



Trabajo de diploma para optar por el título de Ingeniero en
Ciencias Informáticas



**Herramienta para la actualización de blogs en Wordpress sin
acceder a la administración en línea desde sistemas GNU/Linux.**

Autor

Vladis Miriel Perdomo Hernández

Tutores

Ing. Lilian Cid Escalona

Ing. Juan Manuel Álvarez Tur

La Habana, junio de 2015

Universidad de las Ciencias Informáticas

DECLARACIÓN DE AUTORÍA

Declaro ser el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a que haga el uso que estime pertinente con el mismo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2015.

Firma del Autor
Vladis Miriel Perdomo Hernández

Firma del Tutor
Lilian Cid Escalona

Firma del Tutor
Juan Manuel Álvarez Tur

DATOS DE CONTACTO

Ing. Lilian Cid Escalona

Correo electrónico: lescalona@uci.cu

Graduada en Ciencias Informáticas (2009). Al graduarse se vinculó a la docencia en la propia universidad durante tres cursos impartiendo asignaturas de la Disciplina de Ciencias Empresariales como Comercio Electrónico y Contabilidad y Finanzas. Perteneció al proyecto productivo SIGEP-Cuba donde se desempeñó como Administrador de Calidad y fue responsable en el despliegue piloto del producto CEDRUX en el ICID que llevó el Departamento de Soluciones Empresariales del Centro para la Informatización de la Gestión de Entidades (CEIGE) de la Facultad 3 Actualmente se desempeña como especialista del Departamento de Operaciones Web y Análisis de Información (DOWAI) del Centro de Ideoinformática (CIDI).

Ing. Juan Manuel Álvarez Tur

Correo electrónico: jmalvarez@uci.cu

Ing. Juan Manuel Álvarez Tur. Graduado en Ciencias Informáticas (2011). Ha trabajado en proyectos de desarrollo y despliegue de software como el sistema Smart Keeper y el sitio web Catálogo Promocional para el Campismo Popular. Ha participado en eventos científicos nacionales e internacionales y publicados en revistas referenciadas.

Autor del Trabajo: Vladis Miriel Perdomo Hernández

Correo electrónico: vladis91@nauta.cu



"El futuro tiene muchos nombres. Para los débiles es lo inalcanzable. Para los temerosos, lo desconocido. Para los valientes es la oportunidad."

Victor Hugo

Dedicatoria

Dedico el presente trabajo especialmente a mi madre querida, por darme la vida, su amor y su apoyo incondicional para lograr este sueño que hoy se hace realidad, por su entrega y dedicación, por confiar en mí y por forjarme como un hombre de bien, por darme una carrera digna para mi futuro y que también le pertenece.

TE QUERO MUCHO MAMA.

AGRADECIMIENTOS

A mi madre por ser la mejor de este mundo, por ser tan valiente y decidida, por estar siempre ahí, por salir adelante aun cuando las cosas iban mal, por lograr lo que soy, te quiero mamá.

A mi padre por estar siempre presente a pesar de la distancia, por brindarme su amor y cariño, por su tolerancia, por sus consejos, gracias por todo pito, te quiero.

A la memoria de mi abuela Mercedes, que no logró ver su primer nieto ingeniero, donde quiera que esté te queremos mucho.

A mi abuelo Héctor, por ser como mi segundo padre, por estar siempre a mi lado y por todo el apoyo que me ha dado, te quiero papá.

A toda mi familia que de una forma u otra me apoyaron, a mi hermano por todo su apoyo incondicional, por ser una de las personas que más quiero en mi vida, te quiero mamá.

A mi hermanita lelita, que a pesar de estar lejos está muy cerca, te quiero peluca.

A mi prima Yane que es como una hermana para mí, gracias por todo, gracias por siempre estar presente, a mi prima Lisi, ustedes fueron parte de este sueño, las adoro mis feas.

A mi primo Juan Carlos por sus verdades directas y concretas pero muy constructivas, por su ayuda, por todo gracias primo.

A quien durante estos 5 años ha significado como una segunda madre para mí, gracias Ana por tu apoyo, por tu ayuda, por cuidarme, por tenerme siempre presente y por darme a mis hermanitas que tanta alegría nos transmiten a diario.

A mi profe Pílciner, que a pesar de compartir poco tiempo con ella fue mi profesora, pero sobre todo mi amiga, consejera y guía, muchas gracias profe.

A todas mis amistades, gracias, sin ustedes no lo hubiese logrado, gracias a Lorea y Ernesto por ser más que amigos como hermanos, por siempre estar ahí ustedes son lo máximo.

Gracias, a yuyo por sus comentarios, consejos y sinceridad, a Betty por apoyarme en los momentos difíciles y por ser la mejor amiga que tengo, gracias putí.

A Jorge Alberto por la calma conservada y su ayuda, a mis tutores por su preocupación durante todo el proceso, a Elián por toda su preocupación y ayuda.

A Yoelkís por su tiempo, ayuda y consejos, a Laura, Carlos Jordán por ser de esos amigos que serán inolvidables, Abel, Raul, Margarita, a todos gracias por siempre estar presente.

A todos mis compañeros de aula, a todos mis profesores, Olivia, Dasiel, Alíen. Alejandro, Damian, Javier, Yaumara, Niurvis, Gendri, Antonio en fin a todos los que de una forma u otra contribuyeron a mi aprendizaje y por los cuales hoy soy ingeniero.

A la Revolución por darme la oportunidad de estudiar, de ser mejor persona, por permitirme ser parte de este sueño que es la UCI y que tantas cosas me ha enseñado durante estos 5 años.

RESUMEN

Los blogs han evolucionado hasta convertirse en una herramienta de expresión y en la actualidad constituye una popular forma de comunicarse y compartir información. Desde el Departamento de Operaciones Web y Análisis de Información del Centro de Ideoinformática, adscrito a la Facultad 1 de la Universidad de las Ciencias Informáticas se administran tres blogs alojados en la plataforma internacional Wordpress con el fin de potenciar la divulgación de la realidad cubana en Internet. La intención fundamental de este trabajo es desarrollar una herramienta informática sin conexión que posibilite la actualización de los blogs y de esa forma evitar ingresar constantemente a los servidores en línea donde se alojan los mismos. El proceso de desarrollo fue guiado por la metodología ágil OpenUP y la implementación se basó en lenguajes como Java para la programación y Lenguaje Unificado de Modelado para construir y documentar el sistema. Además se utilizaron tecnologías y herramientas como el GIMP para el tratamiento de imágenes, NetBeans como Entorno de Desarrollo Integrado, VisualParading como herramienta para el modelado, XML-RPC como protocolo de comunicación, Ekit como procesador de textos adaptado al sistema desarrollado y Junit para las pruebas automatizadas al software siguiendo como principio desarrollar software libre para uso de todos. Para verificar el cumplimiento de las funcionalidades propuestas se realizaron varias pruebas de software clasificadas en unitarias y funcionales.

Palabras claves: Wordpress, blogs, software libre, herramienta, actualización.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
Introducción	6
1.1 Conceptos y definiciones asociadas a la investigación	6
1.2 Herramientas existentes para la actualización de blogs.	7
1.2.1 Extensiones de Firefox.....	7
1.2.2 Aplicaciones de escritorio.....	8
1.2.3 Análisis comparativo de las herramientas estudiadas.	10
1.3 Metodología de desarrollo.....	11
1.3.1 Metodologías Ágiles	12
1.4 Tecnologías y herramientas seleccionadas para el desarrollo del software.....	16
Conclusiones parciales.....	21
CAPÍTULO 2: ANÁLISIS Y DISEÑO	22
Introducción	22
2.1 Propuesta de solución.....	22
2.2 Modelo de Dominio.....	22
2.3 Especificación de los Requisitos del Sistema.....	23
2.3.1 Técnicas de Obtención de Requisitos	23
2.3.2 Requisitos Funcionales (RF).....	24
2.3.3 Requisitos No Funcionales (RNF).....	25
2.3.4 Descripción de los Requisitos Funcionales.....	26
2.4 Modelo de Casos de Uso del Sistema	31
2.4.1 Definición de los actores del sistema.....	31
2.4.2 Descripción de los Casos de Uso del Sistema.....	32
2.5 Diseño del Sistema	39
2.5.1 Patrón Arquitectónico	39
2.5.2 Patrones de Diseño.....	41
2.5.3 Diagrama de Clases del Diseño	43
Conclusiones parciales.....	45

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA.....	46
Introducción	46
3.1 Diagrama de Componentes	46
3.2 Diagrama de Despliegue	47
3.3 Estilos de programación	47
3.4 Estándares de codificación	48
3.5 Estrategia de Prueba	50
3.5.1 Métodos de Prueba.....	51
Conclusiones parciales	60
CONCLUSIONES GENERALES.....	61
RECOMENDACIONES	62
REFERENCIAS.....	63
ANEXOS	66
GLOSARIO DE TÉRMINOS.....	68

ÍNDICE DE FIGURAS

Figura 1 Fases de la Metodología OpenUP	15
Figura 2 Arquitectura del protocolo XML-RPC.....	19
Figura 3 Modelo de Dominio	23
Figura 4 Diagrama de Casos de Uso del Sistema.....	31
Figura 5 Diagrama de la Arquitectura.....	40
Figura 6 Patrón Experto.....	41
Figura 7 Patrón Creador	42
Figura 8 Diagrama de Clases del Diseño.....	44
Figura 9 Diagrama de Componentes	46
Figura 10 Diagrama de Despliegue del Sistema	47
Figura 11 Identación y llaves.....	48
Figura 12 Nomenclatura para crear variables	48
Figura 13 Nomenclatura para crear las clases.	49
Figura 14 Nomenclatura para operadores aritméticos.....	49
Figura 15 Documentación de la clase controladora WP_Blog_Edit.....	49
Figura 16 Documentación de la funcionalidad Lista de Categorías	49
Figura 17 Estructuras de control	50
Figura 18 Resultado de las pruebas unitarias para la clase controladora.....	57
Figura 19 Resultado de las pruebas unitarias para la interfaz Establecer Conexión.	57
Figura 20 Resultado de las no conformidades por iteraciones.	59

INDICE DE TABLAS

Tabla 1 Comparación de herramientas.....	10
Tabla 2 Comparación de metodologías	11
Tabla 3 Descripción del requisito funcional Autenticar usuario en el blog	26
Tabla 4 Descripción del requisito funcional Gestionar Entrada.....	28
Tabla 5 Actores del sistema	32
Tabla 6 Descripción del CU 1 Autenticar en el Blog.....	32
Tabla 7 Descripción CU 2: Gestionar Entrada	33
Tabla 8 Diseño del CP Autenticar Usuario.....	52
Tabla 9 Diseño del CP Crear y Modificar Entrada del CU Gestionar Entrada	53
Tabla 10 Diseño del CP Obtener Entrada del CU Gestionar Entrada	55
Tabla 11 Diseño de CP Eliminar Entrada del CU Gestionar Entrada	56
Tabla 12 Resultado de las pruebas de Caja Negra en la Iteración 1	57
Tabla 13 Resultado de las pruebas de Caja Negra en la Iteración 2	58
<i>Tabla 14 Relación de los entrevistados durante el proceso de desarrollo.</i>	<i>66</i>

INTRODUCCIÓN

En la actualidad, la tecnología juega un papel primordial en las comunicaciones y la forma de difundir la información. El uso de las Tecnologías de la Informática y las Comunicaciones (TIC) ha representado una diversificación en la sociedad, en las relaciones interpersonales y en la forma de difundir y generar conocimiento. A través de este importante desarrollo tecnológico se ha creado un universo de cambios donde se combinan la interactividad y la forma de compartir información.

Una de las formas de compartir información es mediante los blogs, donde la explosión de millones de estos, puebla hoy la red de redes convirtiéndose en uno de los medios más populares de la web social. Los blogs “son un medio de comunicación colectivo que promueve la creación y consumo de información original y diversa, y que provoca, con mucha eficiencia, la reflexión personal y social sobre los temas de los individuos, de grupos y de la humanidad” (1). Estos, presentan un formato de publicación en línea que se centra en los contenidos y en los usuarios. Se encuentran estructurados originalmente como diarios personales, basados en vínculos, noticias y comentarios que se actualizan de modo regular. En la actualidad su proliferación se ha convertido en un avance mundial, ya que se convierten en una alternativa de comunicación e información personalizada.

La estructura de los blogs se caracteriza por poseer un formato específico. En la parte principal de la página aparecen los últimos artículos publicados, generalmente incluyen enlaces a otros blogs, pero esto es de acuerdo a las preferencias del usuario, junto a ellos pueden aparecer los aportes de los visitantes (comentarios), ya sea directamente o mediante vínculos o consumo de “Sindicación Realmente Simple” por sus siglas en inglés RSS. Están organizados cronológicamente y es posible consultar los temas tratados con anterioridad, que por lo general, están organizados en orden cronológico decreciente y también es posible buscar en los archivos mediante algún motor de búsqueda.

Para facilitar el trabajo con la edición y publicación en los blogs existen herramientas informáticas como los editores sin conexión que son la vía más factible para escribir o actualizar una entrada si se presentan dificultades cuando la conexión a Internet es limitada o nula, el editor del servidor no es eficiente o simplemente no se desea ingresar constantemente al servidor en línea donde se aloja el blog. La mayoría de los editores de blogs sin conexión, tienen características comunes tales como: escribir y editar nuevas entradas, corrección ortográfica, gestión de los contenidos, previsualización de los artículos, alineación de

texto, estilos de fuente ajustable, entre otras. Incluso hay algunos que también administran notificaciones de Twitter o pueden publicar en varias plataformas de blogs una vez que se conectan a Internet.

El Departamento de Operaciones Web y Análisis de Información (DOWAI) del Centro de Ideoinformática (CIDI), adscrito a la Facultad 1 de la Universidad de las Ciencias Informáticas (UCI) tiene entre sus misiones potenciar en Internet una imagen más inclusiva de la realidad cubana, en contraposición con el modelo antagónico que intentan mostrar sectores que se oponen al régimen social vigente. A fin de potenciar la existencia de contenidos sobre diversas aristas del acontecer cubano, desde DOWAI se administran tres blogs alojados en la plataforma internacional Wordpress (wordpress.com). Para la actualización de estos blogs, los especialistas encargados de la edición y administración se auxiliaban de Windows Live Writer, una aplicación de escritorio para realizar estas acciones cuyo uso permite evitar los problemas de conectividad para el acceso a los servicios de administración en línea de Wordpress que se presentan en la UCI. En esos problemas de conectividad se comprenden la intermitencia en la disponibilidad de acceso a Internet o su ausencia prolongada, así como la velocidad de la conexión.

En la actualidad, Cuba busca insertarse en las nuevas corrientes tecnológicas, regidas en parte por el principio de desarrollar software libre para uso de todos. En aras de apoyar al país y favorecer la soberanía tecnológica, la UCI decidió migrar sus estaciones de trabajo a software libre, lo cual no permite hacer uso de las posibilidades que ofrece Windows Live Writer, al ser esta una herramienta privativa de Microsoft que no se emula en sistemas GNU/Linux. La situación afecta el proceso de actualización de los blogs ya que como se explicó antes, la creación de las publicaciones en la plataforma en línea de Wordpress está altamente condicionada por factores relacionados con la velocidad de conexión y la disponibilidad de las cuotas de Internet o acceso en general.

En un estudio realizado por especialistas de DOWAI para encontrar alternativas a Windows Live Writer en sistemas GNU/Linux, se detectó que ninguna de las herramientas encontradas cumplía los requisitos que exige el trabajo de los editores del departamento, por tanto, no se tienen referencias de una aplicación que sustituya al Windows Live Writer en su función de facilitar, a través de su capacidad como editor WYSIWYG¹ la actualización de los contenidos en los blogs administrados desde el centro CIDI.

¹ Acrónimo de What You See Is What You Get traducido al español “Lo que tú ves es lo que obtienes” y caracteriza a editores en los que de manera visual se pueden colocar distintos elementos sobre una vista previa de la página.

Teniendo en cuenta la **situación problemática** antes planteada se define el siguiente **problema de la investigación**: ¿Cómo favorecer la actualización de los blogs del departamento DOWAI sin acceder a la administración en línea desde sistemas GNU/Linux?

El **objeto de estudio** se centra en las herramientas sin conexión para la actualización de blogs en Wordpress, mientras que el **campo de acción** se enmarca en las herramientas sin conexión para la actualización de blogs en Wordpress en sistemas GNU/Linux.

Como **idea a defender** se tiene que la implementación de una herramienta sin conexión para la edición de blogs en Wordpress permitirá mejorar el proceso de actualización de los blogs administrados por DOWAI.

Se plantea como **objetivo general** desarrollar una herramienta que permita la actualización de los blogs del departamento DOWAI sin acceder a la administración en línea desde sistemas GNU/Linux.

El objetivo general se desglosa en los siguientes **objetivos específicos**:

1. Analizar las herramientas disponibles para la actualización de blogs sin acceder a la administración en línea.
2. Diseñar una herramienta para la actualización de los blogs administrados por el departamento DOWAI.
3. Implementar una herramienta sobre la base del diseño propuesto.
4. Validar la herramienta desarrollada.

A partir de los objetivos específicos planteados se planifican las siguientes **tareas de la investigación**:

1. Caracterización de las herramientas disponibles para la actualización de blogs sin acceder a la administración en línea.
2. Selección de las herramientas, metodologías y tecnologías a utilizar en el proceso de desarrollo del producto de software.
3. Obtención de los artefactos generados según la metodología de desarrollo aplicada para la realización de la propuesta de solución.
4. Validación de los requerimientos funcionales y no funcionales para el desarrollo de la solución aplicando las técnicas definidas.
5. Implementación de las funcionalidades definidas para cumplir con el objetivo planteado.
6. Aplicación de pruebas de funcionalidad a la solución desarrollada.

Una vez realizadas las tareas antes expuestas se espera obtener como **posible resultado**: una herramienta que permita la actualización de los blogs administrados por el departamento DOWAI, sin acceder a la

administración en línea.

Para dar cumplimiento a las tareas de la investigación planteadas se emplearon los siguientes **métodos científicos**:

1. Métodos teóricos:

- a) **Analítico-sintético**: Haciendo uso de este método científico se investigó la información relacionada con la edición de blogs, las herramientas usadas a nivel nacional e internacional, sus aspectos y relaciones, y se analizaron documentos que permitieron obtener datos necesarios para entender la esencia de los elementos que se describen en este trabajo.
- b) **Modelación**: Durante el diseño de la solución propuesta se realizaron varios modelados, tanto para entender el problema y la solución en sí, como para el diseño de una aplicación de escritorio y su posterior implementación.

2. Métodos Empíricos:

- a) El método de **observación científica** y la **medición** se emplean para la identificación del problema, el estudio objetivo del estado del arte, el diagnóstico y la evaluación de la aplicación.
- b) **Entrevista**: Este tipo de método es muy importante para tener los elementos necesarios para el desarrollo del producto que se quiere elaborar. En este sentido se realizará una entrevista a los responsables de los tres blogs administrados en el departamento DOWAI para conocer cómo funciona el proceso de publicación y actualización de artículos y así definir todos los requerimientos necesarios para que el sistema satisfaga sus necesidades.
- c) El método **coloquial** se emplea para la presentación y discusión de los resultados en sesiones científicas (Seminarios de Tesis, Jornada Científica, Pre-defensa y Defensa).

3. Procedimientos Científicos:

- a) El procedimiento de **análisis** se emplea para la descomposición del objeto de estudio en partes y propiedades para el estudio de sus relaciones y componentes. Se complementa con la síntesis en la sistematización de los conocimientos mediante el entendimiento y descubrimiento de las relaciones esenciales y características generales entre las partes previamente analizadas.
- b) El procedimiento de la **abstracción** se emplea para la comprensión del objeto de estudio.

El documento está estructurado como sigue:

En el Capítulo 1: se hace un estudio acerca de las herramientas existentes para la actualización sin conexión de blogs a través del uso de sistemas GNU/Linux. Se presentan los conceptos y definiciones que ayudan al desarrollo y comprensión de la solución propuesta. Además se justifican las herramientas seleccionadas para la implementación de la misma.

En el Capítulo 2: se describe y desglosa la solución desde la ingeniería de software, describiendo el ciclo seguido desde el modelo de dominio hasta la etapa de diseño. Además se mencionan y relacionan las herramientas que se van a emplear en la realización del trabajo.

En el Capítulo 3: son expuestos los resultados de las etapas de implementación y prueba, brindando una panorámica de las técnicas de programación utilizadas, su funcionamiento, descripción de las clases y operaciones sobre el lenguaje escogido. También se relacionan los resultados de las pruebas realizadas a la aplicación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

Introducción

En el presente capítulo son presentados los conceptos y definiciones que permiten sentar las bases teóricas para la investigación. Se realiza un estudio sobre el surgimiento de los blogs y las herramientas sin conexión existentes para su actualización desde sistemas GNU/Linux. Se expone una breve descripción de las herramientas, marco de trabajo, metodologías de desarrollo de software y lenguaje de programación a utilizar.

1.1 Conceptos y definiciones asociadas a la investigación

Se relaciona una serie de definiciones que servirá como base para entender el desarrollo de la herramienta, formando así el marco teórico de la investigación.

Blogs:

Los blogs son un medio de comunicación colectivo que promueve la creación y consumo de información original y veraz, y que provoca, con mucha eficiencia, la reflexión personal y social sobre los temas de los individuos, de grupos y de la humanidad (1).

Un blog, desde un punto de vista técnico, no es más que una página web, en la que el sistema de edición y publicación se ha simplificado hasta el punto que el usuario no necesita conocimientos específicos del medio electrónico ni del formato digital para poder aportar contenidos de forma inmediata, ágil y permanente, desde cualquier punto de conexión a internet (2).

Un blog es una jerarquía de textos, imágenes, objetos multimedia y datos ordenados cronológicamente que pueden ser vistos a través de un navegador (3).

Luego de analizar los conceptos asociados a los blog, el autor de esta investigación lo define como un conjunto de entradas organizadas cronológicamente y escritas por un individuo, un grupo o un negocio donde el contenido del blog es completamente determinado por el (los) autor(es).

Wordpress:

WordPress es un sistema de gestión de blogs desarrollado con PHP y MySQL-, bajo licencia GPL. Fue creado por Matt Mullenweg a partir de otro sistema, el b2/cafelog y se ha convertido en un software muy popular por su licencia, su facilidad de uso y sus características como gestor de contenidos (4).

Se puede definir además como un sistema de gestión de contenidos que permite crear y mantener un blog

u otro tipo de web. Posee temas y plantillas que sirven de ayuda a la hora de crear un blog, facilitando así a los usuarios, que no poseen conocimientos técnicos (5).

Protocolo:

Los protocolos son “reglas de comunicación que permiten el flujo de información entre equipos que manejan lenguajes distintos o como más sencillo, un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red por medio de intercambio de mensajes” (6),

Los protocolos son reglas o estándares que sirven para controlar y permitir la comunicación de una manera más simple entre computadoras. Los protocolos pueden ser definidos como las reglas que dominan la sintaxis, semántica y sincronización de la comunicación. Los protocolos pueden ser implementados por hardware, software, o una combinación de ambos (7).

Luego de analizar estos conceptos, el autor de esta investigación define que un protocolo no es más que un conjunto de normas, acuerdo o reglas que determinan el formato y la transmisión de datos de forma tal que la comunicación en una red sea más eficiente.

Editores sin conexión:

Son herramientas que permiten gestionar el contenido de un blog cuando no se tiene conexión a Internet, es decir permiten hacer todo el trabajo de edición con las mismas funcionalidades del editor en línea de la plataforma de blog, posteriormente se conecta a internet y se sincronizan con los servicios necesarios y los archivos se actualizan de forma rápida, siendo necesario en algunos casos solo unos pocos segundos para tener la nueva entrada publicada en línea correctamente.

1.2 Herramientas existentes para la actualización de blogs.

Se relacionan un conjunto de herramientas existentes a nivel mundial para la publicación y actualizaciones de blogs basadas en sus principales características, ventajas, desventajas y criterios básicos, se realiza un estudio para encontrar alternativas a Windows Live Writer en los sistemas GNU/Linux.

1.2.1 Extensiones de Firefox.

Las extensiones son sub-aplicaciones, que se integran con el navegador Firefox donde las mismas pueden ser desarrolladas de forma independiente al navegador, permiten añadir nuevas funcionalidades y comportamientos, adaptadas a los requerimientos del usuario aunque no se puede asegurar la compatibilidad con la versión del navegador instalada. Se presenta el estudio de dos populares extensiones para la publicación y actualización de artículos en plataformas de blogs.

SCRIBE FIRE

ScribeFire es una extensión para el navegador Firefox, que hace posible la publicación de artículos directamente desde el navegador, cuenta con una interfaz atractiva y útil. Es considerado uno de los mejores editores de blogs dentro de Firefox.

En pruebas realizadas en los blogs administrados en el departamento DOWAI del centro CIDI, cuando se intenta archivar una entrada para su posterior edición no permite almacenarlo en un archivo local, sino que guarda solo la última entrada en la que se trabajaba, muchas veces no reconoce las etiquetas que se le asignan al artículo, es inestable para la conexión al blog pues debe permanecer conectado a la plataforma para poder trabajar sobre él, siendo por tanto una réplica en línea. En un monitoreo de la cuota de internet UCI se pudo detectar que en un minuto consumió 0.95 Megabytes (MB) lo cual no es conveniente para su uso en CIDI.

DEEPEST SENDER

Entre las principales características de esta herramienta se encuentra la gestión de perfiles de manera sencilla, las entradas se editan en modo texto aunque también permite ver el código HTML para usuarios avanzados, cuenta con el modo vista previa y permite el trabajo con categorías.

Durante las pruebas realizadas se logró conectar con la plataforma Wordpress y publicar, incluyendo imágenes. Se descarta la posibilidad de utilizarlo pues al enviar imágenes la manera de redimensionarlas no es factible y amigable respecto al usuario, además de que la forma de cargar la imagen es definiendo manualmente la ruta del archivo desde un vínculo HTTP o localmente desde el ordenador. Si se ingresa una ruta desde el ordenador no actualiza la imagen en la publicación, además de no mostrarse en la interfaz de edición local. Tampoco permite definir o ingresar categorías a la publicación, no posee un corrector ortográfico y no existe la posibilidad de editar una entrada ya publicada en la plataforma.

1.2.2 Aplicaciones de escritorio.

A continuación se presentan un conjunto de aplicaciones de escritorio que se utilizan como herramientas para una operación o tarea específica en una computadora y que están diseñadas para acciones puntuales del usuario sobre un sistema operativo de interfaz visual como Windows o Linux y que son instaladas en cada uno de los equipos personales. Se presentan una serie de características y funciones de las

herramientas de escritorio más potentes existentes en el mundo para la edición y publicación de artículos sobre las plataformas de blogs.

BLOGILO

Es una herramienta de escritorio de las más utilizadas en el mundo sobre plataformas Linux, por su uso puede llegar a compararse con el Windows Live Writer. Entre las principales características de Blogilo está la presencia de un editor HTML para usuarios avanzados, una vista previa de la entrada que se está editando, brinda la posibilidad de añadir nuevas etiquetas y variadas opciones de edición de texto e imagen.

En el período de pruebas mostró inconsistencia en su funcionamiento. Accediendo a internet mediante la herramienta CNTLM se intentó conectar y se detiene por el siguiente error:

```
(An error occurred in the last transaction. Server (XMLRPC) error: Received invalid XML markup: tag mismatch at 106:7)
```

Entre otras desventajas encontradas es que no permite añadir nuevas categorías al blog a pesar de si cargar las existentes en el blog y poder asignárselas a la entrada que se esté editando.

GNOME BLOG

Es una herramienta de escritorio muy sencilla de manejar, las opciones de edición de texto son muy pocas, solo permite editar los estilos de fuentes como negrita, cursiva y añadir enlaces. Es rápido a la hora de añadir una nueva entrada. Durante las pruebas se pudo detectar que no es recomendable para la actualización de entradas en la plataforma Wordpress por las pocas opciones de edición que posee, no se tuvo éxito en establecer una conexión efectiva con un blog de prueba, se intentó publicar una entrada y la aplicación deja de responder durante varios minutos. Tampoco soporta el trabajo con categorías ni imágenes, no posee ayuda de usuario y no brinda la posibilidad de mantener varios blogs simultáneamente.

QTM

Herramienta diseñada para el escritorio KDE por sus siglas en inglés (*K Desktop Environment*), aunque también trabaja sobre GNOME, entre sus características fundamentales están la presencia

de un editor de texto con opciones para especificar categorías, etiquetas además contiene plantillas de entrada rápida lo cual es muy factible para publicar mensajes rápidos.

En las pruebas realizadas a esta herramienta se pudo apreciar la ausencia de configuración de proxy o parámetros de conexión necesarios en la infraestructura de red utilizada en la UCI, aunque permite editar los parámetros relacionados con la cuenta de blog. Cuando se intenta publicar una entrada se muestra una barra de progreso que no pasa de un 50% y permanece por más de 10 minutos sin mostrar error alguno o bloquear otras opciones. Otra desventaja es que a la hora de insertar una imagen se debe hacer de forma manual escribiendo la ruta donde se encuentra almacenada.

1.2.3 Análisis comparativo de las herramientas estudiadas.

A continuación se realiza una comparación entre las herramientas anteriormente estudiadas teniendo en cuenta las características fundamentales.

Tabla 1 Comparación de herramientas

Características	Guardar Fichero Local	Programar Publicación	Editar parámetros de conexión	Gestión Categorías	Edición de Artículos ya publicado
SCRIBE FIRE		X		X	
DEEPEST SENDER	X			X	
BLOGILO			X	X	
GNOME BLOG					
QTM				X	

Después de realizar un análisis comparativo entre las principales características de los sistemas homólogos estudiados para la actualización de blogs, se puede apreciar que las herramientas analizadas anteriormente no satisfacen las necesidades del departamento DOWAI. Entre las principales características para realizar dicha comparación se encuentran: la posibilidad de guardar el fichero de forma local, donde solo el Deepest Sender lo permite, la opción de programar la publicación del artículo en edición, solo ScribeFire lo permite,

se pudo apreciar además que solo el Blogilo permite editar los parámetros de conexión necesarios para cuando se utiliza un proxy. En la gestión de categorías a pesar de que algunas herramientas permiten su gestión, no se brinda una interfaz de usuario amigable y no todas permiten el manejo de las mismas. Como principal característica se pudo identificar que ninguna de las herramientas estudiadas permite editar una entrada previamente publicada, representando esto una de las principales funcionalidades que se precisa por el departamento.

1.3 Metodología de desarrollo

Cuando se va a realizar un proceso de desarrollo de software es necesario controlar los flujos de trabajo que este origina. Si no se sigue un patrón, es muy posible que como salida final existan clientes insatisfechos o resultados no esperados. Uno de los principios de la UCI para el desarrollo de software, es seguir desde su concepción una metodología de desarrollo, definida como “un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación a ser llevados a cabo por los desarrolladores de sistemas informáticos” (8).

Es por esto que el impacto de elegir la metodología para un equipo en un determinado proyecto es trascendental para el éxito del producto. Entre ellas se encuentran las ágiles, que se han convertido en una de las más utilizadas por las empresas desarrolladoras de software debido al bajo costo en la producción de software y a la reducción en el tiempo de entrega de los productos, y también existen las tradicionales que se caracterizan por ser robustas y para grandes equipos de desarrollo.

Tabla 2 Comparación de metodologías

Metodologías Tradicionales	Metodologías Ágiles
<ul style="list-style-type: none"> Rigidez ante los cambios, de manera lenta o moderada 	<ul style="list-style-type: none"> Flexibilidad ante los cambios del proyecto de forma moderada a rápida
<ul style="list-style-type: none"> Los clientes interactúan con el equipo de desarrollo mediante reuniones 	<ul style="list-style-type: none"> Los clientes son parte del equipo de desarrollo
<ul style="list-style-type: none"> Grupos de gran tamaño y varias veces distribuidos en diferentes sitios 	<ul style="list-style-type: none"> Grupos pequeños (promedio 10 participantes) en el mismo lugar.

<ul style="list-style-type: none"> • Dependencia de la arquitectura de software mediante modelos 	<ul style="list-style-type: none"> • Menor dependencia de la arquitectura de software
<ul style="list-style-type: none"> • Basadas en normas de estándares de desarrollo 	<ul style="list-style-type: none"> • Basadas en heurísticas a partir de prácticas de producción de código
<ul style="list-style-type: none"> • Procesos muy controlados por políticas y normas 	<ul style="list-style-type: none"> • Procesos menos controlados, pocas políticas y normas
<ul style="list-style-type: none"> • Seguimiento estricto del plan inicial de desarrollo 	<ul style="list-style-type: none"> • Capacidad de respuesta ante los cambios

1.3.1 Metodologías Ágiles

Las metodologías se enfocan en los clientes y los resultados. Se basan en promover iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto, logrando que se minimicen los riesgos desarrollando software en cortos períodos de tiempo.

Programación Extrema (XP) (9)

Programación Extrema es un enfoque de la ingeniería del software formulado por Kent Beck. Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y disposición para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Entre las ventajas de XP se encuentran las siguientes:

- Apropiado para entornos volátiles.
- La preparación para el cambio, significa reducir su coste.
- Planificación más transparente para los clientes, conocen las fechas de entrega de funcionalidades.
- Permite tener retroalimentación de los usuarios.

- La presión está a lo largo de todo el proyecto y no en una entrega final.

SCRUM + Programación Extrema (SXP) (10)

SXP es una metodología compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica. Se parte de la introducción de procedimientos ágiles que permiten actualizar los procesos de software para el mejoramiento de la actividad productiva. Fomenta el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo y ayudando al líder del proyecto a tener un mejor control del mismo.

SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos. XP más bien es una metodología encaminada para el desarrollo; consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

SXP consta de cuatro fases principales:

- **Planificación-Definición:** Donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- **Desarrollo:** Es la fase donde se realiza la implementación y construcción del sistema hasta que esté listo para ser entregado, luego de un conjunto de iteraciones de desarrollo.
- **Entrega:** Fase donde se pone en marcha el producto desarrollado y se genera la documentación necesaria para hacer la entrega al cliente.
- **Mantenimiento:** En esta se realiza el soporte para los problemas que pueda presentar el software durante su despliegue.

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día como progresa el trabajo. (10)

OpenUP (11)

Dentro de estas metodologías se encuentra la metodología Proceso Unificado Abierto (OpenUP). Caracterizada por ser una metodología ágil, diseñada para un equipo de desarrollo reducido con un conjunto mínimo de prácticas que ayudan a ser más eficiente el proceso de desarrollo de software. Además por ser extensible, con iteraciones cortas y permitir mantener la filosofía de “*Rational Unified Process*” (RUP), la cual es una de las más empleadas actualmente en la UCI, pues mantiene sus características esenciales: centrado en la arquitectura, dirigida por Casos de Uso (CU) y desarrollo iterativo e incremental dentro del ciclo de vida de un proyecto de software. No provee lineamientos para todos los elementos que se manejan en un proyecto, pero tiene los componentes básicos que pueden servir de base a procesos específicos. La mayoría de los elementos de OpenUP están declarados para fomentar el intercambio de información entre los equipos de desarrollo y mantener un entendimiento compartido del proyecto, sus objetivos, alcance y avances. Esta metodología excluye la mayoría de las piezas opcionales de RUP, y muchos artefactos han sido fusionados. El resultado final es un proceso mucho más sencillo que sigue siendo fiel a los principios de RUP. A continuación se exponen algunos de los beneficios que trae consigo el empleo de esta.

Beneficios (11):

- Es apropiada para proyectos pequeños y de bajos recursos, permite disminuir las probabilidades de fracaso en los proyectos pequeños e incrementar las probabilidades de éxito.
- Permite detectar errores en etapas tempranas del desarrollo de un producto de software a través de un ciclo iterativo.
- Genera la documentación necesaria para el desarrollo del proyecto, lo que hace que se entreguen en el tiempo estimado.
- Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas.

La metodología de desarrollo OpenUP está caracterizada por cuatro principios básicos interrelacionados:

- Colaboración para unificar intereses y compartir conocimientos.
- Equilibrio de prioridades competentes a maximizar el valor de los involucrados con el resultado del proyecto.

- Enfoque en la articulación de la arquitectura.
- Desarrollo continuo para obtener retroalimentación y realizar las mejoras respectivas.

Esta metodología está compuesta por cuatro fases las cuales se exponen a continuación:

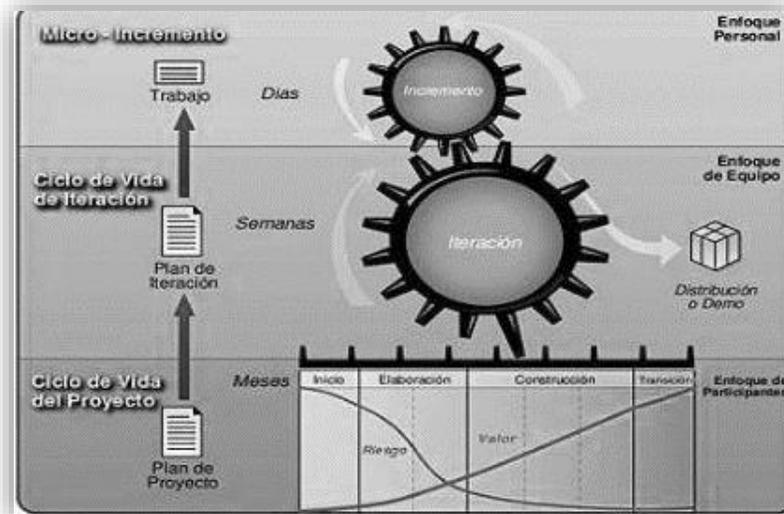


Figura 1 Fases de la Metodología OpenUP

- **Concepción:** se enfoca en el entendimiento del propósito y objetivos del proyecto que se desarrollará, obteniendo suficiente información para confirmar lo que se debe hacer. En esta fase se debe entender qué construir, identificar los actores, quién está interesado en este sistema.
- **Elaboración:** en esta fase se tratan los riesgos significativos para la arquitectura. El propósito de esta fase es establecer la base de elaboración de la arquitectura del sistema. Además obtener una comprensión más detallada de los requerimientos que debe cumplir el sistema.
- **Construcción:** fase enfocada al diseño, implementación y prueba de las funcionalidades para desarrollar un sistema completo. El propósito de esta fase es completar el desarrollo del sistema basado en la arquitectura definida.
- **Transición:** es la última fase, cuyo propósito es asegurar que el sistema es entregado a los usuarios, y evalúa la funcionalidad del último entregable de la fase de construcción.

Luego de analizar las fundamentales metodologías ágiles se decide seleccionar Open Up, debido a su adecuación para el desarrollo de la solución a proponer en la investigación. Proporciona una comprensión detallada del proyecto de software, beneficiando a clientes y desarrolladores sobre los productos a entregar, centrándose en una arquitectura temprana para reducir al mínimo los riesgos y organizar el desarrollo.

1.4 Tecnologías y herramientas seleccionadas para el desarrollo del software.

Para el desarrollo exitoso de un producto de software es fundamental la correcta selección de las tecnologías y herramientas a emplear, pues esto influye directamente en la calidad del producto final y el nivel de esfuerzo del equipo de desarrollo para obtenerlo. Una de las características que debe de presentar la nueva solución es poder administrar desde la aplicación, los blogs alojados en la plataforma Wordpress pertenecientes al departamento DOWAI. A continuación se relacionan un conjunto de tecnologías y herramientas necesarias para el desarrollo exitoso de la solución propuesta.

Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado, del inglés *Integrated Development Environment (IDE)*, es un programa informático compuesto por un conjunto de herramientas de programación. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (12)

NetBeans: 8.0

NetBeans es un IDE disponible para Windows, Mac, Linux y Solaris. Ofrece todas las funciones de los IDE avanzados como diseño de interfaces, asistentes para la conexión con bases de datos, creación automática de propiedades y clases. Es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para otros lenguajes de programación. Es un producto libre y gratuito, sin restricciones de uso. Consta de un importante número de módulos que permiten extenderlo. Permite el desarrollo de aplicaciones multiplataforma de forma muy sencilla y consta de versiones exclusivas para el desarrollo con PHP. (13)

Lenguaje de programación

Un lenguaje de programación es un conjunto de sintaxis y reglas semánticas que definen los programas del ordenador. Es una técnica estándar de comunicación para entregarle instrucciones al computador. Le proporciona la capacidad al programador de especificar qué tipo de datos actúan y que acciones tomar bajo una variada gama de circunstancias, utilizando un lenguaje relativamente próximo al lenguaje humano (14).

Para la selección del lenguaje de programación a utilizar se tuvieron en cuenta aspectos tales como: la interacción con los usuarios, integración con otras aplicaciones, soporte sobre sistemas GNU/Linux, entre otras características que se explicarán a continuación.

Java

Java es un lenguaje originalmente desarrollado por un grupo de ingenieros de Sun, utilizado por Netscape posteriormente como base para JavaScript. Si bien su uso se destaca en el Web, sirve para crear todo tipo de aplicaciones (locales, intranet o internet). La compañía Sun describe el lenguaje Java como *“simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico”*.

Es un lenguaje de objetos e independiente de la plataforma, entre sus características más notables se destaca por ser un lenguaje robusto, gestionar la memoria automáticamente, no permitir el uso de técnicas de programación inadecuadas, subprocesamiento múltiple, modelo cliente-servidor, mecanismos de seguridad incorporados y herramientas de documentación incorporadas. (15)

Se decide escoger Java por su integración con la plataforma Wordpress y su fácil manejo con las APIs que esta brinda a través la librería jWordpress y el protocolo de comunicación XML-RPC. El cual se argumentará más adelante en este documento, además de las características antes mencionadas y su amplia gama de documentación disponible.

JDK 7

Existen distintos programas comerciales que permiten desarrollar código Java. La compañía Sun, creadora de Java, distribuye gratuitamente el *“Java™ Development Kit”* (JDK). Se trata de un conjunto de programas y librerías que permiten desarrollar, compilar y ejecutar programas en Java. Incorpora además la posibilidad de ejecutar parcialmente el programa, deteniendo la ejecución en el punto deseado y estudiando en cada momento el valor de cada una de las variables (con el denominado Debugger). Una parte muy importante del tiempo destinado a la elaboración de un programa se destina a la detección y corrección de errores. Existe también una versión reducida del JDK, denominada *“Java Runtime Environment”* (JRE) destinada únicamente a ejecutar código Java (no permite compilar). (15)

Protocolo de comunicación

Un protocolo de comunicación es un conjunto de normas que están obligadas a cumplir todas las máquinas y programas que intervienen en una comunicación de datos entre ordenadores sin las cuales la comunicación resultaría caótica y por tanto imposible. (16)

Por ejemplo, dos equipos diferentes de marcas diferentes se pueden comunicar sin problemas en caso de usar el mismo protocolo de comunicación. A lo largo del tiempo han ido mejorando las tecnologías de las comunicaciones, y han podido comenzar a utilizarse protocolos más útiles para aumentar la eficiencia de estos. Es por ello que han aparecido nuevos protocolos a los que se han adaptado paulatinamente productos de cada fabricante para asegurar la compatibilidad con el resto de las marcas. A continuación se explica en qué consiste el protocolo XML-RPC que se usará en la implementación de la solución a proponer.

XML-RPC

Es una especificación y un conjunto de implementaciones que permiten que un software corra en sistemas operativos diferentes, corriendo en ambientes distintos para llevar a cabo llamadas entre dispositivos de comunicación y llevar a cabo las llamadas de procedimientos sobre internet. El llamado remoto usa HTTP como transporte y XML como codificación. Está diseñado para ser tan simple como sea posible, permitiendo la transmisión de estructuras de datos complejas, procesarlas y regresarlas. Su arquitectura se basa en el modelo cliente-servidor donde el cliente ejecuta un RPC en el servidor y este procesa cada RPC llamando el correspondiente manejador de RPC. (17)

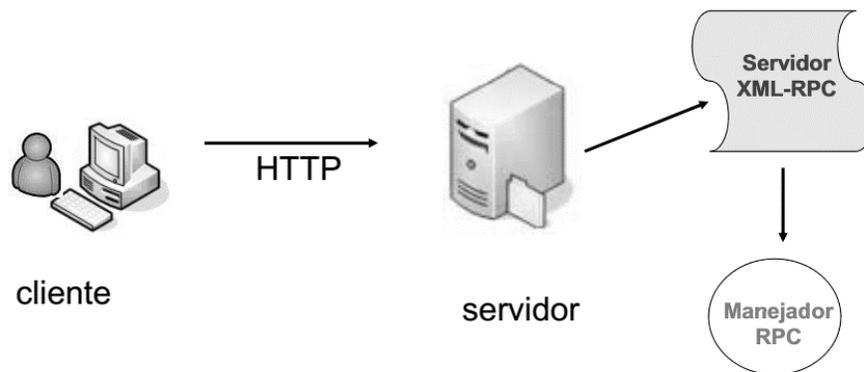


Figura 2 Arquitectura del protocolo XML-RPC

Entre los lenguajes de programación que soporta este protocolo se encuentran:

- Perl, Python, Microsoft .NET, Rebol, Python, **Java**, Frontier, C/C++, Lisp, Rebol, Real Basic, Tcl, Delphi, WebObjects, PHP y Zope.

Procesador de texto

Se utilizó el código de un procesador de textos publicado en el sitio *Hexidec Codex* (hexidec.com). El cual se dedica a la distribución de software libre y de código abierto. Esta herramienta posee un código fácil de entender y con eficientes componentes de edición de texto, por lo que representa una gran ventaja para el desarrollo de la solución que se propondrá, donde el nivel básico es la presencia de un editor de textos. Para el desarrollo del presente trabajo se adaptaron las funcionalidades a la solución que se propondrá.

Es un editor de texto HTML desarrollado en java y de código abierto, libre de nuevas contribuciones. Es un sistema autónomo que permite cargar y guardar archivos en formato HTML, además de exportarlo como RTF. Este editor WYSIWYG ha demostrado ser muy popular por un centenar de desarrolladores que lo usan en más de 30 países (18), entre las características fundamentales de la aplicación se destacan:

Características (18):

- Estilos de texto y colores, que no es más que la posibilidad de darle estilos al texto y colores deseados a gusto del usuario.

- Copiar, cortar y pegar.
- Negrita, cursiva y subrayar.
- Inserción de imagen, donde se puede especificar las dimensiones de dicha imagen en el panel de edición así como eliminarla en caso deseado.
- Buscar y reemplazar, consiste en buscar la palabra específica y reemplazarla por lo que se desee en el momento.
- Ventana de edición de código fuente, permite editar el código HTML para usuarios avanzados.
- Diálogo selector de fuente le permite usar y ver fuentes del sistema en la página HTML.
- Cargar y guardar documentos en seguros para la web, codificación en Base64.
- Compatible con herramientas de inserción de caracteres Unicode Dingbats, símbolos de las matemáticas y alfabetos extendidos.
- Selector de fuente desplegable con el objetivo del fácil acceso a editar la fuente del texto.
- Insertar imágenes desde URLs.
- Impresión, brinda la posibilidad de imprimir el documento en edición.
- Opción de pegar "texto plano" elimina el formato del contenido del portapapeles.

Lenguaje Unificado de Modelado (UML) 8.0

Lenguaje Unificado de Modelado del inglés *Unified Modeling Language* (UML) es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un modelo, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables (19). Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. Incluye conceptos semánticos, notación y principios generales. Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo, pero no especifica en sí mismo qué metodología o proceso usar.

Herramienta CASE

Las herramientas CASE, acrónimo de *Computer Aided Software Engineering* (Ingeniería de Software Asistida por Computadoras) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y dinero (20).

Visual Paradigm 8.0

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de mejor calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta CASE UML también proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML (21).

Conclusiones parciales

En el presente capítulo luego de una revisión bibliográfica y un estudio de los sistemas homólogos al que se desea implementar, se pudo identificar que las soluciones existentes no cubren las funcionalidades deseadas por los especialistas del departamento DOWAI y presentan varias inconsistencias respecto a problemas de conexión y características fundamentales, siendo necesaria la implementación de un nuevo editor de blog que cumpla con las necesidades reales del departamento DOWAI. Se abordaron las características, ventajas y desventajas de los diferentes editores de blog existentes sobre la plataforma Wordpress, demostrándose así en pruebas realizadas directamente sobre la plataforma en Internet. Del análisis sobre las tendencias actuales en el desarrollo de software, se seleccionaron las tecnologías y herramientas más adecuadas para realizar la aplicación informática que se requiere, seleccionando como metodología de desarrollo ágil OpenUP, como lenguaje de modelado UML 8.0, utilizándose Visual Parading como herramienta UML, como lenguaje de programación Java y como IDE NetBeans 8.0. Además de importar el código de un procesador de textos desarrollado en Java con el objetivo de agregarle las funcionalidades necesarias que den solución al problema planteado y que sirviera de base como editor de texto en la aplicación, como protocolo de comunicación con la plataforma Wordpress se decidió utilizar XML-RPC.

CAPÍTULO 2: ANÁLISIS Y DISEÑO

Introducción

Se realiza el levantamiento de los requisitos, con el objetivo de dar una solución factible al problema existente. En este capítulo se definen el Modelo de Dominio, entidades y actores que intervienen, los requisitos funcionales y no funcionales, así como diagrama de caso de uso del sistema y las relaciones existentes entre ellos. El proceso de desarrollo a seguir estará guiado por las pautas que se rigen en la metodología de desarrollo de software OpenUP. Lo anterior guiará la construcción del sistema informático que fundamenta la presente investigación.

2.1 Propuesta de solución

Se propone la implementación de una aplicación de escritorio para la publicación de artículos sobre la plataforma Wordpress que cumpla con los requerimientos de los especialistas del departamento DOWAI y contenga las funcionalidades identificadas, basado en el funcionamiento de los sistemas homólogos y que satisfaga la problemática antes planteada en la introducción de este trabajo.

2.2 Modelo de Dominio

Un modelo de dominio o mapa conceptual es una representación de conceptos en un dominio del problema. Comunica (a los interesados, como pueden ser los desarrolladores) cuáles son los términos importantes y como se relacionan entre sí. Puede ser tomado como el punto de partida para el diseño del sistema. (11)
Para la modelación de la propuesta de solución es necesario comprender los conceptos con los que debe trabajar en la aplicación y brindar una representación visual del entorno de la misma. Con la realización del modelo de dominio se pretende unificar el vocabulario entre los usuarios y los desarrolladores y tener más centrado cómo funciona el proceso de actualización de los blog.

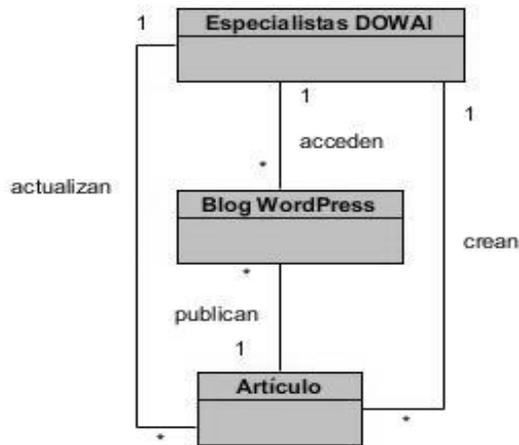


Figura 3 Modelo de Dominio

2.3 Especificación de los Requisitos del Sistema

Para comenzar el desarrollo de un sistema informático, una etapa muy importante dentro del proceso de la Ingeniería de Software, es la Ingeniería de Requisitos, donde se lleva a cabo el proceso de descubrir, analizar, escribir y verificar los servicios y restricciones, del sistema de software que se desea desarrollar; este proceso se realiza mediante la obtención, el análisis, la especificación, la validación y la administración de los requerimientos del software.

Los requisitos de software son las necesidades de los clientes, los servicios que los usuarios desean que proporcione el sistema de desarrollo y las restricciones en las que debe operar. (22)

2.3.1 Técnicas de Obtención de Requisitos

La obtención de requisitos “es el proceso de recoger información sobre el sistema propuesto y los existentes, extraer los requisitos del usuario y del sistema de esta información” (22). Para identificar y extraer los requisitos se emplean técnicas que ofrece la ingeniería de requisitos, por ejemplo: puntos de vista, entrevistas, escenarios, casos de uso, prototipos, entre otros. De estas técnicas se utilizaron para la captura de los mismos, las siguientes:

Entrevistas: se realizaron entrevistas con el cliente en aras de identificar las funcionalidades que requiere el sistema y así identificar los problemas reales de los sistemas homólogos para tenerlo en cuenta durante la implementación de la aplicación, estas entrevistas continúan realizándose durante todo el ciclo de desarrollo y son de gran importancia para los desarrolladores.

Puntos de vista: se realizaron reuniones con el cliente, en este caso con los editores del departamento DOWAI y representantes del proyecto donde se definieron varios puntos de vista que eran de gran importancia para la solución. Con las diferentes opiniones recogidas se lograron definir algunas funcionalidades que requería la aplicación.

2.3.2 Requisitos Funcionales (RF)

Los requisitos funcionales de un sistema describen lo que el sistema debe hacer. Estos requisitos dependen del tipo de software que se desarrolle, de los posibles usuarios y del enfoque general tomado por la organización al redactar el requisito. Los requisitos funcionales del sistema describen con detalle la función de este, sus entradas, salidas y excepciones. (22) Se identifican los siguientes requisitos funcionales del sistema:

RF 1 Autenticar usuario en el blog.

RF 2 Gestionar Entrada.

RF 2.1 Crear Entrada.

RF 2.2 Obtener Entrada.

RF 2.3 Modificar Entrada.

RF 2.4 Eliminar Entrada.

RF 3 Gestionar Categorías.

RF 3.1 Crear Categoría.

RF 3.2 Obtener Categoría.

RF 3.3 Eliminar Categoría.

RF 4 Programar publicación.

RF 5 Insertar Imagen.

RF 6 Asignar autor a la entrada.

RF 7 Publicar o actualizar la entrada.

2.3.3 Requisitos No Funcionales (RNF)

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema (22). Estas restricciones son las que hacen ver al producto usable, rápido y confiable, a continuación se identifican los siguientes:

RNF 1.Requisitos de Usabilidad

- La usabilidad mide cómo el producto puede ser usado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado. Teniendo en cuenta este concepto el sistema puede ser usado por cualquier persona que posea conocimientos básicos de informática.

RNF 2.Requisitos de Confiabilidad

- Cuando ocurre una excepción el sistema mostrará un mensaje explicativo del error ocurrido y permanecerá en el mismo estado sin realizar ninguna otra operación.
- Cuando se realiza alguna operación que implique hacer cambios significativos en el blog, se mostrará un mensaje de confirmación con el objetivo de brindar mayor seguridad al usuario, en sus acciones.

RNF 3.Requerimientos de Software

- Sistema operativo con interfaz gráfica (GNU/Linux en cualquiera de sus distribuciones)
- Máquina virtual de Java previamente instalada a partir de la versión 7.0.
- Tener configurado el servicio NTLM en el ordenador.

RNF 4.Requerimientos de Hardware (Mínimos)

- 1 Gigabyte (GB) de memoria RAM o superior.
- Tarjeta de red a 100 Megabytes (MB)/s o superior.
- 500 MB de espacio libre en el disco duro o superior.
- Procesador Pentium IV a 2.5 Giga Hertz (GHz) o superior

RNF 5.Requerimientos de apariencia o interfaz externa

Este tipo de requisitos se enfatizan en la apariencia del producto, no se trata del diseño de la interfaz en detalle sino que se especifica cómo se pretende que sea la interfaz externa de la aplicación.

- La interfaz del sistema se realizará a través de una aplicación de escritorio.
- Iconos estándares de edición de texto para facilitar su usabilidad.
- Se diseñó una interfaz con el objetivo de que el cliente pueda explotar al máximo las funcionalidades que brinda para agilizar su trabajo a través de una interfaz amigable.

RNF 6. Requerimientos de Seguridad

- Se garantiza la conexión segura a través del protocolo HTTPS protegiendo así la integridad de la información.
- Se garantiza que la información solo pueda ser modificada por algún editor del blog y los autores solo tendrán acceso a modificar sus artículos y publicaciones.
- Verificación sobre acciones irreversibles (eliminaciones).

2.3.4 Descripción de los Requisitos Funcionales

Los requisitos son el producto del trabajo final que genera la ingeniería de requisitos. Describen la función y el desempeño de un sistema basado en computadoras y las restricciones que regirán su desarrollo (11). A continuación se muestra la descripción de dos requisitos funcionales seleccionados y que serán los que se tomarán en cuenta durante todo el documento, para observarlos en su totalidad dirigirse al anexo 2.

Tabla 3 Descripción del requisito funcional Autenticar usuario en el blog

No.	Nombre	Descripción
RF 1	Autenticar usuario en el blog	<ul style="list-style-type: none"> • El requisito permite autenticarse en el blog especificando la URL deseada con el usuario y la contraseña correspondiente a dicha URL, si alguno de los tres parámetros necesarios esta incorrecto se mostrará un mensaje de error de conexión. • También permite establecer los parámetros de conexión con Internet, para esto se debe de tener previamente instalado y configurado el servicio NTLM en el ordenador. • Si la casilla “Editar Proxy” no es activada el sistema tomará por defecto los siguientes valores:

Servidor: localhost o 127.0.0.1

Puerto: 3128

- Si se activa la casilla se mostrarán los campos para modificar los valores.

Prototipo

Establecer Conexión con el Blog

Conexión

URL:

Usuario:

Contraseña: Cantidad de Entradas:

Editar Proxy Servidor: Puerto:

Atributos

Reglas o Restricciones

- URL

- Obligatorio
- Admite texto solamente.
- Admite hasta 250 caracteres.

- Usuario

- Obligatorio
- Admite texto y números.

- Contraseña

- Obligatorio
- Admite texto, números y caracteres especiales.

- Cantidad de Entradas

- Opcional
- Solo admite números.

	<ul style="list-style-type: none"> • Servidor 	<ul style="list-style-type: none"> • Opcional • Admite texto y números.
	<ul style="list-style-type: none"> • Puerto 	<ul style="list-style-type: none"> • Opcional • Admite solo números
	Observaciones	<ol style="list-style-type: none"> 1. Si al escribir la URL o el usuario se exceden de los 250 caracteres el sistema no permitirá seguir escribiendo. 2. Si los parámetros de conexión son incorrectos mostrará un mensaje informativo. 3. En caso de cancelar la autenticación, se cerrará la ventana y se mostrará la interfaz principal del sistema.

Tabla 4 Descripción del requisito funcional Gestionar Entrada.

No.	Nombre	Descripción
RF 2	Gestionar Entrada	<ul style="list-style-type: none"> • El requisito permite crear una nueva entrada, obtener una cantidad determinada de los artículos más recientes ya publicados, modificar una entrada o eliminarla • En caso de seleccionar en la lista de entrada recientes el ítem “nueva entrada”, se refiere a crear una nueva publicación donde los campos de edición quedarán en blanco para redactar la nueva entrada en el blog, la interfaz contiene varias herramientas de edición de texto y otras funcionalidades de gran importancia, como lo son: edición de la fuente, abrir nuevo documento, guardar documento, imprimir documento, ver o editar código HTML, copiar, cortar y pegar, paso atrás, paso adelante, buscar, insertar

		<p>símbolos, seleccionar categoría, insertar hipervínculo, insertar imagen y programar publicación para una hora determinada.</p> <ul style="list-style-type: none">• El usuario escribe el título de la entrada, redacta el artículo e inserta alguna imagen si así lo desea, selecciona las categorías a asignar a la entrada, y de forma opcional puede programar la publicación para una fecha determinada, si el usuario desea seguir la edición posteriormente, puede guardar lo que ha hecho hasta el momento; si desea publicarlo, una vez comprobado que dicho artículo está listo, tiene la opción de publicar la entrada, el sistema devuelve un mensaje informativo si la publicación fue actualizada con éxito o no.• Para obtener las entradas recientes el usuario especificará la cantidad de publicaciones más recientes que desea obtener al conectar un nuevo blog y de estos se mostrará el título en una lista despegable donde al seleccionar el deseado cuando se escoge la opción obtener entrada, se actualizarán los campos de título, contenido y lista de categorías de la interfaz principal.• Si desea eliminar algún artículo se selecciona de la lista despegable el deseado y se selecciona el botón de eliminar.
Prototipo		

WPBlogEditor

Fichero Editar Ver Fuente Formato Buscar Insertar Tabla Formulario Ayuda

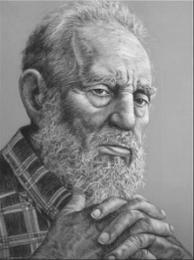
Entradas Recientes: Vivien... Tue May 26 15:44:04 CDT 2015

Autenticado como: vladistest
Autor: Vladis Miriel

Panel de Edición

TÍTULO: **Viviendo de cerca la historia de la Revolución.**

El miércoles 20 de mayo los cuadros y dirigentes de las organizaciones políticas y de masas de la Facultad decidieron tener un encuentro cercano con la Historia de nuestra Revolución, "Donde si no", claro en el Museo de la Revolución.



A pesar de la reparación capital que recibe el edificio por etapas, su personal alega que es inconcebible cerrar las puertas de este tesoro de nuestra historia al público tanto nacional como foráneo, en el caso de este último; resaltando que aunque constituye una fuente de ingresos nada despreciable lo más importante es y cito "Dar a conocer la verdad histórica y contemporánea de Cuba", entendimos entonces que esta premisa constituye la misión fundamental del Centro declarado Monumento Nacional en enero de 2010.

Quedamos prendados con la explicación de la guía asignada quien en un tono educorado y lleno de amor por su labor nos resumió la composición de las áreas expositivas y su historia. "Ojo las delegaciones estudiantiles universitarias y de otras enseñanzas tienen acceso al edificio libre de costo y la posibilidad además de contar con un guía", esto significa que para asistir allí con nuestras brigadas solo necesitamos el deseo, la reservación, bueno y el transporte (más complejo pero soluble). Antes de regalar nuestras instantáneas te decimos que fue definitivamente un viaje casi vivido a cada momento de la Revolución cubana, y sobre todas las cosas una oportunidad más para compartir en un ambiente sano y culto.dweqd
asdkmasdk

Categorías

- autor
- fghfg
- festval
- himno
- himno1
- ispasimuto
- holguin
- informacion
- jovenes
- lor
- marti

Categorías de la Entrada

prueba

Añadir Categoría

Estado de la Entrada

Programar Publicación

Fecha y Hora: 10/06/15 15:12

Atributos	Reglas o Restricciones
<ul style="list-style-type: none"> Título 	<ul style="list-style-type: none"> Admite texto, números y símbolos. Admite hasta 250 caracteres.
<ul style="list-style-type: none"> Contenido 	<ul style="list-style-type: none"> Admite texto, números, símbolos e imágenes.
<ul style="list-style-type: none"> Imagen 	<ul style="list-style-type: none"> Opcional Dimensiones mínimas de 1x1.
<ul style="list-style-type: none"> Categoría 	<ul style="list-style-type: none"> Opcional
Observaciones	<ol style="list-style-type: none"> El usuario no tiene que estar autenticado en el blog para crear una nueva entrada.

2.4 Modelo de Casos de Uso del Sistema

Un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema (11). Los diagramas de casos de uso sirven para detallar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y otros sistemas. A continuación se muestra el diagrama de modelo de casos de uso definido en el desarrollo del sistema.

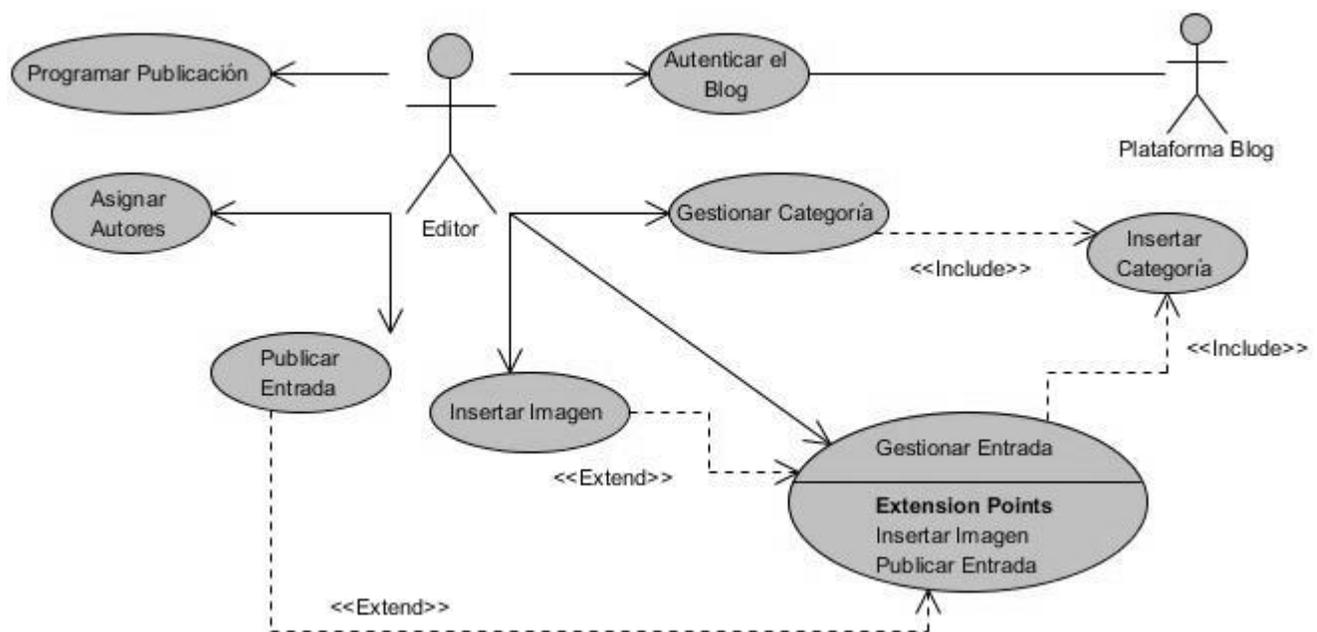


Figura 4 Diagrama de Casos de Uso del Sistema

2.4.1 Definición de los actores del sistema

Los actores del sistema son entidades distintas a los usuarios en el sentido de que estos son las personas reales que utilizarán el sistema, mientras que los actores representan cierta función que una persona real realiza (23). Se identificaron como actores del sistema todos aquellos usuarios que tienen cuentas creadas en la plataforma de blog Wordpress y deseen administrarlas desde la solución propuesta.

Tabla 5 Actores del sistema

Actor	Descripción
Editor	Especialistas del departamento DOWAI que sean editores de los blog administrados.
Plataforma Blog	Es el encargado de establecer la conexión a través de la autenticación permitiendo la gestión del blog.

2.4.2 Descripción de los Casos de Uso del Sistema

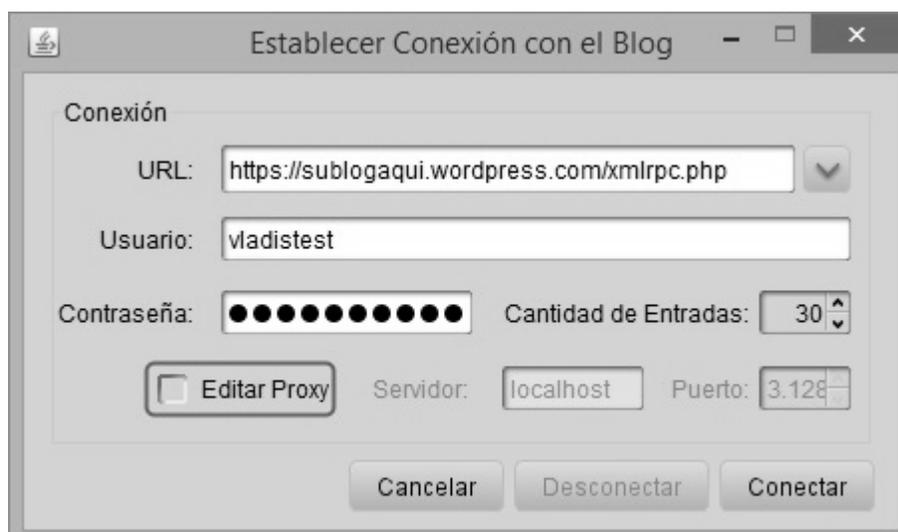
Habitualmente los casos de uso del sistema se describen textualmente, argumentando cada funcionalidad que se le implementará a la aplicación a desarrollar, a continuación se muestra la descripción del caso de uso Autenticar el Blog, para verlos en su totalidad dirigirse al anexo 3.

Tabla 6 Descripción del CU 1 Autenticar en el Blog

Caso de Uso	<i>Autenticar en el Blog</i>	
Actores	<i>Editores, Blog</i>	
Resumen	<i>Autentica en el blog, previamente creado y administrado por un usuario determinado, definiendo la autenticación por los siguientes parámetros:</i> <ul style="list-style-type: none"> • <i>URL del blog</i> • <i>Usuario</i> • <i>Contraseña</i> 	
Complejidad	<i>Baja</i>	
Prioridad	<i>Alta</i>	
Precondiciones	<i>Para autenticarse con éxito, el usuario tiene que tener una cuenta creada en la plataforma Wordpress.</i>	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
	1. <i>Se autentica en la plataforma Wordpress.</i>	
		2. <i>Se autentica en la plataforma mostrando un mensaje informativo, “Se estableció la conexión con el blog”</i>
		3. <i>Termina el caso de uso.</i>

Flujos alternos	
Actor	Sistema
	2b. En caso de tener incorrecto algún parámetro requerido en los datos de autenticación o los parámetros de conexión, el sistema muestra un mensaje de error de autenticación brindando la posibilidad de corregirlos.

Prototipo de Interfaz



Relaciones	CU Incluidos	No procede
	CU Extendidos	No procede
Requisitos no funcionales	RNF1, RNF 5, RNF 6.	
Asuntos pendientes	No procede	

Tabla 7 Descripción CU 2: Gestionar Entrada

Caso de Uso	Gestionar Entrada
Actores	Editores
Resumen	Se gestionan los artículos más recientes existentes en el blog, que incluye crearlos, obtenerlos, modificarlos y eliminarlos.
Complejidad	Alta
Prioridad	Alta
Precondiciones	Para que se ejecute el caso de uso, el editor debe de estar previamente autenticado en la plataforma.
Postcondiciones	En dependencia de la acción que ejecute: <ul style="list-style-type: none"> • Crear Entrada.

	<ul style="list-style-type: none"> • <i>Obtener Entrada.</i> • <i>Modificar Entrada.</i> • <i>Eliminar Entrada.</i> 	
Referencias	<i>RF2.1, RF2.2, RF2.3, RF2.4</i>	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
	1. <i>Se autentica al sistema y elije la opción deseada.</i>	
		2. <i>Permite realizar varias acciones:</i> <ul style="list-style-type: none"> - <i>Crear Entrada. Ver Sección 1: “Crear Entrada”.</i> - <i>Obtener Entrada. Ver Sección 2: “Obtener Entrada”.</i> - <i>Modificar Entrada. Ver Sección 3: “Modificar Entrada”.</i> - <i>Eliminar Entrada. Ver Sección 4: “Eliminar Entrada”.</i>
		3. <i>Termina el caso de uso</i>
Flujos alternos		
	Actor	Sistema
		2b. <i>En caso de no estar conectado a un blog se mostrará un mensaje “Debe autenticarse a algún blog para poder realizar acciones sobre este”.</i>
Prototipo de Interfaz		

WPBlogEditor

Fichero Editar Ver Fuente Formato Buscar Insertar Tabla Formulario Ayuda

Entradas Recientes: Vivien... Tue May 26 15:44:04 CDT 2015

Panel de Edición

TÍTULO: **Viviendo de cerca la historia de la Revolución.**

Autenticado como: vladistest
Autor: Vladis Miriel

Ver Entrada en el Navegador

Categorías

- autor
- fdghfg
- festival
- himno
- himno1
- hipervínculo
- holguin
- informacion
- jovenes
- lor
- marti

Categorías de la Entrada

prueba

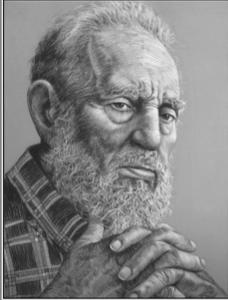
Añadir Categoría

Estado de la Entrada

Programar Publicación

Fecha y Hora: 10/06/15 15:12

El miércoles 20 de mayo los cuadros y dirigentes de las organizaciones políticas y de masas de la Facultad decidieron tener un encuentro cercano con la Historia de nuestra Revolución, "Donde si no", claro en el Museo de la Revolución.



A pesar de la reparación capital que recibe el edificio por etapas, su personal alega que es inconcebible cerrar las puertas de este tesoro de nuestra historia al público tanto nacional como foráneo, en el caso de este último; resaltando que aunque constituye una fuente de ingresos nada despreciable lo más importante es y cito "Dar a conocer la verdad histórica y contemporánea de Cuba", entendimos entonces que esta premisa constituye la misión fundamental del Centro declarado Monumento Nacional en enero de 2010.

Quedamos prendados con la explicación de la guía asignada quien en un tono edulcorado y lleno de amor por su labor nos resumió la composición de las áreas expositivas y su historia. "Ojo las delegaciones estudiantiles universitarias y de otras enseñanzas tienen acceso al edificio libre de costo y la posibilidad además de contar con un guía", esto significa que para asistir allí con nuestras brigadas solo necesitamos el deseo, la reservación, bueno y el transporte (más complejo pero soluble). Antes de regalar nuestras instantáneas te decimos que fue definitivamente un viaje casi vivido a cada momento de la Revolución cubana, y sobre todas las cosas una oportunidad más para compartir en un ambiente sano y culto.dweqd
asdkmasdk

Sección 1 "Crear Entrada"

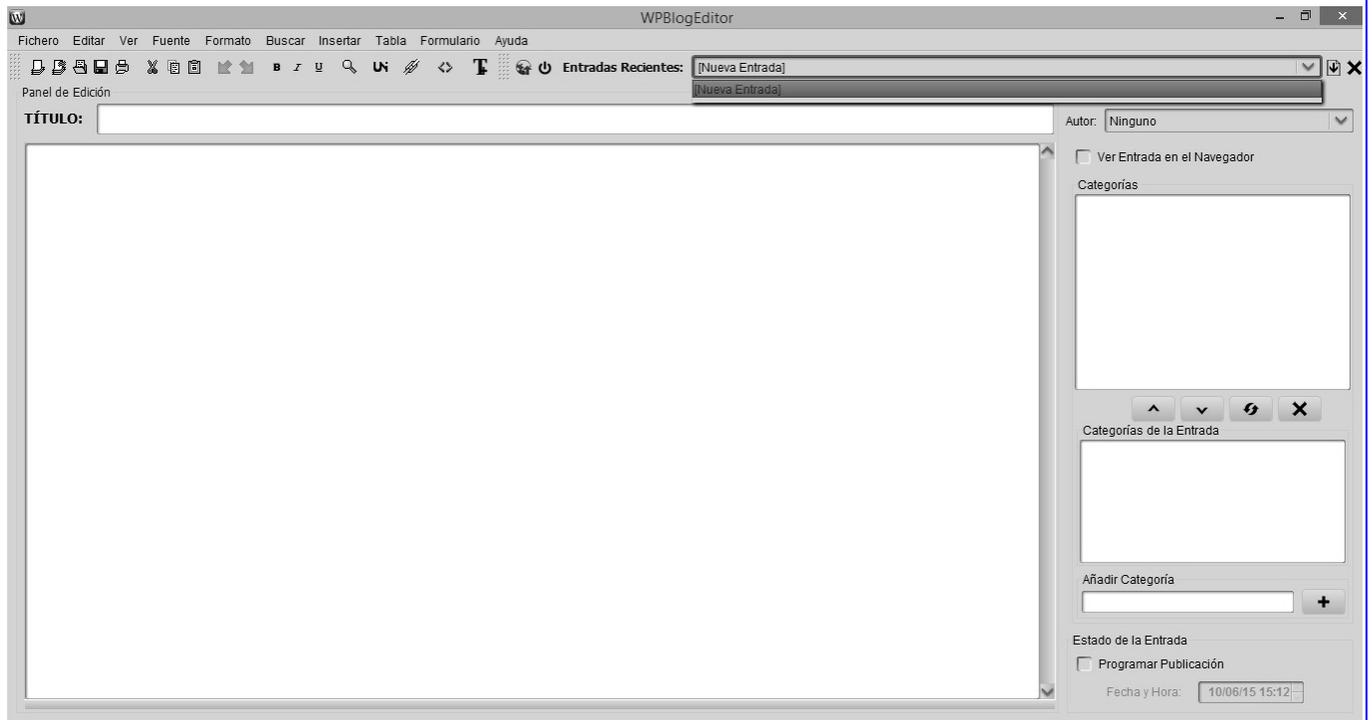
	Actor	Sistema
	1. <i>Selecciona de la lista de Entrada el ítem "Nueva Entrada" o pincha en la opción "Limpiar Campos".</i>	2. <i>Si los campos Título, Contenido o Lista de Categorías asignadas al artículo tienen datos, estos se borran para la nueva entrada.</i>
	3. <i>Redacta el artículo especificando Título, Contenido donde se especifica de manera opcional imágenes e hipervínculos, además selecciona las categorías que se le asignarán a la nueva entrada y de forma opcional programa la publicación para una fecha determinada.</i>	4. <i>Permite escribir en los diferentes campos y seleccionar las categorías deseadas así como seleccionar si programará la publicación especificando la fecha deseada.</i>
	5. <i>Selecciona la opción publicar entrada.</i>	6. <i>Publica la nueva entrada y actualiza la lista de entradas recientes si no fue programada su publicación.</i>

Flujos alternos

	Actor	Sistema
	1b. <i>Selecciona una Entrada reciente</i>	2b. <i>Se muestra un mensaje "¿Está seguro que desea cancelar la entrada?"</i>

6b. Si no ha seleccionado alguna categoría se le asignará una por defecto a la entrada y si no especifica la fecha para la publicación se publica inmediatamente.

Prototipo de Interfaz



Sección 2 “Obtener Entrada”

	Actor	Sistema
	1. Cuando se autentica el blog especifica la cantidad de artículos que desea obtener.	2. Lista el título de la cantidad de entradas determinadas por el usuario, ordenadas por fecha.
	3. Selecciona una entrada de la lista obtenida.	
	4. Selecciona la opción “Obtener Entrada”	5. Muestra el contenido de la entrada, dígame Título, Descripción y Categorías.

Flujos alternos

	Actor	Sistema
	1b. Si la cantidad de entradas es 0.	2b. Muestra un mensaje “La cantidad de entradas debe de ser mayor que 0”.

Prototipo de Interfaz

The screenshot shows the WPBlogEditor interface. At the top, there is a menu bar with options like 'Fichero', 'Editar', 'Ver', 'Fuente', 'Formato', 'Buscar', 'Insertar', 'Tabla', 'Formulario', and 'Ayuda'. Below the menu is a toolbar with various icons for text formatting and media insertion. The main content area has a title field containing 'Fidel Castro recibe distinción mexicana'. Below the title is a paragraph of text and a black and white photograph of Fidel Castro. To the right of the main content is a sidebar with several sections: 'Autenticado como: vladistest', 'Autor: Vladis Miriel', a checkbox for 'Ver Entrada en el Navegador', a list of 'Categorías' (autor, fdghfg, festival, himno, himno1, hipervinculo, holguin, informacion, juvenes, lor, marti), a list of 'Categorías de la Entrada' (hipervinculo, informacion, juvenes, noticia, politico, wordpress), an 'Añadir Categoría' field, and an 'Estado de la Entrada' section with a checkbox for 'Programar Publicación' and a 'Fecha y Hora' field set to '10/06/15 15:12'.

Sección 3 “Modificar Entrada”

	Actor	Sistema
	1. <i>Selecciona la entrada deseada de la lista entradas más recientes y pincha en botón “Obtener Entrada”.</i>	2. <i>Muestra el contenido de la entrada.</i>
	3. <i>Modifica el contenido deseado.</i>	4. <i>Permite escribir en los diferentes campos y seleccionar las categorías deseadas así como eliminar de las ya asignadas, también permite seleccionar si programará la publicación especificando la fecha deseada.</i>
	5. <i>Selecciona la opción modificar entrada.</i>	6. <i>Modifica la entrada y actualiza la lista de entradas recientes si no fue programada su publicación.</i>

Flujos alternos

	Actor	Sistema
		6b. <i>Si la lista de categorías está vacía se le asignará una aleatoria por defecto a la entrada y si no especifica la fecha para la publicación se publica inmediatamente.</i>

Prototipo de Interfaz

The screenshot shows the WPBlogEditor interface. At the top, there's a menu bar with options like 'Fichero', 'Editar', 'Ver', 'Fuente', 'Formato', 'Buscar', 'Insertar', 'Tabla', 'Formulario', and 'Ayuda'. Below the menu is a toolbar with various icons. The main editing area has a title field containing 'Fidel Castro recibe distinción mexicana'. Below the title is a text area with a paragraph of text and a photo of Fidel Castro. To the right, there's a sidebar with a 'Categorías' list, a 'Categorías de la Entrada' list, and a 'Fecha y Hora' field set to '10/06/15 15:12'.

Sección 4 “Eliminar Entrada”

	Actor	Sistema
	1. <i>Selecciona de la lista de entradas reciente la que desee eliminar.</i>	
	2. <i>Selecciona la opción Eliminar Entrada</i>	3. <i>Elimina el artículo seleccionado y actualiza la lista de entradas recientes.</i>

Flujos alternos

	Actor	Sistema
	1b. <i>Selecciona el ítem “Nueva Entrada”</i>	2b. <i>Borra el contenido del artículo anteriormente seleccionado.</i>
	3b <i>Selecciona la opción Eliminar Entrada.</i>	4b. <i>Muestra un mensaje “No ha seleccionado ninguna entrada”.</i>

Prototipo de Interfaz

8

Relaciones	CU Incluidos	<i>No procede</i>
	CU Extendidos	<i>No procede</i>
Requisitos no	<i>RNF1, RNF 2, RNF 5 y RNF 6.</i>	

funcionales	
Asuntos pendientes	<i>No procede</i>

2.5 Diseño del Sistema

“El diseño de software es un proceso iterativo mediante el cual los requisitos se traducen en un “plano” para construir el software” (11). El diseño del sistema define la arquitectura y estructura de hardware, software, componentes, módulos y datos de un sistema de cómputo. Se refiere a la formulación de especificaciones para el nuevo sistema propuesto, de manera que satisfaga los requisitos determinados durante la fase de análisis, tanto desde el punto de vista funcional como del no funcional.

2.5.1 Patrón Arquitectónico

Son patrones de diseño de software que ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados (24).

Arquitectura de 3 capas.

La utilización de este patrón permite una separación de conceptos, de forma tal que el desarrollo de la aplicación esté estructurado de una mejor manera, también facilita la programación de forma paralela e independiente en las diferentes capas, además de brindarle una mayor escalabilidad y facilitar el tratamiento de errores (25). A continuación se muestra un diseño de la arquitectura de la solución propuesta, más adelante en la descripción de las clases del diseño se especifica en qué consiste cada una de ellas.

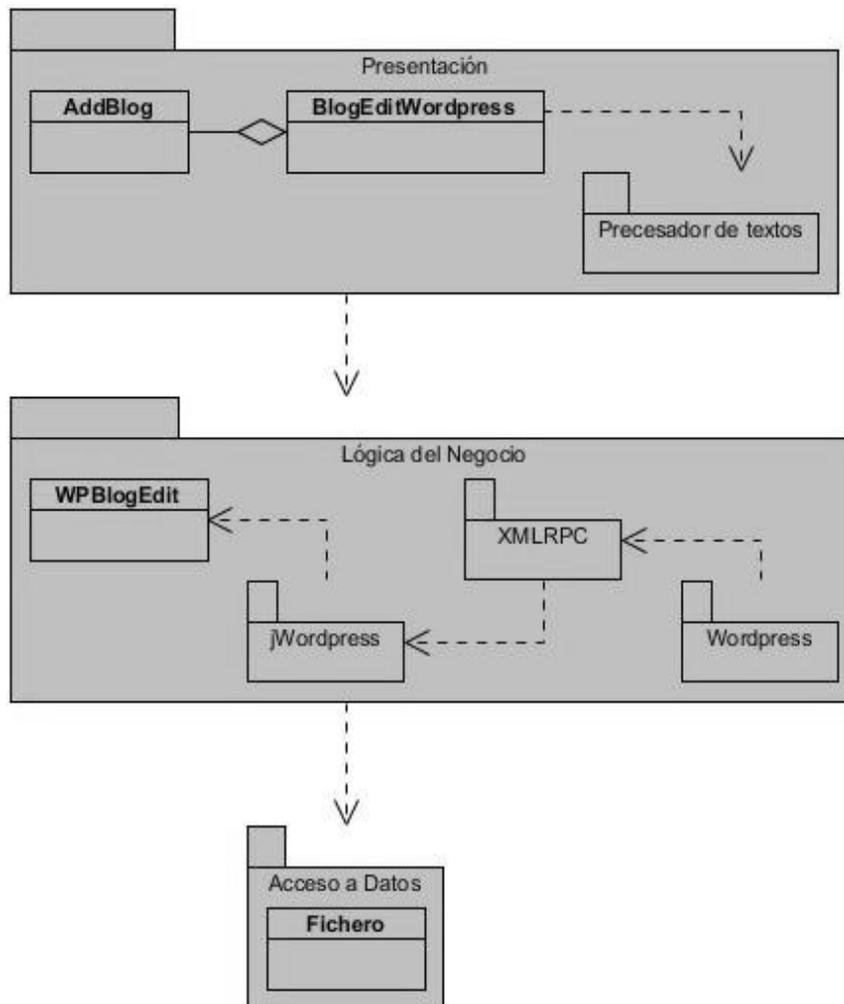


Figura 5 Diagrama de la Arquitectura

Descripción de las capas

Presentación: esta capa contiene las vistas o interfaces de usuario del sistema, además de un paquete que contiene el editor de textos utilizado en la aplicación.

Lógica del Negocio: contiene las clases del diseño y los principales paquetes utilizados para el desarrollo de la solución, donde la clase controladora interactúa con ellos a través de peticiones de forma jerárquica con el fin de mantener el flujo de datos y la comunicación con la plataforma de blogs.

Acceso a Datos: contiene el fichero donde se almacenarán los datos temporales de forma local, el sistema no contiene base de datos, pues los datos que se manejan se obtienen directamente de la plataforma de blogs Wordpress.

2.5.2 Patrones de Diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular (26).

Patrones de asignación de responsabilidad (GRASP)

Patrones de asignación de responsabilidad (GRASP, por sus siglas en inglés “*General Responsibility Assignment Software Patterns*”) son parejas de problema/solución con un nombre, que codifican buenos principios y sugerencias relacionadas frecuentemente con la asignación de responsabilidades. Describen la asignación correcta de responsabilidades en el diseño orientado a objetos (25). A continuación se describen los patrones utilizados y su aplicación en el desarrollo de la solución:

Experto: asigna una responsabilidad al experto en información, que es la clase la cual posee la información necesaria para cumplir con la responsabilidad (25). La solución planteada se desarrolla bajo el concepto de que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para hacerlo. Se evidencia en la clase `WPEditorBlog`, que es la que cuenta con la información necesaria para interactuar y utilizar las librerías encargadas de hacer las peticiones pertinentes en aras de dar cumplimiento al desarrollo de los requisitos propuestos.

```
public WPBlogEditor(String user, String pass, String url)
    Proxy();
    cxn = new Wordpress(user, pass, url);
}
```

Figura 6 Patrón Experto

Controlador: es el encargado de manejar los eventos de entrada del sistema. Define quién debe encargarse de atender un evento del sistema. El controlador es una parte vital de todo el sistema, su función es procesar las diferentes peticiones hechas por el cliente con el objetivo de presentar una respuesta deseada (25), se

evidencia en la clase controladora WPBlogEditor quien se encarga de manejar cada evento en el sistema.

Creador: guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se conecte con el objeto producido en cualquier evento (25). Se evidencia en la clase BlogEditWordPress cuando se crea un objeto de la clase controladora, como se muestra en la figura 7:

```
WPBlogEditor wp = new WPBlogEditor(user, pass, url);
```

Figura 7 Patrón Creador

Bajo Acoplamiento: asigna una responsabilidad para mantener bajo acoplamiento. El grado de acoplamiento no puede considerarse aisladamente de otros principios como Experto y Alta Cohesión. Es un patrón evaluativo que tiene en cuenta el diseñador al realizar mejoras al diseño (25). Se evidencia en todo el diseño del sistema por la poca dependencia entre una clase y otra, asignándole la responsabilidad necesaria a cada una de ellas.

Alta Cohesión: asignar una responsabilidad de modo que la cohesión siga siendo alta. También como el patrón Bajo Acoplamiento es un principio que se deben tener presente en todas las decisiones de diseño. La información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase (25). Evidenciado así en cada clase, almacenando solo la información necesaria para ella.

GoF

Los patrones Gang of Four (GoF), traducido al español Banda de los Cuatro, describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad. Permiten crear grupos de objetos para realizar tareas complejas. Existen tres tipos de estos patrones: de creación que abstraen el proceso de creación de instancias, estructurales que se ocupan de cómo clases y objetos son utilizados para componer estructuras de mayor tamaño y de comportamiento que atañen a los algoritmos y

a la asignación de responsabilidades entre objetos (26). Seguidamente se mencionan los patrones utilizados y sus aplicaciones en la investigación.

Como parte de los patrones **creacionales** que ayudan a solucionar los problemas de creación de instancias, ayudan a encapsular y abstraer dicha creación se utilizó el siguiente:

Instancia única (Singleton): garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia (26). Se muestra en la instancia de la clase Wordpress que inicialmente se declara y es utilizada a lo largo de toda la ejecución del programa y de la que depende la interacción entre el software y la plataforma.

Los patrones de **comportamiento** ofrecen soluciones respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan, para el desarrollo de la solución se utilizó el siguiente:

Mediador (Mediator): define un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto (26). Se evidencia en las librerías que funcionan como mediadoras entre la clase controladora y la plataforma Wordpress con la cual interactúa para el acceso a los datos.

2.5.3 Diagrama de Clases del Diseño

Los diagramas de clases del diseño permiten describir gráficamente las especificaciones de las clases del software. Muestran las clases (descripciones de objetos que comparten características comunes) que componen el sistema y cómo se relacionan entre sí (11)

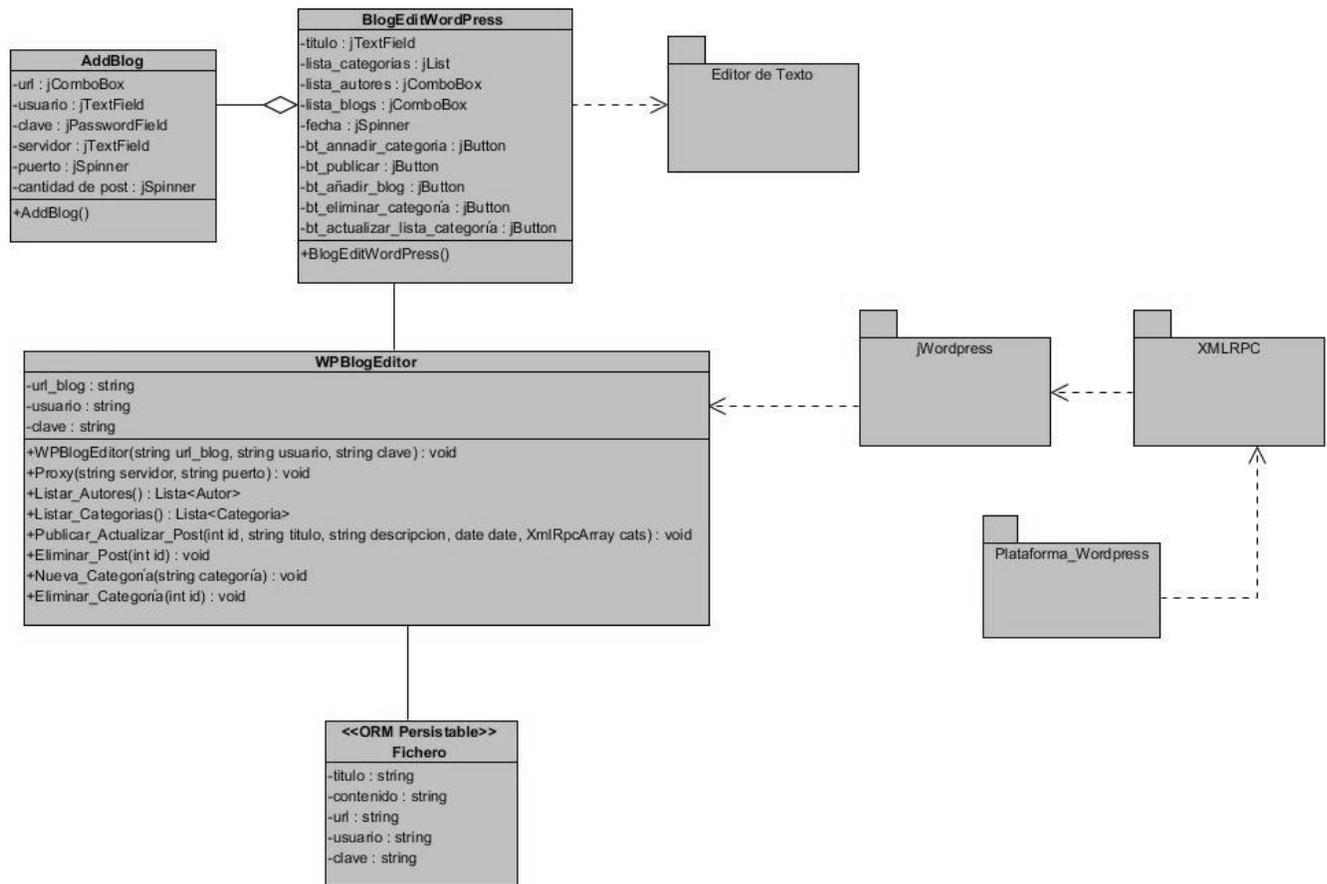


Figura 8 Diagrama de Clases del Diseño

Descripción de las clases del diseño.

BlogEditWordPress: esta clase es la interfaz principal del sistema, en ella se crean o editan los artículos, se añaden nuevos blog, se gestionan las categorías, se listan los autores, se publica la entrada, entre otras funcionalidades del sistema, a través de la clase controladora WPBlogEditor, además se hace uso de una librería representada en el diagrama como un paquete para la edición de textos.

AddBlog: la interfaz encargada de recibir los datos de autenticación del blog y enviárselos a la controladora a través de la interfaz principal para su posterior procesamiento.

WPBlogEditor: es la clase controladora del sistema donde se implementan las principales funcionalidades del sistema interactuando con las librerías usadas para la comunicación con el blog y así enviárselas a la

interfaz principal, también es la encargada de guardar o abrir un fichero local previamente almacenado con el título y contenido de la aplicación.

Fichero: es el encargado de almacenar la información contenida en la interfaz principal del sistema, dígase título y contenido de la entrada.

Descripción de los Paquetes

Los paquetes son un conjunto de librerías y clases encapsuladas como código compilado que son utilizados durante la implementación del sistema y que sirven de gran ayuda al programador.

Editor de Texto: es un editor de textos con herramientas básicas con dicho fin y que es de gran utilidad para la edición de los artículos, en la interfaz principal de hace una llamada a su constructor para así asignarle las funcionalidades incluidas en él, a dicha interfaz.

JWordpress: es una librería de java, utilizada para la comunicación a través del protocolo XMLRPC con la plataforma en línea de Wordpress, contiene la forma de comunicarse con las APIs necesarias para la implementación de todas las funcionalidades que requiere el sistema.

XMLRPC: es un protocolo de comunicación que se encarga de establecer la conexión entre el sistema propuesto y las APIs que brinda Wordpress.

Plataforma_Wordpress: es el servidor en línea de Wordpress desplegado en internet el cual se encarga de dar respuesta a todas las peticiones realizadas desde la clase controladora a través de las librerías y protocolos utilizados.

Conclusiones parciales

En este capítulo se crearon las condiciones necesarias para la implementación y validación de la propuesta de solución, analizando profundamente y obteniendo como base el dominio de la aplicación. Se logró obtener una visión clara y objetiva de los requisitos del cliente. Se especificaron los elementos fundamentales de la arquitectura y el diseño con el empleo de una arquitectura de 3 capas y los diferentes patrones GRASP y GoF respectivamente. A partir de este momento es posible comenzar a construir el sistema teniendo en cuenta el cumplimiento de los requerimientos especificados en este capítulo.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

Introducción

En este capítulo se describen los elementos establecidos por el proceso de desarrollo empleado, que responden a la implementación de la solución. Se plantea la solución desarrollada, se definen los estándares de codificación a emplear para el desarrollo de la propuesta. Se realiza el diseño y aplicación de las pruebas que permiten verificar el funcionamiento de la solución propuesta y la forma de responder a los requerimientos para el cual fue concebido.

3.1 Diagrama de Componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes o bibliotecas cargadas dinámicamente. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente (22).

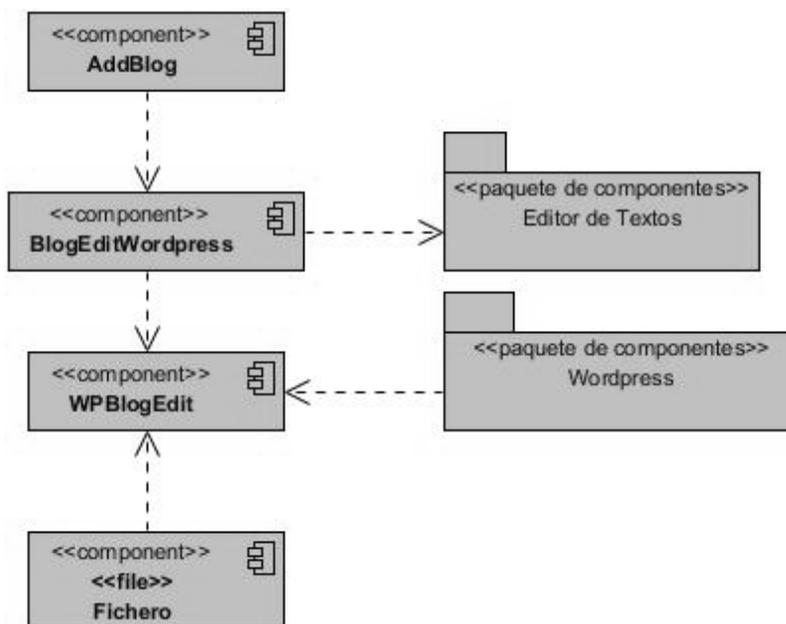


Figura 9 Diagrama de Componentes

3.2 Diagrama de Despliegue

El modelo de despliegue describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional, que generalmente tiene memoria y a menudo, capacidad de procesamiento. Los nodos se utilizan para modelar la topología del hardware sobre el que se ejecuta el sistema. Representan un procesador o un dispositivo sobre el que se pueden desplegar los componentes. La relación entre un nodo y el componente que despliega puede mostrarse con una relación de dependencia (25).

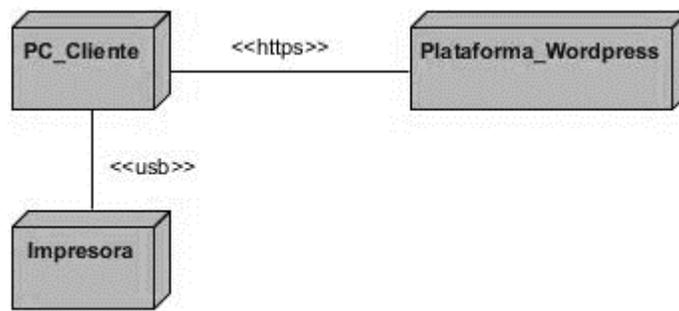


Figura 10 Diagrama de Despliegue del Sistema

Especificación de los nodos y protocolos.

La **PC_Cliente** con los requerimientos mínimos de hardware (detallados en el **RNF 4**) se conecta a través del protocolo seguro **HTTPS** al servidor en línea de la plataforma **Wordpress**, siempre que se acceda a ella mediante el sistema propuesto, la misma se encargará de gestionar todas las peticiones del usuario a través de los servicios que brinda, además la **PC_Cliente** se podrá conectar mediante un cable **USB** a una impresora de manera opcional para imprimir el artículo en edición.

3.3 Estilos de programación

Para el desarrollo del sistema se utilizó el estilo de Programación Orientado a Objetos, por sus siglas en inglés OOP es un método de implementación en el que los programas se organizan como colecciones cooperativas de objetos, cada una representa una instancia de alguna clase, y cuyas clases son, todas ellas,

miembros de una jerarquía de clases unidas mediante relaciones de herencia (27). Java es un lenguaje que permite reutilizar el código utilizando el paradigma de programación orientado a objetos. En él, cada objeto es responsable de inicializarse y destruirse de forma correcta, no existe la necesidad de llamar explícitamente al procedimiento de creación o de terminación debido a que el marco de trabajo se encarga de esto.

3.4 Estándares de codificación

Los estándares de codificación son reglas o patrones de codificación que permiten tener una programación homogénea y que regula la realización de ciertos procesos, de esta forma el sistema resulta fácil de entender y mantener, independientemente del autor (14). En la implementación de la aplicación a desarrollar, se utilizaron varios estándares de codificación, que certificarán legibilidad y organización al código de la misma, simplificando esfuerzos a la hora de darle mantenimiento y seguimiento a la aplicación.

Indentación, llaves de apertura y cierre, y tamaño de las líneas

Se deben utilizar una indentación sin tabulaciones con un equivalente a cuatro espacios. El uso de las llaves “{}” será: la de apertura al final de la línea de código y la de cierre en una nueva línea y la longitud de las líneas es aproximadamente de 75-80 caracteres para mantener la legibilidad del código. A continuación se muestra un ejemplo.

```
public void Nueva_Categoria(String c) throws XmlRpcFault {
    cxn.newCategory(c, c, 89);
}
```

Figura 11 Indentación y llaves

Convención de nomenclatura

Variables: la declaración de las variables se rige por la nomenclatura camelCase. Siempre comienzan con minúscula y en caso de nombres compuestos la separación entre cada palabra será por un guión bajo “_”.

```
public void Prueba() {
    String url_blog = "https://vladistest.wordpress.com/xmlrpc.php";
}
```

Figura 12 Nomenclatura para crear variables

Clases: siempre comienzan con mayúscula, en caso de nombre compuesto cada palabra comenzará con mayúscula.

```
public class BlogEditWordPress extends javax.swing.JFrame {
```

Figura 13 Nomenclatura para crear las clases.

Operadores aritméticos: antes y después de cada operador aritmético debe de existir un espacio.

```
for (int i = 0; i < autores.size(); i++) {  
    nombres[i] = autores.get(i).getDisplay_name();  
}
```

Figura 14 Nomenclatura para operadores aritméticos

Documentación

Todos los archivos deben de tener la documentación asociada al mismo cumpliendo con los siguientes bloques al principio de cada clase o método.

```
/**  
 * Clase controladora WP_Blog_Edit  
 * Esta clase funciona como intermediaria entre la vista  
 * y las librerías.  
 * @category Controladora  
 * @package wp_blog_editor  
 * @author Vladis Miriel Perdomo Hernández <vmperdomo@estudiantes.uci.cu>  
 */
```

Figura 15 Documentación de la clase controladora WP_Blog_Edit.

```
/**  
 * Devuelve una lista de categorías extraídas del blog al que fue conectado.  
 * @author Vladis Miriel Perdomo Hernández <vmperdomo@estudiantes.uci.cu>  
 */
```

Figura 16 Documentación de la funcionalidad Lista de Categorías

Estructuras de control

Las estructuras de control utilizan siempre llaves de apertura y cierre “{}”, incluso en situaciones en las que técnicamente son opcionales. Con esto se aumenta la legibilidad del código y se disminuye la probabilidad de errores lógicos. Incluyen if, for, foreach, while, switch.

```
public String[] Lista_Categorias() throws MalformedURLException, XmlRpcFault {  
    List<Category> categorias = cxn.getCategories();  
    String[] nombres = new String[categorias.size()];  
    for (int i = 0; i < categorias.size(); i++) {  
        nombres[i] = categorias.get(i).getCategoryName();  
    }  
    return nombres;  
}
```

Figura 17 Estructuras de control

3.5 Estrategia de Prueba

Una estrategia de prueba de software integra los métodos de diseño de caso de pruebas del software en una serie bien planeada de pasos que desemboca en la eficaz construcción del software. Esta debe incluir pruebas de bajo nivel para confirmar la correcta implementación de todos los pequeños segmentos de código fuente, así como pruebas de alto nivel que validen las principales funciones del sistema frente a los requerimientos del cliente (11).

Se les denomina pruebas de software al conjunto de técnicas experimentales que se le aplican a los sistemas de software; son un elemento crucial para garantizar la calidad del producto y permiten validar las especificaciones, el diseño y la programación. Estas tienen como objetivo, además de descubrir errores, medir el grado en que el software cumple con los requerimientos definidos (11).

Para probar la solución se definió una estrategia de prueba de software a seguir. Las pruebas fueron realizadas a dos niveles dependiendo de los objetivos que se perseguían con las mismas. Se definieron los métodos de pruebas a aplicarse, las técnicas según el nivel y finalmente se evaluaron las consideraciones y criterios de éxito.

Tipo de Prueba

Cada tipo de prueba tiene un objetivo específico y una técnica que lo soporte. El tipo de prueba utilizado para comprobar el nivel de calidad del software fueron las de funcionalidad, las cuales van enfocadas a la validación de las funciones, métodos, servicios y casos de uso.

Niveles de Prueba

El proceso de realización de pruebas está compuesto por una serie de niveles entre los que se pueden encontrar: el nivel de pruebas unitarias, el nivel de pruebas de integración, el nivel de pruebas del sistema y nivel de pruebas de aceptación. Concluida la implementación el sistema fue sometido a los niveles de prueba que se detallan a continuación, los cuales propiciaron la detección de los errores existentes.

Pruebas Unitarias: Las pruebas de unidad se concentran en la lógica del procesamiento interno y en las estructuras de datos de los límites de un componente. El objetivo de las pruebas unitarias es aislar cada parte del programa y mostrar que las partes individuales son correctas (28).

Las pruebas unitarias se aplican para verificar que el software cumple los requisitos funcionales y no funcionales de su especificación y también son empleadas para asegurar la calidad del código entregado. Además, son la mejor forma de detectar fallas tempranamente en el desarrollo y está demostrado que mientras más pronto se encuentren los errores, menos costará corregirlos.

Pruebas del Sistema: Tienen como propósito fundamental ejercitar profundamente el sistema desarrollado, con el objetivo de verificar que se hayan integrado correctamente todos los elementos del sistema y que realizan correctamente las funciones descritas. Este tipo de pruebas estudia el producto completo para analizar defectos globales o para estudiar aspectos específicos de su comportamiento, tales como seguridad o rendimiento (28). En la investigación se utilizaron las pruebas funcionales.

3.5.1 Métodos de Prueba

La metodología Open UP propone 2 métodos fundamentales que serán usados en el proceso desarrollo de pruebas al sistema, los cuales serán descritos a continuación.

Método de Caja Blanca

Es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante este método, el ingeniero del software puede obtener casos de prueba que garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo, ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa, ejecuten todos los ciclos en sus límites y con sus límites operacionales, y ejerciten las estructuras internas de datos para asegurar su validez. Estas pruebas se gestionan por informáticos, algunas veces ayudados por analistas (29).

Para la aplicación de este método se utilizó una **técnica automatizada de software**, la cual permite realizar pruebas al código del sistema utilizando la herramienta JUnit que está diseñada especialmente para

implementar y automatizar la realización de pruebas de unidad en Java.

Método de Caja Negra

También denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software. Se llevan a cabo sobre la interfaz del software para intentar descubrir diferentes tipos de errores en estas. El objetivo es demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene. Estas pruebas se realizaron a nivel de sistema y se centran principalmente en los requisitos funcionales del software. Si las salidas no son las previstas, entonces la prueba ha detectado un fallo en el software.

La técnica utilizada en este método fue **partición de equivalencia** mediante el diseño de Casos de Pruebas (CP), esta técnica se basa en la división del campo de entrada en un conjunto de clases de equivalencia, estas clases representan un grupo de datos de entrada que definen estados válidos y no válidos del sistema y se obtienen a partir de los casos de prueba. Se elaboran por CU o por requisitos, su objetivo principal es proveer una guía de cómo realizar la prueba, mostrando los pasos a seguir y los datos a introducir con la finalidad de obtener el resultado esperado (11).

A continuación se muestran los diseños de CP correspondientes a los dos CU definidos anteriormente, para verlos en su totalidad dirigirse a los anexos.

Tabla 8 Diseño del CP Autenticar Usuario.

Escenario	Descripción	URL	Usuario	Contraseña	Respuesta del sistema	Flujo central
EC 1.1 Autenticar Usuario	Mediante este escenario se accede al blog con el usuario previamente creado en la plataforma	V	V	V	El sistema se autentica en la plataforma.	El usuario introduce la URL del blog, posteriormente el usuario y la contraseña correspondientes a ese
		https://vladistest.wordpress.com/	vladistest	*****		
		I	V	V	Muestra un mensaje de error "Verifique los parámetros	
		https://wordpress.com/	vladistest	*****		

	Wordpress.				de conexión o la URL no es válida”	blog y de ser necesario edita los parámetros de conexión que son el Servidor y el Puerto de la salida a internet, seguido pincha en aceptar.
		V	I	V	Muestra un mensaje de error	
		https://vladist est.wordpress.com/	vlad93test	*****	“Usuario o Contraseña incorrecta”	
		V	V	I		
		https://vladist est.wordpress.com/	vladistest	*****		

Tabla 9 Diseño del CP Crear y Modificar Entrada del CU Gestionar Entrada

Escenario	Descripción	Título	Contenido	Categorías	Respuesta del sistema	Flujo central
EC 1.1 Crear una nueva entrada	Mediante este escenario se introducen los datos correctos o se deja en blanco alguna variable de los diferentes campos para crear una	V	V	V	Se publica la entrada con todas las variables introducidas de forma válida o dejando las demás en blanco.	El usuario una vez autenticado en el blog selecciona de la lista de entradas despegable el ítem “Nueva Entrada” el cual limpiará todos los
		Probando una nueva entrada.	Vamos a probar la creación de una nueva entrada.	- Educación - Material - Cuba		
		V	V	NA		
		Probando una nueva entrada.	Vamos a probar la creación de una nueva entrada.			
V	NA	V				

	nueva entrada.	Probando una nueva entrada.		- Educación - Material - Cuba		campos de las variables necesarias para la creación de una nueva entrada. Una vez creado y listo para publicar
		NA	V	V		selecciona el botón
			Vamos a probar la creación de una nueva entrada.	- Educación - Material - Cuba		publicar en la barra de herramientas . Se muestra un mensaje indicando que la
		NA	NA	NA	Se publica la entrada con título y contenido en blanco y una categoría que el sistema le asigna por defecto.	que la entrada fue publicada satisfactoriamente.
EC 1.2 Modificar Entrada	Mediante este escenario se modifican los datos de los diferentes campos que se pueden editar.	V	V	V	Se publica la entrada modificando todas las variables de forma válida o dejando las demás en blanco.	
		Modificando la entrada seleccionada .	Vamos a probar modificar un artículo.	- Educación - Material - Cuba		
		V	V	NA		
		Modificando la entrada seleccionada .	Vamos a probar modificar un artículo.			
		V	NA	V		

		Modificando la entrada seleccionada		- Educación - Material - Cuba		
		NA	V	V		
			Vamos a probar modificar un artículo.	- Educación - Material - Cuba		
		NA	NA	NA	Se modifica la entrada con título y contenido en blanco y una categoría que Wordpress le asigna por defecto.	

Tabla 10 Diseño del CP Obtener Entrada del CU Gestionar Entrada

Escenario	Descripción	Cantidad de Entrada a obtener	Respuesta del sistema	Flujo central
EC 1.1 Obtener Entrada	Mediante este escenario se obtienen una cantidad de entradas recientes definida por el usuario al	V 5	Muestra una lista despegable con el título de los artículos mas recientes del blog autenticado y muestra el contenido de la entrada seleccionada.	El usuario una vez autenticado en el blog especifica la cantidad de entradas que desea obtener y se muestra el título de estos en una lista despegable. Permittedole al usuario elegir el artículo y visualizar su

	autenticarse en el blog.	I	El sistema muestra un mensaje de error: "Debe de especificar una cantidad de artículos mayor que 0"	contenido.
		0		

Tabla 11 Diseño de CP Eliminar Entrada del CU Gestionar Entrada

Escenario	Descripción	Título	Respuesta del sistema	Flujo central
EC 1.1 Eliminar Entrada	Mediante este escenario se elimina una entrada seleccionada.	V	Elimina la entrada seleccionada y actualiza la lista de entradas recientes.	El usuario una vez autenticado en el blog selecciona de la lista despegable de artículos el que desea eliminar y selecciona el botón eliminar de la barra de herramientas.
		Visitó a Fidel el Presidente serbio.		
		I	El sistema muestra un mensaje de error: "Debe de seleccionar un ítem válido"	
		Nueva Entrada		

Resultados de las pruebas de Caja Blanca

Las pruebas unitarias aplicadas al sistema fueron los test unitarios haciendo uso de junit. Estas pruebas fueron llevadas a cabo por el desarrollador, encargado de verificar el código de forma automática, se aplicaron a todas las funcionalidades implementadas en el sistema comprobando el correcto funcionamiento de cada una de estas y obteniendo resultados satisfactorios.

A medida que se fueron realizando las pruebas correspondientes se detectaron un total de 4 no conformidades, las cuales se resolvieron mientras se desarrollaban las funcionalidades. Una vez realizada la corrección de los errores detectados en pequeñas iteraciones, los métodos de cada clase se ejecutaron de forma correcta, obteniendo de ellos los resultados esperados, como se muestra a continuación:

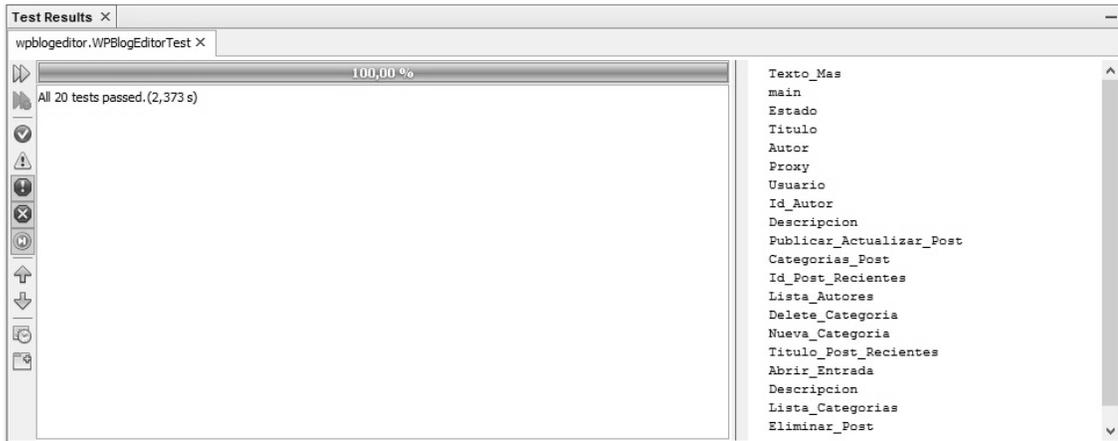


Figura 18 Resultado de las pruebas unitarias para la clase controladora

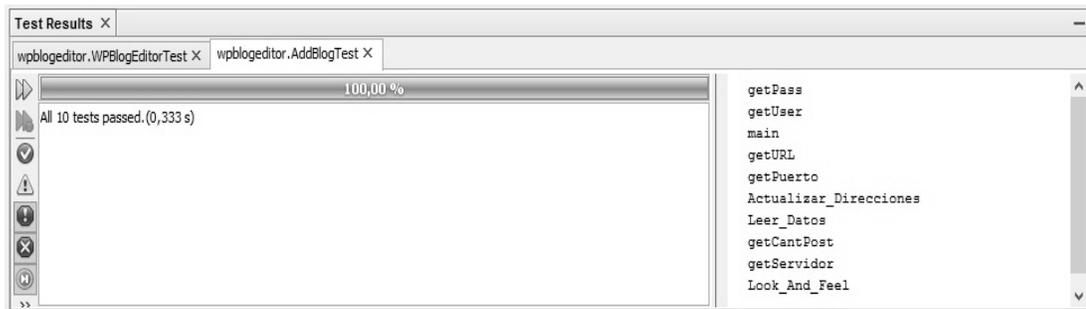


Figura 19 Resultado de las pruebas unitarias para la interfaz Establecer Conexión.

Resultados de las pruebas de Caja Negra

Para evaluar la solución se realizaron 2 iteraciones donde se probó el software íntegramente, finalmente se realizó una prueba final donde se comprobó la resolución de todas las no conformidades detectadas. A continuación se presentan los resultados arrojados durante las diferentes pruebas aplicadas:

Tabla 12 Resultado de las pruebas de Caja Negra en la Iteración 1

Casos de Prueba	No Conformidades			
	Alta	Media	Baja	Total
Autenticar Usuario en el blog.	1	0	1	2
Gestionar Entrada.	3	2	1	6
Programar Publicación.	1	0	1	2
Insertar Imagen.	3	2	1	6

Asignar Autores.	1	0	0	1
Gestionar Categoría	2	0	3	5
Actualizar Entrada.	0	1	1	2
Publicar Entrada.	0	1	1	2
Total.	11	6	9	26

Tabla 13 Resultado de las pruebas de Caja Negra en la Iteración 2

Casos de Prueba	No Conformidades			
	Alta	Media	Baja	Total
Autenticar Usuario en el blog.	0	0	1	1
Gestionar Entrada.	1	0	0	1
Programar Publicación.	0	0	1	1
Insertar Imagen.	1	1	1	3
Listar Autores.	0	0	0	0
Gestionar Categoría	1	0	1	2
Actualizar Entrada.	0	1	0	1
Publicar Entrada.	0	0	1	1
Total.	3	2	5	10

Las pruebas se centraron en el cumplimiento de las funcionalidades descritas en el listado de requerimientos y de casos de uso del sistema, además, se revisó toda la documentación generada. Las No Conformidades (NC) se clasificaron en Alta (A), Media (M) o Baja (B) en dependencia del impacto que tuvieran, generalmente las altas responden a errores técnicos relacionados directamente con la funcionalidad interna del CU y las bajas tienden a ser errores ortográficos, validaciones entre otros errores de bajo impacto.

Se llevaron a cabo 3 iteraciones donde se obtuvieron varias NC, en la primera iteración se detectaron 26 NC de las cuales 11 fueron A, 6 M y 9 B, las cuales fueron eliminadas mediante el transcurso de cada una de las iteraciones definidas; para la segunda iteración se detectaron 10 NC distribuidas en 3 A, 2 M y 5 B; en la iteración final ya no se encontraron NC, por lo tanto las pruebas de funcionalidad en el sistema fueron realizadas satisfactoriamente. Las mismas permitieron garantizar que el sistema cumple con las

especificaciones que se trazaron y de este modo se dio cumplimiento a cada uno de los requisitos definidos en el transcurso de la investigación. Para mayor entendimiento de este proceso a continuación se muestra un gráfico de barras que ilustra las NC por cada iteración.

El gráfico de barras (Figura 17) muestra los principales resultados de las pruebas de caja negra realizadas. Como se puede apreciar, se realizaron dos iteraciones de pruebas, detectando en la primera y segunda iteración un total de 26 y 10 no conformidades respectivamente. Comparando los resultados entre dichas iteraciones se evidencia una reducción considerable de los principales errores detectados en el sistema una vez ejecutada la segunda iteración. Finalmente se realizó una última prueba donde se corrigieron todas las no conformidades.

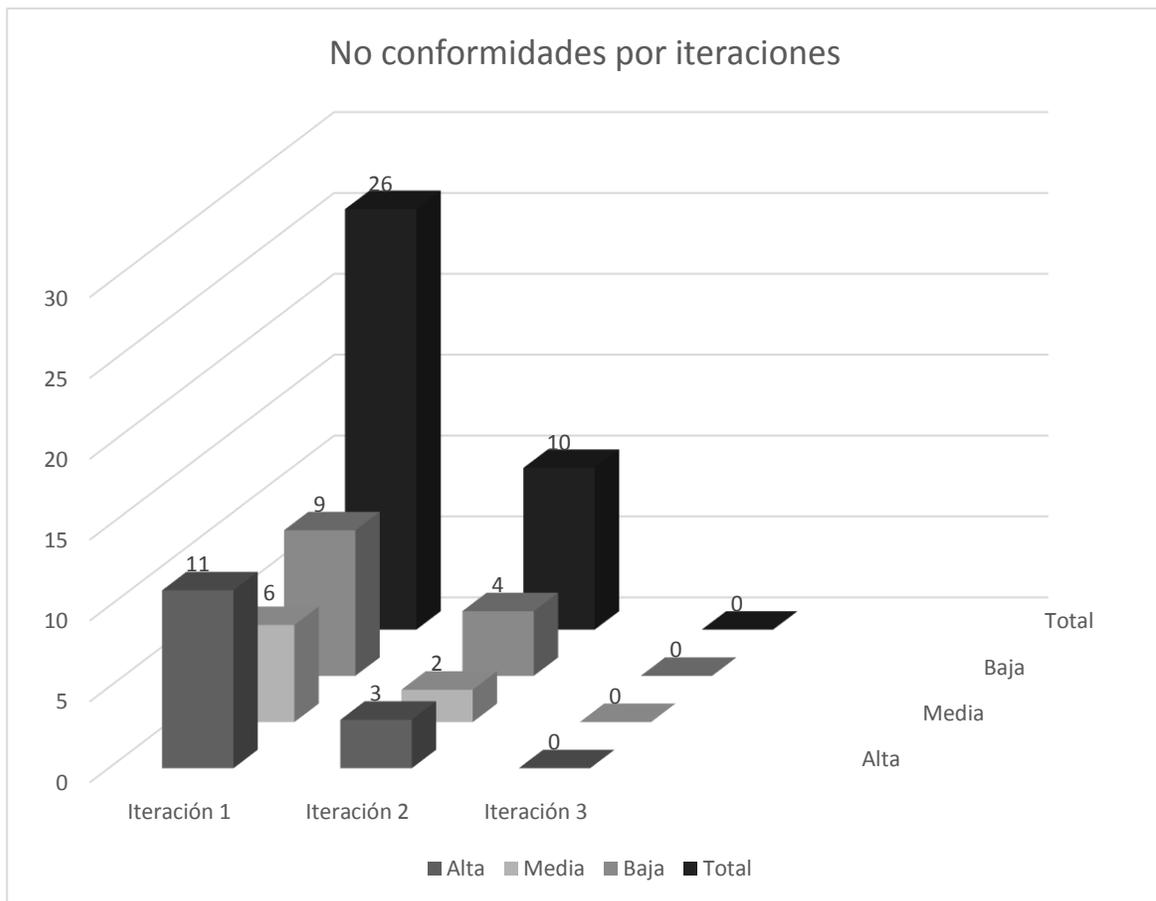


Figura 20 Resultado de las no conformidades por iteraciones.

Conclusiones parciales

En este capítulo se ha llevado a cabo la descripción de cómo las clases y los objetos se organizan en términos de componentes. Se explicó cómo está estructurada la aplicación físicamente, mediante los modelos de despliegue y de componentes. La calidad del sistema desarrollado fue el elemento clave del capítulo que recién concluye. En ese sentido se efectuaron pruebas de software en el nivel de unidad mediante casos de pruebas, para los cuales se tuvieron en cuenta las entradas, las salidas, los resultados esperados y el tratamiento de errores en caso de anomalías, las cuales arrojaron resultados satisfactorios para cada uno de los casos correspondientes

CONCLUSIONES GENERALES

Una vez terminado el proceso de desarrollo de la solución propuesta se puede concluir lo siguiente.

1. El estudio de homólogos realizado en los inicios de la investigación, permitió determinar que no existían herramientas similares que fueran capaces de dar solución a la problemática propuesta, a su vez permitió nutrirse de ideas, conceptos y procedimientos que fueron utilizados para implementar la solución.
2. La utilización de los diagramas de ingeniería tomando como guía la metodología seleccionada, permitió modelar una abstracción de la solución propuesta en forma de diseño para facilitar la implementación posterior de los los requerimientos funcionales identificados.
3. Los artefactos generados durante la implementación de la aplicación permitieron guiar la materialización de los requisitos en forma de componentes de código y elementos que, guiados por patrones de diseño, arquitectura y estándares de código, posibilitaron la creación de la herramienta que da solución al problema planteado.
4. La utilización de las pruebas de caja negra y caja blanca permitió comprobar el éxito del cumplimiento de las funcionalidades del sistema propuesto e implementado en este trabajo.
5. La herramienta desarrollada constituirá un aporte para el departamento DOWAI, ya que en lo adelante contará con una forma sencilla e innovadora de actualizar sus blogs y brindará la posibilidad de acceder directamente a la plataforma sin utilizar herramientas privativas evitando las limitaciones que acarrea su uso, y cumpliendo las políticas del país en cuanto al uso de herramientas y tecnologías libres y la ampliación de la soberanía tecnológica.

RECOMENDACIONES

Tomando como base la investigación realizada y la experiencia acumulada durante la realización de este trabajo de diploma, se proponen las siguientes recomendaciones:

- Agregar al sistema la opción de gestionar etiquetas, que es de gran utilidad para la publicación de los artículos.
- Almacenar localmente las entradas obtenidas para una mayor velocidad de edición.
- Implementar un sistema de trazas en aras de aumentar la seguridad del sistema.
- Contribuir al perfeccionamiento e integración de nuevas funcionalidades que posibiliten una mayor integridad y eficiencia del mismo.

Se estimula a todos los interesados en el estudio de este trabajo a seguir investigando, para que de esta forma, sirva de material de consulta en la realización de posteriores trabajos relacionados con el tema.

REFERENCIAS

1. *WEBLOGS EN EDUCACIÓN*. **Contreras, Fanny Contreras**. 10, 2004, Revista Digital Universitaria, Vol. 5, págs. 3, 4.
2. **Enric Bruguera, Joan Campás**. *El hipertexto y Los blogs*. 1. Barcelona : UOC, 2007.
3. **Winner, Dave**. What makes a weblog a weblog? [En línea] 2003. [Citado el: 5 de diciembre de 2014.] <http://blogs.law.harvard.edu/whatM>.
4. *'WordPress' y la creación de un sitio Web dinámico: metodología de instalación y puesta en marcha*. **Claudio Escobar Arraigada, Joel Lagos Llancao**. 10, Chile : Serie Bibliotecología y Gestión de Información, 2006.
5. **Wordpress.org**. Wordpress. [En línea] Wordpress.com. [Citado el: 12 de diciembre de 2014.] <https://www.wordpress.com>.
6. **Rodríguez-Aragón, Licesio J**. *Internet y Teleinformática*. Buenos Aires : s.n., 2012.
7. *Protocolos de Comunicación Utilizados en Cloud Computing*. **Quispe, Carmen Mamani**. Buenos Aires : REVISTA DE INFORMACIÓN TECNOLOGÍA Y SOCIEDAD, 2012.
8. **Scribd**. [En línea] 9 de Junio de 2011. [Citado el: 6 de diciembre de 2014.] <http://es.scribd.com/doc/64077925/Metodologias-de-Desarrollo-de-Software>.
9. *Metodologías y ciclo de vida*. **INTECO**. 2009.
10. *Metodologías Tradicionales vs. Metodologías Ágiles*. **Figueroa, Roberth G., Solís, Camilo J., Cabrera, Armando A**. 2010.
11. **Pressman, Roger**. *Ingeniería de Software: Un Enfoque Práctico*. 6ta. Edición. s.l. : McGraw-Hill, 2005. pág. 900. ISBN: 9701054733.
12. **González, Daniel, Gutiérrez Hernández, Miguel y Hernández Garduño, José**. *JAVA para estudiantes*. Tercera edición. México : Person Educación, 2003. pág. 664. ISBN: 970-26-0144-4.
13. **Oracle Corporation**. NetBeans. [En línea] enero de 2010. [Citado el: 12 de enero de 2015.] <https://netbeans.org>.
14. **SEBESTA, ROBERT W**. *CONCEPTS OF PROGRAMMING LANGUAGES*. New Jersey : TENTH EDITION, 2012.
15. **Javier García de Jalón, José Ignacio Rodríguez, Íñigo Mingo, Aitor Imaz, Alfonso Brazalez, Alberto Larzabal, Jesús Calleja, Jon García**. *Aprenda Java como si estuviera en primero*. San Sebastián : TECNUN, 2000.

16. **Suyama, Maria.** DesarrolloWeb.com. [En línea] 30 de Agosto de 2004. [Citado el: 10 de Febrero de 2015.] <http://www.desarrolloweb.com/articulos/1617.php>.
17. **Cárdenas, Dr.Roberto Gómez.** Desarrollo de Aplicaciones Distribuidas. *XML-RPC*. [En línea] [Citado el: 20 de Febrero de 2015.] <http://homepage.cem.itesm.mx/rogomez>.
18. **Hexidec Codex.** [En línea] 8 de Junio de 2010. [Citado el: 8 de Febrero de 2015.] www.hexidec.com.
19. **Fowler, Martin y Sccott, Kendall.** *UML Gota a Gota*. 1999.
20. **López, Patricia.** *Herramienta CASE Visual Paradigm*. INGENIERÍA DEL SOFTWARE, Universidad. Cantabria : Universitaria, 2014.
21. **Dueñas Rodríguez, Ernesto, Cabrera González, Lianet y Pompa Torres, Enrique Roberto.** *uciencia*. [En línea] 19 de Diciembre de 2013. [Citado el: 14 de enero de 2015.] <http://uciencia.uci.cu/?q=node/2405>.
22. **Sommerville, Ian.** *Ingeniería de Software*. [ed.] Miguel Martín Romo. 7ma. Edición. Madrid : PEARSON EDUCACION, S.A., 2005. ISBN: 84-7829-074-5.
23. **WEITZENFELD, A.** *Ingeniería de software orientada a objetos con UML, Java e Internet*. s.l. : Cengage Learning Editores, 2005.
24. **Raul Pérez, Yelaine.** *Propuesta de arquitectura para agilizar el desarrollo de sistemas de gestión web sobre portales empresariales libres*. . 2009.
25. **Larman, Craig.** *UML y Patrones Introducción al análisis y diseño orientado a objetos*. [trad.] Luz Maria Henhndez Rodriguez y Humberto Cárdenas Anaya. México : PRENTICE HALL, 1999. pág. 536. ISBN: 970-17-0261-1.
26. **Gamma, Erich, y otros.** *Desing Patterns: Elements of Reusable Object-Oriented Software*. s.l. : Addison Wesley, 1995. ISBN:0-201-63361-2.
27. **Booch, Grady, Cueva Lovelle, Juan Manuel y Cernuda del Río, Agustín.** *Analisis y Diseño Orientado a Objetos con Aplicaciones*. s.l. : ALHAMBRA MEXICANA, S.A, 2011. pág. 638. ISBN: 9789684443525.
28. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El Proceso Unificado de Desarrollo de Software. Manual de Referencia*. s.l. : S.I.: Addison Wesley.
29. **Guérin, Brice Arnaud.** *Gestión de Proyectos Informáticos. desarrollo, análisis y control*. Barcelona : s.n., 2012. ISBN: 978-2-7460-7518-4.

30. Ecured. Enciclopedia Colaborativa Cubana. [En línea] 14 de diciembre de 2010. [Citado el: 6 de diciembre de 2014.] http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3n..
31. JACOBSON, I. *El proceso unificado de desarrollo de software*. Pearson Educación S.A. 2000.

ANEXOS

Anexo 1 Entrevista a los especialistas editores de blog y analista de la información del departamento DOWAI.

Tabla 14 Relación de los entrevistados durante el proceso de desarrollo.

Nombre y Apellidos	Cargos
Ing. Martha Luisa Gala Rodríguez	Jefa de línea de Operaciones web y editora de Blog Ciberlady
Lic. Waldo Barrera Martínez	Especialista de la línea de Operaciones web y editor del Blog Jóvenes por los 5
Lic. Andy Bermellón Campos	Especialista de la línea de Operaciones web y editor del Blog CubaxDentro
Ing. Lilian Cid Escalona	Especialista de la línea de Operaciones web y editora del Blog Cybermambi
Ing. Juan Manuel Álvarez Tur	Analista de Información

Entrevista

Soy estudiante de la carrera de Ingeniería en Ciencias Informáticas de la UCI, realizo un proyecto de investigación del que usted forma parte del proceso objeto de estudio. Considero que tiene información valiosa para el desarrollo de esta investigación, por lo que solicito su colaboración. Sus respuestas serán de carácter confidencial.

Gracias de antemano.

Objetivo: Obtener información acerca del proceso de actualización de los blogs y la mayor parte de los requisitos funcionales del sistema que se está desarrollando.

1. ¿Cómo valora el empleo de los editores de blog sin conexión en el departamento DOWAI?
2. ¿Con que frecuencia utilizaba Windows Live Writer antes de migrar a Software Libre?

3. ¿Aproximadamente cuál es el valor aproximado del consumo de su cuota de acceso a internet actualmente con la edición en línea de los blog administrados por DOWAI?
4. ¿Qué funcionalidades le interesaría a usted como autor de artículos en Wordpress, que tuviera una herramienta de publicación sin conexión?
5. ¿Cuáles serían a su consideración las opciones que no podrían faltar en una aplicación de este tipo?
6. ¿Qué flujo de trabajo propondría para que la herramienta de publicación sin conexión sea idónea y práctica?

Nota: Para ver todos los anexos dirigirse a la versión digital del presente documento.

GLOSARIO DE TÉRMINOS

Etiqueta: Son palabras claves relevantes que están asociadas a una entrada de un blog, tales como una imagen, un video clip o que se le asignan a un conjunto de datos y recursos para organizarlos, y así poder relacionar sus contenidos entre sí.

Categoría: Es la clasificación de los artículos de un blog por temas para facilitar su búsqueda en un repositorio de datos.

Sin conexión: Desconectado.

En línea: Conectado.

Web Social: Se centra en posibilitar y promover que los usuarios creen y compartan información, participen, interactúen en tiempo real y colaboren entre sí convirtiéndose en protagonistas de la misma.

RSS: Son las siglas de *Really Simple Syndication*, es un formato XML para syndicar o compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos.

Artículo: Es un texto que presenta la postura personal de un periodista respecto a un determinado acontecimiento, o problema, o asunto actual, o de interés general o histórico.

Entrada: Son los artículos que se van publicando en la estructura del blog, y que se ordenan de manera cronológica.

WYSIWYG: Es el acrónimo de *What You See Is What You Get* ("lo que tú ves es lo que obtienes", en español) y caracteriza editores en los que de manera visual se pueden colocar distintos elementos sobre una vista previa de la página, encargándose el programa de generar el documento HTML. La manera de trabajar en este tipo de editores es similar a la que se usa cuando se trabaja con un procesador de texto.

Windows Live Writer: Desarrollado por Microsoft, es una aplicación para la edición y publicación de blogs. Incluye edición **WYSIWYG**, la funcionalidad de foto-publicación y mapa-publicación.