

ANÁLISIS Y DESARROLLO DE UN PROTOTIPO DE PORTAL ADAPTATIVO
PARA LA UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA

JOSÉ ANDRÉS LARA VECINO
NANCY PAOLA RUEDA CABRALES

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
FACULTAD DE INGENIERÍA DE SISTEMAS
BUCARAMANGA

2007

ANÁLISIS Y DESARROLLO DE UN PROTOTIPO DE PORTAL ADAPTATIVO
PARA LA UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA

JOSE ANDRÉS LARA VECINO
NANCY PAOLA RUEDA CABRALES

Trabajo de grado presentado para optar el título de
Ingeniero de Sistemas

Director:
Mcc. Jaime Alfredo Rangel Caballero

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
FACULTAD DE INGENIERÍA DE SISTEMAS
BUCARAMANGA

2007

Nota de aceptación

Firma del presidente del jurado

Firma del Jurado

Firma del Jurado

Bucaramanga, Noviembre de 2007

AGRADECIMIENTOS

Queremos expresar nuestros agradecimientos a:

A Dios por habernos guiado en este camino y por darnos siempre la fortaleza para seguir.

A nuestro director Jaime Rangel por guiarnos en el proyecto y por darnos buenas pautas y ánimo para seguir y por confiar siempre en nuestro proceso.

A nuestras familias por acompañarnos incondicionalmente, apoyarnos, aceptar la ausencia en momentos importantes y por estar presentes en todas las etapas de nuestra vida.

A nuestros amigos su compañía, apoyo y comprensión.

DEDICATORIA

Este trabajo esta dedicado a mi Papito que en paz descanse quien siempre soñó con verme graduada y con mi tesis lista. Aquí esta papito para ti con todo el cariño, gracias por tu apoyo y por dejarme saber que siempre contare contigo.

Para mi mamita quien siempre entendió todos mis problemas y quien fue mi fuerte en momentos de dificultad un abrazo con todo mi corazón y quien siempre con su mirada me alentó para seguir en la gran tarea que había emprendido.

A mi tía Yiya quien siempre me ha apoyado en todos mis proyectos, esto también es para ti por que nunca tendré las suficientes palabras para agradecer todo el cariño y apoyo que he recibido.

Para mi Tata gracias por siempre estar en los momentos más importantes de mi vida con tus consejos y palabras que siempre eran adecuadas.

Con amor para todos aquellos que hacen parte de mi vida...

PAOLITA!

TABLA DE CONTENIDO

| | pág. |
|--|------|
| INTRODUCCIÓN | 17 |
| 1. OBJETIVOS | 19 |
| 1.1 OBJETIVO GENERAL | 19 |
| 1.2 OBJETIVOS ESPECÍFICOS | 19 |
| 2. MARCO DE REFERENCIA | 21 |
| 2.1 SISTEMAS WEB | 21 |
| 2.2 TECNOLOGÍAS DE DESARROLLO WEB | 21 |
| 2.2.1 CGI | 21 |
| 2.2.1.1 Forma de actuación de un CGI | 22 |
| 2.2.1.2 Programación de un CGI | 22 |
| 2.2.1.3 Tipos habituales de CGI | 23 |
| 2.2.2 Lenguajes de programación para páginas dinámicas | 24 |
| 2.2.2.1 PHP | 24 |

| | |
|--|----|
| 2.2.2.2 ASP | 24 |
| 2.2.2.3 JSP | 25 |
| 2.2.3 Servlets | 26 |
| 2.2.3.1 Aspectos técnicos | 26 |
| 2.2.4 Servicios Web | 27 |
| 2.2.5 El método hipertextual orientado a objetos | 27 |
| 2.3 PERSONALIZACIÓN | 28 |
| 2.3.1 Fuentes de información | 28 |
| 2.3.1.1 Perfiles de usuario | 28 |
| 2.3.1.2 El contexto | 29 |
| 2.3.1.3 Datos preexistentes | 29 |
| 2.3.2 Personalización individual | 29 |
| 2.3.2.1 Portales sin personalización | 30 |
| 2.3.2.2 Aspectos de diseño | 30 |
| 2.4 APLICACIONES CONSCIENTES DEL CONTEXTO | 31 |
| 2.4.1 Contexto | 31 |

| | |
|---|----|
| 2.5 ADAPTATIVIDAD | 33 |
| 2.5.1 Hipermedia adaptativa | 33 |
| 3. TÉCNICAS DE ADAPTATIVIDAD | 35 |
| 3.1 SISTEMAS ADAPTATIVOS | 35 |
| 3.1.1 Modelo de usuario | 35 |
| 3.1.2 Modelo de dominio | 35 |
| 3.1.3 Mecanismo de adaptación | 35 |
| 3.2 MODELO DE USUARIO | 36 |
| 3.2.1 Creación de creencias | 37 |
| 3.3 FIRST CLICK | 38 |
| 3.4 LOG MINNING | 39 |
| 3.4.1 Aplicaciones | 41 |
| 3.4.1.1 Herramientas analizadoras y generadoras de reportes | 41 |
| 3.4.1.1.1 ANALOG | 41 |
| 3.4.1.1.2 WEBALIZER | 41 |
| 3.4.1.1.3 LOGROVER | 41 |

| | |
|--|----|
| 3.5 COOKIES | 42 |
| 3.5.1 Ventajas de las cookies | 44 |
| 3.5.2 Desventajas del uso de las cookies | 44 |
| 3.6 MODELO VISTA CONTROLADOR | 44 |
| 4. PRODUCTO FINAL | 48 |
| 4.1 MODELO DE USUARIOS | 50 |
| 4.1.1 Estudiantes | 50 |
| 4.1.2 Aspirantes | 51 |
| 4.1.3 Empresarios | 52 |
| 4.1.4 Egresados | 53 |
| 4.2 REGLAS DE ADAPTATIVIDAD DE NAVEGACIÓN | 54 |
| 4.3 LOG DEL SERVIDOR DE LA UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA | 57 |
| 4.4 MECANISMOS DE ADAPTATIVIDAD | 60 |
| 4.4.1 Ventajas y desventajas de las técnicas de adaptatividad estudiadas | 60 |
| 4.5 LA CLASE COOKIE | 61 |
| 4.5.1 Script generador de una cookie | 62 |

| | |
|--|----|
| 4.5.2 Script para capturar la información de una cookie | 62 |
| 4.6 ALGORITMO DE CONSTRUCCIÓN DE LA INFORMACIÓN ADAPTADA DE ACUERDO AL PERFIL DE USUARIO | 63 |
| 4.6.1 Pseudocódigo | 63 |
| 4.6.1.1 Detalle de procedimientos | 64 |
| 4.7 REASIGNACIÓN DEL PERFIL DE USUARIO DE ACUERDO A LAS VISITAS HECHAS | 65 |
| 4.7.1 Detalle de procedimientos | 65 |
| 4.8 BASE DE DATOS | 66 |
| 4.8.1 Modelo ER | 67 |
| 4.8.2 Listado de consultas | 67 |
| 4.8.2.1 Perfil | 67 |
| 4.8.2.2 Visita | 68 |
| 4.8.2.3 Usuario | 68 |
| 5. CONCLUSIONES | 69 |
| 6. TRABAJOS FUTUROS | 71 |
| BIBLIOGRAFÍA | 72 |
| ANEXOS | 79 |

TABLA DE FIGURAS

| | pág. |
|--|------|
| Figura 1. Diagrama de flujo de datos Algoritmo para cargar perfil | 63 |
| Figura 2. Diagrama de flujo de datos Algoritmo para recalculer el perfil | 65 |
| Figura 3. Modelo Entidad – Relación de la base de datos | 67 |

TABLA DE ANEXOS

| | pág. |
|-------------------------------------|------|
| Anexo A. Diccionario de datos | 80 |
| Anexo B. Script de la base de datos | 82 |

GLOSARIO

DESARROLLO WEB: conjunto de tecnologías de software del lado del servidor y del cliente que involucran una combinación de procesos de base de datos con el uso de un navegador en Internet a fin de realizar determinadas tareas o mostrar información.

HIPERMEDIA: integración de gráficos, sonido y vídeo en un sistema que permite el almacenamiento y recuperación de la información de manera relacionada, por medio de referencias cruzadas.

HIPERTEXTO: método de presentación de la información que permite hacer una lectura no secuencial de la misma. Trata de emular el modo en que el cerebro humano almacena y recupera la información, por medio de asociación de ideas, y no en el orden en el que los sucesos ocurrieron. En el hipertexto, la información se organiza en torno a una serie de palabras clave, que permiten que, al llegar a ellas, el documento pueda avanzar hacia otra parte del mismo (delante o detrás de esa posición) o presentar información que se encuentra en otro documento.

INGENIERIA WEB: área de la Ingeniería del Software que trabaja en el entorno de los sistemas Web encargada de estudiar entornos metodológicos orientados hacia las aplicaciones Web en general.

INTERFAZ: conjunto de componentes empleados por los usuarios para comunicarse e interactuar con las computadoras. El usuario dirige el funcionamiento de la máquina mediante instrucciones, denominadas genéricamente entradas.

PAGINA: una página Web es un documento electrónico escrito en un lenguaje de computación llamado HTML. Cada página Web tiene una dirección única, llamada URL ó Localizador Uniforme de Recursos, que identifica su localización en la red¹.

PORTAL: un portal de Internet es un sitio que está dirigido a resolver necesidades específicas de un grupo de personas. Los portales normalmente tienen programación que requiere muchos recursos computacionales y por su alto tráfico generalmente se hospedan en servidores dedicados. Un portal de Internet es un sitio Web cuyo objetivo es ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios².

ROL: patrón de conducta de las personas en las situaciones sociales. El rol puede ser entendido como el papel que pone en práctica una persona.

SCRIPT: procedimiento que se utiliza generalmente para llevar a cabo tareas secuenciales, paso a paso, como son las definidas en macros de aplicaciones

SITIO WEB: un sitio Web, dependiendo de cómo esté diseñado, contiene una o más páginas relacionadas entre sí. Las páginas de un sitio Web están vinculadas entre sí a través de un sistema de hipervínculos, que permiten que el visitante pueda moverse de unas a otras haciendo click sobre un vínculo³.

TRIGGER: un *trigger* o un disparador en una Base de datos, es un evento que se ejecuta cuando se cumple una condición establecida al realizar una Michael operación de inserción (INSERT), actualización (UPDATE) o borrado (DELETE).

¹ Michael Lerner Productions, Estructura de una Página Web
<http://www.learnthenet.com/Spanish/html/14wbpganat.htm>. Página visitada el 3 de abril de 2006.

² Colaboradores de Wikipedia. Portal (Internet),
http://es.wikipedia.org/wiki/Portal_%28internet%29. Página visitada el 3 de abril de 2006.

³ Michael Lerner Productions, Navegacion a través de hipervinculos.
<http://www.learnthenet.com/Spanish/html/18wbpganat.htm>. Página visitada el 3 de abril de 2006.

USUARIO: el usuario final de un producto informático (bien sea hardware o software), es la persona a la que va destinada dicho producto una vez ha superado las fases de desarrollo correspondientes. Normalmente, el software se desarrolla pensando en la comodidad del usuario final, y por esto se presta especial interés y esfuerzo en conseguir una interfaz de usuario lo más clara y sencilla posible.

RESUMEN

Actualmente los portales Web están encontrando en la adaptatividad una herramienta interesante para cautivar y generar fidelidad en los usuarios, ya que tanto el contenido como la interfaz se adaptan a los requerimientos de los diversos perfiles de usuario.

En los canales virtuales de la Universidad Autónoma de Bucaramanga UNAB se tienen identificados varios perfiles de usuarios, pero la estrategia que se maneja es desarrollar un portal para cada perfil, lo que causa que se tenga información distribuida en varios portales, adicionalmente, el usuario tiene que evidenciar cual es su perfil y además, los tiempos de desarrollo y mantenimiento de los servicios son extensos.

Este trabajo pretende recopilar los perfiles de usuarios detectados por los canales virtuales, realizar un estudio sobre el modelo adecuado de adaptatividad y generar un prototipo de portal genérico donde la aplicación maneje el perfil del usuario y adapte su contenido y su interfaz.

Este proyecto aporta investigación sobre nuevas técnicas de desarrollo en la generación de aplicaciones, lo que contribuye con el mejoramiento de procesos en el desarrollo de software. Además de la innovación que representa el aplicar técnicas de vanguardia en el mejoramiento de la experiencia de los usuarios del sitio Web institucional de la Universidad Autónoma de Bucaramanga.

LINEA DE INVESTIGACIÓN

Telecomunicaciones y Tecnologías Web

PALABRAS CLAVES

Adaptatividad, sitio Web, interfaz, navegabilidad.

INTRODUCCIÓN

La Universidad Autónoma de Bucaramanga UNAB creó su primer sitio Web hacia el año de 1.996. Esta primera versión, era usada como un medio de información a la comunidad, pero solo prestaba este servicio. Tanto la interfaz gráfica, como la información presentada al visitante era lo más fundamental y no pasaba de un nivel básico. Durante esta primera versión, la actualización de la información era distante en el tiempo y constaba de remplazar texto por otro.

El segundo cambio importante en el sitio Web de la universidad, se llevó a cabo durante el periodo de tiempo comprendido entre los años 2.000 y 2.005. En esta recreación del sitio, se mejoró la interfaz y el contenido, ofreciendo más y mejor información, pero aun no ofrecía la de mayor relevancia para cada usuario. Es importante anotar que hasta este punto el sitio Web de la universidad había sido pensado desde la organización, por lo que no estaba enfocada al usuario o visitante.

El sitio Web de la Universidad Autónoma de Bucaramanga maneja diversos perfiles o tipos de usuarios que pueden acceder a la información publicada. Actualmente, la estrategia aplicada para brindar información a estos perfiles, es el desarrollo de un sitio Web a cada perfil existente. Debido a esta situación, en algunos casos se tiene redundancia de información, es decir, existe la misma información dispersa en los diferentes portales.

Este esquema hace que el mantenimiento de los portales sea una labor compleja, ya que cuando se hace necesario un cambio de la información publicada, se debe modificar la misma información en los diferentes sitios Web.

Esto puede causar desactualización de la información, o distintas versiones de la misma información, lo cual trae confusión al visitante.

Este proyecto, pretende recopilar un marco teórico sobre el comportamiento de los diferentes tipos de usuario que acceden al sitio Web de la Universidad Autónoma de Bucaramanga, y con base a esta, construir un prototipo de sitio Web institucional de la Universidad Autónoma de Bucaramanga que adapte la información existente a cada perfil de usuario y que el sitio Web se comporte según las necesidades del mismo.

En este momento el departamento de comunicaciones de la Universidad Autónoma de Bucaramanga busca generar un portal que preste la información pertinente a cada perfil de usuario, y que se adapte a las necesidades de cada uno de estos, logrando así satisfacer a todos los usuarios.

1. OBJETIVOS

1.1 OBJETIVO GENERAL

Analizar y desarrollar un prototipo de sitio Web adaptativo para la Universidad Autónoma de Bucaramanga, mediante un estudio de los modelos de adaptatividad de contenido e interfaz para mejorar el manejo de la información de los distintos perfiles en la UNAB.

1.2 OBJETIVOS ESPECÍFICOS

Redactar un documento donde se recopile un estado del arte sobre la personalización y adaptatividad, mediante el análisis de las diferentes técnicas de adaptatividad y personalización de contenido e interfaz, para tener un marco de referencia sobre el tema para su posterior utilización en la implementación del prototipo.

Definir los distintos perfiles de los visitantes del sitio Web institucional de la Universidad Autónoma de Bucaramanga y los contenidos que se desean presentar para cada perfil, mediante un estudio de las visitas al portal y entrevistas con el administrador de contenidos de los canales virtuales y el director de la oficina de mercadeo de la UNAB, para tener el contenido del sitio para su posterior análisis.

Desarrollar un prototipo de portal Web adaptativo para la Universidad Autónoma de Bucaramanga mediante la utilización de la técnica definida por el

estudio anterior para presentar los contenidos e interfaz adaptada a los requerimientos de cada uno de los perfiles definidos

2. MARCO DE REFERENCIA

2.1 SISTEMAS WEB

El desarrollo de documentos Web a cambiado a lo largo de la historia de las tecnologías Web, de mostrar solo información estática tomada de un servidor Web se ha pasado a montar sistemas y aplicaciones en línea que prestan servicios basados en tecnologías Web, y páginas que se crean y modifican según las circunstancias.

En sus inicios, la Web fue pensada como un medio para desplegar información estática que reposa en un servidor⁴. Pero, actualmente, aunque los sistemas cliente continúan cargando la información desde un servidor Web, esta no tiene por que ser estática, sino que, dependiendo de las circunstancias (consultas, peticiones, etc.) adecuará su contenido para la visualización que se le enseñara al usuario.

2.2 TECNOLOGÍAS DE DESARROLLO WEB

2.2.1 CGI. Common Gateway Interface CGI es un mecanismo de comunicación entre el servidor WEB y una aplicación externa desarrollada en casi cualquier lenguaje, que solo debe cumplir la condición de ser soportado

⁴ Cingler M. Desarrollo de aplicaciones y servicios web. http://glud.udistrital.edu.co/glud/areas/doc/articulos/1_articulo_ws/aplicaciones-Web.html. Pagina visitada el 5 de febrero de 2006.

por el servidor http, es común encontrar que la mayoría de las aplicaciones CGI se encuentren desarrolladas con el lenguaje Perl.⁵

2.2.1.1 Forma de actuación de un CGI. A continuación se describe la forma de actuación de un CGI de forma esquemática:

- En primera instancia, el servidor recibe una petición (el cliente ha activado un enlace que contiene el CGI), y comprueba si se trata de una invocación de un CGI.
- Posteriormente, el servidor prepara el entorno para ejecutar la aplicación. Esta información procede mayoritariamente del cliente.
- Seguidamente, el servidor ejecuta la aplicación, capturando su salida estándar.
- A continuación, la aplicación realiza su función: como consecuencia de su actividad se va generando un objeto MIME que la aplicación escribe en su salida estándar.
- Finalmente, cuando la aplicación finaliza, el servidor envía la información producida, junto con información propia, al cliente, que se encontraba en estado de espera. Es responsabilidad de la aplicación anunciar el tipo de objeto MIME que se genera (campo CONTENT_TYPE), pero el servidor calculará el tamaño del objeto producido.

2.2.1.2 Programación de un CGI. Un programa CGI puede ser escrito en cualquier lenguaje de programación que produzca un fichero ejecutable. Entre los lenguajes más habituales se encuentran: C, C++, Perl, Visual Basic... No

⁵ Ibíd.

obstante, debido a que el CGI recibe los parámetros en forma de texto será útil un lenguaje que permita realizar manipulaciones de las cadenas de caracteres de una forma sencilla, como por ejemplo Perl. Perl es un lenguaje interpretado que permite manipulaciones sencillas de ficheros y textos, así como la extracción y manipulación de cadenas de caracteres, unidas a unas búsquedas rápidas y fáciles.

2.2.1.3 Tipos habituales de CGI.

Contador de accesos: Cuenta el número de veces que se ha solicitado una página determinada. Se guarda el valor en un fichero. Cada vez que se invoca se incrementa, para su posterior visualización.

Buscador: Localiza páginas que contengan los términos especificados. Utiliza una tabla que enumera las palabras y para cada una especifica las páginas dónde se encuentra.

Correo: Obtiene información estructurada del usuario.

Contribuciones: Permite añadir enlaces o anotaciones a una página, indicando la procedencia de la adición.

Estadísticas de uso: Presenta información sobre los acontecimientos producidos en el servidor de Web. El servidor mantiene un registro (*log*) de los acontecimientos que se han producido.

Administración remota del servidor: Permite interactuar con el servidor desde WWW. Invoca los programas que controlan o modifican el comportamiento del servidor.

2.2.2 Lenguajes de programación para páginas dinámicas

2.2.2.1 PHP. PHP Hypertext Preprocessor. Es un lenguaje de programación usado generalmente para la creación de contenido para sitios Web. Es un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web, y últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica de usuario.

El fácil uso y la similitud con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores experimentados crear aplicaciones complejas con una curva de aprendizaje muy suave. Igualmente les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones y prácticas^{6,7}.

2.2.2.2 ASP. Con la tecnología *Active Service Pages* (ASP) se puede crear un sitio activo que cambie cada semana, cada día o cada hora. Si se tiene una bases de datos de los productos y sus precios han cambiado solamente hay que introducir la nueva bases de datos en el servidor, y todas sus páginas estarán actualizadas.

Una base de datos permite crear una parte del sitio Web restringido con clave de acceso. Esto puede ser útil si se quiere crear fichas de información para los clientes para los cuales no se tenga libre acceso, creando así su propia Extranet^{8,9}

⁶ Manual de Referencia, <http://www.php.net/>. Página visitada el 1 de abril de 2006.

⁷ Colaboradores de Wikipedia, PHP, <http://es.wikipedia.org/wiki/Php>. Página visitada el 5 de febrero de 2006.

⁸ Microsoft, Centro de desarrollo de aplicaciones ASP .NET, http://www.microsoft.com/spanish/msdn/centro_recursos/asp_net/default.msp. Página visitada el 5 de febrero de 2006.

2.2.2.3 JSP. Java Server Pages JSP es la tecnología para generar páginas Web de forma dinámica en el servidor, desarrollado por Sun Microsystems, basado en scripts que utilizan una variante del lenguaje java.

La tecnología JSP, es una tecnología Java que permite a los programadores generar dinámicamente HTML, XML o algún otro tipo de página Web. Esta tecnología permite al código Java y a algunas acciones predefinidas ser embebidas en el contenido estático. En las JSP, se escribe el texto que va a ser devuelto en la salida (normalmente código HTML) incluyendo código java dentro de él para poder modificar o generar contenido dinámicamente.

En una posterior especificación, se incluyeron *taglib*; esto es, la posibilidad de definir etiquetas nuevas que ejecuten código de clases java.^{10,11}

La principal ventaja de JSP frente a otros lenguajes es que hace posible una integración con clases Java (.class) lo que permite separar en niveles las aplicaciones Web, almacenando en clases java las partes que consumen más recursos, así como las que requieren más seguridad, y dejando la parte encargada de dar formato al documento html en el archivo JSP.

Además Java se caracteriza por ser un lenguaje que puede ejecutarse en cualquier sistema, lo que sumado a JSP le da mucha versatilidad.

Sin embargo JSP no se puede considerar un *script* al 100% ya que antes de ejecutarse el servidor Web compila el *script* y genera un servlet, por lo tanto se puede decir que aunque este proceso sea transparente para el programador no deja de ser una aplicación compilada. La ventaja de esto es algo más de rapidez y disponer del API de Java en su totalidad.

⁹ José Valle, Bases de Datos, <http://www.monografias.com/trabajos24/bases-de-datos/bases-de-datos.shtml>. Página visitada el 1 de abril de 2006.

¹⁰ <http://java.sun.com/products/jsp/>. Página visitada el 5 de febrero de 2006.

¹¹ <http://es.wikipedia.org/wiki/JSP>. Página visitada el 1 de abril de 2006.

2.2.3 Servlets. Los *servlets* son objetos que corren dentro del contexto de un servidor Web (como Tomcat) y extienden su funcionalidad. También podrían correr dentro de un servidor de aplicaciones (ej: OC4J Oracle) que además de contenedor para servlet tendrá contenedor para objetos más avanzados como son los EJB's ó *Enterprise JavaBeans* (Tomcat solo es un contenedor de servlets).

La palabra *servlet* deriva de otra anterior, *applet*, que se refiere a pequeños programas escritos en Java que se ejecutan en el contexto de un navegador Web. Por contraposición, un *servlet* es un programa que se ejecuta en un servidor Web.

El uso más común de los *servlets* es generar páginas Web de forma dinámica a partir de los parámetros de la petición que envíe el navegador Web.

2.2.3.1 Aspectos técnicos. Un *servlet* es un objeto que se ejecuta en un servidor o contenedor J2EE, fue especialmente diseñado para ofrecer contenido dinámico desde un servidor Web, generalmente es HTML. Los *servlets* forman parte de J2EE (Java 2 Enterprise Edition).

Un *servlet* es un objeto Java que implementa la interfaz `javax.servlet.Servlet` o hereda alguna de las clases más convenientes para un protocolo específico (ej: `javax.servlet.HttpServlet`).

Entre el servidor de aplicaciones (o contenedor Web) y el *servlet* existe un contrato que determina cómo han de interactuar. La especificación de éste se encuentra en los JSR (Java Specification Requests) del JCP (Java Community Process)^{12,13}.

¹² Stephanie Bodoff, Java Servlet Technology, http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/Servlets.html. Página visitada el 8 de febrero de 2006.

¹³ Colaboradores de Wikipedia. Java Servlet, <http://es.wikipedia.org/wiki/Servlet>. Página visitada el 1 de abril de 2006.

2.2.4. Servicios Web. Los servicios Web son la revolución informática de la nueva generación de aplicaciones que trabajan en colaboración en las cuales el software esta distribuido en diferentes servidores. Actualmente, se usa la tecnología Internet para solventar las problemáticas asociadas a fallo de aplicación centralizado. Los Web services son un paso adelante en la computación ya que de esta forma un computador ya no se considera como un núcleo de cómputo sino como un repositorio de servicios de aplicaciones distribuidas por Internet¹⁴.

2.2.5 El método hipermedial orientado a objetos (OO-H). Este paradigma, se enfoca en ofrecer una respuesta a la necesidad de personalización dinámica, partiendo de la problemática que generan las políticas de personalización en el ciclo de desarrollo de aplicaciones personalizables, desde la toma de requerimientos hasta la implementación. Este método, específicamente busca llevar a cabo las actividades de diseño de personalización manera paralela a las demás tareas de modelado de sistemas y subsistemas.

El OO-H, se caracteriza principalmente por que la ingeniería de software aplicada durante el proceso de desarrollo (que en este caso se denomina ingeniería Web) debe considerar un modelado en función de los distintos tipos de audiencia, con el fin de permitir la personalización de los entornos ofrecidos, bien sean estos últimos en cuanto a contenido, apariencia, comportamiento o una mezcla de todas estas propiedades. El OO-H, nace como una respuesta a la necesidad de involucrar al usuario dentro del proceso de propagación de aplicaciones interactivas, ofreciéndole al mismo la visión de usar un aplicativo diseñado específicamente para sus requerimientos.

¹⁴ Juan Antonio Breña Moral, Introducción a .NET, <http://www.monografias.com/trabajos11/winnet/winnet.shtml>. Página visitada el 6 de febrero de 2006.

2.3 PERSONALIZACIÓN

Durante los últimos años se ha despertado un gran interés por todo lo que a tecnologías Web se refiere. La interacción entre las personas y los PC's es cada vez más creciente y el complejo acceso al los sistemas de información hace que se generen campos de estudio para hacer mas fácil la interacción entre las partes. La Web es parte de estos sistemas complejos debido a su heterogeneidad y es necesario que el usuario sea capaz de realizar las tareas de forma cómoda y eficaz.

La personalización de aplicaciones, es la capacidad de alteración dinámica con el fin de proporcionar al usuario la impresión de estar trabajando en una aplicación específicamente diseñada para dar satisfacción a sus necesidades particulares partiendo en base a tres tipos de fuentes de información¹⁵.

2.3.1 Fuentes de información.

2.3.1.1 Perfiles de usuario. Para hacer de las tareas Web algo sencillo ante el usuario debemos como primera medida definir unos perfiles de usuario que permitirán darle el enfoque a la navegación de acuerdo a las necesidades del usuario. Cuando buscamos que la interfaz satisfaga a los usuarios muchas veces no podremos lograrlo en la primera vez por lo que hay que implementar técnicas para que el contenido sea completamente adaptado a las necesidades y a los requisitos del usuario.

Estas técnicas nos permiten crear grupos para los posibles usuarios con una serie de características en común y con comportamientos similares. Cuando

¹⁵ Cachero Cristina, Garrigos Irene y Gómez Jaime. Tratamiento de Personalización dinámica en modelos conceptuales de aplicaciones Web. IWAD Group. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante. España.

clasificamos al usuario en uno de estos grupos estamos adaptando la interfaz en todo momento dado el perfil facilitándole la forma de iteración con el PC.

Los grupos de usuarios se crean a través de varias sesiones de navegación en las que el usuario siempre seguirá algunas pautas en función de lo que la información requiera.

2.3.1.2 El contexto. Se debe tener en cuenta también el contexto en el que se produce la interacción usuario-aplicación, incluyendo localización física, momento, y prestaciones de la red, entre otras características que se recogen (si están disponibles) de forma implícita por la aplicación como parte integrante de esa interacción¹⁶.

2.3.1.3 Datos preexistentes. Estos, son información tomada desde otras fuentes, formularios de inscripción, formatos de encuesta (vía Web) ajenos al sistema que se esta construyendo, y que se importan para llevar a cabo la personalización del sitio Web.

2.3.2 Personalización individual¹⁷. La personalización individual es una implementación la personalización que permite a cada usuario fijar la configuración de la información desplegada en el sitio al que accede. Aun así, mientras que la presentación de diversas vistas del contenido a diversos grupos de interés tiene ventajas, aún no es claro si la personalización (en un nivel individual) es algo necesario, agradable, o inútil.

¹⁶ Cachero Cristina, Garrigos Irene y Gómez Jaime. Personalización de aplicaciones en OO-H. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante. España

¹⁷ Goodwin C., Schwartz M. and Nielsen J. Usability of Intranet Portals— a Report From the Trenches. En : Experiences From Real-Life Portal Projects. Segunda edición. Nielsen Norman Group.

Como son tantos los aspectos de diseño a tener en cuenta, las necesidades de los usuarios influenciarán grandemente el grado en el cual un portal incluye la personalización. ¿Cuándo se debe evitar la personalización individual? No es fácil responder esta pregunta, pero es de observar que en sitios altamente funcionales (como herramientas de trabajo o aplicaciones Web), los usuarios pueden percibir la personalización individual como una interferencia con el acceso rápido a su información. Muchas compañías en esta situación optaron por asumir que los usuarios solo desean información, y realizar su trabajo tan rápida y eficientemente como sea posible. Precisamente, *Hewlett Packard* eliminó la característica individual de la personalización de su sitio corporativo debido a la carencia de interés de los usuarios, y en algunos de los cuales se encontró contraproducente esta característica.

2.3.2.1 Portales sin personalización. Mientras que muchos consideran la personalización como una característica que un portal debe tener, en la realidad son muchos los portales que existen sin implementar la personalización. Esto se debe a que algunas organizaciones creen que toda la información corporativa debe estar disponible para todos los empleados. Estas organizaciones no tienen argumentos fuertes a favor de la personalización, y en cambio, si ven como una desventaja el restringir la información que la gente ve.

2.3.2.2 Aspectos de diseño

- Muchos usuarios no realizan la tarea de personalizar páginas, y aceptan fácilmente lo que se les ofrece.
- Decidir que información necesitan ver los usuarios es muy difícil. Usar la personalización para desenfocar (en lugar de ocultar) la información menos

relevante, le ofrece a los usuarios una mayor capacidad de búsqueda y navegabilidad.

- La personalización basada en perfiles de usuario usualmente le permite a los usuarios acceder a la información que necesitan, sin gastar tiempo ni energía en configurar sus preferencias.
- Es necesario hacer una diferencia entre personalización (*personalization*), en la que el autor decide que puede ver cada usuario, y caracterización (*customization*), en la que el usuario decide que se le debe mostrar.

2.4 APLICACIONES CONSCIENTES DEL CONTEXTO

Cuando interactuamos con otras personas y con el ambiente utilizamos información que podemos captar de nuestro alrededor. Podemos deducir e interpretar el contexto de la situación en curso intuitivamente y reaccionar apropiadamente. Los computadores no tienen estas capacidades por si mismas para deducir información situacional de su ambiente y usarlo en sus interacciones. Este es el desafío para la interacción entre el computador y los humanos.

2.4.1 Contexto¹⁸. El contexto es cualquier información que puede ser usada para caracterizar la situación de una entidad. Una entidad es una persona o una posición que son considerados relevantes para la interacción entre un usuario y la aplicación. Cualquier información disponible en el momento de la interacción puede ser vista como información de contexto. Algunos ejemplos son:

¹⁸ Korkea M. Context-Aware Applications Survey. Department of Computer Science. Helsinki University of Technology. Publicado el 25 de abril de 2000. <http://users.tkk.fi/~mkorkeaa/doc/context-aware.html>

- La información espacial (Ubicación, Orientación, Velocidad y aceleración).
- La información temporal (La hora del día, fecha y estación del año).
- La información ambiental (Temperatura, calidad del aire, luz y nivel de ruido).
- La situación de compromiso social (Con quien esta y personas cercanas).
- Recursos cercanos. (Dispositivos accesibles).
- Disponibilidad de recursos (Tipo de red y ancho de banda).
- Mediciones fisiológicas (Presión sanguínea, tasa de respiración, actividad muscular, tono de la voz).
- Actividad (Que habla, que lee, si corre o anda).

Un sistema es consiente del contexto si puede extraer, interpretar y usar la información del contexto. El desafío para estas aplicaciones esta en la complejidad para captar, representar y procesar datos contextuales. Para captar este tipo de datos son necesarios ciertos sensores y aplicaciones especiales que deben tener un formato de representación común y cierto nivel de "inteligencia" para procesar la información y deducir el significado. Este tipo de aplicaciones son muy estimulantes para el usuario ya que el contexto en la mayoría de veces es indirecto y combinado. Un ejemplo de estas aplicaciones es si en una oficina 3 personas coinciden cada cierto tiempo en una ubicación especifica del recinto el sistema puede deducir que se trata de una reunión de estrategia semanal.

El uso de información de contexto es aun muy limitado ya que es muy complicado de captar, representar, y procesar los datos contextuales. El tipo de información más usada para realizar este procesamiento son la ubicación, la

identidad y la información de tiempo. El desarrollo de aplicaciones conscientes del contexto es principalmente grande alrededor de interfaces de usuarios, y ha aumentado bastante en el campo de realidad virtual. También estas tecnologías son usadas actualmente en el campo móvil y computadores de mano. Las aplicaciones existentes están principalmente sobre una fase de creación o prueba ya que debido a su complejidad aun no son muy comerciales. Estas aplicaciones son usadas en áreas de oficina, como guías, o herramientas de trabajo de campo.

2.5 ADAPTATIVIDAD

Los sistemas adaptativos son aquellos que se adaptan al usuario automáticamente, basándose en las suposiciones que el sistema realiza de las necesidades del usuario¹⁹, obtenidas a partir del estudio de un rol o perfil de usuario. Son sistemas que monitorizan el comportamiento de los usuarios y adaptan la presentación (interfaz y contenido) teniendo en cuenta dicho comportamiento. En estos, el crecimiento del sistema, puede ser deducido por el mismo a partir de los accesos realizados a las páginas. Aunque también se podrían necesitar cuestionarios a fin de obtener información más fiable respecto a las preferencias del usuario. La mayor parte de la adaptación es realizada, sin embargo, sobre la base de las acciones de navegación del usuario y también considerando el comportamiento de otros usuarios, agrupándolos por roles o perfiles.

2.5.1 Hipermedia adaptativa²⁰. La hipermedia adaptativa es un sistema de hipertexto que refleja algunas características de sus diferentes usuarios, y

¹⁹ Rojas Roger E. Aplicando técnicas de Minería de Datos en el desarrollo de un sitio Web adaptativo. Trabajo de grado (Ingeniero de Sistemas) - Universidad Autónoma de

²⁰ Rodríguez Paulo, de Brito Silvana, Fonseca Orlando y Jose Lima. Adaptive interfaces to virtual learning communities. Programa de Pós-Graduação em Engenharia Elétrica (PPGEE).

adapta algunos de los aspectos más relevantes del sistema a las necesidades, deseos y preferencias de cada usuario²¹. Con la adaptatividad hipermedial, se busca ofrecer al usuario una interfaz, cuyos contenidos, estilo y vínculos sean seleccionados automáticamente y se generen dinámicamente a partir de los objetivos, necesidades y preferencias de los usuarios²².

²¹ Brusilovsky, Peter. Adaptative and Intelligent Technologies for Web-based Education, In: Künstliche Intelligenz, Special Issue on Intelligent System and Teleaching, 19-25 /4, 1999.

²² Ibid.

3. TÉCNICAS DE ADAPTATIVIDAD

3.1 SISTEMAS ADAPTATIVOS

Los sistemas de hipermedia adaptativa tienen tres componentes básicos en su arquitectura, el modelo de usuario, el modelo de dominio y el mecanismo de adaptación.

3.1.1 Modelo de usuario. Describe el usuario ante el sistema, señala sus preferencias y objetivos. Representa la navegabilidad del usuario y el comportamiento del mismo a la hora de interactuar con el sistema²³.

3.1.2 Modelo de dominio. Describe como se estructurará la información de la aplicación, usando páginas y conceptos definidos por el autor.

3.1.3 Mecanismo de adaptación. Ensambla los dos modelos anteriores para definir la adaptatividad del sistema. Para el caso de nuestro portal adaptativo, es necesario tener en cuenta la diversidad de áreas del conocimiento dentro de la universidad, pues cada estudiante tiene diferentes niveles, estilos y tipos de educación.

²³ Vicari, Rosa; GIRAFFA. "Introdução aos Sistemas Tutores Inteligentes". Aulas do programa de pósgraduação em ciência da computação, UFRGS, 2000.

3.2 MODELO DE USUARIO

El modelo surgió como una necesidad frente a la variedad y número posible de usuarios del sistema. Esta variedad ha sido motivo de estudio por distintos investigadores^{24,25, 26, 27}, los que señalan algunos aspectos determinantes de la misma. Los beneficios que pueden esperarse del uso del modelo para alcanzar la adaptatividad del sistema son varios, entre los cuales se encuentra:

- Adaptarse a distintas tareas
- Alcanzar diferentes objetivos
- Adecuarse a necesidades cambiantes
- Atender distintos niveles de conocimiento y destreza

El modelado del usuario es el proceso de identificar intereses o preferencias de los usuarios, analizando su comportamiento. Este proceso de modelado sufre varios inconvenientes ya que este es dinámico y los intereses de cada uno cambian a través del tiempo.

Un modelo de usuario debe incluir²⁸:

²⁴ Sánchez Villalobos F. Modelado del usuario para ambientes de aprendizaje colaborativo en Internet. 2.000. Tesis profesional presentada como requisito parcial para obtener el título en Maestría en *Ciencias con Especialidad en Ingeniería en Sistemas Computacionales*. Universidad de las Américas, Puebla, México.

²⁵ López-Jaquero V., Montero F., Fernandez A. y Lozano M. Adaptabilidad de Interfaces de Usuario por Reflexión. 2.003. Grupo de Investigación LoUISE, Universidad de Castilla, Albacete, España.

²⁶ Macías J. y Castells P. Diseño interactivo de cursos adaptativos. 2002. E.T.S. de Informática, Universidad Autónoma de Madrid.

²⁷ Medina-Medina N., García-Cabrera L., Rodríguez-Fortiz M. y Parets-Llorca J. Adaptación al Usuario en Sistemas Hipermedia: El Modelo SEM-HP. 2.004. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Granada. España.

²⁸ Macías J. y Castells P.. Diseño interactivo de cursos adaptativos. 2002. E.T.S. de Informática, Universidad Autónoma de Madrid.

- Datos personales del usuario.
- Conjunto de creencias respecto al material visitado por el usuario.

Las creencias son mantenidas por un conjunto de reglas. Estas reglas son consideradas como los intereses de cada usuario de acuerdo con el material visitado.

Cada creencia tiene un grado de adaptatividad que se encuentra dado por el grado de confianza que el usuario tiene acerca del contenido visitado.

Las creencias están clasificadas en dos grupos:

Básicas: son aquellas que directamente se crea con base en el comportamiento del usuario, cuando este visita algún material contenido en el sitio Web.

Derivadas: son aquellas que el usuario infiere a partir de las visitas a material relacionado. Estas creencias representan una probabilidad *a-priori* de que el usuario pueda estar interesado en páginas relacionadas con aquellas que el ha visitado.

3.2.1 Creación de creencias²⁹. El sistema puede crear nuevas creencias dependiendo de algunas de las siguientes consideraciones:

Creación de una creencia básica

Si el usuario consulta un material X de la página Web, que no había sido consultado antes por el mismo, o el usuario no estaba considerado como interesado en el material, entonces se crea la creencia básica según la cual el

²⁹ O'Brien M., Keane M. y Smith B. Predictive Modeling of First-Click Behavior in Web-Search. 2001. Adaptive information cluster, School of computer science and informatics. University College Dublin, Irlanda.

usuario tiene interés en este tipo de material y se justifica la creencia con el hecho de que el material forma parte de la información mostrada.

Creación de una creencia derivada

Si el usuario consulta un material *X* el cual no ha sido visitado anteriormente pero esta relacionado a *Y*, o el usuario no estaba considerado como interesado en *X*, entonces se crea la creencia básica de que el usuario tiene interés en el material *X*, también se crea la derivada de que el usuario tiene interés en *Y*, además se crea la creencia con los hechos de que *X* es una material relacionado con *Y*, y que se encuentran en el acervo. Esto quiere decir que la creencia creada para *Y* es derivada de la creencia básica de *X*.

3.3 FIRST CLICK

Dentro de la navegación por Internet existen modelos relacionados con los procesos cognoscitivos. Estos modelos asumen que los usuarios son sensibles a cierta información que es relevante para ellos. Las estrategias de búsqueda del usuario son predicativamente útiles cuando están aplicadas al comportamiento dado por el primer click³⁰. Este comportamiento se refiere principalmente a lo que sucede cuando un usuario plantea una pregunta al buscador y este evalúa la lista de resultados, escogiendo uno de ellos. Este comportamiento del primer click nos presenta un panorama amplio para realizar deducciones acerca de preferencias del usuario, detectando el material de interés del mismo para posteriormente asumir a que rol o grupo de usuarios pertenece.

La principal aplicación de la técnica del primer click, es la utilizada por motores de búsqueda de última generación, que implementan la detección del material

³⁰ O'Brien M., Keane M. y Smith B. Predictive Modeling of First-Click Behavior in Web-Search. 2001. Adaptive information cluster, School of computer science and informatics. University College Dublin, Irlanda

seleccionado entre los resultados de búsqueda, para en búsquedas subsiguientes, encontrar información más relevante.

En cuanto a la adaptatividad, la técnica del *first click*, construye un mapa de navegación del usuario a partir del primer material examinado por el mismo, pero la implementación de esta técnica, en aplicaciones adaptativas ha mostrado una baja tasa de efectividad, debido a la dificultad que representa para el sistema realizar suposiciones correctas sobre las preferencias del usuario, sin embargo, esta efectividad está ligada al buen modelado de los roles de usuarios determinados en el modelo de usuarios³¹.

3.4 LOG MINING

Los logs de acceso a servidores, consisten en una serie de registros de todas las operaciones sobre archivos generadas por los usuarios que accedan las páginas Web. Cada entrada consiste en una dirección IP de usuarios, fecha y hora de acceso, método de solicitud, URL de la página accesada, protocolo usado, código retornado, cookie perteneciente al usuario y otros campos especificados por el formato del log del servidor³². Cada formato de log varía según el servidor Web y la configuración del mismo.

El log es una fuente de información que se debe tener presente para valorar el rendimiento de un portal Web, ya que contiene toda la información sobre qué material está siendo solicitado por los usuarios y qué información les resulta más interesante³³.

³¹ Church K., Keane M. y Smyth B. The First Click is the Deepest: Assessing Information Scent Predictions for a Personalized Search Engine. 2001. Adaptive Information Cluster, Department of Computer Science, University College Dublin, Irlanda.

³² Kamdar T. y Joshi A. On creating adaptive Web servers using weblog mining. 2000. Department of Computer Science and Electrical Engineering. University of Maryland.

³³ Koutri M, Avouris N. y Daskalaki S. A survey on Web usage mining techniques for Web-based adaptive hypermedia systems. 2002. University of Patras, Grecia.

Cada vez que se lleva acabo una petición en el sitio Web, el servidor escribe una línea en el log, por ejemplo:

```
172.16.110.121 - - (30/Jul/2006:17:02:03 +0100) "GET
/directorio/archivo.html HTTP/1.1" 304 -
"http://www.dominio.com/directorio/index.html" "Firefox
(compatible; Mozilla 4.0 Linux Kernel 2.4)".
```

En donde cada una de estas cosas representa:

| | |
|---|---|
| 172.16.110.121 | Es la dirección IP de la maquina cliente. |
| - | Usuario (si se requiere). |
| - | Contraseña (si se requiere). |
| (30/Jul/2006:17:02:03 +0100 | Fecha y hora de acceso |
| "GET/directorio/archivo.html HTTP/1.1" | Método de solicitud y archivo solicitado |
| 304 | ID de respuesta (2xx es OK, 3xx son de redirección, 4xx son de autenticación y 5xx son de errores de servicio). |
| - | Número de bytes enviados. |
| "http://www.dominio.com/director io/index.html" | Página desde la que se realiza la solicitud. |
| "Firefox (compatible; Mozilla 4.0 Linux Kernel 2.4)" | Información sobre la maquina cliente. |

En los sistemas de hipermedia adaptativa que implementan *log mining*, se suele filtrar el contenido del archivo *log*, a fin de eliminar información de poca relevancia (como etiquetas Web, solicitudes de archivos de imágenes y entradas de error), así como para reducir el tamaño del archivo hasta en un 60%³⁴.

³⁴ Panneso H y Murillo C. Aplicación de técnicas de Web mining para la búsqueda de patrones de comportamiento en clientes dentro de la Web. 2005. Trabajo de grado para optar al título de Magíster en Ciencias Computacionales. Convenio Instituto Tecnológico de estudios superiores de Monterrey Universidad Autónoma de Bucaramanga Universidad Autónoma de Occidente. Santiago de Cali, Colombia.

3.4.1 Aplicaciones

3.4.1.1 Herramientas analizadoras y generadoras de reportes. Estas herramientas, lo que generalmente hacen es tomar la información del log del servidor y a partir de ahí, generar informes, gráficos y resúmenes de estadísticas sobre tráfico y peticiones en un sitio Web. Son de gran utilidad para conocer el material más solicitado por los usuarios, horas de acceso, etc. A continuación se exponen algunos ejemplos de estas herramientas.

3.4.1.1.1 ANALOG³⁵. El ANALOG es un analizador de logs de distribución libre, que funciona sobre plataformas Windows, y que analiza logs de cualquier servidor Web. Genera informes con un alto grado de exactitud y aunque carece de interfaz gráfica de usuario (la aplicación se ejecuta en línea de comandos), genera reportes que se pueden visualizar en un navegador Web. Su mayor aporte se encuentra en la información detallada sobre tráfico y rendimiento del servidor. Genera informes que incluyen las páginas más visitadas e informe de peticiones, lo que lo convierte en una potente herramienta al implementar adaptatividad.

3.4.1.1.2 WEBALIZER. Herramienta también de software libre, que analiza archivos log del servidor. Ofrece un generador de informes en formato HTML configurable (a fin de eliminar información poco relevante), que pueden ser visualizados en un navegador Web estándar.

3.4.1.1.3 LOGROVER. LogRover es un sistema profesional para análisis de archivos log generados por servidores Web. Usa una base de datos en la que almacena los resultados de sus análisis, por lo que los archivos log (que, como

³⁵ Ibid

ya se ha dicho suelen ser muy grandes) solo necesitan ser leídos una sola vez. Ofrece una interfaz Web que permite generar y ver logs y estadísticas de sitios Web, analizados durante un determinado periodo de tiempo. En su sitio Web ofrece una versión Web de la aplicación, y se puede descargar gratis un *trial* que presta la mayoría de las funciones de la aplicación durante 30 días.

3.5 COOKIES³⁶

Una *cookie* es un fragmento de información que se almacena en el disco duro del visitante de una página Web a través de su navegador, a petición del servidor de la página. Esta información puede ser luego recuperada por el servidor en posteriores visitas. Las inventó Lou Montulli, un antiguo empleado de *Netscape Communications*. Originalmente, sólo podían ser almacenadas por petición de un CGI desde el servidor, pero *Netscape* dio a su lenguaje *JavaScript* la capacidad de introducir las directamente desde el cliente, sin necesidad de CGIs. Las *cookies* pueden ser borradas, aceptadas o bloqueadas según desee, para esto solo debe configurar convenientemente el navegador Web.

Las *cookies* son utilizadas habitualmente por los servidores Web para diferenciar usuarios y para actuar de diferente forma dependiendo del usuario. Las *cookies* se inventaron para ser utilizadas en una cesta de la compra virtual, que actúa como dispositivo virtual en el que el usuario va "almacenando" los elementos que desea adquirir, de forma que los usuarios pueden navegar por el sitio donde se muestran los objetos a la venta y añadirlos o eliminarlos de la cesta de compra en cualquier momento. Las *cookies* permiten que el contenido de la cesta de la compra dependa de las acciones del usuario.

³⁶ Colaboradores de Wikipedia, Cookies, <http://es.wikipedia.org/wiki/Cookie>. Pagina visitada por ultima vez el 30 de Septiembre de 2006.

Otro uso de las *cookies* es identificarse en un sitio Web. Los usuarios normalmente se identifican introduciendo sus credenciales en una página de validación; las *cookies* permiten al servidor saber que el usuario ya está validado, y por lo tanto se le puede permitir acceder a servicios o realizar operaciones que están restringidas a usuarios identificados.

Otros sitios Web utilizan las *cookies* para personalizar su aspecto según las preferencias del usuario. Los sitios que requieren identificación a menudo ofrecen esta característica, aunque también está presente en otros que no la requieren. La personalización incluye tanto presentación como funcionalidad. Por ejemplo, las páginas de *Wikipedia* permiten a los usuarios identificados elegir un estilo de presentación a su gusto; el motor de búsqueda de *Google* permite a los usuarios (incluso a los no registrados) decidir cuántos resultados de búsqueda quieren ver en cada página.

Técnicamente, las *cookies* son trozos de datos arbitrarios definidos por el servidor Web y enviados al navegador. El navegador los devuelve sin modificar al servidor, reflejando así un estado (memoria de eventos anteriores) en las transacciones HTTP, que de otra manera serían independientes de estado. Sin las *cookies*, cada petición de una página Web o un componente de una página Web sería un evento aislado, sin ninguna relación con el resto de peticiones de otras páginas del mismo sitio. Pero devolviendo una *cookie* al servidor Web, el navegador le proporciona un medio para relacionar la solicitud de la página actual con solicitudes de páginas anteriores. Además de ser definidas por un servidor Web, las *cookies* también pueden ser definidas por un script en un lenguaje como JavaScript, si éste está soportado y habilitado en el navegador Web.

Las especificaciones de *cookies* sugieren que los navegadores deben soportar un número mínimo de *cookies* o una cantidad mínima de memoria para almacenarlas. En concreto, se espera que un navegador sea capaz de

almacenar al menos 300 *cookies* de 4 *kilobytes* cada una y al menos 20 *cookies* por servidor o dominio.

El servidor que establece la *cookie* puede especificar una fecha de borrado, en cuyo caso la *cookie* será borrada en esa fecha. Un sitio de compras podría querer ayudar a clientes potenciales recordando las cosas que había en su cesta de la compra, incluso si cierran el navegador sin realizar la compra y vuelven más tarde, para evitar que tengan que buscar los productos de nuevo. En ese caso, el servidor crearía una *cookie* con fecha de borrado según el deseo del diseñador del sitio Web. Si no se define una fecha de borrado, la *cookie* es borrada cuando el usuario cierra su navegador. Por lo tanto, definir una fecha de borrado es una manera de hacer que la *cookie* sobreviva entre sesiones. Por esta razón, las *cookies* con fecha de borrado se llaman *persistentes*.

3.5.1 Ventajas de las cookies. Entre las mayores ventajas de las cookies se cuenta el hecho de ser almacenadas en el disco duro del usuario, liberando así al servidor de una importante sobrecarga. Es el propio cliente el que almacena la información y quien se la devolverá posteriormente al servidor cuando éste la solicite. Además, las cookies poseen una fecha de caducidad, que puede oscilar desde el tiempo que dure la sesión hasta una fecha futura especificada, a partir de la cual dejan de ser operativas.

3.5.2 Desventajas del uso de las cookies. Si se utiliza más de un navegador en un ordenador, cada uno tiene su propio almacenamiento de *cookies*. Por lo tanto, las *cookies* no identifican a una persona, sino a una combinación de cuenta de usuario, ordenador y navegador. De esta manera, cualquiera que utilice varias cuentas, varios ordenadores, o varios navegadores, tiene también múltiples conjuntos de cookies.

De la misma manera, las cookies no diferencian entre varias personas que utilicen el mismo ordenador o navegador, si éstos no utilizan diferentes cuentas de usuario.

3.6 MODELO VISTA CONTROLADOR

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.³⁷ La arquitectura MVC fue introducida como parte de la versión Smalltalk-80 del lenguaje de programación Smalltalk. Fue diseñada para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas³⁸.

Modelo: Ésta es la representación específica del dominio de la información sobre la cual funciona la aplicación. El modelo es otra forma de llamar a la capa de dominio. La lógica de dominio añade significado a los datos.

Vista: Esta presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.

³⁷ Colaboradores de Wikipedia. Modelo Vista Controlador, http://es.wikipedia.org/wiki/Modelo_Vista_Controlador. Visitada por ultima vez el 1 de Octubre de 2006

³⁸ Agustín Froufe, Tutorial de Java, http://www.cica.es/formacion/JavaTut/Apendice/arq_mvc.html. Visitada por ultima vez el 1 de Octubre de 2006.

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace).
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta dirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice.

5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

4. PRODUCTO FINAL

Dentro del marco de referencia se incluyen técnicas que son usadas actualmente como mecanismos de adaptatividad. Los conceptos incluidos relativos a *log mining* y *first click* hacen parte de una investigación que se realiza a manera de documentación para llegar a una técnica optima que permitiera generar un algoritmo con los resultados esperados dentro de lo que se busca como parte de el proyecto. La investigación de *log mining* permitió indagar acerca de la confiabilidad de la implementación de un algoritmo que usara esta técnica, dejando ver que pueden llegar a presentarse confusiones en la definición de perfiles de personas que acceden desde maquinas distintas pero a través del mismo *Proxy* ya que técnicamente, quien accede a los recursos es el *Proxy*. Al analizar el log del servidor de la universidad se detectó esta constante, que hace casi imposible identificar conexiones desde diferentes maquinas ya que la mayoría de accesos se hacen desde la misma universidad. Otra de las desventajas encontradas dentro del uso de *log mining* es el tamaño de los archivos *log*; el análisis del mismo ocupa demasiado espacio dentro de la aplicación lo que no permite la rápida ejecución tanto de la aplicaron como del algoritmo.

Siguiendo con las técnicas estudiadas, se encontró un patrón muy presente en sistemas de búsqueda de Internet conocido como *first click*. Esta técnica implementa la adaptación de un rol o perfil a partir del primer click que realiza el usuario dentro del área de trabajo. Basado en este principio básico de la técnica se encuentra una de las principales desventajas de la misma, ya que esta dada a presentar errores para dar un resultado realmente fiel alrededor del verdadero perfil del usuario. En el caso específico de la aplicación que se esta desarrollando se encontró que varios perfiles pueden tomar una conducta inicial muy parecida lo que da campo a no presentar los perfiles adecuados por

ejemplo tomando un caso específico si consideramos una persona que entra y su primer comportamiento está dado por un primer click en el link Financiación y becas es muy difícil diferenciar entre una persona de pregrado o un egresado que está buscando financiación para su especialización o diplomado. Lo anterior deja ver que la técnica no es confiable para realizar el prototipo debido a la dificultad para realizar las suposiciones.

Basado en lo anterior se creó un algoritmo que combinara diferentes técnicas y que arrojara resultados óptimos y confiables. Aplicando la arquitectura de sistemas adaptativos en donde se manejan tres capas de información (modelo de usuario, modelo de dominio, mecanismo de adaptación) se lograron crear reglas para decidir el perfil del usuario y la información que era pertinente para cada uno de ellos. También se decidió hacer manejo de cookies dentro del portal para en ellas guardar la información acerca del perfil del usuario en el portal. Cada vez que el usuario accede al portal se hace un estudio de la última visita, se evalúa la navegación realizada y se reasigna un perfil de acuerdo a esto o también si el usuario tuvo acceso al sitio nunca se asigna un perfil general que estudia el comportamiento para la próxima visita. En este orden se toma solo la última visita para almacenarla a fin que la consulta sea rápida y también debido a que un usuario puede tener varios perfiles o tomar perfiles aleatorios.

Dentro del desarrollo de la aplicación se decidió usar la arquitectura MVC debido a que esta es la usada por la universidad para el desarrollo de sus aplicaciones Web, además provee mecanismos ágiles que permiten hacer reducción en tiempo y esfuerzo debido a que cada una de sus capas son manejadas como entidades separadas lo que hace que cualquier cambio producido se refleje en las demás.

A continuación mostramos el desarrollo del proyecto basado en lo anteriormente mencionado.

4.1 MODELO DE USUARIOS

En esta sección se describirá la información concerniente al modelo de usuarios propio del sitio Web de la Universidad Autónoma de Bucaramanga UNAB donde se analizarán los siguientes ítems.

| | |
|----------------------|--|
| Nombre | Es el nombre con el que se identifica cada uno de los grupos de usuarios. |
| Descripción | Detalle de las características que identifican los usuarios que pertenecen a uno u otro rol. |
| Rango de edad | Edad o intervalo de edad en el que se encuentran los usuarios. |
| Material de interés | Información específica que clasifique como importante para un rol de usuario. |
| Frecuencia de acceso | Fechas ó intervalos de tiempo en los que estos usuarios acceden al sitio web. |

4.1.1 Estudiantes

Descripción: Este grupo se encuentra integrado por personas de las diferentes facultades de la universidad y que están cursando un programa académico, bien sea de pregrado o postgrado.

Rango edad: Personas entre 17 y 35 años.

Material de interés:

- Información institucional relacionada con el programa específico al que pertenecen.
- Información acerca de trámites académicos.
- Información sobre actividades de interés tanto académicas como culturales.

- Información de bienestar universitario.
- Contacto a docentes (páginas Web de docentes, correo electrónico de los mismos).

Frecuencia de acceso: Diaria.

4.1.2 Aspirantes

Descripción: Grupo de personas de la región y el país, que se encuentren interesados en acceder a cualquier programa de educación de la universidad, se consideran también aspirantes los padres de familia que estén interesados en tener información sobre la carrera por la que sus hijos optan. En su gran mayoría, este grupo de usuarios son estudiantes del último grado de educación secundaria, aunque también se encuentran profesionales que deseen iniciar especializaciones o maestrías.

Rango de edad: Personas entre 17 y 50 años

Material de interés:

- Información institucional.
- Descripción de carreras.
- Catalogo de cursos por carreras.
- Tour virtual por las instalaciones de la universidad.
- Modalidades de estudio.

- A distancia.
- Tecnologías.
- Presenciales.
- Nocturnas.
- Virtual.
- Especializaciones.
- Proceso de admisión.
- Canales de financiación.
- Transferencias y homologaciones.
- Valores de matrícula.
- Convenios con otras instituciones.

Frecuencia de acceso: Periodos previos a inicios de los programas académicos.

4.1.3 Empresarios

Descripción: Personas pertenecientes a empresas que actúan en el campo laboral. También personas que estén interesadas en de alguna forma realizar contactos para diferentes negocios.

Rango de edad: Personas entre 30 y 50 años

Material de interés:

- Información sobre educación informal a fin de capacitar empleados ó mejorar la productividad.
- Programas de postgrado.
- Practicas profesionales.
- Pasantías.
- Muestras empresariales.
- Gestión de calidad.
- Programas de asesorías empresariales

Frecuencia de acceso: Esporádica

4.1.4 Egresados

Descripción: Grupo de personas que fueron estudiantes de la Universidad Autónoma de Bucaramanga y que recibieron grado de alguno de los programas ofrecidos. Estas personas están interesadas en continuar su educación con programas especializados en la universidad bien sea de postrado o maestría como también diplomados.

Rango de edad: Personas ente los 27 y 50 años de edad.

Material de interés:

- Programas de especialización.
- Programas de maestría.
- Seminarios de actualización.
- Vínculos con relaciones internacionales para becas.
- Investigación UNAB.

Frecuencia de acceso: Esporádica.

4.2 REGLAS DE ADAPTATIVIDAD DE NAVEGACIÓN

De acuerdo a la selección de material que hace un usuario en las páginas del sitio Web de la universidad, se puede intuir el tipo de usuario que accede a la información publicada en dicha página (Estos datos son tomados de los *clicks* realizados en el menú izquierdo de la página), estas son algunas de las reglas de asociación que relaciona una petición con un rol.

| MATERIAL DE INTERES | TIPO DE USUARIO |
|--|-------------------|
| <p>En el menú Conoce la UNAB:</p> <p>Mensaje del Rector.</p> <p>Presentación institucional.</p> <p>Gestión de calidad.</p> <p>Relaciones nacionales e internacionales.</p> <p>Biblioteca.</p> <p>Bienestar universitario.</p> <p>Escuelas, facultades y departamentos.</p> <p>Proyección social.</p> <p>Campus.</p> <p>En el menú Estudia en la UNAB:</p> <p>Sub-menú Programas por modalidad :</p> <p>Presencial.</p> <p>A distancia virtual.</p> <p>Dual.</p> <p>Sub-menú Programas por nivel de formación :</p> <p>Profesionales.</p> <p>Tecnológicos.</p> <p>Especializaciones.</p> <p>Maestrías.</p> <p>Técnicos laborales.</p> <p>Diplomados y seminarios.</p> <p>Admisiones.</p> <p>Financiación y becas.</p> | <p>Aspirantes</p> |

| | |
|---|--------------------|
| <p>En el menú Investigación Investigación en la UNAB. Ciencias básicas, naturales e ingeniería. Ciencias biomédicas. Ciencias de la educación Ciencias económicas, administrativas y contables Ciencias políticas Ciencias sociales Psicología Sociojurídicas.</p> | <p>Estudiantes</p> |
| <p>En el menú Estudia en la UNAB: Sub-menú Programas por nivel de formación Especializaciones Maestrías Diplomados y seminarios</p> | <p>Egresados</p> |
| <p>En el menú Estudia en la UNAB: Sub-menú Programas por nivel de formación : Diplomados y seminarios. En el menú Conoce la UNAB: Gestión de calidad. Proyección social.</p> | <p>Empresarios</p> |
| <p>En el menú Conoce la UNAB: Relaciones nacionales e internacionales. Biblioteca. Bienestar universitario. Escuelas, facultades y departamentos.</p> | <p>Estudiantes</p> |

4.3 LOG DEL SERVIDOR DE LA UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA

En el fragmento siguiente (El archivo log completo tiene un tamaño de 1 GB, pero se puede separar en archivos mas pequeños, llegando a archivos de 4MB que contienen 31.000 registros) perteneciente a un log del servidor que hospeda el sitio Web institucional de la Universidad Autónoma de Bucaramanga, se observa que la dirección IP de la maquina cliente es igual durante un largo lapso de tiempo, lo que deja ver el principal problema de la utilización de log mining. Este problema consiste en que cuando un usuario o un grupo de usuarios se conectan a través de un Proxy, la identificación es la misma para todos en el log del servidor. En caso de que se trate de un único usuario, es imposible determinar automáticamente en que momento deja el servidor, por lo que se debe realizar un proceso conocido como “sesionar” (del ingles sessionizing), que consiste en realizar supuestos sobre la información en el log, a fin de crear sesiones de usuario y determinar las acciones realizadas por el mismo, para poder realizar un correcto filtrado de la información.

```
200.69.124.43 - - [11/Jul/2006:07:38:45 -0500] "GET /pls/portal
HTTP/1.1" 302 5 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
5.1; SV1)"
200.69.124.43 - - [11/Jul/2006:07:38:48 -0500] "GET
/pls/portal/portal.home HTTP/1.1" 302 5 "-" "Mozilla/4.0 (compatible;
MSIE 6.0; Windows NT 5.1; SV1)"
200.69.124.43 - - [11/Jul/2006:07:38:48 -0500] "GET
/portal/page?_pageid=6,1,6_13&_dad=portal&_schema=PORTAL HTTP/1.1" 200
31736 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)"
200.69.124.43 - - [11/Jul/2006:07:38:49 -0500] "GET
/pls/portal/PORTAL.wwpob_page_util.select_tabs_for_persistence?p_conta
inerid=1&p_siteid=6&p_mode=3&p_tabstring=6_13 HTTP/1.1" 200 55
"http://caribdis.unab.edu.co/portal/page?_pageid=6,1,6_13&_dad=portal&
_schema=PORTAL" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1)"
```

```

200.69.124.43      -      -      [11/Jul/2006:07:38:50      -0500]      "GET
/pls/portal/PORTAL.wwsec_app_priv.login?p_requested_url=PORTAL.home&p_
cancel_url=PORTAL.home      HTTP/1.1"      302      5
"http://caribdis.unab.edu.co/portal/page?_pageid=6,1,6_13&_dad=portal&
_schema=PORTAL" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1)"
200.69.124.43      -      -      [11/Jul/2006:07:38:57      -0500]      "GET
/pls/portal/portal.wwsec_app_priv.process_signon?urlc=v1.2~C9167B6DCA6
830E62C7AC2BE332136864285B1380ABA31DFF938A72FB129E8C6C7F7FC0EDC61F4D8C
EF10C0572B2784FD5422CFDC58BA8EF9E36FE675A86600A91F9349E967AEDBDB2161
75A3952812E786E1142A18E432488C35D3635D253A2556E95CCFDFAF70D57EB8912DBB
7036998B6F86A9EF17933093F4EA21D1E139A466EDFFEF2AA991477599DE1869792421
D77A470A364254CFDE9D9338B536AE1EA754D6AF36D27A36E76B61C449C32FB0E8DCA8
BC1B1560B56E54191B485E830024E2AD0F00EC9C55E51DAE57BCD974037C9195217599
2A2876C20CD7678152F594A9BADAFBAF804271208D6CE545D3851964DA312D868B0A04
27A7D0C3C5E015235C89AB4B324E28C019F160BF49068E0DC459D24C7B9BCC76314ED9
C840E      HTTP/1.1"      302      5
"http://hades.unab.edu.co:7777/sso/jsp/login.jsp?site2pstoretoken=v1.2
~3A8566B9~E45CD047F05F2480160A76500B2A374654B246C09F5EB55AD858C37F2732
A082D74673CF938E128A1122F93EDCEF581CDB534F0B74B9401408F674EEBC821952FB
97085067CBE77EC3B355933AD63F95EDCB351EF47435953C3D4E7F7596AEF89CC1383A
FA896BA0788D52F3AA53B440B85AD04771C9845ECF9F4C0D8339983B4017376690D784
3BCE29728BC48526C6511FBED4C16EB0D97BBB17E222574F73EE8DBFC7EBFA57B6AAF7
8B748F0C13B925E0E8C2C3025FBCFA9E1B02590161FFA2877CA176886CAF4A76E0ECC1
0A0803&p_error_code=&p_submit_url=http%3A%2F%2Fhades.unab.edu.co%3A777
7%2Fsso%2Fauth&p_cancel_url=http%3A%2F%2Fcaribdis.unab.edu.co%2Fpls%2F
portal%2FPORTAL.home&ssusername=" "Mozilla/4.0 (compatible; MSIE 6.0;
Windows NT 5.1; SV1)"
200.69.124.43      -      -      [11/Jul/2006:07:38:57      -0500]      "GET
/portal/page?_pageid=6,1,6_13&_dad=portal&_schema=PORTAL HTTP/1.1" 200
33174
"http://hades.unab.edu.co:7777/sso/jsp/login.jsp?site2pstoretoken=v1.2
~3A8566B9~E45CD047F05F2480160A76500B2A374654B246C09F5EB55AD858C37F2732
A082D74673CF938E128A1122F93EDCEF581CDB534F0B74B9401408F674EEBC821952FB
97085067CBE77EC3B355933AD63F95EDCB351EF47435953C3D4E7F7596AEF89CC1383A
FA896BA0788D52F3AA53B440B85AD04771C9845ECF9F4C0D8339983B4017376690D784
3BCE29728BC48526C6511FBED4C16EB0D97BBB17E222574F73EE8DBFC7EBFA57B6AAF7
8B748F0C13B925E0E8C2C3025FBCFA9E1B02590161FFA2877CA176886CAF4A76E0ECC1
0A0803&p_error_code=&p_submit_url=http%3A%2F%2Fhades.unab.edu.co%3A777

```

```

7%2Fsso%2Fauth&p_cancel_url=http%3A%2F%2Fcaribdis.unab.edu.co%2Fpls%2F
portal%2FPORTAL.home&ssusername=" "Mozilla/4.0 (compatible; MSIE 6.0;
Windows NT 5.1; SV1)"
200.69.124.43      -      -      [11/Jul/2006:07:38:57      -0500]      "GET
/pls/portal/PORTAL.wwpob_page_util.select_tabs_for_persistence?p_conta
inerid=1&p_siteid=6&p_mode=3&p_tabstring=6_13      HTTP/1.1"      200      55
"http://caribdis.unab.edu.co/portal/page?_pageid=6,1,6_13&_dad=portal&
_schema=PORTAL" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1)"
200.69.124.43      -      -      [11/Jul/2006:07:39:08      -0500]      "GET
/pls/portal/PORTAL.wwexp_explore.explore      HTTP/1.1"      302      5
"http://caribdis.unab.edu.co/portal/page?_pageid=6,1,6_13&_dad=portal&
_schema=PORTAL" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1)"
200.69.124.43      -      -      [11/Jul/2006:07:39:08      -0500]      "POST
/pls/portal/!PORTAL.wwpob_page.show?_pageid=6,3      HTTP/1.1"      302      5      "-"
"Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1) RPT-
HTTPClient/0.3-3"
200.69.124.43      -      -      [11/Jul/2006:07:39:08      -0500]      "GET
/portal/page?_pageid=6,3&_dad=portal&_schema=PORTAL      HTTP/1.1"      302      0
"http://caribdis.unab.edu.co/portal/page?_pageid=6,1,6_13&_dad=portal&
_schema=PORTAL" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1)"
200.69.124.43      -      -      [11/Jul/2006:07:39:10      -0500]      "GET
/portal/page?_pageid=6,3,6_23&_dad=portal&_schema=PORTAL      HTTP/1.1"      200
27169
"http://caribdis.unab.edu.co/portal/page?_pageid=6,1,6_13&_dad=portal&
_schema=PORTAL" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1)"
200.69.124.43      -      -      [11/Jul/2006:07:39:10      -0500]      "POST
/pls/portal/!PORTAL.wwpro_app_provider.execute_portlet/89765307/15
HTTP/1.1"      200      16435      "-"      "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
5.1; SV1) RPT-HTTPClient/0.3-3 RPT-HTTPClient/0.3-3"
200.69.124.43      -      -      [11/Jul/2006:07:39:10      -0500]      "GET
/pls/portal/PORTAL.wwpob_page_util.select_tabs_for_persistence?p_conta
inerid=3&p_siteid=6&p_mode=3&p_tabstring=6_23      HTTP/1.1"      200      55
"http://caribdis.unab.edu.co/portal/page?_pageid=6,3,6_23&_dad=portal&
_schema=PORTAL" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1)"

```

```
200.69.124.43 - - [11/Jul/2006:07:39:13 -0500] "POST
/pls/portal/PORTAL.wwexp_api_engine.other_actions HTTP/1.1" 302 5
"http://caribdis.unab.edu.co/portal/page?_pageid=6,3,6_23&_dad=portal&
_schema=PORTAL" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1) "
```

4.4 MECANISMO DE ADAPTATIVIDAD

El mecanismo seleccionado para llevar a cabo la implementación de la adaptatividad será el de la utilización de *cookies*. Este método consiste en almacenar en una *cookie* la información necesaria para construir la información adaptada al perfil de usuario, en nuestro caso un identificador de usuario que se cotejaría con la información almacenada en una base de datos, a fin de obtener este perfil de usuario.

4.4.1 Ventajas y desventajas de las técnicas de adaptatividad estudiadas: Como ya se dijo, la implementación de otras técnicas de adaptatividad como las de *First Click* y *Log Mining*, necesitan de otros algoritmos auxiliares como de predicción y *sesionar*, respectivamente. Además de otros problemas expuestos también con anterioridad, como la falta de consideración en caso de solicitudes erróneas por parte de los usuarios en el caso de *first click*, y la incapacidad de detectar conexiones a través de servidores *Proxy*, en caso de *Log mining*.

La utilización de cookies, presenta como principal desventaja el hecho de la pérdida de la adaptatividad cada vez que se pierda la *Cookie*, cosa que sucede cuando se eliminan los archivos temporales de la máquina cliente.

Una de las ventajas de la utilización de *Cookies* es que existe una implementación en JSP de una clase que facilita la manipulación de esta información.

4.5 LA CLASE COOKIE

Esta clase pertenece al paquete `javax.servlet.http` y tiene como constructor `Cookie(java.lang.String Nombre, java.lang.String Valor)`, donde `Nombre` es el identificador del dato a almacenar y `Valor` es el dato propiamente dicho, por ejemplo el método

```
Cookie myCookie = new Cookie("Nombre", "Pedro");
```

Crearía una *Cookie* que contendría el identificador `Nombre` junto con el valor `Pedro`.

Métodos de la clase:

| MÉTODO | DESCRIPCIÓN |
|---------------------------------|--|
| <code>getComment()</code> | Retorna una cadena de caracteres que describe el propósito de la <i>Cookie</i> o <code>null</code> en caso de que no se haya definido un comentario. |
| <code>getMaxAge()</code> | Retorna la máxima duración especificada de la <i>Cookie</i> . |
| <code>getName()</code> | Retorna el nombre de la <i>Cookie</i> . |
| <code>getPath()</code> | Retorna el prefijo de todos los URL para los que la |
| <code>getValue()</code> | Retorna el valor de la <i>Cookie</i> . |
| <code>setComment(String)</code> | Especifica el propósito o describe la <i>Cookie</i> . |
| <code>setMaxAge(int)</code> | Especifica la duración máxima (en segundos) de la <i>Cookie</i> . |

| | |
|------------------|---|
| setValue(String) | Especifica el valor de la <i>Cookie</i> . |
|------------------|---|

4.5.1 Script generador de una cookie. El siguiente *script* de una pagina Web JSP, crea una *cookie* identificada con la etiqueta `idUsuario` y que contiene un identificador que posteriormente será almacenado en la base de datos, relacionándolo con un perfil de usuario.

```
<%@ page language="java" import="java.util.*"%>
<%
Cookie Usuario = new Cookie ("idUsuario",Userid);
response.addCookie (Usuario);
%>
```

4.5.2 Script para capturar información de una cookie. El siguiente *script* obtiene el valor identificado como `idUsuario` almacenado en una *Cookie* del sitio Web al que pertenece la pagina que implementa el *script*.

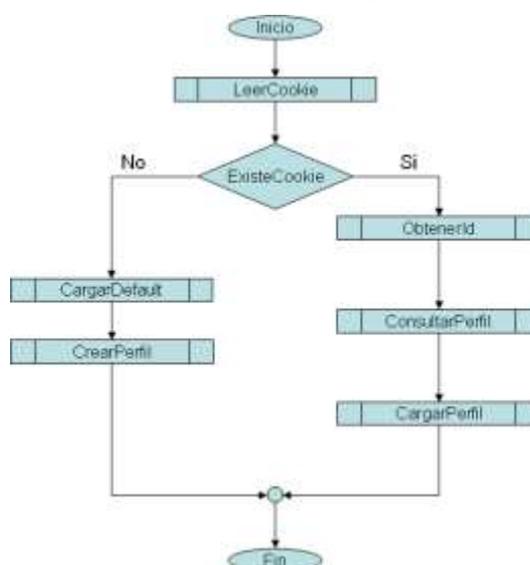
```
<%@ page language="java" %>
<%
Cookie AllCookies [] = request.getCookies();
Cookie myCookie = null;
if (AllCookies != null) {
    for (int i = 0;i<AllCookies.length;i++) {
        if (AllCookies[i].getName().equals("idUsuario")) {
            myCookie = cookies[i];
            break;}
    }
}
```

Con lo que se obtiene un Objeto `myCookie`, que será la referencia a la información de nuestro interés. Basta con ejecutar el método `myCookie.getValue()`; para acceder a la información almacenada.

4.6 ALGORITMO DE CONSTRUCCIÓN DE LA INFORMACIÓN ADAPTADA DE ACUERDO AL PERFIL DE USUARIO

La información que automáticamente será adaptada al usuario, será los accesos que se encuentran el menú principal de la página de la Universidad Autónoma de Bucaramanga, ubicado en la sección izquierda de la misma y para lo que se implementará el siguiente algoritmo.

Figura 1. Diagrama de flujo de datos Algoritmo para cargar perfil.



Fuente: Autores

4.6.1 Pseudocódigo

Inicio

LeerCookie

Si ExisteCookie entonces:

ObtenerId

ConsultarPerfil

CargarPerfil

DOM:

CargarDefault

CrearPerfil

Fin

4.6.1.1 Detalle de procedimientos

LeerCookie

Verificar que existe el archivo que contiene la *cookie*.

ObtenerId

Obtiene el valor almacenado en la *cookie*.

ConsultarPerfil

Obtiene el perfil asociado a un usuario específico identificado con el *id* adquirido en la *cookie*.

CargarPerfil

Adapta la información dependiendo del perfil.

CargarDefault

Debido a que el usuario no se encuentra registrado en nuestra base de datos, se le mostrara información de carácter general.

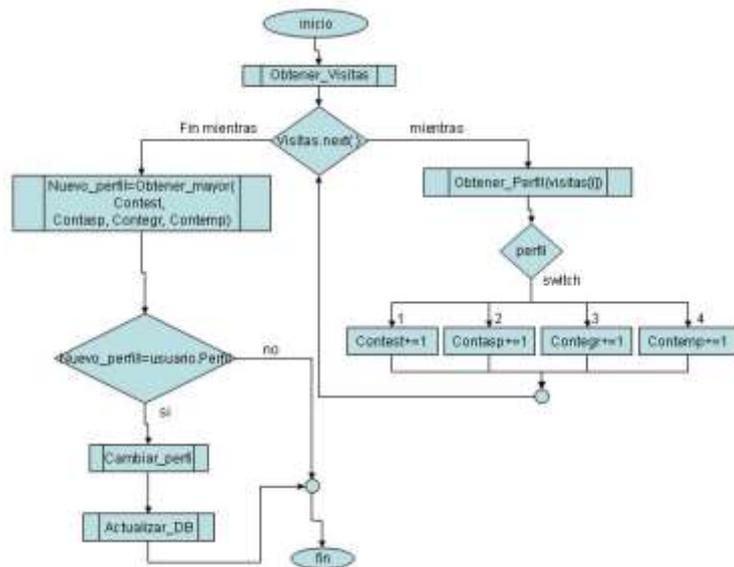
CrearPerfil

Una vez que se muestre al usuario información de carácter general, se almacenaran tanto en una *cookie* como en la base de datos, las solicitudes que se hagan los materiales publicados, esta información, será la que permita deducir el perfil de usuario al que pertenece este visitante. El perfil de usuario será almacenado en la *cookie* que contiene el perfil, para que la próxima vez que el usuario visite la página, obtenga un menú adaptado de acuerdo a su anterior visita.

4.7 REASIGNACIÓN DE PERFIL DE USUARIO DE ACUERDO A LAS VISITAS HECHAS

Cuando el usuario ingresa al portal se debe hacer una revisión del perfil de acuerdo a las visitas que este ha hecho anteriormente. Para esto se creo un algoritmo que hace este estudio y hace una reasignación si es necesario del perfil. A continuación se muestra el algoritmo:

Figura 2. Diagrama de flujo de datos Algoritmo para recalcular el perfil.



Fuente: Autores

4.7.1 Detalle de procedimientos

Obtener_visitas

Se toman de la base de datos las visitas realizadas por el usuario y se guardan en un arreglo, donde después se evaluarán para asignarle a cada una de ellas un perfil. Dentro de un switch se va llevando la cuenta de cada perfil cuantas páginas visitadas tiene.

Obtener_mayor(Contest, Contasp, Contegr, Contemp)

Se evalúan los resultados de los contadores que quedan de las páginas con el fin de obtener el perfil con el mayor número de visitas.

Obtener_perfil

El perfil con el mayor número de visitas dentro del arreglo será el perfil a asignar al usuario del portal.

Cambiar_Perfil

Se hace una comparación del perfil actual que tiene el usuario con el nuevo arrojado de la función *Obtener_perfil*, si este es diferente se asignara el nuevo si son iguales continuara de la misma forma.

Actualizar_DB

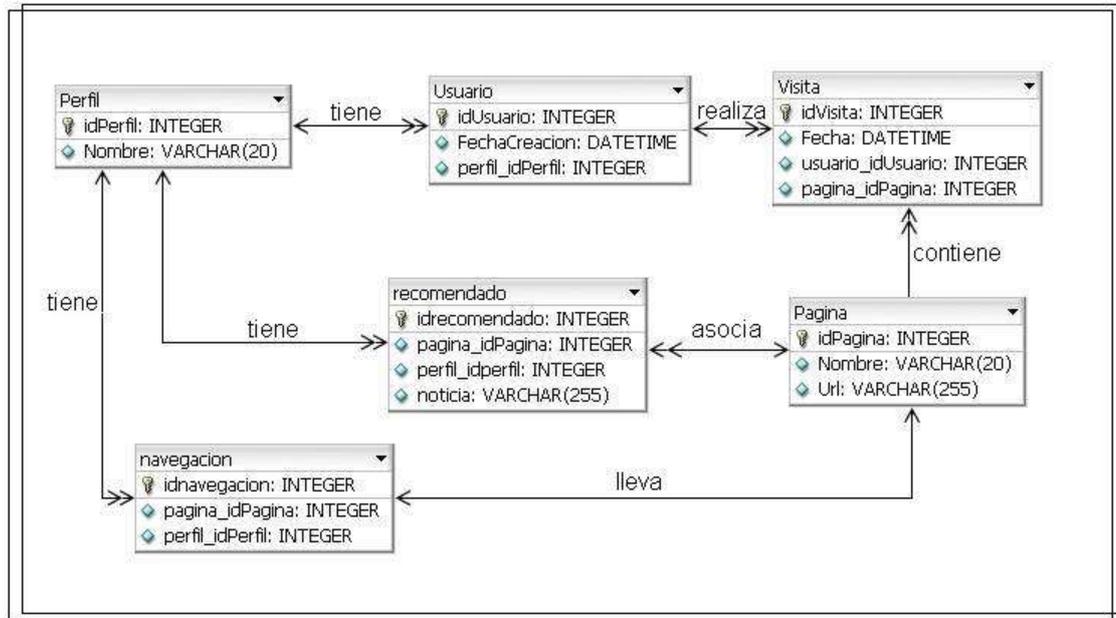
Si el perfil debe ser actualizado por que es diferente al que existía se hará un UPDATE en la base de datos para que el perfil este disponible para la próxima vez que el usuario entre.

4.8 BASE DE DATOS

También se implementa una pequeña base de datos que almacenará datos sobre las visitas a la página principal del sitio Web de la Universidad Autónoma de Bucaramanga, lo que permitirá de manera sencilla obtener información sobre la actividad en el mismo.

4.8.1 Modelo ER

Figura 3. Modelo Entidad – Relación de la base de datos



Fuente: Autores

4.8.2 Listado de consultas

4.8.2.1 Perfil

- Agregar perfil

Esta consulta crea un nuevo perfil en la base de datos

- Editar perfil

Se modifican las creencias del perfil y sus características

- Consultar

Se hace una consulta del estado actual del perfil

- Listar

Se muestran los diferentes perfiles

4.8.2.2 Visita

- Página más visitada

Se lista la página más visitada por cada perfil recaudado de la información de cada usuario perteneciente a un perfil determinado.

Hora de mayor demanda (periodos de tiempo)

Se trata de establecer en que periodos de tiempo cada perfil está teniendo más visitas al portal y a su información.

- Perfil más activo

Establece desde que perfil los usuarios del portal están accediendo, es decir de que forma se comportan y cual de ellos es el más usado.

- Perfil menos activo

Establece que perfil está siendo menos usado por las personas que acceden al portal.

4.8.2.3 Usuario

- Crear usuario

Se crea un usuario en la base de datos en donde se va a guardar el id y además las páginas visitadas para establecer el perfil.

- Actualizar usuario

Se revisa las últimas visitas del usuario con el fin de establecer si estas siguen perteneciendo al perfil establecido o de lo contrario se modifica el perfil buscando el más adecuado a sus necesidades.

5. CONCLUSIONES

El desarrollo de aplicaciones Web adaptativas es un campo nuevo que se está comenzando a explorar y a tener una implementación bastante amplia pero no tiene algoritmos ni procedimientos establecidos que estandaricen el funcionamiento de los mismos. Existen portales que actualmente utilizan técnicas de adaptatividad que mediante inicio de sesiones mantienen sus preferencias con respecto a apariencia y diseño de las páginas. También existen otros sistemas que hacen búsqueda basados en las preferencias de los usuarios, a fin de obtener una mayor eficiencia en los resultados mostrados al usuario. El portal Web de la universidad Autónoma de Bucaramanga maneja mucha información que en algunos casos se encuentra repetida o el usuario no encuentra debido a la complejidad de la navegación por el sitio.

Por esta razón el proyecto permitió estudiar las técnicas que se implementan en diferentes campos y que sirven como apoyo para la construcción de portales adaptativos. El portal Web adaptativo construido para la Universidad Autónoma de Bucaramanga fue desarrollado bajo diferentes técnicas que permitieron lograr el objetivo propuesto en un comienzo. Como primer paso se trabajó en un modelo hipermedia adaptativo de tres capas en las que se visualizan cada uno de los perfiles presentes en el portal y sus creencias aquellas que están relacionadas con el interés de cada uno de ellos. También se implementó una técnica por la cual se pudiera acceder al perfil adecuado de acuerdo al usuario. En este se utilizaron cookies como técnica de almacenamiento, mantenimiento y actualización de perfiles. En las cookies se almacenan las páginas visitadas por cada usuario y basado en ello se hace un estudio mediante un algoritmo para detectar el perfil del usuario que accede a la página.

Por medio de estas técnicas se busco hacerle al usuario una navegación mucho mas agradable en donde se le recuerden las ultimas paginas visitadas, links de interés y noticias relacionadas con su perfil, con el fin que siempre encuentre información que sea pertinente y que además su interfaz se haga agradable debido a la interacción con el usuario acerca de sus preferencias.

6. TRABAJOS FUTUROS

En base a la labor realizada durante este proyecto, se generan cimientos para la construcción de trabajos posteriores como el análisis de la información contenida en la base de datos, lo que permitiría incluir dinamismo al sitio Web de la UNAB, pues a partir de consultas mas precisas y especificas se puede obtener información sobre el material publicado con mayor solicitud, igualmente sobre la actividad de los perfiles y crear a partir de allí un modelo de información cambiante para publicar en el sitio y mejorar la experiencia de los visitantes.

También es posible, posterior a la implementación de la adaptatividad, enriquecer de manera dinámica el modelo de usuarios, bien sea incluyendo o excluyendo perfiles de usuario, dependiendo de la información recolectada de la base de datos.

BIBLIOGRAFÍA

ADOMAVICIUS G. y TUZHILIN A. Personalization Technologies: A process-oriented perspective. Communications of the ACM. Volume 48, Issue 10 (Octubre 2005).

BARRERA-SANABRIA G., ARENAS-SELEEY D., García-Ojeda Juan C. y Méndez-Ortiz F. Designing Adaptive Educational Web Sites: General Framework. En: IEEE Computer Society. 2004.

BRUSILOVSKY, P. Adaptive and Intelligent Technologies for Web-based Education, In: Künstliche Intelligenz, Special Issue on Intelligent System and Teleaching, 19-25 /4, 1999.

CACHERO C., GARRIGOS I. y GOMEZ J. Personalización de aplicaciones en OO-H. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante. España. <http://www.dlsi.ua.es/~ccachero/papers/ideas02.pdf>

_____ Tratamiento de Personalización dinámica en modelos conceptuales de aplicaciones Web. IWAD Group. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante.
<http://www.dlsi.ua.es/~igarrigos/art/jisbd03.pdf>

CHURCH K., KEANE M. y SMYTH B. The First Click is the Deepest: Assessing Information Scent Predictions for a Personalized Search Engine. 2001. Adaptive Information Cluster, Department of Computer Science, University College Dublin, Irlanda.

GOODWIN C., SCHWARTZ M. y NILSEN J. Usability of Intranet Portals a Report From the Trenches. En: Experiences From Real-Life Portal Projects. Segunda edición. Nielsen Norman Group. <http://www.nngroup.com/reports/intranet/portals/>

JIAN-CHIH, O., LEE, C. y CHEN M. Web Log Mining with Adaptive Support Thresholds. 2004. Department of Electrical Engineering. National Taiwan University Taipei, Taiwan.

KAMDAR T. y JOSHI A. On creating adaptive Web servers using weblog mining. 2000. Department of Computer Science and Electrical Engineering. University of Maryland.

KORKEA M. Context-Aware Applications Survey. Department of Computer Science. Helsinki University of Technology. Publicado el 25 de abril de 2000. <http://users.tkk.fi/~mkorkeaa/doc/context-aware.html>

KOUTRI M, AVOURIS N. y DASKALAKI S. A survey on Web usage mining techniques for web-based adaptive hypermedia systems. 2002. University of Patras, Grecia.

LOPEZ-JAQUERO V., MONTERO F., FERNANDEZ A. y Lozano M. Adaptabilidad de Interfaces de Usuario por Reflexión. 2.003. Grupo de Investigación LoUISE, Universidad de Castilla, Albacete, España.

MACIAS J. y CASTELLS P. Diseño interactivo de cursos adaptativos. 2002. E.T.S. de Informática, Universidad Autónoma de Madrid.

MEDINA-MEDINA N., GARCIA-CABRERA L., RODRIGUEZ-FORTIZ M. y PARETS-LLORCA J. Adaptación al Usuario en Sistemas Hipermedia: El Modelo SEM-HP. 2.004. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Granada. España.

NANOPOULOS A., KATSAROS D. y MANOLOPOULOS Y. Exploiting Web Log Mining for Web Cache Enhancement. 2001. Department of Informatics, Aristotle University, Grecia.

O'BRIEN M., KEANE M. y SMITH B. Predictive Modeling of First-Click Behavior in Web-Search. 2001. Adaptive information cluster, School of computer science and informatics. University College Dublin, Irlanda.

PANNESO H y MURILLO C. Aplicación de técnicas de Web mining para la búsqueda de patrones de comportamiento en clientes dentro de la Web. 2005. Trabajo de grado para optar al título de Magíster en Ciencias Computacionales. Convenio Instituto Tecnológico de estudios superiores de Monterrey Universidad Autónoma de Bucaramanga Universidad Autónoma de Occidente. Santiago de Cali, Colombia.

QIANG YANG, HUI WANG y WEI ZHANG. Web-log Mining for Quantitative Temporal-Event Prediction. 2003. Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong, China.

RODRIGUEZ PAULO, de BRITO SILVANA, FONSECA Orlando y Jose Lima. Adaptive interfaces to virtual learning communities. Programa de Pós-Graduação em Engenharia Elétrica (PPGEE).

ROJAS Roger E. Aplicando técnicas de Minería de Datos en el desarrollo de un sitio Web adaptativo. 2.004. Trabajo de grado (Ingeniero de Sistemas) - Universidad Autónoma de Bucaramanga, UNAB. Escuela de Ciencias Naturales e Ingenierías. Facultad de Ingeniería de Sistemas, Bucaramanga, Colombia.

SANCHEZ VILLALOBOS F. Modelado del usuario para ambientes de aprendizaje colaborativo en Internet. 2.000. Tesis profesional presentada como requisito parcial para obtener el título en Maestría en Ciencias con Especialidad en Ingeniería en Sistemas Computacionales. Universidad de las Américas, Puebla, México.

VICARI, R. Introdução aos Sistemas Tutores Inteligentes. Aulas do programa de pósgraduação em ciência da computação, UFRGS, 2000.

Colaboradores de Wikipedia. *Java Server Pages*,
http://es.wikipedia.org/w/index.php?title=Java_Server_Pages&oldid=12006064.
Página visitada el 1 de abril de 2006.

Colaboradores de Wikipedia, *PHP*,

<http://es.wikipedia.org/w/index.php?title=PHP&oldid=12019741>. Página visitada el 5 de febrero de 2006.

Colaboradores de Wikipedia. *Portal (Internet)*,

http://es.wikipedia.org/w/index.php?title=Portal_%28Internet%29&oldid=11788430. Página visitada el 3 de abril de 2.006.

Colaboradores de Wikipedia. *Java Servlet*,

http://es.wikipedia.org/w/index.php?title=Java_Servlet&oldid=11902527. Pagina visitada el 1 de abril de 2006.

Cingler M. Desarrollo de aplicaciones y servicios web.

http://glud.udistrital.edu.co/glud/areas/doc/articulos/1_articulo_ws/aplicaciones-Web.html. Pagina visitada el 5 de febrero de 2006.

Sun Microsystems, JavaServer Pages Technology.

<http://java.sun.com/products/jsp/>. Página visitada el 5 de febrero de 2006.

Stephanie Bodoff, Java Servlet Technology,

http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/Servlets.html. Página visitada el 8 de febrero de 2006.

Michael Lerner Productions, Estructura de una Página Web

<http://www.learnthenet.com/Spanish/html/14wbpganat.htm>. Página visitada el 3 de abril de 2.006.

Michael Lerner Productions, Navegación a través de hipervínculos.

<http://www.learnthenet.com/Spanish/html/18wbpganat.htm>. Página visitada el 3 de abril de 2.006.

Master-net, La personalización cada vez más popular,

<http://www.masterdisseny.com/master-net/atrasadas/20.php3>. Página visitada el 5 de febrero de 2006.

Microsoft, Centro de desarrollo de aplicaciones ASP .NET,

http://www.microsoft.com/spanish/msdn/centro_recursos/asp_net/default.msp.
Página visitada el 5 de febrero de 2006.

Juan Antonio Breña Moral, Introducción a .NET,

<http://www.monografias.com/trabajos11/winnet/winnet.shtml>. Página visitada el 6 de febrero de 2006.

José Valle, Bases de Datos, <http://www.monografias.com/trabajos24/bases-de-datos/bases-de-datos.shtml>. Página visitada el 1 de abril de 2006.

Manual de referencia, <http://www.php.net/>. Página visitada el 1 de abril de 2006.

Colaboradores de Wikipedia. *Modelo Vista Controlador*,

http://es.wikipedia.org/w/index.php?title=Modelo_Vista_Controlador&oldid=11911428, Visitada por ultima vez el 1 de Octubre de 2006

Agustín Froufe, Tutorial de Java,

http://www.cica.es/formacion/JavaTut/Apendice/arq_mvc.html. Visitada por

ultima vez el 1 de Octubre de 2006

ANEXOS

Anexo A. Diccionario de datos

Navegación

| Campo | Tipo | Nulo | Predeterminado |
|---------------------|---------|------|----------------|
| <u>idNavegacion</u> | Int(11) | No | 0 |
| pagina_idPagina | Int(11) | No | 0 |
| perfil_idperfil | Int(11) | No | 0 |

Pagina

| Campo | Tipo | Nulo | Predeterminado |
|-----------------|--------------|------|----------------|
| <u>idPagina</u> | int(11) | No | |
| Nombre | Varchar(55) | No | |
| Url | Varchar(255) | No | |

Perfil

| Campo | Tipo | Nulo | Predeterminado |
|-----------------|-------------|------|----------------|
| <u>idPerfil</u> | int(10) | No | |
| Nombre | varchar(20) | Sí | <i>NULL</i> |

Recomendado

| Campo | Tipo | Nulo | Predeterminado |
|----------------------|--------------|------|----------------|
| <u>idRecomendado</u> | int(10) | No | |
| pagina_idPagina | int(10) | No | 0 |
| perfil_idPerfil | int(10) | No | 0 |
| noticia | varchar(255) | No | |

Usuario

| Campo | Tipo | Nulo | Predeterminado |
|------------------|-------------|------|----------------|
| <u>idUsuario</u> | Int(10) | No | |
| Perfil_idPerfil | Int(10) | No | 0 |
| fechacreacion | varchar(19) | No | |

Visita

| Campo | Tipo | Nulo | Predeterminado |
|-------------------|-------------|------|----------------|
| <u>idVisita</u> | Int(10) | No | |
| Usuario_idUsuario | Int(10) | No | 0 |
| pagina_idPagina | Int(11) | No | 0 |
| Fecha | varchar(19) | No | |

Anexo B. Script de la base de datos

```
DROP TABLE IF EXISTS `navegacion`;
CREATE TABLE `navegacion` (
  `idNavegacion` int(11) NOT NULL default '0',
  `pagina_idPagina` int(11) NOT NULL default '0',
  `perfil_idperfil` int(11) NOT NULL default '0',
  PRIMARY KEY (`idNavegacion`)
) TYPE=MyISAM;
```

```
INSERT INTO `navegacion` VALUES (2, 2, 2);
INSERT INTO `navegacion` VALUES (1, 1, 2);
INSERT INTO `navegacion` VALUES (3, 3, 2);
INSERT INTO `navegacion` VALUES (4, 3, 3);
INSERT INTO `navegacion` VALUES (5, 4, 2);
INSERT INTO `navegacion` VALUES (6, 5, 2);
INSERT INTO `navegacion` VALUES (7, 6, 3);
INSERT INTO `navegacion` VALUES (8, 6, 2);
INSERT INTO `navegacion` VALUES (9, 7, 2);
INSERT INTO `navegacion` VALUES (10, 7, 3);
INSERT INTO `navegacion` VALUES (11, 8, 2);
INSERT INTO `navegacion` VALUES (12, 9, 2);
INSERT INTO `navegacion` VALUES (13, 9, 3);
INSERT INTO `navegacion` VALUES (14, 9, 4);
INSERT INTO `navegacion` VALUES (15, 10, 2);
INSERT INTO `navegacion` VALUES (16, 11, 2);
INSERT INTO `navegacion` VALUES (17, 12, 2);
INSERT INTO `navegacion` VALUES (18, 12, 1);
INSERT INTO `navegacion` VALUES (19, 13, 1);
INSERT INTO `navegacion` VALUES (20, 14, 1);
INSERT INTO `navegacion` VALUES (21, 15, 4);
INSERT INTO `navegacion` VALUES (22, 16, 1);
INSERT INTO `navegacion` VALUES (23, 16, 3);
INSERT INTO `navegacion` VALUES (24, 17, 1);
INSERT INTO `navegacion` VALUES (25, 18, 1);
INSERT INTO `navegacion` VALUES (26, 19, 1);
INSERT INTO `navegacion` VALUES (27, 20, 4);
INSERT INTO `navegacion` VALUES (28, 21, 2);
INSERT INTO `navegacion` VALUES (29, 22, 3);
INSERT INTO `navegacion` VALUES (30, 22, 1);
INSERT INTO `navegacion` VALUES (31, 23, 1);
INSERT INTO `navegacion` VALUES (32, 24, 1);
INSERT INTO `navegacion` VALUES (33, 25, 1);
INSERT INTO `navegacion` VALUES (34, 26, 1);
INSERT INTO `navegacion` VALUES (35, 27, 1);
INSERT INTO `navegacion` VALUES (36, 28, 1);
INSERT INTO `navegacion` VALUES (37, 29, 1);
INSERT INTO `navegacion` VALUES (38, 31, 1);
```

```
DROP TABLE IF EXISTS `pagina`;
CREATE TABLE `pagina` (
  `idPagina` int(11) NOT NULL auto_increment,
  `Nombre` varchar(55) NOT NULL default '',
```

```

`Url` varchar(255) NOT NULL default '',
PRIMARY KEY (`idPagina`)
) TYPE=MyISAM AUTO_INCREMENT=32 ;

INSERT INTO `pagina` VALUES (6, 'Especializaciones',
'pnfespecializaciones.jsp');
INSERT INTO `pagina` VALUES (3, 'Dual', 'ppmdual.jsp');
INSERT INTO `pagina` VALUES (2, 'A Distancia Virtual',
'ppmdistancia.jsp');
INSERT INTO `pagina` VALUES (1, 'Presencial', 'ppmpresencial.jsp');
INSERT INTO `pagina` VALUES (5, 'Tecnologicos',
'pnftecnologicos.jsp');
INSERT INTO `pagina` VALUES (4, 'Profesionales',
'pnfprofesionales.jsp');
INSERT INTO `pagina` VALUES (7, 'Maestrias', 'pnfmaestrias.jsp');
INSERT INTO `pagina` VALUES (8, 'Tecnicos Laborales',
'pnftecnicos.jsp');
INSERT INTO `pagina` VALUES (9, 'Diplomados y Seminarios',
'pnfDyS.jsp');
INSERT INTO `pagina` VALUES (10, 'Preuniversitarios',
'pnfpreuniversitarios.jsp');
INSERT INTO `pagina` VALUES (11, 'Admisiones', 'admisiones.jsp');
INSERT INTO `pagina` VALUES (12, 'Financiacion y Becas',
'FyBecas.jsp');
INSERT INTO `pagina` VALUES (13, 'Mensaje del Rector',
'cUmensaje.jsp');
INSERT INTO `pagina` VALUES (14, 'Presentacion Institucional',
'cUpresentacion.jsp');
INSERT INTO `pagina` VALUES (15, 'Gestion de la Calidad',
'cUgestion.jsp');
INSERT INTO `pagina` VALUES (16, 'Pelaciones Nacionales e
Internacionales', 'cUrelaciones.jsp');
INSERT INTO `pagina` VALUES (17, 'Bilbioteca', 'cUbiblioteca.jsp');
INSERT INTO `pagina` VALUES (18, 'Bienestar Universitario',
'cUbienestar.jsp');
INSERT INTO `pagina` VALUES (19, 'Escuelas Facultades y
Departamentos', 'cUescuelas.jsp');
INSERT INTO `pagina` VALUES (20, 'Proyeccion Social',
'cUproyeccion.jsp');
INSERT INTO `pagina` VALUES (21, 'Campus', 'cUcampus.jsp');
INSERT INTO `pagina` VALUES (22, 'Investigacion en la Unab',
'iinvestiga.jsp');
INSERT INTO `pagina` VALUES (23, 'Ciencias Basicas, Naturales e
Ingenieria', 'ibasicas.jsp');
INSERT INTO `pagina` VALUES (24, 'Ciencias Biomedicas',
'ibiomedicas.jsp');
INSERT INTO `pagina` VALUES (25, 'Ciencias de la Educacion',
'ieducacion.jsp');
INSERT INTO `pagina` VALUES (26, 'Ciencias economicas, Administrativas
y Contables', 'ieconomicas.jsp');
INSERT INTO `pagina` VALUES (27, 'Ciencias Politicas',
'ipoliticas.jsp');
INSERT INTO `pagina` VALUES (28, 'Psicologia', 'ipsicologia.jsp');
INSERT INTO `pagina` VALUES (29, 'Socio Juridicas',
'isociojuridicas.jsp');
INSERT INTO `pagina` VALUES (30, 'Inicio', 'index.jsp');
INSERT INTO `pagina` VALUES (31, 'Ciencias Sociales',
'isociales.jsp');

```

```

DROP TABLE IF EXISTS `perfil`;
CREATE TABLE `perfil` (
  `idPerfil` int(10) unsigned NOT NULL auto_increment,
  `Nombre` varchar(20) default NULL,
  PRIMARY KEY (`idPerfil`)
) TYPE=MyISAM AUTO_INCREMENT=6 ;

INSERT INTO `perfil` VALUES (1, 'estudiante');
INSERT INTO `perfil` VALUES (2, 'aspirantes');
INSERT INTO `perfil` VALUES (3, 'egresados');
INSERT INTO `perfil` VALUES (4, 'empresarios');
INSERT INTO `perfil` VALUES (5, 'General');

DROP TABLE IF EXISTS `recomendado`;
CREATE TABLE `recomendado` (
  `idRecomendado` int(10) NOT NULL auto_increment,
  `pagina_idPagina` int(10) NOT NULL default '0',
  `perfil_idPerfil` int(10) NOT NULL default '0',
  `noticia` varchar(255) NOT NULL default '',
  PRIMARY KEY (`idRecomendado`)
) TYPE=MyISAM AUTO_INCREMENT=9;

INSERT INTO `recomendado` VALUES (1, 17, 1, 'Nuevos plazos para entregas de libros, los plazos se extenderan con la intencion que los estudiantes puedan tener un mejor acceso a los libros');
INSERT INTO `recomendado` VALUES (2, 13, 1, 'Le rector deja un nuevo mensaje para los estudiantes');
INSERT INTO `recomendado` VALUES (3, 12, 2, 'La UNAB ofrece nuevos planes de financiacion');
INSERT INTO `recomendado` VALUES (4, 11, 2, 'La Unab comienza un nuevo ciclo de admisiones');
INSERT INTO `recomendado` VALUES (5, 6, 3, 'Nuevos programas de especializaciones');
INSERT INTO `recomendado` VALUES (6, 7, 3, 'Nuevos Programas de maestrias');
INSERT INTO `recomendado` VALUES (7, 9, 4, 'Diplomados con cupos especiales para empresas');
INSERT INTO `recomendado` VALUES (8, 15, 4, 'Nuevos programas de gestion empresarial');

DROP TABLE IF EXISTS `usuario`;
CREATE TABLE `usuario` (
  `idUsuario` int(10) unsigned NOT NULL auto_increment,
  `Perfil_idPerfil` int(10) unsigned NOT NULL default '0',
  `fechacreacion` varchar(19) NOT NULL default '',
  PRIMARY KEY (`idUsuario`)
) TYPE=MyISAM AUTO_INCREMENT=6 ;

INSERT INTO `usuario` VALUES (4, 3, '2006-10-21 18:22:54');
INSERT INTO `usuario` VALUES (3, 2, '2006-10-17 18:21:43');
INSERT INTO `usuario` VALUES (5, 4, '2006-10-14 18:23:00');

DROP TABLE IF EXISTS `visita`;
CREATE TABLE `visita` (
  `idVisita` int(10) unsigned NOT NULL auto_increment,
  `Usuario_idUsuario` int(10) unsigned NOT NULL default '0',
  `pagina_idPagina` int(11) NOT NULL default '0',
  `Fecha` varchar(19) NOT NULL default '',
  PRIMARY KEY (`idVisita`)

```

```
) TYPE=MyISAM AUTO_INCREMENT=214 ;

INSERT INTO `visita` VALUES (1, 4, 30, '2006-10-31 11:05:38');
INSERT INTO `visita` VALUES (2, 4, 30, '2006-10-31 11:04:39');
INSERT INTO `visita` VALUES (3, 4, 30, '2006-10-31 11:02:54');
INSERT INTO `visita` VALUES (4, 4, 30, '2006-10-31 10:59:05');
INSERT INTO `visita` VALUES (5, 4, 30, '2006-10-31 09:27:21');
INSERT INTO `visita` VALUES (6, 4, 30, '2006-10-31 09:27:19');
```