



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

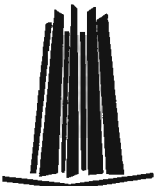
FACULTA DE ESTUDIOS SUPERIORES ARAGÓN

**“SISTEMA DE ADMINISTRACIÓN DE EVENTOS DEL
LABORATORIO L3 DE LA DIVISIÓN DE LAS CIENCIAS
FÍSICO MATEMÁTICAS Y DE LAS INGENIERÍAS DE LA
ENEP ARAGÓN”**

T R A B A J O E S C R I T O
EN LA MODALIDAD DE SEMINARIOS
Y CURSOS DE ACTUALIZACIÓN Y
CAPACITACIÓN PROFESIONAL, QUE
PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN
P R E S E N T A:

CARLOS VÁZQUEZ OROZCO

**ASESOR :
ING. RODOLFO VÁZQUEZ MORALES**



MÉXICO, 2005.

0350983



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.



INDICE

Capítulo I Diplomado de Sistemas Web

1.1 Módulo I Fundamentos del desarrollo de sistemas en web-----	1
1.2 Módulo II Análisis y diseño de sistemas orientados al web -----	9
1.3 Módulo III interfase de usuario -----	15
1.4 Modulo IV Capa de datos -----	21
1.5 Módulo V Reglas de negocios -----	29
1.6 Módulo VI Implementación-----	32

Capítulo II Análisis y diseño

2.1 planteamiento del problema-----	35
2.2 Funciones del sistema-----	36
2.3 Casos de uso del sistema -----	37
2.4 Diagramas de secuencia del sistema -----	52

Capítulo III Interfase de usuario

3.1 Mapa de navegación -----	66
3.2 Diseño de las paginas web -----	69

Capítulo IV Capa de datos

4.1 Concepción de la base de datos -----	79
4.2 Creación de la base de datos -----	86
4.3 Acceso a la base de datos -----	91
4.4 Seguridad -----	93

Capítulo V Capa de reglas de negocios

5.1 Reglas de negocios del sistema -----	96
5.2 Plan de trabajo para la implementación del sistema -----	98
5.3 Estándares de programación -----	99
5.4 Pruebas del sistema -----	100
5.5 Mapa del sitio web -----	101

Conclusiones -----	104
---------------------------	-----

Bibliografía -----	106
---------------------------	-----

INTRODUCCION

Hoy en día las organizaciones del mundo moderno, basan su operación cada vez más en el empleo de las herramientas informáticas, siendo una de las más utilizadas los Sistemas de Información en Web, difundiendo a todo mundo, información acerca de los servicios y/o productos que ofrecen.

Los diferentes tipos de organismos informan acerca de sus actividades y establecen comunicación con otras personas u organismos, y toda esta información siempre esta disponible las 24 horas del día, los 365 días del año, por lo que hace a la Web un medio muy importante y estratégico para compartir información y hacer negocios.

De ésta manera la implementación de aplicaciones en Web, es una nueva y excelente forma de desarrollo o crecimiento, lo cual implica retos no enfrentados previamente por los programadores, administradores de proyectos, analistas de sistemas etc.

Las aplicaciones Web cambian profundamente los conceptos preestablecidos acerca de la programación de aplicaciones para escritorio. Teniendo que considerar cuestiones de seguridad, presentación de datos, disponibilidad de información, etc. toman un giro diferente, que pueden tomar desprevenido a la persona que desea participar en este tipo de desarrollos. Por lo que es necesario que el profesional de la computación actualice sus conocimientos para enfrentar este tipo de proyectos.

El presente trabajo escrito presenta los contenidos del Diplomado Desarrollo de Sistemas en Web, con una aplicación real que demuestra los conocimientos adquiridos, demostrando que el objetivo principal del Diplomado se cumplió, pues sus participantes se actualizaron en el desarrollo de estas aplicaciones, adquiriendo los conocimientos para poder enfrentar el desarrollo e implementación de aplicaciones en Web.



El trabajo esta dividido en cinco capítulos, el primer capítulo es una reseña de los temas vistos en el Diplomado Desarrollo de Sistemas en Web, en los capítulos restantes se describe el análisis y desarrollo de el caso práctico Sistema de Administración de Eventos del Laboratorio L3 de la División de las Ciencias Físico Matemáticas y de las Ingenierías de la FES Aragón. Cabe señalar que el nombre del proyecto se refiere a la ENEP Aragón, ya que el registro se realizó cuando aún no se otorgaba el status de Facultad de Estudios Superiores Aragón, pero el diseño de la interfaz se ha realizado con el nuevo título de Facultad, por lo tanto, en la redacción del trabajo hablamos de FES y no de ENEP.



CAPITULO I

DIPLOMADO DE SISTEMAS WEB

En este capítulo se presenta una reseña de cada uno de los seis módulos, que formaron parte de el Diplomado de Desarrollo de Sistemas web, el cual tuvo una duración de 160 horas.

1.1 Modulo I: Fundamentos del desarrollo de Sistemas en Web

El objetivo de este modulo fue analizar, proponer y diseñar una infraestructura de hardware y software para dar soporte a un sistema que sera utilizado en línea vía www, así como también conocer los fundamentos del modelo de tres capas.

Al desarrollar un sitio web, se pueden utilizar una gran variedad de productos diferentes:

- Hardware para el servidor web
- Sistema Operativo
- Software de servidor web
- Sistema de administrador de base de datos
- Lenguaje de secuencia de comandos o de programación

Alguna de estas opciones depende de otras, por ejemplo, no todos los sistemas operativos se ejecutan sobre todo el hardware, ni todos los lenguajes de secuencia de comandos se pueden conectar a la base de datos, etc. En la actualidad existen diversos modelos de desarrollo de aplicaciones web, cada uno con diversas ventajas y desventajas. Estos modelos utilizan diferentes tipos de tecnología, tanto de hardware como de software, y tienen costos de licencias de uso muy variados, desde las propuestas desarrolladas con software que se puede utilizar de manera gratuita, hasta los modelos que usan software con costo de varios millones de dólares.



Elementos de hardware

Para elegir el hardware que se requiere para montar un servidor web, es muy importante tener en cuenta las tareas que haya que cumplir el servidor, es decir, si las aplicaciones web manejaran páginas estáticas, páginas dinámicas, multimedia, bases de datos y estimar el número de vistas que recibirá, así como también el software que se utilizara (Sistema Operativo, servidor, etc.). Todo esto servirá para hacer una estimación de la potencia y características de la computadora que necesitará.

Para elegir un buen hardware hay que considerar cuatro factores:

- Memoria: nunca como en este caso conviene no escatimar y evitar ahorros contraproducentes.
- Procesador: absolutamente aconsejable orientarse a procesadores de gran potencia.
- Disco: mejor orientarse hacia SCSI, por que libera casi del todo al procesador de cálculos que interesan a los discos. También en este caso cuanto más rápidos son, mejor responde el servidor a los problemas de tráfico elevado.
- Tarjeta de video: una tarjeta de video baja calidad es perfecta. El uso que se hace de esta tarjeta se limita a pilotear un monitor, casi siempre apagado.

En seguida se muestra las características de tres servidores:

<i>Servidor dedicado 1</i>	<i>Servidor dedicado 2</i>	<i>Servidor dedicado 3</i>
Intel Pentium IV 2.8 Ghz	Intel Pentium IV 2.67 Ghz	Intel Pentium IV 3.6 Ghz
256 Mb de RAM	256 Mb de RAM	512 Mb de RAM
1 disco EIDE 40 Gb	1 disco SCSI 18.2 Gb ultra 160	2 discos SCSI 18.2 Gb ultra 160
no incluye RAID	RAID disponible	RAID si

Elementos de software

Los elementos de software que se eligieron para los proyectos de desarrollo en web en este diplomado fueron: linux con el servidor web apache, PHP y MySQL. Se eligieron



estos elementos de software por sus cualidades y por estar disponibles para su uso de manera gratuita, las cualidades de cada uno de ellos son:

LINUX. Posee todas las características de los sistemas comerciales usados en los entornos de trabajos más exigentes, debido a su eficacia actualmente lo hacen dominar el mercado de los grandes servidores. Algunas de sus principales características son:

- Multiproceso. Permite la ejecución de varias aplicaciones simultáneamente.
- Multiusuario. Distintos usuarios pueden acceder a los recursos del sistema simultáneamente aunque se trate de una instalación en una sola maquina.
- Multiplataforma. Funciona con la mayoría de las plataformas del mercado: Intel 386/486/Pentium, Motorola 680, Sun Sparc,...
- Shell Programables que lo convierten en el sistema más flexible que existe.
- Soporte para cualquier cantidad y tipo de dispositivos directamente en el núcleo.
- Soporte para la mayoría de sistema de ficheros.

APACHE. Es un servidor de paginas (http), el mas utilizado en el mundo por ser poderoso y flexible. Entre las características principales de Apache se encuentran:

- Es un servidor web potente, flexible y ajustado al http/1.1.
- Es altamente configurable y extensible.
- Puede ser ajustado a través de la definición de modulo empleando su propio API (Application Programming Interface).
- Provee todo su código fuente de forma libre y se distribuye bajo una licencia no restrictiva.
- Se ejecuta en diversas plataformas operativas tales como: Windows 9x/NT, Macintosh, Novell NetWare, OS/2, Linux y la mayoría de los Unix existentes.

PHP. Es uno de los lenguajes de lado servidor más extendidos en el Web. PHP nos permite adjuntar sus pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz, sin tener que generar programas creados íntegramente en un lenguaje distinto al HTML.



Entre los competidores principales de PHP se puede citar a Perl, Microsoft Active Server Pages(ASP), Java Server Pages (JSP) Y Allaire ColdFusion.

En comparación con estos productos, PHP cuenta con muchas ventajas, entre las que se encuentran las siguientes:

- Alto rendimiento. PHP es muy eficiente. Mediante el uso de un único servidor, puede servir millones de accesos al día. Los indicadores comparativos de rendimiento muestran que PHP supera ampliamente a sus competidores en esta faceta.
- Interfaces para una gran cantidad de sistemas de bases de datos diferentes. PHP dispone de una conexión propia a todos los sistemas de base de datos. Además de MySQL, puede conectarse directamente a las bases de datos PostgreSQL, Msq, Oracle, filePro, Hyperwave, Informix, Internase, Sybase, entre otras. El uso de ODBC permite establecer una conexión a cualquier base de datos que suministre un controlador ODBC, Entre ellas, se incluyen los productos de Microsoft.
- Bibliotecas incorporadas para muchas tareas web habituales. Como se ha diseñado para uso en la web, PHP incorpora una gran cantidad de funciones integradas para realizar útiles tareas relacionadas con la web. Puede generar imágenes GIF al instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con cookies y generar documentos PDF, todo en unas pocas líneas de código.
- Bajo coste. PHP es gratuito. Se puede descargar la última versión de <http://www.php.net>
- Facilidad de aprendizaje y uso. La sintaxis de PHP se basa en otros lenguajes de programación, principalmente en C y Perl.
- Portabilidad. PHP está disponible para una gran cantidad de sistemas operativos diferentes. Se puede escribir código en todos los sistemas operativos gratuitos del tipo Unix, como Linux y FreeBSD, versiones comerciales de Unix, como Solares, IRIX o en las diferentes versiones de Microsoft Windows. Su código funciona sin necesidad de aplicar ninguna modificación a los diferentes sistemas que ejecuten PHP.



- Acceso a código abierto. Dispone de acceso al código fuente de PHP. A diferencia de los productos comerciales y de código cerrado, si desea modificar algo o agregar un elemento al programa, puede hacerlo con total libertad.

MySQL. Es un sistema para la administración de base de datos relacional (RDBMS) rápido y sólido. El servidor de MySQL controla el acceso a los datos para garantizar el uso simultáneo de varios usuarios, para proporcionar acceso a dichos datos y para asegurarse de que sólo obtienen acceso a ello los usuarios con autorización. Por lo que, MySQL es un servidor multiusuario y de subprocesamiento múltiple. Entre sus competidores principales se puede citar a PostgreSQL, Microsoft SQL Server y Oracle, cuenta con muchas ventajas, entre las que se encuentran las siguientes:

- Alto rendimiento. MySQL es muy rápido, según indicadores comparativos de sus desarrolladores en el sitio web mysql.com.
- Bajo costo. Esta disponible de manera gratuita, bajo una licencia de código abierto.
- Facilidad de configuración y aprendizaje. Las bases de datos más modernas utilizan SQL. Si se ha utilizado otros RDBMS, no debería tener problemas para adaptarse a este. MySQL resulta además más sencillo de configurar que otros productos similares
- Portabilidad. MySQL se puede utilizar en una gran cantidad de sistemas Unix diferentes así como bajo Microsoft Windows.
- Accesibilidad a código fuente. Como en el caso de PHP, puede obtener y modificar el código fuente de MySQL.

En la figura 1 se presenta la arquitectura básica para una aplicación Web y el lugar que ocupa cada elemento de software mencionado en esta arquitectura.

Web Broser. El navegador es concerniente al cliente

Servidor Web. La mayoría de las aplicaciones que trabajan en la Web toman un lugar sobre el servidor Web. Una aplicación específica, llamada servidor Web será responsable de la comunicación con el navegador.

Middleware. PHP pertenece a la clase de lenguajes conocidos como middlewre, estos lenguajes trabajan estrechamente con el servidor Web para interpretar la solicitud hecha desde la World Wide Web, procesan esta solicitud interactúan con otros programas sobre el servidor para completar las solicitud, e indican al servidor Web exactamente cual servicio va para el navegador del cliente.

Base de datos relacional. El sistema manejador de bases de datos relacional provee una gran forma de almacenar y acceder a información compleja.

Sistema operativo. Sin duda alguna, el factor más importante para un buen desempeño en el rendimiento, es el Sistema Operativo. Esto determina la posible potencia de la base de datos.

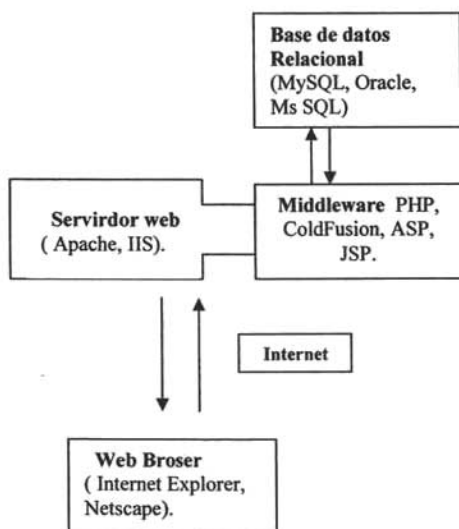


Figura 1 Arquitectura básica para una aplicación Web.

Arquitectura de tres capas

Como se menciona, la arquitectura básica para una aplicación Web se basa en una arquitectura de tres capas. Se puede decir que todas las aplicaciones tienen la misma arquitectura básica y se pueden subdividir en tres partes:

- Capa de presentación (capa del cliente).
- Capa de negocios (capa intermedia).
- Capa de datos (capa de servidor).

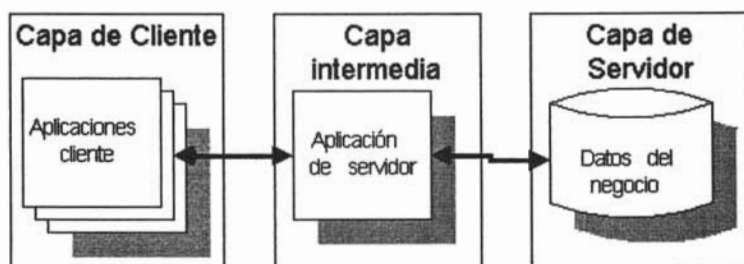


Figura 2 Arquitectura de tres capas

Cada una de estas partes o capas se especializa en una función en particular, teniendo comunicación entre sí, a continuación se describe cada una de las capas.

Capa de presentación. Los servicios de presentación proporcionan la interfaz necesaria para presentar información y reunir datos. Normalmente la interfaz de usuario reside en un programa ejecutable localizado en la estación de trabajo del usuario final. La capa de servicios de presentación es responsable de:

- Obtener información del usuario.
- Enviar información del usuario a los servicios de negocios para su procesamiento.
- Recibir los resultados del procesamiento de los servicios de negocios.
- Presentar estos resultados al usuario.

Capa de negocios. Los servicios de negocios son el puente entre un usuario y los servicios de datos. Responden a peticiones del usuario (u otros servicios de negocios)



para ejecutar una tarea de este tipo. Cumplen con esto aplicando procedimientos formales y reglas de negocio a los datos relevantes. Una tarea de negocios es una operación definida por los requerimientos de la aplicación, como introducir una orden de compra o imprimir una lisita de clientes. Las reglas de negocios son políticas que controlan el flujo de las tareas. El nivel de servicios de negocios es responsable de:

- Recibir la entrada del nivel de presentación.
- Interactuar con los servicios de datos para ejecutar las operaciones de negocios para los que la aplicación fue diseñada.
- Enviar el resultado procesado al nivel de presentación.

Capa de datos. Se encarga de la gestión del sistema de base de datos, de modo que toda petición, tanto de lectura como de escritura, a una base de datos a de pasar por ella, estableciendo las reglas de acceso. El nivel de servicios de datos es responsable de:

- Almacenar datos.
- Recuperar los datos.
- Mantener los datos.
- Mantener la integridad de los datos.

Este modelo de diseño de aplicaciones permite (de manera especial para aplicaciones web) que las aplicaciones pueden correr en ambientes pequeños o grandes, modificar el sistema en periodos de tiempo reducido, modificar o rediseñar una capa produciendo un impacto mínimo en la estructura general del sistema, independencia de la aplicación con respecto a la tecnología o producto que permita el acceso a la información. Como se aprecia este tipo de arquitectura nos permite desarrollar, modificar, actualizar de forma más rápida e eficiente.

A continuación se describe como interactúan los elementos de software en una arquitectura de 3 capas:

1. Un usuario de navegador Web envía una petición HTTP solicitando una página Web dada. Por ejemplo, el usuario podría solicitar todos los libros escritos por



Isaac Asimov que tenga una librería en internet utilizando un formulario HTML. Los resultados de la búsqueda se almacenan en results.php.

2. El servidor Web (Apache) recibe la petición de results.php, recupera el archivo y lo pasa al motor de PHP para su procesamiento.
3. El motor de PHP comienza a analizar la secuencia de comandos. Dentro de la secuencia de comandos hay un comando que establece la conexión a la base de datos y ejecuta una consulta (realiza la búsqueda de libros). PHP abre una conexión al servidor MySQL y remite la consulta pertinente.
4. El servidor MySQL recibe la consulta de la base de datos y la procesa. A continuación, envía los resultados (una lista de libros) al motor de PHP.
5. El motor de PHP termina de ejecutar la secuencia de comandos, lo que suele implicar la aplicación de formato a los resultados en HTML. Seguidamente, devuelve el código HTML resultante al servidor Web Apache.
6. El servidor Web Apache devuelve el código HTML al navegador donde el usuario puede ver la lista de los libros solicitados.

1.2 Modulo II: Análisis y diseño de sistemas orientado al web

El objetivo de este modulo fue obtener los conocimientos y habilidades para analizar, diseñar y modelar Sistemas de Información así como la importancia de documentar dichos sistemas. El análisis de sistema es la etapa en la que se realiza una investigación acerca del problema que se pretende resolver y cómo puede ser resuelto. En esta fase de desarrollo, la comunicación entre analistas, programadores y usuarios es de gran importancia, para que el sistema cumpla con las expectativas planteadas.

Ciclo de vida de un proyecto

El objetivo del ciclo de vida de un proyecto es definir las actividades a ejecutar en un proyecto. El ciclo de vida estructurado consiste de las siguientes etapas:



Estudio: es la etapa de estudio de viabilidad o estudio inicial, su principal objetivo es el estudio e identificación de las necesidades actuales en el ambiente del usuario (a través de encuestas), y proponer escenarios viables.

Análisis: conforme a las alternativas generadas por el estudio, en esta etapa se modelan las necesidades del usuario a través de diagramas de flujo de datos (DFD), diagramas de entidad relación (ER), diagramas de transición de estados y demás herramientas (UML), dando como resultado las especificaciones estructuradas.

Diseño: en esta etapa se diseña el sistema, determinando los módulos componentes del sistema, de acuerdo a una jerarquía apropiada, a los procesadores (hardware) y a la función. En esta activada se desarrolla algo conocido como modelo de implantación del usuario. Se concentra básicamente en las especificaciones de frontera humano máquina (la parte que alguien realizara a mano y las partes que se implantaran en la máquina) y de la interfaz sobre la máquina (el diseño de pantallas y diálogos en línea entre el usuario y la máquina).

La implantación: esta etapa incluye la codificación e integración de los módulos con técnicas de programación estructurada.

Generación de pruebas de aceptación: consiste en preparar un conjunto de casos para efectuar las pruebas del sistema.

Garantía de calidad: en esta etapa se efectúa la prueba final de aceptación del sistema.

La descripción del procedimiento: el resultado de esta actividad es un manual para el usuario.

Conversión de la base de datos: está actividad sólo se realiza cuando existen sistemas funcionando.

La instalación: existen varias estrategias de instalación gradual, distribuida, completa. Un aspecto muy importante de esta actividad es la capacitación.



Hoy en día, es necesario contar con un plan bien analizado, un cliente tiene que comprender que es lo que hará un equipo de desarrolladores, además tiene que ser capaz de señalar cambios si no se han captado claramente sus necesidades (o si cambia de opinión durante el proceso). A su vez, el desarrollo es un esfuerzo orientado a equipos, por lo que cada uno de sus miembros tiene que saber que lugar toma su trabajo en la solución final (así como saber cuál es la solución general).

La clave esta en organizar el proceso de diseño de tal forma que los analistas, clientes, desarrolladores y otras personas involucradas en el desarrollo del sistema lo comprendan y convengan con él, UML proporciona tal organización.

UML (Lenguaje Unificado de Modelado): Es una herramienta que tiene la función de servir de enlace entre quien tiene la idea y el desarrollador, es decir, ayuda a capturar la idea del sistema de una manera convencional y fácil de comprender para comunicarla posteriormente a quien este involucrado en un proceso de desarrollo. El UML esta compuesto por diversos elementos gráficos que se combinan para conformar diagramas. La Finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. El modelo UML de un sistema es similar a un modelo a escala de un edificio junto con la interpretación del artista del edificio. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice como implementar dicho sistema. A continuación se listan los diagramas más comunes de UML:

- Diagrama de clases.
- Diagrama de objetos.
- Diagrama de casos de uso.
- Diagrama de estados.
- Diagrama de secuencias.
- Diagrama de actividades.
- Diagrama de colaboraciones.
- Diagrama de componentes.
- Diagrama de distribución.

Los diagramas UML permiten examinar un sistema desde distintos puntos de vista. Es importante hacer notar que en un modelo UML no es necesario que aparezcan todos los diagramas. De hecho, la mayoría de los modelos UML contienen un subconjunto de los diagramas listados anteriormente.

En seguida se describe en que consisten los diagramas de caso de uso y los diagramas de secuencias vistos en el diplomado:

Caso de uso: es una descripción de las acciones de un sistema desde el punto de vista del usuario. Para los desarrolladores del sistema, está es una herramienta valiosa, ya que es una técnica de aciertos y errores para obtener los requerimientos del sistema desde el punto de vista del usuario. Esto es importante, si la finalidad es crear un sistema que pueda ser utilizado por la gente (no sólo por expertos en computación). Un caso de uso es muy poderoso, pero lo es más cuando se visualiza por medio de UML, los elementos para representar un caso de uso son:

Representación: una elipse representa un caso de uso y una figura agregada representa un actor. El nombre del actor aparece justo debajo de él y el nombre del caso de uso aparece ya sea dentro de la elipse o justo debajo de ella. Una línea asociativa conecta a un actor con el caso de uso, y representa la comunicación entre el actor y el caso de uso, en la figura 3 se muestra la representación un diagrama de caso de uso.

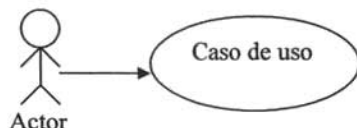


Figura 3 Diagrama de caso de uso

Un caso de uso debe ser simple, inteligible, claro y conciso, generalmente hay pocos actores asociados a cada caso de uso. Las preguntas clave son:



1. ¿Cuáles son las tareas del actor?
2. ¿Que información crea, guarda, modifica, destruye o lee el actor?
3. ¿Debe el actor notificar al sistema los cambios externos?
4. ¿Debe el sistema informar al actor de los cambios internos?

Diagrama secuencias: UML permite expandir el campo de visión y muestra la forma en que un objeto interactúa con otros. En este campo de visión expandido, se incluye una importante dimensión: el tiempo. La idea primordial es que las iteraciones entre los objetos se realizan en una secuencia establecida y que la secuencia se toma su tiempo en ir del principio al fin.

Representación: el diagrama de secuencias consta de objetos que se representan con un rectángulo con nombre (subrayado), mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical.

Objetos: los objetos se colocan cerca de la parte superior del diagrama de izquierda a derecha y se acomodan de manera que simplifique el diagrama. La extensión que está debajo (en forma descendente) de cada objeto será una línea discontinua conocida como la línea de vida de un objeto. Junto con ésta se encuentra un pequeño rectángulo conocido como activación, el cual representa la ejecución de una operación que realiza el objeto.

Mensaje: un mensaje que va de un objeto a otro pasa de la línea de vida de un objeto a otro. Un objeto puede enviarse un mensaje a sí mismo (es decir desde su línea de vida hacia su propia línea de vida).

Tiempo: el diagrama representa al tiempo en dirección vertical. El tiempo se inicia en la parte superior y avanza hacia la parte inferior. Un mensaje que este en la parte superior ocurrirá antes que uno que este más cerca de la parte inferior.

En la figura 4 se muestra el conjunto básico de símbolos del diagrama de secuencias.

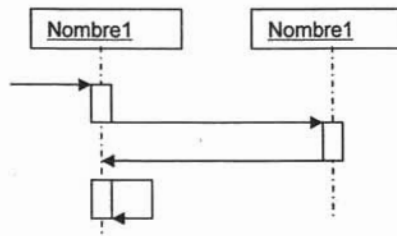


Figura 4 Diagrama de secuencias

A continuación se describe lo que se debe analizar, diseñar, modelar y documentar de un sistema:

Lo que se debe analizar.

- Análisis de hardware y software. Lo que se debe analizar, es con que software y hardware se va a desarrollar el sistema y analizar con que software y hardware se va implementar el sistema que son dos cosas diferentes.
- Análisis de factibilidad. Determina si es factible automatizar un sistema y de ser así, sugerir escenarios aceptables.
- Estimación de tiempos y costos. Se tiene que elaborar estimaciones (muy exactas) del costo y el tiempo necesario para tener un sistema nuevo y los beneficios que derivaran de ello.
- Análisis de costo beneficio. El propósito de esta actividad es calcular todos los costos anticipados asociados con el sistema: no el costo de construirlo, sino también el costo de instalarlo, operarlo y mantenerlo, además de los costos extras.

Lo que se debe diseñar.

Lo que debemos diseñar es la base de datos, ya que su buen diseño es fundamental para las etapas del desarrollo, implementación y utilización del sistema. Se debe comenzar por identificar las entidades involucradas en el sistema; estas entidades tienen características o atributos que las describen. Lo anterior no es más que la identificación de las tablas y sus campos. Y siempre se deberá cuidar la integridad referencial y evitar la redundancia. Relacionado con el diseño de base de datos está la



elaboración de un diccionario de datos. El cual describe claramente el diseño físico de la base, es decir, indica el nombre de cada tabla y de cada campo que la constituye, así como tipo de dato que se almacena y las condiciones que se deben cumplir.

Lo que se debe modelar.

Debemos modelar los requerimientos del usuario, para llevar a cabo esta tarea se hace uso de herramientas de modelado, con el fin de concentrarse en las propiedades importantes del sistema, discutir cambios y correcciones de los requerimientos del usuario y verificar que el analista, comprenda correctamente el ambiente del usuario y que lo documente para que los diseñadores del sistema y los programadores puedan construir el sistema.

Lo que se debe documentar.

Como resultado de todas las actividades anteriores se debe haber generado toda la documentación. Todos se deben considerar parte del sistema y así serán guardados, copias de algunos, o todos, se entregaran a diferentes participantes del proyecto.

1.3 Modulo III: Interfase de usuario

El objetivo del modulo fue analizar los requerimientos de interacción entre el usuario y el sistema.

Mediante la interfase, el usuario interactúa con el sistema o con un sitio web, por lo que es muy importante diseñar una interfase entendible y agradable, para que el usuario se sienta cómodo frente a la interfase, de lo contrario el usuario se resistirá a utilizar la aplicación o la pagina web. Los aspectos fundamentales que se deben procurar al diseñar la interfase son:

Usabilidad: como se trabaja para seres humanos, que quieren realizar una tarea de una forma sencilla y eficaz y en este caso particular, la deben realizar frente a una computadora en un entorno grafico (interfase de usuario) en la web. La usabilidad ayuda a que esta tarea se realice de una forma sencilla.



Diseño amigable: toda interfaz debe ser amigable, pero más en sistemas Web, ya que el usuario la visualiza como un elemento de Internet, aunque la funcionalidad del sistema lo puede hacer olvidar lo vistoso de las paginas comerciales, siendo fundamentales el diseño y formato.

Congruencia entre los procesos: el sistema debe estar estrictamente apegado a los procesos establecidos en el análisis, siendo fundamental su funcionalidad y organización, una interfaz amigable no sirve de nada si los procesos no son los necesarios ni adecuados.

Distribución lógica: los elementos de la interfaz como formularios, botones, listas, tablas, celdas, imágenes, mensajes, etc., deben tener una distribución lógica para lograr la funcionalidad y entendimiento.

Navegación lógica: el orden y secuencia en que se navegue entre los elementos de la interfaz y sus pantallas deben ser adecuados y lógicos según los procesos y perfiles de usuario.

Retroalimentación: se refiere a la información o respuesta que da el sistema a través de mensajes, a las acciones o procesos que realiza el usuario.

Consistencia: la uniformidad de la distribución, navegación, diseño, entradas y salidas, uso de colores, formato de mensajes, etc. de todas las pantallas debe conservarse siempre, pues el usuario se adaptara rápidamente al uso del sistema con un comportamiento homogéneo de éste.

Implementación de ayuda: en ocasiones es necesaria la implementación de módulos de ayuda, esto se decide a través del análisis de los perfiles de usuario.

Longitud de la página: debiendo tener un orden y secuencia adecuado en base a los procesos y distribuidos de manera lógica, cuidando que la longitud de las páginas no exceda el tamaño de las pantallas, considerando 800X600 como la resolución estándar.



Una vez diseñada la interfaz y teniendo en cuenta los aspectos mencionados anteriormente, se debe implementar la interfaz. Para ello se utiliza HTML como columna vertebral, pues el dúo conformado por HTTP y HTML es el más adecuado.

HTML (HyperText Markup Language): es una aplicación SGML1.8 (Standard Generalized Markup Language) con forme al Estándar Internacional ISO 8879, y es ampliamente considerado como el lenguaje estándar para la publicación en la Web. El HTML es un lenguaje compuesto por un conjunto predeterminado de elementos con los que se componen etiquetas para dar formato de hipertexto a documentos publicados en la web. Las etiquetas se componen con los caracteres menor que (<) y mayor que (>) que contienen el nombre y los atributos de un elemento HTML. Básicamente se trata de un conjunto de etiquetas (tags) que sirven para definir la forma en la que presenta el texto y otros elementos de la página.

Los documentos escritos en HTML están estructurados en dos partes diferenciadas: la **HEAD** (cabecera) y el **BODY** (cuerpo). Por ejemplo:

```
<HTML>
<HEAD>
<TITLE> Documento de prueba </TITLE>
</HEAD>
<!-- Esto es un comentario-->
<BODY>
<H1> Esto es una "demo" de documento HTML </H1>
Esto es el más sencillo de los documentos HTML.
</BODY>
</HTML>
```

El elemento **<HTML> </HTML>** no es obligatorio. Solo sirve como identificación del tipo de contenido del fichero para ciertos programas que hacen cambios masivos en muchas páginas a la vez.



El encabezado, delimitado por **<HEAD>** y **</HEAD>** es donde colocaran las etiquetas de índole informativo como por ejemplo el título de nuestra página.

Para escribir comentarios en la página (que sólo se ven en el texto fuente, pero no en el visualizador) se utilizará el elemento **<!-- -->**

El cuerpo, flanqueado por las etiquetas **<BODY>** y **</BODY>**, que será donde colocara el texto e imágenes delimitados a su vez por otras etiquetas como las que se ha visto.

Los elementos mas utilizados en una interfaz HTML son:

- Listas.
- Tablas. Permite la representación de datos por filas y columnas, en forma tabular. La definición es muy flexible indicando sólo los elementos que forman cada fila y columna, calculándose de forma automática el tamaño que deben tener las celdas. En una tabla podemos introducir todo tipo de elementos del lenguaje HTML como imágenes, enlaces, texto, listas, formularios, etc.
- Imágenes. En un documento HTML se puede incluir cualquier imagen en formatos gráficos: **GIF**, **JPEG** o **XBM**. El formato más extendido y práctico es el **GIF**, todos los navegadores gráficos de la Web soportan este formato. Este formato, así mismo, utiliza algoritmos de compresión que hacen que sus ficheros sean cortos en tamaño y apropiados para su transmisión por la red.
- Hipervínculos. Es la utilidad básica del hipertexto, permite indicar zonas de texto o imágenes que si son seleccionados por el lector del documento nos traslada a otros documentos HTML u otras zonas del documento actual.
- Formas. Los formularios son plantillas que permiten la creación de documentos HTML con peticiones de datos. La principal utilidad de los formularios es la posibilidad de crear cuestionarios, encuestas, páginas de comentarios o cualquier documento en el que se desee una interacción por parte del usuario.

El lenguaje HTML está limitado a la hora de aplicarle forma a un documento. Esto es así porque fue concebido para otros usos (científicos sobretodo), distinto a los actuales, mucho más amplios. Para resolver esta limitación se utilizan las hojas de estilo CSS.



CSS: Las hojas de estilo representan un gran avance que permite aumentar la capacidad de mejorar la apariencia de las páginas, nos permiten definir el formato de una Web, la apariencia visual: colores, imágenes de fondo, fuentes, márgenes y sangría etc. Aunque esto también se puede hacer solo con HTML, las CSS permiten lograr una presentación muy uniforme, ya que basta definir un estilo una sola vez para que éste se aplique a todos los elementos del mismo tipo que existan en una página.

Para definir estilos en secciones reducidas de una página se utiliza la etiqueta ``. Con su atributo `style` se indica en sintaxis CSS las características de estilos. Ejemplo, para poner un párrafo en el que determinadas palabras se visualicen en color verde, se tendría que escribir la siguiente línea.

```
<p>  
Esto es un párrafo en varias palabras  
<SPAN style="color:green">en color verde</SPAN>.  
resulta muy fácil.  
</p>
```

Se puede hacer que toda una etiqueta muestre un estilo determinado. Por ejemplo, se puede definir un párrafo entero en color rojo y otro en color azul. Para ello se utiliza el atributo `style`, que es admitido por todas las etiquetas del HTML (siempre y cuando se disponga de un navegador compatible con CSS).

```
<p style="color:#990000">  
Esto es un párrafo de color rojo.  
</p>  
<p style="color:#000099">  
Esto es un párrafo de color azul.</p>
```

Con la etiqueta `<DIV>` se definen secciones de una página y aplicarle estilos con el atributo `style`, es decir, se definen estilos de una vez a todo un bloque de la página.



```
<div style="color:#000099; font-weight:bold">
<h3>Estas etiquetas van en <i>azul y negrita</i></h3>
<p>
Seguimos dentro del DIV, luego permanecen los estilos
</p>
</div>
```

También se puede definir en la cabecera del documento, estilos para que sean aplicados a toda la página. Es una manera muy cómoda de darle forma al documento y muy potente, ya que estos estilos serán seguidos en toda la página y así se ahorrara muchas etiquetas HTML que apliquen forma al documento. Además, si se desea cambiar los estilos de la página se hará de una sola vez. En el siguiente ejemplo se utiliza la etiqueta <STYLE> colocada en la cabecera de la página para definir los distintos estilos del documento. A grandes rasgos, entre <STYLE> y </STYLE>, se coloca el nombre de la etiqueta que queremos definir los estilos y entre llaves {} colocamos en sintaxis CSS las características de estilos.

```
<html>
<head>
<title>Ejemplo de estilos para toda una página</title>
<STYLE type="text/css">
<!--
H1 {text-decoration: underline; text-align:center}
P {font-Family:arial,verdana; color: white; background-color: black}
BODY {color:black;background-color: #cccccc; text-indent:1cm}
// -->
</STYLE>
</head>
<body>
<h1>Página con estilos</h1>
Bienvenidos...
<p>Siento ser tan hortera, pero esto es un ejemplo sin importancia</p>
</body>
</html>
```

Mapa del sitio web. El mapa de navegación de un sitio web, es una representación grafica del sitio, esta representación permite ver la estructura y el contenido del sitio de un solo vistazo. En el mapa del sitio deben quedar representadas la totalidad de las páginas.

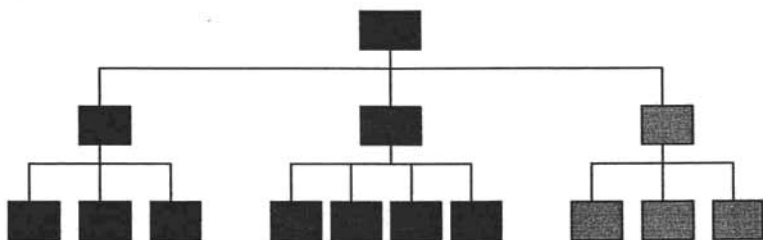


Figura 5 Mapa del sitio

El mapa es de gran utilidad para el diseño del sitio, ya que permite representar todas las páginas que se deben diseñar, así mismo permite ver el flujo que se debe seguir para llegar a una página en particular.

1.4 Modulo IV: Capa de datos

El objetivo de este modulo fue entender y utilizar el lenguaje SQL, para diseñar consultas y manipular información de una base de datos, explicando el acceso universal a los datos.

Las Bases de Datos son el núcleo del Sistema de Información de cualquier empresa u organización. La administración y gestión de las mismas constituye, por tanto, un trabajo muy importante dentro de cualquier organización. El ciclo de vida de una base de datos puede descomponerse en 3 etapas:

1. **Concepción:** El esquema conceptual se construye utilizando la información que se encuentra en la especificación de los requisitos de usuario. El diseño conceptual es completamente independiente de los aspectos de implementación, como puede ser el SGBD que se vaya a usar, los programas de aplicación, los lenguajes de programación, el hardware disponible o cualquier otra consideración física.



2. Creación de la BD vacía: la segunda fase consiste en traducir este esquema en instrucciones comprensibles para el SGBD, como resultado se obtiene la estructura de la base de datos desprovista del cualquier tipo de información.
3. Explotación: en esta fase es donde los registros serán manipulados con la ayuda de los lenguajes de programación. Es ahora cuando los usuarios pueden consultar los datos y ponerlos a punto durante el resto de la vida de la base de datos.

Con el fin de obtener el mejor desempeño y flexibilidad de la aplicación Web, es recomendable seguir ciertas guías para el diseño de la base de datos. Bajo estos lineamientos, es posible desarrollar un sistema de base de datos escalables y que puedan manejar desde la más sencilla consulta hasta la más compleja, los cuales son:

- Al crear una base de datos, lo que hacemos es modelar elementos y relaciones del mundo real, almacenar la información sobre dichos objetos y relaciones. Por regla general, cada clase de objeto del mundo real que se modele necesitara su propia tabla.
- Evitar el almacenamiento redundante, ya que esto nos genera problemas como el malgastar el espacio innecesariamente, otro problema son las anomalías que pueden surgir al guardar datos, es decir, situaciones en las que al cambiar los datos se producen inconsistencia. Por lo que la integridad de los datos quedaría alterada y no se sabría que datos son correctos.
- Uso de valores de columna únicos, es decir cada atributo de una fila debe almacenar un solo valor.
- Selección de claves lógicas, se debe asegurar de que las claves seleccionadas son exclusivas.
- Reflexionar sobre las preguntas que desea formular a la base de datos, es decir, asegurarse que la base de datos contiene todos los datos necesarios y que se han establecido los vínculos adecuados entre las tablas para responder a las preguntas deseadas.
- Evitar diseños con varios atributos nulos, no conviene tener muchos valores nulos en una tabla, ya que esta practica contribuye a desperdiciar espacio de



almacenamiento, ocasionando problemas al calcular totales o al aplicar funciones a columnas numéricas.

SQL

El lenguaje de consulta estructurado SQL, es un lenguaje de base de datos de consulta normalizado, utilizado por los diferentes motores de bases de datos para realizar determinadas operaciones, sobre los datos o sobre la estructura de los mismos. Lo cual asegura que cualquier sentencia escrita en ANSI, será interpretada por cualquier motor de base de datos.

El lenguaje SQL esta compuesto por comandos, cláusulas, operadores y funciones de agregado, estos elementos se combinan para crear, actualizar y manipular las bases de datos. La estructura básica de SQL es la siguiente:

- Base de datos: esta compuesta de un conjunto de tablas del sistema, creadas implícitamente por él y un conjunto de tablas y vistas creadas por el usuario.
- Tablas: el universo de un modelo esta descrito con la ayuda de tablas, cada una representa a una entidad o a una asociación entre entidades. Las tablas están compuestas de columnas de filas llamadas convencionalmente campos y registros. Una columna representa un atributo de la entidad y para describirla es necesario especificar un nombre y un tipo de datos, una particularidad de las columnas es que pueden permitir o no valores nulos.
- Fila (tupla): es una combinación de los diferentes atributos del objeto registro.
- Vistas: es una tabla virtual definida sobre las tablas bases descritas por el usuario, su objeto es permitir a los usuarios manipular un subconjunto de datos.
- Usuarios: uno de los objetivos del sistema de gestión de base de datos es afianzar la seguridad de los datos, para hacer esto, el lenguaje de programación genera todos los usuarios y sus permisos de acceso para acceder a una base de datos, cada usuario debe disponer de su autorización correspondiente y es el propietario de la base de datos el que debe dar y retirar los permisos de acceso.



- **Índices:** se utilizan para aumentar el rendimiento del sistema y la integridad de los datos. A cada tabla que se crea se asocia automáticamente una tabla índice que contiene la posición del registro según la columna especificada como índice.

MySQL

El gestor de base de datos relacional que se usa en esta parte del desarrollo es MySQL, para crear y administrar la base de datos. Enseguida se muestra la estructura creada al instalar el administrador de base de datos MySQL ya que es importante saber donde se guardan las bases de datos que son creadas y donde localizar archivos que nos pueden ser de utilidad.

En el directorio `/bench` encontraremos ejemplos de script y SQL., en el directorio `/share` estan los mensajes de error del servidor para los distintos idiomas, los directorios `/include` y `/lib` contienen los ficheros `*.h` y las librerías necesarias, en `/bin` estan los ficheros ejecutables y `/data` encontraremos como subdirectorio cada una de las bases de datos que hayamos creado.

Para cada base de datos que se cree, MySQL crea un directorio con el nombre que se le asigne a la base de dato. Dentro de este directorio, por cada tabla que se define MySQL va a crear tres archivos: `mitabla.isd`, `mitabla.ism` y `mitabla.frm`.

El archivo con extensión `isd` contiene los datos de la tabla, el `ism` contiene información de las claves y otros datos que MySQL utiliza para buscar datos en el fichero `isd`. Y el archivo `frm` contiene la estructura de la propia tabla.

Todo el sistema de permisos MySQL se guarda en una base de datos llamada `mysql`, la cual se compone de cinco tablas: `host`, `user`, `db`, `tables_priv` y `columns_priv`.

La tabla `user` contiene información sobre los usuarios, desde que máquinas pueden acceder a nuestro servidor MySQL, su clave y sus diferentes permisos. La tabla `host` informa sobre que máquinas podrán acceder a nuestro sistema, así como la base de datos a que tendrán acceso y sus diferentes permisos. Finalmente, las tablas `db`,



tables_priv y columns_priv proveen un control individual de las bases de datos, tablas y columnas (campos).

Las tareas que se pueden realizar con MySQL son las siguientes:

- Crear una base de datos.
- Eliminar una base de datos.
- Crear una tabla.
- Eliminar una tabla.
- Cargar datos a una tabla.
- Eliminar datos de una tabla.
- Recuperar información de una tabla.
- Seleccionar registros.
- Ordenar registros.
- Trabajar con valores nulos.
- Conteo de filas.

Para realizar estas tareas cuenta con comandos, cláusulas, operadores lógicos, operadores de comparación y funciones de agregado. Un comando normalmente consiste de una sentencia SQL seguida por un punto y coma. Cuando se emite un comando, Mysql lo manda al servidor para que lo ejecute, nos muestra los resultados y regresa al prompt indicando que esta listo para recibir más consultas.

Una sentencia se puede escribir en una sola línea, pero es posible escribir más de una sentencia por línea, siempre y cuando estén separadas por punto y coma. También es posible escribir una sentencia en varias líneas si así se requiere; MySQL determinara en donde finaliza la sentencia cuando encuentre el punto y coma, no cuando encuentre el fin de línea.

Se puede usar de manera interactiva ejecutar comandos y obtener los resultados. Sin embargo, es posible usar MySQL en batch. Para hacer esto tenemos que poner comandos que deseemos ejecutar dentro de un archivo, y entonces decirle a MySQL que lea los comandos de dicho archivo.



Si se usa MySQL de esta manera, se esta creando un script, posteriormente se esta ejecutando dicho script.

Algunas razones para usar un script:

- Si se ejecuta un cierto número de consultas frecuentemente (cada día o semana), al hacer un script nos evitamos tener que volver a teclear cada una de las consultas.
- Se pueden generar nuevas consultas que sean similares a las existentes al copiar y editar algún script.
- Al escribir consultas de varias líneas, los script ayudan para que no se tengan que escribir todas las líneas nuevamente si se comete un error.
- Se pueden distribuir script's a otras personas para que puedan ejecutar nuestros comandos y sentencias
- Cabe mencionar que el formato de salida es diferente (más conciso) cuando se ejecuta Mysql en modo batch, que cuando se usa de manera interactiva.

Acceso universal a datos

El acceso universales a los datos significa que, teóricamente por lo menos, los usuarios finales sin tener en cuenta la herramienta de acceso a la información o ubicación, deberían ser capaces de acceder a cualquier o todos los datos en la empresa u organización que es necesaria para ellos, para hacer su trabajo.

El nivel de acceso a los datos entonces es responsable de la interfase entre las herramientas de acceso a la información y las bases de datos operacionales. En algunos casos, esto es todo lo que un usuario final necesita.

Acceder a la información y los recursos de una empresa u organización desde cualquier lugar, en cualquier momento y desde cualquier dispositivo, ha sido el paradigma de numerosos proyectos y tecnologías innovadores (Java, Wap, GPRS, Bluetooth, etc.) desarrollados en el mundo de la informática y las telecomunicaciones. Finalmente, dicho paradigma está transformándose en realidad de la mano de las soluciones de Internet móvil.



Uno de los adelantos claves de los últimos años ha sido el desarrollo de una serie de "filtros" de acceso a datos, tales como EDA/SQL para acceder a casi todo los Sistemas de Gestión de Base de Datos (Data Base Management Systems - DBMSs) y sistemas de archivos de datos, relacionales o no. Estos filtros permiten a las herramientas de acceso a la información, acceder también a la data almacenada en sistemas de gestión de base de datos que tienen veinte años de antigüedad.

Seguridad

La evaluación de este punto es uno de los más importantes en la interconexión de la Web con Bases de datos. Por su propia naturaleza, la Web es un lugar peligroso, este medio está diseñado para permitir que numerosos usuarios soliciten servicios desde sus equipos. La mayor parte de estas peticiones solicitan páginas Web perfectamente legítimas, pero existe la posibilidad de que la gente realice otros tipos de conexiones.

Al analizar el tema de seguridad, lo primero que se tiene que valorar es la importancia de los datos que se protegerán. Puede resultar tentador que siempre se debería aplicarse el nivel más alto de seguridad a todos los sitios Web, pero la protección tiene un costo. Antes de decidir el esfuerzo o los gastos que se deben invertir en el sistema de seguridad, es necesario determinar el valor de la información. El valor de la información en el equipo de un usuario que utilice su computadora, el de una empresa, el de un banco y el de una organización militar es diferente.

Los riesgos que se ciernen sobre un sitio Web varían en función del tipo de sitio pero se pueden citar las siguientes:

- Exposición de los datos. Para reducir el riesgo de la exposición de datos, debe limitar los métodos de acceso a la información y los usuarios que disponen de dicho acceso, lo cual implica desarrollar las labores de diseño teniendo presente la seguridad, configurando el servidor y el software de manera correcta, programar con atención, realizar pruebas exhaustivas, eliminar los servicios innecesarios del servidor Web y exigir autenticación. Para proteger la información confidencial, se pueden cifrar los datos antes de enviarlos a través de una red y descifrados en el otro extremo. Los servidores Web suelen utilizar



el sistema SSL (Secure Socket Layer) para realizar esta tarea mientras los datos entre los servidores Web y los navegadores.

- Pérdida o destrucción de los datos. Se pueden adoptar varias medidas para reducir la probabilidad de perder datos. Se debe proteger los servidores contra ataques informáticos. Reducir al mínimo los empleados con acceso al equipo. Comprar discos duros de calidad Utilizar discos RAID para permitir el uso de varios discos duros. Independientemente del causante de las pérdidas de datos sólo hay una forma de protección real, las copias de seguridad.
- Modificación de los datos. La protección de los archivos almacenados en un servidor frente a su modificación exige el uso de la funciones de permisos de acceso a archivos que incorpore su Sistema Operativo y protege el sistema de acceso no autorizado. La aplicación de permisos de archivos permite autorizar el uso del sistema, pero no modificarlo los archivos del sistema ni de otros usuarios.

Se debe buscar un equilibrio entre la seguridad, la usabilidad, el costo y rendimiento, si se aumenta la seguridad de un servicio se puede reducir su capacidad de uso, es decir, limitar lo que puede hacer una gente o pedir que se identifique. El incremento de la seguridad también puede reducir el nivel de rendimiento de los equipos. La ejecución de software para lograr que el sistema resulte más seguro (como sistemas de encriptación, detención de intrusos y operaciones de registro de usuarios) absorbe recursos. Se necesita mucho más potencial de procesamiento para suministra una sesión cifrada, como una conexión SSL a un sitio Web, que una sesión normal. La pérdida de rendimiento se puede compensar con la adquisición de equipos o hardware más rápido que este especialmente diseñado para operaciones de encriptación.

El rendimiento, la usabilidad, el costo y la seguridad se pueden considerar como objetivos contrapuestos. Se debe examinar los pros y los contras de cada opción y tomar una decisión que logre un compromiso en función del valor de la información, del presupuesto, de la cantidad de visitas esperadas y de los obstáculos que se consideran que aceptarán los usuarios legítimos.



La mayor parte de los navegadores y servidores Web incorporan mecanismos de autenticación. Los servidores Web se pueden configurar para solicitar un nombre de usuario y su contraseña, para permitir el acceso a determinados archivos y directorios del servidor. Apache permite proteger de manera sencilla una parte del sitio o su totalidad. PHP y MySQL incorporan muchas otras formas de lograr el mismo objetivo, MySQL resulta más rápido que la autenticación integrada. PHP suministra una autenticación más flexible o presenta la solicitud de forma más atractiva.

1.5 Modulo V: Regla de negocios

Los objetivos de este modulo fueron: identificar las necesidades del proceso que se desea incorporar al Web, reconocer la necesidad de mantener y controlar un sitio apoyado por una base de datos.

Toda aplicación trata de reflejar parte del funcionamiento del mundo real, para automatizar tareas que de otro modo serían llevadas a cabo de modo menos eficiente, o bien no podrían realizarse. Para ello, es necesario que cada aplicación refleje las restricciones que existen en el negocio dado, de modo que nunca sea posible llevar a cabo acciones no válidas.

A las reglas que debe seguir la aplicación para garantizar lo anterior se las llama reglas de negocio o business rules. Estas reglas de negocios se clasifican en varios grupos:

El primer grupo de reglas engloba a todas aquellas que controlan la información básica para cada atributo es válida:

- No permitir precios negativos de artículos.
- El sexo de una persona solo puede ser Masculino o Femenino.
- Una fecha de compra válida.

Otro grupo de reglas incluye todas aquellas que controlan las relaciones entre los datos.



- Todo pedido debe ser realizado por un cliente registrado y conocido.
- Una vez registrado un pedido por un cliente, no es posible eliminar este cliente del sistema, a menos que se eliminen todos aquellos pedidos que ha realizado.

Es frecuente que a partir de cierta información se pueda derivar otra, es decir, especifican y controlan la obtención de información que se calcula a partir de la existente:

- El total de un pedido se puede calcular a partir de las distintas partidas que lo componen.
- El total de cada partida se puede calcular a partir de las unidades vendidas y el precio por unidad.

Otro grupo de reglas de negocio esta compuesto por las reglas de restricción, estas restringen los datos que el sistema puede contener, más allá de las restricciones básicas que sobre los mismos existen.

- Regla del modelo de datos: un saldo debe ser un número.
- Regla de restricción: para clientes de chequera, el saldo debe ser mayor o igual que 1500.

El último grupo son aquellas reglas determinan y limitan como fluye la información a través del sistema.

- El cliente solicita una consulta de saldo en la cuenta 3547.
- Esta solicitud genera eventos para uno o más reportadores.
- Solo participan aquellos eventos que estén contemplados para dar respuesta al flujo de trabajo solicitado para dar respuesta adecuada.

Herramientas para la implementación

Para cada una de las tareas de implementación será imprescindible contar con una herramienta adecuada. Al ser un trabajo multidisciplinar, se requieren muchas herramientas distintas, aunque unos pocos programas bastan para posibilitar la implementación de un sitio Web.



Teniendo en cuenta que la mayoría de los servidores Web utilizan Linux y los visitantes Windows, es importante poder probar el sitio con ambos sistemas. Por lo que se debe tener en la PC de desarrollo instalado Linux y acceder a él desde de otro equipo, con Windows, para las pruebas del sitio.

Como se comento al principio el servidor es Apache, ya que se encuentra en la mayoría de los sitios y es gratuito. Para el desarrollo de páginas Web en diferentes lenguajes (PHP, HTML y CSS), es recomendable contar con un editor web: UltraEdit-32 o CrimSom Editor (estos programas cuentan con funciones muy interesantes, como poder escribir por columnas) y aparte soporta conexión FTP con nuestro servidor Web.

Es recomendable disponer de Dreamweaver MX, que nos permite una creación de código mucho más rápida y facilita la integración con las hojas de estilo (CSS). El equipo debe de contar con los principales navegadores para poder probar la portabilidad del sitio, Internet Explorer y Netscape.

La Base de Datos será MySQL, la más extendida y soportada por todos los servidores Web, gratuita y de fácil uso, debe ir acompañada de alguna herramienta de administración como phpMyAdmin.

Enlace con la base de datos

El acceso a los datos debe ser a través de servicios abstractos, es decir, consiste en definir las rutinas de acceso primarias a la base de datos:

- Conectar / desconectar de la base de datos.
- Ejecutar consultas.
- Ejecutar sentencias (insertar, actualizar y eliminar).

No se deben construir soluciones que sean dependientes de alguna versión de manejadores de base de datos.



Catálogo electrónico

Cuando se trabaja en la implementación de servicios a través de un sitio Web, regularmente se requiere de un catalogo electrónico como parte de la infraestructura. Este catalogo contiene el listado completo de productos y servicios que se ofrecen en el sitio. Las operaciones que se pueden realizar sobre el catalogo, son solo en dos modalidades:

- 1- como usuario del sitio Web, con acceso libre y publico.
- 2- como administrador de contenidos del mismo, con acceso restringido.

Normalmente un usuario tiene interés en ver todo el catalogo de forma que pueda tomar una decisión sobre lo que se oferta, mientras que el administrador de contenidos solo accede a una sección controlada donde podrá dar mantenimiento al mismo. Este proceso del mantenimiento del catalogo es parte de las actividades logísticas para el sitio.

1.6 Modulo VI: Implementación

El objetivo de este modulo fue Implementar los productos y servicios de software que fueron conceptualizados.

Elaboración del plan de trabajo

El desarrollar un plan de trabajo permite identificar las tareas a realizar y el tiempo estimado que se llevara realizar cada una de ellas, así como el orden de ejecución de las mismas, es decir, que tareas se deben realizar primero y cuales se pueden realizar simultáneamente, de tal modo que se puedan evitar periodos ociosos e innecesarios. El desarrollo de este plan también nos permitirá observar los avances del proyecto, en virtud de que se pueden comparar de forma conveniente la planificación original con la del desarrollo real. Para representar este plan de trabajo se puede utilizar las gráficas de Gantt.



Elaboración del plan de pruebas

Todo Sistema o aplicación, debe ser probado mediante una ejecución controlada antes de ser entregado al cliente, estas ejecuciones o ensayos de funcionamiento, posteriores a la terminación del código del software, se denominan habitualmente pruebas. Ya que no se pueden probar todas las posibilidades de funcionamiento del software, la idea fundamental para el diseño de casos de prueba consiste en elegir algunas de ellas que, por sus características, se consideren representativas del resto.

El objetivo del plan de pruebas es señalar el enfoque, los recursos y el esquema de actividades de prueba, así como los elementos a probar, las características, las actividades de prueba, el personal responsable y los riesgos asociados.

En general, la estrategia de pruebas suelen seguir las siguientes etapas:

1. La prueba de módulo (prueba de unidad) centra sus actividad en ejercitar la lógica del módulo y los distintos aspectos de la especificación de las funciones que debe realizar el modulo.
2. La prueba de integración debe tener en cuenta los mecanismos de agrupación de módulos fijados en la estructura del programa, así como, en general, las interfaces entre componentes de la arquitectura de software.
3. La prueba de validación (o funcional) debe comprobar si existen desajustes entre el software y los requisitos fijados para su funcionamiento.
4. La prueba del sistema debe centrar sus comprobaciones en el cumplimiento de los objetivos indicados para el sistema.
5. La prueba de aceptación sirve para que el usuario pueda verificar si el producto final se ajusta a los requisitos por él fijados o, en último caso en función de lo que indique el contrato.

Estándares de programación

Un aspecto muy importante para escribir código es definir una terminología específica para ello, esto es lo que se denomina estándares de programación, que no es más que



el usar y seguir ciertas reglas de notación y nomenclatura durante la fase de implementación (codificación) de una aplicación.

Un estándar de programación no solo busca definir la nomenclatura de las variables, objetos, métodos y funciones, sino también tiene que ver con el orden y legibilidad del código escrito, siguiendo esta idea, se puede definir tres partes principales dentro de un estándar de programación:

- Convenciones de nomenclatura: nombrar variables, funciones, métodos etc.
- Convenciones de legibilidad de código: como indentar el código, etc.
- Convenciones de documentación: como establecer comentarios, archivos de ayuda, etc.

Teniendo en cuenta lo expuesto en todos los módulos del Diplomado Desarrollo de Sistemas en Web, en los siguientes capítulos se analizará y diseñará el Sistema de Administración de Eventos del Laboratorio L3.

CAPITULO II

ANÁLISIS Y DISEÑO

Para que un sistema orientado a la Web, así como cualquier otro Sistema de Información cumpla con las expectativas para lo cual será creado, se debe ante todo, hacer un análisis de los requerimientos del sistema en cuestión, para poder realizar el diseño correspondiente. Estas dos actividades son las primeras que se deben desarrollar de acuerdo al ciclo de vida estructurado de un proyecto.

UML (lenguaje de modelado unificado) es la herramienta que se utilizara modelar y documentar el sistema de una manera grafica. UML esta compuesto por diversos elementos gráficos, de los cuales solos se utilizaran los casos de uso (que permiten obtener los requerimientos del sistema desde el punto de vista del usuario) y diagramas de secuencias (estos diagrama permiten ver en que forma un objeto interactúan con otros).

El sistema a analizar y diseñar es el sistema de Administración de Eventos del laboratorio L3 de la División de Ciencias Físico Matemáticas y de las Ingenierías de la FES Aragón.

2.1 Planteamiento del problema

Con frecuencia el laboratorio L3 organiza eventos de índole académica (cursos, talleres, exposiciones), de modo que se requiere difundir estos eventos y realizar el registro de los alumnos a estos, así como también, difundir temas de interés actual. Hasta la fecha esta el registro de los asistentes se realiza "a mano" y a veces hasta el momento en que se está desarrollando el evento, lo cual complica la emisión de la papelería de soporte (gafetes, recibos, listas de asistentes, etc.), dificulta la gestión de los espacios reservados (identificación de sobre cupo o subocupación) y complica la entrega de constancias.

Necesidad de implementación de un nuevo esquema

Con base en lo anterior, se hace necesario implantar un nuevo esquema, el cual permita resolver los problemas que se generan al organizar estos eventos actualmente (de forma manual), para ello se requiere contar con una aplicación adaptable de distintos eventos, que permita a los interesados vía Web ver una lista de todos los eventos disponibles, ver con detalles las características de cada uno de los eventos y además pueda registrarse a el evento de su elección, generándose en forma automatizada la papelería correspondiente, además de permitir elaborar listas de asistentes y reportes estadísticos. También debe permitir a los organizadores de los eventos poder administrar todo lo relacionado a los eventos desde Internet.

2.2 Funciones del sistema

Haciendo un análisis más detallado de los requerimientos tenemos que se requiere que se realicen las siguientes tareas:

- Administrar tipos de eventos
- Administrar eventos
- Administrar alumnos
- Administrar profesores
- Administrar lugares (ubicación donde se llevara acabo el evento)
- Enviar mensajes
- Generar reportes
- Generar consultas
- Administrar usuario de sistema
- Iniciar sesión
- Validar usuario.

De acuerdo a las tareas antes mencionadas se desprenden las funciones que deberá realizar el sistema.

Funciones del sistema		
Iniciar sesión	Modificar evento	Registra alumno
Registrar usuario	Eliminar evento	Modificar alumno
Modificar usuario	Consultar evento	Eliminar alumno
Eliminar usuario	Registrar lugar	Consultar alumno
Consultar usuario	Modificar lugar	Inscribir alumno
Validar usuario	Eliminar lugar	Modificar inscripción
Registrar tipo evento	Consultar lugar	Cancelar inscripción
Modificar tipo evento	Registrar profesor	Enviar mensaje
Eliminar tipo evento	Modificar profesor	Generar reportes
Consultar tipo evento	Eliminar profesor	Generar consultas
Registrar evento	Consultar profesor	Salir del sistema

Enseguida se representan los requerimientos del Sistema de Administración Eventos del Laboratorio L3 mediante los diagramas de caso de uso.

2.3 Casos de uso del sistema

Los diagramas de caso de uso del sistema representan como cada uno de los "agentes" humanos o informáticos relacionados con el sistema interactúan con las funciones de este, a estos agentes se les llama actores. En seguida definimos los actores del sistema:



Rol: Su principal tarea será administrar los usuarios de sistema, es decir dar de alta, asignar privilegios, dar de baja los usuarios del sistema, aun que puede realizar cualquier tarea sobre el sistema registrándose como súper usuario.

**Súper usuario**

Rol: Podrá realizar cualquier tarea sobre el sistema a excepción de las que el administrador realiza, pero su tarea principal será Administrar lo tipos de eventos, eventos del sistema y administrar profesores.

**Usuario**

Rol: Solo podrá realizar las tareas que el administrador le autorice, podría realizar las siguientes tareas, administra alumnos, administrar lugares, enviar mensajes (mail), emitir reportes o realizar consultas.

**Alumno**

Rol: Es simplemente consultar los eventos disponibles en la pagina Web y registrarse a un evento.

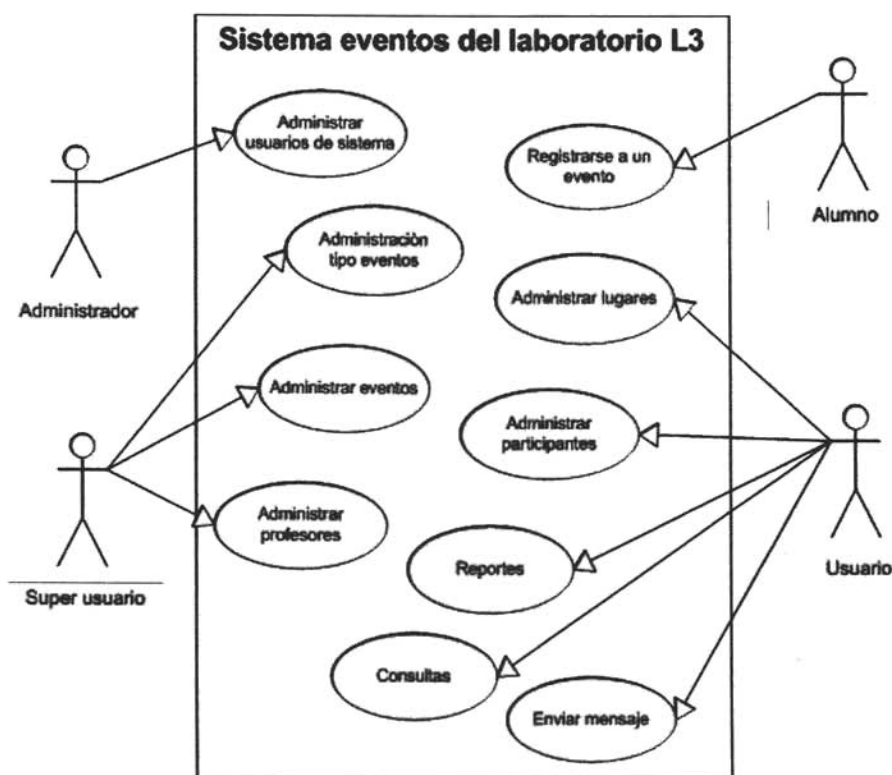


Figura 6. Casos de uso del sistema

En la figura 6 se muestra el diagrama de casos de uso del sistema, en los casos de uso administrar tipo eventos, administrar eventos, administrar usuarios, administrar ponentes, administrar participantes y administrar lugares se convertirán en subcasos de uso.

Subcasos de uso administrar usuarios de sistema.

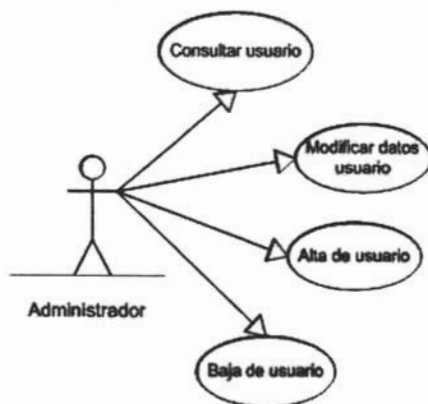


Figura 7 Subcasos de uso Administrar usuarios

Caso de uso:	Alta usuario de sistema
Actores:	Administrador
Propósito:	Dar de alta un usuario en el sistema

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar alta usuario
3. Ingresar datos del usuario
4. Guardar datos
5. Confirmar datos ingresados
6. Si los datos son confirmados se guardan los datos

Caso de uso:	Modificar datos del usuario
Actores:	Administrador
Propósito:	Modificar los datos del usuario

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar modificar usuario
3. Ingresar id del usuario
4. Obtiene datos usuario
5. Modificar datos del usuario
6. Guardar datos
7. Confirmar datos ingresados
8. Si los datos son confirmados se guardan los datos



Caso de uso:	Baja de usuario
Actores:	Administrador
Propósito:	Dar de baja a un usuario del sistema

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar baja usuario
3. Ingresar id del usuario
4. Obtener datos del usuario
5. Confirmar baja
6. Si la eliminación es confirmada se borra el usuario.

Caso de uso:	Consulta usuario
Actores:	Administrador
Propósito:	Consultar los datos de un usuario de sistema

Curso de eventos:

1. Ingresar contraseña
2. Selecciona consultar usuario
3. Ingresa id usuario
4. Si id de usuario es correcto obtiene datos del usuario

Subcasos de uso administrar tipo eventos.

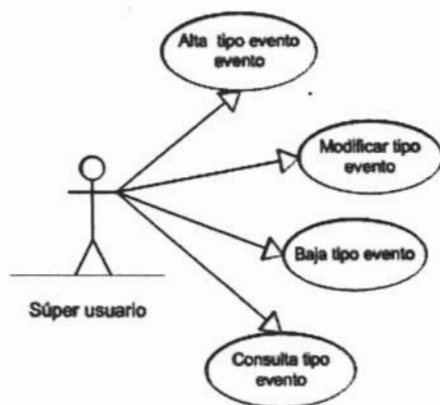


Figura 8 Subcasos de uso Administrar tipo de evento



Caso de uso:	Alta tipo de evento
Actores:	Súper usuario
Propósito:	Dar de alta un tipo de evento

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar Alta tipo evento
3. Ingresar datos del tipo de evento
4. Guardar datos
5. Confirmar datos ingresados
6. Si los datos son confirmados se guardan los datos

Caso de uso:	Modificar tipo de evento
Actores:	Súper usuario
Propósito:	Modificar los datos de un tipo de evento

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar modificar tipo evento
3. Ingresar id tipo evento
4. Obtener datos tipo evento
5. Modificar los datos del tipo evento
6. Guardar datos
7. Confirmar datos ingresados
8. Si los datos son confirmados se guardan los datos

Caso de uso:	Baja tipo evento
Actores:	Súper usuario
Propósito:	Dar de baja un tipo de evento.

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar baja tipo de evento
3. Ingresar id tipo de evento
4. Obtener datos del tipo de evento
5. Confirmar baja
6. Si la eliminación es confirmada se borra el tipo evento

Caso de uso:	Consulta tipo evento
Actores:	Súper usuario
Propósito:	Consultar los datos de un tipo de evento

Curso de eventos:

1. Ingresar su contraseña
2. Seleccionar consultar tipo evento
3. Ingresar id tipo evento
4. Si el id del tipo evento es correcto obtiene los datos correspondientes.

Subcasos de uso administrar eventos.

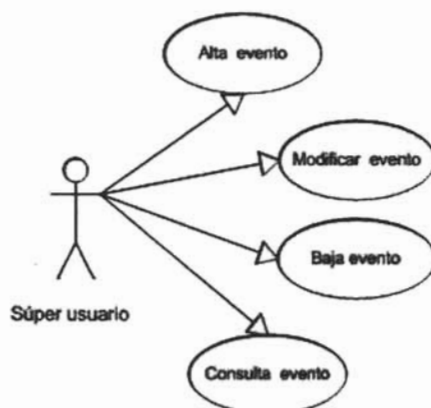


Figura 9 Subcasos de uso Administrar eventos

Caso de uso:	Alta evento
Actores:	Súper usuario
Propósito:	Dar de alta un evento

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar Alta evento
3. Ingresar datos del evento
4. Guardar datos
5. Confirmar datos ingresados
6. Si los datos son confirmados se guardan los datos



Caso de uso:	Modificar datos del evento
Actores:	Súper usuario
Propósito:	Modificar los datos de un evento

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar modificar evento
3. Ingresar id del evento
4. Obtener datos del evento
5. Modificar datos del evento
6. Guardar datos
7. Confirmar datos ingresados
8. Si los datos son confirmados se guardan los datos

Caso de uso:	Baja evento
Actores:	Súper usuario
Propósito:	Dar de baja un evento

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar baja evento
3. Ingresar id del evento
4. Obtener datos del evento
5. Confirmar baja
6. Si la eliminación es confirmada se borra el evento

Caso de uso:	Consultar evento
Actores:	Súper usuario
Propósito:	Consultar los datos de un evento

Curso de eventos:

1. Ingresar su contraseña
2. Seleccionar consultar evento
3. Ingresar id de evento
4. Si el id de evento es correcto obtiene los datos del evento

Subcasos de uso administrar profesores

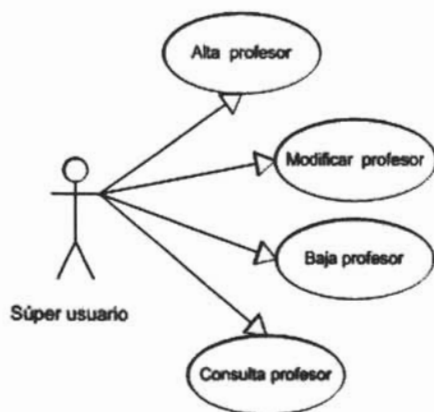


Figura 10 Subcasos de uso Administrar profesores

Caso de uso:	Alta profesor
Actores:	Súper usuario
Propósito:	Consultar los datos de un profesor

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar Alta profesor
3. Ingresar datos del profesor
4. Guardar datos
5. Confirmar datos ingresados
6. Si los datos son confirmados se guardan los datos

Caso de uso:	Modificar profesor
Actores:	Súper usuario
Propósito:	Modificarlos datos del profesor

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar modificar profesor
3. Ingresar id del profesor
4. Obtener datos del profesor
5. Modificar datos del profesor
6. Guardar datos
7. Confirmar datos ingresados
8. Si los datos son confirmados se guardan los datos



Caso de uso:	Baja profesor
Actores:	Súper usuario
Propósito:	Dar de baja a un profesor

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar baja profesor
3. Ingresar id del profesor
4. Obtener datos del profesor
5. Dar de baja al profesor
6. Confirmar baja
7. Si la eliminación es confirmada se borra los datos del profesor

Caso de uso:	Consultar profesor
Actores:	Súper usuario
Propósito:	Permite consultar los datos de un ponente o profesor

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar consulta profesor
3. Ingresar id del profesor
4. Si el id del profesor es correcta obtiene los datos correspondientes

Subcasos de uso administrar alumnos

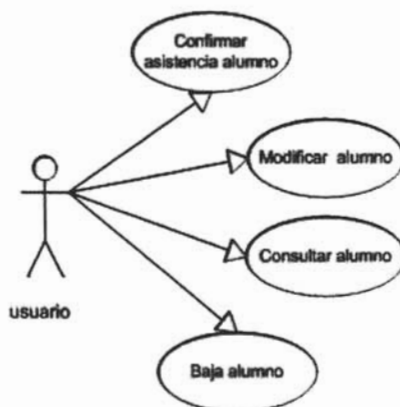


Figura 11 Subcasos de uso Administrar alumnos



Caso de uso:	Confirmar asistencia del alumno
Actores:	Usuario
Propósito:	Permite confirmar la asistencia del participante al evento al que se registro

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar confirmar asistencia
3. Ingresar id del evento
4. ingresar id del alumno
5. Confirmar asistencia del participante

Caso de uso:	Modificar alumno
Actores:	Usuario
Propósito:	Modificar los datos de un alumno

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar modificar datos alumno
3. Ingresar id alumno
4. Obtener datos del alumno
5. Modificar datos del alumno
6. Guardar datos
7. Confirmar datos ingresados
8. Si los datos son confirmados se guardan los datos

Caso de uso:	Baja de alumno
Actores:	Usuario
Propósito:	Dar de baja un participante de la base de datos

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar baja alumno
3. Ingresar id del alumno
4. Obtener datos del asistente
5. Confirmar baja
6. Si la eliminación es confirmada se borra el evento

Caso de uso:	Consultar alumno
Actores:	Usuario
Propósito:	Consultar los datos de un alumno

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar consulta alumno
3. Ingresar id alumno
4. Si el id del alumno es correcto obtiene los datos correspondientes

Subcasos de uso administrar lugares.

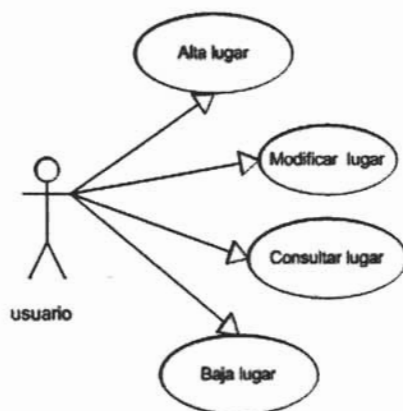


Figura 12 Subcasos de uso Administrar lugar

Caso de uso:	Alta lugar (ubicación donde se realizara un evento)
Actores:	Usuario
Propósito:	Dar de alta un lugar

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar Alta lugar
3. Ingresar datos del lugar
4. Guardar datos
5. Confirmar datos ingresados
6. Si los datos son confirmados se guardan los datos



Caso de uso:	Modificar lugar
Actores:	Usuario
Propósito:	Permite modificarlos datos de un lugar

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar modificar lugar
3. Ingresar id del lugar
4. Obtener datos del lugar
5. Modificar datos del lugar
6. Guardar datos
7. Confirmar datos ingresados
8. Si los datos son confirmados se guardan los datos

Caso de uso:	Baja lugar
Actores:	Usuario
Propósito:	Dar de baja a un lugar

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar baja lugar
3. Ingresar id del lugar
4. Obtener datos del lugar
5. Dar de baja lugar
6. Confirmar baja
7. Si la eliminación es confirmada se borra el lugar

Caso de uso:	Consultar lugar
Actores:	Usuario
Propósito:	Consultar los datos de un lugar

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar consulta lugar
3. Ingresar id lugar
4. Si el id del lugar es correcto obtiene los datos correspondientes

Caso de uso registrarse a un evento



Figura 13 Caso de uso Registrarse a un evento

Caso de uso:	Registrarse a un evento
Actores:	Alumno
Propósito:	Permitir registrarse a un alumno a un evento de su elección

Curso de eventos:

1. Entrar al sitio Web
2. Seleccionar un evento
3. Llenar formulario de registro
4. Envía formulario de registro
5. Recibe confirmación de registro

Caso de uso enviar mensaje.

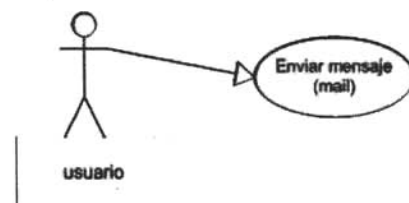


Figura 14 Caso de uso enviar mensaje (mail)

Caso de uso:	Enviar mensaje
Actores:	Usuario
Propósito:	Enviar mensaje (mail) a todos los participantes de un evento

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar enviar mensaje
3. Selecciona evento
4. Escribe el mensaje
5. Envía el mensaje

Caso de uso consultas

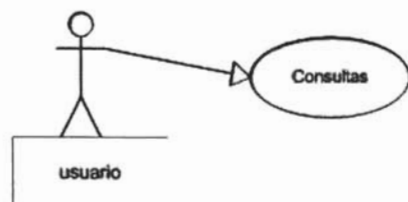


Figura 15 Caso de uso consultas

Caso de uso:	Generar consultas
Actores:	Usuario
Propósito:	Permitir realizar consultas

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar consultas
3. Obtiene opciones de consulta
4. Elige una opción de consulta
5. Obtiene el resultado de la consulta

Caso de uso reportes.

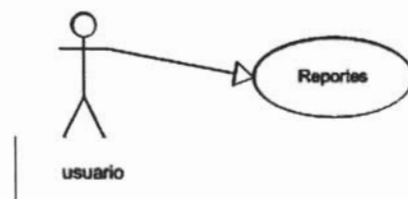


Figura 16 Caso de uso reportes

Caso de uso:	Generar reportes
Actores:	Usuario
Propósito:	Permitir generar reportes

Curso de eventos:

1. Ingresar contraseña
2. Seleccionar reportes
3. Obtiene opciones de reportes
4. Elige una opción de reporte
5. Obtiene el reporte

2.4 Diagrama de Secuencias del sistema

Los diagramas de secuencias conjuntan el trabajo hecho por los casos de uso definidos. Utilizando los diagramas de secuencias se modela la interacción entre los dominios seleccionados que son necesarios para proveer del comportamiento especificado en un caso de uso particular.

Diagrama de secuencia Alta tipo de evento



Figura 17 Diagrama de secuencia Alta tipo de evento

En el diagrama se puede observar que la secuencia inicia cuando se elige crear el tipo de evento, inmediatamente se obtiene un registro vacío del catálogo de tipo evento, se ingresan los datos necesarios del tipo de evento, se guardan los datos en el catálogo tipo evento y finalmente se obtiene un mensaje donde se indica que se ha creado un nuevo tipo de evento.

Diagrama de secuencia Modificar tipo evento

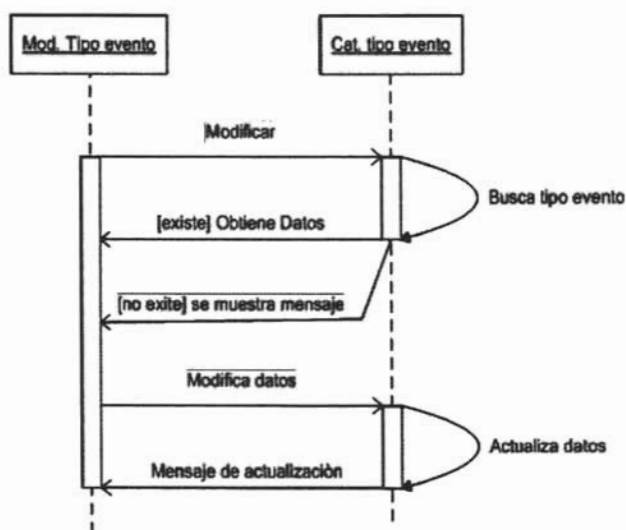


Figura 18 Diagrama de secuencia Modificar tipo de evento

Como se observa en el diagrama la secuencia empieza cuando se elige modificar los datos de un tipo de evento, enseguida se busca el tipo de evento a modificar, si se encuentra el tipo de evento a modificar se muestran los datos del tipo evento, se modifican los datos, se actualizan los datos en el catalogo de tipo de eventos y finalmente se muestra un mensaje de actualización efectuada. En el caso que no exista el tipo de evento a modificar se obtiene un mensaje que lo indica.

Diagrama de secuencia Baja tipo evento

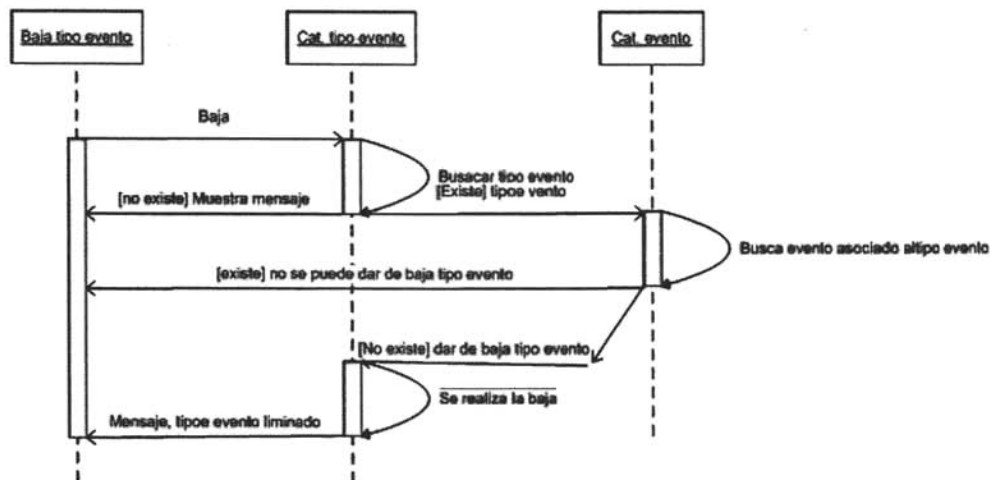


Figura 19 Diagrama de secuencia Baja tipo de evento

La secuencia inicia cuando se elige dar de baja un tipo de evento, se busca el tipo de evento a dar de baja en el catalogo de tipo evento, dependiendo del resultado de la búsqueda se tiene:

1. Si se encuentra el tipo evento en el catalogo, se busca algún evento asociado a este tipo de evento, si hay un evento asociado a este tipo de evento se indica que no se puede dar de baja por que hay un evento asociado. En caso de que no exista un evento asociado se procede a dar de baja el tipo de evento del catalogo y finalmente se obtiene el mensaje tipo de evento eliminado.
2. Si no se encuentra el tipo de evento en el catalogo se muestra un mensaje indicando esta situación.

Diagrama de secuencia Alta evento

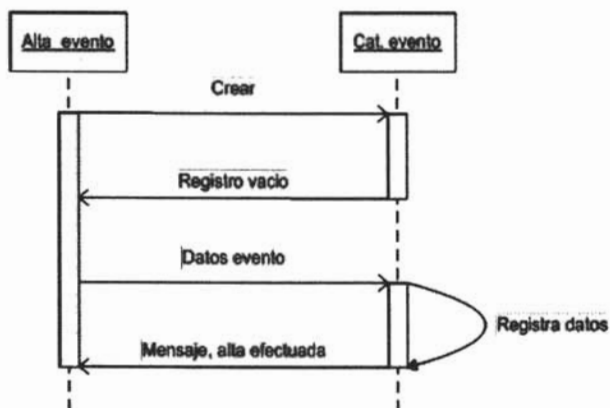


Figura 20 Diagrama de secuencia Alta evento

La secuencia inicia cuando se elije crear el evento, se va al catalogo evento, se obtiene un registro vació del catalogo, se ingresan los datos del evento, se guardan los datos en el catalogo evento y se obtiene un mensaje donde se indica que se a creado un nuevo de evento.

Diagrama de secuencia Modificar evento

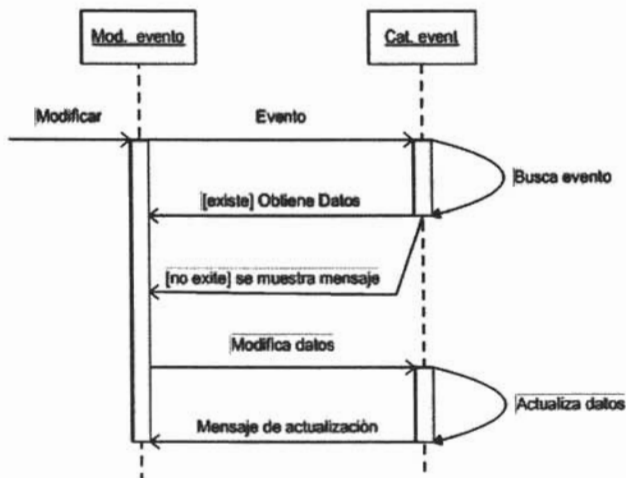


Figura 21 Diagrama de secuencia Modificar evento



La secuencia inicia cuando se desea modificar los datos de un evento, se indica el evento a modificar, se busca el evento a modificar en el catalogo eventos, del resultado de se tienen los siguientes casos:

1. Si no existe el evento a modificar, se indica esta situación.
2. Si existe el evento se obtienen los datos del evento a modificar, se modifican los datos, se actualizan los datos en el catalogo y se notifica la actualización.

Diagrama de secuencia Baja evento

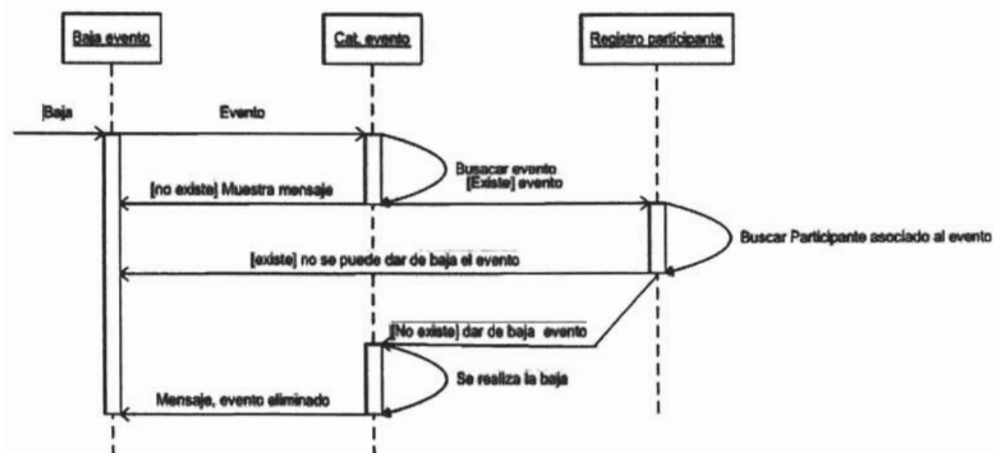


Figura 22 Diagrama de secuencia Baja evento

La secuencia empieza cuando se desea dar de baja un evento, se indica el evento a dar de baja, se busca el evento en el catalogo eventos, dependiendo del resultado se presentan dos situaciones:

1. Si no existe el evento en le catalogo, se indica esta situación.
2. Si existe el evento en el catalogo evento se busca si el evento tiene algún participante asociado en el catalogo registro participantes, si hay algún participante asociado no se puede dar de baja el evento; pero si no existe un participante asociado se realiza la baja del evento y finalmente se indica la baja del evento.

Diagrama de secuencia Alta usuario

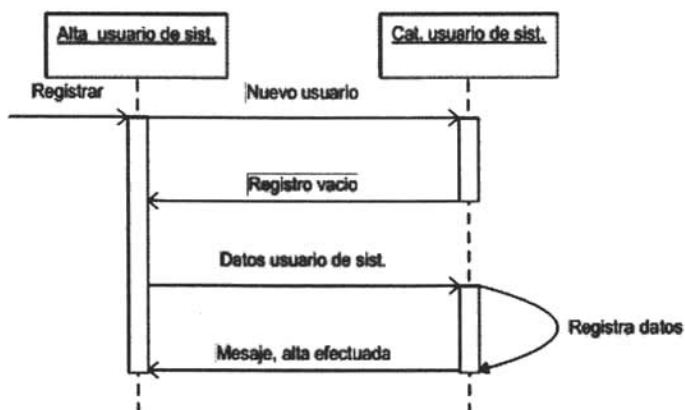


Figura 23 Diagrama de secuencia Alta usuario

La secuencia inicia cuando se quiere registrar un usuario de sistema, se obtiene un registro en blanco, se introducen los datos del usuario, se guardan los datos en el catalogo de usuarios de sistema y se indica que el registro se ha efectuado.

Diagrama de secuencia Modificar usuario

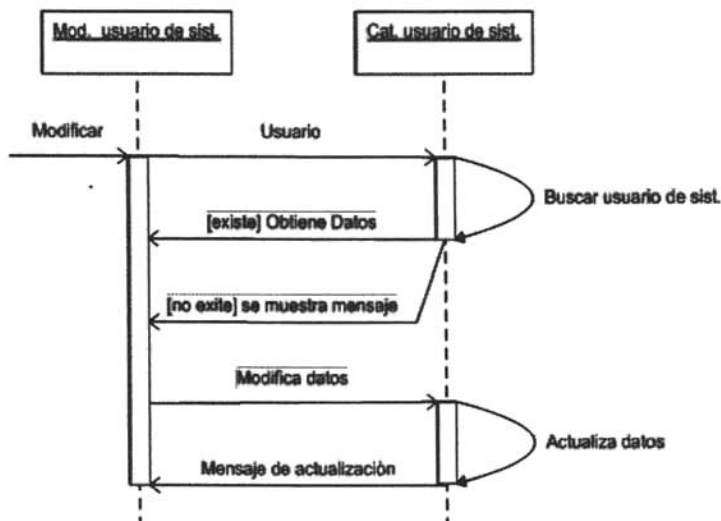


Figura 24 Diagrama de secuencia Modificar usuario



La secuencia inicia cuando se quiere modificar los datos del usuario de sistema, se indica el usuario a modificar, se busca al usuario en el catalogo de usuarios de sistema, dependiendo del resultado se tiene:

1. Si se encuentra el usuario en el catalogo, se obtienen los datos del usuario, se modifican los datos y se actualiza los datos en el catalogo y se indica que la actualización se ha efectuado.
2. Si no existe el usuario a modificar se indica esta situación mostrando un mensaje.

Diagrama de secuencia Baja usuario



Figura 25 Diagrama de secuencia Baja usuario

La secuencia inicia cuando se desea dar de baja un usuario, se indica el usuario a dar de baja, del resultado se desprende dos casos:

1. Si no existe el usuario en el catalogo se indica esta situación.
2. Si el usuario existe se muestran los datos del usuario a dar de baja, se confirma dar de baja, se elimina el registro del catalogo usuario de sistema y se indica que el registro se ha eliminado.

Diagrama de secuencia Alta profesor

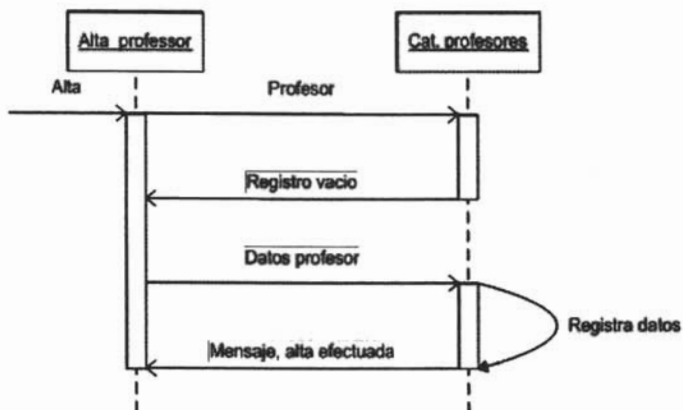


Figura 26 Diagrama de secuencia Alta profesor

La secuencia inicia cuando se requiere dar de alta un profesor, se obtiene un registro en blanco, se introducen los datos del instructor, se guardan los datos en el catalogo instructores y se obtiene indicación de alta efectuada.

Diagrama de secuencia Modificar profesor

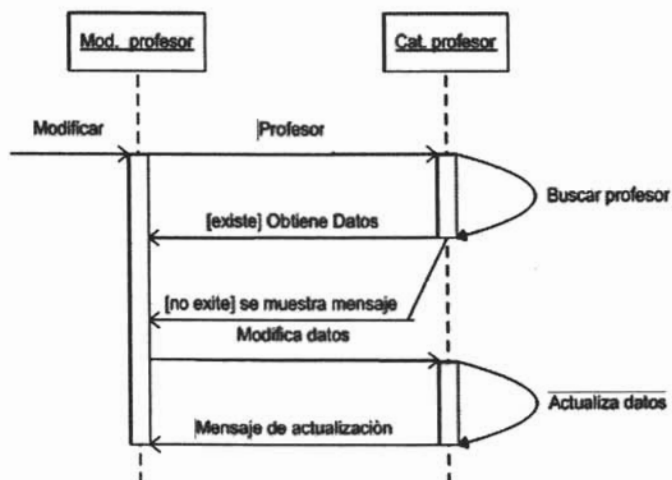


Figura 27 Diagrama de secuencia Modificar profesor

La secuencia inicia cuando se quiere modificar los datos del profesor, se indica el profesor a modificar, se busca al profesor en el catalogo de profesores, dependiendo del resultado se tiene:

1. Si no existe el profesor en el catalogo se indica esta situación.
2. Si se encuentra al profesor en el catalogo, se obtienen los datos del profesor, se modifican los datos y se actualiza los datos en el catalogo y se indica que la actualización se a efectuado.

Diagrama de secuencia Baja profesor



Figura 28 Diagrama de secuencia Baja profesor

La secuencia inicia cuando se desea dar de baja un profesor, se indica el profesor a dar de baja, del resultado se desprende dos casos:

1. Si no existe el profesor en el catalogo se indica esta situación.
2. Si el profesor existe se muestran los datos del profesor a dar de baja, se confirma dar de baja, se elimina el registro del catalogo profesores de sistema y se indica que el registro se ha eliminado.

Diagrama de secuencia Registrarse a un evento

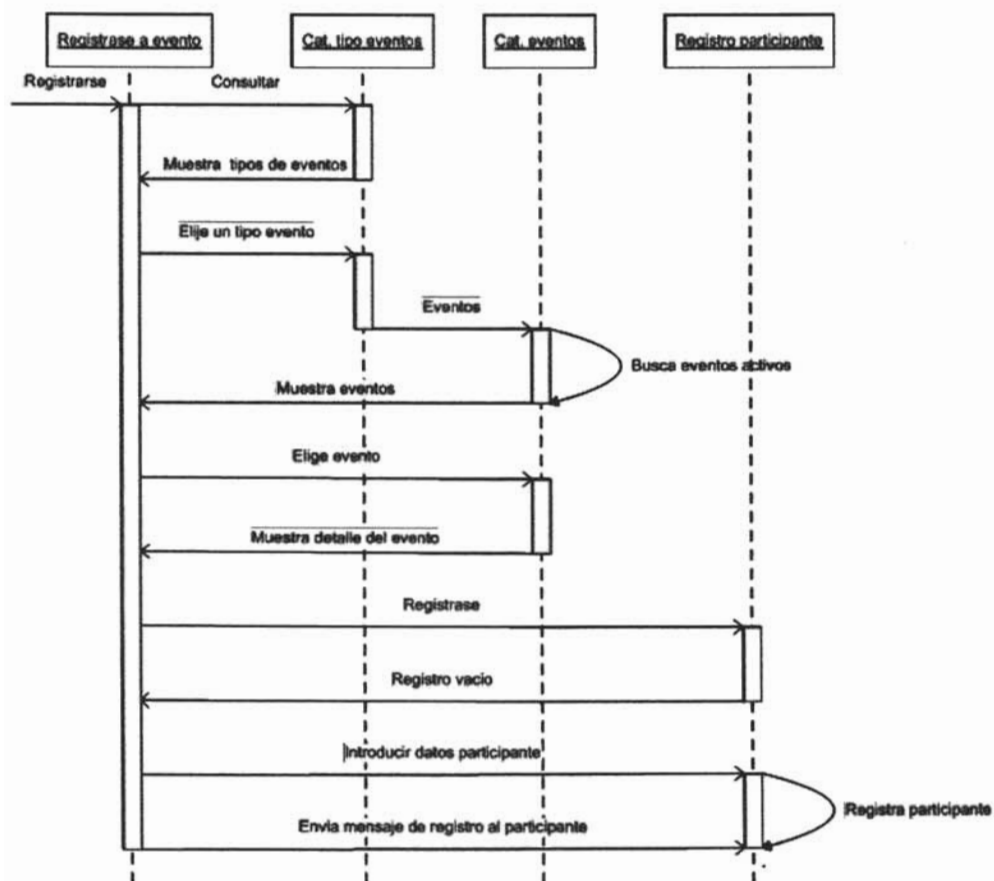


Figura 29 Diagrama de secuencia Registrarse a un evento

La secuencia inicia cuando un participante se quiere registrar a un evento, consulta tipos de eventos disponibles del catalogo, elije un tipo de evento, se buscan los eventos activos de acuerdo al tipo de evento elegido en el catalogo eventos, obtiene una lista de eventos, elije un evento, se muestra la información del evento, elije registrarse al evento, se obtiene registro en blanco, introduce sus datos, se guardan sus datos en el catalogo registro participante y se envía mensaje de registro efectuado.

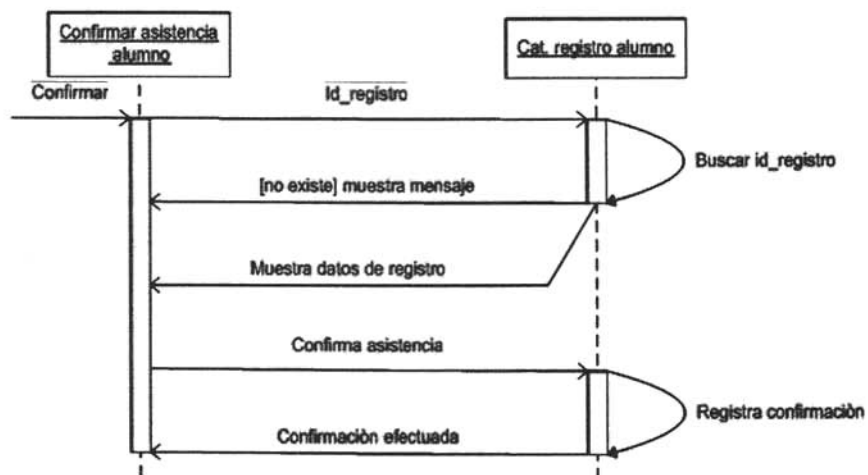
Diagrama de secuencia Confirmar asistencia del alumno a un evento

Figura 30 Diagrama de secuencia Confirmar asistencia del alumno a un evento

La secuencia inicia cuando el alumno confirma su asistencia, se ingresa el id_registro, se buscan en el catalogo registro de alumnos, si no existe se indica esta situación; si existe se confirma la asistencia y se registra en catalogo registro alumnos y se indica que se a efectuado es operación.

Diagrama de secuencia Modificar registro a un evento

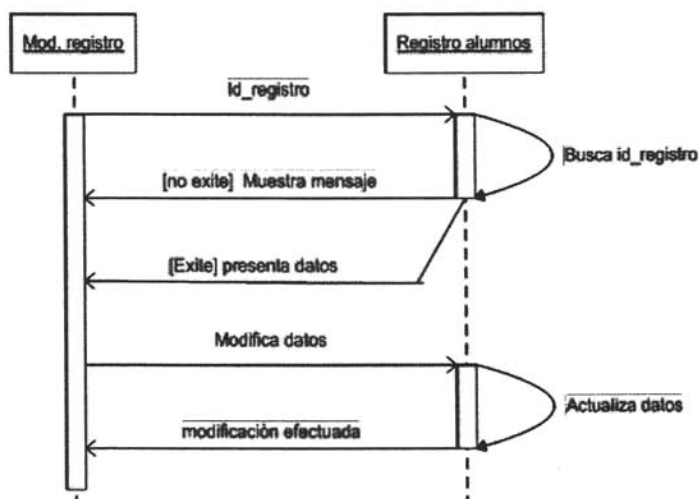


Figura 31 Diagrama de secuencia Modificar registro a un evento

La secuencia inicia cuando se quiere modificar el registro de evento de un alumno, se solicita el id_registro, se busca id_registro en el catalogo registro alumno, si no se encuentra se indica esta situación; pero si existe se modifican los datos del registro, se actualizan los datos en el catalogo registro alumnos y se indica que se han modificado los datos del registro alumno.

Diagrama de secuencia Eliminar registros no confirmados de un evento

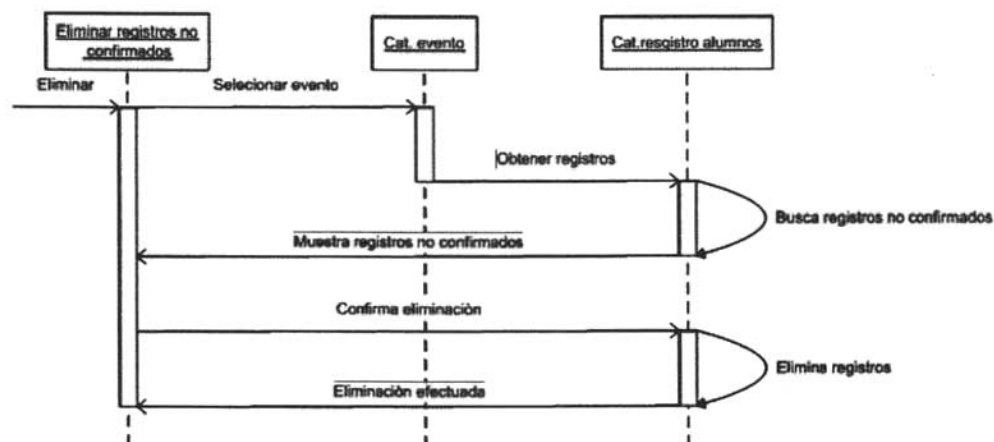


Figura 32 Diagrama de secuencia Eliminar registros no confirmados a un evento

La secuencia inicia cuando se quiere eliminar los registros de alumnos que no confirmaron su asistencia a un evento en particular, se selecciona el evento en cuestión, se buscan los registros en el catalogo registro de alumnos, se muestran los alumnos que no confirmaron sus asistencias, se confirma su eliminación, se eliminan de el catalogo registro de alumnos y se indica que se eliminaron los registros de alumnos sin confirmar.

Diagrama de secuencia Modificar alumno

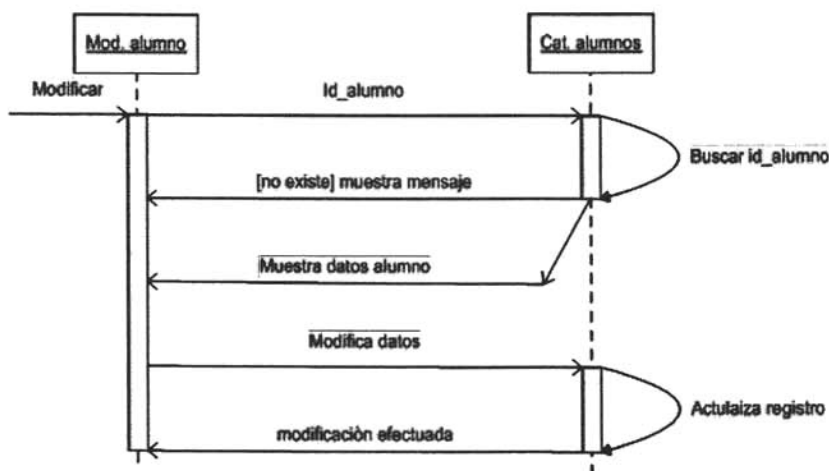


Figura 33 Diagrama de secuencia Modificar participante

La secuencia comienza cuando se quiere modificar los datos del alumno, se solicita el `id_alumno`, se busca en el catalogo alumno, si no existe se indica esta situación; de lo contrario se muestran los datos del alumno, se modifican los datos, se actualizan los datos en el catalogo alumnos y se indica que la actualización se realizó.

En el siguiente capítulo se diseñará la interfase (paginas Web) del usuario, que es una parte muy importante en el diseño de sistemas orientados a la Web.

CAPITULO III

INTERFASE DE USUARIO

El diseño de la interfaz (página Web) de usuario es uno de los aspectos más importante para un sitio Web, ya que será nuestra carta de presentación; del buen diseño de la interfase y de su contenido, depende que el visitante vuelva a visitar el sitio Web.

3.1 Mapa de navegación

Antes de empezar a diseñar las pantallas del sitio del sistema Administración de Eventos del Laboratorio L3, es conveniente hacer un mapa de navegación, donde se muestra el orden de presentación de las pantallas, esto es con la finalidad de no perderse al ir diseñándolas, además permite que varios desarrolladores trabajen al mismo tiempo en el diseño de las mismas.

De acuerdo al análisis del proyecto se requiere de dos mapas, el mapa de navegación del alumno y el mapa de navegación del usuario de sistema. Cabe mencionar que se dividió de esta forma, ya que son dos procesos diferentes, además es conveniente que la página principal del alumno no contenga opciones concernientes al usuario de sistema y viceversa, ya que esto ayuda a la claridad y la seguridad del sitio Web.

El mapa de navegación del alumno, solo representa la navegación y las páginas que han de diseñarse para que el participante pueda consultar los tipos de eventos disponibles, los eventos activos, pueda inscribirse a un evento, pueda consultar si esta inscrito a un evento y cancelar su registro a un evento. Este inicia con la página principal del alumno y de ahí se sigue una ruta de navegación según la tarea a realizar. Por ejemplo para que un participante se pueda inscribir a un curso se tendrá que diseñar las secuencia de pantallas como se muestra en el mapa de la figura 33, página principal (esta página solo se diseñaría una sola vez), página tipo de eventos, página eventos, página detalles del curso, página formulario de registro y una pantalla de notificación de registro.

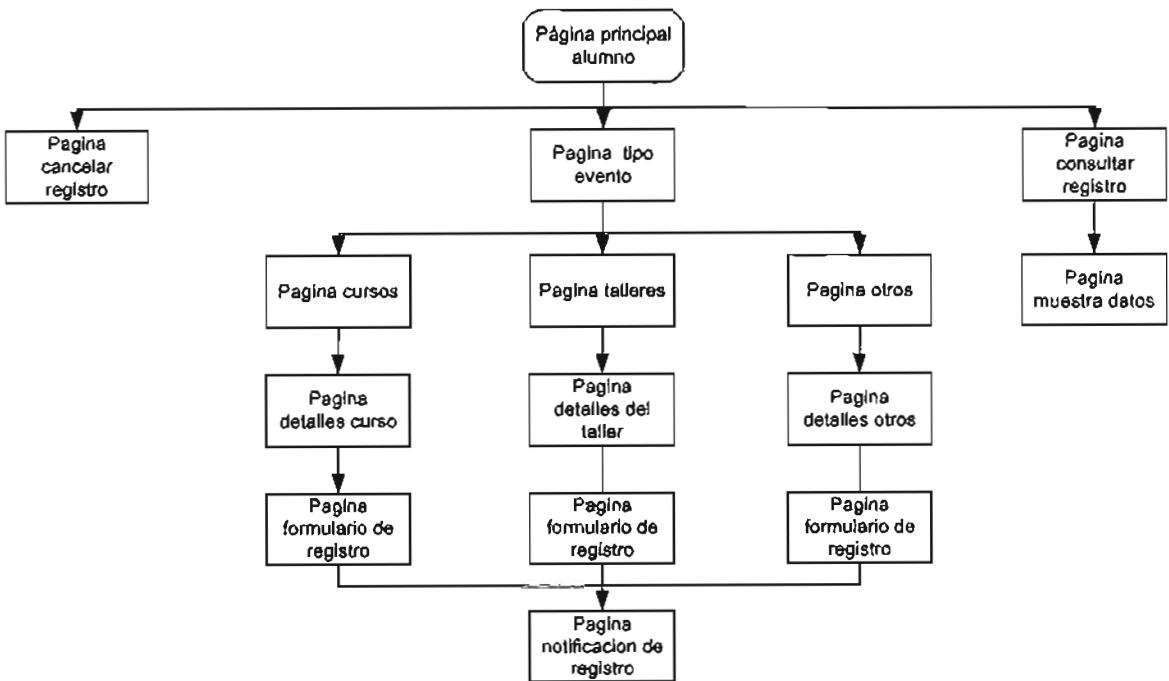


Figura 33 Mapa de navegación alumno

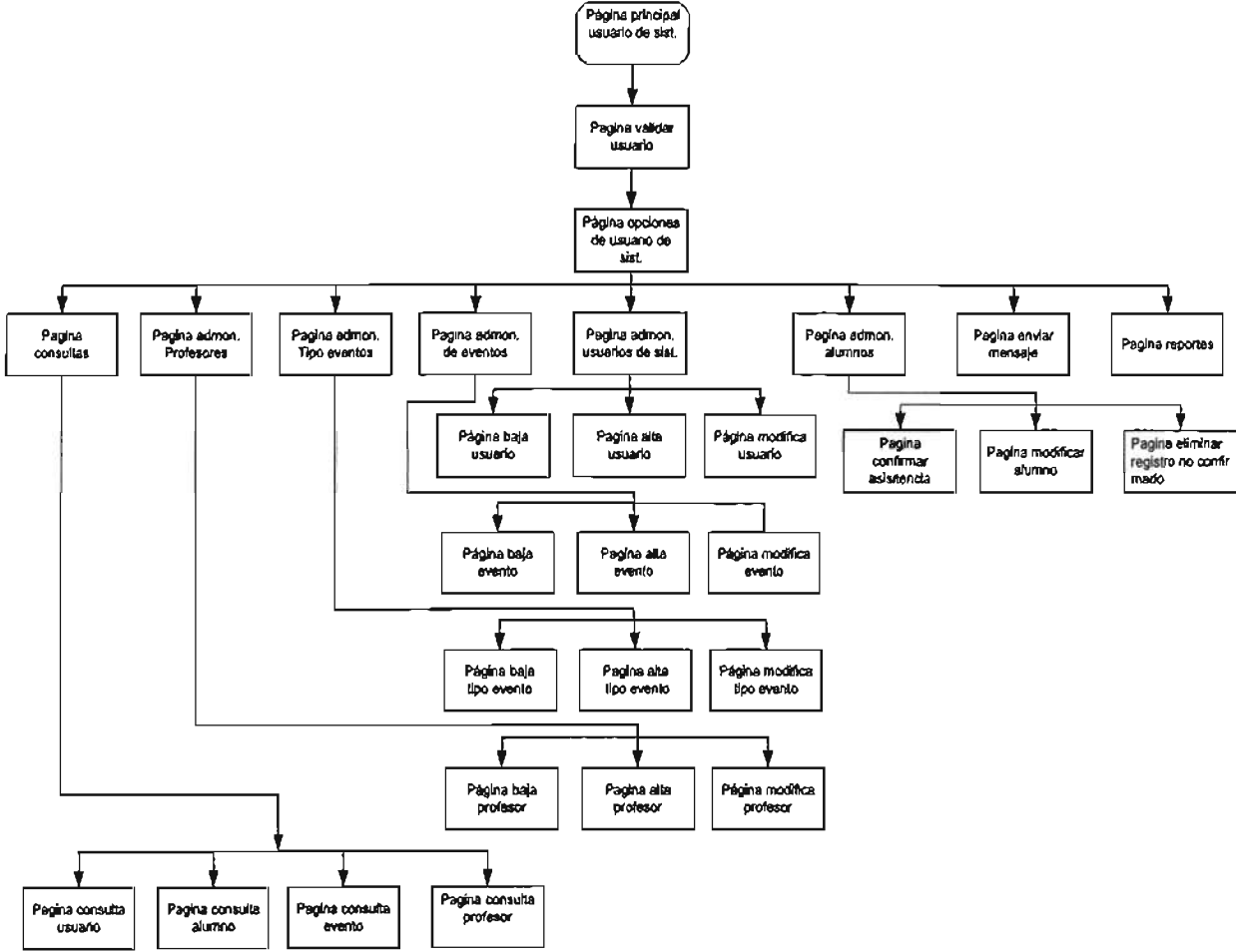


Figura 34 Mapa de navegación usuario de sistema

El mapa navegación del usuario de sistema mostrado en la figura 34, se representa la navegación de las páginas que han de diseñarse para que el usuario de sistemas pueda realizar las siguientes tareas: administrar tipos de evento, administrar eventos, administrar usuarios, administrar instructores, administra participantes, enviar mensajes, consultas y reportes. Se inicia con la página principal de usuario de sistema, enseguida se presenta la página validar usuario, después de esta página se presentan las opciones de usuario de sistema y de ahí se sigue una ruta de navegación, según la tarea a realizar. Por ejemplo, si queremos dar de alta un evento tendremos que pasar por la página administrar eventos y finalmente pasamos a la página alta evento.

3.2 Diseño de las páginas Web

La interfaz ha de ser lo más uniforme posible a través del sitio Web: un mismo sistema de colores, uso de tipografía, colocación de los elementos de navegación, etc. Se han de proporcionar al mismo tiempo, pistas para informar del lugar específico en que nos encontramos, con una cierta variedad dentro de la homogeneidad del conjunto.

En seguida se muestra los aspectos que se deberán tomar en cuenta para diseñar las páginas del sistema o sitio:

- Definir como va ser la página, es decir idear e imaginar la página
- Identificar los elementos con que se formara las partes de la página: textos, imágenes, gráficos, fotografías, colores o audio.
- Definir el contenido informativo.
- Definir los nexos o ligas
- Definir la estructura u organización de los contenidos, la información y las ligas.

La estructura en la que se basarán las páginas es como sigue:

Cada una de las páginas se dividirá en tres partes, un cabecera donde se ubicara el logotipo de la página Web y el nombre del sitio, un área para menú que estará por

debajo del encabezado del lado izquierdo y una área principal donde se pondrá el contenido de la página.

En la figura 35 se muestra la página principal del usuario de sistema, esta página sigue la estructura que se definió para toda página del sitio, es decir, se tiene una sección para el logotipo y nombre del sitio (SAE LABORATORIO DE ELECTRONICA L3), una sección para el menú (aunque en esta página no se mostrara un menú) y una sección para el contenido que se mostrara en la página (sitio solo para personal autorizado). Esta página debe contar con una liga (botón aceptar) hacia la página autenticación de usuario.

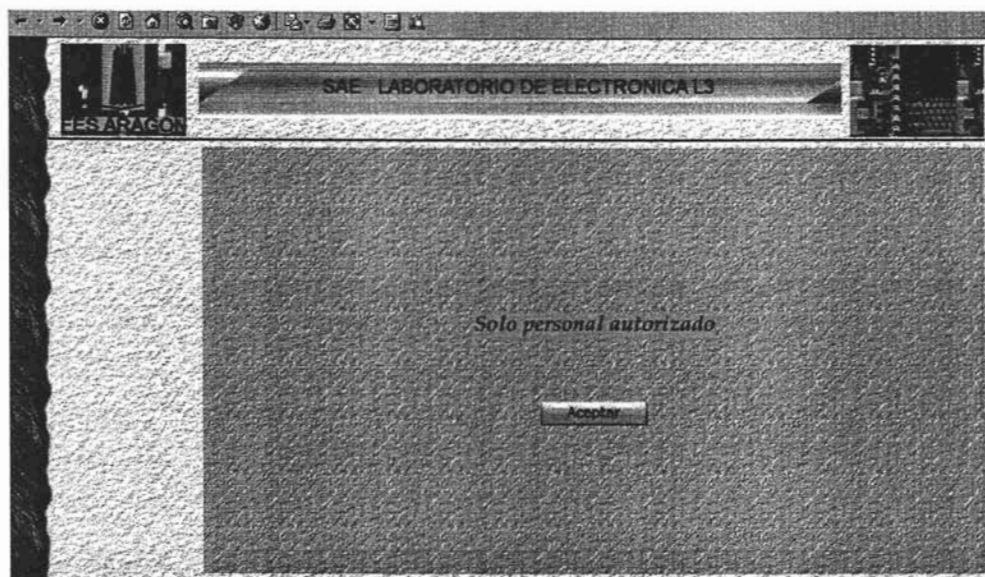


Figura 35 Página principal usuario

La página autenticación de usuario (figura 36) sigue la misma estructura definida, en la sección principal de la página se presenta un formulario donde al usuario se le pide que se identifique, también contiene un botón para enviar los datos para su autenticación. En caso de que los datos enviados sean correctos se presentara la página opciones de usuario de sistema.

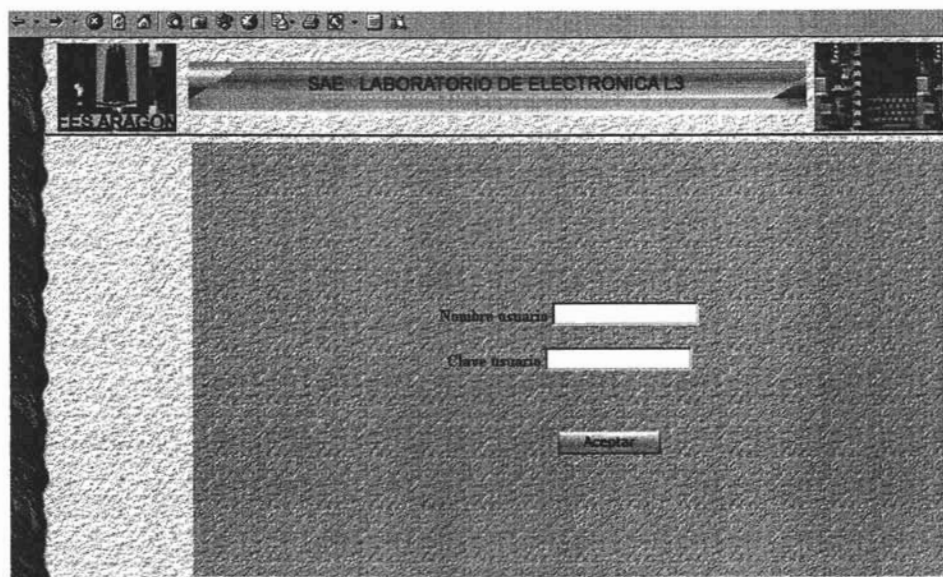


Figura 36 Página autenticación de usuario

La página opciones de usuario de sistema (figura 37) debe contener un menú donde se muestren las acciones que puede realizar el sistema. Cuando se pase el puntero del ratón sobre un elemento del menú debe mostrar en la sección principal de la página una descripción de las tareas en particular que se pueden llevar acabo y dando un clic en el elemento del menú debe llevarlo a la página para realizar esas tareas.

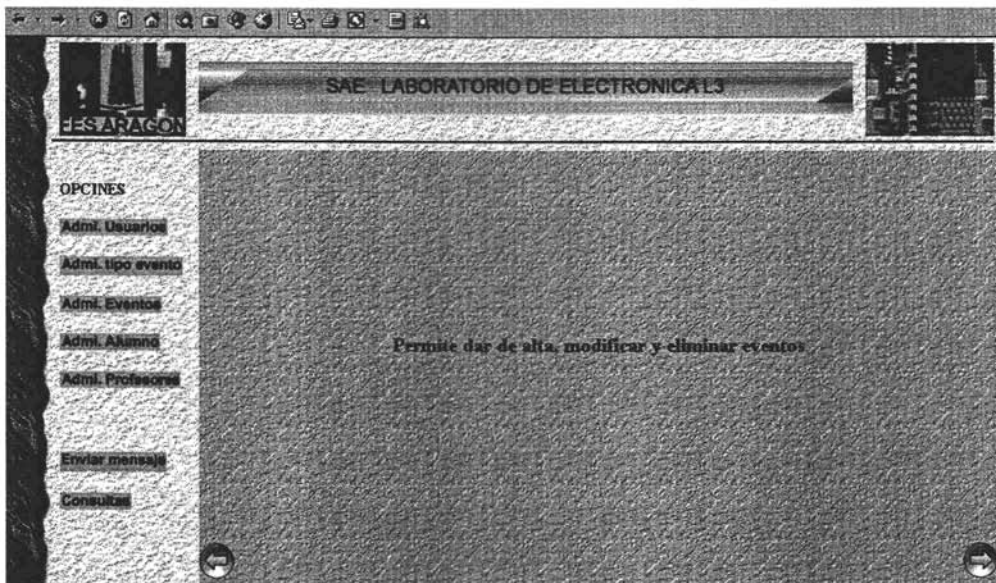


Figura 37 Página opciones de usuario

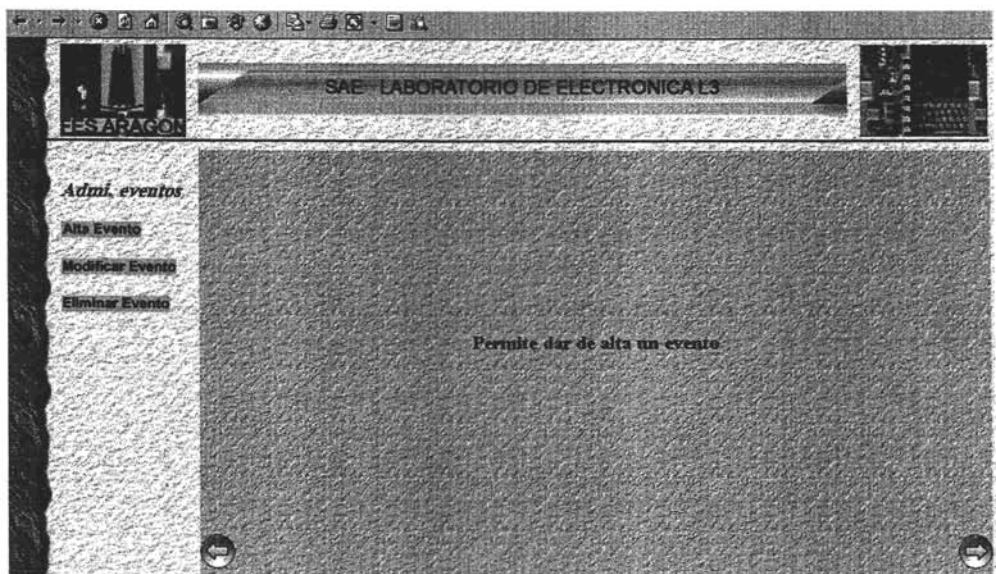


Figura 38 Página administrar eventos

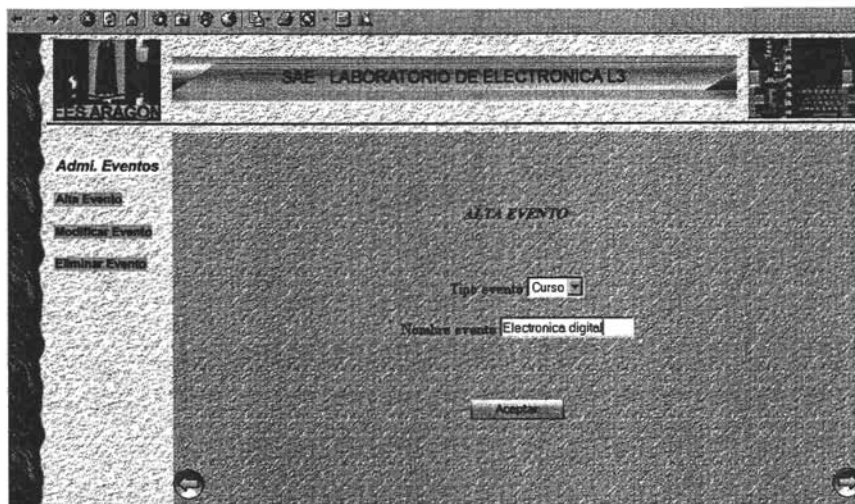


Figura 39 Página alta evento

La página administrar eventos (figura 38) debe tener un encabezado en la sección de menú, el cual debe corresponder al nombre del elemento seleccionado en la página opciones de sistema y contendrá los elementos de menú que indicarán las acciones en particular que se pueden realizar. Al igual que en la página opciones de sistema, al colocarse el puntero del ratón sobre un el elemento del menú se presentara en la sección principal una breve descripción de la tarea a realizar. Dando un clic sobre un elemento del menú se mostrara la página que permitirá realizar una tarea en particular. Además deberá contener un botón para regresar a la página opciones de sistema.

La página alta evento (figura 39) se debe mostrar en la sección principal un formulario, que constara de un menú que permita elegir un tipo de evento (curso, taller etc.), un campo para poner el nombre del evento (electrónica digital), un botón aceptar para efectuar el alta, un botón limpiar el cual nos permite limpiar los campos, un botón regresar que nos permita regresar a la página anterior y otro botón que nos regrese a la página opciones de sistema. En la sección de menú aparecerá todos los elementos desactivados y opacados excepto el elemento en uso estará visible pero inactivo.



Como se puede observar, se siguió la ruta que marca el mapa de navegación para el diseño de las páginas que son necesarias para realizar el alta de un evento. Puede parecer que son muchas pantallas por las que hay que pasar para llegar al objetivo alta evento, pero si tomamos en cuenta que solo se pasarán por la página principal y la página autenticación de usuario solo una vez para entrar al sitio, entonces solo se tendrá que pasar por tres páginas para llegar a un objetivo. Siguiendo con el diseño de las páginas para el sitio, a continuación veremos las correspondientes al mapa de navegación del alumno.

La estructura para el diseño de la página principal del alumno (figura 40) se usa la misma estructura para las páginas del usuario, es decir, debe de contar con tres secciones encabezado, menús y contenido principal. En la sección de menú se colocara las opciones a que tiene acceso el alumno y la sección principal tendrá una imagen relacionada con la electrónica.

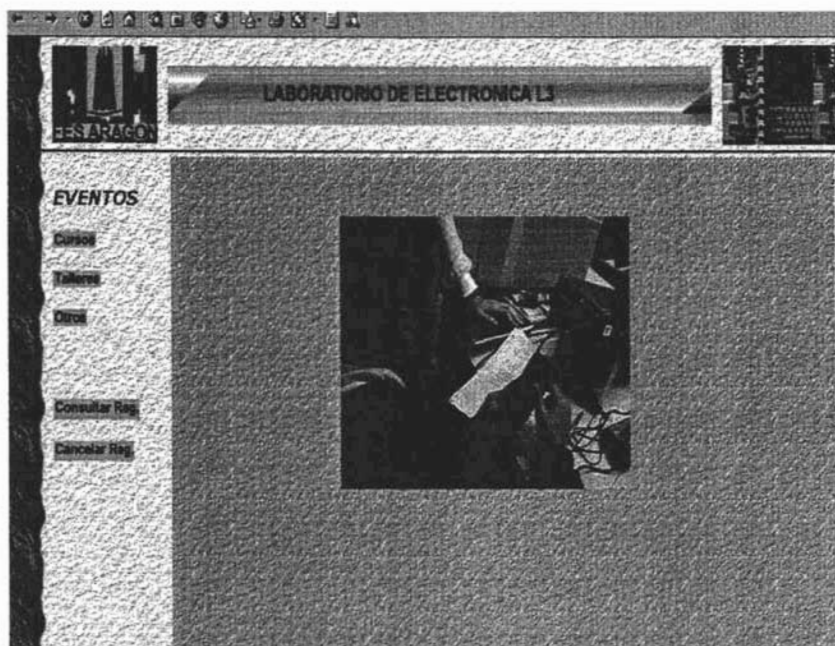


Figura 40 Página principal alumno

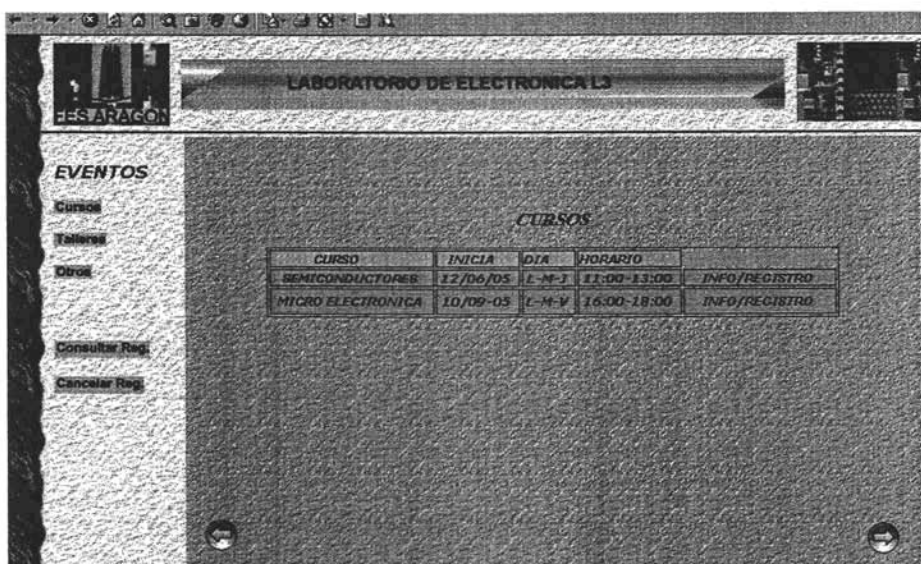


Figura 41 Página cursos

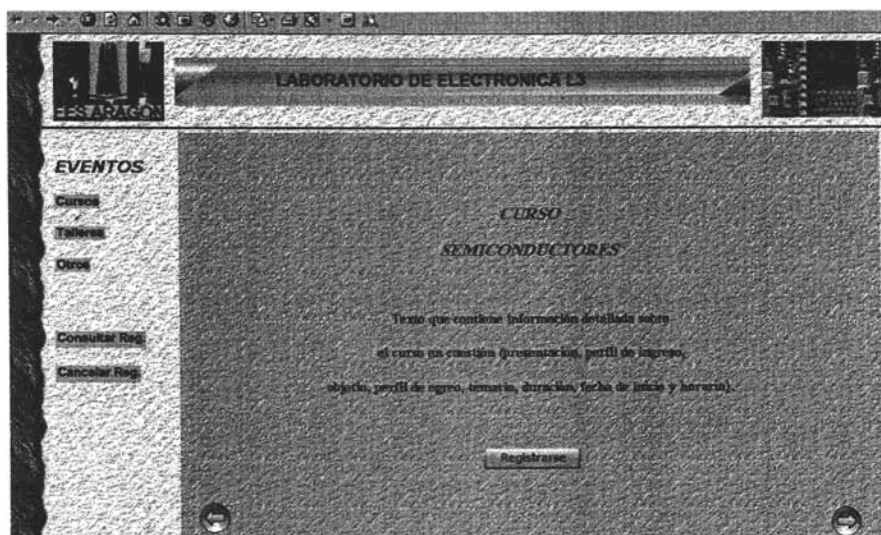


Figura 42 Página información/registro



LABORATORIO DE ELECTRONICA L3

CURSO DE SEMICONDUCTORES

Para registrarse a este curso llene el siguiente formulario:

No. cta

Nombre

Carrera

Mail

Telefono

DATOS DEL CURSO AL QUE SE REGISTRA

Curso	Semiconductores
Inicio	Junio 12
Horario	11:00-13:00

Aceptar Limpiar campos

Figura 43 Página registro

En la página cursos (figura 41) se debe mostrar una lista de los cursos que se impartirán conteniendo los siguientes datos el nombre del curso, la fecha de inicio y horario, así como la opción de mostrar más información sobre el curso y poder registrarse. En el menú solo la opción cursos quedara activa y las demás opciones quedaran deshabilitadas. Y deberá contar con un botón regresar a la página anterior.

La página información/registro (figura 42) debe contener en la sección principal un encabezado (el tipo de curso y el nombre del curso) y la información detallada sobre el curso en cuestión como: la presentación, perfil de ingreso, objetivo, perfil de egreso, temario, duración, fecha de inicio y horario. También en la sección principal debe tener un botón para presentar el formulario que se debe llenar para registrarse, un botón regresar a la página anterior. Todas las opciones de menú quedan inhabilitadas excepto la opción elegida.

En la página registro (figura 43) en su sección principal contendrá un formulario para que el participante introduzca los datos necesarios (No. Cta., nombre, carrera, mail y teléfono) para su registro al curso, también debe tener los datos del curso al que se escribe (curso, inicia, horario). Debe contener un botón (aceptar) para enviar registro y un botón (regresar) para ir a la página anterior.

Como se puede observar se siguió la ruta que marca el mapa de navegación para el diseño de las páginas que son necesaria para registrarse a un curso en particular. Para llegar al objetivo (inscribirse a un curso) se tiene que pasar por cuatro páginas página principal, página cursos, página información/registro y página registro; pero una vez entrando a la página principal solo se tiene que pasar por tres páginas para llegar a un objetivo.

La gama de colores que se emplearan para la construcción de las páginas en sus distintas partes, debe basarse en un color base, que de un espíritu propio al sitio y este color se debe mantener en todas las páginas del sitio. Para establecer una identidad propia y una continuidad entre todas las páginas que lo forman, los colores se podrían basar en el logotipo que se utilizará para la página, es decir trabajando las diferentes partes de la página con los degradados claros y oscuros de estos colores, aunque se podrán usar otros colores que armonicen con los colores base.

En la construcción de las páginas, no se deberá usar más de tres tipos de fuentes para el texto (se usaran fuentes del tipo Sans Serif que pintan las letras planas que son mas fáciles de leer en la pantalla del monitor). El color del texto deberá ser tal que destaque claramente del fondo de la página.

La resolución que se tomarán en cuenta para la construcción de las páginas Web deberá ser de 800x600 tomando en cuenta que estas son las resoluciones más comunes en la actualidad.

Como en todo, la mejor forma es experimentar y experimentar, hasta que encontremos aquella combinación que exprese lo que deseamos transmitir, pero

poniéndonos siempre en el lugar del usuario potencial, no en el lado del diseñador, ya que es este el verdadero destinatario y juez de nuestro trabajo final.

Hasta aquí se ha definido como se serán las páginas del sitio, es decir como estarán estructuradas las páginas, los elementos con que deberán contar (texto, imágenes, colores, ligas, referencias).

El lenguaje que se utilizara como base para construir las páginas Web es HTML, ya que permite aglutinar textos, sonidos e imágenes y combinarlos. Pero como HTML tiene algunas limitaciones (aplicar formato a un documento) e inconvenientes, (la lectura del código HTML se hace pesada y difícil a la hora de buscar errores o depurar las páginas) se utilizaran las hojas de estilo en cascada ya que ayudan a separar el contenido de la forma, es decir, los elementos que componen una página de la forma con la que se muestran. Además, CSS ayuda en gran medida a la definición de estilos en la página, ya que permite ajustar de una manera mucho más precisa cualquier aspecto de cualquier elemento de la página.

Siguiendo con el análisis y diseño del sistema, en el siguiente capítulo se diseñara la base de datos del sistema. La base de datos es muy importante ya que de ella depende que se pueda almacenar y obtener la información requerida de la forma más eficiente.



CAPITULO IV

CAPA DE DATOS

Las bases de datos están presentes en numerosas aplicaciones, sobre todo en aquellas que manejan grandes cantidades de información que deben ser almacenadas y manipuladas de forma rápida y eficiente; las bases de datos en la Web no son la excepción. Las consideraciones a tener en cuenta a la hora de desarrollar una aplicación de bases de datos en la Web no son en principio distintas de las necesarias para cualquier otra aplicación. El ciclo de vida de una base de datos consta de tres etapas: concepción, creación y explotación de la base de datos.

4.1 Concepción de la base de datos

Para el esquema conceptual de la base de datos se utilizara el modelo entidad-relación, ya que es una herramienta de modelado para representar el mundo real, es decir, establece una visión global de los datos de una organización o Sistema de Información, en un nivel próximo al usuario e independiente de las características del equipo donde después se vaya a instrumentar el sistema.

Modelo entidad-relación

Una entidad es cualquier objeto real o abstracto sobre el cual queremos tener información que tiene existencia por si misma y se puede identificar de manera clara y precisa. Un atributo es una propiedad, característica o unidad básica de una entidad.

Los atributos de cada una de las entidades del SAE del Laboratorio L3 se representaran en un diagrama entidad relación.

La base de datos del SAE del Laboratorio L3 contendrá las siguientes entidades:

**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**



Entidad	Descripción
Tipo evento	Almacena los tipos de eventos que existen (cursos, talleres, conferencias, etc.)
Eventos	Almacena los datos correspondientes de un evento en particular. Como son el nombre del curso, fecha de inicio, cupo entre otros.
Alumnos	Almacena los datos personales del alumno.
Profesores	Almacena los datos del profesor.
Registro	Almacena el numero de registro de los alumnos registrados a un evento
Usuario	Almacena los datos del usuario del sistema (Nombre de usuario, contraseña, etc.)
Encuesta	Almacena información respecto de la evaluación del curso y el profesor.
Lugar	Almacena información del lugar físico disponibles para la realización de los eventos.
Horario	Almacena los horarios de los eventos (día y hora)

Diagrama entidad relación

Toda entidad se convierte en una tabla que toma el nombre de la entidad, los atributos de la entidad serán las columnas de la tabla y el atributo identificador principal será la clave primaria. Los atributos que no sean el identificador principal podrán tomar valores nulos.

Las relaciones (asociación o correspondencia entre entidades) uno a muchos (1: M) o uno a uno (1:1), se transforman propagando el atributo identificador principal de la entidad que tiene la cardinalidad máxima 1 a la que tiene la cardinalidad máxima M. Si la relación fuese 1:1 la propagación de la clave podría hacerse en cualquier sentido. El atributo propagado es una clave ajena que hace referencia a la tabla con cardinalidad máxima de 1. El diagrama entidad relación que describe la base de datos del Sistema Eventos L3 se presenta en la figura 44.

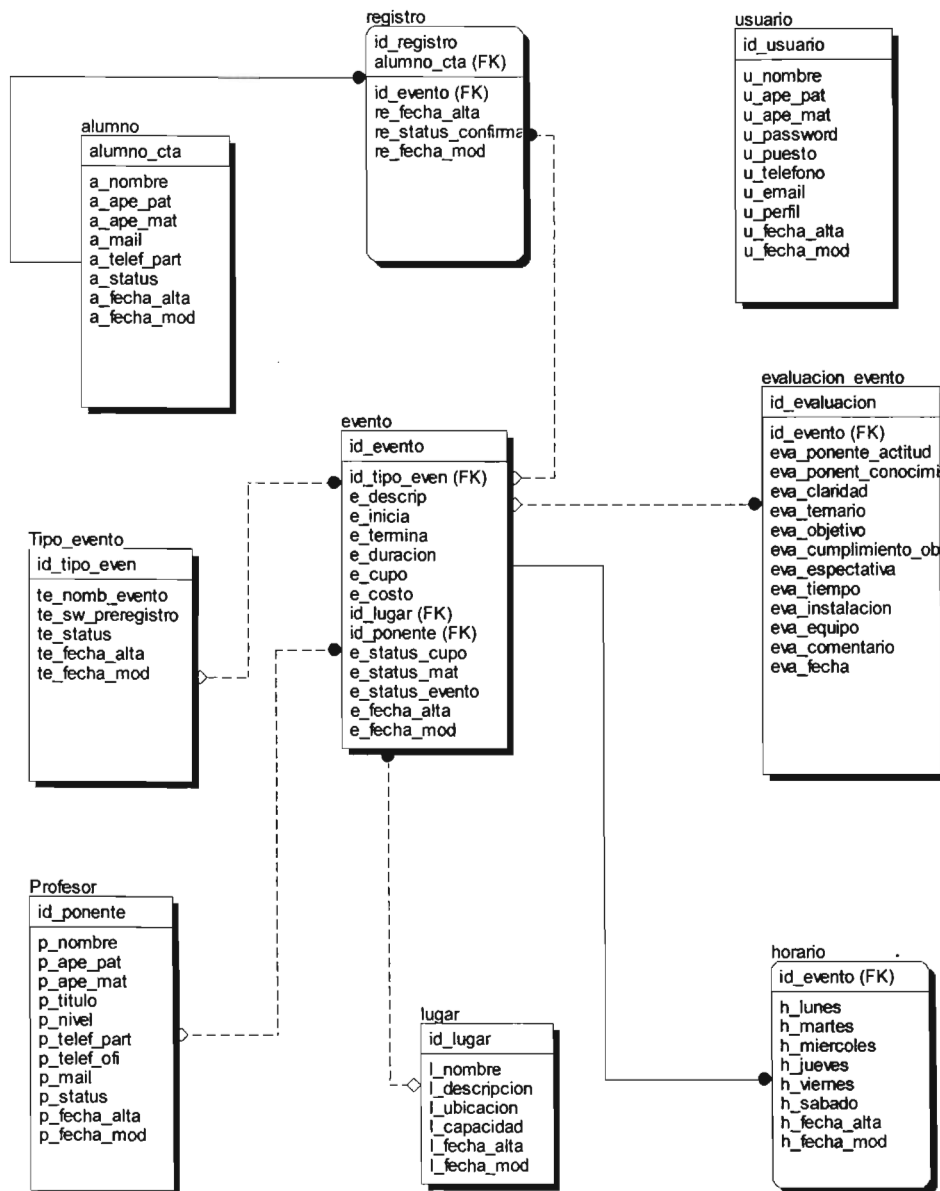


Figura 44 Diagrama entidad relación del Sistema Eventos L3

**Diccionario de datos**

El diccionario de datos contiene los detalles que describen los datos utilizados por el sistema, como es el nombre de los atributos, a que tabla pertenecen los atributos, su descripción, tipo de dato, longitud del campo. El diccionario de datos del Sistema Eventos L3 se muestra a continuación:

Column Table Name	Column Name	Column Datatype	Column Attribute Definition	Column Attribute Is PK
alumno	alumno_cta	NUMBER(10)	Numero de Cta. del Alumno	Yes
alumno	a.fecha_mod	DATE	Fecha de modificación del registro	No
alumno	a_fecha_alta	DATE	Fecha de alta del registro	No
alumno	a_mail	VARCHAR(30)	Dirección electrónica del alumno	No
alumno	a_nombre	VARCHAR(20)	Nombre del alumno	No
alumno	a_ape_pat	VARCHAR(15)	Apellido paterno del alumno	No
alumno	a_ape_mat	VARCHAR(15)	Apellido materno del alumno	No
alumno	a_telef_part	CHAR(8)	Teléfono particular del alumno	No
tipo_evento	id_tipo_even	INTEGER	Llave para acceder a los tipos de eventos	Yes
tipo_evento	te_fecha_alta	DATE	Fecha de alta del tipo de evento	No
tipo_evento	te_fecha_mod	DATE	Fecha de ultima modificación al tipo de evento	No
tipo_evento	te_nomb_evento	CHAR(20)	Nombre del tipo de evento	No
tipo_evento	te_status	SMALLINT	Estado del tipo de evento	No
lugar	id_lugar	INTEGER	Clave de identificación del Lugar	Yes
lugar	l_capacidad	INTEGER	Capacidad del lugar para el evento	No
lugar	l_descripcion	VARCHAR2(40)	Descripción del Lugar	No
lugar	l_fecha_alta	DATE	Fecha de alta del Lugar	No
lugar	l_fecha_mod	DATE	Fecha ultima modificación del Lugar	No
lugar	l_nombre	VARCHAR2(30)	Nombre del Lugar	No
lugar	l_ubicacion	VARCHAR2(40)	Ubicación del Lugar dentro o fuera de la Escuela	No



evalua_evento	id_evaluacion	INTEGER	Clave de identificación de la evaluación	Yes
evalua_evento	eva_claridad	SMALLINT	Indica evaluación sobre claridad del tema	No
evalua_evento	eva_comentario	SMALLINT	Sugerencia por parte del asistente	No
evalua_evento	eva_cumplimiento_objetivo	SMALLINT	indica evaluación sobre cumplimiento de objetivo	No
evalua_evento	eva_equipo	SMALLINT	Indica evaluación sobre el equipó dispuesto	No
evalua_evento	eva_espectativa	SMALLINT	Indica evaluación sobre cumplimiento de expectativas	No
evalua_evento	eva_fecha	DATE	Fecha de sugerencia	No
evalua_evento	eva_instalacion	SMALLINT	Indica evaluación sobre las instalaciones donde se desarrollo el evento	No
evalua_evento	eva_objetivo	SMALLINT	Indica evaluación sobre exposición del objetivo	No
evalua_evento	eva_prof_conocimiento	SMALLINT	Indica evaluación sobre el conocimiento del ponente	No
evalua_evento	eva_prof_actitud	SMALLINT	Indica evaluación sobre la actitud del ponente	No
evalua_evento	eva_temario	SMALLINT	Indica evaluación sobre el temario del evento	No
evalua_evento	eva_tiempo	SMALLINT	Indica evaluación sobre la duración del evento	No
evalua_evento	id_evento	SMALLINT	Clave de identificación del evento	No
evento	id_evento	INTEGER	Identificación del Evento	Yes
evento	e_costo	DECIMAL(0,000)	Costo de un evento en particular	No
evento	e_cupo	INTEGER	Cupo de un evento en particular	No
evento	e_descrip	VARCHAR(50)	Nombre particular de un evento	No
evento	e_duracion	NUMBER	Duración del evento en horas	No
evento	e_fecha_alta	DATE	Fecha de alta del evento	No
evento	e_fecha_mod	DATE	Fecha de ultima modificación al evento	No
evento	e_inicia	DATE	Fecha de inicio de un evento en particular	No
evento	e_status_cupo	SMALLINT	status del cupo del evento	No



			(disponible o agotado)	
evento	e_status_evento	SMALLINT	Estatus del evento (activo o inactivo)	No
evento	e_status_mat	SMALLINT	Estatus de del material que se ocupara en el evento	No
evento	e_termina	DATE	Fecha de termino de un evento en particular	No
evento	id_lugar	INTEGER	Clave de identificación del Lugar (T. lugar)	No
evento	id_ponente	INTEGER	Llave para acceder al ponente (T. Ponente)	No
evento	id_tipo_even	INTEGER	Llave para acceder a los tipos de eventos (T. tipo_evento)	No
horario	id_evento	INTEGER	Identificación del Evento (t evento)	Yes
horario	h_fecha_alta	DATE	Fecha de alta del Horario	No
horario	h_fecha_mod	DATE	Fecha de ultima modificación al horario	No
horario	h_jueves	CHAR(13)	Horario de un evento en particular	No
horario	h_lunes	CHAR(13)	Horario de un evento en particular	No
horario	h_martes	CHAR(13)	Horario de un evento en particular	No
horario	h_miercoles	CHAR(13)	Horario de un evento en particular	No
horario	h_sabado	CHAR(13)	Horario de un evento en particular	No
horario	h_viernes	CHAR(13)	Horario de un evento en particular	No
profesor	id_profesor	INTEGER	Llave para acceder al ponente	Yes
profesor	p_fecha_alta	DATE	Fecha de alta del ponente	No
profesor	p_fecha_mod	DATE	Fecha ultima modificación del ponente	No
profesor	p_mail	VARCHAR(30)	Dirección electrónica del ponente	No
profesor	p_nivel	NUMBER	Indica nivel académico del ponente	No
profesor	p_nombre	VARCHAR(20)	Nombre del ponente	No
profesor	p_ape_pat	VARCHAR(15)	Apellido paterno del ponente	No
profesor	p_ape_mat	VARCHAR(15)	Apellido materno del ponente	No



profesor	p_status	SMALLINT	Indica el estado actual del ponente (0 disponible, 1 no disponible)	No
profesor	p_telef_ofi	CHAR(8)	Teléfono de la oficina del ponente	No
profesor	p_telef_part	CHAR(8)	Teléfono particular del ponente	No
profesor	p_titulo	VARCHAR(25)	Titulo del ponente (Ing., Arq. etc.)	No
registro	id_registro	INTEGER	Identificación del registro al evento	Yes
registro	alumno_cta	NUMBER(10)	Numero de Cta. del Alumno (T. Alumno)	Yes
registro	id_evento	INTEGER	Identificación del Evento (T. evento)	No
registro	re_fecha_alta	DATE	Fecha de registro al evento	No
registro	re_fecha_mod	DATE	Fecha de ultima modificación del registro	No
registro	re_confirma	SMALLINT	Estatus para confirmar asistencia al evento	No
usuario	id_usuario	INTEGER	Llave para acceder al usuario	Yes
usuario	u_email	VARCHAR(30)	Dirección electrónica del usuario	No
usuario	u_fecha_alta	DATE	Fecha de alta del registro de un usuario	No
usuario	u_fecha_mod	DATE	Fecha de ultima modificación al registro del usuario	No
usuario	u_nomb	VARCHAR(20)	Nombre del usuario	No
usuario	u_ape_pat	VARCHAR(15)	Apellido paterno del usuario	No
usuario	u_ape_mat	VARCHAR(15)	Apellido materno del usuario	No
usuario	u_password	CHAR(10)	Contraseña del usuario para ingresar al sistema	No
usuario	u_perfil	INTEGER	Perfil del usuario	No
usuario	u_puesto	VARCHAR(20)	Puesto o cargo que ocupa	No
usuario	u_telefono	CHAR(8)	Teléfono del usuario	No

Después del diseño de la estructura de base de datos, el siguiente paso es crear la base de datos y las tablas correspondientes dentro de la base de datos. A la hora de



crear las tablas, se deben ajustar al diseño de la base de datos para conseguir un almacenamiento óptimo con la menor utilización de espacio.

4.2 Creación de la base de datos *del sistema*

Para crear la base de datos y sus tablas se utilizara el gestor de base de datos relacional MySQL. Como se sabe, MySQL además de permitir crear la base de datos y las tablas, permite realizar más tareas como insertar datos, recuperar datos, actualizar registros, eliminar registros entre otras tareas relacionadas con la base de datos.

Estas tareas se realizan a través de la línea de comandos, es decir, hay que escribir el comando con sus parámetros correspondiente para realizar una tarea en particular. Otra posibilidad que resulta más práctica, que evita trabajar continuamente tecleando órdenes, es utilizar phpMyAdmin, a través de él se puede administrar todos los procesos o funciones típicas de la base de datos MySQL de manera más sencilla, ya que cuenta con una interfaz Web muy intuitiva.

Para crear la base de datos `Eventos_L3` y sus tablas correspondientes del sistema Web, se tiene que escribir la siguiente línea de comandos de MySQL (aunque se puede utilizar phpMyAdmin para el mismo fin):

```
mysql> create database Eventos_L3;
```

Después de crear la base de datos se tiene que crear las tablas correspondientes. Para realizar esta tarea puede crear tabla por tabla:

```
CREATE TABLE alumno (  
alumno_cta    INTEGER (10) NOT NULL,  
a_nombre     VARCHAR (20) NULL,  
a_ape_pat    VARCHAR (15) NULL,  
a_ape_mat    VARCHAR (15) NULL,  
a_fecha_na   DATE NULL,  
a_telef_part CHAR (8) NULL,  
a_cve_lada   CHAR (4) NULL,  
a_mail       VARCHAR(30) NULL,  
a_fecha_alta DATE NULL,  
a_fecha_mod  DATE NULL,  
a_status     SMALLINT NULL  
);
```



O usar un archivo de texto que tenga los comandos necesarios para crear las tablas de un solo paso. Como se muestra e las siguientes líneas:

```
DROP TABLE IF EXISTS ALUMNO;
DROP TABLE IF EXISTS CTL_EVENTO;
DROP TABLE IF EXISTS CTL_LUGAR;
DROP TABLE IF EXISTS ENCUESTA_DIPLO;
DROP TABLE IF EXISTS EVALUACIÓN_EVENTO;
DROP TABLE IF EXISTS EVENTO;
DROP TABLE IF EXISTS HORARIO;
DROP TABLE IF EXISTS PONENTE;
DROP TABLE IF EXISTS REGISTRO;
DROP TABLE IF EXISTS USUARIO;

CREATE TABLE alumno (
    alumno_cta    INTEGER (10) NOT NULL,
    a_nombre     VARCHAR (20) NULL,
    a_ape_pat    VARCHAR (15) NULL,
    a_ape_mat    VARCHAR (15) NULL,
    a_telef_part CHAR (8) NULL,
    a_mail       VARCHAR(30) NULL,
    a_fecha_alta DATE NULL,
    a_fecha_mod  DATE NULL,
    a_status     SMALLINT NULL
);
ALTER TABLE alumno ADD (PRIMARY KEY (alumno_cta) );

CREATE TABLE tipo_evento (
    id_tipo_even    INTEGER NOT NULL,
    ce_nomb_evento  VARCHAR(20) NULL,
    ce_sw_preregistro SMALLINT NULL,
    ce_fecha_alta   DATE NULL,
    ce_fecha_mod    DATE NULL,
    ce_status       SMALLINT NOT NULL
);
ALTER TABLE ctl_evento ADD (PRIMARY KEY (id_tipo_even) ) ;
```

```
CREATE TABLE ctl_lugar (
    id_lugar          INTEGER NOT NULL,
    cl_nombre         VARCHAR(30) NOT NULL,
    cl_descripcion    VARCHAR(40) NOT NULL,
    cl_ubicacion      VARCHAR(40) NOT NULL,
    cl_capacidad      INTEGER NULL,
    cl_fecha_alta    DATE NOT NULL,
    cl_fecha_mod      DATE NOT NULL
);
ALTER TABLE ctl_lugar ADD (PRIMARY KEY (id_lugar) );

CREATE TABLE evaluacion_evento (
    id_evaluacion     INTEGER NOT NULL,
    id_evento         INTEGER NULL,
    eva_ponente_actitud SMALLINT NOT NULL,
    eva_ponent_conocimiento SMALLINT NOT NULL,
    eva_claridad      SMALLINT NOT NULL,
    eva_temario       SMALLINT NOT NULL,
    eva_objetivo      SMALLINT NOT NULL,
    eva_cumplimiento_objetivo SMALLINT NULL,
    eva_espectativa   SMALLINT NOT NULL,
    eva_tiempo        SMALLINT NOT NULL,
    eva_instalacion   SMALLINT NOT NULL,
    eva_equipo        SMALLINT NOT NULL,
    eva_comentario    VARCHAR(60) NULL,
    eva_fecha         DATE NULL
);
ALTER TABLE evaluacion_evento ADD (PRIMARY KEY (id_evaluacion) );
```



```
CREATE TABLE evento (
  id_evento      INTEGER NOT NULL,
  id_tipo_even  INTEGER NULL,
  e_descrip     VARCHAR (50) NULL,
  e_inicia      DATE NULL,
  e_termina     DATE NULL,
  e_duracion    INTEGER NULL,
  e_cupo       INTEGER NULL,
  e_costo       DECIMAL (00,000) NULL,
  id_lugar      INTEGER NULL,
  id_ponente    INTEGER NULL,
  e_status_cupo SMALLINT NULL,
  e_status_mat  SMALLINT NULL,
  e_status_evento SMALLINT NULL,
  e_fecha_alta  DATE NULL,
  e_fecha_mod   DATE NULL
);
ALTER TABLE evento ADD (PRIMARY KEY (id_evento) );

CREATE TABLE horario (
  id_evento      INTEGER NOT NULL,
  h_lunes        CHAR(13) NOT NULL,
  h_martes       CHAR(13) NOT NULL,
  h_miercoles    CHAR(13) NOT NULL,
  h_jueves       CHAR(13) NOT NULL,
  h_viernes      CHAR(13) NOT NULL,
  h_sabado       CHAR(13) NOT NULL,
  h_fecha_alta  DATE NULL,
  h_fecha_mod   DATE NULL
);
ALTER TABLE horario ADD ( PRIMARY KEY (id_evento) );
```



```
CREATE TABLE profesor (  
    id_profesor INTEGER NOT NULL,  
    p_nombre VARCHAR (20) NULL,  
    p_ape_pat VARCHAR (15) NULL,  
    p_ape_mat VARCHAR (15) NULL,  
    p_titulo CHAR (25) NULL,  
    p_nivel INTEGER NULL,  
    p_telef_part CHAR (8) NULL,  
    p_telef_ofi CHAR (8) NULL,  
    p_mail VARCHAR(30) NULL,  
    p_status SMALLINT NULL,  
    p_fecha_alta DATE NULL,  
    p_fecha_mod DATE NULL  
);  
ALTER TABLE profesor ADD (PRIMARY KEY (id_profesor) );  
  
CREATE TABLE registro (  
    id_registro INTEGER NOT NULL,  
    alumno_cta INTEGER (10) NOT NULL,  
    id_evento INTEGER NULL,  
    re_status_confirma SMALLINT NULL,  
    re_fecha_alta DATE NULL,  
    re_fecha_mod DATE NULL  
);  
ALTER TABLE registro ADD (PRIMARY KEY (id_registro, alumno_cta) );  
  
CREATE TABLE usuario (  
    id_usuario INTEGER NOT NULL,  
    u_perfil INTEGER NOT NULL,  
    u_password CHAR (10) NULL,  
    u_nombre VARCHAR (20) NULL,  
    u_ape_pat VARCHAR (15) NULL,  
    u_ape_mat CHAR (15) NULL,  
    u_puesto VARCHAR (20) NULL,  
    u_telefono CHAR (8) NULL,  
    u_email VARCHAR (30) NULL,  
    u_fecha_alta DATE NULL,  
    u_fecha_mod DATE NULL  
);
```



Como se menciono anteriormente estas líneas se guardan en un archivo de texto con la extensión sql, que en este caso particular tiene el nombre T_eventos_L3.sql.

4.3 Acceso a la base de datos

Para acceder a la base de datos Eventos_L3 desde Web se deberá seguir los siguientes pasos:

1. Comprobar y filtrar los datos procedentes del usuario.

Eliminar todos los espacios en blanco que se puedan introducir por parte del usuario de manera involuntaria al principio o al fina, esto se pude hacer con la función trim(). Filtrar los datos para evitar problemas de seguridad.

2. Configurar una conexión a la base de datos Eventos_L3. Para establecer una conexión al servidor de MySQL se utilizara la siguiente línea de comandos:

```
$db = mysql_pconnect('localhost','root',' ', Eventos_L3');
```

3. Consultar la base de datos.

Para realizar una consulta se utilizara la función mysql_query(), sin embargo, antes de realizar la tarea, conviene configurar la consulta que se desea ejecutar:

```
$query = "select * from alumno where " . $alumn_cta."  
Like '%" . $No. Cta. "%";
```

Ahora se puedes ejecutar la consulta:

```
$reult = mysql_query($query);
```

4. Recuperar los resultados.

Para recuperar los resultados se pueden utilizar las funciones:

```
mysql_num_rows()
```

```
mysql_fectch_array()
```

5. Desconexión de la base de datos

Se puede utilizar: `mysql_close ()`



Como siempre se debe escribir código que se ejecute correctamente sin importar en que servidor lo hace, es decir, código portátil y fácil de mantener. En el SAE del Laboratorio L3 se utilizará para la conexión a la base de datos la librería ADOdb de PHP; esta es una librería de abstracción de datos que permite asegurar en gran medida la portabilidad de la aplicación. Provee una serie de funciones comunes para comunicarse con las distintas bases de datos como MySQL, PostgreSQL, Oracle, Interbase, Microsoft SQL Server, Access, FoxPro, Sybase, ODBC y ADO. Con esta librería solo se tendrá que hacer una vez el código sin importar si el DBMS cambia, así se evitará el tener que cambiar el código y no se tendrá que hacer un sitio para cada DBMS, pues el tipo de base de datos lo especificamos en una variable y tan solo bastará cambiar esa variable por el nombre del actual DBMS.

Para conectarse a la base de datos MySQL se tienen que escribir las siguientes líneas:

```
$DOCUMENT_ROOT=$HTTP_SERVER_VARS['DOCUMENT_ROOT'];
include('adodb/adodb.inc.php');
$conn = &ADONewConnection('mysql');
$conn->PConnect('localhost','usuario','contraseña','Eventos_L3');
```

En la primer línea se determina la ruta absoluta donde se encuentran las paginas en el servidor Web, para esto se utiliza la variable incorporada `HTTP_SERVER_VARS['DOCUMENT_ROOT'];` que apunta a la base del árbol de documentos del servidor Web, en la segunda línea se indica el archivo con los datos que necesita el programa para reconocer ADODB, en la tercer línea se indica el tipo manejador de base de datos en este caso MySQL y en la última línea se hace la conexión con la base de datos; en esta línea se indica el nombre del host en que se ejecuta el servidor MySQL, el nombre de usuario, contraseña del usuario y el nombre de la base de datos a conectarse.

4.4 Seguridad

Por su propia naturaleza, la Web es un lugar peligroso, este medio esta diseñado para permitir que numerosos usuarios soliciten servicios desde sus equipos, por tanto, algunos riesgos que se corren en un sitio Web es la exposición de datos confidenciales, perdida o destrucción de datos, modificación de datos entre otros.

Teniendo esto en cuenta el sistema debe de contar con:

- Un proceso para asignar nombre y contraseña a cada usuario del sistema.
- Un proceso para autenticar a los usuarios del sistema.
- Un proceso para asignar el nivel de privilegios necesarios para realizar su trabajo.

Para implementar estos procesos se puede realizar:

El servidor Apache que permite proteger de manera sencilla una parte del sitio o el sitio en su totalidad, presentando una ventana al usuario y pidiendo su nombre de usuario y contraseña cuando intenta visitar un directorio de acceso restringido. También se puede usar PHP y MySQL, que incorporan muchas otras formas de lograr el mismo objetivo. MySQL resulta más rápido que la autenticación integrada, PHP suministra una autenticación más flexible o presenta la solicitud más atractiva.

Para implementar la seguridad del sitio que se esta diseñando se utilizará `phpsecurepages` que es un modulo de PHP que asegura las paginas con un nombre y contraseña. La instalación de esta aplicación permite asegurar páginas de forma más rápida y simple. Solo basta con adicionar unas líneas al código PHP en la página. También permite niveles de seguridad, se pueden crear grupos y darle ciertos privilegios, también permite modificar la pantalla que tiene como predeterminada.

Para asegurar las páginas del sitio Web primero se debe configurar el modulo de `phpsecurepages`, editando el archivo `secure.php` en cual hay que indicar ciertos parámetros como:



<code>\$cfgIndexpage = '/index.php';</code>	//indicar que pagina ir si el login a fallado
<code>\$admEmail = '';</code>	//indicar el mail del administrador cuando se presenta un error
<code>\$noDetailedMessages = true '';</code>	//elegir una descripción breve o de tallada cuando ocurre un error
<code>\$useDatabase = false; ó \$useData = true;)</code>	//Elegir usar base de datos o entradas de archivo
<code>\$cfgServerHost = 'localhost';</code>	// MySQL hostname
<code>\$cfgServerPort = '';</code>	// MySQL port - leave blank for default port
<code>\$cfgServerUser = 'root';</code>	// MySQL user
<code>\$cfgServerPassword = '';</code>	// MySQL password
<code>\$cfgDbDatabase = 'phpSecurePages';</code>	// MySQL database name containing pSecurePagetable
<code>\$cfgDbTableUsers = 'phpSP_users';</code>	//MySQL table name containing phpSecurePagesuserfields
<code>\$cfgDbLoginfield = 'user';</code>	// MySQL field name containing login word
<code>\$cfgDbPasswordfield = 'password';</code>	// MySQL field name containing password
<code>\$cfgDbUserLevelfield = 'userlevel';</code>	// MySQL field name containing user level

Una vez configurado el archivo secure.php, el siguiente paso para hacer una página segura, es insertar en las primeras líneas del código de cada una de las páginas las siguientes líneas:

```
<?PHP
$requiredUserLevel = array(2, 4);
$minUserLevel=6;
$cfgProgDir = 'phpSecurePages/';
include($cfgProgDir . "secure.php");?>
```



En la primera línea se indica que los usuarios con el nivel 2 y 4 tienen acceso a la página, la segunda línea indica que todos los usuarios con nivel 6 o con un nivel superior tienen acceso a la página, la tercera línea indica donde se encuentran los archivos phpSecurePages, en la última línea se indica el archivo con los datos que necesita el programa para reconocer phpSecurePages.

Todo Sistema de Información, debe permitir que la información almacenada sea correcta, que no se puedan hacer operaciones no válidas y que cierta información importante o valiosa, quede restringida para la mayoría de los usuarios. Las reglas de negocios nos permiten definir estos aspectos, en el capítulo siguiente se muestran las reglas de negocio del sistema.

CAPITULO V

REGLAS DE NEGOCIOS

Como todo sistema o aplicación trata de representar el funcionamiento del mundo real, es necesario que refleje las restricciones que existen en el negocio dado, para que no sea posible llevar acciones no validas.

5.1 Reglas de negocios

Para las operaciones de la base de datos Eventos_L3 de nuestro sistema, las reglas de negocios que se deben ver reflejadas en la aplicación son las siguientes:

Reglas del modelo de datos

- Los campos fechas deberán ser siempre validas.
- El campo cupo para un evento no podrá ser un número negativo.
- El campo precio para un evento siempre deberá ser un numero positivo.
- Los campo llave nunca podrán ser nulos.
- El campo confirmar asistencia solo podrá tomar el valor 1 ó 0.
- Los campos estatus solo podrán tomar el valor 1 ó 0.

Reglas de relación

- Un tipo evento no se puede borrar si existe un evento asociado a el.
- Para dar de alta un evento debe existir un tipo de evento al que se pueda asociar.
- Antes de registrar o inscribir a un participante se deberá verificar si hay cupo para dicho evento.
- Se le pondrá al participante, una marca de confirmación de asistencia al evento, cuando haya concluido sus tramites correspondientes al evento en el que este inscrito.
- Un evento no se podrá borrar si existen al menos un participante asociado a el.



- El usuario (participante) puede estar inscrito en uno o más eventos.
- Para que un usuario (participante) pueda inscribirse a un evento debe existir en la base del sistema.
- Solo se podrá aceptar inscripciones dos días antes del inicio del evento.

Reglas de restricción

- Tanto el administrador, como los usuarios del sistema tendrán que autenticarse en el sistema antes de poder realizar cualquier tarea sobre el.
- Solo el administrador puede dar de alta, modificar, dar de baja y consultar a un usuario de sistema.
- El usuario de sistema solo podrá cambiar su contraseña previa autenticación ante el sistema.
- No pueden existir dos tipos de eventos con el mismo nombre.
- Un alumno solo podrá estar registrado una vez en la base de datos del sistema.
- Se requerirá como mínimo para registrar a un alumno en la base de datos nombre completo, No. de cta., teléfono y dirección electrónica.
- Pueden existir uno o más eventos con el mismo nombre.
- La fecha de inicio de un evento no deberá ser menor que la fecha de alta del evento en el sistema.
- La fecha de termino de un evento no deberá ser menor que la fecha de inicio del evento.

Aunque estas reglas de negocio son las que se identificaron a la hora del análisis pueden surgir otras.

Ubicación de las reglas de negocios

Como el SAE del Laboratorio L3 se implementara en una arquitectura de tres capas, las reglas de negocios descritas anteriormente, se ubicaran en este esquema de tres capas, tomando en cuenta ciertas pautas a la hora de tomar la decisión de donde



ubicar las reglas de negocios, para esto se tomara como base la clasificación de las reglas de negocios.

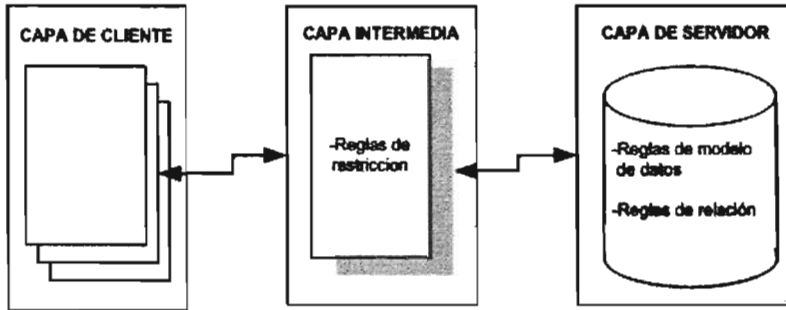


Figura 45 Ubicación de las Reglas de negocios

En general, lo más recomendable suele ser implementar todas las reglas de negocios relativas al modelo de datos y las de relaciones en el servidor, ya que los modernos servidores suelen llevar acabo estas tareas eficazmente. Por lo que respecta al resto de las reglas, la mejor solución suele implementarlas en la capa intermedia.

5.2 Plan de trabajo para la implementación del sistema

El plan de trabajo para la implementación del sistema se muestra en el siguiente diagrama:

ACTIVIDAD	OCTUBRE				NOVIEMBRE				DICIEMBRE			
	1	2	3	4	1	2	3	4	1	2	3	4
Preparación del equipo (PC) y el software necesario para desarrollar el sistema												
Implementación de la base de datos												
Diseño de la interfase												
Programación de los módulos necesarios que integraran el sistema												
Subir el sistema al servidor												
Pruebas de funcionamiento del sistema												
Validación por parte del usuario												

Figura 46 Diagrama de actividades para la implementación del sistema

El plan de trabajo se diseño teniendo en cuenta solo una persona para desarrollar el sistema. Como se observa en el diagrama, el tiempo estimado para desarrollar el sistema fue de 3 meses aproximadamente, es decir, puede que algunas de las actividades pueda concluirse antes de lo previsto, o se necesite un poco más de tiempo para alguna actividad.

Todo el desarrollo se realizara sobre una PC y posteriormente se subirá al servidor para hacer las pruebas de funcionamiento necesarias.

5.3 Estándares de programación

A continuación se listan algunas técnicas que se utilizarán para programar, con el fin de que el código sea entendible y poder encontrar más fácilmente errores en el programa si estos se llegan a dar:

- Hacer una sangría de todo el cuerpo de cada función en un nivel.
- Definir un tamaño de la sangría (tres espacios por cada uno de los niveles de sangría).



- Seleccionar nombres de variables significativos que ayuden a la documentación del programa.
- Evitar juntar palabras separadas como en totalcomision, en vez de ello se deberán separar las palabras con subrayado como total_comisión que es más legible.
- Separar las declaraciones y los enunciados ejecutables en un procedimiento o función, mediante una línea en blanco para enfatizar donde terminan las declaraciones y donde empiezan los enunciados ejecutables.
- Colocar espacios en blanco a ambos lados de un operador binario, ya que esto hace resaltar al operador y hace al programa más legible.
- Hacer una sangría en los enunciados del cuerpo de una estructura de decisión.

5.4 Pruebas del sistema

Las pruebas del sistema se realizaran de la siguiente manera:

TIPO DE PRUEBA	ACCIONES A REALIZAR
Pruebas unitarias	Se probaran cada uno de los módulos que integran el sistema.
Pruebas de integración	Se probará la integración de los módulos que conforman la estructura del sistema.
Pruebas del sistema	Probar si el sistema cumple con los objetivos indicados.
Pruebas de aceptación	El usuario probará el sistema para comprobar que cumple con los requisitos por él fijados.

Las pruebas unitarias y las pruebas de integración se podrán realizar sobre la computadora donde se desarrollara el sistema, pero las pruebas del sistema y las pruebas de aceptación se deberán realizar sobre el servidor donde se alojará el sistema.

En el diagrama de la figura 46 se puede observar el tiempo que se dedicará para las pruebas del sistema y pruebas de aceptación, las pruebas de unidad y de integración se realizarán conforme se vaya programando los módulos y se integrarán a la estructura del sistema el sistema.

5.5 Mapa del sitio

Es recomendable tener una página dedicada al mapa del sitio Web (más aun cuando el sitio es muy grande). El principal beneficio de contar con el mapa del sitio, es dar al usuario una descripción de la estructura del sitio en un solo vistazo; dedicando toda una página a la visualización de la arquitectura de la información. Si está bien diseñada, esta descripción incluirá varios niveles de jerarquía, sin ser tan grande como para que el usuario pierda la capacidad de percibirlo como un todo. El mapa debe ser una simple y compacta disposición de links; y debe mostrar todo en un vistazo.

En la figura 47 se presenta el diseño la página que contendrá el mapa del sitio Eventos L3, que solo podrán observar los usuarios del sistema, este mapa del sitio muestra de un solo vistazo la estructura del sitio. El usuario podrá elegir rápidamente la tarea que le interesa realizar ya que cada una de las opciones deberá tener un link que lo lleve a la página correspondiente para realizar la tarea elegida. La página del sitio deberá también contar con la opción de ir a la página principal.

En cada una de las páginas del sitio deberá tener la opción que permita acceder a la página que contiene el mapa del sitio, esta opción deberá ser fácilmente identificable.

El mapa de la figura 48 muestra las categorías de los tipos eventos y los eventos que pertenecen a cada tipo de evento. Cada evento en particular tiene una liga que lleva a la página que contiene los detalles del evento. También la página mapa del sitio concerniente al alumno, debe tener la opción de llevar este a la página principal.

En la página principal y las demás páginas a la que tiene acceso el alumno deberá tener la opción de ir directamente a la página que contiene el mapa de sitio, esta opción debe ser fácilmente identificable.



Figura 47 Página mapa de sitio usuario

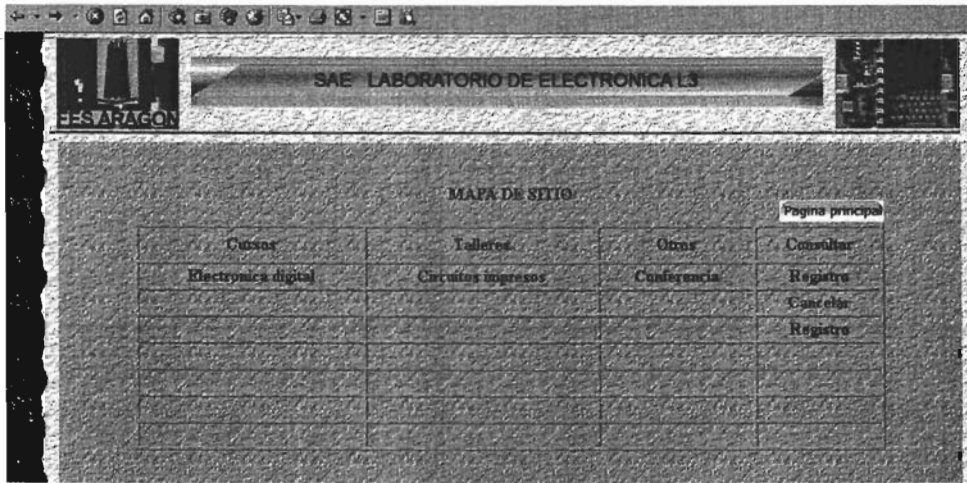


Figura 48 Página mapa de sitio alumno

Conclusiones

El desarrollo de un Sistema Web implica conocer varios aspectos que van desde conocer los elementos de hardware donde se desarrollara la aplicación Web, y donde se instalaran los elementos necesarios de software con la que se desarrollará la aplicación.

Pero uno de los aspectos más importantes, es hacer un análisis del sistema a desarrollar y documentarlo, ya que de esto depende que el sistema cumpla con requerimientos para el cual se desarrollara. Para este análisis se deben utilizar las herramientas que nos permitan hacer este análisis y documentarlo, siendo una excelente opción UML.

Un buen diseño de la base de datos es fundamental, ya que permitirá un manejo de la información más eficiente. Por lo que las consideraciones a tener en cuenta a la hora de desarrollar una aplicación de bases de datos en la Web, no son en principio distintas de las necesarias para cualquier otra aplicación.

También es importante identificar las reglas de negocios, ya que estas nos permiten mantener la integridad de los datos que se manipulan en la base de datos.

Algunos aspectos importantes que se deben tomar en cuenta al desarrollar la aplicación web:

- La interfase de usuario en los sistemas Web es un aspecto fundamental a tener en cuenta, ya que un buen diseño de la interfase de usuario ayudara a que el sitio Web tenga éxito.
- La seguridad es un aspecto muy importante que se tiene que tomar en cuenta al desarrollar una aplicación Web, esto para garantizar la integridad de la información. El nivel de seguridad dependerá de la importancia de la información que se quiere proteger.





- La portabilidad de la aplicación Web es otro aspecto que se debe considerar. El hacer una aplicación Web portátil es conveniente ya que se podrá migrar a otro servidor sin hacer muchos cambios al código para que se pueda ejecutar correctamente.

Se puede concluir que el éxito de una aplicación Web depende en primer lugar de un buen análisis y diseño del sistema a desarrollar, de una interfase de usuario bien estructurada, de una buena implementación de seguridad, la elección del software adecuado para desarrollar y ejecutar la aplicación Web, y el hardware adecuado donde se instalara la aplicación.



BIBLIOGRAFIA

Fitzgerald, Dennis
Análisis y Diseño de Sistemas de Información
Prentice Hall

Jamsa, Kris; King, Konrad y Anderson, Andy
Superutilidades para HTML y Diseño WEB
McGraw Hill

Welling, Luke
Desarrollo Web con PHP y MySQL
Anaya Multimedia

Gallego Vázquez, Antonio
Desarrollo WEB Con PHP y MySQL
Anaya Multimedia

DOCUMENTACIÓN ELECTRÓNICA

Documentación del proyecto GNU, Free Software Foundation.
<http://www.gnu.org/>

Documentación de herramientas para diseño de aplicaciones web.
<http://www.maestrosdelweb.org/editorial/linux/>

Documentación de PHP.
<http://directorio.adfound.com/visitar.php>

Documentación de herramientas para diseño de aplicaciones web.
<http://www.desarrolloweb.com>

Documentación de herramientas para diseño de aplicaciones web.
<http://www.webstyleguide.com>

Documentación de herramientas para diseño de aplicaciones web.
<http://www.tejedoresdelweb.com>

Documentación de herramientas para diseño de aplicaciones web.
<http://www.webestilo.com>

