

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN TECNOLÓGICA

ANÁLISIS DE SISTEMAS INFORMÁTICOS

**DISEÑO E IMPLEMENTACION DE UN SITIO WEB PARA
LA IMPORTADORA WANG**

SANTIAGO RENÉ TIBÁN TOAPANTA

CRISTIAN VINICIO MORALES QUINGA

DIRECTOR: ING. LUZ MARINA VINTIMILLA

Quito, Marzo del 2006

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Cristian Vinicio Morales Quinga y Santiago René Tibán Toapanta, bajo mi supervisión.

Ing. Luz Marina Vintimilla
DIRECTOR DE PROYECTO

DECLARACIÓN

Nosotros, Cristian Vinicio Morales Quinga , Santiago René Tibán Toapanta, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo , a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

CRISTIAN VINICIO MORALES

SANTIAGO RENÉ TIBÁN

AGRADECIMIENTO

A Dios por darme salud, sabiduría y fortaleza
para poder culminar este trabajo
y a todas aquellas personas que confiaron en mí,
y que, con su apoyo, dedicación y sacrificio
hicieron posible que culminara esta etapa de mi vida,
en especial a mis padres y hermanos.

También un agradecimiento especial a
nuestros profesores, pues gracias
a sus conocimientos impartidos,
estamos culminando esta etapa estudiantil,
en especial a nuestra Directora de Proyecto,
Ing. Luz Marina Vintimilla

Vinicio

AGRADECIMIENTO

A Dios por ser mi guía espiritual en mi carrera,
por darme la luz cuando más lo necesitaba
y cambió el rumbo de mi vida

A mi esposa que con su cariño
y comprensión pudo llevarme
a la culminación de esta meta,
en especial a mi hijo
que con su ternura y amor
hace que en esta vida todo sea posible

Santiago

DEDICATORIA

Dedico este trabajo con todo mi corazón
a mis padres y hermanos,
pues ellos han sido, son y serán,
parte fundamental en mi vida, por el sacrificio,
dedicación y confianza que pusieron en mí.

En especial lo dedico a mi Madre
que con su comprensión, cariño
y apoyo incondicional
me ha motivado a conquistar mis metas.

Vinicio

DEDICATORIA

A mi familia a quienes con su apoyo
económico y moral hicieron posible
la culminación de este trabajo
y en especial a mi madre
un agradecimiento profundo por el
apoyo brindado para alcanzar una
meta más en la vida.

Santiago

CONTENIDO

PRESENTACIÓN	I
RESUMEN	II

CAPÍTULO 1 **01**

1.1. ÁMBITO	01
1.2. EL PROBLEMA	01
1.3. FORMULACIÓN	02
1.4. SISTEMATIZACIÓN	02
1.5. OBJETIVO GENERAL	02
1.6. OBJETIVOS ESPECÍFICOS	02
1.7. JUSTIFICACIÓN PRÁCTICA	03
1.8. ALCANCE	03
1.9. LIMITACIONES	03
1.10. PRESUPUESTO Y FUENTES DE FINANCIAMIENTO	04
1.10.1. Recursos Humanos	04
1.10.2. Software	04
1.10.3. Hardware	04
1.10.4. Varios	04

CAPÍTULO 2 **05**

2.1. APLICACIÓN E INGENIERIA WEB	05
2.2. DISEÑO DE PÁGINAS WEB	06
2.2.1. ETAPAS	06
2.2.2. FUNDAMENTOS	07
2.2.3. APLICACIONES WEB Y EL DESARROLLO DE CAPAS	08
2.2.4. SERVIDORES WEB	09
2.2.4.1. Software Intermedio	10
2.2.5. PÁGINAS DINÁMICAS Y PÁGINAS ESTÁTICAS	12
2.2.5.1. Páginas Dinámicas	13
2.2.5.2. Páginas Dinámicas Cliente	14
2.2.5.3. Páginas Dinámicas Servidor	15
2.3. CONCEPTOS RELACIONADOS	16
2.3.1. DIRECCIÓN IP	16
2.3.1.1. Máscara de Red	17
2.3.1.2. Máscara de Red por Defecto	17
2.3.2. INTERNET	18
2.3.2.1. Internet y Sociedad	19
2.3.3. W.W.W.	19
2.3.3.1. Características	19
2.3.4. SITIO WEB	20
2.3.5. PORTAL WEB	21
2.3.6. HTTP	21

2.3.6.1. Características	21
2.3.6.2. Etapas en una Conexión HTTP	22
2.3.7. JAVASCRIPT	23
2.3.8. HTML	24
2.3.9. PHP	25
2.3.10. DOMINIO	26
2.3.11. WEB HOSTING	27
2.3.11.1. WINDOWS HOSTING Y LINUX HOSTING	28
2.3.11.1.1. Windows Hosting	28
2.3.11.1.2. Linux Hosting	28

CAPÍTULO 3 **30**

3.1. TABLA DE RESUMEN DE LA METODOLOGÍA	30
3.2. PARADIGMA ESPIRAL ORIENTADO A LA WEB	31
3.3. METODOLOGÍA	32
3.4. METODOLOGÍA OOHDM	33
3.5. FASES DE LA METODOLOGÍA OOHDM	34
3.5.1. CAPTURA DE REQUERIMIENTOS	34
3.5.1.1. Descripción de los Usuarios del Sistema	35
3.5.2. DISEÑO CONCEPTUAL	35
3.5.3. DISEÑO NAVEGACIONAL	36
3.5.4. DISEÑO ABSTRACTO DE INTERFAZ	38
3.5.5. IMPLEMENTACIÓN	39
3.5.6. PRUEBAS	39
3.6. U.M.L. (Lenguaje Unificado de Modelamiento)	40
3.6.1. OBJETIVOS DE U.M.L.	41
3.6.2. AREAS CONCEPTUALES DE U.M.L.	41
3.6.2.1. Estructura Estática	41
3.6.2.2. Comportamiento Dinámico	42
3.6.2.3. Construcciones de Implementación	42
3.6.2.4. Organización del Modelo	42
3.6.3. DIAGRAMAS U.M.L.	43
3.6.3.1. DIAGRAMAS DE CASOS DE USO	43
3.6.3.2. DIAGRAMAS DE CLASES	45
3.6.3.3. DIAGRAMAS DE OBJETOS	48
3.6.3.3.1. OID (Object Identifier)	49
3.6.3.3.2. Características de un Objeto	50
3.6.3.4. DIAGRAMAS DE SECUENCIA	50
3.6.3.4.1. Elementos de un Diagrama de Secuencia	51
3.6.3.5. DIAGRAMA DE COLABORACIÓN	52
3.6.3.5.1. Elementos de un Diagrama de Colaboración	53
3.6.3.6. DIAGRAMAS DE ESTADOS	54
3.6.3.6.1. Elementos de un Diagrama de Estados	54
3.6.3.7. DIAGRAMA DE ACTIVIDADES	55
3.6.3.7.1. Características del Diagrama de Actividades	57
3.6.3.8. DIAGRAMA DE COMPONENTES	57
3.6.3.8.1. Componentes	58

3.6.3.8.2. Identidad	59
3.6.3.9. DIAGRAMA DE DESPLIEGUE	59

CAPÍTULO 4 **61**

4.1. HERRAMIENTAS DE CONSTRUCCIÓN WEB	61
4.2. MICROSOFT FLASH MX 2004	61
4.3. MICROSOFT DREAMEAVER MX 2004	62
4.4. MICROSOFT FIREWORKS MX 2004	63
4.5. SERVIDOR WEB APACHE	63
4.6. MYSQL	65

CAPÍTULO 5 **66**

5.1. CONCLUSIONES	66
5.2. RECOMENDACIONES	67
BIBLIOGRAFÍA	68

ANEXOS **70**

PRESENTACIÓN

En vista del gran desarrollo que a tenido el Internet en los últimos tiempos, el crecimiento explosivo de las transacciones vía Web, a abierto la posibilidad de acceder de una forma rápida y efectiva a la información desde cualquier parte del mundo.

La importadora desea mostrar al exterior su imagen mediante la implementación de un sitio web que le permita al visitante obtener información de los productos y servicios que ofrece la importadora.

Para el desarrollo de nuestro Sitio se ha empleado Software Libre (Open Source), con el fin específico de reducir los costos finales en desarrollo e implantación.

En el desarrollo de nuestro Sitio se ha utilizado Apache 1.3.23, el mismo que al estar integrado con MySQL 3.23.48 permite una interconexión consistente y de calidad de la web con la base de datos

La interfaz fue diseñada con software de apoyo como Macromedia Flash MX2004, Macromedia Dreamweaver MX2004 y Macromedia Fireworks MX2004, las mismas que proporcionan una combinación potente de herramientas visuales para el desarrollo de aplicaciones Web, ésto, además permite que los diseñadores y desarrolladores manejen cualquier sitio web con facilidad.

La metodología OOHDM nos propone una herramienta sencilla, probada, y ajustable a las necesidades del usuario para planificar, diseñar y producir un sitio web de calidad

RESUMEN

La Importadora requiere de un Sitio Web dinámico que responda a las expectativas de sus usuarios y permita mostrar una imagen positiva, responsable, y seria al medio externo.

En el proyecto se incluye los elementos necesarios para un buen diseño de aplicaciones web.

Capítulo 1

En el presente documento se detallara los aspectos generales como son: El ámbito del problema, el objetivo general, los objetivos específicos, el alcance, las limitaciones y el presupuesto para el proyecto.

Capítulo 2

Se hace una descripción de la ingeniería de software aplicado al diseño de aplicaciones web en la que se incluyen: Ingeniería Web y conceptos relacionados al desarrollo de las mismas.

Capítulo 3

Aquí se detalla el diseño amparados en el paradigma espiral orientado a la web que proporciona un modelo evolutivo para el desarrollo de sistemas de software complejos y la metodología OOHDM

Capítulo 4

En esta secciona se realiza una descripción de las herramientas de diseño web utilizadas para la elaboración de nuestro sitio

Capitulo 5

Al finalizar este documento se presentan las conclusiones y recomendaciones, basadas en el estudio y en el desarrollo del sitio web de la importadora

CAPITULO 1

1.1. ÁMBITO

LA IMPORTADORA WANG esta ubicada en la Av. 6 de Diciembre N22-39 y Veintimilla, es un distribuidor oficial de copadoras con una experiencia de 4 años dentro del sector, ofrece sus servicios de venta de copadoras, impresoras, repuestos, suministros, mantenimiento y recarga de cartuchos. Además del local principal (Matriz) cuenta con una sucursal en el Dorado Iquique N14-296 y Yaguachi donde al mismo tiempo brinda los servicios de un cibercafé.

En vista del gran desarrollo que a tenido el Internet en los últimos años, el crecimiento explosivo de transacciones vía Web, han motivado a que la IMPORTADORA WANG decida extender sus horizontes hacia todo el territorio ecuatoriano, mediante la implementación de un portal Web que permita a los usuarios obtener información de la Importadora, registrarse como potenciales clientes, descargar catálogos, descargar proformas y hacer los respectivos pedidos de los productos y servicios que ofrece la Importadora Wang.

1.2. EL PROBLEMA

Las necesidades de la comunicación externas a la importadora son mas necesarias día a día por lo que se ha planteado el desarrollo de un proyecto de información Web que permita mejorar nuestros servicios y la atención al cliente.

El Sitio Web de la Importadora, además, debe permitir promocionar y atender solicitudes de pedidos de venta los diferentes productos con su respectiva variedad de características y costos, brinda la oportunidad de realizar sus consultas técnicas y comerciales a través de esta Web y de esta manera mejorar las interrelaciones hacia sus clientes o visitantes en el exterior de la misma.

Ya que su ubicación geográfica se encuentra en una zona muy comercial se pretende dar a conocer nuevos productos de una forma mas detallada y especifica en el menor tiempo posible.

1.3. FORMULACIÓN

¿Como dar a conocer nuevos productos y servicios además de promocionar los productos tradicionales de la Importadora WANG, con sus respectivas características y costes, utilizando tecnología de punta y manejando la información para el cliente a través de la red Internet?

1.4. SISTEMATIZACIÓN

- ¿Es posible administrar una información detallada de cada uno de los productos y servicios que ofrece la importadora?
- ¿Es posible tener un registro de los clientes que tiene la importadora?
- ¿Es posible descargar proformas y catálogos interactivamente a través del Internet?
- ¿Es posible mejorar la imagen de la importadora a través de una correcta información de ella, su razón de ser, su misión, visión, etc. Y además de ayudar al visitante con información de los lugares donde se puede hacer la entrega a domicilio con sus respectivos costos y tiempos de entrega?.
- ¿Se pueden manejar estadísticas de inquietudes comentarios y sugerencias que los visitantes o clientes tienen para con la importadora?

1.5. OBJETIVO GENERAL

Diseñar e implementar un portal Web que permita a la empresa promocionar y atender solicitudes de pedidos de ventas de sus diferentes productos, tanto los nuevos como los tradicionales con sus respectivas características y costes.

1.6. OBJETIVOS ESPECIFICOS

- 1) Implementar un módulo que permita a la empresa anunciar sus Productos y Servicios, actualizando la información constantemente.
- 2) Implementar un módulo que permita acceder a una zona exclusiva de clientes.

- 3) Implementar un modulo que permita al usuario acceder y descargar catálogos y/o proformas , así como también las promociones
- 4) Implementar un modulo que permita al usuario acceder a la información de la empresa, alcance geográfico, tiempo de entrega, y costes.
- 5) Implementar un modulo que permita al usuario registrar sus inquietudes comentarios y sugerencias

1.7. JUSTIFICACIÓN PRÁCTICA

El portal Web que se ha presentado esta enfocado a facilitar y dar comodidad, al momento de acceder a la información a visitantes, clientes y administradores:

- A Visitantes al momento de buscar información (costos y características) sobre un tipo de producto que les agrada o les interese adquirir.
- A Clientes al momento en que reciban catálogos de nuevos productos o servicios que pueda ofrecer la empresa, así como también de las ofertas y promociones.
- A Administradores al momento de recibir vía correo electrónico, las inquietudes, comentarios y sugerencias de sus clientes o visitantes, lo que servirían como referencia para mejorar la imagen de la Empresa.

1.8. ALCANCE

Diseñar y Programar los componentes necesarios para el desarrollo del Sitio Web de la Importadora Wang. Además de Incluir la información documental definida por la institución dentro del sistema propuesto

1.9. LIMITACIONES

El Sitio Web desarrollado para la Importadora Wang presenta las siguientes limitaciones:

- No se incluye lo que son ventas directas On-Line.
- No se implementará un foro de soporte técnico

1.10. PRESUPUESTO Y FUENTES DE FINANCIAMIENTO

1.10.1. Recursos Humanos

PERSONAL	TOTAL HORAS (6 MESES)	COSTO HORA	TOTAL
Analista	150	8.00	1200
Desarrollador	150	6.00	900
TOTAL			2100

1.10.2. Software

DETALLE	COSTO TOTAL
MACROMEDIA Dreamweaver Mx 2004 Flash Mx 2004 Fireworks Mx 2004	Free
APACHE (1.3.23) Mysql (3.23.48) Php (4.1.1)	
Rational Rose 2000 Enterprise Edition	300
TOTAL	300

1.10.3. Hardware

DETALLE	COSTO TOTAL
Computador Pentium IV	700
Impresora	70
TOTAL	770

1.10.4. Varios

DETALLE	COSTO TOTAL (6 MESES)
Suministros de Oficina	60
Internet	120
Otros	120
TOTAL	300

TOTAL COSTO PROYECTO \$ 3470

CAPITULO 2

2.1 APLICACION E INGENIERIA WEB

Una Aplicación Web es un sistemas de software cuya funcionalidad es mostrada a través de la plataforma World Wide Web (WWW). Desarrollar este tipo de aplicaciones es un proceso muy complicado que demanda altos conocimientos técnicos y experiencia por parte del personal involucrado. Además, este proceso de desarrollo no parece estar acompañado de métodos adecuados que garanticen la calidad la calidad de las aplicaciones Web.

Una Aplicación Web consiste en un software basado en Internet, en el cual una población extensa de usuarios, por medio de un navegador, hacen peticiones remotas y esperan una respuesta que puede implicar una mezcla de publicación impresa y desarrollo de software, de mercadeo e informática, de comunicaciones internas y relaciones externas y de arte y tecnología. Una aplicación Web se distingue por utilizar hipermedia para presentar al usuario el contenido de textos, gráficos, sonido y video, por su actualización constante, por su inmediatez y por la capacidad de interactuar con otros elementos de Internet . (*M. Mendoza 2005*).

El desarrollo de aplicaciones Web es más que un medio de manipulación y creación de presentaciones ya que además debe incluir análisis de necesidades, diseño, gerencia, métricas, mantenimiento, etc.

La ingeniería Web es el establecimiento y uso de conocimiento científico, principios de gerencia e ingeniería y el enfoque sistemático y disciplinado para desarrollar, entregar y mantener, aplicaciones y sistemas basados en Web de alta calidad.

La aplicación de principios de ingeniería pueden evitar el caos potencial al que nos enfrentamos, y poner bajo control el desarrollo de las aplicaciones *Web*, minimizando riesgos y mejorando el mantenimiento y la calidad. (*M. Mendoza 2005*).

2.2 DISEÑO DE PAGINAS WEB

El diseño de una buena aplicación web conlleva la investigación, una planificación cuidadosa y una comprobación completa de las ideas de su aplicación web, al crear un sitio se debe tener en cuenta como se va a organizar la información y como explorarán los usuarios la aplicación web,

El diseño de páginas web es una amplia área de aplicación del diseño gráfico en la cual se integran conocimientos propios del diseño como son la composición, el uso de color y la tipografía, con conocimientos técnicos del medio como son los lenguajes HTML (HiperText Markup Language) y CSS (Cascade Style Sheets), así como también con conocimientos sobre usabilidad, accesibilidad y estructura de un sitio web.

Actualmente, el diseño de páginas Web se ha convertido en algo cotidiano. Se trata de realizar una página y darle formato, imprimirle un estilo, etc.

Lo anterior se puede conseguir de varias maneras:

- Creando directamente el código HTML con un procesador de texto plano, como el mismo Block de Notas de Windows.
- Utilizando un programa WYSIWYG de creación de páginas como Adobe GoLive, Macromedia Dreamweaver, NVU o Microsoft Frontpage.
- Utilizando lenguajes de programación del lado del servidor como ASP, .NET, JSP, PHP, Perl, etc para generar la página web.
- Usando un sistema CMS (Web Content Management System) que permite separar el contenido del diseño y que permite al diseñador generar solamente las plantillas, de modo que el contenido sea administrado en un área separada. Algunos sistemas CMS son el PhpNuke, Plone y Mambo. (*Enciclopedia Wikipedia, 2005*)

2.2.1 ETAPAS

El diseño de páginas web tiene dos etapas. La primera es el diseño visual de la información que se desea editar. En esta etapa se trabaja en el papel distribuyendo el texto, los gráficos, los vínculos a otros documentos y otros objetos multimedia que se consideren pertinentes.

Una vez que se tiene este boceto se pasa a 'programar' la página web. Para esto, y fundamentalmente para manejar los vínculos entre documentos, se creó el lenguaje de marcación de hipertexto o HTML.

Al diseñar nuestro sitio la primera etapa estuvo constituida en el Diagrama Navegacional, mientras que la segunda etapa constituyó el diseño en sí del Sitio Web.

El HTML consta de una serie de etiquetas que estructuran el texto y son presentados en forma de hipertexto por los navegadores. Esto se puede hacer con un simple editor de textos (debe guardarse como texto plano, sin ningún tipo de formato). Aprender HTML es relativamente fácil, así que es sencillo crear páginas web de este modo. Esta era la única manera de generarlas hasta que aparecieron, a mediados de 1996, algunos editores visuales de HTML, como MS FrontPage y Macromedia Dreamweaver. Con estas herramientas no es absolutamente necesario aprender HTML (aunque sí aconsejable), con lo cual el desarrollador se concentra en lo más importante que es el diseño del documento. *(Enciclopedia Wikipedia, 2005)*

2.2.2. FUNDAMENTOS

Un correcto diseño web implica conocer cómo se deben utilizar cada una de las etiquetas permitidas en HTML, es decir, hacer un uso correcto de este lenguaje dentro de los estándares establecidos por la W3C y en lo referente a la web semántica. Debido a la persimibilidad de algunos navegadores web como Internet Explorer, esta premisa original se ha perdido. Por ejemplo, este navegador permite que no sea necesario cerrar las etiquetas del marcado, utiliza código propietario, etc. Esto impide que ese documento web sea universal e independiente del medio que se utilice para ser mostrado.

La web semántica, por otra parte, aboga por un uso lógico de las etiquetas según el significado para el que fueron concebidas. Por ejemplo se utilizará la etiqueta <P> para marcar párrafos, y <TABLE> para tabular datos (nunca para disponer

visualmente los diferentes elementos del documento). En su última instancia, esto ha supuesto una auténtica revolución en el diseño web puesto que apuesta por separar totalmente el contenido del documento de la visualización.

De esta forma se utiliza el fichero HTML únicamente para organizar y estructurar la información y las hojas de estilo CSS para indicar como se mostrará dicha información en los diferentes medios (como por ejemplo, una pantalla de ordenador, un móvil, impreso en papel, leída por un sintetizador de voz, etc.).

También existen páginas dinámicas, las cuales permiten mayor interactividad entre la web y el visitante, proporcionándole herramientas dinámicas tales como buscadores, chat, foros, sistema de encuestas, etc. y poseen de un panel de control de administración de contenidos. Este permite crear, actualizar y administrar cantidades ilimitadas de contenido. (*Enciclopedia Wikipedia, 2005*).

2.2.3. APLICACIONES WEB Y EL DESARROLLO DE CAPAS

Las aplicaciones hipermedia han evolucionado en los últimos años y se han concentrado mayormente en la web. Las antiguas aplicaciones distribuidas en cd's dieron lugar a aplicaciones dinámicas, de constante actualización e incluso personalizables, capaces de adaptarse a los tipos de usuarios y en casos avanzados, a cada usuario en particular. Estas características encuentran el medio ideal en la web, ya que de otra forma sería costoso su mantenimiento y evolución. (*D. Silva, 2005*)

La complejidad del desarrollo ocurre a diferentes niveles: dominios de aplicación sofisticados (financieros, médicos, geográficos, etc.); la necesidad de proveer acceso de navegación simple a grandes cantidades de datos multimediales, y por último la aparición de nuevos dispositivos para los cuales se deben construir interfaces web fáciles de usar. Esta complejidad en los desarrollos de software sólo puede ser alcanzada mediante la separación de los asuntos de modelización en forma clara y modular.

La metodología OOHDMM empleada en nuestro proyecto ha sido utilizada para diseñar diferentes tipos de aplicaciones hipermedia como galerías interactivas, presentaciones multimedia. El éxito de esta metodología es la clara identificación de los tres diferentes niveles de diseño en forma independiente de la implementación.

La justificación de tanto trabajo puede encontrarse en cualquier aplicación que requiera navegación: en términos de programación orientada a objetos, si los elementos por los que se navega son los del diseño conceptual se estaría mezclando la funcionalidad hipermedia con el comportamiento propio del objeto. Por otro lado, si los nodos de la red de navegación tienen la capacidad de definir su apariencia, se estaría limitando la extensión de la aplicación para ofrecer nuevas presentaciones del mismo elemento y eventualmente se estaría dificultando la personalización de la interfaz.

Es necesario, entonces, mantener separadas las distintas decisiones de diseño según su naturaleza (conceptual, navegacional, de interfaz) y aplicar las tecnologías adecuadas a cada capa en el proceso de implementación. (*D. Silva, 2005*).

2.2.4. SERVIDORES WEB

Un servidor web es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.

Sin embargo, el hecho de que HTTP y HTML estén íntimamente ligados no debe dar lugar a confundir ambos términos. HTML es un formato de archivo y HTTP es un protocolo

Un servidor web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que

el cliente solicita. A modo de ejemplo, al teclear <http://www.google.com/> en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

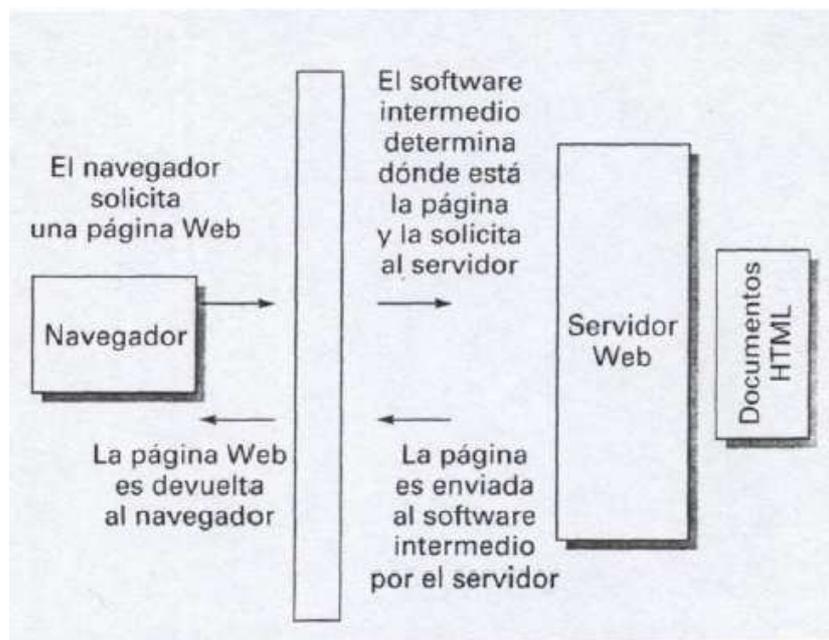
Sobre el servicio web clásico podemos disponer de aplicaciones web. Éstas son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- Aplicaciones en el lado del cliente: el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Normalmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje javascript y java, aunque pueden añadirse mas lenguajes mediante el uso de plugins
- Aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP. (*Enciclopedia Wikipedia, 2005*).

2.2.4.1. SOFTWARE INTERMEDIO (Middleware)

Hasta el momento probablemente tengamos la impresión de que la comunicación entre cliente y servidor es directa. Desgraciadamente, esto no es verdad: normalmente existe por lo menos una capa de software entre ellos. Esta capa se llama software intermedio (*middleware*). Esta muestra la comunicación entre un cliente ejecutando un navegador como Internet Explorer y un servidor Web.

GRÁFICO 1.1



Fuente: <http://www.desarrolloweb.com/articulos/middleware>

Aquí el software intermedio que se encuentra entre el servidor Web y el cliente que ejecuta el navegador Web intercepta las peticiones que proceden del navegador. Si se hace una petición para una página Web entonces determina la localización del documento Web y envía una petición para esa página. El servidor responde a la petición y devuelve la página al software intermedio, quien la dirige al navegador que la visualizará en la pantalla del monitor que utiliza el cliente.

Existen dos categorías de software intermedio: el software intermedio general y el software intermedio de servicios.

El software intermedio general es el que está asociado a los servicios generales que requieren todos los clientes y servidores. El software típico que se utiliza como tal es:

- El software para llevar a cabo comunicaciones utilizando el protocolo TCP/IP y otros protocolos de red.
- El software del sistema operativo que por ejemplo, mantiene un almacenamiento distribuido de archivos.
- El software de autenticación, el cual comprueba que un usuario que desee utilizar un sistema distribuido pueda en efecto hacerlo.

- El software intermedio orientado a mensajes que gestiona el envío de mensajes desde clientes a servidores y viceversa.

El software intermedio de servicios es el software asociado a un servicio en particular. Entre los ejemplos típicos de este tipo de software se incluyen:

- Un software que permite a bases de datos diferentes conectarse a una red cliente/servidor.
- Un software específico de objetos distribuidos.
- Un software intermedio de red asociado al protocolo http
- Un software intermedio asociado a productos de seguridad específicos.

(Enciclopedia Wikipedia, 2005)

2.2.5. PÁGINAS DINÁMICAS Y PÁGINAS ESTÁTICAS

En la web podemos encontrar, o construir, dos tipos de páginas:

- Las que se presentan sin movimiento y sin funcionalidades más allá de los enlaces
- Las páginas que tienen efectos especiales y en las que podemos interactuar.

Las primeras páginas son las que denominamos páginas estáticas, se construyen con el lenguaje HTML, que no permite grandes florituras para crear efectos ni funcionalidades más allá de los enlaces.

Estas páginas son muy sencillas de crear, aunque ofrecen pocas ventajas tanto a los desarrolladores como a los visitantes, ya que sólo se pueden presentar textos planos acompañados de imágenes y a lo sumo contenidos multimedia como pueden ser videos o sonidos.

El segundo tipo de páginas se denomina página dinámica. Una página es dinámica cuando se incluye cualquier efecto especial o funcionalidad y para ello es necesario utilizar otros lenguajes de programación, aparte del simple HTML.

Mientras que las páginas estáticas todo el mundo se las puede imaginar y no merecen más explicaciones, las páginas dinámicas son más complejas y versátiles.

2.2.5.1. PÁGINAS DINÁMICAS

Como hemos visto, una página es dinámica cuando realiza efectos especiales o implementa alguna funcionalidad o interactividad.

Además, hemos visto que para programar una página dinámica necesitaremos otros lenguajes aparte del HTML. Sin embargo, nunca hay que olvidarse del HTML, ya que éste es la base del desarrollo web: generalmente al escribir una página dinámica el código de los otros lenguajes de programación se incluye embebido dentro del mismo código HTML.

Una razón por la que construiremos una página dinámica es la simple vistosidad que pueden alcanzar los trabajos, ya que podemos hacer presentaciones más entretenidas de las que se consiguen utilizando únicamente HTML. Pero vamos a ver con calma algunas razones menos obvias pero más importantes.

En nuestro proyecto hemos decidido realizar un sitio para la Importadora Wang donde una de las informaciones principales a proveer podría ser la presentación de productos. Efectivamente, esta información suele ser dada por la proveedora de la Importadora y podría ser muy fácilmente almacenada en una base de datos. Si trabajásemos con páginas HTML, tendríamos que construir una página independiente para cada vez que salga un nuevo producto o cada vez que un producto se deje de producir o que ya no se tenga en stock, además tendríamos que introducir "a mano" cada uno de los productos en cada página "nueva". Asimismo, cada vez que se realizáramos un cambio, nos tendríamos que acordar de descolgar la última página y colgar la actual. Todo esto podría ser fácilmente resuelto mediante páginas dinámicas. En este caso, lo que haríamos sería crear un programa (solo uno) que se encargaría de recoger de la base de datos todos los productos que siguen en producción o que se tendrían en stock y por otro lado se tendría un administrador que ingrese, actualice o elimine los productos de la importadora según sea el caso.

Este hecho lo podríamos aplicar a otras situaciones: como en las promociones de un nuestro sitio, en las solicitudes de productos, etc.

Podemos hacer una clasificación a las páginas dinámicas en función de dónde se lleva a cabo el procesamiento de la página, es decir, el computador que cargará con el peso adicional que supone que la página realiza efectos y funcionalidades.

2.2.5.2. Páginas Dinámicas Cliente

Conocemos el primer tipo de páginas dinámicas: Las del lado del cliente, es decir, las que se ejecutan en el navegador del usuario.

Son las páginas dinámicas que se procesan en el cliente. En estas páginas toda la carga de procesamiento de los efectos y funcionalidades la soporta el navegador.

Usos típicos de las páginas de cliente son efectos especiales para webs como rollovers o control de ventanas, presentaciones en las que se pueden mover objetos por la página, control de formularios, cálculos, etc.

El código necesario para crear los efectos y funcionalidades se incluye dentro del mismo archivo HTML y es llamado SCRIPT. Cuando una página HTML contiene scripts de cliente, el navegador se encarga de interpretarlos y ejecutarlos para realizar los efectos y funcionalidades.

Las páginas dinámicas de cliente se escriben en dos lenguajes de programación principalmente: Javascript y Visual Basic Script (VBScript).

Las páginas del cliente son muy dependientes del sistema donde se están ejecutando y esa es su principal desventaja, ya que cada navegador tiene sus propias características, incluso cada versión, y lo que puede funcionar en un navegador puede no funcionar en otro.

Como ventaja se puede decir que estas páginas descargan al servidor algunos trabajos, ofrecen respuestas inmediatas a las acciones del usuario y permiten la utilización de algunos recursos de la máquina local.

En nuestro sitio tenemos éste tipo de páginas, cuando utilizamos lenguaje javascript en la validaciones de campos, como códigos, mail, fechas, teléfonos, etc. Asimismo, tenemos éstas páginas en el proceso de solicitar productos.

2.2.5.3. Páginas Dinámicas Servidor

Las páginas dinámicas del servidor son reconocidas, interpretadas y ejecutadas por el propio servidor.

Las páginas del servidor son útiles en muchas ocasiones. Con ellas se puede hacer todo tipo de aplicaciones web. Desde agendas a foros, sistemas de documentación, estadísticas, juegos, chats, etc. Son especialmente útiles en trabajos que se tiene que acceder a información centralizada, situada en una base de datos en el servidor, y cuando por razones de seguridad los cálculos no se pueden realizar en el ordenador del usuario.

Es importante destacar que las páginas dinámicas de servidor son necesarias porque para hacer la mayoría de las aplicaciones web se debe tener acceso a muchos recursos externos al ordenador del cliente, principalmente bases de datos alojadas en servidores de Internet. Un caso claro es un banco: no tiene ningún sentido que el cliente tenga acceso a toda la base de datos, sólo a la información que le concierne.

Las páginas dinámicas del servidor se suelen escribir en el mismo archivo HTML, mezclado con el código HTML, al igual que ocurría en las páginas del cliente. Cuando una página es solicitada por parte de un cliente, el servidor ejecuta los scripts y se genera una página resultado, que solamente contiene código HTML. Este resultado final es el que se envía al cliente y puede ser interpretado sin lugar a errores ni incompatibilidades, puesto que sólo contiene HTML

Luego es el servidor el que maneja toda la información de las bases de datos y cualquier otro recurso, como imágenes o servidores de correo y luego envía al cliente una página web con los resultados de todas las operaciones.

Para escribir páginas dinámicas de servidor existen varios lenguajes, que veremos con detenimiento más adelante. Common Gateway Interface (CGI) comúnmente escritos en Perl, Active Server Pages (ASP), Hipertext Preprocesor (PHP), y Java Server Pages (JSP).

Las ventajas de este tipo de programación son que el cliente no puede ver los scripts, ya que se ejecutan y transforman en HTML antes de enviarlos. Además

son independientes del navegador del usuario, ya que el código que reciben es HTML fácilmente interpretable.

Como desventajas se puede señalar que será necesario un servidor más potente y con más capacidades que el necesario para las páginas de cliente. Además, estos servidores podrán soportar menos usuarios concurrentes, porque se requerirá más tiempo de procesamiento para cada uno. (*Desarrolloweb, 2005*)

Dentro de nuestro Sitio, éste tipo de páginas son utilizadas para realizar la conexión con la Base de Datos, dentro de los procesos de Administración del Sitio Web, como por ejemplo en las administraciones de: productos, proveedores, marcas, usuarios, promociones, marcas, etc.

2.3. CONCEPTOS RELACIONADOS

En este punto revisaremos las características principales de los conceptos más utilizados durante el desarrollo del sitio web de la Importadora Wang

2.3.1. DIRECCIÓN IP

Dirección IP es la dirección que tiene un ordenador conectado a una red, ya sea Internet, red local, etc. Todo ordenador que se comunique con otros necesita tener una dirección para poder mandar y recibir información correctamente. Sólo decir que en una comunicación punto a punto (es decir, entre sólo dos equipos), no sería absolutamente necesario tener una IP, ya que el tráfico de información va a seguir siempre el mismo camino.

Una IP está formada por 4 números de 8 bits cada uno, separados por puntos. Es decir, una IP está formada por 4 octetos; por lo tanto, toda dirección IP tendrá 32 bits ($8 \times 4 = 32$). Cada número de la dirección IP va desde 0 a 255.

Octeto 1	Octeto 2	Octeto 3	Octeto 4
xxx.	xxx.	xxx.	xxx

Una IP tiene esta forma: 255.255.255.255

Una dirección IP sirve para identificar de manera única a una entidad (ordenador, router, etc.) en una red.

Una Dirección IP puede ser de tipo A, B, o C

	Octeto 1	Octeto 2	Octeto 3	Octeto 4	Máximo Equipos
Tipo A	Red	Equipos	Equipos	Equipos	$(2 \wedge 8) - 1$
Tipo B	Red	Red	Equipos	Equipos	$(2 \wedge 16) - 1$
Tipo C	Red	Red	Red	Equipos	$(2 \wedge 24) - 1$

2.3.1.1. MASCARA DE RED

Una mascara de red es una dirección de 32 bits usada para ‘enmascarar’ una parte de la dirección IP para distinguir el ID de red del ID de *host*. Esto es necesario para que el TCP/IP pueda determinar cuando una dirección IP pertenece a la red local o a una red remota.

Cada maquina en una red TCP/IP requiere una mascara de red, bien una mascara de red por defecto usada cuando una red no está dividida en subredes, o una mascara ‘personalizada’ cuando la red está dividida en segmentos.

2.3.1.2. MASCARA DE RED POR DEFECTO

Una mascara de red por defecto se usa en las redes TCP/IP cuando estas no están divididas en subredes. Todos los *hosts* TCP/IP requieren esta mascara aunque estén en un solo segmento de red. La mascara por defecto que podemos utilizar, depende de la ‘clase’ de dirección.

En la mascara de red, todos los bits que corresponden a un ID de red están colocados a 1. El valor decimal de un octeto con todos unos, es 255. Todos los bits que corresponden al ID *host* estarán colocados a cero.

Clase	Bits usados por la mascara de red	Valor decimal
Clase A	11111111 00000000 00000000 00000000	255.0.0.0
Clase B	11111111 11111111 00000000 00000000	255.255.0.0
Clase C	11111111 11111111 11111111 00000000	255.255.255.0

2.3.2. INTERNET

El Internet, algunas veces llamado simplemente "La Red", es un sistema mundial de redes de computadoras, un conjunto integrado por las diferentes redes de cada país del mundo, por medio del cual un usuario en cualquier computadora puede, en caso de contar con los permisos apropiados, acceder a información de otra computadora y poder tener inclusive comunicación directa con otros usuarios en otras computadoras.

Fue concebido por la agencia de nombre ARPA (Advanced Research Projects Agency) del gobierno de los Estados Unidos en el año de 1969 y se le conocía inicialmente como ARPANET. El propósito original fue crear una red que permitiera a los investigadores en un Campus poder comunicarse a través de los sistemas de cómputo con investigadores en otras Universidades.

Hoy en día, el Internet es un medio de comunicación público, cooperativo y autosuficiente en términos económicos, accesible a cientos de millones de personas en el mundo entero. Físicamente, el Internet usa parte del total de recursos actualmente existentes en las redes de telecomunicaciones.

Técnicamente, lo que distingue al Internet es el uso del protocolo de comunicación llamado TCP/IP (Transmission Control Protocol/Internet Protocol).

Para muchos usuarios del Internet, el correo electrónico (e-mail) ha reemplazado prácticamente al servicio postal para breves mensajes por escrito. El correo electrónico es la aplicación de mayor uso en la red. También se pueden realizar conversaciones "en vivo" con otros usuarios en otras localidades usando el IRC (Internet Relay Chat). Más recientemente, el software y hardware para telefonía en Internet permite conversaciones con audio y video en vivo. (*Informaticamilenium, 2005*).

2.3.2.1. Internet y Sociedad

Hoy en día, Internet ha llegado a gran parte de los hogares y de las empresas de los países ricos, en este aspecto se ha abierto una brecha digital con los países pobres, en los cuales la penetración de Internet y las nuevas tecnologías es muy limitada para las personas.

Es muy común en países ricos y en vías de desarrollo el acceso a Internet en establecimientos llamados cybercafés. (*Informaticamilenium, 2005*).

2.3.3. WWW

El World Wide Web fue creado en 1989-1990 por un grupo de científicos del CERN (Consejo Europeo para la Investigación Nuclear) en Geneva, con un objetivo muy concreto: compartir información de manera flexible, sencilla y distribuida.

El WWW es, además, una ventana a la información mundial, muy fácil de utilizar. Es esta una de las causas fundamentales de la explosión en el uso de esta herramienta por parte de los usuarios de la Internet.

2.3.3.1. Características

- Es multimedial porque maneja textos, imágenes estáticas, imágenes dinámicas y sonido.
- Es distribuido porque no hay un gran punto central donde la información está concentrada, sino que cubre todo el Mundo con un gran número de servidores, cada uno conteniendo algo de información organizada como un gran hipertexto.
- Es heterogéneo porque incluye varios servicios de Internet que lo precedieron como por ejemplo Gopher, FTP, Newsgroups, WAIS, etc. Es colaborativo porque permite que cualquiera pueda agregar nueva información y la misma esté disponible rápidamente para todos los demás.

(*Enciclopedia Wikipedia, 2005*).

2.3.4. SITIO WEB

Es un conjunto de archivos electrónicos y páginas Web referentes a un tema en particular, que incluye una página inicial de bienvenida, generalmente denominada home page, con un nombre de dominio y dirección en Internet.

Los Sitios Web son empleados por las instituciones públicas y privadas, organizaciones e individuos para comunicarse con el mundo entero. En el caso particular de las empresas, este mensaje tiene que ver con la oferta de sus bienes y servicios a través de Internet, y en general para efectivizar sus funciones de mercadotecnia.

Un Sitio Web no necesariamente debe localizarse en el sistema de cómputo de un negocio. Los documentos que integran el Sitio Web pueden ubicarse en un equipo en otra localidad, inclusive en otro país. El único requisito es que el equipo en el que residan los documentos esté conectado a la red mundial de Internet. Este equipo de cómputo o Servidor Web, como se le denomina técnicamente, puede contener más de un sitio Web y atender concurrentemente a los visitantes de cada uno de los diferentes sitios .

Al igual que los edificios, oficinas y casas, los Sitios Web requieren de una dirección particular para que los usuarios puedan acceder a la información contenida en ellos. Estas direcciones, o URLs (por sus siglas en inglés Uniform Resource Locator), aparecen cotidianamente en todos los medios de comunicación como son prensa escrita, radio, televisión, revistas, publicaciones técnicas y en el propio Internet a través de los motores de búsqueda (por su denominación en inglés search engines). Los nombres de estos sitios Web obedecen a un sistema mundial de nomenclatura y están regidos por el ICANN (Internet Corporation for Assigned Names and Numbers).

Los Sitios Web pueden ser de diversos géneros, destacando los sitios de negocios, servicio, comercio electrónico en línea, imagen corporativa, entretenimiento y sitios informativos. (*Informaticamilenium, 2005*).

2.3.5. PORTAL WEB

Portal es un término, sinónimo de puente, para referirse a un Sitio Web que sirve o pretende servir como un sitio principal de partida para las personas que se conectan al World Wide Web. Son sitios que los usuarios tienden a visitar como sitios ancla. Los portales tienen gran reconocimiento en Internet por el poder de influencia que tienen sobre grandes comunidades.

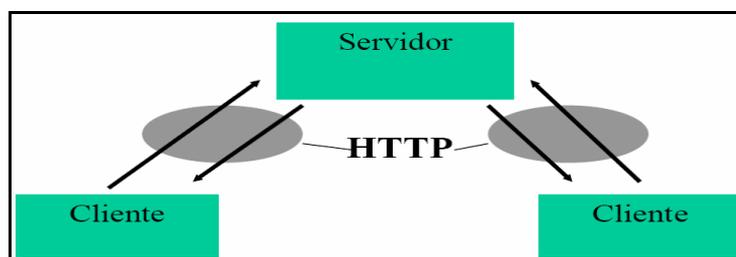
La idea es emplear estos portales para localizar la información y los sitios que nos interesan y de ahí comenzar nuestra actividad en Internet. Un Sitio Web no alcanza el rango de portal sólo por tratarse de un sitio robusto o por contener información relevante. Un portal es más bien una plataforma de despegue para la navegación en el Web. (*Informaticamilenium, 2005*).

2.3.6. HTTP

El protocolo estándar de transferencia de la Web es el HTTP (HyperText Transfer Protocol, el protocolo de transferencia de hipertexto).

El HTTP es el protocolo utilizado por los clientes y servidores WWW y está diseñado para lograr una rápida y eficiente entrega de documentos y objetos hipermediales.

GRAFICO 2.1



Fuente: <http://www.monografias.com/html>

2.3.6.1. Características

- Es un protocolo sin estados (stateless).

- Tiene la peculiaridad de realizar una única transacción por conexión (connectionless).
- Todas las comunicaciones HTTP emplean caracteres de 8 bits lo cual permite una transmisión segura de todos los datos.
- Permite manejar metainformación.

Metainformación es información acerca de la información. Esto permite que el visualizador decida la mejor manera de aprovechar la misma. Por ejemplo, podría enviarse un objeto y especificar el idioma de manera que se pueda elegir si se desea hacer el download o no, la fecha de expiración, etc.

2.3.6.2. Etapas en una Conexión HTTP

Una conexión HTTP tiene cuatro etapas:

1- *Apertura de la conexión*: el cliente contacta al servidor en la dirección Internet y puerto especificado en la URL.

2- *La solicitud*: el cliente envía un mensaje al servidor solicitándole un objeto o documento hipermedial en particular. Este mensaje está formado por cabecera de solicitud HTTP que definen el método y suministra información acerca de las posibilidades del cliente.

3- *La respuesta*: el servidor envía una respuesta al cliente. Esta respuesta está formada por cabecera de respuesta HTTP describiendo el estado de la transacción, seguida por los datos.

4- *Cierre de la conexión*

El procedimiento formado por las cuatro etapas de una conexión HTTP nos indica que dicha conexión puede transferir solamente un único objeto o documento, o procesar una única transacción (connectionless), mientras que la naturaleza sin estados del HTTP significa que cada conexión no conoce nada acerca de las que la precedieron. (stateless). (*Monografías.com, 2005*).

2.3.7. JAVASCRIPT

Javascript es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro de una página web.

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

Con Javascript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Javascript es el siguiente paso, después del HTML, que puede dar un programador de la que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica.

Entre las acciones típicas que se pueden realizar en Javascript tenemos dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, javascript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

Javascript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Javascript pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente. (*Desarrolloweb, 2005*).

Dentro de nuestro Sitio Web se ha utilizado lenguaje Javascript en la validación de campos, para enviar mensajes al usuario, para hacer vínculos, etc.

2.3.8. HTML

El HTML, Hyper Text Markup Language (Lenguaje de marcación de Hipertexto) es el lenguaje de marcas de texto utilizado normalmente en la www (World Wide Web). Fue creado en 1986 por el físico nuclear Tim Berners-Lee, el cual tomo dos herramientas preexistentes: El concepto de Hipertexto (Conocido también como link o ancla) el cual permite conectar dos elementos entre si y el SGML (Lenguaje Estándar de Marcación General) el cual sirve para colocar etiquetas o marcas en un texto que indique como debe verse. HTML no es propiamente un lenguaje de programación como C++, Visual Basic, etc., sino un sistema de etiquetas. HTML no presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizara en la forma como éste lo entienda.

El entorno para trabajar HTML es simplemente un procesador de texto, como el que ofrecen los sistemas operativos de Windows (Bloc de notas), UNIX (el editor vi o ed) o el que ofrece MS Office Word. El conjunto de etiquetas que se creen, se deben guardar con la extensión .htm o .html

Estos documentos pueden ser mostrados por los visores o "browsers" de paginas Web en Internet, como Netscape Navigator, Mosaic, Opera y Microsoft Internet Explorer. También existe el HTML Dinámico (DHTML), que es una mejora de Microsoft de la versión 4.0 de HTML que le permite crear efectos especiales como, por ejemplo, texto que vuela desde la página palabra por palabra o efectos de transición al estilo de anuncio publicitario giratorio entre páginas y página. (*Monografías.com, 2005*)

2.3.9. PHP

PHP (acrónimo recursivo de "PHP: Hypertext Preprocessor", originado inicialmente del nombre PHP Tools, o Personal Home Page Tools) es un Lenguaje de programación interpretado, con licencia open-source.

El fácil uso y la similaridad con los más comunes lenguajes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores experimentados crear aplicaciones complejas con una curva aprendizaje muy suave. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones y prácticas.

Debido al diseño de PHP, también es posible crear aplicaciones con una interfaz gráfica para el usuario o GUI, utilizando la PHP-GTK. También puede ser usado desde la Línea de comandos, como Perl o Python.

Su interpretación y ejecución se da en el servidor en el cual se encuentra almacenada la página y el cliente solo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página web, enriquecida con código PHP, el servidor interpretará las instrucciones mezcladas en el cuerpo de la página y las sustituirá con el resultado de la ejecución antes de enviar el resultado a la computadora del cliente. Además es posible utilizarlo para generar archivos PDF, Flash o JPG, entre otros.

Permite la conexión a numerosas bases de datos de forma nativa tales como MySQL, Postgres, Oracle, ODBC, IBM, Microsoft SQL Server y SQLite, lo cual permite la creación de Aplicaciones web muy robustas.

PHP tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX, Linux, Windows y Mac OS X, y puede interactuar con los servidores de web más populares.

El modelo PHP puede ser visto como una alternativa al sistema de Microsoft que utiliza ASP.NET/C#/VB.NET, a Coldfusion de la compañía Macromedia, a JSP/Java de Sun Microsystems, y al famoso CGI/Perl. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un compilador comercial denominado Zend Optimizer. (Silva, 2005).

2.3.10. DOMINIO

Técnicamente, un nombre de dominio es una dirección usada para identificar y localizar una computadora en la Internet. Mientras que las computadoras usan los números de Internet Protocolo (IP) para localizarse, las personas se encuentran en dificultades a la hora de memorizarlos, por lo cual se crearon los nombres de dominio para identificar aquellas direcciones.

Un ejemplo de nombre de dominio es google.com o epn.edu.ec.

Cuando alguien a través de un browser intenta visitar <http://www.yahoo.com/> el Sistema de Nombres de Dominio, en adelante Domain Name System (DNS) lo traduce al correspondiente número de IP y lo lleva hasta allí.

Existen dos divisiones en orden de importancia: los top level domains (TLDs), o dominios de primer nivel, incluyen a los famosos “.com”, “.net” y “.org”. Y los country code domain names (ccTLDs) incluyen al final del dominio la designación del país al que pertenecen, como por ejemplo “.ar” para el caso argentino, “.br.” para el caso brasilero, etc.

En un principio los únicos caracteres válidos eran las 26 letras del alfabeto en inglés, los 10 dígitos, los guiones (aunque estos no se podían usar para empezar o terminar la dirección), y el punto para dividir una dirección en secciones (ej.: epn.edu.ec). Y decíamos en un principio porque debido a la presión para hacer de Internet un entorno multilinguaje, es que se admitieron otro tipo de caracteres. Así en noviembre de 2000 se admitieron los caracteres en Chino, Japonés y Coreano, que han sido hasta ahora los más solicitados para su registro; en febrero de 2001 se admitieron caracteres en unas 40 lenguas europeas (por fin podíamos incluir la ñ y las diéresis), y en abril del mismo año se admitieron en Thai, Hebreo y Árabe.

Las categorías admitidas más frecuentemente, y sus correspondientes subdominios son:

.com	Organizaciones comerciales
.edu	Entidades educativas

.gov	Gobierno
.net	Proveedores de servicio de Internet
.org	Organizaciones sin fines de lucro
.mil	Militares
.int	Organizaciones internacionales
.aero	Industria aeronáutica
.biz	Negocios
.coop	Cooperativas
.info	Uso irrestricto
.museum	Museos
.name	Para personas individuales
.pro	Profesionales (abogados, médicos, etc)

A nivel mundial, la entidad encargada de la administración del sistema es la Corporación de Asignación de Nombres y Números de Internet (ICANN –Internet Corporation for Assigned Names and Numbers).

La ICANN es la nueva organización sin fines de lucro encargada del Sistema de Nombres de Dominio (DNS), de la ubicación de las direcciones IP, del gerenciamiento del servidor raíz, y de la coordinación de la asignación de los números de protocolo. (*Zamudio, 2005*).

2.3.11. WEB HOSTING

Una vez que se ha realizado una página Web, se necesita enviarla a un servidor que esté permanentemente conectado a Internet para que cualquier internauta pueda verla.

Lo más normal es que uses un servicio de alojamiento que te proporcione una empresa, una empresa que tenga toda la infraestructura necesaria para tener los servidores permanente conectados a Internet.

Este tipo de servicios se suelen llamar: web hosting, hosting, alojamiento o alojamiento web.

Básicamente podemos clasificar el hosting en dos grandes grupos: los gratuitos y los profesionales.

Lo recomendable es que se use un alojamiento profesional ya que actualmente el precio de estos servicios es muy competitivo y nos ofrecen alojar nuestras páginas por muy poco precio. Por otro lado, los gratuitos finalmente no lo son ya

que nos obligan a poner publicidad que normalmente suele ser muy agresiva y desagradable, dando una mala imagen a nuestra página web. (*J. Gracia, 2005*)

2.3.11.1 WINDOWS HOSTING Y LINUX HOSTING

Dependiendo del tipo de programación y los servicios que usemos o tengamos pensado usar nos decidiremos por un tipo de alojamiento u otro. Tenemos dos grandes clasificaciones: Windows Hosting y Linux Hosting.

2.3.11.2. Windows Hosting

El término Windows Hosting se aplica a los alojamientos web (web hosting) cuyo servidor corre bajo sistema operativo Windows, normalmente también está asociado al servidor web IIS que es el servidor web de Microsoft.

Las tecnologías en las que Windows Hosting es superior a Linux Hosting ya bien porque no pueden correr bajo Linux o porque aunque corriendo tienen mejor rendimiento y dan menos problemas en Windows son: Windows, ASP, ASP.NET, ADO, ADO.NET, VB.NET, C#, Visual Studio .NET, SQL Server o IIS.

Un asunto muy a tener en cuenta a la hora de contratar un servicio de Windows Hosting es el soporte de componentes de terceros que tengan instalados.

Debemos buscar este tipo de alojamiento cuando usemos o queramos usar alguna de estas tecnologías: Windows, ASP, ASP.NET, SQL Server o IIS. (*J. Gracia, 2005*)

2.3.11.2. Linux Hosting

Cuando nos referimos a Linux Hosting estamos hablando de un hosting cuyo servidor corre bajo sistema operativo Linux.

Los hosting Linux suelen ser más baratos que los Windows hosting, la razón esta clara, no hay que pagar las licencias del servidor, puesto que las licencias de Linux son gratuitas. Otra de las razones es que hay mejores herramientas para la

administración de servidores Linux que abaratan el coste de administración de un servidor Linux.

A la hora de contratar un Linux hosting tenemos que tener las siguientes cosas en cuenta:

- **Panel de control.** Tenemos un panel de control con acceso web, desde el que podemos hacer la mayoría de las operaciones de administración y mantenimiento del dominio. Esto es importante porque así tenemos bastante independencia del administrador de la máquina y nos permitirá realizar nuestras operaciones mucho más rápido.
- **Soporte 24/7.** Esto quiere decir 24 horas al día, 7 días a la semana, veamos que siempre haya por lo menos una persona al cargo de las máquinas. Si por cualquier motivo hay que realizar algo en las máquinas, una persona debe estar allí para que pueda resolver el problema, no teniendo que esperar al día siguiente para que se solucione. Muchas compañías dicen que tienen este tipo de soporte y luego no es así.
- **Soporte para tu lenguaje de programación favorito.** Asegúrese que el servicio de alojamiento soporta el lenguaje de programación que desea y además, sería muy conveniente que soportase también PHP y Perl. Aunque no le interesen estos lenguajes, se va a encontrar con paquetes externos que posteriormente talvez le gustaría integrar, y la mayoría de ellos están escritos en estos lenguajes.
- **Soporte para base de datos.** Soporte por lo menos de MySQL, adquirir un servicio de alojamiento web sin soporte para MySQL sería todo un error.
- **Acceso mediante FTP.** El acceso mediante FTP es primordial para subir tus páginas al servidor, indispensable.
- **Acceso Telnet o SSH.** Tener acceso a la línea de comando siempre es recomendable, aunque no es imprescindible, si va a hacer algo serio le va a hacer falta.

(J. Gracia, 2005)

CAPITULO 3

3.1. TABLA DE RESUMEN DE LA METODOLOGÍA

En la siguiente tabla se muestra un resumen del paradigma, metodología y métodos utilizados en el desarrollo del Sitio Web de la Importadora Wang.

PARADIGMA ESPIRAL ORIENTADO A LA WEB	METODOLOGÍA OOHDM		METODOS UML
Análisis de Requisitos			Identificación de Actores Diagrama de Casos de Uso
Análisis	Diseño Conceptual	Modelo Estático	Diagrama de Clases Diccionario de Clases Diagrama de Objetos
Diseño	Diseño de Interfaz Abstracta	Modelo Dinámico	Diagrama de Secuencia Diagrama de Colaboración Diagrama de Actividades
	Diseño Navegacional	Modelo Navegacional	Diagrama Arquitectónico Diagrama Navegacional
Construcción	Herramientas de Análisis Rational Rose Enterprise Edition Herramientas de Diseño Software Macromedia Mx 2004 MySQL 3.23.48 PHPMyAdmin 2.2.3 PHP 4.1.1		
Pruebas	Pruebas Funcionales		
Mantenimiento			
Implementación			

3.2. PARADIGMA ESPIRAL ORIENTADO A LA WEB

Existen diferentes modelos de paradigmas, pero para el desarrollo de nuestro proyecto se utilizará el Paradigma Espiral Orientado a la Web, el mismo que muestra las siguientes características:

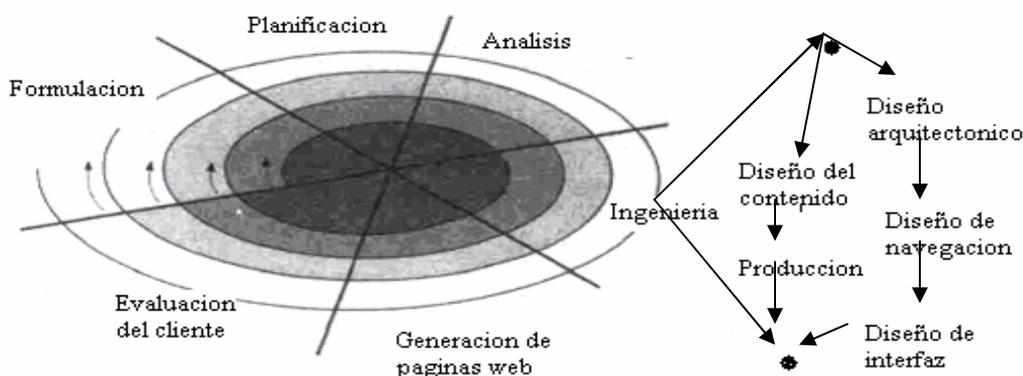
El modelo en espiral trata de desarrollar incrementalmente el proyecto, dividiéndolo en muchos subproyectos. Uno de los puntos mas importantes del proceso es concentrarse primero en los aspectos mas críticos del proyecto. La idea es definir e implementar las características mas importantes primero, y con el conocimiento adquirido para hacerlo, volver hacia atrás y reimplementar las características siguientes en pequeños subproyectos.

El modelo en espiral orientado a la web se divide en un número de actividades estructurales, también llamadas regiones de tareas. Generalmente, existen entre tres y seis regiones de tareas. (*I. Jacobson, 2000*).

- **Comunicación con el cliente:** las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.
- **Planificación:** La planificación estima el coste global del proyecto, evalúa los riesgos asociados con el esfuerzo del desarrollo, y define una planificación del desarrollo bien granulada para el incremento final de la WebApp, con una aplicación mas toscamente granulada para los incrementos subsiguientes.. Son todos los requerimientos.
- **Análisis de riesgos:** establece los requerimientos técnicos para la WebApp e identifica los elementos del contenido que se van a incorporar. También se define los requisitos de diseño gráfico(estética). Es decir se identifica los datos y requisitos funcionales y de comportamiento para la aplicación web.
- **Ingeniería:** Aquí se incorpora dos tareas paralelas, el Diseño del Contenido y Producción, estas dos tareas son llevadas a cabo por dos personas no técnicas del equipo. El objetivo de estas tareas es diseñar, producir y adquirir todo el contenido de texto, gráfico y video que se vayan a integrar en la WebApp.

- **Generación de páginas:** La generación de páginas es una actividad de construcción que hace mucho uso de las herramientas automatizadas para la creación de la aplicación web. El contenido definido en la actividad de ingeniería se fusiona con los diseños arquitectónicos, de navegación y de la interfaz para la elaboración de páginas Web ejecutables en HTML, XML y otros lenguajes orientados a procesos (java). Durante esta actividad también se lleva a cabo la integración con el software intermedio de componentes. Las pruebas ejercitan la navegación, intentan descubrir los errores de los applets, guiones y formularios, y ayuda a asegurar que la aplicación web funcionará correctamente en diferentes entornos
- **Evaluación el cliente:** en este punto es donde se solicitan cambios, tiene lugar la ampliación del ámbito.

GRÁFICO 3.1



Fuente: <http://www.clikear.com/metodo/espinal>

3.3. METODOLOGÍA

La Metodología es el establecimiento de teorías sobre el Método. Entonces la metodología es la descripción y el análisis de los métodos. Podríamos afirmar que la metodología es el estudio analítico y crítico de los métodos de investigación y de prueba, esto incluye:

La descripción, el análisis y la valoración crítica de los métodos que conciernen a la investigación.

Lo trascendente de la metodología es que le interesa más el proceso de investigación que los mismos resultados.

En la era espacial en la cual nos desarrollamos, el bombardeo permanente de información demanda de cada uno de los sujetos la búsqueda de diferentes procedimientos y mecanismos de acción, es decir, las herramientas más idóneas para aplicar esa información y transformarla en conocimiento. La información dista mucho del significado de conocimiento, a la primera se la puede conceptuar como los datos que obtenemos por diversos medios y la segunda como la puesta en práctica de aquellos datos que han sido obtenidos.

Al hablar de metodología hacemos referencia a los diversos mecanismos de rigor lógico-científico, que ayudan a desarrollar el conocimiento, dentro de cada una de las disciplinas científicas. La metodología une y procesa todos los componentes de las áreas del saber, de igual forma, construye sistemas que permiten llegar a los objetivos, metas, desafíos y por ende los consabidos resultados. (C. Villalba, 2004)

3.4. METODOLOGÍA OOHDM

La metodología aplicada para el desarrollo de nuestro proyecto es la OOHDM (Object – Oriented Hypermedia Design Method, Método de Diseño Hipermedia Orientado a Objeto). La misma fue seleccionada ya que satisface los requerimientos de la aplicación, especialmente en lo referente al desarrollo Web.

OOHDM es una propuesta metodológica ampliamente aceptada para el desarrollo de aplicaciones de la web. En sus comienzos no contemplaba la fase de captura y definición de requisitos, pero actualmente propone el uso de User Interaction Diagrams (UIDs) definidos por *Vilain, Schwabe & Sieckenius (2000)*. Esta propuesta parte de los casos de uso, que considera una técnica muy difundida, ampliamente aceptada y fácilmente entendible por los usuarios y clientes no expertos, pero que resulta ambigua para el equipo de desarrollo en fases posteriores del ciclo de vida. Igualmente, resalta la necesidad de empezar el diseño del sistema, especialmente en los entornos web, teniendo un claro y

amplio conocimiento de las necesidades de interacción, o lo que es lo mismo de la forma en la que el usuario va a comunicarse con el sistema. Partiendo de estas dos premisas, OOHDM propone que la comunicación con el usuario se realice utilizando los casos de uso y a partir de ellos los analistas elaboran los UIs. Estos UIs son modelos gráficos que representan la interacción entre el usuario y el sistema, sin considerar aspectos específicos de la interfaz. El proceso de transformación de un caso de uso a un UIs es descrito detalladamente en la propuesta.

OOHDM centra el desarrollo de un sistema de información web entorno al modelo conceptual de clases. Este diagrama debe surgir de los requisitos que se definan del sistema, pero los casos de uso resultan demasiado ambiguos para ello. Así, propone refinar el proceso de desarrollo descrito en UML, de forma que de los casos de uso se generen los UIs que concreten más la definición de los requisitos para, a partir de ellos, obtener el diagrama conceptual. OOHDM propone la descripción de escenarios en forma textual y gráfica para cada tipo de usuario, como etapa previa al diseño de la navegación. (*D. Silva, 2005*).

3.5. FASES DE LA METODOLOGÍA OOHDM

OOHDM propone el desarrollo de aplicaciones hipertexto a través de un proceso compuesto las siguientes etapas: captura de requerimientos, diseño conceptual, diseño navegacional, diseño de interfaces abstractas, implementación y pruebas.

3.5.1. CAPTURA DE REQUERIMIENTOS

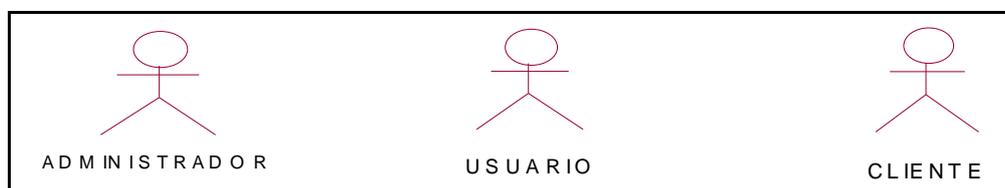
En el desarrollo de software de calidad, es esencial la captura correcta de requerimientos por parte de los diseñadores del sistema a desarrollar. Según *Schwabe, D. & G. Rossi (2002)*, consideran necesario incorporar esta fase al OOHDM. Aquí, es primordial identificar a los actores (stakeholders) y las tareas que ellos van a ejecutar. Luego, los escenarios son recolectados para cada tarea y tipo de actor, produciendo los Casos de Uso, los cuales son una representación

gráfica concisa de la interacción entre el usuario y el sistema durante la ejecución de una tarea.

3.5.1.1. Descripción de los Usuarios del Sistema

Para una mejor comprensión del rol que tienen los usuarios del Sistema se realizó una breve descripción de cada uno de ellos colocando su descripción, responsabilidades y uso del sistema.

GRÁFICO 3.2.



ADMINISTRADOR.- Es la persona que se encarga de administrar todo lo referente al Sitio Web, actualiza la Información de la Importadora, productos, proveedores, marcas, descargas, etc, además administra lo que son los comentarios y sugerencias. También es el encargado de dar soluciones a las solicitudes expuestas en el módulo de Servicio Técnico.

USUARIO.- Es la Persona que visita el Sitio Web, puede revisar la información de la Importadora y de los productos se ofrece en la Importadora.

CLIENTE.- Es un Usuario que se ha registrado en el Sistema puede revisar la Información de la Importadora y sus productos. Además tiene acceso a realizar pedidos, sacar proformas, realizar descargas y solicitar servicio técnico.

3.5.2. DISEÑO CONCEPTUAL

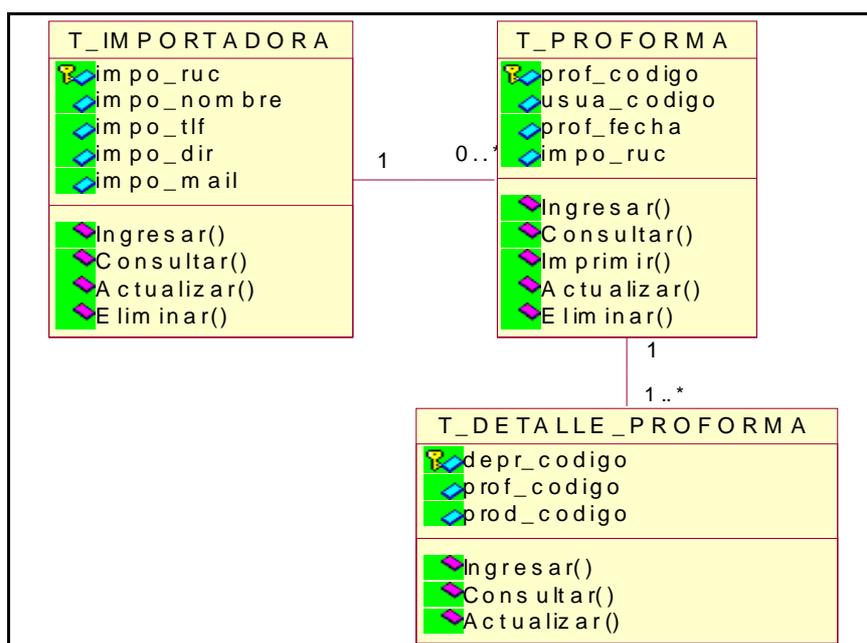
Durante esta actividad se construye un esquema conceptual representado por los objetos del dominio, las relaciones y colaboraciones existentes establecidas entre ellos. En las aplicaciones hipermedia convencionales, cuyos componentes de

hipermedia no son modificados durante la ejecución, se podría usar un modelo de datos semántico estructural (como el modelo de entidades y relaciones). De este modo, en los casos en que la información base pueda cambiar dinámicamente o se intenten ejecutar cálculos complejos, se necesitará enriquecer el comportamiento del modelo de objetos.

En OOHDM, el esquema conceptual está construido por clases, relaciones y subsistemas. Las clases son descritas como en los modelos orientados a objetos tradicionales. Sin embargo, los atributos pueden ser de múltiples tipos para representar perspectivas diferentes de las mismas entidades del mundo real.

Se usa notación similar a UML (Lenguaje de Modelado Unificado). El esquema de las clases consiste en un conjunto de clases conectadas por relaciones

GRÁFICO 3.3.



3.5.3. DISEÑO NAVEGACIONAL

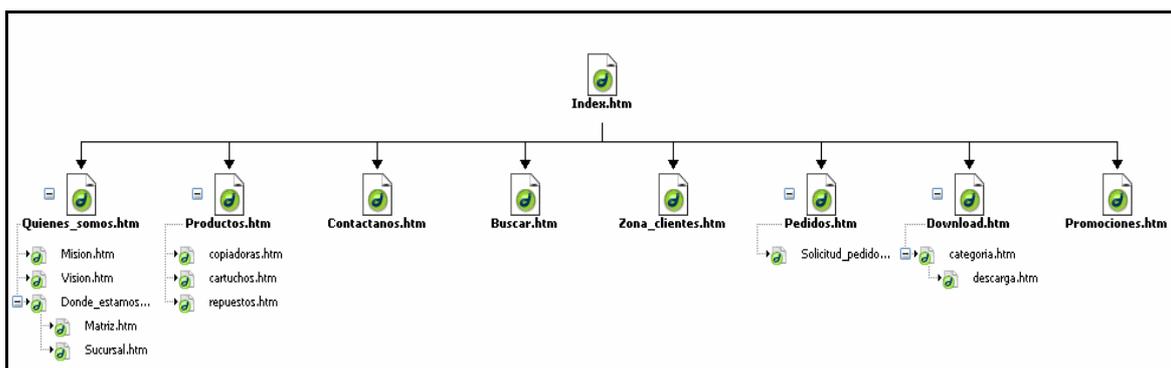
La primera generación de aplicaciones web fue pensada para realizar navegación a través del espacio de información, utilizando un simple modelo de datos de hipermedia. En OOHDM, la navegación es considerada un paso crítico en el diseño aplicaciones. Un modelo navegacional es construido como una vista sobre

un diseño conceptual, admitiendo la construcción de modelos diferentes de acuerdo con los diferentes perfiles de usuarios. Cada modelo navegacional provee una vista subjetiva del diseño conceptual.

El diseño de navegación es expresado en dos esquemas: el esquema de clases navegacionales y el esquema de contextos navegacionales. En OOHDMM existe un conjunto de tipos predefinidos de clases navegacionales:

nodos, enlaces y estructuras de acceso. La semántica de los nodos y los enlaces son las tradicionales de las aplicaciones hipertexto, y las estructuras de acceso, tales como índices o recorridos guiados, representan los posibles caminos de acceso a los nodos.

GRÁFICO 3.4.



La principal estructura primitiva del espacio navegacional es la noción de contexto navegacional. Un contexto navegacional es un conjunto de nodos, enlaces, clases de contextos, y otros contextos navegacionales (contextos anidados). Pueden ser definidos por comprensión o extensión, o por enumeración de sus miembros.

Los contextos navegacionales juegan un rol similar a las colecciones y fueron inspirados sobre el concepto de contextos anidados. Organizan el espacio navegacional en conjuntos convenientes que pueden ser recorridos en un orden particular y que deberían ser definidos como caminos para ayudar al usuario a lograr la tarea deseada.

Los nodos son enriquecidos con un conjunto de clases especiales que permiten de un nodo observar y presentar atributos (incluidos las anclas), así como métodos (comportamiento) cuando se navega en un particular contexto.

3.5.4. DISEÑO ABSTRACTO DE INTERFAZ

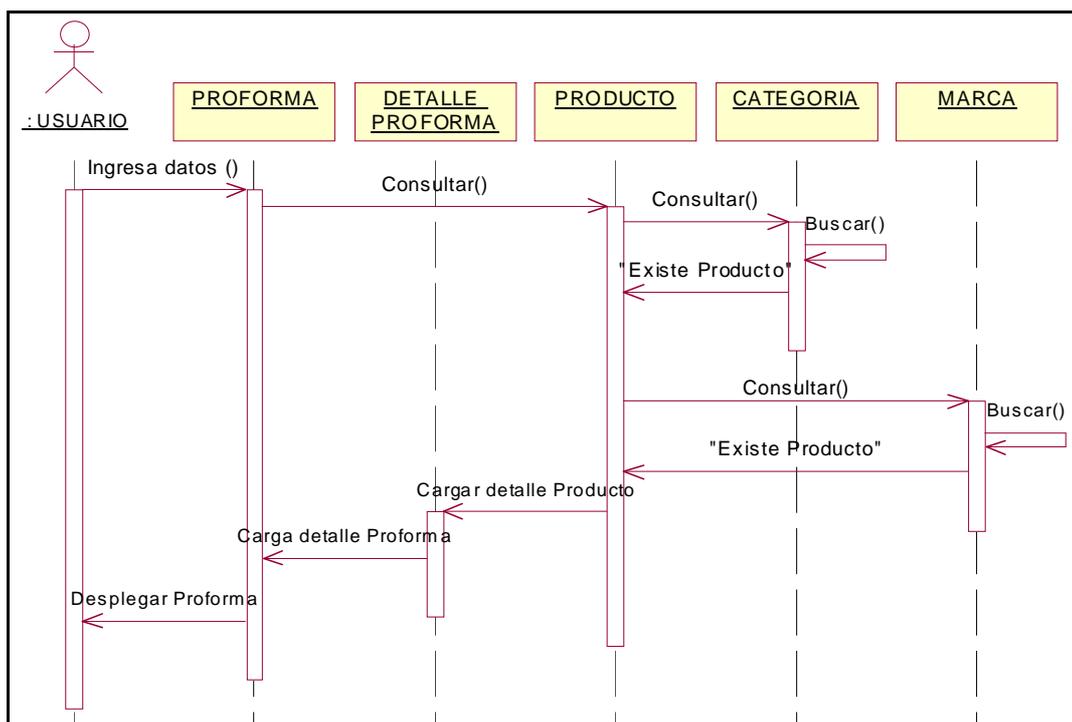
Una vez que las estructuras navegacionales son definidas, se deben especificar los aspectos de interfaz. Esto significa definir la forma en la cual los objetos navegacionales pueden aparecer, cómo los objetos de interfaz activarán la navegación y el resto de la funcionalidad de la aplicación, qué transformaciones de la interfaz son pertinentes y cuándo es necesario realizarlas.

Una clara separación entre diseño navegacional y diseño de interfaz abstracta permite construir diferentes interfaces para el mismo modelo navegacional, dejando un alto grado de independencia de la tecnología de interfaz de usuario.

El aspecto de la interfaz de usuario de aplicaciones interactivas (en particular las aplicaciones web) es un punto crítico en el desarrollo que las modernas metodologías tienden a descuidar. En OOHDM se utiliza el diseño de interfaz abstracta para describir la interfaz del usuario de la aplicación de hipermedia.

El modelo de interfaz ADVs (Vista de Datos Abstracta) especifica la organización y comportamiento de la interfaz, pero la apariencia física real o de los atributos, y la disposición de las propiedades de las ADVs en la pantalla real son hechas en la fase de implementación.

GRAFICO 3.5.



3.5.5. IMPLEMENTACION

En esta fase se implementa el software hipermedia. Los modelos construidos independientemente de la plataforma de implementación son traducidos a un ambiente de ejecución.

Nuestra aplicación está compuesta por los siguientes módulos:

- Información General
- Productos y Promociones
- Servicio Técnico
- Registro de Usuarios
- Descargas
- Proformas
- Contáctenos
- Administración del Sitio

GRAFICO 3.6.



3.5.6. PRUEBAS

En esta fase se llevan a cabo las pruebas a la aplicación antes de colocarla en producción.

En nuestro caso a medida que se codifiquen las tareas, se van a ir aplicando las pruebas unitarias a cada una de ellas, con el fin de ir constatando el buen funcionamiento de los módulos, para luego poder ensamblar el sistema y así realizar las pruebas definitivas con el cliente. *(D. Delgado, 2005)*

3.6. U.M.L.

El Lenguaje Unificado de Modelamiento (U.M.L.) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

U.M.L. capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo.

El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

U.M.L. no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de U.M.L. para una gran variedad de lenguaje de programación, así como construir modelos por ingeniería inversa a partir de programas existentes.

Es un lenguaje de propósito general para el modelado orientado a objetos. U.M.L. es también un lenguaje de modelamiento visual que permite una abstracción del sistema y sus componentes.

Existían diversos métodos y técnicas Orientadas a Objetos, con muchos aspectos en común pero utilizando distintas notaciones, se presentaban inconvenientes para el aprendizaje, aplicación, construcción y uso de herramientas, etc., además de pugnas entre enfoques, lo que genero la creación del U.M.L. como estándar para el modelamiento de sistemas de software principalmente, pero con

posibilidades de ser aplicado a todo tipo de proyectos. (*Consultoría en Seguridad, 2005*).

3.6.1. OBJETIVOS DE U.M.L.

- UML es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.
- UML no pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.
- Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. UML necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son la encapsulación y componentes.
- Debe ser un lenguaje universal.
- Imponer un estándar mundial.

3.6.2. AREAS CONCEPTUALES DE UML

Los conceptos y modelos de UML pueden agruparse en las siguientes áreas conceptuales:

3.6.2.1. Estructura Estática

Cualquier modelo preciso debe primero definir su universo, esto es, los conceptos clave de la aplicación, sus propiedades internas, y las relaciones entre cada una de ellas. Este conjunto de construcciones es la estructura estática. Los conceptos

de la aplicación son modelados como clases, cada una de las cuales describe un conjunto de objetos que almacenan información y se comunican para implementar un comportamiento. La información que almacena es modelada como atributos; La estructura estática se expresa con diagramas de clases y puede usarse para generar la mayoría de las declaraciones de estructuras de datos en un programa.

3.6.2.2. Comportamiento Dinámico

Hay dos formas de modelar el comportamiento, una es la historia de la vida de un objeto y la forma como interactúa con el resto del mundo y la otra es por los patrones de comunicación de un conjunto de objetos conectados, es decir la forma en que interactúan entre sí. La visión de un objeto aislado es una máquina de estados, muestra la forma en que el objeto responde a los eventos en función de su estado actual. La visión de la interacción de los objetos se representa con los enlaces entre objetos junto con el flujo de mensajes y los enlaces entre ellos. Este punto de vista unifica la estructura de los datos y el control de flujo de datos.

3.6.2.3. Construcciones de Implementación

Los modelos UML tienen significado para el análisis lógico y para la implementación física. Un componente es una parte física reemplazable de un sistema y es capaz de responder a las peticiones descritas por un conjunto de interfaces. Un nodo es un recurso computacional que define una localización durante la ejecución de un sistema. Puede contener componentes y objetos.

3.6.2.4. Organización del Modelo

La información del modelo debe ser dividida en piezas coherentes, para que los equipos puedan trabajar en las diferentes partes de forma concurrente. El conocimiento humano requiere que se organice el contenido del modelo en paquetes de tamaño modesto. Los paquetes son unidades organizativas, jerárquicas y de propósito general de los modelos de UML. Pueden usarse para

almacenamiento, control de acceso, gestión de la configuración y construcción de bibliotecas que contengan fragmentos de código reutilizable.

3.6.3. DIAGRAMAS UML

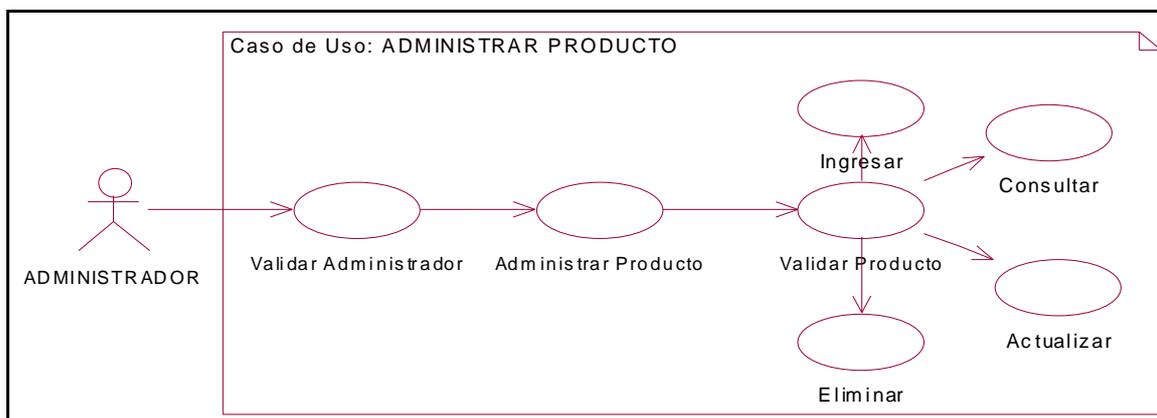
Un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones. En concreto, un diagrama ofrece una vista del sistema a modelar. Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas. Entre los cuales tenemos;

- Diagrama de casos de uso.
- Diagrama de clases.
- Diagrama de objetos.
- Diagrama de secuencia.
- Diagrama de Colaboración
- Diagrama de Actividades
- Diagrama Componentes
- Diagrama Despliegue

3.6.3.1. DIAGRAMAS DE CASOS DE USO

Casos de Uso es una técnica para capturar información de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje. No pertenece estrictamente al enfoque orientado a objeto, es una técnica para captura de requisitos.

GRAFICO 3.7



- Los Casos de Uso (*Ivar Jacobson*) describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista del usuario.
- Permiten definir los límites del sistema y las relaciones entre el sistema y el entorno.
- Los Casos de Uso son descripciones de la funcionalidad del sistema independientes de la implementación.
- Los Casos de Uso cubren la carencia existente en métodos previos (*OMT*, *Booch*) en cuanto a la determinación de requisitos.
- Los Casos de Uso particionan el conjunto de necesidades atendiendo a la categoría de usuarios que participan en el mismo.
- Están basados en el lenguaje natural, es decir, es accesible por los usuarios.

Actores

- Principales: personas que usan el sistema.
- Secundarios: personas que mantienen o administran el sistema.
- Material externo: dispositivos materiales imprescindibles que forman parte del ámbito de la aplicación y deben ser utilizados.
- Otros sistemas: sistemas con los que el sistema interactúa.



La misma persona física puede interpretar varios papeles como actores distintos, el nombre del actor describe el papel desempeñado.

Los Casos de Uso se determinan observando y precisando, actor por actor, las secuencias de interacción, los escenarios, desde el punto de vista del usuario. Los casos de uso intervienen durante todo el ciclo de vida. El proceso de desarrollo estará dirigido por los casos de uso. Un escenario es una instancia de un caso de uso.

UML define cuatro tipos de relación en los Diagramas de Casos de Uso:

- Comunicación

- Inclusión : una instancia del Caso de Uso origen incluye también el comportamiento descrito por el Caso de Uso destino. «include» reemplazó al denominado «uses»
- Extensión : el Caso de Uso origen extiende el comportamiento del Caso de Uso destino. «extend»
- Herencia : el Caso de Uso origen hereda la especificación del Caso de Uso destino y posiblemente la modifica y/o amplía.

Parámetros para la construcción de un caso de uso:

Un caso de uso debe ser simple, inteligible, claro y conciso. Generalmente hay pocos actores asociados a cada Caso de Uso. Preguntas clave:

- cuáles son las tareas del actor?
- qué información crea, guarda, modifica, destruye o lee el actor?
- debe el actor notificar al sistema los cambios externos?
- debe el sistema informar al actor de los cambios internos?

La descripción del Caso de Uso comprende:

- el inicio: cuándo y qué actor lo produce?
- el fin: cuándo se produce y qué valor devuelve?
- la interacción actor-caso de uso: qué mensajes intercambian ambos?
- objetivo del caso de uso: qué lleva a cabo o intenta?
- cronología y origen de las interacciones
- repeticiones de comportamiento: qué operaciones son iteradas?
- situaciones opcionales: qué ejecuciones alternativas se presentan en el caso de uso?

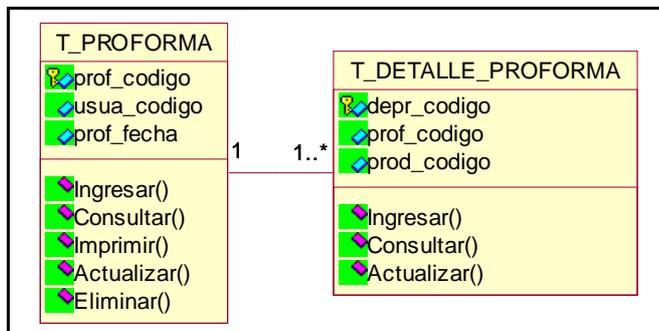
3.6.3.2. DIAGRAMA DE CLASES

El Diagrama de Clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para

atributos y operaciones. El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones.

El mundo real puede ser visto desde abstracciones diferentes (subjetividad).

GRÁFICO 3.8.



Mecanismos de abstracción:

- Clasificación / Instanciación
- Composición / Descomposición
- Agrupación / Individualización
- Especialización / Generalización

La clasificación es uno de los mecanismos de abstracción más utilizados. La clase define el ámbito de definición de un conjunto de objetos, y cada objeto pertenece a una clase, Los objetos se crean por instanciación de las clases. Cada clase se representa en un rectángulo con tres compartimientos:

- nombre de la clase
- atributos de la clase
- operaciones de la clase

Los atributos de una clase no deberían ser manipulables directamente por el resto de objetos. Por esta razón se crearon niveles de visibilidad para los elementos:

- Privado : es el más fuerte. Esta parte es totalmente invisible (excepto para clases friends en terminología C++)
- Los atributos/operaciones protegidos están visibles para las clases friends y para las clases derivadas de la original.

- Los atributos/operaciones públicos son visibles a otras clases (cuando se trata de atributos se está transgrediendo el principio de encapsulación)

Los enlaces entre objetos pueden representarse entre las respectivas clases y sus formas de relación son:

- Asociación y Agregación (vista como un caso particular de asociación)
- Generalización/Especialización.

Las relaciones de Agregación y Generalización forman jerarquías de clases.

Asociación:

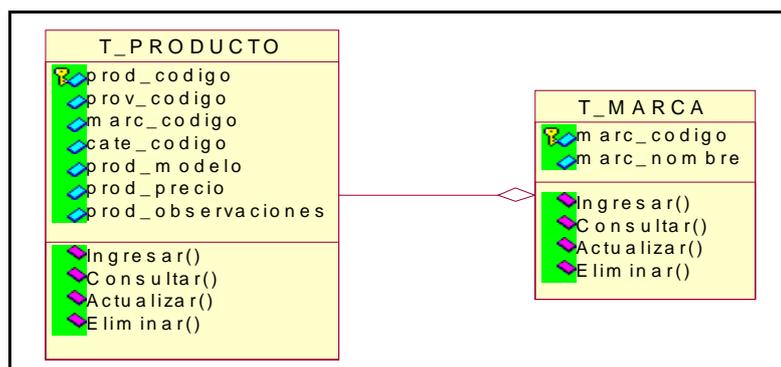
La asociación expresa una conexión bidireccional entre objetos. Una asociación es una abstracción de la relación existente en los enlaces entre los objetos. Puede determinarse por la especificación de multiplicidad (mínima...máxima)

- Uno y sólo uno
- 0..1 Cero o uno
- M..N Desde M hasta N (enteros naturales)
- 0..* Cero o muchos
- 1..* Uno o muchos (al menos uno)

Agregación:

La agregación representa una relación parte de entre objetos. En UML se proporciona una escasa caracterización de la agregación. Esta relación puede ser caracterizada con precisión determinando las relaciones de comportamiento y estructura que existen entre el objeto agregado y cada uno de sus objetos componentes.

GRÁFICO 3.9.



Generalización:

Permite gestionar la complejidad mediante un ordenamiento taxonómico de clases, se obtiene usando los mecanismos de abstracción de Generalización y/o Especialización. La Generalización consiste en factorizar las propiedades comunes de un conjunto de clases en una clase más general. Los nombres usados: clase padre - clase hija. Otros nombres: superclase - subclase, clase base - clase derivada. Las subclases heredan propiedades de sus clases padre, es decir, atributos y operaciones (y asociaciones) de la clase padre están disponibles en sus clases hijas. La Generalización y Especialización son equivalentes en cuanto al resultado: la jerarquía y herencia establecidas. Generalización y Especialización no son operaciones reflexivas ni simétricas pero sí transitivas. La especialización es una técnica muy eficaz para la extensión y reutilización. La noción de clase está próxima a la de conjunto.

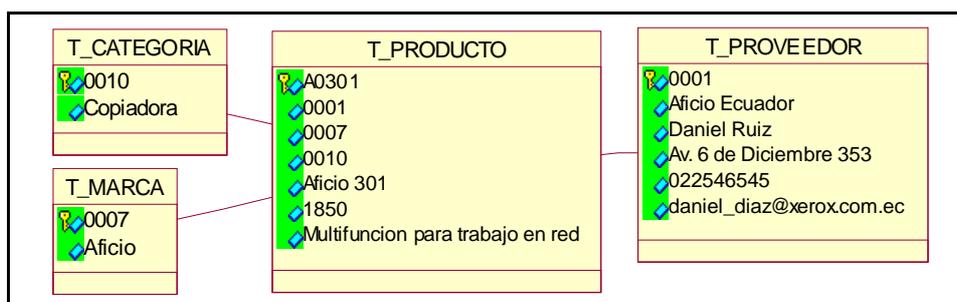
3.6.3.3. DIAGRAMA DE OBJETOS

Objeto es una entidad discreta con límites bien definidos y con identidad, es una unidad atómica que encapsula estado y comportamiento. La encapsulación en un objeto permite una alta cohesión y un bajo acoplamiento. el Objeto es reconocido también como una instancia de la clase a la cual pertenece.

La encapsulación presenta tres ventajas básicas:

- Se protegen los datos de accesos indebidos
- El acoplamiento entre las clases se disminuye
- Favorece la modularidad y el mantenimiento

GRÁFICO 3.10.



Un objeto se puede ver desde dos perspectivas relacionadas: como una entidad de un determinado instante de tiempo que posee un valor específico y como un poseedor de identidad que tiene distintos valores a lo largo del tiempo.

Cada objeto posee su propia identidad exclusiva y se puede hacer referencia a él mediante una denominación exclusiva que permite accederle. El Modelado de Objetos permite representar el ciclo de vida de los objetos a través de sus interacciones. En UML, un objeto se representa por un rectángulo con un nombre subrayado.

- Objeto = Identidad + Estado + Comportamiento
- El estado está representado por los valores de los atributos.
- Un atributo toma un valor en un dominio concreto.

La regla general para la notación de instancias consiste en utilizar el mismo símbolo geométrico que el descriptor. En la instancia se muestran los posibles valores pero las propiedades compartidas sólo se pone de manifiesto en el descriptor. La notación canónica es un rectángulo con tres compartimientos. En el primero va el nombre del objeto, en el segundo sus atributos y en el tercero sus operaciones. Este último puede ser omitido si así se prefiere.

3.6.3.3.1. OID (Object Identifier)

Cada objeto posee un oid. El oid establece la identidad del objeto y tiene las siguientes características:

- Constituye un identificador único y global para cada objeto dentro del sistema.
- Es determinado en el momento de la creación del objeto.
- Es independiente de la localización física del objeto, es decir, provee completa independencia de localización.
- Es independiente de las propiedades del objeto, lo cual implica independencia de valor y de estructura.

No cambia durante toda la vida del objeto. Además, un oid no se reutiliza aunque el objeto deje de existir.

3.6.3.3.2. Características Alrededor de un Objeto:

- Estado:

El estado evoluciona con el tiempo. Algunos atributos pueden ser constantes, el comportamiento agrupa las competencias de un objeto y describe las acciones y reacciones de ese objeto. Las operaciones de un objeto son consecuencia de un estímulo externo representado como mensaje enviado desde otro objeto.

- Persistencia:

La persistencia de los objetos designa la capacidad de un objeto trascender en el espacio/tiempo, podremos después reconstruirlo, es decir, cogerlo de memoria secundaria para utilizarlo en la ejecución (materialización del objeto). Los lenguajes OO no proponen soporte adecuado para la persistencia, la cual debería ser transparente, un objeto existe desde su creación hasta que se destruya.

- Comunicación:

Un sistema informático puede verse como un conjunto de objetos autónomos y concurrentes que trabajan de manera coordinada en la consecución de un fin específico. El comportamiento global se basa pues en la comunicación entre los objetos que la componen.

3.6.3.4. DIAGRAMA DE SECUENCIA

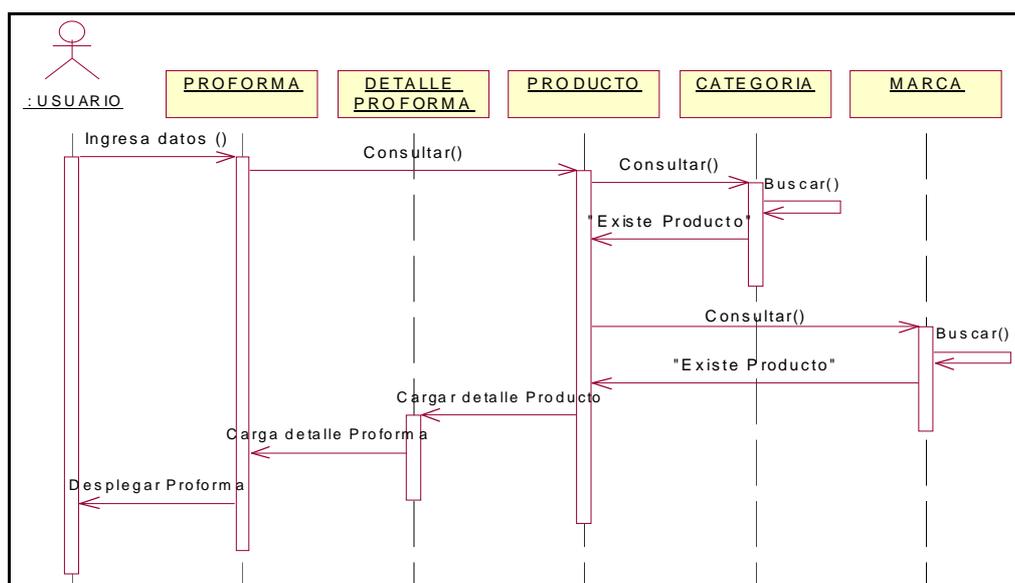
Los diagramas de secuencia proporcionan una vista detallada de un caso de uso. Muestran una interacción organizada en una secuencia de tiempo y ayuda a documentar el flujo de lógica dentro de la aplicación. Los participantes se muestran en el contexto de los mensajes que se transfieren entre ellos. En un sistema de software amplio, el diagrama de secuencia puede incluir un mayor número de detalles y contener miles de mensajes.

Un diagrama de Secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo. Esta descripción es importante porque puede dar detalle a los casos de uso, aclarándolos a nivel de mensajes de los objetos

existentes, como también muestra el uso de los mensajes de las clases diseñadas en el contexto de una operación. Observando que mensajes se envían a los objetos, componentes o casos de uso, y viendo grosso modo cuanto tiempo consume el método invocado, los Diagramas de Secuencia nos ayudan a comprender los cuellos de botella potenciales, para así poder eliminarlos. A la hora de documentar un Diagrama de Secuencia resulta importante mantener los enlaces de los mensajes a los métodos apropiados del Diagrama de Clases.

Mientras que el diagrama de caso de uso permite el modelado de una vista 'business' del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos.

GRÁFICO 3.11.



3.6.3.4.1. Elementos de un Diagrama de Secuencia

- Línea de vida de un objeto

Un objeto se representa como una línea vertical punteada con un rectángulo de encabezado y con rectángulos a través de la línea principal que denotan la ejecución de métodos. El rectángulo de encabezado contiene el nombre del objeto y el de su clase, en un formato: nombreObjeto: nombreClase.

- Activación

Muestra el periodo de tiempo en el cual el objeto se encuentra desarrollando alguna operación, ya sea por sí mismo o por medio de delegación a alguno de sus atributos. Se denota como un rectángulo delgado sobre la línea de vida del objeto.

- Mensaje

El envío de mensajes entre objetos se denota mediante una línea sólida dirigida, desde el objeto que emite el mensaje hacia el objeto que lo ejecuta.

- Tiempos de transición

En un ambiente de objetos concurrentes o de demoras en la recepción de mensajes, es útil agregar nombres a los tiempos de salida y llegada de mensajes.

- Destrucción de un objeto

Se representa como una X al final de la línea de ejecución del objeto.

3.6.3.5. DIAGRAMA DE COLABORACIÓN

Los diagramas de colaboración constituyen otro tipo de diagrama de interacción. Al igual que los diagramas de secuencia, muestran cómo operan los objetos de un grupo entre sí en un caso de uso. A cada mensaje se le asigna un número para documentar el orden en el que tiene lugar.

El Diagrama de Colaboración modela objetos y enlaces significativos dentro de una interacción, muestra los roles en la interacción en disposición geométrica y la implementación de una operación.

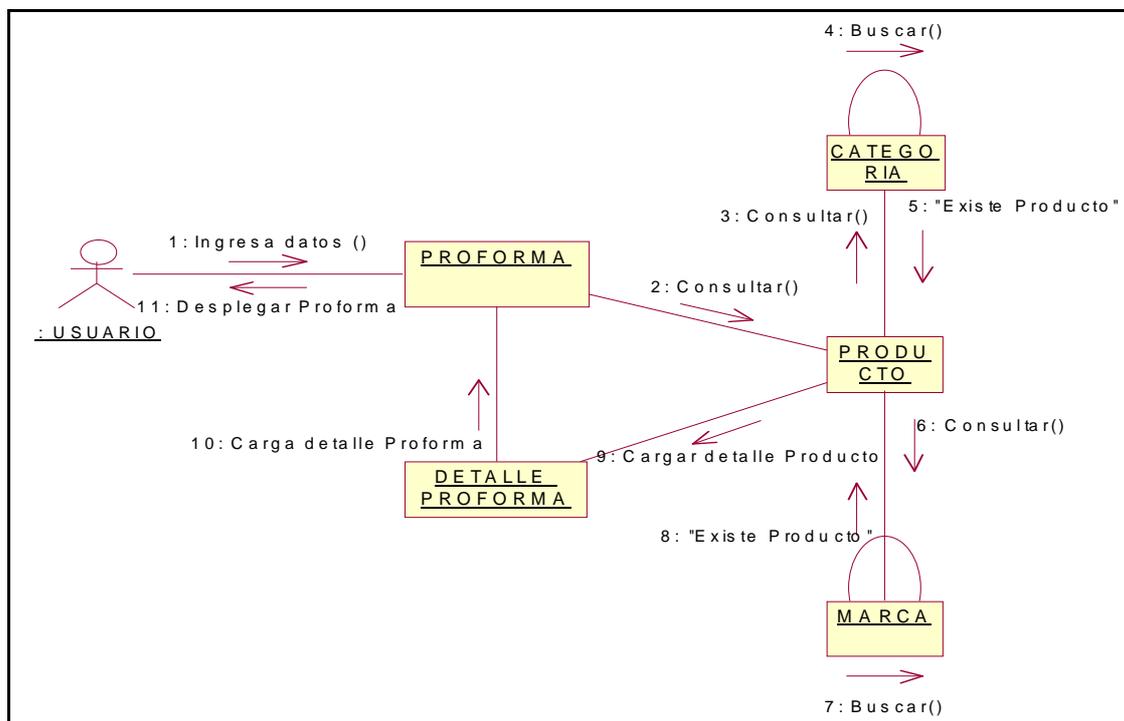
Los mensajes se muestran como flechas ligadas a líneas de relación que conectan a los roles.

La secuencia de mensaje se representa con número secuencial que precede a descripción de mensaje

El Diagrama de colaboración puede obtenerse automáticamente a partir del correspondiente diagrama de Secuencia o viceversa.

La estructura estática viene dada por los enlaces; la dinámica por el envío de mensajes por los enlaces.

GRÁFICO 3.12.



3.6.3.5.1. Elementos de un Diagrama de Colaboración

- Objeto

Se representa con un rectángulo que contiene el nombre y la clase del objeto en un formato

nombre objeto: nombre clase

- Enlace

Un enlace es una instancia de una asociación en un diagrama de clases. Se representa como una línea continua que une a dos objetos, acompañada por un número que indica el orden dentro de la interacción. Pueden darse varios niveles de subíndices para indicar anidamiento de operaciones. Se pueden utilizar estereotipos para indicar si el objeto que recibe el mensaje es un atributo, un parámetro de un mensaje anterior, si es un objeto local o global.

- Flujo de mensajes

Expresa el envío de un mensaje. Se representa mediante una flecha dirigida cerca de un enlace

- Objeto Compuesto

Es una representación alternativa de un objeto y sus atributos. En esta representación se muestran los objetos contenidos dentro del rectángulo que representan al objeto que los contiene.

3.6.3.6. DIAGRAMA DE ESTADOS

Muestra el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación, junto con los cambios que permiten pasar de un estado a otro. Los Diagramas de Estado representan autómatas de estados finitos, desde el p.d.v. de los estados y las transiciones. Son útiles sólo para los objetos con un comportamiento significativo. Cada objeto está en un estado en cierto instante. El estado está caracterizado parcialmente por los valores algunos de los atributos del objeto. El estado en el que se encuentra un objeto determina su comportamiento. Cada objeto sigue el comportamiento descrito en el Diagrama de Estados asociado a su clase. Los Diagramas de Estados y escenarios son complementarios, los Diagramas de Estados son autómatas jerárquicos que permiten expresar concurrencia, sincronización y jerarquías de objetos, son grafos dirigidos y deterministas. La transición entre estados es instantánea y se debe a la ocurrencia de un evento.

3.6.3.6.1. Elementos de un Diagrama de Estados

- Estado

Identifica un periodo de tiempo del objeto (no instantáneo) en el cual el objeto está esperando alguna operación, tiene cierto estado característico o puede recibir cierto tipo de estímulos. Se representa mediante un rectángulo con los bordes redondeados, que puede tener tres compartimientos: uno para el nombre, otro para el valor característico de los atributos del objeto en ese estado y otro para las acciones que se realizan al entrar, salir o estar en un estado.

- Eventos

Es una ocurrencia que puede causar la transición de un estado a otro de un objeto. Esta ocurrencia puede ser una de varias cosas:

- Condición que toma el valor de verdadero o falso
- Recepción de una señal de otro objeto en el modelo
- Recepción de un mensaje
- Paso de cierto período de tiempo, después de entrar al estado o de cierta hora y fecha particular

El nombre de un evento tiene alcance dentro del paquete en el cual está definido, no es local a la clase que lo nombre.

3.6.3.7. DIAGRAMA DE ACTIVIDADES

Un Diagrama de actividad es una especialización del Diagrama de Estado, organizado respecto de las acciones y usado para especificar

- Un método
- Un caso de uso
- Un proceso de negocio

Un estado de actividad representa una actividad: un paso en el flujo de trabajo o la ejecución de una operación. Un grafo de actividades describe grupos secuenciales y concurrentes de actividades. Los grafos de actividades se muestran en el Diagrama de Actividades

Las actividades se enlazan por transiciones automáticas. Cuando una actividad termina se desencadena el paso a la siguiente actividad.

Un diagrama de actividades es provechoso para entender el comportamiento de alto nivel de la ejecución de un sistema. Sin profundizar en los detalles internos de los mensajes. Los parámetros de entrada y salida de una acción se puede mostrar usando las relaciones de flujo que conectan la acción y un estado de flujo de objeto

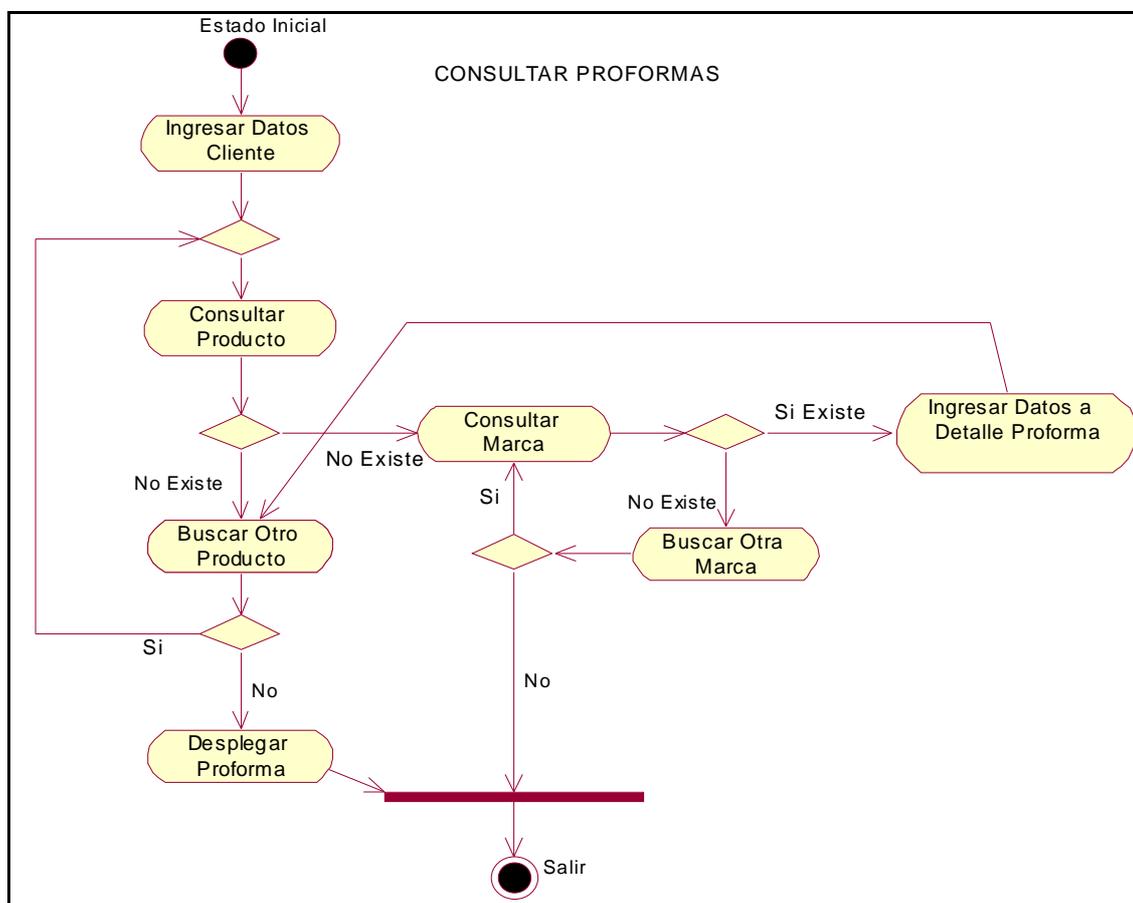
Un gráfico de actividades contiene estados de actividad que representan la ejecución de una secuencia en un procedimiento, o el funcionamiento de una actividad en un flujo de trabajo. En ves de esperar un evento, como en un estado de espera normal, un estado de actividad espera la terminación de su computo. Cuando la actividad termina, entonces la ejecución procede al siguiente estado de

actividad dentro del diagrama. Una transición de terminación es activada en un diagrama de actividades cuando se completa la actividad precedente.

Un grafo de actividades puede contener también estados de acción que son similares a los de actividad pero son atómicos y no permiten transiciones mientras están activos. Los estados de acción se deben utilizar para las operaciones cortas de mantenimiento,

Un diagrama de actividades pueden contener bifurcaciones, así como divisiones de control en hilos concurrentes. Los hilos concurrentes representan actividades que se pueden realizar concurrentemente por los diversos objetos o personas. La concurrencia se representa a partir de la agregación, en la cual cada objeto tiene su propio hilo. Las actividades concurrentes se pueden realizar simultáneamente o en cualquier orden. Un diagrama de actividades es como un organigrama tradicional.

GRÁFICO 3.13



3.6.3.7.1. Características del Diagrama de Actividades

- Notación

Un estado de actividad se representa como una caja con los extremos redondeados que contiene una descripción de actividad. Las transacciones simples de terminación se muestran como flechas. Las ramas se muestran como condiciones o como diamantes con múltiples flechas de salida etiquetadas. Una división o una unión de control se representa con múltiples flechas que entran o salen de la barra gruesa de sincronización.

Cuando es necesario incluir eventos externos, la recepción de un evento se puede mostrar como un disparador en una transición, o como un símbolo especial que denota la espera de una señal. A menudo es útil organizar las actividades en un modelo según su responsabilidad. Esta clase de asignación puede mostrarse organizando las actividades en regiones distintas separadas por líneas en el diagrama. Debido a su aspecto, esto es conocido como Calles.

Un diagrama de actividades puede mostrar el flujo de objetos como valores. Para un valor de salida, se dibuja una flecha con línea discontinua desde la actividad al objeto. Para un valor de entrada, se dibuja una flecha con línea discontinua desde el objeto a una actividad.

3.6.3.8. DIAGRAMA DE COMPONENTES

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, etc. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

Un diagrama de componentes representa las dependencias entre componentes software, incluyendo componentes de código fuente, componentes del código binario, y componentes ejecutables. Un módulo de software se puede representar como componente. Algunos componentes existen en tiempo de compilación,

algunos en tiempo de enlace y algunos en tiempo de ejecución, otros en varias de éstas.

Un componente de sólo compilación es aquel que es significativo únicamente en tiempo de compilación. Un componente ejecutable es un programa ejecutable.

Un diagrama de componentes tiene sólo una versión con descriptores, no tiene versión con instancias. Para mostrar las instancias de los componentes se debe usar un diagrama de despliegue.

Un diagrama de componentes muestra clasificadores de componentes, las clases definidas en ellos, y las relaciones entre ellas. Los clasificadores de componentes también se pueden anidar dentro de otros clasificadores de componentes para mostrar relaciones de definición.

Un diagrama que contiene clasificadores de componentes y de nodo se puede utilizar para mostrar las dependencias del compilador, que se representa como flechas con líneas discontinuas (dependencias) de un componente cliente a un componente proveedor del que depende. Los tipos de dependencias son específicos del lenguaje y se pueden representar como estereotipos de las dependencias.

El diagrama también puede usarse para mostrar interfaces y las dependencias de llamada entre componentes, usando flechas con líneas discontinuas desde los componentes a las interfaces de otros componentes.

El diagrama de componente hace parte de la vista física de un sistema, la cual modela la estructura de implementación de la aplicación por sí misma, su organización en componentes y su despliegue en nodos de ejecución. Esta vista proporciona la oportunidad de establecer correspondencias entre las clases y los componentes de implementación y nodos. La vista de implementación se representa con los diagramas de componentes.

3.6.3.8.1. Componentes

Es una parte física reemplazable de un sistema que empaqueta su implementación y es conforme a un conjunto de interfaces a las que proporciona su realización.

Algunos componentes tienen identidad y pueden poseer entidades físicas, que incluyen objetos en tiempo de ejecución, documentos, bases de datos, etc. Los

componentes existentes en el dominio de la implementación son unidades físicas en los computadores que se pueden conectar con otros componentes, sustituir, trasladar, archivar, etc.

Los componentes tienen dos características: Empaquetan el código que implementa la funcionalidad de un sistema, y algunas de sus propias instancias de objetos que constituyen el estado del sistema.

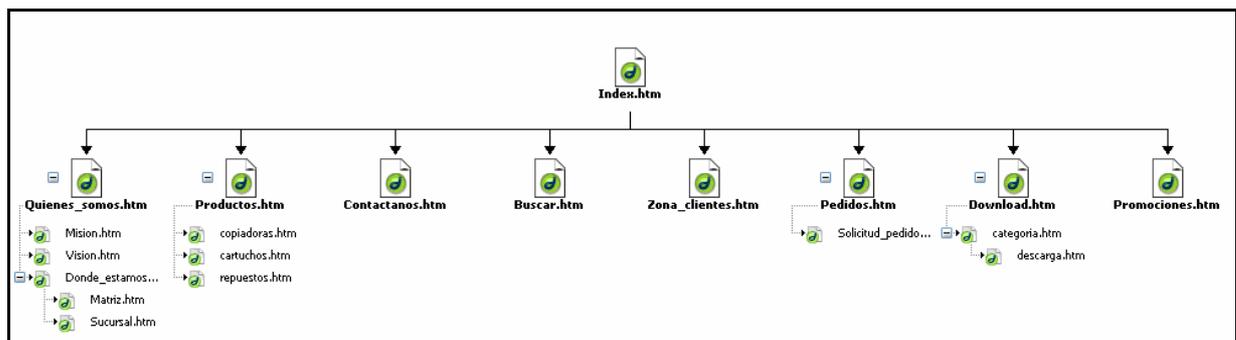
3.6.3.8.2. Identidad

Un componente de identidad tiene identidad y estado. Posee los objetos físicos que están situados en él. Puede tener atributos, relaciones de composición con los objetos poseídos, y asociaciones con otros componentes. Desde este punto de vista es una clase. Sin embargo la totalidad de su estado debe hacer referencia a las instancias que contiene.

3.6.3.9. DIAGRAMA DE DESPLIEGUE

Los Diagramas de Despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo: Dispositivos, procesadores, memoria

GRÁFICO 3.14.



Los nodos se interconectan mediante soportes bidireccionales que pueden a su vez estereotiparse. Esta vista permite determinar las consecuencias de la distribución y la asignación de recursos. Las instancias de los nodos pueden contener instancias de ejecución, como instancias de componentes y objetos. El modelo puede mostrar dependencias entre las instancias y sus interfaces, y también modelar la migración de entidades entre nodos u otros contenedores.

Esta vista tiene una forma de descriptor y otra de instancia. La forma de instancia muestra la localización de las instancias de los componentes específicos en instancias específicas del nodo como parte de una configuración del sistema. La forma de descriptor muestra que tipo de componentes pueden subsistir y qué tipo de nodos se pueden conectar, de forma similar a una diagrama de clases, esta forma es menos común que la primera. (*I. Jacobson, 2000*).

CAPITULO 4

4.1. HERRAMIENTAS DE CONSTRUCCION WEB

En el presente capítulo se describirá las características mas importantes de las software utilizado en el desarrollo del nuestro Sitio Web, dicho software proporcionan una combinación potente como herramientas visuales para el desarrollo de aplicaciones Web.

4.2. MICROSOFT FLASH MX 2004

Flash es la plataforma de software de predominancia indiscutible, usada por más de un millón de profesionales y con una presencia en más del 97% de los equipos de escritorio con conexión a Internet en todo el mundo, así como en una amplia gama de dispositivos.

Macromedia Flash MX 2004 le permite a los diseñadores y desarrolladores integrar video, texto, audio y gráficos en experiencias dinámicas que le permiten al cliente imbuirse en su vivencia y que producen resultados superiores para marketing y presentaciones interactivas, aprendizaje electrónico e interfaces de usuario de aplicaciones. Los Timeline Effects permiten a los diseñadores añadir transiciones comunes como difusiones y sombras sin necesidad de scripting. Comportamientos ("Behaviours") pre-definidos, eliminan la necesidad de un scripting complejo de navegación. El compilador de alto desempeño mejora significativamente el nivel de reproducción del sonido, incluso con contenido creado para versiones anteriores de Flash Player Mejoras en Flash Player 7 entregan un desempeño adicional que ofrecen la mejor experiencia para el usuario final. Flash MX 2004 ofrece integración única a Macromedia Dreamweaver, Fireworks y las otras herramientas de diseño

Flash MX 2004 también incluye Elementos MX para Flash, componentes de aplicación que facilitan a los desarrolladores el crear rápidamente grandes experiencias con una imagen visual consistente. (*Horizonteweb.com, 2005*)

El nuevo lenguaje ActionScript 2.0 permite a los desarrolladores añadir interactividad sofisticada a su contenido. Nuevo soporte para Hojas de Estilo en Cascada –CSS- permite a los usuarios entregar sitios híbridos que mezclan Flash y HTML con una imagen visual consistente.

Los Elementos MX para Flash incluyen plantillas para proyectos comunes como presentaciones y anuncios para la Web, así como componentes útiles para la interfase de usuario como los “data grid”, un “player” para multimedia y un “accordion pane”. (*Horizonteweb.com, 2005*)

4.3. MICROSOFT DREAMWEAVER MX 2004

Dreamweaver MX 2004 es la opción profesional para la creación de sitios y aplicaciones web. Proporciona una combinación potente de herramientas visuales de disposición, características de desarrollo de aplicaciones y soporte para la edición de código. Gracias a las robustas características para la integración y diseño basado en CSS, Dreamweaver permite que los diseñadores y desarrolladores web creen y manejen cualquier sitio web con toda facilidad.

Dreamweaver MX 2004 incluye potentes controles basados en normas para asegurar un diseño de alta calidad. Un entorno de diseño construido en torno a las hojas de estilo en cascada (CSS) hace posible un desarrollo más rápido y más eficiente de sitios profesionales creados con código limpio.

Dreamweaver MX 2004 es abierto y ampliable y cuenta con los productos y las tecnologías que usted usa, para darle la libertad y la flexibilidad de elegir la tecnología que más le conviene, ahora y en el futuro. Ya que permite desarrollar sitios web de HTML, XHTML, XML, ASP, ASP.NET, JSP, PHP y Macromedia ColdFusion. Además Dreamweaver MX 2004 ofrece un FTP seguro para codificar completamente todas las transferencias de archivos y evitar el acceso no autorizado a los datos, contenido de archivos, nombres de usuario y contraseñas. (*Horizonteweb.com, 2005*)

4.4. MICROSOFT FIREWORKS MX 2004

Fireworks MX 2004 tiene las herramientas que los profesionales del web necesitan para crear de todo, desde botones gráficos sencillos hasta sofisticados efectos de rollover. Permite Importar, editar e integrar fácilmente todos los principales formatos gráficos, incluidos imágenes vectoriales y de mapas de bits. Con Fireworks MX 2004 podemos exportar fácilmente imágenes de Fireworks a Flash, Dreamweaver y aplicaciones de terceros.

Fireworks MX 2004 permite crear rápidamente gráficos web de alta calidad e interactividad compleja. Fireworks ofrece todas las herramientas familiares que usted necesita: edición de fotos robusta, control de texto preciso y creación de imágenes profesional.

Fireworks proporciona un conjunto de herramientas completo y profesional para una producción rápida y actualizaciones fáciles con lo cual se reduce el tiempo necesario para el diseño y el desarrollo de imágenes, aun al trabajar con imágenes grandes. (*Horizonteweb.com, 2005*)

4.5. SERVIDOR WEB APACHE

Apache es uno de los servidores Web mas potentes del mercado, ofreciendo una perfecta combinación entre estabilidad y sencillez.

Las principales características de Apache son:

- Funcionalidad en múltiples plataformas (Unix / Linux), Windows y otras).
- Elaborado indice de directorios.
- Soporte del ultimo protocolo http 1.1.
- Sencilla administración basada en la configuración de un único archivo.
- Soporte para CGI (Common Gatewav Interface) y FastCGI

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

En la actualidad (2006), Apache es el servidor HTTP más usado, siendo el servidor HTTP del 68% de los sitios web en el mundo y creciendo aún su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft).

4.5.1. Versión 2.X

El núcleo 2.x de Apache tiene varias mejoras clave sobre el núcleo de Apache 1.x. Estas mejoras incluyen threads de UNIX, mejor soporte para plataformas no Unix (como Windows), un nuevo API, y soporte de IPV6.

4.5.2. Módulos

La arquitectura del servidor Apache es muy modular. El servidor consta de un sección *core* y mucha de la funcionalidad que podría considerarse básica para un servidor web es provista por módulos. Algunos de estos son:

- mod_ssl - Comunicaciones Seguras vía TLS.
- mod_rewrite - reescritura de direcciones servidas.
- mod_dav - Soporte del protocolo WebDAV (RFC 2518).
- mod_deflate - Compresión transparente con el algoritmo deflate del contenido enviado al cliente.
- mod_auth_ldap - Permite autenticar usuarios contra un servidor LDAP.

El servidor de base puede ser extendido con la inclusión de módulos externos entre los cuales se encuentran:

- mod_perl - Páginas dinámicas en Perl.
- mod_php - Páginas dinámicas en PHP.
- mod_python - Páginas dinámicas en Python.
- mod_jk - Conector para enlazar con el servidor Jakarta Tomcat de páginas dinámicas en Java (Servlets y JSP).

(Enciclopedia Wikipedia, 2005).

4.6. MySQL

MySQL es uno de los Sistemas Gestores de bases de Datos (SQL) más populares desarrolladas bajo la filosofía de código abierto.

La desarrolla y mantiene la empresa MySQL AB pero puede utilizarse gratuitamente y su código fuente está disponible libremente.

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad; aquellos elementos faltantes fueron llenados por la vía de las aplicaciones que la utilizan.

Poco a poco los elementos faltantes en MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL.
- Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

(Enciclopedia Wikipedia, 2005).

CAPITULO 5

5.1. CONCLUSIONES

- EL software de Open Source fue muy eficaz en el desarrollo de nuestro sitio web, por su ventaja de ser libre(sin costo) y su confiabilidad.
- El software utilizado en el desarrollo de nuestro sitio, Dreamweaver MX 2004, Fireworks MX 2004 y Flash MX 2004 fueron de gran utilidad a pesar de que se necesitaron adquirir varios conocimientos desde el Internet.
- Php 4.1.1, fue el lenguaje utilizado en la construcción de nuestras paginas dinámicas, para ser ejecutados en el servidor, además se utilizó Javascript como un lenguaje de apoyo muy eficaz para ser ejecutado del lado del cliente.
- El servidor web Apache 1.3.23 fue muy eficiente en nuestro trabajo con PHP 4.1.1 y Base de Datos MySQL 3.23.48, estos son recursos libres, de fácil instalación configuración, consistente y de calidad además presente una interfaz gráfica con lo cual se concluye que la Base de Datos fue eficaz y manejable.
- En el desarrollo de nuestro proyecto se utilizo la metodología OOHDM, la cual nos permitió planificar, diseñar y desarrollar nuestro sitio web de una manera sencilla, rápida y con el resultado de tener un Sitio de calidad.
- La realización de este proyecto nos permitió fortalecer todos los conocimientos adquiridos en nuestra formación académica.

5.2. RECOMENDACIONES

- El sitio debe ser actualizado constantemente, se recomienda un periodo de actualización mensual, especialmente en los módulos de producto y descargas para brindar un servicio eficiente y confiable al usuario.
- Se recomienda crear un foro de preguntas y respuestas que permita una mejor comunicación entre el técnico de la empresa y sus clientes.
- Es importante capacitar al personal que administrará el sitio web para tener un correcto funcionamiento del Sistema, con el fin de brindar un buen servicio al usuario.
- Se recomienda sacar respaldos de la base de datos mensualmente o cada vez que se ingresen o actualice la información y así evitar pérdidas o inconsistencia de la información almacenada en la base de datos .
- Se recomienda instalar seguridades como fireware, antivirus, antispay, etc que protejan al sitio y a su base de datos de intrusos y ataques y Así evitar que la información, tanto de Base de Datos como del presentado en las páginas web, pueda ser alterada de alguna manera.

BIBLIOGRAFÍA

DELGADO D., HERRERA M., VELÁSQUEZ J.; *Sistema de Administración Web*; Barcelona; 2004.

GRAIG L.; *UML y Patrones, Introducción al Análisis y Diseño Orientado a Objetos*; México; ED. Prentice Hall; 1998.

JACOBSON Ivar, BOOCH Grady, RUMBAUGH James; *El Proceso Unificado de Desarrollo de Software*; ED. Pearson Education S.A.; Madrid; 2002.

MENDOZA M., BARRIOS J.; *Sistema de Administración*; Mérida; 2004.

SILVA Darío, MERCERAT Bárbara; *Construyendo Aplicaciones Web con una Metodología Orientada a Objetos*; Buenos Aires; 2002.

VILLALBA Carlos; *Metodología de la Investigación*; ED. SUR; Quito; 2004.

WESLEY Addison, JACOBSON Ivar, BOOCH Grady, RUMBAUGH James; *El Lenguaje Unificado de Modelado*; ED. Pearson Education S.A.; Madrid; 2000

SITIOS WEB DE REFERENCIA

<http://www.telemidia.puc-rio.br/oohdm/oohdm.html>

<http://www.ingenierossoftware.com>

<http://www.creangel.com/uml/intro.php>

<http://www.clikear.com/metodo/espinal>

<http://dewey.uab.es/pmarques/calida2.htm>

<http://www.clikear.com/manuales/uml/>

<http://www.creangel.com/uml/diagramas.php>

http://es.wikipedia.org/wiki/Diseño_de_paginas_web

<http://www.desarrolloweb.com/articulos/715.php?manual=27>

http://www.ulpiano.com/Tutoriales_Nombres_de_Dominio.htm

<http://web.usersnet.net/soporte/categorias.php>

<http://eia.udg.es/~atm/tcp-ip/>

<http://www.interhelp.org/>

<http://lg.msn.com/intl/es/tutorial/intro3.htm>

<http://www.learnthenet.com/spanish/glossary/intadd.htm>

<http://www.monografias.com/html>

ANEXOS

TABLA DE CONTENIDO ANEXOS

1.	MANUAL TÉCNICO	01
1.1.	DIAGRAMAS DE CASOS DE USO DEL USUARIO	01
1.2.	DIAGRAMAS DE CASOS DE USO DEL CLIENTE	01
1.3.	DIAGRAMAS DE CASOS DE USO DEL ADMINISTRADOR	02
1.4.	DIAGRAMA DE CLASES	03
1.5.	DICCIONARIO DEL DIAGRAMA DE CLASES	04
1.6.	DIAGRAMA DE OBJETOS	05
1.7.	DIAGRAMAS DE SECUENCIA	06
1.8.	DIAGRAMAS DE COLABORACIÓN	12
1.9.	DIAGRAMAS DE ACTIVIDADES	17
1.10.	DIAGRAMA ARQUITECTÓNICO	21
1.11.	DIAGRAMA NAVEGACIONAL	22
2.	MANUAL DEL USUARIO	23
2.1.	HOME	24
2.2.	PRODUCTOS	24
2.2.1.	Sección Copiadoras	25
2.2.2.	Sección Impresoras	25
2.2.3.	Sección Cartuchos	26
2.2.4.	Sección Accesorios	27
2.3.	SERVICIO TECNICO	28
2.4.	REGISTRATE	29
2.5.	PROFORMAS	29
2.6.	DOWNLOAD	31
2.7.	PROMOCIONES	32
2.8.	CONTACTANOS	32

3. MANUAL DEL ADMINISTRADOR	33
3.1. MANTENIMIENTO IMPORTADORA	34
3.2. MANTENIMIENTO DESCARGAS	34
3.3. ACTUALIZAR ACCESORIOS	35
3.4. MANTENIMIENTO PROMOCIONES	35
3.5. MANTENIMIENTO USUARIOS	36
3.6. MANTENIMIENTO PRODUCTOS	38
3.7. MANTENIMIENTO PROVEEDORES	40
3.8. MANTENIMIENTO CATEGORÍA	42
3.9. MANTENIMIENTO DE MARCA	44
3.10. MANTENIMIENTO DE CIUDADES	46