



**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

**“APLICACIÓN PARA CONTROLAR INFORMACIÓN DE SITIOS
DE INTERÉS TURÍSTICOS, A TRAVÉS DE WWW”**

TRABAJO ESCRITO

EN LA MODALIDAD DE “SEMINARIOS Y CURSOS DE ACTUALIZACIÓN
Y CAPACITACIÓN PROFESIONAL”

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

PRESENTA:

NANCY RAMOS MARTÍNEZ

ASESOR: ING. CÉSAR GERMÁN ROSAS

MÉXICO, 2007



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.

Gracias a mis padres y a mis hermanos que me apoyaron todo este tiempo, por el esfuerzo que realizaron junto conmigo para poder lograr una meta más en mi vida.

“Nadie sabe de lo que es capaz de hacer, hasta que trata de hacerlo”.

Marden.

Índice	
Diplomado	4
Módulo I. Fundamentos del Desarrollo de Sistemas en Web	5
Módulo II. Análisis y Diseño con UML	7
Módulo III. Introducción a la programación Orientada a Objetos con Java	9
Módulo IV. Capa de Usuario	11
Módulo V. Capa de Datos	14
Módulo VI. Reglas de Negocio	15
Introducción	17
1. Antecedentes	18
1.1 Internet	18
1.1.1 Historia de WWW	20
1.1.2 Actualidad	20
1.1.3 Sistemas Orientados al Web	21
1.2 Arquitectura de tres capas	22
1.2.1 Capa de datos	22
1.2.2 Capa de negocio	22
1.2.3 Capa de presentación	23
1.3. Base de Datos	24
1.3.1 Ciclo de vida de una BD	24
1.3.2 DBMS	24
1.3.3 Modelo de datos	25
1.3.4 MySQL	25
1.4 Lenguaje SQL	26
1.4.1 Funciones de agrupamiento	26
1.4.2 Cláusulas	26
1.4.3 Funciones	27
1.4.4 Creación de una tabla	28
1.4.5 Llave primaria	29
1.4.6 Llave foránea	29
1.4.7 Vista	29
1.5 UML	30
1.5.1 Diagrama de Casos de Uso	30
1.5.2 Diagrama de Estados	31
1.5.3 Diagrama de Colaboración	31
1.5.4 Diagrama de Clases	32
1.5.5 Diagrama de Secuencia	32
1.6 Servidor	34
1.6.1 Clasificación	34
1.6.2 Servidor Apache Tomcat	34
1.6.3 Instalar de Apache Tomcat	35

1.7. Diseño de página Web	37
1.7.1 HTML	37
1.8 Java	40
1.8.1 Instalar JDK	39
1.9 Reglas de negocio	42
1.9.1 Modelo vista controlador	42
1.9.2 Arquitectura de aplicaciones Web	45
2. Propuesta de la Aplicación	46
2.1 Entorno y Fundamentos	46
2.2 Propuesta de la Aplicación	47
2.3 Requerimientos Funcionales	48
2.4 Requerimientos no Funcionales	48
2.5 Objetivo General	49
2.6 Objetivos específicos	49
2.7 Beneficios	49
2.8 Alcance	50
2.9 Entregables	50
3. Diseño de la solución	51
3.1 Requerimientos de hardware y software	51
3.2 Herramientas necesarias	54
3.3 Diagrama Entidad / Relación	55
3.4 Actores del sistema	56
3.5 Diseño del sistema con UML	57
3.6 Desglose de los casos de uso	58
3.7 Diagrama de Clases	64
3.8 Diagrama de Secuencia	65
3.9 Diccionario de Datos	66
3.10 Script de la Base de Datos	70
3.11 Mapa de Navegación	74
3.12 Interfaz del Sistema	75
4. Posibilidades de Integración y crecimiento del Sistema	79
4.1 Presente y futuro de Internet	79
4.2 Tendencias del Software Libre	81
4.3 Tiempos de desarrollo	83
4.4 Etapa de Pruebas	84
4.5 Posibilidades de Integración	86
4.6 Crecimiento del Sistema	88
Conclusiones	89
Bibliografía	91

DIPLOMADO
“DESARROLLO DE SISTEMAS EN WEB (WWW)”

Objetivo:

Proporcionar a los participantes el conocimiento y habilidad requeridos para analizar, diseñar e implementar aplicaciones de uso interactivo y con intercambio de datos a través de una interfase www, de tal forma que se ofrezca una ventaja competitiva con respecto a los otros entornos donde no existe el www.

Temario

Módulo I. Fundamentos del Desarrollo de Sistemas en Web

Módulo II. Análisis y Diseño con UML

Módulo III. Introducción a la programación Orientada a Objetos con Java

Módulo IV. Capa de Usuario

Módulo V. Capa de Datos

Módulo VI. Reglas del Negocio

Módulo I

FUNDAMENTOS PARA EL DESARROLLO ORIENTADO AL WEB (WWW)

Dentro de este capítulo se dieron a conocer los conceptos más importantes referentes al desarrollo de sistemas orientados al Web.

Entre los conceptos más importantes expuestos en el diplomado se encuentra la teoría básica para entender el desarrollo de un sistema orientado al Web y las tecnologías más importantes que existen para desarrollar dichos sistemas.

De los conceptos más significativos presentados en el diplomado se encuentran los siguientes:

Internet es una red mundial de equipos de cómputo que se comunican mediante un lenguaje común conocido como protocolo de red.

Direccionamiento IP es un protocolo de la capa de red encargado del proceso de ruteo de los paquetes a través de la red. Todas las direcciones IP tienen un formato básico y no pueden ser tomadas arbitrariamente. La dirección se divide en dos partes: el número de red y el número de host, tiene una longitud de 32 bits, esta formada por 4 segmentos de bits cada uno y separados por puntos.

De la misma manera se trataron los servicios que proporciona Internet como World Wide Web, el cual brinda información a usuarios de cualquier parte del mundo; otros de los servicios que proporciona como el correo electrónico, foros de discusión, servicios de transferencia de archivos FTP, Servicios de noticias, etc.

Se dio a notar que para el desarrollo de un sistema es fundamental el conocimiento del sistema operativo sobre el cual se implantara el sistema; entre los cuales podemos nombrar Windows y Unix.

Se presentaron opciones gráficas para Linux, en este caso “KDE” y “Gnome”. KDE es uno de los entornos gráficos más populares pues posee un entorno agradable y cómodo muy parecido al entorno de Windows.

Por otro lado uno de los objetivos era tener un panorama acerca de los elementos que se pueden aplicar al Diseño de Sistemas Orientado al Web. Por lo cual se presentaron dentro del diplomado las tecnologías más utilizadas en el desarrollo de sistemas orientados al Web.

Igualmente se trató uno de los programas importantes en la transferencia de información utilizado como auxiliar y de mayor confiabilidad es *Secure Shell* (SSH) que admite realizar conexiones entre maquinas a través de una red abierta de forma segura, pues copia archivos de una máquina a otra.

Este programa provee una fuerte autenticación y comunicación segura sobre un canal inseguro y nace como un reemplazo de los comandos telnet, ftp, rlogin, rsh y rcp, los cuales proporcionan gran flexibilidad en la administración de una red, aunque sin embargo no son confiables en cuanto a seguridad.

Se presentaron algunos elementos de software que se utilizan para el desarrollo de un proyecto en World Wide Web. Entre estas herramientas se encontraron tecnologías de Software Libre, de los cuales se estudió sus características, configuración e instalación de los mismos, aplicaciones tales como: Apache Tomcat, MySQL, Java, JVM, Eclipse, entre otras.

Un DBMS (DataBase Management System) Sistema Manejador de Base de Datos, es un software que se encarga de gestionar datos de una o más aplicaciones. Entre los más conocidos se encuentran: SQL-SERVER, ORACLE (el mejor servidor de BD Objetos-Relacional), MYSQL (bases de datos Relacional), POSTGRESSSQL(es un servidor de bases de datos de gran potencia).

La función de SQL es la de soportar la definición, manipulación y control de los datos de una base de datos. Además permitir realizar vistas de acuerdo a las necesidades del usuario, este lenguaje permite la creación de una base de datos robusta. Así como crear consultas para manipular bases de datos remotas cliente - servidor.

La utilización de Java en el desarrollo de Sistemas es conveniente porque integra tres características de suma importancia: la seguridad, la escalabilidad y la portabilidad. En relación al acceso de datos Java cuenta con un estándar de acceso a los datos, llamado JDBC (Java DataBase Connectivity), el cual se comporta como los controladores ODBC. La aplicación de programación Java permite desarrollar un sistema independiente, con objetos reutilizables, lo cual dará como resultado un sistema eficaz, útil, fácil de administrar y confiable.

El conocer el software de uso libre a través de este módulo, garantiza un Sistema confiable, seguro, funcional y con un costo de desarrollo e implementación más bajo del que se tendría usando software propietario, además de que se asegura la compatibilidad en diferentes plataformas, tanto de lado del servidor como del cliente.

Módulo II

ANÁLISIS Y DISEÑO CON UML

El objetivo de este módulo fue comprender lo importante que es la etapa de análisis y diseño de un sistema. Así como también conocer las metodologías que se utilizan hoy en día para desarrollar esta etapa.

El propósito fundamental fue conocer UML (*Unified Modeling Language*), y obtener como productos finales los Diagramas de la Estructura de cualquier Sistema a diseñar.

En esta etapa es fundamental establecer la arquitectura del mismo antes de su construcción, es decir, especificar completamente los módulos que tendrá el software, la relación que existirá entre ellos y la interfaz con el usuario. Ya que a partir de la documentación del diseño, los programadores podrán construir y probar el software. La documentación del sistema es parte importante ya que garantiza la posibilidad de efectuar correcciones y mejoras al software en el futuro.

Dentro de este apartado se expusieron diversos análisis, los cuales se deben de aplicar al proceso de desarrollo de un software, como son: análisis de costos, en el cual se debe de verificar los gastos que implica desarrollar un nuevo sistema o innovar el actual; análisis de Tecnología para visualizar el tipo de Sistema Operativo, determinar herramientas DBMS que existen, etc.; Análisis de Beneficios que se maneja para examinar los aspectos positivos y negativos de automatizar un proceso; Análisis de Seguridad para verificar que el usuario se encuentre protegido y se proporcionen niveles de seguridad al sistema; Análisis de Factibilidad, nos ayuda a identificar los usuarios responsables y el funcionamiento general del Sistema; así como también las deficiencias actuales en el ambiente del usuario.

Para obtener una documentación completa del Sistema, una vez completados los análisis, es necesario contar con Diagramas como el Diagrama de Flujo de Datos (*D.F.D*), que es la representación gráfica que permite al analista definir entradas, procedimientos y salidas de la información, permitiendo así comprender los procedimientos existentes con la finalidad de optimizarlos.

Otro diagrama importante es el D.E.R. (Diagrama Entidad Relación) en el que se representa la relación de las entidades. Después de obtener los diagramas anteriores es posible generar el Diccionario de Datos mediante un software específico, en este caso por medio del software ERWIN.

Consecutivamente se procede al Modelado del Sistema, esto puede ser utilizando UML (Unified Modeling Language) el cual es un lenguaje que permite modelar, construir y documentar los elementos que forman un Sistema.

El UML cuenta con distintos modelos, los cuales representan a un Sistema desde diferentes perspectivas. Por ejemplo el Diagrama de Casos de Uso, representa los actores (representados por muñecos) y sus relaciones (óvalos). El Diagrama de Estados se utiliza para mostrar los estados, eventos, transiciones y actividades de los diferentes objetos. También existen los diagramas de Secuencia, es útil para observar la vida de los objetos en el sistema, identificar llamadas a realizar o posibles errores del modelado estático, que imposibilite el flujo de información o llamadas entre componentes.

Algunas de las ventajas que nos ofrece el UML, son que nuestro sistema de software es diseñado y documentado antes de que sea codificado, por ende sabremos exactamente los que conseguiremos, por adelantado. Así como también nos permite visualizar el código reutilizable y evita el tropezarse con errores por lo que nos permite ahorrar tiempo en el desarrollo.

El implementar el Lenguaje UML en el análisis de Sistemas es independiente del lenguaje de programación y de las características de los proyectos, este lenguaje ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otro ramo.

A través de la notación UML podemos comunicar y compartir el conocimiento de una arquitectura gracias a la combinación simultánea de sus perspectivas. El resultado, es una mayor comprensión y claridad sobre la naturaleza de los objetos, eventos y hechos que tienen consecuencias dentro de un Sistema.

Módulo III

INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS CON JAVA

El propósito principal de este apartado fue conocer el lenguaje Java, sus características y beneficios que proporciona al aplicarse en los sistemas Orientados al Web y conocer como se puede aplicar este lenguaje a través de ejemplos.

Para comprender la programación orientada a objetos (P.O.O) primeramente se debe definir un objeto, que son las entidades que se pueden distinguir de manera clara y definida dentro de un lenguaje.

Ésta programación tiene varias características entre las que se pueden mencionar: el Encapsulamiento, significa que el objeto es auto-contenido, o sea que la misma definición del objeto incluye tanto los datos que éste usa (atributos) como los procedimientos (métodos) que actúan sobre los mismos; la Herencia, que permite compartir atributos con otros objetos; el Polimorfismo, que se refiere a la posibilidad de construir varios métodos con el mismo nombre pero con relación a la clase a la que pertenece cada uno, con comportamientos diferentes.

Debido a que Java permite aprovechar los servicios que son ofrecidos en el Internet y gracias a que el código compilado de Java (conocido como byte code) es interpretado por una máquina virtual; un programa compilado de Java puede ser utilizado por cualquier computadora que tenga instalado el interprete de Java JVM. Ya que puede funcionar como una aplicación sola o como un “applet” (que es un pequeño programa hecho en Java). Los applets de Java se pueden pegar a una página Web (html), y con esto se puede tener un programa que cualquier persona que tenga un navegador de Internet compatible podrá usar; Java tiene ciertas políticas que evitan la codificación de virus en este lenguaje. Otra de las características de este lenguaje es que existen muchas restricciones en cuanto a seguridad (especialmente para los applets) que limitan lo que se puede y no puede hacer con los recursos de una computadora.

Java ha venido a revolucionar la computación de nuestros días, debido a que integra diferentes características de suma importancia; como por ejemplo proporciona seguridad, escalabilidad y portabilidad.

La Máquina Virtual Java (*JVM “Java Virtual Machine”*) es en sí el núcleo del lenguaje de programación Java. De hecho, es imposible ejecutar un programa Java sin ejecutar alguna implantación de la JVM. En la máquina virtual de Java se encuentra el

motor que en realidad ejecuta el programa Java y es la clave de la independencia de los programas Java sobre el sistema operativo y el hardware en que se ejecutan, pues facilita la verificación estricta de tipos de datos, la utilización de múltiples hilos, un ligado dinámico y recolección automática de basura (limpieza de la memoria de la computadora).

Con Java se pueden construir varios tipos de programas, cada uno con unas características específicas, y que se ejecutan de distintas maneras. A continuación se explican las siguientes:

- Aplicaciones Standalone. Son programas básicos de Java, se ejecutan por el Java Runtime Environment.
- Servlets. Son módulos que permiten sustituir o utilizar el lenguaje Java en lugar de programas CGI, se ejecutan como aplicaciones servidores en Internet.
- JSP. Contienen elementos Html y código de programación escrito en Java.

El lenguaje Java actualmente es uno de los más utilizados alrededor del mundo, dado que sus características permite desarrollar software con gran calidad, comparándolo con otros lenguajes.

Módulo IV

CAPA DE USUARIO

Se abarco la teoría referente a la capa de usuario, quienes la conforman, como se maneja, etc.

Un sistema desarrollado para la Web con metodología de tres capas (*Three Tier Layer*) tiene tres componentes: Cliente, Servidor y Base de Datos.

El cliente es la computadora del usuario, el cual accede a los servicios del navegador, el servidor es el sistema que realiza la lógica de negocios y los cálculos, así como la comunicación entre los mismos. La Base de datos mantiene los datos almacenados en un sistema relacional de base de datos.

La ventaja de esta arquitectura es la independencia de una aplicación cliente, ya que el acceso se realiza a través del navegador, la presentación de los datos es con HTML y la comunicación entre el cliente y el servidor se realiza a través del protocolo http.

HTML es el lenguaje que se utiliza en la creación de páginas Web e interfaz de Sistemas Web, ya que tiene como características ser un lenguaje sencillo y con gran potencial.

A la presentación de datos a través de una pantalla se le conoce como interfaz, y tiene como característica que desde donde el cliente interactúa con la información la presentación se realiza por medio del navegador en html. Por tanto la interfaz es el medio por el cual el usuario puede hacer todas sus operaciones a la base de datos.

La finalidad de este módulo fue conocer la estructura de la interfaz del sistema sobre la cual el usuario interactúa con el Sistema. Para comenzar el diseño de la interfaz es necesario tener en cuenta quién será nuestro usuario final y para que utilizará el sistema en cuestión, ya que la interfaz debe contar con los elementos y secuencias necesarios para que el usuario pueda realizar adecuadamente sus procesos sobre la base de datos.

Primeramente se debe conocer que tipo de cliente es al que esta dirigido el sistema, ya que el tipo de web que se va a diseñar pueden ser de tipo empresarial, comercial, social o híbrido.

Además se debe tomar en cuenta el estándar o normatividad de la empresa u organización, se debe conocer cuales son los colores corporativos de la empresa, si cuenta con un logo, leyenda o slogan publicitario.

El establecer los niveles de usuario, sirven para establecer los tipos de acceso y operaciones que pueden realizar los diferentes usuarios.

Dentro de este módulo se vio Html es óptimo para el desarrollo de la interfaz, utilizando editores visuales, los cuales permiten diseñar sin la necesidad de escribir ni una línea de código. Ejemplo de ellos son: *Dreamweaver MX*, *Front Page*, etc. Además su compatibilidad admite cualquier tecnología, como por ejemplo: *Active Server Pages*, *JavaScript*, *PHP*, *XML*, *ISAPI*, entre otros.

El diseño de la interfaz se basa principalmente en los Diagramas de Casos de Uso y de Secuencia, de estos diagramas se deben de concluir los niveles de usuario que el sistema utiliza, así como los niveles de acceso.

El realizar un mal diseño en los niveles de usuario afectan la seguridad Web, siendo los más vulnerables las sesiones y los clientes.

Se debe de clasificar a que tipo de Web esta dirigido nuestro Sistema, los sitios Web pueden ser de tipo empresarial, comercial, social e híbrido. Por lo tanto de acuerdo al tipo se pueden definir los colores que se pueden utilizar, o que tan seria o llamativa debe de ser.

La usabilidad de la interfaz depende de varias características: primeramente debe ser amigable, llamar la atención del usuario, contar con una distribución lógica en los formularios, navegación lógica, retroalimentación, consistencia, e implementación de ayuda.

La estructura de las paginas debe ser clara, funcional y con una continuidad, es conveniente que la longitud de las paginas no exceda una resolución de 800X600 *pixels* o en su correspondiente porcentaje, debido a que si los excede en la mayoría de los monitores no se vera adecuadamente.

La estructura que se sugiere en un sistema Web en los menús es de tipo vertical con opciones, esta estructura se debe tomar de los procesos identificados.

Debido a que a los usuarios se le hace tedioso llenar formularios se recomienda que estos tengan una distribución adecuada y que no se exceda en el uso de elementos, por ejemplo: en las casillas de verificación deben de existir como máximo 16, las cajas de selección no deben de exceder de tres en sus opciones, etc.

Es recomendable que se incluya dentro de la interfaz una ayuda, la cual se basará sobre todo en la operatividad y la aclaración de funciones. El diseñar una interfaz con basándose en los diagramas de casos de usos nos dará como resultado una interfaz clara, fácil y funcional, en la que el usuario final podrá realizar adecuadamente sus operaciones.

Módulo V

CAPA DE DATOS

El propósito del módulo es conocer los lenguajes que existen, así como también las herramientas que existen para diseñar la Base de Datos. Al terminar será posible implementar el lenguaje SQL en el diseño de la capa de Datos.

Una base de datos es un conjunto de objetos que se interrelacionan entre sí, y los objetos del mismo tipo constituyen una entidad.

Un Sistema Gestor de Base de Datos (SGBD) tiene la función de permitir la definición de todos los datos, manipularlos, establecer controles para la seguridad de estos datos y permitir los accesos concurrentes.

Los ciclos de una base de datos son 3: la “Concepción”, esta fase consiste en reproducir con ayuda de uno de los modelos de datos (relacional) el esquema escrito de la base de datos; la segunda etapa la “Creación” radica en traducir el esquema en ordenes comprensibles para el SGBD; y por último la etapa de la “Explotación” aquí los datos son manipulados por los usuarios por medio de lenguajes de programación.

Desde la creación de la base de datos se pueden implementar algunas reglas del negocio, por medio de triggers o procedimientos almacenados, lo cual beneficiara al sistema ya que a través de esto se podrá tener una integridad de datos.

Para la implementación de la base de datos existen diferentes opciones como SQL Server y MySql, los dos muy potentes y con muchas características en común. También se utilizó DB2, por sus características también es un DBMS muy potente.

Parte importante de este apartado fue conocer como instalar, manejar y crear una base de datos. Igualmente se implementaron las reglas de negocio mediante triggers y stored procedures.

Módulo VI

REGLAS DE NEGOCIO

Las reglas de negocio o *business rules* se encargan de que en la aplicación se refleje las restricciones que existen en el mundo real, de forma que no se puedan llevar a cabo acciones que no son válidas.

Ejemplos de tales reglas son: no permitir crear facturas pertenecientes a clientes inexistentes, controlar que el saldo negativo de un cliente nunca sobrepase cierta cantidad, etc.

El objetivo del módulo era dar a conocer la importancia de las reglas del negocio y aplicarlas en el diseño del sistema. Obteniendo como resultado un sistema apegado a la realidad y funcional.

Se dieron a conocer las reglas de negocio y su clasificación. Se sugiere que para dar la ubicación correcta a las Reglas del Negocio se defina a que tipo de regla pertenece; ya que de acuerdo a su tipo es su ubicación.

Sin embargo las reglas del negocio no tienen un lugar en específico en el cual ubicarse, sin embargo existen algunas sugerencias tomando en cuenta el tipo de regla en la que se está clasificando.

Entre las reglas que podemos mencionar se encuentran las del modelo de datos, las cuales se encargan de verificar que solo se validen los datos para cada campo de cada tabla. La ubicación que se sugiere es el servidor, así como también para las reglas de Relación.

En lo que respecta a las reglas de Restricción se deben de implementar en la capa intermedia; ya que contemplan restricciones en los datos que dependen casi siempre de información que se encuentra en varias tablas.

En cuanto a las reglas de flujo es una excelente opción implementarlas en la capa intermedia, ya que son muy extensas, lo cual hace que no se pueda manejar por un gestor de base de datos.

Las reglas del negocio tienen como objetivo ser independientes del sistema, ya que esto lo beneficiara, debido a que si se necesita realizar algún cambio a nivel de la base de datos o la programación esto no altere nuestras reglas ya establecidas.

En resumen este último módulo sirvió para aplicar todos los conocimientos adquiridos dentro del diplomado, así como también para conocer y aplicar las reglas del negocio al sistema.

INTRODUCCIÓN

Debido a que la Web es uno de los servicios que provee la red de redes “*Internet*”, y uno de los más importantes en cuanto a impacto y uso, es indispensable implementar el desarrollo de sistemas orientados hacia el *World Wide Web (WWW)*.

Actualmente muchas empresas ofrecen sus productos y/o servicios por medio de WWW, es por esto que las empresas deben contar con un sistema bien diseñado, el cual les permita cubrir todas las necesidades del cliente.

En este trabajo se propone un sistema orientado al Web, en el cual se utilizan tecnologías de software libre, ya que dicho software ha sido utilizado para el desarrollo de sistemas teniendo resultados positivos.

En el capítulo I. Se describe brevemente los conceptos para el entendimiento y desarrollo del sistema, se tratarán las tecnologías que están involucradas en el desarrollo del sistema como por ejemplo: Arquitectura de tres capas, HTTP, HTML, lenguaje de consulta SQL, JAVA, Apache Tomcat, entre otros.

En el segundo capítulo, se detallará la problemática a resolver, se darán a conocer los requerimientos del usuario, el alcance del sistema, los beneficios y los posibles productos a generar.

Tercer capítulo “Diseño de la Solución” se refiere al diseño del sistema orientado al Web, desde el análisis de requerimientos, diagramas de la arquitectura del sistema, se proponen las herramientas de software, así como también la interfaz de usuario, y la creación de un prototipo.

Capítulo IV, trata sobre las posibilidades que tiene el sistema de integrarse al mercado, así como también las posibilidades de desarrollo que tendrá el mismo y el trabajo a futuro o mejoras que se pueden hacer al sistema.

CAPITULO I. ANTECEDENTES

Debido a que se desea diseñar un sistema orientado a la Web, es necesario dar a conocer los conceptos más importantes sobre Internet, Arquitectura Cliente/Servidor, bases de datos, etc. Posteriormente se darán a conocer algunas de las tecnologías de uso libre utilizadas para el desarrollo de sistemas orientados al Web. Asimismo se presentarán los beneficios y características de cada uno de los programas que se implementaran en el desarrollo de sistema en cuestión. Así como también los conceptos relacionados al diseño de sistemas, los cuales se implementarán en capítulos siguientes, como lenguajes de programación, servidores, etc.

1.1 INTERNET

El propósito original de Internet fue de interés militar, en el posible caso de una Guerra Nuclear se necesitaba una forma de comunicación rápida y eficiente por medio de la cual se pudiera enviar información de forma segura.

La solución era una red compuesta por computadoras en la que todos los nodos tuvieran la misma importancia, de tal forma que la desaparición de uno de ellos no afectara al tráfico: cada nodo de la red decidiría qué ruta seguirían los datos que llegaran a él. Por último, los datos se dividirían en "paquetes", que podrían seguir distintas rutas, pero que deberían reunirse en el punto de destino.

El proyecto de Internet surge en los años 60's primeramente auspiciado por la agencia ARPANET (Advanced Research Projects Agency NETwork) del gobierno de los Estados Unidos.

Gracias a ARPANET, científicos e investigadores pudieron compartir recursos informáticos en forma remota; esto era una gran ayuda ya que en los años 70's el tiempo de procesamiento por computadora era un recurso realmente escaso.

De ahí es que surge Internet como la red de redes que permite establecer comunicación, compartir recursos, etc.

Internet utiliza como lenguaje común el protocolo de red IP el cual es un estándar para la comunicación de computadoras. "Protocolo IP es el encargado de enviar los paquetes de datos por la red desde el origen al destino, mediante la utilización de las direcciones".

Conforme evoluciono Internet empezó a brindar un conjunto de servicios que consisten en métodos de comunicación y entretenimiento. Entre los servicios que ofrece la red de redes se encuentra World Wide Web (WWW), el cual brinda información a usuarios de cualquier parte del mundo; también servicios como el correo electrónico, foros de discusión, servicios de transferencia de archivos *File Transfer Protocol (FTP)*, servicios de noticias, entre otros.

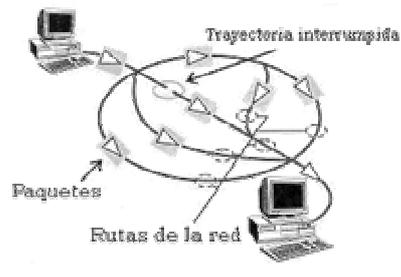


Fig 1. Internet

1.1.1 HISTORIA DE WWW

Por su parte la World Wide Web surge en 1989 cuando Tim Beners- Lee diseña un sistema para hacer accesible fácilmente la información, este sistema utiliza el hipertexto como base para estructurar una red de enlaces entre documentos. Así es como nace el primer navegador Web, llamado World Wide Web. Otro de los primeros navegadores fue Mosaic, el cual más tarde se convertiría en Netscape.

A mediados de los noventas el Internet tiene su auge, ya que se crearon empresas que proporcionaban el servicio, con módems mucho más rápidos y a un menor costo, lo cual permitió que fuera accesible para una gran cantidad de usuarios.

1.1.2 ACTUALIDAD

Actualmente Internet es un conjunto de redes interconectadas entre si a nivel mundial, por medio del cual un usuario puede acceder a todo tipo de información de otra computadora y tener comunicación directa con otros usuarios.

El intercambio de información vía Internet se ha convertido en un fenómeno cultural, lo cual ha llevado a que este evolucione de forma vertiginosa. Durante los últimos años, Internet se ha hecho accesible para la gran mayoría de personas, siendo uno de los medios de comunicación más importante de los últimos siglos.

La evolución de Internet ha permitido incorporar nuevas tecnologías como los servicios inalámbricos de banda ancha en los aviones, conexiones a Internet por medio de teléfonos celulares (agregar mas servicios innovadores de la Internet), lo cual permite que los usuarios que navegan por la Web aumente considerablemente.

A partir de 1994 el crecimiento de la WWW fue constante lo cual permitió que a finales de los noventas se convirtiera en el servicio insignia de Internet dando lugar a servicios en línea como los que se están desarrollando actualmente.

Debido a esto Internet actualmente se ha convertido en una oportunidad de negocio para las empresas. A través de uno de sus servicios World Wide Web “WWW” se esta creando un nuevo mercado mediante el cual se puede realizar diferentes transacciones al mismo tiempo y sin la necesidad de trasladarse de un lugar a otro. Lo cual representa un gran ahorro en cuanto a tiempo y (dinero) tanto para el cliente como para la empresa, ya que mantener un sistema orientado al Web es mucho más económico que pagar publicidad en algún otro medio de comunicación.

1.1.3 SISTEMA ORIENTADO AL WEB

Un sistema desarrollado para la Web, es una aplicación robusta, lo que significa que implica un análisis, diseño y desarrollo como cualquier sistema, ya que este debe ser muy sólido, puesto que el acceso desde la Web lo hace más vulnerable.

Este tipo de sistemas dispone de una lógica de negocio la cual puede reaccionar y alterar su estado a partir de la interacción con varios usuarios. Un sistema de esta índole además puede abarcar a cientos de usuarios simultáneamente.

Una de las ventajas que proporciona el crear sistemas Orientados al Web es que las empresas podrán darse a conocer con miles de usuarios, así como de ofrecer sus servicios. La creciente popularidad de las aplicaciones Web se debe a sus múltiples ventajas, entre las cuales podemos citar:

Multiplataforma: Con un solo programa, un único ejecutable, las aplicaciones pueden ser utilizada a través de múltiples plataformas, tanto de hardware como de software.

Actualización instantánea: Como se hace uso de un sólo programa que radica en el servidor, los usuarios siempre utilizarán la versión más actualizada del sistema.

Aprendizaje: Los usuarios, como utilizan la aplicación a través de un navegador, hacen uso del sistema tal como si estuvieran navegando por Internet, por lo cual su uso es más intuitivo.

Fácil de integrar con otros sistemas: Ya que se basa en protocolos estándares, la información manejada por el sistema puede ser accedida con mayor facilidad por otros sistemas.

Acceso móvil: El usuario puede acceder a la aplicación con la única restricción de que cuente con un acceso a la red privada de la organización o a Internet, dependiendo de las políticas de dicha organización; puede hacerlo desde una computadora de escritorio, una laptop o desde un PDA¹; desde su oficina, hogar u otra parte del mundo.

Disponibilidad de Información: La construcción de un sistema Web permite tanto al usuario como al dueño de la aplicación contar con la información en cualquier momento y en cualquier lugar.

Automatización de Procesos: Esto significa que los procesos que realiza el sistema serán de gran ayuda para mejorar el rendimiento de la empresa, realizando procesos de forma rápida, eficaz y con un mínimo de errores.

¹ PDA. "Personal Digital Assistant" son considerados pequeños ordenadores de bolsillo tremendamente potente y capaz de realizar numerosas funciones.

1.2 ARQUITECTURA DE TRES CAPAS

En las aplicaciones diseñadas según un modelo de tres capas, el sistema es dividido en datos, lógica de negocio y presentación.

1.2.1 CAPA DE DATOS

La capa de datos representa el mecanismo por el cual se manipula y persiste la información. Consiste en un administrador de bases de datos relacional (DBMS), y el esquema de datos propio de cada aplicación.

El nivel de servicios de datos es responsable de:

- Recuperar los datos.
- Almacenar los datos.
- Mantener los datos.

1.2.2 CAPA DE NEGOCIO

En la capa de lógica de negocio se modela el comportamiento del sistema, basándose en los datos provistos por la capa de datos, y actualizándolos según sea necesario. Esta capa describe los distintos procesos de negocio que tienen lugar en las organizaciones, desde el ciclo de aprobación de un documento hasta la política de descuentos para un pedido.

En esta capa se encuentran las reglas y lógica de procedimientos necesarios para realizar las operaciones del sistema, es decir actividades operacionales, validaciones, cálculos, etc. 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 a automatizar.
- Enviar el resultado procesado al nivel de presentación

1.2.3 CAPA DE PRESENTACIÓN

Finalmente, la capa de presentación contiene todos los elementos que constituyen la interfaz con el usuario. Esta capa incluye todo aquello con lo que el usuario puede interactuar, como por ejemplo las pantallas de las aplicaciones, el modelo de navegación del sistema.

El cliente proporciona el contexto de presentación, generalmente un browser como Microsoft Internet Explorer, Netscape Navigator, Firefox, Mozilla, etc. que permite ver los datos remotos a través de una capa de presentación html.

La capa de servicios de presentación es responsable de:

- Obtener información del usuario.
- Enviar la 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.

La ventaja de esta arquitectura es la independencia de una aplicación cliente, pues el acceso se realiza a través del navegador, la presentación de los datos es con html y la comunicación entre el cliente y el servidor se realiza a través del protocolo http.

Los diseños "*three tier*" son ampliamente utilizados en el mercado, y a lo largo del tiempo han probado sus ventajas. Las aplicaciones en tres capas típicamente tienen mayor capacidad de crecimiento y son más sencillas de mantener, dada su naturaleza altamente modular.

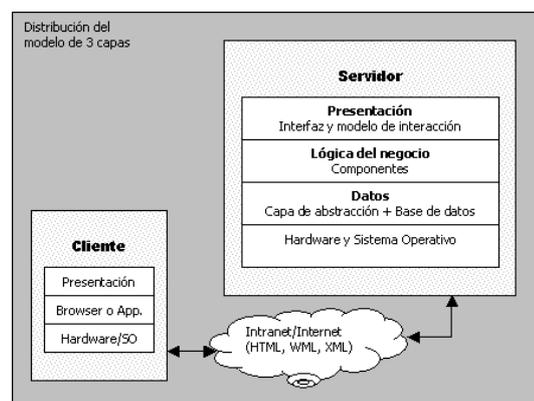


Fig 2. Arquitectura de 3 capas

1.3 BASE DE DATOS

En el desarrollo del Sistema se utilizarán tecnologías de Software Libre por lo cual es esencial el conocimiento de la configuración e instalación de los mismos, en este caso al desarrollar sistemas orientados al Web es necesario elegir o disponer de una base de datos potente, robusta y rápida. Es por ello que se presentan a continuación las características más importantes de un BD.

Existen diferentes definiciones de Base de Datos, aquí se exponen algunas: “Una Base de Datos es un conjunto integrado de datos interrelacionados entre sí.”² “Una Base de Datos es un conjunto exhaustivo, no redundante de datos estructurados, organizados independientemente de su utilización y su implementación en máquina, accesible en tiempo real y compartibles por usuarios concurrentes.”³

1.3.1 CICLO DE VIDA

Una Base de Datos, consta de 3 etapas: Concepción (consiste en un esquema escrito), Creación de BD (esta fase traduce el esquema en ordenes comprensibles para SGBD), Explotación (los registros son manipulados con ayuda de lenguajes de programación).

1.3.2 DBMS

Los sistemas Gestores de Bases de Datos, son un software que se encarga de servir como interfaz entre la base de datos y las aplicaciones. Las principales funciones de un DBMS son: permitir la definición de todos los datos, manipular datos, establecer controles para la seguridad de estos datos, permitir accesos concurrentes. Entre los DBMS de uso libre se encuentran: Postgre SQL, Firebird, SQLite, y MySQL. Los DBMS más populares en la actualidad se encuentran:

- Oracle – Considerado el mejor servidor de base de datos Objeto – Relacional.
- SQL Server – Es un servidor de base de datos de fácil manejo de Microsoft.
- MySQL – Es el servidor de base de datos relacional más popular y fácil de manejar.
- PostgreSQL – Considerado un servidor de base de datos de gran potencia.

² Elmarsi, R, Navathe, S.B. 1989

³ Howe, 1983

1.3.3 MODELO DE DATOS

Es un conjunto de conceptos que sirven para describir la estructura de una base de datos. Los modelos de datos se pueden dividir en 3 grupos:

- Modelo basado en objetos, el cual se utiliza para describir los datos en nivel conceptual, ejemplo de este el Modelo Entidad – Relación.
- Modelo lógico basado en registros. Se utilizan para describir datos en los niveles conceptual y físico.
- Modelos físicos de datos. Se usan para describir a los datos en el nivel más bajo, básicamente capturan aspectos de la implementación de los sistemas de base de datos.

1.3.4 MYSQL

MySQL es un sistema de administración de bases de datos relacional, licenciado bajo la GPL⁴. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

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

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

La parte SQL de MySQL significa Lenguaje Estructurado de Consulta y es en general el lenguaje más usado y estandarizado de acceso bases de datos relacionales.

⁴ GPL. General Public License o licencia pública general, es una licencia creada por la Free Software Foundation a mediados de los 80, y esta orientada a los términos de distribución, modificación y uso de software.

1.4 LENGUAJE SQL

El Lenguaje de Consultas Estructuradas o SQL (Structured Query Lenguaje), es un lenguaje que se utiliza para interactuar con una base de datos y más concretamente con un tipo específico llamado base de datos relacional.

SQL se utiliza para organizar, gestionar y recuperar datos almacenados en una base de datos. Este lenguaje tiene como característica que es fácil de aprender y una herramienta completa para gestionar datos.

El lenguaje SQL está compuesto por comando, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos, por lo cual existen dos tipos de comandos: DDL permite crear y definir nuevas bases de datos, campos e índices y DML permite generar consultas para ordenar, filtrar y extraer datos de la base de datos.

Una de las sentencias más importantes dentro del Lenguaje SQL es la sentencia “select”, la cual se utiliza para realizar consultas. Esta sentencia recupera los datos de una base de datos. Su sintaxis es la siguiente: SELECT Campos FROM Tabla.

1.4.1 FUNCIONES DE AGRUPAMIENTO

SUM Devuelve la suma total de los valores de una expresión de columna o campo numérica.

AVG Devuelve la medida de los valores en una expresión de columna

COUNT Devuelve el número de valores en una expresión de columna

MAX Devuelve el valor más alto de los contenidos en una expresión de columna

MIN Devuelve el valor mas bajo de los contenidos en una expresión de columna

1.4.2 CLAUSULAS

FROM.- Lista de tablas que contienen los datos a recuperar por la consulta.

WHERE.- Se dice a SQL que incluya solo ciertas filas o registros de datos en los resultados de la consulta es decir, que tienen que cumplir los registros que se desean ver.

ORDER BY.- Ordena los resultados en base a los datos de una o mas columnas. Si se omite, los resultados saldrán ordenados por el primer campo que sea clave en el índice que se haya utilizado

GROUP BY.- Especifica una consulta sumaria. En vez de producir una fila de resultados por cada fila de datos de la base de datos, una consulta sumaria agrupa todas las filas similares y luego produce una fila sumaria de resultados para cada grupo.

Seguido de la clausura **GROUP BY** se especifican los nombres de uno o más campos cuyos resultados se desean agrupados.

HAVING.- Esta instrucción se encarga de incluir solo ciertos grupos producidos por la cláusula **GROUP BY** en los resultados de la consulta. Al igual que la clausura **WHERE**, utiliza una condición de búsqueda para especificar los grupos deseados. En otras palabras, especifica la condición que deben de cumplir los grupos. Sólo es valida si previamente se ha especificado la cláusula **GROUP BY**.

Operador Unión.- combina el resultado de dos sentencias **SELECT** en un único resultado. Este resultado se compone de todos los registros devueltos en ambas sentencias. Por defecto, los registros repetidos se omiten. Para no quitarlos se emplea la palabra **ALL**.

Cuando se utilice el operador **UNION**, la lista de selección para cada sentencia **SELECT** debe tener el mismo número de expresiones de columnas con el mismo tipo de datos y en el mismo orden.

Para realizar una consulta de la unión de dos tablas o más, se utiliza la cláusula **JOIN**. El **JOIN** se indica en la clausura **WHERE** como otra condición más, en este caso, el número de **CLAVES** en las dos tablas debe existir, y si no es así, el **ACTIVO** no aparecerá en la relación, aunque figure en uno de las dos.

Alias de las Tablas.- son un instrumento para abreviar los nombres de las tablas o ficheros y poder referirse a ellos en todas las sentencias.

Como en la mayoría de los lenguajes aquí también podemos encontrar los operadores lógicos, aritméticos y de fechas.

1.4.3 FUNCIONES

Las funciones permiten realizar con los datos operaciones adicionales. Una función representa un valor único que se obtiene aplicando determinadas operaciones a otros valores dados que se llaman argumentos. Algunas de las funciones son:

CHR. Convierte un código ASCII en una cadena de carácter

RTRIM o TRIM. Quita los blancos que existan por la derecha en una cadena

LTRIM. Quita los blancos por la izquierda que tenga una cadena

UPPER. Convierte cada letra de una cadena a mayúscula

LOWER. Convierte a minúscula cada letra de una cadena

Funciones que devuelven fechas

SYSDATE. Devuelve la fecha de hoy (SISTEMA)

TO_DATE. Convierte una cadena de caracteres a una fecha

TO_DATE. ('01/30/89', 'DD,MM/YY')

1.4.4 CREACION DE UNA TABLA

La sentencia para crear una tabla tiene la siguiente sintaxis:

```
Create table nombre_tabla (definicion_columna,...)
```

Los nombres de columna no deben tener espacios en blanco y los tipos de datos disponibles son:

- NUMBER.- Este tipo puede almacenar valores numéricos, tanto enteros como reales, en formato decimal.
- FECHA.- Se almacena la información del año, mes, día, hora, minuto, segundo. CARÁCTER.- CHAR Es de longitud fija (0<|<256)
- BOLEANO.- Este tipo puede almacenar los valores NULL, FALSE y TRUE.

Para la eliminación de una tabla se utiliza el siguiente formato:

```
DROP TABLE nombretabla
```

La Sentencia INSERT se utiliza para añadir registros a las tablas de la base de datos. El formato de la sentencia es:

```
INSERT INTO nombre_tabla [(nombre_columna,... )] VALUES (expr,...)
```

Otra sentencia importante es UPDATE se utiliza para cambiar el contenido de los registros de una tabla de la base de datos. En una misma sentencia pueden utilizarse varios campos de cada registro de la tabla. La sintaxis es la siguiente:

```
UPDATE nombre_tabla where (condición)
```

Sentencia DELETE ayuda a borrar registros de una tabla de la base de datos. Su formato es el siguiente:

```
DELETE FROM nombre_tabla[WHERE {condición}]
```

Para el diseño de la base de datos se debe tomar en cuenta los siguientes conceptos:

1.4.5 LLAVE PRIMARIA

Es el conjunto de atributos cuya unión no puede repetirse como contenido de una tabla. Una regla importante es que “no” se pueden descomponer; es decir, pasaran como llaves foráneas todos los atributos que forman la llave primaria.

1.4.6 LLAVE FORANEA

Este se forma cuando un campo en una tabla hace referencia a otra tabla, es decir se conforma por los atributos que forman una llave primaria de una tabla externa.

1.4.7 VISTA

Es un objeto que no contiene datos por si mismo, es un conjunto de datos tomados de otras tablas por medio de una consulta.

1.5 UML

UML (*Unified Modeling Language*) es parte importante para el desarrollo de nuevos sistemas. Como antecedentes debemos saber que fue impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Así como también es importante saber que en el proceso de creación han participado otras empresas como Microsoft, Hewlett-Packard, Oracle o IBM, analistas y desarrolladores.

El UML permite modelar, construir y documentar los elementos de un sistema, ya que cuenta con distintos modelos, los cuales representan a un Sistema desde diferentes perspectivas.

1.5.1 DIAGRAMA DE CASOS DE USO

Los casos de uso se emplean para visualizar el comportamiento del sistema, una parte de el o de una sola clase. El diagrama de uso es muy útil para definir como debería ser el comportamiento de una parte del sistema. Los casos de uso se representan por una elipse, cada caso de uso contiene un nombre, que indica su funcionalidad. Los actores se simbolizan por muñecos.

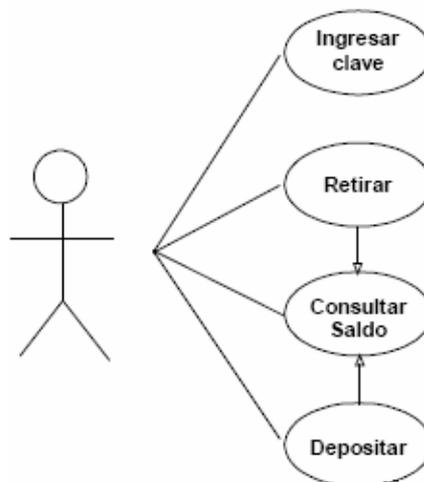


Figura 3. Diagrama de Casos de Uso

1.5.2 DIAGRAMA DE ESTADOS

Diagrama de Estados para mostrar la secuencia de los estados, eventos, transiciones y actividades de los diferentes objetos. En él se indican que eventos hacen que se pase de un estado a otro y cuáles son las respuestas y acciones que genera.

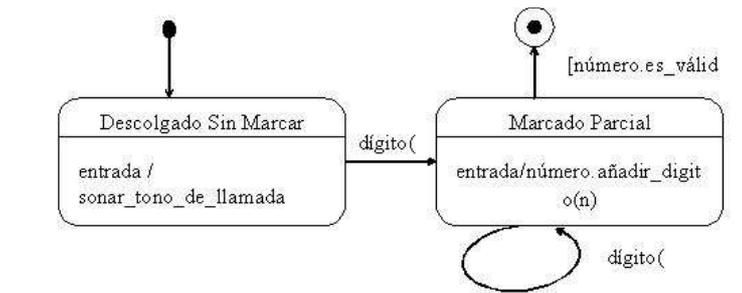


Fig 4. Diagrama de Estados

1.5.3 DIAGRAMA DE COLABORACIÓN

Los diagramas de Colaboración, muestran una serie de objetos con los enlaces entre los mismos y con los mensajes que se intercambian entre ellos.

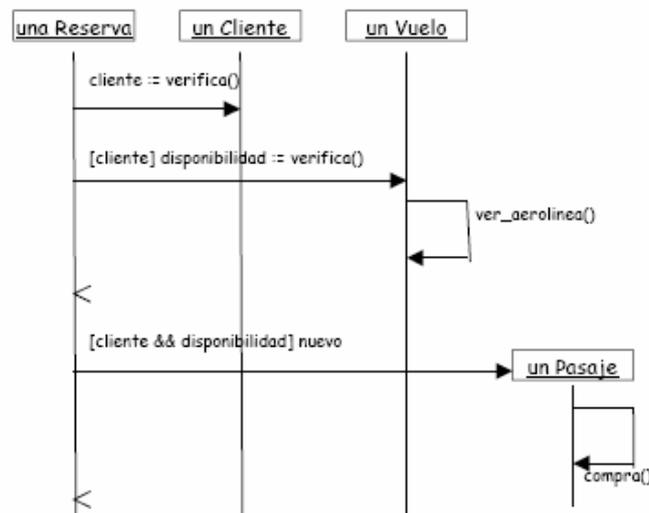


Fig. 5. Diagrama de Interacción

1.5.4 DIAGRAMA DE CLASES

Diagrama de Clases, aquí se definen las características de cada una de las clases, interfaces, colaboraciones y relaciones de dependencia y generalización

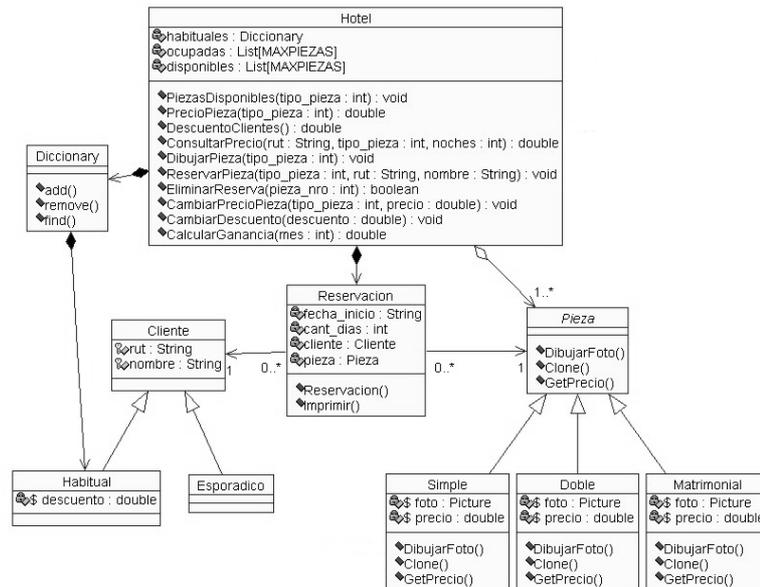


Fig 6. Diagrama de Clases

1.5.5 DIAGRAMA DE SECUENCIA

Diagrama de Secuencia ejemplifica los tiempos de los eventos; por otro lado el de Componentes, muestran la organización de los elementos físicos del sistema.

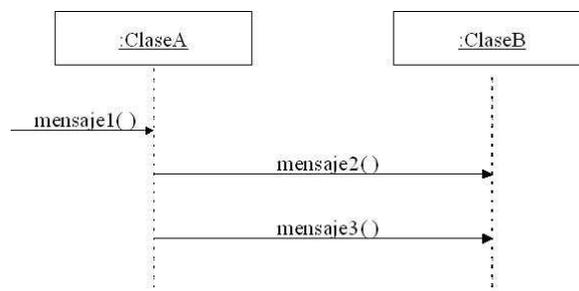


Fig 7. Diagrama de secuencia

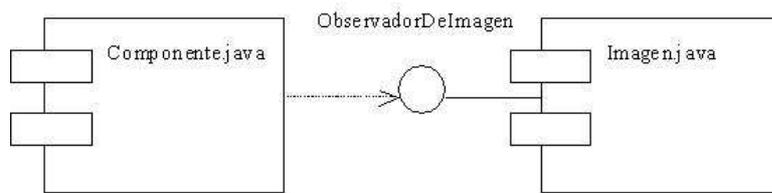


Fig 8. Diagrama de componentes.

Una de las ventajas que ofrece UML, es que al diseñar y documentar un sistema sabremos exactamente lo que conseguiremos antes de ser codificado. Nos permite visualizar el código reutilizable. Evita el tropezarse con errores por lo que nos permite ahorrar tiempo en el desarrollo.

Actualmente para la representación gráfica de los diagramas, podemos encontrar Herramientas de Diseño, como son *Rational Rose*, *Poseidon* que ofrece una interfaz amigable dirigida al desarrollo de diagramas de casos de uso, por otro lado ERWIN como un generador de Diagramas Entidad / Relación, y *Power Designer* que se puede aplicar para la creación del diccionario de datos a partir de un D.E.R.

Por ello se concluye que el crear documentación para todos los sistemas, es parte fundamental para obtener un producto con calidad.

1.6 SERVIDOR

Un servidor es una computadora de gran capacidades, que se encarga de prestar servicios a otras computadoras que se conectan a él.

1.6.1 CLASIFICACION

Los servidores se pueden clasificar de la siguiente forma: Servidores de correo, Servidores Web, Servidores de aplicaciones, Servidores de Correo, Servidores IRC, etc. Para este caso es necesario conocer los siguientes conceptos:

- Servidores Web: Atiende y responde a las diversas peticiones de los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS.
- Servidores de Aplicaciones: Designados a veces como un tipo de middleware (software que conecta dos aplicaciones), como por ejemplo: Jakarta Tomcat.

1.6.2 SERVIDOR JAKARTA TOMCAT

Apache Tomcat (Jakarta Tomcat) empezó siendo una implementación de la especificación de los servlets comenzada por James Duncan Davidson, que trabajaba como arquitecto de software en Sun y que posteriormente ayudó a hacer el proyecto Open Source y en su donación a la Apache Software Foundation.

Tomcat funciona como un contenedor de Servlets y Java Server Pages (JSP) de Sun Microsystems, es por esto que se le considera un servidor de aplicaciones.

Sin embargo Tomcat puede, asimismo, funcionar como servidor Web por sí mismo. Opera de tal manera en entornos de desarrollo poco exigentes en términos de velocidad y de manejo de transacciones.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual. El servidor Jakarta Tomcat es muy poderoso y flexible. Este servidor como ya se mencionó puede correr bajo Windows, Unix, Netware, etc. Por las características que presenta es un servidor muy utilizado, además de ser un Servidor gratuito, bajo la Licencia de GNU.

1.6.3 INSTALAR APACHE TOMCAT

Para instalar, Tomcat es necesario haber instalado previamente Java 2 SDK. Tomcat puede bajarse de <http://jakarta.apache.org/tomcat>. El procedimiento de instalación que se muestra a continuación se basa en un sistema operativo MS Windows XP.

Para instalar se deben de seguir los siguientes pasos:

- Ejecutar el instalador de Tomcat “jakarta-tomcat-4.1.27.exe”.
- Seleccionar la instalación Típica. Elegir el directorio donde se instalará Tomcat, por ejemplo C:\Program Files\Apache Tomcat 4.0.
- El instalador de Tomcat creará, dentro del menú Inicio/Programas, un grupo de programas llamado Apache Tomcat 4.0.

Al terminar la instalación el contenido del directorio de Tomcat veremos una estructura de directorios como la que se muestra en la siguiente figura. Cada uno de estos directorios tiene su propósito por ejemplo en “bin” se encuentran los scripts de arranque del servidor y webapps es el directorio base para las aplicaciones web.

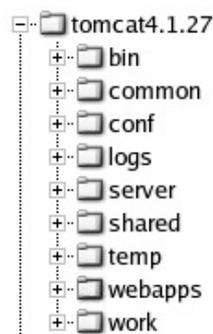


Fig 9. Directorio de Tomcat

Para arrancar Tomcat disponemos de dos accesos directos desde el menú de inicio de Windows para iniciar y detener Tomcat como los siguientes:



Fig 10. Iniciar y detener Tomcat

De acuerdo a la instalación, Tomcat escuchará peticiones en el puerto 8080. Desde el navegador podemos acceder con el siguiente URL: <http://localhost:8080/>. Si el servidor esta funcionando correctamente debe mostrar la siguiente página:

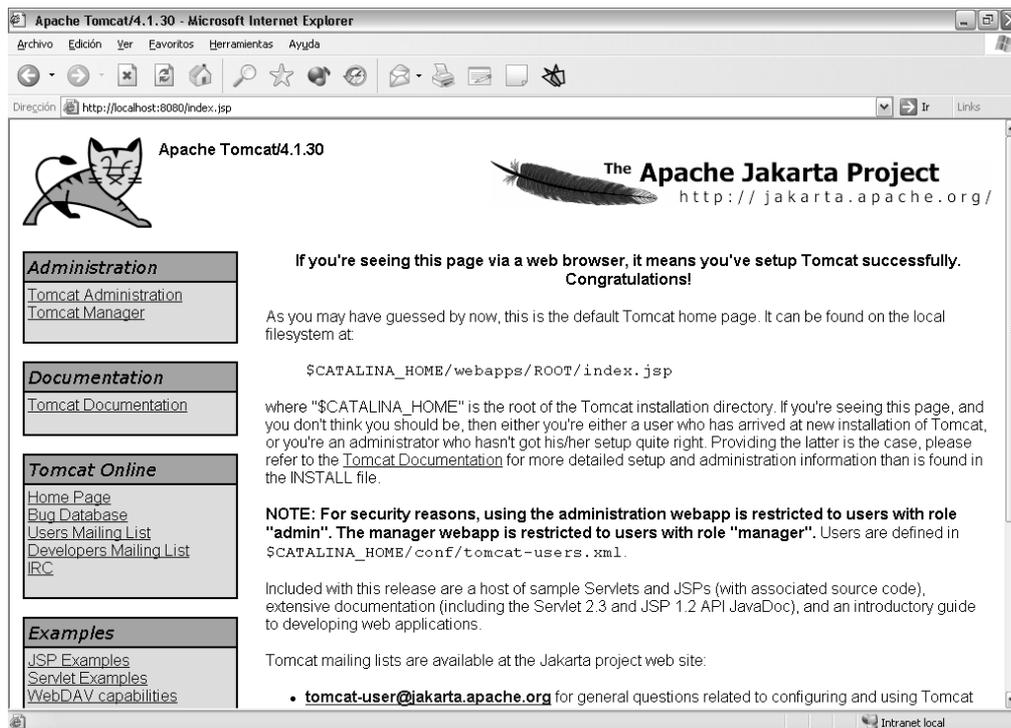


Fig 11. Servidor Apache Tomcat

1.7 DISEÑO DE PÁGINAS WEB

El diseño de aplicaciones Web con metodología de tres capas, se divide en tres componentes principales: Cliente, es la computadora desde la cual se accede a los servicios a través del navegador. Servidor, se encarga de realizar la lógica de negocios y los cálculos. Base de Datos, esta capa se encarga de mantener los datos almacenados.

La ventaja de esta arquitectura es la independencia de una aplicación cliente, pues el acceso se realiza a través del navegador, la presentación de los datos es con HTML y la comunicación entre el cliente y el servidor a través de HTTP.

La interfaz es el medio por el cual el usuario puede hacer todas sus operaciones a la base de datos, a través de Internet, por lo cual debe contar con las siguientes características:

- Diseño Amigable: Este tipo de diseño permitirá que el usuario, lo maneje fácilmente.
- Congruencia con los procesos. Es fundamental la funcionalidad y organización.
- Distribución Lógica. Los elementos de la interfaz como formularios, deben tener una distribución adecuada para lograr funcionalidad y entendimiento.
- Retroalimentación. Conviene el uso de mensajes para indicar que se esta procesando la información.

1.7.1 HTML

Para implementar la interfaz es fundamental conocer el manejo del lenguaje HTML. El lenguaje HTML (*Hypertext Markup Language*) se utiliza para crear documentos que muestran una estructura de hipertexto, además se pueden crear documentos de tipo multimedia, es decir, que contengan información más allá del simple texto. HTML contiene instrucciones llamadas tags o etiquetas, los cuales son interpretados por un visualizador para hacer dinámicas las paginas, lo que permitirá la aplicación de componentes dinámicos como Active Server Pages ⁵(ASP), Jscript, Java, etc, por esta razón se implementara como base para el desarrollo de la interfaz. La estructura de un documento html esta conformada por: un encabezado y el cuerpo. El encabezado incluye datos relacionados con el contenido del documento, como el título, autor, etc. El cuerpo del programa contiene textos e imágenes. La sintaxis de las etiquetas es la siguiente: <tag> contenido </tag>.

⁵ ASP es una tecnología desarrollada por Microsoft para crear páginas web de contenido dinámico apoyándose en scripts ejecutados en el servidor.

La estructura básica de un documento en HTML, es la siguiente:

```

<html> <!-- Etiqueta principal -->

<head>

Contenido de la cabeza del documento...

</head>

<body><!-- Se usa para definir el cuerpo del documento -->

Todo el contenido del cuerpo del documento...

</body>

</html>

```

<H1>, <H2>, <H3>: Titulares. Sirven para definir el tamaño de los títulos. Se pueden definir seis niveles de titulares. Se definen mediante las tags <H1>.....</H1> hasta <H6>.....</H6> siendo el más grande H1, después H2 y así sucesivamente.

<P>: Párrafos. Esta etiqueta, en un principio, se diseñó para saltar de párrafo por lo que puede ir sola "<P>".

Para generar enlaces se realiza mediante el tag <A>...... Ejemplo: comúnmente utilizado para saltar entre diferentes URLs.

Las listas en una interfaz resultan elementos esenciales para el manejo de información. Pueden darse cinco tipos diferentes de listas, y son las siguientes:

Listas numeradas u ordenadas: Se engloban por las tags y cada elemento de la lista estará encabezado por la tag que puede o no llevar la tag de cierre .

```

<OL>

<LI> Primera línea

<LI>Segunda línea </OL>

```

Listas con viñetas o sin orden: Se engloban por las tags y cada elemento de la lista, también estará encabezado por la tag . El resultado es que el navegador inserta viñetas (marcadores) delante de cada elemento.

```
<UL>  
<LI>Primera línea  
<LI>Segunda línea  
</UL>
```

Listas de menú y de directorio: Están en desuso puesto que su resultado suele ser, prácticamente, idéntico al de las listas con viñetas. Sintaxis es la siguiente `<menu>.....</menu>` y cada elemento encabezado por la tag ``.

Las tablas son elementos para ordenar la información dentro de celdas, las cuales se pueden identificar por un número de columna y renglón en que se encuentra. Estas permiten dar un mejor formato en el manejo y presentación de la información.

El contenido de una tabla lo debemos desarrollar entre las tags `<table>.....</table>`. Cada fila de la tabla se indica mediante las tags `<tr>.....</tr>`. Las tags `<th>` y `<td>` con sus correspondientes tags de cierre, indican para indicar las filas individuales dentro de cada fila. Las tags `<th>.....</th>` indican que se trata de celdas que sirven como encabezado de tabla y suelen visualizarse en negrita. Las tags `<td>.....</td>` indican que se trata de celdas comunes.

En cuanto a las imágenes se pueden insertar utilizando la siguiente sintaxis: ``

1.8 JAVA

El lenguaje de programación Java, fue diseñado por la compañía Sun Microsystems Inc, surgen en 1990, con el propósito de crear un lenguaje que pudiera funcionar en redes computacionales (redes de computadoras formadas por más de un tipo de computadora, ya sean PC, Machintosh, estaciones de trabajo, etc.), y que fueran independientes de la plataforma. Convirtiéndose así “Write Once, Run Everywhere” en el principal lema de Java.

Como lenguaje de programación para computadoras, Java se introdujo a finales de 1995. A principios de 1997 apareció Java 1.1, mejorando sustancialmente la primera versión del lenguaje. Java 1.2, más tarde rebautizado como Java 2, nació a finales de 1998, la cual es la tercera versión del Lenguaje de programación.

El lenguaje de programación Java es un lenguaje de alto nivel que cuenta con las siguientes características: es simple, ya que no cuenta con la complejidades de otros lenguajes como C++; es distribuido esto quiere decir que tiene la capacidad de interconexión TCP/IP; multiplataforma ya que los programas Java pueden ejecutarse en cualquier plataforma (Windows95 y NT, Mac, Unix); es multithread, es decir permite actividades simultáneas en un programa; etc.

1.8.1 INSTALACIÓN DE JSDK

Primeramente se puede descargar el j2sdk desde el sitio Web de Sun en <http://java.sun.com/j2se/>. Una vez descargado tenemos en nuestra computadora un archivo llamado: "j2sdk-(versión)-nb-3_6-bin-windows.exe". Para instalar solo tenemos que ejecutar el archivo, seguir las instrucciones y darle una ruta de instalación en el momento en que nos lo requiera. Generalmente es un ejecutable que realiza la instalación en el directorio raíz del disco duro. Dejando el ambiente de trabajo en C:\j2sdk1.4.2.

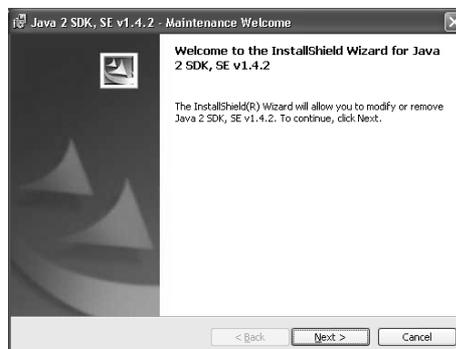


Fig 12. Instalación de JSDK

Una cosa muy importante, para el funcionamiento correcto, es agregar:

- El directorio de los programas en path
- Las variables de entorno classpath = C:\java\lib

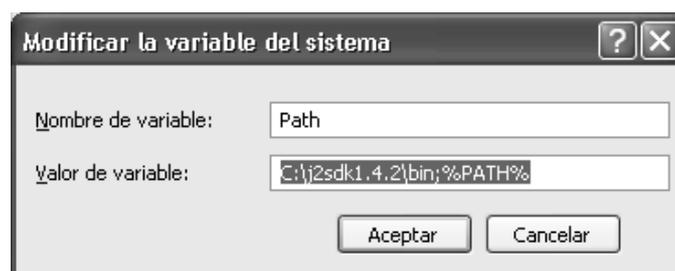
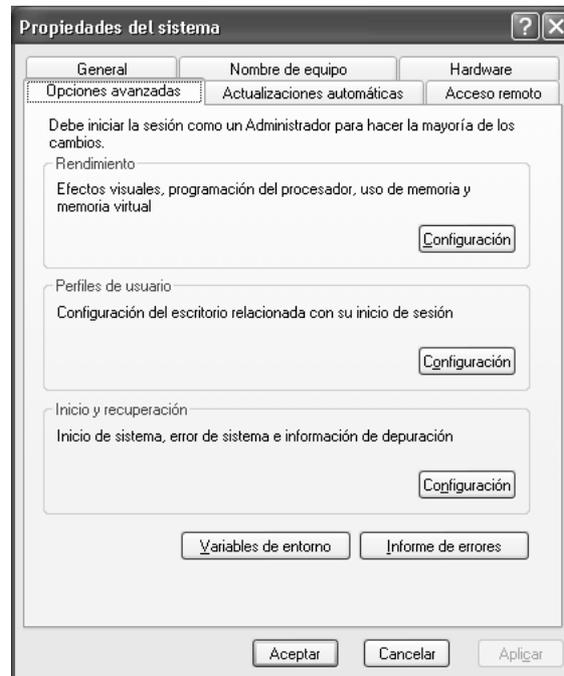


Fig 13. Valor de la variable Path

1.9 REGLAS DE NEGOCIO

Como ya se había mencionado las reglas de negocio se encargan de que en la aplicación se refleje las restricciones que existen en el mundo real, de forma que no se puedan llevar a cabo acciones que no son válidas. Dichas reglas se clasifican de la siguiente forma:

- **REGLAS DE MODELO DE DATOS.** Esta regla tiene como objetivo controlar la información almacenada en cada atributo o propiedad de una entidad u objeto.
- **REGLAS DE RELACIÓN.** Incluyen todas aquellas reglas que controlan las relaciones entre los datos.
- **REGLAS DE DERIVACIÓN.** Son el conjunto de reglas que especifican y controlan la obtención de información que se puede calcular a partir de la ya existente.
- **REGLAS DE RESTRICCIÓN.** Estas reglas requieren para su verificación del acceso a otros fragmentos de información, es decir necesitan consultar información, es decir necesitan consultar información ingresada.
- **REGLAS DE FLUJO.** Son reglas que indican el flujo que debe de seguir la información y obliga a que los caminos sean válidos dentro del sistema.

1.9.1 MODELO VISTA CONTROLADOR

El modelo vista controlador (MVC) plantea que se debe de separar la lógica del negocio de la presentación; permitiendo que la capa de negocio sea independiente de la capa del Cliente.

Los componentes principales del MVC son: el Controlador, el cual se encarga de examinar y extraer parámetros del objeto Request; el Modelo contiene las reglas del negocio y conoce las reglas para mantener el estado del mismo.

La Vista despliega información contenida en el Modelo, tomando en cuenta la responsabilidad de la generación de contenidos, es decir es responsable de la presentación.

Para implementar el MVC en un proyecto Web, se puede utilizar un framework, como los Struts. Un framework es una colección de interfaces que fueron auspiciados por Apache Software Foundation. El objetivo de los framework es ayudar en el desarrollo y mantenimiento de aplicaciones complejas, una de las ventajas de los Struts es que es software de uso libre.

Los principales componentes de los Struts son: Action Servlet, los cuales controlan el flujo de la ejecución; el Action quien se encarga de acceder a los objetos de negocio; el Action Form para validar la entrada de datos, ActionForward, se encarga de realizar un redireccionamiento, Struts-config.xml, contiene la configuración de la aplicación, incluyendo seguridad, acceso a la base de datos, mapeo, etc..

Los Struts interactúan con la aplicación Web a través de hipertexto y formas HTML. Las formas de captura permiten enviar datos a la aplicación mediante alguna acción del negocio.

Algo que se debe de tomar en cuenta es conocer los elementos existentes para la conectividad a la base de datos. Aunque al utilizar los struts no es necesario modificar mucho dentro del proyecto, solo basta con modificar un archivo para cambiar la conexión a diferentes bases de datos.

Podemos encontrar JDBC API el cual es el estándar de la industria para la conectividad independiente a base de datos, entre el lenguaje de programación Java y una amplia gama de Sistemas Manejadores de Bases de Datos (DBMS). El API JDBC hace posible tres cosas: Una de ellas es establecer una conexión con una base de datos o tener acceso a cualquier fuente de datos tabular, la segunda ejecutar sentencias SQL y por último procesar los resultados de las cuentas.

Consiste en un conjunto de clases e interfaces escritos en Java que ofrecen un completo API para la programación de base de datos.

Existen cuatro tipos de Drivers, el API contiene dos tipos importantes de interfaces: el primero es el API JDBC para los desarrolladores de aplicaciones, y el segundo es el driver de bajo nivel para los desarrolladores de drivers. Los drivers de tecnologías JDBC se clasifican en cuatro categorías.

Direct-to-Database Pure Java Driver. Este estilo de driver traduce directamente las llamadas JDBC al protocolo de red utilizado por el DBMS, permitiendo directamente una llamada de la máquina cliente al servidor DBMS y proporciona una solución básica al acceso por intranet.

Driver para Middleware de Bases de Datos. Este estilo de driver traduce las llamadas JDBC a un protocolo de software de tipo Middleware, y después es traducido el protocolo del DBMS, el Middleware provee conectividad a distintos tipos de bases de datos.

Un driver mixto Java - API nativo. Este estilo de driver convierte las llamadas JDBC a llamadas de tipo nativo de DBMS por ejemplo Oracle, Sybase, Informix, DB2, etc. Se requiere que algún tipo de código binario deba ser cargado en la máquina cliente y generalmente es desarrollado por el fabricante de DBMS.

Puente JDBC-ODBC. Esta combinación proporciona el acceso JDBC vía driver ODBC. El código binario ODBC, y en muchos casos el código del cliente de la base de datos, puede ser cargado en cada máquina cliente que utilice el puente. Conecta Java con el ODBC de Microsoft vía métodos nativos. Este driver fue desarrollado por JavaSoft y forma parte del JDK para Windows.

Para el desenvolvimiento del Sistema se debe de considerar la lógica del negocio, la cual es ejecutada detrás de un servidor Web. Debido a que esta nos permite visualizar las necesidades y requerimientos del usuario final.

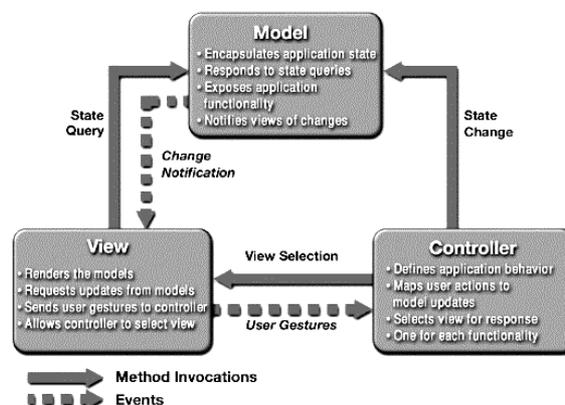


Fig 14. Modelo Vista Controlador (MVC)

1.9.2 ARQUITECTURA DE APLICACIONES WEB CON JAVA “WEB CONTAINER”

Un Web container se encarga de proporcionar la infraestructura de ejecución para las aplicaciones, dicha infraestructura es la siguiente:

- Soporte a Comunicaciones.
- Manejo del ciclo de vida de Servlets
- Soporte de ejecución

El Web Container permite que el programador se concentre en el desarrollo de las reglas de negocio de la aplicación. Un ejemplo de Web Container Apache Tomcat.

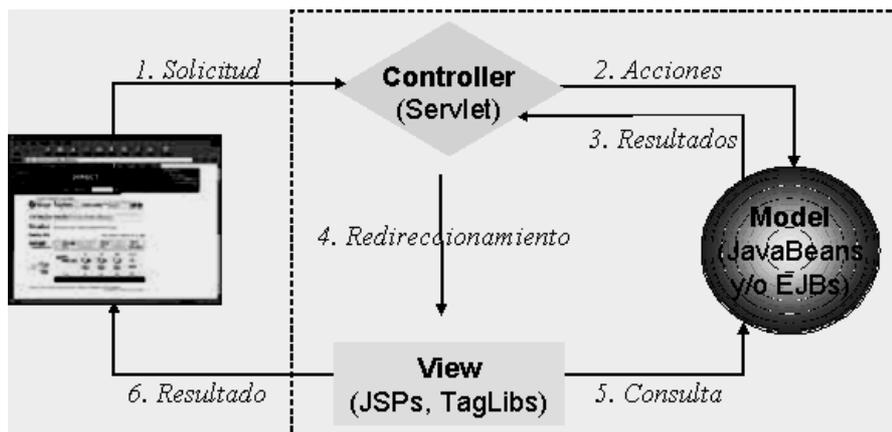


Fig 15. Arquitectura de Struts

CAPÍTULO II

PROPUESTA DE LA APLICACIÓN PARA CONTROLAR INFORMACIÓN DE SITIOS TURÍSTICOS A TRAVÉS DE WWW”

En los últimos años las aplicaciones Web han tenido un gran auge gracias a Internet y el incremento de sitios Web, sobre todo con el fin de fomentar el comercio electrónico.

La facilidad de administración hace ideal la implementación de estos sistemas, tanto para su despliegue en Internet como en intranets corporativas.

El auge de soluciones open source hace que su desarrollo sea sencillo y que por ende se desarrollen cada día más sistemas de este tipo.

2.1 ENTORNO Y FUNDAMENTOS

Actualmente en México el turismo es parte importante para el desarrollo económico, ya que una parte de los ingresos provienen del turismo extranjero. Por lo cual es importante dar a conocer todos los eventos artísticos y culturales, que se presentan.

En México los usuarios Internet llegan a 17 millones aproximadamente, lo que implica, que cada usuario debe investigar en los buscadores, para encontrar un sitio que les proporcione todos los datos sobre un evento, sitio turístico, etc.

En la actualidad los sitios Web que se encuentran en la red, solo pueden ser de dos tipos: uno de ellos proporciona solamente información escrita acerca de los lugares, exposiciones o eventos que se presentan en la ciudad y el otro es el cual permite la venta de boletos solamente para un evento en específico, cada uno de estos sitios presenta la información por separado, no existe un portal o sitio donde se pueda encontrar todos los eventos que se presentan en la ciudad de México y tampoco donde se puedan hacer reservaciones o compras directas.

2.2 PROPUESTA DE LA APLICACIÓN

La propuesta del sistema surge debido a que en la actualidad no existe un portal que proporcione toda la información acerca de los eventos culturales y turísticos con los que cuenta y se presentan en la Ciudad de México y que además se puedan encontrar todos los servicios necesarios para poder adquirir beneficios tales como la compra y reservación de boletos para dichos eventos.

El sistema Web “Sistema para controlar información de sitios de interés turísticos, a través de WWW” se desarrolla con el propósito central de automatizar vía Web el servicio de compra y reservación de boletos para eventos culturales y de entretenimiento, así como también en hacer reservaciones para tener un guía turístico en los recorridos de museos u otros eventos.

Estos servicios beneficiarían tanto a los habitantes de la ciudad de México como también a los turistas, ya que proporcionará información actual acerca de los eventos más importantes que se presentan cada mes.

Lo que se pretende es que el sistema además de brindar servicios específicos también pueda brindar información de utilidad para los clientes, por medio de links dentro de la página, para dar a conocer: restaurantes, teatros, hoteles.

Por lo tanto se plantea un sistema Orientado al Web el cual esta encaminado a la cultura por una parte y por otra al turismo de la ciudad de México.

A continuación se muestra un diagrama de Flujo de datos, en el cual se puede ver los servicios que podrá obtener el usuario a través del sistema.

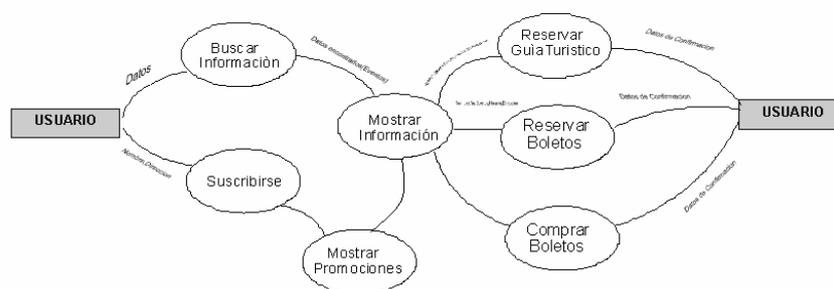


Fig 16. Diagrama de Flujo de Datos

2.3 REQUERIMIENTOS FUNCIONALES

Se desea que el sistema sea accesible a través de la World Wide Web (WWW), por lo cual todas las acciones realizadas se ejecutarán de forma rápida y eficaz desde cualquier parte del mundo. Una vez que se halla ingresado al sistema el usuario, podrá seleccionar las siguientes actividades:

- Registrarse en el sistema.
- Obtener información de los sitios turísticos.
- Consultar los precios de los sitios de interés.
- Reservar boletos para los eventos.
- Pago de boletos.
- Reservar recorrido con Guía Turístico.

2.4 REQUERIMIENTOS NO FUNCIONALES

El sistema en cuestión se pretende que sea seguro y solo permita a usuarios con permisos poder editar y borrar información, con los permisos adecuados, es por esto que el sistema contará con niveles de seguridad para los usuarios, se requiere que en cuanto se comiencen a realizar operaciones de reservaciones y compra el usuario se registre en el sistema, en lo que se refiere a consulta de información el usuario podrá navegar por el sitio sin ningún problema.

Este sistema debe admitir que se realicen reservaciones o compras para uno o más eventos, siempre y cuando se encuentre registrado en el sistema. El pago o reservación se permite al cliente, con una tarjeta de crédito válida. Al realizar una operación de reservación o compra el sistema deberá generar un documento el cual se enviará vía e-mail como notificación al usuario con los datos necesarios para comprobar y hacer valida su transacción.

El usuario podrá en cualquier momento acceder, modificar o cancelar su registro en la página, según las normas establecidas al momento de realizar la operación. Las reservaciones del guía se podrán realizar desde cualquier parte del país, debido a que se realizarán transferencias de dinero por la WWW, siempre y cuando se haya reservado o comprado el boleto para un evento. La mecánica será la misma que al realizar un proceso de reservación o compra.

El sistema contará con una interfaz simple para que el usuario pueda interactuar con el sistema de forma fácil y rápida.

2.5 OBJETIVO GENERAL

El sistema para controlar información de sitios turísticos tiene como objetivo optimizar el tiempo de espera, para reservación, compra de boletos y reservación de guías turísticos, proporcionando servicios al instante y desde cualquier lugar, a través de la WWW.

2.6 OBJETIVOS ESPECIFICOS

Para lograr el objetivo general formulado, es necesario alcanzar las siguientes metas específicas:

1. Definir el esquema de la base de datos para el sistema.
2. Diseñar la interfaz del sistema.
3. Hacer posible el uso remoto del sistema a través de Internet.

2.7 BENEFICIOS

Como ya se había mencionado uno de los objetivos es dar a conocer los sitios de interés de la Ciudad de México, por medio de guías turísticos, que se podrán contratar vía Internet, de forma remota, sin la necesidad de realizar consultas vía telefónica, además permitir realizar consultas, reservaciones y compra de boletos de museos, exposiciones, conciertos, teatros entre otros.

Uno de los mayores beneficios en contra de las compañías que no cuentan con sistema Web es que el sistema se encontrará funcionando los 365 días al año, lo cual incrementa la productividad.

Otro de los beneficios será que los clientes podrán consultar a través del Sistema diferentes sitios Web en los que podrá encontrar información sobre restaurantes, hoteles, parques de diversión, etc.

Crear un sistema de este tipo beneficiara tanto a los usuarios como a las compañías que presentan el evento, ya que se darán a conocer a nivel nacional e Internacional.

2.8 ALCANCE

Se diseñará un sistema orientado al Web con capacidad de manejar la reservación y compra de boletos o guías turísticas, mediante la WWW de forma confiable. Lo que implicaría un sistema configurable, amigable y que sea reutilizable vigente con el tiempo y con el crecimiento.

Se propone construir una aplicación que pueda usarse con servidores estándares de Web y aplicación, como Apache Tomcat. Así como también debe ser compatible con los navegadores actuales como Internet Explorer IE , Netscape, Firefox, Mozilla, etc. La aplicación correrá sobre un servidor 24x7 con supervisión, aire acondicionado, etc.

Por otro lado debido a que la seguridad en la Web requiere ser tratada con gran importancia, el sistema contará con niveles de seguridad, lo que implica que dependiendo del nivel del usuario será la información a la que se tenga acceso.

2.9 ENTREGABLES

El cliente obtendrá una serie de productos finales, los cuales le permitirán la instalación e implementación del sistema de forma rápida y eficiente, dichos productos serán:

- Aplicación Web para “Sistema para controlar información de sitios de interés turísticos, a través de WWW”
- Requerimientos del Sistema
- Manual para la configuración del sistema.

CAPITULO III

DISEÑO DE LA SOLUCIÓN

Diseñar un software significa establecer la arquitectura del mismo antes de su construcción, es decir, especificar completamente los módulos de los cuales constará el sistema, la relación que existirá entre ellos y la interfaz con el usuario. Ya que a partir de la documentación del diseño, los programadores podrán construir y probar el software. La documentación del diseño garantiza la posibilidad de efectuar correcciones y mejoras al software en el futuro.

Debido a que el mantenimiento de un sistema constituye uno de los costos más grandes, se debe de tomar en cuenta diferentes aspectos antes de diseñar un sistema, entre los cuales podemos mencionar los requerimientos de software y hardware que empleara el sistema.

3.1 REQUERIMIENTOS HARDWARE Y SOFTWARE

En cuanto a los requerimientos del Hardware y Software nos debemos enfocar en implementar elementos que sean factibles tanto para el aprovechamiento de la aplicación como para el cliente.

HARDWARE

Los requerimientos mínimos de hardware con los que se debe contar son con un equipo tenga la función de Servidor, y además pueda contar con los siguientes requerimientos:

- Pentium III, IV ó equivalente.
- Memoria RAM 512 MB
- Capacidad en Disco Duro de 40 GB
- Tarjeta de Red
- No Brake

En lo que se refiere al hardware del cliente o del usuario, este se debe considerar como mínimo:

- Procesador Pentium II o superior.
- Memoria RAM 128 MB
- Tarjeta de Red.

En cuanto a las condiciones del servidor se debe de cuidar el lugar donde se establezca el equipo, ya que este debe contar con supervisión 24 x 7, aire acondicionado, temperatura adecuada, etc.

SOFTWARE

Se ha elegido trabajar con software libre, ya que estas herramientas no requieren de un pago de licencia, además han tenido un desarrollo satisfactorio.

Entre las herramientas de software que se utilizan para desarrollar este sistema, se encuentra MySQL¹ el cual se elige como Base de Datos, debido a que este proporciona grandes beneficios, ya que corre en más de 20 plataformas (Linux, Windows, OS/X, Netware, etc), además es un Open Source lo cual permite que el gasto económico al desarrollar el sistema se reduzca.

También se optó por Jakarta Tomcat versión 4.x como servidor de la aplicación, por ser un Servidor Gratis, bajo la Licencia de GNU, además de soportar varias Plataformas como Windows y Linux, al mismo tiempo es uno de los más estables que se encuentran en la actualidad.

Como lenguaje de programación, se utilizó el Lenguaje Java versión 1.4.2, el cual permite la reutilización de código y además es software libre. Además en los últimos años ha demostrado ser una de las herramientas más confiables, efectivas y poderosas en el desarrollo de software.

Se aplicaron tecnologías como los servlets los cuales los desarrolló Sun. Los servlets Java son muy eficientes, debido al esquema de threads en el que se basan y al uso de una arquitectura estándar como la JVM, Java Virtual Machine. También se encuentran los JSP permiten juntar HTML, aplicaciones Java, y componentes como las JavaBeans creando una página Web especial que el servidor Web compila dinámicamente en un servlet la primera vez que es llamada.

Para un sistema robusto como el que se considera diseñar, se debe contar con un diseño adecuado, para ello se utiliza el lenguaje UML, que es uno de los más utilizados y conocidos en la actualidad por su rapidez. El cual cuenta con distintos modelos, los cuales representan al Sistema desde diferentes perspectivas. En este caso para el diseño de los diagramas se utiliza Poseidon for UML, herramienta idónea para el desarrollo de los diagramas.

¹ MySQL es utilizado por más de 8 millones www.mysql.com.

3.2 HERRAMIENTAS NECESARIAS

Las herramientas de software que se han elegido actualmente se implementan en el desarrollo de Sistemas orientados al Web. Las herramientas necesarias para el desarrollo del sistema, son las siguientes:

- Un ambiente de desarrollo para Java, tal como el Java 2 SDK. La versión estándar del SDK 1.4 ya incluye el API JDBC.
- Un servidor de bases de datos MySQL al que se tenga acceso con un nombre de usuario y una contraseña.
- El driver JDBC para MySQL, Connector/ J.
- El servidor de aplicaciones Tomcat que nos permitirá trabajar con los programas en Java. Con éste se incluye el API Java Servlet.

SOFTWARE UTILIZADO PARA DESARROLLAR APLICACIÓN WEB	
ETAPA	SOFTWARE
Análisis	UML Programa Poseidón (Diagrama CU, Diagrama Clases) Erwin 4.01 (Diagrama Entidad / Relación)
Diseño e Implementación	Java Apache Tomcat MySQL
Pruebas	Internet Explorer Excel

3.3 DIAGRAMA ENTIDAD / RELACIÓN

El diagrama Entidad / Relación² representa una estructura lógica general de la base de datos. Es una de las herramientas más utilizadas en el diseño de Bases de Datos. Este diagrama consiste en un conjunto de objetos básicos denominados entidades (representadas por rectángulos) y relaciones.

En el siguiente diagrama se muestra la estructura básica del sistema a partir del cual se podrá diseñar la base de datos.

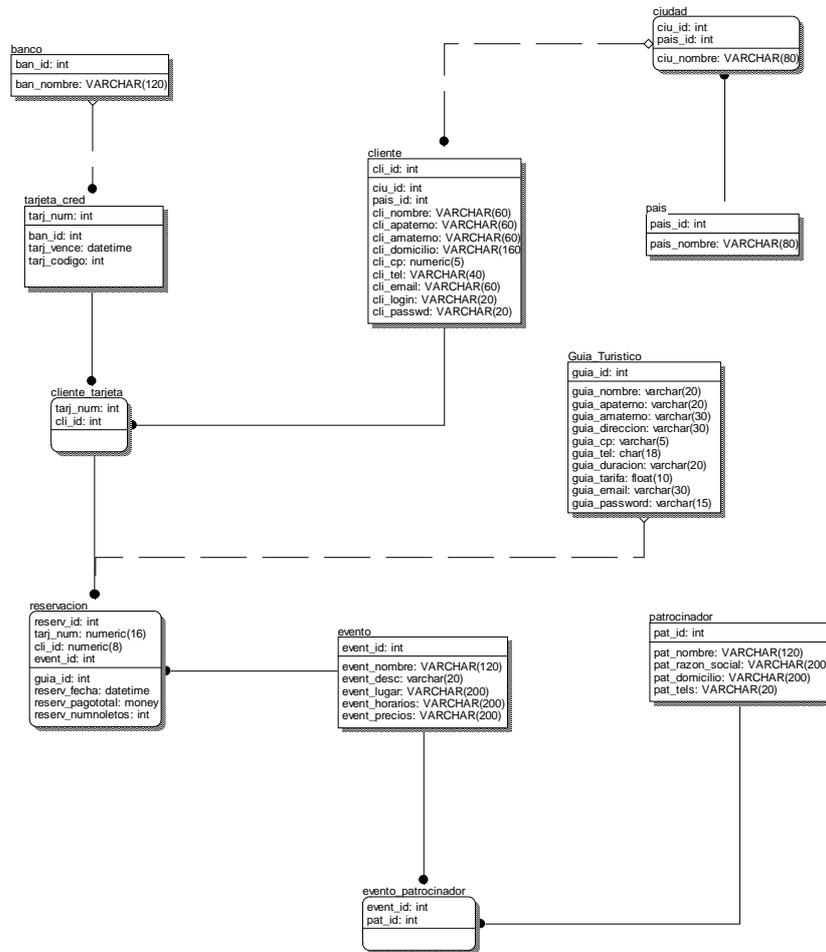


Fig 17. Diagrama E / R

² El Diagrama Entidad / Relación fue introducido por Peter Chen en 1976.

3.4 ACTORES DEL SISTEMA

La aplicación para controlar información de sitios de interés turísticos, a través de WWW, permite la reservación, compra de boletos y la reservación de guías turísticos a través de la Web. En el sistema se encuentran identificados los siguientes actores:

- Administrador
- Usuario
- Guía Turístico

Descripción de los actores:

Usuario. Es la persona que realiza reservaciones, compra de boletos o reserva un guía turístico para un recorrido.

Guía Turístico. Es la persona que presta el servicio de guía, el cual puede verificar los horarios de trabajo por medio de la Web.

Administrador. Es el encargado de toda la gestión del sistema. Entre las actividades que realiza son: administrar catálogos, generar reportes.

3.5 DISEÑO DEL SISTEMA CON UML

Como ya se había mencionado se eligió el Unified Modeling Lenguaje (UML), para el diseño del sistema debido a que desde hace algunos años este lenguaje ha tenido gran auge. Utilizando UML podemos representar al sistema desde diferentes perspectivas, los siguientes diagramas muestran en forma general como funciona el sistema.

DIAGRAMA “CASOS DE USO”

A continuación se muestra el Diagrama de Casos de Uso que se obtuvo a través del lenguaje de diseño de UML. Este diagrama representa gráficamente el sistema, como se comporta el sistema desde el punto de vista de los actores (usuario, administrador, guía turístico) y la interrelación que estos tienen con el sistema.

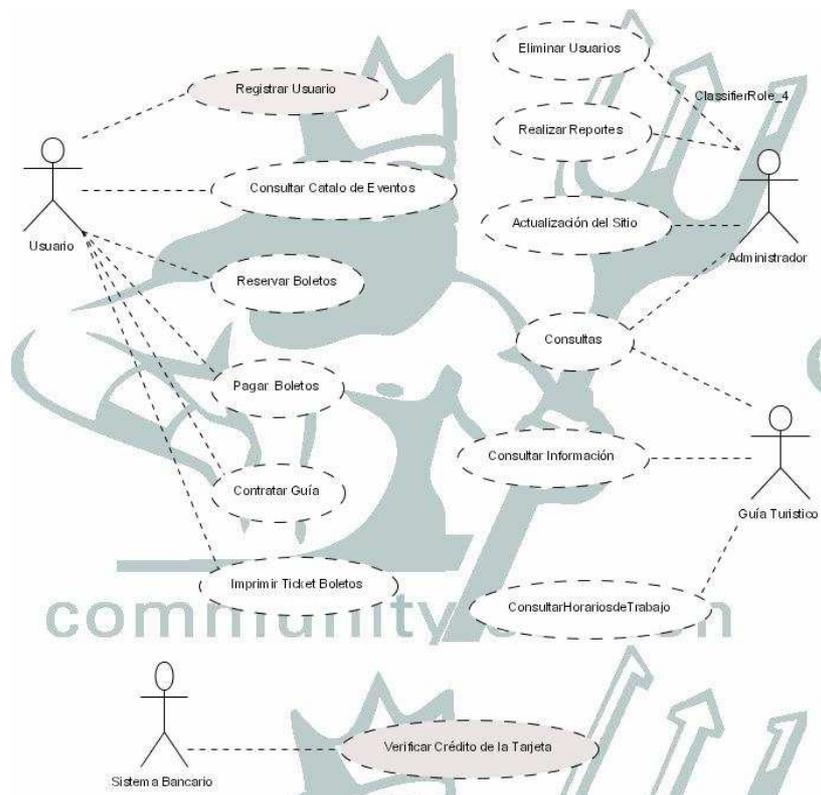


Fig 18. Diagrama de Casos de Uso

3.6 DESGLOSE DE LOS CASOS DE USO

CASOS DE USOS DEL CLIENTE

Nombre:	Registrar Cliente
Descripción:	Permite al usuario registrarse en el sistema para reservar, comprar y reservar guías turísticas, a través de la Web.
Actores:	Cliente o Usuario
Precondiciones:	El cliente debe entrar al sitio Web y entrar al módulo de registro.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario entra al módulo de registro. 2. El sistema muestra un formulario con los siguientes datos: <ul style="list-style-type: none"> - Usuario y password - Nombre del Cliente - Apellidos Paterno y Materno - Dirección - Código postal - Email - Teléfono 3. El cliente captura los datos y los envía. 4. El sistema comprueba la validez de los datos y almacena los datos.
Flujo Alternativo:	<ol style="list-style-type: none"> 1. Si el usuario o password no son validas, enviar mensaje para que escriba nuevamente el usuario. 2. Si falta algún datos de llenar del formulario, el sistema mandará un mensaje de error que indique “Debe llenar el campo es obligatorio”. 3. Si algún tipo de dato en el formulario no es válido, se envía mensaje “Valor incorrecto”.

Nombre:	Registrar Autenticación
Descripción: Este caso de uso el usuario se logea en el sistema para reservar, comprar y reservar guías turísticos, a través de la Web.	
Actores: Cliente	
Precondiciones: El cliente debe entrar al sitio Web y entrar al módulo de registro.	
Flujo Normal: <ol style="list-style-type: none"> 1. El usuario entra al módulo de registro. 2. El sistema solicita usuario y password 3. El cliente introduce su usuario y password. 4. El sistema valida los datos, muestra la página de inicio con un mensaje de bienvenida y el nombre del usuario 	
Flujo Alternativo: <ol style="list-style-type: none"> 1. Si el usuario o password no son validas, enviar mensaje “Usuario o password invalido”. 2. Si falta algún de los datos de mostrar “Es necesario llenar los dos campos”. 	

Nombre:	Consultar Precios
Descripción: Permite consultar el catálogo con los eventos y datos más importantes para cada evento.	
Actores: Cliente, Administrador	
Precondiciones: Entrar al sistema	
Flujo Normal: <ol style="list-style-type: none"> 1. El usuario entra al sistema. 2. No es necesario que se logee 3. Muestra un recuadro para escribir el evento que se busca. 4. El sistema muestra todos eventos relacionados. 	
Flujo Alternativo: <ul style="list-style-type: none"> ▪ Si no se encontró el evento enviar mensaje “No existe evento”. ▪ Si el evento se ha cancelado en las ultimas horas muestra “Evento Cancelado” 	

Nombre:	Reservar Boletos
Descripción:	Permite realizar una reservación de boletos para un evento o reservar un guía turístico para un museo, exposición, etc.
Actores:	Cliente
Precondiciones:	Entrar al sitio Web, el usuario debe haberse registrado e iniciado una sesión.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario se logea 2. El sistema valida usuario y password. 3. Se elige el evento que se desea y se da clic en reservar. 3. Muestra el formulario para verificar información y pide el número de tarjeta. 4. El sistema genera un numero de reservación y envía por correo electrónico los datos de la reservación, por ejemplo: <ul style="list-style-type: none"> - Numero de reservación - Usuario - Fecha <p>Detalle de la reservación:</p> <ul style="list-style-type: none"> - Descripción del evento - Reservación - Numero de Boletos reservados - Precio - Total
Flujo Alternativo:	<ul style="list-style-type: none"> ▪ Si el usuario y password no son validas, registrarse en el sistema. ▪ Si el número de tarjeta no es valido enviar mensaje para que se cambie. ▪ Si la tarjeta no tiene fondos enviar mensaje y no continuar con el proceso.

Nombre:	Pagar Reservación
Descripción:	Permite a un usuario pagar reservaciones con el sistema de reservaciones de boletos.
Actores:	Cliente, BD
Precondiciones:	Validar usuario y tener una reservación.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario entra al sistema. 2. Es necesario que se registre 3. Solicitar el número de tarjeta. 4. Verificar que sea valida tarjeta.
Flujo Alternativo:	<ul style="list-style-type: none"> ▪ Si el numero de cuenta no es válido “Tarjeta no válida”.

CASOS DE USO DEL ADMINISTRADOR

Nombre:	Administrar Catálogos
Descripción:	Permite al administrador elegir un catálogo con el fin de realizar modificaciones como: altas, bajas y cambios.
Actores:	Administrador
Precondiciones:	<ol style="list-style-type: none"> 1. Entrar al sitio Web. 2. Validar usuario.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario entra al sistema. 2. El sistema muestra un menú con las siguiente opciones: <ul style="list-style-type: none"> - Eventos - Clientes - Guía Turístico - Proveedores 3. El administrador selecciona una opción: <ul style="list-style-type: none"> - Alta de evento - Eliminar evento - Alta de Guía Turístico - Eliminar Guía Turístico

<p>Flujo Alternativo:</p> <ul style="list-style-type: none"> - Si no se capturaron los datos obligatorios, mostrara “Datos insuficientes”. - Si se cancela la eliminación de un evento muestra la última lista de eventos - Si se capturan datos no válidos en algún campo el sistema deberá mostrar “El tipo de datos es incorrecto”.

Nombre:	Generar Reportes
Descripción:	El administrador se identifica en el sistema para generar reportes.
Actores:	Administrador
Precondiciones:	<ol style="list-style-type: none"> 3. Entrar al sitio Web. 4. Validar usuario.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario entra al sistema. 2. El sistema muestra un menú con las siguiente opciones: <ul style="list-style-type: none"> - Reporte de clientes - Reporte de reservaciones - Reporte de boletos en existencia -Reporte de ventas 3. El administrador selecciona una opción, por ejemplo para el reporte Reservaciones, deberá mostrar los siguientes campos: <ul style="list-style-type: none"> - Numero de reservación - Fecha - Datos del cliente - Detalles de la reservación.
Flujo Alternativo:	<ul style="list-style-type: none"> - Si no existen datos para alguna de las opciones, el sistema deberá mostrar “No existen reservaciones”, “No existen Ventas”, “No existen boletos en existencia”, de acuerdo al reporte generado.

CASOS DE USO DEL GUIA TURISTICO

Nombre:	Consultar Horarios
Descripción:	El guía turístico podrá consultar los horarios de trabajo para los cuales se encuentra contratado.
Actores:	Guía Turístico
Precondiciones:	<ul style="list-style-type: none"> - Entrar al sitio Web - Logearse en el sistema
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario entra al sistema. 2. El sistema muestra la opción de ver horarios 3. Se muestran la información de la siguiente manera: <ul style="list-style-type: none"> - Evento - Descripción - Hora y Lugar - Detalles del cliente
Flujo Alternativo:	<ul style="list-style-type: none"> - Si no existen recorridos contratados, el sistema deberá mostrar “No existen reservaciones”.

3.7 DIAGRAMA DE CLASES

Este diagrama tiene como objetivo mostrar la relación de las clases del sistema con sus relaciones estructurales y de herencia. Se muestra a continuación parte del Diagrama de clases, donde se muestran las clases de las que puede constar el sistema.

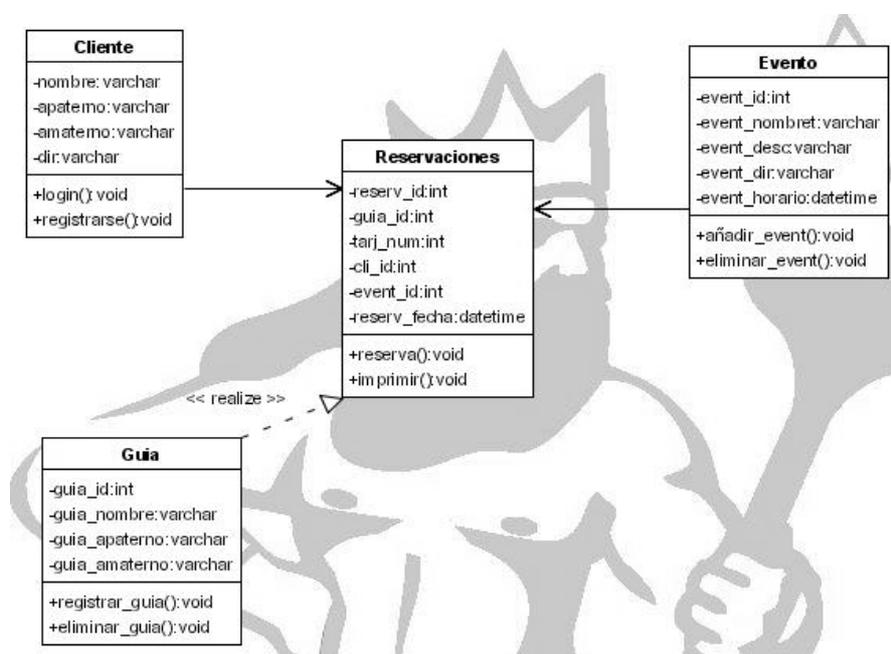


Fig 19. Diagrama de Clases

3.8 DIAGRAMA DE SECUENCIA

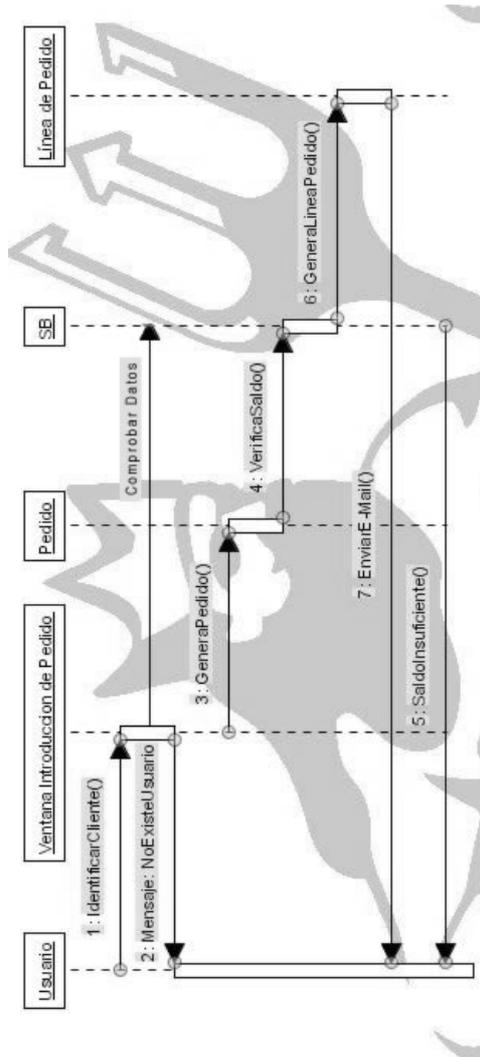


Fig 20. Diagrama de Secuencia

3.9 DICCIONARIO DE DATOS

En el Diccionario de Datos, se encuentra todos los elementos que forman parte del Sistema, los detalles y descripciones de cada uno de estos elementos. El diccionario de datos es útil para cuando se desee conocer cuantos caracteres abarca determinado dato, y donde se utiliza. El diccionario de datos es parte importante de la documentación del sistema, también ayuda a localizar errores, facilita el análisis de los detalles, etc.

"Cliente"					
Nombre	Definición	PK	FK	Tipo de Dato	Referencia
cli_id	Identifica al cliente en el sistema	Si	No	int	
ciu_id	Almacena el numero de ciudad del cliente	No	Si	int	Ciudad
pais_id	Almacena el país de donde es el cliente.	No	Si	int	Pais
cli_nombre	Nombres del cliente	No	No	varchar(60)	
cli_apaterno	Apellido Paterno del cliente	No	No	varchar(60)	
cli_amaterno	Apellido Materno del cliente	No	No	varchar(60)	
cli_direccion	Dirección completa del cliente	No	No	varchar(160)	
cli_cp	Código postal de la dirección	No	No	numeric(5)	
cli_tel	Teléfono del cliente	No	No	varchar(80)	
cli_email	Correo electrónico del cliente	No	No	varchar(80)	
cli_password	Password para identificarse en el sistema	No	No	varchar(80)	

"Evento"					
Nombre	Definición	PK	FK	Tipo de Dato	Referencia
event_id	Identifica el evento	Si	No	int	
event_nombre	Nombre completo del evento	No	No	varchar(120)	
event_desc	Descripción del Evento	No	No	varchar(600)	
event_lugar	Dirección completa del evento	No	No	varchar(200)	
event_horarios	Horario en el que se presenta el evento	No	No	Varchar(200)	
event_precios	Precio del evento por persona	No	No	Varchar(200)	

"Guia_Turistico"					
Nombre	Definición	PK	FK	Tipo de Dato	Referencia
guia_id	Numero para identificar a un guía turístico	Si	No	int	
guia_nombre	Nombre o nombre del guía turístico	No	No	varchar(80)	
guia_apaterno	Apellido paterno del guía turístico	No	No	varchar(80)	
guia_amaterno	Apellido materno del guía turístico	No	No	varchar(80)	
guia_direccion	Dirección del guía turístico	No	No	varchar(80)	
guia_cp	Almacena el código postal	No	No	varchar(80)	
guia_tel	Guarda el numero telefónico del guía	No	No	varchar(20)	
guia_email	Se refiere al correo electrónico del guía	No	No	varchar(80)	
guia_tarifa	Precio que cobra el guía turístico por recorrido	No	No	decimal	
guia_duracion	Se refiere a la duración del recorrido	No	No	varchar(80)	
guia_password	Password con el cual se identifica.	No	No	varchar(80)	

"Patrocinador"					
Nombre	Definición	PK	FK	Tipo de Dato	Referencia
pat_id	Identifica al patrocinador del evento	Si	No	int	
pat_nombre	Nombre con quien se puede contactar	No	No	varchar(120)	
pat_razonsocial	Nombre de la empresa	No	No	varchar(200)	
pat_domicilio	Dirección del patrocinador	No	No	varchar(150)	
pat_tels	Teléfonos del patrocinador	No	No	varchar(80)	

"Evento_Patrocinador"					
Nombre	Definición	PK	FK	Tipo de Dato	Referencia
Event_id	Identifica el evento	No	Si	int	Evento
pat_id	Identificador del patrocinador	No	Si	int	Patrocinador

"Reservaciones"					
Nombre	Definición	PK	FK	Tipo de Dato	Referencia
reserv_id	Numero consecutivo para cada reservación	Si	No	int	Reservaciones
guia_id	Numero para identificar al guía turístico	Si	Si	int	Guia_Turistico
cli_id	Almacena el cliente	Si	Si	int	Cliente
event_id	Identifica el evento que se reserva	Si	Si	int	Evento
tarj_num	Guarda el numero de cuenta	Si	Si	int	Cliente_Tarjeta
reserv_fecha	Guarda la fecha de reservación	No	No	datetime	
reserv_pagototal	Se refiere al pago total por las reservaciones	No	No	decimal	
reserv_numboletos	El numero total de boletos reservados	No	No	int	

"Tarjeta_Cred"					
Nombre	Definición	PK	FK	Tipo de Dato	Referencia
tarj_num	Guarda el numero de cuenta al que cobrará	Si	No	int	
Ban_id	Identifica el banco que expide la tarjeta			Numeric(4)	Banco
tarj_vence	Almacena la fecha de vencimiento de la tarjeta	No	No	datetime	
tarj_codigo	Almacena el código de la tarjeta	No	No	Numeric(4)	

"Banco"					
Nombre	Definición	PK	FK	Tipo de Dato	Referencia
Ban_id	Identifica el Banco	Si	No	int	
Ban_nombre	Guarda el nombre del Banco	No	No	Varchar(80)	

"Cliente_Tarjeta"					
Nombre	Definición	PK	FK	Tipo de Dato	Referencia
Tarj_num	Almacena el numero de Tarjeta del Cliente	No	Si	int	tarjeta_cred
Cli_id	Identifica a que cliente pertenece la Tarjetas	No	Si	Int	Cliente

"Ciudad"					
Nombre	Definición	PK	FK	Tipo de Dato	Referencia
Ciu_id	Identifica la ciudad	Si	No	Int	
Pais_id	Identifica a que país pertenece el cliente	No	Si	Int	Pais
Ciud_nombre	Almacena el nombre de la Ciudad	No	No	Varchar(80)	

"Pais"					
Nombre	Definición	PK	FK	Tipo de Dato	Referencia
Pais_id	Es el identificador del país	Si	No	Int	
Pais_nombre	Guarda el nombre del país donde se esta reservando.	No	Si	Varchar(80)	

3.10 SCRIPT DE LA BASE DE DATOS

Como ya se menciona el manejador de base de datos que se utilizará será MySQL, a continuación se muestra el script para la generación de base de datos que se necesita para la implementación del sistema.

=====
Tabla: Banco
=====

```
CREATE TABLE banco (  
    ban_id          INT NOT NULL,  
    ban_nombre     VARCHAR(120) NOT NULL )
```

=====
Tabla: Ciudad
=====

```
CREATE TABLE ciudad (  
    ciu_id         INT NOT NULL,  
    pais_id       INT NOT NULL,  
    ciu_nombre    VARCHAR(80) NOT NULL )
```

=====
Tabla: Cliente
=====

```
CREATE TABLE cliente (  
    cli_id        INT NOT NULL,  
    ciu_id        INT NULL,  
    pais_id       INT NULL,  
    cli_nombre    VARCHAR(60) NOT NULL,  
    cli_apaterno  VARCHAR(60) NULL,  
    cli_amaterno  VARCHAR(60) NULL,  
    cli_domicilio VARCHAR(160) NOT NULL,  
    cli_cp        NUMERIC(5) NOT NULL,  
    cli_tel       VARCHAR(40) NOT NULL,  
    cli_email     VARCHAR(60) NULL,  
    cli_login     VARCHAR(20) NOT NULL,  
    cli_passwd    VARCHAR(20) NOT NULL )
```

=====
Tabla: Cliente_Tarjeta
=====

```
=====
CREATE TABLE cliente_tarjeta (
    tarj_num      INT NOT NULL,
    cli_id        INT NOT NULL )
=====
```

Tabla: Evento

```
=====
CREATE TABLE evento (
    event_id      INT NOT NULL,
    guia_id       INT NOT NULL,
    event_nombre  VARCHAR(120) NOT NULL,
    event_desc    VARCHAR(20) NULL,
    event_lugar   VARCHAR(200) NOT NULL,
    event_horarios VARCHAR(200) NOT NULL,
    event_precios VARCHAR(200) NOT NULL )
=====
```

Tabla: Evento_Patrocinador

```
=====
CREATE TABLE evento_patrocinador (
    event_id      INT NOT NULL,
    pat_id        INT NOT NULL,
    guia_id       INT NOT NULL )
=====
```

Tabla: Guia_Turistico

```
=====
CREATE TABLE Guia_Turistico (
    guia_id       INT NULL,
    guia_nombre   VARCHAR(20) NULL,
    guia_apaterno VARCHAR(20) NULL,
    guia_amaterno VARCHAR(30) NULL,
    guia_direccion VARCHAR(30) NULL,
    guia_cp       VARCHAR(5) NULL,
    guia_tel      CHAR(18) NULL,
    guia_email    VARCHAR(30) NULL,
    guia_tarifa   FLOAT(10) NULL,
    guia_duracion VARCHAR(20) NULL,
    guia_password VARCHAR(15) NULL )
=====
```

Tabla: Pais

```
=====
CREATE TABLE pais (
=====
```

```
    pais_id          INT NOT NULL,  
    pais_nombre     VARCHAR(80) NOT NULL )
```

Tabla: Patrocinador

```
CREATE TABLE patrocinador (  
    pat_id          INT NOT NULL,  
    pat_nombre     VARCHAR(120) NOT NULL,  
    pat_razon_social VARCHAR(200) NOT NULL,  
    pat_domicilio  VARCHAR(200) NOT NULL,  
    pat_tels       VARCHAR(20) NOT NULL )
```

Tabla: Reservacion

```
CREATE TABLE reservacion (  
    reserv_id      INT NOT NULL,  
    event_id       INT NOT NULL,  
    tarj_num       NUMERIC(16) NOT NULL,  
    cli_id         NUMERIC(8) NOT NULL,  
    guia_id        INT NOT NULL,  
    reserv_fecha   DATETIME NOT NULL,  
    reserv_pagototal MONEY NOT NULL,  
    reserv_numnoletos INT NOT NULL )
```

Tabla: Tarjeta_Cred

```
CREATE TABLE tarjeta_cred (  
    tarj_num       INT NOT NULL,  
    ban_id         INT NULL,  
    tarj_vence     DATETIME NOT NULL,  
    tarj_codigo    INT NOT NULL )
```

```
ALTER TABLE banco
  ADD PRIMARY KEY (ban_id)

ALTER TABLE ciudad
  ADD PRIMARY KEY (ciu_id, pais_id)

ALTER TABLE cliente
  ADD PRIMARY KEY (cli_id)

ALTER TABLE cliente_tarjeta
  ADD PRIMARY KEY (tarj_num, cli_id)

ALTER TABLE evento
  ADD PRIMARY KEY (event_id, guia_id)

ALTER TABLE evento_patrocinador
  ADD PRIMARY KEY (event_id, pat_id, guia_id)

ALTER TABLE Guia_Turistico
  ADD PRIMARY KEY NONCLUSTERED (guia_id)

ALTER TABLE pais
  ADD PRIMARY KEY (pais_id)

ALTER TABLE patrocinador
  ADD PRIMARY KEY (pat_id)

ALTER TABLE reservacion
  ADD PRIMARY KEY (reserv_id, event_id, tarj_num, cli_id, guia_id)

ALTER TABLE tarjeta_cred
  ADD PRIMARY KEY (tarj_num)

ALTER TABLE ciudad
  ADD FOREIGN KEY (pais_id) REFERENCES pais

ALTER TABLE cliente
  ADD FOREIGN KEY (ciu_id, pais_id) REFERENCES ciudad

ALTER TABLE cliente_tarjeta
  ADD FOREIGN KEY (cli_id) REFERENCES cliente

ALTER TABLE cliente_tarjeta
  ADD FOREIGN KEY (tarj_num) REFERENCES tarjeta_cred
```

3.11 MAPA DE NAVEGACIÓN

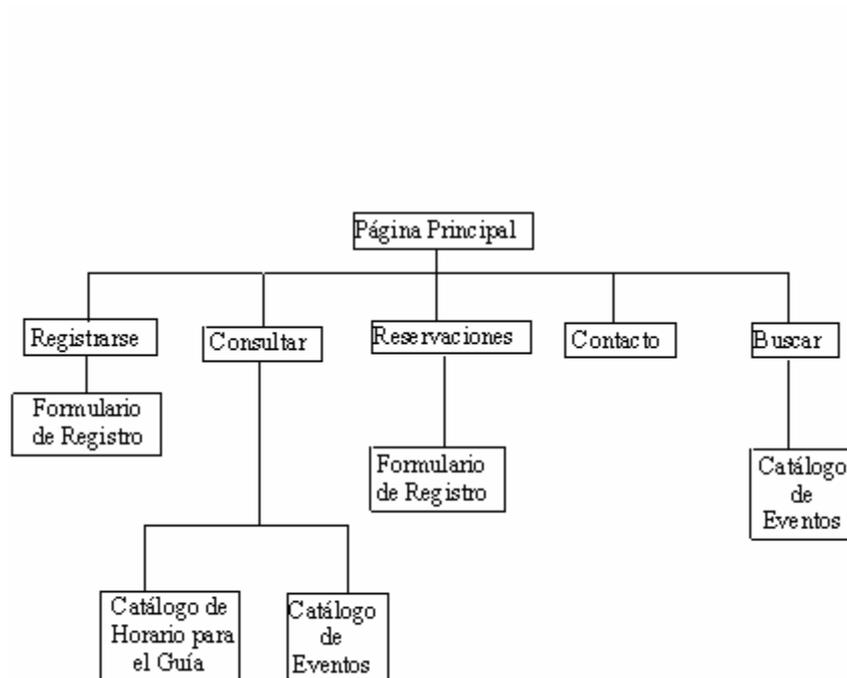


Fig. 21. Mapa de Navegación de la Aplicación

3.12 INTERFAZ DEL SISTEMA

El tipo de Web que se propone es un esquema elegante y con seriedad debido a que esto ayudara a que se considere un portal confiable y seguro, para la adquisición y reservación de boletos. Se propone utilizar colores claros, que no brillen mucho para que no afecten a la vista, la imágenes que se utilizan son pequeñas y en formato jpeg para que se carguen rápidamente y con calidad.

Se busca que la interfaz, permita a cualquier usuario experimentado o no, familiarizarse rápidamente con el sistema. Dicha interfaz pretende ser simple para que cualquier persona pueda acceder al sistema y pueda realizar las transacciones necesarias de forma fácil y rápida. El sistema va a llevar al usuario a través de paginas y formularios indicándole que datos ingresar para poder realizar la transacción que desee ya sea compra o reservación de boletos.



Fig 22. Página de Inicio del sistema

Interfaz del formulario que se tiene que llenar para poder realizar alguna reservación o compra de boletos.



Miércoles 28 de Marzo de 107

MéxicoTurístico.com
Diversión a tu alcance.

[Inicio](#) [Reservaciones](#) [Buscar](#) [Contacto](#)

Por favor, llena estos datos

Nombre:

Apellido Paterno:

Apellido Materno:

Dirección:

Código Postal:

Ciudad: D.F.

Delegación/ Municipio:

Teléfono:

E-mail:

Sexo: femenino masculino

Edad:

[Inicio](#) | [Buscar](#) | [Reservaciones](#) | [Información](#) | [Noticias](#) | [Contacto](#)

Derechos Reservados@2005.

Fig 23. Formulario para registrarse como usuario

Para poder realizar una reservación o compra es necesario que el usuario primero se registre y después ingrese al sistema con el usuario y password creados. Esta es la pantalla de acceso al sistema, donde le usuario ingresará sus datos y si no esta registrado existe un link en la parte inferior donde podrán registrarse.



MéxicoTurístico.com
Diversión a tu alcance.

Miércoles 26 de Abril de 2006

Inicio | Reservaciones | Buscar | Contacto

Por favor ingresa usuario y password.

Usuario :

Password :

Necesitas usuario y password? [Regístrate!](#)

Inicio | Buscar | Reservaciones | Información | Noticias | Contacto

Derechos Reservados©2006

Fig. 24. Formulario para registrarse como usuario

Pantallas del Administrador, en este caso se muestran algunas de las opciones a las que tendrá derecho únicamente el Administrador, como el link “Usuarios” a través del cual se decidirá a que usuarios se les puede quitar los permisos para entrar al sistema, y la opción “Reportes” donde podrá ver información referente a los usuarios que ingresaron al sistema por día, reportes sobre las ventas que se realizaron al día, mensualmente, etc.



Miércoles 28 de Marzo de 107

MéxicoTurístico.com
Diversión a tu alcance.

Inicio Reservaciones Buscar Contacto

Usuarios
Reportes
Inicio

Bloquear Usuarios

USUARIO	PASSWORD	ESTATUS	
Nancy Ramos	*****	BL	<input type="button" value="Guardar"/>
Juan Osorio	*****	AV	<input type="button" value="Guardar"/>
Arturo Ramos	*****	AV	<input type="button" value="Guardar"/>
Julio Santos	*****	AV	<input type="button" value="Guardar"/>
César Palacios	*****	BL	<input type="button" value="Guardar"/>
Julia Rosas	*****	BL	<input type="button" value="Guardar"/>
Alex Rios	*****	BL	<input type="button" value="Guardar"/>
Adrian Reyes	*****	AV	<input type="button" value="Guardar"/>
Natalia Rios	*****	AV	<input type="button" value="Guardar"/>

<< >>

Inicio | Buscar | Reservaciones | Información | Noticias | Contacto

Derechos Reservados@2006.

Fig. 25. Pantalla del Administrador

CAPITULO IV

POSIBILIDADES DE INTEGRACIÓN Y CRECIMIENTO DEL SISTEMA

Hasta hace algunos años Internet no tenía el auge que tiene actualmente, por ejemplo en 1994 había 3 millones de usuarios, en 1999 aproximadamente 300 millones, y en la actualidad existen aproximadamente 6000 millones de usuarios, este servicio ha ido creciendo año con año, por lo cual es evidente que un sistema orientado a la Web actualmente tiene la posibilidad de integración y crecimiento dentro de una empresa.

El propósito de este capítulo es mostrar las posibilidades de integración y crecimiento del sistema desarrollado, así como de exponer las características del mismo.

4.1 PRESENTE Y FUTURO DE INTERNET

Internet no está cambiando sólo nuestra forma de vivir, sino de pensar. Gracias a Internet, se pueden satisfacer deseos, materializar sueños, cubrir necesidades y comunicarse de forma instantánea.

Todos estos cambios se ven reflejados en estadísticas en las que el número de internautas a escala mundial a fines de 2005 sobrepasó mil millones, cuando en 1995 sólo existían 45 millones, y en 2000 eran 420 millones¹. Se espera que para 2011 se llegue a 2 mil millones de usuarios. Por consiguiente se calcula que en 10 años habrá más de 3 mil millones de internautas, es decir tres veces más que el día de hoy.

Internet es un medio poderoso que está cambiando nuestras vidas, pero aún tiene algunas limitaciones: lento, caro, lleno de información basura y con el potencial interactivo todavía limitado.

El conocimiento, la enseñanza, el aprendizaje, el entretenimiento y las transacciones comerciales tienen en Internet un excelente aliado, sobre todo con las comunicaciones por banda ancha. Internet significa interactividad, personalización y convergencia multimedia. Es el primer medio capaz de expresarse mediante textos enlazados, videos, gráficos animados, audio, fotos, y el primer medio accesible desde los más variados soportes, especialmente teléfonos móviles, agendas y libros electrónicos, televisores, coches o computadoras.

¹ Estadísticas de Usuarios Mundiales del Internet. <http://www.exitoexportador.com>.

Se considera que la educación, la salud, la investigación y el comercio serán cuatro de los mundos que más se beneficiarán de Internet, de su capacidad para extender el conocimiento y democratizar el acceso a la información, es por esto una de las razones importantes, para adentrarse en el desarrollo de software orientados a la Web.

Si bien es cierto que aún es un servicio costoso por causa de las facturas telefónicas, y que divide el mundo entre los que tienen acceso a la red y los que no, también es verdad que su capacidad de divulgar conocimiento la convierte en una fuente del conocimiento.

Internet esta transformando radicalmente el proceso de aprendizaje a distancia ya que introduce el concepto de campus virtual. El campus del ciberespacio trata de "crear en el ámbito de la pantalla todas las capacidades de relación y actuación que tiene la educación presencial. No sólo los contenidos educativos, sino también espacios de convivencia, asociación y reflexión".

Se estima que un 55% de los colegios y universidades norteamericanos tienen programas de educación virtual. Además, que hay aproximadamente 1 millón de estudiantes en programas virtuales y 13 millones en programas tradicionales. El Gartner Group estimaba que en el año 2002 más del 80% de los colleges y universidades de Estados Unidos estarán usando metodologías y tecnologías de educación a distancia en por lo menos un programa académico tradicional.

Ahora con los programas a distancia ya no es importante donde se encuentran los alumnos, ni tampoco en donde se encuentra el profesor; la educación a distancia es una realidad actualmente en algunas instituciones de México y en varios sistemas de enseñanza en el mundo, por lo cual se espera que en un futuro no muy lejano la educación a distancia sea una realidad en la mayoría de los lugares.

Algunas de las universidades más conocidas por tener programas a distancia son las de Phoenix, Michigan State, Carnegie Mellon, Colorado State, Indiana, California State, Indiana, Washington State, la Open University de la Gran Bretaña, la de Cataluña y el Instituto Tecnológico de Monterrey en México.

4.2 TENDENCIAS DEL SOFTWARE LIBRE

En los últimos años se ha comprobado que el software libre, es factible para el desarrollo. Ya que este software suministra soluciones factibles para las necesidades de distintos sectores de la información y la comunicación, particularmente para aquellos países “en vías de desarrollo” donde la mayoría de las personas y organizaciones no pueden solventar las elevadas tarifas de licencia que se deben de pagar por el software de dominio privado.

Parte importante del software libre es que abre el código fuente de las aplicaciones, para que los individuos puedan modificar y mejorar el software de acuerdo a sus necesidades y entorno.

En la actualidad los gobiernos como el sector privado están cada vez más al tanto de los beneficios del software libre. Ejemplo de ello es Indonesia, este gobierno ha dado inicio al proyecto “Indonesia se embarca en la fuente abierta” a fin de promover el uso de software libre a nivel nacional, a través de proyectos que habrán de alentar el desarrollo de software libre en el país.

Actualmente muchos países alrededor del mundo se están preocupando por la implementación del software libre, países tales como Malasia - Kuala Lumpur, Filipinas, han implementado planes y proyectos gubernamentales con los cuales tienen como propósito promover e incrementar el uso del software libre a nivel nacional y, por lo tanto, de maximizar los beneficios que pueden ser obtenidos con el software libre.

Hablando de los beneficios del software libre, este se puede vender y revender aplicaciones basadas en este tipo de software sin ningún limitante; con solo incluir en estas el código fuente y los créditos a quienes lo realizaron, esto constituye una oportunidad para las empresas que deseen dar soporte y crecer basados en esta filosofía.

En nuestros días el gobierno esta interesado en adoptar el software libre en la educación, como una plataforma que revolucione la cultura informática en México. Proponiendo crear software educativo para el aprendizaje de los alumnos, creado con software libre con la ayuda de varios países alrededor del mundo, si los países que adopten esta tecnología crean repositorios como los que actualmente existen para el desarrollo de otros paquetes de software, es bastante probable que se junten las condiciones necesarias para que germinen varios proyectos útiles para todos los gobiernos, ya que se contaría con la ayuda de los mejores pedagogos, psicólogos, maestros, etc. del mundo.

En las instituciones o empresas además de contar con aplicaciones para resolver ciertos problemas, se requiere software a la medida, específico, que pueda

atacar los problemas determinados, esto es posible con la utilización del software libre, permitiendo así la reutilización de código y la adaptación del mismo a las necesidades de cada cliente.

La creación de programas a través de la utilización del software libre, beneficiará a parte importante las empresas PYMES (micro, pequeñas y medianas), ya que podrán crecer poco a poco y sin tener problemas con la ley, por contar con software pirata, muy al contrario serán entes que participarán activamente cooperando en este modelo de desarrollo.

4.3 TIEMPOS DE DESARROLLO

En cuanto a los tiempos de desarrollo del sistema se implemento el diagrama de Gantt, para poder considerar cada una de las etapas del sistema y los tiempos en que se debería desarrollar cada etapa.

“El diagrama de Gantt consiste en una representación gráfica sobre dos ejes; en el vertical se disponen las tareas del proyecto y en el horizontal se representa el tiempo. “

Características:

- Cada actividad se representa mediante un bloque rectangular cuya longitud indica su duración; la altura carece de significado.
- La posición de cada bloque en el diagrama indica los instantes de inicio y finalización de las tareas a que corresponden.
- Los bloques correspondientes a tareas del camino crítico acostumbran a rellenarse en otro color.

Diagrama de GANTT

Fechas	2 Agosto 24 Agosto	25 Septiembre – 10 Octubre	11 Noviembre – 19 Diciembre	20 Diciembre – 10 Febrero	20 Marzo
Análisis de la Información					
Propuesta de Diseño Gráfico					
Diseño de la BD					
Integración y Pruebas					
Entrega del Sistema					

4.4 ETAPA DE PRUEBAS

Para la implementación de pruebas se consideraron algunos casos de usos sobre los cuales probar, como por ejemplo:

INICIO DE SESIÓN DE UN USUARIO	
Propósito:	Probar que los usuarios pueden iniciar sesión con el nombre de usuario apropiado y su contraseña.
Pre-Requisitos:	El usuario no ha iniciado sesión todavía. El usuario se registro anteriormente, la cuenta es válida.
Datos de Prueba:	Usuario = nombre del usuario Password = válida, inválida, vacío
Pasos:	<ol style="list-style-type: none"> 1. Entrar en Registrarse 2. teclear usuario 3. teclear password 4. hacer click en Entrar 5. verificar el mensaje de bienvenida si el inicio de sesión es correcto

BÚSQUEDA DE EVENTOS	
Propósito:	Probar que al ingresar palabras sobre un evento muestre todos los eventos relacionados. Deberán ver un mensaje indicando que han encontrado eventos y mostrarlos.
Prerequisitos:	El usuario no ha ingresado a la opción "Buscar".
Datos de Prueba:	Datos = palabras, nombre del evento

Pasos:	<ol style="list-style-type: none">1. Entrar al sistema2. Ingresar en la opción buscar3. Teclear palabras claves del Evento4. hacer click en Buscar5. ver: mostrar todos los eventos que contienen las palabras6. Si no existen coincidencias, mostrar mensaje “No existen eventos que coincidan”
--------	---

También se tomaron en cuenta para probar el sistema los siguientes puntos:

Probar el sistema en diferentes navegadores, debido a que uno de los objetivos del sistema es que este se pueda mostrar y utilizar desde diferentes computadoras y navegadores. Ya que existe la posibilidad de que el navegador interprete la misma página HTML de forma diferente (tamaño de imágenes, acentos, posiciones), se probó la versión en cada los navegadores soportados. En este caso se probó su funcionamiento en FireFox, InternetExplorer y Netscape. El resultado obtenido fue que se puede cargar correctamente el sistema en los exploradores, así como también muestra una buena calidad en las imágenes y velocidad al cargar las mismas.

Así también se revisaron las primeras versiones del sistema ya que existían algunas faltas de ortografía, asimismo se encontraron signos de puntuación capturados y procesados adecuadamente, pero no fueron impresos de forma correcta. Por lo cual se debe considerar la validación y la salida de caracteres especiales.

4.5 POSIBILIDADES DE INTEGRACIÓN

Actualmente el comercio electrónico, proyectos cooperativos, videoconferencia, etc., son muchos de los ambientes que se desarrollan en las empresas. Muchas de estas aplicaciones se están desarrollando o utilizándose ya en cierto grado.

La comunicación por Internet es lo de hoy, hacer publicidad, vender productos, comprar productos, educación a distancia, proporcionar servicios a través de este medio de comunicación es fundamental en estos momentos, es por esto que se considera que un sistema de esta índole puede tener grandes beneficios para el ramo turístico.

La integración de aplicaciones de esta índole permite el acceso a información y documentación de la empresa de forma rápida y en tiempo real.

Como se ha visto el sistema, está orientado al turismo de la ciudad de México, a los eventos que se presentan dentro de la ciudad. Una de las posibilidades de integración del sistema se podría llevar a cabo implantando este sistema en la secretaria de turismo.

La integración del sistema en alguna de las corporaciones gubernamentales, se considera factible, debido a que el turismo es uno de los temas de interés en los estados de la república, y poner una opción en la cual se puedan mostrar todas las opciones turísticas con las que cuentan los estados o con las que cuenta el país ayudara a incrementar la derrama económica que proporciona el turismo.

Actualmente los estados de la república cuentan con una página Web en la cual se exponen los sitios turísticos y eventos que se presentan, este sistema se podría integrar a la página Web para proporcionar servicios de venta, reservaciones y contrataciones de guías turísticos.

Algunas de las características por las que el sistema tiene posibilidades de integración son las siguientes:

- Es fácilmente extensible, el tiempo de creación de una nueva funcionalidad se considera pequeña.
- Está basado en software *open source* ampliamente utilizado y probado, con una gran comunidad de usuarios, lo que proporciona un soporte de gran calidad.
- Dispone de mecanismos de seguridad y gestión de permisos, lo que permite mostrar distinta información según el nivel del usuario así como definir las acciones que pueden realizar y las que no.

- Soporta los sistemas de gestión de bases de datos relacionales: MySQL, PostgreSQL², Oracle³, DB2, etc.
- Es *open source*, proporcionando entre otras ventajas una gran flexibilidad.
- El diseño se basa en tres capas lo cual permite separar la interfaz y el código, lo cual permite una mejor administración.
- Las aplicaciones en tres capas ofrece mayor capacidad de crecimiento y son más sencillas de mantener.

² PostgreSQL. Es un motor de base de datos, es servidor de base de datos relacional libre.

³ Oracle, actualmente es considerado uno de los mejores SGBD

4.6 CRECIMIENTO DEL SISTEMA

El desarrollo de aplicaciones web ha ido tomando importancia con el paso del tiempo, a medida que crece la complejidad del problema, aparecen soluciones nuevas, es por esto que un sistema como este tiene grandes posibilidades debido a que su programación permite reutilizar código.

El sistema en esta primera versión solo abarca la ciudad de México, una parte importante para el crecimiento del sistema sería abarcar la mayoría de los estados del país o que en cada estado se implemente un sistema de este tipo, con lo cual se darían a conocer más sobre los sitios turísticos. Lo que significaría que el sistema tendría que contar con un administrador en cada estado para el mantenimiento del sistema.

Esto también implica que el número de proveedores crecería considerablemente, y por lo tanto el número de usuarios.

Por otra parte el sistema debe presentar algunas mejoras para mantener un alto funcionamiento y satisfacción del cliente, por ejemplo se sugiere:

- Agregar un contador en el inicio de la página, para verificar el número de visitas por día, ayudara al administrador a tener un control sobre el número de usuarios que utilizan el sistema en el día.
- Manejar la publicidad o los banners desde la base de datos.
- Mejoras al diseño grafico de la página web, para que esta pueda contener más información y un mejor aspecto visual para los usuarios.

CONCLUSIONES

El presente trabajo presentó los estándares y lenguajes utilizados para el desarrollo de sistemas orientados al Web. Además se presento una nueva propuesta de sistema en la cual se demuestra que implementando todas las técnicas vistas es posible implementar un sistema que proporcione beneficios.

Una aplicación Web en la actualidad es un sistema completo, robusto el cual debe de contar con todos lo requerimientos de un sistema, comenzando desde el análisis y documentación. El sistema que se diseño permite realizar y administrar las operaciones de compra y reservación de boletos de forma rápida y eficiente.

Un sistema orientado al Web nos permite ser eficientes teniendo como resultados un tiempo de ejecución más rápido que lo normal, además mantener y actualizar un sistema de este tipo sin distribuir e instalar software en miles de clientes.

A través del trabajo expuesto se comprendió que uno de los aspectos claves del diseño de una aplicación es su arquitectura. Ya que lo largo del ciclo de vida de una aplicación, el uso del modelo de 3 capas brinda beneficios tales como:

- Reusabilidad
- Flexibilidad
- Simplicidad de administración y mantenimiento
- Escalabilidad

Por otro lado, el implementar software open source proporciona la posibilidad de modificar y reutilizar código, lo que implica que se puede personalizar o adecuar el código a las soluciones y necesidades de cada cliente.

Al implementar software libre como Java nos brinda además rapidez e independencia de plataforma, parte importante en el desarrollo de sistemas.

El desarrollo de sistemas orientados al Web, proporciona grandes beneficios, entre los cuales podemos citar:

- Facilidad de administración.
- Implementación en Internet como en Intranets.
- Facilidad de uso de las interfaces web.
- Multiplataforma
- Actualización fácil

- Acceso móvil

También otro de los aspectos a tener en cuenta es que una vez realizada una aplicación Web, ya sea para uso interno de una empresa o el público en general, al integrar nuevas funcionalidades tiene un costo mínimo.

Al usar las alternativas propuestas en el presente trabajo para el desarrollo de aplicaciones Web, se obtienen sistemas de información robustos de gran desempeño, estabilidad, escalabilidad y un costo menor al que se tendría al usar software propietario.

El desarrollo de sistemas orientados al Web es un campo que se puede seguir explotando, ya que la red de redes crece día a día.

BIBLIOGRAFÍA

- Schmuller, Joseph; Aprendiendo UML en 24 horas, Pearson.
- Houlette, Forest. Fundamentos de SQL. 2ª Ed, McGraw-Hill 2003.
- Tordio M, Angel M. Java 2, 1ª Ed., Anaya,

Referencias de Internet:

- <http://www.programacion.com/java/>
- <http://www.mysql.com/>
- <http://es.wikipedia.org>
- <http://portals.apache.org>
- <http://www.softwarelibre.gob.mx>