

# **ESCUELA POLITECNICA NACIONAL**

## **ESCUELA DE FORMACION TECNOLÓGICA**

### **DISEÑO DE UN PORTAL WEB DINÁMICO CON ACCESO A DATOS PARA LA AGENCIA DE VIAJES “ABSOLUT JOY EXPEDITIONS TRAVEL AGENCY”**

**PROYECTO PREVIO A LA OBTENCION DEL TITULO DE TECNÓLOGO EN  
ANÁLISIS DE SISTEMAS INFORMÁTICOS**

**MAYRA CARMITA CRIOLLO GUATAPI  
JOSE GUILLERMO SERRANO FUEL**

**DIRECTORA: ING. MYRIAM PEÑAFIEL**

**Quito, Marzo 2007**

## **DECLARACIÓN**

Nosotros MAYRA CARMITA CRIOLLO GUATAPI y JOSE GUILLERMO SERRANO FUEL 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.

---

**ALONSO FERNÁNDEZ CARPIO**

---

**MARIA DEL CARMEN AGUAYO**

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por CARMITA CRIOLLO GUATAPI y JOSE GUILLERMO SERRANO FUEL, bajo mi supervisión.

---

**Ing. Mirian Peñafiel A. MSc**

## **AGRADECIMIENTOS**

Agradecemos a Dios por darnos la vida, y la inteligencia para llevar a cabo nuestro proyecto, a nuestros padres por brindarnos su apoyo y confianza y a nuestro maestros por impartir sus conocimientos y así formarnos como profesionales listos para enfrentarnos a los retos del mercado laboral actual.

## **DEDICATORIA**

Dedicamos el presente Proyecto a nuestros padres, por todo el apoyo y paciencia que nos han brindado durante toda nuestra carrera estudiantil ya que ellos fueron el pilar fundamental para cumplir con nuestro objetivo.

# INDICE DE CAPITULOS

INDICE DE CAPITULOS .....	I
CAPITULO I.....	8
INTRODUCCIÓN .....	8
1.1 Ámbito.....	8
1.2 Planteamiento del Problema.....	8
1.2.1 Solución.....	9
1.3 Formulación y Sistematización del Problema .....	9
1.3.1 Formulación .....	9
1.3.2 Sistematización.....	9
1.4 Objetivos de la Investigación .....	10
1.4.1 Objetivo General .....	10
1.4.2 Objetivos Específicos .....	10
1.5 Justificación del Proyecto .....	10
1.6 Presupuesto.....	11
1.6.1 Costo del Proyecto .....	11
1.6.2 Costo Estimado del Proyecto según COCOMO BÁSICO .....	11
CAPITULO II.....	14
MARCO DE REFERENCIA .....	14
2.1 Ingeniería WEB .....	14
2.2 APLICACIONES WEB .....	15
2.2.1 Arquitectura Web .....	16
2.2.2 Aplicaciones Multinivel .....	19
2.3 Herramientas de Soporte al Desarrollo.....	21
2.3.1 JavaScript .....	21
2.3.2 XHTML .....	22
2.3.3 CSS.....	24
2.3.4 Apache Web Server.....	28
2.3.5 PHP 5 .....	28
2.3.6 XAMPP .....	30
2.3.7 MySQL .....	31
2.4 HERRAMIENTAS DE APOYO .....	32
2.4.1 Dreamweaver MX 2004.....	32
2.4.2 MACROMEDIA FIREWORKS.....	33
2.4.3 PowerDesigner .....	34
2.4.5 Rational Rose .....	35
CAPITULO III.....	36
ASPECTOS METODOLÓGICOS.....	36
3.1 PARADIGMA ESPIRAL ORIENTADO A LA WEB.....	36
3.2 OOHDM: "Object Oriented Hypermedia Design Method" .....	37
3.2.1 Diseño Conceptual o Análisis de Dominio .....	38
3.2.2 Diseño Navegacional.....	38
3.2.3 Diseño de la Interface Abstracta .....	38

3.2.4 Implementación.....	39
3.3 UML.....	40
3.3.1 Diagrama de Casos de Uso .....	40
CAPITULO IV .....	49
CONCLUSIONES Y RECOMENDACIONES .....	49

# CAPITULO I

## INTRODUCCIÓN

### 1.1 Ámbito

La Agencia de Viajes "ABSOLUT JOY EXPEDITIONS Travel Agency", nace el 01 de Octubre de 2003 con la finalidad de promocionar paquetes turísticos en el Ecuador.

Actualmente dicha Institución cuenta con 10 paquetes turísticos a promocionar; y 1200 correos electrónicos de posibles clientes. Para su Administración cuenta con los Departamentos de Gerencia, Contabilidad y Secretaría.

La Agencia esta ubicada la Avenida Amazonas N24-71 y Wison.

### 1.2 Planteamiento del Problema

La Agencia de Viajes "ABSOLUT JOY EXPEDITIONS Travel Agency" basa su marketing de relación con sus clientes vía correo electrónico. Al momento para obtener las direcciones, y posteriormente enviar las promociones turísticas, la Agencia lo hace a través de una visita personal a empresas e instituciones. Dichas direcciones son almacenadas en una hoja electrónica Excel, por lo que se presentan entre otros los siguientes problemas.

- El personal Administrativo ocupa mucho tiempo enviando las promociones a los diferentes correos electrónicos.
- La Agencia no es muy conocida, por lo cual no es posible ampliar el mercado nacional y sobre todo internacional.
- Pocas personas interesadas en reservar los paquetes que la Agencia ofrece.
- Pérdida de información debido a que las reservaciones las almacenan en una hoja electrónica Excel.

### **1.2.1 Solución**

Para ayudar a la solución de los problemas que se presentan en esta Institución proponemos la creación de un portal Web en el cual:

- La información de la Agencia y de los paquetes turísticos que ésta ofrece esté siempre disponible.
- Se amplíe la posibilidad de atraer nuevos clientes nacionales e internacionales.
- Los usuarios pueden consultar paquetes disponibles.
- Se puede reservar paquetes turísticos.
- La información de los e-mails se almacene en una Base de Datos confiable.
- Se agilice el envío de correo electrónico a los clientes.

## **1.3 Formulación y Sistematización del Problema**

### **1.3.1 Formulación**

¿Cómo administrar mejor las reservaciones e información general de la Agencia de Viajes?

### **1.3.2 Sistematización**

¿Cómo mantener una información detallada de las promociones turísticas?

¿Cómo informarse de la disponibilidad de los diferentes paquetes turísticos?

¿Cómo garantizar a los usuarios el acceso al sitio?

¿Cómo la Agencia registra los datos de personas interesadas en un paquete turístico?

¿Cómo promocionar los servicios que presta la Agencia de Viajes?

¿Cómo los clientes pueden solicitar información más detallada y confidencial?

¿Cómo posibilitar que usuarios extranjeros puedan hacer uso de los servicios que presta la página Web?

## **1.4 Objetivos de la Investigación**

### **1.4.1 Objetivo General**

Promocionar todos los paquetes turísticos que ofrece la agencia de viajes “ABSOLUT JOY EXPEDITIONS Travel Agency” a nivel nacional e internacional y facilitar la interrelación con sus clientes, a través del desarrollo y posterior implementación de un sitio Web.

### **1.4.2 Objetivos Específicos**

- 1) Mantener un catálogo actualizado de la oferta de paquetes turísticos a nivel nacional e internacional en una base de datos.
- 2) Consultar a través de la Web, las ofertas de paquetes turísticos nacionales e internacionales.
- 3) Registrar usuarios para el sitio proporcionándoles una garantía de autenticación.
- 4) Permitir la reservación de un paquete turístico a través de un formulario en la Web.
- 5) Proporcionar información de carácter general sobre la Agencia de Viajes.
- 6) Proporcionar interacción con los usuarios del sitio a través del correo electrónico.

## **1.5 Justificación del Proyecto**

El sitio web de la agencia permitirá promocionar y administrar mejor los paquetes turísticos que ésta ofrece.

El sitio Web tiene como prioridad, contribuir para que gran parte del mundo conozca los lugares turísticos del Ecuador promocionados en ésta Agencia, limitándose a cumplir con los ideales de la Institución de una manera eficiente.

## 1.6 Presupuesto

### 1.6.1 Costo del Proyecto

<b>HARDWARE</b>	
PC Escritorio	800
<b>SOFTWARE</b>	
Rational Rose	620
Macromedia Dreamweaver	400
Macromedia Fireworks MX 2004	300
MySQL	S/N
PHP	S/N
Servidor XAMPP	S/N
<b>GASTOS VARIOS</b>	300
(Hojas de papel bond, diskettes, cds, tonner impresora, servicios básicos, investigación, transporte)	
<b>TOTAL</b>	2420

### 1.6.2 Costo Estimado del Proyecto según COCOMO BÁSICO

<b>ESTIMACION DEL PROYECTO</b>
--------------------------------

#### COCOMO BASICO

$$VE = (S_{OP} + 4S_{SMP} + S_{PES}) / 6$$

$$E = a_b KLDC^{bb}$$

$$D = c_b E^{db}$$

$$PRODUCTIVIDAD = LDC/E$$

$$COSTE PROYECTO = (TARIFA LABORAL/PRODUCTIVIDAD) * VE$$

#### ADMINISTRADOR

#### CREAR CUENTA

LDC

v. optimista	v. m. probable	v. pesimista	VE
250	308	350	305,333333

**INGRESAR CLAVE**

LDC

v. optimista	v. m. probable	v. pesimista	VE	
20		30	40	30

**CONFIRMAR RESERVACIÓN**

LDC

v. optimista	v. m. probable	v. pesimista	VE	
350		400	450	400

**ACTUALIZAR TOURS**

LDC

v. optimista	v. m. probable	v. pesimista	VE	
300		396	500	397,333333

**AÑADIR TOURS**

LDC

v. optimista	v. m. probable	v. pesimista	VE	
300		380	460	380

**ELIMINAR TOURS**

LDC

v. optimista	v. m. probable	v. pesimista	VE	
200		250	300	250

**CAMBIAR CONTRASEÑA**

LDC

v. optimista	v. m. probable	v. pesimista	VE	
250		300	350	300

**USUARIO****CREAR CUENTA**

LDC

v. optimista	v. m. probable	v. pesimista	VE	
250		300	350	300

**INGRESAR CLAVE**

LDC

v. optimista	v. m. probable	v. pesimista	VE	
20		30	40	30

**INGRESAR RESERVACIÓN**

LDC

v. optimista	v. m. probable	v. pesimista	VE	
700		760	820	760

**LISTAR RESERVACIONES**

LDC

v. optimista	v. m. probable	v. pesimista	VE	
100		150	200	150

**ELIMINAR RESERVACIÓN**

LDC

v. optimista	v. m. probable	v. pesimista	VE
100	150	200	150

<b>TOTAL VE=</b>	<b>3452,66667</b>
<b>E=</b>	<b>8,70072</b>
<b>D=</b>	<b>5,68814923</b>
<b>PRODUCTIVIDAD=</b>	<b>396,825397</b>
<b>TARIFA LABORAL</b>	<b>500</b>
<b>TARIFA LABORAL/PRODUCTIVIDAD</b>	<b>1,26</b>
<b>COSTE DEL PROYECTO</b>	<b>4350,36</b>

## CAPITULO II

### MARCO DE REFERENCIA

#### 2.1 Ingeniería WEB

Pocos pueden discutir que Internet y la World-Wide Web están cambiando nuestras vidas. Cada día es más común que tareas tales como la lectura del periódico, la compra de libros o discos, operaciones bancarias, reserva de hoteles, compra de billetes de avión o tren, entre otras muchas, las realicemos conectados con nuestro ordenador a Internet. Es así que, durante la última década hemos asistido al crecimiento vertiginoso del desarrollo y uso de aplicaciones y sistemas Web cada vez más complejos y sofisticados.

Desafortunadamente, dicha complejidad no parece estar acompañada de los mecanismos adecuados que garanticen la calidad de unos sistemas de los que cada día tenemos mayor dependencia a nivel social, funcional y económico.

Esta carencia de calidad ha venido generando una preocupación creciente entre la comunidad científica y técnica involucrada en el desarrollo Web. Así pues, en los últimos años surgen varias iniciativas con el objetivo de poner cierto orden dentro de la maraña que estamos creando y en la que nos movemos habitualmente.

La nueva disciplina que ha sido establecida durante los años anteriores se ve por consiguiente como:

***Ingeniería Web:*** *Aplicación de procedimientos sistemáticos, disciplinados y cuantificables en busca de evolución hacia una alta calidad y un desarrollo efectivo de aplicaciones y sistemas en la World Wide Web.*

*<http://www.webengineering.org/beta1/wsls.aspx?vdc=beta1&vurl=start/about>*

## 2.2 APLICACIONES WEB

Inicialmente la Web era simplemente una colección de páginas estáticas, documentos, etc., que podían consultarse o descargarse.

El siguiente paso en su evolución fue la inclusión de un método para confeccionar páginas dinámicas que permitiesen que lo mostrado fuese dinámico (generado o calculado a partir de los datos de la petición). Dicho método fue conocido como CGI (common gateway interface) y definía un mecanismo mediante el cual podíamos pasar información entre el servidor HTTP y programas externos. Los CGI siguen siendo muy utilizados, puesto que la mayoría de los servidores Web los soportan debido a su sencillez. Además, nos proporcionan total libertad a la hora de escoger el lenguaje de programación para desarrollarlos.

El esquema de funcionamiento de los CGI tenía un punto débil: cada vez que recibíamos una petición, el servidor Web lanzaba un proceso que ejecutaba el programa CGI.

En ingeniería de software una aplicación Web es aquella que los usuarios usan accediendo a un servidor Web a través de Internet o de una intranet. Las aplicaciones Web son populares debido a la practicidad del navegador Web como cliente ligero. La habilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad. Aplicaciones como los webmails, wikis, weblogs, MMORPGs.

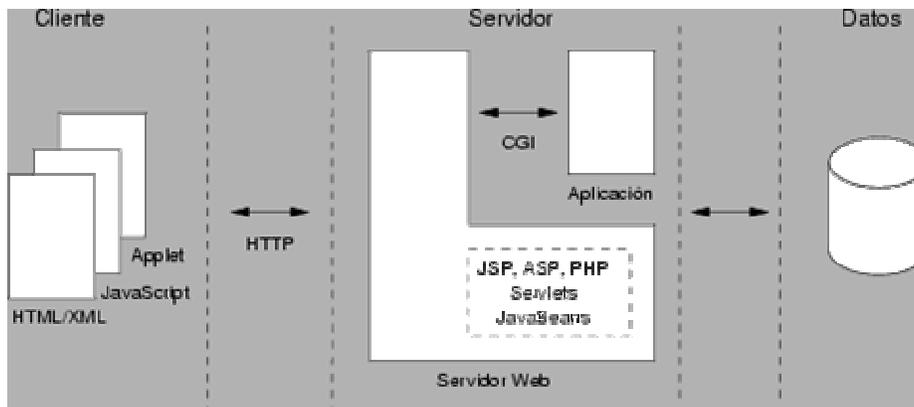
La idea fundamental es que los navegadores, *browsers*, presentan documentos escritos en HTML que han obtenido de un servidor Web. Estos documentos HTML habitualmente presentan información de forma estática, sin más posibilidad de interacción con ellos.

El modo de crear los documentos HTML ha variado a lo largo de la corta vida de las tecnologías Web pasando desde las primeras páginas escritas en HTML almacenadas en un fichero en el servidor Web hasta aquellas que se generan al vuelo como respuesta a una acción del cliente y cuyo contenido varía según las circunstancias.

<http://www.uoc.edu/masters/esp/img/873.pdf>

[http://www.sonork.com/esp/web\\_app.html](http://www.sonork.com/esp/web_app.html)

El esquema general muestra cada tipo de tecnología involucrada en la generación e interacción de documentos Web.



**Figura:** Esquema general de las tecnologías Web.

<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node11.html>

### 2.2.1 Arquitectura Web

La fácil creación de hojas HTML y en general de sitios Web, usando herramientas simples, ha hecho que el desarrollo de este tipo de aplicaciones se haga sin un trabajo serio de análisis y diseño.

Cualquier sistema de complejidad no trivial, necesita ser analizado y modelado. Las aplicaciones Web, al igual que otras aplicaciones, necesitan métodos formales de análisis y diseño.

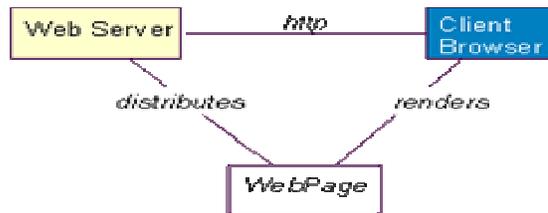
#### **Cuál es la diferencia entre un sitio Web y una aplicación Web?**

Una aplicación Web es un sitio Web donde la navegación a través del sitio, y la entrada de datos por parte de un usuario, afectan el estado de la lógica del

negocio. En esencia, una aplicación Web usa un sitio Web como entrada (front-end) a una aplicación típica.

La arquitectura de un sitio Web tiene tres componentes principales: un servidor Web, una conexión de red, y uno o más clientes (browsers).

El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP.

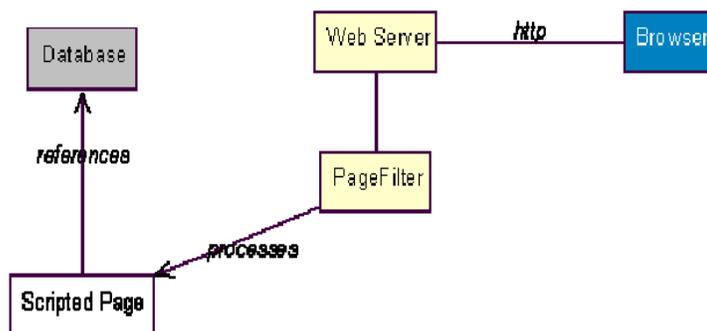


<http://uca.guegue.com/AnalisisyDisenoWeb/web-app.pdf>

### Arquitectura básica de una aplicación / sitio Web

La información mostrada en las páginas está típicamente almacenada en archivos. Sin embargo, muchas veces esta información está almacenada en una base de datos, y las páginas son creadas dinámicamente.

Los sitios Web que usan este esquema, son llamados sitios dinámicos.



*<http://uca.guegue.com/AnalisisyDisenoWeb/web-app.pdf>*

## **Páginas Web**

Las páginas Web son el componente principal de una aplicación o sitio Web. Los browsers piden páginas (almacenadas o creadas dinámicamente) con información a los servidores Web.

En algunos ambientes de desarrollo de aplicaciones Web, las páginas contienen código HTML y scripts dinámicos, que son ejecutados por el servidor antes de entregar la página.

Una vez que se entrega una página, la conexión entre el browser y el servidor Web se rompe (a diferencia de otros esquemas tipo cliente / servidor). Es decir que la lógica del negocio en el servidor solamente se activa por la ejecución de los scripts de las páginas solicitadas por el browser (en el servidor, no en el cliente).

## **Scripts en el cliente**

Cuando el browser ejecuta un script en el cliente, éste no tiene acceso directo a los recursos del servidor.

Hay otros componentes que no son scripts, como los applets o los componentes ActiveX. Los scripts del cliente son por lo general código JavaScript o VBScript, mezclados con código HTML.

## **Formularios**

La forma más común de capturar la información dada por el usuario, es a través de formularios. Un formulario (form) es una colección de campos de entrada: textbox, text area, checkbox, radio button group, button y selection list.

Cuando un formulario es llenado, se envía al servidor usando una operación submit solicitada por el usuario típicamente al hacer click en un botón.

## **Servidor Web**

Un servidor Web es un programa que atiende y responde a las diversas peticiones de los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS (la versión segura, cifrada y autenticada de HTTP).

A partir del esquema anterior se han diseñado y construido todos los programas servidores de HTTP que existen, variando sólo el tipo de peticiones (páginas estáticas, CGI, Servlets, etc.) que pueden atender, en función de que sean o no multi-proceso, multi-hilados, etc.

En muchas aplicaciones Web hay una capa intermedia, compuesta por un conjunto de componentes, que se ejecutan no necesariamente en el servidor Web, sino en otros servidores de aplicaciones. Esta capa encapsula la lógica del negocio, y, al ser componentes compilados puede contener objetos, con sus métodos y atributos (llamados business objects).

*<http://www.uoc.edu/masters/esp/img/873.pdf>*

### **2.2.2 Aplicaciones Multinivel**

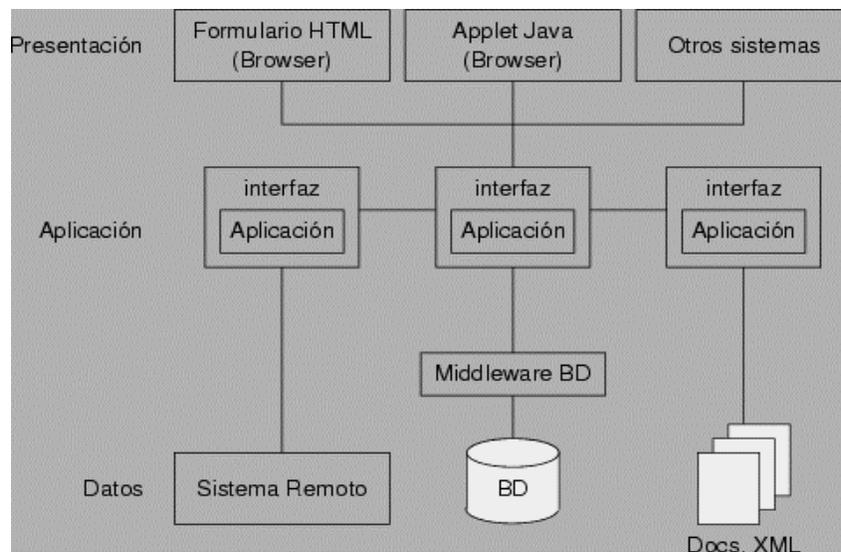
Al hablar del desarrollo de aplicaciones Web resulta adecuado presentarlas dentro de las aplicaciones multinivel. Los sistemas típicos cliente / servidor pertenecen a la categoría de las aplicaciones de dos niveles. La aplicación reside en el cliente mientras que la base de datos se encuentra en el servidor. En este tipo de aplicaciones el peso del cálculo recae en el cliente, mientras que el servidor hace la parte menos pesada, y eso que los clientes suelen ser máquinas menos potentes que los servidores. Además, está el problema de la actualización y el mantenimiento de las aplicaciones, ya que las modificaciones a la misma han de ser trasladada a todos los clientes.

Para solucionar estos problemas se ha desarrollado el concepto de arquitecturas de tres niveles: interfaz de presentación, lógica de la aplicación y los datos.

La capa intermedia es el código que el usuario invoca para recuperar los datos deseados. La capa de presentación recibe los datos y los formatea para

mostrarlos adecuadamente. Esta división entre la capa de presentación y la de la lógica permite una gran flexibilidad a la hora de construir aplicaciones, ya que se pueden tener múltiples interfaces sin cambiar la lógica de la aplicación. La tercera capa consiste en los datos que gestiona la aplicación. Estos datos pueden ser cualquier fuente de información como una base de datos o documentos XML.

Convertir un sistema de tres niveles a otro multinivel es fácil ya que consiste en extender la capa intermedia permitiendo que convivan múltiples aplicaciones en lugar de una sola

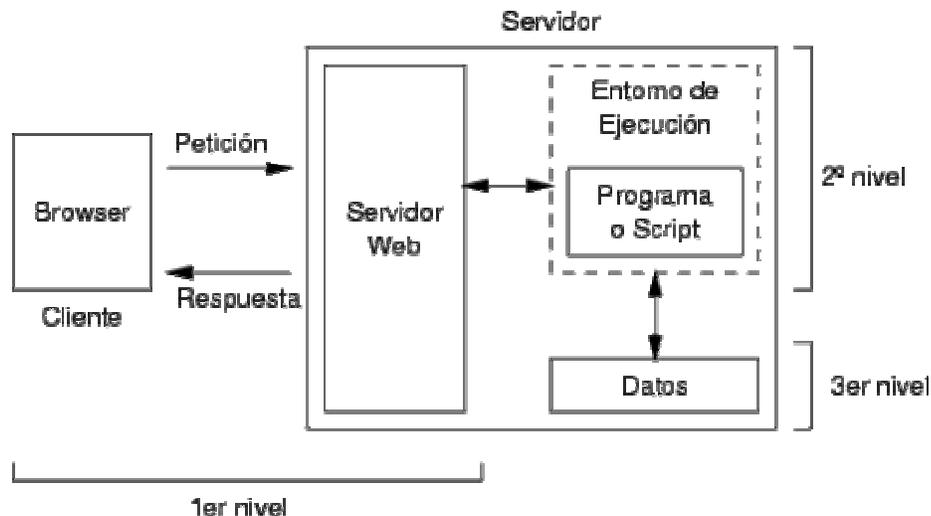


<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node21.html>

La arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles. El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor Web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente a algún tipo de programa o *script*. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo

resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).



<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node21.html>

Lamentablemente, el uso de toda esta tecnología pasa por el dominio de técnicas de programación y de acceso a bases de datos, condición esta que no se puede presuponer en un curso de divulgación como éste. Así, nos vamos a restringir al uso de herramientas básicas a la hora de la construcción de un portal docente.

## 2.3 Herramientas de Soporte al Desarrollo

### Capa Cliente

#### 2.3.1 JavaScript

JavaScript es un lenguaje de programación utilizado para crear programas encargados de realizar acciones dentro del ámbito de una página Web siendo

el siguiente paso, después del HTML, que puede dar un programador de la Web que decida mejorar sus páginas y la potencia de sus proyectos.

JavaScript se inicia en el año de 1995 cuando Netscape introduce la versión 2.0 de Navigator e incluye JavaScript bajo el nombre de Mocha, cuando aparece esta versión de Navigator se le llamaba LiveScript.

Finalmente se le bautiza con el nombre de JavaScript para llamar la atención de los medios y la industria de la informática.

Lo que quería Netscape es que JavaScript fuera un lenguaje de guiones, fácil de usar y que cualquier persona pudiera utilizarlo. Después de 2 años JavaScript se convirtió en una de las herramientas más utilizadas por los desarrolladores Web, incluso se utiliza más que el propio Java y ActiveX.

En el año de 1996 Microsoft empieza a tener gran interés por lograr competir con JavaScript por lo que lanza su lenguaje llamado Jscript, pero que desafortunadamente no ha tenido el gran éxito por lo que no ha podido competir directamente con este lenguaje.

A mediados de 1997, Netscape promociona JavaScript y lanza la versión 1.2 de este lenguaje. Esta nueva versión incluye nuevos componentes que dan gran potencial al lenguaje, pero lamentablemente esta versión solo funcionaba en la última versión del Navigator en ese momento. En la actualizada esta versión de JavaScript es soportada en la mayoría de los navegadores y es utilizado en casi todos los sitios de internet existentes.

<http://www.ortizmania.com/online/articulo.asp?art=6>

### **2.3.2 XHTML**

El XHTML (*Extensible Hypertext Markup Language*) o Lenguaje de Etiquetado Hipertextual es una familia de módulos y tipos de documentos que reproduce, engloba y extiende HTML 4.0.

Los tipos de documentos de la familia XHTML están basados en XML y diseñados fundamentalmente para trabajar en conjunto con aplicaciones de usuario basados en XML; y puede ser extendido por los propios desarrolladores.

El lenguaje XHTML surgió ante los problemas de compatibilidad que surgían cuando se usaba un documento HTML en distintas plataformas.

Utilizar XHTML en lugar de HTML presenta las siguientes mejoras:

- Los documentos XHTML son conformes a XML. Como tales, son fácilmente visualizados, editados y validados con herramientas XML estándar.
- Los documentos XHTML pueden escribirse para que funcionen igual o mejor que lo hacían antes tanto en las aplicaciones de usuario conformes a HTML 4.0 como en las nuevas aplicaciones conformes a XHTML 1.0.
- Los documentos XHTML pueden usar aplicaciones (por ejemplo scripts y applets) que se basen ya sea en el Modelo del Objeto Documento de HTML o XML

### **Variantes de XHTML**

Transitoria.

Estricta.

Con marcos (frames).

La variante transitoria se utiliza en caso de que haya que usar características de presentación de XHTML (sólo si no se pueden utilizar CSS). La variante estricta no permite marcas de presentación, es la que se utiliza con hojas de estilo. La variante con marcos se emplea cuando se utilizan frames en el navegador

### **¿En qué se diferencia de HTML?**

Las diferencias no son excesivas:

- No es necesario utilizar un único DTD.

- Se pueden crear versiones reducidas o extendidas del lenguaje.
- Obliga a seguir reglas más estrictas al construir un documento.
- Todas las etiquetas y atributos deben ir en minúsculas:

NO: <A HREF=...> SI: <a href=...>

Todos los valores de los atributos deben ir entre comillas:

NO: <table border=0> SI: <table border="0">

Todas las etiquetas no vacías deben tener etiqueta de cierre:

NO: <p>en un lugar... SI: <p>en un lugar...</p>

Todas las etiquetas vacías deben indicarse "a la XML":

NO:  SI: 

<http://www.hipertexto.info/documentos/xhtmll.htm>

<http://petra.euitio.uniovi.es/~labra/cursos/ext02/xhtmllWml.PDF>

### **2.3.3 CSS**

Hojas de Estilo en Cascada (Cascading Style Sheets), es un lenguaje que permite a los autores y usuarios enlazar estilos (ej. Fonts, spacing, etc.) a documentos estructurados (ej. Documentos HTML y aplicaciones XML). Mediante la separación de estilo de los documentos de su contenido CSS simplifica el desarrollo Web y el mantenimiento del sitio.

Las hojas de estilo describen cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los *Estilos* definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne:

```
h1 {color: red;}
```

h1 es el selector

{color: red;} es la declaración

El selector funciona como enlace entre el documento y el estilo, especificando los elementos que se van a ver afectados por esa declaración. La declaración es la parte de la regla que establece cuál será el efecto. En el ejemplo anterior, el selector h1 indica que todos los elementos h1 se verán afectados por la declaración donde se establece que la propiedad color va a tener el valor red (rojo) para todos los elementos h1 del documento o documentos que estén vinculados a esa hoja de estilos.

Las tres formas más conocidas de dar estilo a un documento son las siguientes:

Utilizando una hoja de estilo externa que estará vinculada a un documento a través del elemento <link>, el cual debe ir situado en la sección <head>.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN">
```

```
<html>
```

```
<head>
```

```
<title>Título</title>
```

```
<link rel="stylesheet" type="text/css"
```

```
href="http://www.w3.org/css/officeFloats.css" />
```

```
</head>
```

```
<body>
```

```
.  
.   
.   
.
```

```
</body>  
</html>
```

Utilizando el elemento <style>, en el interior del documento al que se le quiere dar estilo, y que generalmente se situaría en la sección <head>. De esta forma los estilos serán reconocidos antes de que la página se cargue por completo.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN">  
<html>  
  <head>  
    <title>hoja de estilo interna</title>  
    <style type="text/css">  
  
      body {  
        padding-left: 11em;  
        font-family: Georgia, "Times New Roman", serif;  
        color: red;  
        background-color: #d8da3d;  
      }  
  
      h1 {  
        font-family: Helvetica, Geneva, Arial, sans-serif;  
      }  
  
    </style>  
  </head>  
  <body>  
    <h1>Aquí se aplicará el estilo de letra para el Título</h1>  
  </body>  
</html>
```

Utilizando estilos directamente sobre aquellos elementos que lo permiten a través del atributo <style> dentro de <body>. Pero este tipo de estilo pierde las

ventajas que ofrecen las hojas de estilo al mezclarse el contenido con la presentación.

Algunas normas básicas a la hora de crear una CSS son las siguientes:

En el siguiente ejemplo, `h1{color: red;}`, el *selector*, `<h1>`, le dice al navegador la parte del documento que se verá afectada por esa regla. Los selectores pueden aparecer individualmente o agrupados, separándolos con comas:

```
h1, h2, h3 {  
  color: red;  
}
```

o lo que es lo mismo

```
h1 {color: red;}  
h2 {color: red;}  
h3 {color: red;}
```

La *propiedad*, que en este caso sería `color`, especifica qué aspecto se va a cambiar. En este ejemplo la propiedad cambiada será el color. Las propiedades que se desean modificar en una CSS para un mismo selector pueden agruparse, pero será necesario separar cada una de ellas con un punto y coma.

```
p {text-align:center;color:red}
```

Normalmente se describe una propiedad por línea, de la siguiente manera:

```
h1 {  
  padding-left: 11em;  
  font-family: Georgia, "Times New Roman", Times, serif;  
  color: red;  
  background-color: #d8da3d;  
}
```

El *valor*, en este caso red, establece el valor de la propiedad. Es importante recordar que si el valor está formado por más de una palabra, hay que ponerlo entre comillas.

p {font-family: "sans serif";

*<http://www.w3c.es/Divulgacion/Guiasbreves/HojasEstilo>*

## **Capa de Servidor Web**

### **2.3.4 Apache Web Server**

Este servidor cuyo mantenimiento está a cargo de Apache Software Foundation, es el servidor Web mas popular debido a su estabilidad, costo, eficiencia y portabilidad. Es un producto de código fuente abierta que se ejecuta bajo las plataformas Unix, Linux y Windows.

El nombre apache tiene un origen un poco discutido, algunos dicen que viene de "**a patchy** Server" debido a numerosos patches, otros dicen de una manera más seria que los instigadores de este proyecto tomaron el nombre en memoria de los Apaches por su gran adaptabilidad al terreno.

Según los estudios de netcraft de mayo de 2003, apache es utilizado en el 62.57 % de los servidores, seguidos por IIS con un 27.45%.

*<http://es.tldp.org/LinuxFocus/pub/mirror/LinuxFocus/Castellano/May2000/article122.shtml>*

### **2.3.5 PHP 5**

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos Web y el cual puede ser embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este

lenguaje es permitir escribir a los creadores de páginas Web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP.

PHP es un lenguaje creado por una gran comunidad de personas. El sistema fue desarrollado originalmente en el año 1994 por Rasmus Lerdorf como un CGI escrito en C que permitía la interpretación de un número limitado de comandos.

El sistema fue denominado Personal Home Page Tools y adquirió relativo éxito gracias a que otras personas pidieron a Rasmus que les permitiese utilizar sus programas en sus propias páginas.

Dada la aceptación del primer PHP y de manera adicional, su creador diseñó un sistema para procesar formularios al que le atribuyó el nombre de FI (Form Interpreter) y el conjunto de estas dos herramientas, sería la primera versión compacta del lenguaje: PHP/FI.

La siguiente gran contribución al lenguaje se realizó a mediados del 97 cuando se volvió a programar el analizador sintáctico, se incluyeron nuevas funcionalidades como el soporte a nuevos protocolos de Internet y el soporte a la gran mayoría de las bases de datos comerciales. Todas estas mejoras sentaron las bases de PHP.

Actualmente PHP se encuentra en su versión 5, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades actuales y solucionar algunos inconvenientes de la anterior versión. Algunas mejoras de esta nueva versión son su rapidez gracias a que primero se compila y luego se ejecuta, mientras que antes se ejecutaba mientras se interpretaba el código, su mayor independencia del servidor Web -creando versiones de PHP nativas para más plataformas y un API más elaborado y con más funciones.

*<http://www.desarrolloweb.com/articulos/436.php?manual=12>*

Existen tres campos en los que se usan scripts escritos en PHP:

- Scripts del lado del servidor. Este es el campo más tradicional y el principal foco de trabajo. Se necesitan tres cosas para que esto

funcione. El intérprete PHP (CGI ó módulo), un servidor Web y un navegador. Es necesario correr el servidor Web con PHP instalado. El resultado del programa PHP se puede obtener a través del navegador, conectándose con el servidor Web.

- Scripts en la línea de comandos. Puede crear un script PHP y correrlo sin ningún servidor Web o navegador. Solamente necesita el intérprete PHP para usarlo de esta manera. Este tipo de uso es ideal para scripts ejecutados regularmente desde cron (en \*nix o Linux) o el Planificador de tareas (en Windows). Estos scripts también pueden ser usados para tareas simples de procesamiento de texto.
- Escribir aplicaciones de interfaz gráfica. Probablemente PHP no sea el lenguaje más apropiado para escribir aplicaciones gráficas, pero si conoce bien PHP, y quisiera utilizar algunas características avanzadas en programas clientes, puede utilizar PHP-GTK para escribir dichos programas. También es posible escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP, no disponible en la distribución principal.

## **Capa de base de datos:**

### **2.3.6 XAMPP**

Es un paquete de desarrollo en php para Windows y Linux. Por defecto este paquete viene preparado para trabajar con mysql.

Una de las formas más fáciles y rápidas de tener Apache, MySQL, PHP y phpMyAdmin en su máquina es, sin lugar a duda, pasando por un paquete de instalación cómodo y automático como XAMPP. Ofrece un pack de instalación automática con lo que podrás alojar y servir tus páginas Web desde tu máquina en local.

Además, XAMPP ofrece una colección de librerías y otras aplicaciones de gran utilidad para el manejo y administración de una página Web, junto a todas las dependencias que resultan imprescindibles para ello.

Un servidor Web, una base de datos MySQL, PHP, un servidor de correo electrónico, Perl y un servidor FTP son los elementos claves de esta distribución. Además incluye Apache 2 y las últimas versiones de MySQL y

PHP; Apache y MySQL se instalarán como servicios. Y todo esto gracias a un asistente que automatizará todo el proceso para que sea lo más leve y rápido posible

*<http://www.codigophp.com/index.php/page/2/>*

### **2.3.7 MySQL**

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

**MySQL es un sistema de administración de bases de datos.** Una base de datos es una colección estructurada de datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto monto de información en un red corporativa. Para agregar, acceder y procesar datos guardados en un computador, se necesita un administrador como MySQL Server.

**MySQL es un sistema de administración relacional de bases de datos.** Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

**MySQL es software de fuente abierta.** Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. MySQL usa el GPL (GNU General Public License) para definir que puede hacer y que no puede hacer con el software en diferentes situaciones

MySQL Database Server es muy rápido, confiable y fácil de usar. MySQL Server fue desarrollado inicialmente para manejar grandes bases de datos mucho más rápidamente que las soluciones existentes y ha sido usado exitosamente por muchos años en ambientes de producción de alta demanda. A través de constante desarrollo, MySQL Server ofrece hoy una rica variedad

de funciones. Su conectividad, velocidad y seguridad hacen a MySQL altamente satisfactorio para acceder bases de datos en Internet.

### **Características de MySQL**

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

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

*[http://www.netpecos.org/docs/mysql\\_postgres/x57.html](http://www.netpecos.org/docs/mysql_postgres/x57.html)*

*<http://www.software-shop.com/Productos/MySQL/mysql.html>*

## **2.4 HERRAMIENTAS DE APOYO**

### **2.4.1 Dreamweaver MX 2004**

Dreamweaver es una herramienta orientada tanto al diseño como al desarrollo Web, que brinda un amplio soporte a las tecnologías más extendidas, y actualmente abarca el 80% del mercado de edición de páginas Web.

Con Dreamweaver se puede controlar todo el proceso de creación de un sitio Web: creación del sitio, estructurar el árbol de links, diseñar las páginas y la subida de nuestro trabajo a nuestro servidor Web.

También ofrece un entorno de codificación con todas las funciones. Incluye herramientas para la edición de código (tales como coloreado de código y terminación automática de etiquetas) y material de referencia sobre HTML,

hojas de estilos en cascada (CSS), JavaScript, ColdFusion Markup Language (CFML), Microsoft Active Server Pages (ASP) y JavaServer Pages (JSP).

Dreamweaver le permite crear aplicaciones Web dinámicas basadas en bases de datos empleando tecnologías de servidor como CFML, ASP, NET, JSP y PHP.

Dreamweaver se puede personalizar totalmente. Puede crear sus propios objetos y comandos, modificar métodos abreviados de teclado e incluso escribir código JavaScript para ampliar las posibilidades que ofrece Dreamweaver con nuevos comportamientos, inspectores de propiedades e informes de sitios.

### **Características principales de Dreamweaver MX 2004:**

- Administrador de proyectos
- Trabajo con capas (layers)
- Amplio soporte para la tecnología CSS
- Integración con otras aplicaciones, como Fireworks y Flash
- Página de inicio para un mejor acceso a documentos, archivos y recursos
- Conexión FTP integrada en la aplicación
- Comportamientos predefinidos para añadir funcionalidad de Javascript
- Barra de inserción con acceso inmediato a los elementos principales
- Arquitectura extensible para ampliar la funcionalidad de la aplicación
- Alto nivel de personalización del espacio de trabajo.

*<http://www.formaselect.com/curso/Macromedia-Dreamweaver-mx-2004/presentacion.htm>*

*[http://cevug.ugr.es/material/curso\\_DreamWeaver04.pd](http://cevug.ugr.es/material/curso_DreamWeaver04.pd)*

### **2.4.2 MACROMEDIA FIREWORKS**

El programa Fireworks MX, software que pertenece a la casa Macromedia, sirve para realizar aplicaciones gráficas de todo tipo, que potencien y hagan más atractivas nuestras páginas Web. MX identifica a la última versión de este producto, y si no tienes una copia, puedes descargar una versión de prueba que dura sólo 30 días, pero que te servirá para entender las nociones básicas del software.

Fireworks, a diferencia de otros productos gráficos como Photoshop, tiene la particularidad de trabajar con vectores y mapas de bit, combinación que lo hace indicado para realizar diseños Web. Esto se potencia gracias a que trae incorporado un sinnúmero de opciones que nos permiten crear aplicaciones bastantes interesantes, como gifs animados, banners y menús desplegados, entre otras imágenes.

*[http://www.macromedia.com/go/thirdparty\\_es](http://www.macromedia.com/go/thirdparty_es)*

### **2.4.3 PowerDesigner**

PowerDesigner, un acercamiento que modela de proceso del negocio para alinear negocio y IT, es datos de la empresa que modelan y la solución del diseño de base de datos que ayuda a arquitectura eficaz de la empresa del instrumento y trae un modelo conceptual de gran alcance de los datos a tu ciclo vital del desarrollo del uso.

PowerDesigner combina únicamente varios datos estándares que modelan las técnicas (UML, el modelar del proceso del negocio y los datos mercado-que conducen que modelan) junto con ambientes principales del desarrollo tales como .NET, espacio de trabajo, PowerBuilder, Java, eclipse, etc., para traer análisis de negocio y soluciones formales del diseño de base de datos al ciclo de vida tradicional del desarrollo del software. Y trabaja con todo el RDBMS moderno.

PowerDesigner es una suite de aplicaciones de Powersoft para la construcción, diseño y modelado de datos a través de diversas aplicaciones.

Es la herramienta para el análisis, diseño inteligente y construcción sólida de una base de datos y un desarrollo orientado a modelos de datos a nivel físico y conceptual, que dan a los desarrolladores Cliente / servidor la más firme base para aplicaciones de alto rendimiento.

Esta suite cuenta con los siguientes productos: bases de datos en plataformas de sistemas analíticos usando modelados dimensionales, esquemas de "estrella" y "nieve", particionamiento y agregación.

También cuenta con un alto desempeño en el indexamiento de esquemas.

*[http://www.pcm.gob.pe/portal\\_ongei/publicaciones/cultura/lib5103/Libro.pdf](http://www.pcm.gob.pe/portal_ongei/publicaciones/cultura/lib5103/Libro.pdf)*

### **2.4.5 Rational Rose**

Rational Rose es la mejor solución de modelado visual en el mundo, y la mejor herramienta para traducir requisitos de alto nivel a una arquitectura flexible basada en componentes.

Rational Rose se encuentra a la cabeza en cuanto al desarrollo del Unified Modeling Language (UML), que se ha convertido en la notación estandarizada empleada en Rational Rose para especificar, visualizar y construir desarrollos de software y sistemas.

#### **Características**

- Mantiene la consistencia de los modelos del sistema software
- Chequeo de la sintaxis UML
- Generación documentación automáticamente
- Generación de código a partir de los Modelos
- Ingeniería inversa(crear modelo a partir código)

#### **El desarrollo de Software abarca las disciplinas de:**

- Modelado de casos de uso y datos
- Modelado de arquitectura

- Modelado de componentes,
- Construcción de claves y;
- Pruebas de unidades.

*[http://www.abits.cl/rational/desa\\_soft.htm](http://www.abits.cl/rational/desa_soft.htm)*

## **CAPITULO III**

### **ASPECTOS METODOLÓGICOS**

#### **3.1 PARADIGMA ESPIRAL ORIENTADO A LA WEB**

Es un modelo de proceso de software evolutivo. En el modelo espiral, el software se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones. La versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez mas completas de ingeniería del sistema.

##### **Características**

- Incluye la etapa de análisis de riesgos, no incluida anteriormente.
- Es ideal para crear productos con diferentes versiones mejoradas como se hace con los software modernos de microcomputadoras.
- La ingeniería puede desarrollarse a través del ciclo de vida clásico o el de construcción de prototipos.

##### **Ventajas**

- El modelado en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora, no terminal cuando se entrega el software.
- Como el software evoluciona, a medida que progresa el proceso, el desarrollador y el cliente comprenden y reaccionan mejor ante riesgos en cada uno de los niveles evolutivos.
- Permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.
- Demanda una consideración directa de los riesgos técnicos en todas las etapas del proyecto.
- Reduce los riesgos antes de que se conviertan en problemáticos.

El modelo espiral orientado a la Web, define seis actividades principales:

**Comunicación con el cliente:** establecer comunicación entre el desarrollador Web y el cliente.

**Planificación:** actividad para definir recursos, tiempo y otras informaciones relacionadas con el proyecto. Son todos los requerimientos.

**Análisis de riesgos:** evaluar riesgos técnicos y otras informaciones relacionadas con el proyecto.

**Ingeniería:** construir una o más representaciones de la aplicación.

**Construcción y adaptación:** construir, probar, instalar y proporcionar soporte al usuario.

**Evaluación el cliente:** las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementación durante la etapa de instalación.

*<http://148.202.148.5/cursos/cc321/fundamentos/unidad1/espiral.htm>*

*<http://www.itlp.edu.mx/publica/tutoriales/analisis/24.htm>*

### **3.2 OOHDM: “Object Oriented Hypermedia Design Method”**

La Metodología de Diseño Hipermedia orientada a objetos, propuesta por D. Schwabe y G. Rossi, establece que el desarrollo de un Hiperdocumento es un proceso de cuatro fases en el que se combinan diferentes estilos de desarrollo como el incremental, iterativo y prototipado. Las tres primeras fases son de diseño, en las que se obtiene un conjunto de modelos orientados a objeto que describen el documento que será construido en la última fase.

### **3.2.1 Diseño Conceptual o Análisis de Dominio**

Se realiza, el modelado del dominio del hiperdocumento utilizando algún método análisis orientado a objetos de Sistemas de Información, por ejemplo OMT , obteniendo un esquema conceptual de clases en el que, además de clases abstractas y objetos, se representan las relaciones entre ellas, incluidas las de herencia y agregación, y los correspondientes atributos (que pueden ser de cualquier tipo, desde simples cadenas de caracteres a gráficos, imágenes, texto, sonido, etc.) y métodos asociados a las clases.

### **3.2.2 Diseño Navegacional**

En la que se ha de definir la estructura de navegación a través del hiperdocumento mediante la realización de modelos navegacionales que representen diferentes vistas del esquema conceptual de la fase anterior. Se trata, en definitiva, de reorganizar la información para adaptarla a las necesidades de los usuarios del sistema. El Diseño Navegacional se expresa, también con un enfoque orientado a objetos, a través de dos tipos de esquemas o modelos: el denominado esquema de clases navegacionales, con las posibles vistas del hiperdocumento a través de unos tipos predefinidos de clases, llamadas navegacionales, como son los "nodos", los "enlaces", y otras clases que representan estructuras o formas alternativas de acceso a los nodos, como los "Indices" y los "recorridos guiados"; y el esquema de contexto navegacional, que permite la estructuración del hiperespacio de navegación en subespacios para los que se indica la información que será mostrada al usuario y los enlaces que estarán disponibles cuando se acceda a un objeto (nodo) en un contexto determinado.

### **3.2.3 Diseño de la Interface Abstracta**

En la que se realiza un modelo, también orientado a objetos, para especificar la estructura y el comportamiento de la interfase del sistema hipermedia con el usuario. Este modelo es abstracto y, por tanto, independiente de la implementación final del sistema. Sin embargo, se basa en las ideas que actualmente se aplican en las Interfaces Gráficas de Usuario (IGUs), por lo que

como la mayor parte de entornos hipermedia comerciales trabajan con IGUs, su implantación en un entorno de este tipo debe ser una tarea sencilla.

El modelo de la interface abstracta se expresa a través de tres tipos de diagramas

que se complementan entre sí. En primer lugar se deben crear los denominados diagramas de Vistas de Datos Abstractos (ADVs) que incluyen una vista (ADV) por cada clase navegacional (nodo, enlace o estructura de acceso) que fue establecida durante la fase de Diseño Navegacional. Un diagrama de este tipo se compone de una serie de cajas (una caja es un ADV) que representan las diferentes clases de objetos que aparecerán ante el usuario. Un segundo tipo de diagramas que componen el modelo de interface es el Diagrama de Configuración, donde se representan principalmente los eventos externos (provocados por el usuario, como ClicDeRatón o DobleClic) que maneja un ADV, los servicios que ofrece el ADV (como "visualización") y las relaciones estáticas entre las Hadas. Por último, el modelo se completa con los denominados Diagramas de Estado que, como ocurre en los modelos orientados a objetos de ingeniería del software [13, representan el comportamiento dinámico del sistema hipermedial mediante el establecimiento de un diagrama de transición de estados para cada ADVs, en el que se reflejan los posibles estados por los que puede pasar cada objeto de la interface (oculto, desactivado, ampliado, reducido, normal, etc.) y los eventos que originan los cambios de estado.

### **3.2.4 Implementación**

Concreción de los modelos navegacionales y de interface en objetos particulares con sus correspondientes contenidos (textuales, visuales sonoros, etc.) y sus posibilidades de navegación. Aunque, al utilizar un enfoque de orientación a objetos podría parecer conveniente que la implementación se hiciera en un entorno de construcción de hiperdocumentos también orientado a objetos, como MacWeb, debido al carácter abstracto del diseño, ésta puede hacerse fácilmente en otros entornos hipermediales como Hypercard, DIRECTOR o cualquiera que permita trabajar con el lenguaje HTML, como HTML Write o HTML Assistant.

*<http://www.ati.es/gt/LATIGOO/OOp96/Ponen6/atiao6p06.html>*

### **3.3 UML**

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.

UML 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.

UML no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML 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. UML es también un lenguaje de modelamiento visual que permite una abstracción del sistema y sus componentes.

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

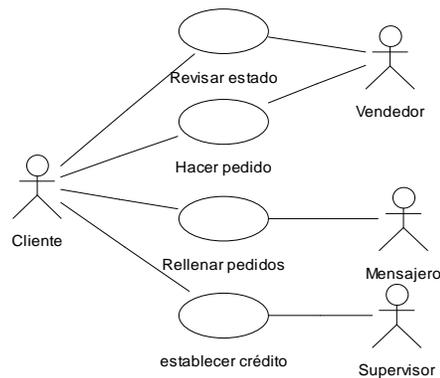
#### **3.3.1 Diagrama de Casos de Uso**

Los casos de uso modelan el sistema desde el punto de vista del usuario. Creados durante la obtención de requisitos, los casos usos deben cumplir los siguientes objetivos.

Definir los requisitos funcionales y operativos del sistema, diseñando un escenario de uso acordado por el usuario final, y el equipo de desarrollo; proporcionar una descripción clara y sin ambigüedad de cómo el usuario final interactúa con el sistema y viceversa, y proporcionar una base para la validación de las pruebas.

Utilizando UML se puede crear una presentación visual de los casos de uso llamada diagrama de casos de usos. Como otros elementos del análisis, los casos de uso pueden representarse a diferentes niveles de abstracción.

Los diagramas de casos de uso contienen casos de uso y actores, siendo estos últimos las entidades que interactúan con el sistema. Pueden ser humanos u otras máquinas o sistemas que tengan definidas interfaces con nuestro sistema. (Pressman, 367).



El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

Un diagrama de casos de uso consta de los siguientes elementos:

- Actor.
- Casos de Uso.
- Relaciones de Uso, Herencia y Comunicación.

### 3.3.2 Diagrama de Clases

Clase
- atributo1 : int
- atributo2 : cadena
+ operacion1(p : bool) : double
# operacion2()

El Diagrama de Clases es una descripción de las clases en un sistema y sus relaciones. No describe el comportamiento dinámico del sistema el primer elemento de un diagrama de clases es una descripción de un diagrama de clases individual. (Pressman, 393)

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).

**Atributos** en UML, los atributos se muestran al menos con su nombre, y también pueden mostrar su tipo, valor inicial y otras propiedades. Los atributos también pueden ser mostrados visualmente:

+ Indica atributos públicos

# Indica atributos protegidos

- Indica atributos privados

**Operaciones.**- las operaciones (métodos) también se muestran al menos con su nombre, y pueden mostrar sus parámetros y valores de retorno.

Las operaciones, al igual que los atributos, se pueden mostrar visualmente:

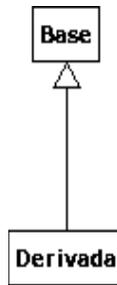
+ Indica operaciones públicas

# Indica operaciones protegidas

- Indica operaciones privadas

## **Herencia**

Es uno de los conceptos fundamentales de la programación orientada a objetos, en la que una clase “recoge” todos los atributos y operaciones de la clase de la que es heredera, y puede alterar/modificar algunos de ellos, así como añadir más atributos y operaciones propias.

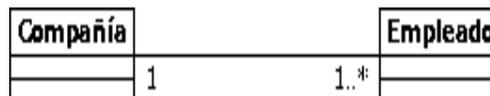


En UML, una asociación de *generalización* entre dos clases, coloca a estas en una jerarquía que representa el concepto de herencia de una clase derivada de la clase base.

### Asociaciones

Una asociación representa una relación entre clases, y aporta la semántica común y la estructura de muchos tipos de “conexiones” entre objetos.

Las asociaciones son los mecanismos que permite a los objetos comunicarse entre sí. Describe la conexión entre diferentes clases (la conexión entre los objetos reales se denomina conexión de objetos o *enlace*).



En UML, las asociaciones se representan por medio de líneas que conectan las clases participantes en la relación, y también pueden mostrar el papel y la multiplicidad de cada uno de los participantes. La multiplicidad se muestra como un rango [mín...máx] de valores no negativos, con un asterisco (\*) representando el infinito en el lado máximo.

### Acumulación

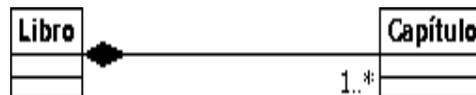
Las acumulaciones son tipos especiales de asociaciones en las que las dos clases participantes no tienen un estado igual, pero constituyen una relación “completa”. Una acumulación describe cómo se compone la clase que asume el rol completo de otras clases que se encargan de las partes.



En UML, las acumulaciones están representadas por una asociación que muestra un rombo en uno de los lados de la clase completa.

### **Composición**

Las composiciones son asociaciones que representan acumulaciones muy fuertes. Esto significa que las composiciones también forman relaciones completas, pero dichas relaciones son tan fuertes que las partes no pueden existir por sí mismas. Únicamente existen como parte del conjunto, y si este es destruido las partes también lo son.



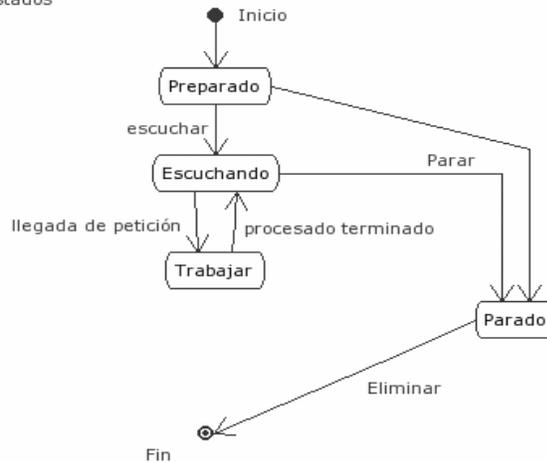
En UML, las composiciones están representadas por un rombo sólido al lado del conjunto.

### **3.3.3 Diagramas de Comportamiento**

#### **3.3.3.1 Diagrama de Estados**

Los diagramas de estado muestran los diferentes estados de un objeto durante su vida, y los estímulos que provocan los cambios de estado en un objeto.

Servidor de estados



<http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>

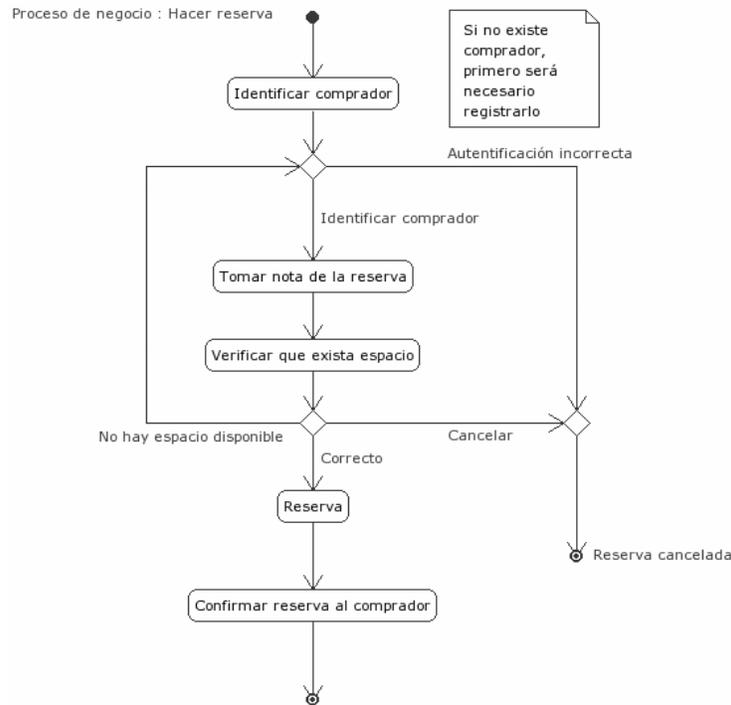
Los diagramas de estado ven a los objetos como *máquinas de estado* o autómatas finitos que pueden estar en un conjunto de estados finitos y que pueden cambiar su estado a través de un estímulo perteneciente a un conjunto finito.

**Estado.-** los estados son los ladrillos de los diagramas de estado. Un estado pertenece a exactamente una clase y representa un resumen de los valores y atributos que puede tener la clase. Un estado UML describe el estado interno de un objeto de una clase particular

Hay dos tipos especiales de estados: inicio y fin. Son especiales en el sentido de que no hay ningún evento que pueda devolver a un objeto a su estado de inicio, y de la misma forma no hay ningún evento que pueda sacar a un objeto de su estado de fin.

### 3.3.3.2 Diagrama de Actividad

Los diagramas de actividad describen la secuencia de las actividades en un sistema. Los diagramas de actividad son una forma especial de los diagramas de estado, que únicamente (o mayormente) contienen actividades



<http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>

Los diagramas de actividad son similares a los diagramas de flujo procesales, con la diferencia de que todas las actividades están claramente unidas a objetos.

Los diagramas de actividad siempre están asociados a una *clase*, a una *operación* o a un *caso de uso*.

**Actividad.-** es un único paso de un proceso. Una activa es un estado del sistema que actividad interna y, al menos, una transición saliente. Las actividades también pueden tener más de una transición saliente, si tienen diferentes condiciones.

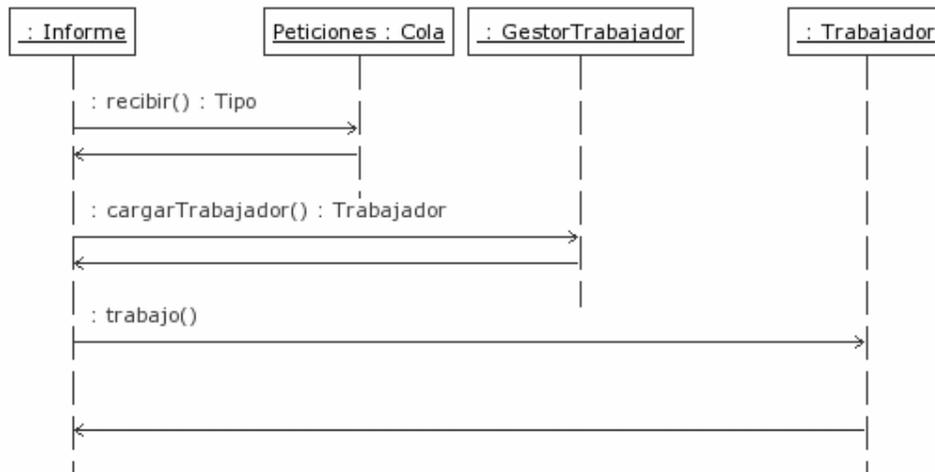
### 3.3.4 Diagramas de Interacción

#### 3.3.4.1 Diagrama de Secuencia

Los diagramas de secuencia muestran el intercambio de mensajes (es decir la forma en que se invocan) en un momento dado.

Los diagramas de secuencia ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos.

En los diagramas de secuencia, los objetos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta. El eje de tiempo también es vertical, incrementándose hacia abajo, de forma que los mensajes son enviados de un objeto a otro en forma de flechas con los nombres de la operación y los parámetros.



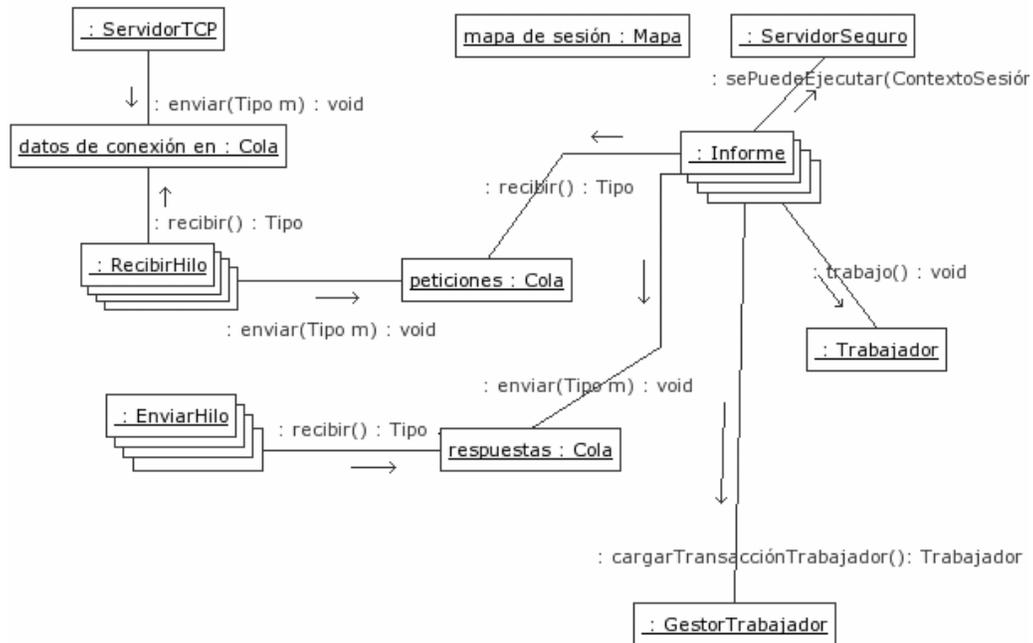
<http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>

Los mensajes pueden ser o bien síncronos, el tipo normal de llamada del mensaje donde se pasa el control a objeto llamado hasta que el método finalice, o asíncronos donde se devuelve el control directamente al objeto que realiza la llamada. Los mensajes síncronos tienen una caja vertical en un lateral del objeto invocante que muestra el flujo del control del programa.

### 3.3.4.2 Diagrama de Colaboración

Los diagramas de colaboración muestran las interacciones que ocurren entre los objetos que participan en una situación determinada. Esta es más o menos la misma información que la mostrada por los diagramas de secuencia, pero destacando la forma en que las operaciones se producen en el tiempo, mientras que los diagramas de colaboración fijan el interés en las relaciones entre los objetos y su topología.

En los diagramas de colaboración los mensajes enviados de un objeto a otro se representan mediante flechas, mostrando el nombre del mensaje, los parámetros y la secuencia del mensaje.



<http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>

Los diagramas de colaboración están indicados para mostrar una situación o flujo programa específicos y son unos de los mejores tipos de diagramas para demostrar o explicar rápidamente un proceso dentro de la lógica del programa.

## **CAPITULO IV**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **CONCLUSIONES**

- XAMPP es un software muy potente, que nos ayudo a instalar de manera conjunta MYSQL, PHP y PHPMyAdmin; además su instalación es bastante sencilla y con una Interfaz bastante amigable.
- Trabajar con Hojas de Estilo resulta muy interesante, ya que trabajando en una sola hoja de código podemos controlar; letras, gráficos, tablas, etc. Así como sus tamaño, color, y sobretodo su ubicación. De todas las páginas sin ser necesario hacerlo en cada una.
- XHTML es un lenguaje que nos permite crear el contenido de las paginas Web (información) sin tener que dar formato a dicho contenido, principal diferencia con respecto a HTML.

## RECOMENDACIONES

- El pago de Tours se los podría hacer vía Web, y de esta manera el Usuario realizaría toda la operación de Reservación sin tener que ir a la Agencia personalmente. Este módulo no se lo desarrolló debido a que el cliente no aceptó en las sugerencias que hicimos.
- Otro módulo que se lo podría implementar es la reservación de pasajes aéreos, ya que la Agencia también presta este servicio.

## **BIBLIOGRAFÍA**

### **LIBROS**

*Pressman, Roger S, Ingeniería de Software, Quinta Edición, Año 1998*

### **INTERNET**

1. <http://es.tldp.org/LinuxFocus/pub/mirror/LinuxFocus/Castellano/May2000/article122.shtml>
2. [http://www.pcm.gob.pe/portal\\_ongei/publicaciones/cultura/lib5103/Libro.pdf](http://www.pcm.gob.pe/portal_ongei/publicaciones/cultura/lib5103/Libro.pdf)
3. <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node21.html>
4. <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node11.html>
5. <http://148.202.148.5/cursos/cc321/fundamentos/unidad1/esprial.htm>
6. <http://www.informandote.com/jornadasIngWEB/articulos/jiw01.pdf>
7. <http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>
8. <http://www.ati.es/gt/LATIGOO/OOp96/Ponen6/atio6p06.html>
9. <http://uca.guegue.com/AnalisisyDisenoWeb/web-app.pdf>
10. <http://www.itlp.edu.mx/publica/tutoriales/analisis/24.htm>
11. [http://www.netpecos.org/docs/mysql\\_postgres/x57.html](http://www.netpecos.org/docs/mysql_postgres/x57.html)
12. <http://www.w3c.es/Divulgacion/Guiasbreves/HojasEstilo>
13. <http://www.ortizmania.com/online/articulo.asp?art=6>
14. <http://www.web-taller.com.ar/dreamweaver.html>
15. [http://www.macromedia.com/go/thirdparty\\_es](http://www.macromedia.com/go/thirdparty_es)
16. <http://www.codigophp.com/index.php/page/2/>
17. <http://www.creangel.com/uml/diagramas.php>
18. <http://www.uoc.edu/masters/esp/img/873.pdf>
19. [http://www.sonork.com/esp/web\\_app.html](http://www.sonork.com/esp/web_app.html)