



UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS
VERITATE SOLA NOBIS IMPONETUR VIRILISTOGA. 1948

Facultad Matemática, Física y Computación
Licenciatura en Ciencia de la Computación



TRABAJO DE DIPLOMA

Desarrollo de QuipusNews 2.0

Autor

Yoandy Sánchez Najarro

Tutores

Yuris Denis, Arturo Chang, Manuel Vázquez

Departamento

Programación e Ingeniería del Software.

Julio de 2009

CON SU ENTRAÑABLE TRANSPARENCIA



Dictamen

Hago constar que el presente trabajo fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del Autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del Tutor

Pensamiento

“Son los problemas sin resolver, no los resueltos, los que mantienen activa la mente”.

Erwin Guido Kolbenheyer.

“No hay enigmas, si un problema puede plantearse, es que puede resolverse”

Ludwig Wittgenstein

Dedicatoria

Esta tesis está dedicada especialmente a mi Familia, y a aquellos que tienen la voluntad de aprender algo nuevo cada día.

Agradecimientos.

Al director del periódico Vanguardia.

Y de forma general al colectivo del periódico Vanguardia del cual me sentí parte, en especial a Javier por su ayuda, a Carolina, Marelys y Yusnel. A las correctoras cuyo granito de arena está aquí también.

A Yuri y Manuel, tutores de la tesis, por su disposición y por su tiempo.

A aquellos profesores que influyeron significativamente en mi formación como profesional, de forma especial José Enrique, Blanca Esther, Yosvel Reyes, Daniel Gálvez, Carlos García.

A los colegas que influyeron en mi vocación y me inspiraron en determinado momento, Yarieldis Claro, Yuniesky Carralero, Ernesto Sarduy.

A Freddy por su ayuda.

Y a aquellos amigos que me brindaron su apoyo El Curra, Ernesto, Dayron, Bish, Yeni, Anailys, Noraida, Erilys, las matemáticas Dania y Yuni, Arsenio, Yimmy, El Ninja.

A mi primo Andrés por la ayuda que me ofreció en el momento oportuno y por su constancia.

Resumen

QuipusNews fue el primer sistema que agrupó herramientas para la producción, control y distribución de trabajos periodísticos. La necesidad de la modernización de los sistemas informáticos en la prensa para garantizar un uso óptimo de los recursos con que se cuenta en esta área trajo consigo la idea del desarrollo de una herramienta que sustituyera a la actual versión de QuipusNews y que garantizará mejor accesibilidad, flexibilidad y usabilidad. Se construye así una aplicación web usando las tecnologías más modernas con que se cuenta.

En el trabajo se hace un estudio de las tecnologías, bibliotecas y frameworks que existen para el desarrollo de aplicaciones web similares a entornos de escritorio. Se diseñan nuevas interfaces de usuarios y se implementan los servicios que existen en QuipusNews y que mantienen el flujo de trabajo establecido.

Para el desarrollo de QuipusNews 2.0 se emplea la biblioteca o conjunto de herramientas denominado ExtJS, lo cual hizo posible la construcción de las nuevas interfaces de usuario y las nuevas características añadidas.

Abstract

QuipusNews was the first grouping tools for production, control and distribution of journalistic work. The need to modernize the computer systems in the Press to ensure optimal use of resources available in this area led to the idea of developing a tool to replace the current version of QuipusNews and ensuring better accessibility, flexibility and usability. It builds a web application using the latest technologies that are available.

This project has made a study of technologies, libraries and framework for developing web applications similar to desktop environments. Designed the new user interfaces, implemented the services that exist in QuipusNews and maintained the workflow established.

For the development of QuipusNews 2.0, the ExtJS library or toolkit was used which made possible the construction of new user interfaces and new features added.

Índice

Introducción.....	12
Capítulo 1 Elementos Teóricos Necesarios en el Desarrollo de Quipus News 2.0.....	16
1.1 Sistemas Manejadores de Contenido (CMS).....	16
1.1.1 Características críticas de un CMS.....	16
1.1.2 Características deseables en un CMS.....	18
1.2 Estado del Arte.....	19
1.2.1 Tecnologías usadas con en el lenguaje php.....	20
1.2.1.1 Patrón de Diseño Modelo-Vista-Controlador (MVC).....	20
1.2.1.2 Compresión de Datos usando Gzip y directivas de Apache.....	22
1.2.1.3 Llamado a procesos remotos usando XML (XML-RPC).	24
1.2.1.4 Sistema Gestor de Base Datos.....	25
1.2.1.5 Clase de php PDO.	26
1.2.1.6 Mapeo Objeto Relacional (ORM).....	27
1.2.2 Tecnologías usadas con el lenguaje Javascript.....	28
1.2.2.1 Javascript y XML asincrónico (AJAX).....	28
1.2.2.2 Problemas Básicos con los navegadores.....	30
1.2.3 Frameworks.....	31
1.2.3.1 Frameworks de servidores web.....	32
1.2.3.2 Frameworks de Javascript a nivel del cliente.....	35
1.3 Conclusiones.....	37
Capítulo 2 Metodología de Diseño e Implementación de QuipusNews 2.0.....	39
2.1 Modelación del sistema.....	39
2.2 Diseño del nuevo prototipo.....	39
2.2.1 Especificaciones Funcionales.....	40
2.3 Diagrama y Descripción de los principales casos de uso del Sistema.....	44
2.3.1 Detalles de algunos Casos de usos.....	45
2.4 Descripción del Flujo de Trabajo. Diagramas de Actividades.....	50
2.5 Diseño de las nuevas interfaces de usuario e interacción con el sistema.....	50
2.6 Implementación del Sistema.....	51
2.6.1 Módulos del sistema.....	51

2.6.2	Diagramas de Clases. Clases Controladoras.....	52
2.6.3	Extensión de Clases en Biblioteca ExtJS.	52
2.6.4	Implementación de ORM. Modelo.....	53
2.6.4.1	Creación de Vistas en el gestor de base datos.....	56
2.7	Conclusiones del Capítulo	57
Capítulo 3 Manual de Usuario.....		59
3.1	Nuevas Características en QuipusNews 2.0.....	59
3.2	Requisitos e instalación del Sistema.	59
3.2.1	Requerimientos de hardware.	60
3.2.2	Requerimiento de software.....	60
3.2.3	Pasos para la Instalación.....	60
3.2.3.1	Instalación de la base datos.	60
3.3	Despliegue de QuipusNews.	61
3.4	Comenzando en QuipusNews 2.0.	62
3.4.1	Entrando en el Sistema.	62
3.4.2	Cambiar Contraseña.	64
3.5	Manual del redactor.	65
3.5.1	Buscador.	66
3.5.2	Panel de Edición.	66
3.5.3	Filtro de Materiales.....	68
3.5.4	Mostrar u Ocultar campos en la Rejilla.	68
3.6	Manual del Editor.	69
3.6.1	Panel de Imágenes.	70
3.6.2	Subir Imagen.	70
3.6.3	Insertar Imagen.	71
3.6.4	Edición Maestra.....	72
3.7	Manual del Publicador.	72
3.8	Repositorio de Ficheros.	73
3.9	Manual del Administrador.	73

3.9.1	Sección Usuario.....	74
3.9.2	Sección Roles.	75
3.10	Análisis de los resultados.....	75
3.10.1	Resultados Observables.....	75
	Conclusiones.....	77
	Bibliografía.....	79
	Anexo A Herramientas.	81
	Anexo C Vistazo detallado para futuro trabajo.	86

Introducción

La informatización de la prensa es una tarea de importancia en el ámbito actual de nuestro país. Desde hace unos años se han dado pasos de avances que han resultado en mejoras y calidad de trabajo del periodismo, incidiendo directamente en el nivel de información de la población y su nivel cultural.

QuipusNews es una aplicación web para la gestión de trabajo en medios de prensa digitales. Es un marco de producción, donde el periodista puede además de introducir su trabajo, retroalimentarse con otras informaciones, encontrar imágenes y otros elementos que puedan apoyar el texto (Acosta, Martínez et al. 2002). La versión anterior de QuipusNews fue desarrollada por el Grupo Chasqui de La Universidad Central de Las Villas en el año 2002, integrado por un grupo selecto de estudiantes de la Facultad de Matemática, Física y Computación, que programaron esa aplicación, la cual se ha mantenido estable hasta el momento. Su última versión 1.5ps, realizada en agosto de 2003, actualizó la interfaz de usuario para incorporarle ciertas facilidades de uso, en especial la mayoría de las operaciones se ejecutan usando una técnica similar al moderno AJAX, en lo que se refiere a la forma en la que el usuario interactúa con la aplicación web, de modo que el usuario no pierde el foco de lo que está realizando.

No obstante, este cambio radical en su interfaz provocó una pérdida: QuipusNews 1.5ps sólo funciona correctamente usando Internet Explorer 6.0. En vistas de que hay una política gubernamental de que se vaya migrando, en la medida de lo posible, hacia el software libre, QuipusNews 1.5ps supone un obstáculo en los medios de prensa que lo utilizan al no ser compatible con ningún navegador libre. Además limita la visibilidad por parte de los usuarios que usan otros navegadores que no necesariamente son software libre, por lo cual debe lograrse un código fuente más cercano al estándar en este manejador de contenido.

Por otra parte, no todos los servicios de QuipusNews 1.5ps usan el “**nuevo**” diseño de interfaz y todavía algunos de ellos tienen la interfaz “**vieja**”.

Luego el problema fundamental está dado por el surgimiento de la necesidad de lograr una nueva herramienta que suplante la versión actual de QuipusNews, manteniendo su flujo de trabajo y su objetivo principal que es la automatización del proceso de producción de materiales de prensa en medios digitales.

OBJETIVO GENERAL DEL TRABAJO DE TESIS.

1. Crear una nueva versión de QuipusNews que garantizará mejor accesibilidad, flexibilidad y usabilidad.

OBJETIVOS ESPECÍFICOS.

1. Estudiar las bibliotecas actuales para la creación de aplicaciones Web, en particular, se propone el estudio de ExtJS y Yahoo! User Interface Toolkit.
2. Rediseñar la interfaz de usuario de QuipusNews, en particular se pretende eliminar algunos obstáculos para la usabilidad del sistema.
3. Diseñar un prototipo para la nueva versión de QuipusNews.
4. Decidir cuál biblioteca es más adecuada para implementar el diseño creado.
5. Evaluar el impacto del cambio en la aplicación Web. En particular la forma en que se comunican los eventos entre navegador y servidor.
6. Implementar un prototipo que incluya los servicios:
 - a. Redacción.
 - b. Edición.

Luego de aquí se derivan las siguientes cuestiones que merecen una respuesta:

- ¿Qué aspectos debemos tener en cuenta como criterio de medida para la usabilidad en una aplicación web?
- ¿Las bibliotecas que son estudiadas permitirán la elaboración del sistema con las características tales que cumplimenten los objetivos?
- ¿Beneficiará en gran medida al trabajo en la prensa digital la nueva aplicación desarrollada?

JUSTIFICACIÓN DE LA INVESTIGACIÓN.

Otra de las dificultades que se presentan con QuipusNews 1.5ps es que no hay un equipo que le esté dando soporte (de esta dificultad se derivan, en gran medida, todas las que se han mencionado antes). Luego de agosto de 2003, han emergido varias bibliotecas (o toolkits) para la creación de interfaces de usuario, y que unifican la interacción entre el navegador y el servidor de aplicaciones web. Estas bibliotecas sí cuentan con comunidades de desarrollo que, de ser usadas, pueden ayudar a comprender mejor, partes de QuipusNews 1.5ps que actualmente sólo tres personas dominan.

Además, estas bibliotecas son compatibles con los navegadores más populares y se eliminaría la restricción de usar solamente IE 6.0. Las bibliotecas han sido desarrolladas usando un código lo más estándar posible, lo que permite que un conjunto amplio de navegadores visualicen el contenido y ejecuten las funciones de igual forma. Por otra parte, las bibliotecas están acompañadas de una gran cantidad de *widgets*, tales como ventanas, paneles, celdas y formas. Estas también interactúan con el usuario y el navegador, respondiendo a muchos eventos de teclado y clics de ratón. De aquí que estos *frameworks* de javascript hacen posible construir interfaces de usuarios más apropiadas y de manera más rápida por un desarrollador.

QuipusNews, no obstante, ha probado ser una herramienta eficiente y de gran utilidad en los medios donde ha sido puesto en funcionamiento. Debido a su estabilidad, un cambio radical a otro CMS es una operación que necesitaría de mucha más planificación y cuidado, y no es una operación que se necesite con urgencia.

Al observar a QuipusNews como un CMS, es conveniente hacer una comparación con otros, pues la mayoría de los más populares como Joomla, Drupal, Mambo han sido creados por una comunidad de desarrolladores que además ofrecen soporte a estas aplicaciones y son capaces de mantenerlas con actualizaciones que responden a las demandas y exigencias de los clientes o usuarios.

Otros CMS son desarrollados por encargo de empresas que cuentan con un equipo de profesionales, lo cual les ha permitido obtener productos comercializables.

La versión 2.0 de QuipuNews que actualmente se encuentra iniciando la fase de implantación, comenzó a desarrollarse en Enero del presente año 2009, a pesar de que su idea se concibió con anterioridad. En el presente trabajo el autor realiza el diseño de la nueva versión de dicho software, contando con la asesoría y el apoyo de la dirección del periódico Vanguardia y los tutores de la tesis.

HIPÓTESIS DE INVESTIGACIÓN.

Con QuipusNews 2.0 se espera una herramienta capaz de maximizar el rendimiento y las posibilidades de este Manejador de Contenidos para la gestión del material de noticias en los periódicos online como Vanguardia. Se tendrá entonces una solución innovadora y acorde a las modernas tecnologías con que se cuenta en estos momentos. Se prevé así que cambios radicales en la interfaz de usuario, y referentes a la

comunicación entre cliente y servidor en esta aplicación brinden al equipo de redacción una experiencia positiva en el proceso de producción de la noticia.

ESTRUCTURA DE LA TESIS.

El **Capítulo 1** cubre un análisis de los elementos teóricos necesarios, así como una breve descripción de las tecnologías que se deben usar para la construcción de una aplicación web, específicamente un manejador de contenido, teniendo en cuenta sus ventajas.

En el **Capítulo 2** se expone la metodología para el diseño e implementación del sistema. Se describe el flujo de trabajo que se debe implementar en el gestor de contenido. Se detallan los principales casos de usos, estructuras, patrón de diseño usado, especificaciones funcionales y además, la integración con las bibliotecas empleadas.

El **Capítulo 3** constituye una guía detallada para los usuarios. Consta de un manual de usuarios para los roles de cada uno. Se exponen los requerimientos recomendados de hardware y software necesarios para el funcionamiento de la aplicación.

Además este trabajo contiene 3 anexos.

En el **Anexo A** se lista un compendio de herramientas que incluye entornos de desarrollo y otros softwares útiles en el desarrollo de aplicaciones y que son utilizados para la construcción de esta aplicación.

El **Anexo B** incluye un conjunto de imágenes relacionadas con la aplicación y de interés para desarrolladores y usuarios de esta.

En el **Anexo C** se expone una serie de elementos a tener en cuenta para el futuro trabajo de perfeccionamiento y adaptabilidad de la aplicación.

Capítulo 1 Elementos Teóricos Necesarios en el Desarrollo de Quipus News 2.0.

En este capítulo se trata un análisis de los principales elementos y tendencias tecnológicas existentes para el desarrollo del software que es objeto de nuestro estudio. En particular se aborda el tema de la construcción de aplicaciones web tales como un CMS.

1.1 Sistemas Manejadores de Contenido (CMS).

Existe una gran cantidad de aplicaciones web clasificadas como Sistemas Manejadores de Contenido. Todas están construidas sobre distintas tecnologías y para diferentes arquitecturas y tienen sus particularidades. Por tanto debemos hacer un análisis de las características de estos que nos aporten experiencia para el desarrollo de este tipo de software.

1.1.1 Características críticas de un CMS.

Si definimos un CMS solamente como un sistema para gestión de contenido en la Web, o lo orientamos demasiado a la manipulación de documentos, sería como mirar hacia atrás en lugar de hacia delante. Es evidente que una de las limitaciones de algunos sistemas es que su diseño se orienta demasiado a gestionar documentos (Brampton 2008). Aunque en los sitios web tiene predominaban páginas de texto, ahora son pocas las que se limitan a eso. Aunque el texto es primordial, los sistemas más antiguos buscaron mayor flexibilidad en la cuestión de qué tiene acceso a qué, y quién puede hacer qué.

Si bien el llamado "núcleo" del sistema Mambo puede ser instalado con la funcionalidad útil, una parte esencial de éxito del mambo fue la posibilidad de añadir extensiones. Fuera del núcleo de desarrollo, muchas funciones adicionales se han creado. La existencia de este grupo de capacidades añadidas es vital para muchos usuarios de Mambo. Para muchos requisitos comunes, se dispone de una extensión de la plataforma. Para casos muy raros, o bien el código puede ser personalizado o un nuevo código adicionado en el *framework* de Mambo. Las grandes ventajas son la capacidad de imponer el estilo general y la existencia de regímenes de sitio para la navegación y otros servicios básicos.

El resultado es que los sistemas han superado la etiqueta de los CMS, ya que el mundo de la Web se ha convertido cada vez más interactivo. Sitios como Amazon y eBay han inspirado a muchas otras innovaciones donde la página web es mucho más que un compendio de artículos. Se refleja una tendencia de los CMS a migrar hacia un framework para la creación de nuevas facilidades en la web. La presentación del texto, a menudo con ilustraciones, es una importante capacidad, pero la flexibilidad y la extensibilidad son fundamentales.

Ahora podemos crear una lista de los requisitos esenciales y otra lista de características deseables para un CMS. Los aspectos fundamentales son:

- **Continuidad:** A pesar de las limitaciones de los protocolos básicos de Internet, muchas funciones de un sitio web necesitan mantener la información a través de una serie de interacciones de los usuarios y la información, estos deben ser protegidos del ataque de delincuentes informáticos. El framework debe manejar esto de una manera que hace fácil para las extensiones mantener cualquier dato que necesiten.
- **Gestión de usuarios:** El marco tiene que proporcionar los fundamentos para un sistema de control de usuarios a través de algún tipo de autenticación. Pero este debe ser flexible a fin de que la menor cantidad de código se ha instalado para manejar el requisito, que puede ir desde un solo usuario administrativo manejo de cientos de miles de usuarios distintos y una variedad de sistemas de autenticación.
- **Control de acceso:** Es siempre necesario, aunque sólo sea para que limite la posibilidad de configurar su página web. A menudo se necesita mucho más que los diversos grupos de usuarios se asignan diferentes privilegios. En la actualidad se está ampliamente de acuerdo en que el mejor enfoque es el de acceso basado en funciones de Control (RBAC) sistema. Esto significa que se trata de funciones que se otorgan los permisos, y de acceso se asignan funciones. Es preferible pensar en lugar de usuarios de acceso, ya que los papeles también tienen que tenerse en cuenta otras cosas que sólo los usuarios, tales como los sistemas informáticos.

- **Extensión de gestión:** Un framework es útil si se puede extender fácilmente. No hay una sola instalación de usuario visible que es esencial para cada sitio web, de modo que el marco ideal es despojado de todas las funciones. Cada Capacidad visible por los usuarios puede ser añadida como una extensión. Cuando se consideran los requisitos para la construcción de un sitio web, resulta que hay varios tipos diferentes de extensión. Una buena clasificación es en componentes, módulos, *plugins*, y plantillas.
- **Seguridad y manejo de errores:** Todo el mundo es consciente de la ola de amenazas de *spam* al agrietamiento de los sitios web maliciosos. Para ser eficaces, la seguridad ha de ser incorporada desde el inicio de modo que no sólo el marco de lograr la mejor seguridad posible, también proporciona un medio útil para garantizar la construcción de extensiones. Los errores son importantes tanto como un problema de usabilidad como un problema potencial de falla de seguridad, de manera que un mecanismo de manejo de error estándar es también necesario.

1.1.2 Características deseables en un CMS.

La mayoría de las personas no se conforman con la lista de características críticas. Aunque son los elementos esenciales, es probable que se necesiten más facilidades en la práctica, especialmente si la creación de extensiones se hace fácil (Brampton 2008). La lista de características deseables ciertamente incluye:

- **Manipulación de Código eficiente y mantenible:** El *framework* es probable que se compondrá de un número de código de los archivos separados. Es esencial que se carga solo cuando es necesario. Los mecanismos utilizados tienen que ser capaces de manejar archivos de código adicional añadido como extensiones.
- **Manejo de Base Datos:** Muchas aplicaciones web necesitan acceso a una base datos para funcionar correctamente. El mismo framework necesita de la base datos para ejecutar sus operaciones. Entonces los frameworks deben implementar funciones de más alto nivel para lidiar con la base datos.
- **Caches:** Estos se utilizan en muchos contextos diferentes de procesamiento de Internet. Hasta la fecha, las dos zonas más productivas han sido objeto de caché y XHTML. La Velocidad de operación y la carga de procesamiento ambas se benefician considerablemente de los caches bien implementados. Por lo tanto,

es muy conveniente para un framework de un CMS proporcionar los mecanismos adecuados, ligeros y fáciles de usar.

- **Menús:** Se trata de una característica común de los sitios web, especialmente cuando se toma en el sentido más amplio para incluir cosas tales como barras de navegación y otras formas de presentar lo que son esencialmente listas de enlaces. Es conveniente que el framework de la lógica para la creación y gestión de menús, incluyendo una interfaz estándar para la creación de extensiones para el menú. El framework también debe proporcionar los datos de menú de una forma que facilita la creación de un menú de pantalla.
- **Idiomas:** Hoy en día, como mínimo, el desarrollo de software debe tener en cuenta las exigencias impuestas por la aplicación en diferentes idiomas, entre ellos los que necesitan caracteres multi-byte. Ahora se está ampliamente de acuerdo en que parte de la solución a este requisito es el uso de UTF-8. Un mecanismo para permitir que se traduzca el texto fijo es muy conveniente. El conjunto de cuestiones planteadas por la demanda de apoyo lingüístico son generalmente descrito con los términos de localización e internacionalización. La primera es la construcción de capacidades en un sistema de apoyo a diferentes formas de hacer las cosas, de los cuales el más destacado es la elección de la lengua. La localización está en el despliegue de las especificidades locales en un sistema que se ha internacionalizado. Aparte de la propia lengua, las cuestiones a tener en cuenta incluyen la presentación de fechas, importes monetarios, y los números.

Muchos otros servicios útiles, tales como el manejo del envío de correo electrónico, la asistencia en la creación de XHTML, las solicitudes de aislamiento del sistema de archivos, y así sucesivamente. Pero antes de considerar un enfoque a la aplicación, hay una importante cuestión acerca de cómo un CMS debe administrarse.

1.2 Estado del Arte.

En este epígrafe se realiza un estudio de las tecnologías existente para la construcción de aplicaciones web, en particular los frameworks y bibliotecas.

1.2.1 Tecnologías usadas con en el lenguaje php.

“PHP es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la Web. Dentro de una página Web se puede incrustar código PHP que se ejecutará cada vez que se visite una página. El código PHP es interpretado en el servidor Web y genera código HTML y otro contenido que el visitante verá.”(Thomson 2003).

El lenguaje php se ha generalizado en el desarrollo de aplicaciones web por las características que posee de manejo de hipertexto, además de que es un lenguaje de muy alto nivel que viene con muchas extensiones que brindan facilidades para la implementación de grandes sistemas, manejo de base datos y otros.

1.2.1.1 Patrón de Diseño Modelo-Vista-Controlador (MVC).

MVC es un patrón de diseño orientado a objetos que describe cualquier aplicación con interacción humana y es particularmente adecuado para aplicaciones web .Es un patrón de arquitectura de *software* que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos (Ree 2008). El Modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio. La Vista representa la página HTML; presenta el modelo en un formato adecuado para interactuar.

Usualmente es una interfaz de usuario. El Controlador responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Véase la siguiente figura.

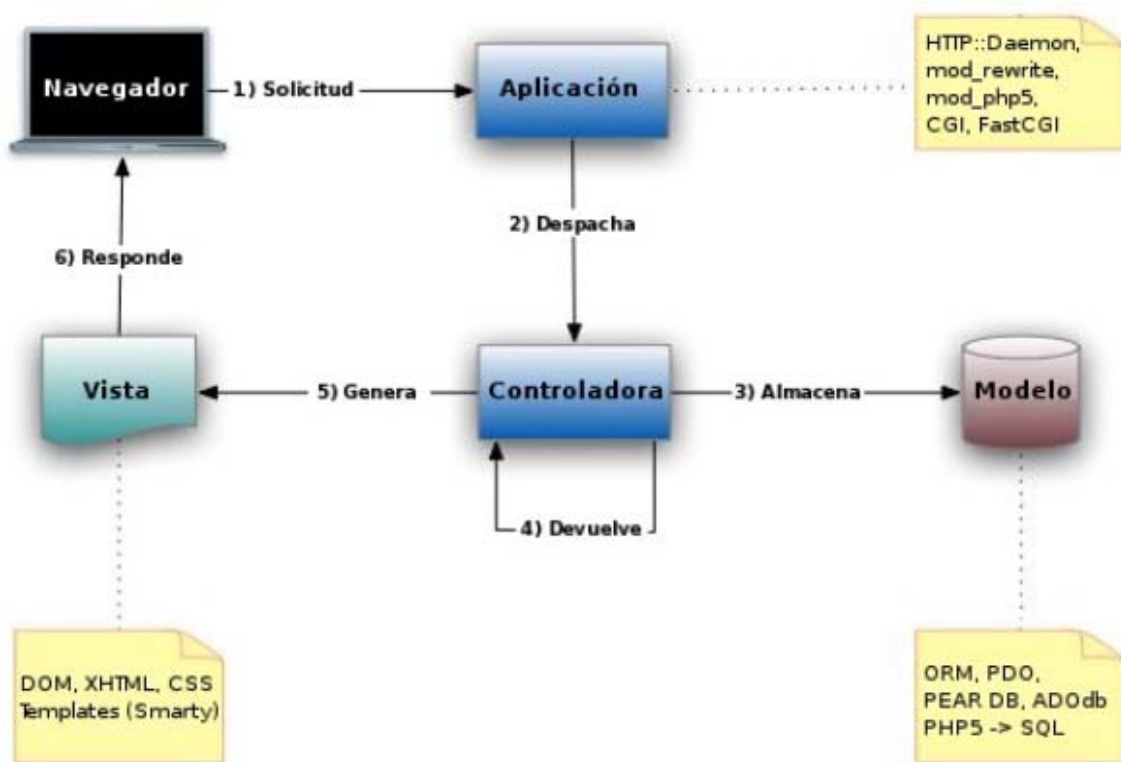


Figura 1. Modelo Vista Controlador.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente(2008):

1. El usuario interactúa con la interfaz de usuario de alguna forma.
2. El controlador recibe la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario. Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

1.2.1.2 Compresión de Datos usando Gzip y directivas de Apache.

La compresión de datos, o sea, la información que fluye entre el navegador del usuario y el servidor es un factor importante a tener en cuenta con respecto a la optimización del ancho de banda empleado para sostener la comunicación. Existen diferentes técnicas que se emplean para reducir el tamaño de los ficheros de código fuente en una aplicación web, podemos mencionar:

1. Minimización de código fuente (javascript).
2. Compresión del texto al ser enviado al navegador por el servidor web.
3. Codificación del texto de los ficheros en ASCII.

En estos aspectos es destacable que en la tercera opción es recomendable la codificación UTF-8 o *unicode* ya que hacen posible eliminar posibles errores que tienen que ver con la interpretación de caracteres de los contenidos en diferentes idiomas.

Por ejemplo: los acentos muchas veces no se visualizan correctamente. Sin embargo se pueden encontrar soluciones existentes hace mucho tiempo como escribir caracteres especiales válidos para HTML o implementar la codificación de los caracteres con javascript en el lado del cliente (en lo adelante en el lado del cliente se refiere a que el código es ejecutado en el navegador del cliente).

Normalmente usamos el método convencional de envío de información descomprimida hacia los navegadores, pero varios de los principales navegadores que existen en la actualidad soportan la funcionalidad de recibir información en formato "gz-encoded", lo que significa que puede recibir datos comprimidos en formato GZ.

A partir de la versión PHP 4.0.5, se incluye la funcionalidad de enviar datos comprimidos hacia los clientes, pero sería una tarea bien difícil adicionar esta funcionalidad a cada uno de las páginas de nuestro sitio. Sin embargo haciendo uso de las funcionalidades ofrecidas por el ".htaccess" podemos adicionar la compresión GZIP a todo nuestro sitio sin necesidad de editar ninguna página.

En el fichero ".htaccess" debemos incluir dos líneas como se describe a continuación:

```
php_value auto_prepend_file /camino/hasta/begin_gzip.php
```

```
php_value auto_append_file /camino/hasta/end_gzip.php
```

Los comandos “auto_prepend_file” y “auto_append_file” permiten incluir scripts en cada ejecución, sin necesidad de utilizar las funciones “include();” o “require();”. Esta funcionalidad hace mucho más fácil incluir nuevas funcionalidades a cualquier sitio sin siquiera editar cualquier script del sitio.

En este ejemplo se usan dos scripts. La primera línea es equivalente a incluir “include("begin_gzip.php");” al comienzo de cada script. La segunda línea es equivalente a incluir “include("end_gzip.php");” al final de la ejecución de cada script.

En estos dos scripts, lo que se debe activar es la funcionalidad en php para controlar la salida del buffer, especificándole al PHP que comprima los datos si detecta que el navegador que esta haciendo el pedido soporta datos comprimidos. Cada agente (ej. I.E. o FF) envía un “header” al servidor indicando si puede o no recibir datos en formato “gz-encoded”. Si el agente no soporta esta funcionalidad (como es el caso de los principales buscadores y spiders), PHP simplemente le envía los datos sin comprimir, así que no se corre el riesgo de que alguien no pueda ver el sitio por no soportar esta funcionalidad.

El primero de los scripts “begin_gzip.php” debe contener la siguientes línea, indicándole al *parser* de PHP que active el buffer de salida y que como resultado de ese buffer los datos deben ir codificados en formato “.gz”.

```
<?php ob_start("ob_gzhandler"); ?>
```

El segundo script “end_gzip.php”, debe contener otra línea como se describe a continuación. Esta línea le dice al PHP que libere el buffer de salida.

```
<?php ob_flush(); ?>
```

Como conclusión podemos decir que incluir esta funcionalidad a un sitio montado sobre PHP no tiene riesgo alguno de que los usuarios no puedan ver el sitio, y sin embargo tiene 3 ventajas principales:

1. Muchos de los usuarios verán el sitio un poco más rápido, más aun si usas grandes volúmenes de datos y página muy extensas.
2. Disminuirá en gran medida el tráfico generado continuamente para tu servidor de hospedaje.
3. Si usas algún framework en javascript como Dojotoolkits, YUI, Prototype, etc., que descargan cientos de Kb hacia el navegador, podrás optimizar la descarga en

un 60% aproximadamente (300 kbytes PlainText sería equivalente a 120 kbyte GZ).

Se debe tener en cuenta que si ya se hace uno de las funcionalidades de buffering de PHP en algunos de los *scripts*, quizás los resultados no sean los esperados, y además se debe configurar correctamente el camino hasta los scripts dentro del fichero “.htaccess”.

1.2.1.3 Llamado a procesos remotos usando XML (XML-RPC).

XML-RPC es una especificación funcional de un conjunto de implementaciones que le permiten a programas que se ejecutan en diferentes sistemas operativos y ambientes realizar llamadas a procedimientos a través de Internet (XML-RPC 2008).

Esta se define como un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes. Está diseñado para ser lo más simple posible, al tiempo que posibilita que complejas estructuras de datos sean transmitidas, procesadas y se devuelva una respuesta.

Los parámetros de los procedimientos pueden ser escalares, numéricos, cadenas de caracteres, fechas, etc.; y pueden ser además registros y listas, así como complejas estructuras de datos.

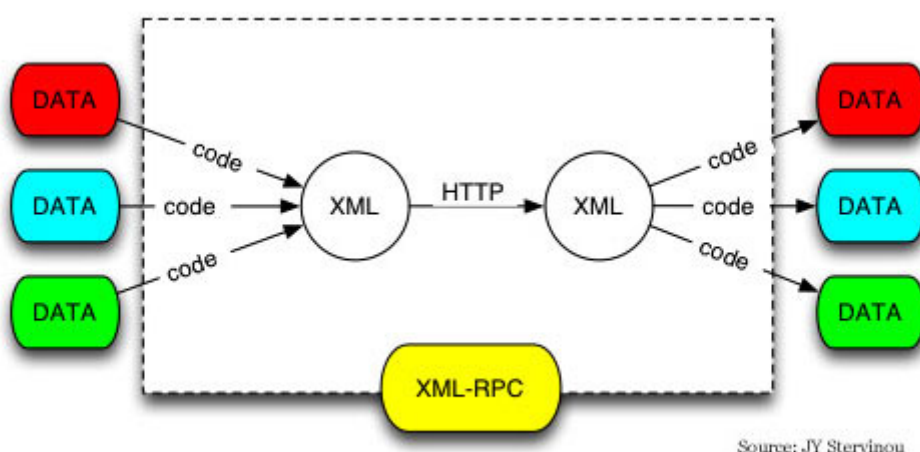


Figura 2. XML-RPC

Veámos ejemplo de invocación en XML-RPC :


```

1  <?xml version="1.0"?>
2  <methodCall>
3    <methodName>cu.vanguardia.setinfo</methodName>
4    <params>
5      <param>
6        <value><i4>360</i4></value>
7      </param>
8      <param>
9        <value><i4>221</i4></value>
10     </param>
11   </params>
12 </methodCall>
13

```

Y como respuesta a esta invocación se obtiene.

```

1  <?xml version="1.0"?>
2  <methodResponse>
3    <params>
4      <param>
5        <value><string>Se ha actualizado la informacion obj_id: 51</string></value>
6      </param>
7    </params>
8  </methodResponse>
9

```

Existen implementaciones para varios sistemas operativos, lenguajes de programación, licencias comerciales y de software libre: C/C++, Delphi, Frontier, Guile, Java, Lisp, Microsoft .NET, Perl, PHP, Python, Real Basic, Rebol, Tcl, WebObjects y Zope (XML-RPC 2008).

Algunas implementaciones conocidas son:

- Apache XML-RPC, una implementación en Java.
- XMLRPC-EPI, una implementación php.
- XML-RPC-C, una implementación para C y C++.

Existen además muchas bibliotecas en PHP que brindan funcionalidades para el trabajo con XML-RPC, lo cual facilita el trabajo con este protocolo en gran medida. Ejemplo de ello son *Zend Framework*, *CodeIgniter*, entre otras, que serán estudiadas en el apartado 1.3.4 “Frameworks”.

1.2.1.4 Sistema Gestor de Base Datos.

Existen un gran número de Gestores de Base Datos que se usan para el almacenamiento y tratamiento de la información por las aplicaciones web. Se hallan razones concluyentes además por que sitúan a Mysql como inmejorable tratándose de aplicaciones web en criterios que se tienen en cuenta como velocidad en las transacciones, capacidad para múltiples conexiones y caché de datos. Mysql además es

Open Source. Estas razones nos conllevan a escoger a Mysql como gestor de base datos para el desarrollo de la aplicación, independientemente de que las versiones anteriores de QuipusNews han sido desarrolladas usando este gestor de base datos, por lo que un cambio de decisión implicaría un trabajo extra de movimiento hacía otro gestor que por consiguiente podría provocar pérdidas de información valiosa que ya se tiene almacenada.

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Este se desarrolla como software libre en un esquema de licenciamiento dual. Este gestor destaca por poseer varios motores de almacenamiento, siendo los más usados MyIsam y InnoDB (Schumacher and Lentz 2008). Este último ofrece muchas características que le dan solidez a la base de datos y permiten la integridad referencial de la misma. También implementa la agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

1.2.1.5 Clase de php PDO.

PHP Data Objects (o PDO) es una extensión que provee una capa de abstracción de acceso a datos para PHP 5, con lo cual se consigue hacer uso de las mismas funciones para hacer consultas y obtener datos de distintos manejadores de bases de datos.

La necesidad de una librería como PDO nace de la gran cantidad de bases de datos soportadas por PHP (Group 2006). Cada una de estas bases de datos necesita de su propia API para hacer tareas generalmente comunes al resto. Esto provoca que nuestro código no sea fácil portar de una base de datos a otra.

Para solucionar este problema en PHP existen múltiples librerías ofrecidas por terceros. ADODB, PEAR DB y MDB2 son algunas de las más populares. Aunque la mayoría de estas librerías son mucho más maduras que PDO y en algunos aspectos ofrecen muchas más funcionalidades, PDO tiene ciertas ventajas que no podemos obviar.

Para empezar: la velocidad. PDO está compilado en C/C++, por lo que ofrece mayor rapidez de ejecución que sus competidoras (de código interpretado). Por otra parte: viene incorporado en PHP5 (es la solución “oficial” y en continua evolución) y desde PHP 5.1 viene habilitado por defecto. De esta manera no tienes que importar ni cargar código de terceros cuando quieres usarlo.

No hay que olvidar que PDO está completamente orientado a objetos, lo que facilita su uso frente a decenas de diferentes funciones de las API de cada sistema de bases de datos.

PDO actualmente soporta: Microsoft SQL Server, Firebird, IBM Informix, MySQL, Oracle, ODBC, PostgreSQL, SQLite.

1.2.1.6 Mapeo Objeto Relacional (ORM).

El mapeo objeto-relacional (más conocido por su nombre en inglés, *Object-Relational mapping*) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional. En la práctica esto crea una base de datos orientada a objetos virtual, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).

Sin herramientas, un desarrollador tendría que definir las tablas en una base de datos y los objetos correspondientes en el ambiente de desarrollo según el modelo de datos diseñado. Luego debe ser definida una interfaz para conectar los dos elementos antes expuestos, convirtiendo la comunicación durante las consultas y los resultados. Para mejorar esta situación se encuentran herramientas de desarrollo que integran el sistema gestor de base de datos en el ambiente de desarrollo, haciendo posible reducir la carga de trabajo en el proceso de desarrollo.

Hay paquetes comerciales y de uso libre disponibles que desarrollan el mapeo relacional de objetos, aunque algunos programadores prefieren crear sus propias herramientas ORM.

En la programación orientada a objetos, las tareas de manejo de datos son implementadas generalmente por la manipulación de objetos, los cuales son casi siempre valores no escalares. Para ilustrarlo, considere el ejemplo de una entrada en una libreta de direcciones, que representa a una sola persona con cero o más números telefónicos y cero o más direcciones. En una implementación orientada a objetos, esto puede ser modelado por un "objeto persona" con "campos" que almacenan los datos de dicha entrada: el nombre de la persona, una lista de números telefónicos y una lista de direcciones. La lista de números telefónicos estaría compuesta por "objetos de números telefónicos" y así sucesivamente. La entrada de la libreta de direcciones es tratada como un valor único por el lenguaje de programación (puede ser referenciada por una sola

variable, una instancia). Varios métodos pueden asociarse con el objeto, como uno que devuelva el número telefónico preferido, la dirección de su casa, etc.

Sin embargo, muchos productos populares de base de datos, como los productos SQL DBMS, solamente puede almacenar y manipular valores escalares como enteros y cadenas, organizados en tablas.

El programador debe convertir los valores de los objetos en grupos de valores simples para almacenarlos en la base de datos (y volverlos a convertir luego de recuperarlos de la base de datos), o solo usar valores escalares simples en el programa. El mapeo relacional de objetos es utilizado para implementar esta primera aproximación.

El vía crucis del problema reside en traducir estos objetos a formas en las cuales puedan ser almacenadas en la base de datos, y los cuáles pueden ser recuperados más tarde fácilmente, mientras se preserven las propiedades de los objetos y sus relaciones; estos objetos se dice entonces que son persistentes.

ORM es usado para el acceso transparente a datos almacenados en la base datos mediante objetos del ambiente de desarrollo. Esto reduce el trabajo, ya que las acciones de creación, lectura, actualización y eliminación son definidas automáticamente. Por lo que se Reducen errores comunes y riesgos de seguridad (Fowler 2008).

1.2.2 Tecnologías usadas con el lenguaje Javascript.

Las tecnologías que podemos encontrar que involucran el lenguaje Javascript tenemos: AJAX y DHTML. Ambas ofrecen un conjunto de técnicas que permiten crear sitios web interactivos y dinámicos (Flannagan 1997).

1.2.2.1 Javascript y XML asincrónico (AJAX).

Por AJAX se entiende *Asynchronous Javascript and XML*, lo que en español es Javascript asincrónico más XML. Este ofrece una técnica en el lenguaje javascript del lado del cliente para realizar llamados al servidor por debajo de la escena, o sea, de forma transparente al usuario. De esta forma se obtiene información adicional para actualizar una porción de la página web sin causar la recarga completa de la página (Darie, Brinzarea et al. 2006).

AJAX es un fenómeno que significa diferente cosas para diferentes tipos de personas, ya que se ofrece al usuario una experiencia muy mejorada, beneficiando tanto a los

diseñadores de sitios web, reduciendo requerimientos de servidores y ancho de banda, como a los usuarios de la aplicación web ya que se incrementa la usabilidad.

Definimos entonces AJAX como un conjunto de tecnologías que permiten el intercambio de información entre el navegador del usuario y el servidor sin recargar completamente la página. Esto no solo hace que las aplicaciones web respondan más rápido, además éstas se hacen más interactivas ya que pequeñas partes de una página pueden actualizarse. Eliminar la recarga de página es solo un paso más para habilitar características más complejas como validación de datos en tiempo real, arrastrar y tirar componentes, y otras tareas asociadas tradicionalmente a las aplicaciones web.

Esto garantiza una mejora en la usabilidad de la aplicación, que es una de las cualidades que buscamos para el desarrollo de QuipusNews. Por lo cual la búsqueda nos lleva a un framework (véase el epígrafe 1.2.3 Frameworks) o plataforma a nivel de cliente que brinde soporte de AJAX.

Desde el punto de vista técnico los ingredientes esenciales de AJAX son los siguientes:

- Uso del *Document Object Model* (DOM) (referenciar) en funciones javascript para manipular partes del HTML de la página.
- El objeto **XMLHttpRequest** de javascript permite acceder al servidor asincrónicamente, de forma que usuario continúa trabajando mientras las funcionalidades son ejecutadas en background.
- Se requiere una tecnología de servidor para manejar y dar respuesta a los pedidos provenientes del cliente, por ejemplo PHP.(definir en anexos).

Podemos resumir que AJAX representa la combinación de técnicas ya existentes.

- Este combina Javascript y XML para lograr sitios web más dinámicos.
- Permite interfaces de usuarios más interactivas.
- Aumenta la velocidad en la interacción cliente-servidor.
- Existen gran cantidad de frameworks que proveen soluciones sólidas incluso para incompatibilidades entre distintos navegadores.

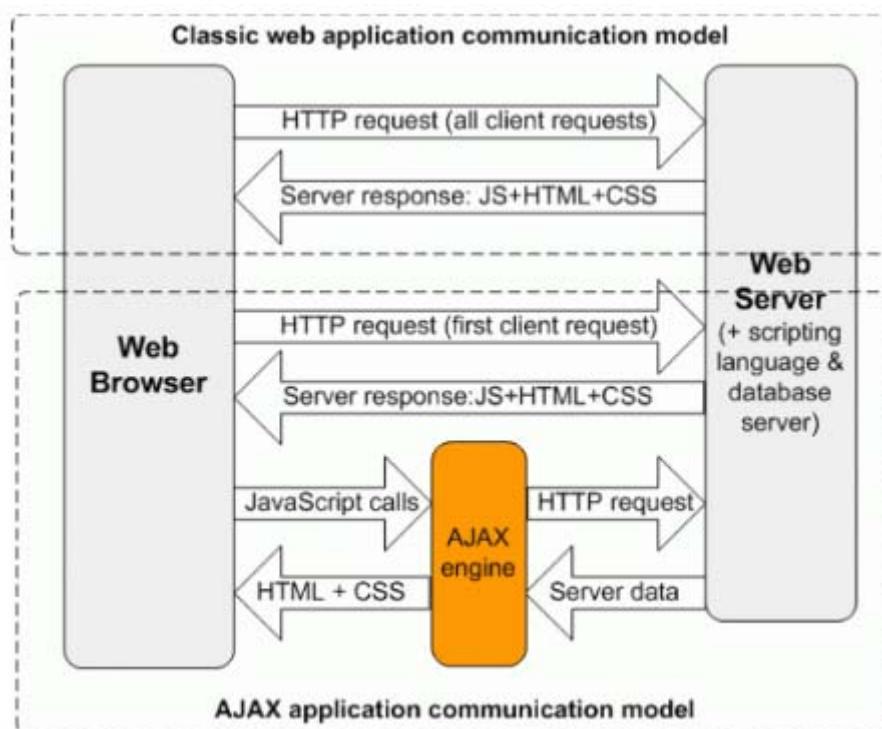


Figura 3. tecnología AJAX

Como se observa en la imagen el centro de la comunicación está en el motor de Ajax que en muchos casos no es más que un script con la responsabilidad de instanciar el objeto XMLHttpRequest y usar sus métodos.

AJAX requiere un nivel significativo de conocimiento, ya que proviene de javascript. Desafortunadamente el javascript es uno de los lenguajes de programación más incomprensidos que existen. Sin embargo Javascript es realmente un poderoso lenguaje de script (interpretado) orientado a objetos. Mientras más desarrolladores se familiaricen con el verdadero poder que contiene javascript podrán tomar mayor ventaja de todo lo que AJAX puede ofrecer para Internet.

1.2.2.2 Problemas Básicos con los navegadores.

Para construir cualquier aplicación Web, tenemos que hacer algunas suposiciones acerca de lo que sucederá en el navegador. Esto se complica por la existencia de diferentes navegadores, cada uno con sus propias peculiaridades. La mayoría de estas se refieren a los detalles del uso de XHTML y CSS, pero hay algunas cuestiones de amplio uso que se puede revisar ahora.

Uno de ellos es adoptar una política sobre el uso de Javascript. Sin duda, es posible mejorar la capacidad de respuesta de aplicaciones Web mediante el uso de un navegador

basado en un lenguaje de script. El código se ejecuta en el propio ordenador del visitante, en lugar de tener que volver al servidor para ejecutar código. Para algunas aplicaciones, como los editores WYSIWYG, no es práctico de usar otra cosa que los mecanismos que existen en el navegador. Aunque hay varias opciones para el navegador de secuencias de comandos, la más utilizada es probablemente JavaScript.

Hay problemas con la estandarización del JavaScript, pero sobre todo hay un problema de accesibilidad. No todos usan un navegador que se soporte JavaScript, y, en particular, los “lectores de pantalla” utilizados por personas que no pueden leer la información de una pantalla.

Otra consideración general es el uso de *cookies*. A pesar de los temores después de su introducción, el uso adecuado de las *cookies* se considera perfectamente normal. Un inconveniente es que los robots de rastreo rechazan las *cookies*.

De lo contrario, ya que estamos interesados en la construcción de un avanzado gestor de contenidos, y características críticas tales como la capacidad para permitir a los usuarios acceder a los compradores o construir un carro de la compra no se pueden proporcionar con seguridad sin el uso de cookies, asumimos que las cookies serán aceptadas. Esto no quiere decir que los visitantes que rechazan las "cookies" son bloqueados, sino que los servicios que reciben serán restringidos.

1.2.3 Frameworks.

En general, con el término framework, nos estamos refiriendo a una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Un framework constituye entonces una plataforma de desarrollo.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones, como el patrón de diseño Modelo-Vista-Controlador abordado en el apartado 1.3.2.1.

La mayoría de los frameworks Web se encargan de ofrecer una capa de controladores de acuerdo con el patrón MVC, ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación.

La construcción de aplicaciones web como los gestores de contenido exige la necesidad de la selección de un frameworks o un conjunto de bibliotecas o componentes que ayuden y agilicen el proceso de desarrollo.

1.2.3.1 Frameworks de servidores web.

La continuación de la búsqueda de un ambiente de desarrollo apropiado, que ofrezca los instrumentos para mejorar el desarrollo de aplicaciones, lleva al mundo del desarrollo web, que está lleno de aplicaciones hechas a la medida y sitios web, muchos de los cuales son conectados a sistemas de base de datos. Además, muchas nuevas tecnologías están siendo usadas en este mundo. Este contexto es perfecto para estudiar los instrumentos de desarrollo que podrían dar con una integración mejorada con un Sistema Gestor de Base Datos y proveer de herramientas flexibles para generar elementos de interfaces de usuario sobre un modelo de datos. Una ventaja adicional de desarrollar una aplicación web es que el resultado es mucho más flexible: esto hace que no se requiera de procedimientos de instalación difíciles y a su vez, pone el sistema a disposición en donde se quiera, mientras haya una conexión a Internet disponible. De esta forma también se reduce la realización y los problemas de mantenimiento y gastos. Con las capacidades del JavaScript Asíncronico y XML (mejor conocido como AJAX) es posible crear aplicaciones similares a las de escritorio. Nuevos navegadores como Mozilla próximo Firefox 3.5 y el Chrome de Google ayudan a estas mejorando la velocidad de tiempo de ejecución del código de JavaScript compilándolo.

La mayoría de los frameworks para aplicaciones web presentan una serie de características que se pueden ilustrar en la siguiente tabla:

Absorción de URL y sesiones	No es necesario manipular directamente las URLs ni las sesiones. sesiones, el framework ya se encarga de hacerlo.
Acceso a datos	Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en BBDD, XML, etc.

Controladores	La mayoría de frameworks implementa una serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables a las necesidades de un proyecto concreto.
Autenticación y control	Incluyen mecanismos para la identificación de usuarios mediante <i>login</i> y contraseña y permiten restringir el acceso a determinadas páginas a determinados usuarios.
Internacionalización.	Herramientas que permiten la creación de diccionarios para el trabajo con varios
Separación entre diseño y contenido.	Consiste en el uso de motores de plantillas y manejo de vistas para lograr una separación entre los archivos el código fuente (PHP) y el código de navegador (XHTML) en el caso de los sistemas web.

Varios frameworks que contienen las técnicas antes descritas han sido listados.

Zend Framework.

Zend es la compañía que desarrolla PHP. Este Framework posee muchas características innovadoras. Este proporciona el soporte y el código profesional apoyado por una compañía corporativa. Esto es para aquellos que quieren construir aplicaciones para empresas grandes. Ellos tendrán programadores profesionales y también serán beneficiados de los componentes que Zend proporciona (Zend 2009).

CakePHP.

Ha mejorado significativamente en cuanto a sus características, aunque ha bajado en cuanto al rendimiento. Este tiene una comunidad muy activa y la diferencia de este framework de otros es que, realmente es usado por muchas compañías y sitios Web. Un sistema manejador de contenido muy popular como Mambo ha decidido usar CakePHP en su próxima versión. Además tiene una base de código enorme de componentes y *plugins*.

Symfony.

Es uno de los frameworks que brinda más características y herramientas para desarrollo, sin embargo aún así es el más lento, lo que se debe la carga de bibliotecas que recaen en un costo de recurso a nivel de servidor. Presenta una capa de base datos separada, o sea, en lo que se refiere al implementación de un ORM. Es usable para proyectos de gran envergadura que requieran un framework que brinde todas las características que ofrece Symfony (Fabien Potencier 2008).

CodeIgniter.

Es el más rápido de los frameworks en estudio. Tiene una comunidad de desarrolladores enorme (Myer 2008).

Ilustremos una comparación para medir el rendimientos en 3 de estos frameworks. Las pruebas han sido realizadas usando un medio de cómputo con un procesador AMD Sempron a 1.8 Ghz, 512 MB de RAM, y sistema operativo Linux. Se usó el software httpperf. Además, se creó la misma vista en para cada framework haciendo uso también de accesos a datos. (Ekerete 2008)

Tabla de Rendimiento

	Corrida 1	Corrida 2	Corrida 3	Corrida 4	Promedio
CakePHP	8.0	8.0	8.0	7.2	7.8
Codeigniter	98.2	98.1	98.3	98.3	98.2
Zend Framework	33.2	33.3	33.5	33.6	33.4

En conclusión a está comparación tenemos que Codeigniter es dos veces más rápido que el framework de Zend y mucho más lento es el CakePHP.

Vemos ahora el comportamiento en cuanto a sus características.

Tabla de Características (Pallett 2006).

	MVC	ORM	Plantillas	Ajax	M.Aut	Modulos
Zend	X					X
CakePHP	X	X		X	X	
Symfony	X	X		X	X	
CodeIgniter	X		X			X

Argumentar que además fue posible integrar en CodeIgniter una implementación de un ORM de terceros, y que un soporte para Ajax no es necesario, ya que al lado del cliente se usa una biblioteca como ExtJS, que ofrece todas las facilidades para Ajax, por lo cual no es necesaria la generación de código a nivel de servidor, sobre todo, porque todas las interfaces de usuario son construida con la biblioteca ExtJS. Además, el módulo de Identificación fue heredado o continuado de la antigua versión de QuipsNews, por lo cual pierde importancia la necesidad de uso de un módulo nativo para este propósito.

1.2.3.2 Frameworks de Javascript a nivel del cliente.

Para resolver el problema de que las aplicaciones web no sean lo suficientemente interactivas, es usado javascript en estas para el navegador en el lado del cliente, que contiene un entendimiento de la lógica de negocio. Un gran problema existente con javascript y las aplicaciones web en general es que al ejecutarse en distintos navegadores hay una tendencia a interpretar el javascript y los estilos con ciertas diferencias. Aunque en la actualidad con la actualización de los navegadores estos tratan de interpretar el estándar más estrictamente, todavía persisten pequeños problemas de compatibilidad entre los *softwares* de navegación.

Sin embargo, hay soluciones disponibles para afrontar estos problemas e inclusive lograr la compatibilidad con navegadores viejos o de versiones anteriores. Para ello se requiere un gran conocimiento de javascript, los estilos y la forma en los diferentes navegadores lo interpretan, para así manejar excepciones para distintas interpretaciones. Afortunadamente, en la actualidad existe un gran número de bibliotecas de javascript capaces de manejar la ejecución del código y los estilos de tal forma que se aprecie igual en todos los navegadores populares. Estas bibliotecas contienen combinaciones de muchas soluciones comunes.

Todo aquel que ha usado Gmail, Microsoft Outlook Web Access, Google Maps está indirectamente familiarizado con las potencialidades que hay tras la tecnología AJAX. which gives the responsiveness associated with desktop applications to website applications. Por AJAX se entiende Asynchronous JavaScript + XML , lo que español es Javascript asincrónico más XML. Este permite una comunicación por debajo del escenario en las aplicaciones web. De esta forma, es posible actualizar solo una pequeña porción de la página que se requiera para responder a una actividad del usuario. Así AJAX es capaz de ofrecer al usuario una experiencia muy mejorada; beneficiando tanto a los diseñadores de sitios web, reduciendo requerimientos de servidores y ancho de banda, como a los usuarios de la aplicación web ya que se incrementa la usabilidad.

Yahoo User Interface o YUI es una biblioteca o también se puede definir como un conjunto de utilidades y controles, escritos en javascript como lenguaje de programación. Yahoo User Interface puede ser usada para construir aplicaciones web que interactúan intensamente a través del uso de la escritura en el DOM (Document Object Model). AJAX, DHTML y otras herramientas muy efectivas. En esta biblioteca se pueden encontrar además recursos de estilos (CSS) en su núcleo. Los componentes de la biblioteca YUI son todos a código abierto, lo cual significa que son libres de usar para cualquier propósito.

DOJO es un conjunto de herramientas para DHMTL de código abierto, escritas en javascript. DOJO está construido sobre otras muchas bases de código de contribuidores que incluyen a BurstLib, F(m), nWidgets, por lo cual también se puede considerar como un conjunto de herramientas o biblioteca unificada. El propósito principal de DOJO es resolver varios de los problemas muy conocidos ya que existen con respecto a DHTML. Lo cual ha impedido el desarrollo masivo de aplicaciones web dinámicas.

DOJO hace que sea sencillo [Dojo makes it easy to build dynamic capabilities directly into web pages or other JavaScript supported environments.]. Los componentes provistos por DOJO pueden ser utilizados para crear sitios web que son más interactivos funcionales y amigables para el usuario. A través del uso de DOJO los desarrolladores pueden construir interfaces de usuarios más rápida y fácilmente que antes, simplificando las escrituras complejas de código mediante el uso de las API de bajo nivel de Dojo y sus capas de compatibilidad.

ExtJS es un framework basado en javascript a nivel del cliente, o sea, es funcional ejecutándose en el navegador del usuario. Es usado para la construcción e aplicaciones web. Ext se originó como un conjunto de utilidades extendidas para YUI. Estas extensiones fueron bien recibidas y rápidamente organizadas en su propia biblioteca independiente. Originalmente fue distribuida como YUI-Ext. Jack Slocum liberó la versión .33 de YUI-Ext, la cual se convirtió en la versión final del código bajo ese nombre. Ya que al final del año la biblioteca había ganado suficiente popularidad su nombre se quedó en Ext simplemente. En el 2007 una compañía fue creada para lidiar con la creciente popularidad de Ext, que actualmente presenta una licencia dual.

Extjs fue desarrollado por un equipo unificado de programadores con una meta singular: proveer un núcleo y biblioteca de interacción de interfaz de usuario estable (Blades, Ramsay et al. 2008). Por esto, el código fuente usado para diferente funcionalidades y widgets es más coherente que en muchas otras bibliotecas de Javascript. ExtJs resalta por lograr aplicaciones web más intuitivas y fáciles de usar. Todo lo cual tiene un significado en el objetivo de QuipusNews 2.0 así como en la hipótesis de investigación de este trabajo.

1.3 Conclusiones.

En este capítulo se abordado la metodología necesaria en la construcción de un manejador de contenido dentro de la que clasifica la herramienta QuipusNews. Además, se ha realizado un estudio de los elementos teóricos necesarios, así como un análisis de las tecnologías existente para el desarrollo de una aplicación web con las características propias de un software de gestión de documentos. El estudio de estas tecnologías nos ha permitido seleccionar las herramientas, ambiente de desarrollo y bibliotecas que se emplearán para el desarrollo e implementación de un sistema capaz de procesar el material de prensa hasta publicarse en uno o varios sitios web online,

siguiendo un flujo de trabajo que garantiza un proceso organizado como es el caso del periódico Vanguardia.

Capítulo 2 Metodología de Diseño e Implementación de QuipusNews 2.0.

2.1 Modelación del sistema.

El modelo del sistema incluye la definición de los casos de uso del sistema, la descripción de los procesos fundamentales que implementa el módulo y el diseño de las interfaces de usuario.

2.2 Diseño del nuevo prototipo.

Para el desarrollo de un nuevo prototipo de interfaz de usuario para la aplicación QuipusNew 2.0 se usará la biblioteca ExtJS. Alternativamente su pudieran usar otras bibliotecas como JQuery, YUI (*Yahoo User Interface*), Prototype y Dojo, algunas de las cuales han sido estudiadas en el apartado 1.3.4.2. Estas al igual que ExtJS brindan facilidades para la construcción de una interfaz de usuario, así como para las funcionalidades de comunicación entre el navegador del usuario, y el servidor, ya que además contienen una gran cantidad componentes. Sin embargo es posible usar un adaptador para usar como base las clases definidas en las otras bibliotecas anteriormente mencionadas en ExtJS, ya que ésta presenta patrones de diseño en su construcción que lo posibilitan. Así podemos usar por ejemplo la biblioteca de YUI “por debajo del telón” funcionando en conjunto con nuestra biblioteca, de manera que todos los componentes funcionaran de forma idéntica. ExtJS además contiene un adaptador interno para él, esto lo hace independiente de otra biblioteca, siendo su uso en conjunción con cualquier otra, solo una opción que podemos emplear.

En el momento en que se realiza esta tesis, estando aún en fase de desarrollo es liberada la versión 3.0 de ExtJS, que trae algunas características nuevas y mejoras, de forma general, por lo que es necesario plantearse el estudio de la nueva versión con vistas a considerar la factibilidad de la migración a esta versión cuyo núcleo a variado.

ExtJS 3.0 presenta las siguientes características:

- Nuevo núcleo base de la biblioteca más liviano con optimización en su código.
- API para diagramas haciendo uso de la tecnología Flash.
- Soporte para conexión remota con streaming de datos.

- Componente ListView.
- Mejoras y refinamiento en los componentes de barraas de herramientas, menú y botones.
- Actualizaciones de estilos que brindan más facilidades para la personalización de temas.
- Actualización de eventos y registro de modelos.
- refinamientos de las funciones Ajax.
- Implementaciones para accesibilidad.
- Integración de flujo de datos cliente servidor.

Sin embargo algunos de los componentes que ya habían sido implementados con la versión 2.0 de la biblioteca ExtJS no se visualizaron correctamente con la versión 3.0. Debe tenerse presente además que la versión 3.0 se encontraba en su fase Beta. Finalmente la decisión adoptada fue construir el módulo de administración solamente con la versión de 3.0 de ExtJS. Debido a que en este momento aún no se había comenzado la implementación de este módulo. Y con el objetivo de usar las nuevas características de esta versión en dicho módulo.

2.2.1 Especificaciones Funcionales.

Una especificación funcional describe cómo funcionará un producto completamente desde la perspectiva del usuario. No le importa cómo se implemente la cosa. Habla de funciones. Especifica pantallas, menús, diálogos, etcétera (Spolsky 2000).

Estas generalmente contienen:

- Una nota de aviso.
- Un autor.
- Escenarios.
- Sección de “No objetivos”.
- Una visión general.
- Detalles.
- Asuntos pendientes.
- Notas al margen.

Además las especificaciones se actualizan con frecuencia. Las puestas al día se van haciendo según el producto va siendo desarrollado y se toman nuevas decisiones. La especificación siempre refleja nuestro mejor conocimiento colectivo de cómo funcionará el producto. Solamente se congela la especificación cuando el producto está en estado de "código completo" (o sea, cuando toda la funcionalidad está completa, pero todavía queda probarlo y depurarlo).

A continuación no se pretende exponer en detalles todas las especificaciones funcionales del sistema, pero si se expondrán algunos escenarios, diagramas y pantallas con el fin de mostrar algunas de las funcionalidades más importantes del nuevo sistema y las ventajas con respecto a la anterior versión.

ESCENARIOS.

Escenario 1. Un redactor.

Martica es una de las redactoras del periódico Vanguardia. Para redactar una nueva noticia en el sitio web online a través de QuipusNews, usualmente accede al sitio de QuipusNews a través de su URL, se identifica y una vez en la página inicial que muestra todos los servicios a los cuales ella tiene acceso, da clic en Redacción y espera varios segundos a que cargue una página que muestra la lista de materiales de trabajo, o sea, los que están siendo redactados. Aquí da clic sobre un enlace que la lleva a los rechazados, esperando otro periodo de tiempo a que cargue la página que le mostrará la información, da clic sobre un título y tras un corto tiempo carga la página que le muestra un formulario con cajas de texto y un editor que le permite llenar los campos, terminar de redactar el documento, al dar clic en el botón Guardar, espera por un pequeño pop up que le notifica el éxito de la acción realizada. Luego Martica corrige los últimos detalles y da clic en el botón Terminar.

Con los esfuerzos realizados por el colectivo de Vanguardia se puso en funcionamiento el nuevo manejador de contenidos, con novedosas características. Entonces, para realizar su trabajo una vez identificada, al dar clic en Redacción, esta muestra instantáneamente la lista de materiales en un panel de pestañas con rejillas y sonríe por lo agradable de la vista o lo rápido que se ha mostrado.

Luego da clic en la pestaña de Materiales Rechazados y “flash” instantáneamente se muestran los rechazados. Ahora da doble clic en uno de los títulos y se crea en el acto otra pestaña que contiene un editor o formulario, mostrando una intuitiva barra de

progreso que le indica cómo se va cargando la información de la noticia en el editor que le permitirá escribir y efectuar los cambios en el texto de la noticia. Al dar clic en el botón Guardar, se muestra en la barra de estado un mensaje que indica que el material está siendo enviado y guardado en el servidor, mientras paralelamente Martica continúa haciendo los últimos ajustes, pocos segundos después se ha mostrado en la barra de estado que el proceso de guardado se ha efectuado satisfactoriamente y la hora en que ha ocurrido. Martica ahora da clic en terminar, entonces se cierra el editor, se muestra un mensaje de notificación y el título desaparece de la vista.

Escenario 2. Un editor.

Mariana es una editora del Periódico Vanguardia. Para realizar su labor en la versión anterior de QuipusNews, una vez identificada en el sistema, hacía clic en el enlace del servicio de Edición y tras unos segundos se cargaba la página con la lista de los materiales de trabajo.

En espera de un material para editarlo, al cabo de cierto tiempo, Mariana actualiza la página, aguardando por la conexión con el servidor para observar si aparece ya el título de lo que espera que en ese momento y que aún está siendo redactada por Martica.

Mariana repite el proceso varias veces y consume una porción de ancho de banda de Internet de la red de Vanguardia, finalmente observa la noticia ya en la página, y dando clic en el título, espera por la respuesta del servidor para que se cargue la página que le muestra el formulario con el editor, y un espacio a la izquierda que le permitirá buscar, subir, y añadir imágenes del repositorio de QuipusNews.

Una vez corregido el material tras haber pasado por los pasos de guardado, subida, búsqueda y selección de las imágenes, cada uno de los cuales conlleva una actualización de la página. Entonces, ella añade una de las imágenes a la zona de imagen de portada y finalmente da clic en terminar, donde la ejecución transcurre cargándose la página de la lista de materiales en edición nuevamente.

Con el uso del nuevo sistema QuipsuNews en Vanguardia la forma de hacer el trabajo de Mariana ha cambiado para bien, pues ella, una vez identificada en el nuevo sistema, al dar clic en el servicio Edición se muestra el panel de pestañas. En el panel de eventos Mariana puede observar un texto que le dice que la noticia X está siendo redactada por Martica.

Sin necesidad de actualizar la página, al cabo de un tiempo, otro material se observa en la rejilla, y en el panel de eventos, Mariana lee la notificación de la terminación de esa noticia por parte de Martica.

Entonces Mariana da clic en el título de la noticia y automáticamente se le muestra el formulario que carga el contenido de la noticia en las cajas de texto y el editor para el resumen y texto de la noticia. Luego Mariana procede con el resto de las operaciones de selección y adición de las imágenes y fichero asociadas a la noticia sin actualizar completamente la página en que se encuentra. Una vez que ha terminado y da clic en el botón Terminar finaliza esta tarea con la misma espontaneidad que Martica en el nuevo Quipus.

Escenario 3. Buscando noticias mediante el uso de filtros.

Con QuipusNews 2.0 es posible en los servicios de redacción, edición, publicación, repositorio de ficheros, la utilización de filtros en cada uno de los campos mediante los cuales se muestran los elementos listados. Si la editora Mariana necesita buscar las noticias en cuyo título se encuentra la palabra “universidad” en el rango de fechas comprendido entre el 15 de mayo y el 3 de junio de este año, da clic en el botón a la izquierda del encabezado de la columna título y se despliega un menú, dando clic entonces en la opción filtro puede añadir la cadena de texto “universidad”.

Luego realiza la misma operación sobre el encabezado de fecha y esta vez en la opción filtro dispondrá de dos calendarios que le permiten especificar la fecha inicial y la fecha final del rango en que desea realizar la búsqueda de las noticias. Una vez que se ha elegido el filtro, el sistema responde actualizando la rejilla con los materiales encontrados que satisfacen los criterios especificados en cada filtro; no es necesario presionar ningún otro botón para ello.

Nota: Es importante tener en cuenta que los filtros permanecerán activos hasta tanto no se desactiven desmarcando el *checkbox* en el menú del filtro correspondiente.

Escenario 4. Asociando imágenes a la noticia.

Estando en el panel de edición, para asociar una imagen a la noticia, el editor presiona el botón Insertar Imagen, que se encuentra en la región inferior del panel de imágenes a la izquierda. El sistema responde entonces abriendo una ventana que muestra las miniaturas de los ficheros en el repositorio que, además, pueden ser filtrados. A la derecha de esta ventana se visualizan las características de cada objeto seleccionado.

Para escoger, el editor da doble clic sobre la imagen o presiona el botón Aceptar. Luego la imagen se coloca en el panel de imágenes, valga la redundancia, después de una transición. La imagen colocada además presenta un *radiobutton* que el editor presiona para asociarlo como imagen de portada. Al realizar esta acción, la imagen es colocada en la zona de imagen de portada localizada en el panel de edición.

2.3 Diagrama y Descripción de los principales casos de uso del Sistema.

Un sistema normalmente involucra muchos tipos de usuarios. Cada tipo de usuario se representa por un actor. Los actores utilizan el sistema interactuando con los casos de uso. Un caso de uso es una secuencia de acciones que el sistema lleva a cabo para ofrecer algún resultado de valor para un actor. El modelo de casos de usos está compuesto por todos los actores y todos los casos de uso de un sistema (Referencia El proceso unificado del software).

El sistema en QuipusNews impone roles bien definidos; a saber: redactor, jefe de información, jefe de redacción, corrector, editor web, publicador, director, y fotógrafo; y también se evidencian casos de uso del sistema a primera vista. El Diagrama muestra los principales casos de uso de QuipusNews, derivados directamente de sus requisitos.

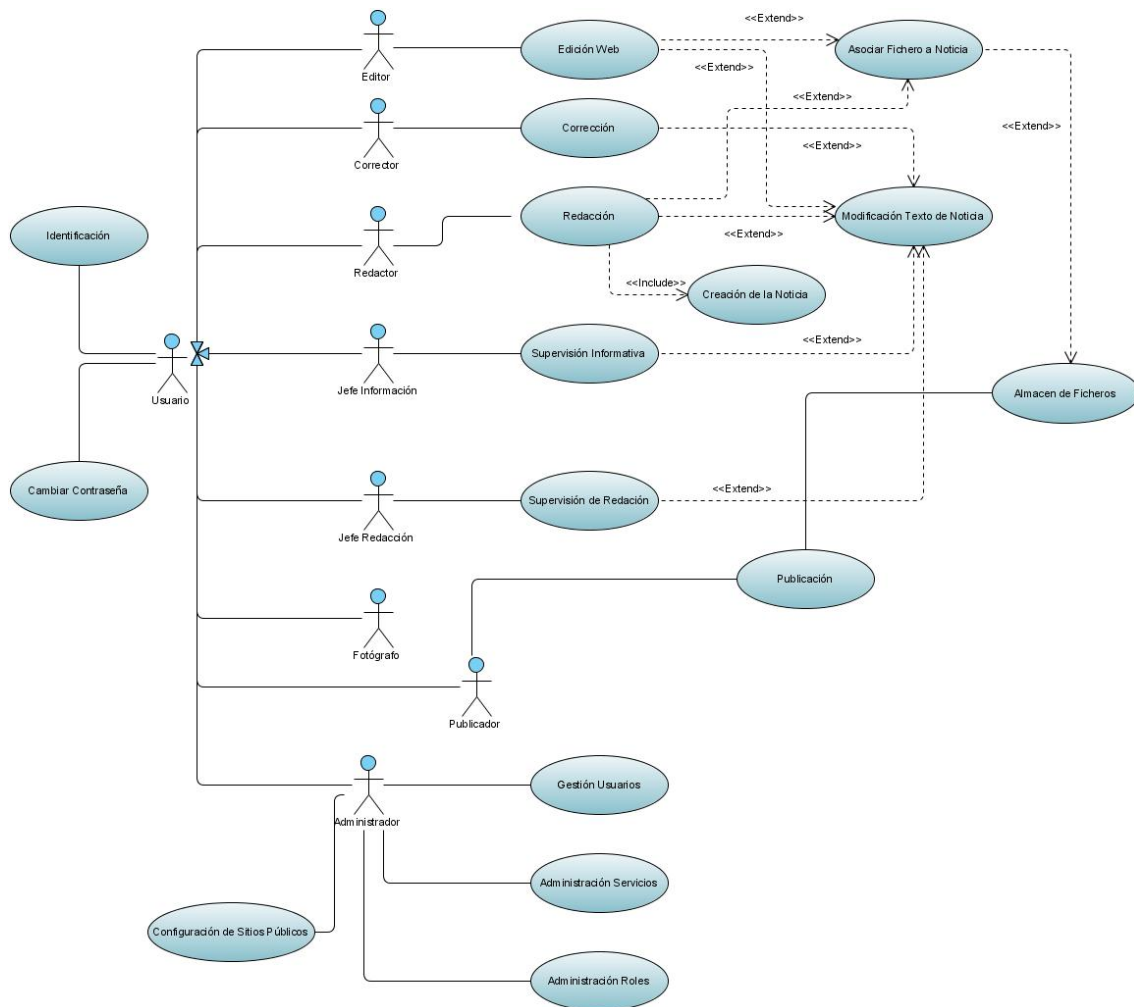


Figura 4. Diagrama de Casos de Uso.

2.3.1 Detalles de algunos Casos de usos.

Entre los casos de usos más importantes del sistema se encuentra los de redacción, edición, asociación de ficheros a la noticia, publicación, Administración de Usuarios, Administración de roles. Estos serán detallados con el fin de ilustrar los procesos de interacción de los actores del sistema con las nuevas interfaces que se han diseñado, además de la importancia que revierten los modelos de casos de usos en el diseño e implementación de un sistema.

IDENTIFICACIÓN.

Caso de uso	Redacción
Actor	Usuario
Precondiciones	Se ha cargado la interfaz del sistema.

Propósito	Chequear identificación del usuario y permitir el acceso al sistema según su rol.
Resumen	El usuario adiciona su nombre de usuario, contraseña y elige un tema de ambiente gráfico. Se le permite el acceso al sistema si los datos suministrados son correctos.
Poscondiciones	Se crea una sesión para el usuario. El usuario accede al sistema con el tema elegido.

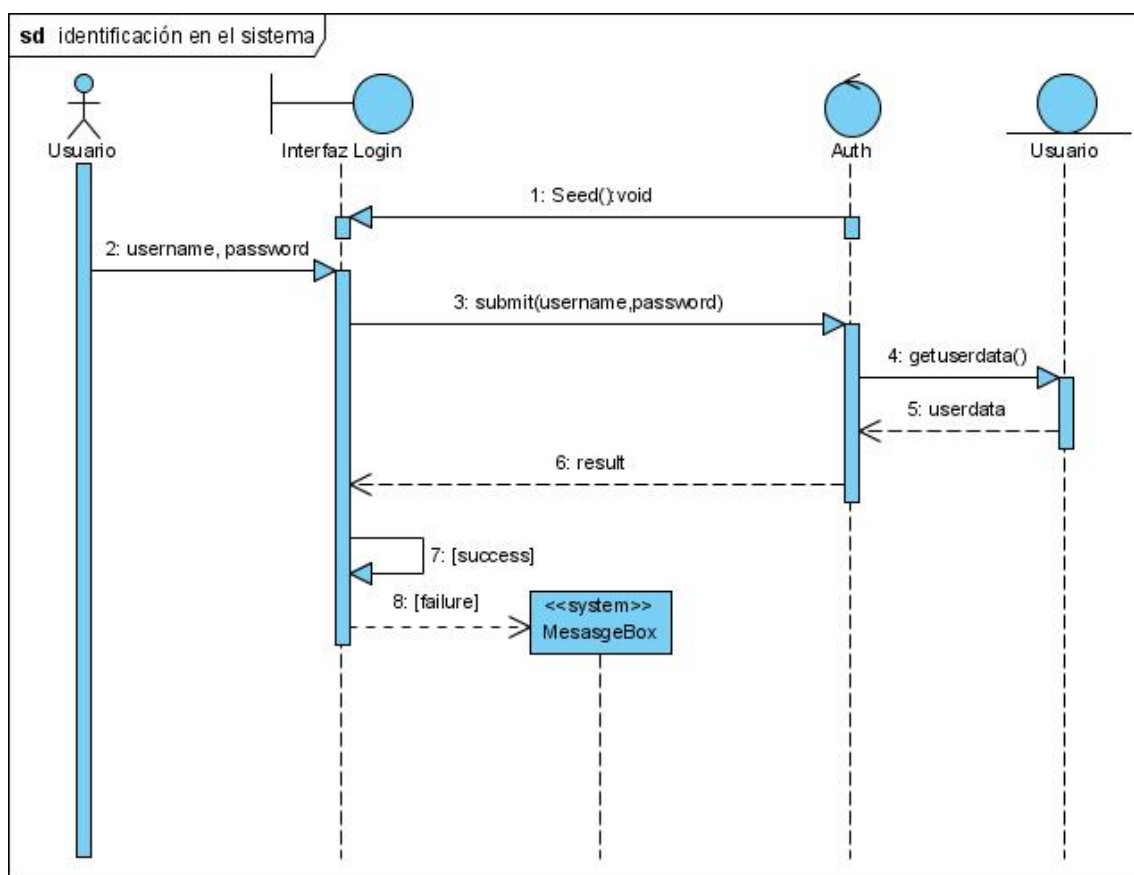


Figura 5. Diagrama de Secuencia.

REDACCIÓN.

Caso de uso	Redacción
Actor	Redactor
Precondiciones	El Redactor está identificado en el sistema. Se ha creado el material

Propósito	Redactar un nuevo material.
Resumen	El redactor debe llenar los campos referente al nuevo material. Escribir el texto del resumen y el cuerpo del material, asociando imágenes.
Poscondiciones	Se ha modificado el texto de la noticia. La noticia pasa al siguiente estado.

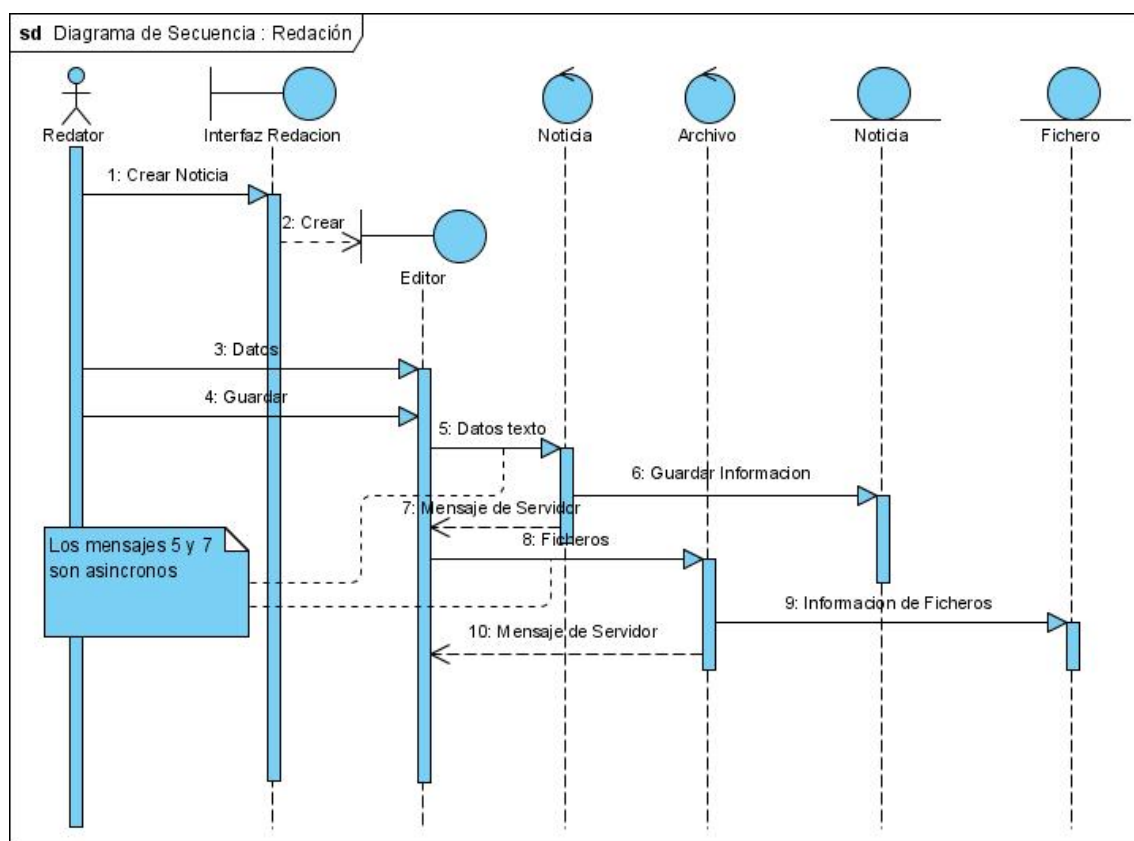


Figura 6. Diagrama de Secuencia Redación.

EDICIÓN.

Caso de uso	Edición.
Actor	Editor
Precondiciones	El Editor está identificado en el sistema. El material a editar ha sido redactado.

Propósito	Editar un nuevo material.
Resumen	El editor puede corregir los campos referente al nuevo material. Se Modifica el texto del resumen y el cuerpo del material, asociando imágenes. Además se añade la imagen de portada.
Poscondiciones	Se ha modificado el texto de la noticia. La noticia pasa al siguiente estado.

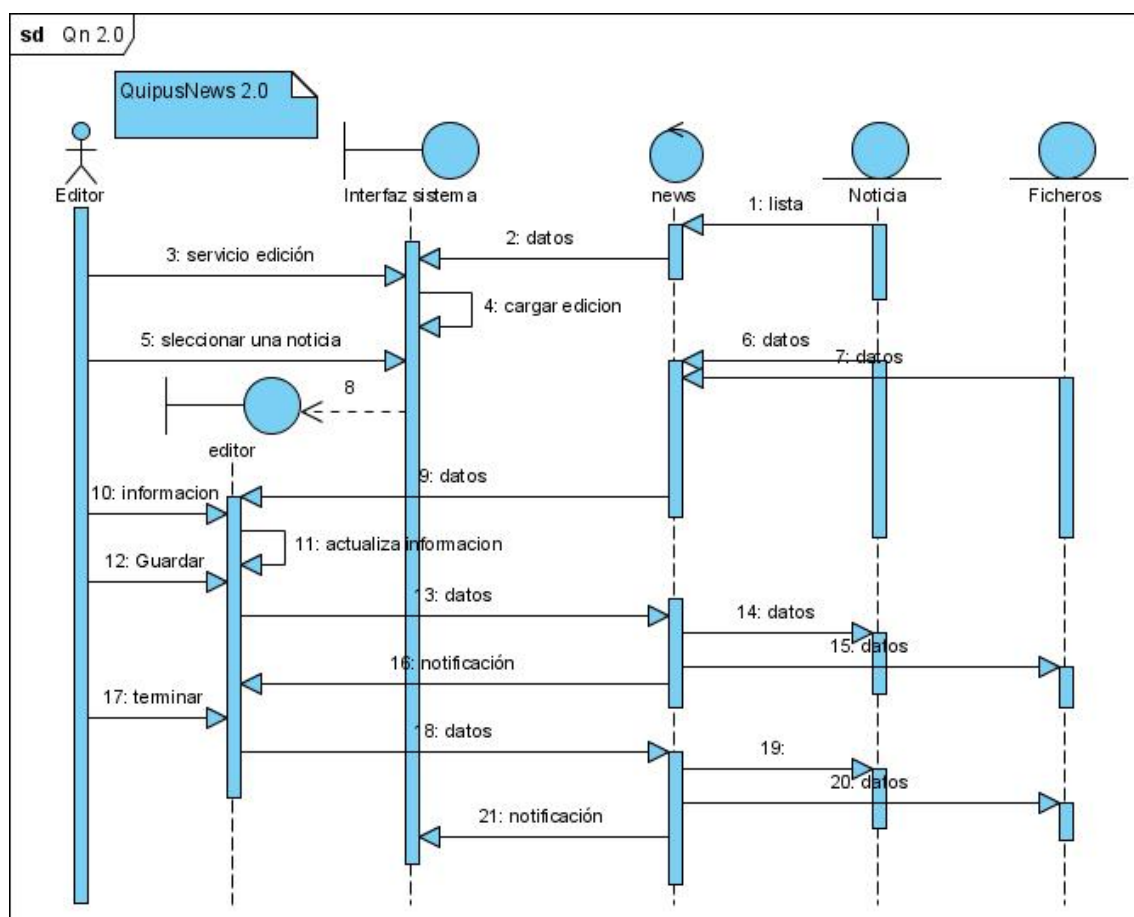


Figura 7. Diagrama de Secuencia Edición.

A continuación se muestra un diagrama de secuencia que muestra la interacción entre las clases de la versión anterior de QuipusNews. El diagrama ha sido realizado usando

técnicas de ingeniería inversa, por lo cual no es tan preciso, sin embargo nos sirve si queremos observar las diferencias o elementos comunes del diseño.

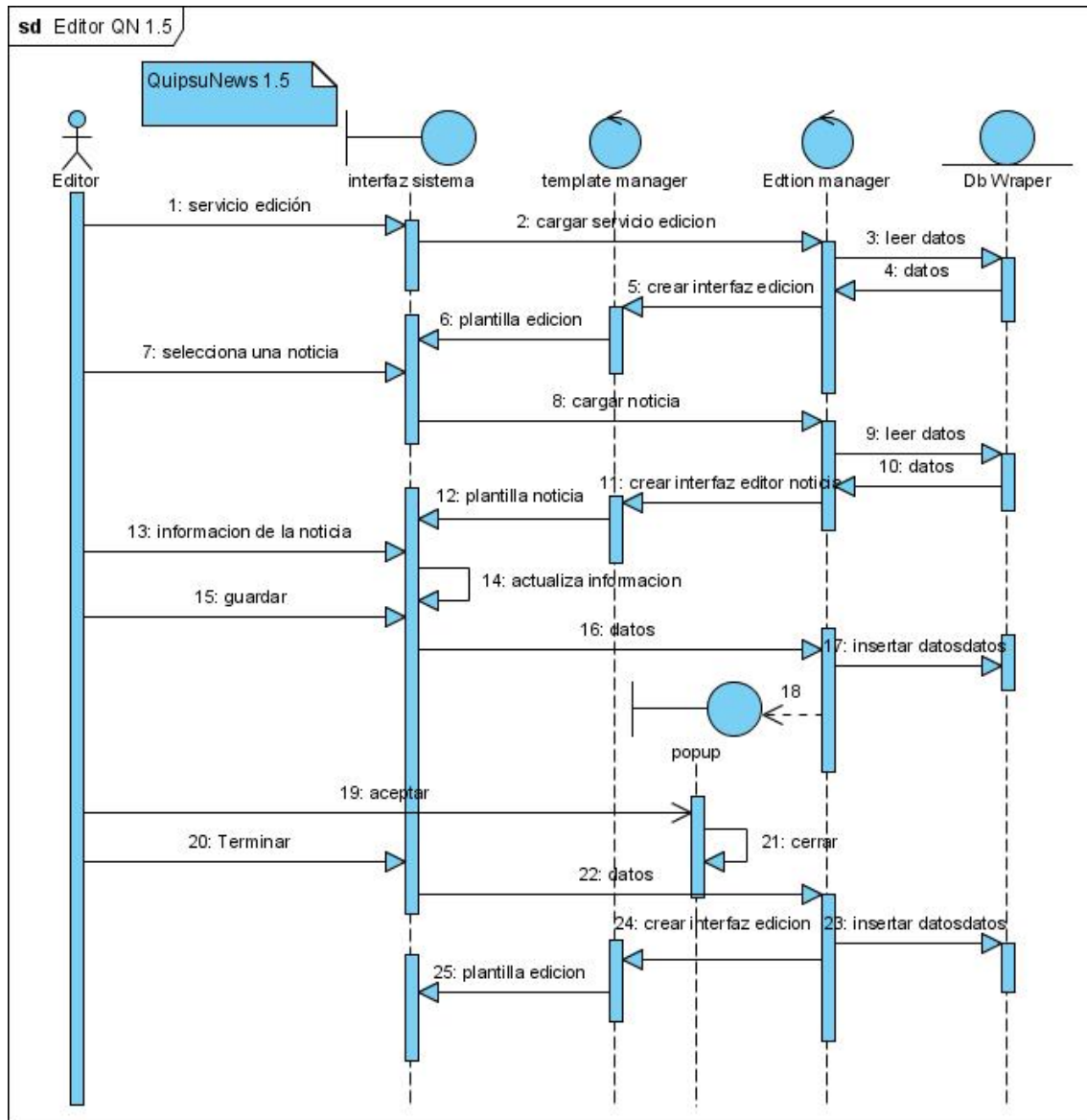


Figura 8. Diagrama de Secuencia QN 1.5.

2.4 Descripción del Flujo de Trabajo. Diagramas de Actividades.

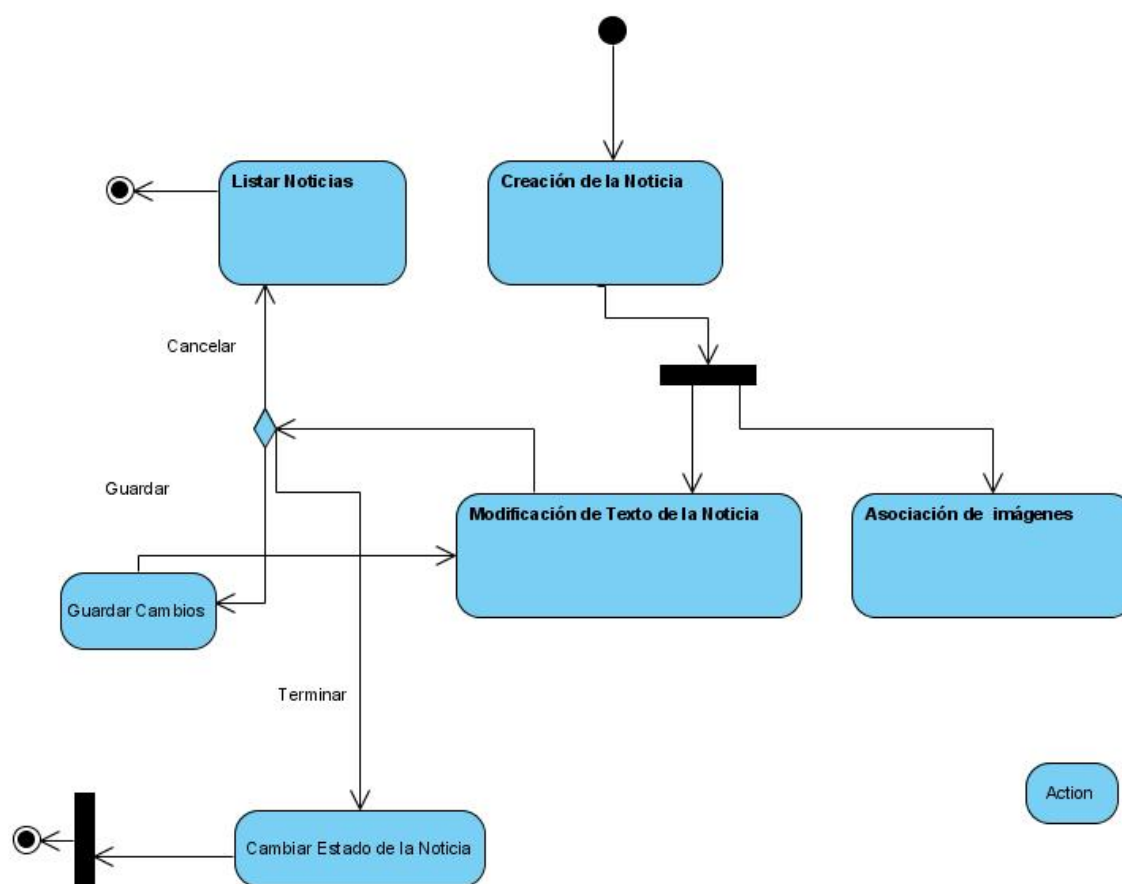


Figura 9. Diagrama de actividades.

2.5 Diseño de las nuevas interfaces de usuario e interacción con el sistema.

ExtJs presenta muchos “widgets”. Estos incluyen componentes tales como diálogos de mensajes, rejillas, ventanas, y otros más llamativos que conforman una función en particular de una interfaz de usuario. Es preferible ver los componentes como “onReady” más como funciones del núcleo y solo referirse a los componentes que constituyen un rol específico de interfaz de usuario como un “widget”-por ejemplo una rejilla que es empleada para presentar datos tabulados al usuario.

Ext provee un conjunto de funciones para construir estos componentes, las cuales reciben como uno de sus parámetros un objeto de configuración de tipo JSON.

Veámos un ejemplo de código para la creación de un *MessageBox* y para la creación de una Ventana.

```
1 Ext.Msg.show({
2     title: title,
3     msg: msg,
4     minWidth: 200,
5     modal: true,
6     icon: Ext.Msg.INFO,
7     buttons: Ext.Msg.OK
8 });
9
```

```
1 var win = new Ext.Window({
2     layout: 'column',
3     title: 'Entrar al Sistema',
4     width: 300,
5     closable: false,
6     resizable: false,
7     plain: true,
8     border: false,
9     items: [login]
10 });
```

Para el diseño y la construcción de estas interfaces de usuario utilizó la herramienta *GuiDesigner* que está hecha sobre *ExtJS* (véase figura en anexo B). Además es posible emplear un *plugins* que existe para *eclipse* para diseñar de forma visual estas interfaces arrastrando componente con el *Mouse* de forma fácil, al estilo de muchos IDEs.

2.6 Implementación del Sistema.

2.6.1 Módulos del sistema.

Estructuralmente la aplicación está dividida en tres módulos o capas.

- **Capa de acceso a datos.** Está incluye la implementación de un ORM para la abstracción del acceso a la base datos.
- **Módulo Cliente.** Este está conformado por las clases y componentes que hacen uso de la biblioteca *ExtJS* y que se ejecutan en el lado del cliente como código Javascript.
- **Módulo de Servicio.** Este reúne las clases y funcionalidades implementadas en PHP 5 vinculadas al framework *CodeIgniter*.

2.6.2 Diagramas de Clases. Clases Controladoras.

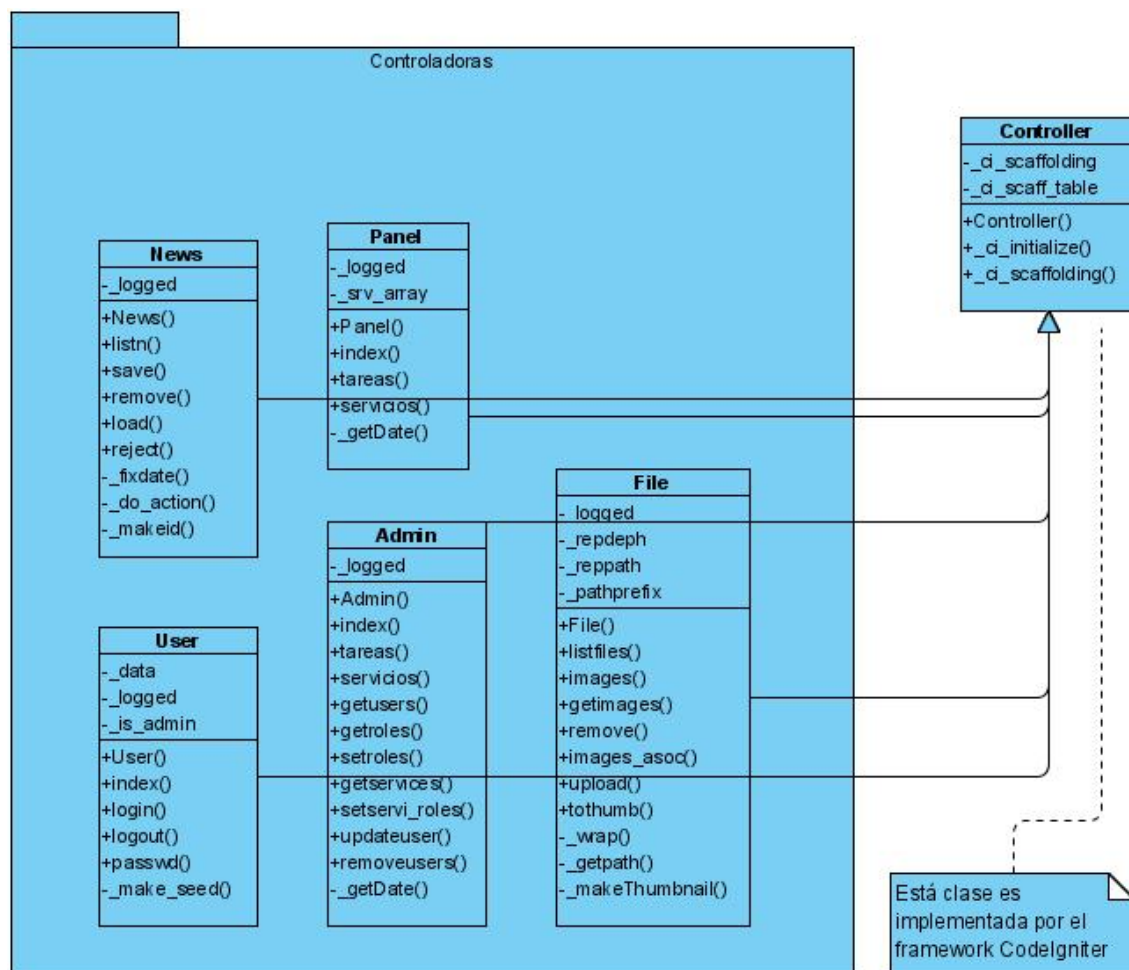


Figura 10. Diagrama de Clases Controladoras.

2.6.3 Extensión de Clases en Biblioteca ExtJS.

En el diagrama de clases se muestra en color verde las clases de ExtJS que fueron extendidas para crear nuevos componentes con funcionalidades particulares.

Para extender de una clase de ExtJS se llama al método “extend” de la clase “singleton” Ext que recibe como parámetros el nombre de la superclase y un objeto de configuración en formato JSON que sobrescribe los métodos de la clase padre. De esta forma no es necesario hacerlo de la forma tradicional en Javascript en el cual se usa el método “prototype”. El objeto de configuración contiene además un miembro especial llamado “constructor” para definir el constructor de la nueva clase, sino se especifica el constructor se crea uno por defecto. Mediante el constructor se pueden definir los nuevos métodos de la subclase. Veamos un ejemplo de código:

siendo una ventaja del desarrollo del software libre y de código abierto. Esto, además, nos permite extender el código de DABL y adaptarlo a nuestras necesidades en caso de ser necesario.

EJ: Fue necesario modificar el generador de clases para que tuviera en cuenta que pueden existir campos en una tabla de la base datos que apunten a otra tabla con la cual está relacionado, y que está tenga igual nombre, lo que trae consigo errores de redefinición de métodos si no se tiene en cuenta.

DBAL contiene un script para generar las clases. Este crea dos clases por cada tabla en la base datos. La primera localizada en `classes/tables/base` contiene todos los métodos nativos de DBAL. Esta clase base es abstracta y no puede ser usada por si misma. La clase base no debe modificarse manualmente, porque será sobrescrita cada vez que se ejecute el script en `tools/mapDB.php`. La segunda clase se localiza en `classes/tables` y tiene el mismo nombre de la tabla, y es la clase que debe ser usada para instanciarla. Esta clase hereda a la primera e inicialmente está vacía, por lo cual puede ser editada con métodos y atributos propios, una vez creada no es modificable por el script generador (Blaisdell 2009).

Aunque no es requerido por DBAL para funcionar es muy recomendado el uso del motor de almacenamiento innoDB de mysql para las tablas de la base datos, ya que este permite el uso de llaves foráneas y relaciones entre tablas para mantener la integridad referencial, lo cual mantiene su base datos limpia y libre de posibles errores. DBAL reconoce las llaves foráneas y crea métodos en las clases para ello (`$customer->getOrders()` - method recognizing relationship between Customer and Order).

Mostrar ejemplos de código de inserción actualización y eliminación de filas de tablas en la base datos.

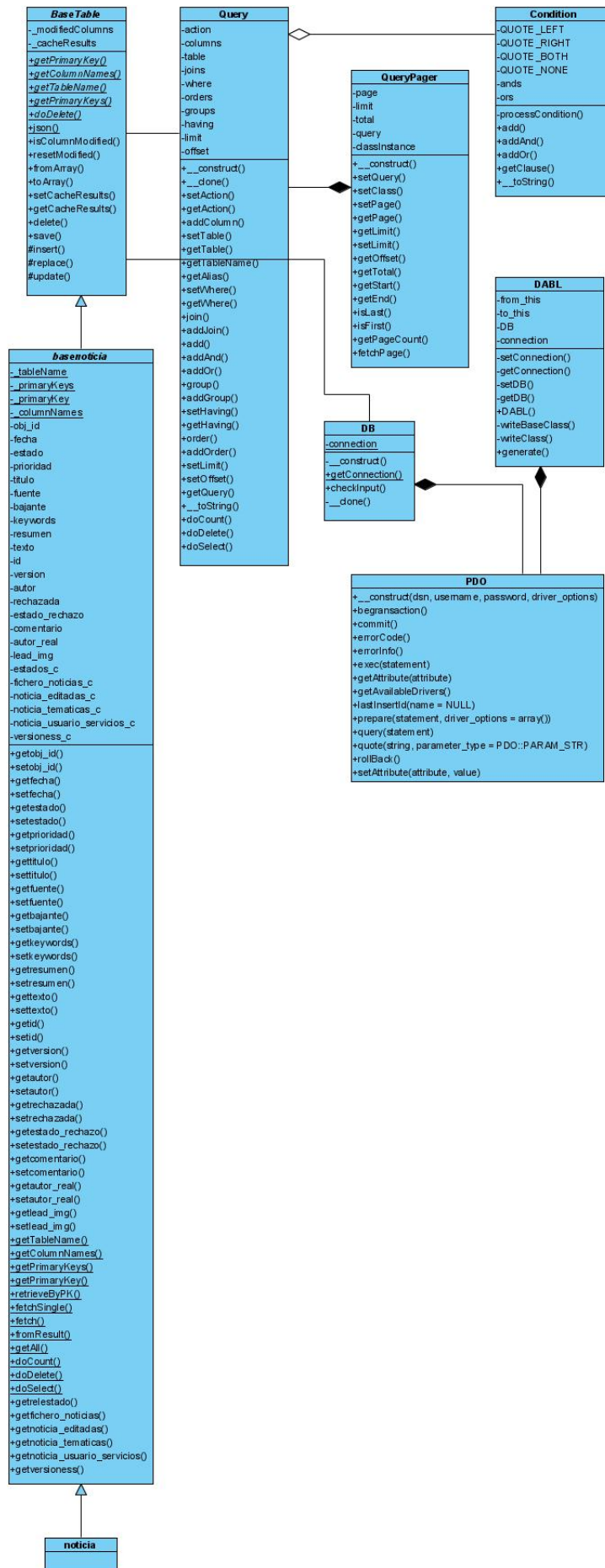


Figura 12. Diagrama de clases ORM.

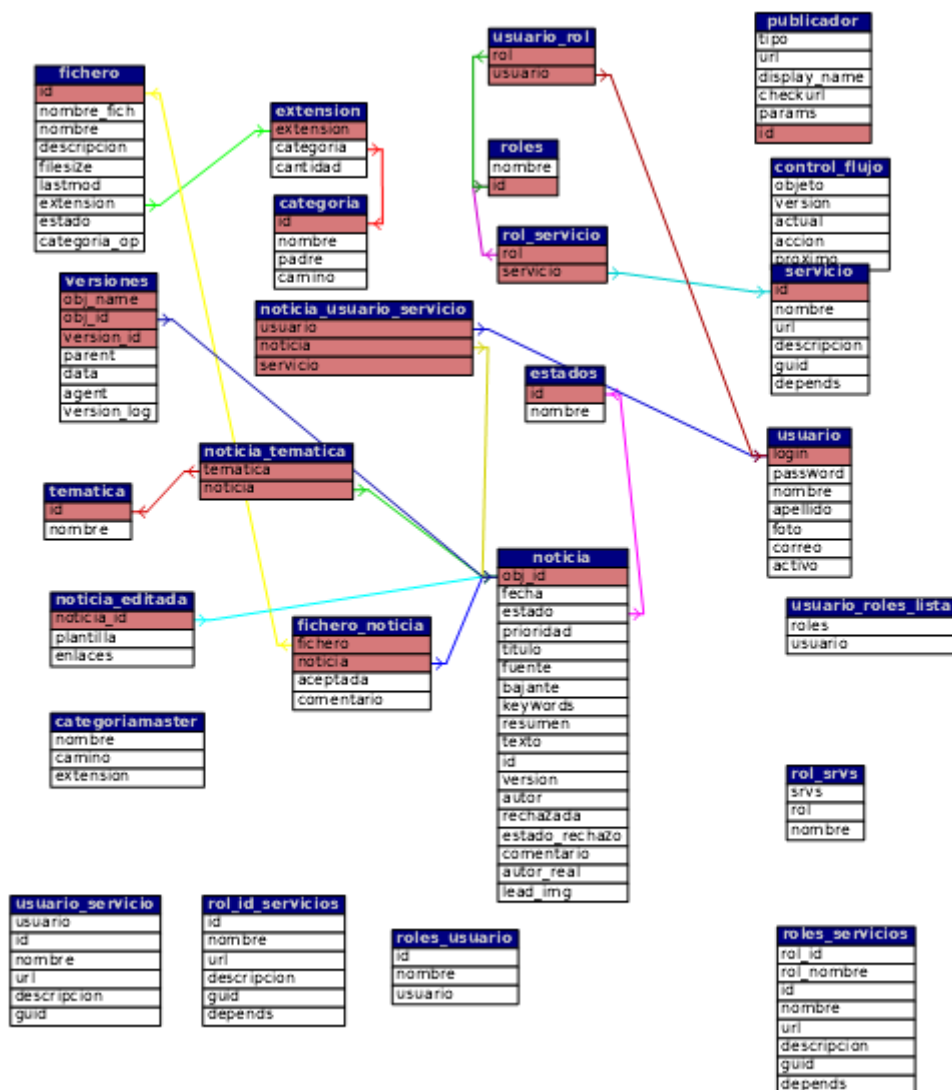


Figura 13. Modelo de Base Datos.

2.6.4.1 Creación de Vistas en el gestor de base datos.

Una de las tareas de investigación que surge en el proceso de implementación de la aplicación es la de crear vistas que simplifiquen la complejidad de algunas consultas realizadas a la base de datos.

Observemos en la siguiente figura el diagrama de Entidad Relación de la base de datos, donde se observa las nuevas vistas que se crearon

(figura obtenida con phpmyadmin.)

Veamos un ejemplo de una de las sintaxis para la creación de esta vista.

(mostrar un ejemplo de código)

2.7 Conclusiones del Capítulo

El estudio y uso de la biblioteca ExtJS ha permitido el diseño de interfaces de usuarios con las características que buscamos para la nueva aplicación web y que de otra manera son muy difíciles de implementar, o sea, sin el uso de una biblioteca que ofrezca esta capacidad. La escritura de especificaciones funcionales ayudó en gran medida en la planificación del diseño de estas interfaces, y sirvió de guía en el proceso de producción del software.

La implantación de un ORM trajo como resultado una metodología de trabajo que le da un mayor nivel al código fuente de la aplicación y por consiguiente lo hace más fácil de entender y escribir a un programador.

Luego con las herramientas usadas se logró implementar el servicio de Redacción, Edición, Repositorio y se continuó en otros de los servicios necesarios en QuipusNew como son el de Edición Maestra, Publicación, y el de Administración que consta de varias secciones. De este modo se han sobrepasado las expectativas en cuanto a la construcción de la aplicación.

Capítulo 3 Manual de Usuario.

El siguiente Capítulo constituye una guía detalla para el usuario en cuanto a la instalación, requisitos necesarios para la utilización del sistema y otros aspectos de ayuda.

3.1 Nuevas Características en QuipusNews 2.0.

Entre las nuevas características de QuipusNews 2.0 podemos listar:

- Componentes añadidos al editor
- Seleccionador de fecha
- Nuevo editor de texto
- Barra de estado con tiempo, mensajes de notificación y Estado de la Noticia
- Nuevo panel de imágenes
- Presentación en pestañas de forma dinámica.
- Componente de Interfaz de Usuario Grid.
- Campos añadidos id Noticia y Título.
- Selección por el usuario de Campos a mostrar
- Barra de paginación con botones de navegación.
- Permite selección de varias filas simultáneamente
- Filtros para búsqueda avanzada fácil de usar
- Actualización asíncrona.
- Contiene barra de Herramientas

3.2 Requisitos e instalación del Sistema.

Una cuestión sin duda determinante en la importancia de uso del software son los recursos necesarios de los que se debe disponer para explotarlo. Elemento este muy importante si tenemos presente los pocos recursos con los que cuenta el país sobre todo en materia de hardware.

El alojamiento de sitios web de forma general requiere de máquinas o medios de cómputos más potentes que los ordenadores de escritorio que empleamos para realizar nuestro trabajo, en este sentido hablamos de servidores. Los servidores se encargan a través de los servicio que brindan a los usuarios, del procesamiento de un gran flujo de

información, el almacenamiento de datos, al tratarse de un servicio web nos ofrecen la posibilidad de acceder desde cualquier geografía.

3.2.1 Requerimientos de hardware.

En cuanto a las características de velocidad de procesamiento, capacidad de memoria RAM y capacidad de almacenamiento en disco duro, QuipusNews no requiere de muchos recursos. Es recomendable entonces su instalación en una máquina con un procesador de al menos 1 GHz, 256 MB de memoria RAM y 15 MB de espacio en disco duro (se necesita más espacio para el repositorio de imágenes, que puede variar). Es suficiente una conexión de 128 Kb/s aunque es recomendable aumentar el ancho de banda para lograr un mejor rendimiento.

3.2.2 Requerimiento de software.

Para el funcionamiento del sistema es indispensable la disponibilidad de **PHP 5** como módulo de **Apache** en el servidor web donde se alojará. Además PHP debe contar con las extensiones **PDO** para **MySQL** (pdo_mysql.so) y **GD2** (php_gd2.so) esta es usada para la gestión de la base datos por parte del framework usado.

Por otra parte la aplicación es multiplataforma tanto del lado del cliente como en el servidor web. Esta es capaz de ejecutarse en cualquier sistema operativo y arquitectura, siempre que se cuente con el software necesario.

3.2.3 Pasos para la Instalación.

Copie y descomprima el archivo con el contenido de QuipusNews en un directorio raíz del servidor web. En la documentación adjunta de CodeIgniter podemos hallar cómo configurar variables específicas del sistema. Es necesario que el directorio en el que se encuentra el repositorio tenga permiso de escritura y lectura.

3.2.3.1 Instalación de la base datos.

1. Cree una base de datos en MySQL y un usuario con permisos de SELECT, INDEX, ALTER, UPDATE, DELETE (ALL PRIVILEGES) para esa base datos.
2. Ejecute el script SQL **data/qn.sql** esto creará todas las tablas y algunos datos necesarios para el correcto funcionamiento de qn.
3. Actualize en el fichero **system/application/config/database.php** las variables \$db['default']['hostname'], \$db['default']['username'] \$db['default']['password']

`$db['default']['database'];` se usan para establecer el servidor de MySQL, el usuario, el password y el nombre de la base datos.

3.3 Despliegue de QuipusNews.

Actualmente el periódico Vanguardia cuenta con dos máquinas para funciones administrativas de la red, que brindan varios servicios necesarios para el proceso productivo en el centro, entre estos se tiene un servidor web para la red interna y un servidor proxy que comunica la red del periódico con Internet. Esta disposición se tiene en cuenta a la hora de pensar en establecer el despliegue de una arquitectura que permita aumentar el rendimiento en el uso de QuipusNews, mediante la optimización del ancho de banda, el incremento de la velocidad de carga del sistema y por consiguiente la seguridad en la transferencia de la información.

Con vistas a implementar esta idea es necesario seguir la siguiente metodología:

1. Todos los archivos de tipo estático (imágenes, script de la biblioteca ExtJS, hojas de estilos y otros ficheros) se alojarán en el servidor web local. De manera que se cargue desde aquí por parte de los navegadores de los usuarios de la red local, evitando así la conexión a Internet con estos fines.
2. Para lo primero debe implementarse un procedimiento que se ejecute a nivel de usuario y detecte la disponibilidad de contenido dinámico en un servidor local y por consiguiente lo cargue desde este. En caso de que no exista esta disponibilidad en un servidor local queda la opción de cargarlo desde el sitio original donde se aloja QuipusNews.
3. En caso de falla en la conexión a Internet: se guardará la información temporalmente en el servidor, de forma tal que una vez restablecida la conexión la información guardada pueda actualizarse.
4. Para el inciso anterior debe implementarse en dicho servidor local un servicio que sea especie de una sucursal de QuipusNews, con el objetivo de mantener el flujo de trabajo en el sistema.

Con los planteamientos vistos anteriormente se puede llegar a una conclusión sin un previo análisis en la práctica, y es que estos satisfacen nuestra hipótesis de que es posible mejorar la velocidad y rendimiento del nuevo QuipusNews a la vez que las

mejoras y extensiones en las nuevas interfaces de usuarios traen aparejadas un sistema más robusto y operacional.

3.4 Comenzando en QuipusNews 2.0.

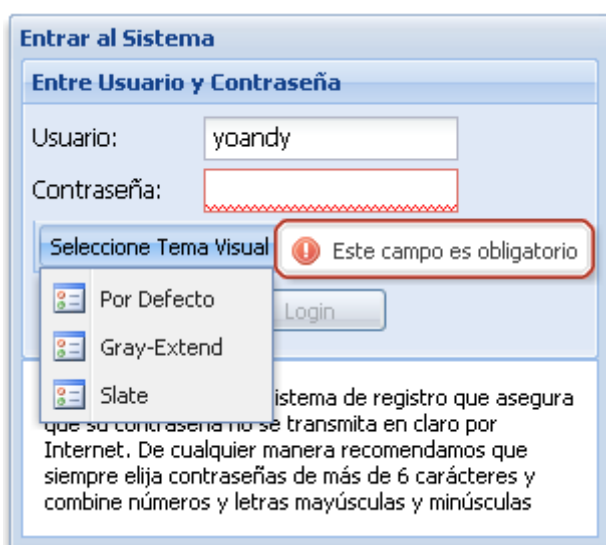
La interfaz de usuario en QN 2.0 está construida usando los “widgets” de la biblioteca ExtJS. Esto quiere decir que los componentes como los paneles son redimensionables y pueden ser contraídos y expandidos. Las ventanas son movibles, redimensionables, se pueden cerrar, maximizar. De maneras que estas interfaces son muy cercanas a la una aplicación de escritorio.

Cuando se menciona “sistema” se está haciendo referencia a la aplicación QuipsuNews 2.0.

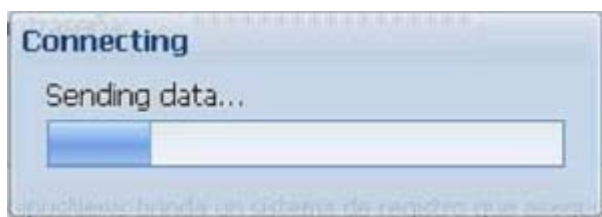
3.4.1 Entrando en el Sistema.

Para acceder a QuipusNews se necesita abrir un navegador de Internet y escribir en la barra de direcciones el URL donde el administrador de la red haya colocado el sistema, o el URL que brinde el proveedor del servicio. En este manual supondremos que el URL es <http://quipusnews.cu>.

Una vez que se conoce el URL donde se encuentra QuipusNews, el usuario debe abrir un explorador, escribirlo y ordenar que se navegue a esa dirección. Hecho esto, aparecerá en su explorador un formulario similar al de la siguiente Figura:



La Ventana de identificación en el sistema contiene un formulario con dos cajas de texto en las cuales se introducirá respectivamente el nombre de usuario y la contraseña. Estos datos deben ser suministrados por el administrador del sistema a cada usuario. Una vez que los datos se han entrado correctamente el botón “Entrar” es activado. La ventana contiene además un menú desplegable para la selección de un tema visual. Al presionar la tecla Enter o el botón Entrar de la ventana se mostrará una ventana con barra de progreso y un mensaje que indica que los datos están siendo enviados al servidor para su chequeo, véase la figura.



En caso de error se mostrará una ventana indicando el tipo de error. Si la identificación transcurre satisfactoriamente se cargará el marco principal de la aplicación. Por marco principal se entiende la página principal, es preferible llamarlo de esta forma para diferenciar de un sitio web convencional, donde una página se espera que contenga información legible e hipervínculos a otras páginas. Debe notarse que en el marco principal de la aplicación cuando se presiona en un botón o una acceso directo a uno de los servicio las interfaces correspondientes se muestran sin actualizar la página, por tanto no se evidencia aquí el concepto de vínculo o hipervínculo. En concreto definimos a QuipsuNews 2.0 como una “aplicación de escritorio” en un entorno web.



1. Fecha del sistema.
2. Panel de Tareas (Muestra accesos directos a los servicios a los cuales tiene acceso el usuario).
3. Botón para expandir y contraer el Panel de Tareas.
4. Panel Contenedor.
5. Muestra en el sitio donde se encuentra el usuario.
6. Botones del sistema para salir y cambiar contraseña respectivamente.

3.4.2 Cambiar Contraseña.

Para cambiar la contraseña se presiona en el botón ubicado en la zona superior derecha del marco principal y se muestra un diálogo como el de la figura:

Observe en la siguiente figura como se señalan los campos del formulario advirtiéndole que es obligatorio llenarlos para completar la acción con éxito. Cuando todos

los campos han sido llenados, se habilita el botón Aceptar que es presionado para completar la operación. Luego se le notificará con *MessageBox* si se efectuó correctamente el cambio de contraseña.

3.5 Manual del redactor.

El servicio redacción tiene una interfaz como la que muestra la figura. En la rejilla se muestran los materiales de trabajo que están en redacción. Para abrir uno de los materiales listados se da doble clic sobre este en la fila de correspondiente de la rejilla. En respuesta a esta acción el sistema muestra el panel de edición en una pestaña adicional que luego puede ser cerrada.

<input type="checkbox"/>	Id Noticia	Titulo	Autor	Fecha
<input type="checkbox"/>	17	bla	yoandy	25/04/2009

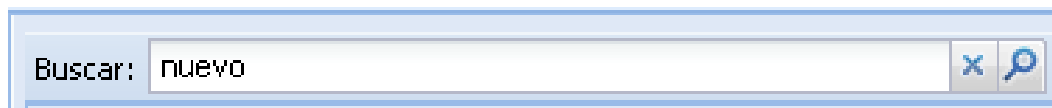
1. Número de elementos listados.
2. Barra de herramienta superior.
3. Buscador.

4. Rejilla.
5. Barra de Paginación.
6. Botón Actualizar.
7. Botón Eliminar (elimina los elementos seleccionados)
8. Botón Crear Material (Abre un nuevo panel de edición)

3.5.1 Buscador.

En la figura se muestra una imagen ampliada del buscador incluido en la mayoría de los servicios de QN 2.0.

Para buscar se inserta un texto y se presiona la tecla Enter o el botón Buscar, como respuesta a esta acción se listarán los materiales que contienen una porción de texto en su título que coincide con el texto añadido. Para mostrar todas las noticias en la rejilla se presiona el botón Cancelar Búsqueda.



1. Caja de texto (se introduce el texto de búsqueda).
2. Botón Cancelar Búsqueda.
3. Botón Buscar.


3.5.2 Panel de Edición.

El panel de Edición es el formulario donde se adicionan los campos de la noticia, estos son: Título, Autor, Fecha, Fuente, Bajante, Palabras Claves, Resumen y Contenido. El título es un campo obligatorio. Si el Autor no se inserta se toma el que está identificado en el sistema por defecto.

Editor 16

Título:


Autor:

Fecha:
 

Fuente:


Bajante:







Palabras Claves:



Resumen:

Contenido:

Tahoma 

B
I
U
A
A
A
ab







Implementado nuevo servicio

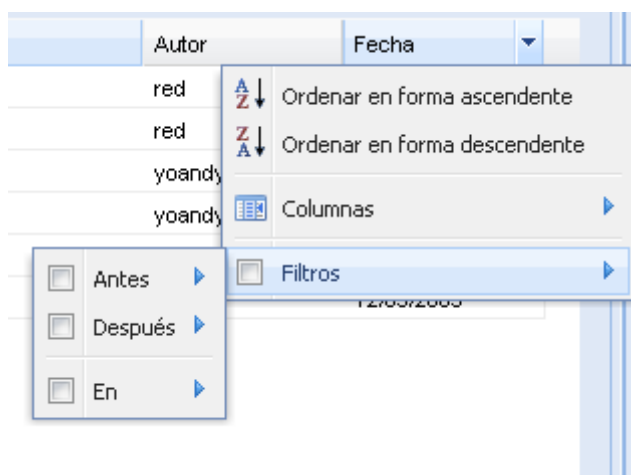
Estado: Edición 2:18:52 AM

1. Botón guardar y Terminar.
2. Barra de Estado.
3. Muestra el estado de la noticia.
4. Hora del sistema.

Cuando se presiona el botón Guardar se muestra una notificación en la barra de Estado como la que se observa en la figura.

3.5.3 Filtro de Materiales.

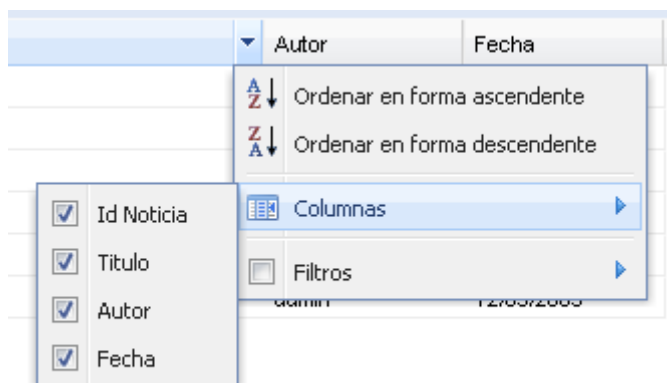
En los componentes “rejillas” o “grid” que se encuentran en cada uno de los servicios para listar los materiales es posible realizar búsquedas avanzadas mediante el filtrado por columnas según el tipo de datos. Por ejemplo el filtro de una cadena de texto es una o varias palabras, para una fecha es una fecha posterior, anterior o específica, para un número es un número mayor, menor, o igual. El filtro se entra a través del menú que se despliega desde el botón de despliegue en el encabezamiento de columna que se encuentra en la Rejilla. Observe la figura donde se muestra la elección de un filtro de fecha.



Es importante que si no se desactiva el filtro en el botón de chequeo del menú siempre se visualizaran los materiales filtrados aunque se actualice la Rejilla.

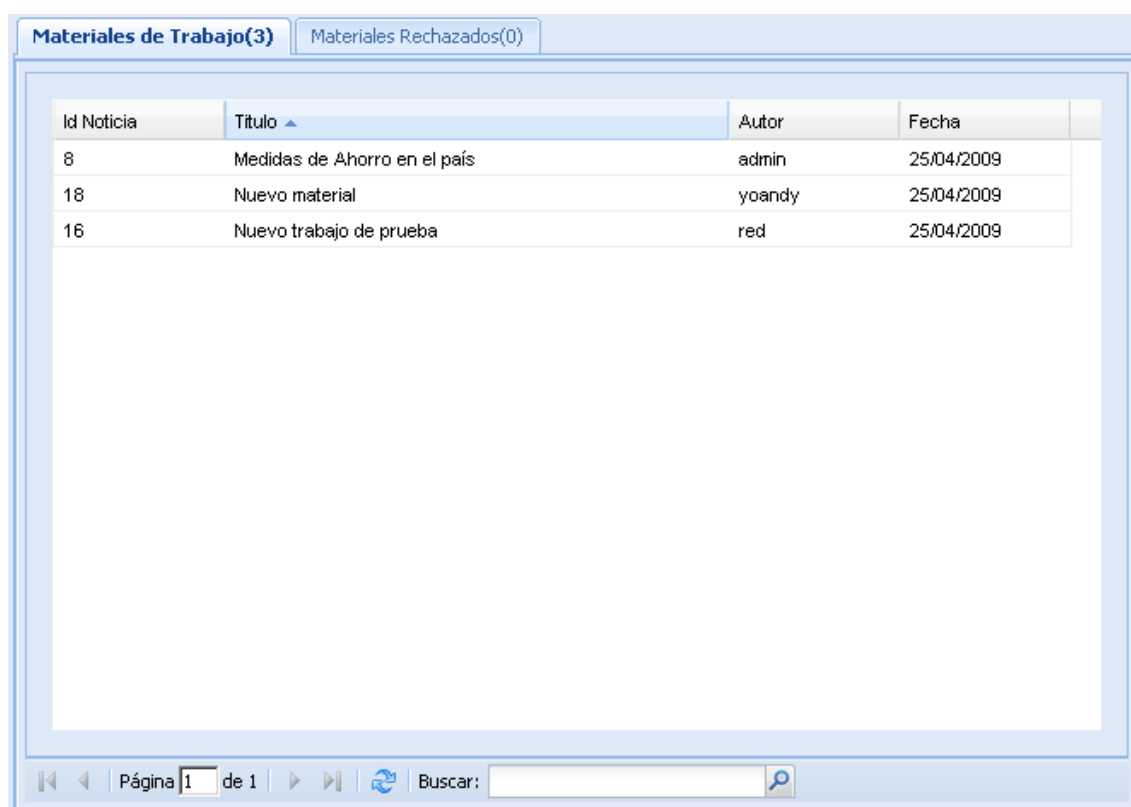
3.5.4 Mostrar u Ocultar campos en la Rejilla.

Desde el menú que se expande en el encabezamiento de las columnas de la Rejilla se puede seleccionar los campos que queremos sean visibles. Para ello expandimos el menú en la opción Columnas y seleccionamos las columnas que se mostrarán. Por defecto se visualizan todas las columnas. Observemos la figura.



3.6 Manual del Editor.

En la figura se muestra la interfaz del servicio edición, similar a la de redacción, con la diferencia de que el buscador se encuentra en la barra de herramientas inferior.



1. Rejilla.
2. Barra de Herramientas inferior.
3. Buscador.

3.6.1 Panel de Imágenes.

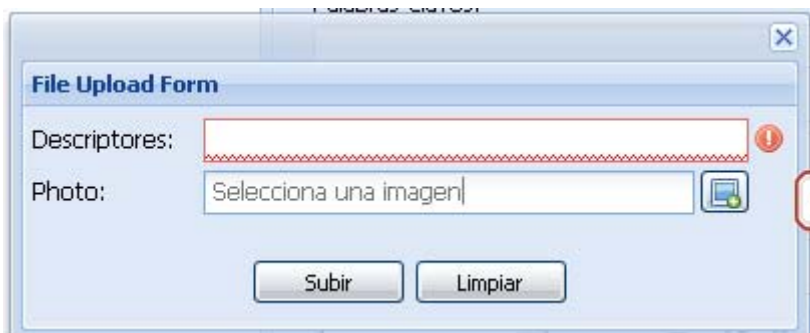
El panel de imágenes es usado para subir, seleccionar y asociar las imágenes al material que se está editando. Además para asociar la imagen de portada al material. Obsérvese en la figura



1. Botón Contraer y Expandir.
2. Botón Radio para seleccionar la imagen de portada.
3. Botón Eliminar Imagen (elimina la imagen respectiva).
4. Imagen en miniatura.
5. Botón Subir Imagen.
6. Botón Insertar Imagen.

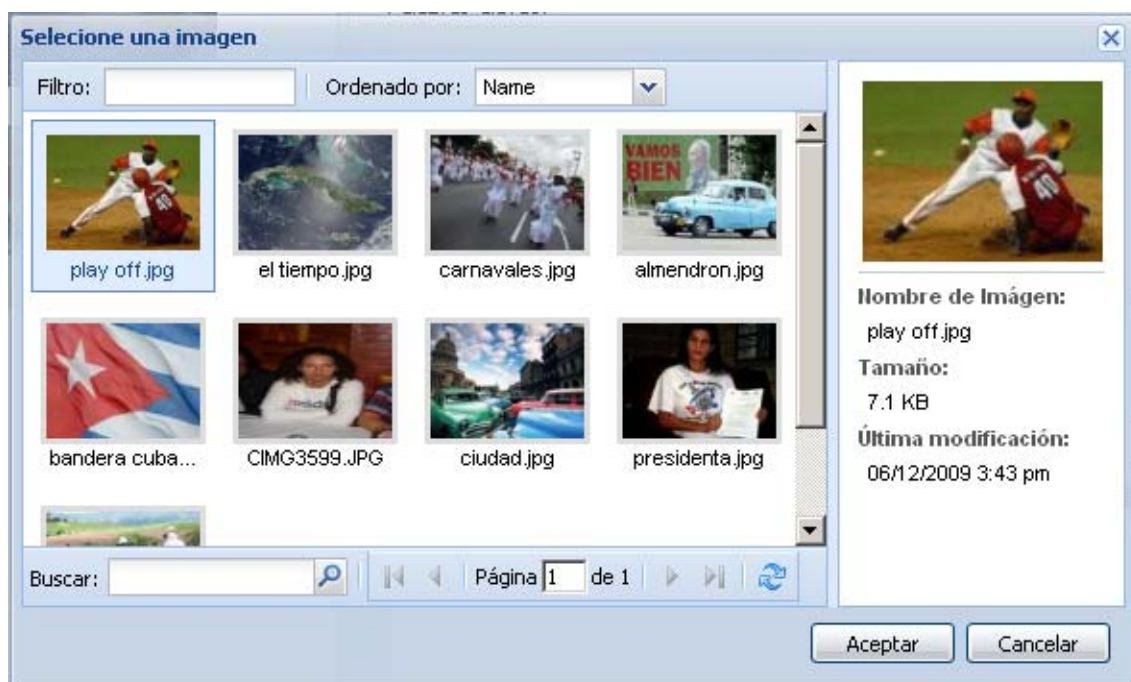
3.6.2 Subir Imagen.

Para subir una imagen al servidor que se almacenará en el repositorio. Presionamos el botón Subir Imagen del Panel de Imágenes. Entonces se muestra una ventana como la de la figura.



3.6.3 Insertar Imagen.

Para insertar una imagen dentro el Panel de Imágenes presionamos el botón Insertar Imagen y se mostrará la ventana de selección de imágenes que se muestra en la figura. Las imágenes en miniatura que se ven en esta ventana se cargan de repositorio de ficheros. En la región a la izquierda se puede observar los datos de la imagen que es seleccionada, como: el nombre, tamaño y la fecha de última modificación. Esta ventana también contiene un filtro y un *combo* para ordenar la lista en la barra de herramientas superior. En la barra de herramientas inferior está ubicado un buscador y una barra de paginación de los elementos listados.



1. Caja de texto para añadir un filtro.
2. Seleccionador de campo para organizar.

3. Buscador.
4. Barra de Paginación.

3.6.4 Edición Maestra.

El servicio de Edición Maestra permite editar aquellas noticias que se encuentran en el estado de publicación. Vea en la figura.

3.7 Manual del Publicador.

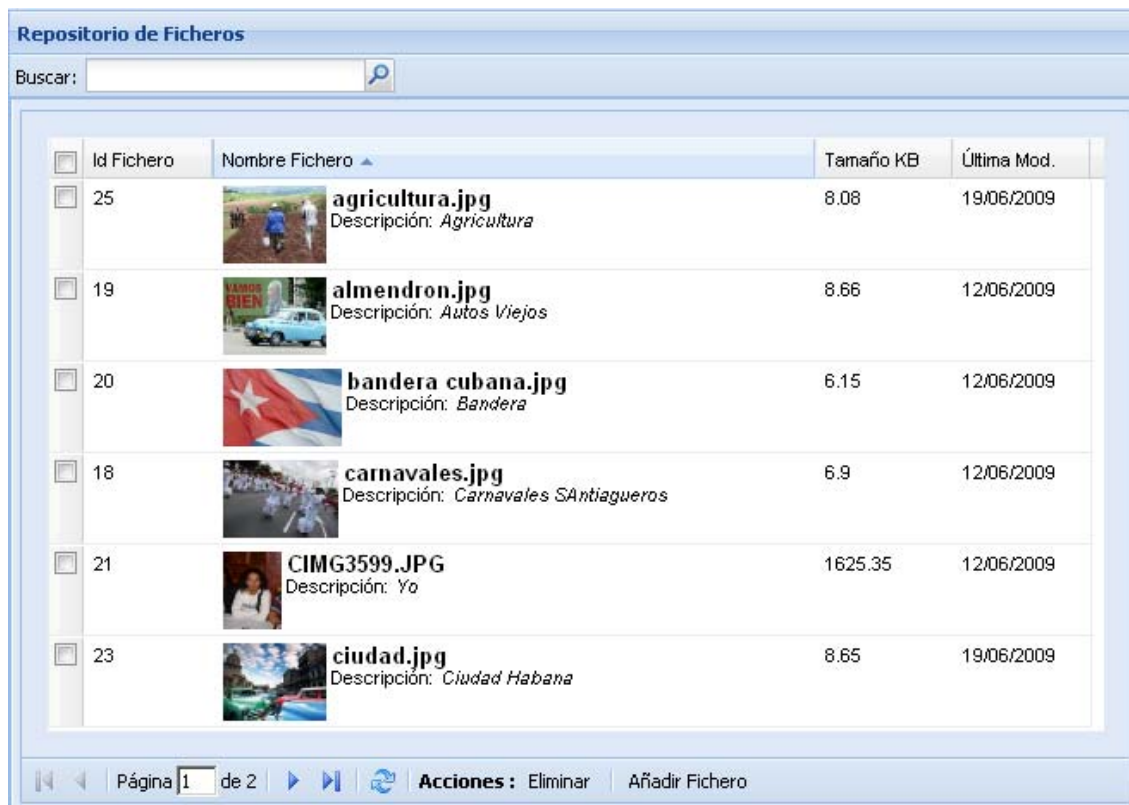
En la figura se muestra la interfaz de usuario para el servicio de publicación. Los materiales en esta interfaz pueden ser expandidos con el objetivo de observar el contenido del resumen y el texto.



1. Menú para selección de las temáticas en las cuales se publicará el material,
2. Menú para selección de los sitios en los cuales se publicará el material.
3. Botón Ejecutar (Efectúa la publicación con las opciones seleccionadas).
4. Barra de Paginación.
5. Elemento Expandido.

3.8 Repositorio de Ficheros.

En la figura se tiene una vista del servicio Repositorio de Ficheros.



1. Buscador.
2. Imagen en miniatura.
3. Barra de paginación
4. Botón Eliminar
5. Botón Añadir Fichero. (Muestra una ventana idéntica a la de la sección 3.6.2)

3.9 Manual del Administrador.

El usuario con el rol de administrador se identifica de la misma forma que los otros usuarios del sistema, pero en este caso accede a la interfaz de administración que es independiente a la del resto de los usuarios. Véase la figura. Aquí se muestra un panel de pestañas expandibles con las opciones del servicio administración.



3.9.1 Sección Usuario.

En esta sección el administrador efectúa las operaciones de añadir un nuevo usuario, otorgarle roles a un usuario, eliminar un usuario; actualizar el nombre, correo, contraseña del usuario, establecer si este está activo o no. Vea en la figura la sección Usuario que consta de un seleccionador de elementos, para seleccionar y visualizar los roles otorgados a un usuario; y una rejilla de edición para añadir, eliminar y actualizar los usuarios del sistema. Ver este último en la figura.

Administración de Usuarios

Roles de Usuario

Disponible

redactor
administrador
corrector
editor
publicador
supervisor
Jefe_Información
director

Seleccionado

Limpiar

↑

↑

→

←

↓

↓

Guardar Cambios

Añadir Usuario

Eliminar Usuario

	Nombre Usuario	Nombre Completo	Correo	Contraseña	Active
1	admin	El Administrador	admin@cip.cu	*****	Yes
2	edi	Ediberto Tomás	edi@cip.cu	*****	Yes
3	efafa	efafa	newfff@cip.cu	*****	Yes

1	admin	El Administrador	admin@cip.cu	<input checked="" type="checkbox"/>
4	luisg	Luis Guzman H		*****	Yes
5	manu	Manuel Vázque		*****	Yes

3.9.2 Sección Roles.

En esta sección el administrador determina a que servicio se tendrá acceso con cada uno de los roles de usuario. Observemos la interfaz de usuario para esta sección en la figura.

Roles de los Usuarios

Servicios Miembros

Disponible

Administración
Redacción
Corrección
Supervisión
Jefe de Información
Control histórico
Sitios públicos
Edición

Seleccionado

Limpiar
 Edición Maestra
 Repositorio
 Publicación

Roles

publicador
redactor
administrador
corrector
publicador
supervisor
Jefe_Información
director

Aplicar

Aplicar la Configuración Seleccionada

3.10 Análisis de los resultados.

A continuación se exponen brevemente alguno de los resultados obtenidos con el desarrollo de la aplicación desde el punto de vista práctico.

3.10.1 Resultados Observables.

Es preciso mostrar una serie de resultados que son palpables y que dejan ver el cumplimiento de los objetivos de investigación trazados.

- Ambiente Visual más amigable.

- Facilidades en la usabilidad y flexibilidad del Sistema.
- Visibilidad con alto por ciento de similitud en todos los navegadores.
- Rendimiento en la comunicación Servidor – Cliente.
- Aumento de la velocidad en la producción de la Noticia por los usuarios.
- Posibilidad de trabajar en varios editores simultáneamente.
- Se evita la pérdida del foco de una operación al cambiar a otra.

Resumiendo de los aspectos ya enumerados es importante acotar que el uso de la biblioteca ExtJS garantiza el por ciento de similitud en cuanto al aspecto visual en las distintas versiones de los navegadores más populares usados en la actualidad. Para que conste se muestra en el Anexo B. una pantalla del sistema en varios de esos navegadores usando una resolución de 800x600 por motivos de escalabilidad, de aquí que la aplicación es usable en disposición de cualquier resolución usada por el usuario.

Conclusiones.

El estudio de las tecnologías estudiadas nos ha permitido seleccionar las herramientas, ambiente de desarrollo y bibliotecas a emplear en el desarrollo e implementación de un sistema capaz de procesar el material de prensa hasta publicarse en uno o varios sitios web online, siguiendo un flujo de trabajo que garantiza un proceso organizado como es el caso del periódico Vanguardia.

El estudio y uso de la biblioteca ExtJS posibilita el diseño de interfaces de usuarios con las características que buscamos para la nueva aplicación web y que de otra manera son muy difíciles de implementar, o sea, sin el uso de una biblioteca que ofrezca esta capacidad. La escritura de especificaciones funcionales ayudó en gran medida en la planificación del diseño de estas interfaces, y sirvió de guía en el proceso de producción del software.

La implantación de un ORM trajo como resultado una metodología de trabajo que le da un mayor nivel al código fuente de la aplicación y por consiguiente lo hace más fácil de entender y escribir a un programador.

Luego con las herramientas usadas se logró implementar el servicio de Redacción, Edición, Repositorio y se continuó en otros de los servicios necesarios en QuipusNew como son el de Edición Maestra, Publicación, y el de Administración que consta de varias secciones. De este modo se han sobrepasado las expectativas en cuanto a la construcción de la aplicación.

Es necesario proponer soluciones que ayuden entonces a concebir la producción de este software en una medida que permita su implementación y despliegue en el menor tiempo posible, o sea, que pueda implantarse en los servidores que se tienen para ese fin, que son provistos por el Ministerio de Comunicaciones de nuestro país. De esta forma comenzaría a usarse en un inicio por el centro de prensa de Vanguardia en la provincia de Villa Clara para la actualización de su periódico en línea en Internet, después por otras instituciones de prensa en el país.

Luego de la experiencia obtenida en la realización de este software proponemos a modo de recomendaciones además, soluciones como:

- Publicar un repositorio de la aplicación en un servidor de Internet, al cual tendrían acceso programadores y diseñadores asociados al desarrollo de este software. El repositorio contendrá código fuente, documentación de la

aplicación y de las tecnologías usadas para la construcción de esta herramienta necesaria para desarrolladores como: IDEs, debuggers, navegadores portables; bibliotecas con su documentación que forman parte del software.

- Esto permitirá que se pueda, en un futuro, organizar una comunidad de desarrolladores que aceleren la producción del software, liberándolo de posibles bugs que puedan afectar su desempeño, además de darle el soporte necesario una vez terminada, así como extender funcionalidades de acuerdo a las exigencias del flujo de trabajo en medios de prensa en general.
- Para lo anterior se pudiera crear un grupo de producción que cuente con la participación de especialistas en la rama de Ciencia de la Computación y estudiantes de Ciencia de la Computación.

Bibliografía

(2008). from <http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>.

Acosta, E. O., N. A. T. Martínez, et al. (2002). SISTEMA QUIPUSNEWS V1.5. PRIMERAS APLICACIONES: PERIÓDICO VANGUARDIA Y QUIPUSNEWSLETTER. Grupo Chasqui. Santa Clara, Universidad Central "Marta Abreu" De Las Villas. **5th**: 152.

Blades, S. C., C. Ramsay, et al. (2008). Learning Ext JS. Birmingham, Packt Publishing Ltd.

Blaisdell, D. (2009). "PHP Database Abstraction Layer." from <http://manifestwebdesign.com/developer-resources/database-abstraction-layer/>.

Brampton, M. (2008). PHP5 CMS Framework Development. Birmingham, Packt Publishing Ltd.

Darie, C., B. Brinzarea, et al. (2006). AJAX and PHP Building Responsive Web Applications. Birmingham, Packt Publishing Ltd.

Ekerete (2008). "PHP framework comparison benchmarks." Retrieved june 16, 2009, 2009, from <http://avnetlabs.com/frameworks/php-framework-comparison-benchmarks.htm>.

Fabien Potencier, F. Z. (2008). Symfony La Guía Definitiva, librosweb.

Flanagan, D. (1997). JavaScript: The Definitive Guide, O'Reilly.

Fowler, M. (2008). "Patrón de diseño Association Table Mapping." from <http://www.martinfowler.com/eaaCatalog/associationTableMapping.html>.

Group, T. P. (2006, 2008). "PHP Data Objects." 2009, from <http://es.php.net/manual/es/intro.pdo.php>.

Myer, T. (2008). Professional CodIgniter. Indianapolis, Indiana, Wiley Publishing, Inc.

Outlet (2009). "Quick Start Guide." from <http://www.outlet-orm.org>.

Pallett, D. (2006). "Ten Different Php Frameworks." Retrieved may 15, 2009, 2009, from <http://www.phpit.net/>.

Ree, L. v. d. (2008). Usable and Adaptable Business Software. Department of Media and Knowledge Engineering Faculty EEMCS. Delft, the Netherlands, Delft University of Technology. **MASTER OF SCIENCE**: 109.

To be able to adapt to the fast changing markets, companies require flexible systems

to support them in their daily business. Instead, most Enterprise Support Software is made for a specific task, not designed to match the needs of a specific company, let alone adapt to changing requirements. To test if this can be changed, the question in this project is: Is it possible to simplify the development of business software, without being restricted in possibilities, to be able to create custom business software that is usable and adaptable? In this project a solution will be developed that will be used in a case study in order to answer this question.

Schumacher, R. and A. Lentz (2008). "Dispelling the Myth." from <http://dev.mysql.com/tech-resources/articles/dispelling-the-myths.html>.

Spolsky, J. (2000) Especificaciones Funcionales sin Esfuerzo.

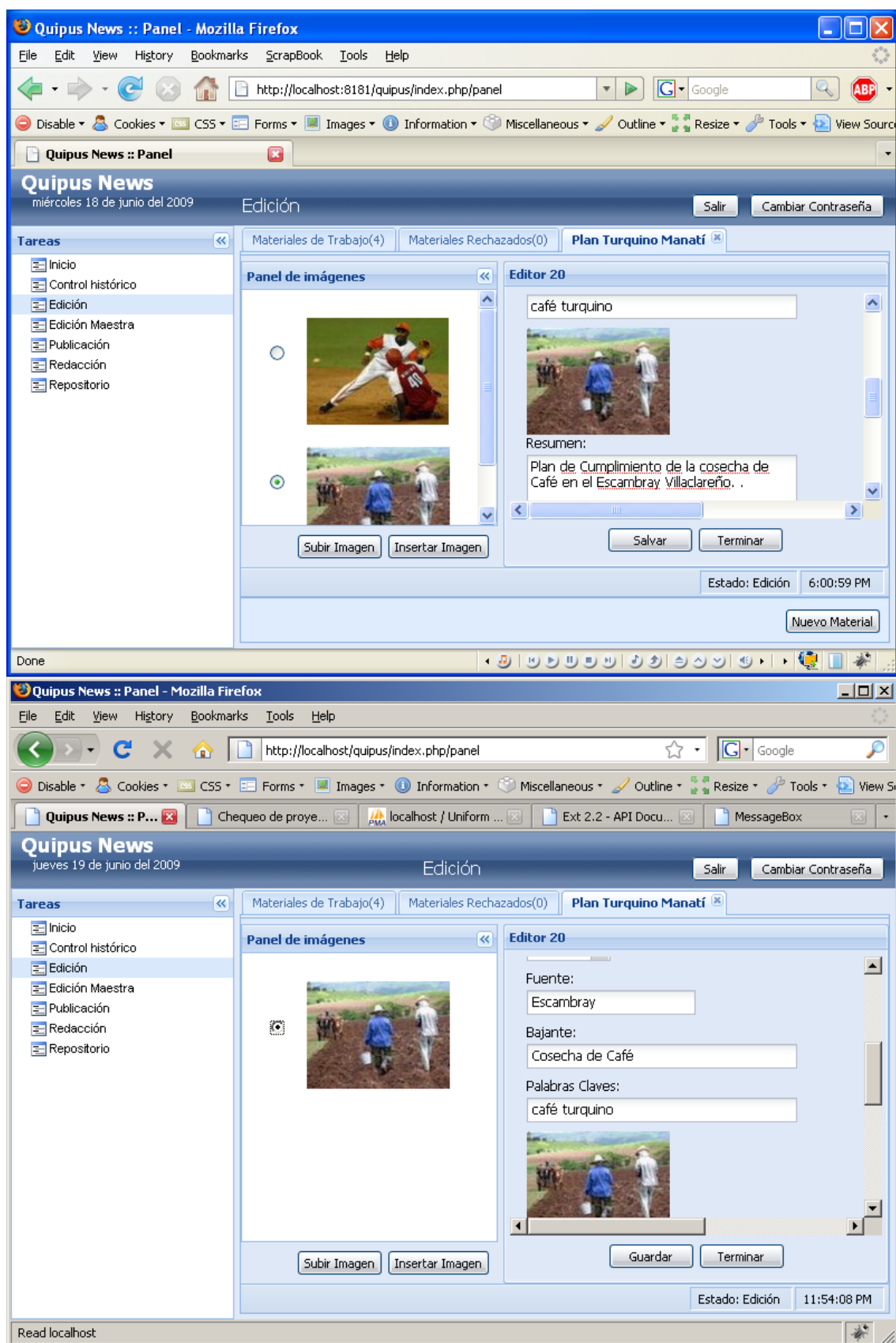
Thomson, W. (2003). php.

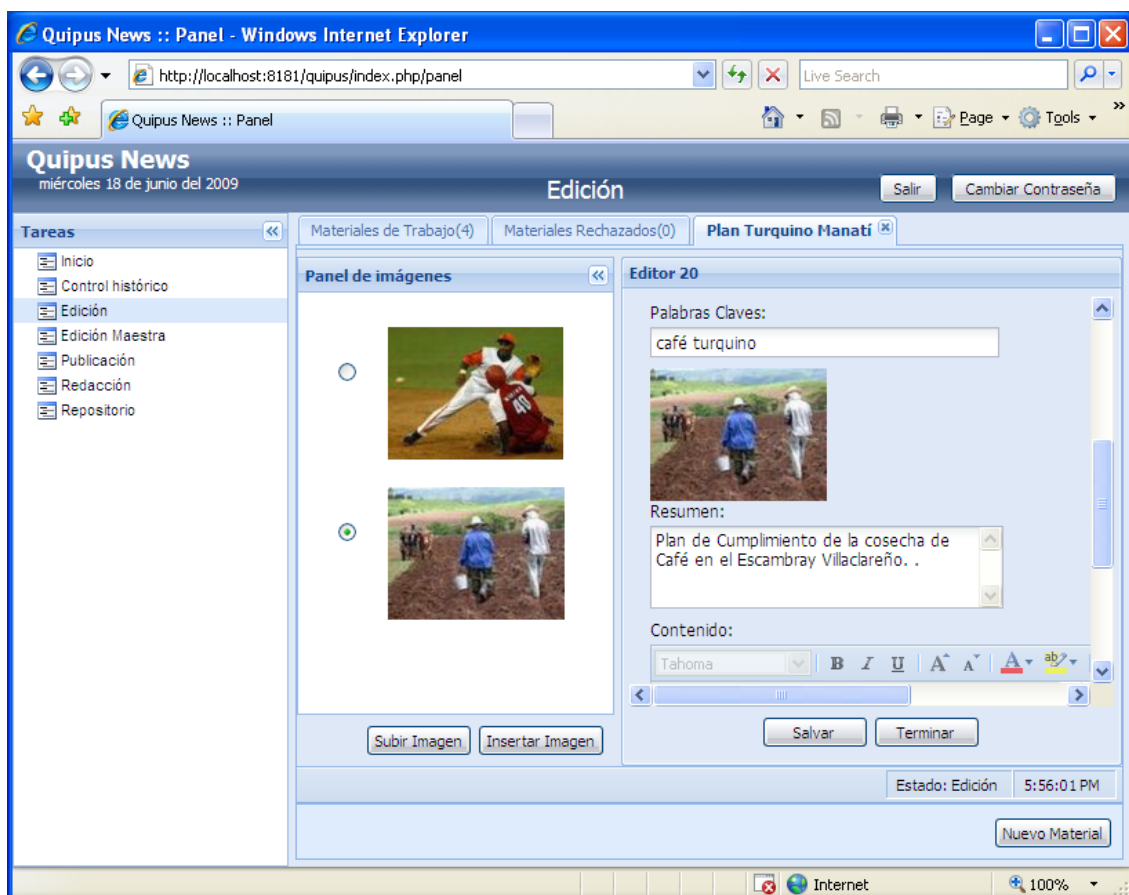
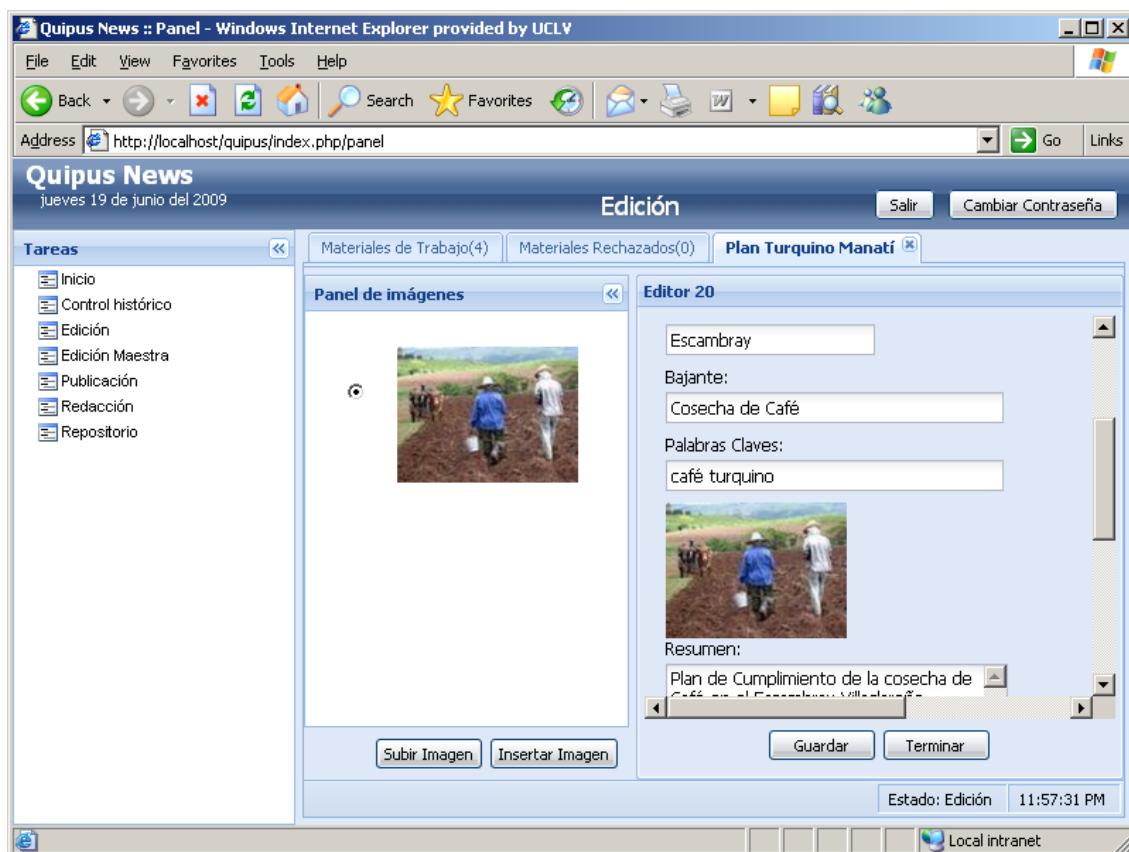
XML-RPC (2008). "XML-RPC Home Page." from <http://www.xmlrpc.com/spec>.

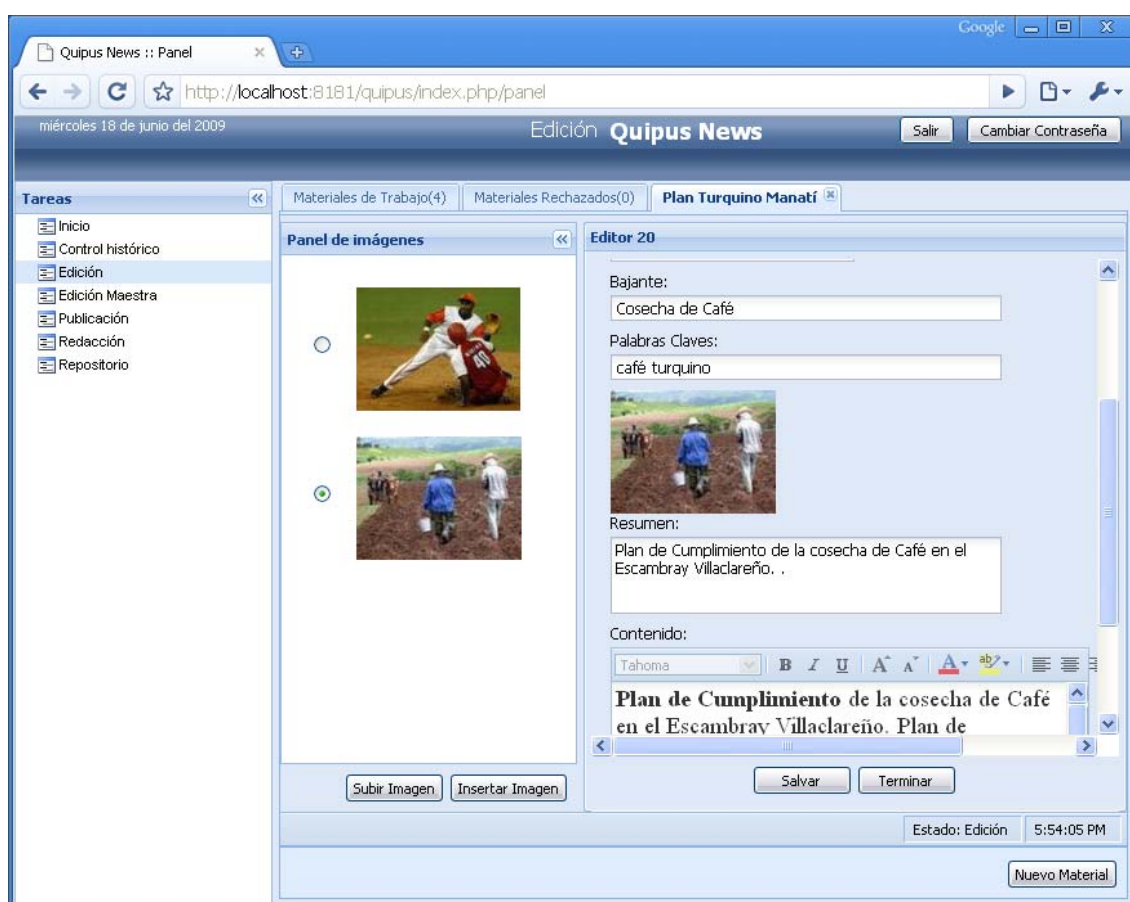
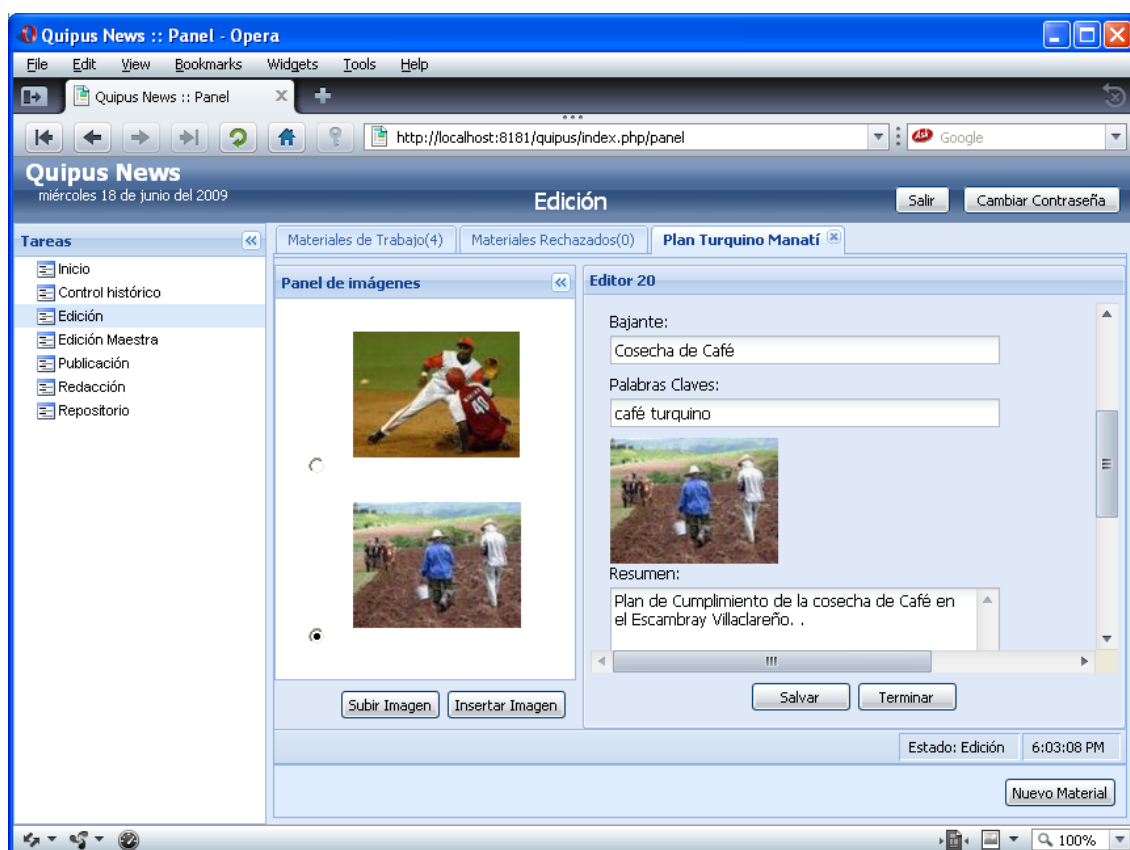
Zend (2009). "Built for Zend Framework: Professional PHP 5 solutions from Zend." Retrieved june 20, 2009, 2009, from <http://www.zend.com/en/community/>.

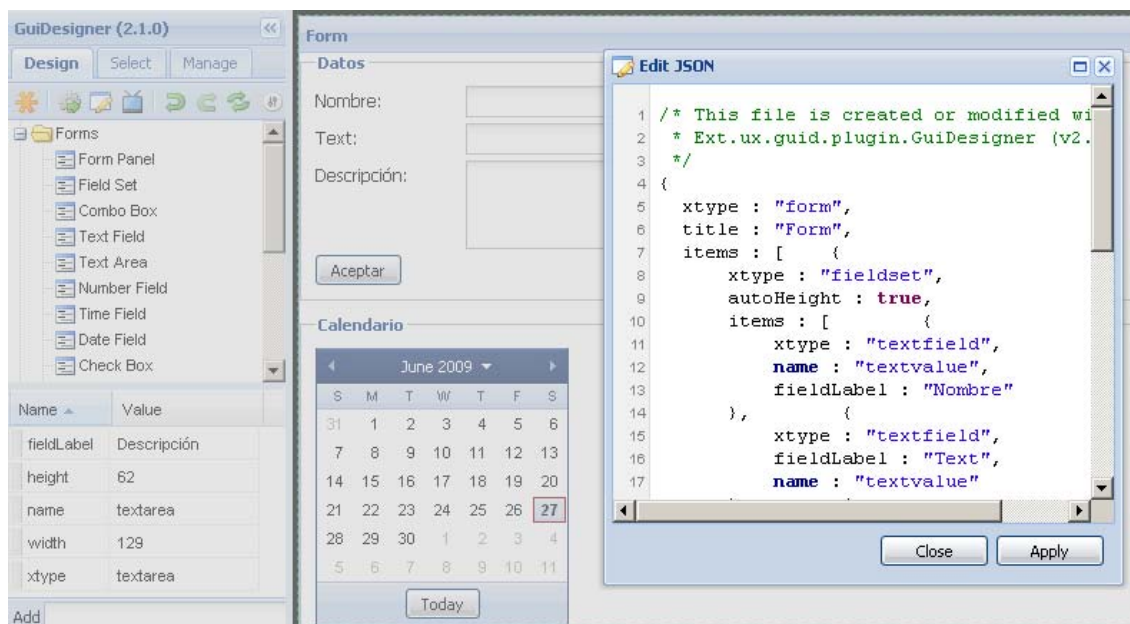
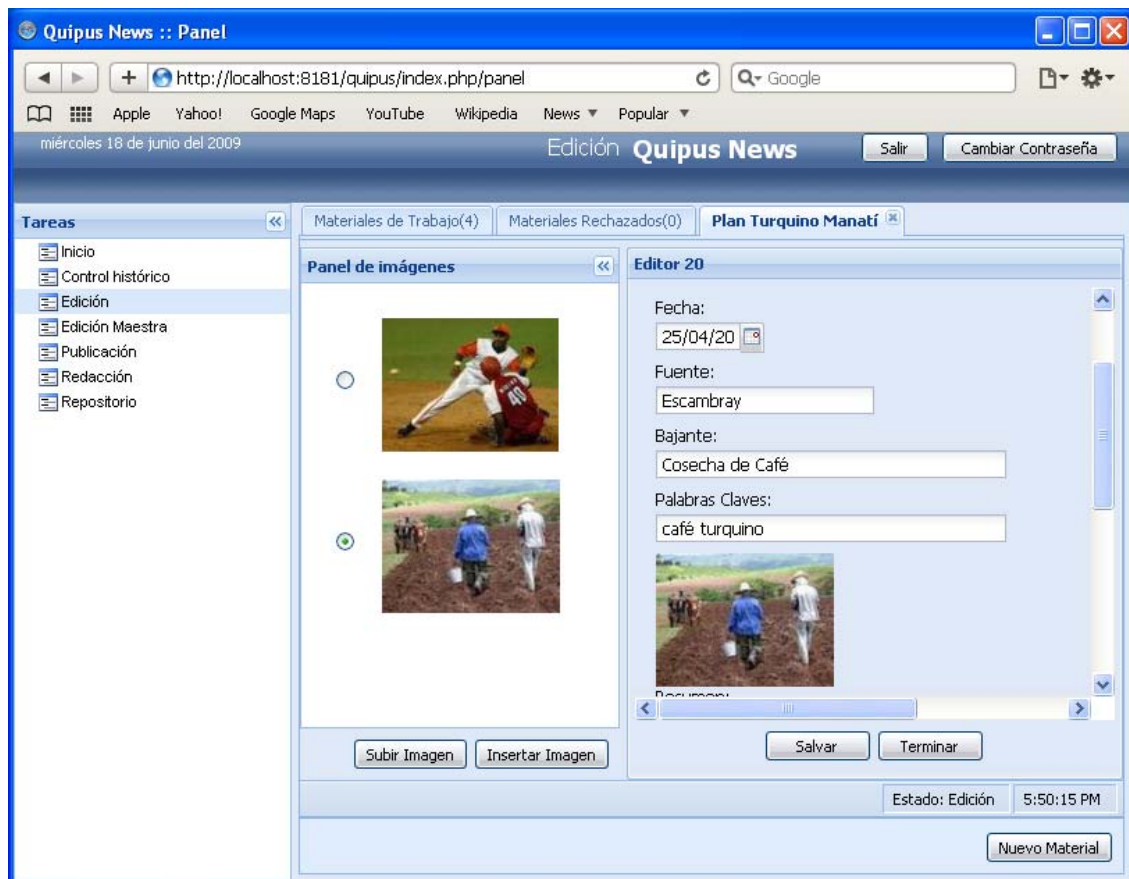
Anexo A Herramientas.

Anexo B Imágenes de la Aplicación.









Anexo C Vistazo detallado para futuro trabajo.