

**Universidad Central de Venezuela  
Facultad de Ciencias  
Escuela de Computación**



**Easy Extractor: Un sistema  
de Clasificación de Información de la Web  
utilizando técnicas de extracción de datos**

Trabajo Especial de Grado  
Presentado ante la Ilustre  
Universidad Central de Venezuela  
Por los bachilleres

**Andrés José Díaz Márquez  
Lucía Francesca Poli López**

Para optar por el título de Licenciado en Computación

Tutores:  
Prof. Andrés Sanoja  
Prof. Claudia León

Caracas, 23 de Octubre 2008

## ACTA

Quienes suscriben, miembros del Jurado designado por el Consejo de la Escuela de Computación de la Universidad Central de Venezuela, para examinar el Trabajo Especial de Grado presentado por los bachilleres Andrés José Díaz Márquez, C.I. 17.884.170 y Lucía Francesca Poli López, C.I. 17.063.479; con el título: **Easy Extractor: Un sistema de Clasificación de Información de la Web utilizando técnicas de extracción de datos**, a los fines de optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído como fue, dicho trabajo por cada uno de los miembros del jurado, se fijó el día **23 de Octubre del 2008** a las **11:00 am**, para que sus autores lo defendieran de forma pública, lo que hicieron en el aula **PB III** de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondieron a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el Jurado decidió aprobarlo.

En fé de lo cual se levanta la presente Acta, en Caracas a los 23 días del mes de Octubre del año dos mil ocho (2008) dejándose también constancia de que actuó como Coordinador del Jurado la Profesora tutora Claudia León.

---

Prof. Andrés Sanoja  
Tutor

---

Profa. Claudia León  
Tutora

---

Prof. Sergio Rivas  
Jurado Principal

---

Prof. Antonio Silva  
Jurado Principal

---

Profa. Yosly Hernández  
Jurado Suplente

---

Prof. Robinson Rivas  
Jurado Suplente

## DEDICATORIAS

La realización de este proyecto está dedicado especialmente:

A Dios nuestro señor, porque de él proviene toda la sabiduría.

A mis padres, coprotagonistas de esta historia, quienes por sus sabios consejos y todos sus sacrificios hicieron posible éste proyecto.

A mis hermanos, por su ayuda incondicional y apoyo en todo momento, fuentes de recreación del buen humor, imprescindible para hacer posible este trabajo.

A mi abuela, la que hoy me acompaña con sus sabios consejos y los que desde el cielo nos bendicen con su amor y continua protección.

A mis tíos, por su afecto, entendimiento y continua ayuda.

A mis amigos, por brindarme su gran apoyo y ánimo en cada instante.

Andrés Díaz

En primer lugar le dedico esta tesis a Dios quien me dio la fé, las fuerzas, la paciencia y la salud para superar todas las dificultades y poder finalizar este trabajo.

También se las dedico a mis padres, Consola y Mercurio, quienes me han apoyado de forma incondicional desde el comienzo de mi vida y siempre han sido mi sostén y apoyo en los momentos difíciles.

A mis abuelos Lucia y Martín quienes siempre me impulsaron a seguir los estudios y me enseñaron a no conformarme y siempre superarme, que aunque no están aquí conmigo, me han acompañado desde el cielo iluminándome el camino a seguir.

A mi hermano Héctor, el cual fue mi ejemplo a seguir y mi apoyo durante mis años de estudio.

A mis tíos, Manuela, Mercedes, Claudio y Adrián, los cuales han sido como mis segundos padres y me han ayudado a ser una mejor persona.

A mis amigos, Andrés, Néstor, Lucy, Enmanuel, Andreina de Sousa, Andreina Jiménez, Rosa, Jesús, Constantino, Blanca, con quienes compartí mis años de estudio y me han servido de apoyo y fuente de estímulo.

Lucía Poli

## **AGRADECIMIENTOS**

A la Universidad Central de Venezuela, alma mater y fuente de conocimientos, "Casa que vence la sombra".

A mis padres por ser la luz en todo momento de oscuridad.

A mis hermanos por darme el apoyo durante toda la carrera universitaria.

A mi novia, Lucy Fernández, por tener la calma, paciencia y palabras que necesite en algún momento de mi carrera.

A mis amigos Néstor, Lucia, Enmanuel, Constantino, Andreina de Sousa, Andreina Jiménez, Rosa, Blanca por haber compartido y disfrutado con ellos, todos estos años de vida universitaria.

A el tutor académico, el profesor Andrés Sanoja por su apoyo incondicional, colaboración y orientación a lo largo del desarrollo de este proyecto.

A la profesora Claudia León por su apoyo en la culminación de mi trabajo especial de grado.

Y a todas aquellas personas que de una u otra manera ayudaron a realizar este trabajo. A todos ellos, muchas gracias.

Andrés Díaz

Quiero realizar un especial agradecimiento a nuestra insigne Universidad Central de Venezuela, la cual fue mi casa durante los 5 años de estudios. A los profesores de la Escuela de Computación por sus enseñanzas y colaboración en mi formación profesional, muy especialmente al Profesor Andrés Sanoja, quien fue nuestro guía y mentor en la realización de este trabajo y sin el cual no habríamos podido lograrlo. También le quiero agradecer a la Profesora Claudia León por la ayuda que nos brindó en el perfeccionamiento de este trabajo.

Quiero agradecer a Andrés Díaz por haber sido mi amigo, compañero de estudios y mi apoyo en los momentos difíciles y cruciales de esta tesis.

A quienes más les debo agradecer es a mis padres, Consola y Mercurio, quienes no sólo me dieron la vida sino que me han apoyado e impulsado a seguir mis sueños y nunca desmayar hasta alcanzarlos, y sin los cuales yo no hubiera podido realizar mi carrera universitaria. Muchas gracias y los quiero mucho.

A mis abuelos, Lucía y Martín, que se que me deben estar viendo y deben estar orgullosos de mis logros, quienes me enseñaron que puedo lograr todo lo que me proponga en la vida esforzándome un poco y nunca conformándome.

Al resto de mi familia, a mis tíos, primos y hermano, gracias por haberme apoyado e impulsado a siempre mejorar y sus constantes palabras de aliento a seguir trabajando y estudiando.

A todos mis amigos, quienes fueron y son una parte importante de mi vida, que me apoyaron en los buenos y malos momentos, con quienes compartí momentos de alegrías y tristezas, muchas gracias por su apoyo incondicional y su amistad.

A todas las personas que han formado parte de mi vida y han ayudado a mi formación personal y profesional, gracias.

Lucía Poli

## RESUMEN

La información en la Web se encuentra en cambio constante y sujeta a la volatilidad de la red, por lo que una página puede existir hoy pero no mañana o encontrarla con un contenido distinto. De esta manera surge la necesidad de extraer, almacenar y clasificar información que el usuario considere importante en la Web. En la actualidad existen algunas herramientas que han solucionado dicha problemática, entre las cuales mencionamos Solvent y Google Bloc de notas. Sin embargo pueden no satisfacer las necesidades de algunos usuarios.

El sistema Easy Extractor, se crea como una herramienta distinta que no sólo engloba funcionalidades de otras aplicaciones, sino que ofrece nuevas posibilidades a los usuarios. En el proceso de desarrollo de esta aplicación se realizó una adaptación de la metodología *eXtreme Programming* (XP). Dentro de las cualidades con las que cuenta este sistema son: compartir información con otros usuarios, exportar la información a formato PDF para su almacenaje local, intercambiar comentarios e ideas con otros usuarios, demarcado y almacenaje de información e imágenes para su posterior recuperación, entre otras.

**Palabras Claves:** Mozilla, XUL, WEB, Aplicación, Servicios Web.

---

## ÍNDICE DE CONTENIDO

INTRODUCCIÓN.....	1
CAPÍTULO I: MARCO REFERENCIAL.....	6
1.1.    XUL (XML-based User-interface Language).....	6
1.2.    Javascript .....	8
1.3.    Extracción de datos de la Web basado en XPATH.....	8
1.4.    Servidor .....	9
1.5.    Servicios Web .....	10
CAPÍTULO II: MARCO APLICATIVO .....	16
2.1.    Estructura del Sistema. ....	20
2.2.    Especificaciones técnicas: .....	22
2.3.    Iteración 1.....	22
2.4.    Iteración 2.....	34
2.5.    Iteración 3.....	38
2.6.    Iteración 4.....	43
CAPÍTULO III: CASOS DE ESTUDIO Y ENCUESTA .....	58
3.1.    Casos de Estudio.....	58
3.2.    Encuesta.....	67
3.3.    Resultados.....	69
CONCLUSIONES .....	71
RECOMENDACIONES.....	72
REFERENCIAS.....	73
LECTURAS RECOMENDADAS .....	74
APÉNDICE: ENCUESTAS REALIZADAS .....	75

**ÍNDICE DE FIGURAS**

Figura 1. Arquitectura del sistema Web.....	4
Figura 2. Ejemplo de REST. ....	12
Figura 3. Arquitectura de Axis2.....	13
Figura 4. Motor Extensible de Mensajería.....	14
Figura 5. Arquitectura de Módulos.....	15
Figura 6: Metáfora (Estructura del Sistema).....	20
Figura 7: Barra de Herramientas.....	24
Figura 8: Diseño de la Barra de Herramientas.....	24
Figura 9: Ventana de Ingreso en la Extensión Firefox. ....	25
Figura 10: Diseño del ingreso en la extensión firefox. ....	26
Figura 11: Ventana de Selección de la Información a Almacenar. ....	27
Figura 12: Diseño de la selección de la información a almacenar.....	28
Figura 13: Ventana del Formulario de Almacenaje de Información. ....	29
Figura 14: Diseño del Formulario de Almacenaje de Información.....	30
Figura 15. Código fuente de la extracción XPath.....	33
Figura 16: Diagrama de clases persistentes.....	35
Figura 17: Llamada REST. ....	36
Figura 18: Llamada REST (usuario registrado).....	37
Figura 19: Menú Superior.....	39
Figura 20: Menú Lateral.....	39
Figura 21: Menú Superior del usuario autenticado. ....	40
Figura 22: Código fuente del archivo header.jsp. ....	40
Figura 23 Código fuente del archivo lateral_menu.jsp. ....	41
Figura 24: Código fuente del archivo superior_menu.jsp. ....	42
Figura 25: Página Principal de la Aplicación Web.....	46
Figura 26: Página de búsqueda.....	47
Figura 27: Artículo seleccionado por el usuario.....	47
Figura 28: Página de Ingreso al sistema. ....	48
Figura 29: Página de Registro. ....	48
Figura 30: Estructura de carpetas del proyecto en Myeclipse.....	49
Figura 31: Código fuente del archivo <i>IndexAction</i> . ....	50
Figura 32: Código fuente del archivo <i>CloseAction</i> . ....	51
Figura 33: Código fuente del archivo <i>SaveCommentArtAction</i> . ....	52
Figura 34: Modelos de la Base de Datos para Hibernate. ....	53
Figura 35: Clases en la carpeta form. ....	53
Figura 36: Clases en la carpeta Model. ....	54
Figura 37: <i>JSP</i> en la carpeta Views. ....	55
Figura 38. Página principal (tercer caso de estudio). ....	59
Figura 39. Fragmento extraído de la página. ....	60
Figura 40. Información a ser almacenada. ....	60

## Índice de Figuras

---

Figura 41. Imagen almacenada en la sesión de usuario.....	61
Figura 42. Página principal (segundo caso de estudio).....	62
Figura 43. Fragmento extraído de la página. ....	63
Figura 44. Información a ser almacenada. ....	63
Figura 45. Imagen almacenada en la sesión de usuario.....	64
Figura 46. Página principal (tercer caso de estudio). ....	65
Figura 47. Fragmentos extraídos de la página. ....	66
Figura 48. Información a ser almacenada. ....	66
Figura 49. Artículo Almacenado en la Página Web. ....	67
Figura 50. Gráfico resultante de la aplicación de la encuesta. ....	69

## ÍNDICE DE TABLAS

Tabla 1: Formato de historias de usuario.....	16
Tabla 2: Formato de registro de pruebas unitarias del lado del cliente. ....	20
Tabla 3: Pruebas relacionadas a las historias de usuario de la iteración 1.....	34
Tabla 4: Prueba relacionada a la historia de usuario de la iteración 2.....	38
Tabla 5: Prueba relacionada a la historia de usuario de la iteración 3.....	43
Tabla 6: Pruebas relacionadas a las historias de usuario de la iteración 4.....	57

## INTRODUCCIÓN

En sus comienzos, el Internet fue diseñado o creado únicamente para intercomunicar a las personas a través de las páginas Web, correos, entre otros; es por esta razón que surgen los navegadores Web, los cuales son los encargados de mostrar la información y las páginas Web a los usuarios.

El surgimiento de Mozilla Firefox trajo consigo la utilización de un lenguaje de programación llamado XUL, el cual fue concebido inicialmente para el desarrollo de las aplicaciones de Mozilla como FileZilla, Thunderbird, entre otros. Sin embargo, con el surgimiento del concepto de la Web 2.0 (la cual es una Web colaborativa donde los usuarios finales se convierten en los productores del contenido que se va a transmitir sin necesidad de tener conocimientos tecnológicos avanzados, debido a que las mismas páginas Web están condicionadas para este tipo de interacción [1]), se hace uso del lenguaje XUL para el desarrollo de aplicaciones sociales. Este es un lenguaje basado en XML que ofrece la facilidad de creación de interfaces de usuario permitiendo el desarrollo de aplicaciones multiplataforma sofisticadas sin necesidad de herramientas especiales.

Las aplicaciones sociales tienen como finalidad proveer la navegación y búsqueda de información en la Web a los usuarios, no obstante, aunque existen varias de estas aplicaciones, aún persisten necesidades de los usuarios por satisfacer. Una de éstas necesidades es la extracción, almacenaje y clasificación de información importante encontrada en la Web; debido a que dicha información se encuentra sujeta a la volatilidad de la red, la cual está cambiando constantemente, debido a que una página puede existir hoy pero no mañana, hasta puede seguir existiendo físicamente pero con un contenido distinto al del día anterior.

Hay algunas aplicaciones que se han enfocado en esta necesidad, pero como la misma es muy amplia, cada aplicación lo ha hecho en un nivel de abstracción diferente, abarcando desde lo más general hasta lo más específico, es decir, algunas se han enfocado en almacenar el *link* de la página de interés (Bookmarks) y hay otras que sólo almacenan lo que el usuario selecciona de la página (Google bloc de notas). Existen varias aplicaciones que permiten extraer la información de la Web, estas son Solvent y Google Bloc de Notas.

Solvent [2] es una extensión de Firefox, que ayuda en la demarcación de la información relevante de una página Web. Haciendo uso de XPath, señala o subraya partes de la página que pueden ser de interés para el usuario y genera extractores para almacenar algunos o todos los fragmentos que se seleccionan de la página, impidiendo que el usuario guarde sólo la información importante de dicho fragmento. Por su parte, Google Bloc de Notas [3] es un servicio que provee Google a sus usuarios, en el cual se puede almacenar todo tipo de información. Con la finalidad de hacer más útil este servicio, se encuentra disponible una extensión de firefox que permite añadir recortes de textos, imágenes y vínculos procedentes de las páginas Web y colocarlos en Google Bloc de notas sin tener que salir de la ventana del navegador, sin embargo, este no le señala al usuario cuáles son las partes de la página que puedan ser de interés.

Estas herramientas son útiles en el enfoque en que fueron diseñadas y desarrolladas, pero aún así pueden no cumplir con los requerimientos o necesidades de otros usuarios.

En virtud de lo expuesto anteriormente, con este Trabajo Especial de Grado se pretende desarrollar un sistema Web capaz de señalar al usuario la información de su interés contenida en una página Web, de modo que posteriormente él (ella) pueda seleccionar lo que realmente desee guardar, resolviendo de esta forma las debilidades de las aplicaciones mencionadas

anteriormente, además de permitirle organizar sus resultados de una forma estructurada, de manera tal que sea posible reutilizar esta información por él (ella) o por otros usuarios, o bien utilizarla en sistemas de software como datos de entrada.

Para dar cumplimiento a esto se plantean lo siguientes objetivos:

- **Objetivo General**

Desarrollar un sistema Web que permita la clasificación de páginas utilizando el marcado, la extracción y el almacenaje de datos relevantes de una página Web seleccionada por el usuario.

- **Objetivos Específicos**

- Diseñar e implementar una aplicación del lado del cliente encargada de realizar el proceso de extracción de los datos.

- Diseñar e implementar una aplicación del lado del servidor encargada de almacenar los datos extraídos por el usuario.

- Elaborar una aplicación Web en donde los usuarios puedan consultar los datos que han almacenado.

En base a estos objetivos, se desarrollará la aplicación Easy Extractor, la cual constará de dos módulos: una extensión de Firefox y una aplicación del lado del servidor. En la figura 1 se puede observar la estructura general del sistema y algunas de las tecnologías involucradas en el desarrollo de cada módulo. La fuente de entrada será un conjunto de páginas Web conformada por cualquier página Web disponible en Internet, accesible mediante un navegador Web, y de la cual el usuario desee almacenar información.

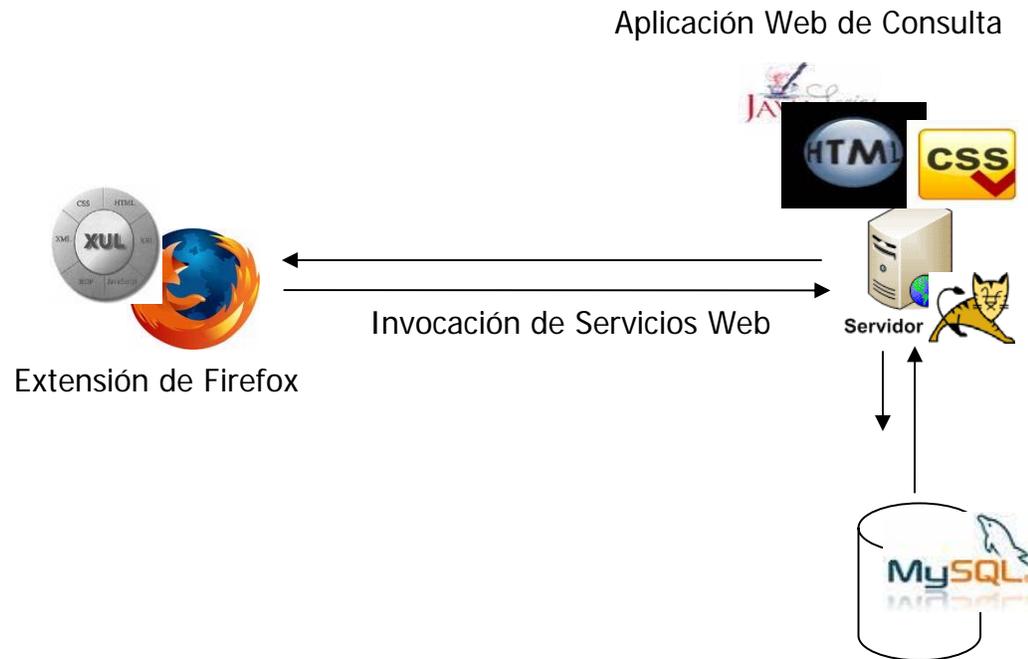


Figura 1. Arquitectura del sistema Web.

- **Extensión de Firefox**

Es una aplicación desarrollada en lenguaje XUL, instalada en el navegador del usuario, en la cual será necesario autenticarse. Además, se hará uso de la tecnología XPath para la localización de la información contenida en las etiquetas HTML que delimitan el contenido de la página y que será de utilidad al usuario para seleccionar la información que le sea relevante y así poder guardarla para una próxima revisión. Una vez que el usuario seleccione la información e imágenes que desee almacenar, deberá asignarles un título, categoría y tags, para su posterior recuperación.

- **Aplicación del Lado del Servidor**

Esta aplicación será desarrollada en Java, bajo el Patrón de Diseño Modelo-Vista-Controlador (MVC), encargada del proceso de almacenaje y recuperación de los datos de la base de datos Mysql utilizando el Servidor de Aplicaciones Apache Tomcat. Se proveerá una página Web para que los usuarios puedan consultar la información que han almacenado, así como la que han guardado otros usuarios, según las categorías (tags) dispuestas por ellos mismos y categorías predefinidas. Además, se proveerá a los usuarios la posibilidad de exportar artículos seleccionados de la página a formato Excel o PDF, para su uso particular. También, se podrá crear nuevos artículos en base a artículos encontrados en la página.

El Trabajo Especial de Grado se estructura de la siguiente manera:

- En el primer capítulo se describe el marco teórico, en el cual se especifica los tipos de lenguajes de programación y las diferentes tecnologías, tanto del lado del cliente como del servidor, a utilizar.
- En el segundo capítulo se muestra el marco aplicativo donde se precisan todos los pasos efectuados para lograr el desarrollo del sistema, usando una adaptación del proceso de desarrollo XP.
- En el tercer capítulo se presenta el caso de estudio y los resultados obtenidos de la interacción por usuarios externos al sistema.

Por último se presentan las conclusiones del sistema realizado.

## **CAPÍTULO I: MARCO REFERENCIAL**

Este capítulo presenta las bases conceptuales para la realización del sistema propuesto, explicando el proceso de desarrollo a utilizar, describiendo brevemente las tecnologías Web involucradas en la implementación de un sistema dinámico e interactivo.

### **1.1. XUL (XML-based User-interface Language)**

XUL es un lenguaje basado en XML utilizado para describir y crear interfaces de usuario, que ha sido diseñado para brindar la portabilidad de las mismas, por lo que permite desarrollar aplicaciones multiplataforma sofisticadas o complejas sin necesidad de herramientas especiales [4].

Inicialmente XUL fue creado para desarrollar los productos de Mozilla tanto en el navegador como en el cliente de correo electrónico, entre otros. Al ser un lenguaje basado en XML, contiene todas las características disponibles para XML y sus mismas ventajas.

Algunos elementos que pueden ser creados son:

- Controles de entrada tales como cuadros de texto y cajas de chequeo
- Barra de herramientas con botones u otros contenidos
- Menús en barras de menú o menú emergente
- Pestañas de diálogo.
- Árbol de información jerárquica o tabulada
- Teclas de accesos directo

Con XUL se pueden crear distintos tipos de aplicaciones, siendo las más usuales las extensiones de Firefox, las cuales permiten agregar funcionalidades al navegador, por ejemplo una barra de herramientas suplementarias, menús contextuales o una interfaz de personalización del navegador. Las extensiones pueden igualmente ser aplicadas a otros productos de Mozilla tal como Thunderbird.

Una interfaz XUL es definida mediante la especificación de tres grupos de componentes:

- **Content:**

Aquí se encuentran los documentos XUL, que definen el diseño de la interfaz, que contienen las declaraciones de las ventanas y los elementos de la interfaz de usuario.

- **Skin:**

Un skin se compone de archivos CSS y de un número de imágenes usadas para definir el aspecto y la interfaz. Las hojas de estilo describen detalles de la apariencia de una ventana. Se almacenan de forma separada a los archivos XUL para facilitar la modificación del skin de una aplicación.

- **Locale:**

Está compuesto por todos los archivos donde se definen los textos a utilizar en la interfaz, en distintos idiomas, facilitando la internacionalización de la aplicación.

## 1.2. Javascript

Es un lenguaje de programación utilizado para crear pequeños programas o scripts encargados de realizar acciones dentro del ámbito de una página Web. Con Javascript se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador Web es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estas acciones [5].

Las principales características son:

- Es un lenguaje basado en objetos y orientado a eventos, diseñado específicamente para el desarrollo de aplicaciones dentro del ámbito de Internet.
- Es sencillo y pensado para proveer rapidez y ligereza en las actividades realizadas sobre plataforma Web.
- Los programas en Javascript pueden estar incluidos en los documentos HTML y/o tener la referencia a un archivo externo de extensión “js”; y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos y manejar los elementos de la página.
- Es un lenguaje que permite tanto la programación de scripts o programas, orientados a objetos, con funciones y estructuras de datos complejas.

## 1.3. Extracción de datos de la Web basado en XPATH

Es un lenguaje para direccionar partes de un documento XML, diseñado para ser utilizado tanto por XSL como por XPointer. Según James Clark y Steve DeRose [6], XPath es el resultado de un esfuerzo para

proporcionar una sintaxis y semántica comunes para funcionalidades compartidas entre XSL Transformations (XSLT) y XPointer; proporciona facilidades para manipulación de tipos de datos básicos, tales como cadenas de caracteres, números y booleanos. Utiliza una sintaxis compacta y no XML para facilitar su uso dentro de URIs y de valores de atributos XML. XPath obtiene su denominación por el uso que hace de una notación de caminos, como en los URLs, para navegar a través de la estructura jerárquica de un documento XML.

El documento XML es analizado a través de un *parser* generando un árbol de nodos y se utilizan los caminos de localización para establecer la ruta a seguir a través de dicho árbol para llegar al dato o componente deseado. Estas rutas pueden ser relativas o absolutas, y consisten en una secuencia de uno o más pasos de localización separados por "/", en donde cada paso selecciona un conjunto de nodos relativos a un nodo actual.

#### **1.4. Servidor**

Un servidor es un programa en ejecución (un proceso) en un computador en red que acepta peticiones de programas que se están ejecutando en otros computadores para realizar un servicio y responder adecuadamente [7]. El servidor más utilizado en aplicaciones java es Apache Tomcat, el cual es un servidor de aplicaciones o contenedor Web, que es usado en la implementación de las especificaciones de Java Servlets y JavaServer Pages (Jsp), de Sun Microsystems, proporcionando un ambiente para que el código Java pueda ejecutarse con la cooperación de un servidor Web.

Tomcat posee herramientas para su configuración y administración, pero también puede ser configurado modificando los archivos de

configuración que poseen formato XML. Además, contiene su propio servidor HTTP interno.

Incluye el compilador Jasper, que compila JSP's convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor Web Apache.

La jerarquía de directorios de instalación de Tomcat incluye:

- **Bin:** arranque, cierre, y otros scripts y ejecutables.
- **Common:** clases comunes que pueden utilizar Catalina y las aplicaciones Web.
- **Conf:** archivos XML y los correspondientes DTD para la configuración de Tomcat.
- **Logs:** logs de Catalina y de las aplicaciones.
- **Server:** clases utilizadas solamente por Catalina.
- **Shared:** clases compartidas por todas las aplicaciones Web.
- **Webapps:** directorio que contiene las aplicaciones Web.
- **Work:** almacenamiento temporal de archivos y directorios.

### 1.5. Servicios Web

Son un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web [8].

Los servicios Web son usados para establecer la comunicación entre dos aplicaciones, ya que estos son una colección de protocolos y estándares

que sirven para intercambiar datos entre aplicaciones, las cuales pueden estar desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma. En el caso de estudio serán utilizados para comunicar una aplicación desarrollada en XUL del lado del cliente y una aplicación desarrollada en Java del lado del Servidor.

La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web ha sido creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

Existen distintas formas de usar los servicios Web, la estudiada en este caso será *Representational State Transfer* (REST), la cual es una técnica de arquitectura de software para sistemas hipermedia distribuidos como la World Wide Web. El término se originó en el año 2000, en una tesis doctoral sobre la Web escrita por Roy Fielding [9], uno de los principales autores de la especificación del protocolo HTTP y ha pasado a ser ampliamente utilizado por la comunidad de desarrollo.

Según el autor, REST está pensada para evocar una imagen de cómo se comporta una aplicación Web bien diseñada; una red de páginas Web, donde los usuarios avancen en la aplicación a través de *links* (transiciones de estado), obteniendo como resultado que la siguiente página (representando el próximo estado de la aplicación) sea transferida al usuario y ofrecida para su uso.

La razón por la cual se llama REST, es porque un usuario hace una referencia a una página Web usando el URL, la cual devuelve una representación del recurso (en la Figura 2 es un documento HTML). Esta representación coloca al usuario en un estado. Cuando el usuario selecciona

un *hyperlink* en la página HTML es otro recurso, por lo que la nueva representación coloca al usuario en un nuevo estado. En conclusión, el usuario es transferido de estado con cada representación (Representación de transferencia de estado).



**Figura 2. Ejemplo de REST.**

Las principales características de REST son:

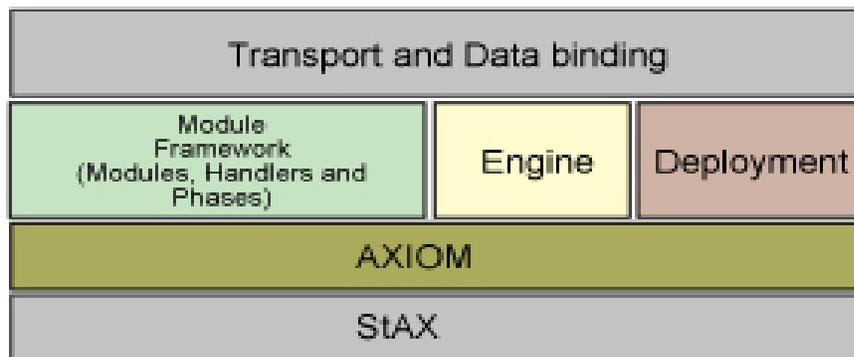
- **Protocolo cliente/servidor sin estado:** cada mensaje HTTP contiene toda la información necesaria para comprender la petición. Como resultado, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes. Sin embargo, en la práctica, muchas aplicaciones basadas en HTTP utilizan *cookies* y otros mecanismos para mantener el estado de la sesión (algunas de estas prácticas, como la reescritura de URLs, no son permitidas por REST).
- **Conjunto de operaciones bien definidas:** se aplican a todos los recursos de información; HTTP en sí define un conjunto pequeño de operaciones, las más importantes son POST, GET, PUT y DELETE. Con frecuencia estas operaciones equivalen a las operaciones CRUD

(create, read, update, delete), que se requieren para la persistencia de datos, aunque POST no encaja exactamente en este esquema.

- **Sintaxis universal:** usado para identificar los recursos. En un sistema REST, cada recurso es direccionable únicamente a través de su URI.
- **Uso de hipermedios:** para la información de la aplicación y las transiciones de estado de la aplicación: la representación de este estado en un sistema REST son típicamente HTML o XML. Como resultado de esto, es posible navegar de un recurso REST a muchos otros, simplemente siguiendo enlaces sin requerir el uso de registros u otra infraestructura adicional.

En cuanto a las plataformas a usar para el desarrollo de servicios Web, existen múltiples arquitecturas que pueden ser usados, en este caso estudiaremos Axis2, ya que posee un componente de soporte a REST.

Axis2 está compuesta por los siguientes componentes (Figura 3) [10]:



**Figura 3. Arquitectura de Axis2.**

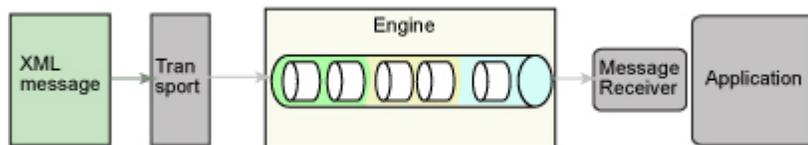
- **AXIOM**

El Modelo del Objeto de Axis (AXIOM), es un modelo de objeto XML desarrollado para mejorar el uso y desempeño de la memoria durante el

procesamiento del XML, basado en *pull parser*. Usando el Streaming API para XML (StAX) *pull parser*, AXIOM puede controlar el proceso de parseo para proveer soporte de construcción diferido. Construcción diferida es la habilidad que posee AXIOM de construir un modelo de objeto parcialmente mientras el resto del modelo es construido según las necesidades del usuario. La ventaja es que la memoria sólo es usada para satisfacer las necesidades del usuario.

- **Motor Extensible de Mensajería**

El motor de Axis2 es un procesador de SOAP puro y no depende de ninguna especificación de Java. El funcionamiento de este motor (Figura 4) comienza cuando un mensaje es recibido a través de un transporte, luego el motor llama un conjunto de interceptores predefinidos llamados manejadores. Los manejadores procesan normalmente la información dentro de las cabeceras de SOAP, aunque ellos no tienen limitaciones para procesar otras partes de mensaje también. El mensaje es entonces entregado a un receptor de mensajes.

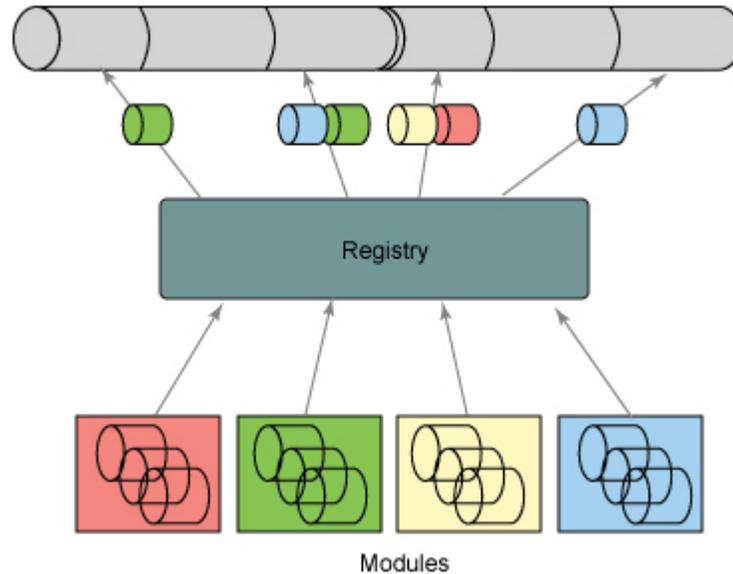


**Figura 4. Motor Extensible de Mensajería**

- **Arquitectura de Módulos**

Los módulos proveen un mecanismo de extensión al servidor. Cada módulo de Axis2 contiene un conjunto de manejadores relacionados. Por ejemplo, existe un módulo denominado WS-Addressing el cual contiene los manejadores para el soporte de ES-Addressing por parte del motor de Axis2. Si un administrador de Axis2 desea agregar este módulo, solo debe desplegar

el mismo en el motor de Axis2. Las reglas, para designar los manejadores que van en cada fase y tubo, se encuentran en el archivo *modules.xml*. (Figura 5).



**Figura 5. Arquitectura de Módulos**

- **Modelo de Desarrollo Mejorado**

Axis2 permite el despliegue de servicios en su motor durante su uso, lo cual permite que los usuarios desplieguen servicios sin necesidad de reiniciar el servidor. Los servicios deben estar comprimidos en archivos con extensión *.aar* (archivos Axis2).

## CAPÍTULO II: MARCO APLICATIVO

En este capítulo se presenta el análisis y diseño de la herramienta Easy Extractor elaborada en este trabajo especial de grado con el fin de describir todas las funcionalidades del sistema y cumplir los objetivos propuestos al comienzo de esta investigación.

La metodología utilizada para el desarrollo del sistema fue una adaptación de XP (*eXtreme Programming*), la cual es una metodología ágil, en donde los individuos e interacciones son más importantes que los procesos y herramientas, y que el funcionamiento del software sea más importante que una documentación exhaustiva. Las fases definidas por XP se aplican en el desarrollo de nuestro sistema de la siguiente manera:

- **Planificación:**

Una vez determinadas las necesidades del cliente (en este caso el tutor), se procedió al análisis de las mismas para modelar un sistema que satisfaga dichos requerimientos, lo que llevó al desarrollo de las historias de usuario.

Las historias de usuario fueron definidas siguiendo el formato mostrado en la tabla 1. Este formato está compuesto por el número de historia de usuario, el nombre de la misma, la prioridad y riesgo de desarrollo con un rango de valores bajo, medio y alto, y la descripción del requerimiento del cliente.

Número:	Nombre:
Prioridad:	Riesgo en desarrollo:
Descripción:	

**Tabla 1: Formato de historias de usuario.**

Las iteraciones se planificaron en base a los módulos del sistema, es decir cada iteración durará el tiempo que tome el desarrollo de cada módulo, los

cuales son: extensión XUL, servicios Web y la aplicación Web. Al concluir cada iteración se le entregará al cliente el módulo desarrollado para la realización de pruebas, correcciones y cambios.

Las historias de usuario obtenidas de los requerimientos del cliente son las siguientes:

<b>Número:</b> 1	<b>Nombre:</b> Elaborar una aplicación cliente para la extracción de datos de páginas Web.
<b>Prioridad:</b> Alta.	<b>Riesgo en desarrollo:</b> bajo.
<b>Descripción:</b> crear una extensión de firefox para realizar la demarcación y extracción de la información relevante para el usuario de una página Web	

<b>Número:</b> 2	<b>Nombre:</b> Realizar el almacenaje de la información seleccionado por el usuario.
<b>Prioridad:</b> Alta.	<b>Riesgo en desarrollo:</b> bajo.
<b>Descripción:</b> diseñar y desarrollar los mecanismos de comunicación necesarios entre las aplicaciones cliente y servidor, para el almacenaje de datos extraídos por el usuario de una página Web.	

<b>Número:</b> 3	<b>Nombre:</b> Elaborar una interfaz de usuario para la consulta de la información guardada por los usuarios.
<b>Prioridad:</b> Alta.	<b>Riesgo en desarrollo:</b> bajo.
<b>Descripción:</b> diseñar y desarrollar una página Web, con las secciones necesarias, para que los usuarios puedan consultar la información que han almacenado.	

<b>Número:</b> 4	<b>Nombre:</b> Permitir conocer el número de usuarios que han marcado información de una página Web.
<b>Prioridad:</b> Baja.	<b>Riesgo en desarrollo:</b> bajo.
<b>Descripción:</b> Desarrollar un icono en la barra de herramientas del navegador Web capaz de informar al usuario el número de usuarios que han realizado el marcado de información de la página Web que tiene abierta en el navegador.	

<b>Número:</b> 5	<b>Nombre:</b> Mostrar toda la información marcada por un usuario.
<b>Prioridad:</b> Baja.	<b>Riesgo en desarrollo:</b> bajo.
<b>Descripción:</b> Desarrollar una opción en la interfaz de usuario para la recuperación de toda la información almacenada por un usuario específico.	

<b>Número:</b> 6	<b>Nombre:</b> Elaborar un motor de búsqueda de información almacenada por otros usuarios.
<b>Prioridad:</b> Baja.	<b>Riesgo en desarrollo:</b> bajo.
<b>Descripción:</b> Elaborar un sistema de búsqueda de información basada en tags o categorías, para que los usuarios puedan encontrar información marcada y almacenada por otros usuarios.	

<b>Número:</b> 7	<b>Nombre:</b> Mostrar todos los usuarios que han marcado y almacenado información de una página Web específica.
<b>Prioridad:</b> Baja.	<b>Riesgo en desarrollo:</b> bajo.
<b>Descripción:</b> Elaborar una opción en la página Web o en la extensión de firefox que muestre todos los usuarios que han marcado y almacenado información de la página Web desplegada en el momento en el navegador.	

<b>Número:</b> 8	<b>Nombre:</b> Mostrar información del usuario en línea.
<b>Prioridad:</b> Baja.	<b>Riesgo en desarrollo:</b> bajo.
<b>Descripción:</b> Elaborar una opción en la aplicación Web que muestre información del usuario en línea.	

<b>Número:</b> 9	<b>Nombre:</b> Hacer comentarios de artículos e imágenes guardadas en la aplicación Web.
<b>Prioridad:</b> Baja.	<b>Riesgo en desarrollo:</b> bajo.
<b>Descripción:</b> Elaborar una opción en la página Web que permita realizar comentarios de artículos e imágenes guardadas por usuarios que se encuentran registrados y autenticados en la aplicación.	

<b>Número:</b> 10	<b>Nombre:</b> Crear un artículo en base a artículos existentes.
<b>Prioridad:</b> Medio.	<b>Riesgo en desarrollo:</b> medio.
<b>Descripción:</b> Elaborar una opción en la página Web que permita componer un nuevo artículo de uno o más artículos existentes.	

<b>Número:</b> 11	<b>Nombre:</b> Crear una opción en la extensión de firefox que recomiende páginas Web.
<b>Prioridad:</b> Medio.	<b>Riesgo en desarrollo:</b> medio.
<b>Descripción:</b> Elaborar una opción en la extensión de firefox que le recomiende al usuario aleatoriamente las páginas Web de las cuales otros usuarios han extraído información.	

<b>Número:</b> 12	<b>Nombre:</b> Crear una opción en la página Web que permita exportar artículos.
<b>Prioridad:</b> Medio.	<b>Riesgo en desarrollo:</b> medio.
<b>Descripción:</b> Elaborar una opción en la página Web que exporte los artículos, seleccionados por el usuario, a formato PDF.	

- **Diseño:**

Para cada iteración en la etapa de diseño se añadieron imágenes y códigos de programación de la interfaz de cada módulo. Y en caso de haberlos creado en alguna iteración anterior, se actualizaron, desechando la anterior.

- **Desarrollo:**

En el proceso de desarrollo se decidió seguir los estándares de programación de lenguaje Java en cuanto a la nomenclatura, declaración de variables, espaciado, indentación y documentación. También se acordó nombrar los archivos javascript y XUL en minúscula. Además se acordó utilizar en todos los tipos de archivos una sentencia o comentario por línea.

Los programadores se reunieron para desarrollar y solucionar las historias de usuario correspondiente a cada iteración, guardándose al finalizar cada día los resultados a un repositorio SVN ubicado en el servidor de desarrollo del Centro de Computación Paralela y Distribuida de la Escuela de Computación.

- **Pruebas:**

Las pruebas fueron realizadas por cada historia de usuario del lado del cliente, utilizando el formato de la tabla 2, para escribir los casos de prueba y los resultados obtenidos de la aplicación. Dicha prueba fueron realizadas antes de mostrárselas al cliente (tutor), él cual posteriormente realizaba las pruebas pertinentes para determinar el cumplimiento de los requerimientos o necesidades

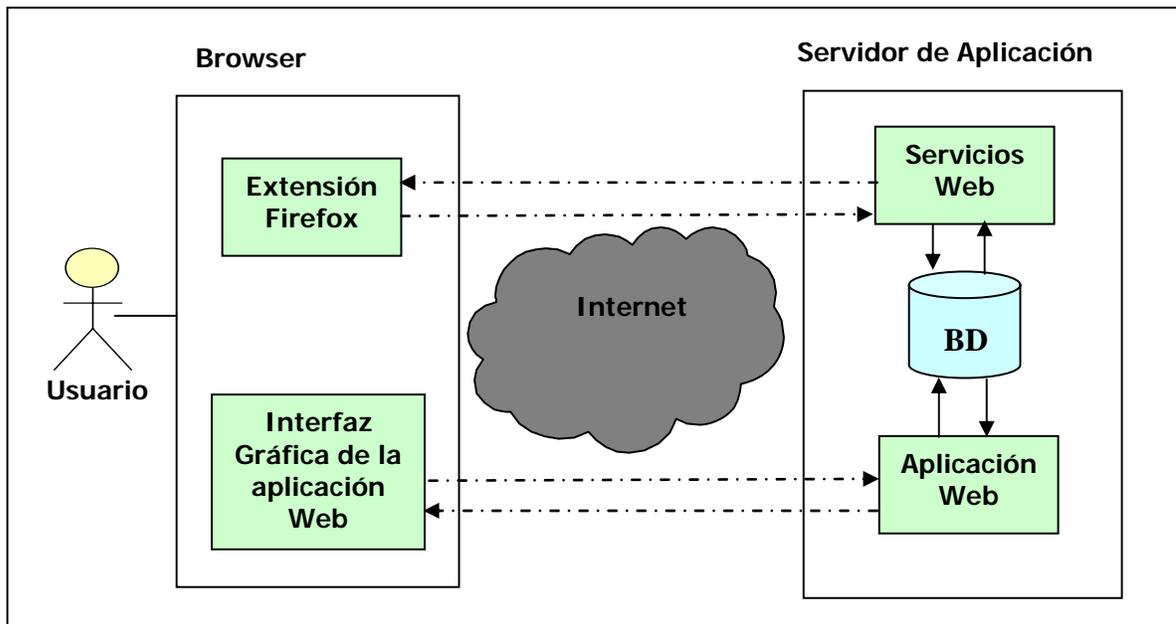
del mismo; aportando ideas y mejoras a la aplicación, las cuales fueron presentadas en las siguientes reuniones.

Id caso prueba	Modulo	Descripción del Caso de Prueba	Resultado Esperado	Resultado Obtenido	Motivo de la Falla	Observaciones

**Tabla 2: Formato de registro de pruebas unitarias del lado del cliente.**

### 2.1. Estructura del Sistema.

Los nuevos usuarios del sistema accederán a una página promocional donde podrán registrarse y obtener la extensión de firefox para instalar en su navegador. En este trabajo especial de grado no se elaborará dicha página y se asume que los usuarios ya poseen la extensión instalada. En la figura 6 se muestran los módulos por los cuales está compuesto el sistema y la secuencia de comunicaciones que se seguirá.



**Figura 6: Metáfora (Estructura del Sistema).**

- **Extensión Firefox:**

Complemento del navegador Mozilla Firefox para la obtención y almacenamiento de información relevante por el usuario a fin de tener organizada su data y poder posteriormente recuperarla.

- **Interfaz Gráfica de la Aplicación Web:**

Conjunto de páginas Web al servicio de los usuarios que se compone de un ambiente de diseño sencillo con un área de trabajo adecuada y un conjunto de herramientas y funcionalidades que facilitan la manipulación de dichas páginas Web.

- **Servicios Web:**

Encargados de controlar las diferentes acciones de autenticación, de almacenamiento de imágenes y textos marcados por el usuario, de saber el número de usuarios que han guardado información de la página Web en la que se encuentra un determinado usuario y de ofrecer la aleatoriedad de las páginas guardadas por todos los usuarios.

- **Aplicación Web:**

Proceso encargado de proveer la eficiencia y eficacia de todas las funcionalidades del sistema, para facilitar al usuario la recuperación de la información almacenada, la construcción de artículos basados en otros artículos, la realización de comentarios de determinados artículos y fotos, la exportación de artículos relevantes en formato de Excel y PDF.

## 2.2. Especificaciones técnicas:

El desarrollo del sistema se llevará a cabo utilizando las siguientes herramientas, tecnologías:

- Servidor Web: Apache Tomcat 5.5.
- Procesamiento de Eventos: JavaScript 1.3.
- Intercambio de información cliente-servidor: Servicios Web.
- Lenguaje de Programación: XUL y Java 1.5.0\_02.
- Gestión de usuarios y sus trabajos: SMD MySQL 5.0.

Una vez establecidos los requerimientos iniciales, seleccionadas las primeras historias de usuario a implementar y haber modelado una idea general de cómo funcionaría el sistema, se planifican 3 iteraciones.

## 2.3. Iteración 1

- **Planificación**

Iteración 1	
<b>Descripción</b>	Desarrollo de la aplicación cliente en lenguaje XUL que el usuario utilizará para la extracción y almacenamiento de la información de las páginas Web.
<b>Historia de usuario a desarrollar</b>	<p>1.- Elaborar una aplicación cliente para la extracción de datos de páginas Web.</p> <p>4.- Permitir conocer el número de usuarios que han marcado información de una página Web.</p> <p>11.- Crear una opción en la extensión de firefox que recomiende páginas Web.</p>

---

<b>Tiempo estimado</b>	15 días.
<b>Fecha Inicio/Fin</b>	01-07-2008 / 15-07-2008

- **Tareas por Historia de Usuario**

HU 1.- Elaborar una aplicación cliente para la extracción de datos de páginas Web.

- Hacer diseño de la extensión de firefox.
- Desarrollar proceso de extracción de datos de la página Web usando XPath.

HU 4.- Permitir conocer el número de usuarios que han marcado información de una página Web.

- Diseñar barra indicadora del porcentaje de usuarios.
- Desarrollar el proceso de obtención de la cantidad de usuarios para ser mostrado en la barra indicadora.

HU 11.- Crear una opción en la extensión de firefox que recomiende páginas Web.

- Diseñar botón encargado del cambio aleatorio de página Web.
- Desarrollar el cargado de la página Web seleccionada de forma aleatorio del pool de páginas almacenadas en el sistema.

- **Diseño:**

En base a los requerimientos y opciones a ofrecer al usuario se diseñó una interfaz fácil de usar y de entender. Luego de realizar varios diseños y discutirlos con el usuario, el diseño final de la barra de herramientas es mostrada en la figura 7, en donde se pueden encontrar dos botones y una imagen, ubicados en

la parte baja del navegador y del margen derecho hacia el izquierdo, primero se encuentra el botón de páginas aleatorias identificado por EE, luego se encuentra la barra indicadora del porcentaje de usuarios y por último el botón que activa la extensión firefox para la extracción de la información.



**Figura 7: Barra de Herramientas.**

La barra de herramientas se obtiene con el siguiente código en lenguaje XUL (figura 8).

```
<statusbar id="status-bar">
<statusbarpanel
  id="status-panel"
  class="statusbarpanel-menu-iconic"
  tooltiptext="Easy Extractor"
  onclick="Function.open();"/>
<image
  src='chrome://easyextractor/skin/0.jpg'
  id="porcent-panel"
  tooltiptext="0%"
  width='50'
  height='22'/>
<statusbarpanel
  id="random-panel"
  class="statusbarpanel-menu-iconic"
  tooltiptext="Paginas Aleatorias"
  onclick="Function.random();"/>
</statusbar>
```

**Figura 8: Diseño de la Barra de Herramientas.**

La extensión de firefox esta compuesta por tres ventanas principales:

- Ingresar en el sistema usando los datos de usuario creado en la página Web (figura 9).
- Selección de la información a almacenar (figura 11).

- Llenar formulario para el guardado de la información seleccionada (figura 13).



**Figura 9: Ventana de Ingreso en la Extensión Firefox.**

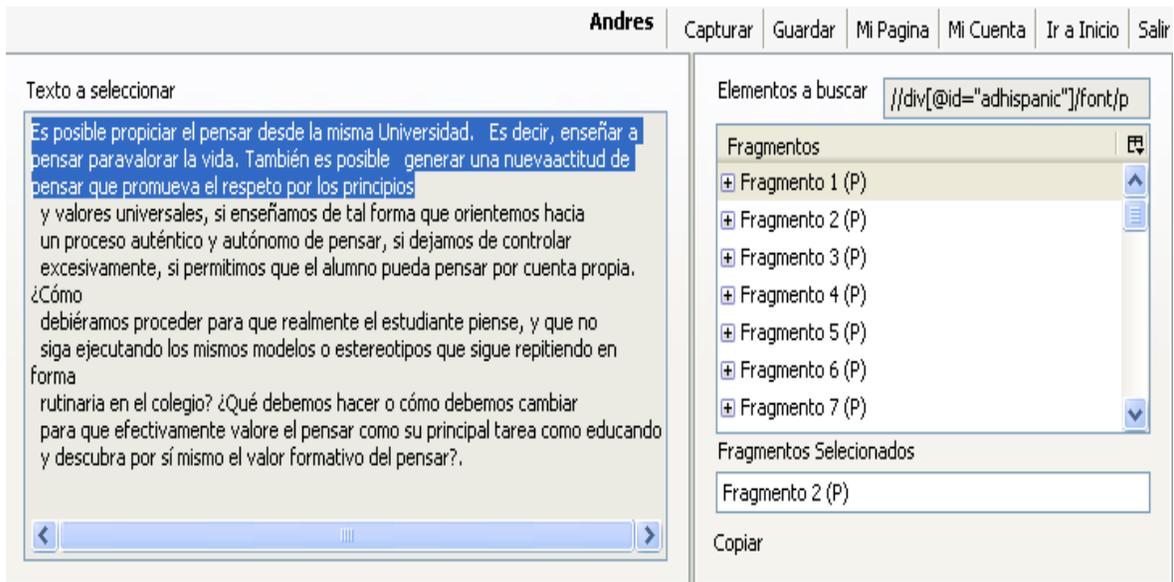
El siguiente código en lenguaje XUL corresponde al diseño de la ventana de ingreso, figura 10.

```

<toolbar id="principal" hidden="true" flex="1" fullscreentoolbar="true"
insertafter="FindToolbar">
  <vbox flex="1">
    <hbox id="box3">
      <hbox id="box1">
        <toolbarbutton id="boton" label="Salir"
          class="hoverable nolabel" tooltip="Salir!!!"
          oncommand="Function.close();" />
      </hbox>
      <spacer flex="1" />
    </hbox>
    <label value="Texto a seleccionar" />
    <textbox flex="1" id="code-textbox" multiline="true" readonly="true" /-->
    <hbox flex="1">
      <tabpanels flex="1" id="panel">
        <!-- ===== Logo ===== -->
        <image src='chrome://easyextractor/skin/logo.jpg' width='100' height='100'/>
      </tabpanels>
      <!-- ===== Panel de Login y Passwird ===== -->
      <tabpanels flex="1" id="login-panel">
        <vbox>
          <label id="loginError" value=" " />
          <hbox>
            <label value="Login " />
            <textbox id="login"/>
          </hbox>
          <hbox>
            <label value="Password " />
            <textbox id="password"/>
          </hbox>
          <hbox>
            <toolbarbutton id="boton-save" label="Entrar"
              class="hoverable nolabel"
              oncommand="Function.enter();"/>
            <toolbarseparator />
            <toolbarbutton id="boton-save" label="Olvido su clave?"
              class="hoverable nolabel"
              oncommand="Function.recovery();"/>
          </hbox>
        </vbox>
      </tabpanels>
    </hbox></vbox></toolbar>

```

Figura 10: Diseño del ingreso en la extensión firefox.



**Figura 11: Ventana de Selección de la Información a Almacenar.**

El siguiente código es desarrollado en lenguaje XUL y le muestra al usuario la información que puede almacenar (figura 12).

```

<tabpanel flex="1" id="left-panel"> <vbox>
    <label value="Texto a seleccionar" />
    <textbox flex="1" id="code-textbox" multiline="true" readonly="true" />
</vbox></tabpanel>
<tabpanel flex="1" id="xpath-panel" hidden="true">
    <!-- ===== Capture Tab ===== -->
    <vbox> <hbox>
        <label value="Elementos a buscar" />
        <textbox flex="1" id="xpath-textbox" readonly="true"
            oninput="Capture.onXPathTextboxInput();" />
    </hbox>
    <deck id="ee-scrapeToolbar-code-captureTab-domTreeDeck" flex="1">
        <tree id="ee-scrapeToolbar-code-captureTab-friendly-tree" flex="1"
            enableColumnDrag="true"
            onselect="DOM.onFriendlyTreeSelect();">
            <treecols>
                <treecol id="item" label="Item" flex="1" primary="true" />
            </treecols>
            <treechildren /></tree>
        <tree id="ee-scrapeToolbar-code-captureTab-dom-tree" flex="1"
            enableColumnDrag="true" onselect="DOM.onTreeSelect();"><treecols>
            <treecol id="tag" label="Tag" flex="1" primary="true" />
            <splitter class="tree-splitter" />
            <treecol id="text" label="Text / URL" flex="3" />
            <splitter class="tree-splitter" />
            <treecol id="var" label="Variable" flex="1" />
            <splitter class="tree-splitter" />
            <treecol id="class" label="Class" flex="1" hidden="true" />
        </treecols>
        <treechildren />
    </tree> </deck>
        <label value="Items Seleccionados" />
        <textbox flex="0.5" id="items-selected" value="" readonly="readonly" />
        <hbox id="box1">
            <toolbarbutton id="boton-copiar" label="Copiar" class="hoverable nolabel"
                oncommand="DOM.copyDocument();" />
        </hbox></vbox>
    <!-- ===== Console Tab ===== -->
    <vbox id="ee-scrapeToolbar-code-consoleTab-console-vbox">
        <vbox id="ee-scrapeToolbar-code-consoleTab-consoleInternal-vbox" />
    </vbox>
    <!-- ===== Results Tab ===== -->
    <textbox id="ee-scrapeToolbar-code-resultsTab-textbox" multiline="true" readonly="true" />
</tabpanel>

```

**Figura 12: Diseño de la selección de la información a almacenar.**

**Figura 13: Ventana del Formulario de Almacenaje de Información.**

Para obtener el formulario con la información necesaria para ser almacenada se utilizó el siguiente código desarrollado en XUL (figura 14).

- **Desarrollo**

En esta iteración se utilizó el lenguaje XUL para el desarrollo de las interfaces de usuario y javascript para el manejo de los eventos.

En la parte superior de la extensión se encuentra una barra de botones, en la cual se encuentra el botón capturar, el cual permite a los usuarios seleccionar un contenido de la página Web, usando XPath (figura 15), dando como resultado un árbol de nodos y mostrándose en el panel derecho, en donde el usuario seleccionará los nodos que desea almacenar. Por último a partir de esos nodos seleccionados se va a poder seleccionar la información final que el usuario guardará.

```

<!-- ===== Panel de Guardar lo seleccionado===== -->
<tabpanel flex="1" id="save-panel" hidden="true">
  <vbox>
    <label value="Debe seleccionar el texto o imagenes a guardar"
      id="text-images-error" hidden="true"/>
    <label value="Texto a Almacenar" />
    <textbox id="text" multiline="true" readonly="true" />

    <label value="Imagenes a Almacenar" />
    <textbox id="images" multiline="true" readonly="true" />

    <hbox>
      <label value="Titulo " />
      <textbox id="title"/>
      <label value="Indique el Titulo bajo el cual seran almacenados el texto y
las imagenes" id="title-error" hidden="true"/>
    </hbox>
    <hbox>
      <label value="Tag " />
      <textbox id="tag"/>
      <label value="Deben separarse por comas (,) " />
      <label value="Indique los Tags bajo el cual seran almacenados el texto y
las imagenes" id="tag-error" hidden="true"/>
    </hbox>
    <hbox>
      <label value="Categoria " />
      <menulist id="category">
        <menupopup>
          <menuitem value="1" label="Artes" selected="true"/>
          <menuitem value="3" label="Ciencias"/>
          <menuitem value="4" label="Deporte"/>
          <menuitem value="6" label="Economia"/>
          <menuitem value="8" label="Educacion"/>
          <menuitem value="2" label="Entretenimiento"/>
          <menuitem value="5" label="Politica"/>
          <menuitem value="7" label="Social"/>
        </menupopup></menulist></hbox>
    <hbox>
      <toolbarbutton id="boton-save" label="Guardar"
        class="hoverable nolabel" oncommand="Function.save();"/>
      <toolbarseparator />
      <toolbarbutton id="boton-cancel" label="Cancelar"
        class="hoverable nolabel"/>
    </hbox> </vbox></tabpanel>

```

**Figura 14: Diseño del Formulario de Almacenaje de Información.**

**//captura el evento click en la sección de la página Web**

```

Capture._capturingMouseListener = {
  _captured : false,
  handleEvent: function(aEvent) {
    var elmt = aEvent.target;
    if (!this._captured && aEvent.type != "click" && elmt != Capture._capturedNode) {
      Capture._captureNode(elmt);
    }
    if (aEvent.type == "mousedown") {
      this._captured = true;
    } else if (aEvent.type == "click" || (aEvent.type == "mouseover" && this._captured)){
      aEvent.stopPropagation();
      aEvent.preventDefault();
      aEvent.cancelBubble = true;
      Capture._doneCapturingXPath();
    }
  }
}

```

**//crea el XPATH del nodo seleccionado**

```

Capture._captureNode = function(node) {
  var xpath = "";
  var namespace = node.ownerDocument.documentElement.namespaceURI;
  var prefix = namespace ? "x:" : "";
  var node2 = node;
  var doc = Capture._currentDoc;
  while (node2 && node2 != doc) {
    var tag = node2.tagName.toLowerCase();
    var id = node2.id;
    var className = node2.className;
    var segment = prefix + tag;
    if (tag == "tr") {
      var rowCount = node2.parentNode.rows.length;
      if (rowCount > 1 && rowCount < 5) {
        segment += '[' + (node2.rowIndex+1) + ']';
      }
    } else if (tag == "td") {
      var cellCount = node2.parentNode.cells.length;
      if (cellCount > 1 && cellCount < 5) {
        segment += '[' + (node2.cellIndex+1) + ']';
      }
    }
  }
}

```

```

if (className && className != "") {
    var classes = className.split(" ");
    var newClasses = [];
    for (var c = 0; c < classes.length; c++) {
        var cl = classes[c];
        if (cl.indexOf("simile-ee-") < 0) {
            newClasses.push(cl);
        }
    }
    if (newClasses.length > 0) {
        segment += '@class="' + newClasses.join(" ") + '"';
    }
}
if (id && id != "") {
    xpath = "/" + segment + '@id="' + id + '"' + xpath;
    break;
}
xpath = "/" + segment + xpath;
node2 = node2.parentNode;
}
Capture.getXPathTextbox().value = xpath;
Capture.highlightNode(node);
DOM.constructTree([ node ]);
Capture._capturedNode = node;
}
//resalta el nodo seleccionado
Capture.highlightXPath = function(skipReconstructingDOM) {
    Capture.clearXPathHighlighting();
    var doc = Code.instrumentContentDocument();
    var xpath = Capture.getXPathTextbox().value;
    var nodes = [];
    try {
        nodes = Capture._getNodesOnXPath(doc, xpath);
        for (var i = 0; i < nodes.length; i++) {
            Highlighter.highlight(nodes[i], Code.highlightStyleClassName, function(m) {
                m.className = "captured";
            });
        }
        Capture.getXPathTextbox().className = "";
    } catch (e) {
        // if the xpath evaluation triggers an error, we signal it by changing the background
        // of the xpath text fields
        Capture.getXPathTextbox().className = "bad";
    }
    if (!skipReconstructingDOM) {
        DOM.constructTree(nodes);
    }
    Capture._highlighting = true;
};

```

```
//obtiene los nodos correspondientes al XPATH
Capture._getNodeOnXPath = function(doc, xpath) {
  var nodes = [];
  var namespace = doc.documentElement.namespaceURI;
  var nsResolver = namespace ? function(prefix) {
    return (prefix == 'x') ? namespace : null;
  } : null;
  var iterator = doc.evaluate(xpath, doc, nsResolver, XPathResult.ANY_TYPE,null);
  var node = iterator.iterateNext();
  while (node) {
    nodes.push(node);
    node = iterator.iterateNext();
  }
  return nodes;
}
```

**Figura 15. Código fuente de la extracción XPath**

- **Pruebas**

Las pruebas relacionadas a las historias de usuario de esta iteración se encuentran descritas y detalladas en la tabla 3, la cual tiene 5 campos: el módulo, la descripción de la prueba, el resultado esperado, el resultado obtenido y el motivo de la falla.

Las pruebas más importantes para este módulo son:

- Validación de formulario de entrada al sistema.
- Validación de formulario de almacenaje de información.
- Verificación de la selección de fragmentos a almacenar.

Módulo	Descripción de la Prueba	Resultado Esperado	Resultado Obtenido	Motivo de la Falla
Extensión Firefox	Validación de formulario de entrada al sistema	Entrada al sistema de manera exitosa y correcta	Entrada al sistema de manera exitosa y correcta	
Extensión Firefox	Validación de formulario de almacenaje de información	Guardado de contenido exitoso	Guardado de contenido exitoso	
Extensión Firefox	Verificación de la selección de fragmentos a almacenar	Seleccionar los fragmentos que el usuario realmente quiere guardar	Selección los fragmentos que el usuario realmente quiere guardar	

**Tabla 3: Pruebas relacionadas a las historias de usuario de la iteración 1.**

## 2.4. Iteración 2

- **Planificación**

<b>Iteración 2</b>	
<b>Descripción</b>	Desarrollo de servicios Web, que se utilizaran para el control de las diferentes acciones que ofrece la Extensión Firefox.
<b>Historia de usuario a desarrollar</b>	2.- Realizar el almacenaje de la información seleccionado por el usuario.
<b>Tiempo estimado</b>	7 días.
<b>Fecha Inicio/Fin</b>	16-07-2008 / 23-07-2008

- **Tarea por Historia de Usuario**

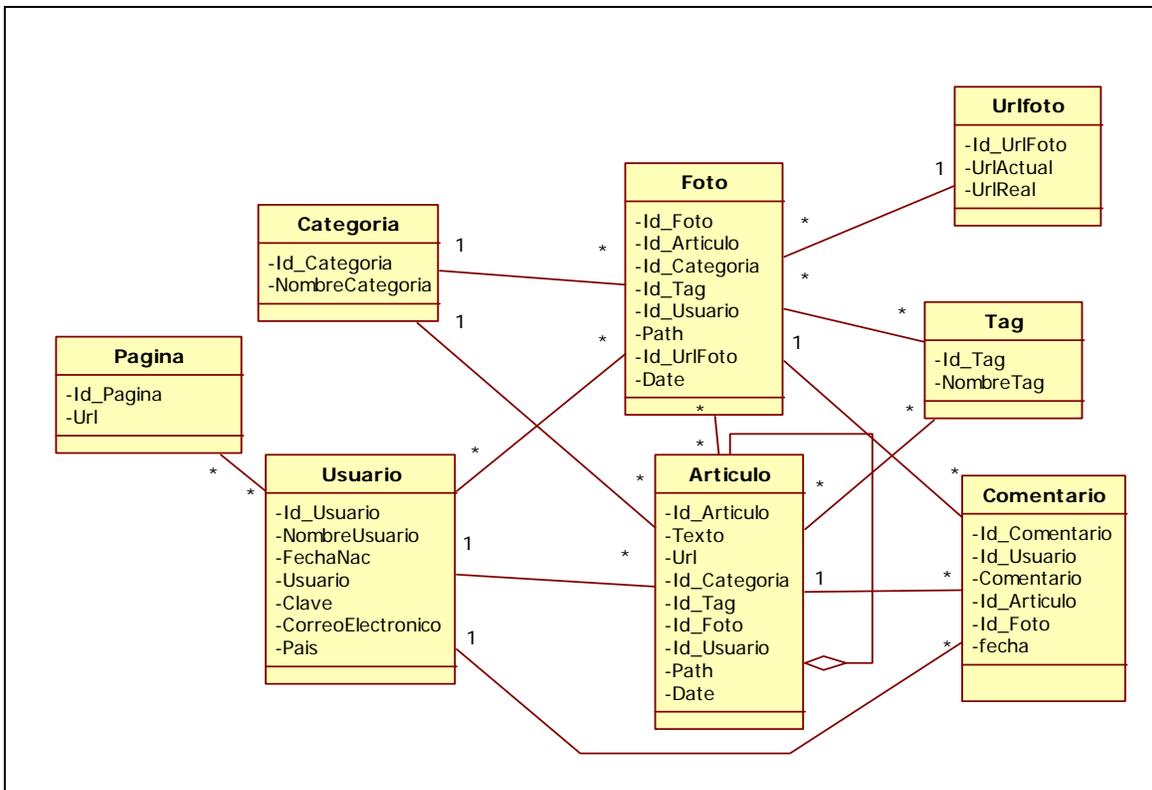
HU 2.- Realizar el almacenaje de la información seleccionado por el usuario.

- Desarrollar proceso de almacenaje de los datos de la página Web seleccionado por el usuario para ser guardado en su sesión.

- **Diseño:**

En esta iteración no se realizó ninguna interfaz para el sistema, ya que se utiliza la interfaz realizada en la iteración anterior para invocar las funcionalidades desarrolladas.

En esta iteración se trabajó con la base de datos, por lo que se realizó el diseño de la misma, determinando cuales son los datos y tablas necesarias para el almacenaje de todos los datos requeridos, obteniendo el diagrama mostrado en la figura 16.



**Figura 16: Diagrama de clases persistentes.**

- **Desarrollo:**

En esta iteración se utilizó el lenguaje Java para el desarrollo de las funcionalidades.

La extensión firefox invoca los servicios Web encargados del almacenaje y obtención de la información de la base de datos, utilizando REST, es decir son invocados a través de un URL pasando los datos de entrada como parámetros del URL. Un ejemplo de este tipo de llamada se muestra en la figura 17, la cual invoca el servicio Web que determina si el usuario que intenta ingresar se encuentra registrado o no, devolviendo en caso afirmativo los datos del usuario (figura 18).

```
var client = new XMLHttpRequest();
var login = document.getElementById("login").value;
var pass = document.getElementById("password").value;
client.open("GET",
"http://127.0.0.1:8080/axis2/services/UserSearch/search?login="+login+"&password="+p
ass, false);
client.send("");
```

**Figura 17: Llamada REST.**

```

public String search(String login, String password){
    String driver="",url_conexion="", usuario_conexion="", clave_conexion="";
    String query="";
    String resultado = "";
    driver="com.mysql.jdbc.Driver";
    url_conexion="jdbc:mysql://localhost:3306/easyExtractor";
    usuario_conexion="userEasyE";
    clave_conexion="passEasyE";
    query="SELECT * FROM usuario WHERE Usuario='"+login+" ' and
Clave='"+password+" ' ";
    try{
        Class.forName(driver);
        Connection db = DriverManager.getConnection(url_conexion, usuario_conexion,
clave_conexion);
        Statement st = db.createStatement();
        ResultSet rs = st.executeQuery(query);
        rs.first();
        if(rs.first()){
            resultado += rs.getString("Id_Usuario")+ " - " +rs.getString("NombreUsuario");
        }
        rs.close();
        st.close();
        db.close();
    }catch(Exception e){
        e.printStackTrace();
    }
    return resultado;}

```

**Figura 18: Llamada REST (usuario registrado).**

- **Pruebas:**

La prueba relacionada a la historia de usuario de esta iteración se encuentra descrita y detallada en la tabla 4, la cual tiene 5 campos: el módulo, la descripción de la prueba, el resultado esperado, el resultado obtenido y el motivo de la falla.

La prueba para este módulo es:

- Validación del proceso de almacenaje de los datos de la página Web seleccionado por el usuario para ser guardado en su sesión.

Módulo	Descripción de la Prueba	Resultado Esperado	Resultado Obtenido	Motivo de la Falla
Servicios Web	Validación del proceso de almacenaje de los datos de la página Web seleccionado por el usuario para ser guardado en su sesión	Almacenaje exitoso del contenido seleccionado por el usuario	Almacenaje exitoso del contenido seleccionado por el usuario	

**Tabla 4: Prueba relacionada a la historia de usuario de la iteración 2.**

### 2.5. Iteración 3

- **Planificación**

Iteración 3	
<b>Descripción</b>	Desarrollo de la interfaz gráfica de la aplicación Web que se compone de un ambiente de diseño sencillo con un área de trabajo adecuada y un conjunto de herramientas y funcionalidades que facilitan la manipulación de dicha aplicación.
<b>Historia de usuario a desarrollar</b>	3.- Elaborar una interfaz de usuario para la consulta de la información guardada por los usuarios.
<b>Tiempo estimado</b>	8 días.
<b>Fecha Inicio/Fin</b>	23-07-2008 / 31-07-2008

- **Tareas por Historia de Usuario**

HU 3.- Elaborar una interfaz de usuario para la consulta de la información guardada por los usuarios.

- Diseñar la interfaz de la aplicación Web donde los usuarios podrán recuperar la información almacenada desde la extensión Firefox.

- **Diseño:**

En esta iteración se diseñó la interfaz de la página Web donde los usuarios pueden consultar la información almacenada por él y por otros usuarios. Dicha interfaz consta de un banner y un menú principal ubicados en la parte superior de la pantalla (figura 19) y banner secundario ubicado en el lateral izquierdo de la pantalla (figura 20).



**Figura 19: Menú Superior**



**Figura 20: Menú Lateral**

El menú superior es modificado cuando el usuario se autentifica en el sistema (figura 21), para ofrecer nuevas opciones a nuestros usuarios.

Andres | [Editar](#) | [Cerrar Sesión](#)



**Figura 21: Menú Superior del usuario autenticado.**

- **Desarrollo:**

En esta iteración se utilizó Java Server Pages (JSP) para el desarrollo de la interfaz y cada .jsp consta de la inclusión de 3 páginas, las cuales son header.jsp (figura 22), superior\_menu.jsp (figura 24) y lateral\_menu.jsp (figura 23) y en la parte central del .jsp se muestra la información acorde a la funcionalidad.

```
<%@ page import="com.ee.form.*, java.util.ArrayList" %>
<%UserForm u=(UserForm)request.getSession().getAttribute("user");%>
<table>
  <tr>
    <td align="right" class="Estilo3"><a
href="/EasyExtractor/views/login.jsp">Iniciar Sesi&oacute;n</a></td>
    <%} else { %>
    <td align="right" class="Estilo3"><%=u.getName() %>&nbsp;| <a
href="/EasyExtractor/views/edit_profile.jsp">Edit</a>&nbsp;| <a
href="/EasyExtractor/close.do">Cerrar Sesi&oacute;n</a></td>
    <%} %>
  </tr>
  <tr>
    <td colspan="2"></td>
  </tr>
  <tr>
    <td colspan="2"></td>
  </tr>
</table>
```

**Figura 22: Código fuente del archivo header.jsp.**

```

<%if(session.getAttribute("type")!=null){
    java.util.ArrayList
    cats=(java.util.ArrayList)session.getAttribute("categorys");
%>
<table align="left" border="1">
    <tr>
        <td align="center" class="Estilo7"
background="/EasyExtractor/images/subtitulos.jpg">Categorias</td>
    </tr>
    <%for(int z=0;z<cats.size();z++){
        com.ee.form.CategoryForm
        c2=(com.ee.form.CategoryForm)cats.get(z);
    %>
        <tr>
            <td bgcolor="#0066FF" class="Estilo7">&nbsp;
                <a
href="/EasyExtractor/search.do?idcat=<%=c2.getId()%>"><%=c2.g
etName() %></a></td>
        </tr>
    <%} %>
        <tr>
            <td bgcolor="#0066FF" class="Estilo7">&nbsp;
                <a href="/EasyExtractor/search.do?idcat=0">Todas</a>
            </td></tr></table><%}%>

```

Figura 23 Código fuente del archivo lateral\_menu.jsp.

```

<td colspan="3">
<% if(session.getAttribute("user")!=null){%>
<table width="1024">
  <tr>
    <td>&nbsp;</td>
    <td class="Estilo8">
      <a href="/EasyExtractor/index.do">Inicio</a>
      |<a href="/EasyExtractor/last10.do?art=1">Articulos</a>
      |<a href="/EasyExtractor/last10.do?ph=1">Imágenes</a>
      |<a href="/EasyExtractor/last10.do?myart=1">Mis
      Articulos</a>
      |<a href="/EasyExtractor/last10.do?myph=1">Mis Imágenes</a>
    </td>
    <%if(request.getAttribute("articles")!=null){ %>
      <td class="Estilo8"><a href="javascript:checkPdf()">Exportar
      Pdf</a></td>
    <%} %>
    <%if(request.getAttribute("articles")!=null){ %>
      <td>|</td>
      <td class="Estilo8"><a href="javascript:check()">
      Componer</a></td>
    <%} %>
    <td>&nbsp;</td>
    <form action="/EasyExtractor/search.do">
      <td align="center" class="Estilo3">Buscar<input type="text"
      name="search"><input type="submit" value="Ir"/></td>
    </form>
  </tr>
</table><br>
<%} else{ %>
  <table width="1024" border="0">
    <tr align="center">
      <td>&nbsp;</td>
      <td class="Estilo8">
        <a href="/EasyExtractor/index.do">Inicio</a>
        |<a href="/EasyExtractor/last10.do?art=1">Articulos</a>
        |<a href="/EasyExtractor/last10.do?ph=1">Imágenes</a>
      </td>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
      <form action="/EasyExtractor/search.do">
        <td align="center" class="Estilo3">Buscar
          <input type="text" name="search">
          <input type="submit" value="Ir"/>
        </td>
      </form>
    </tr>
  </table><br>
<%} %>
</td>

```

Figura 24: Código fuente del archivo superior\_menu.jsp.

- **Pruebas:**

Las pruebas relacionadas a la historia de usuario de esta iteración se encuentran descritas y detalladas en la tabla 5, la cual tiene 5 campos: el módulo, la descripción de la prueba, el resultado esperado, el resultado obtenido y el motivo de la falla.

La prueba importante para este módulo es:

- Verificación de la existencia de las vistas a utilizar en la aplicación Web.

<b>Módulo</b>	<b>Descripción de la Prueba</b>	<b>Resultado Esperado</b>	<b>Resultado Obtenido</b>	<b>Motivo de la Falla</b>
Interfaz Gráfica de la aplicación Web	Verificación de la existencia de las vistas a utilizar en la aplicación Web	Existencia de todas las vistas a utilizar en la aplicación Web	Existencia de todas las vistas a utilizar en la aplicación Web	

**Tabla 5: Prueba relacionada a la historia de usuario de la iteración 3.**

## 2.6. Iteración 4

- **Planificación**

<b>Iteración 4</b>	
<b>Descripción</b>	Desarrollo de la aplicación Web encargada de proveer la eficiencia y eficacia de todas las funcionalidades del sistema.
<b>Historia de usuario a desarrollar</b>	5.- Mostrar toda la información marcada por un usuario. 6.- Elaborar un motor de búsqueda de información almacenada por otros usuarios. 7.- Mostrar todos los usuarios que han marcado y almacenado información de una página Web específica.

	<p>8.- Mostrar información del usuario en línea.            9.- Hacer comentarios de artículos guardados en la página Web.            10.- Crear un artículo en base a artículos existentes.            12.- Crear una opción en la página Web que permita exportar artículos.</p>
<b>Tiempo estimado</b>	8 días.
<b>Fecha Inicio/Fin</b>	01-08-2008 / 09-08-2008

- **Tareas por Historia de Usuario**

HU 5.- Mostrar toda la información marcada por un usuario.

- Desarrollar la búsqueda de información e imágenes almacenadas por un usuario.

HU 6.- Elaborar un motor de búsqueda de información almacenada por otros usuarios.

- Desarrollar un buscador de artículos e imágenes que contengan en su título, texto o tags las palabras indicadas por el usuario.
- Elaborar una opción que permita buscar artículos o imágenes de una categoría determinada.

HU 7.- Mostrar todos los usuarios que han marcado y almacenado información de una página Web específica.

- Desarrollar la búsqueda de los usuarios que han guardado información de una página específica.

HU 8.- Mostrar información del usuario en línea.

- Elaborar una opción en la página Web que permita ver y editar los datos del usuario.
- Desarrollar un buscador de artículos e imágenes del usuario registrado.

HU 9.- Hacer comentarios de artículos guardados en la página Web.

- Desarrollar una opción que permita a los usuarios autenticados realizar comentarios a un artículo o imagen determinada.

HU 10.- Crear un artículo en base a artículos existentes.

- Desarrollar una opción en la página Web que permita a los usuarios poder guardar un artículo propio a partir de uno o más artículos previamente seleccionados.

HU 12.- Crear una opción en la página Web que permita exportar artículos.

- Desarrollar una opción que permita a los usuarios exportar a formato PDF los artículos previamente seleccionados.
- **Diseño:**

Cada funcionalidad posee su correspondiente archivo *JSP* de salida, como por ejemplo: página de inicio que muestra los últimos 10 artículos almacenados en el sistema (figura 25), búsqueda de artículos e imágenes en base a palabras escritas por el usuario (figura 26), mostrar el contenido del artículo seleccionado (figura 27), ingresar al sistema (figura 28), registrarse como un usuario nuevo (figura 29), entre otros.



**Figura 25: Página Principal de la Aplicación Web.**

# Easy Extractor



Inicio | Artículos | Imágenes

Buscar

**Artículos**

**[Caso De Estudio 3](#)**

The bursting of the dot-com bubble in the fall of 2001 marked a turning point for the web. Many people concluded that the web was overhyped, when in fact bubbles and consequent shakeouts appear to be a common feature of all technological revolutions. Shakeouts typically mark the point at which an ascendant technology is ready to take its place at center stage. The pretenders are given the bums rush, the real success stories show their strength, and there begins to be an understanding of what separates one from the other. The concept of "Web 2.0" began with a conference...

**[Nueva Composicion](#)**

The bursting of the dot-com bubble in the fall of 2001 marked a turning point for the web. Many people concluded that the web was overhyped, when in fact bubbles and consequent shakeouts appear to be a common feature of all technological revolutions. Shakeouts typically mark the point at which an ascendant

**Imágenes**

**[Caso De Estudio 3](#)**

Figura 26: Página de búsqueda.

# Easy Extractor



Inicio | Artículos | Imágenes

Buscar

**La educacion publica anda por la calle de la amargura**

La ideologia de los profesores y de la educacion misma comenzo a cambiar en los anos cincuenta al mismo ritmo en que el pais se convertia de agrario en industrial y de servicios, al mismo tiempo en que la ciudad y la cultura urbana crecieron sobre los sacrificios del campo. Los profesores que entonces se identificaban con los padres de familia y con los problemas de la poblacion que vivia en el campo, con el crecimiento de las ciudades y los medios de comunicacion se integraron con otra cultura y otra ideologia.

Categoría: Educación      Tags: educacion, criticas, mexico

Url: <http://tvnoticias.wordpress.com/2008/04/24/el-gobie>

Usuario: andres

Fecha de Creación: 15-10-2008

Comentarios:

[andres](#) dijo el 19-10-2008:  
Buena Practica!

Figura 27: Artículo seleccionado por el usuario.

Inicio | Artículos | Imágenes

Buscar  Ir

### INICIAR SESIÓN

Usuario

Clave

Entrar

Recordar clave. Haga click [aquí](#)

Aún no tiene cuenta... Regístrese [aquí](#)

Figura 28: Página de Ingreso al sistema.

Inicio | Artículos | Imágenes

Buscar  Ir

### REGISTRO DE USUARIO

Nombre

Usuario

Clave

Repita la Clave

Fecha de Nacimiento    ...

Correo Electrónico  Ej: mariaperez@hotmail.com

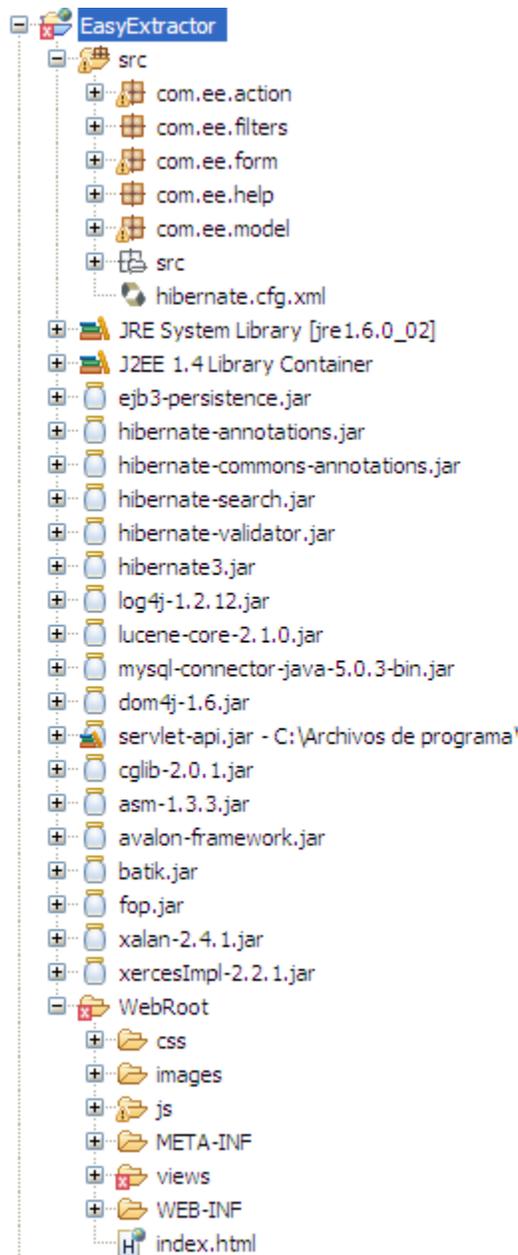
País  Seleccione su país

Registrar

Figura 29: Página de Registro.

- **Desarrollo:**

Esta iteración fue desarrollada siguiendo Modelo Vista Controlador (MVC), usando Java para el Controlador, Hibernate en el Modelo y *JSP* para las vistas, obteniendo la estructura de carpetas mostrada en la figura 30.



**Figura 30: Estructura de carpetas del proyecto en Myeclipse.**

Los controladores se encuentran en la carpeta `com/ee/action` y sus nombres terminan en *Action* e indican la funcionalidad que cumplen, por ejemplo *IndexAction* realiza las búsquedas de los últimos 10 artículos en el sistema para ser mostrados en el index de la página (figura 31). Otro ejemplo sería *CloseAction* que permite cerrar la sesión del usuario autenticado (figura 32). También tenemos *SaveCommentArtAction* que se encarga de almacenar el comentario escrito por el usuario autenticado en un artículo o imagen (figura 33).

```

public void service(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException{
    //declarando la aplicacion
    ServletContext application=getServletContext();

    //Pagina de Inicio del sistema
    Session s=(Session) request.getAttribute("session_hb");

    request.getSession().removeAttribute("type");

    Article art=new Article(s);

    ArrayList list;

    if(request.getSession().getAttribute("user")==null){
        list=art.last10();
    }else{
        UserForm
u=(UserForm)request.getSession().getAttribute("user");
        list=art.last10(u.getId());
    }
    Category c=new Category(s);
    request.getSession().setAttribute("categorys", c.search());
    request.setAttribute("10articles", list);

    //redireccionando
    RequestDispatcher
redireccionador=application.getRequestDispatcher("/views/index.jsp");
    redireccionador.forward(request, response);
}

```

**Figura 31: Código fuente del archivo *IndexAction*.**

```

public void service(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException{
    //declarando la aplicacion
    ServletContext application=getServletContext();

    Session s=(Session) request.getAttribute("session_hb");
    //cerrar la sesion del usuario

    Article a=new Article(s);
    String redirect="";

    request.getSession().removeAttribute("user");
    ArrayList list=a.last10();
    request.setAttribute("10articles", list);
    redirect = "/views/index.jsp";

    request.getSession().removeAttribute("type");

    //redireccionando
    RequestDispatcher
redireccionador=application.getRequestDispatcher(redirect);
    redireccionador.forward(request,response);
}

```

**Figura 32: Código fuente del archivo *CloseAction*.**

```

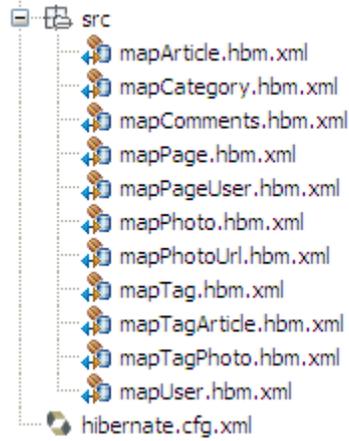
public void service(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    //declarando la aplicacion
    ServletContext application=getServletContext();
    //guardar el comentario a un articulo
    request.getSession().removeAttribute("type");
    Session s=(Session) request.getAttribute("session_hb");
    Article a=new Article(s);
    TagArticle t=new TagArticle(s);
    Photo p=new Photo(s);
    TagPhoto tp = new TagPhoto(s);
    Comment c=new Comment(s);
    String redirect="";
    String comment=request.getParameter("comment");
    if(!comment.equalsIgnoreCase("")){
        ArticleForm
art=(ArticleForm)request.getSession().getAttribute("article");
        UserForm
user=(UserForm)request.getSession().getAttribute("user");
        CommentForm com=new CommentForm();
        com.setId(0);
        com.setComment(comment);
        com.setIdArticle(art.getId());
        com.setIdPhoto(0);
        com.setIdUser(user);
        c.insert(com);
        art.getComments().add(com);
        request.getSession().setAttribute("article", art);
        redirect = "/views/show-article.jsp";
    }else{
        redirect = "/views/show_article.jsp?error=No introdujo el
comentario";
    }
    //redireccionando
    RequestDispatcher
redireccionador=application.getRequestDispatcher(redirect);
    redireccionador.forward(request,response);
}

```

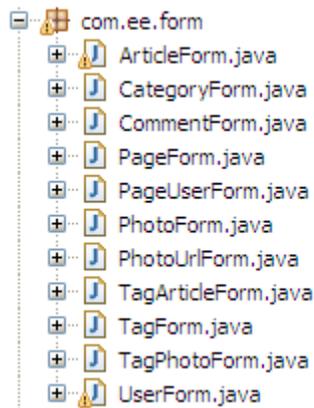
**Figura 33: Código fuente del archivo SaveCommentArtAction.**

El modelado de la base de datos para Hibernate se encuentra en la carpeta *src/* y sus nombres comienzan por *map* y tienen extensión *.hbm.xml*, existiendo uno por cada tabla en la base de datos (figura 34); estos modelos requieren una clase que represente cada uno de los campos de las tablas en java, estas clases se encuentran en la carpeta *com/ee/form* y sus nombres concluyen en *Form* (figura 35); y las clases que se encargan de realizar los

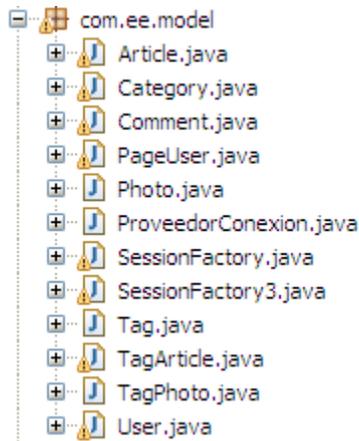
accesos a la base de datos se encuentran en la carpeta *com/ee/model* (figura 36).



**Figura 34: Modelos de la Base de Datos para Hibernate.**



**Figura 35: Clases en la carpeta form.**



**Figura 36: Clases en la carpeta Model.**

Los archivos *JSP* se encuentran en la carpeta views del proyecto (figura 37). El nombre de cada uno permite identificar la función que cumple, por ejemplo *create\_article.jsp* nos muestra un formulario con los datos necesarios para el almacenaje del nuevo artículo, *edit\_profile.jsp* muestra la información del perfil del usuario autenticado y permite modificar algunos campos, *login.jsp* es la página de autenticación al sistema, *recovery\_password.jsp* solicita al usuario su nombre de usuario para poder buscar el password en la base de datos, entre otros.



**Figura 37: JSP en la carpeta Views.**

- **Pruebas:**

Las pruebas relacionadas a la historia de usuario de esta iteración se encuentran descritas y detalladas en la tabla 6, la cual tiene 5 campos: el módulo, la descripción de la prueba, el resultado esperado, el resultado obtenido y el motivo de la falla.

Las pruebas más importantes para este módulo son:

- Verificación de artículos seleccionados por el usuario para componer un nuevo artículo.
- Validación de los datos introducidos en el formulario de autenticación.
- Validación de los datos introducidos en el formulario de registro.
- Validación de los datos introducidos en el formulario de recuperación de password.
- Validación de los datos introducidos en el formulario de modificación del perfil.

- Validación de los datos introducidos en el formulario de creación de un nuevo artículo.
- Verificación de que el usuario haya iniciado sesión para poder realizar comentarios.
- Verificación de artículos seleccionados por el usuario para exportarlos en formato PDF.
- Verificación de la correctitud de los resultados de una búsqueda.

<b>Módulo</b>	<b>Descripción de la Prueba</b>	<b>Resultado Esperado</b>	<b>Resultado Obtenido</b>	<b>Motivo de la Falla</b>
Aplicación Web	Verificación de artículos seleccionados por el usuario para componer un nuevo artículo	Artículos compuestos realizados por los usuarios	Artículos compuestos realizados por los usuarios	
Aplicación Web	Validación de los datos introducidos en el formulario de autenticación	El usuario se autentica exitosamente	El usuario se autentica exitosamente	
Aplicación Web	Validación de los datos introducidos en el formulario de registro	El usuario se registra exitosamente	El usuario se registra exitosamente	
Aplicación Web	Validación de los datos introducidos en el formulario de recuperación de password	El usuario recupera su password de manera exitosa	El usuario recupera su password de manera exitosa	
Aplicación Web	Validación de los datos introducidos en el formulario de modificación del perfil	Edición del perfil del usuario de manera exitosa	Edición del perfil del usuario de manera exitosa	
Aplicación	Validación de	El usuario	El usuario	

Web	los datos introducidos en el formulario de creación de un nuevo artículo	crea un nuevo artículo de manera exitosa	crea un nuevo artículo de manera exitosa	
Aplicación Web	Verificación de que el usuario haya iniciado sesión para poder realizar comentarios	El usuario realiza comentarios de artículos de manera exitosa	El usuario realiza comentarios de artículos de manera exitosa	
Aplicación Web	Verificación de artículos seleccionados por el usuario para exportarlos en formato PDF.	Artículos exportados en formato PDF	Artículos exportados en formato PDF	
Aplicación Web	Verificación de la correctitud de los resultados de una búsqueda.	Obtención de artículos e Imágenes relacionados con el criterio de búsqueda	Obtención de artículos e Imágenes relacionados con el criterio de búsqueda	

**Tabla 6: Pruebas relacionadas a las historias de usuario de la iteración 4.**

En este capítulo se logró apreciar la puesta en práctica de una personalización del proceso de desarrollo XP que incluyó las 4 fases básicas de desarrollo de software (análisis de requerimientos, diseño, implementación y prueba) en iteraciones, previamente planificadas a partir de variables determinantes como prioridad de riesgo y tiempo de desarrollo. Todo lo anterior permitió una implementación organizada, fluida y exitosa del sistema.

Una vez finalizadas las cuatro iteraciones en las que se dividió el proceso de desarrollo, se obtuvo la aplicación Easy Extractor capaz de poder clasificar el contenido Web relevante al usuario y almacenarlo para su próxima revisión.

## **CAPÍTULO III: CASOS DE ESTUDIO Y ENCUESTA**

Una vez finalizado el proceso de desarrollo y prueba del sistema se procedió a realizar un estudio del funcionamiento de la aplicación con tres páginas diferentes y una encuesta a un número de personas, para comprobar la eficacia y eficiencia del sistema realizado en este Trabajo Especial de Grado.

### **3.1. Casos de Estudio**

Para validar el correcto funcionamiento del sistema en contextos reales se utilizaron tres (3) páginas Web.

- **Primer Caso de Estudio:**

Se realizó el proceso de extracción de información de la página Web <http://tvnoticias.wordpress.com/2008/04/24/el-gobierno-ilegitimo-de-calderon-no-hara-nada-para-sacar-a-la-educacion-publica-del-hoyo-en-que-se-encuentra/>, de la cual se desea guardar un fragmento del contenido de dicha página mostrada en la figura 38.

## Agencia de Noticias Independiente (Edomex)

La educación pública anda por la calle de la amargura; Vázquez Mota subordinada a Elba Esther

Publicado en **Agencia de Noticias Independiente (ANI)**, Artículos de Pedro Echeverría V., Educación, Magisterio, Política by Agencia de Noticias Independiente en Abril 24th, 2008

La educación mexicana está en los últimos lugares por culpa de la SEP y del SNTE

Pedro Echeverría V.



1. Yo no se si ya da vergüenza decir que estudiamos en México; quizá por eso aumentan los títulos del extranjero. Todos los informes que llegan de organismos internacionales dicen que estamos en los últimos lugares en todo: en presupuesto aplicado a educación, en calidad educativa (matemáticas, lenguaje,) en porcentaje de investigadores, en proporción de alfabetizados. Quizá por eso los hijos de los millonarios estudian en otros países o esos ricos, con apoyo gubernamental, crean en el país escuelas particulares de alto nivel con cuotas millonarias. Si se siguen publicando esos registros internacionales donde México ocupa los últimos lugares en educación nuestros licenciados y doctores muy poco valor tendrán en el extranjero; peor aún: las investigaciones en nuestra nación no alcanzarán la calidad necesaria para darle prestigio a México. Pareciera que muchos toman esta realidad como chacoteo, pero es una cosa muy seria (no es un campeonato de fútbol), la educación es fundamental para cualquier país.

2. La opinión pública, formada por los medios de comunicación, pero también por su falta de visión y su inmediatez, ha culpado siempre a los profesores del enorme deterioro educativo. Mientras tanto el presidente de la República, la secretaria de Educación y la cacique sindical determinan (sin consultar) las medidas que toman. Sucede a la opinión

### BOLETIN SEMANAL

Diario de Ecatepec Noticias del 1 al 7 de mayo 2008 [Versión impresa](#); [haz clic aquí](#)

### Páginas

[¿Quiéres ser reportero ciudadano?](#)

### CULTURA

### ECATEPEC

[Cambio de rumbo económico y generación de empleos, la solución para reducir la inseguridad](#)

[Policías de Ecatepec arriaron ganado que se escapó de su corral; les tocó hacerla de vaqueros](#)

[En alianza con empresarios se conformará el Consejo Ciudadano de Seguridad Pública de Ecatepec](#)

[La estación del metro](#)

**Figura 38. Página principal (tercer caso de estudio).**

Se hace *click* para la captura del texto con la extensión de firefox. Debido a que la imagen seleccionada se encuentra en una etiqueta "<p>" de HTML, la aplicación sugiere toda la información contenida en las etiquetas "<p>" de la página Web. En este caso devuelve 8 fragmentos de dicha página, en donde solo se necesita el fragmento 4, como se muestra en la figura 39.



Figura 39. Fragmento extraído de la página.

De la información extraída en la parte izquierda de la extensión de firefox, se va guardar solo una parte de dicho fragmento, la cual se muestra en la figura 40.

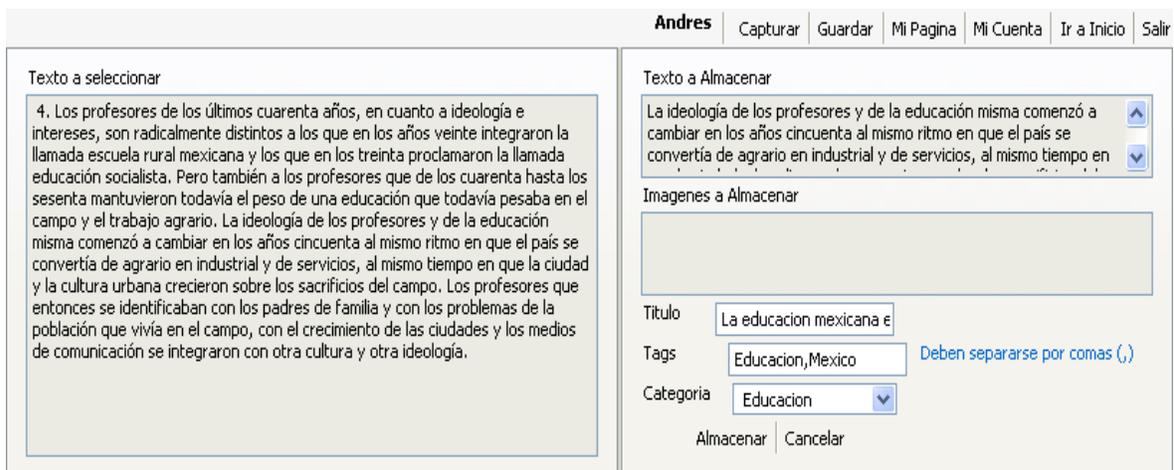
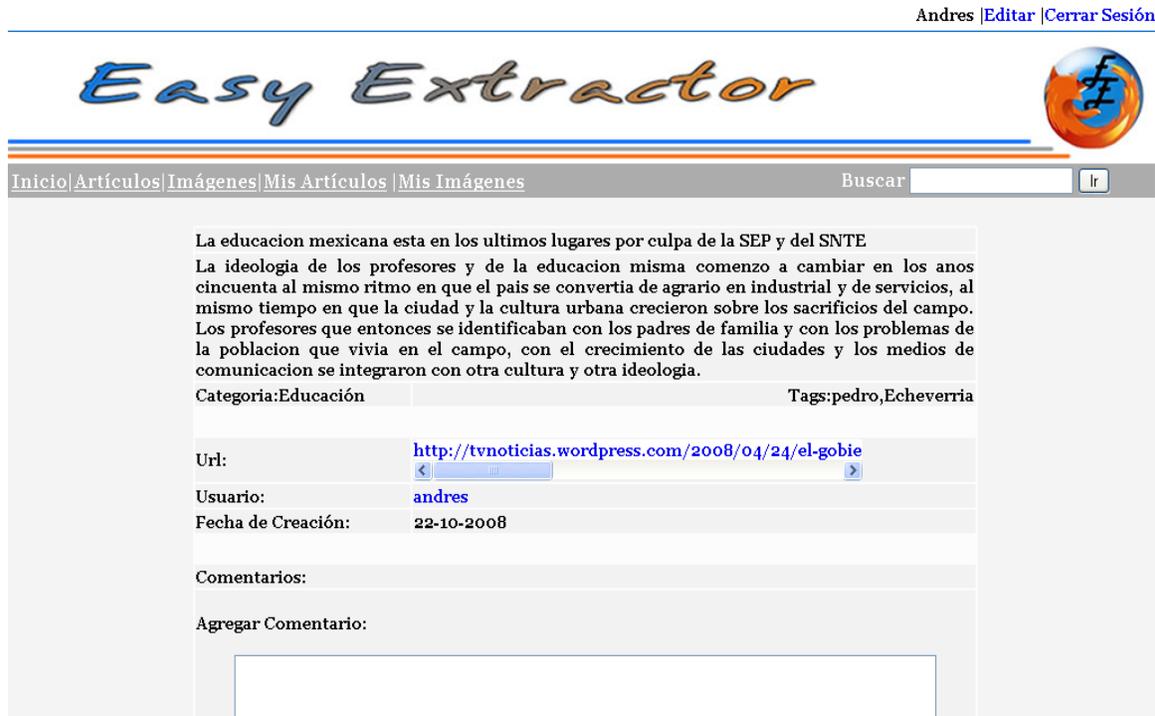


Figura 40. Información a ser almacenada.

Una vez almacenado una parte del fragmento relevante por el usuario, en la página principal de dicho usuario, se verá agregado un nuevo artículo donde

podrá volver a consultarlo y los demás usuarios podrán leerlo y comentarlo, como se muestra en la figura 41.



**Figura 41. Imagen almacenada en la sesión de usuario.**

- **Segundo Caso de Estudio:**

Se realizó el proceso de extracción de información de la página Web <http://www.hp.com/latam/ve/hogar/vivera/index.html>, de la cual se desea guardar la imagen principal contenida en dicha página mostrada en la figura 42.



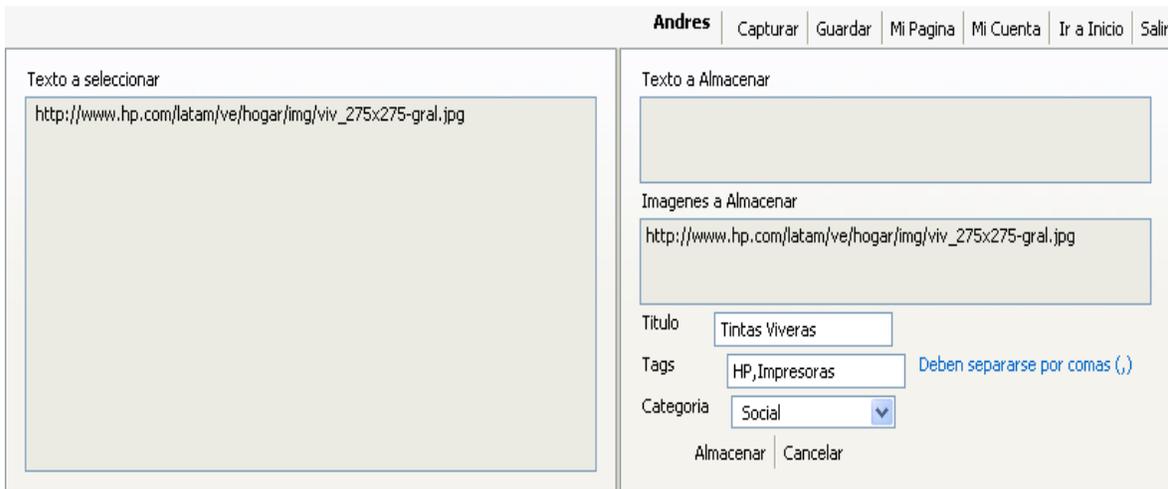
Figura 42. Página principal (segundo caso de estudio).

Se hace *click* para la captura de la imagen con la extensión de firefox. Debido a que la imagen seleccionada se encuentra en una etiqueta "`<img>`" de html la aplicación sugiere toda la información contenida en las etiquetas "`<img>`" de la página Web. En este caso devuelve solo un fragmento de dicha página, por lo cual se tomo solo el fragmento 1, el cual es la imagen "HP, la diferencia en tus impresiones", como se muestra en la figura 43.



**Figura 43. Fragmento extraído de la página.**

De la información extraída en la parte izquierda de la extensión se va a copiar el URL de la imagen mostrado en la figura 44.



**Figura 44. Información a ser almacenada.**

Una vez almacenada la imagen por el usuario, en la página principal de dicho usuario se verá agregada una nueva imagen donde podrá volver a consultarla y los demás usuarios podrán verla y comentarla, como se muestra en la figura 45.

# Easy Extractor



[Inicio](#) | [Artículos](#) | [Imágenes](#) | [Mis Artículos](#) | [Mis Imágenes](#)

Buscar

Tintas Viveras



Categoría: Social

Tags: HP, Impresoras

Url: <http://127.0.0.1:8080/EasyExtractor/search.do?idp=1>

Usuario: andres

Fecha de Creación: 23-10-2008

**Figura 45. Imagen almacenada en la sesión de usuario.**

- **Tercer Caso de Estudio:**

Se realizó el proceso de extracción de información de la página Web <http://oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>, de la cual se desea guardar la información contenida en el artículo "What Is Web 2.0", tanto el texto como la imagen, como se muestra en la figura 46.

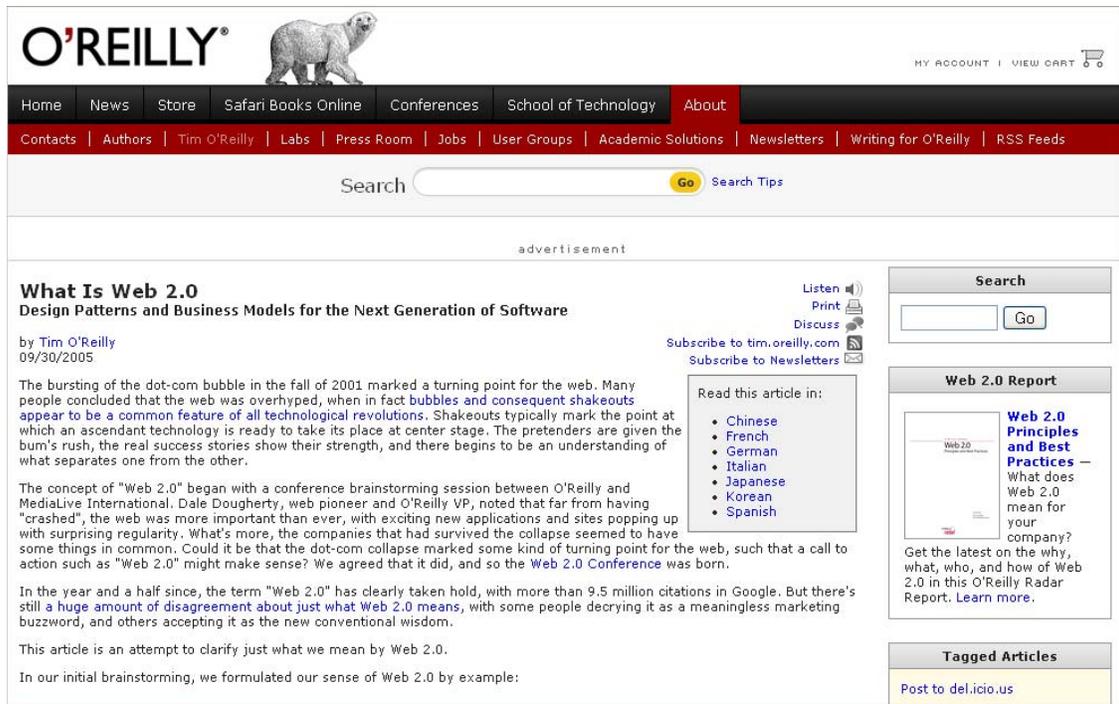


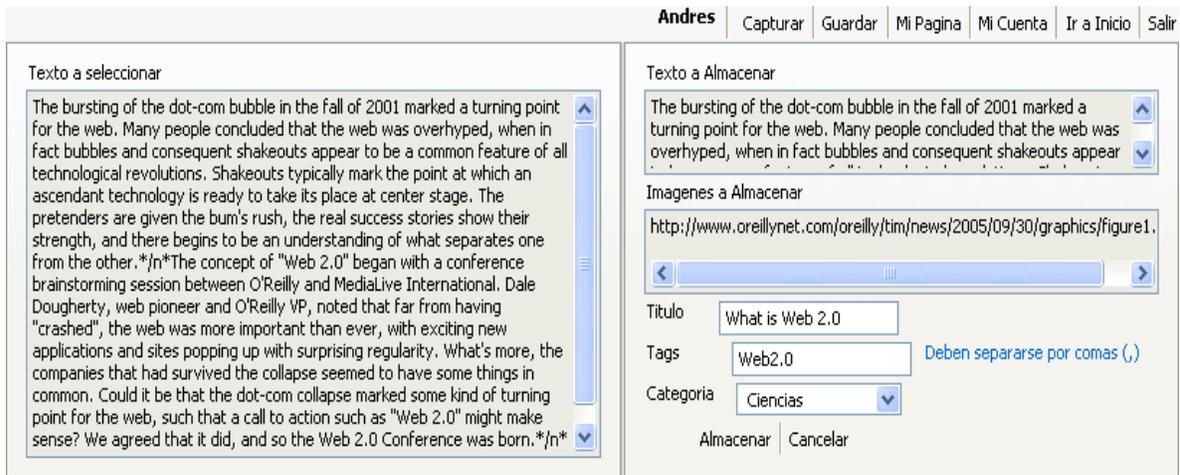
Figura 46. Página principal (tercer caso de estudio).

Al momento de capturar la información con la extensión de firefox, se hace *click* en el primer párrafo del artículo. Debido a que la información seleccionada se encuentra en una etiqueta "`<p>`" de html la aplicación sugiere toda la información contenida en las etiquetas "`<p>`" de la página Web. En este caso devuelve 19 fragmentos que se extrajeron de la página, de los cuales solo se van a copiar los fragmentos 1,2 y 8, los cuales son los dos primeros párrafos del artículo y la imagen "Web 2.0 Meme Map", como se muestra en la figura 47.



**Figura 47. Fragmentos extraídos de la página.**

De la información extraída en la parte izquierda de la extensión se va a copiar todo el contenido obteniendo la información mostrada en la figura 48.



**Figura 48. Información a ser almacenada.**

Una vez almacenada la información por el usuario, en la página principal de dicho usuario en Easy Extractor, se verá agregado un nuevo artículo e imagen donde podrá volver a consultarlo y los demás usuarios podrán leerlo y comentarlo, como se muestra en la figura 49.



**Figura 49. Artículo Almacenado en la Página Web.**

### 3.2. Encuesta

La encuesta se realizó a una población de 20 personas, entre estudiantes y graduados del área de computación, con edades comprendidas entre 20 y 30 años, los cuales son investigadores y realizan trabajos en grupo, convirtiéndose en potenciales usuarios del sistema. La mencionada encuesta es la siguiente:

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.		
2	La terminología del menú es constante en todo el sitio.		

<b>3</b>	Los iconos que aparecen se identifican claramente con lo que representan.		
<b>4</b>	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.		
<b>5</b>	Para actividades similares, los diálogos, formularios son similares.		
<b>6</b>	Se mantiene una navegación consistente y coherente en todas las pantallas.		
<b>7</b>	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.		
<b>8</b>	Se dan indicaciones para completar campos problemáticos.		
<b>9</b>	Los títulos son descriptivos y distintivos.		
<b>10</b>	Los tipos y tamaños de letra son legibles y distinguibles.		
<b>11</b>	El mensaje de error informa cuales son las acciones correctoras.		
<b>12</b>	El acceso a la ayuda está en una zona visible y reconocible.		
<b>13</b>	La demarcación de la información de la página Web es intuitiva		
<b>14</b>	La búsqueda de información es fácil y rápida.		
<b>15</b>	Es fácil identificar las acciones que se pueden realizar en cada pantalla.		

El rango de valores es el siguiente:

0 = No se han detectado errores de este tipo.

1 = Sin importancia. Intentar solucionarlo solo si sobra tiempo.

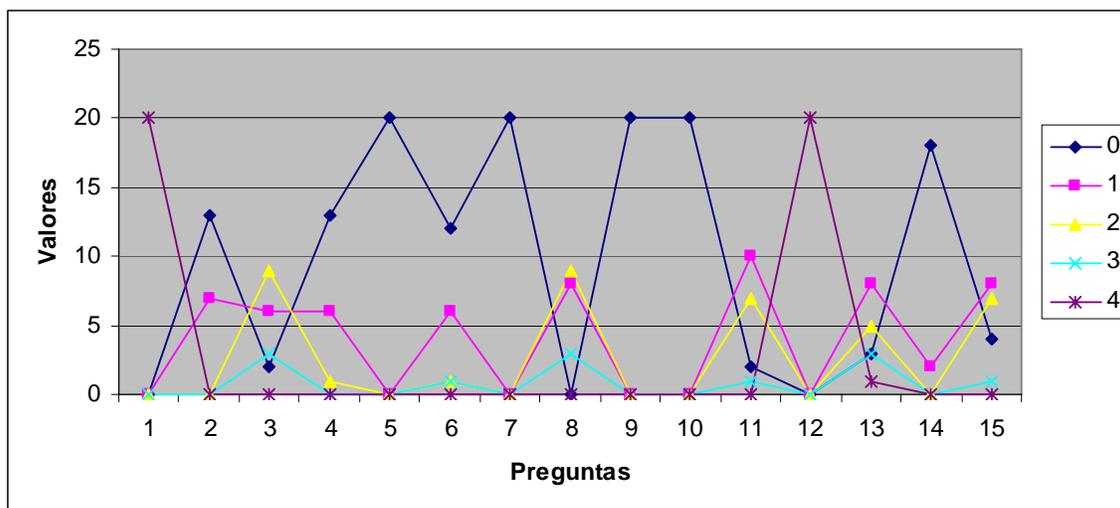
2 = Problema minoritario, de baja prioridad.

3 = Problema grave, de alta prioridad. Es importante solucionarlo.

4 = Catástrofe. Debe ser solucionado inmediatamente.

### 3.3. Resultados

Gracias a los datos recolectados por dicha encuesta aplicada, podemos llegar a realizar un promedio de cada descripción, por el puntaje obtenido por cada uno de las personas encuestadas, y generar el gráfico que se muestra en la figura 50, el cual muestra la tendencia de la correctitud de cada ítem planteada.



**Figura 50. Gráfico resultante de la aplicación de la encuesta.**

Los resultados de la encuesta según los usuarios entrevistados, es que el sistema desarrollado cumple las funciones de extracción, almacenaje y recuperación de la información, dado que los mismos indicaron que utilizarían dicha aplicación en grupos de trabajo para poder compartir información e intercambiar ideas y comentarios con otros miembros del grupo.

Otros usuarios señalaron que dicha aplicación ayudaría a mantener las citas organizadas al momento de realizar proyectos de investigación, como por ejemplo: monografías, informes, entre otros.

Otra característica que fue destacada por los usuarios, es la sugerencia de artículos por parte de la aplicación, dado que es más fácil el proceso de selección

de la información a almacenar, puesto que no deben leer toda la página para encontrar las secciones relevantes.

Ciertos usuarios destacaron que la herramienta Easy Extractor es colaborativa debido a que permite comentar y agregar información que a otros usuarios podría interesar.

En resumen, todos los usuarios quedaron satisfechos con el desempeño de la aplicación, con respecto a la interfaz resaltaron que la misma era usable y sencilla.

La realización de esta encuesta, nos ayudó a determinar que existe una falla en el suministro de ayuda a los usuarios que utilizan el sistema, para facilitar la manipulación y aprendizaje del mismo, ya que no se reconoce con facilidad cuales son las acciones que se pueden realizar con cada opción de la extensión de firefox. Sin embargo, a los usuarios les es fácil recuperar la información almacenada a través de la aplicación Web.

## CONCLUSIONES

Finalizada la realización del Trabajo Especial de Grado, se obtuvo el sistema Easy Extractor, para la extracción, almacenamiento y clasificación de la información en la Web, el cual fue desarrollado basándose en una adaptación de la metodología XP.

Esta herramienta ofrece al internauta una nueva alternativa para la manipulación de la información que se puede encontrar en el Internet.

El uso de la metodología XP en el desarrollo de la aplicación permitió definir claramente los requerimientos y necesidades del usuario, lo cual facilitó el proceso de organización y estructuración de las iteraciones a realizar, que fueron enfocadas al desarrollo de cada módulo del sistema, de esta manera sin la finalización de un módulo no se pudo avanzar al siguiente, garantizándose el correcto funcionamiento y cumplimiento de los requisitos planteados por el usuario en cada módulo.

Además de haber cumplido todos los objetivos planteados, se desarrolló una aplicación que no solo es capaz de extraer, almacenar, organizar y recuperar información sino que también permite que los usuarios puedan relacionarse e intercambiar ideas, ayuda a mantener las citas organizadas, sugiere artículos a los usuarios, permite colaborar con otros usuarios acerca de un tema específico. Adicionalmente compone artículos de unos ya existentes, exporta artículos en formato PDF y nos proporciona una búsqueda sencilla y rápida de la información almacenada.

## RECOMENDACIONES

Difundir este nuevo sistema como herramienta tecnológica para investigadores que se encuentren en el área de computación.

Adiestrar y capacitar a profesionales de otras áreas para que puedan manipular el sistema Easy Extractor y así poder disfrutar de las funcionalidades que el mismo les brinda.

Elaborar un demo que muestre de manera fácil y rápida la manipulación del sistema vía online.

Agregar nuevas funcionalidades al sistema con el fin de satisfacer los intereses de los usuarios.

Incorporar más funcionalidades que permitan la interacción entre los diferentes usuarios.

Ofrecer otros tipos de información que se puedan almacenar, como sonido, audio, video.

## REFERENCIAS

[1] Nafría, Ismael. (2007). *Web 2.0. El usuario, el nuevo rey de Internet*. Editorial Gestión 2000. Grupo Planeta. Madrid / España.

[2] SIMILE project. *Solvent*. <http://simile.mit.edu/wiki/Solvent>.

[3] Google Inc. *Preguntas Frecuentes sobre Google Bloc de Notas*. <http://www.google.com/googlenotebook/faq.html>.

[4] Oeschger, Ian. (2001). *Xul Tools and what they Means*. <http://www.oreillynet.com/pub/a/network/2000/05/19/magazine/xul.html?page=3>.

[5] Mozilla Developer Center. *About Javascript*. [http://developer.mozilla.org/En/About\\_JavaScript](http://developer.mozilla.org/En/About_JavaScript).

[6] Clark, James & Derosé, Steve. (1999). *XML Path Language (XPath) Version 1.0*. <http://www.w3.org/TR/xpath>.

[7] Coulouris G, Dollimore J, Kindberg T. (2001). *Sistemas Distribuidos Conceptos y Diseño*. Tercera Edición. PEARSON EDUCACIÓN, S.A., Madrid, España.

[8] W3C. *Guía Breve de Servicios Web*. <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.

[9] Fielding, Roy. (2000). *Architectural Styles and the Design of the Network-based Software Architectures*. Universidad de California, Irvine. <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.

[10] Chinthaka, Eran (2006). *Web services and Axis2 architecture*. <http://www.ibm.com/developerworks/webservices/library/ws-apacheaxis2/#N10239>.

## LECTURAS RECOMENDADAS

Boswell, David & Colaboradores. *Creating Applications with Mozilla*. Editorial O'Reilly.

Cobo Romani, Cristóbal & Pardo Kuklinski, Hugo (2007) *Planeta Web 2.0. Inteligencia colectiva o medios fast food*. Grup de Recerca d'Interaccions Digitals, Universitat de Vic. Flacso México. Barcelona / México DF.

Costello, Roger L. *Building Web Service the REST Way*.  
<http://www.xfront.com/REST-Web-Services.html>.

*Mozilla: a development platform under the hood of your browser*.  
[http://www.freesoftwaremagazine.com/articles/from\\_java\\_to\\_mozilla?page=0%2C2](http://www.freesoftwaremagazine.com/articles/from_java_to_mozilla?page=0%2C2).

Pérez, Javier. (2008). *Introducción a Javascript*.  
<http://www.librosweb.es/javascript/>

**APÉNDICE: ENCUESTAS REALIZADAS**

**Lucy Fernández**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	0	
3	Los iconos que aparecen se identifican claramente con lo que representan.	0	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	0	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	2	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	1	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	1	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	1	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	2	

**Emmanuel Mestanza**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	0	
3	Los iconos que aparecen se identifican claramente con lo que representan.	2	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	0	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	3	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	2	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	3	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	3	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	1	

**Yesika Fernández**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	0	
3	Los iconos que aparecen se identifican claramente con lo que representan.	0	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	1	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	0	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	1	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	2	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	1	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	0	

**Adrian Rodríguez**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	0	
3	Los iconos que aparecen se identifican claramente con lo que representan.	3	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	2	Religión
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	0	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	3	No se identifica bien que son tags y que se debe colocar en ese campo
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	Usar letra Arial
11	El mensaje de error informa cuales son las acciones correctoras.	1	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	1	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	0	

**Ana Da Silva**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	0	
3	Los iconos que aparecen se identifican claramente con lo que representan.	2	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	1	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	1	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	3	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	1	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	1	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	1	

**Andreina De Sousa**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	1	
3	Los iconos que aparecen se identifican claramente con lo que representan.	2	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	0	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	1	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	2	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	1	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	4	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	1	

**Blanca Pérez**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	0	
3	Los iconos que aparecen se identifican claramente con lo que representan.	2	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	0	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	1	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	1	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	1	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	1	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	1	

**David De Andrade**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	0	Estandarizar el idioma de los menus.
3	Los iconos que aparecen se identifican claramente con lo que representan.	2	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	0	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	1	path
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	3	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	0	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	0	Agregar icono al mouse si es posible
14	La búsqueda de información es fácil y rápida.	1	Repite aciertos
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	1	Fácil luego de usarlo un par de veces. Recomiendo un

			wizard (ayudante para familiarizarse)
--	--	--	---------------------------------------

**Francisco De Freitas**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	0	
3	Los iconos que aparecen se identifican claramente con lo que representan.	2	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	1	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	0	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	1	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	1	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	3	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	1	

**Jesús Rondón**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
<b>1</b>	Cada pantalla empieza con un título que describe su contenido.	4	
<b>2</b>	La terminología del menú es constante en todo el sitio.	1	
<b>3</b>	Los iconos que aparecen se identifican claramente con lo que representan.	3	
<b>4</b>	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	0	
<b>5</b>	Para actividades similares, los diálogos, formularios son similares.	0	
<b>6</b>	Se mantiene una navegación consistente y coherente en todas las pantallas.	0	
<b>7</b>	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
<b>8</b>	Se dan indicaciones para completar campos problemáticos.	2	
<b>9</b>	Los títulos son descriptivos y distintivos.	0	
<b>10</b>	Los tipos y tamaños de letra son legibles y distinguibles.	0	
<b>11</b>	El mensaje de error informa cuales son las acciones correctoras.	1	
<b>12</b>	El acceso a la ayuda está en una zona visible y reconocible.	4	
<b>13</b>	La demarcación de la información de la página Web es intuitiva	2	
<b>14</b>	La búsqueda de información es fácil y rápida.	0	
<b>15</b>	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	2	

**Jonathan Jiménez**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	0	
3	Los iconos que aparecen se identifican claramente con lo que representan.	1	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	1	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	0	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	1	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	0	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	0	
14	La búsqueda de información es fácil y rápida.	1	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	0	

**José Alvarado**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	1	
3	Los iconos que aparecen se identifican claramente con lo que representan.	1	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	1	Religión
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	0	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	2	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	2	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	0	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	1	

**Juan Becerra**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	0	
3	Los iconos que aparecen se identifican claramente con lo que representan.	2	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	0	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	0	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	1	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	2	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	2	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	2	

**María Díaz**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	1	Todos los textos deben estar en un solo idioma
3	Los iconos que aparecen se identifican claramente con lo que representan.	2	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	0	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	1	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	2	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	1	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	1	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	2	

**Néstor Méndez**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	1	
3	Los iconos que aparecen se identifican claramente con lo que representan.	1	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	1	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	0	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	2	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	2	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	1	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	1	

**Pablo Mora**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	1	
3	Los iconos que aparecen se identifican claramente con lo que representan.	3	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	0	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	1	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	2	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	2	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	2	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	3	

**Pedro Gómez**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	1	Mantener un solo idioma
3	Los iconos que aparecen se identifican claramente con lo que representan.	1	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	0	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	0	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	2	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	2	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	3	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	1	

**Rosa Brillante**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	0	
3	Los iconos que aparecen se identifican claramente con lo que representan.	2	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	0	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	0	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	1	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	1	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	2	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	2	

**Sebastian Castro**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	0	
3	Los iconos que aparecen se identifican claramente con lo que representan.	1	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	0	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	0	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	1	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	1	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	1	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	2	

**Tina Burgio**

	Descripción	Cumplimiento (0 sin errores...4 muy grave)	
		Puntaje	Observaciones
1	Cada pantalla empieza con un título que describe su contenido.	4	
2	La terminología del menú es constante en todo el sitio.	0	
3	Los iconos que aparecen se identifican claramente con lo que representan.	1	
4	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.	0	
5	Para actividades similares, los diálogos, formularios son similares.	0	
6	Se mantiene una navegación consistente y coherente en todas las pantallas.	0	
7	La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación.	0	
8	Se dan indicaciones para completar campos problemáticos.	1	
9	Los títulos son descriptivos y distintivos.	0	
10	Los tipos y tamaños de letra son legibles y distinguibles.	0	
11	El mensaje de error informa cuales son las acciones correctoras.	2	
12	El acceso a la ayuda está en una zona visible y reconocible.	4	
13	La demarcación de la información de la página Web es intuitiva	2	
14	La búsqueda de información es fácil y rápida.	0	
15	Es fácil identificar las acciones que se pueden realizar en cada pantalla.	2	