueda por folio del evento.

Reportes

El sistema incluye la generación de varios tipos de reportes, los cuales permitirán al administrador contar con información oportuna y confiable.

Reporte General. El primer tipo de reporte es el general, y consiste en seleccionar una fecha inicial y otra final, ver figura 17.

Figura 17 Página de reporte general.

El resultado del reporte general incluye toda la información de los eventos (ver figura 18). A través del número de folio se puede editar la información del evento.

# Folio	Evento	Horario	Sala	Descripción	Equipo	Alimentos
1 0120022	Ases de Consejo de Ciudad	27/11/2001 08:00 - 14:00	ESTUDIO PARA PROFESORES	Reporte de 11:00 a 12:00		<input type="checkbox"/>
2 0120012	Ases de Coll. de Ciudad	27/11/2001 10:00 - 12:00	SALA DE CONSEJO BIBLIOTECA - F201			<input type="checkbox"/>
3 0120013	Profesional - M.I. San Carlos Bax Bax	27/11/2001 17:00 - 19:00	ATA MAQUINA - 8102	Reporte	INDIVIDUAL MULTIMEDIA 3 Equipo Pnp 1 (191) Via Fideo Module CompuC	<input checked="" type="checkbox"/>

Figura 18 Resultado del reporte general.

Para generar un reporte por alimentos se sigue el mismo procedimiento que el realizado para el reporte general. En la figura 19 se muestra la página para realizar un reporte por alimentos.

Fecha Inicial: 27/11/2001 (dd/mm/yyyy)

Fecha Final: 27/11/2001 (dd/mm/yyyy)

BUSCAR **Cancelar**

Figura 19 Página de reporte de alimentos.

Reporte por Uso de Salas. Cuando se elige un reporte de este tipo aparece un cuadro en donde se despliegan todas las salas registradas y solo es necesario seleccionar una de ellas, ver figura 20.

ALBERCA ALBERCA

AUDITORIUM LANGUAGE LAB - B 102

AULA MAGNA 6104

AULA MAGNA 6105

AULA MAGNA 6106

AULA MAGNA 6207

AULA MAGNA 6208

AULAS VI - EX-LANANIA

Auditorio - AUDITORIUM

BIBLIOTECA - BIBLIOTHECA LATERAL

Fecha Inicial: 01/11/2001 (dd/mm/yyyy)

Fecha Final: 30/11/2001 (dd/mm/yyyy)

*Al dejar en blanco el campo el reporte se realizará sobre todos los eventos del semestre seleccionado

BUSCAR **Cancelar**

Figura 20 Página de reporte de uso de salas.

La figura 21 muestra el resultado de un reporte por uso de salas.

#	Sala	# SubEventos	Horas
0	AULA MAGNA - 8402 0		00 00

Figura 21 Resultado de un reporte de uso de salas.

Reporte por eventos y subeventos. Para la generación de este tipo de reporte debe proporcionar los siguientes datos: semestre folio inicial y folio final. Ver figura 22.

Reporte Eventos y Subeventos

Semestre: 0100

Del folio: _____

Al folio: _____

*Al dejar en blanco el campo el reporte se realizará sobre todos los eventos del semestre seleccionado.

BUSCAR

Figura 22 Página de reporte por eventos y subeventos.

El resultado de un reporte por eventos y subeventos se muestra en la figura 23.

#	Folio	Nombre del Evento	Nivel	Direccion/Division	SubEventos
1	02020001	Evento de Prueba	NIVEL 1 BASICO	DACS-DIRECCION	1
2	02020002	Evento Prueba	NIVEL 1 BASICO	DSA-COMERCIO ELECTRONICO	1
3	02020003	Examen Profesional	NIVEL 3 MAGNO	DI-SISTEMAS DE INFORMACION	0
4	02020004	Examen Profesional	NIVEL 3 MAGNO	DI-SISTEMAS DE INFORMACION	0
5	02020005	Examen Profesional	NIVEL 3 MAGNO	DI-SISTEMAS DE INFORMACION	1

Figura 23 Resultado de un reporte por eventos y subeventos.

A través del número de folio se puede editar la información del evento.

Reporte por Uso de Equipo. Este reporte requiere como dato el nombre del equipo, la fecha inicial y la fecha final. La figura 24 presenta la forma de realizar un reporte por uso de equipo.

Reporte de uso de Equipo

Semestre: 0202

Equipo: [List of equipment types including AUDIOVISUAL 1-5, MULTIMEDIA 1, and AULAS II]

Fecha Inicial: 27/11/2001

Fecha Final: 27/11/2001

Botones: **Buscar**, **Cancelar**

Figura 24 Reporte por uso de equipo.

La figura 25 muestra el resultado del reporte.

Reporte de Uso de Equipo - Semestre 0202
Del 27/11/2001 al 27/11/2001

#	Sala	# SubEventos	Horas
0	AUDIOVISUAL MULTIMEDIA 2 Equipo Fijo (TV31-Vhs-Retro-Modular-CompuPC)	1	02:00

Figura 25 Resultado de un reporte por uso de equipo.

Catálogos

Este módulo permite administrar los catálogos de salas, equipo, alimentos, accesorios, montajes, usuarios, divisiones y niveles.

Salas. El catálogo de salas tiene dos opciones: listado y agregar. La primera permite visualizar las salas registradas en el sistema, ver figura 26. A través del número de sala se puede editar la información de ésta.

#	Sala	Capacidad
1	ALBERCA - ALBERCA	1
2	AUDIOVISUAL LANGUAGE LAB - 8302	42
3	AULA MAGNA - 6103	130
4	AULA MAGNA - 6105	133
5	AULA MAGNA - 6106	123
6	AULA MAGNA - 6207	125
7	AULA MAGNA - 8402	90
8	AULAS VI - EXPLANADA	1
9	Auditorio - AUDITORIO	190
10	BIBLIOTECA - JAREDJ LATERAL	1
11	BIBLIOTECA - SALA DE EXPOSICIONES	1
12	BOBENA CSI - AULAS II	1
13	CABINA CONTROL - 6302	0
14	CABINA CONTROL - 6305	0
15	CAE - LOBBY	1
16	CAE - SALON DE AEROBICS	1
17	CAE - SALON DE BAILE	1
18	CAE - TEATRO	1

Figura 26 Catálogo de salas.

Para agregar una nueva sala es necesario proporcionar los siguientes datos: nombre de la sala y capacidad con la que cuenta. La figura 27 muestra la página para agregar una nueva sala.

SALA

Sala: COMPUTO AVANZADO

Capacidad: 100

[Agregar] [Cancelar]

Figura 27 Página para agregar una nueva sala.

Para la administración de los catálogos de equipo, alimentos, accesorios, montajes, divisiones y niveles; se sigue el mismo procedimiento.

Catálogo de Usuarios. Este catálogo permite listar los usuarios registrados en el sistema, ver figura 28. Por medio del número se puede modificar la información del usuario.

#	Usuario	Tipo de Usuario
1	admin	
2	csernc1	
3	ROGELIO	
4	csernc2	
5	csernc3	
6	csernc4	
7	csernc5	
8	csernc6	
9	DANIEL	
10	JOAQUIN	
11	MANUEL	
12	AGUSTIN	
13	JUAN	
14	RUBEN	
15	FELEX	
16	JUAN RAMON	
17	LORENZO	
18	VICTOR	

Figura 28 Catálogo de usuarios.

Para tener acceso al sistema será necesario contar con un nombre de usuario y una contraseña los cuales deben de ser proporcionados por el administrador del sistema.

Para agregar un nuevo usuario al sistema se deben proporcionar los siguientes datos: nombre, contraseña y tipo de usuario. Ver figura 29.

The screenshot shows a web browser window with the title "Página Nueva 1 - Microsoft...". The browser's address bar and menu bar are visible. On the left, there is a sidebar menu with the following items: Eventos, Busquedas, Reportes, Catalogos, Salas, Equipo, Alimentos, Accesorios, Montajes, Usuarios, Listado, Agregar, Du-Div, and Niveles. The main content area is titled "USUARIOS" and contains a form with the following fields and options:

- Usuario:** A text input field containing "jocinto".
- Password:** A text input field containing "jocintus".
- Tipo de Usuario:** A radio button selection with three options: "Solo Lectura", "Lectura y Escritura", and "Administrador". The "Administrador" option is selected.

At the bottom of the form, there are two buttons: "Agregar" and "Cancelar".

Figura 29 Página para agregar un nuevo usuario.

BIBLIOGRAFÍA

Bibliografía

Análisis y Diseño Orientado a Objetos

Grady Booch

Addison Wesley, 1996

Calidad de la Educación

Carlos Muñoz Izquierdo

México, 2000

CGI Programming on the World Wide Web

Shishir Gundavaram

O'Reilly & Associates, Inc.

Delphi 5 Developer's Guide

SAMS

**Diseño de un sistema para la administración de los horarios de la
carrera de Ingeniería Mecánica Eléctrica**

Yolanda Cuevas Salgado

Tesis, U.N.A.M.

Servidor Apache al descubierto. La solución más completa.

Rich Bowen & Ken Coar

Prentice Hall

FrontPage 2000: The Complete Reference

Martin S. Matthews & Erik B. Poulsen

Fundamentos de Bases de Datos. Segunda Edición

Henry F. Korth & Abraham Silberschutz

Mc Graw Hill

HTML 4 para Web

Laura Lemay

Printice Hall

Ingeniería de Software Manual de Referencia

Robert Pressman, 1998.

Java Script Bible. 3rd Edition

Danny Goodman

IDG Books WorldWide

Microsoft NT Server 4.0

Charle Russell y Sahron Crawford

Mc Graw Hill

Programación Delphi 5

Francisco Chate

Anaya

The Unified Modeling Lenguaje

James Rumbaugh

Addison Wesley, 2001

<http://www.interbase.com.mx/>

<http://www.aytsi.com/>

<http://www.clubdelphi.com/>

<http://www.educacion.org.mx/>

<http://www.geocities.com/elplanetamx/informacionsistemas.html>

<http://www.ifie.edu.mx/premios.htm>

<http://www.presidencia.gov.co/polsoc/salud/amplreno.htm>

<http://www.puc.cl/cextension/academic/gesyprod.html>

<http://www.sep.gob.mx/>

<http://www.sepiensa.or.mx/>

<http://www.rational.com.mx/>

gidis.ing.unlpn.edu.ar/uml/uml.html

si.mty.itesm.mx/InfoGral/pIntro.htm