



Universidad Nacional Autónoma de México

Facultad de Ingeniería

“PORTAL DE LA ACADEMIA REGIONAL DE
INFORMÁTICA DE LA DGETI”

TESIS

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

PRESENTAN:

MEJIA BRAVO ROCIO JOSEFINA

SALGADO GALVÁN SUSANA

SÁNCHEZ MÁXIMO ADRIANA



DIRECTOR DE TESIS: MI JUAN CARLOS ROA BEIZA.

Ciudad Universitaria.

Octubre 2008.

Facultad de Ingeniería UNAM



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.

Dedico esta tesis con gran cariño y respeto a mis sabias amigas y consejeras: ***Lic. Ma. De Lourdes Beltrán Lara, Lic. Gloria Vielma Baños***, que por años me han apoyado y me han alentado admirablemente a seguir y obtener este y otros tantos logros, agradeciendo que me inculcaran sus conocimientos, producto de una amplia experiencia y una exitosa vida profesional.

A mi hermano ***Heriberto Salgado Galván***, que aún en tu ausencia sigo aprendiendo de ti, valentía, coraje e ímpetu para encontrar lo que por mucho tiempo has buscado, la libertad en toda la extensión de la palabra misma.

Con gratitud, admiración y respeto a mi amigo ***Ing. Jesús A. Pérez Espiridión***, quien además de ser una persona exitosa en todos los ámbitos, cuenta con la virtud de ser un gran líder; quien desde que inicie este proyecto me alentó y apoyo para terminar este ciclo aún a pesar de los inconvenientes, y a quien agradezco por ayudarme a impulsar mi carrera profesional.

La lista a agradecer en el desarrollo y elaboración de esta tesis es inmensa, porque desearía mencionar a todos aquellos que positivamente me he encontrado en este recorrido de mi vida y que afortunadamente me esta permitiendo cerrar un ciclo que ha durado por mucho tiempo abierto y que por fin llega a la etapa final, para iniciar o concluir otros más.

A mis *padres y hermanos*: Mi gran familia que han estado siempre en todo momento para apoyarme.

Al *Dr. Gustavo Albarran Hernández*: Quien gracias a su admirable paciencia y sabiduría, me alentó a cerrar este ciclo de mi carrera profesional.

A *Mine*: Por compartir conmigo parte de nuestro recorrido como estudiantes y ser una grandiosa amiga y que gracias a tu apoyo pude llegar a la recta final de la carrera.

A *Anabel*: Que en el camino por la curiosidad de conocer y saber más, el destino me permitió conocerte y que en el desarrollo de la tesis me ayudaste con paciencia a acomodar y resolver mis dudas con la programación.

A *Janelly y Alberto*: Dos nobles Personas con una gran madures e inteligencia que me apoyaron y dieron ánimo para continuar en una etapa muy difícil de mi formación como persona.

A ti *Anita* que me seguiste alentado hasta el final.

Para no omitir a todos aquellos que ya tomaron rumbos, vidas y destinos distintos, pero que me permitieron compartir parte de una etapa de su vida y que gracias a ello aprendí ha valorar, disfrutar y apreciar el presente, a querer más y por ende explorar vidas y mundos diferentes.

¡Gracias!.

OBJETIVO

Diseño e implantación de un portal en internet para la Dirección General de Educación Tecnológica Industrial(DGETI). La DGETI es una dependencia adscrita a la Subsecretaría de Educación Media Superior (SEMS), que depende de la Secretaría de Educación Pública.

Se requiere de elaborar un portal que se acceda vía WEB y que contenga toda la información relevante de la Academia Regional de Informática, que permita informar a los miembros de la academia de las actividades que se están desarrollando y las que se están programando para futuras reuniones; ser analizadas y discutidas, y llegar a posibles acuerdos.

Por lo que se requiere que esté portal cuente con toda la información actualizada referente a las actividades próximas a desarrollarse y que los miembros puedan tomar las consideraciones necesarias para llevar a cabo a buen término estas actividades.

Se deberá analizar, desarrollar e implantar en software de distribución gratuita, para evitar la inversión en software propietario y con esto, evitar gastos en el pago de licencias.

El sitio deberá ser administrado y actualizado por el personal designado por la DGETI en acuerdo con la academia.

En el portal se pondrá la información que se genera en la academia de informática, brindando un servicio a nivel nacional de las actividades que se realizarán durante el año, con respecto a cursos, exposiciones, trabajos de investigación elaboración de secuencias y materiales didácticos, publicación de artículos e intercambio bibliográfico.

El portal estará regido bajo los esquemas sistemáticos de factibilidad, funcionalidad y operatividad, de tal manera que sea atractivo y de interés para los docentes y que coadyuve en su práctica cotidiana laboral.

Es así que el portal, cubrirá la necesidad del intercambio de información de los diferentes planteles a nivel nacional almacenando la información en una base de datos versátil que cubra los requerimientos del sistema.



Índice.

ÍNDICE



Portal de la Academia Regional de Informática de la DGETI.





Índice.

| | |
|--|-----|
| Dedicatoria | iii |
| Agradecimiento | v |
| Objetivo | vii |
| | |
| Capítulo 1 | 1 |
| 1. Entorno del Problema | 3 |
| 1.1. Introducción..... | 3 |
| Resumen..... | 4 |
| 1.2. Rubro de la Empresa..... | 5 |
| Misión..... | 6 |
| Visión..... | 6 |
| Dirección General..... | 7 |
| Dirección de Planeación y Evaluación..... | 10 |
| Dirección Técnica..... | 10 |
| Dirección de Apoyo a la Operación Estatal..... | 12 |
| Coordinación de Organismos Descentralizados Estatales de CECyTES..... | 13 |
| Coordinación Administrativa..... | 15 |
| 1.3. Análisis del Entorno Académico de Informática..... | 16 |
| Redes y Telecomunicaciones..... | 17 |
| Objetivo..... | 18 |
| Estrategias..... | 18 |
| Internet e Intranet..... | 19 |
| Objetivo..... | 19 |
| Estrategias..... | 19 |
| 1.4. Estructura Académica..... | 21 |



| | |
|---|----|
| 1.5. Como se Distribuye Actualmente la Información..... | 21 |
| Capítulo 2 | 25 |
| 2. Marco Teórico | 27 |
| 2.1. Principales tendencias de la WEB y metodologías de desarrollo..... | 27 |
| Funcionamiento Básico de la Web..... | 28 |
| Tecnologías de aplicación en el servidor web..... | 30 |
| Metodologías de Desarrollo de sistemas Web..... | 32 |
| Método ERMM (Relationship Management Methodology)..... | 36 |
| Web Site Design Method(WSDM)(Método de Diseño de un Sitio Web). .. | 37 |
| Web Modeling Language (WebML) (Lenguaje de Modelado Web)..... | 39 |
| UML-based Web Engineering methodology (UWE)(Metodología de Ingeniería Web basado en UML)..... | 41 |
| Ariadne Development Method (ADM)(Método de Desarrollo Ariadne)... .. | 41 |
| 2.2. Características, Ventajas y Desventajas de las Bases de Datos Relacionales..... | 43 |
| Características de las bases de datos..... | 46 |
| Cardinalidad de las relaciones..... | 47 |
| Bases de Datos Relacionales..... | 48 |
| Estructura del Modelo Relacional..... | 49 |
| Reglas de Codd..... | 54 |
| 2.3. Características, Ventajas y Desventajas del Software Gratuito..... | 58 |
| 2.3.1. Características, Ventajas y Desventajas de las Bases de Datos Gratuitas..... | 65 |
| Características..... | 65 |
| Acceso a Datos..... | 66 |
| Funcionalidad..... | 68 |
| Seguridad..... | 69 |
| Ventajas..... | 70 |



| | |
|--|----|
| Desventajas..... | 71 |
| PostGreSQL..... | 71 |
| Ventajas..... | 72 |
| Desventajas..... | 73 |
| MySQL..... | 73 |
| Ventajas..... | 74 |
| Desventajas..... | 74 |
| Firebird..... | 75 |
| Ventajas..... | 76 |
| Desventajas..... | 76 |
| 2.3.2. Características, Ventajas y Desventajas de Apache (tomcat), php y javascript..... | 79 |
| Apache..... | 79 |
| Apache Tomcat..... | 79 |
| Características..... | 80 |
| Ventajas..... | 80 |
| Desventajas..... | 81 |
| PHP..... | 81 |
| Características..... | 83 |
| Ventajas..... | 84 |
| Desventajas..... | 85 |
| Java Scrip..... | 85 |
| Características..... | 87 |
| Ventajas..... | 87 |
| Desventajas..... | 87 |
| 2.4. Arquitectura Cliente/servidor..... | 88 |
| Definición del Modelo Cliente/Servidor..... | 89 |
| Componentes del modelo Cliente/Servidor..... | 91 |
| Clasificación de modelos Cliente/Servidor..... | 94 |



| | |
|--|------------|
| Clasificación por tamaño de componentes..... | 94 |
| Fat Client (Thin Server)..... | 95 |
| Fat Server (Thin Client)..... | 95 |
| Clasificación por planos o capas (Tier)..... | 95 |
| Planos a niveles de software..... | 96 |
| Cliente/Servidor Dos Capas o Planos..... | 96 |
| 1. Implementado con SQL Remoto..... | 97 |
| Ventajas..... | 98 |
| Desventajas..... | 98 |
| 2. Implementado con Procedimientos Almacenados..... | 98 |
| Ventajas..... | 99 |
| Desventajas..... | 99 |
| Cliente/Servidor de Tres Planos..... | 100 |
| Ventajas..... | 101 |
| Desventajas..... | 101 |
| Clasificación por la naturaleza del servicio..... | 102 |
| Servidores de Bases de Datos..... | 103 |
| Procedimientos Almacenados..... | 103 |
| Servidores de transacciones..... | 104 |
| Consideraciones para herramientas de desarrollo cliente/servidor..... | 105 |
| Capítulo 3..... | 107 |
| 3. Definición y Análisis del Problema..... | 109 |
| 3.1. Problemática Actual..... | 109 |
| 3.2. Requerimientos Generales y Particulares del Sistema..... | 114 |
| Requerimientos Generales..... | 117 |
| Requerimientos Particulares..... | 118 |
| 3.3. Clasificación y Organización de la Información a Presentar en el Sitio Web..... | 118 |



| | |
|---|------------|
| Segmentar la información..... | 119 |
| Importancia de la Jerarquía..... | 120 |
| Relaciones..... | 121 |
| Función..... | 121 |
| Estructura del Sitio..... | 124 |
| Estructura de Información Básica..... | 124 |
| Archivos de Sitio y Estructuras de Directorio..... | 126 |
| 3.4. Análisis Jerárquico del Sitio a Desarrollar..... | 129 |
| Ventajas AHP..... | 129 |
| Estructura del Modelo Jerárquico..... | 129 |
| Árbol de Jerarquías..... | 131 |
| 3.5. Opciones de Solución del Front-End y Back-End..... | 136 |
| Editores HTML..... | 137 |
| Mozilla Firefox..... | 140 |
| Internet Explorer 7..... | 141 |
| Opera..... | 142 |
| Uso Actual de los Navegadores..... | 142 |
| Páginas Estáticas y Dinámicas..... | 143 |
| Páginas Activas..... | 145 |
| Decisión de Solución Front-End..... | 145 |
| Decisión de Solución Back-End..... | 147 |
| Capítulo 4..... | 149 |
| 4. Diseño, Programación e Implantación..... | 151 |
| 4.1. Aplicación de la Metodología..... | 151 |
| 4.2. Diseño del Portal..... | 154 |
| Peso de las Páginas..... | 155 |
| Esquemas o Estilos de las Páginas..... | 155 |
| Uso de Presentaciones en Flash..... | 157 |



| | |
|---|-----|
| Uso de Macros o Frames..... | 157 |
| Uso de Imágenes de Background..... | 159 |
| Uso de Meta Tags Adecuados..... | 159 |
| Normas para incorporar Elementos Gráficos y Multimedia..... | 160 |
| Estándares Internacionales..... | 162 |
| 4.2.1. Diseño General del Portal..... | 164 |
| 4.2.2. Diseño de la Página Web Estática y Dinámica..... | 164 |
| 4.3. Diseño de la Base de Datos..... | 172 |
| Diagrama de Procesos Nivel 0: Diagrama de Contexto..... | 172 |
| Diagramas de Flujo de Procesos..... | 173 |
| Diagrama de Procesos Nivel 1: Portal de la Academia Regional de Informática de la DGETI..... | 173 |
| Diagrama de Procesos Nivel 2: Consultas..... | 174 |
| Diagrama de Procesos Nivel 2: Gestión..... | 175 |
| Diagrama de Procesos Nivel 3: Gestión..... | 176 |
| Alta..... | 176 |
| Baja..... | 178 |
| Actualiza..... | 179 |
| Consulta..... | 180 |
| Diagrama Jerárquico de Procesos..... | 181 |
| 4.3.1. Diagrama de Flujo de Datos Generales..... | 182 |
| Validación de Usuario y Opciones de Administración..... | 183 |
| Diagrama de Flujo Actualizar Gestor..... | 184 |
| Diagrama de Flujo Actualizar Administrador de Usuarios..... | 186 |
| 4.3.2. Diccionario de Datos..... | 187 |
| 4.3.3. Diagrama de Entidad-Relación..... | 191 |
| 4.4. Integración de Pruebas..... | 193 |
| Pruebas al Portal de la Academia Regional de Informática..... | 197 |
| Prueba de Acceso..... | 197 |



Índice.

| | |
|--|------------|
| Prueba de Consultas, Altas, Bajas y Cambios en las tablas..... | 200 |
| Prueba de validación de Fecha, Hora, caracteres incorrectos, vacíos, no selección de registro a actualizar..... | 205 |
| 4.5. Implantación y Liberación..... | 209 |
| Criterios para la Selección de Equipo..... | 209 |
| Criterio Alterno..... | 209 |
| Implantación..... | 210 |
| Capacitación..... | 210 |
| Plan de Capacitación..... | 211 |
| Diagrama de Gant..... | 212 |
| Liberación..... | 212 |
| Contrato de Sitios Web..... | 213 |
| 4.6. Tipos de Mantenimiento..... | 220 |
| Mantenimiento Preventivo..... | 220 |
| Mantenimiento Correctivo..... | 220 |
| Mantenimiento Proactivo..... | 220 |
| Conclusiones..... | 221 |
| Bibliografía..... | 227 |
| Glosario..... | 235 |
| Manual Técnico..... | 241 |
| Manual de Usuario..... | 279 |



Portal de la Academia Regional de Informática de la DGETI.





CAPÍTULO I



**Portal de la Academia Regional de Informática de la
DGETI.**





1. Entorno del Problema.

1.1. Introducción.

Cada vez es mayor el volumen de actividades que se desarrollan directamente en la red. Los esquemas formales de actualización en el área de cómputo han generado una actividad próspera y eficaz al realizar trámites o envíos de información, de tal manera que las diferentes entidades regionales puedan estar actualizadas en el contexto académico.

La importancia radica en las diferentes funciones que pueden desarrollarse dentro del portal para generar el interés de los docentes. Los centros de formación imparten sus enseñanzas y se relacionan con los estudiantes. En este sistema educativo, generado de un entorno social con necesidades de transferencia de información actualizada y oportuna la Internet juega un papel trascendental, ya que los sistemas informáticos son una fuente de consulta y actualización para los docentes.

Si bien se sabe de la complejidad educativa, también es justo mencionar la riqueza que emanan de las instituciones, como son: los modelos, los docentes, los alumnos, los avances tecnológicos en los que esta referida esta tesis.

La Academia Regional de Informática, no cuenta a la fecha con una página web que permita informar de los cursos, conferencias, exposiciones, juntas o cualquier otra actividad, este es el motivo que nos lleva proponer el desarrollo de un portal para la academia.



Portal de la Academia Regional de Informática de la DGETI.



La información que emana de cualquiera de las actividades dentro de La Academia Regional de Informática, es requisito mínimo e indispensable para que los docentes estén informados de los acontecimientos actuales y futuros, que se realizarán en cada uno de los CETIS (Centro de Estudios Tecnológicos Industriales y de Servicio), del Distrito Federal. Igualmente se podrá contar con los calendarios de dichas actividades y acceder a material de apoyo, para un mejor desarrollo de sus actividades en el aula.

Resumen.

La tesis se divide en cuatro capítulos con la finalidad de tener una visión más amplia de la problemática de estudio. Que en este caso es la creación de una página para la Academia Regional de Informática.

Primer Capítulo.- En este capítulo se conocerá la actividad, estructura y función principal de la Institución que es la DGETI (Dirección General de Educación Técnica Industrial). Siendo el órgano principal que rige a la Academia de Informática.

Segundo Capítulo.- Se plantea el marco teórico como eje rector de toda tesis su sustento teórico, las tendencias de las nuevas tecnologías para el desarrollo de la página, las bases de datos que resguardarán la información, así como las ventajas y desventajas del software a utilizar analizando su entorno de programación.

Tercer Capítulo.- Se desarrolla un análisis de la problemática, visualizando los requerimientos del usuario para que la página sea factible, eficiente y operable. Es importante que se cumplan con las jerarquías de organización de la página para que el diseño sea eficiente.



Cuarto Capítulo.- Este último capítulo se desarrolla por medio de la programación de la página web aplicando un diseño para el portal; teniendo como meta principal cubrir los requerimientos del usuario como: Modernizar las actividades que realiza la academia. Involucrar al personal dando a conocer las actividades que se desarrollan durante el ciclo escolar. Seleccionar el mejor material didáctico recopilado a través de acuerdos de academia. Información oportuna de las actividades de la academia. Brindar material útil y actual al área de informática, de esta manera se podrá tener la información disponible, para aquel Docente que la requiera.

1.2 Rubro de la Empresa.

La DGETI (Dirección General de Educación Técnica Industrial), es una dependencia adscrita a la Subsecretaría de Educación Media Superior (SEMS), dependiente de la Secretaría de Educación Pública, que ofrece servicio educativo de nivel medio superior tecnológico.

Cuenta con una infraestructura física de 6 mil 231 aulas, 1 mil 533 laboratorios, 1 mil 623 talleres, 311 salas audiovisuales, 317 bibliotecas y 620 áreas deportivas en 429 planteles educativos a nivel nacional, de los cuales 168 son CETIS(Centro de Estudios Tecnológicos Industriales y de Servicio) y 261 CBTIS(Centro de Bachillerato Tecnológico Industrial y de Servicio); ha promovido además la creación de 288 CECyTEs,(Colegio de Estudios Científicos y Tecnológicos), mismos que operan bajo un sistema descentralizado.



Con esta infraestructura se atendió durante el 2001 a una población escolar de 520 mil 678 alumnos, de los cuales 466 mil 110 corresponden al Bachillerato Tecnológico (B.T.), 21 mil 902 al Técnico Profesional y Básico (T.P. y T.B.) y 32 mil 666 en el Sistema Abierto de Educación Tecnológica Industrial (SAETI); asimismo, a través de los planteles descentralizados (CECYTEs) se atendieron 123 mil 151 alumnos. En forma paralela y con la participación de la sociedad civil se atendieron 70 mil 239 alumnos en 789 planteles incorporados

Planificar un servicio educativo de esta magnitud, implica una importante responsabilidad, por el comportamiento heterogéneo regional que presenta y por la gran cantidad de jóvenes alumnos que atiende.

En el marco del Programa para la Modernización Técnica y la Capacitación (PMETyC), se atendió en el 2001 a 12 mil 489 alumnos en el modelo de Educación Basada en Normas de Competencia Laboral (EBNCL) en 47 planteles de 30 Entidades Federativas.¹

Misión.

“Formar personas con conocimientos tecnológicos en las áreas industrial, comercial y de servicios, a través de la preparación de profesionales técnicos y bachilleres, con el fin de contribuir al desarrollo sustentable del país.”¹

Visión.

¹ Datos consultados en: www.dgeti.sep.gob.mx. Marzo 2008.



Capítulo I. Entorno del Problema



“Ser una institución de educación media superior certificada, orientada al aprendizaje y desarrollo de conocimientos tecnológicos y humanísticos.”¹



Dirección General.

La DGETI ha alcanzado niveles de modernidad educativa que permiten a los alumnos estar en contacto con los avances tecnológicos contemporáneos en las diferentes áreas del conocimiento, incorporando la informática como un auxiliar cotidiano en el proceso de enseñanza-aprendizaje.

La DGETI, como institución se estructura de las direcciones siguientes:

- Dirección General.
- Dirección Técnica.
- Dirección de Apoyo a la Operación Estatal.
- Coordinación de Colegios de Estudios Científicos y Tecnológicos.
- Coordinación Administrativa.

Todas ellas nos permiten desarrollar y cumplir con los requerimientos de la educación a nivel bachillerato tecnológico.

La figura 1.2.1 Muestra un organigrama general de la Institución.

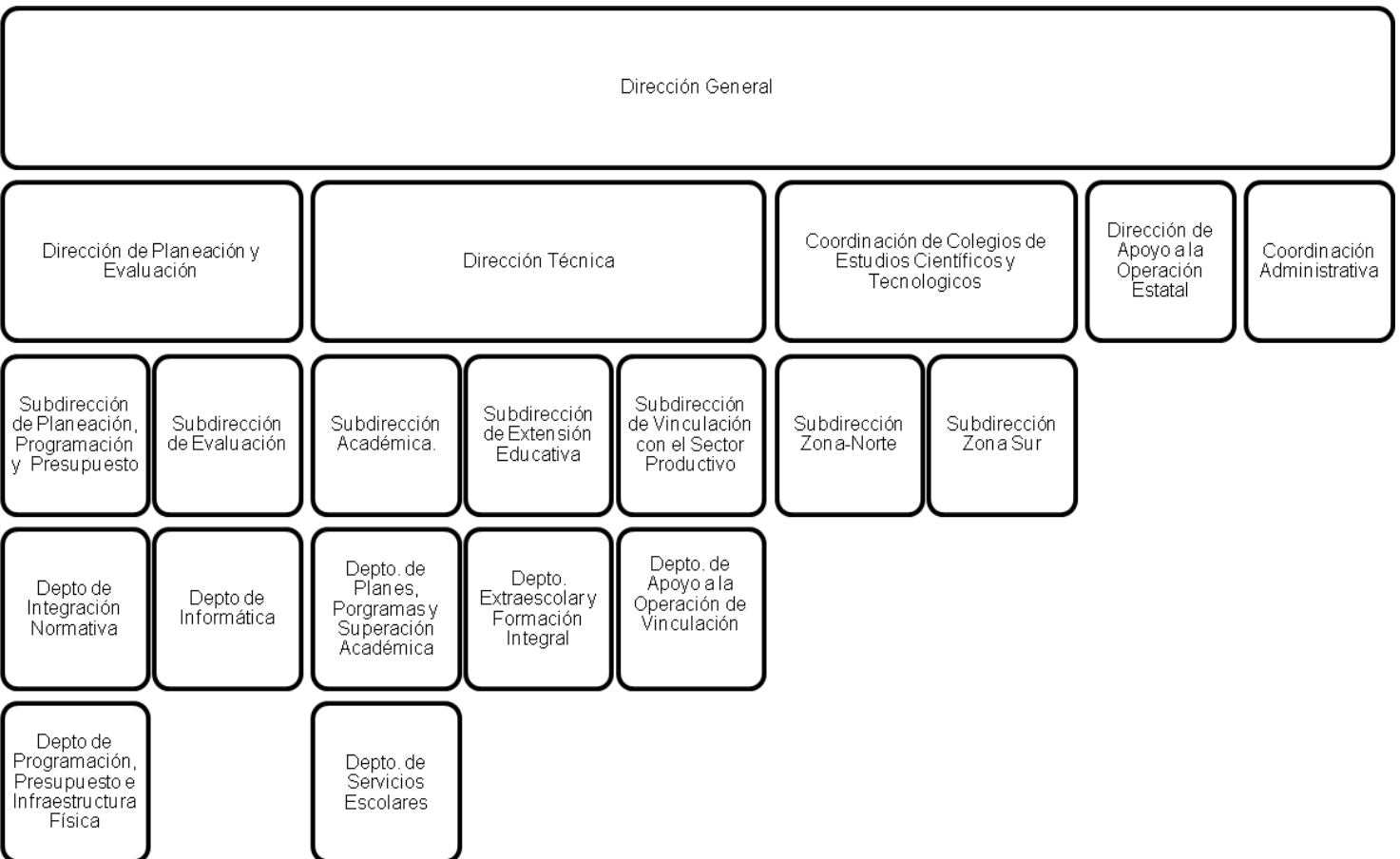


Figura 1.2.1 Organigrama de la Institución. Organigrama de la DGETI. www.dgeti.sep.gob.mx



La Dirección General cuenta con las siguientes funciones:

- Planear, controlar, organizar y evaluar el desempeño de las labores de la dirección general.
- Proponer normas pedagógicas, contenidos, planes y programas de estudio, materiales didácticos e instrumentos para la evaluación del aprendizaje para el nivel medio superior y de servicios.
- Verificar las normas pedagógicas, contenidos, planes y programas de estudio, métodos materiales didácticos para el aprendizaje, aprobados para la educación tecnológica industrial de servicios.
- Formular disposiciones técnicas y administrativas para la organización operación, desarrollo, supervisión y evaluación del proceso de enseñanza-aprendizaje realizada en los planteles de educación tecnológica industrial y de servicios, y verificar su cumplimiento.
- Promover y fomentar las coordinaciones de educación tecnológica industrial, los servicios que ofrece la dirección general.
- Diseñar los lineamientos aprobados, programas de la superación académica y profesional del personal directivo y docente de los planteles de la Secretaría.
- Coordinar las acciones de la dirección general a través de las coordinaciones estatales con los gobiernos de los estados para los servicios de educación tecnológica industrial a nivel medio superior.



Dirección de Planeación y Evaluación.

En el país, la Educación Media Superior Tecnológica ha contribuido decisivamente a la construcción de una sociedad ya que es estratégica por un lado debido a la posición que ocupa dentro del sistema educativo nacional y por otro porque proporciona formación académica en dos modalidades una de carácter propedéutica y otra bivalente.

Responder a retos y oportunidades de esta magnitud implica un gran esfuerzo y una gran responsabilidad por el comportamiento heterogéneo regional del país y por la gran población estudiantil que atiende la Dirección General de Educación Tecnológica Industrial, es por ello que a través de la Dirección de Planeación y Evaluación implanto desde el año 1992 el Sistema Integral de Planeación, el cual considera los principios fundamentales de la planeación estratégica, teniendo carácter participativo, integral, interactivo, obligatorio y perfectible, faculta la posesión de un sistema que capta, encauza, y en general a través de la evaluación pondera la información veraz, oportuna y suficiente de los aspectos cuantitativos y cualitativos del servicio educativo que suministra la DGETI en la búsqueda de la mejora continua tal y como lo establece el Programa Nacional de Educación – PRONAE.

Dirección Técnica.

Esta área se encarga del análisis de los planes y programas que rige a la DGETI, así como los métodos, normas pedagógicas siendo la vía de comunicación entre los planteles y la DGETI además de las funciones siguientes:



- Elaborar el programa operativo anual y el anteproyecto del presupuesto de la Dirección Técnica, con base en los lineamientos establecidos y presentarlos a la Dirección General para lo conducente.
- Planear, dirigir y evaluar la formulación de los contenidos, planes y programas de estudio, métodos y normas pedagógicas para la educación tecnológica industrial y de servicios.
- Determinar los lineamientos para el diseño y la elaboración de los materiales de comunicación educativa, que satisfagan las necesidades de las diferentes modalidades educativas que ofrece la dirección.
- Establecer las normas y lineamientos para la planeación, organización, operación de las reuniones académicas locales y regionales.
- Promover y fomentar la superación académica y actualización profesional del personal docente adscrito al Sistema de Educación Tecnológica Industrial y de Servicios.
- Dirigir y evaluar los procesos de control escolar del Sistema de Educación Tecnológica Industrial y de Servicios, la integración de los programas de servicio social, titulación, orientación educativa, bibliotecas e higiene y emergencias escolares del Sistema de Educación Tecnológica Industrial y de Servicios.
- Proponer, establecer y evaluar las normas y lineamientos a que se sujetarán las actividades de vinculación del Sistema de Educación Tecnológica Industrial y de Servicios con los sectores productivos de bienes y servicios: social, gubernamental y privado. Fomentar el intercambio académico y de desarrollo científico y tecnológico de los planteles del sistema con organismos públicos y privados.



- Participar con las instancias correspondientes de la Dirección General y con los organismos públicos y privados en la realización de estudios para la apertura, modificación o liquidación de carreras y especialidades que ofrece el Sistema de Educación Tecnológica Industrial y de Servicios.
- Mantener comunicación y coordinación con las demás áreas de la Dirección General para el desempeño de la Dirección. Mantener informada a la Dirección General, acerca del desarrollo de las funciones del área.

Dirección de Apoyo a la Operación Estatal.

La Dirección de Apoyo a la Operación Estatal sirve como un puente entre el Órgano Central y las Coordinaciones de Enlace Operativo; dándole fuerza a las funciones de apoyo y supervisión de los centros de trabajo en toda la República, facilitando el desarrollo de estrategias para brindar un servicio de calidad, conformado por la oportunidad, eficiencia y eficacia que el modelo institucional demanda. Sus funciones son:

- Desarrollo de programas de Apoyo a la Operación Estatal del Sistema de Educación Tecnológica Industrial y de Servicios.
- Difusión de la normatividad, lineamientos, procedimientos y criterios establecidos por las autoridades competentes, para la operación de las coordinaciones y planteles del Sistema de Educación Tecnológica Industrial y de Servicios.



- Dirigir y coordinar el desarrollo de los cursos de capacitación y actualización profesional regional y nacional del personal directivo de las coordinaciones y planteles del Sistema de Educación Tecnológica Industrial y de Servicios.
- Verificar la aplicación de las estructuras organizacionales y ocupacionales autorizadas para la operación de las coordinaciones de educación tecnológica industrial y establecer las medidas que procedan a la corrección de las desviaciones que se encuentren.
- Asesorar a las coordinaciones de educación tecnológica industrial en asuntos de carácter laboral, sindical y estudiantil, cuando así lo requieran.
- Proponer a la Dirección General los movimientos de cambios de adscripción y de techos financieros del personal del sistema y en su caso turnarlos a la Coordinación Administrativa para lo conducente.

Coordinación de Organismos Descentralizados Estatales de CECyTEs.

En 1991 la creación de los Colegios de Estudios Científicos y Tecnológicos (CECyTEs), se sustentó en el Programa para la Modernización Educativa 1989-1994, que estableció dentro de sus acciones principales, que el incremento adicional de la demanda estudiantil se atendería con nuevos subsistemas escolares descentralizados de educación bivalente y terminal que propiciarán una participación más efectiva de los Gobiernos Estatales y favorezcan una mejor vinculación regional con el sector productivo.



Capítulo I. Entorno del Problema



El ordenamiento jurídico en el que se sustentan estos Colegios es el Convenio de Coordinación, suscrito entre el Ejecutivo Federal y los Gobiernos de los Estados.

En los Convenios se establecen las bases para impulsar y consolidar los programas de educación media superior tecnológica en las entidades federativas, también se determinan las condiciones para la creación, operación y financiamiento de los colegios como organismos públicos descentralizados de los estados, con personalidad jurídica y patrimonio propio. El primer Convenio de Coordinación se firmó en el Estado de Sonora el 1° de marzo de 1991.

La Dirección General de Educación Tecnológica Industrial, como unidad administrativa responsable de coordinar las acciones encaminadas a atender los acuerdos señalados en los convenios, crea para realizar dichas funciones, la Coordinación de Organismos Descentralizados Estatales de CECyTEs, que tiene como principales funciones:

- Representar a la SEP ante los Órganos de Gobierno de los CECyTEs
- Supervisar la correcta aplicación de los convenios de Coordinación
- Proporcionar asesoría técnico-pedagógica a los CECyTEs
- Coordinar acciones orientadas a la planeación gestión, evaluación y seguimiento del quehacer institucional de los CECyTEs, conforme a las normas, políticas, lineamientos y procedimientos establecidos en el marco de los Convenios de Coordinación.



Coordinación Administrativa.

En su carácter de área adjetiva la Coordinación Administrativa de la Dirección General de Educación Tecnológica Industrial (DGETI), tiene como funciones:

- Establecer los lineamientos, parámetros y mecanismos eficientes que garanticen la administración adecuada y óptima de los recursos humanos, materiales, financieros y tecnológicos.
- La prestación de servicios administrativos, generales, técnicos y complementarios para el despacho de los asuntos competencia de la DGETI.
- Dirigir los procesos de remuneraciones al personal, pago a proveedores, ejercicio presupuestal, adquisiciones menores y suministrar el material de oficina para la operación de la DGETI.
- Participar como representante en los casos que generen procesos jurídicos y que impliquen actos de administración siguiendo siempre los lineamientos establecidos por la SEP y la Dirección General.
- Elevar la funcionalidad de las áreas administrativas de la Coordinación, desarrollando y proponiendo técnicas administrativas acordes con las necesidades de innovación, calidad, modernización, simplificación y transparencia en los procesos que impliquen actos de administración.
- Proporcionar asesoría sobre los trámites y servicios que oferta la DGETI.
- Operar en función directa con los lineamientos que, en sus respectivas materias, dictan los órganos rectores de la Administración Pública Federal y las Direcciones Generales de la SEP.



1.3 Análisis del Entorno Académico de Informática.

El manejo y aprovechamiento de la información se ha visto favorecido con la revolución tecnológica de los últimos años.

Con el desarrollo de la Informática surgida propiamente de la convergencia de la computación, la microelectrónica y las telecomunicaciones, aunado al manejo de grandes volúmenes de información mediante el uso de estas tecnologías es más rápido, sencillo y confiable, ya que esta puede ser consultada y transmitida en diferentes medios electrónicos o almacenada en diferentes dispositivos.

Con el apoyo de la tecnología informática las instituciones públicas y privadas están en posibilidades de mejorar sustancialmente sus mecanismos tradicionales de gestión y de servicios, incrementando la eficiencia y calidad en sus productos y/o servicios.

El sector educativo en México representa un factor de vital importancia para el desarrollo de estas tecnologías en un futuro, por lo cual es indispensable proveer de tecnología de punta, para el estudio y desarrollo de sus actividades educativas, así mismo se requiere contar con redes de datos que satisfagan las necesidades de intercambio de información de manera que incremente la calidad de la educación en nuestro país.

Ante tal reto informático la DGETI y considerando la responsabilidad que implica el ser la institución educativa de nivel medio superior con mayor cobertura nacional, tiene a bien elaborar un plan de desarrollo tecnológico interno, considerando el parque informático con el que cuenta actualmente, los avances tecnológicos, el desarrollo constante de aplicaciones acordes a las nuevas tecnologías y anteponiendo las necesidades nacionales de intercambio de información, así como la satisfacción de los



Portal de la Academia Regional de Informática de la DGETI.



requerimientos de informática que permitan enriquecer la calidad en la educación en el nivel medio superior, impartida en los planteles que conforman este subsistema.

Por todos los aspectos anteriores, se hace necesario contar con una área a nivel nacional que regule y coordine el crecimiento tecnológico en los planteles y unidades administrativas de la DGETI, estableciendo los lineamientos necesarios para una adecuada administración y uso del parque informático, un crecimiento tecnológico acorde a los objetivos institucionales de este subsistema, así mismo impulsar el desarrollo de aplicaciones que contribuyan a la mejora substancial de los mecanismos tradicionales de gestión de trámites administrativos centralizados y de servicios en línea, obteniendo beneficios reales y tangibles en las actividades de apoyo a la educación, así como en el cumplimiento satisfactorio de los planes y programas de estudio de la población escolar de la DGETI.

Redes y Telecomunicaciones.

La interconexión de redes para el departamento de informática de la DGETI es un aspecto de vital relevancia, por tal motivo es importante contar con una infraestructura de redes de datos robusta que permita satisfacer las actuales y futuras necesidades de este subsistema, preparándolo para los avances tecnológico, así como enriquecer los recursos informáticos con que cuenta actualmente, a fin de garantizar una mayor calidad en la educación que ofrece este subsistema, así como la automatización de los procesos de trámites administrativos entre los diferentes planteles y unidades administrativas.



Objetivo.

Promover en todos los planteles y unidades administrativas que conforman la DGETI, el desarrollo eficaz de la infraestructura de redes de computadoras para el acceso y transmisión de datos.

Estrategias.

- Proveer a los planteles y unidades administrativas que lo soliciten, los recursos informáticos necesarios para garantizar la interconexión e interoperabilidad con las redes informáticas actuales con las que está actualmente interconectada la DGETI.
- Establecer mecanismos normativos para la conexión, instalación, administración y uso de las redes y equipos de telecomunicaciones.
- Garantizar el desarrollo de la red académica.
- Evaluar la tecnología de informática y telecomunicaciones con que se cuenta y sugerir cambios.
- Calendariza en conjunto con los planteles y/o unidades administrativas programas para el mantenimiento y/o ampliación de las redes.



Internet e Intranet.

Con base en los recursos informáticos y de telecomunicaciones con que cuenta la DGETI, es importante fomentar la investigación y el desarrollo tecnológico en todos los planteles y unidades administrativas de la misma, mediante la creación, expansión y mantenimiento de la pagina Web de la DGETI, así como el crecimiento de la Red-académica, por lo cual el departamento de Informática mediante el área de Internet e Intranet realiza estas funciones.

Objetivo.

Adoptar, asimilar e Innovar, los recursos informáticos con que cuenta la DGETI, contribuyendo a fomentar el desarrollo educativo y el aprovechamiento de los recursos tecnológicos agilizando los procesos administrativos en los planteles y unidades administrativas que forman la DGETI.

Estrategias.

- Impulsar la amplia disponibilidad de la información relacionada con los planteles y unidades administrativas de la DGETI, a través del uso de medios digitales.
- Establecer criterios que definan los lineamientos necesarios para las publicaciones de información relacionada con los planteles y unidades administrativas de la DGETI.
- Crear una infraestructura de información vía Internet que beneficie al desarrollo de aplicaciones para el mayor aprovechamiento de los recursos informáticos y de telecomunicaciones, así como a la mejora de las actividades educativas y



Capítulo I. Entorno del Problema



procesos administrativos de los planteles y unidades administrativas de la DGETI.

- Promover la difusión masiva de la DGETI a nivel nacional e internacional mediante el aprovechamiento de los recursos informáticos y de telecomunicaciones.
- Mejorar la Interconexión e interoperabilidad con las redes informáticas con que cuenta la DGETI.
- Consolidar los marcos necesarios que normalicen el acceso y transmisión de datos y el desarrollo de servicios públicos y privados a través de medios electrónicos.
- Apoyar el fortalecimiento y desarrollo de redes académicas que permitan extender a los diferentes planteles y unidades administrativas de la DGETI, los servicios de información elevando el nivel académico y mejorando los procesos administrativos de los mismos.
- Desarrollar la infraestructura informática y de telecomunicaciones a fin de atender las necesidades de intercambio de información entre los planteles y unidades administrativas de la DGETI.
- Contar con una infraestructura de redes de alta velocidad.
- Establecer mecanismos para la coordinación, seguimiento y evaluación de las actividades relacionadas con la adquisición de tecnologías de información.

Es así que la DGETI, busca satisfacer las necesidades de las instituciones que pertenecen al subsistema, para apoyar las carreras de informática.



1.4 Estructura académica.

La estructura de la Academia Regional de Informática para los 34 planteles está integrada de la manera siguiente. Ver Figura 1.4.1.

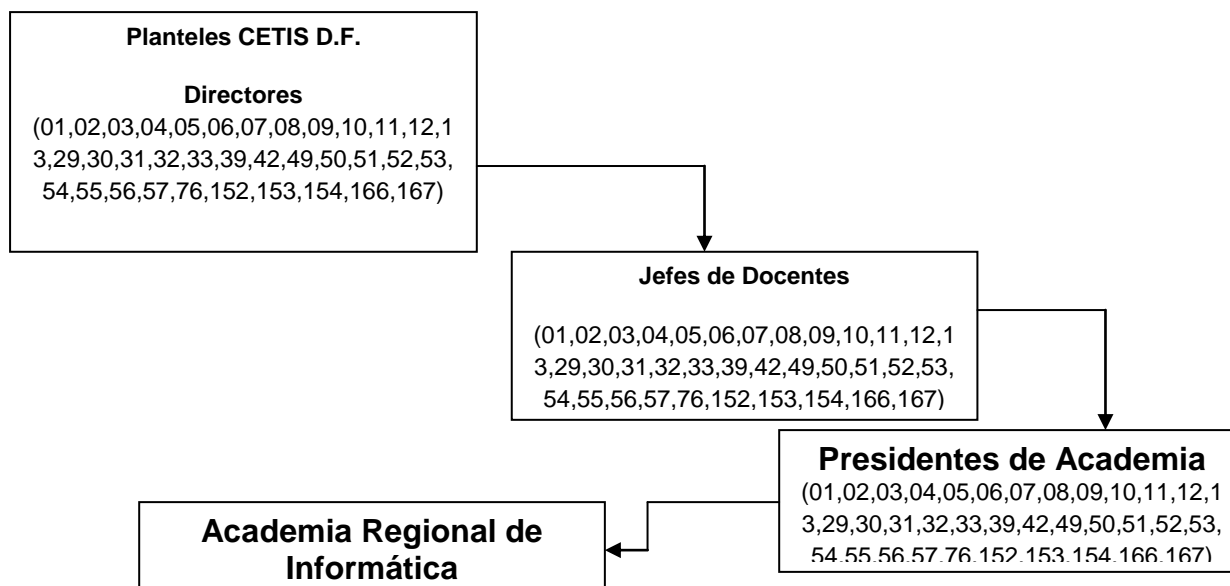


Figura 1.4.1. Estructura Académica.

1.5 Como se distribuye Actualmente la Información.

Para el personal Docente, que deba asistir a:

- Exposiciones
- Elaboración de exámenes
- Reuniones de academia regional



- Cursos intersemestrales
- Reuniones de academia local
- Conferencias
- Simposiums
- Mesas redondas

Todo se maneja a través de:

- Oficios

Que se envía de la manera siguiente vea Figura 1.5.1

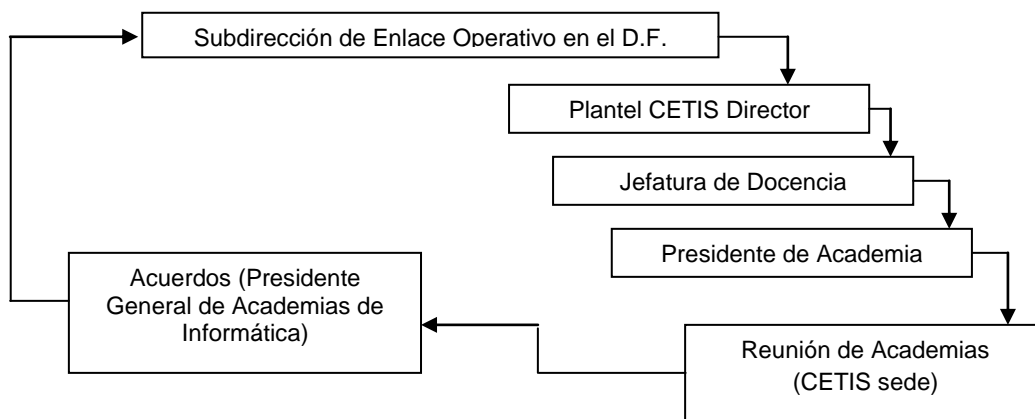


Figura. 1.5.1 Secuencia de Envío.

La información que se tiene en la página de Internet de la DGETI, no contiene lo que la Academia Regional de Informática requiere sabemos que con los oficios nos justifica la inasistencia al plantel de procedencia. A continuación se muestra un ejemplo de Oficio o comisión.



México D.F., 29 de Noviembre del 2007

Asunto: Comisión

Profr.(a)

P r e s e n t e.

En cumplimiento a la disposición oficial emitida por la Subdirección de Enlace Operativo, queda usted comisionada para asistir a la **“Tercera Reunión de Academia de la especialidad de Informática”**. La actividad tendrá lugar en las instalaciones del Centro de Estudios Tecnológicos industrial y de servicio N° 6, ubicado en Av. Cuitláhuac N° 50 esq. Tláhuac Colonia Los Reyes Iztapalapa, Delegación Iztapalapa, el día **11 de Diciembre** del presente año, con un horario de 9:00 a 15:00 horas.

Esperando contar como hasta hoy con su valiosa participación, aprovecho la ocasión para enviarle un cordial saludo.

Nombre del Director

C.c.p.-

Una de las formas en que se da a conocer la información en la institución con la desventaja de que en ocasiones cuenta con poco tiempo de anticipación tendiendo a que en ocasiones las fechas suelen caducar cuando llegan al destino final, generando la baja participación y audiencia de los integrantes en los eventos a llevarse a cabo.



Capítulo I. Entorno del Problema



Un problema primordial es que toda la transacción se realiza de forma manual. La automatización de esta se puede llevar a cabo gestionando los datos, peticiones e incidencias a través de una base de datos basada en software libre por lo que proponer un portal con datos actualizados, gestionados por una base de datos que se actualiza por un periodo semanal o quincenal máximo, evitando estos inconvenientes, es lo óptimo.

Esta página con acceso a datos permite interactuar con la información de una base de datos ya sea para obtener información y mostrarla al usuario o bien para actualizar su contenido. Ayuda a gestionar con facilidad los contenidos del sitio Web y a interactuar con bases de datos.

Al contar con una base de datos, la actualización de las páginas toma menos tiempo ya que la manipulación del código HTML es mínima o nula. Mediante el gestor de base de datos se puede actualizar múltiples páginas y una mayor cantidad de contenido.



CAPÍTULO II



**Portal de la Academia Regional de Informática de la
DGETI.**





2. Marco Teórico.

2.1 Principales tendencias de la WEB y metodologías de desarrollo.

Las redes de computadoras y de comunicación han evolucionado por décadas. Hoy las tecnologías de comunicación e información son justamente un mecanismo de soporte (para el procesamiento de datos y sistemas de información) y en ocasiones hasta un medio para la comercialización.

En la actualidad las grandes empresas suelen usar el correo electrónico como un medio para comunicarse con otras compañías o firmas comerciales; otras organizaciones han adoptado el uso de “páginas web” para crear un enorme negocio basado principalmente en el Internet.

El progreso en varios campos de la tecnología, entre ellos: computadoras, telecomunicaciones, programas informáticos y la información han modificado los estándares cotidianos de vida de una manera difícilmente imaginable dos décadas atrás.

Las tecnologías modernas se emplean de manera combinada, especialmente a través de Internet, para conectar a millones de personas de todos los rincones del mundo. Las comunicaciones cuentan cada vez con menos limitaciones impuestas por la geografía y el tiempo que requería su transmisión.

La información se difunde con mayor amplitud y rapidez. Esta revolución tecnológica alcanzará en creciente medida a todos los campos de actividad en los que la transmisión digital de la información tenga utilidad, sea en la oficina, en el mundo de los negocios, o en el de las compras, el ocio y el esparcimiento.



Una de las principales tecnologías que ha permitido la evolución y el intercambio de paquetes de información entre diferentes redes de computadoras es el TCP/IP (**Transmission Control Protocol**)/(**Internet Protocol**) (Protocolo de Control y transmisión/Protocolo de Internet). El cual ha logrado la estandarización de comunicación entre computadoras.

La **World Wide Web** (Red Global Mundial) es un sistema de archivos de hipertexto enlazados que pueden ser vistos a través de Internet. En el navegador Web los archivos pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navegar a través de ellos usando hiperenlaces. Lo más importante de la Web es su alto nivel de accesibilidad. El desarrollo de estas tecnologías ha logrado que el intercambio de datos crezca a su máximo nivel, simplificándose cada vez más y creando nuevas formas de comunicación.

La naturaleza interactiva de la Web y su entorno hipertexto permiten búsquedas profundas no lineales que son iniciadas y controladas por los clientes, por lo tanto las actividades de mercadeo mediante la Web están más impulsadas por los clientes que aquellas proporcionadas por los medios tradicionales.

Funcionamiento Básico de la Web.

Al realizar el enlace de hipertexto tecleando la dirección **URL/URI** (Localizador Uniforme de Recursos/Identificador Universal de Recursos), el navegador comienza la comunicación transparente para el usuario, para obtener los datos de la página y visualizarla.



Primeramente traduce la parte del nombre del servidor de la URL en una dirección **IP** (Protocolo de Internet) usando la base de datos distribuida de Internet conocida como **DNS** (Sistema de Nombres de Dominio). Esta dirección IP es necesaria para contactar con el servidor web y poder enviarle paquetes de datos. Posteriormente se envía una petición **HTTP** (Protocolo de Transferencia de Hipertexto) al servidor Web solicitando el recurso. En el caso de una página web típica, primero se solicita el texto **HTML** (Lenguaje de Marcado de Hipertexto) y luego es inmediatamente analizado por el navegador, el cual, después, hace peticiones adicionales para los gráficos y otros archivos que forman parte de la página. Luego de que se han recibido los archivos solicitados desde el servidor web, el navegador renderiza la página tal y como se describe en el código HTML, ó con **CSS (Cascading Style Sheet)** (Hojas de estilo en cascada) y otros lenguajes web, los cuales definen la estructura y el contenido de los documentos. Al final se incorporan las imágenes y otros recursos para producir la página que ve el usuario en su pantalla. Como puede observarse en la figura 2.1.1 se describe de manera sencilla los pasos a seguir para realizar una petición Web.

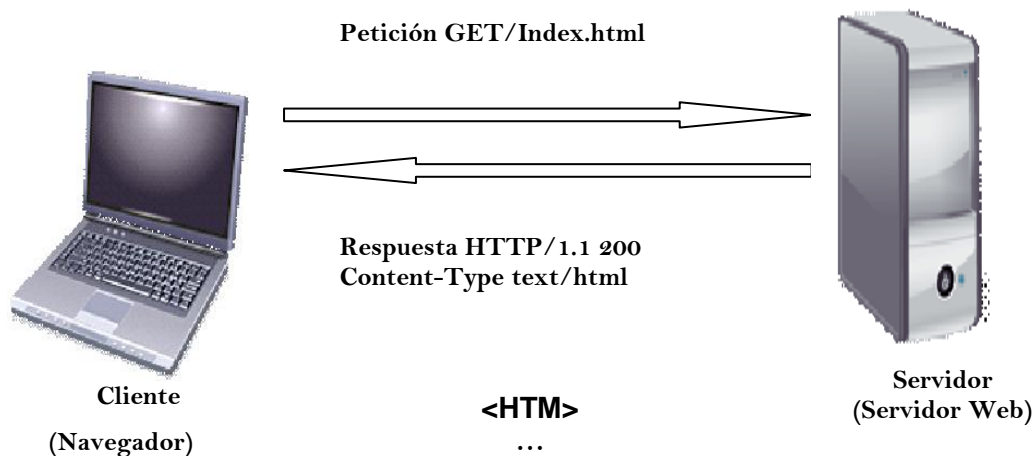


Figura 2.1.1. Funcionamiento Básico de la Web.



Tecnologías de aplicación en el servidor web.

En la actualidad existe el desarrollo de nuevas alternativas que permitan mejorar y subsanar las deficiencias de los programas y herramientas actuales. Por un lado se diseñan sistemas de ejecución de módulos más integrados con el servidor, que evitan que éste tenga que instanciar y ejecutar multitud de programas.

La otra vía consiste en dotar al servidor de un intérprete de algún lenguaje de programación (RXML, PHP, VBScript, etc.) que permita incluir las páginas en el código de manera que el servidor sea quien lo ejecute, reduciendo así el tiempo de respuesta.

La programación basada en componentes es un gran adelanto para la productividad de los desarrolladores web. Las tecnologías heredadas como **DCOM (Modelo de objetos y componentes distribuidos de Microsoft)** es una infraestructura que permite a una aplicación invocar componentes del Modelo de Objetos y componentes instalados en otro servidor los cuales presentan grandes mejoras para estas infraestructuras.

En el desarrollo de un sistema Web es requisito tomar en cuenta las siguientes consideraciones:

- Separar la lógica de la aplicación de la interface de usuario.
- Utilizar métodos estándar de comunicación entre la lógica de aplicación y la interface de usuario.
- Herramientas que permitan una fácil adaptación de las aplicaciones a los nuevos dispositivos que irán apareciendo.
- Definir el costo en comunicaciones que debe asumir la organización.



- Tener en cuenta los procesos de réplica, periodicidad y el ancho de banda que consuman.
- Replantear la idoneidad de la ubicación de cada proceso.
- Extremar las pruebas al diseñar e implementar los protocolos de comunicación.

El World Wide Web Consortium (W3C) Consorcio Internacional de la World Wide Web es un consorcio internacional donde todos los integrantes trabajan conjuntamente para desarrollar estándares Web.

Su misión es llevar a la World Wide Web a su máximo potencial mediante el desarrollo de normas y protocolos que aseguren el crecimiento a largo plazo de la Web.

La parte fundamental de las tecnologías Web es que deben ser compatibles entre sí y permitir que cualquier hardware y software utilizados para acceder a ella trabajen en conjunto.

La figura 2.1.2 presenta una visión general de la infraestructura Web. Soporta búsquedas en cinco áreas:

Temas de la accesibilidad, internacionalización, independencia de dispositivos, acceso móvil y la garantía de calidad de las tecnologías W3C, que se destaca por desarrollar y mantener los estándares tanto anteriores como posteriores, permitiendo almacenar y comunicar de manera eficiente diferentes formas de información (**Web Semántica**).

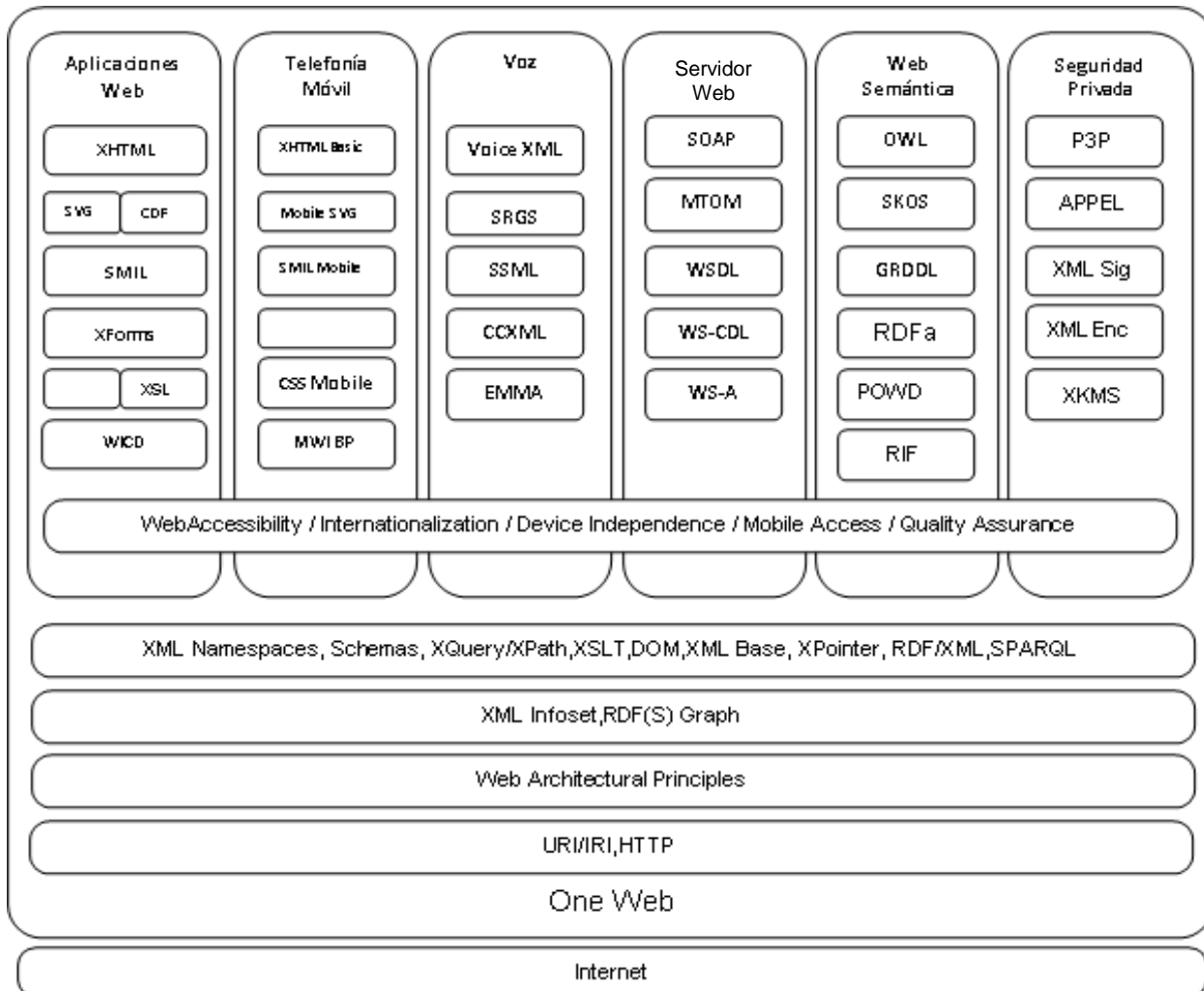


Figura 2.1.2. Visión General de la Infraestructura Web.

Metodologías de Desarrollo de sistemas Web.

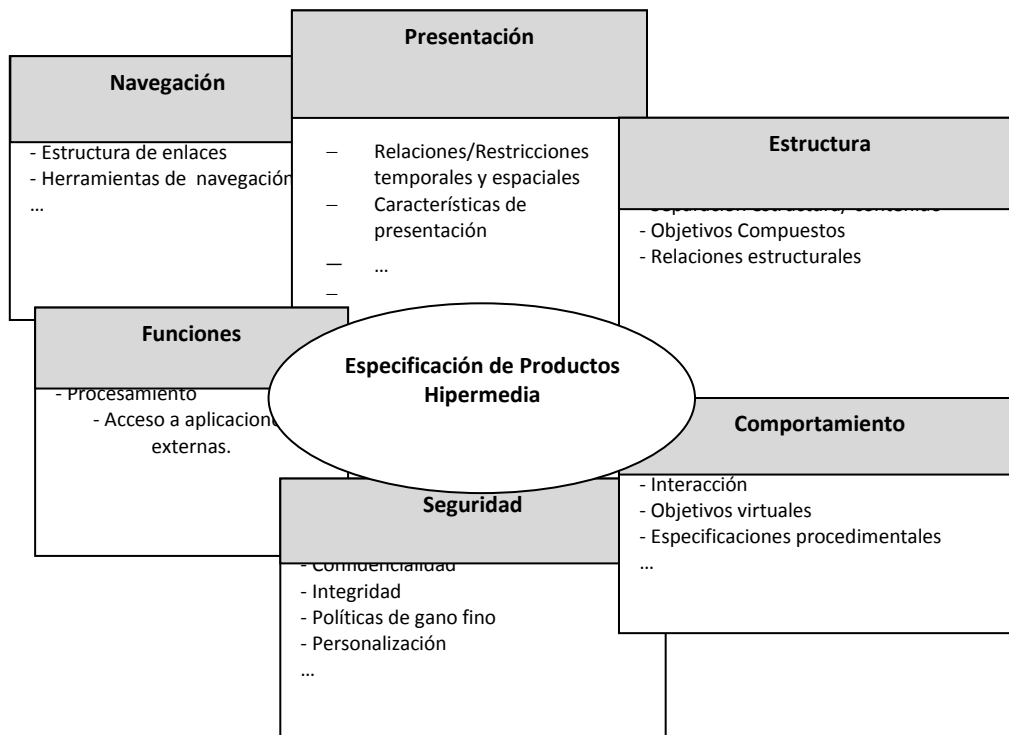
El ciclo de vida de desarrollo de software, se integra por el conjunto de fases entre las que se encuentran el análisis, diseño, implementación, pruebas e instalación; las cuales no siguen una secuencia definida. El modelo de procesos, establece una forma de trabajo concreta en función del paradigma adoptado. El método de desarrollo se encarga de proponer detalladamente qué actividades deben llevarse a cabo durante



cada una de las fases y qué producto o entidades de diseño deben generarse, ofreciendo a su vez principios básicos, guías o consejos para la producción de software con mayor calidad.

Por las características peculiares que presentan los sistemas hipermedia y web requieren de enfoques distintos. Este enfoque se introduce en la fase de evaluación en el ciclo de desarrollo, permitiendo que la construcción del sistema sea iterativa en cuestión de análisis, utilidad y uso para cubrir sus expectativas.

Los requisitos planteados a un método de desarrollo de aplicaciones hipermedia y web al momento del modelado del sistema son seis: navegación, presentación, estructura, comportamiento, acceso y funciones. Estos requisitos están descritos en la figura 2.1.3.



2.1.3. Perspectivas para el desarrollo de sistemas hipermedia.



Los sistemas hipermedia están organizados por medio de nodos relacionados por enlaces navegables que permiten la libre exploración del espacio de información por parte de los usuarios.

Desde el punto de vista de la aplicación, los sistemas web se consideran como un subconjunto de los sistemas hipermedia.

El desarrollo de sistemas hipermedia y web, como todo sistema debe realizarse siguiendo un proceso sistemático y bien definido. El empleo de métodos y técnicas existentes para diseño hipermedia permiten la mejora en calidad, reusabilidad y facilidad de mantenimiento de sistemas ya desarrollados, además de beneficiarse en experiencia y conocimiento de otros diseñadores.

Los modelos de referencia o de diseño y los métodos de desarrollo emplean técnicas, abstracciones y notaciones propias de otros ámbitos, obteniendo un énfasis en el modelo relacional y la orientación a objetos.

Las razones por las que es preciso utilizar métodos y técnicas específicamente desarrollados para esta tecnología, es que se requiere contar con un mecanismo que permitan modelar^{1,1}:

- Sofisticadas estructuras de navegación, las cuales suelen ser efímeras o adaptables a las necesidades y preferencias del usuario.
- Comportamientos interactivos y reacciones ante determinados eventos, ya sean iniciados por el usuario o no. Por ejemplo cuando el usuario selecciona un video

¹ Díaz, P., Aedo, I. y Montero, S. Ariadne, a development method for hypermedia. In proceedings of Dexa 2001, volume 2113 of Lecture Notes in Computer Science, pages 764-774, 2001.



y se reproduce, éste para y vuelve al primer fotograma o cuando dos objetos que se mueven de manera aleatoria por el espacio de visualización chocan, se escucha un sonido de alerta)

- Interfaces con aplicaciones externas, como bases de datos, servicios web u otras aplicaciones hipermedia.
- Composiciones multimedia en las que hay que armonizar contenidos que se presentan en distintas dimensiones, por lo que se da lugar a una presentación de utilidad al mismo tiempo que es estéticamente adecuada a las necesidades y preferencias de sus usuarios.
- Restricciones de acceso que hagan posible indicar cómo distintos tipos de personas van a poder hacer uso del mismo sistema hipermedia de acuerdo con sus necesidades y responsabilidades. Estas restricciones se imponen en función de términos y abstracciones propias del dominio de la hipermedia.

La experiencia en ingeniería de software es útil, principalmente por la disciplina que impone al desarrollo y las capacidades de abstracción y modelado permitido, justificándose la necesidad de contar con métodos que estén fundamentados en elementos y conceptos propios del dominio de la hipermedia, pudiendo funcionar como nodo, enlace o contar con relaciones temporales entre contenidos.

La limitante del tiempo obliga a que los diseñadores más experimentados recurran a emplear soluciones de proyectos anteriores, logrando adaptarlos de tal forma que resuelvan los problemas propios del desarrollo en cuestión.

Los patrones de diseño son de gran ayuda en este tipo de problemáticas y permite dedicar el potencial intelectual de los diseñadores en la búsqueda de nuevas soluciones a problemas novedosos.



Los siguientes métodos para hipermedia y web, ofrecen un marco racional que guía al diseñador desde la concepción inicial del problema hasta una solución lógica y formal para lo cual debe especificar los pasos a seguir y las actividades a realizar para desarrollar la aplicación además de proponer una serie de herramientas o modelos con los que especifica todas las características de la aplicación.

Método ERMM (Relationship Management Methodology).

Es una versión extendida, probablemente, el único método para hipermedia que parece cubrir todo el ciclo de desarrollo, desde el estudio de factibilidad hasta la evaluación del sistema, aunque sólo propone actividades y productos concretos para las fases de análisis y de diseño.

El análisis se desarrolla por medio de un diagrama entidad-relación, no se establecen relaciones reflexivas. Se realiza de manera ascendente a descendente.

Cada una de las unidades se compone partiendo de las entidades del diagrama E-R, generando m-pantallas, quienes especifican los contenidos, enlaces, herramientas de navegación y funciones asociadas a cada unidad.

El diseño ascendente emplea como punto de partida las m-pantallas, generando un nuevo diagrama de aplicación ascendente que contrasta con el descendente.

En la figura 2.1.4 se puede observar las fases que integran el Método ERMM. En el resto de las fases, se emplea el estudio de factibilidad, implementación, pruebas y evaluación.

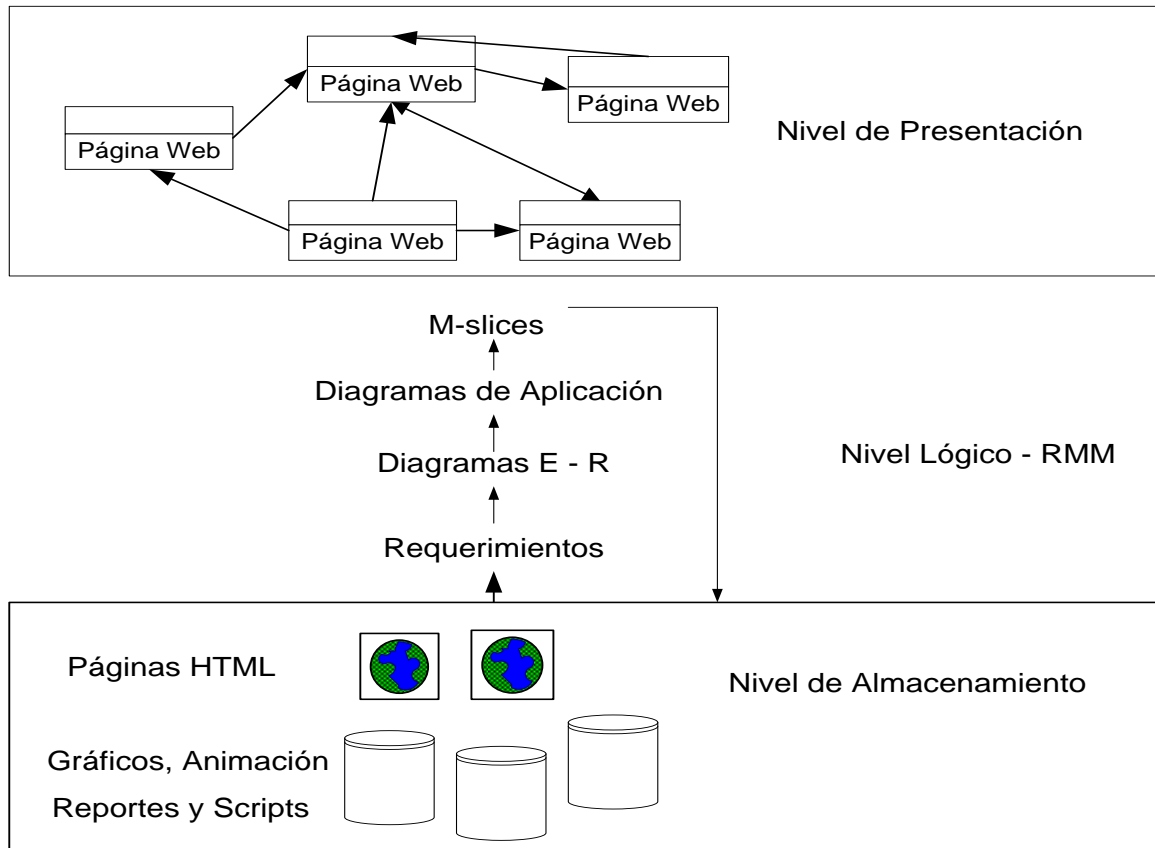


Figura 2.1.4. Fases del Método ERMM

Web Site Design Method (WSDM) (Método de Diseño de un Sitio Web).

Método que se interesa en las necesidades del usuario, más que en los datos, destinado a sistemas web, las características de los sistemas hipermedia, y la sincronización de contenidos, no se toman en cuenta. Este método se orienta principalmente a dar soporte al desarrollo de sistemas personalizados. Para el desarrollo de sistemas web se propone las fases de Modelado de Usuarios, Diseño Conceptual y Diseño de la Implementación.



En la primera fase se analizan y clasifican los tipos de usuarios en clases, atendiendo a los requisitos que estos tienen en cuanto al uso del sistema se refiere. Cada clase de usuarios puede dividirse según las perspectivas que puedan responder a distintos requisitos de usabilidad; entonces el diseño se realiza en función de esas clases y perspectivas. El diseño Conceptual se encarga de modelar los requisitos de información de cada clase de usuarios, por lo que se genera un Modelo de Objetos de Usuario para cada una de ellas junto con un Modelo de Navegación para cada perspectiva de cada clase.

El Diseño de Implementación modela la apariencia de la aplicación por lo que finalmente deberá implementarse haciendo uso de tecnologías propias de los entornos web. En la figura 2.1.5 se muestra las fases integrantes del método WSDM.

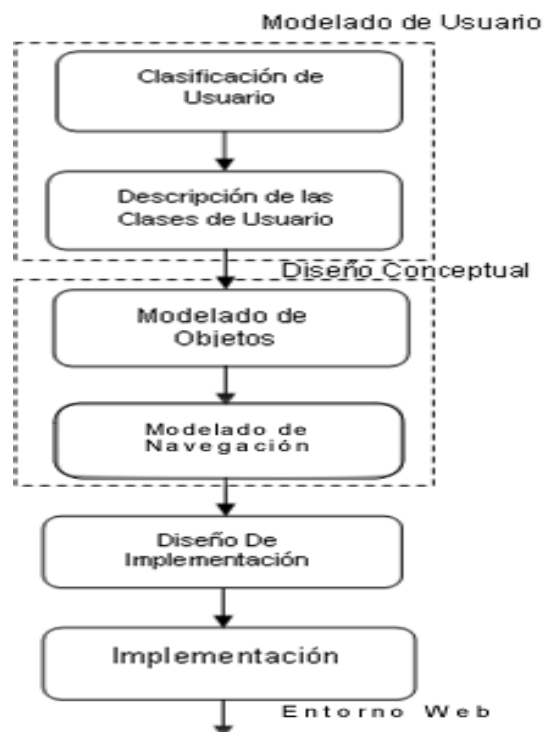


Figura 2.1.5 Fases del Método WSDM.



Web Modeling Language (WebML) (Lenguaje de Modelado Web).

Método orientado a diseñar sitios web desde una perspectiva de alto nivel. Es un diseño iterativo cuya pretensión es guiar al desarrollador desde el proceso de levantamiento de requisitos hasta el diseño personalizado de la aplicación. Cada ciclo de diseño está compuesto de las actividades siguientes: Recolección de Requisitos, Diseño de Datos, Diseño abstracto del Hipertexto (respondiendo al concepto de **in the large** introducido en **HDM**), Diseño detallado del Hipertexto (atendiendo al concepto de **in the small** de **HDM**), Diseño de la presentación, Diseño de los Usuarios y de los Grupos y personalización del Diseño. A este objeto se proponen cuatro herramientas: El Modelo Estructural, el Modelo de Hipertexto, el Modelo de Presentación y el Modelo de Personalización.

El Modelo Estructural representa a los datos que contendrán el sitio web así como las relaciones estructurales que existen entre ellos. En este modelo se incluyen los usuarios y grupos como un tipo de entidad predefinida que puede servir para personalizar el sitio web. El modelo de hipertexto describe que páginas y que contenido integran el sitio, y cómo están enlazadas a través de dos submodelos: el de Composición y Navegación.

El Modelo de Presentación se emplea para especificar la apariencia de las páginas de una manera independiente de las tecnologías de presentación y de la plataforma de implementación, por medio de una sintaxis abstracta basada en XML.

El Modelo de Personalización está basado en el esquema estructural en el que se han definido los tipos de usuario y al que se pueden añadir reglas utilizando una especie de lenguaje **OQL (Object Query Language)** (Lenguaje de consultas de objetos) que indicarán como derivar el contenido atendiendo a ciertas características del usuario.



En la figura, 2.1.6 se muestra el enfoque WebML en el desarrollo de aplicaciones web la cual consta de diferentes fases que se deben aplicar en un iterativo e incremental.

El proceso se somete a varios ciclos, cada ciclo de producción de un prototipo o una versión parcial de la aplicación, permite la realización de las pruebas y la evaluación desde las primeras fases de desarrollo.

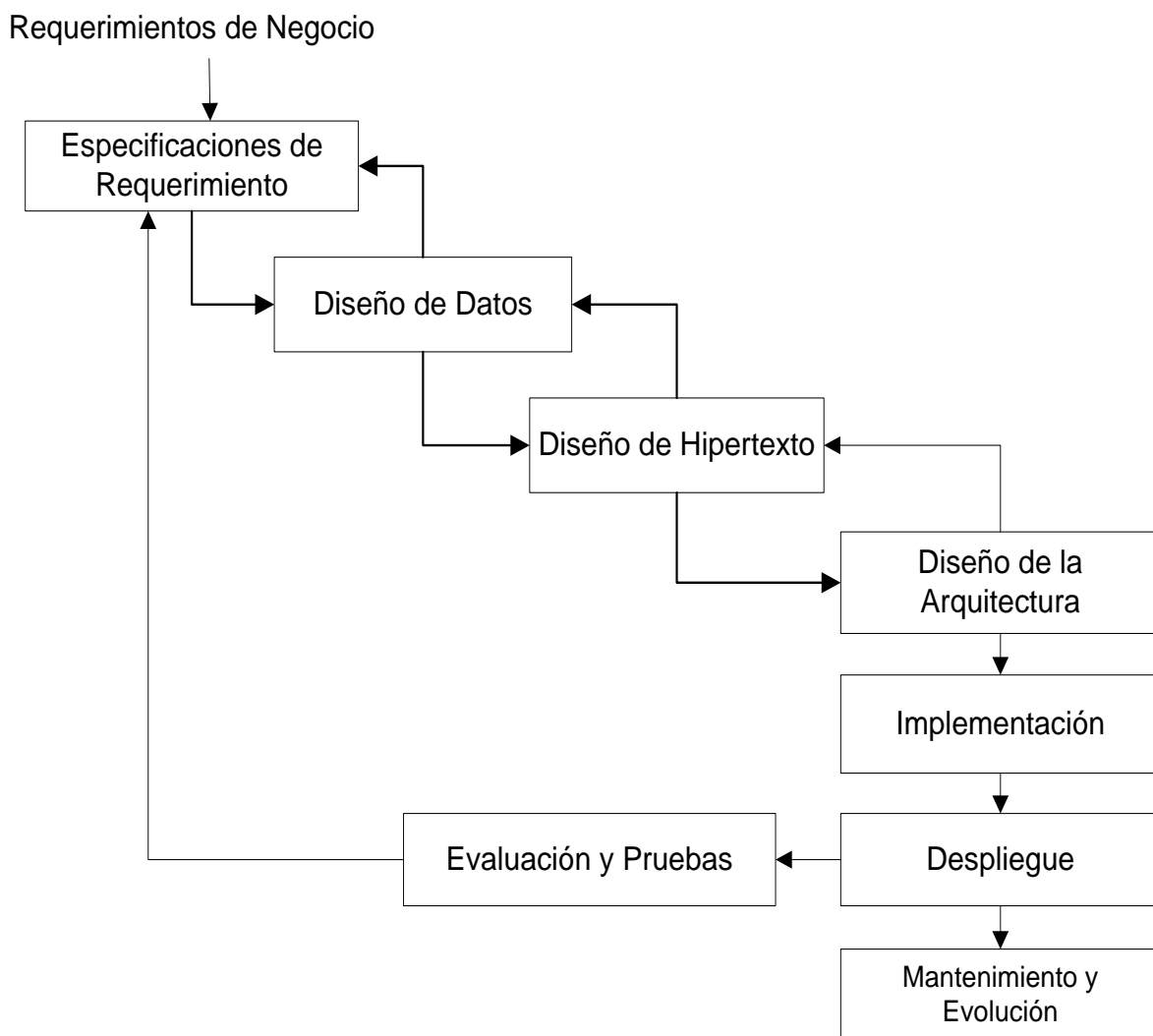


Figura 2.1.6 Enfoque del Método WebML



UML-based Web Engineering methodology (UWE). (Metodología de Ingeniería Web basado en UML).

Método que emplea técnicas procedentes de la orientación a objetos para especificar aplicaciones hipermedia, la cual plantea un enfoque iterativo y progresivo cuyas actividades fundamentales son el análisis de requisitos y el diseño conceptual, de la navegación y de la presentación.

Los elementos hipermedia se representan por medio de elementos propios de los diagramas de clase UML. Para modelar aspectos dinámicos se hace uso de modelos de tarea y diagramas de estado, mientras la navegación y la presentación se representan por medio de UML y de estereotipos creados al efecto.

Ariadne Development Method (ADM). (Método de Desarrollo Ariadne).

Plantea un proceso sistemático integrador e independiente de la plataforma de implementación para modelar y evaluar aplicaciones y sistemas hipermedia y de web.

Plantea un modelo de proceso iterativo y centrado en el usuario con el que se pueden modelar las características del sistema de una forma completa, integradora e independiente de una plataforma ADM aborda estos puntos:

- Elementos de contenido multimedia rigurosamente(imagen, sonido, etc), tomando en cuenta aspectos estéticos y de usabilidad.
- Integrar la especificación de requisitos funcionales no relacionados con la facilidad de navegación.



- Incorporar a los usuarios del sistema en el diseño resultante, de manera que este modelado facilite el de sistemas adaptivos o de reglas de acceso a la información.
- Aportar medios para establecer políticas de acceso con las que se pueda determinar quién puede hacer qué operaciones sobre determinados objetos del sistema.

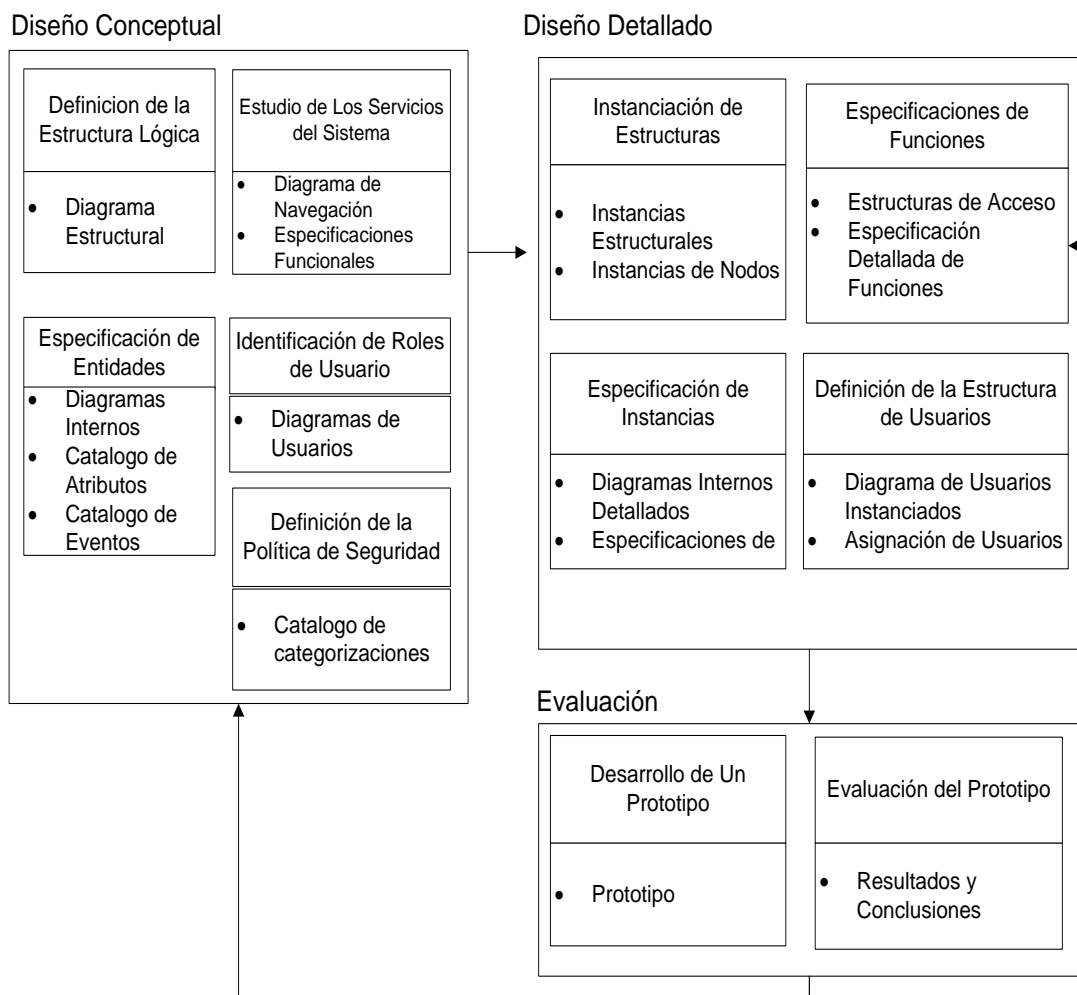


Figura 2.1.7. Fases del Método Ariadne.



En la Figura 2.1.7 se observa la división de fases integrantes del método; se destaca en principio que hay una subdivisión de las mismas, existiendo por tanto una relación entre ambas. Los desarrolladores web requieren una mejor productividad y calidad para satisfacer las necesidades del mercado y reducir los tiempos de entrega y costos. Estos métodos presentados para el desarrollo web aún cuentan con deficiencias en los modelos y herramientas de análisis y diseño que carecen de conceptos adecuados para capturar las propiedades de desarrollo, la documentación es reducida y con baja calidad, los costos y tiempo son impredecibles y escapan del control durante las fases de mantenimiento y evolución.

Pero en un tiempo no muy remoto se espera tener herramientas de desarrollo que ofrezcan: Modelado visual con uso de constructores de modelado de alto nivel, facilitando la generación de código portable y eficiente, soporte de todo el ciclo de vida de la aplicación; finalmente la integración de módulos de lógica preexistentes (**legacy software**) en un proceso de desarrollo optimizado.

2.2. Características, Ventajas y Desventajas de las Bases de Datos Relacionales.

Las bases de datos en la actualidad son vitales para las organizaciones modernas. Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su empleo posterior.

Existe una estrecha relación entre la información y el desarrollo económico y social. Las bases de Datos están consideradas como un sistema, el cual está definido como el “Conjunto de cosas que ordenadamente relacionadas entre sí contribuyen a



determinado objeto” definición proveniente del Diccionario de la Real Academia Española (**DRAE**).

Algunas características importantes y distintivas de las bases de Datos son:

- Persistencia, los datos residen en un área de almacenamiento estable.
- Los datos pueden compartirse por varios usuarios. Múltiples usos y usuarios.
- Existe una interrelación entre los datos almacenados, esto es que los datos pueden estar almacenados en diferentes dispositivos pero pueden conectarse para mostrar la información completa.

Un sistema de administración de Base de Datos (DBMS) son una serie de componentes que permiten crear, usar y mantener los datos almacenados. Gracias a su evolución, ha llevado a proporcionar un amplio rango de características para incorporar, almacenar, divulgar, mantener, recuperar y formatear datos por lo que se han vuelto más complejos. La función más importante de las Bases de Datos es la habilidad de responder a cierta solicitud de datos respondiendo a una pregunta, mejor conocida como consulta. Una base de datos debe tener una identidad separada de las aplicaciones que la usan (programas, formularios y reportes). La identidad separada o independencia de datos, permite que las definiciones de las Bases de Datos se modifiquen sin afectar las aplicaciones relacionadas.

Con el creciente avance de las redes de cómputo y el Internet, el procesamiento distribuido es una función vital en los DBMS; este permite que equipos geográficamente distantes proporcionen el acceso de datos almacenados con la ayuda de la arquitectura cliente servidor, el cual es un arreglo de componentes como son clientes, servidores y datos entre equipos conectados a una red. La base de datos debe dar servicio a un gran número de usuarios, apoyar políticas corporativas, incluir datos relevantes e importantes y dar un acceso eficiente.



En la figura 2.2.1 se presentan las fases necesarias para el desarrollo de una base de datos operacional. Las primeras dos fases se enfocan en la información contenida en la base de datos, las dos últimas se enfocan en una implementación eficiente. El modelado conceptual de datos, utiliza los requerimientos de datos y genera los diagramas entidad relación (**ERD**) para el esquema conceptual y para cada uno de los esquemas externos.

El diseño lógico de bases de datos transforma el modelo conceptual de datos en un formato comprensible para un sistema manejador de bases de datos comercial. El Diseño de bases de datos distribuidas se enfoca en la implementación eficiente; involucra seleccionar la ubicación de los datos y procesos, de manera tal que mejore el desempeño en tiempo de respuesta, disposición de datos y control. La implementación eficiente minimiza el tiempo de respuesta sin tener que usar demasiados recursos, como el espacio en disco y memoria.

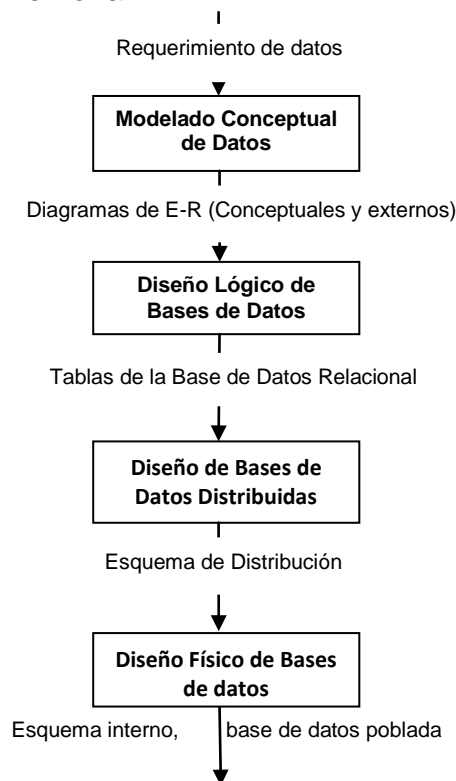


Figura 2.2.1. Fases de Desarrollo de una Base de Datos.



Características de las bases de datos.

Redundancia. Se refiere, a la existencia de información repetida o duplicada en diferentes tablas dentro de una base de datos. Conduce a muchos problemas que tienen que ver con la integridad y consistencia de los datos.

Consistencia. Frecuentemente los problemas de consistencia de datos se deben a la redundancia de éstos. Es muy probable que surjan incongruencias al almacenar la misma información en más de un lugar.

Integridad. La integridad de una base de datos se refiere no sólo a que los datos sean consistentes dentro de la base, sino además que los valores que posean los datos, sean válidos de acuerdo a las dependencias funcionales entre tablas y de acuerdo a las políticas de negocio. La integridad de la base de datos se puede lograr mediante:

- El mantenimiento de una redundancia mínima y controlada.
- El establecimiento de llaves primarias.
- La validación de las dependencias entre tablas relacionadas.
- La creación de reglas de validación durante la inserción y edición de datos.

Seguridad. La seguridad implica asegurar que los usuarios están autorizados para llevar a cabo lo que tratan de hacer.

Por otro lado, una base de datos debe cumplir con las siguientes condiciones:

- Tanto los usuarios finales como los programas de aplicación no necesitan conocer los detalles de las estructuras de almacenamiento ya que lo importante para estos usuarios es la información contenida.



- Los datos son compartidos por diferentes usuarios y programas de aplicación; existe un mecanismo común para inserción, actualización, borrado y consulta de los datos. Los procedimientos de actualización, recuperación común y bien delimitada, habrán de ser capaces de conservar la integridad, seguridad y confidencialidad del conjunto de datos.
- Tanto datos como procedimientos pueden ser transportables conceptualmente a través de diferentes SGBD.

Cardinalidad de las relaciones.

La cardinalidad se define como una restricción sobre el número de entidades que participan en una relación. Las cardinalidades se clasifican por los valores comunes para una cardinalidad mínima y máxima. Una cardinalidad mínima de uno o más indica una relación obligatoria. Una relación obligatoria hace que el tipo de entidad dependa de la existencia de la relación. En contraste, una cardinalidad mínima de cero indica una relación opcional.

Una cardinalidad máxima de uno significa que la relación es de un solo valor o funcional. La palabra función proviene de las matemáticas, la función puede proporcionar un valor. Una relación que tiene una cardinalidad máxima de uno en una dirección y más de una (muchas) en la otra, se llama relación 1-M (uno a muchos). Una relación que tiene una cardinalidad máxima mayor a 1 en ambas direcciones es conocida como relación M-N (muchos-a-muchos). Las relaciones M-N son comunes en las bases de datos de negocios. Las relaciones 1-1 son menos comunes pero existentes, su peculiaridad es que la cardinalidad máxima es igual a uno en ambas direcciones. La tabla 2.2.2 muestra un resumen de la clasificación de cardinalidades.



| Clasificación | Restricciones de las Cardinalidades |
|----------------------------|---|
| Obligatoria | Cardinalidad mínima ≥ 1 |
| Opcional | Cardinalidad mínima = 0 |
| Funcional o de valor único | Cardinalidad máxima = 1 |
| 1-M | Cardinalidad máxima = 1 en una dirección y Cardinalidad máxima > 1 en otra dirección |
| M-N | Cardinalidad máxima > 1 en ambas direcciones |
| 1-1 | Cardinalidad máxima = 1 en ambas direcciones |

Tabla 2.2.2. Resumen Clasificación de Cardinalidades

Bases de Datos Relacionales.

Este es un modelo de datos basado en la teoría de las relaciones, en donde los datos se estructuran lógicamente en forma de relaciones (tablas), siendo un objeto fundamental del modelo mantener la independencia de esta estructura lógica respecto al modo de almacenamiento y a otras características de tipo físico.

El modelo relacional para la gestión de una base de datos es un modelo de datos basado en la lógica de predicado y en la teoría de conjuntos. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Este modelo de datos persigue los siguientes objetivos: Independencia física, independencia lógica, flexibilidad, uniformidad y sencillez.

La peculiaridad de las bases de datos relacionales se centra en que existe una base matemática, incluyendo la conceptualización de las tablas como conjuntos. La base de datos relacional está formada por una recopilación de tablas interrelacionadas.



La comprensión de las conexiones o relaciones entre tablas en una base de datos relacional es necesaria, las relaciones se muestran mediante los valores de una columna en una tabla que coinciden con los valores de una columna en otra tabla.

Su idea fundamental es el uso de «relaciones». Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados «tuplas». En este modelo se representan los datos y las relaciones entre estos, a través de una colección de tablas, en las cuales los renglones (tuplas) equivalen a cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro localizado en la tupla.

En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que estos se almacenen no tiene relevancia. La información puede ser recuperada o almacenada por medio de «consultas» que ofrecen una amplia flexibilidad y poder para administrar la información. Este enfoque permite a los usuarios obtener información de la base de datos sin asistencia de sistemas profesionales de administración de información.

Las entradas en la tabla tienen un solo valor (son atómicos); no se admiten valores múltiples, por lo tanto la intersección de un renglón con una columna tiene un solo valor, nunca un conjunto de valores.

Estructura del Modelo Relacional.

La relación es el elemento básico del modelo relacional. La tabla 2.2.3 muestra los elementos integrantes del Modelo relacional.



| | |
|---------------------|-----------------------------------|
| Relación | Tabla |
| Tupla | Fila |
| Atributo | Columna |
| Número de tuplas | Cardinalidad |
| Número de atributos | Grado |
| Dominio | Colección de valores |
| Clave primaria | Identificador único para la tabla |

Tabla 2.2.3. Elementos básicos del Modelo Relacional

Dominio. Un Dominio es un conjunto finito de valores homogéneos y atómicos caracterizados por un nombre. Esto es, que los valores son todos del mismo tipo y son indivisibles, es decir, si se descomponen se perdería la semántica del dominio; por ejemplo: Dominio de Nacionalidades: Mexicana, Francesa, Norteamericana. La importancia de los dominios es que restringen las comparaciones, es decir, sólo se pueden comparar atributos definidos sobre el mismo dominio.

Atributo. Un atributo es aquel que participa en la descripción de las entidades y que como tal constituye una pieza específica de información para un determinado dominio. Una tabla no puede tener dos atributos con el mismo nombre, sin embargo, existen los dominios compuestos los cuales están conformados por otros dominios, además de tener un nombre y permitir aplicar restricciones.

Relación. Matemáticamente una relación definida sobre los n dominios D_1, D_2, \dots, D_n , es un subconjunto del producto cartesiano de estos dominios donde cada elemento de la relación (tupla), es una serie de n valores ordenados.

Claves. El término clave en una relación es un conjunto no vacío de atributos que identifican unívoca y mínimamente cada tupla. Toda relación siempre tendrá una clave candidata, estas claves pueden ser clasificadas en tres principales grupos:



- **Clave primaria.** Es aquella clave que permite identificar las tuplas de la relación de forma única.
- **Claves alternativas.** Son aquellas claves que no han sido escogidas como claves primarias, pero que también podrían identificar de manera única a una tupla.
- **Clave foránea.** Es un conjunto no vacío de atributos cuyos valores han de coincidir con los valores de la clave primaria de una relación.

Restricciones. Las restricciones son estructuras no permitidas. Hay dos tipos: inherentes y del usuario. Las inherentes son aquellas propias al modelo, por ejemplo, no tener tuplas repetidas y las del usuario son aquellas que validan las instancias de las relaciones.

Justificaciones. En el modelo relacional se manejan algunas justificaciones, como:

- Las entidades del mundo real son identificables y distinguibles.
- Si una entidad es lo bastante importante en el mundo real como para requerir una representación explícita en la base de datos, tal entidad deberá ser susceptible de identificarla sin ambigüedad, pues de lo contrario sería imposible hablar de ella, es por ello que se dice que en una base de datos, nunca se registrara información acerca de algo que no se pueda identificar.

La arquitectura relacional consta de los siguientes componentes:

Modelo relacional de datos: En el nivel conceptual, el modelo relacional de datos está representado por una colección de relaciones almacenadas. Cada registro de tipo conceptual en un modelo relacional de datos se implanta como un archivo almacenado distinto.



Sub-modelo de datos: Los esquemas externos de un sistema relacional se llaman sub-modelos relacionales de datos, cada uno consta de uno o más escenarios (vistas) para describir los datos requeridos por una aplicación dada. Un escenario puede incluir datos de una o más tablas de datos. Cada programa de aplicación está provisto de un buffer (“Área de trabajo de usuario”) donde el DBMS puede depositar los datos recuperados de la base para su procesamiento, o puede guardar temporalmente sus salidas antes de que el DBMS las escriba en la base de datos.

Esquema de almacenamiento: En el nivel interno, cada tabla base se implanta como un archivo almacenado. Para la recuperación sobre la clave principal o secundaria se pueden establecer uno o más índices para acceder a un archivo almacenado.

Sub-lenguaje de datos: Es un lenguaje de manejo de datos para el sistema, el álgebra y cálculo relacional, ambos lenguajes son “relacionalmente completos”, esto es, cualquier relación que pueda derivarse de una o más tablas de datos, también se puede derivar con un solo comando del sub lenguaje. Por tanto, se puede recuperar una tabla en vez de un solo registro con la ejecución de un comando del sub-lenguaje de datos.

Las bases de datos relacionales pasan por un proceso al que se le conoce como normalización de una base de datos, el cual es entendido como el proceso necesario para que una base de datos sea utilizada de manera óptima. Entre las ventajas que este modelo ofrece están:

- Garantiza herramientas para evitar la duplicidad de registros, a través de campos claves o llaves.
- Garantiza la integridad referencial: Así al eliminar un registro elimina todos los registros relacionados dependientes.
- Favorece la normalización por ser más comprensible y aplicable.



La normalización es el proceso de dividir los datos en diferentes tablas para minimizar el número de veces que es necesario repetir los mismos datos. Las formas normales son grados de normalización y están sujetas a una serie de reglas. Las relaciones que describe Codd al definir su modelo relacional, son transformadas en tablas al momento de trabajar con una base de datos. Las reglas si estas se cumplen hacen que las estructuras posean la menor cantidad de redundancia posible. En la figura 2.2.4 se muestra un diagrama correspondiente a las formas normales.

- Primera Forma Normal (1NF): No hay campos múltiples (todos los campos son atómicos). Todas las filas deben tener el mismo número de columnas.
- Segunda Forma Normal (2NF): Todo campo que no sea clave debe depender por completo de toda la clave.
- Tercera Forma Normal (3NF): No hay dependencias transitivas. Un campo debe depender de la clave y no de otro campo.
- Forma Normal de Boyce-Codd (BCNF): Todos los determinantes de la tabla son clave candidata.
- Cuarta Forma Normal (4NF): Una fila no debe contener dos o más campos multivalorados (aquellos que pueden contener más de un valor simultáneamente) sobre una entidad.
- Quinta Forma Normal (5NF): Una tabla puede almacenar atributos dependientes a la clave sólo por unión.

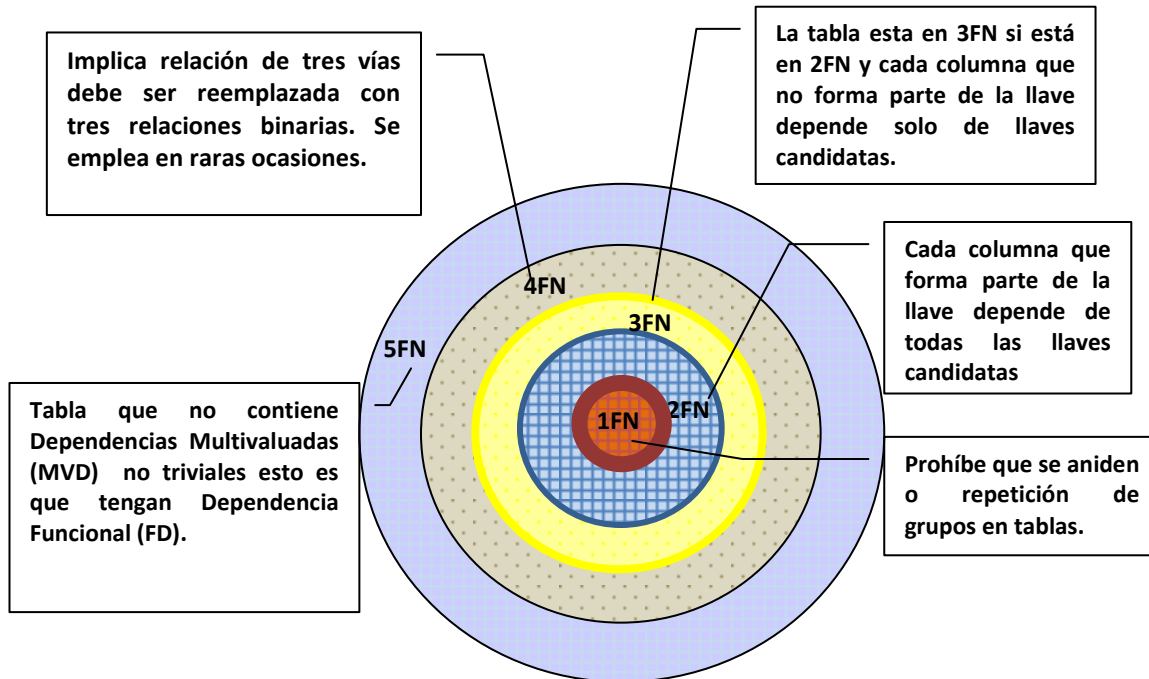


Figura 2.2.4 Forma Normal de Codd

Reglas de Codd.

La existencia de numerosos SGBD que se anuncian como relacionales; carecen de muchas características que se consideran importantes en un sistema relacional, perdiendo muchas ventajas del modelo relacional. Por ello Codd propuso las siguientes 12 condiciones que toda base de datos debe cumplir para considerarse relacionales:

- Representación de la Información. Toda la información dentro de una base de datos relacional se representa de manera explícita a nivel lógico y exactamente de una sola manera, como valores en una tabla.
- Regla del acceso garantizado. Se garantiza que todos y cada uno de los datos (valor atómico) en una base de datos relacional pueden ser leídos recurriendo a una combinación del nombre de la tabla, valor de la llave primaria y nombre de la columna.



- El manejo sistemático de los valores nulos. En un DBMS totalmente relacional se soportan los valores nulos (que son distintos de una cadena de caracteres vacía o de una cadena con caracteres en blanco o de cero o cualquier otro número), para representar información faltante o no aplicable de una forma consistente, independientemente del tipo de dato.
- Catálogo dinámico en línea basado en un modelo relacional. La descripción de la base de datos se representa en el nivel lógico de la misma forma que los datos ordinarios, de tal suerte que los usuarios autorizados puedan aplicar el mismo lenguaje relacional para consultarla, que aquél que emplean para con sus datos habituales.
- Regla del sub-lenguaje de dato completo. Se debe contar con un sub-lenguaje que contemple la definición de datos, la definición de vistas, la manipulación de datos, las restricciones de integridad, la autorización, el inicio y fin de una transacción.
- Regla de actualización de vistas. Todas las vistas que teóricamente sean actualizables deberán ser actualizadas por medio del sistema.
- Inserción, actualización y eliminación de alto nivel. La posibilidad de manejar una relación base o una relación derivada como un sólo operador se aplica a la lectura, inserción, modificación y eliminación de datos.
- Independencia física de los datos. Los programas de aplicación y la actividad en terminales no deberán ser afectados por cambios en el almacenamiento físico de los datos o en el método de acceso.
- Independencia lógica de los datos. Los programas de aplicación y la actividad en terminales no deberán ser afectados por cambios de cualquier tipo que preserven la información y que teóricamente permitan la no afectación en las tablas base.



- Independencia de la integridad. Las restricciones de integridad de una base de datos deberán poder definirse en el mismo sub-lenguaje de datos relacional y deberán almacenarse en el catálogo, no en los programas de aplicación.
- Independencia de la distribución. Un DBMS relacional tiene independencia de distribución. Debe existir un sub lenguaje de datos que pueda soportar bases de datos distribuidas sin alterar los programas de aplicación cuando se distribuyan los datos por primera vez o se redistribuyan éstos posteriormente.
- Regla de la no subversión. Si un sistema relacional tiene un lenguaje de bajo nivel (un sólo registro cada vez), ese bajo nivel no puede ser utilizado para suprimir las reglas de integridad y las restricciones expresadas en el lenguaje relacional de nivel superior (múltiples registros a la vez).

Para manipular la información utilizamos un lenguaje relacional, actualmente se cuenta con dos lenguajes formales el Álgebra relacional y el Cálculo relacional. El Álgebra relacional permite describir la forma de realizar una consulta, en cambio, el Cálculo relacional sólo indica lo que se desea devolver.

Uno de los lenguajes comunes para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

El modelo relacional, lleva asociado en su parte estática (estructura y restricciones) una dinámica que permite la transformación entre estados de la base de datos: Esta transformación de un estado origen a un estado objetivo se realiza aplicando un conjunto de operadores, mediante los cuales se llevan a cabo las siguientes operaciones: Inserción, borrado, modificación y consulta de tuplas.



La dinámica del modelo relacional actúa sobre conjuntos de tuplas y se expresa mediante lenguajes de manipulación relacionales que asocian una sintaxis concreta a las operaciones. Los lenguajes relacionales son lenguajes de navegación y especificación, divididos en dos tipos:

- Algebraicos. Caracterizados por los cambios de estado, especificados mediante operaciones, los operandos son relaciones y el resultado es otra relación. Álgebra relacional
- Predicativos. Los cambios de estado se especifican mediante predicados que definen el estado objetivo sin indicar las operaciones que hay que realizar para llegar al mismo. Conocido como Calculo relacional orientado a tuplas u orientado a dominios.

El algebra relacional consiste en una colección de operaciones sobre relaciones donde cada operación toma una o más relaciones como su operando y produce otra relación como su resultado. Dado que el resultado de una operación del álgebra relacional es una relación, ésta a su vez puede ser sujeto de posteriores operaciones algebraicas.

- El algebra relacional se basa en la teoría de conjuntos, relaciones y en el álgebra de conjuntos
- Adicionalmente al conjunto básico de operadores como: unión, diferencia, producto cartesiano e intersección; incorpora operadores específicos de base de datos tales como proyección, selección y join.
- Las bases de datos relacionales están basadas en el concepto matemático de relaciones entre conjuntos. Así las operaciones que se pueden efectuar entre relaciones son tanto las comunes a los conjuntos, unión, intersección, diferencia, producto cartesiano; como las específicas de las relaciones, selección, proyección, etc.



- Si q , r y s son relaciones con todos los dominios iguales, esto es, con el mismo esquema, se les puede aplicar las operaciones típicas de conjuntos.

Hoy en día se puede observar el cambio de los sistemas de gestión de bases de datos. Inicialmente se trataba de software muy caro, sobre grandes y costosas computadoras. Actualmente existen sistemas de gestión de bases de datos para computadoras personales, siendo muchos de ellos económicos o incluso gratuitos. Esta tendencia al abaratamiento y disminución en tamaño físico de los sistemas contrasta con la cada vez mayor capacidad, potencia y prestaciones de los SGBD.

2.3 Características, Ventajas y Desventajas del Software Gratuito.

Desde décadas atrás el usuario ha tenido que acostumbrarse a la imposición de condiciones y prohibiciones al comprar un software y no poder adaptarlo a las necesidades del cliente o realizar corrección de fallas. Precisamente es el software libre el que concede las libertades que el software propietario niega.

El término software libre (o programas libres) se refiere en concreto a cuatro libertades:

- Libertad para ejecutar el programa en cualquier sitio, con cualquier propósito y para siempre.
- Libertad para estudiarlo y adaptarlo a nuestras necesidades. Esto exige el acceso al código fuente.
- Libertad de redistribución, de modo que se nos permita colaborar con vecinos y amigos.



- Libertad para mejorar el programa y publicar las mejoras. También exige el código fuente.

Estas libertades se pueden garantizar de acuerdo con la legalidad vigente por medio de una licencia. En ella se plasman las libertades, pero también restricciones compatibles con ellas, como dar crédito a los autores originales si redistribuimos. Incluso puede obligarnos a que los programas ajenos mejorados por nosotros también sean libres, promoviendo así la creación de más software libre.

En inglés el término free además de 'libre' significa 'gratis', lo que genera gran confusión. Por ello a menudo en inglés se toman prestadas palabras españolas y se habla de 'software libre', en contraposición a 'software gratis', al igual que nosotros tomamos prestada la palabra software.

En si no se está hablando de software gratuito, el software libre se puede vender si se desea. Pero debido a la tercera libertad, cualquiera puede redistribuirlo sin pedir dinero a cambio ni permiso a nadie, lo que hace prácticamente imposible obtener dinero por distribuirlo, salvo la pequeña cantidad que se pueda cargar por grabarlo en un soporte físico y enviarlo, algo raramente demandado excepto para grandes volúmenes, como es el caso de las distribuciones.

Se han formalizado definiciones más precisas de software libre, como es el caso notable de las directrices de la distribución Debian. En ellas se permite además que el autor exija que los códigos fuentes distribuidos no sean modificados directamente, sino que los originales se acompañen de parches separados y que se generen programas binarios con distinto nombre que el original. Además, exigen que las licencias no contaminen otros programas distribuidos en el mismo medio.



Equivalente a software libre es el término **Open Source Software** ('programas de fuente abierto'), promovido por Eric Raymond y la **Open Source Initiative**. El término hace énfasis en la disponibilidad de código fuente, no en la libertad, pero su definición es prácticamente la misma que la de Debian. Puede dar lugar a ventajas técnicas, como mejores modelos de desarrollo, negocio y mayor seguridad, etc. Fuertemente criticado por Richard Stallman y la **Free Software Foundation**, ha encontrado más eco en la literatura comercial y en las estrategias de las empresas que de una manera u otra apoyan el modelo. Otros términos relacionados de alguna manera con el software libre son:

- **Freeware.** (Programas gratuitos). Normalmente se ceden en binario y con derechos de redistribución. Sin embargo, a veces sólo se pueden obtener de un sitio oficial, normalmente para promocionar otros programas o servicios, como es el caso de los kits de Java gratuitos que proporciona Sun Microsystems.
- **Shareware.** No es software gratis, sino un método de distribución, ya que los programas, generalmente sin fuentes, se pueden copiar libremente, pero no usar continuamente sin pagarlos.
- **Charityware, Careware.** Semejante a shareware, pero cuyo pago se exige para una organización caritativa patrocinada. En muchos casos, el pago no se exige, pero se solicita una contribución voluntaria.
- **Dominio público.** El autor renuncia absolutamente a todos sus derechos, a favor del común, lo cual tiene que estar declarado explícitamente en el programa, ya que si no se dice nada, el programa es propietario y no se puede hacer nada con él. En este caso, y si además se proporcionan los códigos fuentes, el programa es libre.



- **Copyleft.** Un caso particular de software libre cuya licencia obliga a que las modificaciones que se distribuyan sean también libres.
- **Propietario, cerrado, no libre.** Términos usados para denominar al software que no es libre ni de fuente abierta.

Hay dos grandes motivaciones para el desarrollo de software libre. La motivación ética, abanderada por la Free Software Foundation, heredera de la cultura hacker, y partidaria del apelativo libre, que argumenta que el software es conocimiento y debe poderse difundir sin trabas.

La motivación pragmática, abanderada por la Open Source Initiative y partidaria del apelativo fuente abierta, que argumenta ventajas técnicas y económicas.

El software libre trae consigo numerosas ventajas y pocas desventajas. De ellas la que más fundamento tiene es la económica, ya que no es posible obtener mucho dinero de la distribución y ésta la puede y suele hacer alguien distinto al autor. El modelo de costos del software libre es muy distinto del propietario, ya que gran parte de él se ha desarrollado fuera de la economía formal monetaria, muchas veces con mecanismos de trueque: “yo te doy un programa que te interesa y tú lo adaptas a tu arquitectura y le haces mejoras que a ti te interesan”.

Otra consecuencia de las libertades es la calidad que se deriva de la colaboración voluntaria de gente que contribuye o que descubre y reporta errores en entornos y situaciones inimaginables por el desarrollador original. Además, si un programa no ofrece la calidad suficiente, la competencia puede tomarlo y mejorarlo, partiendo de lo que hay. Así, dos poderosos mecanismos: la colaboración y la competencia se combinan en aras de la calidad.



Ya no depende tanto de la fiabilidad del fabricante para intentar deducir la calidad de un producto, sino que la guía nos dará la aceptación de la comunidad y la disponibilidad de los códigos fuentes.

La evaluación de productos antes de adoptarlos es ahora mucho más sencilla, ya que basta con instalar los productos alternativos en nuestro entorno real y probar, mientras que para el software propietario hay que fiarse de informes externos o negociar pruebas con los proveedores, lo cual no es siempre posible.

Dada la libertad de modificar el programa para uso propio, el usuario puede personalizarlo o adaptarlo a sus necesidades, corrigiendo errores si los tuviera. También podemos, directamente o contratando servicios, integrar el programa con otro, o auditar su calidad (por ejemplo, la seguridad). El control pasa, en gran medida, del proveedor al usuario.

La Administración pública es un gran usuario de características especiales, ya que tiene obligaciones con el ciudadano, ya sea proporcionándole servicios accesibles, neutrales respecto a los fabricantes, garantizando la integridad, utilidad, privacidad y seguridad de sus datos a largo plazo. Todo ello la obliga a ser más respetuosa con los estándares que las empresas privadas y a mantener los datos en formatos abiertos y manipulados con software que no dependa de estrategia de empresas, generalmente extranjeras, certificado como seguro por auditoría interna. La adecuación a estándares es una característica notable del software libre que no es tan respetada por el software propietario, ávido en muchos casos notables de crear mercados cautivos.

Asimismo, la Administración tiene una cierta función de escaparate y guía de la industria que la hace tener un gran impacto, que debería dirigirse a la creación de un tejido tecnológico generador de riqueza nacional. Ésta puede crearse fomentando empresas,



cuyo negocio sea en parte el desarrollo de nuevo software libre para la administración, o el mantenimiento, adaptación o auditoría del software existente.

Para el desarrollador y productor de software, la libertad cambia las reglas del juego. Es más fácil competir siendo pequeño y adquirir tecnología de punta. Puede aprovecharse del trabajo de los demás, compitiendo incluso con otro producto modificando su propio código, si bien también el competidor copiado se aprovechará de nuestro código (si es copyleft). Un buen proyecto puede conseguir la colaboración gratuita de mucha gente. Asimismo, la distribución es barata y global.

Para el integrador el software libre es el paraíso. Puede integrar trozos de programas para conseguir el producto necesario, disponiendo de un acervo de software libre de donde extraer las piezas.

Disponer del código fuente lo cambia todo, situándonos casi en las mismas condiciones que el productor. El valor añadido de los servicios es mucho más apreciado, ya que el costo del programa es bajo. Éste es actualmente el negocio más claro con software libre y con el que es posible un mayor grado de competencia.

La documentación que viene con un programa es parte integrante del mismo, como lo son los comentarios del código fuente. Así lo reconoce la Ley de Propiedad Intelectual, y parece lógico que se le apliquen las mismas libertades y evolucione de la misma manera.

Toda modificación que se haga de un programa requiere un cambio simultáneo y consistente en su documentación. La mayor parte de esta documentación suele estar codificada como archivos de texto sin formato, ya que se pretende que sea universalmente accesible con un entorno de herramientas mínimo, y debe incluir una pequeña introducción al programa documentado (README o LEEME), instrucciones de instalación (INSTALL), alguna historia sobre la evolución pasada y futura del programa



(changelog), autoría y condiciones de copia (AUTHORS y copyright), así como las instrucciones de uso.

Todas ellas, menos la autoría y las condiciones de copia, deberían ser libremente modificables según el programa va evolucionando. A la autoría sólo se le deberían añadir nombres y créditos, pero sin borrar nada, y las condiciones de copia sólo deberían modificarse si estas mismas lo permiten.

El software libre exige que esta documentación pueda ser modificada fácilmente, lo que a su vez obliga a usar formatos denominados transparentes, de especificación conocida y procesables por herramientas libres, como son, además del texto puro y limpio, el formato de páginas de manual de Unix, TexInfo, LaTeX o DocBook, sin perjuicio de distribuir también el resultado de transformar esos documentos fuente en formatos más aptos para visualizar o imprimir, como HTML, PDF o RTF (formatos opacos).

La variante tecnológica de las leyes son las normas o estándares. En programación son especialmente importantes los protocolos de comunicaciones, ya sea entre máquinas remotas o entre módulos de la misma máquina.

Es obvio que no debe limitarse su difusión, especialmente si queremos que florezcan los programas libres que operen con otros, pero a pesar de ello, tradicionalmente, los organismos de normalización, como la ISO e ITU, venden sus normas, incluso en formato electrónico, y prohíben su redistribución. Aunque pueda intentar justificarse esto para cubrir parcialmente los gastos, la libre difusión del texto de los estándares ha resultado mucho más productiva.

Éste es el caso de las recomendaciones del W3C y sobre todo las que gobiernan Internet, disponibles desde el principio en documentos llamados RFC, de (Request for Comments), en formatos electrónicos legibles en cualquier editor de textos. Pero no es la disponibilidad la única causa del éxito de los protocolos de Internet. También lo es su modelo de desarrollo, muy similar al del software libre por su carácter abierto a la



participación de cualquier interesado y por la utilización de listas de correo y medios similares.

2.4. Características, Ventajas y Desventajas de las Bases de Datos Gratuitas

Una base de datos gratuita es software de código abierto, lo que significa que es posible para cualquier persona usarlo y modificarlo, se puede bajar el código fuente de estas bases y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades.

Además el tipo de licencia que manejan es GPL **General Public Licence (Licencia Publica General)** la cual está orientada principalmente a proteger la libre distribución, modificación y uso de software.

Características:

- Disponibilidad en gran cantidad de plataformas y sistemas, como: IBM AIX, FreeBSD, GNU/Linux, Mac OS X, NetBSD, Novell Netware, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, todas las versiones de Windows, entre otras.
- Contienen varias API's **Application Programming Interface (Interfaz de Programación de Aplicaciones)** que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos incluyendo C, C++, C#, Pascal, Delphi, Eiffel, Smalltalk, Java, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac), FreeBASIC, y Tcl; cada uno de estos utiliza una API específica.



- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.
- Procesamiento distribuido de transacciones, permite distribuir las transacciones a través de los diferentes servidores.

Acceso a Datos:

- **Índices.** Característica que mejora el rendimiento durante el acceso a una tabla.
- **Diccionario de Datos.** Esta característica de la base de datos permite obtener fácilmente la información de los objetos que pertenecen a la base de datos como son las tablas, vistas, procedimientos almacenados, etc.
- **Vistas y Vistas Actualizables.** Es una característica que permite visualizar los datos de una o varias tablas.
- **Secuencias.** Objeto de base de datos que se utiliza para generar números exclusivos. La réplica utiliza secuencias para ordenar los cambios efectuados en las tablas publicadas.
- **SubSelect (Sub Consultas).** Es una sentencia que se utiliza en lugar de la cláusula **values (valores)**. Puede contener **join (uniones)**, llamar a funciones, y puede incluso consultar en la misma TABLA los datos que se insertan.
- **Triggers (disparador).** Subrutina que es ejecutada de manera automática cuando se produce algún tipo de transacción (inserción, borrado o actualización) en la tabla de una base de datos.
- **Datos especiales** ver tabla 2.4.1:



| Tipo de dato: | Características. |
|------------------|---|
| Bit | Representa valores TRUE o FALSE, YES o NO, 1 ó 0 |
| Uniqueidentifier | Número hexadecimal de 16 Bytes que identifica como única una fila entre muchas. |
| sql_variant | Cualquier tipo de datos permitido, excepto txt, ntext, timestamp) |
| Table | Almacena resultados para más adelante procesarlos. |

Tabla 2.4.1 Datos Especiales

- **Table space (espacio de tabla).** Unidad de almacenamiento de base de datos que es en líneas generales equivalente a un grupo de archivos en SQL Server.
- **Cursores.** Son utilizados dentro de los procedimientos almacenados para seleccionar o modificar datos con cierta lógica de negocio.
- **Integridad de datos.** Es un sistema de reglas que utilizan la mayoría de las bases de datos relacionales para asegurarse que los registros de tablas relacionadas son válidos y que no se borren o cambien datos relacionados de forma accidental produciendo errores de integridad.
- **Funciones.** Encargadas de retornar un valor o información de una llamada a rutina o de una consulta, esto ayuda mucho a la reutilización de código.
- **Procedimientos almacenados.** Ayudan a reducir el tráfico de red ya que utiliza una sola llamada a la base de datos.
- **Uso de Alias.** Se utilizan para escribir código más corto e identificar de forma explícita columnas de varias tablas.
- **Save points (puntos de recuperación).** Permiten deshacer una transacción completa o parte de ella.



Funcionalidad:

- **Replica.** Proceso de copia y mantenimiento de bases de datos múltiples que hacen un sistema distribuido. Consiste en dos servidores: uno Maestro y uno esclavo, El maestro registra todas las consultas que provocan cambios en la base de datos y las almacena en un archivo de registro binario. El servidor esclavo se conecta al maestro, lee las condiciones en el archivo binario y las ejecuta contra su copia local.
- **Cluster (grupo).** Característica que ayuda al balanceo de carga, es decir, se encarga de encaminar las conexiones entrantes a los servidores disponibles que estén menos ocupados.
- **Partición de tablas.** Se trata de una característica de las bases de datos actuales que permite dividir físicamente una tabla de la base de datos en particiones de acuerdo a la lógica del negocio que se necesite, de forma que, reduce significativamente los tiempos de respuesta de la aplicación.
- **Automatic Storage Management (Administración automática de almacenamiento).** La base de datos es capaz de automáticamente expandir el espacio de almacenamiento.
- **Data WareHouse (Depósito de datos).** Está orientado a temas, integrado, no volátil, de tiempo variante, que se usa para el soporte del proceso de toma de decisiones gerenciales.
- **Federated tables.** Esta técnica hace que las distintas bases de datos físicas se vean como una base de datos lógica para la comunidad de usuarios, esto simplifica el acceso a datos.
- **Funciones XML. Extensible Markup language (lenguaje de marcas extensible)** Formato de archivo estándar usado para transferir datos entre sistemas.



- **Precisión math (matemática).** Soportan operaciones complejas para los clientes que necesitan gran precisión en sus cálculos matemáticos.
- **Archive engine (Archivos de maquina).** Esta herramienta permite almacenar de forma eficiente los datos históricos.
- **Migración de datos.** Es el traspaso de información (datos) entre bases de datos.
- **ACID Atomicity, Consistency, Isolation and Durability (Atomicidad, Consistencia, Aislamiento y Durabilidad).** Soportan atomicidad, consistencia, aislamiento y durabilidad.

Seguridad:

- **Control de acceso a usuarios.** Contienen sistemas de autenticación capaces de aceptar o denegar las conexiones, así como herramientas para la administración de los privilegios y métodos de encriptación.
- **Backups (copia de seguridad) y hot back ups.** Se pueden respaldar con las utilidades en ambiente gráfico disponibles, tiene la capacidad de sacar respaldos, esto es permitido gracias a online backup cuando la aplicación está en actividad normal.
- **Protocolo SSL Secure Sockets Layer (Seguridad de la Capa de Transporte).** Tecnología que mantiene seguros los datos a través de Internet.
- **Encriptación de datos.** Almacenamiento de datos con un formato encriptado para asegurar la privacidad y no divulgación de los mismos.
- Importación y exportación de datos.
- Privilegios de acceso a objetos.
- Perfiles de usuario.



Ventajas:

- La principal ventaja es que no existen costos de licencia para el producto en sí mismo.
- El usuario puede obtener el código fuente, lo cual brinda independencia del proveedor y de este modo el usuario no depende de su existencia.
- Toda la información (estado, **bugs (errores)**, etc.) es abierta, no existe política de ocultamiento corporativa ni censura.
- Si algo no funciona, no hay inconveniente en averiguarlo rápidamente. Como consecuencia, los proyectos de Código Abierto son muy rápidos para reaccionar, si surgen problemas.
- Existe una comunidad de usuarios y desarrolladores que hacen una notable diferencia. Debido a la diversidad de usuarios, los productos están usualmente probados y se puede obtener ayuda rápidamente, por lo cual las bases de datos libres tienen menos errores de programación que las propietarias.
- La persona que desarrolla es la responsable de administrar las operaciones de entrada/salida de la base de datos, así como de habilitar y reforzar ciertas características de acuerdo a la particular aplicación, esto hace más eficiente y posee un **performance (funcionamiento)** de la base de datos comparado con las bases de datos propietarias.
- Al ser una herramienta **Open Source (código abierto)**, los clientes pueden crear su propia ingeniería de almacenamiento que se ajuste a las necesidades de una aplicación en particular.
- Una base de datos relacional libre archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad.



Desventajas:

- La licencia GNU GPL de las bases de datos obliga a que la distribución de cualquier producto derivado (aplicación) se haga bajo esa misma licencia.
- Pueden tener problemas de integridad en entornos de alta concurrencia en la modificación.
- Pueden no poseer todas las características de las bases de datos de códigos propietarios.
- La tabla de índices puede crecer más que la misma base si no se tiene el cuidado del uso de estos.
- La falta de soporte técnico garantizado al cual acudir si los manuales y ayuda en línea no son suficientes.

PostgreSQL.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley. Es una derivación libre OpenSource de este proyecto, y utiliza el lenguaje SQL92/SQL99, así como otras características.

Fue el pionero en conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. Incluye características de orientación a objetos, como puede ser herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. No es un sistema de gestión de bases de datos puramente orientado a objetos.



Hoy en día es el sistema libre más avanzado con diferencia, soportando la gran mayoría de las transacciones SQL, control concurrente, teniendo a su disposición varios lenguajes como por ejemplo C, C++, Java, Python, PHP y muchos más.

Las principales características de este gestor de bases de datos son:

- Implementación del estándar SQL92/SQL99.
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP, etc.), cadenas de bits, etc.
- Incorpora una estructura de datos **array (arreglos)**.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc. Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas. Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a éste gestor de bases de datos se le incluye entre los gestores objeto-relación.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

Ventajas:

- Posee una gran escalabilidad. Es capaz de ajustarse al número de CPU's y a la cantidad de memoria que posee el sistema de forma óptima.
- Implementa el uso de **rollback's (recuperación)**, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en las que MySQL no podría.



- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos.

Desventajas:

- Consume gran cantidad de recursos.
- Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.
- Es de 2 a 3 veces más lento que MySQL.

MySQL.

Es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de información de forma muy eficiente. Fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esto es debido en parte a la existencia de infinidad de librerías y herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de una fácil instalación y configuración. Las principales características de este gestor de bases de datos son las siguientes:



Portal de la Academia Regional de Informática de la DGETI.



- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo. Soporta gran cantidad de tipos de datos. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- Gran portabilidad entre sistemas. Soporta hasta 32 índices por tabla. Gestión de usuarios y **passwords (contraseña)**, manteniendo un muy buen nivel de seguridad en los datos.

Ventajas:

- Velocidad a la hora de realizar las operaciones lo que le hace uno de los gestores que ofrecen mayor rendimiento.
- Facilidad de Uso, capacidad. Permite múltiples hilos de proceso, de manera que varios clientes pueden conectarse de forma simultánea.
- Seguridad. Soporta conexiones encriptadas utilizando el protocolo SSL.
- Tiene un tamaño de distribución modesto, especialmente comparado con el enorme espacio que ocupan determinados sistemas de bases de datos comerciales.
- Disponibilidad y costo. Es un proyecto de Código abierto con licencia dual. Además de poseer Integridad Referencial.

Desventajas:

- Carece de soporte para transacciones, rollback's y subconsultas.
- El mal manejo de índices pueden hacer que estos superen el tamaño de la base.



Firebird.

Es una base de datos relacional que ofrece muchas características de SQL ANSI estándar y funciona en Linux, Windows, MacOSX y una variedad de plataformas UNIX. Ofrece una concurrencia excelente, alto rendimiento y un poderoso lenguaje de procedimientos almacenados y disparadores. Otras características principales son:

- Soporte completo de Procedimientos Almacenados y Triggers. Las Transacciones son totalmente **ACID**.
- Integridad referencial. Arquitectura Multi Generacional. Muy bajo consumo de recursos.
- Completo lenguaje para Procedimientos Almacenados y Triggers (PSQL). Soporte para Funciones externas (UDF's).
- Poca o ninguna necesidad de Administradores de Bases de Datos especializados.
- Opción a usar la versión embebida - de un solo archivo - ideal para crear CDROM con catálogos, versiones de evaluación o monousuario de aplicaciones.
- Escritura segura-recuperación rápida sin necesidad de logs de transacciones.
- Formas de acceder a las bases de datos: nativo/API, driver dbExpress, ODBC, OLEDB, .Net provider, driver JDBC nativo de tipo 4, módulo para Python, PHP, Perl, etc.
- Soporte nativo para los principales sistemas operativos, incluyendo Windows, Linux, Solaris, MacOS. Backups incrementales.
- Disponible para arquitecturas de 64bits. Es medianamente escalable. Completa implementación de cursores en PSQL.



Ventajas:

- Es multiplataforma, y actualmente puede ejecutarse en los sistemas operativos: Linux, HP-UX, FreeBSD, Mac OS, Solaris y Microsoft Windows.
- Ejecutable pequeño, con requerimientos de hardware bajos. Arquitectura Cliente/Servidor sobre protocolo TCP/IP y otros **embedded (embebido)**.
- Soporte de transacciones y claves foráneas. Buena seguridad basada en usuarios/roles. Existencia de controladores ODBC, OLEDB y JDBC. Controlador nativo para PHP.
- Requisitos de administración bajos, siendo considerada como una base de datos libre de mantenimiento, al margen de la realización de copias de seguridad.
- Lenguaje completo para la escritura de triggers y procedimientos almacenados. Capacidad de almacenar elementos BLOB **Binary Large Objects (objetos binarios largos)**.
- Instalado como servidor permite acceder concurrentemente a varios servidores distribuidos de una manera muy eficaz a través de una conexión TCP. Soporta todo lo que se necesita de un motor de base de datos permitiendo consultas complejas.
- Tiene un buen manejo de datos aún cuando sean grandes cantidades. Es totalmente gratuito tanto para uso personal, como comercial.

Desventajas:

- Operar con respecto al servidor :



- No permite conexiones remotas. Solo un usuario/aplicación pueden estar accediendo a la base de datos (porque la misma queda bloqueada cuando es abierta).
- No realiza la autenticación de usuarios (es decir, cualquier aplicación puede acceder a la base de datos sin necesidad de poseer usuario y passwords válidos)

Tabla comparativa de las 3 principales Bases de Datos gratuitas.

| Características | MySQL | PostgreSQL | Firebird |
|---------------------|---|---|--|
| Versión | Mysql-4.1.x | PostgreSQL 8.x | Firebird 1.5.x (Roadmap) |
| Licencia | Dual: GPL y comercial | BSD | IDPL & IPL |
| Plataformas | Linux, Solaris, HP-UX, MacOS, AIX, SCO, IRIX, FreeBSD, NetBSD, OpenBSD, Windows, BSDI, DEC, OS/2, Compaq Tru64, Novell NetWare. | Linux, Solaris, HP-UX, AIX, IRIX, FreeBSD, OpenBSD, NetBSD, MacOS, SCO OpenServer, SCO Unixware, BeOS, BSDI, Compaq Tru64, QNX, Windows | Linux, Solaris, freeBSD, HP-UX, MacOS, Windows |
| Velocidad | Media/Alta | Media | Media/Alta |
| Estabilidad | Alta/Muy alta | Alta | Alta |
| ACID | Si | Si | Si |
| Integridad en Datos | Si | Si | Si |
| Seguridad | Alta | Media/Alta | ? |
| SSL | Si | Si | plug-in |



Portal de la Academia Regional de Informática de la DGETI.



| | | | |
|--------------------------------------|---|---|---|
| Concurrencia | Alta | Alta | Alta |
| Vistas | Si | Si | Si |
| Esquemas | Si (>=5.0) | Si | NO |
| Subselects | Si (>=5.0) | Si | Si |
| Stored Procedures | Si (>=5.0) | Si (pl/pgSQL, pl/Perl, pl/TCL, pl/Python, pl/sh) | Si |
| Triggers | Si (>=5.0, rudimentario) | Si | Si |
| XML | Si | ¿ | ? |
| Interfaces de programación | ODBC, JDBC, C/C++, .NET/Mono, ADO.Net, OLEDB, Delphi, Perl, Python, PHP, Embedded (C precompilador), Embedded in Java (Conector/MXJ). | ODBC, JDBC, C/C++, Embedded SQL (in C), Tcl/Tk, Perl, Python, PHP | ODBC, JDBC, C/C++, PHP, Python, Perl, Kylix, Delphi, .NET/Mono, ADO |
| Transacciones | Si | Si | Si |
| Llaves Foráneas Constraints | Si | Si | Si |
| Replica | Si | NO | Comercial plugin available |
| Table spaces | Si | Si | ? |
| Cluster | Si | NO | NO |
| Hot Backups | Si | Si | Si |
| Administrador Visual de Herramientas | Si | Si | Si |



2.5. Características, Ventajas y Desventajas de Apache (tomcat), php y java script.

Apache.

Servidor web de código abierto. Desarrollado por Rob McCool, como tentativa de mejorar el servidor existente en el NCSA. El nombre Apache es un acrónimo de **patch Server** (servidor de remiendos), esto es un servidor construido con código preexistente, piezas y parches de código. Es capaz de utilizar otros interpretadores y lenguajes como "Tcl", "PHP" y "Python". Puede conectarse directamente a una Base de datos.

Posee diversos módulos que le permiten utilizar gran variedad de lenguajes y desarrollar funcionalidades avanzadas. Ofrece una instalación sencilla para sitios pequeños y si se requiere es posible expandirlo hasta el nivel de los mejores productos comerciales.

Apache Tomcat.

También llamado Jakarta Tomcat o Apache Tomcat funciona como un contenedor de servlets (objetos que se ejecutan en un servidor) desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de Java Server Pages (JSP) de Sun Microsystems.

Es un servidor web con soporte de servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache. Es usado como servidor autónomo en entornos con alto nivel de tráfico y alta disponibilidad.



Dado que fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java. Se mantiene y desarrolla por miembros de la **Apache Software Foundation** y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la **Apache Software Licence**.

Características:

Tomcat 5.x.

- Implementado a partir de las especificaciones Servlet 2.4 y JSP 2.0
- Recolección de basura reducida. Capa envolvente nativa para Windows y Unix para la integración de las plataformas. Análisis rápido JSP.

Tomcat 6.x.

- Implementado de Servlet 2.5 y JSP 2.1. Soporte para Unified Expression Language 2.1
- Diseñado para funcionar en Java SE 5.0 y posteriores. Soporte para Comet a través de la interfaz CometProcessor.

Ventajas:

- Es gratis, es fácil de instalar, se ejecuta en máquinas más pequeñas y es compatible con las API más recientes de Java. Puede descargarse, instalarse y probarse en menos de una hora.
- Ocupa muy poco espacio, teniendo su código binario un tamaño total de apenas 1 megabyte, de modo que no es raro que se ejecute rápido.
-



- Pone a disposición de todo el mundo las últimas actualizaciones de Java. Es fiable. Innumerables empresas utilizan Tomcat (aunque es imposible saber cuántas debido a la falta de licencia comercial). Su solidez se basa en que miles de desarrolladores contribuyen con código.
- Como Tomcat puede ejecutarse utilizando la JVM que se quiera, puede utilizarse JDK.
- Es compatible de forma integrada con el último IDE para Java de IBM, WebSphere Studio Application Development (WSAD) y su versión para el iSeries, WebSphere Development Studio Client (WDSC).

Desventajas:

- No cuenta con un servicio de soporte técnico para resolver los problemas. Sólo puede trabajar con un único servidor; no puede utilizarse en clusters.
- No soporta cambios de escala. No es compatible con muchas API's de J2EE.
- No soporta aplicaciones que realmente necesiten invocar de forma remota transacciones complejas con seguridad de nivel de objetos y de métodos.

PHP.

Es un lenguaje de programación intérprete usado normalmente para la creación de páginas web dinámicas. PHP es un acrónimo recursivo que significa "PHP Hypertext Pre-processor". El lenguaje PHP se emplea principalmente para las siguientes tareas:

- Programación de scripts por el lado del servidor. Programación de scripts en línea de comandos del sistema operativo sin necesidad de un servidor web.
- Desarrollo de aplicaciones multiplataforma.



PHP es una opción natural para los programadores en máquinas con Linux que ejecutan servidores web con Apache, pero funciona igualmente en cualquier otra plataforma como UNIX o Windows, con el software de Netscape o del web server de Microsoft. PHP también utiliza las sesiones de HTTP, conectividad de Java.

PHP se ejecuta en el servidor por lo que nos permite acceder a los recursos que tenga como por ejemplo una base de datos. El programa PHP es ejecutado en el servidor y el resultado es enviado al navegador. Este resultado es normalmente una página HTML pero también podría ser una página WML (Wap) como se despliega en la Figura 2.5.1



Figura 2.5.1 Ejecución de PHP

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que un navegador lo soporte, ya que es independiente del navegador.



Características:

- Código abierto: El código fuente del intérprete es accesible para permitir posibles mejoras o sugerencias acerca de su desarrollo. Dispone de extensiones para la conexión con la gran mayoría de los sistemas de gestión de bases de datos para el almacenamiento de información permanente en el servidor.
- Proporciona soporte a múltiples protocolos de comunicaciones en Internet (http, IMAP, FTP, LDAP, SNMP, etc.).
- Eficiente: consume muy pocos recursos en el servidor, por lo que con un equipo relativamente sencillo es posible desarrollar interesantes aplicaciones.
- Alta velocidad de desarrollo: permite realizar rápidamente sitios web dinámicos. Proporciona gran cantidad de extensiones útiles y bien documentadas que ahorran trabajo al programador.
- También dispone de facilidades para el procesamiento de archivos, funciones de tratamiento de textos, generación dinámica de imágenes, tratamiento de documentos XML, entre otras.
- Tiene gran variedad de módulos para crear una interface para una librería en particular, generando una API para esta.
- Contiene API's que vienen implementadas y permiten manejo de gráficos, archivos PDF, Flash, Cybercash, calendarios, XML, IMAP, POP, etc.

La última versión de PHP 5 utiliza el motor Zend-2 y presenta mejoras significativas y un entorno de programación orientado a objetos mucho más completo, que permite que PHP proporcione un alto rendimiento a las aplicaciones Web empresariales a nivel de las plataformas J2EE y .NET. PHP ha sido desarrollado inicialmente para entornos UNIX y es en este sistema operativo donde se aprovechan mejor sus prestaciones y consigue un mayor rendimiento.



Ventajas:

- Portable y multiplataforma. Corre en casi cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en por lo menos 25 plataformas, incluyendo diferentes versiones de Unix, Windows (95, 98, NT, ME, 2000, XP, etc.) y Macs. Esto permite que las aplicaciones PHP puedan ser portadas de una plataforma a otra sin necesidad de modificar ni una sola línea de código.
- Como en todos los sistemas, se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al Sistema operativo. La sintaxis de PHP es similar a la de C.
- Es completamente expandible. Está compuesto de un sistema principal (escrito por Zend), un conjunto de módulos y una variedad de extensiones de código.
- Contiene muchas interfaces distintas para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTPD. Otra alternativa es configurarlo como modulo CGI.
- Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, entre otros. Siempre se podrá disponer de ODBC para situaciones que lo requieran.
- Rapidez. Generalmente es utilizado como modulo de Apache, lo que lo hace extremadamente veloz. Esta completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.
- PHP es Open Source, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no se está forzado a pagar actualizaciones anuales para tener una versión que funcione.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.



- Perceptiblemente más fácil de mantener. Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

Desventajas:

- No posee una abstracción de base de datos estándar, sino bibliotecas especializadas para cada motor (a veces más de una para el mismo motor).
- No posee adecuado manejo de internacionalización, Unicode.
- Por su diseño dinámico no puede ser compilado y es muy difícil de optimizar. Por sus características promueve la creación de código desordenado y complejo de mantener.

Java Script.

Es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página Web.

Es un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, el más utilizado.



Portal de la Academia Regional de Informática de la DGETI.



No se tiene que declarar todas las variables, clases y métodos. La implementación de interfaces puede no realizarse, los métodos pueden ser públicos, privados o protegidos. Los tipos de variables, parámetros y funciones de retorno no se definen explícitamente.

Posee un modelo basado en el prototipo de objetos en lugar del modelo general basado en clases de objetos que provee herencia dinámica, esto es, que la herencia puede variar para objetos individuales.

Soporta funciones sin ningún requerimiento declarativo especial. Las funciones pueden ser propiedades de los objetos, al ejecutarse vagamente como tipos de métodos.

Se complementa con HTML y PHP en la programación de páginas Web. Lo interpreta el navegador en la máquina del usuario y no en el servidor, una de sus mejores características es que puede servir para controlar datos antes de ser enviados al servidor. Dispone de funciones para lanzar mensajes de aviso, de confirmación, de tratamiento de imágenes, de movimiento y posición del ratón.

Basada en la sintaxis de Java, pero con programación diferente. Una de sus principales virtudes es el control de datos que el usuario quiere enviar al servidor, antes que estos sean enviados. Es decir, cuando hay un formulario donde se puede enviar información al servidor, mediante javascript, podemos hacer controles de los datos para que sean de la forma que deseamos, produciendo un filtro previo que nos evita basura en las bases de datos del servidor.

Otra de sus funcionalidades es el tratamiento de imágenes y el manejo de mouse. La creación de algunos efectos funciona muy bien y rápidamente. Permite el tratamiento y gestión de las "cookies", estos son pequeños archivos que guardan información sobre datos introducidos por el usuario.



Características:

- Sintaxis similar a la usada en Java y C, al ser un lenguaje del lado del cliente este es interpretado por el navegador, no necesita tener instalado ningún Framework.
- Funciones. Propias del lenguaje y predefinidas por los usuarios Permite la programación orientada a objetos.
- Es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros.
- XMLHttpRequest es una de las funciones más importantes, que permite intercambiar datos asincrónicamente con el servidor web, puede ser mediante PHP, ASP, entre otros.
- Aunque Javascript sea soportado en gran cantidad de navegadores se puede elegir la opción de Activar/Desactivar el Javascript en los mismos.

Ventajas:

- Facilidad de aprendizaje; la mayoría de los navegadores lo soportan. Es compacto y sencillo por lo que no necesita compilarse ya que es interpretado por el navegador del cliente. La seguridad es casi total.

Desventajas:

- Tienen capacidades limitadas, por razones de seguridad, es necesario usarlo conjuntamente con otros lenguajes evolucionados, posiblemente más seguros, como Java. Dicha limitación es evidente al operar en el hardware del ordenador, como, por ejemplo, la fijación en automático de la resolución de vídeo o la impresión de un documento.



- El código es visible y puede ser leído por cualquiera, incluso si está protegido con las leyes del copyright. El código debe descargarse completamente antes de poderse ejecutar por lo que si los datos que un script utiliza son muchos, el tiempo que tardará en descargarse será muy largo, mientras que la interrogación de la misma base de datos en el servidor sería más rápida. Al ejecutar un script en un servidor limitado y con una continua solicitud por un mayor número de usuarios podría generar problemas en los tiempos de respuesta.

2.6 Arquitectura Cliente/servidor.

La arquitectura cliente/servidor, es una plataforma abierta. Las posibilidades de incluir distintos productos, aplicaciones o componentes de diferentes proveedores nos brinda la oportunidad de hacer una gran variedad de combinaciones de clientes y servidores. Pero esta gran variedad implica que se debe tener en cuenta, también una gran cantidad de elementos a considerar y evaluar al momento de proponer una solución basada en la arquitectura cliente/servidor.

Existen dos aspectos básicos en la problemática de la arquitectura cliente/servidor:

- ¿Qué plataforma elegir? Tiene relación con la respuesta a cuestiones aún más específicas como son: ¿Qué plataforma cliente elegir? ¿Qué plataforma servidor? ¿Qué clase de middleware? ¿Qué administrador o servidor de base de datos? ¿Sobre qué arquitectura de computación distribuida se tendrá que montar la solución? Esto mismo nos obliga a tener conocimientos importantes para la integración de las mismas, dado que el desarrollo de aplicaciones cliente/servidor requiere del manejo de elementos en el área de diseño de bases de datos, comunicación entre procesos, procesamiento de transacciones, generación de



GUI (interfaces gráficas de usuarios) y de Internet, con clientes y servidores distribuidos a lo largo de la Web.

- ¿Qué herramientas de desarrollo elegir? Tiene relación con la toma de decisiones sobre el área de desarrollo y herramientas de cliente/servidor.

Esta tecnología ofrece beneficios en cuanto a que permite y promueve el establecimiento de sistemas abiertos, esto mismo le otorga un alto grado de complejidad, característica propia de los sistemas abiertos, con una gran cantidad de componentes que no siempre tienen un compromiso firme o permanente entre sí, a diferencia de los sistemas propietarios.

Así que las decisiones tomadas respecto del diseño deben ser las adecuadas, de otra manera resultan más críticas con relación a desarrollos tradicionales, lo que obliga siempre a manejarlos en forma mucho más coherente y a tener un gran dominio de herramientas para lograr una adecuada implantación y explotación.

Definición del modelo cliente/servidor.

Es un concepto que sirve para describir las comunicaciones entre procesos que se clasifican como consumidores de servicios (clientes) y proveedores de servicios (servidores).

Se puede definir el modelo cliente/servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aun en entornos multiplataforma.

En el modelo cliente/servidor, (Fig. 2.6.1) el cliente envía un mensaje solicitando un determinado servicio a un servidor, y este envía uno o varios mensajes con la respuesta. En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras.

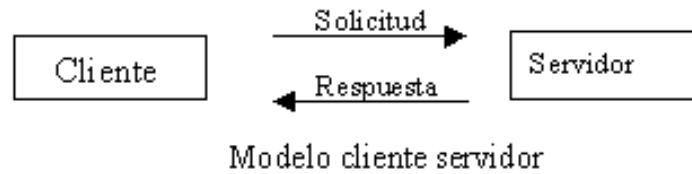


Fig. 2.6.1. Arquitectura Cliente /servidor.

Tanto los clientes como los servidores son entidades independientes que operan conjuntamente a través de una red para realizar una tarea; en la siguiente lista se presentan una serie de características que debe cumplir la arquitectura cliente/servidor:

- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.
- Existe una clara distinción de funciones basada en el concepto de "servicio", que se establece entre clientes y servidores.
- La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos.
- Los clientes corresponden a procesos activos en cuanto a que son éstos los que hacen peticiones de servicios a los servidores. Estos últimos tienen un carácter pasivo ya que esperan las peticiones de los clientes.
- No existe otra relación entre clientes y servidores que no sea la que se establece a través del intercambio de mensajes entre ambos. El mensaje es el mecanismo para la petición y entrega de solicitudes de servicio.
- Las plataformas de software y hardware entre clientes y servidores son independientes. Una ventaja principal de esta arquitectura es la posibilidad de conectar clientes y servidores independientemente de sus plataformas.



- El concepto de escalabilidad tanto horizontal y vertical es aplicable a cualquier sistema cliente/servidor. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

Componentes del modelo Cliente/Servidor.

La arquitectura cliente/servidor es un modelo basado en la idea del servicio, en el que el cliente es un proceso consumidor de servicios y el servidor es un proceso proveedor de servicios. Además esta relación está establecida en función del intercambio de mensajes que es el único elemento de acoplamiento entre ambos. De lo anterior se desprenden los tres elementos fundamentales sobre los cuales se desarrollan e implantan los sistemas cliente/servidor:

- El proceso cliente que es quien inicia el diálogo
- El proceso servidor que pasivamente espera a que lleguen peticiones de servicio.
- El middleware que corresponde a la interfaz que provee la conectividad entre el cliente y el servidor para poder intercambiar mensajes.

Cliente. Proceso que permite al usuario formular los requerimientos y pasarlos al servidor, se le conoce con el término front-end. Normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de la red. El front-end es el cliente y en modernos entornos es un navegador Web.

Las funciones que lleva a cabo el proceso cliente son:



Portal de la Academia Regional de Informática de la DGETI.



- Iniciar solicitudes o peticiones, por tanto tiene un papel activo en la comunicación (dispositivo maestro o amo). Administrar la interfaz de usuario.
- Interactuar con el usuario. Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos. Esperar y recibir resultados del servidor. Por lo general, puede conectarse a varios servidores a la vez.

Servidor. Proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Conocido también con el término back-end. Normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos.

El back-end es una colección de recursos de información y cálculo utilizados por una empresa o institución: sistemas de gestión de bases de datos, contabilidad, almacén procesos de transacciones, recursos humanos etc. Las funciones que lleva a cabo el proceso servidor se presenta en los siguientes puntos:

- Esperar a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación (dispositivo esclavo). Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- Aceptar los requerimientos de bases de datos que hacen los clientes. Procesar requerimientos de bases de datos. Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.
- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado). Casi no hay interacción directa con los usuarios finales.



Middleware. Es la interfaz que provee la conectividad entre aplicaciones clientes, servidores y entre aplicaciones y bases de datos.

Es una capa de software que protege a los desarrolladores de tener que manejar detalles de bajo nivel de diferentes protocolos de comunicación, sistemas operativos y arquitecturas de bases de datos. Este tipo de interfaces incluyen API's, PRC's, Pipes, mensajería de red y accesos a bases de datos.

Para entender en forma más ordenada y clara los conceptos y elementos involucrados en el modelo cliente/servidor se puede aplicar una descomposición o arquitectura de niveles. Esta descomposición principalmente consiste en separar los elementos estructurales de esta tecnología en función de aspectos más funcionales de la misma:

- **Nivel de Presentación.** Agrupa a todos los elementos asociados al componente Cliente.
- **Nivel de Aplicación.** Agrupa a todos los elementos asociados al componente Servidor.
- **Nivel de comunicación.** Agrupa a todos los elementos que hacen posible la comunicación entre los componentes Cliente y servidor.
- **Nivel de base de datos.** Agrupa a todas las actividades asociadas al acceso de los datos.

Este modelo de descomposición en niveles, permite introducir más claramente la discusión del desarrollo de aplicaciones en arquitecturas de hardware y software en planos.



Clasificación de modelos Cliente/Servidor.

Es importante conocer los componentes cliente/middleware/servidor, pero también es preciso analizar ciertas relaciones entre éstos, que pueden definir el tipo de solución que se ajusta mejor a los requerimientos de información que se obtuvieron en la etapa de análisis de un determinado proyecto. El analista o líder deberá conocer los eventos/restricciones del negocio para, a partir de allí, hacer las consideraciones y estimaciones de la futura configuración, teniendo en cuenta aspectos como por ejemplo, la oportunidad de la información, tiempo de respuesta, tamaños de registros, tamaño de bases de datos, estimaciones del tráfico de red, distribución geográfica tanto de los procesos como los datos, etc.

Por lo anterior se presenta una clasificación basado en los conceptos de Fat Client/Thin Client, Fat Server/Thin Server, Two Tier, Three Tier, los cuales están generalizados, pero que se consideran necesarios y útiles para la aplicación del modelo cliente/servidor.

Clasificación por tamaño de componentes.

Este tipo de clasificación se basa en los grados de libertad que brinda el modelo cliente/servidor para balancear la carga del proceso entre los niveles de presentación, aplicación y base de datos. Dependiendo del segmento de las capas que el software tenga que soportar, para la mayor o menor carga de procesamiento, se habla de Fat Cliente (Thin Server) o Fat server (Thin Client). Consideraciones de este tipo son importantes al momento de decidir una plataforma de desarrollo/explotación, al punto que pueden definir la viabilidad o no de las mismas para enfrentar un cierto número de restricciones impuestas por una problemática a resolver.



Fat Client (Thin Server).

En este esquema de arquitectura el grueso de la aplicación es ejecutada en el cliente, es decir, el nivel de presentación y de aplicación corren en un único proceso cliente, y el servidor es relegado a realizar las funciones que provee un administrador de base de datos. En general tiene mejor aplicación en sistemas de apoyo de decisiones (DSS: Decision Support System) y sistemas de información ejecutiva (EIS: Executive Information System), tiene pocas posibilidades de aplicarse en sistemas de misión crítica.

Fat Server (Thin Client).

El proceso cliente es restringido a la presentación de la interfaz de usuario, mientras que el grueso de la aplicación corre por el lado del servidor de aplicación. Este tipo de arquitectura presenta una flexibilidad mayor para desarrollar un gran espectro de aplicaciones, incluyendo los sistemas de misión crítica a través de servidores de transacciones.

Clasificación por planos o capas (Tier).

Se basan en la idea de capas o planos (tier), la cual es una variación sobre la división o clasificación por tamaño de componentes (clientes grandes y servidores amplios). Esto se debe a que se trata de definir el modo en que las prestaciones funcionales de la aplicación serán asignadas, y en qué proporción, tanto al cliente como al servidor.

Dichas prestaciones se deben agrupar entre los tres componentes clásicos para el modelo cliente/servidor:



- Interfaz de usuario
- Lógica de negocios
- Datos compartidos

Cada uno de los cuales corresponde a un plano. Dentro de esta categoría tenemos las aplicaciones en dos planos (two-tier), tres planos (three-tier) y multi planos (multi-tier).

Dado que este término ha sido sobrecargado de significados por cuanto se le utiliza indistintamente para referirse tanto a aspectos lógicos (Software) como físicos (Hardware), aquí se esquematizan solo el primer aspecto.

Planos a niveles de software.

Este enfoque o clasificación es el más generalizado y el que más se ajusta a los enfoques modernos, dado que se fundamenta en los componentes lógicos de la estructura cliente/servidor y en la madurez y popularidad de la computación distribuida.

Cliente/Servidor Dos Capas o Planos.

Esta estructura se caracteriza por la conexión directa entre el proceso cliente y un administrador de bases de datos. Dependiendo de donde se localice el grupo de tareas correspondientes a la lógica de negocios se pueden tener a su vez dos tipos distintos dentro de esta misma categoría:



1. Implementado con SQL Remoto.

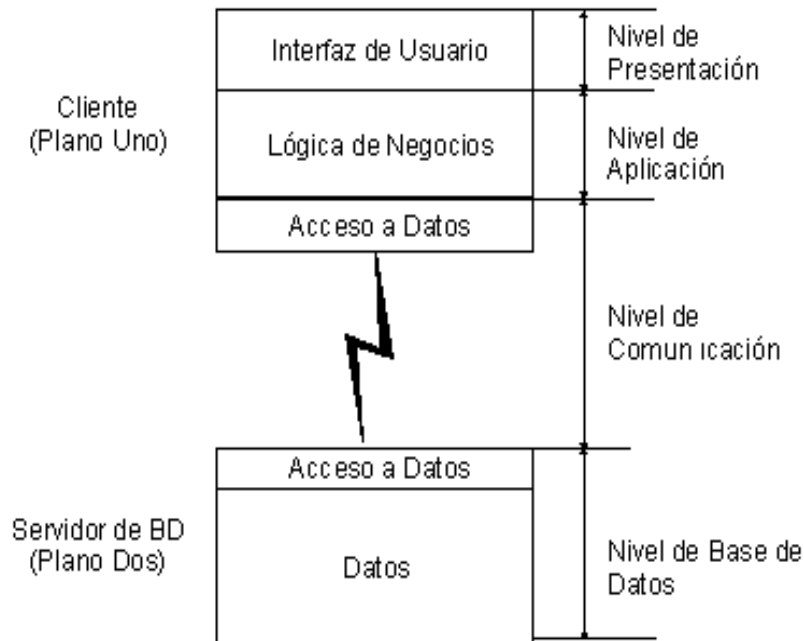


Figura. 2.6.2 Arquitectura dos capas. Con la interfaz y la lógica del negocio juntas.

En este esquema (Fig. 2.6.2) el cliente envía mensajes con solicitudes SQL al servidor de bases de datos y el resultado de cada instrucción SQL es devuelto por la red, no importando si son uno, diez, cien o mil registros. Es el mismo cliente quien debe procesar todos los registros que le fueron devueltos por el servidor de base de datos, según el requerimiento que él mismo hizo.

Esto hace que este tipo de estructura se adecue a los requerimientos de aplicaciones orientadas a los sistemas de apoyo y gestión, pero resultan inadecuados para los sistemas críticos en que se requieran bajos tiempos de respuesta.



Ventajas:

- Presenta una estructura de desarrollo bastante simple por cuanto el programador típicamente maneja un único ambiente de desarrollo (es más simple respecto de cliente/servidor en tres planos, puesto que reduce una capa de programación, como se verá más adelante).

Desventajas:

- La gran cantidad de información que se envía al cliente congestiona demasiado el tráfico de red, lo que se traduce en bajo rendimiento.
- Por su bajo rendimiento esta estructura tiene un bajo espectro de aplicación, limitándose a la construcción de sistemas no críticos.

2. Implementado con Procedimientos Almacenados.

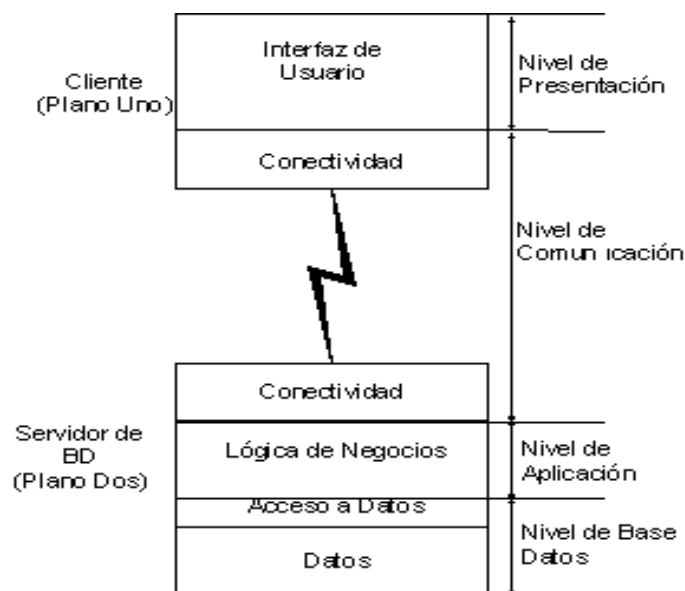


Fig.2.6.3 Arquitectura dos capas. Con acceso a la base de datos y la lógica del negocio juntos.



En este esquema (Fig.2.6.3) el cliente envía llamadas a funciones que residen en la base de datos, y es ésta quien resuelve y procesa la totalidad de las instrucciones SQL agrupadas en la mencionada función.

Ventajas:

- Presenta las mismas ventajas de una arquitectura dos planos con procedimientos almacenados, pero mejora considerablemente el rendimiento sobre ésta, dado que reduce el tráfico por la red al procesar los datos en la misma base de datos, enviando sólo el resultado final de un conjunto de instrucciones SQL.

Desventajas:

- Si bien la complejidad de desarrollo se ve disminuida, se pierde flexibilidad y escalabilidad en las soluciones implantadas (especialmente respecto de cliente/servidor en tres planos).
- Obliga a basar el grueso de la aplicación en SQL extendido, propios del proveedor de la base de datos que se elija. Debiera considerarse que sí bien los procedimientos almacenados (stored procedures), los desencadenantes (triggers) y las reglas (constraint) son útiles, en rigor son ajenos al estándar de SQL. No existen dos implementaciones de proveedores iguales.
- El lenguaje para la descripción de los procedimientos almacenados y probablemente su funcionalidad varía de un proveedor a otro. Lo que implica que los procedimientos almacenados no son totalmente exportables entre plataformas de distintos proveedores.
- Se pierde la independencia entre el código de la aplicación (conocimiento y reglas del negocio) y los datos.



Cliente/Servidor de Tres Planos.

Esta estructura se caracteriza por elaborar la aplicación en base a dos capas principales de software, más la capa correspondiente al servidor de base de datos. Al igual que en la arquitectura dos capas, y según las decisiones de diseño que se tomen, se puede balancear la carga de trabajo entre el proceso cliente y el nuevo proceso correspondiente al servidor de aplicación.

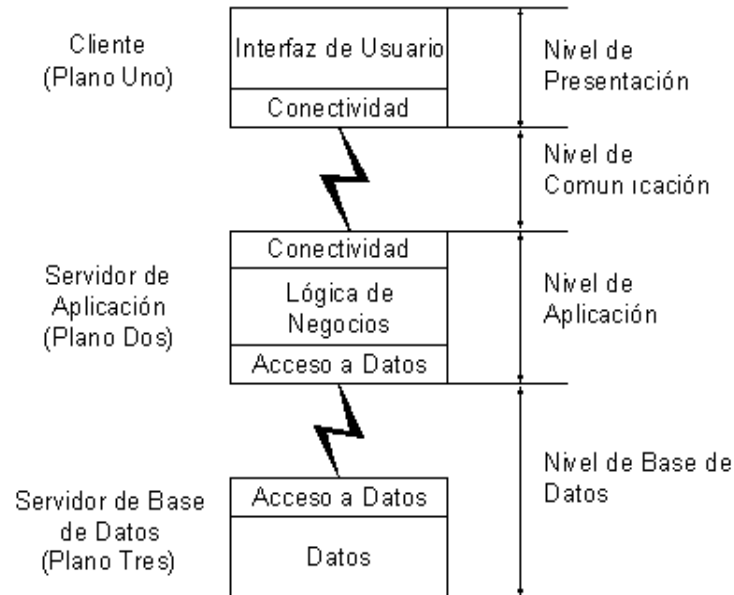


Figura. 2.6.4 Arquitectura de aplicaciones de tres capas.

En este esquema el cliente envía mensajes directamente al servidor de aplicación, el cual debe administrar y responder todas las solicitudes. Es el servidor, dependiendo del tipo de solicitud, quien accede y se conecta con la base de datos.



Ventajas:

- Reduce el tráfico de información en la red por lo que mejora el rendimiento de los sistemas (especialmente respecto de la estructura en dos planos).
- Brinda una mayor flexibilidad de desarrollo y de elección de plataformas sobre la cual montar las aplicaciones. Provee escalabilidad horizontal y vertical.
- Se mantiene la independencia entre el código de la aplicación (reglas y conocimiento del negocio) y los datos, mejorando la portabilidad de las aplicaciones.
- Los lenguajes sobre los cuales se desarrollan las aplicaciones son estándares lo que hace más exportables las aplicaciones entre plataformas.
- Dado que mejora el rendimiento al optimizar el flujo de información entre componentes, permite construir sistemas críticos de alta confiabilidad.
- El mismo hecho de localizar las reglas del negocio en su propio ambiente, en vez de distribuirlos en la capa de interfaz de usuario, permite reducir el impacto de hacer mantenimiento, cambios urgentes de última hora o mejoras al sistema.
- Disminuye el número de usuarios (licencias) conectados a la base de datos.

Desventajas:

- Dependiendo de la elección de los lenguajes de desarrollo, puede presentar mayor complejidad en comparación con cliente/servidor de dos planos. Existen pocos proveedores de herramientas integradas de desarrollo con relación al modelo cliente/servidor de dos planos, y normalmente son de alto costo.
- Debido a estas desventajas es aquí la mayor importancia del Generador (aplicación creada en el presente trabajo).



La clasificación de la arquitectura cliente/servidor en "planos a niveles de hardware", define la configuración física más conveniente sobre la cual montar una determinada aplicación, la cual es estructurada de acuerdo a las consideraciones del modelo lógico o "planos a niveles de software".

Clasificación por la naturaleza del servicio.

Servidores de Bases de Datos.

La creación de aplicaciones cliente/servidor está asociada a la utilización de servidores de bases de datos relacionales SQL, y dependiendo de los requerimientos y restricciones se debe elegir entre una arquitectura de dos o tres planos. Pero una arquitectura centrada en un servidor de bases de datos, cualquiera de las modalidades de dos planos, permite que un proceso cliente solicite datos y servicios directamente a un servidor de bases de datos.

Según las distintas normas de SQL (SQL-89, SQL-92, SQL3) el servidor debe proveer un acceso compartido a los datos con los mecanismos de protección necesarios, así como proveer mecanismos para seleccionar resultados dentro de un conjunto de datos, posibilitando un ahorro en procesos de comunicación. El servidor debe también proveer el mecanismo de concurrencia, seguridad y consistencia de datos, basado principalmente en el concepto de transacción en el que todo se realiza, y por lo tanto se hace permanente, o todo fallará, anulándose la transacción en tal caso.

Los servidores de bases de datos actuales son una mezcla de SQL estándar más otras extensiones propias de cada proveedor.



Por ejemplo casi todas las bases de datos están provistas con procedimientos almacenados (Stored procedures), desencadenantes (Triggers) y restricciones (Constraints) pero presentan diferencias importantes en su implementación.

Procedimientos Almacenados.

Una de las posibilidades de implementar de mejor forma un sistema cliente/servidor en dos planos (two-tier), sin olvidar todas sus restricciones y limitaciones, es a través de procedimientos almacenados, que son funciones que agrupan un conjunto de instrucciones y lógica de procedimientos SQL, los cuales son compilados y almacenados en la misma base. Ya se hizo notar que con estos mecanismos los proveedores de bases de datos introducen la lógica de negocios dentro de su servidor propietario; lo lógico sería mantener una independencia entre datos y códigos.

El rol principal de los procedimientos almacenados es proveer la parte servidora de la lógica de una aplicación cliente/servidor, es decir vendría a reemplazar al servidor de aplicaciones en una arquitectura de tres planos (Three-tier). Si bien los procedimientos almacenados permiten a un servidor brindar servicios aptos para OLTP (On Line Transaction Processing), no se compara con los rendimientos alcanzados por una arquitectura en tres planos con TP pesado.

Desencadenantes: Son mecanismos que permiten realizar acciones automáticamente sobre los datos, las cuales están asociadas a algún evento definido. Normalmente son implementados a través de procedimientos almacenados. Los eventos a los cuales se hace referencia están asociados a las actualizaciones de tablas mediante sentencias delete, insert o update, y son llamados implícitamente al suceder cualquiera de estos eventos, a diferencia de los procedimientos almacenados que son llamados explícitamente por un proceso cliente.



Restricciones: Al igual que los desencadenantes, son acciones que se realizan asociadas a algún evento determinado y están orientadas a llevar a cabo validaciones más simples de datos. Los tipos de eventos son los mismos que para los desencadenantes.

Los proveedores de bases de datos optan también por introducir lenguajes de procedimientos para SQL, con lo que se supone que la lógica de negocios (el servidor dentro del modelo cliente/servidor) se haga más exportable, permitiendo la migración de la aplicación a cualquier plataforma sobre la cual corre la base de datos. Ejemplos de estos lenguajes de procedimientos para SQL son: PL/SQL (Oracle), Ingres/4GL (Ingres), SPL (Informix), Transac SQL (Sybase).

Servidores de transacciones.

Estos tipos de sistemas se pueden implementar con cualquiera de las modalidades cliente/servidor en dos o tres planos, pero incorporan un elemento principal sobre el cual se elabora y basa toda la fortaleza de este modelo, el concepto de transacción.

Tal cual se explica mediante los conceptos de planos, tamaños de componentes y servidores de bases de datos, con un servidor de transacciones el proceso cliente llama a funciones, procedimientos o métodos que residen en el servidor, ya sea que se trate de un servidor de bases de datos o un servidor de aplicaciones. Lo importante es que el intercambio a través de la red se realiza mediante un único mensaje de solicitud/respuesta, es decir, independientemente de que se necesite ejecutar una o más funciones, una o más instrucciones o sentencias SQL, éstas son agrupadas en una unidad lógica llamada transacción; evitando así el intercambio a través de la red de un mensaje solicitud/respuesta por cada sentencia SQL, el cual es el caso de los sistemas cliente/servidor de dos planos, implementados a través de SQL remoto. Estas



aplicaciones denominadas OLTP (On Line Transaction Processing) están orientadas a dar soporte a los procedimientos y reglas de los sistemas de misión crítica.

Existe una estrecha relación, o equivalencia, entre las clasificaciones por tamaño de componente y por capas, dado que se trata de una misma idea, en la que todo se reduce a la forma en que la lógica del negocio se distribuye entre los niveles lógicos del modelo: presentación, aplicación y base de datos. En el caso de sistemas de dos planos, la lógica de negocios se puede distribuir en el cliente (Fat Client/Thin Server), en la base de datos (Thin Client/Fat Server), o una combinación de ambos. En los sistemas de tres planos en cambio, la lógica de la aplicación se localiza normalmente en el servidor de aplicaciones, lo que lo convierte en un esquema (Thin Client/Fat Server).

Consideraciones para herramientas de desarrollo cliente/servidor.

En la segunda generación de herramientas de desarrollo se incorporó la separación del nivel de presentación (interfaz de usuario) de la lógica de negocios, esto es, una arquitectura de tres planos. Las únicas opciones válidas, de acuerdo a los requerimientos de desarrollo y explotación de sistemas críticos, son la implantación de soluciones cliente/servidor en dos planos a través de la utilización de procedimientos almacenados, o la estructura cliente/servidor en tres planos. Ambos modelos están presentes en el mercado a través de herramientas que apoyan las etapas de análisis, diseño y construcción de aplicaciones cliente/servidor y representan mejoras a lo largo de todo el ciclo de desarrollo, principalmente a través de la incorporación de procedimientos estándares, además de documentación desde las primeras etapas de vida de un proyecto.

La elección de dos planos presenta ventajas de simplicidad en el ciclo de desarrollo, pero difícilmente en el mantenimiento, pero los mecanismos de bases de datos pueden realizar funciones de servidores más allá del modelo de datos relacional, a la vez que



Portal de la Academia Regional de Informática de la DGETI.



obliga a depender demasiado de un proveedor de base de datos y de lenguajes propietarios (SQL extendido), lo que puede dificultar la exportación de las aplicaciones a otras plataformas. La elección de tres planos brinda más grados de libertad en cuanto a la disponibilidad de plataformas, pero debe tenerse en cuenta que las decisiones en la elección del software de base, si no son correctas, pueden hacer más compleja la implementación de una solución.

Como recomendación técnica se propone como mejor alternativa la estructura cliente/servidor en tres planos para aplicaciones de alto rendimiento, como son los sistemas críticos, dadas sus características de flexibilidad, escalabilidad y probado rendimiento.



CAPÍTULO III



**Portal de la Academia Regional de Informática de la
DGETI.**





3. Definición y Análisis del Problema.

3.1 Problemática actual.

Un proyecto relevante de la Academia Regional de Informática consiste en involucrar las diferentes áreas existentes en la institución para el desarrollo de trabajos de investigación y de las estrategias de enseñanza a través de una interacción con la WEB. Siendo este en la actualidad uno de los medios más importantes como medio de comunicación y difusión de información a distancia, la cual permite tener una comunicación con un gran número de individuos en un corto tiempo. Aunque la comunicación para muchos es impersonal no deja de ser una herramienta útil y eficiente. Actualmente La Academia Regional de Informática, no cuenta con un portal que le permita realizar el intercambio de la información de una manera fácil y eficiente.

La DGETI en la zona regional del Distrito federal está conformada por 34 CETIS, en los cuales ciertos profesores de cada dependencia se han integrados a la Academia Regional de Informática; está a su vez está conformada dirigida por un presidente de academia. Una de sus principales funciones que está academia desarrolla consiste en informar de las actividades a realizar durante el ciclo escolar, así como dar un seguimiento al realizar cambios y adecuaciones pertinentes, e informar en el menor tiempo posible de cada acto. Sin embargo la información llega a destiempo, y en ocasiones se genera en el momento, teniendo como resultado que no todas las instituciones les llegue la información en tiempo y forma.

Por tal motivo, no se ha podido optimizar e incluso dar calidad en los proyectos o actividades de la especialidad, por los tiempos y el mal manejo de la información.



Como podemos observar en la Figura 3.1.1 el procedimiento que se sigue actualmente, es muy largo, el Presidente General de Academia entrega personalmente a la DGETI un oficio solicitando una fecha de reunión.

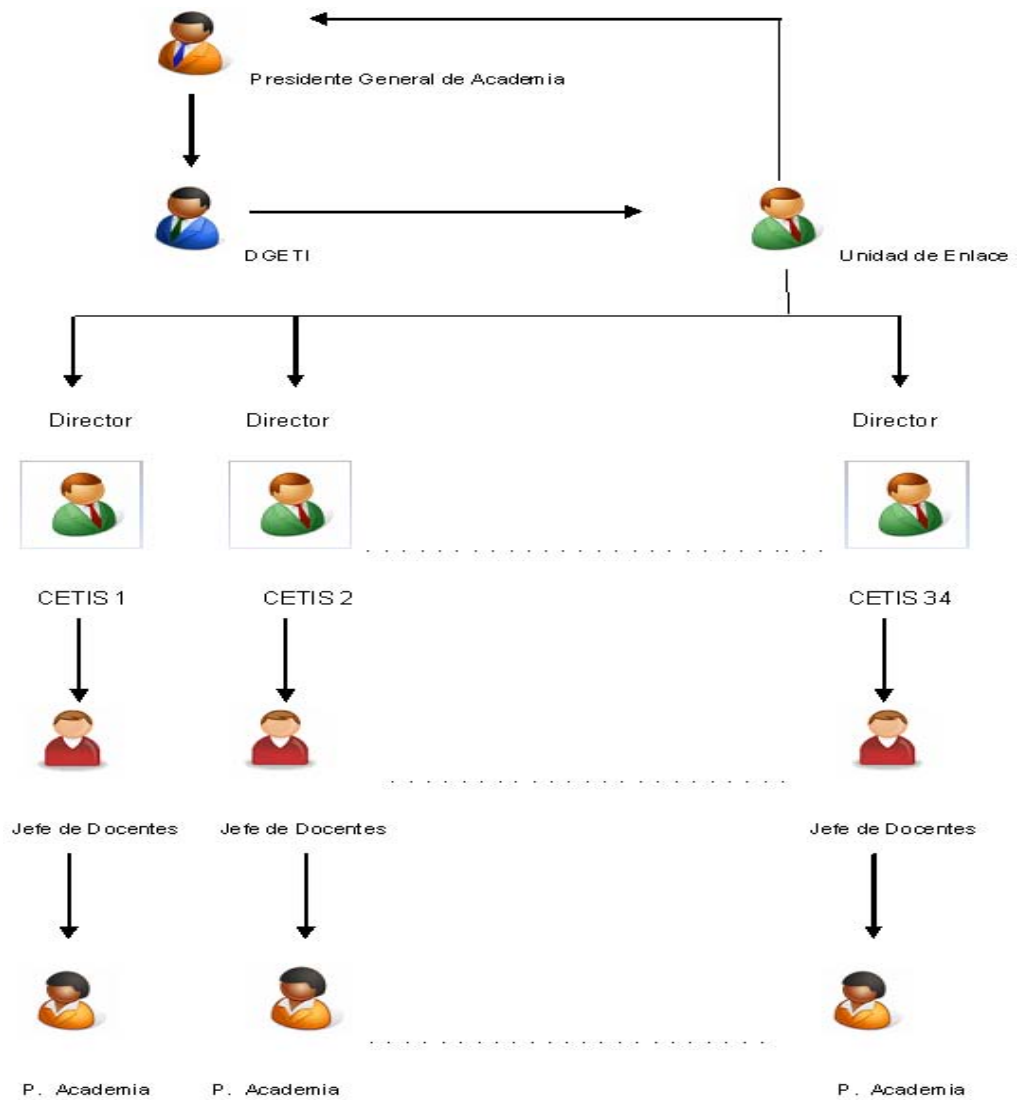


Figura 3.1.1 Proceso Antes de Reunión.



Capítulo III. Definición y Análisis del Problema.



La DGETI recibe el oficio y propone una fecha, respondiendo con otro oficio el cual entrega a su Unidad de Enlace, para que este sea el encargado de hacerle llegar tanto al Presidente General de Academia como a los directores de cada uno de los CETIS su respuesta. Este envío de información se hace por mensajería, personalmente o vía telefónica, ocasionando que la misma llegue a destiempo. La problemática presentada en este punto es que entre el envío y contestación del oficio se lleva aproximadamente 2 semanas.

Una vez girado el documento a los directores de los CETIS, estos turnan la información a su Jefe de Docentes para que este a su vez se lo haga llegar al Presidente de Academia. En este punto el tiempo del proceso de generar oficios y entregar los mismos es aproximadamente 1 semana.

Posteriormente cuando llega el día de la reunión el Presidente General de Academia, entrega una orden del día con los temas a tratar, se desarrollan cada uno de los puntos en cuestión y al terminar dicha reunión generan una minuta de los acuerdos generados y se le entrega una copia a cada uno de los participantes que asistieron; esto en un periodo de 5 días hábiles después de realizada la reunión. Ver Figura 3.1.2

A lo largo de este proceso uno de los problemas que se presentan, es que los asistentes desconocen los temas a tratar y la manera en que se entrega la minuta ya que si uno de los Presidentes de Academia no asistió no recibe dicha minuta y tiene que llamar a sus colegas para que le informen los acuerdos a los que concluyeron.

El tiempo en que la persona interesada localiza a sus colegas que hayan asistido y posean la minuta, pueden transcurrir en el mejor de los casos 3 días.

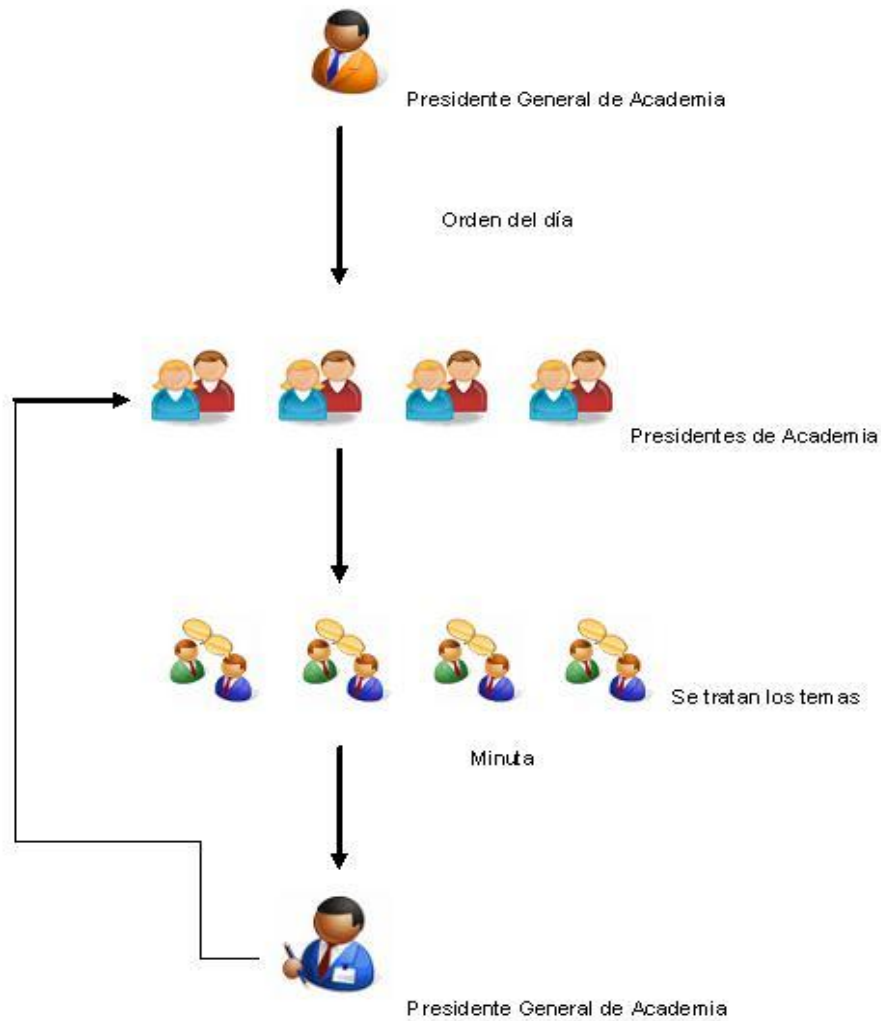


Figura 3.1.2. Durante Reunión.

Como conclusión podemos decir que si todo el proceso va bien y sin retrasos estamos hablando de entre 4 a 6 semanas entre organización, orden del día, acuerdos, repartición de minuta y próxima reunión, por lo cual se desea reducir los tiempos y la vía de comunicación. Ver figura 3.1.3.

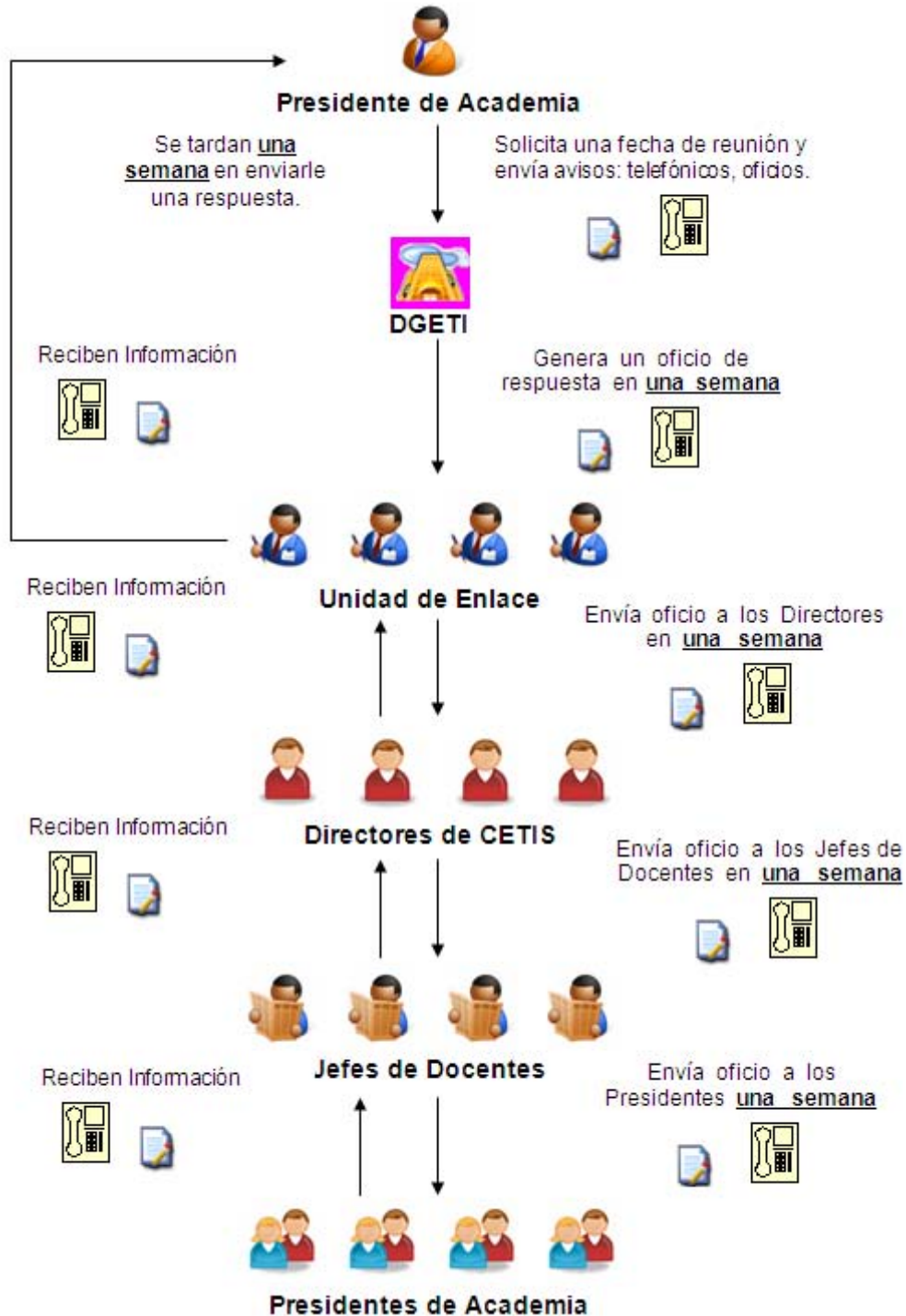


Figura 3.1.3 Problemática Real, Tiempos de Generación/Recepción de Oficios.



3.2. Requerimientos Generales y Particulares del Sistema.

Para tener un panorama de los requerimientos generales con los cuales contara el sistema, nos basaremos en la Ingeniería de Software o Ingeniería de Requerimientos.

La Ingeniería de Requerimientos (IR) cumple un papel primordial en el proceso de producción de software, ya que se enfoca en un área fundamental: la definición de lo que se desea producir. Los requerimientos pueden dividirse en dos tipos:

- Funcionales. Define las funciones que el sistema será capaz de realizar, es decir describen las transformaciones que el sistema realiza sobre las entradas para producir las salidas.
- No Funcionales. Tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc.

Las características de un requerimiento son las propiedades principales. Aquí se presentan las más importantes:

- Necesario. Si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto o del proceso.
- Conciso. Esto es si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.
- Completo. No necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.



Capítulo III. Definición y Análisis del Problema.



- Consistente. Que no exista contradicción con otro requerimiento.
- No ambigüedad. Cuando tiene una sola interpretación. El lenguaje usado en su definición, no debe causar confusiones al lector.
- Verificable. Cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

Los principales beneficios obtenidos al llevar a cabo estos requerimientos son:

- Permite gestionar las necesidades del proyecto en forma estructurada. Cada una de las actividades consiste en una serie de pasos organizados y bien definidos.
- Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados, es decir proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.
- Disminuye los costos y retrasos del proyecto, muchos estudios han demostrado que reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente caro.
- Mejora la calidad del software, esto tiene que ver con cumplir un conjunto de requerimientos (funcionalidad, facilidad de uso, confiabilidad, desempeño, etc.).
- Mejora la comunicación entre equipos, es decir representa una forma de consenso entre clientes y desarrolladores. Si este consenso no ocurre, el proyecto no será exitoso.
- Evita rechazos de usuarios finales, debido a que en cierta forma se obliga al cliente a considerar sus requerimientos cuidadosamente y revisarlos dentro del marco del problema, por lo que se le involucra durante todo el desarrollo del proyecto.



Realmente, son muchas las personas involucradas en el desarrollo de los requerimientos de un sistema. Los roles más importantes pueden clasificarse como sigue:

- Usuario final. Son las personas que usarán el sistema desarrollado. Están relacionados con la disponibilidad y la confiabilidad del sistema; están familiarizados con los procesos específicos que debe realizar el software, dentro de los parámetros de su ambiente laboral. Serán quienes utilicen las interfaces y los manuales de usuario.
- Usuario Líder. Son los individuos que comprenden el ambiente del sistema o el dominio del problema en donde será empleado el software desarrollado. Ellos proporcionan al equipo técnico los detalles y requerimientos de las interfaces del sistema.
- Personal de Mantenimiento. Para proyectos que requieran un mantenimiento eventual, estas personas son las responsables de la administración de cambios, de la implementación y resolución de anomalías. Su trabajo consiste en revisar y mejorar los procesos del producto ya finalizado.
- Analistas y programadores. Son los responsables del desarrollo del producto en sí; ellos interactúan directamente con el cliente.
- Personal de pruebas. Se encargan de elaborar y ejecutar el plan de pruebas para asegurar que las condiciones presentadas por el sistema son las adecuadas. Son quienes van a validar si los requerimientos satisfacen las necesidades del cliente.
- Otras personas que pueden estar involucradas, dependiendo de la magnitud del proyecto, pueden ser: administradores de proyecto, documentadores, diseñadores de base de datos, entre otros.



Requerimientos Generales.

- Elaborar un portal que se acceda vía WEB, que contenga toda la información relevante de la Academia Regional de Informática, que permita informar a todos sus miembros de las actividades que se están desarrollando o que se están programando para futuras reuniones ser analizadas y discutidas, y llegar a posibles acuerdos.
- Se requiere que esté portal cuente con toda la información actualizada referente a las actividades próximas a desarrollarse y que los miembros puedan tomar las consideraciones necesarias para llevar a buen término estas actividades.
- Se deberá analizar, desarrollar e implantar en software de distribución gratuita, para evitar la inversión en software propietario y con esto, evitar gastos en el pago de licencias.
- El sitio deberá ser administrado y actualizado por el personal designado por la DGETI en acuerdo con la academia.
- En el portal se pondrá la información que se genera en la academia de informática, brindando un servicio a nivel nacional de las actividades que se realizarán durante el semestre lectivo, con respecto a cursos, exposiciones, trabajos de investigación elaboración de secuencias y materiales didácticos, publicación de artículos e intercambio bibliográfico.
- El portal estará regido bajo los esquemas sistemáticos de factibilidad, funcionalidad y operatividad, de tal manera que sea atractivo y de interés para los docentes y que coadyuve en su práctica cotidiana laboral.
- Es así que el portal, cubrirá la necesidad del intercambio de información de los diferentes planteles a nivel nacional almacenando los datos en una base de datos versátil que cubra los requerimientos del sistema.



Requerimientos Particulares.

- Ingreso de artículos para publicar. Fechas de reuniones y acuerdos. Información de exposiciones, cursos, etc.
- Secuencias didácticas. Material didáctico. Trabajos de investigación. Subir material de eventos académicos.
- Implantación de un portal integral. Modernizar las actividades que realiza la academia.
- Utilizar la tecnología que tiene disponible la academia de informática para otros asuntos.
- Involucrar a que el personal se entere de las actividades que se desarrollan durante el ciclo escolar.
- Intercambiar información y puntos de vista sobre actividades relacionadas a nivel de Docentes para una misma materia.
- Seleccionar el mejor material didáctico que se pueda recopilar a través de acuerdos de academia.
- Información oportuna de las actividades de la academia. Brindar material útil y actual a los docentes del área de informática.
- El Presidente de Academia requiere un enlace en el cual le permita hacer las solicitudes de reunión y pueda ver la respuesta en un mínimo de tiempo.

3.3 Clasificación y organización de la información a presentar en el sitio Web.

Un sitio Web no funcionará adecuadamente si no se organiza sobre fundamentos sólidos y lógicos, aunque los contenidos básicos sean precisos, atractivos, y estén bien descritos. La manera en que la gente busca y usa la información de referencia,



Capítulo III. Definición y Análisis del Problema.



sugiere que las unidades de información sean pequeñas y acotadas ya que son más funcionales y más fáciles de explorar que los textos largos y uniformes.

Existen cinco pasos básicos en la organización de la información:

- Dividir el contenido en unidades lógicas.
- Establecer una jerarquía de la importancia entre las unidades.
- Utilizar la jerarquía para estructurar relaciones entre unidades.
- Construir un sitio de acuerdo con la estructura de información propuesta.
- Analizar el éxito funcional y estético de su sistema.

Segmentar la información.

La mayor parte de la información que se encuentra en la Web (World Wide Web) se recopila en documentos de referencia cortos que pueden ser leídos de forma no secuencial. Esto se observa en aquellos sitios Web donde los contenidos son en su mayor parte técnicos o administrativos. Esta manera de presentar la información es útil en la Web por varias razones:

- Pocos usuarios de Web pasan tiempo leyendo largos párrafos del texto en pantalla. La mayoría de los usuarios guarda e imprime los documentos para poder leerlos más cómodamente.
- Los bloques de información no muy extensos se prestan a convertirse en enlace Web. El usuario de un enlace Web espera generalmente encontrar una unidad específica y relevante de información, y no un volumen exagerado de contenidos generales. Pero tampoco hay que subdividir excesivamente la información o el usuario puede cansarse. La dimensión máxima para una correcta división de la



información en la Web sería la correspondiente a una o dos páginas impresas en papel.

- Segmentar la información puede ayudar a organizar y a presentar el contenido en un formato uniforme. Esto permite al usuario aplicar su experiencia del sitio visitado a futuras búsquedas y predecir cómo puede estar organizada una sección desconocida de una Web.
- Los fragmentos concisos de la información se adaptan mejor a la pantalla del monitor, donde no hay espacio para visualizar documentos muy extensos. Las Páginas Web grandes tienden a desorientar al usuario medio, pues se le fuerza a utilizar la barra de desplazamiento y a recordar lo que está fuera de pantalla.
- El concepto de fragmento de información debe ser flexible y consistente, acorde con el sentido común, y responder a una organización lógica y cómoda. Dejar que la naturaleza de los contenidos sea lo que sugiera cómo debe ser subdividido y estructurado.
- A veces es conveniente presentar los documentos largos en forma de sucesión enlazada de un conjunto de páginas. A pesar de que es preferible limitar la longitud, parece obvio que dividir un documento largo arbitrariamente puede tener muy poco sentido, especialmente si se quiere que el usuario pueda imprimir o guardar el documento entero con facilidad en un solo paso.

Importancia de la Jerarquía.

La organización jerárquica es una necesidad en la Web. La mayoría de los sitios dependen de una jerarquía, que va desde un plano general de conjunto, (que correspondería a la página principal, de inicio, o Home Page) hasta las páginas más específicas de submenús y contenidos.



Los fragmentos de información deberían clasificarse según su importancia y organizarse en función de las relaciones entre unidades. Una vez determinado un conjunto lógico de prioridades, se puede edificar la jerarquía desde los conceptos más importantes o generales hasta los más específicos o detallados.

Relaciones.

Cuando se enfrentan a un nuevo y complejo sistema de información, el usuario construye su propio esquema mental; este le sirve para evaluar las relaciones entre los temas y para predecir dónde encontrar lo que aún no ha visto.

El éxito en la organización del sitio Web se determina en gran medida por cómo se ajusta el sistema que se construya a las expectativas del usuario. Una organización lógica del sitio permite al usuario prever eficazmente donde puede encontrar las cosas. Si se muestra la información mediante un método adecuado, el usuario podrá extender su conocimiento de las páginas que le son más familiares a páginas completamente desconocidas.

Función.

Una vez que se construyó un sitio Web, hay que analizar su funcionalidad. Un diseño Web eficaz supone encontrar un equilibrio en la relación entre páginas menú y las páginas con contenido específicos. El fin es conseguir un orden de menús y páginas que el usuario medio perciba como natural, y que no interfiera con el uso que haga del sitio.



Los sitios Web con una estructura jerárquica poco profunda (ver Figura 3.3.1), depende demasiado de extensas páginas de menú, que pueden confundir como lo haría una “lista de compra” con información no relacionada:

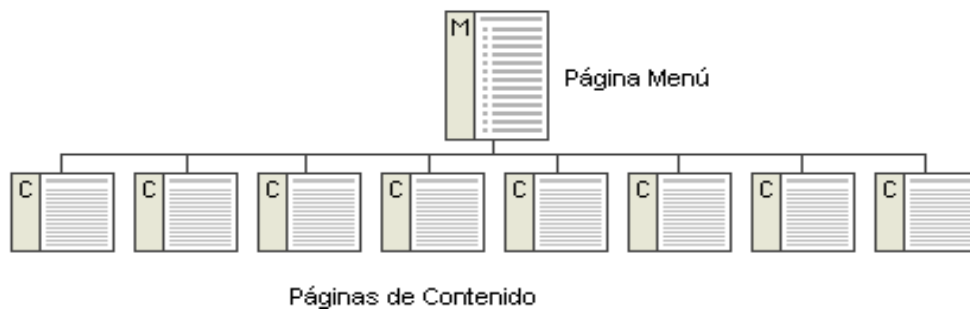


Figura. 3.3.1. Esquema jerárquico poco profundo.

Los esquemas del menú pueden también ser demasiado profundos, llevando la información bajo varias capas del menú. Cualquier usuario puede desanimarse si para acceder a los contenidos reales tiene que atravesar numerosas capas de menús vinculadas entre sí como se muestra en la Figura 3.3.2

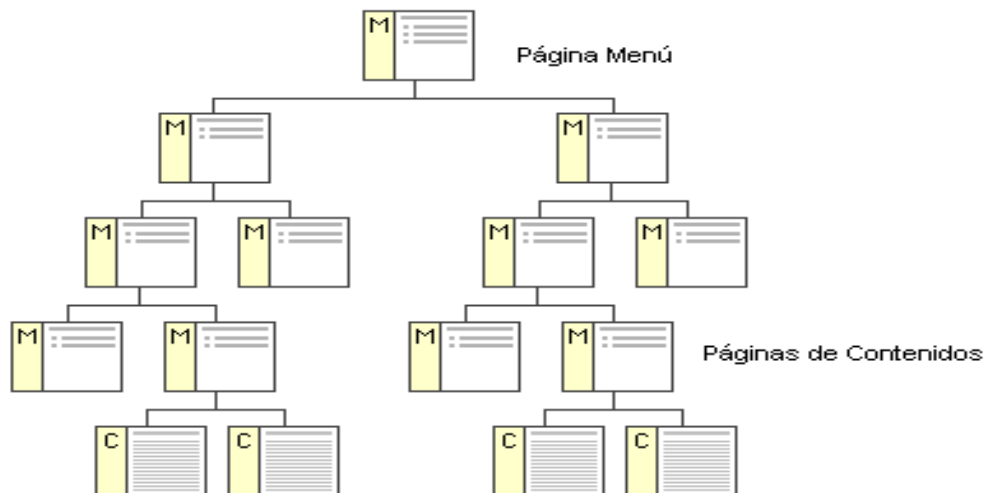


Fig. 3.3.2. Esquema jerárquico profundo.

Si el sitio Web esta en continuo crecimiento, el equilibrio correcto entre menús y páginas de contenidos es un objetivo dinámico fundamental. La retroalimentación o feedback por parte del usuario (y el propio análisis del uso que se hace al sitio) puede



Capítulo III. Definición y Análisis del Problema.



ayudar a decidir si el esquema de menús resiste el uso o tiene puntos débiles. Las estructuras complejas de documentos requieren jerarquías de menús más profundas, pero nunca se debe forzar al usuario a visitar una página de menús tras otra si es posible un acceso directo. Con una jerarquía funcional y equilibrada de menús, se puede ofrecer al usuario un acceso directo a la información y, a su vez, refleja la estructura general de la Web.

El paso más importante de planear el sitio Web, es organizar su información. Pensar cuidadosamente sobre qué se quiere decir y cómo se quiere decir implica un conocimiento profundo de los contenidos de la Web. Conviene hacer esbozos, dividir la información en secciones y sub-secciones, pensar en cómo éstas se relacionan entre sí y construir una tabla de contenidos. Una tabla de contenidos (Figura. 3.3.3) bien organizada puede ser una herramienta fundamental para la navegación. No es tan solo una lista de enlaces en cada una de las secciones, sino que permite al usuario percibir de un vistazo la estructura, la extensión y el flujo narrativo de lo que se presenta.

The image shows a screenshot of a Kodak website page titled "Digital Cameras and Technology". The page features a navigation menu with the following items:

- How to Get Started in Digital Photography**: Products and services to help you start working and playing with pictures on your computer.
- From Prints to Pixels**: Learn how film, digital, computers and the Internet work together.
- Digital Learning Center**: Learn the technology behind digital pictures, and discover exciting projects you can do on your own.
- Sample Digital Pictures**: Photo CD samples and digital camera picture galleries.
- KODAK Picture Ready:** Pictures can help you sell your stuff faster online. Upload, edit, and add pictures to your online auction posting or classified ad.

Below the menu, there are several product categories with brief descriptions:

- Digital Cameras**: Our complete line, for everyone from the casual snapper to the professional.
- Inkjet Media (Desktop)**: Specially formulated paper and transparency film for your inkjet printer.
- Home Inkjet Printers**: Make long-lasting, photo-quality inkjet prints at home using KODAK Personal Picture Makers.
- KODAK Picture CD**: Get your pictures on a CD when you process your film. Includes software for fixing, enhancing, and sharing pictures.
- KODAK Picture Disk**: Get your screen-resolution pictures on floppy disk when...
- Digital Projectors**: For displaying picture-rich presentations.
- Large Format Inkjet Products**: For large format printing (> 36"), printers, inks and multiple media formats.
- Thermal Printers**: For professional quality prints of your digital pictures.
- Scanners**: For getting your pictures and documents into your computer.
- KODAK Photo CD**: Have your pictures archived in high resolution on a CD for use in desktop publishing, web viewing, high quality printing.

The page also includes a small image of a child holding a butterfly and a "shop@kodak" logo.

Figura. 3.3.3 Tabla de Contenido en una página Web. www.kodak.com



Estructura del sitio.

Esta parte responde a la pregunta ¿Qué es lo que el sitio Web realmente quiere transmitir? La comunicación es la esencia de un sitio Web ya sea que el objetivo implique informar, persuadir, educar, o entretener la intención al publicar un contenido en la Web es presentar al público información que pueda entender codificar y actuar de acuerdo a ella.

En realidad, cualquiera que sea el contenido, este debe concentrarse en el público, debe ser relevante y pertinente. Debe atraer y mantener la atención, al mismo tiempo que comunica por completo el mensaje.

Estructuras de información básica.

Los sitios Web se construyen alrededor de temas estructurales básicos. Estas arquitecturas fundamentales gobiernan la interfaz de navegación del sitio Web y modelan los esquemas mentales del usuario de cómo se organiza la información.

Existen tres formas de estructurar un sitio Web:

- **Secuencias.** La manera más simple de organizar la información es colocarla en una secuencia. Todas las páginas siguen un orden lineal, de tal forma que cada una (excepto las extremas) tiene una anterior y una siguiente, son muy predecibles y evitan confusión. En sitios didácticos y de auto-aprendizaje resultan muy convenientes. Ver Figura 3.3.4.

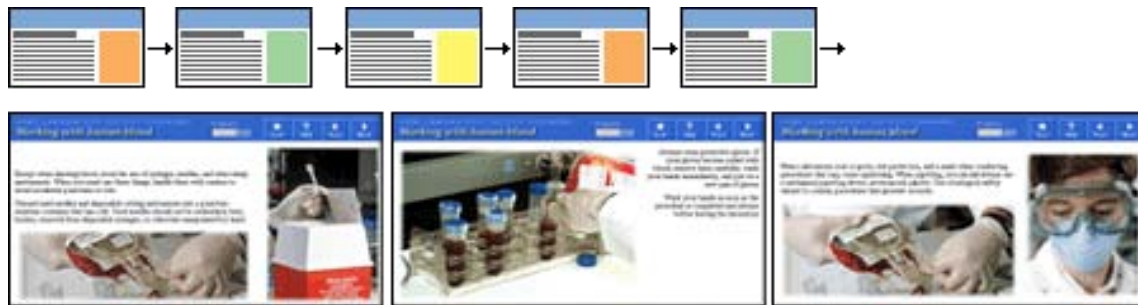


Figura.3.3.4 Estructura de la información en secuencias.

Las Web más complejas pueden también estar organizadas en forma de secuencias lógicas, pero cada una de las páginas de la secuencia principal puede tener enlaces hacia uno o más páginas paralelas donde se presentan observaciones e información colateral o de otras Web. Ver Figura 3.3.5.

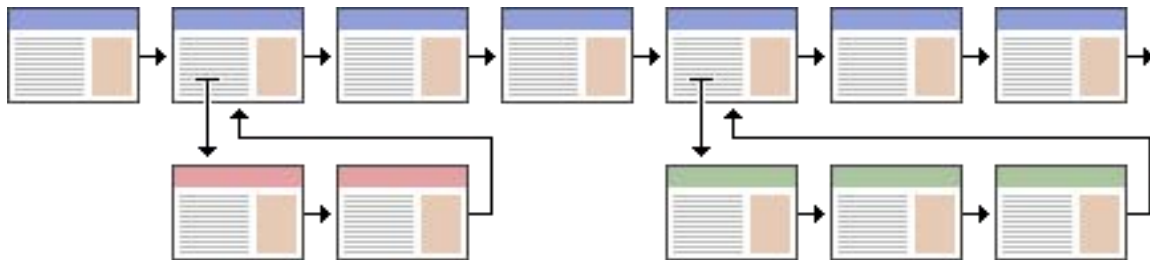


Figura. 3.3.5 Esquema de una Web compleja, con enlace a páginas paralelas.

- **Jerarquías.** Cada página se encuentra ubicada con otras a un mismo nivel, tiene una de nivel superior y puede tener una o más de nivel inferior. Los diagramas jerárquicos se emplean en sitios corporativos e institucionales, ya que la mayoría de los usuarios encuentran esta estructura fácil de entender. Una organización jerárquica también impone al constructor de la Web una eficaz disciplina en su análisis de los contenidos, ya que las jerarquías funcionan si el material está bien organizado. Ver Figura 3.3.6

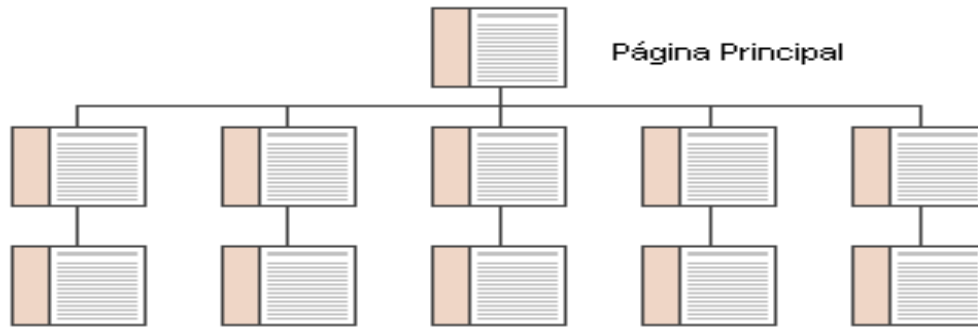


Fig. 3.3.6 Estructura jerárquica de un sitio Web.

- **Red (Web).** No sigue una orden en especial, se usa esta estructura, con público que conoce el tema y busca aspectos muy específicos; en general no es recomendable ya que genera confusión. Ver Figura 3.3.7.

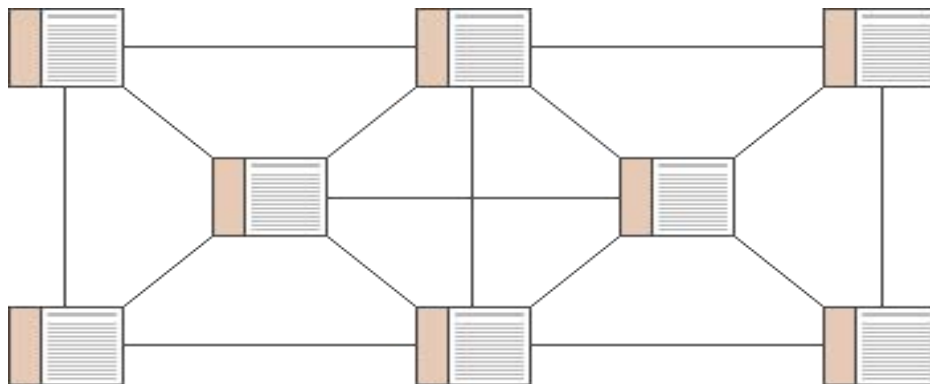


Figura. 3.3.7 Estructura en Red (Web).

Archivo de sitio y estructuras de directorio.

Los diagramas del sitio (Figura. 3.3.8) son también útiles cuando el proyecto pasa de la fase de planificación a la fase de producción real de las páginas Web. En el momento de colocar el sitio en el servidor, el diagrama suele ser el primer documento que utilizan los programadores para entender cómo deben colocarse los distintos archivos que compondrán el sitio Web en los directorios o las carpetas del servidor.



Capítulo III. Definición y Análisis del Problema.



El esquema de directorios y de subdirectorios de los archivos del sitio refleja las principales secciones y estructuras según se indica en el diagrama del sitio.

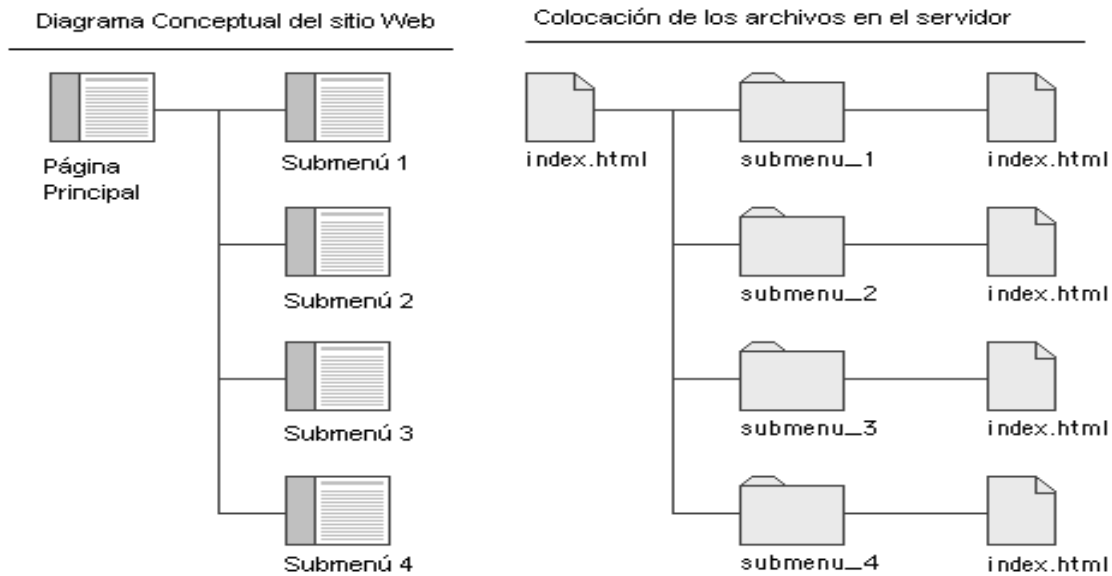


Figura. 3.3.8. Esquema de directorios y subdirectorios colocados en el servidor.

La mayoría de los sitios Web complejos comparten aspectos de los tres tipos de estructuras de información que se han descrito anteriormente. La Figura 3.3.9 resume los tres modelos de organización básica en relación con la linealidad de la narrativa y la complejidad de los contenidos.

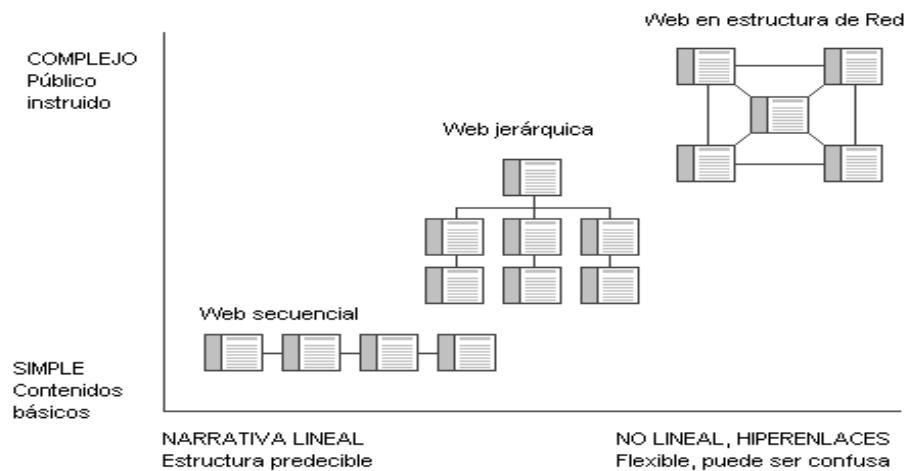


Figura 3.3.9. Gráfica comparativa de las estructuras de información.



La presentación de información responde a parámetros determinados por los objetivos, la logística práctica del medio elegible y el público al que se dirige.

La Figura 3.3.10 mostrada a continuación desglosa los temas más importantes de la entrega de información en función de dos variables fundamentales; el carácter lineal de la estructura de lo presentado y el tiempo de contacto de un usuario.

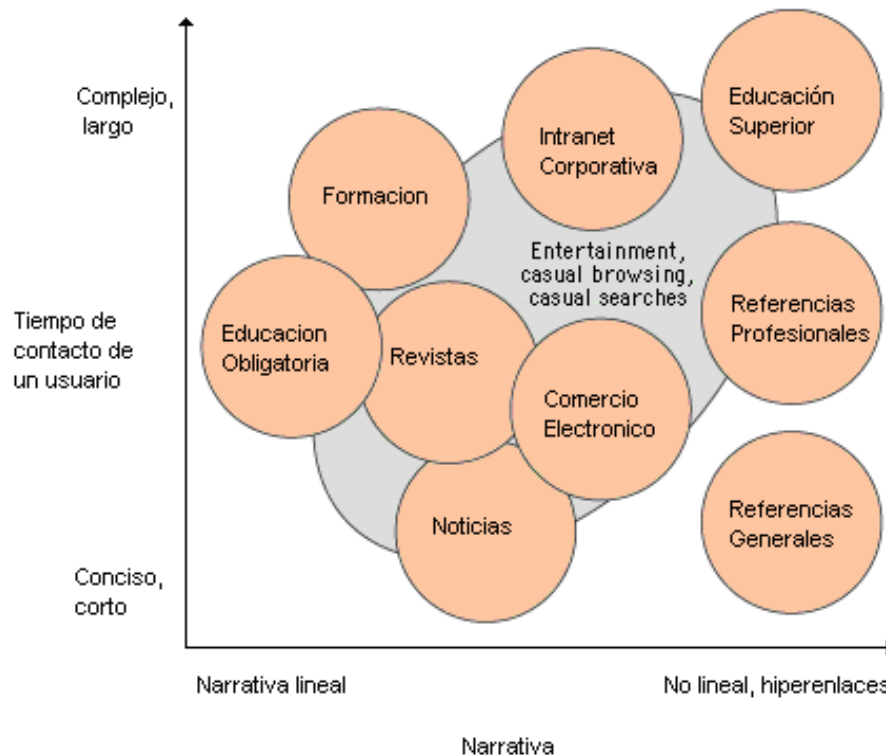


Figura 3.3.10 Gráfica comparativa de los diferentes tipos de sitios Web.

Por lo tanto, la propuesta para desarrollar el portal es una estructura jerárquica sin caer en la profundidad excesiva de capas. El tipo de información que mostrará el portal serán actividades y/o información que maneja y requiere la academia de informática.



3.4 Análisis Jerárquico del Sitio a Desarrollar.

El Proceso de Análisis Jerárquico (Analytic Hierarchy Process AHP) está diseñado para resolver problemas complejos de criterios múltiples. El proceso requiere que el que toma decisiones proporcione evaluaciones subjetivas respecto a la importancia relativa de cada uno de los criterios y a su vez especificar su preferencia con respecto a cada una de las opciones propuestas. El resultado del AHP es una jerarquía con prioridades que nos muestran la preferencia global para cada una de las opciones de decisión.

El AHP mediante la construcción de un modelo jerárquico, permite visualizar de una manera más eficiente y grafica la información de un problema para descomponerla y analizarla por partes, visualizar los efectos de cambio en los niveles y sintetizarlos.

Ventajas AHP.

- Permite desglosar y analizar un problema por partes. Permite medir criterios cuantitativos y cualitativos mediante una escala común.
- Incluye la participación de diferentes personas o grupos de interés y generar un consenso.
- Permite identificar conflictos. Permite determinar el grado de importancia de cada criterio.

Estructura del Modelo Jerárquico (AHP).

Una de las partes más relevantes del AHP, consiste en la estructuración de la jerarquía del problema, etapa en la cual el grupo de decisión involucrado debe lograr desglosar el problema en sus componentes relevantes. La jerarquía básica está conformado por: meta u objetivo General, criterios y opciones. Los pasos a seguir para la estructuración



del modelo jerárquico son:

- Identificación del problema.
- Definición de Objetivo.
- Identificación de Criterios.
- Identificación de Alternativas.

Identificación del Problema. Es la situación que se desea resolver mediante la selección de una de las opciones de las que se dispone. Dichas opciones serán comparadas unas con otras mediante la evaluación de criterios establecidos que permitan conocer los pros y los contras incorporados en cada una de ellas.

Definición de Objetivo. Es una dirección identificada para mejorar una situación existente. El objetivo está en un nivel independiente y los otros elementos de la jerarquía que serán los sub-objetivos o criterios y opciones apuntan en conjunto a la ejecución del mismo. Hay objetivos de largo, mediano y corto plazo, esta diferencia influirá en la construcción del modelo jerárquico.

Identificación de Criterios. Son las dimensiones relevantes que afectan de una manera significativa a los objetivos y deben de expresar las preferencias de los implicados en la toma de decisión. En este punto se deben de incluir aspectos vitales cuantitativos y cualitativos.

Identificación de las Opciones. Son las propuestas factibles mediante las cuales se puede alcanzar un objetivo y cada una de ellas tiene sus pros y sus contras.



Árbol de Jerarquías.

Consiste en elaborar una grafica del problema en términos generales, los criterios y las opciones de decisión, esta grafica se le conoce como Árbol de Jerarquías. Figura 3.4.1.

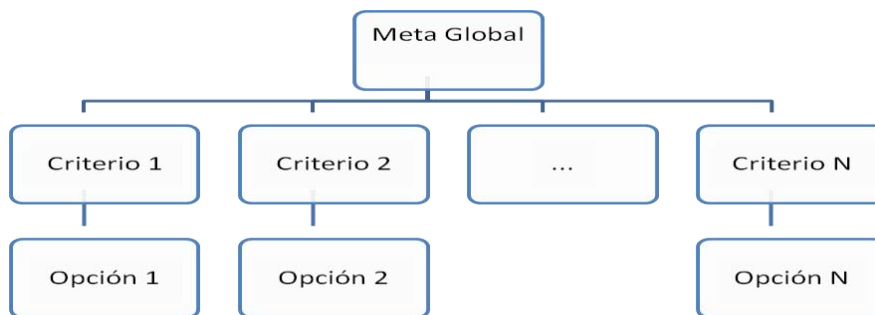


Figura 3.4.1 Árbol de Jerarquías.

El método AHP consiste en lograr que el líder especifique sus opiniones con respecto a la importancia relativa de cada uno de los criterios para la consolidación de la meta global. Cuando se construye una jerarquía se puede hacer de arriba hacia abajo o de abajo hacia arriba.

La construcción de arriba hacia abajo, se inicia con la identificación de los criterios globales, es decir desde lo más general a lo mas particular. Así todos los aspectos generales que recopilamos en la definición del problema estarán presentes en ese primer nivel a manera de criterios, cada uno de estos criterios debe de ir acompañado de una descripción de lo que significa. En la construcción de abajo hacia arriba el proceso es de manera inversa. Primero se generan todas aquellas características que nos permitan ver las opciones y después se construye el modelo jerárquico agrupando todas aquellas características que mantengan un factor común a manera de criterios o subcriterios, según sea el caso hasta llegar al objetivo general.



El sentido en que se comienza a construir va a depender de los datos disponibles. Si en la elaboración están bien definidas todas las opciones y además se conocen sus pros y sus contras, entonces se puede comenzar a hacer el modelo de abajo hacia arriba.

A continuación se presenta el análisis jerárquico del portal dando una explicación de cada uno de sus componentes. Este cuenta con 5 secciones principales cada una de las cuales posee la información requerida para facilitar el trabajo. Figura 3.4.2

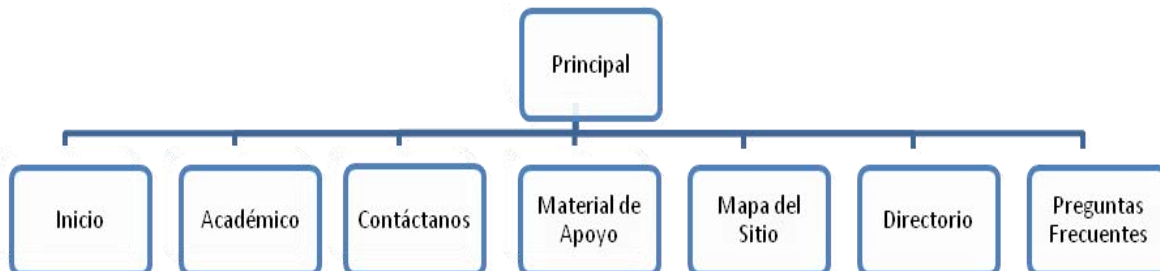


Figura 3.4.2. Primer Nivel.

La primera sección cuenta brevemente con una descripción de la institución incluyendo todas sus funciones. Figura 3.4.3

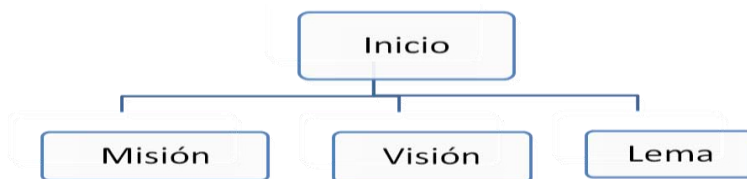


Figura 3.4.3 Primera Sección.

La segunda sección cuenta con todo lo que tiene que ver con cursos, eventos y reuniones así como cada uno de sus temas importantes. Figura 3.4.4

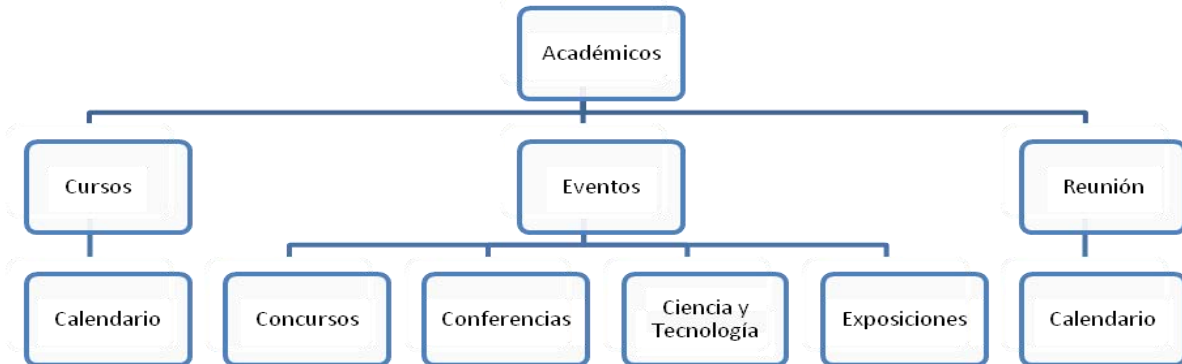


Figura 3.4.4 Segunda Sección.

La tercera sección cuenta con un apartado para que el usuario tenga a bien dar su opinión o sugerencias mediante un correo electrónico al administrador para que este lo considere y pueda darle una solución o en su caso dar un seguimiento y exponerlo ante la Academia y dar una solución favorable. Figura 3.4.5



Figura 3.4.5 Tercera Sección.

La cuarta sección cuenta con la información como su nombre lo dice de apoyo al docente, el cual estará clasificado por Semestre, Materias y Temas. Figura 3.4.6

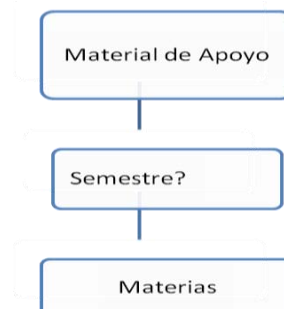


Figura 3.4.6 Cuarta Sección.



La quinta sección contiene el Mapa del sitio, la cual es la representación del sitio web. Está nos permite planificar el diseño de la página, el formato y su estándar para presentar el contenido del sitio. Figura 3.4.7

Mapa del Sitio

Figura 3.4.7 Quinta Sección.

La Sexta sección cuenta con los datos de los Directivos de cada una de las dependencias integrantes de la DGETI (CETIS). Figura 3.4.8.

Directorio

Figura 3.4.8 Sexta Sección.

La Séptima sección cuenta con la sección de preguntas frecuentes. Figura 3.4.9.

Preguntas
Frecuentes

Figura 3.4.9 Séptima Sección.

A continuación se presenta el esquema jerárquico completo del portal propuesto con cada una de sus secciones. Ver Figura 3.4.10

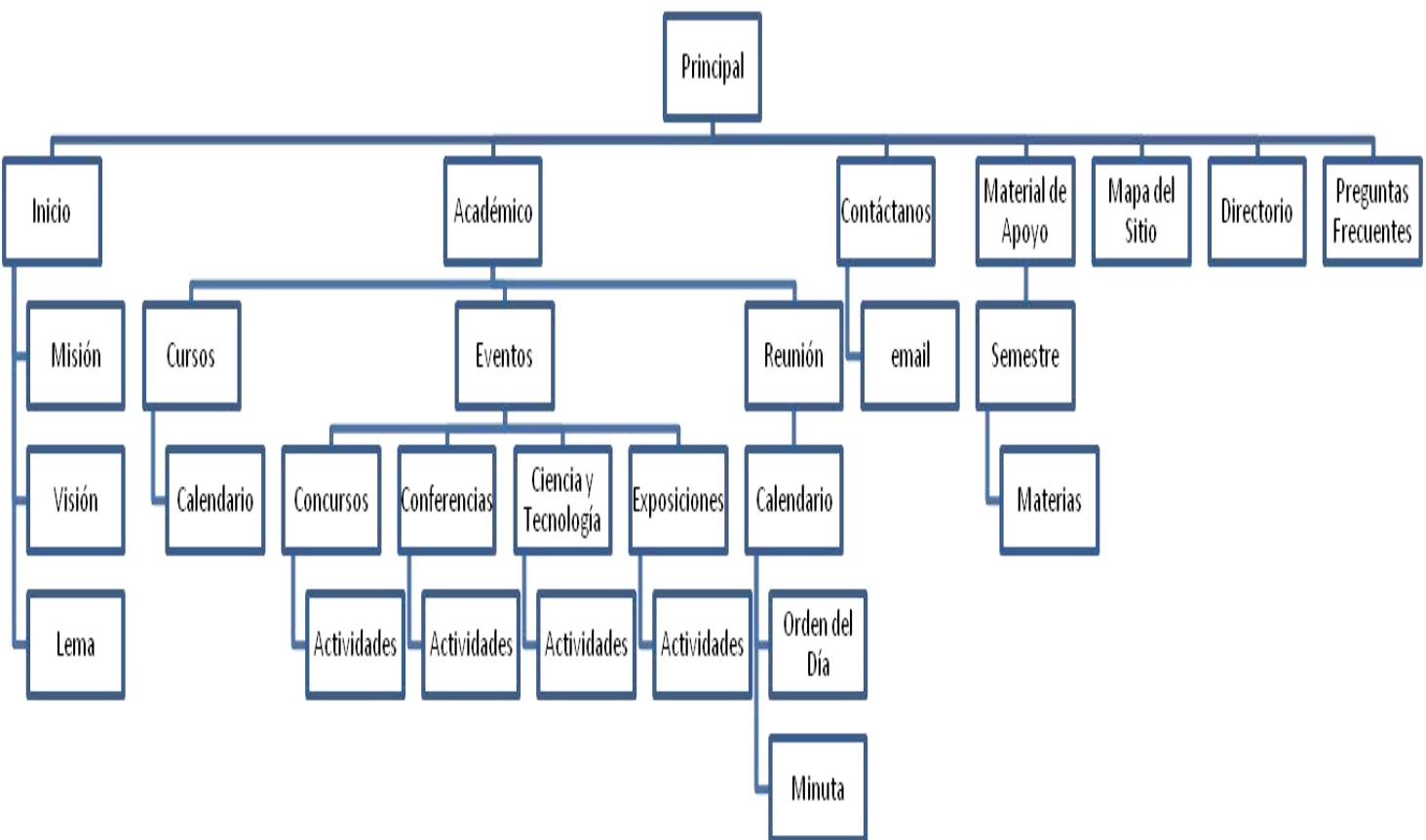


Figura 3.4.10 Portal Propuesto.



3.5 Opciones de solución del Front –End y Back-End.

Para la elección del Front-End se describe el software requerido para el desarrollo de una página Web.

HTML (Hyper Text Markup Lenguaje) es un lenguaje utilizado para crear páginas Web. En resumen el HTML es el lenguaje usado por los navegadores para mostrar las páginas Web al usuario.

Así pues, existen dos herramientas fundamentales e imprescindibles asociadas al lenguaje HTML:

- Los editores HTML (para crear documentos HTML) y,
- Los navegadores (para visualizar dichos documentos).

Aunque también existen otras herramientas automatizadas para generar páginas Web, como son los convertidores de otros formatos y otro tipo de herramientas que revisan y validan datos permitiéndonos analizar los documentos HTML ya creados para ver si se ajustan a los parámetros de este lenguaje.

Los documentos HTML son archivos de texto plano (también conocidos como ASCII) que pueden ser creados mediante cualquier editor de texto, aunque también existen programas específicos para editar HTML (los editores más conocidos son Microsoft FrontPage, Netscape Composer, Macromedia Dreamweaver y Adobe PageMill), concebidos específicamente para editar páginas Web en HTML.



Editores HTML.

La forma más sencilla de realizar un archivo HTML es utilizar cualquier editor de texto sencillo como el Bloc de notas o el WordPad de Windows.

Los editores aceleran o automatizan determinadas tareas comunes en la edición de HTML, a la vez que realzan la presentación del código fuente, por ejemplo, muestran el texto en colores para los diferentes elementos del HTML. Esto hace que la creación de documentos HTML sea más rápida, cómoda y legible. Se pueden distinguir tres tipos de editores de HTML

Editores en modo texto. Son editores de texto plano, tienen un conjunto de menús, barras de botones, etc., a través de los cuales es posible seleccionar las diferentes etiquetas HTML. Algunos ofrecen cuadros de diálogo a través de los cuales se pueden cambiar los diferentes atributos de cada etiqueta. Como se edita directamente el código HTML, se tiene un control total sobre el mismo, sin embargo su principal inconveniente es que el código generado no es demasiado legible, y el proceso de edición es más lento.

Editores gráficos o "WYISWYG" ("What You See Is What You Get", "lo que ves es lo que obtienes"). Son los más intuitivos, pero se pierde algo de control sobre el código HTML que producen ya que sólo se pueden hacer aquellas cosas que el editor 'sabe' hacer, de forma que realmente "lo que ves no es lo que obtienes".

Algunos Editores WYSIWYG de este tipo son:

- **Dreamweaver.** Es un editor visual de páginas web que ha alcanzado un gran nivel de sofisticación sin perder la simplicidad, lo cual lo convierte en la opción ideal. Permite un buen manejo de capas, efectos de HTML dinámico, JavaScript, hojas de estilo y otras herramientas más profesionales.



- **FrontPage2000.** Es un programa pensado para aprovechar la gran cantidad de temas que es posible incorporar al sitio, que incluyen gráficos, botones, titulares, barras de navegación, diseños de fondos, colores y fuentes, ideal para los que quieren un buen diseño, y rápido. Tienen una utilidad para vincular una hoja de estilo a todas las páginas del sitio. Pero no permite manejarlas con facilidad.

Editores mixtos. Son una mezcla de los anteriores. Realzan la presentación del código generado, y disponen de algún sistema para mostrar, de forma diferenciada, las etiquetas HTML introducidas, y modificar sus parámetros.

También hay procesadores de texto (Microsoft Word, WordPerfect) que permiten transformar un documento en una página web, es decir, de alguna forma convierten un documento de texto en un documento HTML. Esta opción no es recomendable ya que estos editores suelen generar código basura.

Ver tabla 3.5.1 Ejemplos de algunos editores

| Editor | Tipo | Ejemplos |
|--------|-------------------------|--|
| HTML | Modo Texto | Notepad |
| | Modo Gráfico (WYSISWYG) | Dreamweaver de Macromedia Front Page de Microsoft |
| | Modo Mixto | Arachnophilia., Word Perfect |

Tabla 3.5.1 Clasificación de editores de HTML.

Existen dos tendencias para realizar las páginas Web. Por un lado tenemos a desarrolladores que prefieren crear páginas programando el HTML y por otro lado quienes utilizan los editores de HTML. Se presentan algunas diferencias entre hacerlo de un modo u otro ver tabla 3.5.2:



| Escribiendo el HTML | Con un editor WYSIWYG |
|---|---|
| Dominas con mayor precisión el código de la página, queda más limpio. Si dominas bien el HTML nunca tendrás ningún problema para hacer lo que deseas. | El código de la página tiene peor calidad, incluso puede llegar a tener errores, más o menos visibles, que cuesta arreglar. Es la máquina la que domina el trabajo. |
| Es más complicado el aprendizaje, más lento y cuando se llega a un nivel avanzado también se hace considerablemente más difícil | El aprendizaje es muy sencillo, tal como puede ser trabajar en Word. Solo se trata de manejar un programa más. |
| Hacer una página cuesta más trabajo y tiempo. | Es muy rápido |

Tabla. 3.5.2 Tendencias para el desarrollo de páginas Web.

Se debe escoger el camino que más convenga o el que parezca más atractivo. De cualquier forma se pueden emplear las dos formas sin problema alguno. Incluso, cuando el desarrollador profundiza en el diseño de páginas Web llega un momento en el que le hace falta conocer las dos maneras de construir Web.

En el mercado existen multitud de editores de HTML WYSIWYG, es importante elegir un editor bueno porque el desarrollo de páginas Web depende de sus resultados. Actualmente el principal editor es Dreamweaver, fabricado por Macromedia. Otras posibilidades son editores como GoLive de Adobe o Front Page de Microsoft y editores de texto preparados para escribir HTML. Posiblemente sea aconsejable empezar con el Block de notas, porque es lo más sencillo, pero utilizar un programa de estos será imprescindible con el tiempo. Algunos navegadores existen son:



Mozilla Firefox.

- La versión Firefox 2.0.0.2 ofrece las siguientes ventajas:
- Características anti-phishing(Anzuelo o Estafa electrónica): que en un principio ya las tuvieron las primeras versiones de Internet Explorer.
- Opciones ocultas para configurar las pestañas: que si bien fue una funcionalidad que primero apareció en los navegadores Opera, hoy está perfeccionada en la nueva versión de Firefox y permite establecer una serie de páginas de inicio que se abren en distintas pestañas, simultáneamente. Esto es útil para personas que necesitan estar actualizadas en diversos canales de información o administran varios blogs, por ejemplo.
- Restauración automática de la sesión de navegación del usuario en caso de caída del sistema, lo que no permite la pérdida de búsquedas o datos de las páginas.
- Tema por defecto nuevo: con nuevos iconos y barra de tabs, a través del ingreso en el lenguaje XUL, el usuario puede modificar y adaptar el entorno de navegación a su gusto, establecer el programa de Gmail Notifier, o establecer un almacenamiento de ficheros similar al FTP, en asociación con la cuenta de Gmail.
- Soporte para texto con gráficos vectoriales escalables: SVG usando svg: textPath.
- Integración de una función para administrar los motores de la barra de búsqueda: que también pueden ser seleccionados por el usuario sin necesidad de abrir una nueva pestaña o ventana.
- La administración y ordenamiento de los marcadores en carpetas y subcarpetas: al igual que su conversión en marcadores de actualizaciones de feeds y sindicación a determinados sitios.



- Bloqueo de pop-ups y un cortafuego: eficiente que protege a la computadora, además de establecer el nivel de protección por sitios. También esta herramienta se vuelve eficaz a la hora de bloquear contenidos para niños.
- Uso de keywords y descripción verbal en la barra de direcciones: para obtener el sitio exacto donde está la información, sin necesidad de escribir la URL exacta.

Internet Explorer 7 (IE7).

Entre sus características encontramos:

- Anti-phishing
- Interfaz mejorada
- Navegación por pestañas.
- Modo seguro de navegación para Windows Vista, solamente
- Sindicación a sitios RSS
- Búsqueda directa por elección de proveedor en la barra de herramientas
- Opciones mejoradas de impresión

Entre sus ventajas están su velocidad, seguridad, soporte de estándares (especialmente CSS), tamaño reducido, internacionalidad y constante innovación. La versión 9, lanzada en junio de 2006, ofrece las siguientes funcionalidades:

- Cliente BitTorrent
- Bloqueador de contenidos. Widgets
- Compatibilidad con la edición Rich Text Editing.(Edición de texto enriquecido)
- Soporte completo de SVG 1.1 Basic



Opera.

Es un navegador pequeño pero con buenas características como: soporte para frames, tablas códigos JavaScript, CSS o Java. Además cuenta con características muy especiales como el zoom de una página o navegación simultanea de varias páginas en un solo ambiente de navegación.

Una de las cosas más importantes de Opera es que es rápido, no se necesita de una maquina muy poderosa para navegar por la Web. Opera tiene un costo, pero en versiones beta se puede obtener de Internet. Se utiliza principalmente para tecnologías de telefonía Celular.

Las opciones de navegadores cada vez son más amplias, se cuenta desde los conocidos por sus casas creadoras así como los de software libre. Para escoger un navegador es importante que se tenga en mente cuáles son las características de los mismos, se realicen comparaciones y selecciones el que mejor convenga según las necesidades del usuario.

Uso actual de navegadores.

Actualmente el navegador que mayormente se emplea en el mundo es Internet Explorer en todas sus versiones, algunas empresas indican que esta ventaja se debe a que viene integrado con Windows, detrás de éste está el navegador de Mozilla Firefox, el cual se está popularizando cada vez más. Firefox es un competidor serio al producto de Microsoft. Existen también los navegadores Safari, Netscape Navigator y Opera los cuales tienen un uso de menos de 2% de uso en el mercado.

Hay que tomar en cuenta una consideración el HTML no permite definir de forma estricta la apariencia de una página, aunque en la práctica, se utiliza también como un lenguaje de presentación. Los archivos de HTML se leen en un navegador Web tal



como Netscape Navigator, Microsoft Explorer, Mozilla, etc. La presentación de la página es dependiente del navegador o browser utilizado ya que el mismo documento no produce el mismo resultado en la pantalla si se visualiza con uno u otro, esto es, HTML se limita a describir la estructura y el contenido de un documento, y no el formato de la página y su apariencia.

Paginas Estáticas Y Dinámicas.

A medida que el uso de Internet se ha ido popularizando como sistema de Información, ha surgido la necesidad de dotar de cierto dinamismo a las páginas Web que muestran la información de forma estática. De esta manera resulta habitual que las aplicaciones Web actuales incorporen páginas que permitan interactuar con los usuarios. Podemos definir las como:

Páginas Estáticas.

Un documento Web estático reside en un archivo asociado al servidor Web. El autor de este documento determina el contenido al momento de escribirla. Dado que el contenido no cambia, una solicitud de un documento estático, da siempre la misma respuesta.

Ventaja. Se siguen utilizando ya que forman la base necesaria para la presentación de datos, las principales ventajas son su sencillez, confiabilidad y desempeño. Un visualizador puede presentar con rapidez un documento estático.

Desventaja. Es su inflexibilidad (el documento debe modificarse siempre que cambie la información). Además los cambios llevan tiempo, porque se requiere que alguien corrija el documento.



Páginas Dinámicas.

Un documento de Web dinámico es creado por un servidor Web, cuando un visualizador lo solicita. Al llegar una solicitud, el servidor Web ejecuta un programa de aplicación que crea el documento dinámico. El servidor devuelve como respuesta al visualizador que solicitó el documento, la salida del programa. El contenido de un documento dinámico puede variar entre una y otra solicitud.

Ventaja. Los documentos dinámicos tienen la capacidad de reportar información actual.

Desventajas. Tienen mayor costo y la incapacidad de presentar información cambiante, ya que los documentos no cambian una vez que el visualizador trae una copia.

La tecnología primaria para crear documentos dinámicos se llama interfaz común de puerta de enlace (CGI).

La norma CGI especifica la manera en que un servidor llama a un programa CGI y el modo en que interpreta la salida del programa. Un ejemplo de esta situación es la existencia de páginas Web con formularios que los usuarios rellenan; posteriormente se hacen búsquedas en bases de datos alojadas en el servidor y los resultados se transmiten al cliente en formato HTML.

Las páginas ASP (Active Server Pages) de Microsoft proporcionan un medio sencillo y eficiente de programar páginas Web, ofreciendo un modelo alternativo a las aplicaciones CGI.



Páginas Activas.

Un documento activo no es especificado completamente por el servidor, consta de un programa que entiende la manera de calcular y presentar valores. Cuando un visualizador solicita un documento activo, el servidor regresa una copia del programa, que deberá ejecutarse localmente. Por lo tanto, el contenido de un documento activo nunca es fijo.

Ventaja. Su capacidad de actualizar continuamente la información.

Desventaja. El costo extra de crearlos y ejecutarlos con la tecnología Java se crea y ejecutan documentos activos.

Decisión de la solución Front –End.

Existe una gran variedad de editores y herramientas para el desarrollo de proyectos Web, por lo que no es fácil elegir. Depende del uso que se le vaya a dar y/o a las preferencias de cada usuario. Es recomendable apoyarse en ambos editores, uno en modo texto para poder manipular con comodidad el código, y otro gráfico para la inserción de código de forma rápida y cómoda.

La decisión incluye el empleo de programación PHP para la obtención de los datos que se encuentran en la Base. Un servidor Web Tomcat, el lenguaje de validación del lado del cliente Java Script y hojas de estilo o CSS. Estas decisiones se discuten a continuación:

El Servidor Web Tomcat permite la utilización de JSPs, si bien no es el único que lo permite, trabaja de manera sencilla, eficaz y eficiente. No obstante la razón de peso



para utilizar tanto el lenguaje como este servidor, es que la herramienta es funcional para realizar el procesamiento de páginas dinámicas.

El lenguaje Java Script se ha incorporado por la amplia variedad de procesos que permite realizar del lado del cliente. En un sistema que pretende dar servicio de análisis de datos a un gran número de usuarios, es importante intentar minimizar la cantidad de operaciones realizadas por el servidor. Esto involucra la necesidad de evitar que cada vez que se envíen datos para ser procesados sea el servidor quien deba validar su consistencia y corrección. JavaScript permite hacer este análisis de datos en el lado del cliente asegurando, de esta manera, que los datos enviados al servidor serán siempre válidos.

Se optó por emplear hojas de estilo las cuales facilitan el diseño de imagen del sitio. Desde la invención de las CSS se ha provisto a los desarrolladores una forma de evitar las distintas vistas mediante la definición de plantillas de estilo. Que incluso ayudan a quien viene detrás para generar adiciones o modificaciones a un módulo existente.

En este caso se eligió, Dreamweaver como la herramienta para el diseño de la página debido a que cumple perfectamente el objetivo de diseñar páginas con aspecto profesional, y soporta gran cantidad de tecnologías, además muy fáciles de usar:

- Hojas de estilo y capas.
- Javascript para crear efectos e interactividades.
- Inserción de archivos multimedia.

Este programa fue descargado desde la página www.macromedia.com, en una versión limitada en el tiempo, pero suficiente para desarrollar el diseño.



Decisión de solución del Back –End.

Para la elección del Back-End se analizó distintas bases de datos en el capítulo anterior, todas son de distribución y utilización libre.

Se selecciono MySQL ya que es un sistema de gestión de base de datos relacional y el hecho de que sea gratuita es un factor a favor de esta base. La administración de la misma es un tanto sencilla; existe documentación disponible en Internet, principalmente en el sitio www.mysql.com el cual ofrece una mayor cobertura a los posibles problemas de los usuarios.

Se utilizó phpMyAdmin el cual es un programa de libre distribución en PHP. Es una herramienta muy completa que permite acceder a todas las funciones típicas de la base de datos MySQL a través de una interfaz web muy intuitiva.



**Portal de la Academia Regional de Informática de la
DGETI.**





CAPÍTULO IV



**Portal de la Academia Regional de Informática de la
DGETI.**





4. Diseño, Programación e Implantación.

4.1. Aplicación de la Metodología.

De acuerdo al análisis que se realizó anteriormente y apoyados en las metodologías señaladas en el tema 3.3 y 3.4 se va a emplear el método jerárquico para el desarrollo del portal ya que este permite visualizar las inconsistencias que pueda tener la página.

- **Identificación del Problema.** El problema primordial es el tiempo de circulación de formatos, solicitudes y dependencias en que la información de gran relevancia tarda en llegar a las autoridades para la difusión pertinente, en cuanto a cursos, reuniones, exposiciones, conferencias, etc.
- **Definición del Objetivo.** El objetivo **principal** es la creación de un portal que muestre el contenido de la información de cursos, eventos, etc que se organizan en la academia regional; al emplear **esta forma o (vía)** se reducirán tiempos tanto de entrega de oficios, una participación activa de académicos en cada evento, además del intercambio de información que podrá llevarse a cabo sobre temas de importancia entre la comunidad, repercutiendo en el nivel académico de forma positiva.
- **Identificación de Criterios.** Como parte del procedimiento y al determinar los aspectos característicos, se identificaron una serie de criterios que gracias al análisis desarrollado en el apartado anterior, deben ser prácticos y óptimos para el portal; el siguiente diagrama presenta los criterios siguientes que se consideran como los óptimos para su desarrollo. Figura 4.1.1.

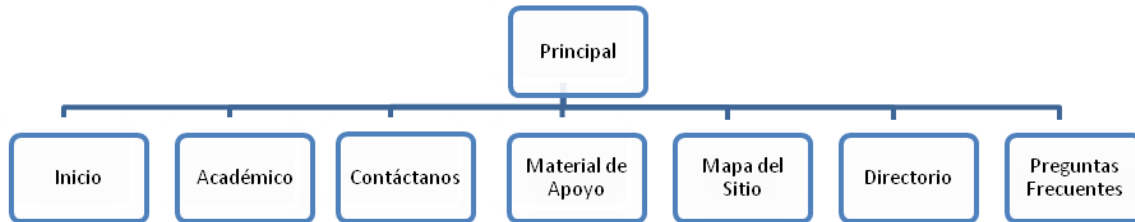


Figura 4.1.1. Identificación de Criterios.

- **Identificación de Opciones.** Tomando en cuenta la clasificación y organización de la información además de la fundamentación de datos sólidos y lógicos, así como de una segmentación de la información que permite contar con una buena organización y a presentar el contenido en un formato uniforme, las opciones consideradas óptimas para llegar al objetivo principal que es el desarrollo del portal, mostrándose en la Figura 4.1.2.1 en la Figura 4.1.2.2 se observa una segmentación de mayor grado llevando finalmente a un diagrama con la segmentación óptima.

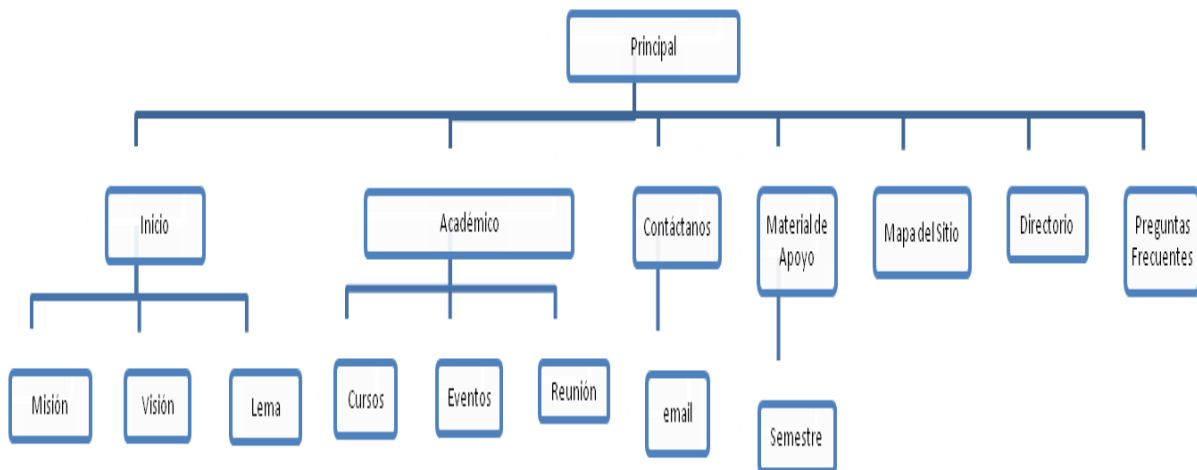


Figura 4.1.2.1. Identificación de Opciones.

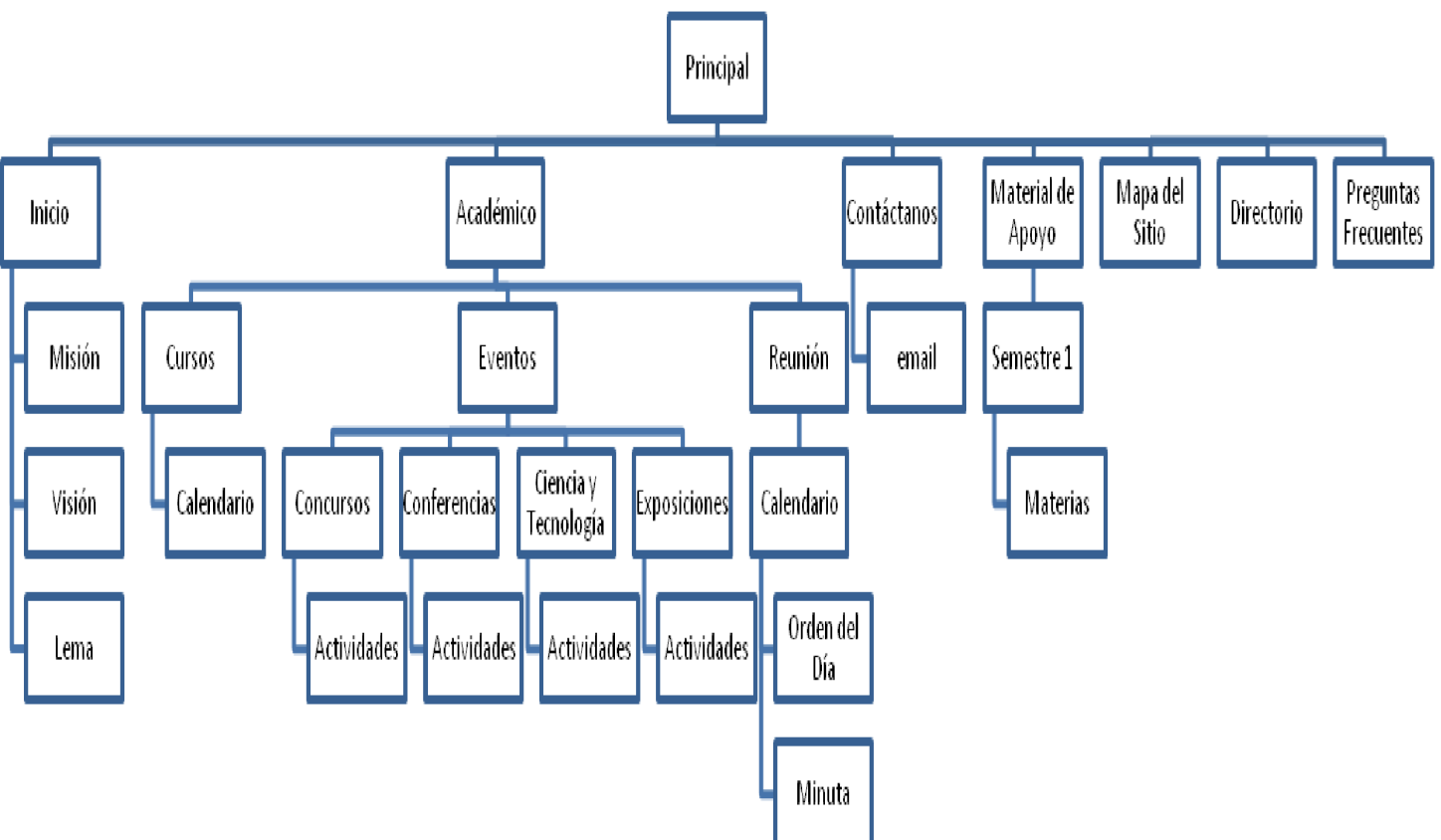


Figura 4.1.2.2. Identificación de Opciones segmentación mayor grado.



Considerando la sección 3.3 se han encontrado puntos débiles en el desarrollo del portal, que han sido corregidos en su momento, uno de ellos es la jerarquía y relaciones así como la estructuración del sitio para que el usuario no se vea forzado a observar demasiados submenús logrando así que el sitio sea funcional.

4.2 . Diseño del Portal.

Una de las características que hace tan popular a la tecnología web es su facilidad para mostrar contenidos de manera gráfica y para vincular fácilmente documentos de origen diferente. No obstante, para que esta pueda darse efectivamente, es necesario que al desarrollar un sitio Web contenido en cierta plataforma, cumplan con estándares que aseguren que la mayor parte de los usuarios podrán ver lo que se publica.

Para ello, es de suma importancia que los sitios que se construyan cumplan con ciertas características de publicación que permitan conseguir dos objetivos concretos:

- El despliegado rápido de las páginas, sin dificultades técnicas en las computadoras de los usuarios;
- La visualización de las páginas, por parte de los usuarios de forma tal como los desarrolladores la construyen.

Para conseguir ambos objetivos, se requiere que al construir los sitios web, se haga uso de un conjunto de experiencias en la construcción de este tipo de contenidos digitales, además de asegurar que se cumpla con los estándares mundiales en este ámbito, que se describen a continuación:



Peso de las Páginas.

Los sitios web deben tener un peso máximo permitido por página que no supere una cantidad razonable de kilobytes(kb), que impidan su visualización. En este sentido, lo razonable dependerá directamente del tipo de sitio desarrollado y de la conexión que tiene la mayor parte de los usuarios.

Esquemas o estilos de las Páginas.

Aunque existen nuevas tecnologías para los esquemas de las páginas web (como las Hojas de Cascadas de Estilo o CSS), lo habitual es que los contenidos que se muestran se dispongan en tablas con el fin de que cada elemento ocupe el lugar que se le ha asignado dentro de la página.

Ejemplo de código que se utilizó en el diseño del portal.

```
<style type="text/css">
```

```
.Estilo1 { font-family: Arial, Helvetica, sans-serif;
```

```
font-size: 12px; text-align:center;
```

```
color:#003366; background-color:#CEEAF7}
```

```
.Estilo2 { font-family: Arial, Helvetica, sans-serif;
```

```
font-size: 12px; text-align:center;
```

```
color:#FFFFFF; background-color:#005782 }
```

```
</style>
```



Al respecto se recomienda construir una estructura de presentación de los contenidos que se pueda fragmentar en varias tablas. De esa manera, cuando el sitio se presente en el programa visualizador del cliente, siempre mostrará la primera tabla (que normalmente llevará el logotipo y la identificación del sitio) de manera rápida, dando al usuario la sensación de haber llegado al destino elegido. Luego en las siguientes tablas se van visualizando los elementos restantes del sitio.

Se debe evitar el uso de tablas generales que incluyen en sí mismas a otras (tablas anidadas), ya que el programa visualizador usará una parte del tiempo en calcular esa relación de dependencia entre las tablas, antes de mostrar algo útil en la pantalla.

Por lo que el tiempo de espera del usuario sería mayor. Ver Ejemplo de Uso de Tablas en la página en la Figura 4.2.1.

| | |
|---|---|
| NOTICIAS | |
| Circular del mes de Mayo <i>Concurso Nacional estudiantil Interinstitucional de la SEMS 2008.</i> <i>Fecha del 12,13,14 de Mayo 2008</i> <i>Cede CETIS NO 30</i> <i>Sito Av Cienfuegos No.1017 esquina Ticomán Colonia San Pedro Zacantenco. Delegación Gustavo A. Madero.</i> |  AVISOS |
| INGRESA A NUESTRO SITIO | Las Secuencias didácticas del próximo semestre pronto las podras encontrar en el sitio. |

Figura 4.2.1 Uso de Tablas en la pantalla Principal.



Uso de Presentaciones en Flash.

La razón para evitar el uso de Flash en la portada es que su uso excede la carga de la presentación del sitio y si la presentación no está bien hecha, puede impedir el acceso. La buena práctica en este sentido es ofrecer una portada con la identificación de la institución y dos enlaces: uno para ver la presentación y otro para ingresar directamente al sitio. Adicionalmente se debe ofrecer la información que sea necesaria para que los usuarios puedan ver el contenido sin experimentar problemas. Ver Figura 4.2.2.



Figura 4.2.2. Página Principal.

Uso de Marcos o Frames.

La tecnología de marcos o frames consiste en agrupar varios archivos para que se desplieguen de manera simultánea, permitiendo a los usuarios ver varios contenidos al mismo tiempo.



Esta tecnología tiene aspectos positivos y negativos, que se detallan brevemente:

Positivos:

- Permite tener ciertos contenidos presentes todo el tiempo, como un cabezal o menú.
- Facilita la navegación ya que el usuario nunca pierde de vista dónde se encuentra.

Negativos:

- Impide que el usuario pueda marcar una página como favorita (bookmark) porque nunca se le muestra cuál es su dirección web.
- Cuando un usuario llega a un contenido desde un enlace provisto por un buscador, verá el sitio sin los otros marcos y no sabrá cómo navegar en él.
- La existencia de varios archivos en uno genera una carga mayor para el usuario que llega al sitio; eso lo obliga a esperar a que aparezcan todos los contenidos de los archivos para poder usarlo.

Ejemplo de código utilizado en el portal. Ver figura 4.2.3.

```
<frameset rows="28%,*,1" cols="*" frameborder="0">  
<frame src="menu_cliente.html" name="menu" scrolling="NO" noresize>  
<frame src="principal.html" name="principal" scrolling="si" noresize>  
<frame src="UntitledFrame-2.html"></frameset>158rchive158r158rchive
```




Capítulo IV. Diseño, Programación e Implantación.



ACADEMIA REGIONAL DE LA DGETI

[Académico](#)

[Contáctanos](#)

[Mapa del sitio](#)



Quinta Reunión Academica

Se llevará a cabo en el Cetus 21 el día
23-09-2008. A las 16:00:00 hrs.



[INGRESA A NUESTRO SITIO](#)

Las Secuencias didácticas del próximo semestre
pronto las podrás encontrar en el sitio.

Figura 4.2.3. Frame de la página principal.

Uso de Imágenes de background.

Tecnología muy popular que se puso de moda en el año 1996, cuando el software Netscape Navigator lo implementó fue en el empleo de imágenes como fondos o backgrounds de las páginas web.

Salvo en casos que sea estrictamente necesario, esta práctica debe usarse poco porque su único efecto es el de agregar un paso excesivo a los sitios, afectando el tiempo de descarga y acceso a la información.

Uso de Meta Tags Adecuados.

Los meta tags son marcas en lenguaje HTML que van en la parte superior del código fuente de cada página, a través de las cuales se entrega a los sistemas de indexación y búsqueda, la información mínima para hacer una correcta indexación del contenido que incluye.



- Los meta tags son un conjunto de elementos que obedecen a un estándar definido por el World Wide Web Consortium por lo que su uso está regulado y mediante los cuales sobrescribe información concreta en la página, tal como título, autor, descripción, idioma y otros. Los más importantes, aunque no los únicos, son los siguientes:

```
<title>Nombre del Sitio o Institución</title>
```

```
<meta name="title" content="Nombre del Sitio o Institución">
```

```
<meta name="description" content="Descripción del Sitio o Institución">
```

```
<meta name="keywords" content="Palabras claves del Sitio o Institución">
```

Adicionalmente sobre este tema en los últimos años se ha desarrollado la estandarización a través del grupo Dublín Core que ha definido los elementos meta tags a usar y que deben ser consultados en <http://www.dublincore.org>.

Normas para Incorporar Elementos Gráficos y Multimedia.

Cuando en un Sitio Web se incorporan elementos gráficos y de multimedia, se deben seguir normas muy concretas para evitar que su peso afecte el desempeño de la página cuando sea solicitada por los usuarios del Sitio Web.

A continuación se describen algunas recomendaciones tendientes a asegurar la correcta inclusión de dichos elementos:

- Optimizar el peso de las imágenes: se debe bajar al máximo posible el peso de las imágenes; cuando no sea posible hacerlo por su tamaño, se debe reducir el número de colores disponibles y la resolución (72 dpi es la norma).



Capítulo IV. Diseño, Programación e Implantación.



- Elegir el formato adecuado: ante un mismo tamaño de imagen, el peso varía dependiendo si estas son procesadas para desplegarse en formato GIF respecto del formato JPG. Normalmente una imagen con colores planos (como un icono) tendrá un peso menor si se guarda en GIF respecto de si es guardada en JPG. Lo contrario ocurrirá con una imagen con muchos colores diversos (como una foto). Se recomienda probar ambos formatos para determinar el óptimo.
- Ubicación de imágenes: se recomienda usar un solo directorio para almacenar las imágenes repetidas, tales como los iconos y otros elementos gráficos que son utilizados en diferentes páginas del sitio. Al ubicarlos en un directorio único se puede aprovechar la función de caché del programa visualizador para mejorar el rendimiento de las páginas. Para efectos de seguridad, se recomienda impedir que un programa visualizador pueda ver el contenido de dicho directorio o cualquier otro dentro del sitio.
- Usar el atributo ALT en imágenes: en el código HTML se debe usar el atributo ALT (texto alternativo) en las imágenes para que el código se despliegue antes que las imágenes y facilite de esta forma la comprensión del contenido a los usuarios.
- Imágenes con alto y ancho: las imágenes (dibujos, fotos, iconos, botones) deben tener tamaño para el ancho y el alto, para que el programa visualizador pueda dejar reservado el espacio para dicho contenido antes de que se realice su despliegue visual.
- Ofrecer plug-ins: cuando se utilizan archivos con multimedia que requiere el uso de plug-ins (programas visualizadores especiales) para revisarlos, se recomienda poner el programa para ser bajado u ofrecer un enlace a lugares para obtenerlo. Esto es especialmente válido en sitios que ofrecen presentaciones de portada en tecnología Flash, las cuales deben ser anunciadas para que el usuario tenga la opción de verlas o avanzar directo al sitio.



- Indicar el peso de los archivos: cuando se ofrecen elementos gráficos o audiovisuales para que el usuario lo baje por medio de la computadora (especialmente Video, Audio, imagen u otros), se recomienda indicar el peso de los mismos, con el objeto de ofrecerle información útil para efectuar la operación.
- Interoperabilidad.
- Los sitios web pueden accederse sin problemas desde computadoras que utilizan diferentes sistemas operativos, en un sitio Electrónico se debe cuidar ese aspecto de la diversidad. Para ello se debe asegurar que desde la mayor parte de ellos las páginas pueden verse sin mayores contratiempos.
- Para asegurar esto, las recomendaciones son las siguientes:
- Utilizar código HTML estándar, no mejorado para un visualizador en especial.
- Probar el sitio con las versiones para diferentes sistemas operativos de diversos visualizadores de páginas (browsers); especialmente hacerlo con versiones de Microsoft Internet Explorer, Netscape Communicator, Mozilla, Opera y Safari.
- Asegurarse de que el sitio puede ser visualizado de alguna forma cuando no se cumplen ciertas condiciones mínimas, por ejemplo, cuando se usan versiones antiguas de un programa visualizador que no soporta las nuevas características del lenguaje HTML (por ejemplo Netscape Navigator versión 4.6).

Estándares Internacionales.

Además de las buenas prácticas basadas en la experiencia, la tecnología web cuenta con un conjunto de estándares que deben ser respetados para obtener la certificación que acredita al sitio respecto de su conformidad con ellos.



Capítulo IV. Diseño, Programación e Implantación.



La entidad encargada del tema es el Word Wide Web Consortium (<http://www.w3c.org>) que se encarga de velar por las mejoras en la tecnología y por hacer avanzar los estándares y que los programas visualizadores cumplan efectivamente con mostrar lo que el lenguaje HTML permite construir.

Dentro de los estándares que más interesa cumplir, se cuentan dos que tienen que ver con la forma de hacer la presentación de contenidos en un Sitio Web y que se detallan a continuación:

Validación de HTML. Es un sistema basado en Internet y presentado en el propio sitio del W3C (<http://validator.w3.org/>) que permite detectar errores en la forma de utilizar el lenguaje HTML y XML en la construcción de un Sitio Web. Lo interesante del programa es que muestra en detalle los errores del código en la página que se prueba, con lo cual se puede llegar a una directa corrección de los problemas que se hayan detectado.

La importancia de tener un código correctamente validado es que se asegura, a partir de esa certificación, que la página web puede ser vista sin problemas, desde cualquier programa visualizador que cumpla con los estándares internacionales en la materia.

Validación de CSS. Es un sistema basado en Internet y presentado en el propio sitio del W3C (<http://jigsaw.w3.org/css-validator/>) mediante el cual se puede validar la sintaxis de una Hoja de Estilo en Cascada (Cascade Style Sheet o CSS, en inglés), mediante la cual se describe la forma de presentar contenidos en una página web.

Este programa muestra en detalle los errores del CSS en la página que se prueba, con lo cual se pueden aislar los problemas y hacer la corrección correspondiente. Cabe indicar que la ventaja de usar la tecnología CSS es que facilita la manutención de un sitio mediante la separación de la presentación (diseño) del contenido.



4.2.1. Diseño General del Portal.

El Sitio Web a desarrollar emplea 3 capas lógicas:

- Capa de presentación o de interfaz de usuario.
- Capa de procesamiento para la recuperación de información en la base de datos y el mantenimiento del Sitio.
- Capa de datos.

La información a procesar tendrá el siguiente recorrido; el servidor de aplicación o servidor Web(Apache Tomcat), procesará las peticiones de los clientes de Internet devolviendo páginas HTML estáticas, las peticiones se realizarán al servidor de base de datos MySQL, procesando los resultados y construyendo las páginas HTML dinámicas que finalmente visualizará el usuario de Internet.

El uso de una página dinámica con acceso a datos nos permitirá interactuar con la información de la base de datos ya sea para obtener información y mostrarla al usuario o bien para actualizar su contenido.

4.2.2. Diseño de la Página Web Estática y Dinámica.

La pantalla de inicio del Portal mostrará la misión, la visión, objetivo y su lema de la Academia Regional de Informática. Ver figura 4.2.2.1.



Capítulo IV. Diseño, Programación e Implantación.



Misión

La Academia Regional de Informática del Distrito Federal, esta conformada por un grupo de docentes profesionales dedicados a organizar, planear, dirigir, eventos académicos, exposiciones, reuniones de trabajo, cursos de actualización, con el objetivo principal, de formar alumnos de espíritu crítico, que con bases sólidas en el área tecnológica, contribuyan al desarrollo laboral del país.

Visión

La Academia Regional de Informática del Distrito Federal, se proyecta como una plantilla de docentes profesionales, que con aplicación de planes y programas del Bachillerato Tecnológico, garantiza para sus egresados una formación tecnológica de calidad, competente para la resolución de problemas, permitiéndoles afrontar exitosamente las exigencias no sólo del ámbito educativo, sino del social y laboral.

“SUPERATIO TECHNICUM, HUMANITATEM”

Figura 4.2.2.1. Página de Inicio.

El Portal utilizara un menú principal y un secundario. Ver figura 4.2.2.2.



Figura 4.2.2.2. Menú del Portal.

El menú principal está compuesto por:

- Académico ver Figura 4.2.2.3.

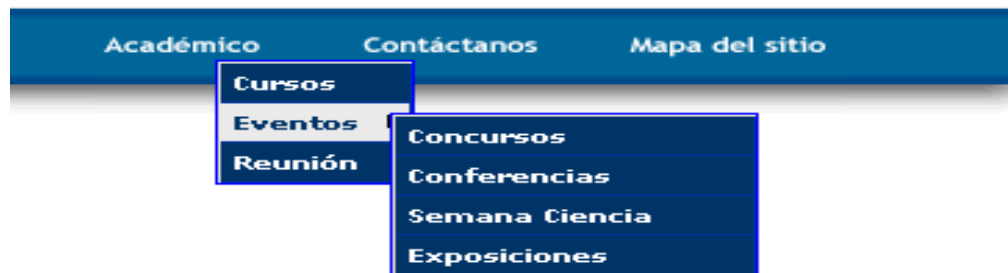


Figura 4.2.2.3. Submenús de Académico.



- Cursos. Consulta a la base de datos, de la tabla curso. Ver figura 4.2.2.4.

| CURSOS | | | | |
|--------------------------------|--------------------------|----------|---------|------|
| Nombre | Profesor | Duración | Plantel | Cupo |
| C# para principiantes.... | Ramón Oceja | 20 | 12 | 25 |
| Control de Calidad en la Indus | Bernardino Sierra Merino | 20 | 35 | 25 |
| Programación Avanzada de JavaS | Christian Cartori | 20 | 50 | 30 |
| Química para principiantes | Mario Pintor Quiroz | 20 | 23 | 20 |

Figura 4.2.2.4. Pantalla de Curso.

- Eventos

- Concursos. Consulta a la base de datos, de la tabla eventos. Ver figura 4.2.2.5.



CONCURSOS

Series y más Series ...

Se llevará a cabo en el CETIS 21

Iniciará el 04/11/2008 hasta el 10/12/2008

En el Auditorio 2 del Edificio A, dando inicio a las 10:30:00 horas.

Figura 4.2.2.5. Pantalla de Concursos.

- Conferencias. Consulta a la base de datos, de la tabla eventos. Ver figura 4.2.2.6.



Capítulo IV. Diseño, Programación e Implantación.



ACADEMIA REGIONAL DE LA DGETI

Académico

Contáctanos

Mapa del sitio



CONFERENCIAS

Matemáticas Finitas ¿Para qué?

Se llevará a cabo en el CETIS 32
Que dará inicio el 21/11/2008 hasta el 21/11/2008
en el Auditorio Principal, la conferencia dará inicio a las 16:00:00

Congreso magistral de matemáticas e ingeniería

Se llevará a cabo en el CETIS 45
Que dará inicio el 10/09/2008 hasta el 11/09/2008
en el Auditorio Principal, la conferencia dará inicio a las 12:00:00

Quinta feria de matemáticas

Se llevará a cabo en el CETIS 32
Que dará inicio el 21/04/2008 hasta el 25/05/2009
en el Auditorio Principal, la conferencia dará inicio a las 18:00:00

Figura 4.2.2.6. Pantalla de Conferencia.

- Semana Ciencia y Tecnología. Consulta a la base de datos, de la tabla eventos. Ver figura 4.2.2.7



ACADEMIA REGIONAL DE LA DGETI

Académico

Contáctanos

Mapa del sitio



Semana Ciencia y Tecnología

Se llevará a cabo en el CETYS 13
El día 2008-07-24 hasta el día 2008-07-27
Se llevará a cabo en el Explanada Principal, gran inauguración a las 13:00:00

Figura 4.2.2.7. Pantalla Semana de Ciencia y tecnología.

- Exposiciones. Consulta a la base de datos, de la tabla eventos. Ver figura 4.2.2.8



Portal de la Academia Regional de Informática de la DGETI.



ACADEMIA REGIONAL DE LA DGETI

Académico

Contáctanos

Mapa del sitio



EXPOSICIONES

Exposición de Tecnologías

Se llevará a cabo en el CETyS 3

El día 2008-07-08 hasta el día 2008-07-15

En el cetyis 34, la exposición dará inicio a partir de las 13:00:00

Figura 4.2.2.8. Pantalla Semana de Exposiciones.

- Reunión. Consulta los datos generados en Asamblea. Para mantener informados los puntos a tratar o tratados durante la reunión. Ver Figura 4.2.2.9.



ACADEMIA REGIONAL DE LA DGETI

Académico

Contáctanos

Mapa del sitio



TERCER ASAMBLEA

Se llevará a cabo en el 335, el día 25-08-2008

Los puntos que se trataron fueron los siguientes: Revisión de calendario de nuevas actividades.

Los puntos a tratar son los siguientes Acuerdos de nuevos planes de estudio.

Figura 4.2.2.9. Pantalla Asamblea.

- Contáctanos. Despliega el módulo de outlook, que permitirá enviar correos al administrador. Ver figura 4.2.2.10.



Capítulo IV. Diseño, Programación e Implantación.

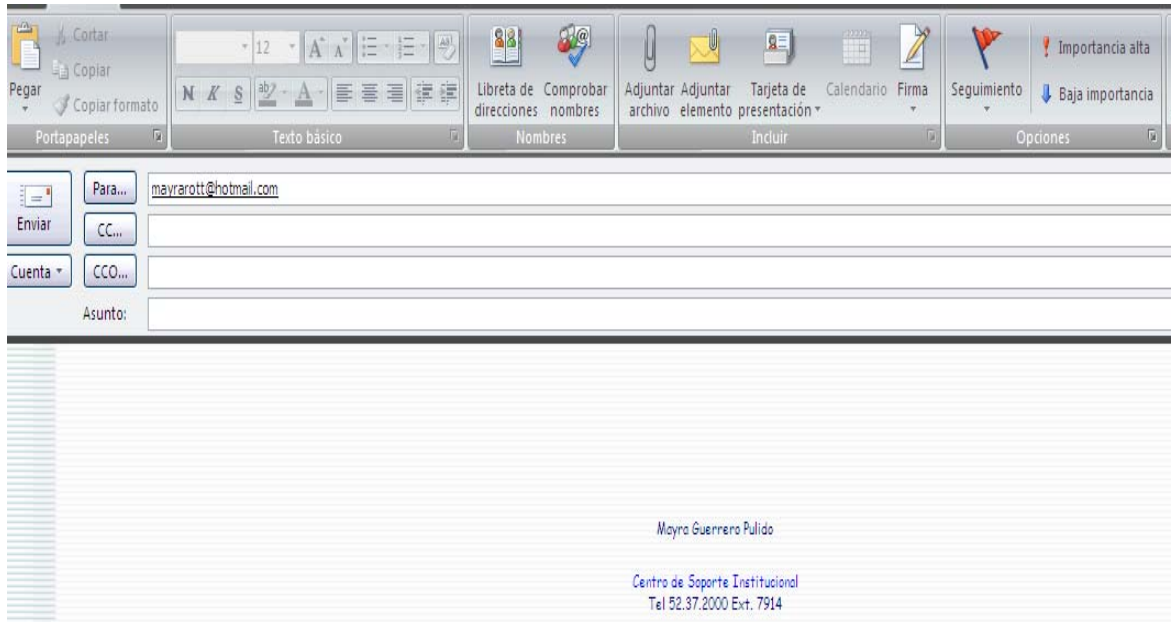


Figura 4.2.2.10. Contáctanos.

- Mapa del Sitio. Organigrama del sitio. Ver Figura 4.2.2.11



Figura 4.2.2.11. Pantalla Mapa del Sitio.



Portal de la Academia Regional de Informática de la DGETI.



Menú secundario contiene:

- Material de Apoyo. Contiene Las Secuencias Didácticas. Ver Figura 4.2.2.12.



Figura 4.2.2.12. Material de Apoyo.

- Directorio. Información principal de las dependencias de la DGETI (CETIS). Ver Figura 4.2.2.13.



[Siguiete](#)

Figura 4.2.2.13. Directorio.



Capítulo IV. Diseño, Programación e Implantación.



- Preguntas Frecuentes. Preguntas sobre la página que ayudaran al usuario a resolverle algunas dudas. Ver figura 4.2.2.14.



Figura 4.2.2.14. Pantalla Preguntas Frecuentes.

- Inicio. Regresa a la pantalla Principal. Ver figura 4.2.2.15. Consulta a la base de datos, en tabla Reunión, contenido de los datos actualizados de la próxima Asamblea, para informar a los integrantes de la Academia.



Figura 4.2.2.15. Pantalla Principal.



Este diseño es la pantalla de inicio, la cual se desarrollo como se explicó en capítulos anteriores con la finalidad de mantener informada a la comunidad que integra a esta dependencia de los acontecimientos que se desarrollan periódicamente en cada una de las instituciones integrantes, disminuyendo los tiempos así como modificando el procedimiento y manejo de la información.

4.3. Diseño de la Base de Base de Datos.

Diagrama de Procesos Nivel 0: Diagrama de Contexto.

Este es un caso especial de diagrama de flujo de datos, en donde una sola burbuja representa todo el sistema. Enfatiza tanto el personal, organizaciones, sistemas, los datos que el sistema recibe y produce, así como el almacenamiento de los mismos y la frontera existente entre el sistema y el resto del mundo. Este diagrama consiste en terminadores, flujos de datos, flujos de control, almacenes de datos y un solo proceso que representa a todo el sistema.

En la figura 4.3.1. se muestra el Diagrama de Contexto (DC), donde el proceso único representa el portal de la Academia de Informática. Los agentes externos son el administrador que se encargara de la actualización de la Base de Datos y los usuarios que utilizaran el portal como medio informativo.

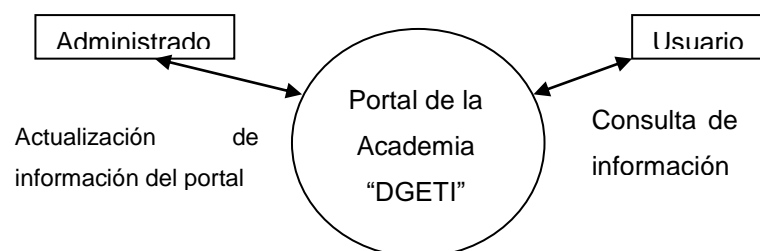


Figura 4.3.1 Diagrama de Contexto.



Diagramas de Flujo de Procesos.

Diagrama de Procesos Nivel 1: Portal de la Academia Regional de Informática de la DGETI.

Este diagrama permite identificar los procesos que definen la funcionalidad del sistema, junto con sus agentes externos y repositorios de información. Figura 4.3.2.

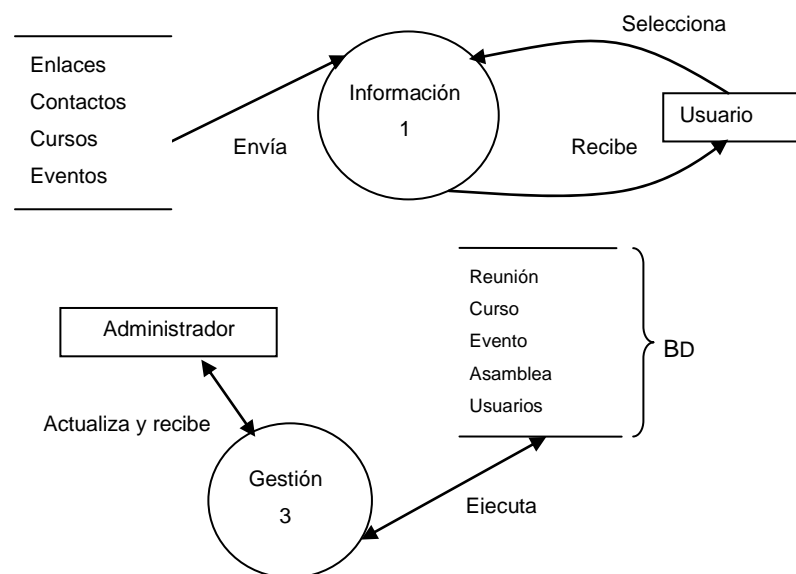


Figura 4.3.2. Diagrama de procesos Nivel 1: Portal de la Academia Regional de Informática de la DGETI.

Información: Es el proceso donde el usuario interactúa con el portal para obtener la información actualizada.

Gestión: Proceso de actualización que realiza el administrador para dar mantenimiento a la información que cada cierto tiempo debe de modificarse.

Usuario: entidad externa que hará uso de la información que se publicará en el Portal.

Administrador: Persona encargada de gestionar la información que será publicada en el Portal.



Diagrama de Procesos Nivel 2: Consultas.

Procesos que se ven involucrados al realizar una consulta dentro del Portal. Figura 4.3.3.

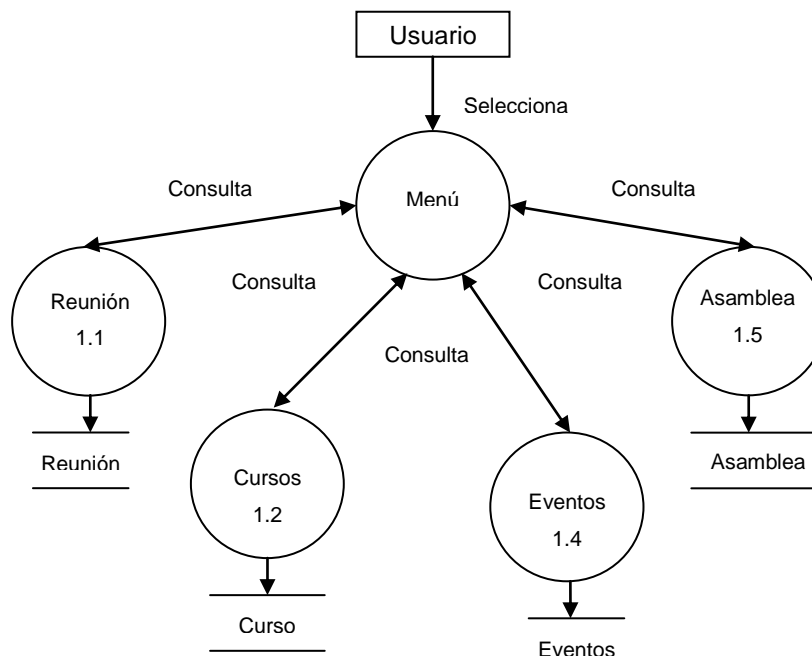


Figura 4.3.3. Diagrama de procesos Nivel 2: Consultas.

Este diagrama describe como la entidad usuario por medio del subproceso Menú del portal lleva a cada subproceso realizar una consulta que corresponde a la información de su interés.

Reunión: Subproceso que genera la consulta a la tabla Reunión para obtener la información de las diferentes reuniones que se efectuaran en todos los planteles.

Cursos: Subproceso que consulta los cursos semestrales dados de alta por el administrador; donde el usuario podrá adquirir la información de éste. (Fecha de



inicio, finalidad, nombre del curso, duración, lugar donde se impartirá, nombre del usuario, clave, etc.)

Eventos: Subproceso que muestra la información de los eventos mensuales y de importancia para la Academia.

Asamblea: Subproceso que muestra la información de la Asamblea (Hora, Fecha, y sede donde se realizará; así como la información de interés que se genera después de éstas.

Diagrama de Procesos Nivel 2: Gestión.

Procesos que se ven involucrados al realizar una actualización de información del Portal. Figura 4.3.4. Este diagrama describe como la entidad administrador por medio del subproceso Menú del gestor lleva a cada subproceso para realizar una consulta e interactuar con la base de datos.

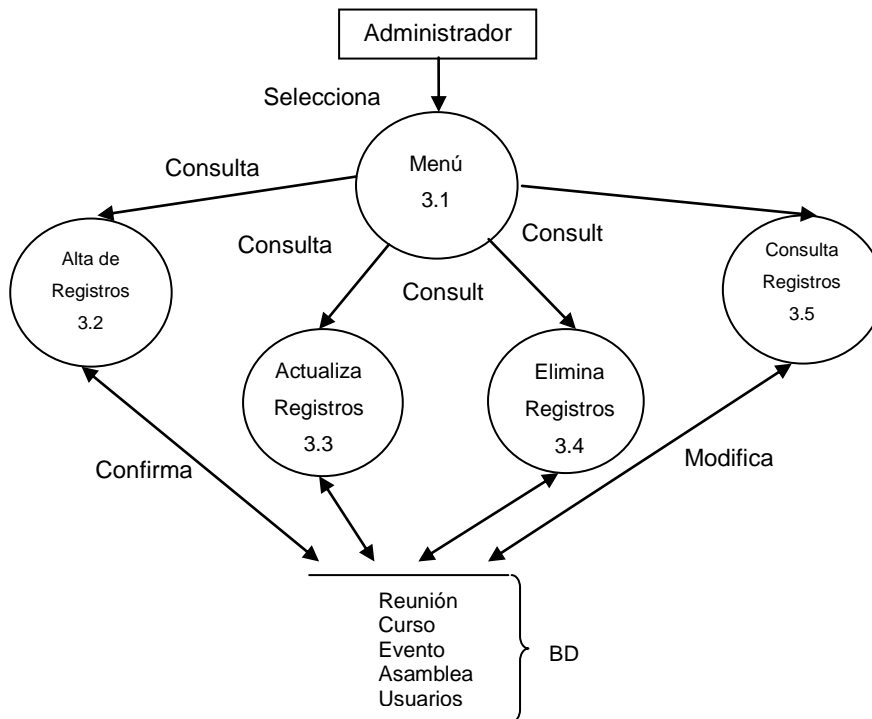


Figura 4.3.4. Diagrama de procesos Nivel 2: Gestión.



Menú. Proceso donde se muestra la página principal del gestor y en el cual el administrador podrá elegir la tabla que desee modificar.

Alta de Registros. Proceso de captura de datos de los nuevos eventos, reuniones cursos, etc. que se vayan a realizar durante un lapso de tiempo estipulado.

Actualiza Registros. Proceso donde se muestran los registros de una tabla en específico, el cual permite modificar los registros.

Elimina Registros. Proceso donde se muestran los registros de una tabla en específico, para poder eliminarlos.

Consulta Registros. Proceso que devuelve la información que hay dentro de una tabla.

Administrador. Entidad externa encargada de administrar los procesos antes mencionados.

Diagrama de Procesos Nivel 3: Gestión.

Alta.

En el siguiente diagrama se describe con detalle los subprocesos que se llevaran a cabo para agregar un nuevo registro dentro de la tabla seleccionada por el Administrador. Ver figura 4.3.5.

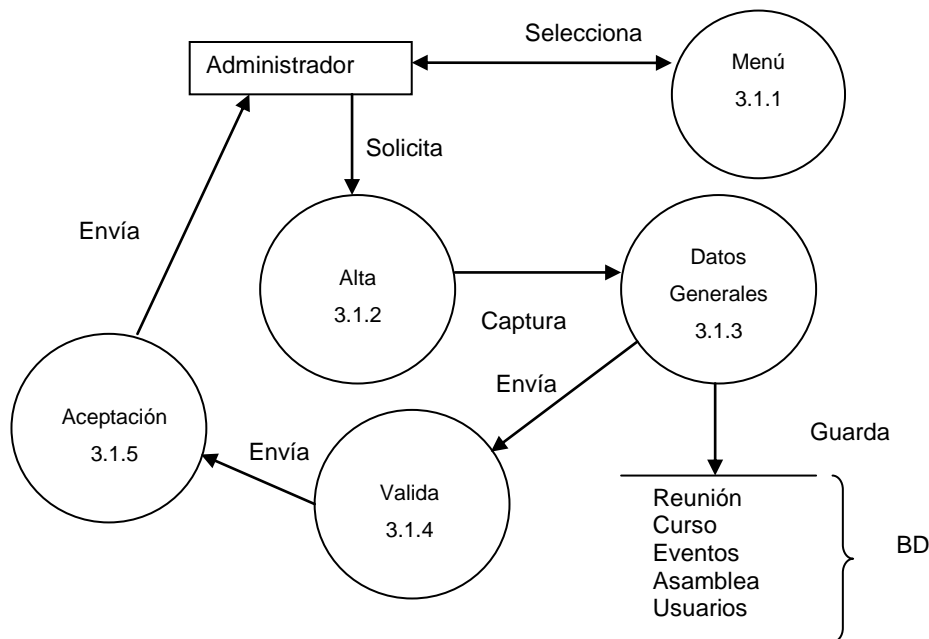


Figura 4.3.5. Diagrama de Procesos Nivel 3. Gestión (Alta).

Subproceso **menú**, el administrador podrá elegir la tabla en la que desee realizar el proceso de modificaciones.

Subproceso **alta**, el administrador ha solicitado una vez que ha elegido la tabla agregar un registro nuevo.

Subproceso **datos generales**, en esta parte el administrador puede capturar los datos necesarios para agregar un nuevo registro el cual se encargará de guardar los datos correspondientes a las tabla de su elección.

Subproceso **valida**, se encarga de hacer la validación de que el tipo de dato que se está introduciendo corresponda al tipo de dato esperado.

Subproceso **aceptación**, es el que recibe por parte de valida la confirmación de la tabla, se encarga de enviar la notificación al administrador.



Baja.

En el siguiente diagrama se describe detalladamente los subprocesos que se llevaran a cabo para eliminar un registro dentro de la tabla seleccionada por el Administrador. Ver figura 4.3.6.

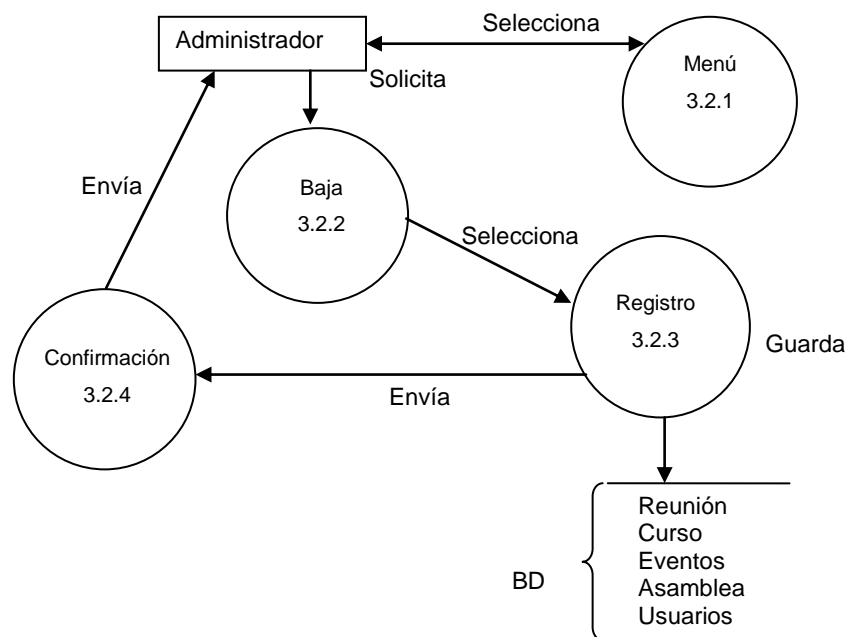


Figura 4.3.6. Diagrama de Procesos Nivel 3. Gestión (Baja).

Subproceso **menú**, el administrador podrá elegir la tabla en la que desee hacer las modificaciones correspondientes.

Subproceso **baja**, el administrador ha solicitado una vez que ha elegido la tabla eliminar un registro.

Subproceso **registros**, en el cual se selecciona el registro que se desea eliminar y el que se encargará de borrar los datos correspondientes a las tabla de su elección.

Subproceso **confirmación**, éste recibe por parte de registros la confirmación de la tabla, y es el encargado de enviarle la notificación al administrador.



Actualiza.

En el siguiente diagrama se describe detalladamente los subprocesos que se llevarán a cabo para actualizar un registro dentro de la tabla seleccionada por el Administrador. Ver figura 4.3.7.

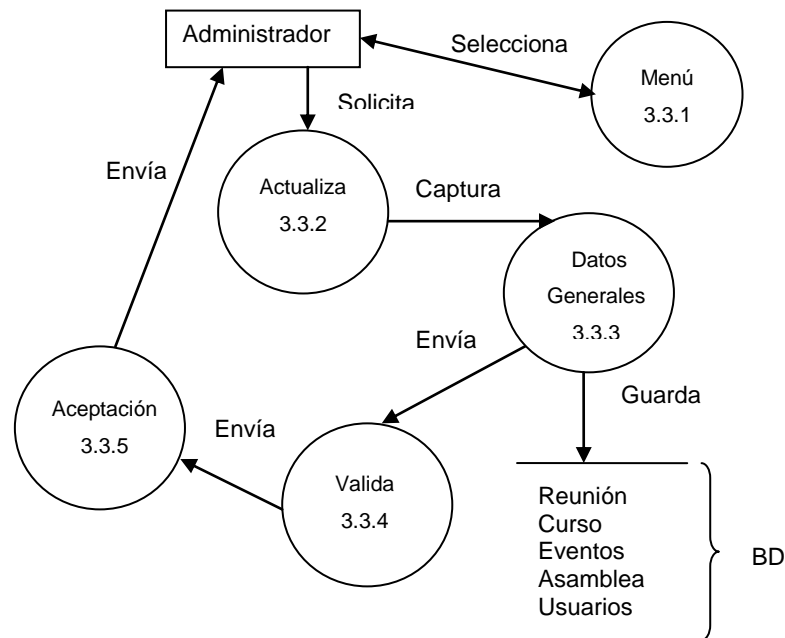


Figura 4.3.7. Diagrama de Procesos Nivel 3. Gestión (Actualiza).

Subproceso **menú**, el administrador podrá elegir la tabla en la que desee hacer sus modificaciones.

Subproceso **actualiza**, el administrador ha solicitado una vez que ha elegido la tabla modificar un registro dado de alta anteriormente.

Subproceso **datos generales**, en éste se devuelven todos los registros que contiene la tabla que se seleccionó y el cual irá a guardar los datos correspondientes a la tabla.

Subproceso **valida**, se encargará de hacer la validación de que el tipo de dato que se está introduciendo corresponda al tipo de dato esperado.



Subproceso **aceptación**, se recibe por parte de valida la confirmación de la tabla, y es el encargado de enviarle la notificación al administrador.

Consulta.

En el siguiente diagrama se describen con detalle los subprocesos que se llevaran a cabo para actualizar un registro dentro de la tabla seleccionada por el Administrador. Ver figura 4.3.8.

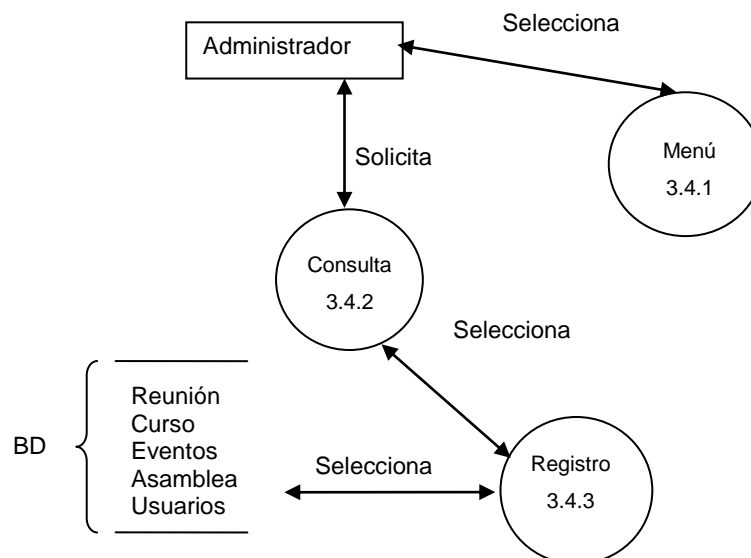


Figura 4.3.8. Diagrama de Procesos Nivel 3. Gestión (Consulta).

Subproceso **menú**, el administrador podrá elegir la tabla en la que desee hacer sus consultas.

Subproceso **consulta**, el administrador ha solicitado una vez que ha elegido la tabla consultar un registro dado de alta anteriormente.

Subproceso **registro** en él se devuelven todos los registros que contiene la tabla que se seleccionó.



Diagrama Jerárquico de Procesos.

De acuerdo al análisis anterior se tiene finalmente una estructura ordenada jerárquicamente de los procesos y subprocesos del portal de la Academia detallándose en la Figura 4.3.9.

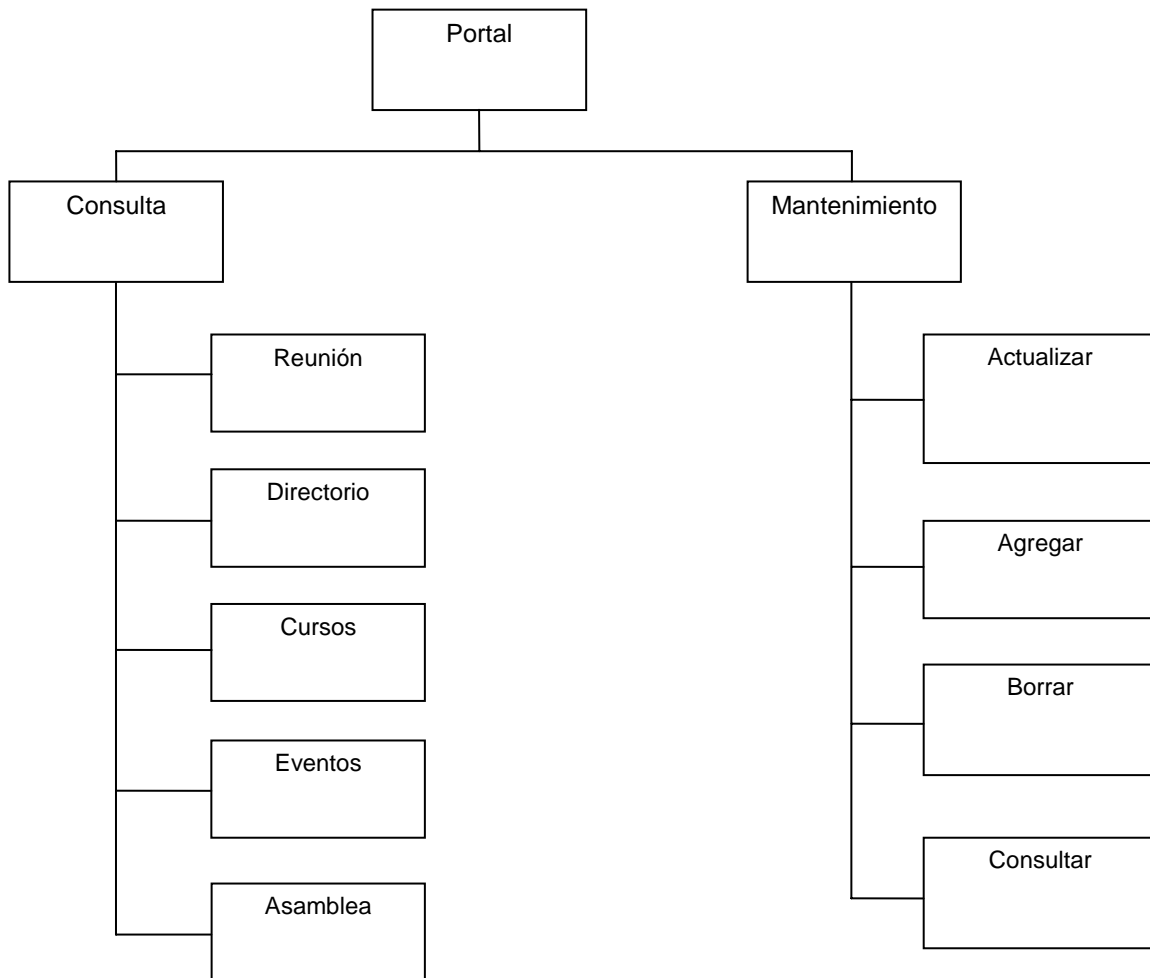


Figura 4.3.9 Diagrama jerárquico de procesos.



4.3.1 Diagrama de Flujo de Datos Generales.

Un diagrama de flujo es un esquema gráfico que muestra los pasos o procesos a seguir para la solución de un problema. Es una manera de especificar los detalles algorítmicos de un proceso.

Se llama diagrama de flujo porque los símbolos empleados se conectan por medio de flechas para indicar la secuencia de operación. Como en la mayoría de las herramientas se estableció una norma para que la simbología fuera universal. La simbología es variable y se ajusta a cierto patrón definido.

El diagrama de flujo de datos, es una herramienta que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por “conductos” y “elementos de almacenamiento” de datos. No solo se emplean para modelar sistemas de procesos de información, sino también como manera de modelar organizaciones enteras, esto es, como una herramienta para la planeación estratégica y de negocios.

En los diagramas de flujo se presuponen los siguientes aspectos:

- Existe siempre un camino que permite llegar a una solución (finalización del algoritmo).
- Existe un único inicio del proceso.
- Existe un único punto de fin para el proceso de flujo (salvo del rombo que indica una comparación con dos caminos posibles).



Validación de Usuario y Opciones de Administración.

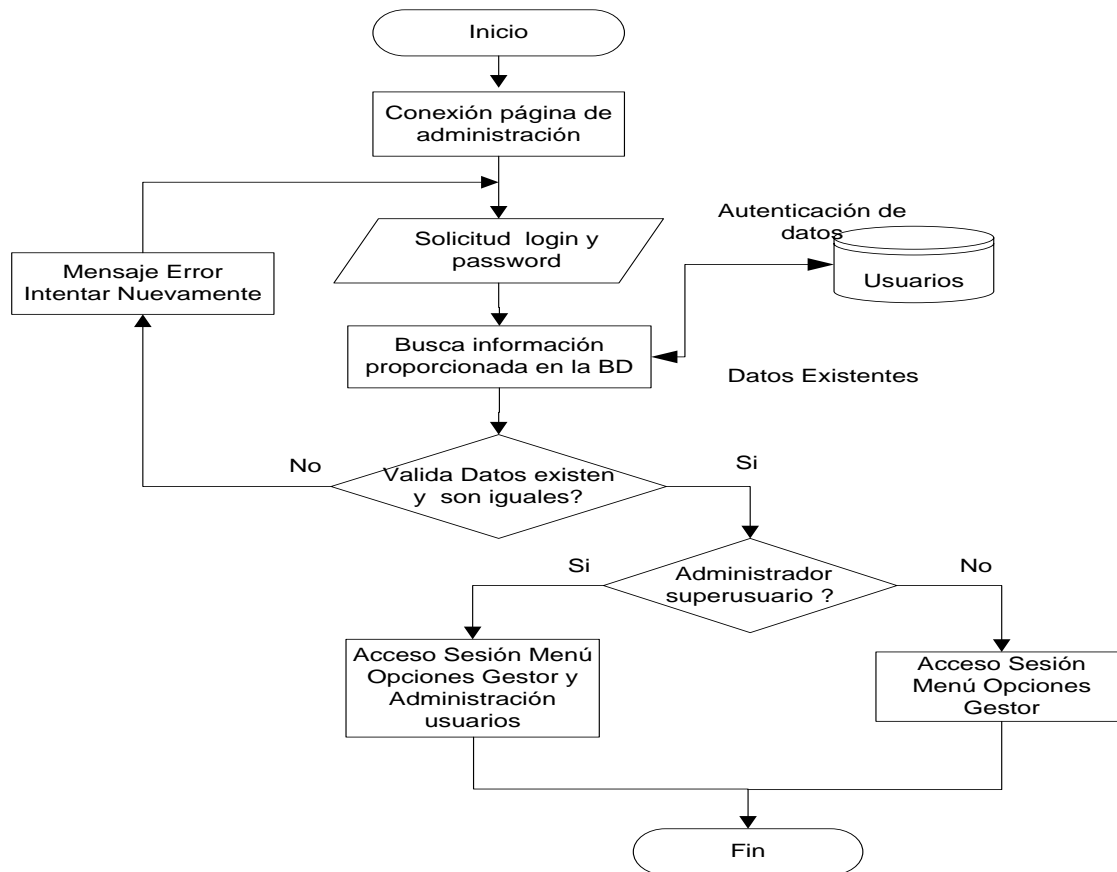


Figura 4.3.1.1 Diagrama de Flujo Autenticación de Usuario.

La descripción de la figura 4.3.1.1 es la siguiente: Cuando el usuario valida una sesión y accede al gestor el sistema verifica la existencia de los datos proporcionados en la tabla usuarios de la base de datos la cual realiza la búsqueda de usuarios dados de alta y la validación de que los datos que se introdujeron son iguales. Si los datos son correctos e iguales se despliega la página con el menú de opciones disponibles. Además si este tiene privilegios de Administrador, desplegará un menú adicional que es la Administración de usuarios. Si los datos no son validos o no existen se solicita llenar el formulario nuevamente para acceder.



Diagrama de Flujo. Actualizar Gestor.

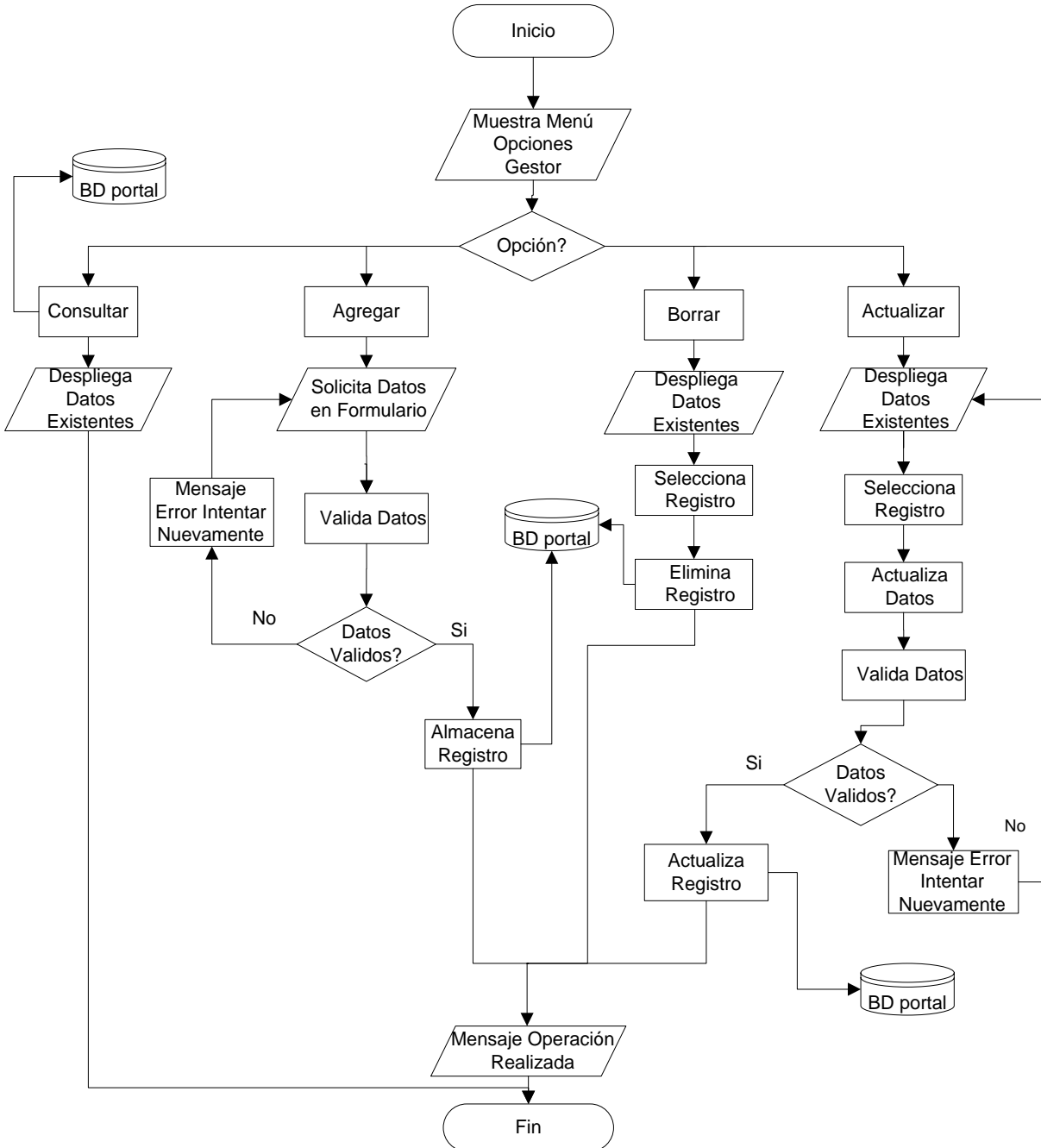


Figura 4.3.1.2. Diagrama de Flujo Actualizar Gestor.



Capítulo IV. Diseño, Programación e Implantación.



El seguimiento del proceso del diagrama de flujo de la Figura 4.3.1.2 se describe a continuación:

El usuario común podrá acceder al Menú de opciones para validar solo las tablas de asamblea, reunión, curso y evento de la base de datos.

Consultar. Despliegan los registros existentes en la base de datos de la tabla seleccionada.

Agregar. Permite mostrar en cada opción un formulario en el cual se solicita al usuario agregar datos en cada campo, estos datos se validan antes de ser enviados a la base de datos si cumplen con los requerimientos el registro se agrega a la base de datos, en caso contrario se muestra un mensaje de error y se solicita llenar el campo con datos validos.

Borrar. Permite borrar registros de cada tabla, al seleccionar el registro este procederá a borrar el registro almacenado en la base de datos. Posteriormente indica si el proceso se realizo o no.

Actualizar. Este proceso permite realizar cambios a los campos existentes de cada registro, despliega los registros almacenados en la base de datos, se realizan los cambios correspondientes en los campos a modificar, el sistema valida los datos, si son correctos estos se actualizan en la base de datos, en caso contrario se envía un mensaje pidiendo al usuario realizar las modificaciones correctamente.

El diagrama 4.3.1.3. es similar al diagrama anterior, con la única diferencia de que en la opción agregar esta tiene dos condicionales que valida si el registro que se ha insertado ya existe en la base de datos, si la respuesta es afirmativa muestra un mensaje de error en caso contrario valida los datos siguientes si estos son correctos se almacena el registro, en caso contrario mostrara un mensaje de error y regresará al formulario solicitando nuevamente los datos a insertar.



Diagrama de Flujo. Actualizar Administrador de Usuarios.

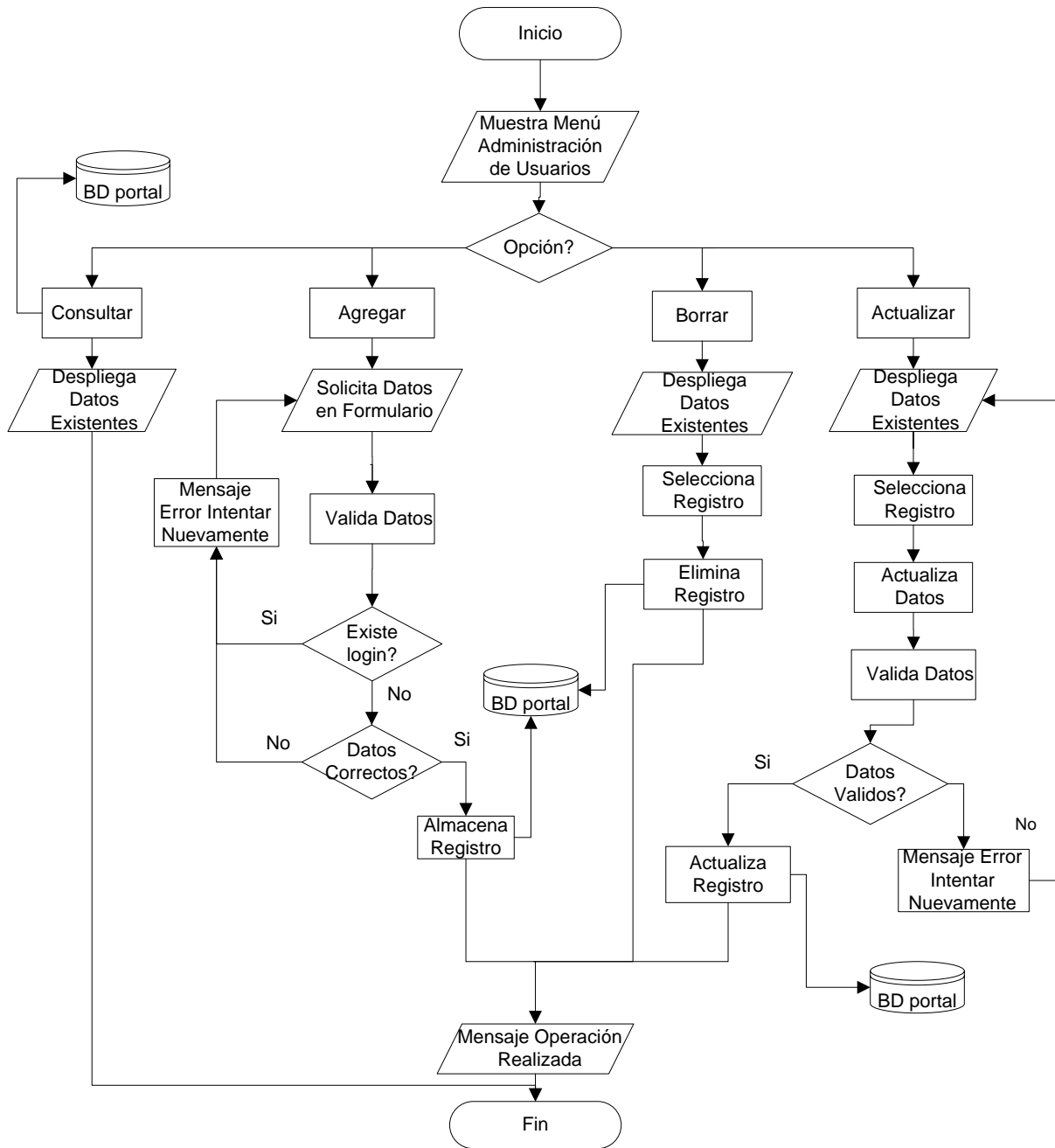


Figura 4.3.1.3. Diagrama de Flujo Actualizar Gestor.



4.3.2. Diccionario de Datos.

El diccionario de datos describe las bases de datos individuales y su entorno. Esta parte contiene descriptores que definen el uso, valor y significado de los datos.

Los diccionarios de datos como documentos normativos contienen las especificaciones particulares que rigen los procesos de producción de los datos. La base para elaborar los diccionarios de datos de los diferentes temas y escalas lo constituye el modelo de datos, el cual debe considerarse como el grupo de especificaciones mínimas de cumplimiento obligatorio para los generadores de información de los sistemas.

En las tablas siguientes 4.3.2.1 a 4.3.2.4 se presentan cada una de las tablas integrantes de la base de datos y sus correspondientes entidades. Cada una de las tablas o entidades mostrará los datos particulares de los atributos y la descripción de las mismas.

- Atributo. Propiedades de cada uno de los acrónimos. (Unsigned, unsigned zerofill, on update current_timestamp)
- Acrónimo. Nombre corto que se da a un atributo.
- Tipo. Tipo de Dato. Es la forma en la que se caracteriza de manera precisa el tipo de valores que puede contener o representar la columna especificada. Son muchos los tipos pero en este caso se definirán algunos como: caracter, numérico, fecha, hora, variable, etc.
- Varchar. Para texto de longitud variable. Cadena de longitud variable no binaria.
- Numeric. Para tipos de datos numéricos.
- Date. Para las columnas que contienen fechas. Valor de fecha en formato 'SSAA-MM-DD'



- Time. Para las columnas que contienen horas. Valor de fecha en formato 'hh:mm:ss'.
- Counter. Contador automático de cada registro por dar de alta.
- Long. Longitud de la cadena.
- Llave PK/FK. Indicador del atributo que permitirá identificar a la entidad con claves únicas e irrepetibles y permitirá relacionar datos y consultas con otras entidades. Pueden ser Primary Key o Foreign Key.
- Solic. Valor solicitado. Indicará si el valor es nulo o no. Se emplea para representar un valor "desconocido".
- Relación-Tablas. Asociación entre tablas. Indica la relación que existe en cada una de las tablas existentes.
- Descripción. Es una explicación breve de las entidades y atributos. Es probable que no todos los atributos tengan una definición ya que frecuentemente, el nombre del atributo es suficiente para definirlo.

Asamblea. Entidad que almacena la información de las asambleas a realizarse. Ver tabla 4.3.2.1.

| Atributo | Acrónimo | Tipo | Long | Llave | Solic. | Rela-Tabla | Descripción |
|---------------|----------|---------|---------|-------|----------|------------|--|
| Clave | Ab_clve | int | 10 | PK | not null | | Clave identificador de la Asamblea |
| Plantel | Ab_plan | int | 3 | PK | not null | | Plantel. Número de las instituciones existentes |
| Fecha | Ab_fech | date | default | PK | not null | | Fecha de la asamblea |
| Nombre | Ab_nomb | varchar | 80 | | not null | | Nombre de la Asamblea |
| Minuta | Ab_mnuta | varchar | 500 | | not null | | Minuta. Notas y puntos acordados en la reunión. |
| Orden del día | Ab_o_dia | varchar | 500 | | not null | | Orden del Día. Puntos importantes a discutir para llevar a cabo la reunión |

Tabla 4.3.2.1 Descripción Tabla Asamblea.



Capítulo IV. Diseño, Programación e Implantación.



Curso. Entidad que almacenará los datos de los cursos a impartir en el periodo. Ver tabla 4.3.2.2.

| Atributo | Acrónimo | Tipo | Long | Llave | Solic. | Relac Tabla | Descripción |
|---------------|----------|---------|---------|-------|----------|-------------|---|
| Clave | Cu-clve | int | 2 | PK | Not null | | Número que identificará el curso |
| Nombre | Cu_nomb | varchar | 100 | | not null | | Nombre del curso. |
| Profesor | Cu_prof | varchar | 60 | | Not null | | Nombre del profesor que impartirá el curso. |
| Duración | Cu_dura | int | 2 | | Not null | | Duración en horas del curso |
| Plantel | Cu_plan | int | 3 | | Not null | | Plantel dónde se impartirá el curso |
| Cupo | Cu_cupo | int | 2 | | null | | Cupo. Limite de personas por curso. |
| Fecha inicial | Cu_fchi | date | default | | Not nul | | Fecha en la que incia el curso |
| Fecha Final | Cu_fchf | date | default | | Not nul | | Fecha en la que finaliza el curso |
| Hora | Cu_hora | time | default | | Not null | | Hora en la el curso se realizará |

Tabla 4.3.2.2. Descripción Tabla Curso.

Evento. Entidad que almacenara los datos principales de cada uno de los eventos a realizar en cada plantel y en el periodo. Ver tabla 4.3.2.3.

| Atributo | Acrónimo | Tipo | Long. | Llave | Solic | Rela-Tabla | Descripción |
|-----------|----------|---------|---------|-------|----------|------------|---|
| Clave | Ev_clve | int | 4 | PK | not null | | Clave identificador del usuario |
| Número | PI_nmro | Int | 3 | | not null | | Número del plantel. |
| Nombre | Ev_nomb | Varchar | 100 | | not null | | Nombre del evento |
| Fecha ini | Ev_fchi | date | Default | | not null | | Fecha de inicio del evento |
| Fecha fin | Ev_fchf | date | Default | | null | | Fecha de fin del evento. |
| Sede | Ev_sede | varchar | 40 | | not null | | Lugar específico donde se llevará a cabo el evento. |
| Hora | Ev_hora | time | Default | | null | | Hora en que llevará a cabo el evento. |
| Tipo | Ev_tipo | varchar | 15 | | Not null | | Tipo de evento. |

Tabla 4.3.2.3. Descripción Tabla Evento.



Reunión. Entidad que almacenará los datos de cada una de las reuniones que se llevarán a cabo en el periodo. Ver tabla 4.3.2.4.

| Atributo | Acrónimo | Tipo | Long | Llave | Solic | Rela-Tablas | Descripción |
|----------|----------|---------|---------|-------|----------|-------------|--|
| Clave | Re_clve | Int | 4 | PK | not null | | Clave de la reunión |
| Fecha | Re_fech | date | default | | not null | | Fecha en que se llevará a cabo la reunión |
| Sede | Re_Sede | Varchar | 40 | | not nul | | Nombre de lugar donde se realizará la reunión. |
| Nombre | Re_nomb | varchar | 80 | | null | | Nombre de la reunión. |
| Hora | Re_hora | Time | Default | | | | Hora en que se llevará a cabo la reunión. |

Tabla 4.3.2.4. Descripción Tabla Reunión.

Usuarios. Entidad que almacenará los datos de las personas que tendrán permiso de acceder y manipular el gestor de la base de datos.

| Atributo | Acrónimo | Tipo | Long | Llave | Solic | Rela-Tabla | Descripción |
|---------------|-----------|---------|------|-------|----------|------------|--|
| Identificador | id | tinyint | 4 | | not null | | Clave única |
| Nombre | nombre | varchar | 30 | | not null | | Nombre del Usuario. |
| Apellidos | apellidos | varchar | 80 | | not null | | Apellidos del Usuario. |
| Login | login | varchar | 30 | PK | not null | | Identificación del Usuario. |
| Password | password | varchar | 30 | | not null | | Clave del Usuario |
| email | email | varchar | 80 | | not null | | Correo electrónico de usuario |
| Tipo Usuario | tipous | varchar | 5 | | Not null | | Privilegio como administrador general o común. |

Tabla 4.3.2.10. Descripción Tabla Usuarios.



4.3.3. Diagrama Entidad-Relación.

Un diagrama o modelo entidad-relación es una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información, sus inter-relaciones y propiedades. Es una representación conceptual de la información.

El modelo presenta una vista unificada de los datos, centrándose en la estructura lógica y abstracta de los mismos, como representación del mundo real, con independencia de consideraciones de tipo físico.

El modelo entidad-relación es el más ampliamente aceptado que se le da soporte en las herramientas CASE.

Brevemente consiste en los siguientes pasos:

- Se parte de una descripción textual del problema o sistema de información a automatizar (los requisitos).
- Se hace una lista de los sustantivos y verbos que aparecen.
- Los sustantivos son posibles entidades o atributos.
- Los verbos son posibles relaciones.
- Analizando las frases se determina la cardinalidad de las relaciones y otros detalles.
- Se elabora el diagrama (o diagramas) entidad-relación.



- Se completa el modelo con listas de atributos y una descripción de otras restricciones que no se pueden reflejar en el diagrama.

Este modelo se basa en entidades que se relacionan o asocian entre sí. La figura 4.3.3.1 ilustra el diagrama entidad-relación del sistema a implementar.

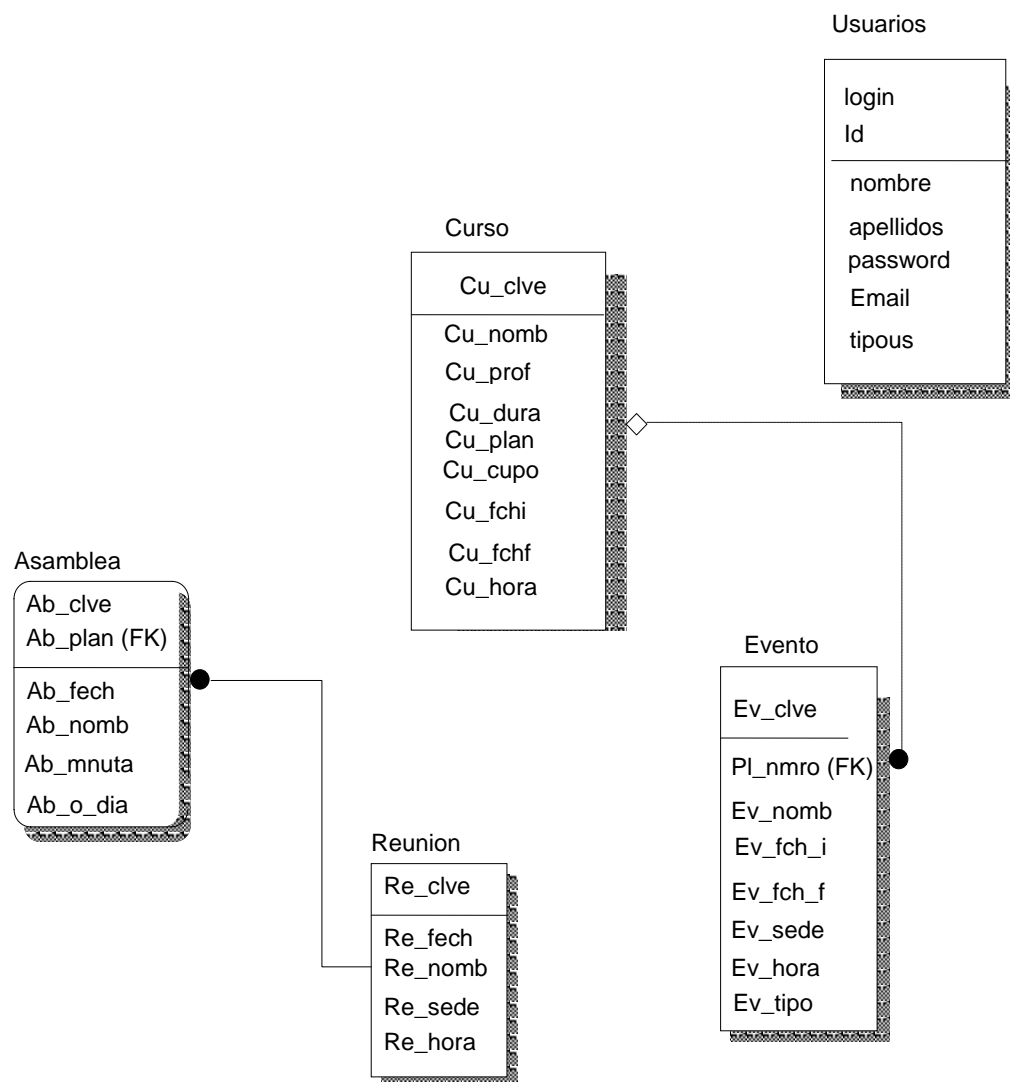


Figura 4.3.3.1 Diagrama Entidad-relación del Sistema.



4.4. Integración de Pruebas.

La prueba del software es un elemento crítico, su objetivo es garantizar la calidad del producto desarrollado. Además, esta etapa implica:

- Verificar la interacción de componentes y la integración adecuada de los componentes.
- Verificar que todos los requisitos se han implementado correctamente.
- Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el software al cliente.
- Diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.
- La prueba no es una actividad sencilla, tampoco es una etapa del proyecto en la cual se asegura la calidad, sino que esta debe ocurrir durante todo el ciclo de vida: permitiendo probar la funcionalidad de los primeros prototipos, la estabilidad, cobertura y rendimiento de la arquitectura además del producto final etc.
- La prueba es un proceso que se enfoca sobre la lógica interna del software y las funciones externas, así como de la ejecución de un programa con la intención de descubrir un error. La prueba no asegura la ausencia de defectos.
- Cualquier proceso de ingeniería puede ser probado por alguna de las formas siguientes: Prueba de la caja negra y prueba de la caja blanca.
- **Prueba de la caja negra:** Las pruebas se llevan a cabo sobre la interfaz del software, y es completamente indiferente el comportamiento interno y la estructura del programa.



Los casos de prueba de la caja negra pretende demostrar que:

- Las funciones del software son operativas. La entrada se acepta de forma adecuada.
- Se produce una salida correcta. La integridad de la información externa se mantiene.
- Se derivan conjuntos de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales del programa.

La prueba de la caja negra intenta encontrar errores de las siguientes categorías:

Funciones incorrecta o ausente.

Errores de interfaz.

Errores en estructuras de datos o en accesos a bases de datos externas.

Errores de rendimiento, inicialización y terminación.

- **Prueba de la caja blanca:** Permite examinar la estructura interna del programa. Se diseñan casos de prueba para examinar la lógica del programa. Es un método de diseño de casos de prueba que usa la estructura de control del diseño procedural para derivar casos de prueba que garanticen que:

Se ejercitan todos los caminos independientes de cada módulo.

Se ejercitan todas las decisiones lógicas.

Se ejecutan todos los bucles.

Se ejecutan las estructuras de datos internos.



Capítulo IV. Diseño, Programación e Implantación.



Los casos de prueba deben satisfacer los siguientes criterios:

Reducir el número de casos de prueba adicionales.

Que exista un indicador sobre la presencia o ausencia de clases de errores.

Tipos de Pruebas.

- **Prueba de unidad.** Se centra en el módulo. Usando la descripción del diseño detallado como guía, se prueban los caminos de control importantes con el fin de descubrir errores dentro del ámbito del módulo. La prueba de unidad hace uso intensivo de las técnicas de prueba de caja blanca.
- **Prueba de integración.** El objetivo es que con los módulos probados en la prueba de unidad, construir una estructura de programa que esté de acuerdo con lo que dicta el diseño.

Hay dos formas de integración:

Integración no incremental. Se combinan todos los módulos por anticipado y se prueba todo el programa en conjunto.

Integración incremental. El programa se construye y se prueba en pequeños segmentos.

En la prueba de integración el foco de atención es el diseño y la construcción de la arquitectura del software.

Las técnicas que más prevalecen son las de diseño de casos de prueba de caja negra, aunque se pueden llevar a cabo algunas pruebas de caja blanca.



Prueba del sistema. Verifica que cada elemento encaja de forma adecuada y que se alcanza la funcionalidad y el rendimiento del sistema total. La prueba del sistema está constituida por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora.

Algunas de estas pruebas son:

Prueba de validación. Proporciona cierta seguridad final de que el software satisface todos los requerimientos funcionales y de rendimiento. Además, valida los requerimientos establecidos comparándolos con el sistema que ha sido construido. Durante la validación se usan exclusivamente técnicas de prueba de caja negra.

Prueba de recuperación. Obliga a un fallo del software y verifica que la recuperación se lleva a cabo apropiadamente.

Prueba de seguridad. Verificar los mecanismos de protección.

Prueba de resistencia. Enfrenta los programas a situaciones anormales.

Prueba de rendimiento. Prueba el rendimiento del software en tiempo de ejecución.

Prueba de instalación. Centrado en asegurar que el sistema de software desarrollado se puede instalar en diferentes configuraciones de hardware y software bajo condiciones de excepciones, por ejemplo con espacio de disco insuficiente o continuas interrupciones.

Pruebas de regresión. Las pruebas de regresión son una estrategia en la cual las pruebas que se han ejecutado anteriormente se vuelven a realizar en la nueva versión modificada, para asegurar la calidad después de añadir la nueva funcionalidad.



Capítulo IV. Diseño, Programación e Implantación.



En concreto, los objetivos de la estrategia de prueba son:

Planificar las pruebas necesarias en cada iteración, incluyendo pruebas de unidad, integración y de sistema. Las pruebas de unidad y de integración son necesarias dentro de la iteración, mientras que las pruebas de sistema son necesarias sólo al final de la iteración.

Diseñar e implementar las pruebas creando los casos de prueba que especifican qué probar, cómo realizar las pruebas y creando, si es posible, componentes de prueba ejecutables para automatizar las mismas.

Pruebas al Portal de la Academia Regional de Informática.

Algunas pruebas que se le practicaron al gestor fueron las siguientes:

- Las condiciones principales fueron: Tener el software Apache y MySQL ejecutándose, instalada la última versión del Gestor.

Prueba de Acceso.

Datos de entrada:

Contraseña y usuario correctos para entrar al gestor.

Caso 1:

Paso 1: Ingresar la ruta del Gestor.

Paso 2: Capturar el usuario y contraseña **existentes en la base de datos.**



Resultados obtenidos:

El formulario que solicita el login y password se conecta con MySQL. **Figura 4.4.1**

La imagen muestra una interfaz web con un encabezado que dice "Gestor del Portal" en un recuadro azul. Debajo, hay un logo con las letras "AR" y el texto "Inicia sesión". Hay dos campos de entrada: "Login:" con el texto "superusuario" y "Password:" con caracteres ocultos por puntos. Un botón "Entrar" está centrado debajo de los campos.

Figura 4.4.1. Ingreso de login y password para acceder al gestor.

Una vez capturado el login y password correcto, abre la sesión para entrar a la página principal del gestor.

Si el usuario es el administrador principal además del menú de opciones de las tablas muestra el menú de administración de usuarios. Si el usuario es común se muestra solo el menú de opciones de las tablas. Ver Figura 4.4.1.1.



Administrador Superusuario



Usuario Común

Figura 4.4.1.1. Menú de Opciones SuperUsuario y Usuario Común.

Caso 2:

Paso 1: Ingresar la ruta del Gestor.

Paso 2: Capturar el usuario y contraseña no existentes o con errores.

Resultados obtenidos:



El formulario que solicita el login y password intenta conectarse a la Base de Datos de MySQL.

Logo de la Academia Regional de Informática de la DGETI

Inicia sesión

Login:

Password:

Figura 4.4.2. Ingreso de zona restringida.

Prueba de Consultas, Altas, Bajas y Cambios en las tablas.

Datos de entrada:

Seleccionar la tabla a Editar.

Caso 1:

Paso 1: Realizar una consulta desde el administrador (Phpmyadmin) de la tabla Asamblea

Paso 2: Entrar a la página principal del gestor.

Paso 3: Ingresar a tabla Asamblea

Paso 4: Seleccionar Consultar. Ver Figura 4.4.3.



Capítulo IV. Diseño, Programación e Implantación.



Show: 30 row(s) starting from record # 0
 in horizontal mode and repeat headers after 100 cells
 Sort by key: None

| | Ab_clve | Ab_plan | Ab_fech | Ab_nomb | Ab_mnuta | Ab_o_dia |
|--------------------------|---------|---------|------------|---|------------|---|
| <input type="checkbox"/> | 83 | 33 | 30-11-2008 | Tercera Asamblea Estatal | Pendiente | Basificación Nivel A |
| <input type="checkbox"/> | 154 | 25 | 23/09/2008 | Cuarta Asamblea Ordinaria | Pendiente | Lista Asistencia, Evaluación Final Semestre |
| <input type="checkbox"/> | 152 | 24 | 21-09-2008 | Revisión Plenario de Academia de Matemática | En proceso | Lista de Asistencia, Asignación de plazas |
| <input type="checkbox"/> | 153 | 54 | 03-09-2008 | Primera Reunión | Pendiente | Propuesta cambio de directivos |

Check All / Uncheck All With selected:

Show: 30 row(s) starting from record # 0
 in horizontal mode and repeat headers after 100 cells

Query results operations
[Print view](#) [Print view \(with full texts\)](#) [Export](#) [CREATE VIEW](#)

Asamblea








| Plantel | Fecha | Nombre | Minuta | O. Día |
|---------|------------|---|------------|---|
| 23 | 21/11/2008 | Asamblea Ordinaria | Pendiente | Lista de Asistencia. Acuerdos de Academias |
| 33 | 30-11-2008 | Tercera Asamblea Estatal | Pendiente | Basificación Nivel A |
| 25 | 23/09/2008 | Cuarta Asamblea Ordinaria | Pendiente | Lista Asistencia, Evaluación Final Semestre |
| 24 | 21-09-2008 | Revisión Plenario de Academia de Matemática | En proceso | Lista de Asistencia, Asignación de plazas |
| 54 | 03-09-2008 | Primera Reunión | Pendiente | Propuesta cambio de directivos |

Figura 4.4.3 Caso1. Consultar.

Caso 2:

Paso 1: Realizar una consulta desde el administrador (Phpmysadmin) de la tabla Eventos.

Paso 2: Entrar a la página principal del gestor.

Paso 3: Ingresar a tabla Eventos

Paso 4: Seleccionar Agregar. Ver Figura 4.4.4.



Eventos

Agregar Registro

| | |
|-----------------------------|----------------------------------|
| No. Plantel: | 51 |
| Nombre del Evento: | Avances Científicos en Robótica. |
| Fecha inicial (dd/mm/aaaa): | 21/10/2008 |
| Fecha final (dd/mm/aaaa): | 24/10/2008 |
| Sede del Evento: | Auditorio Principal |
| Hora (HH:MM): | 12:00 |
| Tipo de Evento: | Conferencia |

Inserción de nuevo registro.

Show: 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells
Sort by key: None

| | | | Ev_clve | PI_nmro | Ev_nomb | Ev_fchi | Ev_fchf | Ev_sede | Ev_hora | Ev_tipo |
|--------------------------|--|--|---------|---------|---|------------|------------|------------------------------|----------|-------------|
| <input type="checkbox"/> | | | 127 | 50 | Nuevas Tecnologías de Desarrollo de bases de | 23/10/2008 | 25/10/2008 | Auditorio Principal | 13:00:00 | exposicion |
| <input type="checkbox"/> | | | 107 | 32 | Matemáticas Finitas ¿Para qué? | 21/11/2008 | 21/11/2008 | Auditorio Principal Cetus 51 | 16:00:00 | conferencia |
| <input type="checkbox"/> | | | 106 | 21 | Tercer Congreso de Matemáticas Básicas | 21/09/2008 | 29/09/2008 | Auditorio Principal Cetus 34 | 16:30:00 | ciencia |
| <input type="checkbox"/> | | | 114 | 50 | 3er. Encuentro regional de Ciencias Humanisti | 22/09/2007 | 22/10/2008 | Explanada Principal | 10:00:00 | exposicion |
| <input type="checkbox"/> | | | 108 | 45 | Congreso magistral de matematicas e ingenieri | 10/09/2008 | 11/09/2008 | Auditorio Principal Cetus 15 | 12:00:00 | conferencia |
| <input type="checkbox"/> | | | 105 | 32 | Quinta feria de matemáticas | 21/04/2008 | 25/05/2009 | Auditorio Principal | 18:00:00 | conferencia |

Check All / Uncheck All With selected:

Show: 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

Figura 4.4.4. Caso 2. Insertar.

Caso 3:

Paso 1: Realizar una consulta desde el administrador (Phpmyadmin) de la tabla Curso

Paso 2: Entrar a la página principal del gestor.

Paso 3: Ingresar a tabla Curso.

Paso 4: Seleccionar Borrar.



Capítulo IV. Diseño, Programación e Implantación.



Paso 5: Marcar registro. Ver Figura 4.4.5.



Registro borrar en phpmyadmin.

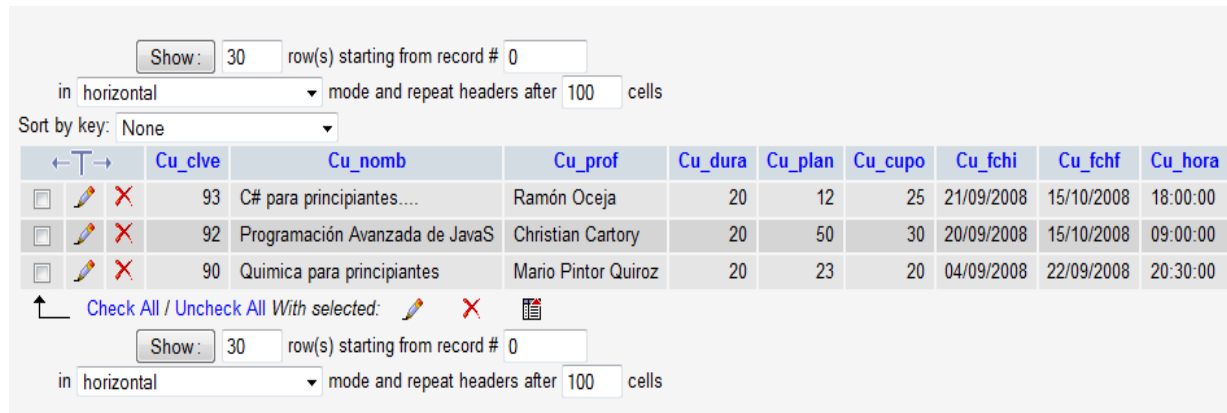


Figura 4.4.5. Caso 3. Eliminar.

Caso 4:

Paso 1: Realizar una consulta desde el administrador (Phpmyadmin) de la tabla Curso

Paso 2: Entrar a la página principal del gestor.

Paso 3: Ingresar a tabla Curso.

Paso 4: Seleccionar Actualizar. Ver Figura 4.4.6.



Curso

| sección: | Nombre | Profesor | Duración | Plantel | Cupo | Fecha Inicio | Fecha Fin | Hora |
|--------------------------|--------------------------|---------------------|----------|---------|------|--------------|------------|----------|
| <input type="checkbox"/> | C# para principiantes... | Ramón Oceja | 20 | 12 | 25 | 21/09/2008 | 15/10/2008 | 18:00:00 |
| <input type="checkbox"/> | Programación Avanzada | Christian Cartori | 20 | 50 | 30 | 20-09-2008 | 15-10-2008 | 09:00:00 |
| <input type="checkbox"/> | Química para principian | Mario Pintor Quiroz | 20 | 23 | 20 | 04/09/2008 | 22/09/2008 | 20:30:00 |

Registro modificado

Show: 30 row(s) starting from record # 0
 in horizontal mode and repeat headers after 100 cells
 Sort by key: None

| | Cu_clve | Cu_nomb | Cu_prof | Cu_dura | Cu_plan | Cu_cupo | Cu_fchi | Cu_fchf | Cu_hora |
|--------------------------|---------|--------------------------------|---------------------|---------|---------|---------|------------|------------|----------|
| <input type="checkbox"/> | 93 | C# para principiantes.... | Ramón Oceja | 20 | 12 | 25 | 21/09/2008 | 15/10/2008 | 18:00:00 |
| <input type="checkbox"/> | 92 | Programación Avanzada de JavaS | Christian Cartori | 20 | 50 | 30 | 20-09-2008 | 15-10-2008 | 09:00:00 |
| <input type="checkbox"/> | 90 | Química para principiantes | Mario Pintor Quiroz | 20 | 23 | 20 | 04/09/2008 | 22/09/2008 | 20:30:00 |

Check All / Uncheck All With selected:

Show: 30 row(s) starting from record # 0
 in horizontal mode and repeat headers after 100 cells

Figura 4.4.6. Caso 4. Modificar.

Resultados generales obtenidos:

Caso 1: Los datos reflejados fueron los mismos que contiene la tabla Asamblea.

Caso 2: Al insertar un nuevo registro, éste se guardó en la tabla Eventos.

Caso 3: Al seleccionar uno o varios registros y borrarlos, estos se eliminaron de la tabla Curso.

Caso 4: Al modificar algún registro, este se actualizó en la tabla.



Prueba de validación de Fecha, Hora, caracteres incorrectos, vacíos y no selección de registro a actualizar.

Caso 1:

Paso 1: Ingresar a tabla Eventos

Paso 2: Seleccionar Agregar.

Paso 3: Ingresar fecha inválida. Ver Figura 4.4.7

Resultado: Envió mensaje de fecha incorrecta y no insertó registro.

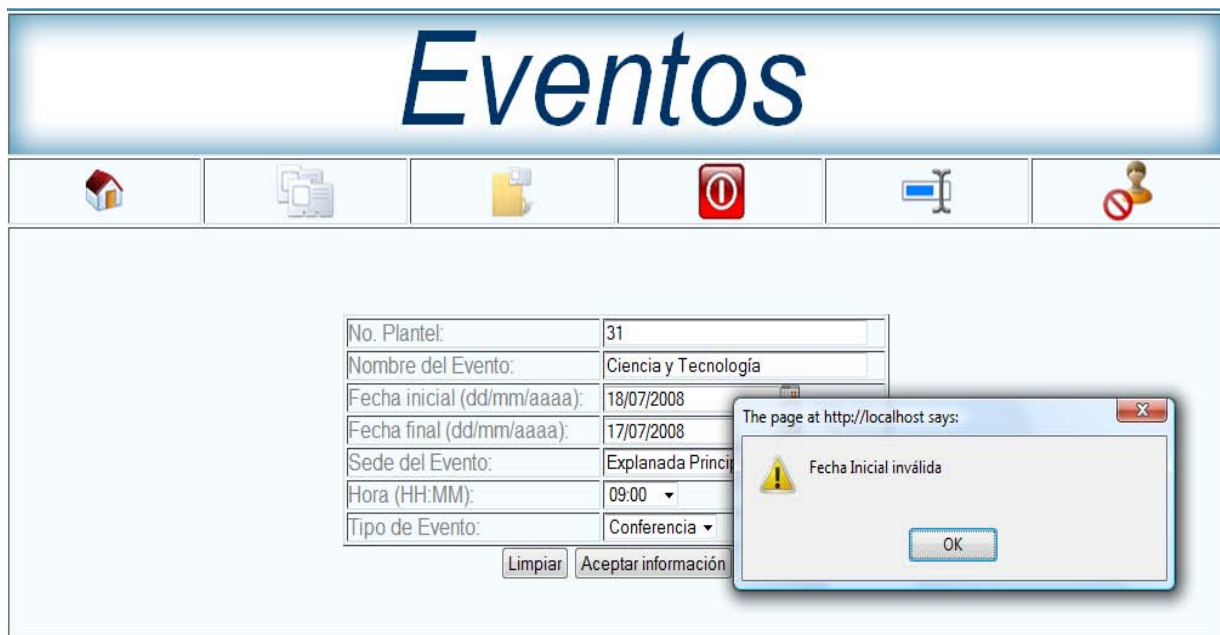


Figura 4.4.7. Validación de Fechas.

Caso 2:

Paso 1: Ingresar a tabla curso.

Paso 2: Seleccionar Actualizar.

Paso 3: Ingresar campo vacío. Ver Figura 4.4.8



Resultado: Envió de mensaje REGISTRO(S) VACIO(S) INTENTE NUEVAMENTE y no modifica el registro.

The screenshot shows a web application interface for course management. At the top, the word "Curso" is displayed in a large blue font. Below this is a navigation bar with several icons. The main content area contains a table with the following columns: Selección, Nombre, Profesor, Duración, Plantel, Cupo, Fecha Inicio, Fecha Fin, and Hora. The first row of the table is highlighted with a red box, and the "Nombre" cell is empty. Below the table, there is a "Limpiar" button. The bottom part of the screenshot shows the same "Curso" header and navigation bar, but the main content area is mostly blank, with a red-bordered box containing the text "REGISTRO(S) VACIO(S) INTENTE NUEVAMENTE".

| Selección | Nombre | Profesor | Duración | Plantel | Cupo | Fecha Inicio | Fecha Fin | Ho |
|-------------------------------------|-------------------------|---------------------|----------|---------|------|--------------|------------|----------|
| <input checked="" type="checkbox"/> | | Ramón Oceja | 20 | 12 | 25 | 28-09-2008 | 15/10/2008 | 18:00:00 |
| <input type="checkbox"/> | Desarrollo WEB | Christian Canton | 30 | 50 | 35 | 21-09-2008 | 15/10/2008 | 13:00:00 |
| <input type="checkbox"/> | Algebra Lineal | Maria Carlos | 30 | 53 | 30 | 02-10-2008 | 12/11/2008 | 17:00:00 |
| <input type="checkbox"/> | Quimica para principian | Mario Pintor Quiroz | 20 | 23 | 20 | 07-09-2008 | 22/09/2008 | 20:30:00 |
| <input type="checkbox"/> | Desarrollo Personal | Armando Hernandez | 20 | 30 | 25 | 23-09-2008 | 20-10-2008 | 12:00:00 |
| <input type="checkbox"/> | Técnicas de Redacción | Armando Quintero | 20 | 30 | 25 | 01-09-2008 | 03/10/2008 | 10:30:00 |
| <input type="checkbox"/> | Redacción Literaria | Manu Perez | 20 | 10 | 25 | 26/09/2008 | 06/11/2008 | 10:00:00 |
| <input type="checkbox"/> | Calculo Financiero | Mario Chua | 30 | 5 | 25 | 01/10/2008 | 03/11/2008 | 13:00:00 |
| <input type="checkbox"/> | Ortografía práctica | Orlando Funes | 20 | 1 | 30 | 25/09/2008 | 16/10/2008 | 11:30:00 |

Figura 4.4.8. Validación de datos vacios.



Capítulo IV. Diseño, Programación e Implantación.



Caso 3:

Paso 1: Ingresar a tabla curso.

Paso 2: Seleccionar Actualizar.

Paso 3: Seleccionar opción actualizar sin elección de registro a actualizar. Ver Figura 4.4.9.

The screenshot shows a web application interface for course management. At the top, the word "Curso" is displayed in a large blue font. Below it is a navigation bar with icons for home, file management, a red stop sign, a vertical ruler, and a person with a red prohibition sign. The main area contains a table with the following columns: "seleccionar", "Nombre", "Profesor", "Duración", "Plazas", "Cupo", "Fecha Inicio", "Fecha Fin", and "Hora". The table lists ten courses, with the first row highlighted in red. At the bottom right of the table are "Limpiar" and "Actualizar" buttons. Below the table is another navigation bar with the same icons. At the bottom of the page, a red-bordered box contains the text "Debe seleccionar el registro a actualizar".

| seleccionar | Nombre | Profesor | Duración | Plazas | Cupo | Fecha Inicio | Fecha Fin | Hora |
|--------------------------|---------------------------|---------------------|----------|--------|------|--------------|------------|----------|
| <input type="checkbox"/> | Ofi para principiantes... | Ramón Ojea | 20 | 12 | 25 | 29-09-2008 | 15/10/2008 | 18:00:00 |
| <input type="checkbox"/> | Desarollo WEB | Christian Carton | 30 | 50 | 35 | 21-09-2008 | 15/10/2008 | 13:00:00 |
| <input type="checkbox"/> | Algebra Lineal | Maria Carlos | 30 | 53 | 30 | 02-10-2008 | 12/11/2008 | 17:00:00 |
| <input type="checkbox"/> | Quimica para principian | Mario Pintor Quiroz | 20 | 23 | 20 | 07-09-2008 | 22/09/2008 | 20:30:00 |
| <input type="checkbox"/> | Desarollo Personal | Armando Hernandez | 20 | 30 | 25 | 23-09-2008 | 20-10-2008 | 12:00:00 |
| <input type="checkbox"/> | Técnicas de Redacción | Armando Quintero | 20 | 30 | 25 | 01-09-2008 | 03/10/2008 | 10:30:00 |
| <input type="checkbox"/> | Redacción Literaria | Maru Perez | 20 | 10 | 25 | 26/09/2008 | 06/11/2008 | 10:00:00 |
| <input type="checkbox"/> | Calculo Financiero | Mario Chua | 30 | 5 | 25 | 01/10/2008 | 03/11/2008 | 13:00:00 |
| <input type="checkbox"/> | Onografia práctica | Oriando Funes | 20 | 1 | 30 | 25/09/2008 | 16/10/2008 | 11:30:00 |

Debe seleccionar el registro a actualizar

Figura 4.4.9. Validación no selección de registro a actualizar.



Caso 4:

Paso 1: Ingresar a tabla Eventos.

Paso 2: Seleccionar Agregar.

Paso 3: Ingresar campo incorrecto. Ver Figura 4.4.10.

Resultado: Envió de mensaje dependiendo del campo que se validó.

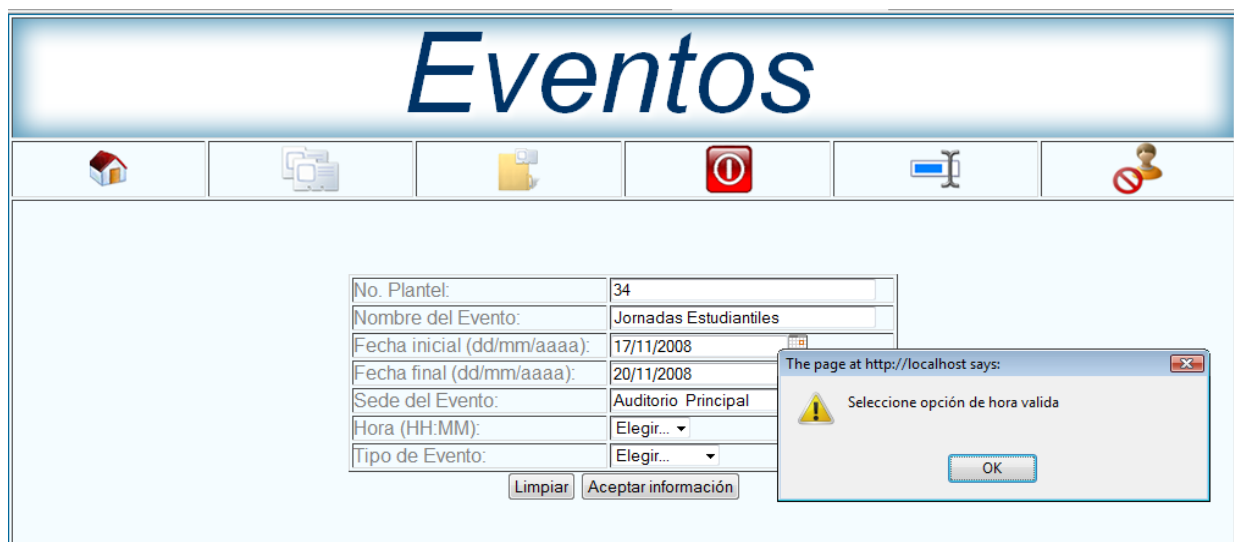
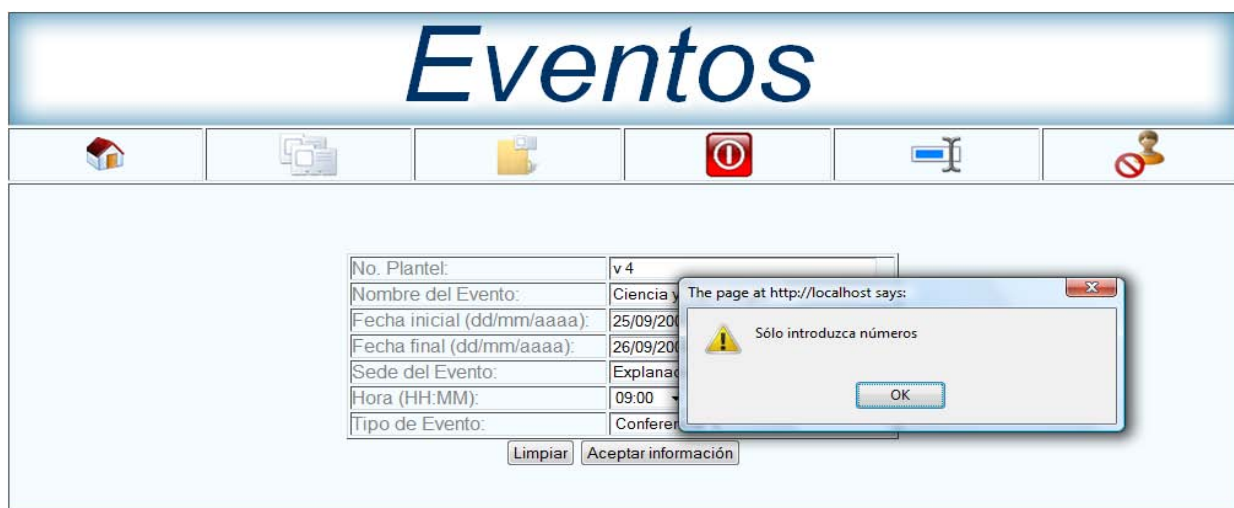


Figura 4.4.10. Validación de Caracteres Numéricos y Hora.



4.5 Implantación y Liberación.

Con la terminación de las dos primeras etapas del estudio de factibilidad, o sea el análisis de sistemas y el diseño del sistema, se está listo para llevar a cabo la última etapa principal del estudio de factibilidad, la cual consiste en la selección del equipo de procesamiento de datos, considerado el adecuado. La aplicación de un nuevo equipo de procesamiento de datos se establece completamente hasta que se realizan las órdenes con los fabricantes respectivos.

Criterios para la Selección del Equipo.

Consiste en presentar a cada uno de los proveedores fabricantes diagramas de flujo y tablas de decisión en donde se denoten las especificaciones del nuevo sistema. Esta información incluye la información general acerca de la compañía, sus planes futuros de procesamiento, y una lista de las especificaciones del nuevo sistema.

Criterio Alternativo.

Este criterio pasa por alto los datos compilados por el estudio de factibilidad a la fecha y le solicita a los proveedores fabricantes del equipo que hagan una investigación propia, por lo que los proveedores fabricantes llevan a la compañía su propio personal de sistemas para el estudio del sistema y diseñan un nuevo sistema ajustado a sus intereses.

El problema del tiempo es otro factor importante ya que cada proveedor fabricante debe de conducir una larga revisión de sistemas.



Implantación.

La preparación para la implantación debe empezar antes de que el nuevo sistema sea instalado y la aplicación anterior sea dada de baja.

Capacitación

Las actividades típicas de la capacitación empiezan antes de que el sistema sea instalado y continuará como sea necesaria para los nuevos usuarios y por cambios y actualizaciones del sistema. Para la capacitación se debe de tomar en cuenta la selección de participantes, currículum, métodos y materiales, personal de capacitación y calendario.

En la selección de participantes, sólo algunos usuarios deben ser capacitados. Una buena estrategia de capacitación es motivar a los de rápido aprendizaje a que trabajen en conjunto con otras personas, para que puedan ir asimilando mejor lo aprendido. Se deben de considerar diferentes niveles y enfoques de capacitación para personal que trabaja con diferentes aspectos del sistema.

Algunas opciones de la metodología y materiales para la capacitación son las siguientes:

- Sistemas de capacitación basados en multimedia interactivo.
- Programas tutoriales interactivos en línea.
- Manuales.



Capítulo IV. Diseño, Programación e Implantación.

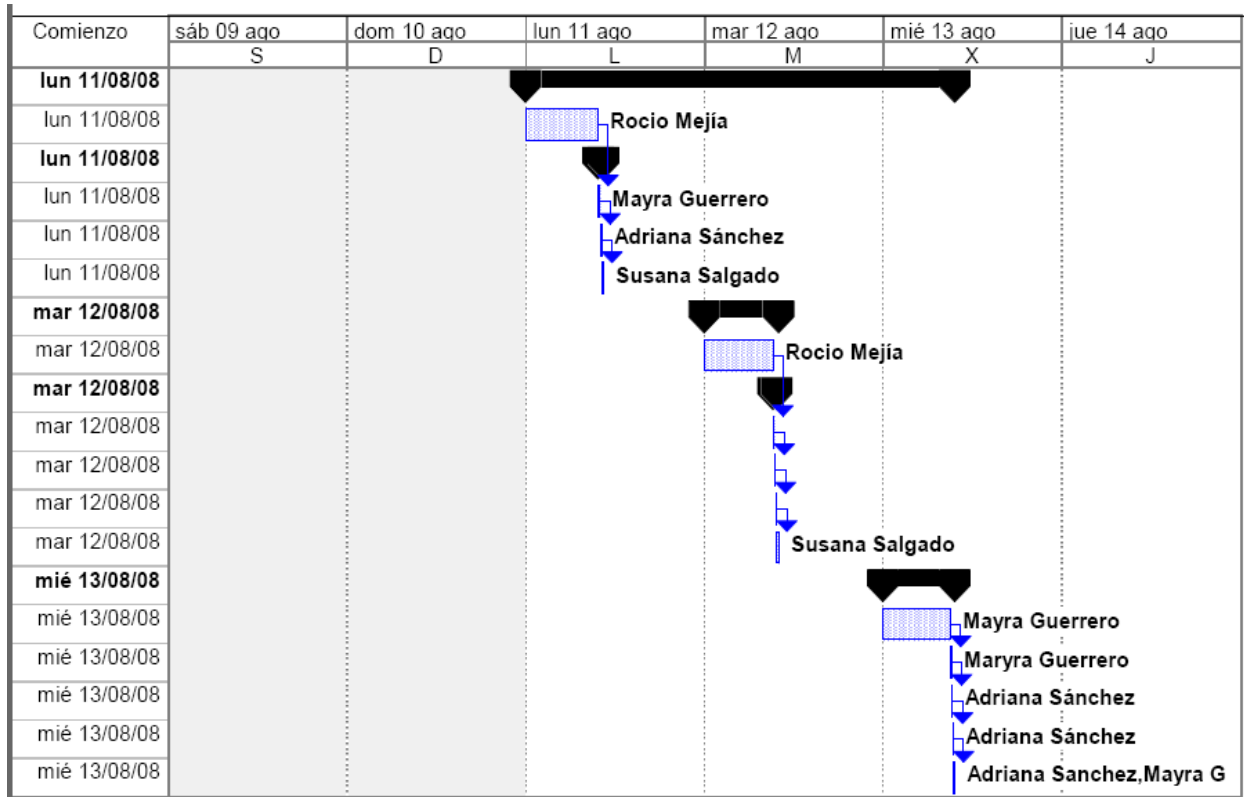


Plan de capacitación.

| Id | | Nombre de tarea | Duración | Comienzo |
|----|--|--|-----------------|---------------------|
| 1 | | Capacitación para uso de gestor | 998 mins | lun 11/08/08 |
| 2 | | Presentación de los expositores | 45 mins | lun 11/08/08 |
| 3 | | Introducción | 45 mins | lun 11/08/08 |
| 4 | | Software Libre (XAMMP) | 15 mins | lun 11/08/08 |
| 5 | | Base de Datos Gratuita (MYSQL) | 15 mins | lun 11/08/08 |
| 6 | | Resolucion de dudas | 15 mins | lun 11/08/08 |
| 7 | | Manejo de XAMPP | 60 mins | mar 12/08/08 |
| 8 | | Herramientas básicas | 15 mins | mar 12/08/08 |
| 9 | | Características de Phpmyadmin | 30 mins | mar 12/08/08 |
| 10 | | Creacion de una base de datos | 10 mins | mar 12/08/08 |
| 11 | | Respaldar la base de datos | 10 mins | mar 12/08/08 |
| 12 | | Recuperación de la base de datos | 10 mins | mar 12/08/08 |
| 13 | | Resolucion de dudas | 15 mins | mar 12/08/08 |
| 14 | | Manejo del gestor | 38 mins | mié 13/08/08 |
| 15 | | Componentes del gestor | 7 mins | mié 13/08/08 |
| 16 | | Dar de alta registros en la tabla | 7 mins | mié 13/08/08 |
| 17 | | Borrar registros de las tablas | 7 mins | mié 13/08/08 |
| 18 | | Actualizar registros | 7 mins | mié 13/08/08 |
| 19 | | Resolucion de dudas | 10 mins | mié 13/08/08 |



Diagrama de Gant.



Liberación.

La liberación es la culminación de cualquier sistema, por que se determina el cumplimiento de un contrato legal entre la empresa y el usuario quedando satisfechas ambas partes.



Capítulo IV. Diseño, Programación e Implantación.



A continuación se muestran dos ejemplos de tipos de contratos uno con el prestador de servicios de publicación de páginas en Internet y otro con el prestador de servicios del desarrollo del Portal.

Contrato de Sitios Web.

Contrato de prestación de servicios corporativo de publicación de páginas y servicios de mensajería en la red Internet que celebra por una parte _____ _representada legalmente en este acto por e _l_____ , en lo sucesivo denominada "El Proveedor" y por la otra representada legalmente por nombre del cliente, en lo sucesivo denominada "Cliente" al tenor de las siguientes declaraciones y cláusulas:

Declaraciones:

- Declara " El Proveedor"
- A. Que dentro de su objeto social se encuentra la publicación de Directorios de Talentos, el diseño, alojamiento, la producción y la promoción del sitio web www.portaldelaacademiaregionaldeinformatica.gob.mx y la prestación de servicios de valor agregado a usuarios de INTERNET, de diseño de páginas para World Wide Web y el desarrollo y operación de proyectos de comunicación social en medios electrónicos .
- B. Que cuenta con la infraestructura de interconexión con la red Internet a través de un sistema autorizado por Internet, de alta confiabilidad y con enlace telefónico a todo el mundo por la red de Teléfonos de México que proporciona el ancho de banda para un servicio de alta calidad.
- C. Que este sistema sirve de enlace entre el cliente y otros sistemas electrónicos de información, nacionales e internacionales.



Portal de la Academia Regional de Informática de la DGETI.



Declara el Cliente:

- A. Que conoce la forma de operar de la red de Internet y que es su intención contratar el servicio corporativo de publicación de páginas y servicios de mensajería a dicho sistema con " El Proveedor".
- B. Que cuenta con el equipo necesario y las cuentas de acceso para acceder al sistema.

Ambas partes declaran:

- A. Que reconocen su personalidad jurídica amplia y suficiente para obligarse y contratar en los términos aquí señalados.
- Expuesto lo anterior, las partes están de acuerdo en otorgar las siguientes:

Cláusulas:

Premisas

- A. El Cliente desea publicar un Web-site a través de " El Proveedor", con lo cual podrá contar con páginas Web en su propio servidor virtual.
- B." El Proveedor" le ofrece para ello la conexión de su servidor a Internet y la tecnología de punta, tanto en diseño, como en programación.

Políticas de pago

- A. El Cliente se obliga a pagar por la colocación de páginas y otros servicios en los servidores de " El Proveedor", según la "Tabla de servicios, tarifas y formas de pago vigente, la cual estará permanentemente a disposición del usuario dentro del sistema.
- B. El Cliente tendrá dos opciones de realizar su pago:



Capítulo IV. Diseño, Programación e Implantación.



- Si desea pagar mensualmente el pago será en efectivo, mediante depósito en cuenta de cheques.
- Si desea pagar trimestral, semestral o anualmente podrá ser con cheque y los procedimientos de pago serán de igual forma mediante depósito bancario.
- C. La cuota anual por la prestación de servicios en ningún caso será reembolsable. Las cancelaciones de servicios que sean de contratación anual deberán hacerse por lo menos 3 meses antes del inicio del siguiente período anual.
- D. Cuando el cliente cubra el importe de la cuota con cheque y éste sea devuelto por la Institución bancaria, " El Proveedor" se reserva el derecho de cobrar 20% sobre el importe del documento devuelto en los términos del artículo 193 de la Ley General de Títulos y Operaciones de Crédito.

Costos

- Los precios por los servicios de publicación a INTERNET a través de la colocación de páginas en un servidor de " El Proveedor", son las siguientes:
- A. La renta mensual será de \$00 Dólares más I.V.A. al tipo de cambio de compra de dólar en ventanilla
- B. Si el Cliente paga anualmente, " El Proveedor" le otorgará un 15% de descuento; un 7% si el pago es semestral y un 4% si es trimestral.
- C. Con un costo de \$00 Dólares al tipo de cambio de ventanilla el día de la contratación, El cliente tendrá derecho a contar con los registros de dominio que requiera dentro del mismo servidor sin cargos adicionales.
- Esto incluye los gastos de registro ante la entidad responsable del control de Nombres de Dominio en INTERNET y la asignación de una dirección IP exclusiva.
- El registro en los 600 motores de búsqueda o "portales" así como la indización del sitio en los mismos tendrá un costo de \$00 pesos mensuales o \$00.00 pesos por pago anual.



Responsabilidad del Cliente.

- A. El Cliente se compromete a hacer un buen uso del sistema absteniéndose de interferir en otras redes con usuarios, enviar o propagar virus informáticos o realizar cualquier actividad que rompa con el código de ética de las Sociedades de Redes Internacionales.
- B. El Cliente se obliga a utilizar el servicio de enlace a la red únicamente para actividades lícitas. Cualquier documento o material transmitido en contra de las leyes y/o acuerdos nacionales y/o internacionales está prohibido. Esto incluye de manera enunciativa y no limitada, material protegido por los Derechos de Autor, material que vaya en contra de la moral y las buenas costumbres, material que sea contrario a la concordia internacional, a la paz, al orden público o material que atente contra la vida privada de alguien.
- C. El Cliente acepta liberar a " El Proveedor" de toda responsabilidad frente a terceros, en caso de que el usuario incumpla consciente o inconscientemente la cláusula anterior de este contrato.
- D. El Cliente acepta que tiene prohibido ofrecer acceso comercial directo o indirecto, a la red Internet sin autorización previa y por escrito de " El Proveedor".
- E. El uso de la información obtenida es a riesgo del Cliente, ya que " El Proveedor" y su personal no son responsables de la calidad, legalidad, veracidad, exactitud o utilidad de la información que el Cliente encuentre en el sistema o en otros sistemas consultados por medio de " El Proveedor".
- F. El Cliente se obliga a no transferir, reproducir, publicar, explotar o distribuir total o parcialmente la información obtenida a través del sistema con fines de lucro, sin el consentimiento del autor, o titular del derecho patrimonial, toda vez que el contenido e información que ofrece el sistema es una obra intelectual que confiere derechos de autor y la cual se encuentra protegida por la ley correspondiente.

Responsabilidades de " El Proveedor".

- A." El Proveedor" se compromete a ofrecer un espacio suficiente en el servidor para la colocación de las páginas web del Cliente.



Capítulo IV. Diseño, Programación e Implantación.



- B. " El Proveedor" manifiesta que cuenta con los conocimientos, el equipo y la tecnología para la buena funcionalidad del servicio motivo del presente contrato.
- C. " El Proveedor" designará contactos técnicos en caso de que el Cliente tenga alguna duda o emergencia en la cual sea posible apoyarlo.
- D. " El Proveedor" no será responsable de las interrupciones y fallas en el servicio provenientes de caso fortuito o fuerza mayor, así como de ninguna falla que pudiera originarse con las conexiones del ISP o proveedor del servicio de acceso, por la cual se le impidiera el acceso al sistema.

Duración

- A. El presente contrato tendrá una duración indefinida.

Terminación

- A. El presente contrato podrá ser dado por terminado por cualquiera de las partes con un aviso por escrito con tres días de anticipación al vencimiento del último pago (con excepción de la Cláusula 2c).
- La Terminación anticipada será sin perjuicio de las obligaciones que hubiera contraído el Cliente antes de la fecha efectiva de terminación.
- A. El presente contrato es intransferible, por lo que no podrá ser cedido o transferido por ningún medio o título legal, a persona alguna ya sea física o moral.

Notificaciones.

- Para todos los efectos a que hubiese lugar, las partes señalan como sus domicilios los siguientes:
- " El Proveedor" Desarrolladores SA de CV
- El Cliente:
- Mientras las partes no notifiquen por escrito el cambio de sus respectivos domicilios, los avisos, notificaciones y demás diligencias jurídicas o extrajudiciales serán practicadas en los domicilios antes señalados surtiendo plenamente sus efectos.



**Portal de la Academia Regional de Informática de la
DGETI.**



El presente contrato se firma por las partes interesadas en la Ciudad de México a los días del mes de _____ de _____.

Por " El Proveedor"

Por el Cliente

Desarrolladores SA de CV

Contrato de Desarrollo de Programas Informáticos.

Conste por el presente documento el Contrato para el Desarrollo de Programas Informáticos que celebran de una parte....., debidamente representado por su Director de Administración....., identificado con DNI..... y su Director de Abastecimiento....., identificado con Libreta Electoral N° con domicilio en....., a quienes en adelante se les denominar la ENTIDAD, y de la otra parte la Empresa inscrita con N° del Registro Mercantil de, debidamente representado por su, Sr....., identificado con DNI, con domicilio en el N°....., a quien en adelante se le denominar el PROVEEDOR bajo los términos y condiciones siguientes:

CLAUSULA PRIMERA.- ANTECEDENTES

CLAUSULA SEGUNDA.- OBJETO

CLAUSULA TERCERA.- COMUNICACION ENTRE LAS PARTES

CLAUSULA CUARTA.- PRECIO CONVENIDO Y FORMA DE PAGO

CLAUSULA QUINTA.- PROPIEDAD DE LOS PROGRAMAS INFORMATICOS

Los Programas Informáticos son propiedad del PROVEEDOR hasta su cancelación total por parte de la ENTIDAD.



Capítulo IV. Diseño, Programación e Implantación.



CLAUSULA SEXTA.- DE LAS OBLIGACIONES
CLAUSULA SEPTIMA.- DE LAS MODIFICACIONES
CLAUSULA OCTAVA.- DE LA ENTREGA
CLAUSULA NOVENA.- PERSONAL DE SERVICIO DEL PROVEEDOR
CLAUSULA DECIMA.- RESPONSABILIDAD DE LAS PARTES
CLAUSULA DECIMA PRIMERA.- ANEXOS
CLAUSULA DECIMA SEGUNDA.-PRUEBA DE FUNCIONAMIENTO
CLAUSULA DECIMA TERCERA.- DE LA RECEPCION
CLAUSULA DECIMA CUARTA.- GARANTIA DE EVICCION
CLAUSULA DECIMA QUINTA.- CAPACITACION
CLAUSULA DECIMA SEXTA.- GASTOS ADICIONALES
CLAUSULA DECIMA OCTAVA.- CASO FORTUITO O FUERZA MAYOR
CLAUSULA DECIMA NOVENA.- FIANZA DEL FIEL CUMPLIMIENTO
CLAUSULA VIGESIMA.- GARANTIA TECNICA
CLAUSULA VIGESIMA PRIMERA.- VENTA
CLAUSULA VIGESIMA SEGUNDA.- RESOLUCION DEL CONTRATO
CLAUSULA VIGESIMA TERCERA.- PENALIDADES
CLAUSULA VIGESIMA CUARTA.- MODIFICACION
CLAUSULA VIGESIMA QUINTA.- DE LA VIGENCIA
CLAUSULA VIGESIMA SEXTA.- DEL TÉRMINO
CLAUSULA VIGESIMA SEPTIMA.- ARBITRAJE
CLAUSULA VIGESIMA OCTAVA.- COMPETENCIA

ENTIDAD
(sello y firma)

PROVEEDOR
(sello y firma)



4.6 Tipos de Mantenimiento.

Mantenimiento Preventivo. Este mantenimiento tiene lugar antes de que ocurra una falla o avería, se efectúa bajo condiciones controladas sin la existencia de algún error en el sistema. Se realiza a razón de la experiencia y pericia del personal a cargo, los cuales son los encargados de determinar el momento necesario para llevar a cabo dicho procedimiento; se puede también estipular el momento adecuado a través de los manuales técnicos.

- Se realiza en un momento en que no se interfiere en las labores cotidianas.
- Se lleva a cabo mediante un programa previamente elaborado donde se detalla el procedimiento a seguir, y las actividades a realizar, a fin de tener las herramientas necesarias "a la mano".

Mantenimiento Correctivo. Este mantenimiento tiene lugar luego que ocurre una falla o avería, es decir, solo actuará cuando se presenta un error en el sistema. En este caso si no se produce ninguna falla, el mantenimiento será nulo, por lo que se tendrá que esperar hasta que se presente el desperfecto para tomar medidas de corrección de errores.

Mantenimiento Proactivo. Este mantenimiento tiene como fundamento el trabajo en equipo, de tal manera que todos los involucrados directa o indirectamente en la gestión del mantenimiento deben conocer las actividades que se llevan a cabo para desarrollar las labores de mantenimiento. Este mantenimiento a su vez debe brindar indicadores (informes) hacia la gerencia, respecto del progreso de las actividades, los logros, aciertos, y también errores.



Conclusiones.



CONCLUSIONES



**Portal de la Academia Regional de Informática de la
DGETI.**





Conclusiones.

Para realizar la implementación de cualquier servicio en un sistema de cómputo se requiere la existencia de un equilibrio entre conocimiento y experiencia en el desarrollo tecnológico y el impacto que ocasione este ante la sociedad. Las opciones de automatización en distintas áreas son requisito indispensable para estar a la vanguardia empresarial.

La búsqueda de minimizar costos entre las empresas ha llevado al desarrollo de alternativas como son la implementación de software libre de código abierto (“Open Source Software”); los cuales permiten adaptarse a los cambios e innovar los sistemas, permitiendo tener la ventaja de que este pueda ser adaptado a las necesidades que cada organización requiere.

La relación de las bases de datos con la web va en aumento, esta interrelación se lleva a cabo por medio de herramientas que se adaptan entre sí por medio de interfaces.

La evolución de la web, ha obligado al empleo de tecnologías cliente/servidor de tres capas (cliente / servidor de aplicaciones /servidor de datos). Permitiendo que la gestión de datos abarque una gran variedad de tipos de información que pueden estar disponibles a cualquier hora, ventaja que se distingue principalmente por el objetivo planteado al implantar este portal.

Los accesos a la web muestran contenido almacenado en la base de datos. La recopilación de datos puede estar automatizada y el manejo de datos masivo se realiza por medio de un sistema gestor de bases de datos que pueden ser empleados en una computadora normal.



El mantenimiento, integridad y seguridad de la información ha avanzado, previendo la actuación de ciertos actos para que la información que se recopile sea verídica.

Una de las opciones más económicas y dependiendo del volumen de información a manejar es la elección de una base de datos de código abierto siendo una alternativa válida que gerentes y desarrolladores deberían considerar al momento de escoger una solución informática.

El desarrollo de este tipo de bases y la forma en la que miles de usuarios aportan a la solución pronta de problemas existentes en el motor de base de datos, las hacen ver como serias competidoras ante las bases de datos propietarias, poco a poco sus funcionalidades rebasan estas últimas.

Aunque Tomcat no es directamente compatible con todas las API y objetos de J2EE (como JDBC, JNDI, JavaMail, RMI, JMS, XML), todas esas API, con la notable excepción de EJB, están disponibles agregando archivos Java (JAR) disponibles gratuitamente.

En caso de no contar con un departamento de servicio técnico, se pueden plantear sus dudas y problemas en multitud de foros y de grupos de noticias. Generalmente, las preguntas que se envían a un foro de código libre se responden más pronto que las que se envían al servicio técnico de un producto comercial.

Hay que tomar en cuenta que la compra del motor de base de datos no es el único gasto que las empresas tendrán que realizar para automatizar sus procesos, existen otros costos como administración, mantenimiento, capacitación, el costo es entonces



Conclusiones.

un punto muy importante para tomar en cuenta a la hora de seleccionar un sistema gestor de bases de datos.

Javascript es un lenguaje que ofrece un sin numero de ventajas a todo. Además se complementa muy bien con php y html, permitiendo pasar datos de un lenguaje a otro con bastante facilidad. Con esto se podrá tener una página más robusta, de navegación cómoda y con algunos efectos visuales interesantes.

Las bases de datos de código abierto están ganando terreno en el mercado rápidamente, esto ha preocupado mucho a las firmas propietarias de software como Oracle a tal punto que esta empresa ya lanzó su versión libre de la base de datos.

Después de realizar un estudio sobre las distintas soluciones existentes y buscando aquella que ofreciera ventajas desde el punto de vista técnico y económico, se llegó a la conclusión de utilizar herramientas de código abierto.

Después del estudio a lo largo de este trabajo, crear una página para la Academia Regional de Informática no ha sido fácil, sobre todo al evaluar la mejor manera de simplificar y automatizar la información a la academia y por ende tener la información actualizada y de fácil acceso. El proyecto está orientado a obtener un producto útil con contenido didáctico.

De manera personal se han adquirido conocimientos y agilidad en el desarrollo de aplicaciones en PHP. También se ha desarrollado habilidades en la administración de bases de datos MySQL, incluyendo el desarrollo de scripts que permitan la automatización de algunas tareas, como la instalación y configuración del entorno de desarrollo; permitiendo poner en prueba las aptitudes y la capacidad de razonar en la resolución de problemas y plantear alternativas viables.



Portal de la Academia Regional de Informática de la DGETI.



Evaluando el objetivo que persigue este trabajo y de acuerdo con lo desarrollado a lo largo del mismo, consideramos que el objetivo se cumplió de manera satisfactoria.

Hay que mencionar que es la primera página con la que cuenta la Academia Regional de Informática, por lo tanto consideramos que es una primera versión y está sujeta a mejoras de acuerdo a las necesidades que vayan surgiendo con el transcurso del tiempo de la Institución.

Finalmente se proponen una serie de mejoras buscando la portabilidad total de la aplicación. Esta mejora consiste en ampliar el rango de servicios ofrecidos desde el portal, o bien mejorar el funcionamiento de algunos de los ya existentes.



BIBLIOGRAFÍA



**Portal de la Academia Regional de Informática de la
DGETI.**





BIBLIOGRAFÍA.

- Castaño Adoración de Miguel, *Fundamentos y Modelos de Bases de Datos*, 2ª. Ed. México, Editorial: Alfa Omega 1999. 544 pp.
- Díaz María Paloma, et al. *Ingeniería de la WEB y patrones de Diseño*. Editorial: Pearson Educación Prentice Hall. Madrid 2005. 500 pp.
- González Barahona: Jesús. Seoane Pascual Joaquín. *Introducción al Software Libre*. 1ª. Ed. Madrid España. Editorial: UOC Formación de posgrado 2003.
- González López Óscar Rodrigo, *Comercio Electrónico*. 1ª. Ed. España Editorial: Pearson Educación 2002. 300 pp
- Gutiérrez Rodríguez Abraham, *PHP 5 a través de Ejemplos*. 1ª. Ed. México Editorial: Alfa Omega 2005. 576 pp.
- Hanke Johann Christian. *PC Cuaderno Los Secretos de PHP y MySQL*. 1ª. Ed. Francia. Editorial: Pearson Educación de México 2006. 95 pp.
- Hanke Johann Christian. *PC Cuaderno PHP5*. 1ª. Ed. Francia. Editorial: Pearson Educación de México 2005. 95 pp.
- Hernández Orallo José. (jorallo@dsic.upv.es), *La Disciplina de los Sistemas de Bases de Datos. Historia, Situación Actual y Perspectivas*. Departamento de Sistemas Informáticos y Computación de la Universidad Politécnica de Valencia. Mayo 2002.
- Kendal Kenneth E. *Análisis y Diseño de Sistemas*. 1ª. Ed. México. Editorial: Prentice Hall 2005.
- Kew Nick, *Desarrollo de Módulos y Aplicaciones con Apache*. 1ª. Ed. España Editorial: Anaya Multimedia 2008. 528 pp.
- Mannino Michael V. *Administración de Bases de Datos. Diseño y Desarrollo de Aplicaciones*. 3ª. Ed. México, Editorial: MacGraw Hill Interamericana 2007.



- Mateu Carles. *Software Libre. Desarrollo de Aplicaciones Web* 1ª. Ed. Barcelona España Editorial: Fundació per a la Universitat Oberta de Catalunya. Marzo 2004 378pp.
- Oz Effy, *Administración de Sistemas de Información*. 2ª. Ed. México. Editorial: Thomson Corporation 2001.
- Pavona Puertas Jacobo *Creación de un Portal con PHP y MySQL*. 3ª. Ed. México. Editorial: Alfa Omega 2007 256 pp.
- Pérez López César, *Dreamweaver 8: Desarrollo de Páginas Web Con PHP y MySQL*. 1ª. Ed. México. Editorial: Alfa Omega 2007 460 pp.
- Piattini Velthuis Mario Gerardo, *Tecnología y Diseño de Bases de Datos*, 1ª. Ed. México, Editorial: Alfa Omega 2007. 980 pp
- Post Gerald V., *Sistemas de Administración de Bases de Datos*, 1ª. Ed. México. Editorial: MacGraw Hill Interamericana 2006.
- Pressman Alan H., *Ingeniería de Software*. 5ª. Ed. México. Editorial: MacGraw Hill Interamericana 2000.
- Quigley Ellie, *PHP y MySQL Practico: Para Programadores y Diseñadores WEB*. 1ª. Ed. España. Editorial: Anaya Multimedia 2007 452 pp
- Schach : Stephen R. *Ingeniería de Software Clásica y orientada a Objetos*. 6ª. Ed. México. Editorial: MacGraw Hill Interamericana 2006.
- Silberschatz Abraham. *Fundamentos de Bases de Datos*. 5ª. Ed. España. Editorial: MacGraw Hill Interamericana 2006. 944 pp
- Senn James A., *Análisis y Diseño de Sistemas de Información*, 2ª. Ed. México. Editorial: MacGraw Hill Interamericana 1992.
- Welling Luke, Thompson Laura. “*PHP and MySQL Web Development*.” 2a. Ed. Indianapolis, Indiana Editorial: Sams Publishing 2003.



CIBEROGRAFÍA.

<http://www.dgeti.sep.gob.mx/> 13/03/2008

<http://www.apachefriends.org/en/xampp.html> 18/03/2008

<http://es.wikipedia.org/wiki/XAMPP> 19/03/2008

http://es.wikipedia.org/wiki/Bases_de_datos_relacionales 20/03/2008

<http://web.firebird.com.mx/modules/mylinks/singlelink.php?cid=6&lid=6> 18/03/2008

<http://web.firebird.com.mx/modules/news/> 18/03/2008

<http://www.mastermagazine.info/termino/3866.php> 21/03/2008

<http://es.wikipedia.org/wiki/.php> 17/03/2008

<http://tomcat.apache.org/> 25/03/2008

<http://www.help400.es/asp/scripts/nwart.asp?Num=131&Pag=10&Tip=T> 24/03/2008

<http://www.forosdelweb.com/f18/desventajas-php-153799/> 25/03/2008

<http://www.programacion.net/bbdd/tutorial/warehouse/2/17/04/2008>

http://www.ciao.es/Opiniones/JavaScript_2_0__346524 22/04/2008

<http://www.trucoswindows.net/foro/topico-75231-tema-1---introduccion-a-php.html>
17/03/2008.

<http://www.maestrosdelweb.com/editorial/phpmysqlap/#a4> 18/04/2008

<http://technet.microsoft.com/es-es/library/ms152563.aspx> 22/04/2008

<http://www.adrformacion.com/cursos/aspnet35av/leccion1/tutorial3.html> 22/04/2008

http://www.aulaclie.es/sql/b_8_1_1.htm 17/04/2008

<http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/> 18/04/2008

<http://mitecnologico.com/Main/PruebasElImplantacionSistema> 22/05/2008



http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/viveros.sca/capitulo6.pdf
23/05/2008

http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/viveros.sca/capitulo5.pdf
24/05/2008

<http://citip.org.mx/servicios/software.aspx> 19/05/2008

http://www.rogeliodavila.com/tcs/TCS%20Notes%20JAVega/Parte_16_TestWhite.ppt
20/05/2008

<http://uxmcc1.iimas.unam.mx/~ramiro~/too/PlanPruebas.com> 26/05/2008

<http://www.modelosdecontratos.com/ficha.asp?id=93&pagina=1&busqueda=0>
28/05/2008

http://compusys.net.mx/contrato_servic.htm 29/05/2008

http://www.video.com.mx/internet/contrato_de_paginas_web.htm 29/05/2008

<http://www.ujat.mx/publicaciones/hitos/ediciones/15/capacitacion.pdf> 26/05/2008

<http://www.organiza.com.mx/importancia-del-entrenamiento-capacitacion.htm>
27/05/2008

<http://www.mitecnologico.com/Main/ConceptoImportanciaCapacitacion> 27/05/2008

http://es.wikipedia.org/wiki/Diagrama_de_Gantt 18/05/2008

<http://www.scribd.com/doc/2878605/Carta-Gantt-2008> 18/05/2008

<http://www.php-hispano.net/articulos/guia-de-seguridad-en-php-1.html>.20/09/2008

<http://phpes.com/blog/category/seguridad/> 22/09/2008

<http://mx.php.net/manual/es/features.safe-mode.php> 26/09/2008

http://www.niquelao.net/acc_calendar-calendario-accesible-en-javascript-innecesario-pero-util/01/10/2008

<http://www.librosweb.es/>05/10/2008



Bibliografía.

<http://www.anieto2k.com/2006/10/15/javascript-no-obstructivo-manual-de-buenas-maneras/> 07/10/2008

<http://alexsancho.name/archives/2005/02/formulario-de-contacto-con-php/> 08/10/2008

http://foro.elhacker.net/php/seguridad_en_php-t63190.0.html 09/10/2008

<http://www.scribd.com/doc/4106702/PHP-MySQL-for-Dummies> 09/10/2008

<http://www.codigojavascript.com/> 10/10/2008



**Portal de la Academia Regional de Informática de la
DGETI.**





GLOSARIO



**Portal de la Academia Regional de Informática de la
DGETI.**





Renderizar: La renderización es el proceso de generar una imagen desde un modelo. Los medios por los que se puede hacer un renderizado van desde lápiz, pluma, plumones o pastel, hasta medios digitales en dos y tres dimensiones. La palabra renderización proviene del inglés render, y no existe un verbo con el mismo significado en español, por lo que es frecuente usar las expresiones renderizar o renderear.

Instancia: En programación una instancia se produce con la creación de un objeto perteneciente a una clase (instanciar una clase), que hereda entonces sus atributos, propiedades y métodos para ser usados dentro de un programa, ya sea como contenedores de datos o como partes funcionales del programa al contener en su interior funcionalidades de tratamiento de datos y procesamiento de la información que ha sido programada con anterioridad en la clase a la que pertenece.

J2EE: Java Platform, Enterprise Edition o Java EE (anteriormente conocido como Java 2 Platform, Enterprise Edition o J2EE hasta la versión 1.4), es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

API's: (Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos).

JVM: Máquina virtual Java (Java Virtual Machine) es un programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java.



Portal de la Academia Regional de Informática de la DGETI.



JDK: (Java Development Kit), es un software que provee herramientas de desarrollo para la creación de programas en java. Puede instalarse en una computadora local o en una unidad de red. En la unidad de red se puede tener la aplicación distribuida en varias computadoras y trabajar como una sola aplicación.

Transacciones ACID: ACID es un acrónimo de Atomicity, Consistency, Isolation and Durability: Atomicidad, Consistencia, Aislamiento y Durabilidad en español. En bases de datos se denomina ACID a un conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción. Así pues, si un sistema de gestión de bases de datos es ACID compliant quiere decir que el mismo cuenta con las funcionalidades necesarias para que sus transacciones tengan las características ACID.

WSAD: Websphere Studio Application Developer.

PIPES: Son una cadena de procesos conectados de forma tal que la salida de cada elemento de la cadena es la entrada del próximo. Es común el uso de buffer de datos entre elementos consecutivos. Las pipes están implementadas en forma muy eficiente en los sistemas operativos multitarea, iniciando todos los procesos al mismo tiempo, y atendiendo automáticamente los requerimientos de lectura de datos para cada proceso cuando los datos son escritos por el proceso anterior.

Frames: Son marcos o cuadros en gráficos por computadora, es el contenido de una pantalla de datos o su espacio de almacenamiento equivalente.



Widgets: Es un componente gráfico, o control, con el cual el usuario interactúa, como por ejemplo, una ventana, una barra de tareas o una caja de texto.

CSS: Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

Bit Torrent: Es un protocolo diseñado para el intercambio de archivos entre iguales (peer to peer o P2P).

RSS: Es un sencillo formato de datos que es utilizado para transmitir contenidos a suscriptores de un sitio web. El formato permite distribuir contenido sin necesidad de un navegador.

Feed: Es un documento (a menudo basado en XML) el cual contiene referencias electrónicas a versiones extendidas, es decir a un enlace para ampliar información, descripción con un resumen del contenido entrada (a veces la entrada entera) y la fecha de publicación.

Phishing: Es un término informático que denomina un tipo de delito encuadrado dentro del ámbito de las estafas, y que se comete mediante el uso de un tipo de ingeniería social caracterizado por intentar adquirir información confidencial de forma fraudulenta (como puede ser una contraseña o información detallada sobre tarjetas de crédito u otra información bancaria). El estafador, conocido como phisher, se hace pasar por una persona o empresa de confianza en una aparente comunicación oficial electrónica, por lo común un correo electrónico, o algún sistema de mensajería instantánea¹ o incluso utilizando también llamadas telefónicas.



Portal de la Academia Regional de Informática de la DGETI.



XUL: (acrónimo de XML-based User-interface Language, lenguaje basado en XML para la interfaz de usuario) es la aplicación de XML a la descripción de la interfaz de usuario en el navegador Mozilla.

SVG: Scalable Vector Graphics (SVG) es un lenguaje para describir gráficos vectoriales bidimensionales, tanto estáticos como animados (estos últimos con ayuda de SMIL), en XML.

Adobe Pagemill: Fue uno de los primeros editores HTML para la World Wide Web.

Pop-Ups: Son ventanas que emergen automáticamente (generalmente sin que el usuario lo solicite) mientras se accede a ciertas páginas Web. A menudo, las ventanas emergentes se utilizan con el objeto de mostrar un aviso publicitario de manera intrusiva.

Tabs: Barra de herramientas de Mozilla.

Keywords: Palabras clave.

Texpath: Es una función propietaria de Mozilla para soportar SVG TEXT, es decir es una forma de convertir texto a gráficos.



MANUAL TÉCNICO



**Portal de la Academia Regional de Informática de la
DGETI.**





1. Introducción a Xampp.

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache e intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de **X** (para cualquiera de los diferentes sistemas operativos), **A**pache, **M**ySQL, **P**HP, **P**erl. El programa esta liberado bajo la licencia GNU, actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas.

XAMPP es un archivo con extensión zip, tar, o exe que se puede descargar y ejecutar, dotado de algunas configuraciones en alguno de sus componentes que el servidor web necesitará. XAMPP es regularmente actualizado para incorporar las últimas versiones u actualización de Apache/MySQL/PHP y Perl.

Oficialmente, los diseñadores de XAMPP solo pretendían su empleo como una herramienta de desarrollo, para permitir a los diseñadores de sitios webs y programadores probar su trabajo en sus propios ordenadores sin ningún acceso a Internet. En la práctica es utilizado actualmente para servidor de sitios webs en WWW, y con algunas modificaciones es generalmente lo suficientemente seguro para serlo.

Existen varias distribuciones, de las cuales se encuentra Windows 98, NT, 2000, 2003, XP y Vista, cuya versión contiene: Apache, MySQL, PHP + PEAR, Perl, mod_php, mod_perl, mod_ssl, OpenSSL, phpMyAdmin, Webalizer, Mercury Mail Transport System for Win32 and NetWare Systems v3.32, Ming, JpGraph, FileZilla FTP Server, mcrypt, eAccelerator, SQLite, and WEB-DAV + mod_auth_mysql.



2. Instalación de Xampp versión para Windows.

PhpMyAdmin es una utilidad que sirve para interactuar con una base de datos de forma sencilla y desde una interfaz web. Sirve para crear bases de datos, tablas, borrar o modificar datos, añadir registros, hacer copias de seguridad, etc. Es una aplicación útil que la mayoría de los hosting con MySQL disponen de ella. Al ser una aplicación escrita en PHP, necesita de Apache y MySQL para poder funcionar.

Pasos para obtener las versiones de la herramienta a instalar:

Buscar la dirección <http://www.apachefriends.org/en/xampp.html> para descargar la aplicación, se sugiere bajarlo en un directorio con un nombre que lo identifique. Una vez que se bajo el archivo identificado con el nombre de xampp- win32-1.6.61-installer este es un archivo ejecutable; al seleccionarlo dando doble clic, este desplegará el asistente de instalación que se muestra en la **Figura 2.1**



Figura 2.1. Proceso de Instalación de Xampp. Asistente de instalación.



Seleccionar el botón Next y este mostrará la ruta por default de instalación, se recomienda dejarlo en esta carpeta si se desea instalar en un lugar en específico se debe indicar y seleccionar el botón Next después de realizar los ajustes.

La siguiente pantalla mostrará los servicios que pueden ser instalados. Seleccionar Install Apache as service, Install MySQL as service e Install Filezilla as service, esto para que el servidor corra de forma optima. **Figura 2.2.**

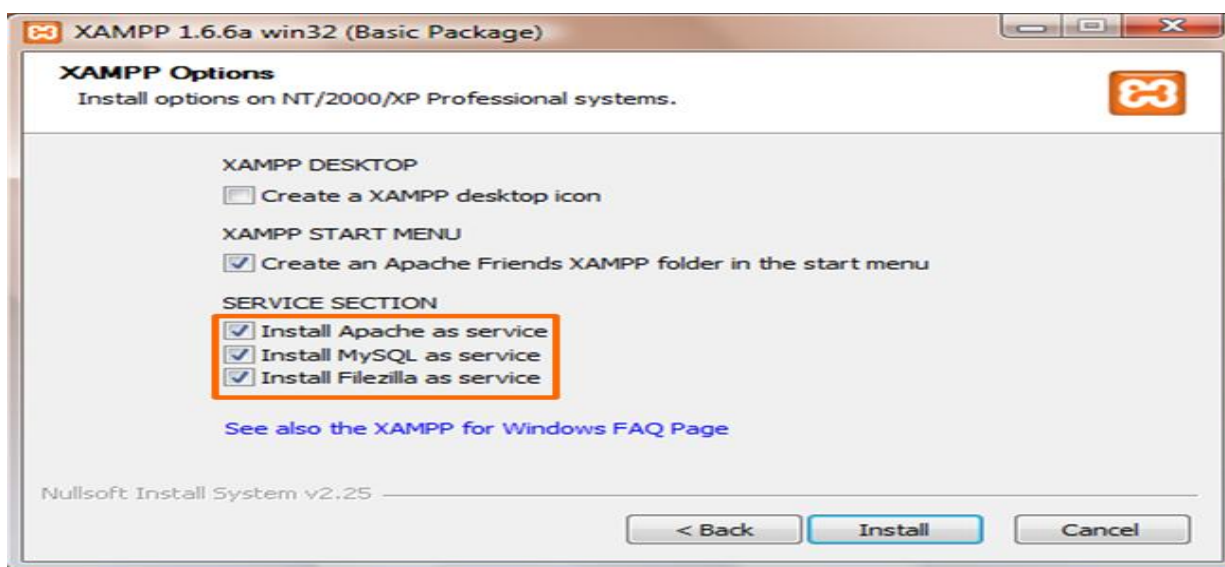


Figura 2.2. Selección de aplicaciones de Xampp a Instalar.

Debe seleccionar el botón de Install, en este paso se descomprimen e instalan los componentes seleccionados.

Aparecerán un par de ventanas de consola, que es la instalación de los servicios. Atentos a esto si tenemos algún antivirus o firewall instalado ya que nos avisará, seguramente, de que se está accediendo a algunos puertos o se esta intentando instalar servicios.

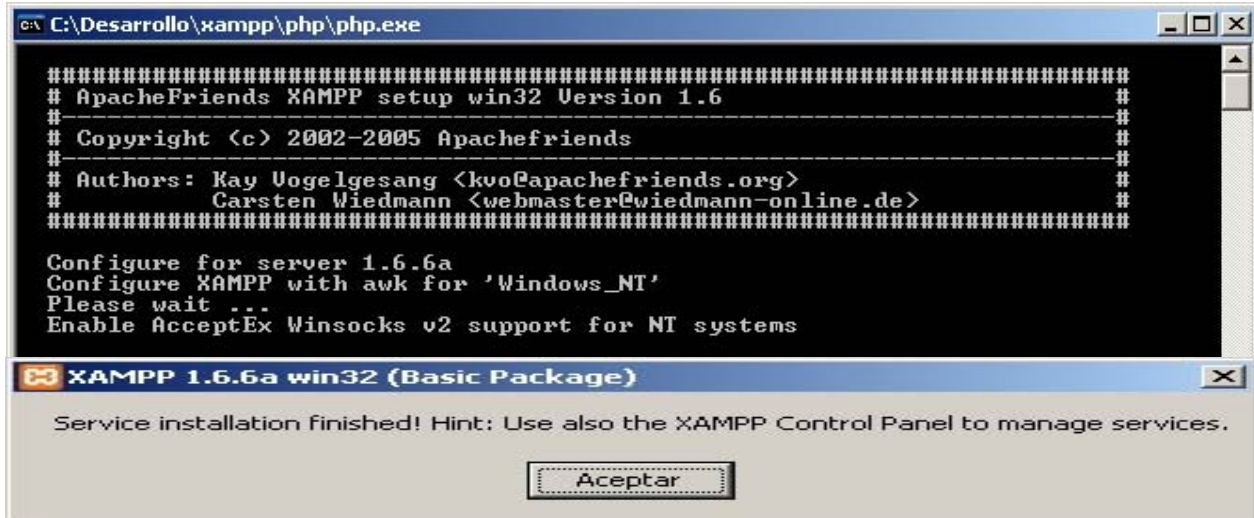


Figura 2.3. Instalación de Servicios y detección de puertos.

Posteriormente mostrará una pantalla en la cual se requiere seleccionar el botón **finish** (finalizar), que desplegará la pantalla como la que se muestra en la **Figura 2.4** en la cual indica que la instalación se realizó con éxito preguntando si desea iniciar el programa debe aceptar y checar si está funcionando adecuadamente.

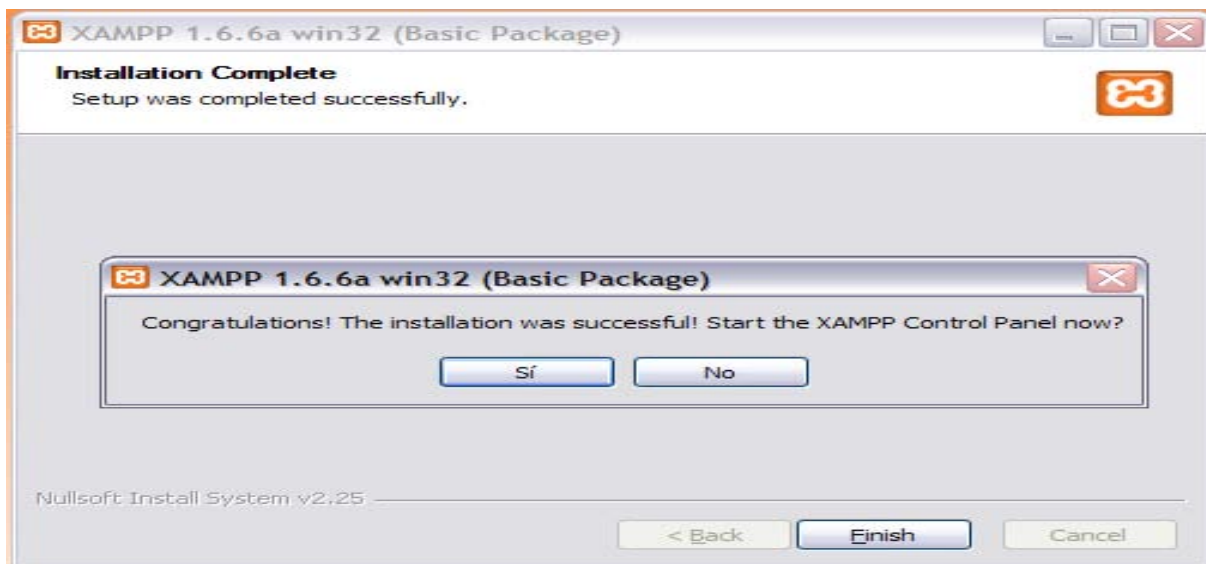


Figura 2.4. Instalación de Xampp realizada con éxito.



Al seleccionar la opción sí; se mostrará la pantalla de la **Figura 2.5** el cual es el panel de control de Xampp y mostrará las aplicaciones que están activadas en ese momento. Comprobar que la aplicación de MySQL y apache estén activadas esto es que estas opciones resalten con un rectángulo de color verde el cual indica que están funcionando correctamente.



Figura 2.5 Panel de Control de la Aplicación Xampp.

3. Creación y Verificación de la Base de Datos.

La aplicación instalada cuenta con una utilidad gráfica, en el menú inicio y en el escritorio puede aparecer el icono de la aplicación instalada Xampp; al momento de inicializarla aparece la aplicación Panel de Control de Xampp.

El primer paso para el acceso de la base de datos es el siguiente, se debe inicializar Internet explorer, Firefox o cualquier navegador de Internet luego se debe introducir



esta dirección <http://localhost/xampp> mostrando la página principal de Xampp con su correspondiente menú de opciones, ver **Figura 3.1**.

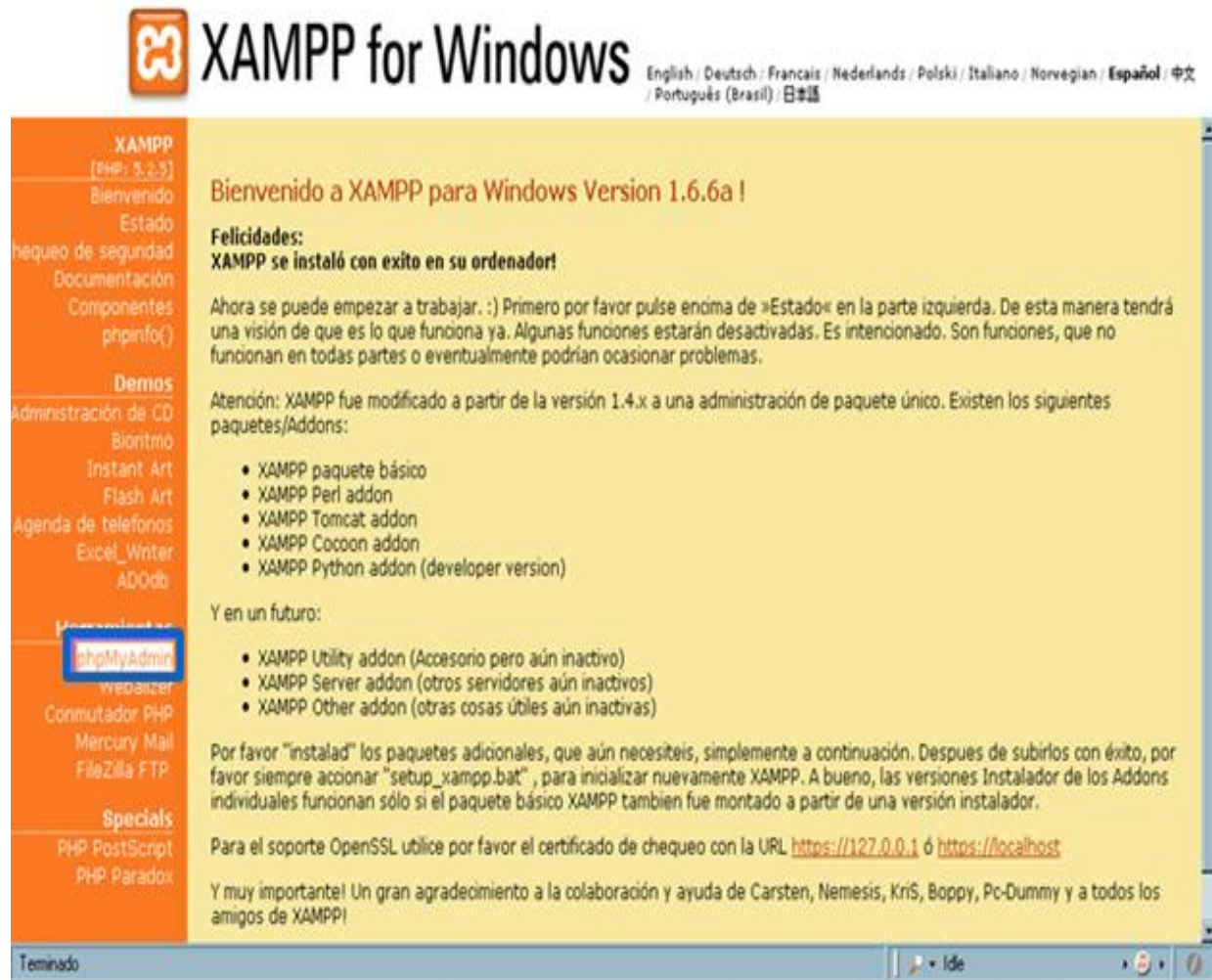


Figura 3.1 Página inicio de Xampp.

Posteriormente se selecciona en el menú la herramienta phpMyAdmin para acceder de manera gráfica a la base de datos MySQL la **Figura 3.2** muestra la pantalla principal de la utileria.

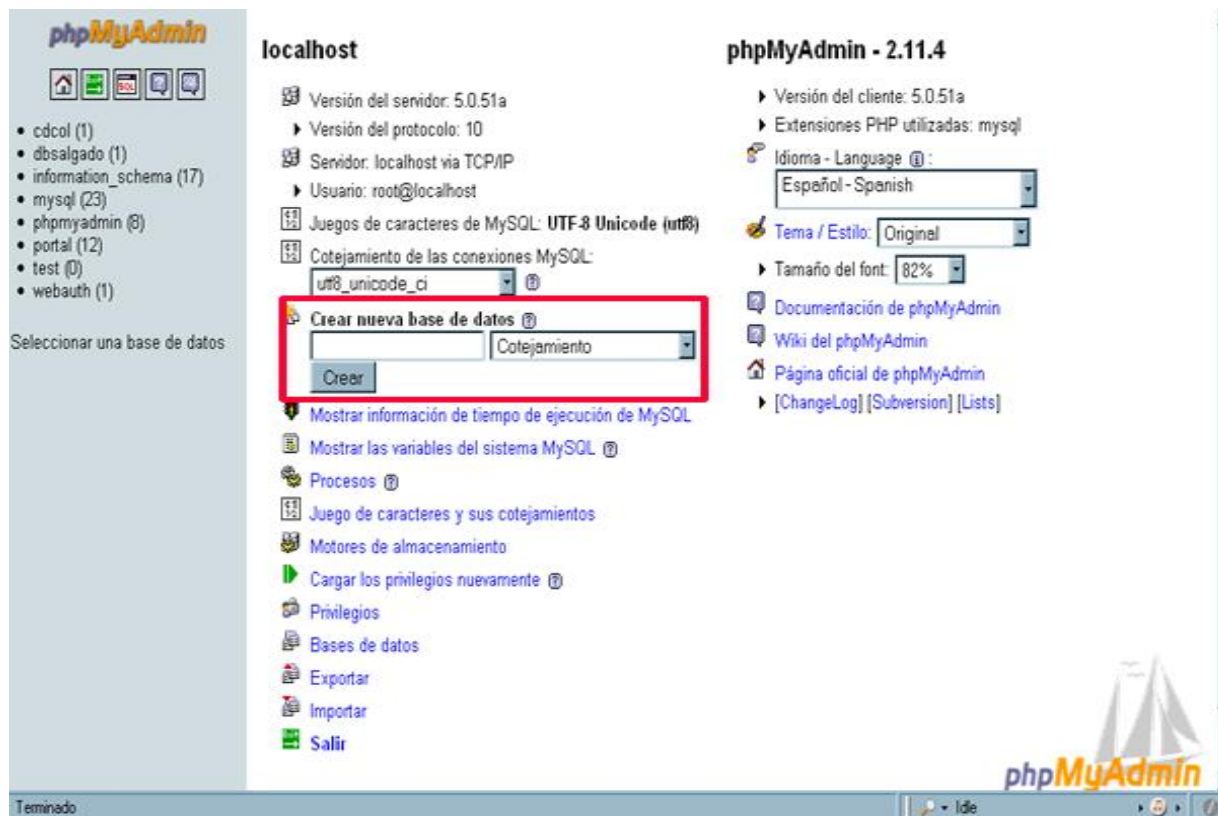


Figura 3.2 Herramienta phpMyAdmin que permite manipular la B. D. MySQL.

Para crear una base de datos como se muestra en la figura, existe una opción que señala **Crear nueva base de datos**, se proporcionan el nombre y el tipo de carácter soportado llamado cotejamiento para esta base de datos se opta por utf8_bin y se selecciona el botón crear. Aparecerá el código y la indicación de que la base de datos fue creada como se puede observa en la **Figura 3.3**.

Al mismo tiempo mostrará la sentencia con que se crea una base de datos cuando no se cuenta con una herramienta grafica.

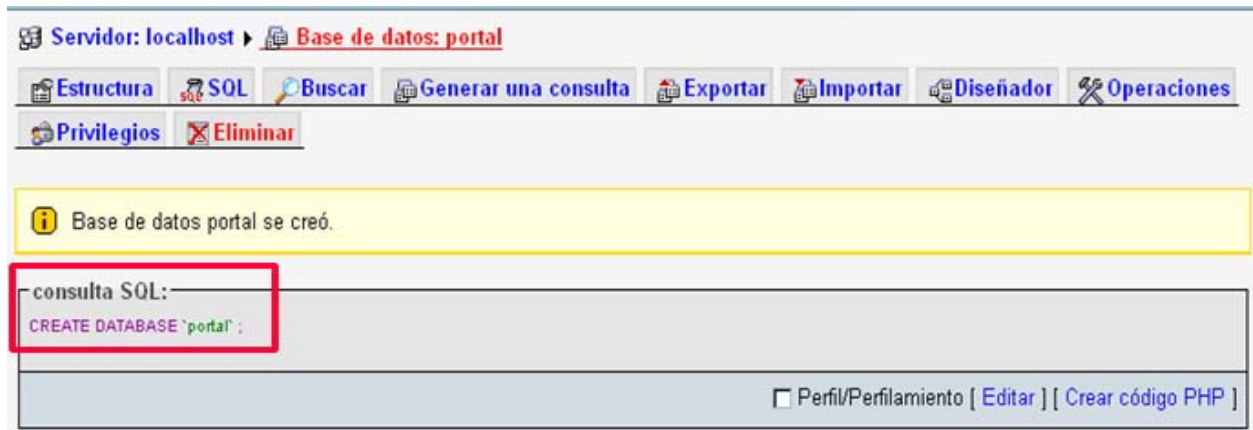


Figura 3.3 Creación de la Base de Datos.

Para generar las correspondientes tablas en la misma pantalla en la parte inferior muestra la opción para crear las tabla de la base de datos ver **Figura 3.4**,



Figura 3.4 Crear nueva tabla en la base de datos.

El siguiente paso es ingresar el nombre de la tabla y el número de atributos que tendrá, seleccionar el botón que indica Continuar.



A continuación se mostrará una pantalla como la que se despliega en la **Figura 3.5** donde se deberán llenar los campos con los nombres correspondientes y sus características particulares de cada uno.

Servidor: localhost ▶ Base de datos: portal ▶ Tabla: plantel

| Campo | Tipo | Longitud/Valores ¹ | Cotejamiento | Atributos |
|---------|---------|-------------------------------|--------------|-----------|
| PL_rfc | VARCHAR | 18 | | |
| PL_nmro | ENUM | 5,0 | | |
| PL_nomb | VARCHAR | 35 | | |
| PL_doml | VARCHAR | 50 | | |
| PL_telf | ENUM | 10,0 | | |
| PL_entd | VARCHAR | 25 | | |
| PL_cpos | ENUM | 5,0 | | |

Comentarios de la tabla:

Motor de almacenamiento: MyISAM

Cotejamiento:

O Añadir campo(s)

Figura 3.5 Creación de Tablas y Columnas.

Ingresar el numero de columnas que se agregaran a nuestra tabla y seleccionar el botón Grabar. En caso de haber concluido exitosamente el paso anterior se desplegará la pantalla con las sentencias empleadas para crear una tabla incluyendo sus correspondientes atributos; esto sirve como apoyo como ya se menciona cuando no se cuenta con una herramienta gráfica **Figura 3.5**.

Tabla "portal"."Plantel" se creó.

```

consulta SQL:
CREATE TABLE `portal`.`Plantel` (
  `PL_rfc` VARCHAR(18) NOT NULL ,
  `PL_nmro` INT(5) NOT NULL ,
  `PL_nomb` VARCHAR(35) NOT NULL ,
  `PL_doml` VARCHAR(50) NOT NULL ,
  `PL_telf` INT(10) NOT NULL ,
  `PL_entd` VARCHAR(25) NOT NULL ,
  `PL_cpos` INT(5) NOT NULL ,
  PRIMARY KEY (`PL_rfc`,`PL_nmro`)
) ENGINE = MYISAM

```

| Campo | Tipo | Cotejamiento | Atributos | Nulo | Predeterminado | Extra |
|----------------------------------|-------------|-------------------|-----------|------|----------------|-------|
| <input type="checkbox"/> PL_rfc | varchar(18) | latin1_swedish_ci | | No | | |
| <input type="checkbox"/> PL_nmro | int(5) | | | No | | |
| <input type="checkbox"/> PL_nomb | varchar(35) | latin1_swedish_ci | | No | | |
| <input type="checkbox"/> PL_doml | varchar(50) | latin1_swedish_ci | | No | | |
| <input type="checkbox"/> PL_telf | int(10) | | | No | | |

Figura 3.6 Sentencia para Crear una tabla.



4. Árbol de Tablas.

Para poder observar el arbol de tablas, en este caso lista del contenido de las tablas en la base de datos; se deberá seleccionar la base de datos a observar y esta mostrará cada una de las tablas que la integran, como se muestra en la **Figura 4.1**.

| Tabla | Acción | Registros | Tipo | Cotejamiento | Tamaño | Residuo a depurar |
|-------------|-----------------|-----------|--------|-------------------|---------|-------------------|
| acceso | | 2 | MyISAM | latin1_swedish_ci | 2.1 KB | - |
| asamblea | | 6 | MyISAM | latin1_swedish_ci | 2.8 KB | 152 Bytes |
| curso | | 10 | MyISAM | latin1_swedish_ci | 2.9 KB | - |
| directorio | | 4 | MyISAM | latin1_swedish_ci | 1.1 KB | - |
| ejemplo | | 19 | MyISAM | latin1_swedish_ci | 2.4 KB | - |
| evento | | 10 | MyISAM | latin1_swedish_ci | 3.1 KB | - |
| personal | | 2 | MyISAM | latin1_swedish_ci | 2.1 KB | - |
| plantel | | 0 | MyISAM | latin1_swedish_ci | 1.0 KB | - |
| reunion | | 11 | MyISAM | latin1_swedish_ci | 2.7 KB | - |
| usuarios | | 4 | MyISAM | latin1_swedish_ci | 3.6 KB | 348 Bytes |
| 10 tabla(s) | Número de filas | 68 | MyISAM | latin1_swedish_ci | 23.6 KB | 500 Bytes |

Figura 4.1 Arbol de Tablas de la Base de Datos Portal.



5. Generación de Consultas en la base de Datos.

Existen varias formas de realizar una consulta, en este caso se muestra la forma gráfica. Seleccionar la opción “**Generar una Consulta**” que se encuentra en el menú superior la cual desplegará otra pantalla con varias opciones y con la opción seleccionar una base de datos, una tabla o una de las entidades de las tablas que se muestra en la **Figura 5.1**.

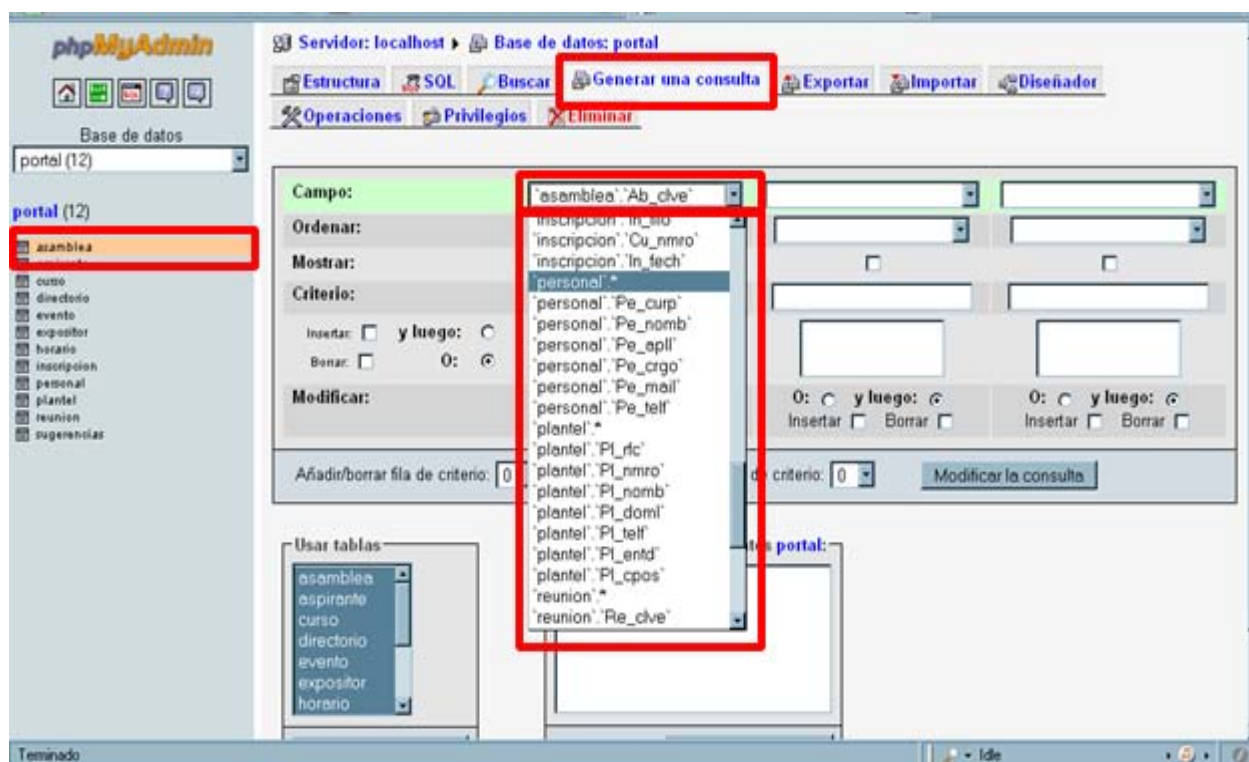


Figura 5.1 Consultas en la base de Datos.

Otra forma de realizar una consulta es seleccionar en el menu de la parte superior de la página del gestor de phpMyAdmin, mostrandose una pantalla donde debe formular la sentencia para realizar la consulta esto se observa en la **Figura 5.2**.

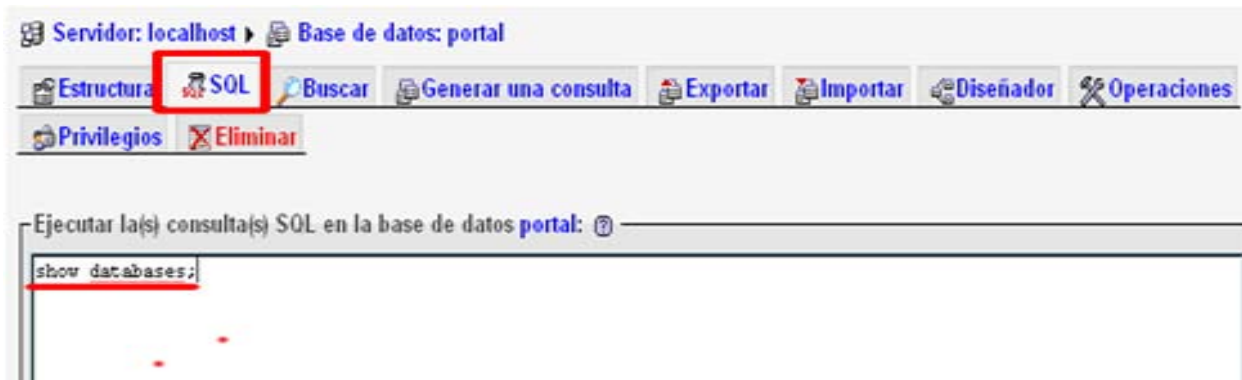


Figura 5.2. Otra forma de realizar una consulta.

6. Respalda Bases de datos utilizando Phpmyadmin.

Una vez que seleccionamos nuestra base de datos, tenemos una página como la mostrada en la **Figura 6.1**.



Figura 6.1 Pantalla principal.

Estructura: Muestra el contenido de las tablas, permitiendo editarlas.



SQL: Se emplea para agregar manualmente datos a las tablas o hacer cambios en ellas, crear, borrar, vaciar, modificar, generar consultas etc. Realiza consultas de manera textual.

Buscar: Como su nombre lo dice ayuda a la búsqueda de texto, frase, palabras, en una o todas las tablas.

Generar una Consulta: Se emplea para facilitar las consultas de las tablas de manera intuitiva y cómoda.

Importar: Esta herramienta es usada cuando se desea agregar una base de datos o tablas provenientes de un programa externo, independiente que sea compatible con esta versión, puede utilizarse la opción de subir archivos docsql sql, mysql, ANSY,DB2, modalidades compatibles con sql y comprimidos , gzip, zip, además de importaciones parciales.

Operaciones: Son opciones extras como vaciar la base de datos, cambiar el nombre, crear nueva tabla, cambiar el tipo de cotejamiento (latin, utf8, ASCII, etc.). Crear esquemas en PDF de las bases de datos, dar mantenimiento preventivo y correctivo de la(s) tabla(s).

Exportar: Permite hacer un respaldo de la base de datos de una o varias tablas o la base de datos completa, en formatos comprimidos, Sql, SVC, Pdf, etc. La opción de exportar permite seleccionar el tipo de formato en que la base de datos puede ser almacenada; las opciones son:

- Datos CSV
- CSV para datos de MS Excel
- Microsoft Excel 2000
- Microsoft Word 2000
- LaTeX
- Hoja de cálculo Open Document
- Texto Open Document
- PDF



- SQL
- XML

o generar archivos descargables tales como gzip, zip.

Seleccionar **SQL** que es el formato que se requiere para hacer el respaldo de la base.

También todas las tablas que se respaldaran. **Figura 6.2.**



Figura 6.2 Respaldo de Tablas.

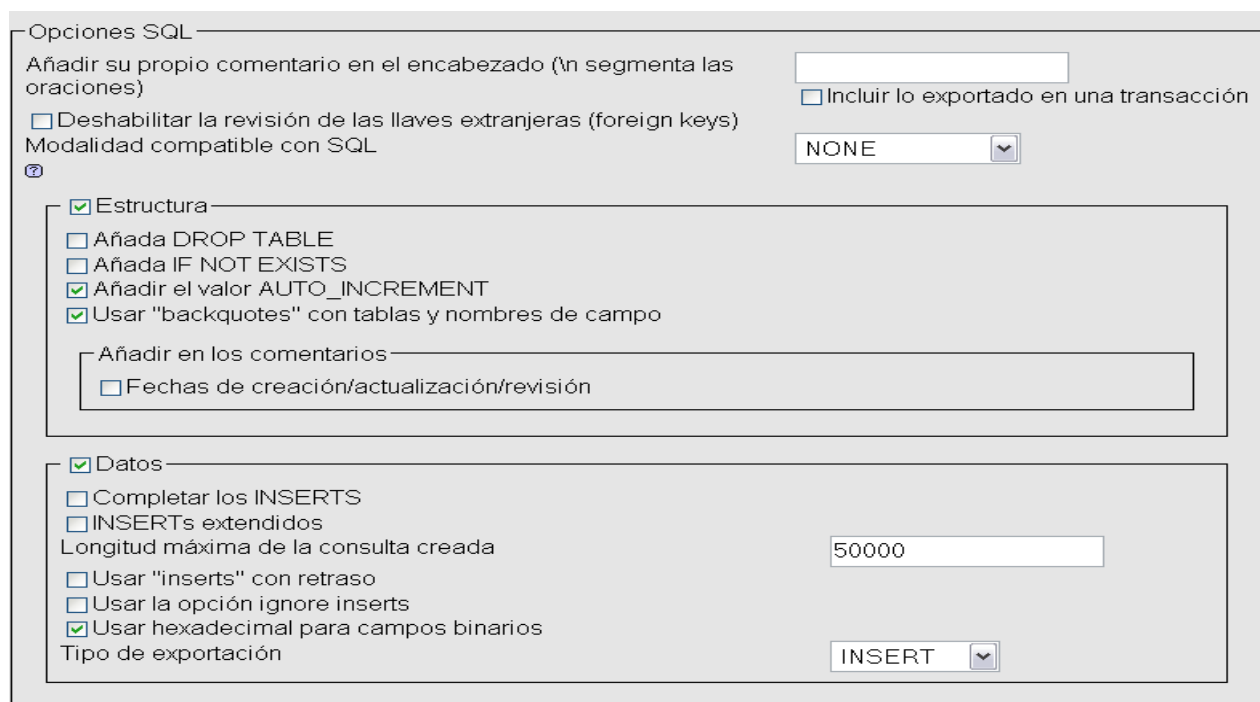


Figura 6.3 Opciones SQL.



En estructura tenemos diferentes opciones ver Figura 6.3.

- **Drop Table (Eliminar Tabla):** Proporciona el código para eliminar una tabla, está debe estar deshabilitada.
- **If not Exist (Si no existe):** Proporciona el código para agregar una tabla siempre y cuando no exista; está deberá estar deshabilitada.
- **Auto_Increment y backquotes (Usar "backquotes" con tablas y nombres de campo):** estas opciones son genéricas y deben ser habilitadas a la hora de respaldar la base de datos.

Datos:

- **Inserts:** Datos que contienen una tabla si no seleccionamos la opción de añadir inserts solo se copiaran las tablas pero no su contenido.
- Nota: si se inserta un respaldo sobre la misma base de datos y esta encuentra un insert igual se detendrá la importación y marcara error.
- **Insert extendido:** Al encontrar un dato dentro de una tabla lo re-escribe esta es una buena opción cuando se importa una tabla y ya se encuentra en la base de datos (tiene que ser en conjunto con IF NOT EXISTS).

Use hexadecimal para BLOB.

Condiciones que se deben seleccionar para generar el respaldo **Figura 6.4.**

Enviar (genera un archivo descargable)

Plantilla del nombre del archivo ⁽¹⁾: (recordar la plantilla)

Compresión: Ninguna "comprimido con zip" "Comprimido con gzip"

Figura 6.4 Selección Condiciones de Respaldo.

Por último se debe seleccionar el nombre del respaldo, si se desea que se comprima en un archivo con extensión zip o gzip o simplemente que te muestren los datos en



pantalla. Para generar una descarga se debe seleccionar la opción **Enviar (generando un archivo)** y luego el botón de continuar.

Resumiendo los pasos a seguir para generar un respaldo son:

- Entrar a phpmyadmin, seleccionar la base de datos.
- Seleccionar la pestaña de Exportar
- Seleccionar exportar SQL y seleccionar la(s) tabla(s) que se requiere exportar.
- En opciones SQL agregar las opciones de la Figura 6.3.
- Seleccionar la forma de descarga, nombre y muy importante debe estar habilitada la opción de Enviar (generando un archivo declarable).

7. Configuración de cuenta de Outlook.

En el menú **Inicio** se selecciona la opción de **Correo electrónico**, como se muestra en la Figura 7.1.



Figura 7.1 Selección de Correo electrónico.



En seguida se observa la ventana de inicio de configuración de Outlook como se muestra en la Figura 7.2 en la cual se debe seleccionar dando click en el botón **Siguiente**.

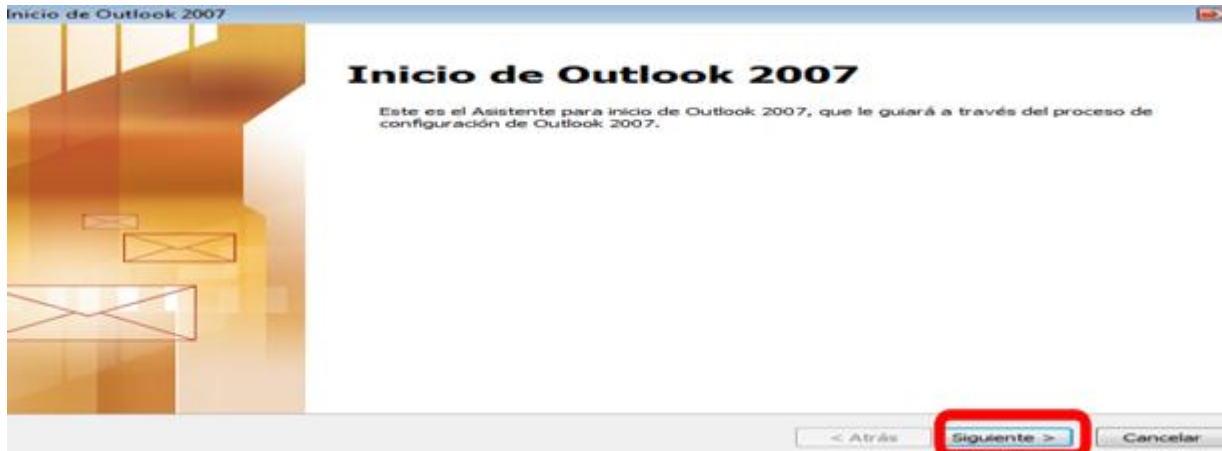


Figura 7.2 Inicio de configuración.

Preguntará si se desea configurar una cuenta de correo, seleccionando **Si**, posteriormente se selecciona el botón **Siguiente**, como se muestra en la Figura 7.3.

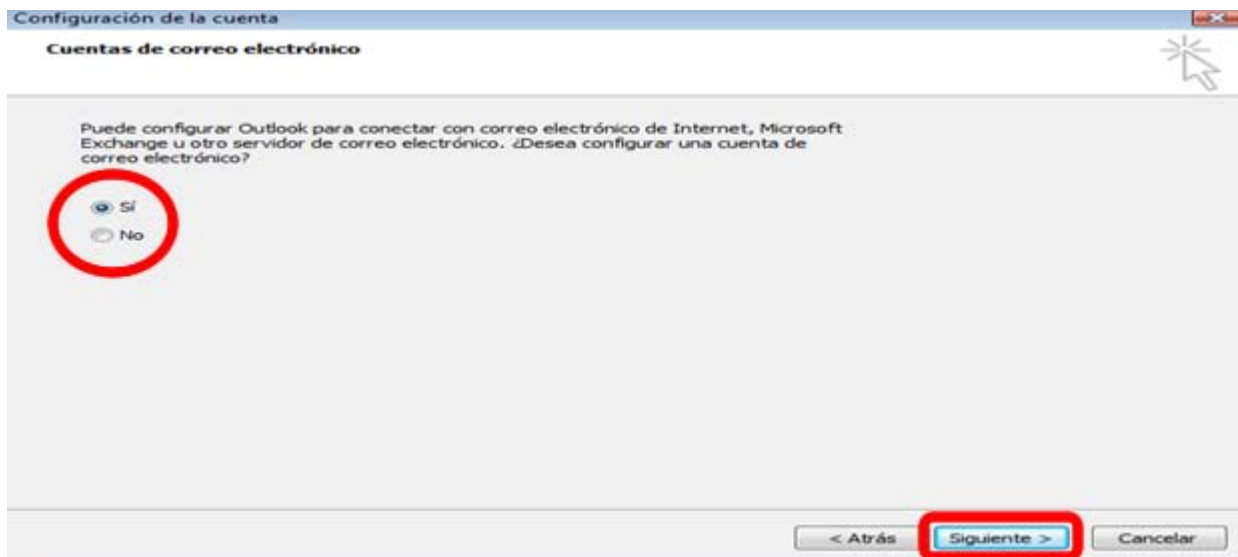


Figura 7.3 Cuentas de correo electrónico.



Posteriormente se piden los datos tales como el nombre con el que se desea que aparezcan los mensajes, el correo electrónico y por último la contraseña, al terminar se selecciona el botón **Siguiente**, como se ve en la figura 7.4.

Agregar una nueva cuenta de correo electrónico

Configuración automática de la cuenta
Haga clic en Siguiente para ponerse en contacto con su servidor de correo electrónico y configurar los parámetros de su cuenta de su proveedor de servicios de Internet o del servidor de Microsoft Exchange.

Su nombre: Mayra Guerrero
Ejemplo: Yolanda Sánchez

Dirección de correo electrónico: mayrarott@hotmail.com
Ejemplo: yolanda@contoso.com

Contraseña: [oculto]
Repita la contraseña: [oculto]
Escriba la contraseña proporcionada por su proveedor de servicios Internet.

Configurar manualmente las opciones del servidor o tipos de servidores adicionales

< Atrás **Siguiente >** Cancelar

Figura 7.4 Configuración automática de la cuenta.

Finalmente muestra una siguiente pantalla en la que menciona que ya se configuro la cuenta seleccionando el botón **Finalizar**, como se observa en la Figura 7.5.

¡Felicidades!

Ha escrito correctamente toda la información necesaria para configurar la cuenta.
Para cerrar el asistente, haga clic en Finalizar.

< Atrás **Finalizar**

Figura 7.5 Fin de Configuración.



En la Figura 7.6 se muestra el proceso de correo electrónico funcional del portal.

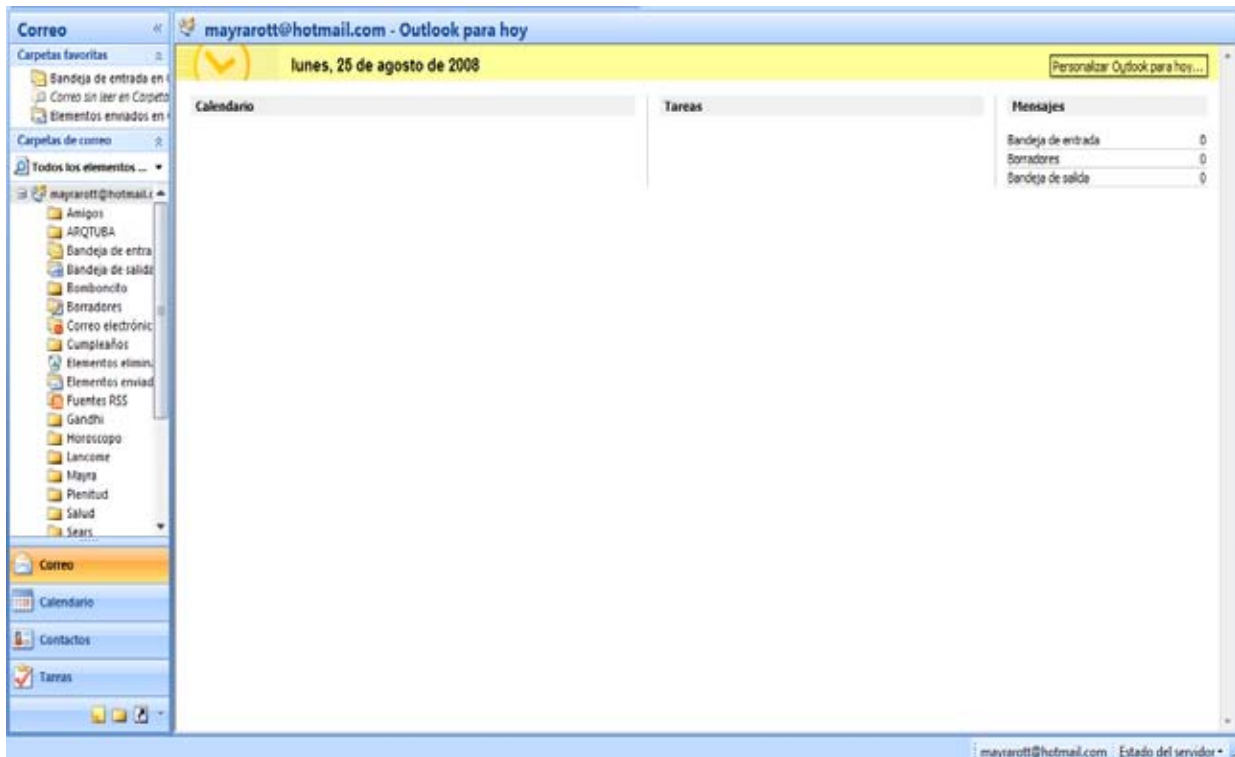


Figura 7.6 Vista de Correo configurado.

8. Medidas de seguridad.

Asegurar que los recursos del sistema de información (material informático o programas) de una organización sean utilizados de la manera que se decidió y que el acceso a la información allí contenida así como su modificación sólo sea posible a las personas que se encuentren acreditadas y dentro de los límites de su autorización.



Se entiende como seguridad un estado de cualquier tipo de información (informático o no) que nos indica que ese sistema está libre de peligro, daño o riesgo. Este peligro es entendido como aquello que pueda afectar su funcionamiento directo o los resultados que se obtienen del mismo.

No existe un sistema 100% seguro. Para que un sistema se pueda definir como seguro debe contar al menos con cuatro características: **Integridad:** La información sólo puede ser modificada por quien está autorizado. **Confidencialidad:** La información sólo debe ser legible para los autorizados. **Disponibilidad:** Debe estar disponible cuando se necesita. **Irrefutabilidad:** (No-Rechazo o No Repudio) Que no se pueda negar la autoría.

Los activos son los elementos que la seguridad informática tiene como objetivo proteger. Son tres elementos que conforman los activos: **Información.** El resguardo de la información en algún medio electrónico o físico. **Equipos que la soportan.** Software, hardware y organización. **Usuarios.** Individuos que utilizan la estructura tecnológica y de comunicaciones que manejan la información.

El activo más importante que se posee es la información y, por lo tanto, deben existir técnicas que la aseguren, más allá de la seguridad física que se establezca sobre los equipos en los cuales se almacena. Estas técnicas las brinda la seguridad lógica que consiste en la aplicación de barreras y procedimientos que resguardan el acceso a los datos y sólo permiten acceder a ellos a las personas autorizadas para hacerlo.

El objetivo de este documento es que el usuario que lo lea adquiera nociones básicas sobre seguridad que pueda aplicar en este desarrollo.



Seguridad en Xampp.

Xampp cuenta ya con ciertas medidas de seguridad que permiten al usuario tener un servidor seguro y que solo ciertos usuarios pueden manipular este. La figura siguiente muestra el estado en que se encuentra xampp además de cómo continuar el chequeo de seguridad que xampp realiza. Apache proveen a XAAMP, un instalador de aplicaciones LAMP fácil de usar y gratuito que trabaja en muchos sistemas operativos, incluyendo GNU/Linux y Windows.



Figura. 8.1. Estado actual de Xampp.

Después de seleccionar el submenú de chequeo de seguridad aparece la figura 8.2. la cual muestra el resumen de los puntos de instalación inseguros y que deben



controlarse, se sugiere seleccionar la liga siguiente <http://localhost/security/xamppsecurityty.php>, que inicia un programa interactivo para configurar los archivos para administrar la seguridad de algunas herramientas.

XAMPP for Windows

Por medio de este resumen puede verse que puntos de la instalación aún son inseguros y tendrían que ser controlados. (Siga leyendo debajo de la tabla.)

| Concerniente a | Estado |
|--|-------------|
| Estas paginas XAMPP NO se visualizan a través de la red | SEGURO |
| MySQL-root tiene clave de acceso | SEGURO |
| PhpMyAdmin password login is enabled. | SEGURO |
| A FTP server is not running or is blocked by a firewall! A FTP server is not running or is blocked by a firewall! | DESCONOCIDO |
| PHP is NOT running in "safe mode" If do you want to offer PHP executions for outside persons, please think about a "safe mode" configuration. But for standalone developer we recommend NOT the "safe mode" configuration because some important functions will not working then. More Info | INSEGURO |

Los puntos marcados en verde estan seguros; los puntos en rojo son definitivamente inseguros y en los amarillos no se pudo comprobar la seguridad (por ejemplo porque el programa a comprobar no estaba en marcha).

Para solucionar estos agujeros en la seguridad llame simplemente al siguiente comando:

=> <http://localhost/security/xamppsecurity.php> <= [allowed only for localhost]

De esta manera se inicia un programa interactivo, que cerrará todos estos agujeros de seguridad.

Please consider this: With more XAMPP security some examples will NOT execute error free. If you use PHP in "safe mode" for example some functions of this security frontend will not working anymore. Often even more security means less functionality at the same time.

The XAMPP default ports:

| | | |
|-------|---------|---|
| ftp | 21/tcp | # File Transfer [Control] (XAMPP: FTP Default Port) |
| smtp | 25/tcp | mail # Simple Mail Transfer (XAMPP: SMTP Default Port) |
| http | 80/tcp | # World Wide Web HTTP (XAMPP: Apache Default Port) |
| pop3 | 110/tcp | # Post Office Protocol - Version 3 (XAMPP: POP3 Default Port) |
| imap | 143/tcp | # Internet Message Access Protocol (XAMPP: IMAP Default Port) |
| https | 443/tcp | # https protocol over SSL (XAMPP: Apache SSL Port) |

Figura 8.2. Chequeo de seguridad Xampp.

Este archivo muestra varias opciones en las cuales se pide un login y un password para la administración y sobre todo el acceso de usuarios en MySQL y XAMPP. Con ello se limita el acceso y solo quien conozca estos datos podrá acceder a estas herramientas y poder modificar su contenido.

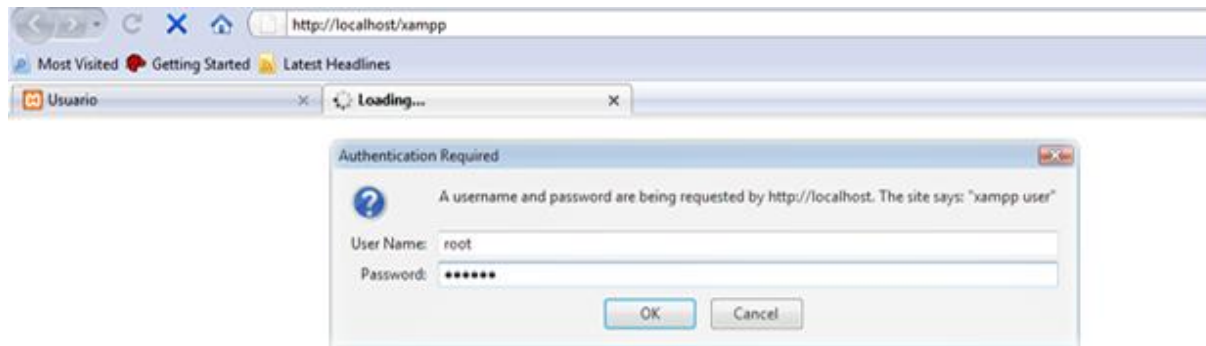


Figura 8.3. Configuración Acceso MySQL y XAMPP.

Después de realizar estos cambios se tendrá un acceso seguro a estas herramientas. A continuación se muestran varias figuras que comprueban que quien no tenga conocimiento de la modificación de los datos agregados no podrá acceder y por tanto hacer uso de la administración de estos servicios.



Figura 8.4. Acceso seguro a Xampp.



¡Autenticación requerida!

El servidor no puede certificar que usted este autorizado para acceder al enlace "/xampp". Usted pudo suministrar información errónea accidentalmente (ejem. una contraseña inválida) o, el navegador no sabe como suministrar la información requerida.

En caso de que a usted le este permitido el uso del documento requerido, le solicitamos de la manera más atenta que por favor vuelva a intentar la operación suministrando nuevamente su identificador y su contraseña.

Por favor contacte con el [webmaster](#) en caso de que usted crea que existe un error en el servidor.

Error 401

[localhost](#)

07/06/08 04:45:36

Apache/2.2.8 (Win32) DAV/2 mod_ssl/2.2.8 OpenSSL/0.9.8g mod_autoindex_color PHP/5.2.5

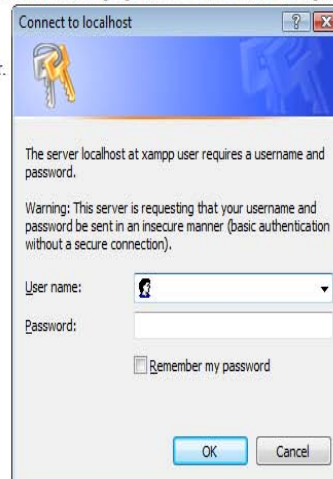


Figura 8.5 Autenticación requerida.

Esta es una medida básica que asegura que el acceso sea solo para quienes conocen este servicio, posteriormente si después de acceder a xampp, el usuario que accedió desea o encuentra el gestor de phpmyAdmin y pretende acceder pedirá el login y el password si estos no se conocen entonces mostrará un error este proceso se muestra en la figuras siguientes.



Figura 8.5. Acceso seguro a phpMyAdmin.

Esta medida también prevé que si el usuario olvida o el sistema detecta que no hay actividad se cierra automáticamente la sesión y el usuario deberá autenticarse nuevamente esta opción podrá deshabilitarse pero se sugiere dejarlo por default.



Figura 8.6 Cierre de sesión por inactividad.

PHP se ha convertido en un lenguaje de scripting esencial para gran parte de sitios Web, y si se tiene planeado utilizarlo, es fundamental realizar una configuración pormenorizada.



Otro detalle que es necesario configurar, es la ruta en la cual se muestra la página principal que generalmente es conocido con el nombre de index.html; en este caso es local y el nombre de la página es sitio.html.

La ruta en la que hay que realizar esta modificación es en la ruta donde se realizó la aplicación de xampp que en este caso es `c://xampp/htdocs/index.html`. Editar este archivo y realizar el cambio siguiente en el renglón donde se encuentran estas líneas `header('Location: '.$uri.'/xampp/');` por esta otra `header('Location: '.$uri.'/Pagina/sitio.html');`; esto permitirá que en lugar de que aparezca la pantalla principal de xampp se observe la página principal del portal, después de que el usuario seleccione esta ruta `http://localhost`. La figura 8.7 muestra la página principal del portal.



8.7. Ruta de la página principal del sitio del portal.

Una ventaja y medida de seguridad es que este cambio permite que pueda observarse la página en cuestión y ocultar la del gestor de xampp.



A continuación se da una definición de algunos de los elementos que deben tomarse en cuenta y con los cuales este sistema trabaja. Debe considerarse que todos estos detalles son medidas que por experiencia los expertos han detectado y sugerido soluciones al mismo.

php.ini. Archivo de configuración de PHP. Puedes ver que php.ini se esta empleando con phpinfo.

httpd.conf. En él guarda su configuración el servidor HTTP Apache; al ser este el más utilizado por desarrolladores y servicios de alojamiento (tanto gratuitos como de pago). Normalmente se encuentra en el directorio conf en el directorio raíz de Apache.

.htaccess. Fichero especial que usa Apache en el que se pueden poner ciertas llamadas a sus directivas o las de algunos módulos que haya cargado. Requiere permisos para poder utilizarlo.

The screenshot shows the XAMPP for Windows control panel on the left and the output of the phpinfo() function on the right. The phpinfo() output is as follows:

| | |
|-----------------------------------|--|
| PHP Version 5.2.4 | |
| System | Windows NT PCRITA 6.0 build 6001 |
| Build Date | Oct 1 2007 20:06:04 |
| Configure Command | cmdscript\inologo\configure.js "--enable-prefix=C:\cygwin\tmp\build\core-php-nts-5.2.4.3992\core-php-nts-5.2.4\php-5.2.4\." "--with-openssl=C:\cygwin\tmp\build\core-php-nts-5.2.4.3992\core-php-nts-5.2.4\php-5.2.4\." "--with-extra-includes=C:\cygwin\tmp\build\core-php-nts-5.2.4.3992\core-php-nts-5.2.4\php-5.2.4\usr\include" "--with-extra-libs=C:\cygwin\tmp\build\core-php-nts-5.2.4.3992\core-php-nts-5.2.4\php-5.2.4\usr\lib" "--disable-ipv6" "--disable-bcmath" "--disable-calendar" "--disable-ftp" "--without-gd" "--disable-odbc" "--without-sqlite" "--disable-tokenizer" "--with-libxml" "--with-iconv" "--disable-force-cgi-redirect" "--enable-fastcgi" "--enable-cli-win32" "--enable-pdo" "--disable-its" "--disable-xmlwriter" |
| Server API | CGIFastCGI |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | C:\Windows |
| Loaded Configuration File | C:\Program Files\Zend\Core\etc\php.ini |
| PHP API | 20041225 |
| PHP Extension | 20060613 |

Figura 8.8. Información phpinfo().



Desactivando Opciones no Recomendables.

Register_globals. Esta opción está deshabilitada por defecto en la versión 4.0.2 de PHP. Register_globals transforma los parámetros incluidos en las peticiones en parámetros globales de PHP.

allow_url_fopen. Esta opción permite a los programadores tratar las URLs como si fueran ficheros, esta opción viene activada por defecto, es habitual usar los datos de la petición para identificar el nombre del fichero a leer. La función include() muestra el contenido del fichero en la página web resultante.

Existen otras funciones como readfile() que también muestran el contenido del fichero por pantalla.

Carga dinámica de módulos. PHP utiliza módulos para añadir funcionalidades de forma dinámica. Puede cargar módulos desde programas usando la función dl() desde un script.

Cuando se carga un módulo, se integra en PHP y se ejecuta con permisos totales. Por lo tanto es factible escribir una extensión para intentar eludir las limitaciones que se establecen a través de la configuración. El ataque describe como un usuario utiliza las opciones de PHP para añadir funcionalidades para cargar en la memoria de Apache código y tomar el control del servidor web. Por lo tanto los módulos que se han de usar son aquellos que están indicados en el fichero php.ini por lo que desactivaremos esta función, enable_dl = Off.

Información sobre PHP en las cabeceras del servidor web. PHP permite incluir su firma en el servidor Web, en la cabecera Header esta característica está orientada a sacar estadísticas de uso de php. Desactivar expose_php = Off



Otra de las opciones para no mostrar esta información en las cabeceras es utilizando la directiva `SecServerSignature` de modsecurity.org [3] con el valor que queramos que se muestre entre comillas.

Desactivando Funciones Y Clases.

Mediante el uso de las directivas `disable_functions` y `disable_classes` es posible deshabilitar funciones y clases. Para evitar estas situaciones configuraremos `disable_functions = openlog`.

Existen otras funciones de integración de PHP/Apache que son potencialmente peligrosas. Y son las siguientes:

| | |
|-------------------------------------|------------------------------------|
| <code>apache_child_terminate</code> | <code>apache_get_modules</code> |
| <code>apache_get_version</code> | <code>apache_getenv</code> |
| <code>apache_note</code> | <code>apache_setenv virtual</code> |

Restricciones de Acceso al Sistema de Ficheros.

La directiva más útil en relación con la seguridad en PHP es `open_basedir`. Esta directiva indica a PHP a que ficheros puede acceder y a cuáles no. El valor de esta directiva es una lista de prefijos de ficheros separados por una coma en Unix y por punto y coma en Windows.

Las restricciones configuradas aquí, afectan a los scripts de PHP y a los ficheros de datos. Se recomienda que se active esta opción incluso en aquellos servidores con un único sitio web, y debe de apuntar un nivel por encima del directorio raíz del servidor web, la configuración sería `open_basedir = /var/www/`



Activando Opciones de Registro.

Por defecto no se registran todos los eventos en PHP. Es importante activar el registro de todos los eventos con:

```
error_reporting = E_ALL          log_errors = On
```

Para ver cualquier error, es necesario activar el registro de errores. Para ello usaremos la opción de configuración `error_log`. Por defecto los errores irán a la salida estándar de error, en este caso, el log de error de Apache. Sin embargo podemos cambiar el destino de los mensajes de error especificando por ejemplo `syslog` en este caso `syslog` registraría los eventos.

En los servidores en producción se deberá desactivar la opción de mostrar errores de inicialización de PHP como errores de ejecución. Para ello:

```
display_errors = Off          display_startup_errors = Off
```

Estableciendo Límites de Memoria.

Php permite establecer la cantidad de memoria que consume un script. En previsión de scripts que puedan estar mal programados y que podrían llegar a consumir toda la memoria y bloquear el sistema. Para ello emplearemos:

Establecemos este valor con:

```
memory_limit = 8M ; 8 MB es la cantidad de memoria por defecto.
```

```
post_max_size = 8M ; 8 MB es el tamaño máximo del POST por defecto.
```

Es posible establecer el tiempo máximo que puede estar PHP procesando la entrada de datos. El límite por defecto son 60 segundos, es conveniente incrementar este valor si hay usuarios con conexiones lentas `max_input_time = 60`



El límite de tiempo que un script PHP puede estar ejecutándose. Para establecer este límite usaremos: `max_execution_time = 30`.

Control de Subida de Ficheros.

Las consideraciones relativas al control de "upload" de ficheros se establecen en el parámetro de `php.ini`, identificado por `file_uploads` que pondremos `file_uploads =Off` o a `On` si queremos permitir subir ficheros mediante las aplicaciones en `php`.

Los parámetros relacionados con esta opción son `upload_max_filesize = 2M` donde se indica el tamaño máximo de ficheros que se pueden subir a la vez, y que por defecto es de 2 MB. Se pueden subir varios ficheros simultáneamente, y el tamaño total es el que se indique en este parámetro.

Control de Sesiones.

HTTP es un protocolo sin control de estados. Lo que quiere decir que el servidor trata cada petición de usuario de forma independiente y sin tener en cuenta que ha sucedido antes. No solo no lo toma en cuenta si no que ni siquiera lo "recuerda". A la hora de realizar desarrollos que utilizan las sesiones para agrupar peticiones, puede ser un inconveniente que no exista un control de estados.

Las sesiones funcionan asignando una porción de información a cada usuario cuando conecta al sitio Web por primera vez. Esta porción de información se denomina identificador de sesión `sessionid`. El mecanismo mediante el cual se asigna el `sessionid` esta ideado para que el navegador del usuario devuelva información al servidor en cada petición.

El servidor Web utiliza la información del `sessionid` para identificar datos del usuario y peticiones anteriores. Desde que el identificador de sesión se utiliza para identificar a un usuario anterior, se podría decir que actúa como un identificador único de usuario



temporal. Con esto surge la problemática de los ataques de secuestro de sesiones, donde conociendo el sessionid se podría suplantar la identidad del usuario autenticado, es decir, se podría conectar a la aplicación con los privilegios del usuario autenticado. El soporte de sesiones en PHP permite a las aplicaciones recordar a un usuario, manteniendo cierta información entre peticiones. Por defecto, esta información se almacena en la ruta /tmp. La información de las sesiones tiene el siguiente formato: sess_ed6128354913y91y3249183y42.

Analizando este código se puede identificar que PHP utiliza identificadores de sesión cuando construye los nombres de los ficheros para los datos de las sesiones (el identificador de sesión después de sess_). Como consecuencia, cualquier usuario del sistema con acceso a la carpeta /tmp podría utilizar los identificadores de sesión y secuestrar la sesión de los usuarios activos. Para evitarlo, especificaremos a PHP como los datos de las sesiones en un directorio separado, con acceso únicamente para el usuario de Apache (www-data). Para ello:

Debe indicarse esta ruta en el fichero php.ini mediante `session.save_path = /var/www/sessions`. Este cambio de configuración no soluciona todos los problemas. Los usuarios del sistema no podrán utilizar los identificadores de sesión si los permisos de la carpeta de sesiones /var/www/sessions están configurados para no permitirlo. Aún así, para cualquier usuario que pueda escribir y ejecutar scripts en PHP en el sistema, será muy sencillo escribir un programa que obtenga la lista de identificadores de sesión ya que el script se ejecutará con los privilegios del usuario que ejecuta el servicio Web.

NOTA:Un directorio de sesiones nunca ha de ser compartido por varias aplicaciones, grupos de usuarios, o sitios Web. Las fugas de identificadores de sesión y los intentos de secuestro pueden evitarse con la ayuda de la opción `session.referer_check`. Activando esta opción PHP analizará el contenido de la cabecera Referer en busca de



la cadena que le indiquemos. Se debería establecer este valor a una cadena que contenga parte del nombre de dominio del sitioWeb.

En este caso se debe indicar `session.referer_check = http://localhost/Pagina/sitio.html`. Como la cabecera. Referer contiene la URL de la página anterior que ha visitado el usuario, contendrá el nombre del dominio del sitio Web en todas las peticiones. Sin embargo, si alguien sigue un link desde otro sitio y llega a el site con un identificador valido de sesión `sessionid`, PHP rechazará la petición.



Figura 8.9. Empleo de sesiones gestor de la base de datos.

Activando las Opciones de Modo Seguro (SAFE_MODE). Es una aproximación a establecer elementos de seguridad para los desarrollos en PHP. Activando esta opción, `safe_mode = On` el motor de PHP impone una serie de restricciones, haciendo que la ejecución de scripts sea más segura.

Restricciones de acceso a ficheros. Las verificaciones que establece `safe_mode` tienen su mayor impacto en el acceso a ficheros. Con este modo se realizan



verificaciones especiales antes de cualquier operación sobre un fichero. Se comprobará que el uid del propietario del fichero coincida con el uid de la cuenta de usuario que posea el script. Esta comprobación puede generar problemas en los siguientes casos:

- Si existe más de un usuario con permisos de escritura en el árbol de directorios del servidor web. Tarde o temprano un script de otro usuario intentará acceder a un fichero propiedad de otro usuario.
- Si las aplicaciones crean ficheros en tiempo de ejecución.

Este segundo caso es el que genera más problemas, la mayoría de las aplicaciones en PHP se usan como CMS (Content Management Systems) o sistemas de gestión de contenidos y todos ellos crean ficheros. La solución más sencilla para evitar este problema pasa por tener la cuenta de desarrollo y la cuenta de Apache en el mismo grupo, y relajar la verificación de uid usando en su lugar el gid. Para ello se establece `safe_mode_gid = On`.

Restricciones de variables de entorno. La posibilidad de escritura en variables de entorno usando la función `putenv()` esta restringida cuando se usa el modo seguro.

La primera de las siguientes dos directivas `safe_mode_allowed_env_vars`, contiene una lista de prefijos separados por coma que indican que variables de entorno pueden ser modificadas. La segunda directiva `safe_mode_protected_env_vars` prohíbe que ciertas variables de entorno separadas por coma, puedan ser modificadas.

```
# allow modification of variables beginning with PHP_ safe_mode_allowed_env_vars =  
PHP_
```

```
# no one is allowed to modify LD_LIBRARY_PATH safe_mode_protected_env_vars =  
LD_LIBRARY_PATH
```

Restricciones de ejecución de procesos externos. El modo seguro establece restricciones en cuanto a la ejecución de procesos externos. Sólo aquellos binarios



indicados en el directorio "seguro" podrán ser ejecutados en los scripts desde PHP, indicando la ruta de estos binarios con: `safe_mode_exec_dir = /var/www/bin`

Las funciones afectadas son:

| | |
|-------------------------|-----------------------|
| <code>exec()</code> | <code>system()</code> |
| <code>passthru()</code> | <code>popen()</code> |

Algunos métodos de programación no funcionarán en modo seguro: `shell_exec()`, `backtick operator` entre otros.

Protecciones más allá de PHP.ini.

Existen proyectos desarrollados exclusivamente para establecer más mecanismos de control en relación con la seguridad en php, por ejemplo `hardened-php.net` [es un parche para PHP que habilita nuevas opciones de configuración desde `php.ini`, estas características afectan a elementos como:

- Elementos de registro o log, permitiendo gestionar los eventos en función de distintas clasificaciones además de la posibilidad de establecer prioridades y criticidad. Se podrán enviar los eventos a scripts de registro externos, o incluir las direcciones IP en los eventos extrayéndolas de las cabeceras `X-Forwarded-For`.
- Filtrado del contenido de GET, POST o COOKIE, en cuanto a número de variables, longitud máxima del nombre de la variable, longitud máxima del índice del array, longitud máxima del valor de la variable, máxima profundidad del array, etc.
- Permite limitar el número máximo de ficheros que se pueden subir.
- Protección del núcleo de PHP y extensión contra vulnerabilidades de formato de cadena (format string).
- Protecciones en relación con la gestión de la memoria.
- Protecciones en tiempo de ejecución, como:
 - Limite de profundidad de ejecución.



- Registro de peticiones fallidas de extensiones de bases de datos *SQL.
- Permite detener un script tras un fallo de ejecución de SQL.
- Es posible prohibir múltiples cabeceras HTTP en llamadas a header() por defecto.
- Límites en relación con ficheros como, filtro en la longitud del nombre de un fichero, filtro de URL, filtro de ficheros subidos, permite truncar el nombre de los ficheros.
- Control del límite de memoria para que no pueda crecer por encima del límite configurado.

Es importante mantener actualizada la versión de phpMyAdmin para corregir problemas de seguridad. Instalar la aplicación en un directorio con un nombre y ubicación no convencional, de esta forma, si llegara a descubrirse un nuevo problema, la instalación no será fácilmente accesible. Asegurar el directorio completo con autenticación de Apache utilizando AuthType, AuthName, etc.

Regularmente revise los registros de acceso en busca de actividad sospechosa. No confié en sumarios y graficas. Revise los "raw logs" (registros en crudo) para detalles más reales. Configure los filtros de Apache mod_security y mod_rewrite para que bloqueen ataques PHP.

Las anteriores son una serie de funcionalidades que se han detectado en el desarrollo de las aplicaciones que integra este sistema hoy en día, por lo que se sugiere documentarse y prever las nuevas funcionalidades en posteriores versiones.



MANUAL DE USUARIO



**Portal de la Academia Regional de Informática de la
DGETI.**





Manual de usuario Gestor de Base de Datos.

Entrar a la página de inicio que tiene la ruta: <http://localhost/> (Figura 1).



Figura 1. Página de inicio.

Debe insertar un login y password validos y que solo él(los) administrador(es) conocen y ha(n) almacenado anteriormente en la base de datos. Si el proceso se realiza satisfactoriamente se ingresa a la página principal (Menú Principal).



Figura 2. Página Principal.



Para realizar consultas, agregar, borrar o actualizar datos de alguna de las tablas existentes se debe seleccionar en el menú principal mostrado en la figura anterior uno de los iconos existentes (Asamblea, Reunión, curso, etc.), posicionando el cursor en la imagen o texto y dando un clic en el botón derecho del mouse. Tomar en cuenta que estos procedimientos serán los mismos para cada una de las opciones del submenú (consultar, agregar, borrar y actualizar) de las tablas existentes.

Una vez que se selecciona la tabla a editar, se mostrará la pantalla que se muestra en la Figura 3.



Figura 3. Página tabla Asamblea.

El link Inicio tiene la función de regresar a la Página del Menú Principal mostrada en la figura 2.

El link Cerrar tiene la función primordial de cerrar la sesión que inicio el usuario al ingresar con login y password, todo esto como medida de seguridad del gestor.



Consulta.

1.- Seleccionar el link **consultar** de la Figura 3. A continuación aparecerá la pantalla con los registros que contiene dicha tabla. Figura 4.

| Plantel | Fecha | Nombre | Minuta | O. Día |
|---------|------------|---|------------|---|
| 23 | 21/11/2008 | Asamblea Ordinaria | Pendiente | Lista de Asistencia. Acuerdos de Academias |
| 33 | 30-11-2008 | Tercera Asamblea Estatal | Pendiente | Basificación Nivel A |
| 25 | 23/09/2008 | Cuarta Asamblea Ordinaria | Pendiente | Lista Asistencia, Evaluación Final Semestre |
| 24 | 21-09-2008 | Revisión Plenario de Academia de Matemática | En proceso | Lista de Asistencia, Asignación de plazas |
| 54 | 03-09-2008 | Primera Reunión | Pendiente | Propuesta cambio de directivos |

Figura 4. Pantalla de Consulta.

Agregar.

- 1.- Seleccionar link **agregar**, parecerá la pantalla de captura. Figura 5.
- 2.- Ingresar los datos.
- 3.- Seleccionar botón Aceptar información.

| | |
|---------------------|---|
| No. de Plantel: | 21 |
| Fecha de Asamblea: | 22/10/2008 |
| Nombre de Asamblea: | Reunión de Colegio |
| Minuta: | Pendiente |
| Orden del día : | de reunión anterior. Asuntos Generales. |

Limpiar Aceptar información

Figura 5. Pantalla Agrega Nuevo Registro.



Borrar.

- 1.- Seleccionar link **borrar**.
- 2.- Seleccionar el registro que se desea borrar ver Figura 6.
- 3.- Seleccionar botón Borrar.

Asamblea

| elección | Plantel | Fecha | Nombre | Minuta | O. Día |
|-------------------------------------|---------|------------|---|------------|--|
| <input type="checkbox"/> | 21 | 21/11/2008 | Asamblea Ordinaria | Pendiente | Lista de Asistencia. Acuerdos de Academias |
| <input type="checkbox"/> | 40 | 30/11/2008 | Tercera Asamblea Ordinaria Estatal | Pendiente | Basificación Nivel A. Diálogo Abogado |
| <input type="checkbox"/> | 24 | 23/09/2008 | Cuarta Asamblea Ordinaria | En proceso | Lista Asistencia, Evaluación Final Semestre |
| <input type="checkbox"/> | 23 | 21/09/2008 | Revisión Plenario de Academia de Matemática | En proceso | Lista de Asistencia, Asignación de plazas |
| <input type="checkbox"/> | 52 | 03/09/2008 | Primera Reunión | Pendiente | Propuesta cambio de directivos |
| <input type="checkbox"/> | 40 | 19/09/2008 | Asamblea Interina Septiembre | Pendiente | 1. Lista de Asistencia. 2. Asignación Horas Base 3. Asuntos varios |
| <input checked="" type="checkbox"/> | 21 | 22/10/2008 | Reunión de Colegio | Pendiente | Lista de Asistencia. Lectura y aprobación de la minuta de reunión anterior. Asuntos Generales. |

| Plantel | Fecha | Nombre | Minuta | O. Día |
|---------|------------|---|------------|--|
| 21 | 21/11/2008 | Asamblea Ordinaria | Pendiente | Lista de Asistencia. Acuerdos de Academias |
| 40 | 30/11/2008 | Tercera Asamblea Ordinaria Estatal | Pendiente | Basificación Nivel A. Diálogo Abogado |
| 24 | 23/09/2008 | Cuarta Asamblea Ordinaria | En proceso | Lista Asistencia, Evaluación Final Semestre |
| 23 | 21/09/2008 | Revisión Plenario de Academia de Matemática | En proceso | Lista de Asistencia, Asignación de plazas |
| 52 | 03/09/2008 | Primera Reunión | Pendiente | Propuesta cambio de directivos |
| 40 | 19/09/2008 | Asamblea Interina Septiembre | Pendiente | 1. Lista de Asistencia. 2. Asignación Horas Base 3. Asuntos varios |

Figura 6. Pantalla Borrar Registro y Consulta proceso realizado con éxito.



Actualizar.

- 1.- Seleccionar link **actualizar**, parecerá la pantalla de captura.
- 2.- Seleccionar el registro que se desea actualizar ver Figura 7.
- 3.- Modificar los datos que se desean actualizar.
- 3.- Seleccionar botón Actualizar.



Figura 7 Pantalla Actualizar Registro.

Se debe aclarar que el procedimiento anterior aplica para todos los demás submenús al igual que el submenú usuarios.

A continuación se mostraran las pantallas de los submenús existentes describiendo para cada uno de ellos el contenido que debe agregarse a cada uno de los campos.



Pantalla de Captura Asamblea.

Asamblea



| | |
|---|---|
| No. de Plantel: | <input type="text" value="21"/> |
| Fecha de Asamblea: | <input type="text" value="22/10/2008"/>  |
| Nombre de Asamblea: | <input type="text" value="Reunión de Colegio"/> |
| Minuta: | <input type="text" value="Pendiente"/> |
| Orden del día : | <input type="text" value="de reunión anterior. Asuntos Generales."/> |
| <input type="button" value="Limpiar"/> <input type="button" value="Aceptar información"/> | |

| Campo | Descripción Captura |
|---------------------------|--|
| No. de Plantel | Número del Plantel donde se realizará la Asamblea. |
| Fecha de Asamblea | Fecha en que se realizará la Asamblea. |
| Nombre de Asamblea | Nombre general que se le quiera dar a la Asamblea. |
| Minuta | Después de Realizada la asamblea se puede agregar los puntos más relevantes que se trataron. |
| Orden del día | Agregar uno o varios puntos a tratar en la reunión que se esta dando de alta. |



Pantalla de Captura Curso.

Curso



| | |
|-----------------------------|--|
| Nombre: | Mmodelado con Plastilina |
| Profesor: | Julia Cuevas Garcia |
| Horas: | 30 horas ▾ |
| Plantel: | 32 |
| Cupo: | 30 personas ▾ |
| Fecha inicial (dd/mm/aaaa): | 23/10/2008  |
| Fecha Final (dd/mm/aaaa): | 04/11/2008  |
| hora(HH:MM): | 15:00 ▾ |

| Campo | Descripción Captura |
|-----------------------|---|
| Nombre: | Nombre general que se le quiera dar a la Asamblea. |
| Profesor: | Nombre del profesor que impartirá el curso. |
| Horas: | Tiempo de duración del curso en horas. |
| Plantel: | Nombre del Plantel donde se impartirá el curso. |
| Cupo: | Agregar uno o varios puntos a tratar en la reunión que se esta dando de alta. |
| Fecha Inicial: | Fecha en que el curso se llevara a cabo. |
| Fecha Final: | Fecha en que se pretende dar por finalizado el curso |
| Hora: | Hora en que el curso iniciará |



Pantalla Captura Eventos.

Eventos

| | |
|-----------------------------|---|
| No. Plantel: | <input type="text" value="51"/> |
| Nombre del Evento: | <input type="text" value="Avances Científicos en Robótica."/> |
| Fecha inicial (dd/mm/aaaa): | <input type="text" value="21/10/2008"/> |
| Fecha final (dd/mm/aaaa): | <input type="text" value="24/10/2008"/> |
| Sede del Evento: | <input type="text" value="Auditorio Principal"/> |
| Hora (HH:MM): | <input type="text" value="12:00"/> |
| Tipo de Evento: | <input type="text" value="Conferencia"/> |

| Campo | Descripción Captura |
|--------------------------|---|
| No. Plantel | Número de identificación del plantel. |
| Nombre del Evento | Nombre del Evento a llevarse a cabo |
| Fecha Inicial | Fecha de Inicio de Evento |
| Fecha Final | Fecha de Finalización del Evento |
| Sede del evento | Lugar donde se realizará el evento. |
| Hora | Hora de inicio |
| Tipo de Evento | Clasificación de Evento. (concurso, conferencia, ciencia, exposición) |



Pantalla de Captura Reunión.

Reunión

| | | |
|---------|------------------------|---|
| Fecha: | 16/10/2008 | |
| Lugar: | Sala Junta tercer piso | |
| Nombre: | Reunión de Colegio | |
| Hora: | 13:30 | ▼ |

| Campo | Descripción Captura |
|--------|--|
| Fecha | Fecha en que se realizará la Asamblea. |
| Lugar | Ubicación donde se realizará la Asamblea. |
| Nombre | Nombre general que se le quiera dar a la Asamblea. |
| Hora | Hora en que dará comienzo la Asamblea. |



Pantalla de Captura Usuario.

Usuario



Login:

Password:

Repite Password:

Nombre:

Apellidos:

E-mail:

| Campo | Descripción Captura |
|-----------------|--|
| Login | Nombre con que se identificará al usuario |
| Password | Clave que permitirá al usuario acceder al gestor |
| Repite Password | Valida que el password introducido anteriormente sea el mismo. |
| Nombre | Nombre del Usuario |
| Apellidos | Apellidos del Usuario |
| E-mail | Correo electrónico del usuario. |