

Universidad de las Ciencias Informáticas

Facultad 1



**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

**Título: Componente para la visualización personalizada de
noticias en el diario Mella**

Autor

Pedro Luis León Fundora

Tutores

Ing. Yunieski Diéguez García
Ing. César González Hernández

La Habana, Junio de 2013
“Año 55 de la Revolución”



Todo es posible. Lo imposible simplemente nos lleva más tiempo.

Dan Brown.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 1 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los _____ días del mes de _____ del año _____.

Pedro Luis León Fundora

Ing. Yunieski Diéguez García

Ing. César González Hernández

AGRADECIMIENTOS

Le agradezco a mis padres Mercedes y Pedro Luis por haber hecho de mí lo que soy hoy, por el apoyo y la confianza brindada, por haber estado a mi lado en todo momento, y sobre todo por ser quienes son.

A mis hermanos que los quiero con el alma.

A mis abuelos Berta y Juan por su amor incondicional.

A mis familiares por estar siempre pendiente de mi vida universitaria.

A mis amistades de toda la vida Leandro, Lesly y Yasser (por cierto este último no me mencionó en sus agradecimientos) bueno, ustedes más que nadie entienden quién soy.

A mis tutores Yuniesky y César por toda la ayuda brindada, a liuris que también me acogió como un tesista más.

A mis profes que siempre me ponían el deo, gracias por ponérmelo de verdad.

Al grupo SIETE, que juntos compartimos muchos tiempos lindos.

En fin, gracias a todos los que de una forma u otra me ayudaron para que el presente trabajo saliera adelante.

Muchas Gracias a todos.

Pedro Luis León

DEDICATORIA

A la memoria de mi bisabuela Caridad.

Pedro Luis León

RESUMEN

En el presente trabajo de diploma se realiza un estudio a los elementos tecnológicos emergentes en el mundo de la Web 2.0. Se efectúan investigaciones relacionadas con los sistemas de prensa digital, tanto nacionales como internacionales, recopilando información referente a la forma de mostrar contenido, así como los componentes en los que se apoyan para lograr un mejor funcionamiento del sistema. Se describe el proceso de desarrollo, las principales herramientas, tecnologías y lenguajes de programación utilizados durante el desarrollo de dicho componente.

Se describen las características y facilidades que brinda el componente para los usuarios. Con el objetivo de lograr un mayor acercamiento al tema, se muestran las principales particularidades del componente, implementación y pruebas realizadas durante el proceso de desarrollo de la aplicación.

Utilizando la plataforma para la publicación de noticias en la Universidad, creada con el Sistema de Gestión de Contenidos WordPress, se induce mediante la propuesta reorganizar el contenido personalizado en la página portada del sistema.

PALABRAS CLAVE

Estadísticas, Prensa digital, Web 2.0, WordPress

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	1
Capítulo 1: REVISIÓN TEÓRICA Y CONCEPTUAL DE LOS SISTEMAS DE PRENSA DIGITAL	5
1.1. Introducción	5
1.2. Prensa escrita y prensa digital	5
1.2.2. Principales medios de prensa internacionales	6
1.2.3. Principales medios de prensa en Cuba.....	8
1.2.4. Principal medio de prensa en la UCI.....	9
1.2.5. Conceptos asociados con las tecnologías a utilizar.	10
1.2.6. Componentes organizadores de contenido en WordPress.....	11
1.3. Tecnologías	12
1.3.1. Proceso de desarrollo de software.....	12
1.3.2. Metodologías de desarrollo de software.....	14
1.3.3. Herramientas para el desarrollo	17
1.4. Conclusiones parciales	19
Capítulo 2: PROPUESTA DE SOLUCIÓN PARA EL COMPONENTE	21
2.1. Introducción:	21
2.2. Modelo de dominio.....	21
2.3. Requerimientos funcionales.....	22
2.4. Requerimientos no funcionales	23
2.5. Historias de usuario	25
2.6. Diseño de la Base de Datos.....	27
2.7. Arquitectura	29
2.7.1. Patrones GRASP	30
2.7.2. Patrones GOF.....	31
2.8. Concepto y clasificación de la audiencia.....	32
2.9. Conclusiones parciales	33
Capítulo 3: IMPLEMENTACIÓN Y PRUEBAS	34
3.1. Introducción	34
3.2. Estándares de codificación	34
3.3. Análisis y descripción de las nuevas funcionalidades y el uso de las ya existentes	36

3.4. Diagrama de despliegue	38
3.5. Pruebas	41
3.5.1. Pruebas funcionales	41
3.5.2. Pruebas de carga y estrés	43
3.6. Conclusiones parciales	45
CONCLUSIONES GENERALES.....	46
RECOMENDACIONES	47
BIBLIOGRAFÍA REFERENCIADA	48
BIBLIOGRAFÍA CONSULTADA.....	50
ANEXOS.....	54

ÍNDICE DE FIGURAS

Ilustración 1: Diagrama del modelo de dominio.....	21
Ilustración 2: Modelo de datos del componente	27
Ilustración 3: Arquitectura de la plataforma	29
Ilustración 4: Comillas simples y dobles	34
Ilustración 5: Estilo de llaves	35
Ilustración 6: Estilo de llaves para ciclos	35
Ilustración 7: Uso de espacios	35
Ilustración 8: Etiquetas y atributos.....	36
Ilustración 9: Diagrama de despliegue	38

ÍNDICE DE TABLAS

Tabla 1: Requisitos funcionales	22
Tabla 2: Artículo.....	27
Tabla 3: Categoría	28
Tabla 4: Registro.....	28
Tabla 5: Usuario.....	28
Tabla 6: Histórico	28
Tabla 7: Historia de usuario “Obtener visitas de los usuarios sobre una noticia”	25
Tabla 8: Historia de usuario “Extraer categoría de las noticias”	26
Tabla 9: Funciones de WordPress	37
Tabla 10: Funciones implementadas.....	37
Tabla 11: Caso de prueba “Obtener visitas”	41
Tabla 12: Caso de prueba “Extraer categoría”	42
Tabla 13: Iteraciones de pruebas funcionales	43
Tabla 14: Sistema sin el componente	44
Tabla 15: Sistema con el componente	44

INTRODUCCIÓN

La Web, se ha ido desarrollando a través de los años. Esto en gran parte se debe, al alto consumo de Internet por los usuarios, quienes principalmente lo utilizan para la búsqueda de información. Debido a esto, la competencia por la audiencia desatada por los líderes de la Web se ha incrementado, provocando una inevitable evolución hacia la Web 2.0. Es por ello, que en aras de lograr una mayor aceptación en la comunidad, se hace necesario mejorar la usabilidad de nuevas aplicaciones.

Las ventajas que brinda la Web 2.0, juegan un rol importante en el desarrollo de sitios. Esto influye en una mayor interacción de los usuarios con la información publicada. De esta manera, los consumidores son considerados creadores de contenidos de una comunidad virtual, a diferencia de antiguos sitios web donde se limitan a la observación pasiva de la información creada para ellos. Uno de los sectores que aprovecha estos espacios, son los medios de comunicación. Para ello la prensa ha digitalizado sus contenidos poniéndolos al alcance de todos, dando lugar a la prensa digital.

En la Universidad de las Ciencias Informáticas (UCI) existe un gran movimiento periodístico, denominado “Código y Letra”. Está conformado en su mayoría por periodistas aficionados que publican artículos en el órgano oficial de la UCI, diario Mella. Este sistema desde su origen, se le catalogó como medio de prensa digital, a diferencia de otros periódicos como el Granma o Juventud Rebelde. El diario Mella fue desarrollado principalmente, con el propósito de mantener informada a la comunidad universitaria en el ámbito noticioso local, aunque también refleja la actualidad nacional e internacional. Sus artículos se clasifican por categorías, los cuales son generados previamente o en el momento de crear las noticias que van a ser publicadas. Esto permite ayudar a los usuarios a realizar una mejor navegación y lograr una mejor organización en el sistema.

Actualmente el sistema muestra en portada las últimas noticias publicadas sin importar la categoría a la que pertenecen. Si un usuario accede con la intención de leer las noticias correspondientes a una categoría, debe acceder primero a la portada y luego seleccionar la categoría de su interés. En caso que desee leer en más de una categoría, deberá acceder a cada una por separado. Esta manera de organizar la información, prolonga el tiempo en encontrar los contenidos de su interés, generando insatisfacción en el uso del diario Mella. Tampoco provee a los administradores de un mecanismo para conocer el comportamiento y las

preferencias de los usuarios, y con ello realizar estudios de tendencias en la comunidad universitaria, elementos de gran utilidad para intensificar el trabajo periodístico en función de lograr calidad y satisfacción.

Teniendo en cuenta la situación problemática planteada anteriormente se formula el siguiente **problema a resolver**:

¿Cómo identificar y mostrar las preferencias noticiosas del usuario autenticado en el diario Mella de la UCI?

Se define para la presente investigación como **objeto de estudio**, los sistemas de prensa digital, enmarcada en el **campo de acción** la representación de información en el diario Mella de la UCI.

Se define como **objetivo general del trabajo**: desarrollar un componente de WordPress, aplicando las metodologías XP y SCRUM, para identificar y mostrar las preferencias noticiosas del usuario en el diario Mella de la UCI.

Para lograr el cumplimiento de los objetivos se proponen las siguientes **tareas de investigación**:

- Investigación de la representación de información en los sistemas de prensa digital y su retroalimentación de los usuarios.
- Identificación de las posibles herramientas a utilizar en el desarrollo del componente para la visualización de noticias personalizadas en el periódico digital universitario, diario Mella.
- Definición de las funcionalidades a desarrollar en cada iteración.
- Implementación del componente para la gestión del proceso de consumo de noticias personalizadas en el periódico digital universitario, diario Mella.
- Integración del componente para la visualización de noticias personalizadas con el periódico digital universitario, diario Mella.
- Realización de pruebas funcionales al componente para la visualización de noticias personalizadas de acuerdo a las preferencias de los usuarios en el periódico digital universitario, diario Mella.

- Realización de pruebas de integración al componente para la visualización de noticias personalizadas de acuerdo a las preferencias de los usuarios en el periódico digital universitario, diario Mella.

Para la realización del trabajo se aplicarán los siguientes **métodos de investigación**:

Histórico-lógico: se usa con el objetivo de estudiar el origen y evolución de todo lo referente a los procesos de visualización de noticias en el ámbito nacional e internacional, para de esta forma conocer más del tema, sus características e importancia. Este método ayuda a dar cumplimiento al estudio del estado del arte de los procesos de visualización de noticias.

Método de la modelación: partiendo de todo lo investigado y aprendido se realizan los modelos correspondientes al ciclo de vida del *software*, que ayudan a dar cumplimiento a las tareas de diseño de los procesos involucrados en la solución.

Método de observación: es utilizado como apoyo para la obtención de información sobre el tema a tratar, y ayuda a dar solución a prácticamente todas las tareas. Haciendo uso de él se puede ver cómo funciona actualmente la visualización de noticias. Además permite apreciar que es lo que realmente necesitan los clientes del sistema de acuerdo a las nuevas concepciones.

Del trabajo se esperan los siguientes **aportes prácticos**:

- Proporcionará una forma más sencilla y directa de ver las noticias publicadas por el periódico digital de acuerdo a las preferencias de los usuarios.
- Facilitará la comunicación y retro-alimentación entre los actores del proceso de publicación y consumo de noticias.
- Favorecerá a los periodistas al conocer el comportamiento de los lectores, teniendo en cuenta los intereses de los mismos respecto al tipo de información publicada.

Estructuración del documento:

Introducción: se explica de forma general lo que se va a tratar en el presente documento, el diseño teórico metodológico. Además se presentan los métodos de investigación y la estructura del documento.

Capítulo 1. Revisión teórica y conceptual de los sistemas de prensa digital: se realiza un estudio de sistemas y componentes homólogos. Se describe el proceso de desarrollo del *software*, así como las tendencias, técnicas, metodologías y *software* usados en la propuesta de solución y se definen las tecnologías y las herramientas a utilizar para el desarrollo del componente.

Capítulo 2. Propuesta de solución para el componente: se especifican los requisitos funcionales y no funcionales con su descripción correspondiente y se muestran las técnicas utilizadas para su obtención. Se describen los diagramas de despliegue y de componentes concernientes para la implementación de la solución.

Capítulo 3. Implementación y pruebas: en este capítulo se realiza la implementación y las pruebas de integración para validar el correcto funcionamiento del componente.

Además el documento contiene las Conclusiones, Bibliografía, Bibliografía referenciada y los Anexos.

CAPÍTULO 1: REVISIÓN TEÓRICA Y CONCEPTUAL DE LOS SISTEMAS DE PRENSA DIGITAL

1.1. Introducción

El tema de la prensa digital, es muy tratado en la literatura por diversos autores a partir de la demanda de información noticiosa por parte de los usuarios. Es importante conocer los conceptos asociados, para un mayor entendimiento sobre el tema, dentro de los que están: prensa y prensa digital, así como los principales medios que existen.

La progresiva popularización de Internet y finalmente la consolidación de la denominada Web 2.0, caracterizada por su eminente orientación hacia el usuario y sus actividades sociales en la Web, ha propiciado la presencia de la mayoría de los medios de comunicación en este entorno. Especialmente relevante ha sido el caso de la prensa escrita, que ha tenido que adaptarse progresivamente a este nuevo escenario, para poder satisfacer las demandas de información, de unos usuarios más familiarizados cada día, con el uso de las nuevas tecnologías (GONZÁLEZ, 2001).

El desarrollo de la prensa digital tiene mucha relación con los Sistemas de Gestión de Contenidos (CMS), pues muchas de estas aplicaciones web se desarrollan fundamentalmente en estas tecnologías entre las que se destacan: WordPress y Drupal. Estos brindan una mayor facilidad de manipulación de contenido y su administración es igual de sencilla. También mejora considerablemente la usabilidad del sistema mediante componentes agregados, que permiten adicionar funcionalidades nuevas sin muchas complicaciones.

1.2. Prensa escrita y prensa digital

La prensa escrita es el conjunto de publicaciones impresas en papel que tienen como objetivo informar a la sociedad. Dentro de la prensa escrita podemos encontrar publicaciones diarias dedicadas a la información de la actualidad.

Los periódicos de información diaria conforman la variedad de prensa escrita de mayor difusión. Se publican a diario y ofrecen información general sobre temas como política, sociedad, deportes, cultura, etcétera (DEPARTAMENTO DE EDUCACIÓN, 2013).

Por su parte la prensa digital según el autor Ramón Salaverría Aliaga “es la especialidad del periodismo que emplea el ciberespacio para investigar, producir y, sobre todo, difundir contenidos periodísticos” (RIVERA, 2011). Esta nueva área del periodismo tiene que ver directamente con el desarrollo de las nuevas tecnologías en los últimos años. Además con este fenómeno surge una nueva versión del periodista: el digital, es decir, aquel que elabora la información básicamente utilizando la Web.

Estos diarios digitales tienen como objetivo, publicar las principales noticias del día. Ofrecen servicios de autenticación de usuario para brindarles más privilegios a los consumidores. Un ejemplo de esto es la interacción con las noticias publicadas, donde pueden comentar u opinar. Entre estos servicios además se encuentra el completamiento de cuestionarios, facilitándole a los periodistas recaudar información sobre las preferencias e intereses de los usuarios, identificando las noticias que más audiencia atrae.

1.2.2. Principales medios de prensa internacionales

Una de las principales aplicaciones destinadas a la prensa digital es **El Mundo**¹. Este periódico es de origen español y su contenido está a cargo de la Unidad Editorial Información General S.L.U (SOCIEDAD ANÓNIMA INFORMACION Y CONTROL DE PUBLICACIONES, 2013). De sus características tecnológicas poco se sabe, pues no brinda información alguna de este tema. Por lo que, se desconoce la existencia de algún medio tecnológico destinado a la reubicación de publicaciones, teniendo presente los gustos de sus usuarios.

Este diario no reorganiza su contenido en dependencia de las preferencias de los usuarios autenticados. Después de un período de tiempo accediendo al mismo, no se apreciaron cambios claros en su contenido. Por lo que carece de alguna posible solución tecnológica para la problemática planteada.

¹ Aplicación destinada a la prensa digital. Actualmente es uno de los periódicos españoles más visitados en el mundo. <http://www.elmundo.es/>

CNN² es también otro de los sistemas monitoreados, el cual tiene como propietario la empresa Turner Broadcasting System Incorporated, oriunda de Norteamérica. Este diario presenta sucursales en diferentes países del mundo, entre los más conocidos de habla hispana, se encuentra: España, Méjico y Chile (TURNER BROADCASTING SYSTEM INCORPORATED, 2013). A partir del análisis del código de fuente de este diario, se identificó que su desarrollo se realizó con el CMS WordPress, pero se desconoce la versión y el listado de componentes que utiliza. El entorno se muestra más agradable al consumidor, pero al igual que el periódico mencionado anteriormente, ignora los intereses de sus usuarios.

Por su parte **El País**³, con cerca de 13 millones de usuarios únicos al mes es el líder mundial de la prensa online en español. Cerca de 5 millones del total de lectores online acceden desde fuera de España, lo que corrobora su definición como "el periódico global en castellano" (EDICIONES EL PAÍS, 2013).

En su estructura de contenido no refleja las tecnologías utilizadas para su desarrollo. A partir de los servicios que ofrece, tampoco es de interés las preferencias de su audiencia. Esto se debe a que sus artículos son publicados en un orden cronológico y no se tiene en cuenta la categoría a la que pertenecen, generando así desorganización en los contenidos mostrados.

Por las características que se plantean, se puede deducir que los intereses de los usuarios, no son una prioridad para las organizaciones encargadas de patrocinar estos sistemas. Actualmente, esta es una de las causas por la cual hoy en día muchos sitios web pierden audiencia.

² *Cable News Network* (CNN) por sus siglas en inglés, es otra de las aplicaciones web que se dedican a la prensa digital. <http://cnnespanol.cnn.com/>

³ Es uno de los sistemas de prensa más visitados en el mundo. <http://elpais.com/>

1.2.3. Principales medios de prensa en Cuba

En Cuba también existen sistemas dedicados a la prensa digital, **Cubadebate**⁴ es uno de ellos. El sitio también conocido como *blog*⁵ está desarrollado sobre la herramienta WordPress y utiliza varios componentes para completar sus prestaciones además de su funcionamiento. Todas estas aplicaciones adheribles están desarrolladas bajo la licencia GNU/GPL⁶ y realizaron modificaciones a todas para garantizar su mejor integración, traducción al español y en sentido general ajustar sus funciones a los intereses de Cubadebate (CÍRCULO DE PERIODISTAS CUBANOS, 2009).

Cubadebate cuenta con un número bien nutrido de componentes que reorganizan contenidos. En el estudio se detectaron varios componentes homólogos que se muestran a continuación:

Category Order (Versión 1.0.2), posibilita el reordenamiento de las categorías de la manera deseada, muy útil si se quieren manejar prioridades. No requiere un control especial, sólo instalar y volver a ordenar (CÍRCULO DE PERIODISTAS CUBANOS, 2009).

Organize Series (Versión 2.0.8) Organiza los artículos en series. Esto ayuda a que sea más fácil para los lectores de un *blog*, descubrir todas las series que se han escrito y también para encontrarlos fácilmente, puesto que forman parte de la misma serie (CÍRCULO DE PERIODISTAS CUBANOS, 2009).

Estos componentes reorganizan los contenidos a partir de la intencionalidad de los administradores y no de la preferencia de los usuarios.

⁴ *Blog* cubano destinado a la lucha contra el terrorismo mediático. <http://www.cubadebate.cu/>

⁵ *Blog*: aplicación web periódicamente actualizada que tiene como objetivo principal mantener a una comunidad web actualizada, permitiéndole a los consumidores una mayor interacción sobre el contenido publicado.

⁶ GPL significa *General Public License* (Licencia Pública General). La más difundida de tales licencias es la Licencia Pública General de GNU, o GPL de GNU, para abreviar. Puede reducirse aún más, a GPL, cuando se sobreentienda que nos estamos refiriendo a la «GPL de GNU».

El diario **Juventud Rebelde**⁷ es otro de los sistemas que residen en Cuba. En la actualidad es uno de los más modernos en cuanto a tecnología. Estudios a su código de fuente, revelan que fue desarrollado con el CMS WordPress. El mismo no ofrece en su contenido datos sobre la tecnología utilizada, a diferencia de Cubadebate. Este sistema no reorganiza información, compartiendo las mismas características que sus homólogos internacionales.

Juventud Rebelde no refleja claramente el uso de componentes para la reorganización de contenidos. En él además, no se evidencia la existencia de algún mecanismo para identificar las preferencias de los usuarios. Es por ello que se descartan para el desarrollo de esta investigación. Se puede destacar, que Cubadebate sí ofrece información sobre su tecnología y servicios, las cuales tributan conocimientos valiosos, debido a los estudios realizados. De esta manera se le podrá dar cumplimiento al objetivo que persigue la investigación, de una forma más eficiente.

1.2.4. Principal medio de prensa en la UCI

En la UCI existen disimiles espacios informativos pero no se catalogan como sistema de prensa digital. Muchos de ellos publican noticias de diversos temas, pero su función principal no es la de los diarios tradicionales. Solamente se puede denominar como sistema dedicado a la prensa el diario Mella. Este es un sistema desarrollado con el CMS WordPress en su versión 3.1 y utiliza un grupo de componentes que mejoran la usabilidad de la propia aplicación.

Actualmente el diario Mella, no es capaz de identificar las noticias que prefiere su comunidad. Por lo que se hace difícil brindar un servicio de noticias más eficiente. Tampoco provee a los periodistas del sistema, información de lo que realmente es noticia para los internautas, para así mejorar el trabajo periodístico. En aras de mejorar los servicios del diario Mella, se propone desarrollar un componente que le permita al sistema conocer las noticias preferidas de los usuarios, para así lograr una mejor presentación del contenido.

Para la realización del componente, se hace necesario la selección de un grupo de tecnologías, que van a guiar el desarrollo de la solución. En la orientación del progreso de ejecución de la

⁷ Se identifica como uno de los principales diarios de Cuba, tanto impreso como digital. <http://www.juventudrebelde.cu/>

propuesta se necesita de una metodología. Una herramienta CASE⁸ para registrar el ciclo de vida completo del *software* y un IED⁹ de desarrollo para la programación. Así como de técnicas para la validación del sistema a través de pruebas pertinentes que verifiquen su correcto funcionamiento.

1.2.5. Conceptos asociados con las tecnologías a utilizar.

¿Qué es un componente?

Un componente es una aplicación que se adhiere a un sistema para permitirle realizar funciones específicas que antes no hacía. Estos *plugin*¹⁰ como también se les conocen, son adicionados y son ejecutados por la aplicación principal. Se puede decir también que interactúan mediante una API¹¹ (WORDPRESS ORGANIZATION, 2012).

Tendencias y características de los componentes

El principal objetivo de estas aplicaciones es auxiliar a los sitios web, ya sea para mejorar sus funcionalidades u optimizar su rendimiento. Son dispositivos que permiten a sistemas mayores extender sus capacidades normales o aportar una función, generalmente muy específica, de manera que no se afecten las funciones ya existentes ni se complique el desarrollo del programa principal.

Estos cuentan con una amplia gama de características: son aplicaciones que se instalan y se desinstalan, permitiendo de esta manera incorporar un conjunto específico de características o servicios a una web. Esto permite que los administradores, dispongan del mismo en el momento que lo necesiten y si su presencia ya no es necesaria se puede desvincular del

⁸ CASE (*Computer Aided Software Engineering*), sus siglas traducidas al español significan Ingeniería de *software* Asistida por Computadora.

⁹ Entorno de Desarrollo Integrado, traducido del inglés *Integrated Development Environment* (IDE).

¹⁰ *Plug-in* del inglés enchufable o inserción.

¹¹ Interfaz de programación de aplicaciones (IPA) o API (del inglés *Application Programming Interface*) es el conjunto de funciones y procedimientos en la programación orientada a objetos, que ofrece cierta biblioteca para ser utilizado por otro *software* como una capa de abstracción.

sistema sin que cause efectos secundarios. Además pueden ser modificados para así lograr un rendimiento óptimo del mismo (WORDPRESS ORGANIZATION, 2012).

Los sistemas de gestión de contenidos

Un Sistema de Gestión de Contenidos (en inglés *Content Management System*, abreviado CMS) es un programa que permite crear una estructura de soporte *framework*¹² para la creación y administración de contenidos, principalmente en páginas web, por parte de los participantes.

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite maniobrar de manera independiente el contenido y el diseño. Así, es posible darle en cualquier momento un diseño distinto al sistema sin tener que darle formato al contenido nuevamente, además de permitir la fácil y controlada publicación de varios editores (CASAS CERVERO, 2011).

Uno de los CMS más utilizados es WordPress, una plataforma de un alto nivel de desarrollo. Posee publicación personal orientada a la estética, los estándares web y la usabilidad. WordPress es libre y al mismo tiempo gratuito.

1.2.6. Componentes organizadores de contenido en WordPress

A partir del análisis de diversas fuentes, se encontraron componentes que muestran distintas formas de manejar el orden de los *post*¹³ en WordPress, a conveniencia de los desarrolladores. Uno de ellos es el **Query Post**. Esta función se usa para sacar un listado de las noticias publicadas en el sistema, con los cuales posteriormente se trabaja para ser mostrados de forma personalizada a los usuarios (WORDPRESS ORGANIZATION, 2012). La función que realiza este componente, sirve como base para llegar al desarrollo de la solución que propone el presente trabajo de diploma.

¹² La palabra inglesa *framework* define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular.

¹³ Un *post* se traduce en español como artículo, aunque se suele utilizar con la terminología inglesa o el nombre de 'entradas' en las publicaciones hechas en *blogs*.

Otro componente estudiado es el **Order Post**. Este se utiliza para organizar las entradas manualmente de cada categoría como uno desee y no de forma cronológica como lo hace normalmente WordPress. Muy útil para proyectos de libros, revistas y portafolios electrónicos en donde la estructura cronológica no es la más adecuada (WORDPRESS ORGANIZATION, 2012). Al igual que el anterior presenta características similares de funcionamiento.

A partir de las características mostradas en los homólogos anteriores, se puede afirmar que no resultan factibles para dar solución a la problemática planteada. Pues se necesita un componente que realice el servicio de forma independiente, sin la interacción de un intermediario. Con la realización del siguiente trabajo de diploma, se obtendría un resultado óptimo y genérico, que reorganice los artículos de la portada sin la interacción del administrador, haciendo posible el aprendizaje del mismo sobre los usuarios. De esta manera, se logra generar una nueva página inicial con las preferencias noticiosas de cada internauta.

1.3. Tecnologías

Con el objetivo de que el resultado del desarrollo sea factible, se realiza el estudio y la selección de las tecnologías a utilizar para la implementación del componente.

1.3.1. Proceso de desarrollo de software

Las grandes industrias de *software* continúan enfrentando problemas, pues aún resulta significativo el número de proyectos que no culminan con la calidad deseada. Las organizaciones exitosas, han aprendido que aún los mejores profesionales necesitan de un ambiente organizado, disciplinado y estructurado. Los centros que carecen de dicho ambiente, arriesgan a su personal a interminables horas de trabajo para solucionar problemas.

La UCI no exenta a los problemas presentes en la industria de *software*, se lleva a cabo un programa de mejora de *software*, en el cual se disemina el modelo CMMI. Dicho programa busca aumentar la madurez y capacidad de la organización en la producción de *software*.

CMMI

CMMI es un modelo de referencia para el crecimiento de capacidades y madurez, que se enfoca tanto en procesos de Administración como de Ingeniería de Sistemas y *Software*. Fue

creado por el SEI (*Software Engineer Institute*) y está compuesto por 22 Áreas de proceso las que se distribuyen en 5 niveles según las representaciones del modelo. El método usado por los creadores del modelo, para certificar el ascenso de una organización por los diferentes niveles es el conocido como SCAMPI2 (BETH CHRISSIS *et al.*, 2009).

El programa de mejora que se desarrolla actualmente en la UCI, tiene como objetivo la implementación de los procesos necesarios para cubrir las 7 áreas de proceso del nivel 2 del modelo. Las áreas que incluyen son: Planeación del Proyecto (PP), Monitoreo y Control de Proyecto (PMC), Administración de Requisito (REQM), Aseguramiento de la Calidad para Procesos y Productos (PPQA), Medición y Análisis (MA), Administración de la Configuración (CM) y Administración de Acuerdos con Proveedores (SAM). Con ella se garantiza la reducción de varios de los problemas que acucian la producción de *software* en la Universidad.

En el centro (Centro de Informatización Universitaria de la UCI) CENIA se llevan a cabo procesos de mejora dirigidos a productos, los cuales están vinculados a las características del mismo. La evolución del *software* está guiada por el "Proceso de desarrollo con enfoque ágil orientado al segundo nivel de CMMI". Dentro de los elementos que forman parte de la solución se define el uso de herramientas libres, compatibles con el desarrollo tecnológico de la Universidad (MENA, 2012). El presente trabajo de diploma utiliza este proceso de desarrollo.

El nivel 2 de CMMI se caracteriza como un proceso gestionado y define siete áreas:

- Gestión de Requisitos (*Requirements Management* REQM): El propósito de esta área de proceso, es gestionar los requisitos de los productos del proyecto y componentes del producto e identificar inconsistencias entre dichos requisitos y la planificación del proyecto y los productos de trabajo.
- Planificación de Proyectos (*Project Planning* PP): el objeto de esta área de proceso es el de establecer y mantener la planificación que define las actividades del proyecto.
- Seguimiento y Control del Proyecto (*Project Monitoring and Control* PMC): su propósito es comprender el progreso del proyecto para tomar acciones correctivas adecuadas cuando el desempeño del proyecto se desvíe del plan.
- Gestión de Acuerdos con Proveedores (*Supplier Agreement Management* SAM): su propósito es administrar la adquisición de productos de los proveedores para los cuales existe un acuerdo formal.

- Medición y Análisis (*Measurement and Analysis* MA): su propósito es desarrollar y sostener la capacidad de medición que se usa para apoyar las necesidades de información de la administración.
- Aseguramiento de Calidad de Procesos y Productos (*Process and Product Quality Assurance* PPQA): su propósito es proporcionar a la administración y al personal, una visión objetiva de los procesos y productos de trabajo asociados.
- Gestión de Configuración (*Configuration Management* CM): su propósito es establecer y mantener la integridad de los productos de trabajo usando identificación, control, contabilidad de estado y auditorías de configuración.

1.3.2. Metodologías de desarrollo de software

El desarrollo y la evolución constante experimentada por los procesos de Ingeniería de *Software*, ha traído consigo la realización de varias tareas en este campo, como son: análisis de requisitos, especificación, diseño y arquitectura, programación, prueba, documentación y mantenimiento.

El proceso de desarrollo del *software*, define el conjunto de actividades precisas para convertir los requisitos de los usuarios en el conjunto seguro y resistente de artefactos que componen un producto de *software*.

Las metodologías se definen por pasos a seguir para el cumplimiento de un objetivo. El objetivo dentro del desarrollo del *software* es realizar un producto de alta calidad que cumpla con los requerimientos del cliente. Por lo que se hace de vital importancia definir metodologías para guiar el proceso de desarrollo de un producto de *software*.

Entre las metodologías de desarrollo de *software* que utiliza en CENIA se encuentran XP y SCRUM, las cuales se utilizan en el presente trabajo de diploma:

XP

Basándose en la simplicidad, la comunicación y la reutilización del código, surge la Programación Extrema (*eXtreme Programming XP*), siendo así una metodología ágil de desarrollo de *software*. Como requisito para alcanzar el éxito del proyecto se tiene al usuario final como parte del equipo. Es una metodología con reconocido éxito y se utiliza en proyectos con entregas a cortos plazos.

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de *software*, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios (CANÓS *et al.*, 2003).

Las características fundamentales de XP son:

- Desarrollo iterativo e incremental: que permite pequeñas mejoras, unas tras otras, consecutivamente.
- Pruebas unitarias: estas son continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- Programación en parejas: recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Permitiendo que el código sea revisado al mismo tiempo que se escribe.
- Frecuente interacción del equipo de programación con el cliente o usuario: recomendando que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir una nueva funcionalidad: haciendo entregas frecuentes.
- Refactorización del código: reescribir ciertas partes del código para aumentar su legibilidad.

- Propiedad del código compartido: no se dividen las responsabilidades en el desarrollo de cada módulo. Este método promueve que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

Lo esencial de esta metodología es lograr la comunicación entre desarrolladores y usuarios, la retroalimentación entre ellos, con los usuarios finales y la simplicidad en el código.

SCRUM

SCRUM define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 15 años. Sus principales características se centran en dos aspectos fundamentales. El desarrollo de *software* se realiza mediante iteraciones, denominadas *sprint*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración (ORJUELA y ROJAS, 2008).

Características:

- Equipos auto dirigidos.
- Utiliza reglas para crear un entorno ágil de administración de proyectos.
- No prescribe prácticas específicas de ingeniería.
- Los requerimientos se capturan como elementos de la lista reserva del producto.
- El producto se construye en una serie de *sprint* de un mes de duración.
- Usado para proyectos complejos con requerimientos cambiantes.
- Basado en un control de proceso empírico.

Prácticas:

- Iteración (*Sprint*).
- Reunión de planificación de las iteraciones.
- Reuniones diarias.
- Reuniones de revisión de las iteraciones.
- Reuniones de revisión del diseño.
- Estabilización de las iteraciones.
- Meta *Scrum*.

1.3.3. Herramientas para el desarrollo**Herramientas CASE**

Acrónimo de Computer Aided *Software* Engineering (Ingeniería de *Software* Asistida por Computadoras). Las herramientas CASE son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. Las herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de *software* y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un *software* (INSTITUTO NACIONAL DE ESTADÍSTICA E INFORMÁTICA, 1999).

Visual Paradigm 8

Visual Paradigm es una herramienta multiplataforma de modelado visual UML¹⁴ y una herramienta CASE muy potente y fácil de utilizar. Soporta las últimas versiones de UML. Mediante Visual Paradigm se puede realizar el modelado, la captura de requisitos, diseño de

¹⁴ Acrónimo de Unified Modeling Language (Lenguaje Unificado de Modelado). Metodología que permite crear un entorno de análisis y diseño orientado a objetos. UML tiene como objetivo ayudar a los desarrolladores de *software* a seguir un proceso ya estándar en la implementación de nuevas aplicaciones que permita optimizar tiempo y recursos.

base de datos, modelado de procesos de negocio. Es apoyado por un conjunto de lenguajes, tanto en la generación del código como en la ingeniería inversa, algunos ejemplos son: Java, C++, PHP y Python (VALDÉS, 2007).

Entorno de desarrollo integrado (IDE)

Un entorno de desarrollo integrado (Integrated Development Environment, IDE por sus siglas en inglés) es un programa compuesto por un conjunto de herramientas para un programador. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (ORACLE CORPORATION, 2013).

Netbeans 7.0.1

Netbeans es una herramienta para la redacción, compilación, depuración y ejecución de programas. Está desarrollado en Java y soporta otros lenguajes de programación. Este consta con un amplio número de módulos que permiten la extensión del mismo. Es un producto libre y gratuito sin restricciones de uso (ORACLE CORPORATION, 2013).

Principales características de Netbeans:

- Creación de Proyectos PHP¹⁵: integra lenguajes como HTML¹⁶, JavaScript¹⁷ y CSS¹⁸.

¹⁵ Hypertext Preprocessor o Procesador de Hipertexto (PHP) por sus siglas en inglés. Es un lenguaje de programación del lado del servidor, gratuito e independiente de plataforma. Soporta la programación orientada a objetos y es una alternativa para la creación de portales web dinámicos.

¹⁶ Hypertext Markup Lenguaje o Lenguaje de Marcas de Hipertexto (Html) por sus siglas en inglés. La información publicada en la web debe ser mediante un lenguaje que sea entendido universalmente. El lenguaje utilizado por la *World Wide Web* es el HTML es un lenguaje de hipertexto, es decir, un lenguaje que permite escribir texto de forma estructurada y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento.

¹⁷ JavaScript es un lenguaje script desarrollado por la empresa Netscape Communications. Se utiliza en páginas web HTML para realizar tareas y operaciones en el marco de la aplicación cliente.

¹⁸ Cascade Style Sheet (CSS) u Hoja de Estilos en Cascadas Es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML.

- Editor de código fuente: el editor de PHP es ágil y a la vez robusto, contiene ayuda en línea, reconocimiento de sintaxis y todo lo que proveen las últimas versiones de PHP.
- Integración con PHP *Unit Testing*: es posible crear *test* con PHPUnit, para diferentes funciones, luego realizar la comprobación y ver todos los resultados.
- Depuración de PHP: Netbeans integra muy bien la utilización Xdebug¹⁹, gracias a esto podemos inspeccionar y examinar cada variable local, establecer puntos de interrupción y evaluar el código en nuestra lógica. El IDE de Netbeans para PHP también ofrece la línea de comandos de depuración, la salida del programa PHP aparece en una pantalla de línea de comandos en el IDE de sí mismo y se puede inspeccionar el código HTML generado sin tener que cambiar a un navegador.
- Integración con MySQL²⁰: Netbeans posee una integración completa en términos de administración básica y avanzada de MySQL, y todo desde el mismo entorno.

1.4. Conclusiones parciales

- La revisión bibliográfica y la perspectiva dada por diversos autores, permitió identificar los elementos comunes en los conceptos encontrados desde un enfoque informático, adoptando una posición en la investigación ajustada al objeto de estudio.
- El estudio de diferentes sistemas de prensa digital y de componentes homólogas, permitió conocer el desinterés de estos diarios con relación a las preferencias de los usuarios. Además de la inexistencia de aplicaciones capaces de identificar estos intereses.
- Se descarta la utilización de las aplicaciones encontradas, por no ajustarse a las particularidades de la solución que se necesita, pero sirvieron como base para observar en algunos sistemas, los servicios que brindan a los usuarios.

¹⁹ Xdebug es una extensión para PHP que proporciona un soporte muy completo para la depuración de scripts. Un script es un archivo que incluye un conjunto de comandos. Son ejecutados desde la primera línea hasta la última (de forma secuencial).

²⁰ MySQL es un sistema gestor de base de datos (SGBD). Es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos.

- Con el análisis de las tecnologías establecidas para la construcción de la solución, se seleccionaron los elementos del entorno de desarrollo necesarios para el componente reorganizador de contenidos del diario Mella.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN PARA EL COMPONENTE

2.1. Introducción:

Después de haber analizado y estudiado varios sistemas de prensa digital, tanto nacionales como internacionales en cuanto a la forma de mostrar su información. Se identificaron disímiles características que estos sistemas no tienen en cuenta, una de las principales es que no obtienen información de las preferencias de los usuarios. Debido a su importancia se incorporarán en el diario Mella, por lo que el sistema debe contar con un componente, que le permita reorganizar todas las noticias de la portada de los usuarios registrados según sus preferencias.

2.2. Modelo de dominio

El Modelo de Dominio o Modelo Conceptual muestra (a los modeladores) clases conceptuales significativas en un dominio del problema; es el artefacto más importante que se crea durante el análisis orientado a objetos (CRAIG LARMAN, 2003).

Es una representación visual de los conceptos u objetos del mundo real, significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema.

Representación del modelo de dominio

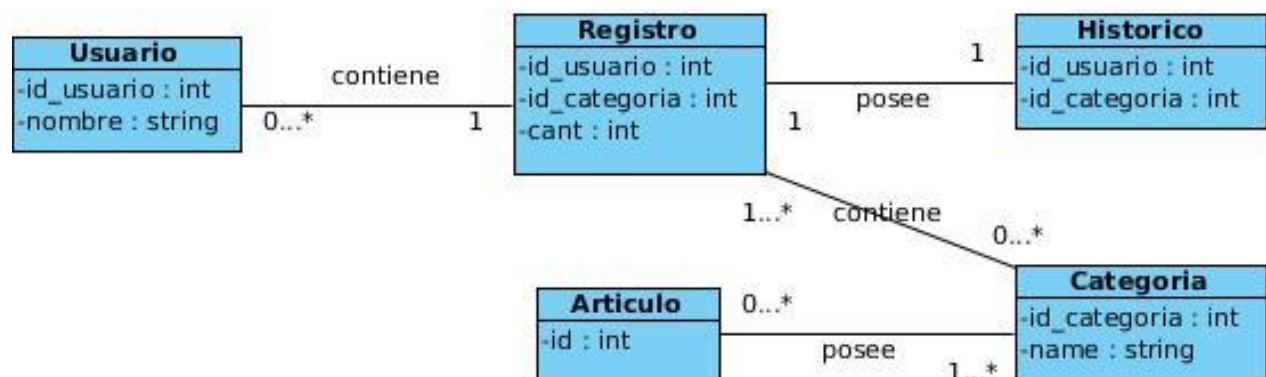


Ilustración 1: Diagrama del modelo de dominio

Fuente: elaboración propia

Glosario de conceptos del modelo de dominio

Categoría: en esta clase se almacenan todos los tipos de categorías a la que puede pertenecer una publicación.

Usuario: en la clase usuario se almacenan las características de los usuarios que acceden al sistema como su identificador y su nombre.

Artículo: esta clase es la que se dedica a almacenar todos los artículos creados por los usuarios en los *blog*.

Registro: en la clase registro es donde se almacena la cantidad de veces que un usuario visita un artículo.

Histórico: es la clase que va a contener las categorías que más visitan los usuarios.

2.3. Requerimientos funcionales

Después de la descripción del modelo de dominio, se hace necesario identificar lo que debe hacer el sistema, por lo que se hace necesario conocer los requisitos funcionales, estos representan las capacidades con las que el sistema debe cumplir, los que deben ser comprendidos tanto por los desarrolladores como por los usuarios. A continuación se relacionan los que debe cumplir el componente a desarrollar.

Tabla 1: Requisitos funcionales

Código	Descripción del requisito funcional	Prioridad
RF1	Contar visitas de los usuarios sobre una noticia.	Alta
RF2	Extraer categoría de las noticias.	Alta
RF3	Obtener la cantidad de visitas por categoría de un usuario determinado.	Alta
RF4	Guardar historial de visitas por categoría.	Alta
RF5	Definir categoría favorita del usuario.	Alta
RF6	Mostrar noticias en la portada de la categoría favorita del visitante.	Alta

RF7	Actualizar categoría favorita.	Alta
RF8	Establecer el tiempo que el componente va a aprender del usuario antes de actuar sobre el mismo.	Media
RF9	Establecer el tipo de rol sobre el cual el componente debe actuar.	Media
RF10	Restablecer a valores predeterminados los datos de un usuario específico.	Baja
RF11	Restablecer a valores predeterminados en general.	Baja

2.4. Requerimientos no funcionales

Los requerimientos no funcionales especifican propiedades o cualidades que el producto de *software* debe tener, como restricciones en cuanto al diseño, rendimiento, facilidad de mantenimiento, seguridad, entre otras.

Entre los requerimientos no funcionales del sistema propuesto se encuentran:

RNF1 - Usabilidad

El componente debe poseer una arquitectura de información simple, que posibilite al administrador del sistema usarlo y configurarlo de forma sencilla.

RNF2 - Soporte

El componente debe dar la posibilidad de ser mejorado, así como de incorporarle nuevas funcionalidades en caso de ser necesarias.

Se debe contar con un manual, con la descripción de las funcionalidades del componente para los roles que ejecutan tareas de gestión de información.

RNF3 - Restricciones de diseño

Como el componente se integra a WordPress, el correcto funcionamiento del mismo depende del comportamiento del sistema, el cual necesita las siguientes herramientas y tecnologías:

Servidor web Apache 2.2.21.

Se requiere un navegador capaz de interpretar JavaScript 1.5 y CSS 3, HTML5, se recomienda Mozilla Firefox en su versión 13.x o superior.

Gestor de base de datos MySQL en su versión 5.0.6.

La versión de PHP debe ser la 5.3.10 o superior.

RNF4 - Legales

El CMS escogido como plataforma para el desarrollo del portal está basado en la licencia GNU/GPL versión 2.

RNF5 - Seguridad

Garantizar que la información sea editada únicamente por las personas que tienen permisos para realizar esta actividad.

Se debe implementar un sistema de salvadas de seguridad de la base de datos y de los ficheros del sistema de la forma siguiente: una copia total al año, una diferencial cada semestre y una incremental cada mes.

RNF6 - Eficiencia

Debe ser capaz de responder en un tiempo menor de 1 segundo para las páginas de información.

RNF7 - Hardware

Por parte del servidor web y de base de datos, varía en dependencia de la aplicación en la cual sea acoplada, por lo que se establece como requerimiento estándar:

Poseer 4 GB de memoria RAM para cumplir con una carga de trabajo media.

Poseer 1024 GB (1 Terabyte) de almacenamiento, y se debe revisar la capacidad pues según el histórico de los eventos a registrar el sitio puede crecer en cuanto a tamaño.

El micro procesador debe tener cuatro núcleos o más y no menos de 1.5 MHz.

RNF8 - Por parte del cliente:

Debe tener mínimo 256 Mb de RAM

Un micro procesador Intel Pentium 4 a 1.2 GHz o superior.

RNF9 - Disponibilidad

El componente debe estar disponible a toda hora, siempre y cuando la aplicación esté operativa.

Se debe realizar tareas de mantenimiento una vez al año para revisar las capacidades óptimas de trabajo del componente.

RNF10 - Portabilidad

El componente debe ser genérico, permitiendo su instalación en cualquier sistema que utilice el CMS WordPress en su versión 3.0 en adelante.

2.5. Historias de usuario

Se realiza una descripción de las historias de usuarios (HU) con el mínimo nivel de detalle. Es necesario destacar que las HU han jugado un papel importante en la estimación de los tiempos requeridos para el desarrollo del proyecto.

A continuación se muestran dos ejemplos de los contenidos de las fichas de las HU, para consultar el resto debe remitirse al Anexo 1.

Tabla 2: Historia de usuario “Obtener visitas de los usuarios sobre una noticia”

HU1- HU Contar visitas de los usuarios sobre una noticia.

Historia de Usuario	
Número: HU1	Nombre Historia de Usuario: HU1- HU Contar visitas de los usuarios sobre una noticia.
Modificación de Historia de Usuario Número: 1	
Referencia:	
Programador: Pedro Luis León Fundora	Iteración Asignada: 1
Prioridad en Negocio: alta	Puntos Estimados: 1
Riesgo en Desarrollo: alta	Puntos Reales: 1

Descripción: Luego que el usuario accede al sistema este comienza a obtener el identificador del consumidor junto con el identificador de la noticia que está visitando en ese momento.

Observaciones: El usuario debe estar previamente autenticado en el sistema.

Prototipo de interfaz:

Tabla 3: Historia de usuario “Extraer categoría de las noticias”

HU2- HU Extraer categoría de las noticias.

Historia de Usuario	
Número: HU2	Nombre Historia de Usuario: HU2- HU Extraer categoría de las noticias.
Modificación de Historia de Usuario Número: 1	
Referencia:	
Programador: Pedro Luis León Fundora	Iteración Asignada: 1
Prioridad en Negocio: alta	Puntos Estimados: 1
Riesgo en Desarrollo: alta	Puntos Reales: 1
Descripción: Posteriormente se procede a identificar las categorías a la que pertenece cada noticia visitada por un usuario.	
Observaciones: El usuario debe estar previamente autenticado en el sistema.	
Prototipo de interfaz:	

2.6. Diseño de la Base de Datos

En el siguiente diagrama solo se muestran las tablas con las que interactúa el componente.

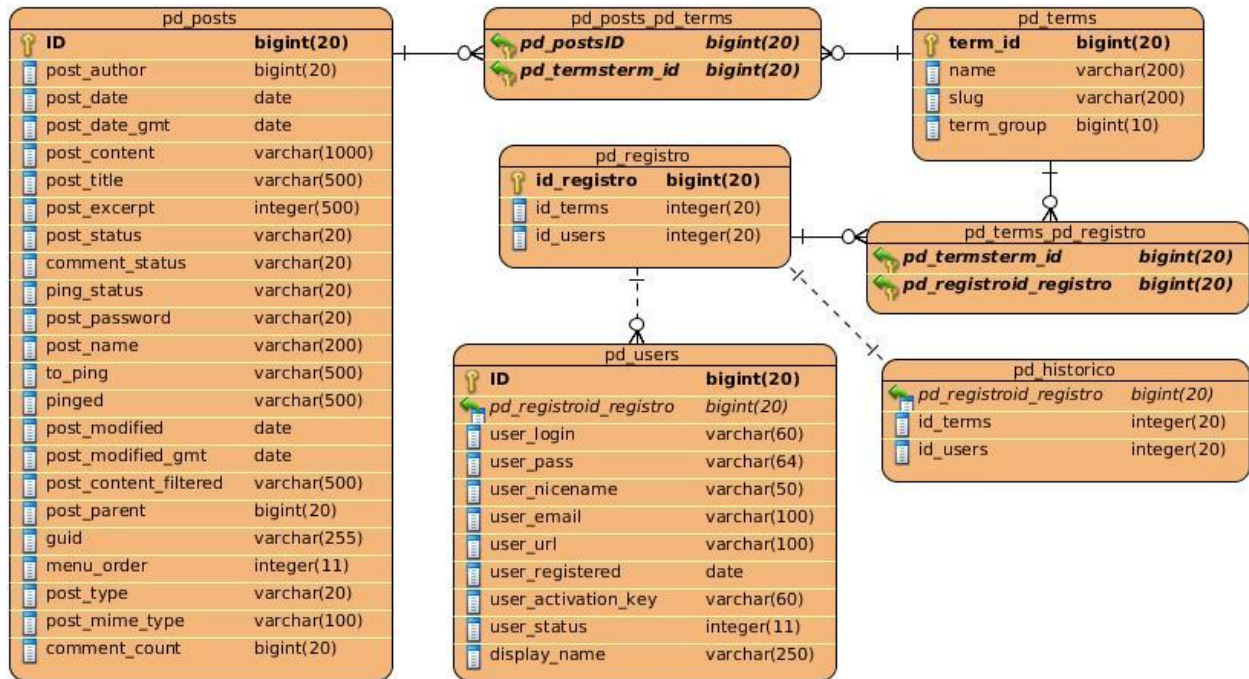


Ilustración 2: Modelo de datos del componente

Fuente: elaboración propia

A continuación se hace una breve descripción de las tablas del modelo de datos conjunto sus principales atributos:

Tabla 4: Artículo

pd_posts		
Descripción	Tabla que contiene la información de los artículos.	
Atributo	Tipo	Descripción
post_autor	bigint	Nombre del autor del artículo.
post_date	date	Fecha de publicación.
post_content	varchar	Contenido o cuerpo del artículo.
post_title	varchar	Título del artículo.
post_type	varchar	Tipo de artículo.

Tabla 5: Categoría

pd_terms		
Descripción	Tabla que contiene la información de las categorías.	
Atributo	Tipo	Descripción
<i>name</i>	<i>varchar</i>	Nombre de la categoría.
<i>term_group</i>	<i>bigint</i>	Grupo al que pertenece.

Tabla 6: Registro

pd_registro		
Descripción	Tabla que registra todas las acciones de los usuarios en el sistema, almacenando identificador del usuario e identificador de la categoría del artículo que este visitando.	
Atributo	Tipo	Descripción
<i>id_terms</i>	<i>integer</i>	Identificador de una categoría.
<i>id_users</i>	<i>integer</i>	Identificador de un usuario.

Tabla 7: Usuario

pd_users		
Descripción	Tabla que contiene la información de los usuarios.	
Atributo	Tipo	Descripción
<i>id</i>	<i>bigint</i>	Nombre del autor del artículo.
<i>user_login</i>	<i>varchar</i>	Usuario de autenticación en el sistema.
<i>user_pass</i>	<i>varchar</i>	Contraseña del usuario.
<i>user_email</i>	<i>varchar</i>	Dirección electrónica del usuario.
<i>user_registered</i>	<i>date</i>	Fecha en que se registró.

Tabla 8: Histórico

pd_historico		
Descripción	Tabla que almacena el identificador de un usuario y el identificador de su categoría favorita.	
Atributo	Tipo	Descripción
<i>id_terms</i>	<i>integer</i>	Identificador de una categoría.
<i>id_users</i>	<i>integer</i>	Identificador de un usuario.

2.7. Arquitectura

El componente fue realizado para WordPress el cual adquiere la propia arquitectura de este CMS. WordPress contiene 3 ficheros principales: *wp_admin*, *wp_includes*, *wp_content* este último posee 2 ficheros importantes (*themes* y *plugins*).

- *wp_admin*: posee todas las funcionalidades, páginas, enlaces, usuarios entre otros elementos por la parte de la administración.
- *wp_includes*: contiene un conjunto de ficheros indispensables para el funcionamiento de WordPress. Incluye todo un conjunto de implementaciones de clases, que son empleados por los *plugins* (se encuentran también las clase de acceso a datos).
- *wp_content*: contiene 2 directorios, uno para los temas y otro para los *plugins*. Los temas son los que controlan la apariencia de todas las páginas y brinda una mejor vista del diseño externo y los *plugins* son complementos que agregan funciones al sistema.

Esta arquitectura se encuentra estructurada por una única página servidor que genera el contenido que a la vez se muestra en la página final, estas páginas generadas en WordPress pueden o no contener formularios. En la figura a continuación se representa el funcionamiento de WordPress como aplicación web (MENA, 2012).

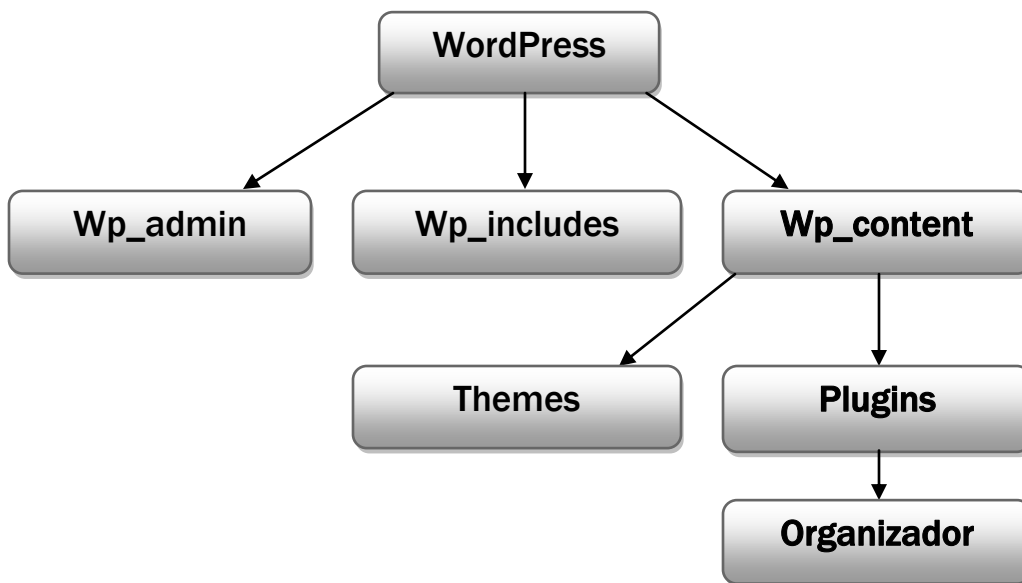


Ilustración 3: Arquitectura de la plataforma

Fuente: elaboración propia

2.7.1. Patrones GRASP

Bajo acoplamiento

Nos indica la intensidad con que una clase está relacionada con otras, con que las conoce y con que recurre a ellas. Bajo acoplamiento nos indica que una clase no depende de muchas clases.

La alta dependencia entre las clases presentes no existe, ya que una puede existir sin necesidad de que la otra esté creada y a la vez no sufre ningún daño si alguna es eliminada.

La dependencia entre las clases como artículos (*post*), usuarios (*user*), comentarios (*comment*) es baja ya que una puede existir sin necesidad de que la otra exista, por lo que el cambio no afecta el funcionamiento de las demás clases (MENA, 2012).

Alta cohesión

Muestra el nivel de relación que tienen estas clases entre sus responsabilidades. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas para que no realicen un enorme trabajo.

Se manifiesta en la coherencia de la información que hay en la composición de las clases, pero a la vez no aparece ninguno que dé a entender que no pertenece a esta clase (MENA, 2012).

Se manifiesta en la coherencia de la información que hay en la composición de las clases, por ejemplo la clase usuario (*user*) está compuesta por atributos relacionados con ella misma, como nombre (*name*), nombre completo (*nickname*), correo (*email*), estado (*status*), entre otros, pero a la vez no aparece ninguno que dé a entender que no pertenece a esta clase.

Experto

La clase responsable de realizar una tarea relacionada con algún atributo es aquella que lo contenga. Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada (MENA, 2012).

Estos son los aspectos del sistema:

- Lógica de negocio

- Persistencia a la base de datos
- Interfaz de usuario

La clase usuario (*user*), es la que contiene los atributos necesarios para la creación de un usuario, y no para la creación de un artículo, o comentario, ya que las clases artículo (*post*) y comentario (*comment*) son las que contienen los atributos necesarios para realizar esta función.

Controlador

Como su nombre lo indica, este patrón asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema (MENA, 2012).

En WordPress, las responsabilidades de las clases están bien distribuidas, de forma tal que todas las actividades relacionadas con los usuarios, son manejadas por la clase usuario (*user*), de igual forma ocurre con el resto de las clases del sistema.

2.7.2. Patrones GOF

Para lograr un diseño de calidad, se tuvo en cuenta disímiles patrones de diseño. Su objetivo principal es solucionar problemas relacionados con las clases, pues estos, a partir de la información que posee cada una de ellas, determina cuál debe asumir o no una responsabilidad (MENA, 2012).

Patrón creacional:

Singleton: garantiza que solamente se crea una instancia de la clase y provee un punto de acceso global a él, en el caso de los diferentes objetos existentes, como usuario (*user*), en cualquier parte que se llame dentro del proyecto se pueden usar sus atributos, sin necesidad de crear nuevas instancias de dichos objetos (MENA, 2012).

Patrón de comportamiento:

Mediador: se pone en práctica cuando se comunican los diferentes objetos de WordPress, ya que existe una variedad de ellos que se comunican entre sí. Por ejemplo, el objeto usuario (*user*) se comunica con el objeto artículo (*post*). Ya que cada artículo tiene un autor que es un usuario (*user*) (MENA, 2012).

2.8. Concepto y clasificación de la audiencia**¿Qué son las audiencias?**

Son las personas clave o estratégicas para realizar determinadas funciones en un sistema. Una estrategia fundamental es la segmentación de la audiencia para que se pueda llegar a cada audiencia con mensajes focalizados y de acuerdo a los intereses personales, con base en su perfil²¹ y en el rol²² que desempeñan.

Se hace necesario clasificar la audiencia porque no todos los roles tienen un seguimiento de su comportamiento a partir del componente desarrollado, sino que son los usuarios comunes con el rol de suscriptor. A los demás roles que interactúan en la administración del sistema, como los autores, editores y administrador, no se les tiene en cuenta para mostrarle el contenido personalizado del sistema.

Clasificación de la audiencia**Roles común:**

- **Suscriptor:** es un usuario que no contribuye con el sistema en cuanto a la publicación de contenido, aunque si participa de forma indirecta mediante comentarios, y puede colaborar enviando noticias desde la interfaz principal del sistema.
- **Colaborador:** este usuario ya puede redactar sus propias noticias que se mantienen en forma de borrador hasta que usuarios con más permisos lo moderen. Además, este

²¹ Es la descripción detallada de las posibles transacciones que puede realizar un usuario en el sistema.

²² El rol es el nombre que se le confiere al conjunto de perfiles que le son asignados al usuario para el ejercicio de sus funciones.

perfil permite editar y borrar sus propios borradores y visualizar categorías, comentarios y el número de comentarios pendientes de moderar.

- **Autor:** permite que el usuario pueda crear, publicar, editar, borrar artículos, tanto pendientes como publicados. Además tiene la posibilidad de subir adjuntos a los artículos. Como su nombre indica, únicamente pueden generar contenido y por ello no les está permitido moderar comentarios, categorías, entre otros.
- **Editor:** dispone de más conocimiento y está capacitado para administrar completamente sus propias entradas y las de otros usuarios, además de administrar la creación, modificación, publicación de páginas, categorías, comentarios, enlaces, entre otros.
- **Administrador del sistema:** además de todo lo que los usuarios anteriores pueden realizar, este tiene la capacidad de administrar *plugins*, temas y otros elementos, así como llevar un control completo sobre los usuarios que tienen un nivel inferior.

Usuarios especiales:

Administrador del sistema: dispone de todo el acceso al sistema y controla a todos los roles anteriores, es el que posee los permisos superiores, se encarga de mantener y controlar la plataforma.

2.9. Conclusiones parciales

- La descripción del modelo de dominio, permitió representar cómo se obtiene la información ingresada por los usuarios, además de la forma en que realiza el proceso de reorganización y publicación del componente.
- La elaboración de los artefactos: requisitos funcionales y no funcionales, historias de usuarios, todos definidos por las metodologías XP y SCRUM guiadas por el modelo de desarrollo CMMI en su nivel 2, permitió describir las características de la solución propuesta para documentar el proceso de desarrollo y guiar el trabajo.
- Teniendo en cuenta que el presente trabajo de diploma, tributa a una aplicación consolidada. Se plantea que el desarrollo de la propuesta de solución basada en la arquitectura y los patrones con que cuenta WordPress, brindó la flexibilidad y adaptabilidad presentes en el componente.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

3.1. Introducción

En el presente capítulo se abordan aspectos fundamentales acerca del desarrollo del componente. Se muestra cómo ha sido implementada la propuesta basándose en el modelo de datos, además se representan los estándares de código utilizados en la programación del componente. Para asegurar que las historias de usuario estén terminadas y que cumplen con las expectativas del usuario final es necesario probar el componente y verificar los resultados mediante el diseño de los casos de prueba.

3.2. Estándares de codificación

Para la implementación del componente se propone un estándar de código basado en la librería PEAR²³. Esto se debe al constante cambio de los desarrolladores y que los sistemas deben ser mantenidos por equipos de soporte, que necesitan entender el código para posibles cambios.

Comillas simples y dobles

Estas deben usarse apropiadamente, las simples se deben utilizar cuando en una cadena no se está evaluando nada. Por lo general nunca se prescinde del uso de ambos tipos de comillas, puesto que el estilo de comillas se puede alternar en el código, por ejemplo:

```
$table_name = $wpdb->prefix . "historico";  
$sql = "INSERT INTO $table_name(id_terms , id_users) VALUES ('.$obj[$i]->id_terms.', '$obj[$i]->id_users.')";  
$wpdb->query($sql);
```

Ilustración 4: Comillas simples y dobles

Fuente: elaboración propia

²³ PEAR, o PHP Extension and Application Repository, es un entorno de desarrollo y sistema de distribución para componentes de código PHP. El proyecto PEAR fue fundado por Stig S. Bakken en 1999 para promover la reutilización de código que realizan tareas comunes.

Estilo de llaves

Se define el uso de llaves para bloques con varias líneas como se muestra a continuación:

```
function ordenar_post_panel() {  
    include('template/menu.html');  
}
```

Ilustración 5: Estilo de llaves

Fuente: elaboración propia

Los ciclos también deben contener llaves al inicio y fin, por ejemplo:

```
foreach($post_categories as $c) {  
    $sql = "INSERT INTO $table_name(id_terms, id_users) VALUES ('$c', '$user_ID')";  
    $wpdb->query($sql);  
}
```

Ilustración 6: Estilo de llaves para ciclos

Fuente: elaboración propia

Uso de espacios

Siempre se debe utilizar espacios después de las comas y antes y después de los operadores lógicos de asignación, por ejemplo:

```
$table_name = $wpdb->prefix . "registro";  
$obj = $wpdb->get_results("SELECT id_users , id_terms , MAX(cantidad)  
FROM (SELECT id_users , id_terms , COUNT( id_terms ) AS cantidad  
FROM $table_name  
GROUP BY id_users , id_terms  
ORDER BY id_users , COUNT( id_terms )) AS result  
GROUP BY result.id_users , id_terms  
ORDER BY result.id_users , cantidad;");
```

Ilustración 7: Uso de espacios

Fuente: elaboración propia

Etiquetas y atributos

Las etiquetas y atributos del lenguaje HTML, deben estar correctamente escritos, iniciados y finalizados con corchetes angulares, las etiquetas que pueden encerrar textos y otros elementos, la terminación indica la finalización de la tarea que realicen. Para las etiquetas de auto cierre, el *slash* para el cierre debe estar precedido por un espacio.

```
<html>
<form method="post" action="" id="radio">
Definir tiempo de aprendizaje del plugin sobre el usuario:<br />
<input name="tiempo" type="radio" value="rb1dia" checked="checked" />1 día<br />
<input name="tiempo" type="radio" value="rb3dias" />3 días<br />
<input name="tiempo" type="radio" value="rb7dias" />7 días<br />
<input name="tiempo" type="radio" value="rb15dias" />15 días<br />
<input type='submit' name='' value='Activar' />
</form>
</html>
```

Ilustración 8: Etiquetas y atributos

Fuente: elaboración propia

3.3. Análisis y descripción de las nuevas funcionalidades y el uso de las ya existentes

Se conoce como reutilización de código al uso de una parte o la totalidad de un programa informático existente, para la construcción de otro. De esta forma se aprovecha el trabajo anterior, se economiza tiempo, y se reduce la redundancia.

La ubicación del código reusable en un único lugar y llamarlo desde los diferentes archivos es la forma más común de reutilizar código. Este proceso se conoce como abstracción y puede verse claramente en el archivo *functions.php* que es donde se implementan diferentes funcionalidades que van a ser llamadas cuando se necesiten, evitando así que el desarrollador tenga que implementar o duplicar el código en las vistas finales que necesitan la funcionalidad requerida.

Para el desarrollo del componente se utilizan los beneficios que ofrece la reutilización de código, debido a que se utilizan extensiones implementadas con anterioridad que facilitan el trabajo a los desarrolladores, a continuación se mencionan y explican las más importantes:

WP-Slimstat: Esta funcionalidad se encarga de almacenar en la base de datos las estadísticas que va generando WordPress.

Si bien es cierto que las extensiones sirven de gran ayuda a la hora de implementar funcionalidades complejas, también es cierto que WordPress pone en manos de los desarrolladores una serie de funciones necesarias para el desarrollo de aplicaciones web. Estas funciones vienen implementadas en el sistema ya que el código que brindan es de uso frecuente en todas las implementaciones. A continuación se describen las más utilizadas:

Tabla 9: Funciones de WordPress

Funciones	Descripción
is_user_logged_in()	Comprueba que el usuario esta autenticado.
get_the_id()	Obtiene el identificador de una categoría.
wp_get_post_categories()	Obtiene de una publicación su categoría.
get_current_user_id()	Obtiene el identificador del usuario autenticado que este visitando una publicación.
get_results()	Obtiene el resultado que devuelve una consulta de base de datos.
query()	Recibe una consulta de base de datos y la ejecuta en la base de datos.
add_options_page()	Adiciona una nueva opción en el menú de WordPress.
add_action()	Adiciona una función a WordPress.
function_exists()	Comprueba la existencia de una funcionalidad, para evitar llamar a alguna no existente.

Tabla 10: Funciones implementadas

Funciones	Descripción
get_id_category_user()	Obtiene el ID del usuario y el ID del artículo que se esté visitando en ese momento.
get_categoria_favorita()	Obtiene los artículos más visitados por cada usuario y los archiva en la base de datos.
categoria_favorita()	Obtiene la categoría favorita dado un usuario.
contador_visitas()	Función que recibe como parámetro un usuario para obtener sus visitas al sistema.

obtener_rol()	Función que recibe como parámetro el identificador de un usuario para obtener su rol correspondiente.
puede_mostrar()	Función que recibe como parámetro el identificador de un usuario para conocer si el componente puede mostrarle contenido.
ordenar_post_instala()	Adiciona tablas a la base de datos.
ordenar_post_desinstala()	Elimina tablas de la base de datos.
ordenar_post_panel()	Crear una ventana dentro del panel de administración que nos permita modificar las opciones del componente.

3.4. Diagrama de despliegue

El diagrama de despliegue muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

La siguiente figura muestra cómo será desplegado el *software*:

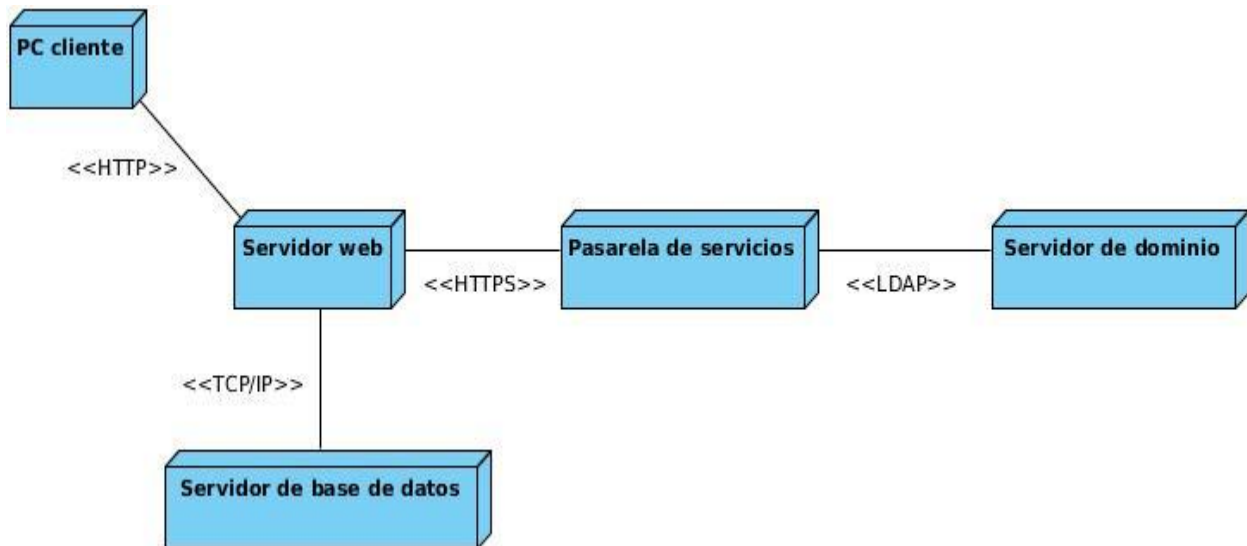


Ilustración 9: Diagrama de despliegue

Fuente: elaboración propia

Descripción de los nodos

PC-cliente

Las PC-cliente acceden al sistema con un navegador web, que puede ser: Mozilla Firefox o Chrome. Se comunican con el servidor web mediante el protocolo HTTP.

Servidor web

Este tiene el dominio del contenido de la aplicación y una fuerte integración con las bases de datos, ya que la mayoría de las transacciones relacionadas con el acceso a los datos de la aplicación utilizan la conexión TCP/IP. Este servidor permite que las PC-cliente interactúen y tengan acceso al sistema.

Servidor de base de datos

Los servidores de bases de datos son aplicaciones que permiten gestionar datos. Se utilizan para cualquier aplicación que necesite almacenamiento, acceso y análisis de datos estructurados.

Servidor de dominio

Un directorio activo almacena información referente a los usuarios de una infraestructura y el acceso o permisos que estos tienen para interactuar o acceder al resto de los recursos de la red.

Pasarela de servicios

La pasarela de autenticación es un servicio diseñado con el objetivo de garantizar seguridad, y confiabilidad a la hora de interactuar con el servidor de dominio, a través de ella se realiza el proceso de autenticación de los usuarios en las diferentes aplicaciones que usan el servicio sin necesidad de que las mismas interactúen de forma directa con el servidor que contiene el directorio activo.

Descripción de elementos e interfaces de comunicación

HTTP

El protocolo HTTP (*HyperText Transfer Protocol*) define la sintaxis y la semántica que utilizan los elementos de *software* de la arquitectura web (clientes, servidores, proxy) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador web o un *spider*) se lo conoce como "*user agent*" (agente del usuario). A la información transmitida se le llama recurso y se le identifica mediante un URL²⁴. Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc.(CAROLINA GIL MEZA, 2013)

TCP/IP

El conjunto TCP/IP (Protocolo de comunicación y transmisión/ Protocolo de Internet) está diseñado para enrutar y tiene un grado elevado de fiabilidad, es adecuado para redes grandes y medianas, así como en redes empresariales. Se utiliza a nivel mundial para conectarse a Internet y a los servidores web. Es compatible con las herramientas estándar para analizar el funcionamiento de la red. Es más rápido en redes con un volumen de tráfico grande donde haya que enrutar un gran número de tramas. El conjunto TCP/IP se utiliza tanto en campus universitarios como en complejos empresariales, en donde utilizan muchos enrutadores y conexiones a mainframe o a ordenadores UNIX, así como también en redes pequeñas o domésticas, en teléfonos móviles y en domótica (CHACA, 2013).

LDAP

El Protocolo Ligero de Acceso a Directorios (en inglés, *Lightweight Directory Access Protocol*, LDAP) es un conjunto de protocolos abiertos usados para acceder a información guardada de forma centralizada a través de la red. Está basado en el estándar X.500 para compartir directorios, pero es menos complejo e intensivo en el uso de recursos. Es un sistema cliente/servidor. El servidor puede usar una variedad de bases de datos para guardar un directorio, cada uno optimizado para operaciones de lectura rápidas y en gran volumen.

²⁴ Sigla en inglés de Uniform Resource Locator (localizador de recursos uniforme), es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación.

Cuando una aplicación cliente LDAP se conecta a un servidor LDAP puede, o bien consultar un directorio, o intentar modificarlo. En el evento de una consulta, el servidor, puede contestarla localmente o puede dirigir la consulta a un servidor LDAP que tenga la respuesta. Si la aplicación cliente está intentando modificar información en un directorio LDAP, el servidor verifica que el usuario tiene permiso para efectuar el cambio y después añade o actualiza la información (DONNELLY, 2000).

HTTPS

Acrónimo de *Hypertext Transfer Protocol Secure* (Protocolo Seguro de Transferencia de Hipertexto). Es la versión segura del protocolo HTTP (que se utiliza a su vez, para visualizar las páginas web) y viene dada por el protocolo de seguridad SSL²⁵, el cual crea un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. HTTPS es utilizado por los navegadores más populares, como Internet Explorer, Firefox o Safari (BARRENO, 2013).

3.5. Pruebas

Son procesos que se realizan con el objetivo de ejercitar un programa, tiene como intención encontrar errores antes de ser entregado al usuario final. Se ejecutan bajo condiciones o requerimientos específicos, los resultados son observados y registrados, al final se realiza una evaluación y rectificación de algunos aspectos de sistema o componente.

3.5.1. Pruebas funcionales

A continuación se muestran dos de los casos de pruebas diseñados a las historias de usuario. El resto de los casos de pruebas asociados a las restantes historias de usuarios pueden consultarse en el anexo 2.

Tabla 11: Caso de prueba “Obtener visitas”

Caso de prueba de aceptación	
Código: HU1_P1	Historia de Usuario: HU1- HU Contar visitas de los usuarios sobre una noticia.
Nombre: Obtener visitas.	

²⁵ Security Sockets Layer, en español Capa de Conexión Segura. Es un sistema de intercambio y autenticación de información utilizado para establecer un canal de comunicaciones seguro.

Descripción: Probar que el sistema obtenga correctamente las visitas de un usuario a una noticia específica.
Condiciones de Ejecución: La aplicación debe ser ejecutada con privilegios de rol suscriptor.
Entrada / Pasos de Ejecución: El usuario suscriptor debe navegar por el sistema.
Resultado Esperado: El sistema debe obtener las visitas de un usuario sobre una noticia.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 12: Caso de prueba “Extraer categoría”

Caso de prueba de aceptación	
Código: HU2_P1	Historia de Usuario: HU2- HU Extraer categoría de las noticias.
Nombre: Extraer categoría.	
Descripción: Probar que el sistema extraiga correctamente el identificador de una categoría de una noticia visitada por un usuario suscriptor.	
Condiciones de Ejecución: La aplicación debe ser ejecutada con privilegios de usuario suscriptor.	
Entrada / Pasos de Ejecución: El usuario suscriptor debe navegar por el sistema.	
Resultado Esperado: El sistema debe obtener el identificador de la categoría a la que pertenece la noticia visitada.	
Evaluación de la Prueba: Prueba satisfactoria.	

Análisis de los resultados:

Para comprobar que el componente emita los resultados esperados, se realizan 4 iteraciones de pruebas. En la primera iteración se ejecutan 6 pruebas funcionales en conjunto cliente-desarrollador para las 3 historias de usuario que se implementan en esa iteración, de las cuales 5 coinciden con los resultados esperados representando un 83%. Por otra parte una prueba resulta fallida representando un 17%. En una segunda iteración para las otras 4 historias de usuarios desarrolladas, se realizan 7 pruebas funcionales y las 7 resultan satisfactorias constituyendo un 100%. En una tercera iteración se realizan 11 pruebas a las 4 historias de usuarios faltantes, de las cuales 9 resultan satisfactorias representando el 82%, quedando dos pruebas insatisfactorias lo que representa el 18%. En la última iteración luego de finalizar el desarrollo del componente, se le realizan pruebas a todas las funcionalidades, obteniendo resultados satisfactorios, pues para el desarrollo de 17 pruebas funcionales se obtiene un total de 17 pruebas exitosas, para un 100% de pruebas satisfactorias.

A continuación se muestra como quedan reflejados los resultados por cada una de las iteraciones de pruebas funcionales realizadas el sistema:

Tabla 13: Iteraciones de pruebas funcionales

Iteración	Historias de Usuarios	Satisfactoria	Insatisfactoria	Total
1	3	5	1	6
2	4	7	0	7
3	4	9	2	11
4	11	17	0	17

3.5.2. Pruebas de carga y estrés

Con esta prueba, se puede observar el comportamiento del sistema con el componente integrado y sin él, para realizar una comparación del rendimiento del sistema en condiciones de extrema carga, brinda además una idea de la respuesta del sistema en tiempo real si se supera la carga esperada. De esta manera se puede comprobar la poca afectación que sufre el sistema una vez acoplada la propuesta. Para la realización de la prueba, se utiliza de la herramienta Jmeter²⁶.

Resultados de las pruebas de carga y estrés

A continuación se muestra los resultados de las pruebas, tras la ejecución de las pruebas de carga y estrés a solo una de las vistas del sistema, para así validar requisitos no funcionales, los cuales ya han sido definidos con anterioridad en la presente investigación. Como el componente solo trabaja sobre la página de inicio esta fue la vista escogida para realizarle las pruebas.

²⁶ <http://jmeter.apache.org/>

Tabla 14: Sistema sin el componente

No. usuarios	Peticiones al servidor	Tiempo min. de respuesta (ms)	Tiempo máx. de respuesta (ms)	Rendimiento (ms)
1	1625	1	295	1.8
5	3147	1	364	2.2
10	5721	1	1061	6.5
50	13592	1	2045	7.1

Tabla 15: Sistema con el componente

No. usuarios	Peticiones al servidor	Tiempo min. de respuesta (ms)	Tiempo máx. de respuesta (ms)	Rendimiento (ms)
1	1820	1	223	2.2
5	6632	1	402	2.3
10	6030	1	1423	6.9
50	14612	1	2777	7.2

3.6. Conclusiones parciales

- La utilización de los estándares de codificación con que cuenta la Universidad, permitió la elaboración del código del componente de manera limpia y organizada, siendo este entendible para cualquier programador que utilice el código desarrollado.
- La especificación del Diagrama de Despliegue, posibilitó contar con una vista de cómo quedará el sistema en el momento de su explotación por parte de los usuarios de la comunidad universitaria.
- Para la validación de la solución obtenida, se utilizaron las pruebas de funcionalidad, así como las de carga y estrés. Esto corroboró un buen desempeño de las funcionalidades implementadas de manera satisfactoria, demostrando que la aplicación con el componente incorporado, no afecta el rendimiento de la misma.

CONCLUSIONES GENERALES

La presente investigación cumple un papel fundamental, al servir de ayuda para la elaboración de componentes similares para sistemas de prensa digital existentes en la Universidad de las Ciencias Informáticas o fuera de ella realizados con el CMS WordPress.

Los aspectos teóricos y conceptos abordados en la primera etapa de la investigación, hizo posible el desarrollo del componente y un mejor entendimiento de las principales características y tendencias de los sistemas de prensa digital en la actualidad.

Las herramientas Netbeans y Visual Paradigm permitieron desarrollar satisfactoriamente el componente para el sistema de prensa digital de la Universidad.

El uso del CMS WordPress y las buenas prácticas de las metodologías ágiles SCRUM y XP enfocadas hacia el nivel 2 de CMMI, aportó gran fiabilidad, rapidez y organización en el proceso de desarrollo de *software*.

Las pruebas de funcionalidad, así como las de carga y estrés, comprobaron el buen funcionamiento del componente y del sistema de acuerdo con los requisitos planteados.

Se documentó toda la investigación desde su comienzo, generando los artefactos propios de las metodologías utilizadas, facilitando de esta manera que otros usuarios o desarrolladores puedan comprender la concepción del sistema en poco tiempo.

Con el cumplimiento de las tareas investigativas, se desarrolló un componente novedoso de WordPress para el sistema de prensa digital (Mella) de la Universidad de las Ciencias Informáticas, facilitando a los usuarios autenticados un entorno con las noticias de su preferencia en la portada del sistema. Dando cumplimiento al objetivo de la investigación.

RECOMENDACIONES

Se recomienda la integración del componente realizado, a otros sitios de la Universidad desarrollados con WordPress, de esta forma los contenidos publicados estarían mejor dirigidos a su público objetivo. Al brindarles la información de sus intereses en la portada de estos sitios, sin necesidad de navegar a páginas interiores.

BIBLIOGRAFÍA REFERENCIADA

1. BARRENO, B. Protocolo HTTPS [Página Web: Sitio Oficial]. SlideShare Incorporated, [Consultado el: abril de 2013]. Disponible en: <http://es.slideshare.net/babp/protocolo-https-3617275>.
2. BETH CHRISSIS, M.; KONRAD, M., et al. CMMI Guía para la integración de procesos y la mejora de productos. España: Pearson Educación Sociedad Anónima., 2009. ISBN 9788478290963.
3. CANÓS, J; LETELIER, P, et al. Metodologías Ágiles en el desarrollo de *Software*. España: Universidad Politécnica de Valencia, 2003.
4. CAROLINA GIL MEZA, A. Protocolos de transferencia SlideShare Incorporated, [Consultado el: abril de 2013]. Disponible en: <http://es.slideshare.net/ingalegil/protocolos-de-transferencia>.
5. CASAS CERVERO, J. Content Management System. Trabajo de diploma, Facultad de Informática de Barcelona. Universidad Politécnica de Cataluña, 2011.
6. CÍRCULO DE PERIODISTAS CUBANOS. Para webmasters y desarrolladores Cuba: Cubadebate, [Consultado el: febrero de 2013]. Disponible en: <http://www.cubadebate.cu/editores/webmasters/>.
7. CRAIG LARMAN. Modelo del Dominio. Prentice Hall, 2003.
8. CHACA, J. Protocolos de red [Página web]. SlideShare Incorporated, [Consultado el: abril de 2013]. Disponible en: http://es.slideshare.net/j_chaca/protocolos-de-red.
9. DEPARTAMENTO DE EDUCACIÓN, P. L. Y. C. ¿Qué es la prensa escrita? [Página web]. España: Departamento de Educación, Universidades e Investigación del Gobierno Vasco, [Consultado el: mayo de 2013]. Disponible en: <http://www.hiru.com/medios-de-comunicacion/que-es-la-prensa-escrita>.
10. DONNELLY, M. Una Introducción a LDAP Plone Foundation, [Consultado el: abril de 2013]. Disponible en: http://softwarelibre.mes.edu.cu/index_html/centro-de-asistencia-tecnica/manual/una-introduccion-a-ldap/una-introduccion-a-ldap.

11. EDICIONES EL PAÍS, S. L. Corporativo [Página web]. España: El País, [Consultado el: mayo de 2013]. Disponible en: <http://elpais.com/corporativos/>.
12. GONZÁLEZ, M. A. C. Convivencia de la prensa escrita y la prensa on line en su transición hacia el modelo de comunicación multimedia. Estudios sobre el mensaje periodístico, 2001, nº 7, p. 71-78. ISSN 1134-1629.
13. INSTITUTO NACIONAL DE ESTADÍSTICA E INFORMÁTICA. Herramientas Case [Página Web: Institucional]. Perú: INE, [Consultado el: abril de 2013]. Disponible en: <http://www.inei.gob.pe/biblioineipub/bancopub/Inf/Lib5103/Libro.pdf>.
14. MENA, C. M. C. Desarrollo del Sistema de Prensa Digital de la Universidad de las Ciencias Informáticas. Tutor: Hernández, I. C. G. y García, I. Y. D. Trabajo de diploma, Facultad 1. Universidad de las Ciencias Informáticas, 2012.
15. ORACLE CORPORATION. ¿Qué es NetBeans? [Página Web]. NetBeans, [Consultado el: febrero de 2013]. Disponible en: https://netbeans.org/index_es.html.
16. ORJUELA, A. y ROJAS, M. Las metodologías de desarrollo ágil como una oportunidad para la ingeniería del *software* educativo. Avances en Sistemas e Informática, 2008, vol. 5, nº 2, p. 159-170. ISSN 1909-0056.
17. RIVERA, D. El ciberperiodismo en Ecuador. Casos de estudio La Hora y Crónica, periódicos de Loja. QUÓRUM ACADÉMICO, 2011, vol. 8, nº 16, p. 12. Disponible en: <http://dialnet.unirioja.es/servlet/articulo?codigo=3865226>. ISSN 1690-7582.
18. SOCIEDAD ANÓNIMA INFORMACION Y CONTROL DE PUBLICACIONES. [Página web]. España: El Mundo, [Consultado el: abril de 2013]. Disponible en: <http://www.elmundo.es/>.
19. TURNER BROADCASTING SYSTEM INCORPORATED. [Página web]. Cable News Network (CNN), [Consultado el: mayo de 2013]. Disponible en: <http://cnnespanol.cnn.com/>.
20. VALDÉS, J. M. Tecnologías del *software*. Barcelona - España: 2007. 43 p. ISBN 978-84-9849-050-3.
21. WORDPRESS ORGANIZATION. Description WordPress, [Consultado el: febrero de 2013]. Disponible en: http://codex.wordpress.org/Function_Reference/query_posts.

BIBLIOGRAFÍA CONSULTADA

1. AUBRY, C. Objetivo: Web WordPress 3-un CMS para crear su sitio Web. Ediciones ENI, 2011. ISBN 2746068494.
2. BARRENO, B. Protocolo HTTPS [Página Web: Sitio Oficial]. SlideShare Incorporated, [Consultado el: abril de 2013]. Disponible en: <http://es.slideshare.net/babp/protocolo-https-3617275>.
3. BAUTISTA, D. C. y BAUTISTA, J. G. Servidor web [Página Web: Sitio Oficial]. SlideShare Incorporated, [Consultado el: febrero de 2013]. Disponible en: <http://es.slideshare.net/josegregoriob/servidor-web-8451426>.
4. BETH CHRISSIS, M.; KONRAD, M., et al. CMMI Guía para la integración de procesos y la mejora de productos. España: Pearson Educación Sociedad Anónima., 2009. ISBN 9788478290963.
5. CABELLO, V. N. Introducción a las Base de Datos Relacionales. Madrid: Visión Libros, 2008. 30-33 p. ISBN 987-84-9886-809-8.
6. CANÓS, J.; LETELIER, P., et al. Metodologías Ágiles en el desarrollo de Software. España: Universidad Politécnica de Valencia, 2003.
7. CAROLINA GIL MEZA, A. Protocolos de transferencia SlideShare Incorporated, [Consultado el: abril de 2013]. Disponible en: <http://es.slideshare.net/ingalegil/protocolos-de-transferencia>.
8. CASAS CERVERO, J. Content Management System. Trabajo de diploma, Facultad de Informática de Barcelona. Universidad Politécnica de Cataluña, 2011.
9. CHACA, J. Protocolos de red [Página web]. SlideShare Incorporated, [Consultado el: abril de 2013]. Disponible en: http://es.slideshare.net/j_chaca/protocolos-de-red.
10. CHRISSIS, M. B.; KONRAD, M., et al. CMMI Guía para la integración de procesos y la mejora de productos. España: Pearson Educación S.A., 2009. ISBN 9788478290963.

11. CÍRCULO DE PERIODISTAS CUBANOS Para webmasters y desarrolladores [Página web]. Cuba: Cubadebate, [Consultado el: febrero de 2013]. Disponible en: <http://www.cubadebate.cu/editores/webmasters/>.
12. COBO, Á. PHP y MySQL: Tecnología para el desarrollo de aplicaciones web. Ediciones Díaz de Santos, 2005. ISBN 8479787066.
13. CRAIG LARMAN. Modelo del Dominio. Prentice Hall, 2003.
14. DATE, C. J. Introducción a los sistemas de bases de datos. Pearson Educación, 2001. ISBN 9684444192.
15. DEPARTAMENTO DE EDUCACIÓN, P. L. Y. C. ¿Qué es la prensa escrita? [Página web]. España: Departamento de Educación, Universidades e Investigación del Gobierno Vasco, [Consultado el: mayo de 2013]. Disponible en: <http://www.hiru.com/medios-de-comunicacion/que-es-la-prensa-escrita>.
16. DONNELLY, M. Una Introducción a LDAP Plone Foundation, [Consultado el: abril de 2013]. Disponible en: http://softwarelibre.mes.edu.cu/index_html/centro-de-asistencia-tecnica/manual/una-introduccion-a-ldap/una-introduccion-a-ldap.
17. EDICIONES EL PAÍS, S. L. Corporativo [Página web]. España: El País, [Consultado el: mayo de 2013]. Disponible en: <http://elpais.com/corporativos/>.
18. FIGUEREDO, R. R. SfAspectPlugin extensión de Symfony para la programación orientada a aspectos. 2009.
19. GNU ORGANIZATION. Preguntas frecuentes acerca de las licencias de GNU [Página Web: Sitio Oficial]. Sistema Operativo GNU, Última actualización: 2013/04/13. [Consultado el: febrero de 2013]. Disponible en: <http://www.gnu.org/licenses/gpl-faq.es.html>.
20. GONZÁLEZ, M. A. C. Convivencia de la prensa escrita y la prensa on line en su transición hacia el modelo de comunicación multimedia. 2001, 71-78 p. ISBN 1134-1629.
21. GONZÁLEZ, M. A. C. Convivencia de la prensa escrita y la prensa on line en su transición hacia el modelo de comunicación multimedia. Estudios sobre el mensaje periodístico, 2001, nº 7, p. 71-78. ISSN 1134-1629.

22. GUTIÉRREZ, E. Conceptos Básicos y avanzados. Barcelona-España: 2009. 13 p. ISBN 978-2-7460-5220-8.
23. HEURTEL, O. Desarrollar un sitio web Dinámico e interactivo. Barcelona-España: ENI, 2011. ISBN 978-2-7460-6666-3.
24. LÓPEZ, P. Herramienta CASE Visual Paradigm. España: Universidad Cantabria – Facultad de Ciencias, 2007. 25 p.
25. MARTINEZ, R. R. y JIMÉNEZ, R. P. Prensa digital y Web 2.0 [Página Web]. España: Universitat Pompeu Fabra, Barcelona, Última actualización: 05-06-2012. [Consultado el: febrero de 2013].
26. MARTINEZ, S. G. y CAMPOS, A. V. Características de la Web 2.0 [Página Web: Sitio Web]. SlideShare Incorporated, [Consultado el: febrero de 2013]. Disponible en: <http://es.slideshare.net/SaraGuillenCampos/caracteristicas-de-la-web-20-14671985>.
27. MENA, C. M. C. Desarrollo del Sistema de Prensa Digital de la Universidad de las Ciencias Informáticas. Tutor: Hernández, I. C. G. y García, I. Y. D. Trabajo de diploma, Facultad 1. Universidad de las Ciencias Informáticas, 2012.
28. MUSCIANO C. Y KENNEDY B. HTML La Guía Completa. México: O'Reilly & Associates, Inc., 1999. 7-8 p. ISBN 1-6592-235.
29. ORACLE CORPORATION. ¿Qué es NetBeans? [Página Web]. NetBeans, [Consultado el: febrero de 2013]. Disponible en: https://netbeans.org/index_es.html.
30. ORJUELA, A. y ROJAS, M. Las metodologías de desarrollo ágil como una oportunidad para la ingeniería del software educativo. Avances en Sistemas e Informática, 2008, vol. 5, nº 2, p. 159-170. ISSN 1909-0056.
31. RIVERA, D. El ciberperiodismo en Ecuador. Casos de estudio La Hora y Crónica, periódicos de Loja. QUÓRUM ACADÉMICO, 2011, vol. 8, nº 16, p. 12. Disponible en: <http://dialnet.unirioja.es/servlet/articulo?codigo=3865226>. ISSN 1690-7582.
32. SALAVERRÍA, R. Redacción periodística en Internet. España: Eunsa Pamplona, 2005. ISBN 8431322594.

33. SCHULZ, R. G. Diseño Web con CSS. Barcelona-España: 2008. 263 p. ISBN 978-3-8266-1616-2.
34. SOCIEDAD ANÓNIMA INFORMACION Y CONTROL DE PUBLICACIONES. [Página web]. España: El Mundo, [Consultado el: abril de 2013]. Disponible en: <http://www.elmundo.es/>.
35. SOFTNEWS NET SRL. Descripción de Visual Paradigm for UML Community Edition [Página Web: Sitio Oficial]. Softpedia, [Consultado el: febrero de 2013]. Disponible en: <http://www.softpedia.es/programa-Visual-Paradigm-for-UML-Community-Edition-4164.html>.
36. TURNER BROADCASTING SYSTEM INCORPORATED. [Página web]. Cable News Network (CNN), [Consultado el: mayo de 2013]. Disponible en: <http://cnnespanol.cnn.com/>.
37. VAN LANCKER, L. HTML5-Los fundamentos del lenguaje. Ediciones ENI, 2012. ISBN 2746075164.
38. VELASCO, S. y MARTÍN, A. Creación de scripts en Linux. España: Universidad Complutense, 2008, 32 p. Disponible en: <http://pendientedemigracion.ucm.es/info/aulasun/archivos/SCRIPTS.pdf>.
39. WORDPRESS ORGANIZATION. Description WordPress, [Consultado el: febrero de 2013]. Disponible en: http://codex.wordpress.org/Function_Reference/query_posts.

ANEXOS

Anexo 1: fichas de historias de usuarios.

Tabla 16: Historia de usuario “Obtener la cantidad de visitas por categoría”

HU3- HU Obtener la cantidad de visitas por categoría.

Historia de Usuario	
Número: HU3	Nombre Historia de Usuario: HU3- HU Obtener la cantidad de visitas por categoría.
Modificación de Historia de Usuario Número: 1	
Referencia:	
Programador: Pedro Luis León Fundora	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alta	Puntos Reales: 1
Descripción: Luego se procede a obtener la cantidad de visitas que se realizaron por parte de un usuario a cada categoría.	
Observaciones: El usuario debe estar previamente autenticado en el sistema.	
Prototipo de interfaz:	

Tabla 17: Historia de usuario “Guardar historial de visitas”

HU4- HU Guardar historial de visitas por categoría.

Historia de Usuario	
Número: HU4	Nombre Historia de Usuario: HU4- HU Guardar historial de visitas por categoría.
Modificación de Historia de Usuario Número: 1	
Referencia:	
Programador: Pedro Luis León Fundora	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alta	Puntos Reales: 1

Descripción: A continuación se procede a almacenar en la tabla registro cada usuario del sistema junto con las categorías que visitó.

Observaciones: El usuario debe estar previamente autenticado en el sistema.

Prototipo de interfaz:

Tabla 18: Historia de usuario “Definir categoría favorita”

HU5- HU Definir categoría favorita del usuario.

Historia de Usuario	
Número: HU5	Nombre Historia de Usuario: HU5- HU Definir categoría favorita del usuario.
Modificación de Historia de Usuario Número: 1	
Referencia:	
Programador: Pedro Luis León Fundora	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alta	Puntos Reales: 1
Descripción: Apoyándose en los datos almacenados en la Base de Datos se procede a obtener la categoría más visitada por el usuario, denominándose esta categoría favorita, luego se pasa a almacenar en la tabla histórico, de forma tal que solo quede un único usuario junto con su categoría favorita.	
Observaciones: El usuario debe estar previamente autenticado en el sistema.	
Prototipo de interfaz:	

Tabla 19: Historia de usuario “Mostrar noticias”

HU6- HU Mostrar noticias.

Historia de Usuario	
Número: HU6	Nombre Historia de Usuario: HU6- HU Mostrar noticias.
Modificación de Historia de Usuario Número: 1	
Referencia:	
Programador: Pedro Luis León Fundora	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1

Riesgo en Desarrollo: Alta	Puntos Reales: 1
Descripción: La aplicación debe mostrar a cada usuario autenticado en el sistema, solamente las noticias nuevas de la categoría que se le definió como favorita.	
Observaciones: El usuario debe estar previamente autenticado en el sistema.	
Prototipo de interfaz:	

Tabla 20: Historia de usuario “Actualizar categoría favorita”

HU7- HU Actualizar categoría favorita.

Historia de Usuario	
Número: HU7	Nombre Historia de Usuario: HU7- HU Actualizar categoría favorita.
Modificación de Historia de Usuario Número: 1	
Referencia:	
Programador: Pedro Luis León Fundora	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alta	Puntos Reales: 1
Descripción: La aplicación cada cierto tiempo actualizará la categoría de cada usuario, pues esta puede cambiar por parte de los consumidores.	
Observaciones: El usuario debe estar previamente autenticado en el sistema.	
Prototipo de interfaz:	

Tabla 21: Historia de usuario “Establecer tiempo”

HU8- HU Establecer tiempo.

Historia de Usuario	
Número: HU8	Nombre Historia de Usuario: HU8- HU Establecer tiempo.
Modificación de Historia de Usuario Número: 1	
Referencia:	
Programador: Pedro Luis León Fundora	Iteración Asignada: 1
Prioridad en Negocio: Media	Puntos Estimados: 1
Riesgo en Desarrollo: Media	Puntos Reales: 1

Descripción: El administrador tiene la opción de escoger el tiempo que desea que el componente trabaje sobre los usuarios en general, mediante la cantidad de acciones que los mismos vayan realizando en el sistema.

Observaciones: El administrador de estar previamente autenticado en el sistema.

Prototipo de interfaz:

Definir aprendizaje del plugin según las acciones del usuario en el sistema:

- Predeterminado
- 10 acciones
- 50 acciones
- 100 acciones
- 1000 acciones

Activar

Tabla 22: Historia de usuario "Establecer el tipo de rol"

HU9- HU Establecer el tipo de rol.

Historia de Usuario	
Número: HU9	Nombre Historia de Usuario: HU9- HU Establecer el tipo de rol.
Modificación de Historia de Usuario Número: 1	
Referencia:	
Programador: Pedro Luis León Fundora	Iteración Asignada: 1
Prioridad en Negocio: Media	Puntos Estimados: 1
Riesgo en Desarrollo: Media	Puntos Reales: 1
Descripción: El administrador tiene la opción de escoger el tipo de usuario sobre el cual desea que trabaje el componente.	
Observaciones: El administrador de estar previamente autenticado en el sistema.	
Prototipo de interfaz:	

Especificar el tipo de usuario sobre el cual va a trabajar el plugin:

Suscriptor
 Colaborador
 Autor
 Editor
 Administrador

Tabla 23: Historia de usuario “Restablecer un usuario a valores predeterminados”

HU10- HU Restablecer un usuario a valores predeterminados.

Historia de Usuario	
Número: HU10	Nombre Historia de Usuario: HU10- HU Restablecer un usuario a valores predeterminados.
Modificación de Historia de Usuario Número: 1	
Referencia:	
Programador: Pedro Luis León Fundora	Iteración Asignada: 1
Prioridad en Negocio: Baja	Puntos Estimados: 1
Riesgo en Desarrollo: Baja	Puntos Reales: 1
Descripción: El administrador puede restablecer los valores de un usuario en específico a predeterminados.	
Observaciones: El administrador de estar previamente autenticado en el sistema.	
Prototipo de interfaz:	
Restablecer los valores a predeterminados de un usuario específico:	
<input type="text" value="usuario"/> <input type="button" value="Resetear"/>	

Tabla 24: Historia de usuario “Restablecer a valores predeterminados en general”

HU11- HU Restablecer a valores predeterminados en general.

Historia de Usuario	
Número: HU11	Nombre Historia de Usuario: HU11- HU Restablecer a valores predeterminados en general.
Modificación de Historia de Usuario Número: 1	

Referencia:	
Programador: Pedro Luis León Fundora	Iteración Asignada: 1
Prioridad en Negocio: Baja	Puntos Estimados: 1
Riesgo en Desarrollo: Baja	Puntos Reales: 1
Descripción: El administrador puede restablecer los valores a predeterminados de todos los usuarios del sistema.	
Observaciones: El administrador de estar previamente autenticado en el sistema.	
Prototipo de interfaz:	
Restablecer los valores a predeterminados de toda la base de datos:	
<input type="button" value="Resetear"/>	

Anexo 2: casos de prueba

Tabla 25: Caso de prueba “Obtener cantidad de visitas por categoría de un usuario”

Caso de Prueba de Aceptación	
Código: HU3_P1	Historia de Usuario: HU3- HU Obtener la cantidad de visitas por categoría.
Nombre: Obtener cantidad de visitas por categoría de un usuario.	
Descripción: Probar que se obtienen correctamente la cantidad de visitas que realiza un usuario a cada una de las categorías.	
Condiciones de Ejecución: La aplicación debe ejecutarse con privilegios de suscriptor.	
Entrada / Pasos de Ejecución: El usuario debe navegar por el sistema.	
Resultado Esperado: El sistema debe obtener la cantidad de visitas por categoría del usuario autenticado en el sistema.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 26: Caso de prueba “Guardar historial”

Caso de Prueba de Aceptación	
Código: HU4_P1	Historia de Usuario: HU4- HU Guardar historial de visitas por categoría.
Nombre: Guardar historial.	
Descripción: Probar que se almacenen correctamente en la base de datos el historial de visitas por categoría.	
Condiciones de Ejecución: La aplicación debe ejecutarse con privilegios de suscriptor.	

Entrada / Pasos de Ejecución: El usuario debe navegar por el sistema.
Resultado Esperado: Las visitas deben ser almacenadas por categoría en la tabla registro de la base de datos.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 27: Caso de prueba “Definir categoría favorita”

Caso de Prueba de Aceptación	
Código: HU5_P1	Historia de Usuario: HU5- HU Definir categoría favorita del usuario.
Nombre: Definir categoría favorita.	
Descripción: Probar que se define correctamente la categoría favorita del usuario autenticado en el sistema.	
Condiciones de Ejecución: La aplicación debe ejecutarse con privilegios de suscriptor.	
Entrada / Pasos de Ejecución: El usuario debe navegar por el sistema.	
Resultado Esperado: El sistema debe identificar la categoría favorita del usuario suscriptor autenticado en el sistema utilizando los datos guardados en la tabla registro de la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 28: Caso de prueba “Mostrar noticias”

Caso de Prueba de Aceptación	
Código: HU6_P1	Historia de Usuario: HU6- HU Mostrar noticias.
Nombre: Mostrar noticias.	
Descripción: Probar que las noticias de la categoría preferida del usuario autenticado en el sistema se muestren correctamente en la página principal de la aplicación.	
Condiciones de Ejecución: La aplicación debe ejecutarse con privilegios de suscriptor.	
Entrada / Pasos de Ejecución: El usuario debe navegar por el sistema.	
Resultado Esperado: Las noticias pertenecientes a la categoría predilecta del usuario autenticado deben ser mostradas en la página principal del sistema.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 29: Caso de prueba “Actualizar categoría”

Caso de Prueba de Aceptación	
Código: HU7_P1	Historia de Usuario: HU7- HU Actualizar categoría favorita.
Nombre: Actualizar categoría.	
Descripción: Probar que la categoría favorita de un usuario autenticado en el sistema se actualice correctamente según sus acciones en la aplicación.	
Condiciones de Ejecución: La aplicación debe ejecutarse con privilegios de suscriptor.	
Entrada / Pasos de Ejecución: El usuario debe navegar por el sistema.	
Resultado Esperado: La categoría favorita del usuario se debe actualizar en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 30: Caso de prueba “Establecer tiempo”

Caso de Prueba de Aceptación	
Código: HU8_P1	Historia de Usuario: HU8- HU Establecer tiempo.
Nombre: Establecer tiempo.	
Descripción: Probar que se establezca correctamente el tiempo de aprendizaje del componente sobre los usuarios autenticados en el sistema.	
Condiciones de Ejecución: La aplicación debe ejecutarse con privilegios de administrador.	
Entrada / Pasos de Ejecución: Se selecciona una opción.	
Resultado Esperado: El componente no comenzará a mostrar las noticias de la categoría favorita del usuario suscriptor registrado en el sistema, hasta que se cumpla la condición seleccionada.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 31: Caso de prueba “Establecer rol”

Caso de Prueba de Aceptación	
Código: HU9_P1	Historia de Usuario: HU9- HU Establecer el tipo de rol.
Nombre: Establecer rol.	
Descripción: Probar que se definan los tipos de rol correctamente sobre el cual el componente va a accionar.	
Condiciones de Ejecución: La aplicación debe ejecutarse con privilegios de administrador.	
Entrada / Pasos de Ejecución: Se selecciona los tipos de roles de usuario sobre el cual va a trabajar el componente.	
Resultado Esperado: El sistema debe almacenar y mostrar datos, solamente de los usuarios	

autenticados que pertenezcan al rol seleccionado.

Evaluación de la Prueba: Prueba satisfactoria.

Tabla 32: Caso de prueba “Restablecer a valores predeterminados un usuario”

Caso de Prueba de Aceptación	
Código: HU10_P1	Historia de Usuario: HU10- HU Restablecer un usuario a valores predeterminados.
Nombre: Restablecer a valores predeterminados un usuario.	
Descripción: Probar que se restablezcan a predeterminados los datos de un usuario dado.	
Condiciones de Ejecución: La aplicación debe ejecutarse con privilegios de administrador.	
Entrada / Pasos de Ejecución: El administrador debe insertar el usuario del consumidor que se desea restablecer a valores predeterminados y luego accionar en el botón de resetear.	
Resultado Esperado: El sistema debe restablecer a valores predeterminados al usuario insertado.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 33: Caso de prueba “Restablecer a valores predeterminados un usuario”

Caso de Prueba de Aceptación	
Código: HU11_P1	Historia de Usuario: HU11- HU Restablecer a valores predeterminados en general.
Nombre: Restablecer a valores predeterminados un usuario.	
Descripción: Probar que se restablezcan a predeterminados todos los datos de los usuarios del sistema	
Condiciones de Ejecución: La aplicación debe ejecutarse con privilegios de administrador.	
Entrada / Pasos de Ejecución: El administrador debe accionar el botón de resetear de la opción de restablecer a valores predeterminados general.	
Resultado Esperado: Los datos de todos los usuarios en la base de datos deben restablecerse a valores predeterminados.	
Evaluación de la Prueba: Prueba satisfactoria.	