



UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS

Facultad 5



UCIBlogs, plataforma para la gestión de blogs educativos en la Universidad de las Ciencias Informáticas.

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

**Autor(es): Ana María Domínguez Garcell.**

**Rigoberto Jiménez Pérez.**

**Tutor: Ing. Yaima Fiallo Valle.**

La Habana, junio de 2015

Año 57 de la Revolución



*"La mejor manera de proteger el conocimiento es hacerlo libre."*

*Tiscar Lara*

## Agradecimientos

---

Quiero agradecer a todas las personas que en algún momento de mi vida tuvieron algo que ver con la realización de este largo camino que es lograr un título universitario. A mi madre Vivian a quien le dedico especialmente esta tesis, a mi padre Julio César por estar ahí siempre cuando lo he necesitado, a los dos por darme una vida digna y feliz, a Andro por estar al lado de mi madre todo este tiempo apoyándome y aconsejándome como un padre, a Andrito mi único hermano por ser tan especial y cariñoso, a mi novio Rigo por ser el mejor amigo y compañero de estudio durante estos 5 años de carrera, siendo una pieza fundamental en esta parte de mi vida. A toda mi linda familia tanto la materna como la paterna por ser tan unidos y personas de bien. A mis suegros Leonor y Santi por ser tan especiales conmigo.

**¡UN MILLON DE GRACIAS A TODOS!**

Ana María Domínguez Garcell

## Agradecimientos

---

*En primer lugar le agradezco a mi padre que aunque Dios se lo llevó muy temprano de mi vida, desde el cielo me ha guiado y me ha dado la fuerza para lograr todos los resultados que he alcanzado y pensar siempre en él me ha dado la fuerza para superar todos los obstáculos que me ha puesto la vida. A mi abuela Leonor (mi buli) por ser una abuela especial, por apoyarme en todo, por quererme tanto y por ser la mejor abuela del mundo. A mi madre por ser una madre ejemplar y exigente, la cual hizo que sea una persona de bien, por apoyarme en todas mis decisiones, por dedicarme toda la vida desde que nací sin importarle sus intereses personales, por poner a sus hijos en primer lugar ante todo, por quererme tanto y por ser tan especial.*

*A Santi mi padrastro, que es una persona a la cual admiro mucho por su inteligencia y por ser una persona tan brillante, gracias Santi por haberme apoyado tanto desde que me conociste, por tus consejos, por quererme tanto como si fuera tu hijo, gracias por ser como eres y por cuidar de mi madre como lo has hecho siempre, quiero que sepas que si no hubiera tenido un padre tan especial como el que tuve, te llamaría padre sin ningún problema.*

*A mis hermanos Rauli, Yamilé y Laura, gracias por ser mis hermanos, A mi hermano Rauli por ser el hermano mayor y darme todos los consejos que me dio para ser una mejor persona, gracias por todo lo que haces por mí. A yami por ser una hermana muy cariñosa y atenta. A mi hermanita Lauri que no por ser la última es la menos importante, gracias por llenarnos de alegría cuando más la necesitamos, gracias por ser tan cariñosa conmigo, por quererme tanto y siempre pensar en mí.*

*A Ana María (mi Puli) mi compañera de tesis, de estudio, mi novia, mi todo, gracias por estos cinco años tan maravillosos que pasamos juntos, gracias por estar conmigo siempre estos cinco años de mi vida, por el apoyo, por el amor, por el cariño y por haber estado junto a mí en los momentos buenos y malos por los que hemos pasado, quiero que sepas que te amo mucho y siempre lo haré.*

*A Vivian, Andro y Andrito por haberme acogido en su familia como un miembro más, por el apoyo, por los consejos. A Vivian mi suegra, por ser una suegra especial, le agradezco muchísimo por haberme apoyado tanto y por tratarme como si fuera su hijo y le agradezco infinitamente por haber apoyado tanto mi relación con Ana María.*

*A Dariel, Richard, Eric y Lester por ser los mejores amigos que he tenido en la vida, gracias por su apoyo y por haber estado conmigo en los momentos malos y buenos de mi vida. Gracias a los 4 por haber estado junto a mí en el momento más difícil de mi vida, cuando perdí a mi padre, gracias por haber estado ahí conmigo.*

*A Carli, Yaikel, Coco y Melisa por permitirme el placer de conocerlos y haber pasado el último tiempo de mi carrera con ellos y haber escuchado tantas veces la exposición de la tesis, siempre ayudando a que todo saliera perfecto. A todos mis compañeros de aula y de la UCI en general, no los pongo a todos porque es imposible, me harían falta 100 hojas, gracias por ser buenos compañeros.*

*Por último y no por ser la menos importante a mi tutora: Yaima Fiallo, gracias por haber aceptado ser nuestra tutora a pesar de que fue a última hora, gracias por todo el apoyo, la ayuda y la contribución a lograr este gran resultado. Te agradezco el doble porque a pesar de tus problemas personales siempre estuviste para ayudarnos.*

**Rigoberto Jiménez Pérez**

## Dedicatoria

---

*Le dedico esta tesis a mi madre Vivian Garcell, esto representa muy poco comparado con todo lo que tú has hecho por mí y no me alcanzará la vida para agradecerte, eres la mejor madre del mundo. Gracias por darme de lo bueno siempre lo mejor, por tu apoyo incondicional, gracias por la vida tan linda y grandiosa que me has regalado, mil gracias por hacerme este largo camino fácil y por ayudarme a cumplirlo. Te lo debo todo a ti madre querida.*

*Ana María.*

*Les dedico todo este logro, esfuerzo y sacrificio a las tres personas más importantes de mi vida:*

*A mi padre que está en el cielo y fue el mejor padre del mundo, que en donde quiera que esté, sé que está muy orgulloso de mí y lo estará siempre.*

*A mi madre porque sé que le he hecho un sueño realidad: ver a su hijo graduarse de Ingeniero.*

*A mi buli, mi abuelita linda, te dedico esto con todo el amor de mi corazón.*

*Rigo.*

## Declaración Jurada de Autoría

---

Por este medio declaramos que nosotros Ana María Domínguez Garcell y Rigoberto Jiménez Pérez somos los únicos autores de este trabajo y autorizamos al Centro de Informatización Universitaria de las Universidad de las Ciencias Informáticas para que haga el uso que estime conveniente.

Para hacer constar lo antes planteado firmamos a los \_\_\_\_ días del mes \_\_\_\_ del año \_\_\_\_\_.

Firma de los autores:

\_\_\_\_\_

Ana María Domínguez Garcell

\_\_\_\_\_

Rigoberto Jiménez Pérez

Firma del tutor

\_\_\_\_\_

Ing. Yaima Fiallo Valle

## Resumen

---

La educación en el siglo XXI se encuentra en un proceso de transformación donde tiene gran impacto la introducción de las tecnologías de la información y las comunicaciones (TICs). La Web 2.0 se caracteriza por los aspectos colaborativos, el diálogo y la participación en la construcción del conocimiento, potenciando los procesos de trabajo colectivo. Formando parte de ésta surgen los blogs educativos, los cuales se usan con fines docentes o en entornos de aprendizaje, tanto por alumnos como por profesores. Como solución se desarrolló una plataforma web para la gestión de blogs educativos que permita a los estudiantes y profesores de la Universidad de las Ciencias Informáticas publicar, archivar, comentar y compartir información académica que favorezca el intercambio del conocimiento. Se determinaron así las principales características o funcionalidades que debe cumplir un blog educativo, destacando los beneficios que aporta al apoyo del proceso de enseñanza y aprendizaje. También se realizó un análisis de las plataformas internacionales para la creación y gestión de los mismos. Se emplearon las herramientas de desarrollo, PhpStorm, Slim, PhpMyAdmin, Apache, XAMPP, VisualSVN Server Manager y TortoiseSVN. Los lenguajes del lado del cliente establecidos fueron HTML 5, CSS 3, JavaScript, como lenguaje del lado del servidor PHP y como gestor de bases de datos MySQL. Todo el proceso de desarrollo fue guiado por la metodología XP, capaz de adaptarse y responder con agilidad al cambio.

Palabras clave: blogs educativos, gestión, plataforma.

# Índice

## Tabla de contenido

Índice .....	8
Introducción .....	10
Capítulo 1: Fundamentación teórica .....	14
1.1 Los Blogs.....	14
1.2 Los blogs como apoyo al proceso de enseñanza-aprendizaje .....	16
1.3 Los edublogs o blogs educativos.....	16
1.3.1. Beneficios de los blogs educativos .....	17
1.4 Conclusiones del estudio de los blogs.....	18
1.5 Plataforma web.....	18
1.5.1. Plataformas web para la creación de blogs.....	19
1.6 Metodología, tecnologías y herramientas para el desarrollo. ....	20
1.6.1. Metodologías tradicionales o pesadas. ....	20
1.6.2. Metodologías ágiles o ligeras.....	22
1.6.3. Valoración de la selección.....	23
1.7 Herramienta para el modelado de la solución .....	25
1.8 Lenguajes para el desarrollo .....	25
1.8.1. Lenguaje de modelado.....	25
1.8.2. Lenguajes y tecnologías del lado del cliente .....	26
1.8.3. Lenguaje del lado del servidor .....	27
1.9 Gestor de bases de datos.....	28
1.10 Herramientas para el desarrollo .....	28
1.11 Consideraciones finales.....	30
Capítulo 2: Solución propuesta.....	31
2.1. Descripción del Sistema Propuesto .....	31
2.2. Levantamiento de requisitos de software .....	31
2.2.1. Requisitos funcionales .....	32
2.2.2. Roles de usuario .....	34
2.2.3. Requisitos no funcionales .....	35
2.3. Exploración.....	37
2.3.1. Historias de usuarios.....	37
2.4. Planificación de la entrega.....	41
2.4.1. Plan de iteración .....	41
2.4.2. Plan de estimación.....	42



---

2.4.3. Plan de entrega.....	49
2.5. Arquitectura y diseño .....	53
2.5.1. Tarjetas CRC.....	53
2.5.2. Patrón Arquitectónico .....	55
2.5.3. Patrones de diseño .....	57
2.5.4. Modelo de datos.....	58
2.6. Consideraciones finales.....	59
Capítulo 3: Implementación y pruebas del sistema .....	60
3.1. Fase de implementación.....	60
3.2. Pruebas del Sistema.....	67
3.2.1. Pruebas de caja blanca.....	67
3.2.2. Pruebas de aceptación .....	71
3.2.3. Resumen de los resultados obtenidos .....	74
3.2.4. Cobertura de las pruebas.....	75
3.3. Consideraciones finales.....	76
Conclusiones Generales.....	77
Recomendaciones .....	78
Referencias bibliográficas.....	79
Anexos.....	81
Anexo 1: Historias de usuarios.....	81
Anexo 2: Tarjetas CRC.....	99
Anexo 3: Tareas de ingeniería.....	102
Anexo 4: Pruebas de caja blanca.....	106
Anexo 5: Interfaces principales de UCIBlogs .....	109

## Introducción

---

La Internet se ha convertido en un recurso en el proceso de enseñanza-aprendizaje. La educación en el siglo XXI se encuentra en un proceso de transformación donde tiene gran impacto la introducción de las TICs. El empleo de las TICs en la formación de la enseñanza superior aporta múltiples ventajas en la mejora de la calidad docente, materializadas en aspectos tales como el acceso desde áreas remotas, la flexibilidad en tiempo y espacio para el desarrollo de las actividades docentes o la posibilidad de interactuar con la información por parte de los diferentes agentes que intervienen en dichas actividades (1).

Se adiciona a esta situación la Web 2.0, que se caracteriza por los aspectos colaborativos, el diálogo y la participación en la construcción del conocimiento, potenciando los procesos de trabajo colectivo (2). Investigaciones recientes en el área de la educación incorporan el término de Web Educativa 2.0, como un espacio abierto y común donde el alumno toma parte activa y gestiona su proceso de aprendizaje, mientras el profesor tiene un importante papel de guía, motivador y coordinador, de manera tal que juntos forman una red de aprendizaje (3). La Web 2.0 ofrece la oportunidad de utilizar Internet o la red local de cualquier centro docente como una extensión del aula, convirtiéndola en una herramienta para promover el aprendizaje y aumentar las posibilidades del profesor para dar dinamismo a su tarea docente. Al mismo tiempo implica que la comunidad docente debe estar abierta a estos nuevos sistemas de aprendizaje, formando al profesorado para que puedan hacer un correcto uso de los mismos.

Los *blogs* forman parte de esta nueva filosofía de la Web 2.0, “tienen un gran potencial como herramienta en el ámbito de la enseñanza, ya que se pueden adaptar a cualquier disciplina, nivel educativo y metodología docente” (4). Los blogs tienen grandes aplicaciones en la educación y constituye un desafío, en la actualidad, incluirlos en las prácticas pedagógicas, donde se potencien, enriquezcan y contribuyan a la mejora de los procesos de enseñar y aprender (5). Surgen así los llamados blogs educativos o *edublogs*, forma abreviada como se conoce en internet. Un blog educativo es un blog que se usa con fines docentes o en entornos de aprendizaje, tanto por profesores como por

alumnos (4). Para los profesores puede servir como ventana de información y opinión acerca de la asignatura o elementos de su ámbito de trabajo, o cualquier tema con fines educativos para compartir. Para los alumnos como foro de opinión para mostrar sus progresos en una determinada actividad, asignatura, o conocimientos adquiridos a lo largo de su vida (5).

Cuba a pesar de todas las dificultades económicas no se ha quedado atrás en cuanto a la implementación y desarrollo de las TICs y la Web 2.0 en la educación superior para potenciar el proceso de enseñanza-aprendizaje. Ejemplo de su implementación es la creación de la red Infomed, que constituye un paradigma en cuanto al uso de estas tecnologías en la educación médica superior cubana (6).

La Universidad de las Ciencias Informática (UCI), no está ajena a la incorporación de las TICs y la Web 2.0 en el proceso docente, “cuenta actualmente con varias herramientas educativas, dentro de las que sobresalen la Herramienta de Autor para la creación de los Objetos de Aprendizaje (OA), el Repositorio de Objetos de Aprendizaje y el Entorno Virtual de Aprendizaje” (7). Sin embargo se aprecia la ausencia de un carácter colaborativo entre todos los individuos que integran la comunidad universitaria para el trabajo de tipo investigativo, extracurricular y demostración de aprendizaje. No se propicia un espacio donde, tanto estudiantes como profesores, de forma individual o en colectivo, tengan la posibilidad de publicar:

- ❖ La información académica sobre cualquier tema de interés.
- ❖ Las conclusiones obtenidas mientras construyen sus conocimientos.
- ❖ Las investigaciones relacionadas con una temática en particular.

Por otra parte no se propicia un espacio donde dicha información pueda ser archivada, accedida, comentada y debatida por todas aquellas personas que se sientan motivadas de dar su opinión de lo que se ha publicado.

Teniendo en cuenta la situación problemática surge el siguiente **problema de investigación**: ¿Cómo propiciar un espacio virtual donde estudiantes y profesores de la UCI puedan publicar, archivar, comentar y compartir información académica que favorezca el intercambio del conocimiento?

Se define como el **objeto de estudio** los blogs educativos.

Una vez determinado el objeto de estudio el **campo de acción** se enmarca en la gestión de blogs educativos en la Plataforma UCIBlogs.

Para dar solución a la problemática antes planteada se define como **objetivo general**: Desarrollar una plataforma web para la gestión de blogs educativos que permita a los estudiantes y profesores de la UCI publicar, archivar, comentar y compartir información académica que favorezca el intercambio del conocimiento.

### **Tareas investigativas:**

1. Análisis del estado del arte sobre los blogs, la seguridad y estándares de registro de usuarios en espacios virtuales sociales.
2. Fundamentación de los aspectos teóricos que sustentan la investigación.
3. Modelación del perfil del usuario de la Plataforma UCIBlogs.
4. Análisis de las formas de clasificación de blogs.
5. Selección de las tecnologías informáticas y metodología de desarrollo de software apropiada para el desarrollo de la solución.
6. Generación de los artefactos ingenieriles de acuerdo a la metodología seleccionada.
7. Documentación del proceso de desarrollo de la plataforma UCIBlogs.
8. Diseño e implementación de la plataforma UCIBlogs.
9. Validación de la solución.

Los **métodos teóricos** aplicados en la presente investigación son:

- ❖ **Histórico – lógico**: se empleó para conocer la evolución y el desarrollo de los blogs educativos.
- ❖ **Analítico – sintético**: su aplicación permitió el análisis de la teoría y extracción de los principales conceptos a incluir en el marco teórico.
- ❖ **Inductivo – deductivo**: su utilización facilitó arribar a conclusiones específicas, después de haber analizado un conjunto de bibliografías sobre los blogs.
- ❖ **Análisis documental**: viabilizó el trabajo de búsqueda y extracción de información con la literatura especializada en el tema abordado en la presente investigación.

- ❖ **Modelación:** permitió la creación y el esbozo de los diagramas del proceso de análisis y diseño de la aplicación desarrollada.

El **método empírico** aplicado en la investigación es:

- ❖ **Observación:** apoyó la selección de las principales funcionalidades a tener en cuenta para la gestión de los blogs.

### **Posibles resultados:**

- ❖ La plataforma para la gestión de blogs educativos: UCIBlogs.
- ❖ La documentación y los artefactos de la herramienta desarrollada.

A continuación se describe de manera resumida el contenido que se muestra en cada uno de los 3 capítulos:

### **Capítulo 1: Fundamentación teórica:**

En este capítulo, se describen los fundamentos teóricos que sustentan la investigación, haciendo una exposición detallada de los principales conceptos relacionados con el objeto de estudio y el campo de acción, así como una selección de las tecnologías y herramientas a utilizar en el desarrollo de la aplicación. Además se hace un estudio de las plataformas de gestión de blogs existentes en el mundo.

### **Capítulo 2: Análisis y diseño de la solución propuesta:**

En el capítulo se establece la solución al problema a través de la definición de las historias de usuarios. Muestra la descripción de las características fundamentales que posee el sistema identificando todos los requisitos funcionales y no funcionales de la aplicación. Además se muestra la planificación de las iteraciones y también el diseño de la aplicación.

### **Capítulo 3: Implementación y pruebas del sistema:**

Se tratan los aspectos relacionados con la implementación de la solución propuesta y se valida mediante las pruebas de cajas blancas y las pruebas de aceptación.

## Capítulo 1: Fundamentación teórica

---

En este capítulo se explicará todo lo relacionado con la fundamentación teórica que incluye el estudio de los principales conceptos asociados con la solución del problema. También se analizarán algunas soluciones existentes y se abordarán las herramientas y tecnologías que serán utilizadas para darle solución al problema.

### 1.1 Los Blogs

El término *web-log* lo acuñó Jorn Barger en el 1997, para referirse a un diario personal en línea que su autor o autores actualizan constantemente. Más adelante, las dos palabras *web* y *log*, se comprimieron para formar una sola, *Weblog* y luego se redujo al término de *Blog* con el cual se referencia en las bibliografías actuales. El acto de escribir un *blog* se conoce como *blogging*, por extensión, la persona que lo escribe es un *blogger* y el espacio donde se concentran los *blogs* se conoce como *blogosphere* (5). En esencia un *blog* es un archivo electrónico que contiene los enlaces, comentarios, opiniones y reflexiones de un autor o conjunto de autores, organizados de forma cronológica inversa sobre un tema o un área del conocimiento. Consecuentemente, se puede afirmar que los *blogs* son herramientas ideales para fomentar la comunicación multidireccional y la diseminación de información (8).

Con el fin de conceptualizar el término *blog* dentro de la presente investigación se han analizado un conjunto de definiciones tomadas de distintas fuentes y con enfoques diferentes, dentro de las que se encuentran:

- ❖ **Definición enciclopédica:** “Un *blog* es un sitio web frecuentemente actualizado que consiste en accesos fechados cronológicamente en orden inverso de tal forma que los mensajes más recientes aparecen primero. Generalmente son publicados por individuos y su estilo es personal e informal” (9).
- ❖ **Definición sencilla:** “*Blog* es una bitácora en línea” (10).
- ❖ **Definición técnica:** “Un *blog* es una jerarquía de textos, imágenes, objetos multimedia y datos ordenados cronológicamente que pueden ser vistos a través de un navegador” (11).

A partir del análisis de diferentes blogs internacionales se determinaron las principales características que debe cumplir un blog, las cuales se describen a continuación.

Se determinó que los blogs disponen de dos interfaces importantes:

- ❖ **Área pública del blog:** Muestra los artículos publicados por el autor del blog, estos pueden ser leídos y comentados por cualquier usuario que accede al mismo.
- ❖ **Área de acceso restringido o administración del blog:** En este espacio entra el autor/autores del blog para gestionar su contenido.

El **área pública del blog** posee los siguientes elementos:

- ❖ *Banner* o cabecera: corresponde a la zona superior y es donde se coloca el título del blog, además se pueden incluir imágenes, animaciones, descripción.
- ❖ *Post* o entradas publicadas: contiene los artículos ordenados cronológicamente, situando de primero el más reciente.
- ❖ Barras laterales: contienen las entradas publicadas ordenadas por fecha de creación, búsqueda de entradas, artículos por etiquetas y lista de enlaces.

En la lectura de una entrada o artículo se suelen ofrecer los siguientes elementos:

- ❖ Fecha de publicación del artículo.
- ❖ Título del artículo.
- ❖ Contenido del artículo: puede contener texto, enlaces, imágenes, audios, vídeos.
- ❖ Autor del artículo: se muestra el nombre del autor del artículo. Puede resultar interesante cuando en un blog publican varios usuarios.
- ❖ Comentarios: muestra el número de comentarios realizados a esa entrada. Cuando el usuario hace clic en este enlace accede a los comentarios publicados.

El **área de acceso restringido** posee los siguientes elementos:

- ❖ Área de creación: pestaña de opciones de entradas que permite crear, editar y/o borrar artículos.

- ❖ Área de configuración: para modificar los parámetros generales del blog (zona horaria, título...) y la administración de sus usuarios (ej.: añadir o borrar otros editores).
- ❖ Área de personalización: desde la pestaña Diseño se permite cambiar la apariencia de tu blog.

### 1.2 Los blogs como apoyo al proceso de enseñanza-aprendizaje

La potencialidad principal de los *blogs* es que “[...] permiten desarrollar capacidades de trabajo colaborativo a través de la distribución de funciones en el grupo y del establecimiento de un modelo de tutoría mutua entre sus integrantes [...]” (4). Cabe resaltar además que los *blogs* aportan valor agregado al ejercicio docente, ya que éste se ve enriquecido por las nuevas formas de aprender y de interactuar socialmente de los estudiantes, en tanto, le permite a los docentes procurar una educación que desarrolle habilidades de pensamiento y para un aprendizaje autónomo que posibilite la capacidad de aprender a aprender (12).

En la actualidad existe un conjunto bastante amplio de docentes que se apoyan en los *blogs* como recurso para aprender y enseñar, compartiendo las experiencias con el resto de la comunidad de docentes que lo utilizan, promoviendo el aprendizaje colaborativo y la integración de sus estudiantes y colegas (5).

### 1.3 Los edublogs o blogs educativos

El término *edublog*, que nace de la unión de *education* y *blog*, se considera que en este caso su principal objetivo es apoyar el proceso de enseñanza-aprendizaje en un contexto educativo. Los blogs en la educación tienen como fin entre otros, eliminar las barreras técnicas para que los estudiantes escriban y publiquen en línea, ya que como tienen forma de una bitácora, anima a los estudiantes a conservar una grabación de su propio pensamiento o proceder en un determinado espacio de tiempo (4).

Los primeros profesores que experimentaron con los blogs fueron los ingleses, con el portal británico Schoolblogs.com, que funciona desde 2001, y el grupo *Education Bloggers Network*, con sede en Estados Unidos. Sin embargo, uno de los mayores apoyos a la introducción de los blogs en un entorno académico, fue liderado por la Universidad de Harvard, de la mano de Dave Winer, en la primavera de 2003.



Otra de las razones por la cual el término *edublog* sea tan popular es por su gran impacto en el apoyo a la clase presencial, brindando la posibilidad de transmitir a los alumnos gran cantidad de noticias y de conocimiento que de otra manera, y debido a la limitación de tiempo que las clases implican, sería imposible abordar (13). En otras palabras, los *edublog* permiten al profesor ofrecer información adicional a los alumnos y la posibilidad de proponer actividades complementarias que contribuyen a su auto-superación. Esto último es viable mediante la variedad de recursos que ofrece la escritura hipertextual, el soporte multimedia, los comentarios, proposición de enlaces relacionados, entre otros, aprovechando la interactividad que ofrece el medio. Con ello, se puede ampliar el contexto presencial del aula, extrapolarlo la confrontación directa y el intercambio creado en el ámbito docente hacia la casa, la biblioteca, la oficina o en cualquier otro espacio donde la tecnología permita el acceso a los recursos de información (13).

Las formas de representación de la información que se utilizan en el momento de creación de un artículo, son los textos e imágenes, pero cada vez son más los *blogs* que permiten la publicación de videos dentro de ellos. De este modo se incorporan formas de aprendizaje visual que diversifican el público al cual va dirigida la información, ya que generalmente los videos se suelen utilizar para reforzar los conocimientos tratados en clases o aclarar visualmente un contenido de difícil comprensión. En consecuencia, se favorece una mejor comprensión de los conceptos teóricos, la asimilación del contenido en un ambiente agradable y se demuestra cómo el conocimiento asociado se puede aplicar de forma práctica en situaciones de la vida real (13).

### **1.3.1. Beneficios de los *blogs* educativos**

Los beneficios que ofrecen los *blogs* en el ambiente educativo son diversos por su sencillez, comodidad y facilidad de uso. Son muchos los autores que han abordado sobre estos beneficios, a continuación se muestra una recopilación de los más destacables:

Según Fernández Rodicio (14):

- ❖ Facilitan el auto-aprendizaje en el estudiante.
- ❖ Se rompen las restricciones de espacio y tiempo que impone el aula.
- ❖ La posibilidad de utilizar contenido multimedia, videos, tutoriales y otros recursos, enriquecen y motivan a los estudiantes en su proceso de aprendizaje.

Según Marzal García-Quismondo (15) :

- ❖ Integrar los temas de las asignaturas con el uso de las TICs.
- ❖ La mejora por parte de los estudiantes en la redacción.
- ❖ Facilita la difusión de la investigación, habilitando contextos de interacción social entre diversos agentes y ofrecen enormes posibilidades, principalmente como espacio de participación y actualización docente.
- ❖ Facilita a los alumnos un entorno de aprendizaje global, mediante la interacción de diferentes materiales didácticos y de comunicación personal y tecnológica, permitiendo que profesores y alumnos puedan trabajar de manera continuada.
- ❖ La inclusión de enlaces a otras páginas y herramientas en el blog facilitan el poder resolver dudas, investigar, ampliar información sobre el contenido que se está trabajando de un modo muy eficiente y veloz (16).
- ❖ Permiten crear vínculos entre ellos, según las preferencias de los bloggers y visitantes y construir un sin fin de redes de información y conocimiento, tanto para la educación formal como la no formal (17).

### **1.4 Conclusiones del estudio de los blogs**

Una vez analizados los blogs y dentro de ellos los destinados a la educación, se puede llegar a la conclusión que la principal diferencia entre ambos está en el contenido que se publica en ellos. Por tanto se decide que los blogs que permitirá gestionar la plataforma serán blogs educativos debido a que el contenido que se publicará en los mismos será académico, teniendo en cuenta que el objetivo que se persigue es permitir a los estudiantes y profesores de la UCI publicar, archivar, comentar y compartir información académica que favorezca el intercambio del conocimiento.

### **1.5 Plataforma web**

Una plataforma web es un sistema que permite la ejecución de diversas aplicaciones bajo un mismo entorno, dando a los usuarios la posibilidad de acceder a ellas a través de Internet (18). Esto quiere decir que, al utilizar una plataforma web, el usuario no debe estar en un espacio físico determinado, sino que sólo necesita contar con una conexión a la Web que le permita ingresar a la plataforma en cuestión y hacer uso de sus servicios.

Las plataformas o como también se conoce, herramientas web, proporcionan alojamiento y asignan a las personas usuarias una dirección web y las proveen de una interfaz, a

través de la cual pueden añadir y editar contenido simplificándole en gran medida el proceso (19).

Se puede llegar a la conclusión que una plataforma web puede definirse como: cualquier sistema que permita crear y administrar un sitio web, crear un blog; empleando un sistema web donde podemos subir artículos, crear categorías y administrar los usuarios que se suscriban a un blog determinado.

### 1.5.1. Plataformas web para la creación de blogs

Existen muchas plataformas de creación y mantenimiento de un blog que permiten, muchas de ellas de forma gratuita y sin necesidad de elevados conocimientos técnicos, crear y administrar un blog: coordinar, borrar o reescribir los artículos, moderar los comentarios de los lectores y personalizar su diseño.

Basándose en el tipo de características y servicios que ofrecen estas plataformas, se estudiaron varias de ellas, a continuación se describen algunas y sus principales características:

**Blogger:** Es una herramienta sencilla y gratuita para que los usuarios publiquen en Internet aunque no tengan ningún conocimiento de programación (20). Blogger aloja los blogs en su propio servidor (*Blogspot*). Brinda la opción de invitar a otros usuarios a publicar entradas en un mismo blog.

**Wordpress.com:** Es una plataforma para la creación de blogs que permite abrir una cuenta gratuita y ligado a esa cuenta crea un espacio que es el blog y que tendrá una dirección en Internet del estilo de "minombre.wordpress.com" (21). WordPress permite jerarquizar los contenidos en categorías para una mejor organización y gestión del contenido. Adicionalmente, es posible etiquetar los contenidos con palabras clave para identificar temas comunes que no tienen que estar en la misma categoría.

**Tumblr:** Es una plataforma que permite publicar y compartir cualquier tema de interés educativo como: textos, fotos, citas, enlaces, música, audio y vídeos. Los usuarios pueden "seguir" a otros usuarios registrados, por lo cual Tumblr puede ser considerado una herramienta social. Tumblr tiene un directorio de blogs organizados por categorías y con un ranking sobre su popularidad dentro de la plataforma. El contenido que se publica en Tumblr está abierto a toda la comunidad. Al tratarse de una plataforma social, otros

profesores y profesionales de la educación pueden seguirse y comentar su trabajo desde el mismo Tumblr (22).

Después del estudio de estas plataformas en cuanto a funcionalidades, organización del contenido, bondades y diseño, se identificaron requisitos y elementos comunes que no pueden faltar en el desarrollo de la plataforma web para la creación de blogs educativos a desarrollar.

A partir del análisis de Blogger se identificaron características en cuanto al diseño de las interfaces, así como la idea de un servidor único para que la gestión de los blogs sea sencilla y rápida. También se identificó la idea de que en los blogs puedan publicar más de un usuario, esto propiciará blogs colaborativos. Después del análisis de Wordpress.com se identificaron características como organizar las entradas publicadas por categorías y etiquetar los contenidos con palabras clave para identificar entradas comunes. A través del estudio de Tumblr se seleccionaron las características de seguir a otros usuarios registrados, un *ranking* sobre la popularidad de los blogs dentro de la plataforma y que todo el contenido que se publique quede abierto a toda la comunidad.

### **1.6 Metodología, tecnologías y herramientas para el desarrollo.**

En este epígrafe se hace alusión a las metodologías, herramientas, lenguajes y tecnologías que se utilizarán en el desarrollo de la plataforma. A continuación se precisan algunas características de las mismas.

#### **1.6.1. Metodologías tradicionales o pesadas.**

Teniendo en cuenta la filosofía de desarrollo de las metodologías, aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de Metodologías Tradicionales o Pesadas.

Estas metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no

son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar (23).

Entre las metodologías tradicionales o pesadas podemos nombrar:

**Rational Unified Process (RUP):** RUP es un conjunto de metodologías adaptables al contexto y necesidades de cada organización (24). Constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. RUP es un proceso de desarrollo de software que se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). También es una herramienta determinada por ciclos y fases para el proceso del modelado, estas son: Fase de Inicio, Fase de Elaboración, Fase de Construcción y Fase de Transición. Además brinda la facilidad de utilizar Lenguaje de Unificado (UML, por sus siglas en inglés, Unified Modeling Language), de forma práctica, como apoyo para realizar muchos procesos que existen para modelar o documentar el sistema de una organización.

**Microsoft Solutions Framework (MSF):** Es una metodología desarrollada por *Microsoft Consulting Services* que define un marco de trabajo de referencia para construir e implantar sistemas empresariales distribuidos basados en herramientas y tecnologías de Microsoft para cualquier plataforma (Linux, Citrix, Microsoft, Unix). MSF provee un conjunto de principios, modelos, disciplinas, conceptos y lineamientos para la entrega de tecnología de la información utilizando soluciones Microsoft. MSF no se limita sólo al desarrollo de aplicaciones, también es aplicable a otros proyectos de TI como, proyectos de implementación de redes o infraestructura. También no obliga al desarrollador a utilizar una determinada metodología (Waterfall, Agile), permitiéndole decidir qué utilizar.

**Win-Win Spiral Model:** Esta es una adaptación del modelo de espiral que hace hincapié explícitamente en la participación del cliente en un proceso de negociación en la génesis del desarrollo de productos. Idealmente, el desarrollador simplemente preguntaría al cliente lo que se requiere y el cliente proporcionaría el suficiente detalle para proceder. El modelo "win-win" deriva su nombre del objetivo de estas negociaciones, es decir, de "ganar-ganar". El cliente recibe el producto que satisface la mayoría de sus necesidades,

y el desarrollador trabaja para alcanzar presupuestos y fechas de entrega. Para lograr este objetivo, el modelo define un conjunto de actividades de negociación al principio de cada paso alrededor de la espiral. En vez de una sola actividad de los clientes de comunicación, las actividades se definen como sigue:

- ❖ Identificación de los actores del sistema.
- ❖ Determinación de las partes interesadas de "ganar" condiciones.
- ❖ Las negociaciones de las condiciones de la victoria de las partes interesadas a fin de reconciliarlos en un conjunto de condiciones de ganar-ganar para todos los interesados.

### 1.6.2. Metodologías ágiles o ligeras

Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc. Históricamente, las metodologías tradicionales han intentado abordar la mayor cantidad de situaciones de contexto del proyecto, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños y con requisitos muy cambiantes. Las metodologías ágiles ofrecen una solución casi a medida para una gran cantidad de proyectos que tienen estas características. Una de las cualidades más destacables en una metodología ágil es su sencillez, tanto en su aprendizaje como en su aplicación, reduciéndose así los costos de implantación en un equipo de desarrollo. Esto ha llevado hacia un interés creciente en las metodologías ágiles (23).

Existen diversas metodologías que coinciden en llamarse metodologías ágiles. Y aunque entre ellas comparten muchas características tienen también diferencias significativas. A continuación se presentan algunas de las metodologías ágiles más representativas (25).

**SCRUM.** Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto. Éstas son las

verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

**Dynamic Systems Development Method (DSDM).** Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD unificada. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases.

**Adaptive Software Development (ASD).** Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

**eXtreme Programming (XP):** La programación extrema o *eXtreme Programming* (de ahora en adelante, XP) es una metodología de desarrollo de la ingeniería de software formulada por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de la XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

### 1.6.3. Valoración de la selección

Después de haber hecho un análisis entre las características de las metodologías tradicionales y ágiles, se selecciona una metodología ágil debido a que se corresponde

con el proceso de desarrollo de software a seguir, con el equipo de desarrollo, en cuanto metas a seguir y organización del mismo, con la propuesta de un proceso iterativo y adaptativo para garantizar la realimentación de la información y de los requisitos, así como la validación continua del sistema de información, además con el hecho de poder permitir que sea aplicado a diferentes entornos y características.

Se selecciona XP como metodología de desarrollo, donde algunas características que se tienen en cuenta para su selección son:

- ❖ La aplicación disciplinada de las diferentes prácticas, por ejemplo la programación en pareja (2 integrantes) donde especifica que toda la producción de código debe realizarse en parejas de programadores, lo cual tiene como ventajas que muchos errores sean detectados conforme son introducidos en el código, por consiguiente la tasa de errores del producto final es más baja, los diseños son mejores y el tamaño del código menor, se posibilita la transferencia de conocimientos de programación entre los miembros del equipo, los programadores conversan mejorando así el flujo de información y la dinámica del equipo. Otra práctica utilizada es la definición de estándares de programación con el objetivo de mantener el código legible para los miembros del equipo, facilitando los cambios.
- ❖ Ofrece una solución factible para proyectos con requisitos muy cambiantes, lo que se manifiesta en el desarrollo de la plataforma pues los requisitos pueden variar en dependencia de las necesidades que presente la Universidad. Por otra parte, al contar con diferentes usuarios que presentan diferentes características, pueden surgir nuevas necesidades y con ello nuevas funcionalidades.
- ❖ Es una metodología sencilla tanto en su aprendizaje como en su aplicación, permitiendo con esto al equipo de desarrollo minimizar el tiempo empleado para su familiarización, entendimiento y puesta en práctica de la misma.
- ❖ Está dirigida a equipos pequeños o medianos, donde el entorno físico debe ser un ambiente que permita la comunicación y colaboración entre todos durante todo el tiempo, siendo el clima de trabajo y la colaboración elementos claves del proceso, al estar conformado el equipo por 2 integrantes permite cumplir con dichos elementos.



- ❖ Uso de tecnologías con un ciclo rápido de realimentación y que soporten fácilmente el cambio, característica que poseen las tecnologías empleadas en el desarrollo de la plataforma.

### 1.7 Herramienta para el modelado de la solución

Las herramientas para el modelado son aplicaciones informáticas usadas para aumentar la productividad en el desarrollo de software, reduciendo considerablemente el tiempo. Estas aumentan la calidad del software; mejoran la planificación del proyecto; apoyan la reutilización, portabilidad y estandarización de la documentación del software; facilitan a su vez el uso de las distintas metodologías de desarrollo de software.

**Visual Paradigm for UML Versión 8.0:** “Es una herramienta de modelado que permite realizar diagramas UML, ingeniería tanto directa como inversa (26).” Está concebida para soportar el ciclo completo de desarrollo del software. Permite generar informes en PDF, imágenes y páginas web. Está disponible en múltiples plataformas (Windows y Linux) y posibilita trabajar a varios desarrolladores a la vez sobre un mismo diagrama, permitiendo ver los cambios en tiempo real.

### 1.8 Lenguajes para el desarrollo

Un lenguaje de programación brinda la posibilidad de crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis, con las cuales el programador se puede comunicar con los dispositivos hardware y software existentes. Con el auge de las nuevas tecnologías ha surgido una nueva serie de lenguajes que facilitan el desarrollo de plataformas web.

#### 1.8.1. Lenguaje de modelado

**Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language):** Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (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 compuestos reciclados. Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema,

para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo (27).

Los principales beneficios de UML (28) son:

- ❖ Mejores tiempos totales de desarrollo (de 50 % o más).
- ❖ Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.
- ❖ Establecer conceptos y artefactos ejecutables.
- ❖ Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- ❖ Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- ❖ Mejor soporte a la planeación y al control de proyectos.
- ❖ Alta reutilización y minimización de costos.

### 1.8.2. Lenguajes y tecnologías del lado del cliente

**Lenguaje de Mercado de Hipertexto (HTML) Versión 5:** “Es la actualización de HTML el lenguaje en el que es creada la web. HTML5 también es un término de marketing para agrupar las nuevas tecnologías de desarrollo de aplicaciones web: HTML5, CSS3 y nuevas capacidades de JavaScript” (29). Incorpora nuevas etiquetas como *canvas*, audio, video, *header*, *footer*, *article*, *nav*, *time* entre otras. Permite generar tablas dinámicas que pueden filtrar, ordenar y ocultar contenido en el cliente.

**Hojas de Estilo en Cascada (CSS) Versión 3:** “Es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura”. (30) Ofrece la posibilidad de definir reglas y estilos de representación en diferentes dispositivos ya sean pantallas de equipos de escritorio, portátiles, móviles, impresoras u otros dispositivos capaces de mostrar contenidos web. Cuenta con opciones de sombreado y redondeado y funciones avanzadas de movimiento y transformación.

**jQuery Versión 1.8:** Es una biblioteca de JavaScript rápida y concisa que simplifica el recorrido en documentos HTML, manejo de eventos, animaciones y las interacciones Ajax para el desarrollo web rápido. Está diseñado para cambiar la forma en que escribe el JavaScript (31). Es compatible con los navegadores Mozilla Firefox 10+, Internet Explorer 6+, Safari 5+, Opera y Chrome. Permite agregar efectos a las páginas haciéndolas más interactivas y con una considerable disminución de código. Posee una versión compacta

que recoge toda la implementación en un solo fichero y lo hace más fácil de cargar. Se pueden acoplar librerías como JQuery UI que se encarga de la creación de elementos visuales con poderosas características como los acordeones y calendarios.

**Bootstrap Versión 3:** Es un framework que simplifica el proceso de creación de diseños web combinando CSS y JavaScript. Ofrece una serie de plantillas CSS y ficheros JavaScript que permiten integrar el framework de forma sencilla y potente en proyectos web (32). Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tabletas y móviles, a distintas escalas y resoluciones. Se integra perfectamente con las principales librerías JavaScript como JQuery.

**Ajax (Asynchronous JavaScript And XML):** Es un conjunto de tecnologías independientes que se unen de formas nuevas y sorprendentes. Las tecnologías que forman AJAX (33) son:

- ❖ XHTML y CSS, para crear una presentación basada en estándares.
- ❖ DOM, para la interacción y manipulación dinámica de la presentación.
- ❖ XML, XSLT y JSON, para el intercambio y la manipulación de información.
- ❖ XMLHttpRequest, para el intercambio asíncrono de información.
- ❖ JavaScript, para unir todas las demás tecnologías.

Desarrollar aplicaciones con AJAX requiere un conocimiento avanzado de todas y cada una de las tecnologías anteriores. Permite mejorar completamente la interacción del usuario con la aplicación evitando las recargas constantes de la página ya que el intercambio de información con el servidor se produce en un segundo plano. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

### 1.8.3. Lenguaje del lado del servidor

**PHP (Hypertext Preprocessor):** “Es un lenguaje de código abierto, que se ejecuta del lado del servidor ,orientado a objeto y especialmente adecuado para el desarrollo web el cual puede ser incrustado en código HTML” (34). Está orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos. Tiene la capacidad de conectarse con muchos de los motores de bases de datos que se utilizan en la actualidad y cuenta con manejo de excepciones.

### 1.9 Gestor de bases de datos

**MySql versión 5.1.37:** Es un sistema de gestión de bases de datos relacional, fue creado por la empresa sueca MySQL AB, la cual tiene el *copyright* del código fuente del servidor SQL, así como también de la marca. MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL (35).

Se seleccionó el Gestor de base de datos MySQL debido a que el mismo tiene como objetivo principal la velocidad y la robustez además de soportar gran cantidad de tipos de datos para las columnas. El mismo se destaca por su gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos. Presenta un flexible sistema de contraseñas (*passwords*) y gestión de usuarios, con un buen nivel de seguridad en los datos. Un aspecto muy importante que se tuvo en cuenta en la selección es que todos los proveedores de alojamiento web (*web hosting*) tienen soporte para MySQL lo que posibilita que en un futuro despliegue de la plataforma a desarrollar, el gestor de base de datos no sea un inconveniente, además de que el mismo tiene una gran integración con el lenguaje del lado del servidor seleccionado.

### 1.10 Herramientas para el desarrollo

**PhpStorm versión 7.0:** PhpStorm es un IDE de programación desarrollado por JetBrains. Es uno de los entornos de programación más completos de la actualidad, permite editar código no sólo del lenguaje de programación php como lo indica su nombre (36). Este IDE permite la gestión de proyectos fácilmente, proporciona un fácil autocompletado de código, permite la interpretación y visualización inmediata de código php hasta en 5 de los navegadores web más populares.

Para la selección del entorno de desarrollo se analizaron un conjunto de IDEs de programación, haciendo énfasis en las facilidades que brindan para los lenguajes seleccionados. Los IDEs estudiados fueron NetBeans, PhpDesigner, PhpEdit, Zend Studio y PhpStorm, de los cuales los únicos posibles son NetBeans y PhpStorm debido a que los otros son software privativos. Se seleccionó **PhpStorm** porque proporciona una finalización inteligente de código, comprobación de errores al instante y mezcla de

lenguajes. Este IDE permite la instalación de *plugins* para HTML5, CSS3 y JQuery, los lenguajes de programación del lado del cliente seleccionados anteriormente, lo cual facilita el desarrollo proporcionando un autocompletado excelente, una muy buena visualización y estructuración del código, también admite el trabajo con controles de versiones lo que facilita el trabajo en equipo, además de contar con un historial de versiones propio, el cual almacena todas las modificaciones que se le realizan al código desde que se crea por primera vez con la posibilidad de revertir los cambios en caso de errores.

**Slim versión 2:** Slim es un micro framework PHP que ayuda a escribir con rapidez aplicaciones web simples pero poderosas y permite la construcción sencilla de APIs de servicio (37).

**PhpMyAdmin versión 3.2.0.1:** Es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar bases de datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 62 idiomas. Se encuentra disponible bajo la licencia GPL Versión 2 (38).

**Apache versión 2.2.6:** Es uno de los servidores web más utilizados internacionalmente, su configuración, robustez y estabilidad hacen que cada vez millones de usuarios reiteren su confianza en este programa. Apache es una herramienta gratuita de código fuente abierta. La arquitectura de este servidor es modular ya que cuenta con diversos módulos que le aportan muchas funcionalidades. Presenta versiones tanto para Windows como para Linux, aportando características extendidas como personalización de las variables de entorno y soporte de reparación de errores (39).

**XAMPP versión 1.7.2:** XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. “Es el entorno más popular de desarrollo con PHP” (40).

**VisualSVN Server Manager versión 3.2.2:** Es el software que permite administrar con un visual agradable y de manera sencilla el servidor de *SubVersion*, una herramienta de control de versiones de código abierto que facilita el trabajo en equipo, ya que los códigos

fuentes se encuentran centralizados en este servidor, donde todos los miembros del equipo pueden descargarlo y subir sus cambios para el resto de los desarrolladores y estos obtener nuevamente la última versión (41).

**TortoiseSVN versión 1.8.10:** Es un cliente *SubVersion*, implementado como una extensión al *shell* de Windows. Es software libre liberado bajo la licencia GNU GPL que permite a los desarrolladores interactuar con el servidor *SubVersion*.

### 1.11 Consideraciones finales

Los conceptos asociados al dominio del problema brindaron un mayor entendimiento de los elementos fundamentales de la investigación. El estudio de los sistemas similares permitió la selección de un conjunto de características comunes como la gestión de entradas, comentarios, enlaces y el soporte multimedia, las mismas apoyarán el levantamiento de los requisitos funcionales. Para guiar el proceso de desarrollo se optó por la metodología XP. Los lenguajes del lado del cliente establecidos fueron HTML 5, CSS 3, JavaScript y como lenguaje del lado del servidor PHP 5. El gestor de bases de datos seleccionado fue MySQL, y como herramientas de desarrollo, PhpStorm, Slim, PhpMyAdmin, Apache, XAMPP, VisualSVN Server Manager, TortoiseSVN.

## Capítulo 2: Solución propuesta

---

### 2.1. Descripción del Sistema Propuesto

Se propone como solución desarrollar una plataforma web nombrada “UCIBlogs”, la cual tiene como objetivo principal permitir a cualquier usuario, la creación de blogs educativos, debido a que el contenido que se gestiona es académico. Los blogs son clasificados por categorías educativas predefinidas en la plataforma, estas categorías pueden ser sugeridas por cualquier usuario y son aprobadas o creadas por los administradores de UCIBlogs. La plataforma contará con una interfaz inicial donde los usuarios podrán ver los blogs más visitados, más comentados, más seguidos, además de brindar la opción de buscar blogs por diferentes criterios de búsqueda. Permitirá navegar por todos los blogs creados con la posibilidad de leer las entradas publicadas en dichos blogs, comentarlas, imprimirla, seguir un blog y compartirlo con otro usuario. También se relacionarán las entradas de los blogs mediante palabras claves, y así identificar publicaciones de diferentes autores, con elementos comunes. La plataforma también admitirá crear blogs colaborativos invitando a otros usuarios a ser coautores, los cuales podrán crear nuevas entradas.

### 2.2. Levantamiento de requisitos de software

Antes de establecer una solución al problema siempre se hace necesario llevar a cabo una reunión con el cliente del software o aplicación, para llegar a acuerdos y establecer los requisitos deseados para el proyecto. De ahí es común establecer en los primeros encuentros que se tienen con el cliente, peticiones informales de requisitos de software y diseños que a la larga aportarán mucho a la concepción del software. XP propone que se realice un levantamiento de requisitos en las primeras reuniones, en las que se tomen todos los aspectos relacionados con las necesidades del sistema a desarrollar. En estos encuentros se pueden llevar a cabo prácticas muy efectivas de captura de requisitos como:

- ❖ Entrevistas frecuentes para aclarar dudas que puedan aparecer.
- ❖ Observación de los procesos a informatizar para lograr claridad y precisión en el proceso.

- ❖ Juegos de roles donde el cliente simula ser un usuario que interactúa con el sistema, que es simulado por el desarrollador.

### **2.2.1. Requisitos funcionales**

Funcionalidades de la Plataforma:

RF1- Gestionar autores.

RF1.1- Insertar autores.

RF1.2- Modificar autores.

RF1.3- Eliminar autores.

RF1.4- Mostrar autores.

RF2- Gestionar administradores.

RF2.1- Insertar administradores.

RF2.2- Modificar administradores.

RF2.3- Eliminar administradores.

RF2.4- Mostrar administradores.

RF3- Autenticar usuario.

RF4- Gestionar blogs.

RF4.1- Crear blog.

RF4.2- Modificar blog.

RF4.3- Eliminar blog.

RF4.4- Mostrar blog.

RF5- Gestionar comentarios.

RF5.1- Crear comentario.

RF5.3- Eliminar comentario.

RF5.4- Mostrar comentario.

RF6- Gestionar entradas de un blog.

RF6.1- Crear entrada de un blog.

RF6.2- Modificar entrada de un blog.

RF6.3- Eliminar entrada de un blog.

RF6.4- Mostrar entrada de un blog.

RF7- Gestionar entradas en un blog para el rol coautor.

RF7.1- Crear entradas en un blog para el rol coautor.

RF7.2- Modificar entradas en un blog para el rol coautor.

RF7.3- Mostrar entradas en un blog para el rol coautor.



- RF8- Compartir blogs.
- RF9- Imprimir entrada.
- RF10- Buscar blogs.
- RF11- Buscar entradas.
- RF12- Listar entradas similares al crear una entrada en un blog.
- RF13- Crear servicio de autenticación a través de una Api de servicios.
- RF14- Listar blogs relacionados de acuerdo a las categorías.
- RF15- Listar entradas relacionadas de acuerdo a sus palabras claves.
- RF16- Listar entradas de acuerdo a sus categorías.
- RF17- Listar entradas de un blog.
- RF18- Listar entradas ordenadas por fecha en un blog.
- RF19- Calcular número de visitas de un blog.
- RF20- Seleccionar si le gusta un blog.
- RF21- Crear seguimiento de un blog.
- RF22- Mostrar seguidores de un blog.
- RF23- Enviar notificación al usuario al crearse un blog.
- RF24- Enviar notificación al usuario al recibir un comentario en su blog.
- RF25- Enviar notificación de una nueva entrada a los seguidores de un blog.
- RF26- Enviar notificación a los seguidores de un blog cuando un blog sea eliminado.
- RF27- Invitar a uno o varios usuarios a leer un blog.
- RF28- Invitar a uno o varios usuarios a leer y/o comentar una entrada.
- RF29- Solicitar ser coautor de un blog.
- RF30- Aceptar solicitud de coautor de un blog.
- RF31- Denegar solicitud de coautor de un blog.
- RF32- Enviar notificación de que se ha denegado o aceptado la solicitud de coautor.
- RF33- Invitar a un usuario a ser coautor de un blog creado.
- RF34- Aceptar la invitación de coautor.
- RF35- Eliminar coautor de un blog.
- RF36- Seleccionar entradas similares como referencia al crear una entrada en un blog.
- RF37- Sugerir categoría.
- RF38- Aceptar o denegar categorías.
- RF39- Notificar a un usuario si la categoría sugerida fue aceptada o denegada.
- RF40- Generar reportes.
  - RF40.1- Listar los blogs más visitados.

RF40.2- Listar los blogs más comentados.

RF40.3- Listar los blogs con más seguidores.

RF40.4- Listar los blogs más gustados.

RF40.5- Listar los blogs más recientes.

RF40.6- Mostrar las categorías más usadas en la plataforma.

RF40.7- Mostrar las categorías más seguidas en la plataforma.

RF41- Gestionar documentos relacionados a una entrada.

RF41.1- Insertar documento.

RF41.2- Mostrar documento.

RF41.3- Descargar documento.

RF42- Gestionar categorías para el rol administrador

RF42.1- Insertar categorías para el rol administrador.

RF42.3- Eliminar categorías para el rol administrador.

RF42.4- Mostrar categorías para el rol administrador.

RF43- Gestionar imágenes relacionados a una entrada.

RF43.1- Insertar imágenes.

RF43.2- Mostrar imágenes.

RF43.3- Descargar imágenes.

RF44- Gestionar video relacionados a una entrada.

RF44.1-Insertar video.

RF44.2-Mostrar video.

RF44.3- Descargar video.

### **2.2.2. Roles de usuario**

Los roles de usuario permiten restringir el acceso al sistema, lo que permite garantizar la confiabilidad y disponibilidad de la información. A continuación se definen los roles de usuario de la plataforma UCIBlogs.

- ❖ Autor: Tiene todos los permisos sobre su propio blog.
- ❖ Coautor: Solo tiene acceso a la publicación y edición de entradas siempre que sean las creadas por este, puede invitar a las personas a leer el blog y puede actualizar su perfil.
- ❖ Administrador: Es el encargado de eliminar desde un usuario hasta un comentario en caso de ser necesario, la plataforma cuenta con una administración independiente que le permite al administrador poder realizar cualquier operación de

eliminación, así como el resto de sus funciones. No tiene acceso a la administración de un blog.

### **2.2.3. Requisitos no funcionales**

Los requerimientos no funcionales del sistema son propiedades del producto. Son características que hacen a la aplicación más atractiva, confiable y usable.

#### **Diseño e implementación**

**RnF1-** Lenguajes de programación: Se utilizará el lenguaje de programación PHP para el servidor y en el cliente HTML5, CSS3 y JavaScript.

**RnF2-** Herramienta de desarrollo: Para la implementación del sistema se utilizará PhpStorm versión 7.0. Visual Paradigm como herramienta CASE para el modelado del sistema.

#### **Usabilidad**

El sistema debe permitir un alto nivel de facilidades de uso, basado en el cumplimiento de los siguientes aspectos:

**RnF3-** Permitir una navegación sencilla, esto se logra a partir de una estructura de la información correcta, los usuarios tendrán conocimiento del lugar donde se encuentra, así como todas las opciones que brinda el sistema, a través de una barra de navegación estática que aparece en todas las interfaces de usuarios. Los enlaces y botones serán fáciles de asociar con las operaciones que realizan.

**RnF4-** Permitir la creación de perfiles de usuario usando imágenes y datos de fácil identificación.

#### **Apariencia o interfaz externa**

**RnF6.** La interfaz gráfica uniforme incluyendo pantallas, menús y opciones. La consistencia de la interacción entre usuario y aplicación estará determinada por el diseño de la interfaces de usuario que mantendrán los elementos como menús, paneles de información, en posiciones fijas, además de la mayor uniformidad posible entre cuadros de texto y botones.

**RnF7-** Interfaz con un diseño sin usar animaciones ni componentes dígame *plugins* o *widgets* con vista a acelerar la velocidad de respuesta.

### Software

#### ❖ Para el cliente:

**RnF8-** Utilizar un navegador web (Mozilla Firefox a partir de su versión 25, Google Chrome a partir de su versión 19.0.1084.52 u Opera a partir de su versión 6.00).

**RnF9-** Sistema operativo XP o superior, GNU/Linux o MAC.

#### ❖ Para el servidor:

**RnF10-** Sistema operativo Windows, GNU/Linux o MAC en cualquiera de sus distribuciones.

**RnF11-** Servidor de base de datos MySQL.

**RnF12-** Servidor con módulo PHP disponible (WAMP, XAMPP, u otros).

### Seguridad:

**RnF14-** El sistema deberá garantizar el acceso controlado a la información. Este debe influir sobre cómo se presentan las interfaces para cada usuario dependiendo del nivel de acceso a la información.

### Acceso a la información:

El acceso a la información depende de los roles de usuario definidos por el sistema. Esto garantiza la confiabilidad y disponibilidad de la información.

### Protección de la información:

Se utilizaron funciones definidas por expertos para evitar inyecciones SQL, XSS, así como se implementó en todas las funcionalidades del sistema una correcta validación de todos los datos entrados por el usuario tanto del lado del cliente como del servidor con el objetivo de preservar la integridad de la información.

## 2.3. Exploración

La metodología de desarrollo XP comienza con la fase de exploración. En esta fase, los clientes plantean a grandes rasgos las historias de usuario (HU) que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo (25).

Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita mediante la redacción de sencillas HU. Los programadores estiman los tiempos de desarrollo en base a esta información. Debe quedar claro que las estimaciones realizadas en esta fase son primarias (ya que estarán basadas en datos de alto nivel), y podrían variar cuando se analicen más en detalle en cada iteración.

### 2.3.1. Historias de usuarios

Las HU son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido (25). Las HU deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios. A continuación aparecen descritas las HU de la presente solución. Para realizar la estimación del tiempo de implementación de cada HU se utilizó la estimación por puntos, para lo cual se define un punto igual a un día de trabajo.

A continuación se describen las principales HU del sistema, las restantes se encuentran en el anexo 1 (Ver anexo 1).

Tabla 1: Autenticar usuario

Historias de Usuario	
Número: 9	Usuario: Todos
Nombre Historia: Autenticar usuario	

Prioridad en el negocio: Alta	Nivel de complejidad: Medio
Tiempo de estimación: 1	Iteración asignada: 1
Descripción: El sistema debe permitir al usuario autenticarse, a través de un formulario con los campos, usuario y contraseña.	

Tabla 2: Crear blog

Historias de Usuario	
Número: 10	Usuario: Todos
Nombre Historia: Crear blog	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 1
Descripción: El sistema debe permitir al usuario crearse uno o varios blogs mediante un formulario con los campos título del blog, dirección del blog, descripción del blog y categorías del blog, siempre y cuando el usuario esté autenticado o se haya registrado previamente en el sistema.	

Tabla 3: Eliminar blog

Historias de Usuario	
Número: 12	Usuario: Administrador
Nombre Historia: Eliminar blog	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 2
Descripción: El sistema debe permitir al administrador eliminar los blogs existentes en la plataforma. Para ello se muestran todos los blogs almacenados, dando la opción de seleccionar el que desea eliminar.	

Tabla 4: Mostrar blog

Historias de Usuario	
Número: 13	Usuario: Todos
Nombre Historia: Mostrar blog	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto

Tiempo de estimación: 1	Iteración asignada: 1
Descripción: El sistema debe permitir mostrar el blog al que se quiere acceder.	

Tabla 5: Crear comentario

Historias de Usuario	
Número: 14	Usuario: Todos
Nombre Historia: Crear comentario	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 1
Descripción: El sistema debe permitir al usuario insertar comentarios en los blog que existen en el sistema. Mediante de un formulario con los campos nombre y comentario. El usuario no tiene que estar autenticado en la plataforma.	

Tabla 6: Crear entrada de un blog

Historias de Usuario	
Número: 17	Usuario: Autor y coautor
Nombre Historia: Crear entrada de un blog.	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 1
Descripción: El sistema debe permitir al autor y coautor crear entradas en los blog que existen en el sistema donde ello son autores o coautores, mediante de un formulario con los campos, título de la entrada, palabras claves, categoría, redactar entrada, subir video, documentos e imagen de la entrada.	

Tabla 7: Modificar entrada de un blog

Historias de Usuario	
Número: 18	Usuario: Autor y coautor
Nombre Historia: Modificar entrada de un blog.	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 1
Descripción: El sistema debe permitir al autor y coautor modificar entradas creadas por ellos, mediante de una tabla donde se listan todas las entradas del blog perteneciente al	

autor; o sólo las creadas por el coautor, con un botón para seleccionar la opción de editar.

Tabla 8: Buscar blogs

Historias de Usuario	
Número: 23	Usuario: Todos
Nombre Historia: Buscar blogs	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 2
Descripción: El sistema debe permitir al usuario buscar blogs, por título de blog, autor, categorías, descripción, fecha.	

Tabla 9: Buscar entradas

Historias de Usuario	
Número: 24	Usuario: Todos
Nombre Historia: Buscar entradas	
Prioridad en el negocio: Baja	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 3
Descripción: El sistema debe permitir al usuario buscar entradas, por título, texto y categoría.	

Tabla 10: Solicitar ser coautor de un blog

Historias de Usuario	
Número: 42	Usuario: Todos
Nombre Historia: Solicitar ser coautor de un blog.	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 2
Descripción: Permitir a un usuario solicitar ser coautor de un blog mediante un formulario con los campos nombre y correo.	

Tabla 11: Sugerir categoría



Historias de Usuario	
Número: 50	Usuario: Todos
Nombre Historia: Sugerir categoría	
Prioridad en el negocio: Alta	Nivel de complejidad: Medio
Tiempo de estimación: 1	Iteración asignada: 1
Descripción: El sistema debe permitir al usuario sugerir categoría, mediante un formulario con los campos categoría y correo del usuario.	

Tabla 12: Listar entradas relacionadas de acuerdo a sus palabras claves

Historias de Usuario	
Número: 28	Usuario: Todos
Nombre Historia: Listar entradas relacionadas de acuerdo a sus palabras claves.	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 2
Descripción: El sistema debe permitir listar entradas relacionadas de acuerdo a sus palabras claves en un blog determinado.	

## 2.4. Planificación de la entrega

Durante la fase de planificación se realiza una estimación del esfuerzo que costará implementar todas las HU, se parte del tiempo asignado a cada una en la fase de exploración. Una vez definidas y descritas las HU se procede a organizarlas en las iteraciones correspondientes, teniéndose en cuenta la prioridad especificada por el cliente y del tiempo de desarrollo de cada una. A continuación se explica el proceso que se llevó a cabo en el desarrollo de la aplicación.

### 2.4.1. Plan de iteración

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el

cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción. Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.

A continuación se presentan las iteraciones necesarias para la producción del software:

**Iteración 1:** Se implementa las HU de mayor importancia para el funcionamiento del sistema, de las cuales se encuentran la 1, 9, 10, 11, 13, 14, 16, 17, 18, 19, 20, 21, 27, 29, 30, 31, 32, 33, 34, 35, 50, 53, 54, 55, 56, 57, 58 y 59. De esta iteración pueden derivarse nuevos requerimientos funcionales.

**Iteración 2:** Se implementan las historias de usuarios con prioridad media en el negocio, de las cuales son: 2, 3, 4, 5, 6, 7, 8, 12, 15, 23, 25, 28, 36, 37, 38, 39, 42, 43, 44, 45, 46, 47, 48, 49, 51, 52. Luego se pasará a solucionar los errores y no conformidades que fueron detectadas en la iteración anterior.

**Iteración 3:** Se implementan las historias de usuario con prioridad baja en el negocio, de las cuales son: 22, 24, 26, 20, 41, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74. Al finalizar esta iteración, se constará con una versión estable del producto con todas las funcionalidades implementadas y probadas.

### 2.4.2. Plan de estimación

Tabla 13: Plan de estimación

Historias de Usuario	Estimación por puntos
Insertar autores	0.5
Modificar autores	0.5
Eliminar autores	0.5

Mostrar autores	0.5
Insertar administradores	0.5
Modificar administradores	0.5
Eliminar administradores	0.5
Mostrar administradores	0.5
Autenticar usuario	1
Crear blog	1
Modificar blog	1
Eliminar blog	1
Mostrar blog	1
Crear comentario	1
Eliminar comentario	1
Mostrar comentario	1
Crear entrada de un blog	1
Modificar entrada de un blog	1
Mostrar entradas en un blog	1
Eliminar entradas de un blog	1
Compartir blogs	0.5
Imprimir entrada	0.5
Buscar blogs	1
Buscar entradas	1
Listar entradas similares al crear una entrada en un blog	0.5
Crear servicio de autenticación a través de una Api de servicios	1
Listar blogs relacionados de acuerdo a las categorías	0.5

Listar entradas relacionadas de acuerdo a sus palabras claves	1
Listar entradas de acuerdo a sus categorías	0.5
Listar entradas de un blog	0.5
Listar entradas organizadas por fecha en un blog	0.5
Calcular número de visitas de un blog	0.5
Seleccionar si le gusta un blog	0.5
Crear seguimiento de un blog	0.5
Mostrar seguidores de un blog	0.5
Enviar notificación al usuario al crearse un blog	0.5
Enviar notificación al usuario al recibir un comentario en su blog	0.5
Enviar notificación de una nueva entrada a los seguidores de un blog	0.5
Enviar notificación a los seguidores de un blog cuando un blog sea eliminado	0.5
Invitar a uno o varios usuarios a leer un blog	0.5
Invitar a uno o varios usuarios a leer y/o comentar una entrada	0.5
Solicitar ser coautor de un blog	1
Aceptar solicitud de coautor de un blog	0.5
Denegar solicitud de coautor de un blog	0.5
Enviar notificación de que se ha denegado o aceptado la solicitud de coautor	0.5
Invitar a un usuario a ser coautor de un blog creado	1

Aceptar la invitación de coautor	1
Eliminar coautor de un blog	0.5
Seleccionar entradas similares como referencia al crear una entrada en un blog	1
Sugerir categoría	1
Aceptar o denegar categorías	0.5
Notificar a un usuario si la categoría sugerida fue aceptada o denegada	0.5
Listar los blogs más visitados	0.5
Listar los blogs más comentados	0.5
Listar los blogs con más seguidores	0.5
Listar los blogs más gustados	0.5
Listar los blogs más recientes	0.5
Mostrar las categorías más usadas en la plataforma	0.5
Mostrar las categorías más seguidas en la plataforma	0.5
Crear entradas en un blog para el rol coautor	1
Modificar entradas en un blog para el rol coautor	1
Mostrar entradas en un blog para el rol coautor	1
Insertar documento	0.5
Mostrar documento	0.5
Descargar documento	0.5

Insertar categorías para el rol administrador	0.5
Eliminar categorías para el rol administrador	0.5
Mostrar categorías para el rol administrador	0.5
Insertar imágenes	0.5
Mostrar imágenes	0.5
Descargar imágenes	0.5
Insertar video	0.5
Mostrar video	0.5
Descargar video	0.5
Total	49

**Tabla 14:** Plan de duración de las iteraciones

Iteración	Historias de Usuario	Estimación por puntos
<b>1</b>	Insertar autores	19.5
	Mostrar las categorías más usadas en la plataforma	
	Mostrar las categorías más seguidas en la plataforma	
	Autenticar usuario	
	Crear blog	
	Modificar blog	
	Mostrar blog	
	Crear comentario	
	Mostrar comentarios	
	Crear entrada de un blog	
	Modificar entrada de un blog	

	Eliminar entrada de un blog	
	Mostrar entrada de un blog	
	Compartir blog	
	Listar blogs relacionados de acuerdo a las categorías	
	Listar entradas de acuerdo a sus categorías	
	Listar entradas de un blog	
	Listar entradas organizadas por fecha en un blog	
	Calcular número de visitas de un blog	
	Seleccionar si le gusta un blog	
	Crear seguimiento de un blog	
	Mostrar seguidores de un blog	
	Sugerir categoría	
	Listar los blogs más visitados	
	Listar los blogs más comentados	
	Listar los blogs más seguidos	
	Listar los blogs más gustados	
	Listar los blogs más recientes	
<b>2</b>	Modificar autores	<b>17</b>
	Eliminar autores	
	Mostrar autores	
	Insertar administradores	
	Modificar administradores	
	Eliminar administradores	
	Mostrar administradores	
	Eliminar blog	
	Eliminar comentario	
	Buscar blogs	
	Listar entradas similares al crear una entrada en un	

	blog	
	Listar entradas relacionadas de acuerdo a sus palabras claves	
	Enviar notificación al usuario al crearse un blog	
	Enviar notificación al usuario al recibir un comentario en su blog	
	Enviar notificación de una nueva entrada a los seguidores de un blog	
	Enviar notificación a los seguidores de un blog cuando un blog sea eliminado	
	Solicitar ser coautor de un blog	
	Aceptar solicitud de coautor de un blog	
	Denegar solicitud de coautor de un blog	
	Enviar notificación de que se ha denegado o aceptado la solicitud de coautor	
	Invitar a un usuario a ser coautor de un blog creado	
	Aceptar la invitación de coautor	
	Eliminar coautor de un blog	
	Seleccionar entradas similares como referencia al crear una entrada en un blog	
	Aceptar o denegar categoría sugerida	
	Notificar a un usuario si la categoría sugerida fue aceptada o denegada	
<b>3</b>	Imprimir entrada	<b>12.5</b>
	Buscar entradas	
	Crear servicio de autenticación a través de una Api de servicios	
	Invitar a uno o varios usuarios a leer un blog	
	Invitar a uno o varios usuarios a leer y/o comentar una entrada	
	Crear entradas en un blog para el rol coautor	



Modificar entradas en un blog para el rol coautor	
Mostrar entradas en un blog para el rol coautor	
Insertar documento	
Mostrar documento	
Descargar documento	
Insertar categorías para el rol administrador	
Eliminar categorías para el rol administrador	
Mostrar categorías para el rol administrador	
Insertar imágenes	
Mostrar imágenes	
Descargar imágenes	
Insertar video	
Mostrar video	
Descargar video	
<b>Total</b>	<b>49</b>

### 2.4.3. Plan de entrega

**Tabla 15:** Plan de entrega

Historias de Usuario	Final 1era Iteración (29 de marzo)	Final 2da Iteración (29 de abril)	Final 3era Iteración (10 de mayo)
Insertar autores	V1.0		
Mostrar las categorías más usadas en la plataforma	V1.0		
Mostrar las categorías más seguidas en la plataforma	V1.0		
Autenticar usuario	V1.0		
Crear blog	V1.0		
Modificar blog	V1.0		
Crear comentario	V1.0		

Mostrar comentarios	V1.0		
Crear entrada de un blog	V1.0		
Modificar entrada de un blog	V1.0		
Eliminar entrada de un blog	V1.0		
Mostrar entrada de un blog	V1.0		
Compartir entrada	V1.0		
Listar blogs relacionados de acuerdo a las categorías	V1.0		
Listar entradas de acuerdo a sus categorías	V1.0		
Listar entradas de un blog	V1.0		
Listar entradas organizadas por fecha en un blog	V1.0		
Calcular número de visitas de un blog	V1.0		
Seleccionar si le gusta un blog	V1.0		
Crear seguimiento de un blog	V1.0		
Mostrar seguidores de un blog	V1.0		
Sugerir categoría	V1.0		
Listar los blogs más visitados	V1.0		
Listar los blogs más comentados	V1.0		
Listar los blogs más seguidos	V1.0		
Listar los blogs más gustados	V1.0		
Listar los blogs más recientes	V1.0		
Modificar autores		V1.1	
Eliminar autores		V1.1	
Mostrar autores		V1.1	
Insertar administradores		V1.1	
Modificar administradores		V1.1	
Eliminar administradores		V1.1	

Mostrar administradores		V1.1	
Eliminar blog		V1.1	
Eliminar comentario		V1.1	
Buscar blogs		V1.1	
Listar entradas similares al crear una entrada en un blog		V1.1	
Listar entradas relacionadas de acuerdo a sus palabras claves		V1.1	
Enviar notificación al usuario al crearse un blog		V1.1	
Enviar notificación al usuario al recibir un comentario en su blog		V1.1	
Enviar notificación de una nueva entrada a los seguidores de un blog		V1.1	
Enviar notificación a los seguidores de un blog cuando un blog sea eliminado		V1.1	
Solicitar ser coautor de un blog		V1.1	
Aceptar solicitud de coautor de un blog		V1.1	
Denegar solicitud de coautor de un blog		V1.1	
Enviar notificación de que se ha denegado o aceptado la solicitud de coautor		V1.1	
Invitar a un usuario a ser coautor de un blog creado		V1.1	
Aceptar la invitación de coautor		V1.1	
Eliminar coautor de un blog		V1.1	
Seleccionar entradas similares como referencia al crear una entrada en un blog		V1.1	

Aceptar o denegar categoría sugerida		V1.1	
Notificar a un usuario si la categoría sugerida fue aceptada o denegada		V1.1	
Imprimir entrada			V1.2
Buscar entradas			V1.2
Crear servicio de autenticación a través de una Api de servicios			V1.2
Invitar a uno o varios usuarios a leer un blog			V1.2
Invitar a uno o varios usuarios a leer y/o comentar una entrada			V1.2
Crear entradas en un blog para el rol coautor			V1.2
Modificar entradas en un blog para el rol coautor			V1.2
Mostrar entradas en un blog para el rol coautor			V1.2
Insertar documento			V1.2
Mostrar documento			V1.2
Descargar documento			V1.2
Insertar categorías para el rol administrador			V1.2
Eliminar categorías para el rol administrador			V1.2
Mostrar categorías para el rol administrador			V1.2
Insertar imágenes			V1.2
Mostrar imágenes			V1.2
Descargar imágenes			V1.2
Insertar video			V1.2

Mostrar video			V1.2
Descargar video			V1.2

## 2.5. Arquitectura y diseño

La arquitectura y diseño del *software* son procesos fundamentales ya que a semejanza de los planos de un edificio o construcción, estas indican la estructura, funcionamiento e interacción entre las partes del software. A continuación se presentan las tarjetas CRC, el patrón arquitectónico, los patrones de diseño y el modelo de datos que se emplearon y ayudaron en el desarrollo de la plataforma UCIBlogs.

### 2.5.1. Tarjetas CRC

Para las restantes tarjetas CRC (Ver anexo 2).

Tabla 16: Tarjeta CRC Conexión

Tarjeta CRC	
Clase: Conexión	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Conectar la aplicación con la base datos MySql.</li> </ul>	

Tabla 17: Tarjeta CRC modelo\_blog

Tarjeta CRC	
Clase: modelo_blog	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Crear un blog.</li> <li>• Actualizar un blog.</li> <li>• Eliminar un blog.</li> <li>• Obtener los datos de un blog.</li> <li>• Verificar si un blog existe.</li> <li>• Eliminar un blog.</li> </ul>	Conexión

Tabla 18: Tarjeta CRC modelo\_coautores

Tarjeta CRC	
Clase: modelo_coautores	

Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Invitar a una persona a ser coautor de un blog.</li> <li>• Solicitar ser coautor de un blog.</li> <li>• Aceptar invitación de coautor.</li> <li>• Aprobar solicitud de coautor.</li> <li>• Denegar solicitud de coautor.</li> <li>• Actualizar los datos de un coautor.</li> <li>• Seleccionar los datos de un coautor de un blog.</li> <li>• Verificar invitación de coautor.</li> <li>• Verificar solicitud de coautor.</li> <li>• Verificar si un usuario es coautor.</li> <li>• Verificar el usuario y la contraseña de un coautor.</li> <li>• Seleccionar los coautores de un blog.</li> <li>• Seleccionar las solicitudes de coautores de un blog.</li> <li>• Eliminar un coautor de un blog.</li> </ul>	<p>Conexión</p>

Tabla 19: Tarjeta CRC modelo\_comentarios

Tarjeta CRC	
Clase: modelo_comentarios	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Insertar un comentario.</li> <li>• Seleccionar los comentarios de una entrada.</li> <li>• Eliminar un comentario.</li> </ul>	<p>Conexión</p>

Tabla 20: Tarjeta CRC modelo\_entrada

Tarjeta CRC	
Clase: modelo_entrada	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Crear una entrada en un blog.</li> <li>• Actualizar una entrada de un blog.</li> <li>• Actualizar la cantidad de visitas de una entrada.</li> <li>• Seleccionar los datos de una entrada.</li> <li>• Seleccionar la cantidad de comentarios de una entrada.</li> <li>• Seleccionar las entradas de un blog.</li> <li>• Seleccionar las entradas de un blog de determinada categoría.</li> <li>• Seleccionar entradas para referenciar.</li> <li>• Seleccionar entradas relacionadas de acuerdo a sus</li> </ul>	<p>Conexión</p>

- palabras claves.
- Eliminar una entrada.

### 2.5.2. Patrón Arquitectónico

La arquitectura de un sistema de software es la organización o estructura de los componentes importantes que interactúan en el mismo. La aplicación está diseñada bajo un patrón Modelo-Vista-Controlador (MVC). Este patrón separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes diferentes. En la siguiente imagen se muestra como se aplica lo anteriormente explicado a la plataforma UCIBlogs. El patrón MVC se aprecia en la Figura 1, donde la vista es la página HTML; el modelo es el Sistema de Gestión de Base de Datos y la lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista.

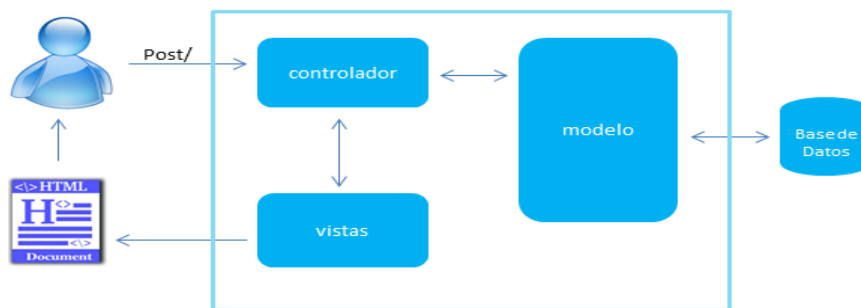


Figura 1: MVC de la plataforma UCIBlogs.

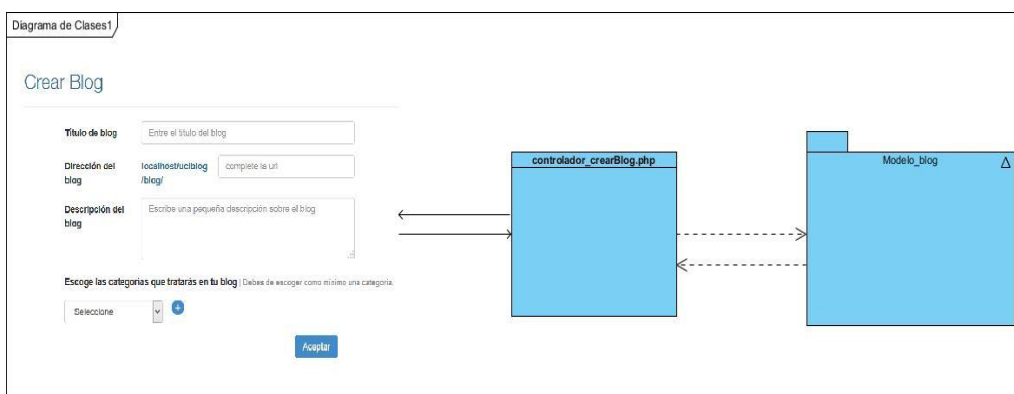


Figura 2: Ejemplo de uso del MVC en la plataforma UCIBlogs.

**Modelo:** Esta capa administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y a instrucciones de cambiar el estado (habitualmente desde el controlador). Representa la información con la que trabaja la aplicación es decir, su lógica de negocio.

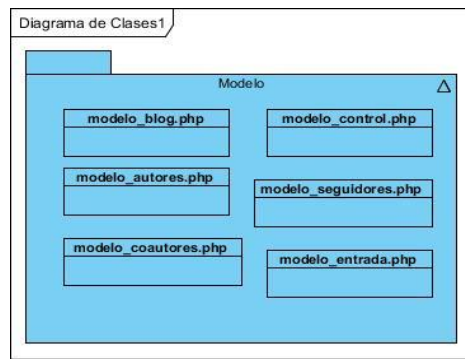


Figura 3: Ejemplo de modelos de la plataforma UCIBlogs.

**Vista:** Esta capa maneja la visualización de la información es decir, presenta el modelo en un formato adecuado para interactuar, que usualmente es la interfaz de usuario.

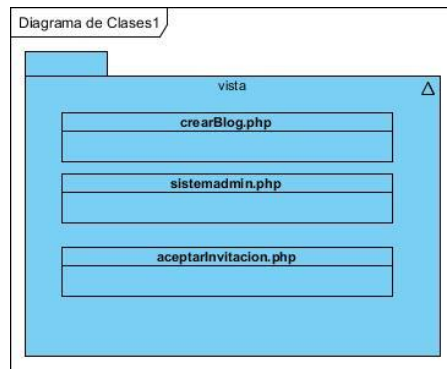


Figura 4: Ejemplo de vistas de la plataforma UCIBlogs.

**Controlador:** Esta capa controla el flujo de datos entre la vista y el modelo es decir, responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y, probablemente en la vista. Tanto la vista como el controlador dependen del modelo el cual no depende de las otras capas.

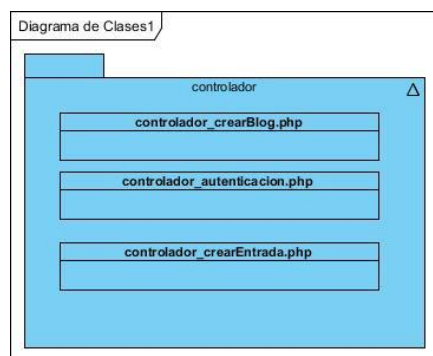




Figura 5: Ejemplo de controladores de la plataforma UCIBlogs.

### 2.5.3. Patrones de diseño

Los **patrones de diseño** son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Para el diseño de la plataforma UCIBlogs se tuvieron en cuenta los 4 patrones GRASP: Experto, Bajo acoplamiento, Alta cohesión y Controlador.

Se hizo uso del patrón **experto** debido a que se diseñaron las clases necesarias cuyas funcionalidades están relacionadas directamente con las entidades que representan y a su vez contienen la información necesaria de las tablas asociadas, trayendo como beneficio la conservación del encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide.

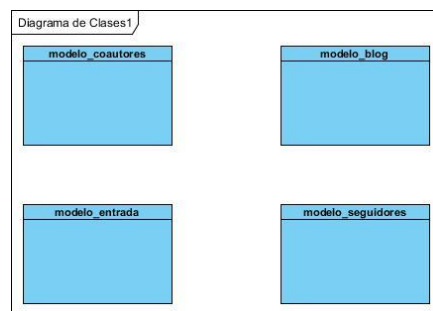


Figura 6: Ejemplo de uso del patrón experto.

Se evidencia el uso del patrón **bajo acoplamiento**. Este patrón propone tener las clases lo menos relacionadas entre sí posible. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases. Se garantiza con su aplicación la reutilización y se disminuye la dependencia entre las clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

Se utiliza el patrón **controlador**, su uso se evidencia en la separación de la lógica del negocio de la capa de presentación, aumentando así la reutilización de código y un mayor control. En la aplicación las peticiones Web son manipuladas por un solo controlador frontal que es el punto de entrada único de toda la aplicación.

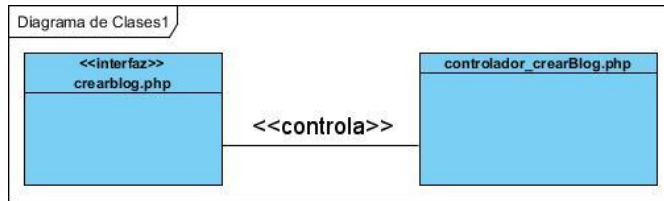


Figura 7: Ejemplo de uso del patrón controlador.

Se evidencia también el uso del patrón **Alta Cohesión** el cual plantea que la información que almacena una clase debe ser coherente y debe estar en lo posible relacionada con la clase.

Por otra parte dentro de los patrones GOF (*Gang of Four*, en español Pandilla de los Cuatro) se destaca la utilización del patrón **Singleton** (instancia única) en la clase conexion.php lo que posibilita que se cree una instancia única de la conexión a la base de datos.

#### 2.5.4. Modelo de datos

El modelo relacional para la gestión de una base de datos es un modelo de datos basado en la lógica de predicado y en la teoría de conjuntos. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar dinámicamente los datos. Su idea fundamental es el uso de «relaciones». Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados «tuplas». La mayoría de las veces se conceptualiza de una manera fácil de imaginar (42).

El siguiente modelo de datos está compuesto por 9 tablas las cuales son descritas a continuación:

Tabla 21: Descripción del modelo de datos

Nombre de la tabla	Descripción
usuarios	Almacena los datos de los autores de un blog.
blog	Almacena todo lo referente a un blog.
entrada	Almacena la información de una entrada.
comentarios	Almacena toda la información referente a los comentarios.

coautores	Almacena los datos de los coautores de los blogs.
seguidores	Almacena los datos de los seguidores de los blogs.
control	Almacena el control en cuanto a cantidad de visitas, comentarios, me gusta, y seguidores de los blogs.
administradores	Almacena los datos de los administradores del sistema.
categoría	Almacena los datos de las categorías educativas por las que son creados los blogs.

La siguiente figura muestra el modelo relacional realizado para almacenar toda la información necesaria.

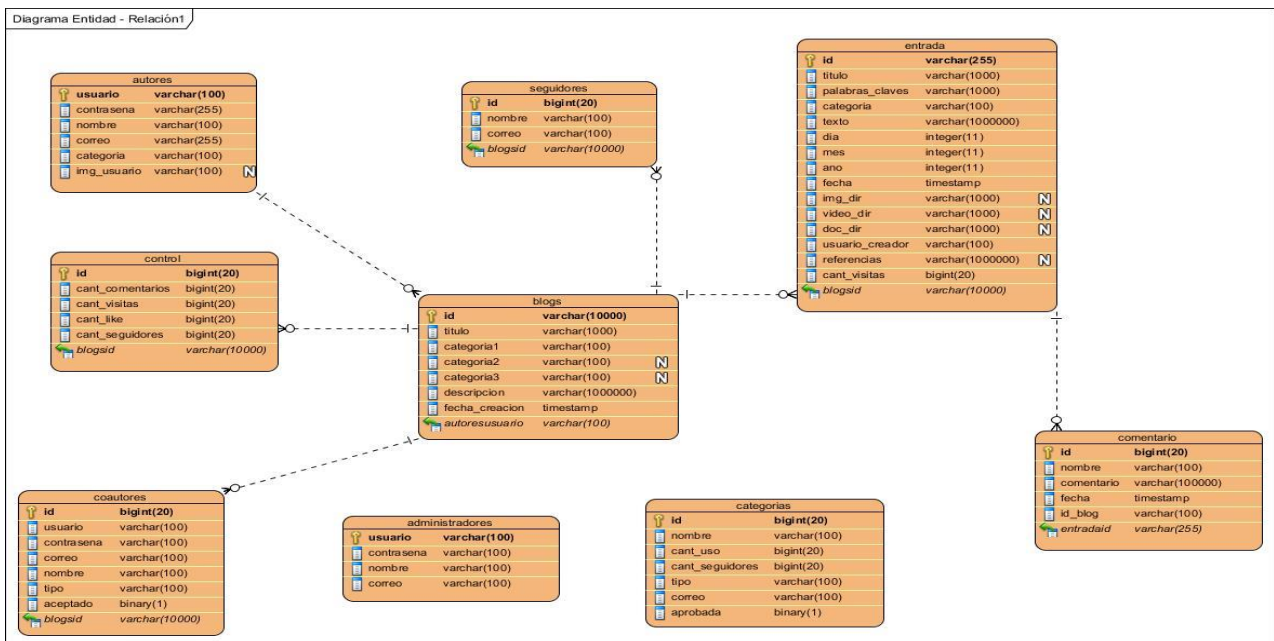


Figura 8: Modelo de dato de la plataforma UCIBlogs.

## 2.6. Consideraciones finales

La arquitectura propuesta y los requisitos del sistema permitieron obtener un diseño sólido de la aplicación. Por otra parte las definiciones de las HU permitieron tener la base para llevar a cabo la implementación de la plataforma.

## Capítulo 3: Implementación y pruebas del sistema

En el presente capítulo se procede a realizar la implementación del sistema. Luego de concluida se prueba la herramienta web. Primeramente se realiza la descripción de las tareas de ingeniería por cada HU. También se aplican las pruebas pertinentes para la validación del correcto funcionamiento del sistema.

### 3.1. Fase de implementación

En esta fase se realiza la implementación de las HU correspondientes a cada iteración, se chequea el plan de iteraciones por si es necesario realizar modificaciones y se crean las tareas de programación para implementar exitosamente cada HU. A continuación se detallan las tareas de ingeniería de las HU más relevantes de la plataforma.

Tabla 22: Tareas de ingeniería

Número	Historias de usuario	Tareas por Historias de Usuario
1	Crear blog	<ol style="list-style-type: none"> <li>1. Implementar la vista con el formulario para crear un blog.</li> <li>2. Implementar el controlador encargado de procesar la información entrada por el usuario para crear un blog.</li> <li>3. Insertar los valores en la entidad 'blogs' de la base de datos de la plataforma.</li> </ol>
2	Crear entrada	<ol style="list-style-type: none"> <li>1. Implementar la vista con el formulario para crear una entrada.</li> <li>2. Implementar el controlador encargado de procesar la información entrada por el usuario para crear una entrada.</li> <li>3. Insertar los valores en la entidad 'entrada' de la base de datos de la plataforma.</li> </ol>
3	Crear comentario	<ol style="list-style-type: none"> <li>1. Implementar el formulario para crear un comentario.</li> <li>2. Implementar el controlador encargado de procesar la información entrada por el usuario para crear un comentario.</li> <li>3. Insertar los valores en la entidad 'comentario' de la base de datos de la plataforma.</li> </ol>
4	Listar entradas relacionadas de acuerdo a sus palabras claves	<ol style="list-style-type: none"> <li>1. Implementar la parte de la vista de la entrada encargada de mostrar la lista de entradas relacionadas de acuerdo a sus palabras claves.</li> <li>2. Implementar en el modelo 'entrada' la función que lista las entradas relacionadas</li> </ol>

## Capítulo 3: Implementación y pruebas del sistema

		de acuerdo a sus palabras claves
5	Buscar blogs	<ol style="list-style-type: none"> <li>1. Implementar la vista con el formulario encargado de permitir al usuario buscar un blog.</li> <li>2. Implementar el controlador encargado de procesar la información entrada por el usuario para buscar un blog.</li> <li>3. Implementar en el modelo 'buscarBlog' la función que se encarga de encontrar el blog que busca el usuario.</li> </ol>
6	Solicitar ser coautor de un blog	<ol style="list-style-type: none"> <li>1. Implementar el formulario encargado de permitir al usuario solicitar ser coautor de un blog.</li> <li>2. Implementar el controlador encargado de procesar la información entrada por el usuario para solicitar ser coautor.</li> <li>3. Insertar los valores en la entidad 'coautores' de la base de datos de la plataforma.</li> </ol>
7	Invitar a un usuario a ser coautor de un blog creado	<ol style="list-style-type: none"> <li>1. Implementar el formulario encargado de permitir al autor de un blog invitar a una persona a ser coautor de su blog.</li> <li>2. Implementar el controlador encargado de procesar la información entrada por el autor para invitar a ser coautor.</li> <li>3. Insertar los valores en la entidad 'coautores' de la base de datos de la plataforma.</li> <li>4. Implementar la función que notificará a la persona invitada.</li> </ol>
8	Sugerir categoría	<ol style="list-style-type: none"> <li>1. Implementar el formulario encargado de permitir al usuario sugerir una categoría.</li> <li>2. Implementar el controlador encargado de procesar la información entrada por el usuario para sugerir una categoría.</li> <li>3. Insertar los valores en la entidad 'categorías' de la base de datos de la plataforma.</li> </ol>
9	Buscar entradas	<ol style="list-style-type: none"> <li>1. Implementar el formulario encargado de permitir al usuario buscar una entrada.</li> <li>2. Implementar el controlador encargado de procesar la información entrada por el usuario para buscar una entrada.</li> <li>3. Implementar en el modelo 'entrada' la función que se encarga de encontrar la entrada que busca el usuario.</li> </ol>

Tabla 23: Tarea 1 de HU 10

Tarea de Ingeniería	
<b>Número de tarea:</b> 1	<b>Número de Historia de Usuario:</b> 10
<b>Nombre de la tarea:</b> Implementar la vista con el formulario para crear un blog.	

## Capítulo 3: Implementación y pruebas del sistema

<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 4/3/2015	<b>Fecha Fin:</b> 4/3/2015
<b>Descripción:</b> Se diseña e implementa la vista (interfaz de usuario) y el formulario con los campos necesarios para la creación de un blog.	

Tabla 24: Tarea 2 de HU 10

Tarea de Ingeniería	
<b>Número de tarea:</b> 2	<b>Número de Historia de Usuario:</b> 10
<b>Nombre de la tarea:</b> Implementar el controlador encargado de procesar la información entrada por el usuario para crear un blog.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha Inicio:</b> 4/3/2015	<b>Fecha Fin:</b> 4/3/2015
<b>Descripción:</b> Se implementa el controlador crearBlog.php el cual se encarga de realizar una correcta validación a los datos para garantizar la seguridad de la información, y una vez pasada la validación se le da paso al modelo para que sea almacenada la información en la base de datos, y una vez concluido este proceso el controlador se encarga de enviar la respuesta a la vista para que el usuario sea informado si el proceso se realizó con éxito o si hubo errores.	

Tabla 25: Tarea 3 de HU 10

Tarea de Ingeniería	
<b>Número de tarea:</b> 3	<b>Número de Historia de Usuario:</b> 10
<b>Nombre de la tarea:</b> Insertar los valores en la entidad 'blogs' de la base de datos de la plataforma.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha Inicio:</b> 4/3/2015	<b>Fecha Fin:</b> 4/3/2015
<b>Descripción:</b> Se implementa en el modelo_blog.php la función que se encarga de almacenar en la entidad 'blogs' de la base de datos de la plataforma la información referente a un blog suministrada por el controlador de este proceso.	

Tabla 26: Tarea 1 de HU 17

Tarea de Ingeniería	
<b>Número de tarea:</b> 1	<b>Número de Historia de Usuario:</b> 17
<b>Nombre de la tarea:</b> Implementar la vista con el formulario para crear una entrada.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 7/3/2015	<b>Fecha Fin:</b> 7/3/2015
<b>Descripción:</b> Se diseña e implementa la vista (interfaz de usuario) y el formulario con los campos necesarios para la creación de una entrada.	

Tabla 27: Tarea 2 de HU 17

Tarea de Ingeniería	
---------------------	--

## Capítulo 3: Implementación y pruebas del sistema

<b>Número de tarea:</b> 2	<b>Número de Historia de Usuario:</b> 17
<b>Nombre de la tarea:</b> Implementar el controlador encargado de procesar la información entrada por el usuario para crear una entrada.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha Inicio:</b> 7/3/2015	<b>Fecha Fin:</b> 7/3/2015
<b>Descripción:</b> Se implementa el controlador crearEntrada.php el cual se encarga de realizar una correcta validación a los datos para garantizar la seguridad de la información, y una vez pasada la validación se le da paso al modelo para que sea almacenada la información en la base de datos, y una vez concluido este proceso el controlador se encarga de enviar la respuesta a la vista para que el usuario sea informado si el proceso se realizó con éxito o si hubo errores.	

Tabla 28: Tarea 3 de HU 17

Tarea de Ingeniería	
<b>Número de tarea:</b> 3	<b>Número de Historia de Usuario:</b> 17
<b>Nombre de la tarea:</b> Insertar los valores en la entidad 'entrada' de la base de datos de la plataforma.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha Inicio:</b> 7/3/2015	<b>Fecha Fin:</b> 7/3/2015
<b>Descripción:</b> Se implementa en el modelo_entrada.php la función que se encarga de almacenar en la entidad 'entrada' de la base de datos de la plataforma la información referente a una entrada suministrada por el controlador de este proceso.	

Tabla 29: Tarea 1 de HU 14

Tarea de Ingeniería	
<b>Número de tarea:</b> 1	<b>Número de Historia de Usuario:</b> 14
<b>Nombre de la tarea:</b> Implementar el formulario para crear un comentario.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 6/3/2015	<b>Fecha Fin:</b> 6/3/2015
<b>Descripción:</b> Se diseña e implementa el formulario con los campos necesarios para la creación de un comentario.	

Tabla 30: Tarea 2 de HU 14

Tarea de Ingeniería	
<b>Número de tarea:</b> 2	<b>Número de Historia de Usuario:</b> 14
<b>Nombre de la tarea:</b> Implementar el controlador encargado de procesar la información entrada por el usuario para crear un comentario.	

## Capítulo 3: Implementación y pruebas del sistema

<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 6/3/2015	<b>Fecha Fin:</b> 6/3/2015
<b>Descripción:</b> Se implementa el controlador comentarios.php el cual se encarga de realizar una correcta validación a los datos para garantizar la seguridad de la información, y una vez pasada la validación se le da paso al modelo para que sea almacenada la información en la base de datos, y una vez concluido este proceso el controlador se encarga de enviar la respuesta a la vista para que el usuario sea informado si el proceso se realizó con éxito o si hubo errores.	

Tabla 31: Tarea 3 de HU 14

Tarea de Ingeniería	
<b>Número de tarea:</b> 3	<b>Número de Historia de Usuario:</b> 14
<b>Nombre de la tarea:</b> Insertar los valores en la entidad 'comentario' de la base de datos de la plataforma.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 6/3/2015	<b>Fecha Fin:</b> 6/3/2015
<b>Descripción:</b> Se implementa en el modelo_comentario.php la función que se encarga de almacenar en la entidad 'comentario' de la base de datos de la plataforma la información referente a una entrada suministrada por el controlador de este proceso.	

Tabla 32: Tarea 1 de HU 28

Tarea de Ingeniería	
<b>Número de tarea:</b> 3	<b>Número de Historia de Usuario:</b> 28
<b>Nombre de la tarea:</b> Implementar la parte de la vista de la entrada encargada de mostrar la lista de entradas relacionadas de acuerdo a sus palabras claves.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 8/4/2015	<b>Fecha Fin:</b> 8/4/2015
<b>Descripción:</b> Se implementa la parte de la vista correspondiente a una entrada que se encarga de mostrar todas las entradas que se relacionan con la que el usuario está observando de acuerdo a sus palabras claves, y se implementa los pasos necesarios para poder mostrar estas entradas en la parte de la vista implementada.	

Tabla 33: Tarea 2 de HU 28

Tarea de Ingeniería	
<b>Número de tarea:</b> 2	<b>Número de Historia de Usuario:</b> 28
<b>Nombre de la tarea:</b> Implementar en el modelo 'entrada' la función que lista las entradas relacionadas de acuerdo a sus palabras claves.	



## Capítulo 3: Implementación y pruebas del sistema

<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.6
<b>Fecha Inicio:</b> 8/4/2015	<b>Fecha Fin:</b> 8/4/2015
<b>Descripción:</b> Se implementa en el modelo_entrada.php la función que se encarga de dividir las palabras claves de la entrada solicitada por la vista y realizar una búsqueda por todas las entradas de la plataforma y seleccionar las que coincidan en sus palabras claves y a su vez brindarle el listado a la vista.	

Tabla 34: Tarea 1 de HU 23

Tarea de Ingeniería	
<b>Número de tarea:</b> 1	<b>Número de Historia de Usuario:</b> 23
<b>Nombre de la tarea:</b> Implementar la vista con el formulario encargado de permitir al usuario buscar un blog.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 6/4/2015	<b>Fecha Fin:</b> 6/4/2015
<b>Descripción:</b> Se diseña e implementa la vista (interfaz de usuario) y el formulario con los campos necesarios para que el usuario pueda buscar el blog que desee.	

Tabla 35: Tarea 2 de HU 23

Tarea de Ingeniería	
<b>Número de tarea:</b> 2	<b>Número de Historia de Usuario:</b> 23
<b>Nombre de la tarea:</b> Implementar el controlador encargado de procesar la información entrada por el usuario para buscar un blog.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 6/4/2015	<b>Fecha Fin:</b> 6/4/2015
<b>Descripción:</b> Se implementa el controlador buscarBlog.php el cual se encarga de realizar una correcta validación a los datos para garantizar la seguridad de la información, y una vez pasada la validación se le da paso al modelo con la información necesaria para que realice la búsqueda, y una vez concluido este proceso el controlador se encarga de enviar a la vista los datos de los blogs obtenidos por el modelo o algún mensaje de error en caso de existir.	

Tabla 36: Tarea 3 de HU 23

Tarea de Ingeniería	
<b>Número de tarea:</b> 3	<b>Número de Historia de Usuario:</b> 23
<b>Nombre de la tarea:</b> Implementar en el modelo 'buscarBlog' la función que se encarga de encontrar el blog que busca el usuario.	

## Capítulo 3: Implementación y pruebas del sistema

<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 4/3/2015	<b>Fecha Fin:</b> 4/3/2015
<b>Descripción:</b> Se implementa en el modelo_buscarBlog.php la función que se encarga de buscar en la base de datos los blogs con el criterio de búsqueda obtenido por el controlador de este proceso.	

Tabla 37: Tarea 1 de HU 24

Tarea de Ingeniería	
<b>Número de tarea:</b> 1	<b>Número de Historia de Usuario:</b> 24
<b>Nombre de la tarea:</b> Implementar el formulario encargado de permitir al usuario buscar una entrada.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 2/5/2015	<b>Fecha Fin:</b> 2/5/2015
<b>Descripción:</b> Se diseña e implementa el formulario con los campos necesarios para que el usuario pueda buscar la entrada que desee.	

Tabla 38: Tarea 2 de HU 24

Tarea de Ingeniería	
<b>Número de tarea:</b> 2	<b>Número de Historia de Usuario:</b> 24
<b>Nombre de la tarea:</b> Implementar el controlador encargado de procesar la información entrada por el usuario para buscar una entrada.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 2/5/2015	<b>Fecha Fin:</b> 2/5/2015
<b>Descripción:</b> Se implementa el controlador buscarEntrada.php el cual se encarga de realizar una correcta validación a los datos para garantizar la seguridad de la información, y una vez pasada la validación se le da paso al modelo con la información necesaria para que realice la búsqueda, y una vez concluido este proceso el controlador se encarga de enviar a la vista los datos de las entradas obtenidas por el modelo o algún mensaje de error en caso de existir.	

Tabla 39: Tarea 3 de HU 24

Tarea de Ingeniería	
<b>Número de tarea:</b> 3	<b>Número de Historia de Usuario:</b> 24
<b>Nombre de la tarea:</b> Implementar en el modelo 'modelo_entrada.php' la función que se encarga de encontrar la entrada que busca el usuario.	

<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 2/5/2015	<b>Fecha Fin:</b> 2/5/2015
<b>Descripción:</b> Se implementa en el modelo_entrada.php la función que se encarga de buscar en la base de datos las entradas con el criterio de búsqueda obtenido por el controlador de este proceso.	

### 3.2. Pruebas del Sistema

Una vez concluida la fase de implementación se pasó a la realización de las pruebas del sistema, las cuales permiten a los desarrolladores y al cliente de la plataforma concluir si la misma tiene la calidad requerida o si cuenta con errores en su funcionamiento. Existen varios tipos de pruebas de software, las cuales permiten determinar si un software tiene calidad o presenta no conformidades. Según la metodología utilizada, se decidieron aplicar dos tipos de pruebas al sistema:

- ❖ **Pruebas de caja blanca:** Están centradas en el código y le permiten al desarrollador comprobar si las funciones o algoritmos implementados en el software tienen la calidad requerida o si tienen algún error. Para su aplicación se analizan los posibles caminos que puede tener una determinada función o algoritmo y luego es probada para comprobar si el resultado esperado es el obtenido.
- ❖ **Pruebas de aceptación:** Estas pruebas las realiza el cliente. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario, las mismas permiten al cliente dar el visto bueno al sistema desarrollado.

#### 3.2.1. Pruebas de caja blanca

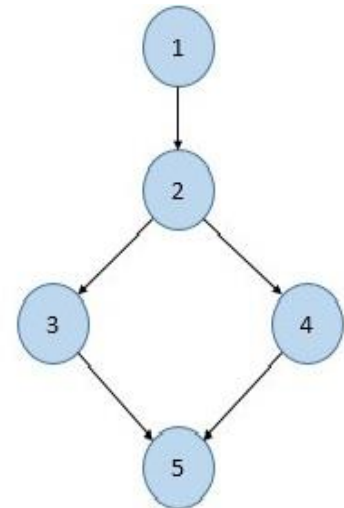
A continuación se muestran algunas de las pruebas de caja blanca realizadas a las funcionalidades de la plataforma:

<p><b>Método: Crear Blog</b></p> <pre>public function CrearBlog() //funcion que permite crear un blog {     \$consulta = "INSERT INTO `blogs`(`id`, `titulo`, `categoria1`,`categoria2`,`categoria3`,     `descripcion`,`autor_user`) VALUES (?,?,?,?,?,?,?);"     \$params = array(\$this-&gt;id, \$this-&gt;titulo, \$this-&gt;categoria1,\$this-&gt;categoria2,     \$this-&gt;categoria3, \$this-&gt;desc,\$this-&gt;autor_user);     \$sql = \$this-&gt;conexion-&gt;consulta_base_de_datos(\$consulta, \$params);     if (!\$sql)     return false;</pre>	<p><b>Grafo resultante:</b></p> <pre> graph TD     1((1)) --&gt; 2((2))     2((2)) --&gt; 3((3))     2((2)) --&gt; 4((4))     3((3)) --&gt; 5((5))     4((4)) --&gt; 5((5))     </pre>
---	--

<pre>else return true; }</pre>	<p>(4)</p> <p>(5)</p>
<p><b>Complejidad Ciclomática:</b></p> <p><math>V(G) = \# \text{ de regiones} = 2</math></p> <p><math>V(G) = A - N + 2 = 5 - 5 + 2 = 2</math></p> <p><math>V(G) = P + 1 = 1 + 1 = 2</math></p>	<p><b>Caminos básicos:</b></p> <p>{1235}, {1245}</p>

<p><b>Método: Crear Entrada</b></p> <pre>public function CrearEntrada() //funcion que permite crear una entrada { \$consulta = INSERT INTO `entrada` (`id`,`titulo`,`palabras_claves`,`categoria`,`texto`,`dia`,`mes`,`ano`,`fecha`,`img_dir`,`video_dir`,`doc_dir`,`usuario_creador`,`blogs_id`)VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?); \$params = array(\$this-&gt;id,\$this-&gt;titulo,\$this-&gt;palabras_claves,\$this-&gt;categoria,\$this-&gt;texto,\$this-&gt;dia,\$this-&gt;mes,\$this-&gt;ano,\$this-&gt;fecha,\$this-&gt;img_dir,\$this-&gt;video_dir,\$this-&gt;doc_dir,\$this-&gt;user_creador,\$this-&gt;blog_id); \$sql = \$this-&gt;conexion-&gt;consulta_base_de_datos(\$consulta,\$params); if (!\$sql) return false; else return true; }</pre>	<p>(1)</p> <p>(1)</p> <p>(1)</p> <p>(2)</p> <p>(3)</p> <p>(4)</p> <p>(5)</p>
<p><b>Complejidad Ciclomática:</b></p> <p><math>V(G) = \# \text{ de regiones} = 2</math></p> <p><math>V(G) = A - N + 2 = 5 - 5 + 2 = 2</math></p> <p><math>V(G) = P + 1 = 1 + 1 = 2</math></p>	<p><b>Caminos básicos:</b></p> <p>{1235}, {1245}</p>

**Grafo resultante:**



<p><b>Método: Seleccionar entradas relacionadas según sus palabras claves</b></p> <pre>public function SelectEntradasRelacionadas(\$palabras,\$id) { \$palabrasarray=explode(',',\$palabras);</pre>	<p>(1)</p>
<p><b>Complejidad Ciclomática:</b></p> <p><math>V(G) = \# \text{ de regiones} = 2</math></p> <p><math>V(G) = A - N + 2 = 5 - 5 + 2 = 2</math></p> <p><math>V(G) = P + 1 = 1 + 1 = 2</math></p>	<p><b>Caminos básicos:</b></p> <p>{1235}, {1245}</p>

**Grafo resultante:**

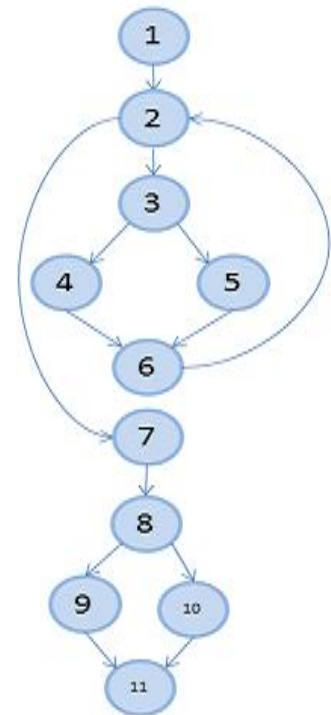
```

$consulta = "SELECT * FROM `entrada` WHERE "; (1)
$params = array(); (1)
for($i=0;$i<count($palabrasarray);$i++){ (2)
    if($i==0){ (3)
        $consulta.="(palabras_claves LIKE ?"; (4)
        $like='%'.$palabrasarray[$i].'%'; (4)
        array_push($params,$like); (4)
    }
    else{ (5)
        $consulta.="OR palabras_claves LIKE ?"; (5)
        $like='%'.$palabrasarray[$i].'%'; (5)
        array_push($params,$like); (5)
    } (6)

} (7)
$consulta.=") AND `id` <>?"; (7)
array_push($params,$id); (7)
$result = $this->conexion->Obtener_Resultados($consulta, $params); (7)
if ($result) { (8)
    return $result; (9)
} else (10)
    return false;

} (11)

```



**Complejidad Ciclomática:**

$V(G) = \# \text{ de regiones} = 4$   
 $V(G) = A - N + 2 = 13 - 11 + 2 = 4$   
 $V(G) = P + 1 = 3 + 1 = 4$

**Caminos básicos:**

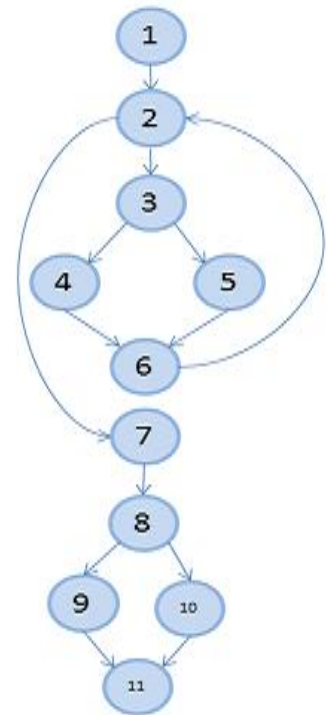
{1-2-3-4-6-2-7-8-9-11}, {1-2-3-5-6-2-7-8-9-11},  
 {1-2-3-4-6-2-7-8-10-11},  
 {1-2-3-5-6-2-7-8-10-11}, {1-2-7-8-10-11}

**Método: Seleccionar entradas para referenciar**

```

public function SelectEntradasParaReferenciar ($palabras) {
    $palabrasarray=explode(',',$palabras);           (1)
    $consulta = "SELECT * FROM `entrada` WHERE ";    (1)
    $params = array();                               (1)
    for($i=0;$i<count($palabrasarray);$i++){        (2)
        if($i==0){                                  (3)
            $consulta.="palabras_claves LIKE ?";    (4)
            $like='%'.$palabrasarray[$i].'%';       (4)
            array_push($params,$like);              (4)
        }
        else{                                        (5)
            $consulta.="OR palabras_claves LIKE ?"; (5)
            $like='%'.$palabrasarray[$i].'%';       (5)
            array_push($params,$like);              (5)
        }
    }
    $consulta.=";";                                  (7)
    $result = $this->conexion->Obtener_Resultados($consulta, $params); (7)
    if ($result) {                                   (8)
        return $result;                             (9)
    } else {                                        (10)
        return false;
    }
}
    
```

**Grafo resultante:**



**Complejidad Ciclomática:**

$V(G) = \# \text{ de regiones} = 4$   
 $V(G) = A - N + 2 = 13 - 11 + 2 = 4$   
 $V(G) = P + 1 = 3 + 1 = 4$

**Caminos básicos:**

- {1-2-3-4-6-2-7-8-9-11}, {1-2-3-5-6-2-7-8-9-11},
- {1-2-3-4-6-2-7-8-10-11},
- {1-2-3-5-6-2-7-8-10-11}, {1-2-7-8-10-11}

### 3.2.2. Pruebas de aceptación

A continuación se muestran algunas de las pruebas de aceptación realizadas a las funcionalidades de la plataforma.

Tabla 40: Prueba de aceptación para la HU "Crear blog"

Caso de Prueba de Aceptación	
<b>Código:</b> H10_P1	<b>Historia de Usuario:</b> 10
<b>Nombre:</b> Crear blog.	
<b>Descripción:</b> El sistema permite al usuario crear un blog.	
<b>Condiciones de ejecución:</b> Estar autenticado.	
<b>Entrada/Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Acceder al sistema a través de la dirección pertinente.</li> <li>2. Autenticarse en el sistema.</li> <li>3. Click en el vínculo 'Crear blog' en el menú de la plataforma.</li> <li>4. Introducir los datos requeridos (título de blog, dirección del blog, Descripción del blog, escoger las categorías que tratarás en tu blog).</li> <li>5. Click en el botón 'Aceptar'.</li> </ol>	
<b>Resultado esperado:</b> El sistema debe crear el blog, notificar al usuario de que se ha creado un blog y mostrar el blog creado.	
<b>Resultado obtenido:</b> Se creó el blog, se notificó al usuario que se creó un blog y mostró el blog creado.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Tabla 41: Prueba de aceptación para la HU "Crear entrada"

Caso de Prueba de Aceptación	
<b>Código:</b> H17_P2	<b>Historia de Usuario:</b> 17
<b>Nombre:</b> Crear entrada.	
<b>Descripción:</b> El sistema permite al autor o al coautor de un blog crear una entrada.	
<b>Condiciones de ejecución:</b> Estar autenticado.	
<b>Entrada/Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Acceder al sistema a través de la dirección pertinente.</li> <li>2. Autenticarse en el sistema.</li> <li>3. Click en el vínculo 'Administrar mis blogs' y entrar a la administración del blog.</li> <li>4. Click en la opción de 'Crear entrada'.</li> <li>5. Introducir los datos requeridos (título, texto, palabras claves, categoría).</li> <li>6. Click en el botón 'Publicar'.</li> </ol>	
<b>Resultado esperado:</b> El sistema debe crear la entrada, notificar a los seguidores del blog que se ha publicado una nueva entrada y mostrar la nueva entrada en las entradas publicadas.	
<b>Resultado obtenido:</b>	

Se creó la entrada, se notificó a los seguidores que se creó una entrada y se mostró la entrada creada en las entradas publicadas.

**Evaluación de la prueba:** Satisfactoria.

Tabla 42: Prueba de aceptación para la HU "Crear comentario"

Caso de Prueba de Aceptación	
<b>Código:</b> H14_P3	<b>Historia de Usuario:</b> 10
<b>Nombre:</b> Crear comentario.	
<b>Descripción:</b> El sistema permite al usuario crear un comentario en una entrada de un blog.	
<b>Condiciones de ejecución:</b> Abrir una entrada de un blog.	
<b>Entrada/Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Acceder a algún blog de la plataforma.</li> <li>2. Click en el botón 'Ver entrada' de cualquier entrada.</li> <li>3. Introducir los datos requeridos (nombre, texto).</li> <li>4. Click en el botón 'Publicar'.</li> </ol>	
<b>Resultado esperado:</b> El sistema debe crear el comentario, notificar que se ha hecho un nuevo comentario al autor del blog y mostrar el nuevo comentario.	
<b>Resultado obtenido:</b> Se creó el comentario, se notificó que se hizo un nuevo comentario al autor del blog y se mostró el comentario creado.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Tabla 43: Prueba de aceptación para la HU "Buscar blog"

Caso de Prueba de Aceptación	
<b>Código:</b> H23_P4	<b>Historia de Usuario:</b> 23
<b>Nombre:</b> Buscar blog.	



<b>Descripción:</b> El sistema permite al usuario buscar blogs.
<b>Condiciones de ejecución:</b> Acceder a la página principal de la plataforma.
<b>Entrada/Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Acceder a la página principal del sistema a través de la dirección pertinente.</li> <li>2. Click en la opción 'Buscar blogs'.</li> <li>3. Introducir el criterio de búsqueda deseado.</li> <li>4. Click en el botón 'Buscar' si se añadió algún filtro de búsqueda o click en el icono buscar si no se añadió ningún filtro.</li> </ol>
<b>Resultado esperado:</b> El sistema debe mostrar el/los vínculo(s) de el/los blog(s) encontrados según el criterio de búsqueda especificado o informar que no hay blogs que coinciden con ese criterio de búsqueda.
<b>Resultado obtenido:</b> El sistema mostró los vínculos a los blogs que coincidieron con el criterio de búsqueda especificado.
<b>Evaluación de la prueba:</b> Satisfactoria.

Tabla 44: Prueba de aceptación para la HU "Sugerir categoría"

Caso de Prueba de Aceptación	
<b>Código:</b> H50_P5	<b>Historia de Usuario:</b> 50
<b>Nombre:</b> Sugerir categoría.	
<b>Descripción:</b> El sistema permite al usuario sugerir una categoría.	
<b>Condiciones de ejecución:</b> Acceder a cualquier interfaz de la plataforma.	
<b>Entrada/Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Acceder a cualquier interfaz de la plataforma.</li> <li>2. Click la opción 'Sugerir categoría' en el menú de la plataforma.</li> <li>3. Introducir los datos requeridos (categoría, correo).</li> <li>4. Click en el botón 'Aceptar'.</li> </ol>	
<b>Resultado esperado:</b> El sistema debe crear la categoría, mostrar al usuario un mensaje que indica que la categoría será aprobada por el administrador del sistema.	
<b>Resultado obtenido:</b> El sistema creó la categoría y le mostró al usuario un mensaje indicándole que la categoría será aprobada por el administrador del sistema.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Tabla 45: Prueba de aceptación para la HU "Buscar entrada"

Caso de Prueba de Aceptación	
<b>Código:</b> H24_P6	<b>Historia de Usuario:</b> 24
<b>Nombre:</b> Buscar entrada.	
<b>Descripción:</b> El sistema permite al usuario buscar una entrada en un blog.	

<b>Condiciones de ejecución:</b> Acceder a un blog de la plataforma.
<b>Entrada/Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Acceder a un blog de la plataforma.</li> <li>2. Introducir el criterio de búsqueda deseado.</li> <li>3. Click en el icono de buscar.</li> </ol>
<b>Resultado esperado:</b> El sistema debe mostrar en el blog solo las entradas que coincidan con el criterio de búsqueda especificado o indicarle al usuario que no hay resultados para esa búsqueda.
<b>Resultado obtenido:</b> El sistema mostró en el blog solo las entradas que realmente coincidieron con el criterio de búsqueda especificado.
<b>Evaluación de la prueba:</b> Satisfactoria.

Tabla 46: Prueba de aceptación para la HU " Solicitar ser coautor de un blog"

Caso de Prueba de Aceptación	
<b>Código:</b> H42_P6	<b>Historia de Usuario:</b> 42
<b>Nombre:</b> Solicitar ser coautor de un blog.	
<b>Descripción:</b> El sistema permite al usuario solicitar ser coautor de un blog.	
<b>Condiciones de ejecución:</b> Acceder a un blog de la plataforma.	
<b>Entrada/Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Acceder a un blog de la plataforma.</li> <li>2. Click en la opción 'Solicitar ser coautor'</li> <li>3. Introducir los campos requeridos.</li> <li>4. Click en el botón 'Aceptar'.</li> </ol>	
<b>Resultado esperado:</b> El sistema debe crear el coautor, y mostrar un mensaje que diga 'Solicitud realizada. Espere a que sea aprobado'	
<b>Resultado obtenido:</b> El sistema creó al coautor y mostró el mensaje 'Solicitud realizada. Espere a que sea aprobado'	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Tras haber identificado los casos de prueba de Caja Blanca y de Aceptación del sistema, se procedió a su ejecución obteniendo los resultados que se muestran en el siguiente epígrafe.

### 3.2.3. Resumen de los resultados obtenidos

Para la ejecución de las pruebas de Aceptación se contó con un total de 74 requisitos funcionales. Con las pruebas que fueron ejecutadas, se cubrió el 100 % de los requisitos funcionales del sistema. De las pruebas ejecutadas el 100 % fueron satisfactorias.

En la ejecución de las pruebas de Caja Blanca se probaron las principales funcionalidades del sistema. En una primera iteración de pruebas se identificaron 50 casos de prueba de los cuales 45 fueron satisfactorios y 5 fueron fallidos dando lugar a 5 no conformidades

las cuales fueron corregidas inmediatamente por los desarrolladores, lo que trajo consigo que en una segunda iteración de pruebas de 50 casos de pruebas identificados todos fueran satisfactorios.

### 3.2.4. Cobertura de las pruebas

Para obtener la cobertura de las pruebas de Aceptación que fueron ejecutadas en el sistema se recogieron los siguientes resultados:

- ❖ Casos de prueba diseñados: 74
- ❖ Casos de prueba aplicados: 74
- ❖ Casos de prueba satisfactorios: 74
- ❖ Cobertura de pruebas (ejecutadas) =  $74/74 = 100\%$
- ❖ Cobertura de pruebas (exitosas) =  $74/74 = 100\%$

En la ejecución de las pruebas de Caja Blanca se obtuvo:

#### 1. Primera iteración (ver gráfico 1):

- ❖ Casos de prueba diseñados: 50
- ❖ Casos de prueba aplicados: 50
- ❖ Cobertura de pruebas (ejecutadas) = 100%
- ❖ Cobertura de pruebas (exitosas) =  $45/50 = 90\%$

#### 2. Segunda iteración (ver gráfico 2):

- ❖ Casos de prueba diseñados: 50
- ❖ Casos de prueba aplicados: 50
- ❖ Cobertura de pruebas (ejecutadas) = 100%
- ❖ Cobertura de pruebas (exitosas) =  $50/50 = 100\%$

Casos de pruebas: Caja blanca

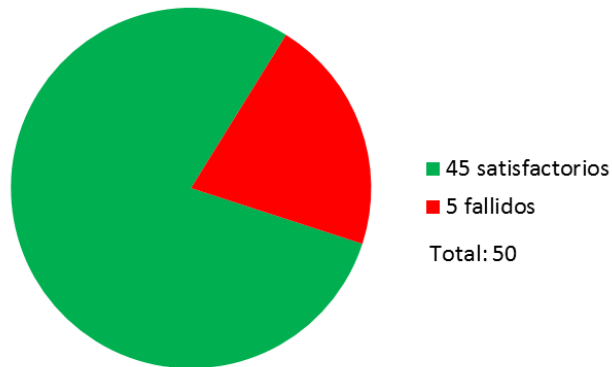


Gráfico 1: Primera iteración de pruebas de caja blanca.

Casos de pruebas: Caja blanca

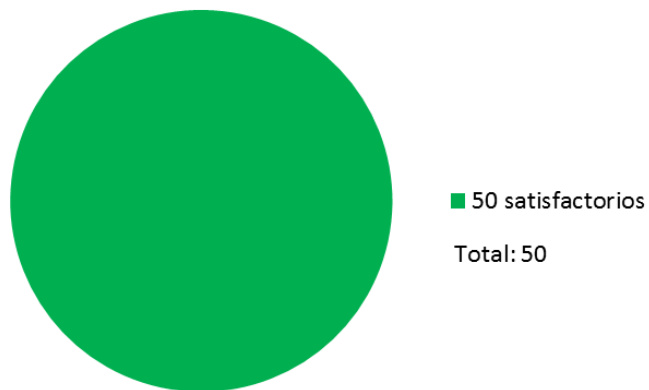


Gráfico 2: Segunda iteración de pruebas de caja blanca.

### 3.3. Consideraciones finales

El proceso de implementación satisfizo las necesidades de la plataforma, lo cual fue comprobado mediante las pruebas realizadas al sistema dando lugar a un correcto funcionamiento de la plataforma. Con el fin de este capítulo se da por terminada la propuesta que trae consigo este trabajo.

## Conclusiones Generales

---

Con la realización del presente trabajo se ha dado cumplimiento a los objetivos planteados en la investigación, obteniendo como resultado principal la plataforma para la gestión de blogs educativos, “UCIBlogs”, la cual permitirá a los estudiantes y profesores de la UCI archivar, publicar, comentar y compartir información académica para favorecer el intercambio del conocimiento. Además se concluyó que los blogs educativos son una herramienta factible para potenciar el trabajo colaborativo y la disseminación de la información en construcción del conocimiento, destacando así su papel esencial en el desarrollo de dicha plataforma.

- ❖ El análisis del estado del arte, la fundamentación de los aspectos teóricos, como parte de la presente investigación, sirvió de ayuda en la selección de las tecnologías informáticas y de la metodología de desarrollo de software, así como en la definición de los requisitos funcionales y no funcionales, lo que garantizó un correcto desarrollo de la plataforma.
- ❖ Con los artefactos generados por la metodología XP, se crearon las bases para la implementación de la plataforma UCIBlogs, permitiendo una correcta documentación del proceso de desarrollo.
- ❖ Se diseñaron e implementaron un conjunto de clases que dan cumplimiento a los requisitos funcionales y no funcionales, aplicando un conjunto de patrones y buenas prácticas.
- ❖ Se demostró, a partir de las diferentes iteraciones de pruebas practicadas al software, que este satisface los requisitos funcionales definidos durante el flujo de trabajo.

## Recomendaciones

---

Una vez terminada la aplicación y analizados los resultados obtenidos de la presente investigación, se mantiene la idea de continuar perfeccionando la solución final, por lo que se recomienda:

- ❖ Crear diseños de blogs diferentes para que el usuario pueda escoger como se verá su blog.
- ❖ Integrar a la plataforma una aplicación externa que permita convertir automáticamente cualquier formato de video al formato webm para que los videos subidos por los autores o coautores de los *blogs* puedan ser reproducidos en las entradas correspondientes.
- ❖ Utilizar la plataforma en la UCI para que todos los profesores y estudiantes puedan transmitir sus conocimientos a toda la comunidad universitaria.
- ❖ Hacer extensible la plataforma para la creación de blogs educativos, hacia otros centros universitarios, instituciones periodísticas, o cualquier otra empresa que le sea apropiada.

## Referencias bibliográficas

1. *Ventajas del uso de las TICs en el proceso de enseñanza-aprendizaje desde la óptica de los docentes universitarios españoles*. **Ferro Soto, Carlos A. , Martínez Senra, Ana Isabel y Otero Neira, María del Carmen**. 29, España : Edutec: Revista electrónica de tecnología educativa, 2009. 1135-9250.
2. *O aspecto relacional das interações na Web 2.0*. **Primo, A.** 1-21, Brasilia : Revista de Biblioteconomia de Brasília, 2009, Vol. 9. 1808-2599.
3. *Web Educativa 2.0*. **de la Torre, Aníbal** . 20, España : Edutec: Revista electrónica de tecnología educativa, 2006. 1135-9250.
4. **TÍSCAR, I. LIGHTNING**. *Usos de los blogs en una pedagogía constructivista*. [En línea] 2005. [Citado el: 20 de 11 de 2014.] <http://www.campusred.net/telos/articulocuaderno.asp?idArticulo=2&rev=65..> 65.
5. *Los Blog como recurso educativo para el proceso de enseñanza* . **Suarez, Z. y Viera, L.** 4, IPLAC : Publicaciones, 2012, Vol. 7. 1993-6850.
6. *El concepto zona de desarrollo próximo y su manifestación en la educación médica superior cubana*. **Lic. González López, Alejandro David, Lic. Rodríguez Matos, Anay de los Ángeles y Lic. Hernández García , Damaris**. La Habana : Scielo, 2011. Disponible en: [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S0864-21412011000400013.0864-2141](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-21412011000400013.0864-2141).
7. *Conceptualización de una red social educativa que integre de forma*. **ARIAS, Y. y ARIAS, Y.** 3, La Habana : Revista Cubana de Ciencias Informáticas, 2011, Vol. 5. Disponible en: <http://rcci.uci.cu/index.php/rcci/article/view/144>. 1994-1536.
8. *El uso de las tic como herramienta para el aprendizaje significativo del inglés*. **Anaya Velasco, Yully , Díaz Rodríguez, Silvia y Martínez Hernández, Jenny**. 27, Colombia : Revistas de la Universidad Cooperativa de Colombia, 2012, Vol. 14. Disponible en: <http://revistas.ucc.edu.co/index.php/ra/article/view/490>.
9. **WALKER, J.** jilltxt. *Weblog Definition for the Routledge Encyclopedia of Narrative Theory*. [En línea] 2006. [Citado el: 5 de 12 de 2014.] [http://jilltxt.net/archives/blog\\_theorising/final\\_version\\_of\\_weblog\\_definition.html](http://jilltxt.net/archives/blog_theorising/final_version_of_weblog_definition.html).
10. *Weblogs en la educación*. **MARTÍNEZ-PENICHE**. 10, s.l. : Revista Digital Universitaria, 2004, Vol. 5. 1067-6079.
11. **WINNER**. *What makes a weblog a weblog?* [En línea] 2003. [Citado el: 14 de 12 de 2014.] <http://blogs.law.harvard.edu/whatmakesaweblogaweblog.html>.
12. **TRUJILLO**. *cognicion. USO EDUCATIVO DE LOS BLOGS*. [En línea] 2010. [Citado el: 15 de 12 de 2014.] [http://www.cognicion.net/index.php?option=com\\_content&task=view&id=384&Itemid=1](http://www.cognicion.net/index.php?option=com_content&task=view&id=384&Itemid=1).
13. *Trabajando con nuestro blog de aula: un estudio de caso en un Centro de Educación Primaria de Valladolid*. **Mantilla Fernández, Daniel**. España : Universidad de Valladolid. Facultad de Educación y Trabajo Social, 2013. Disponible en: <http://uvadoc.uva.es/handle/10324/3898>.
14. *Innovación educativa en la enseñanza presencial universitaria: utilización de blogs*. **Fernández Rodicio, Clara Isabel**. 20, Santiago de Compostela : Universidade de Santiago de Compostela. Servizo de Publicacións e Intercambio Científico, 2010. Disponible en: <http://hdl.handle.net/10347/5006.020>.
15. *Los blogs en el nuevo modelo educativo universitario: posibilidades e iniciativas*. **Marzal García-Quismondo , Miguel Ángel y Butera Fajardo, Maria Jesús**. 19, Madrid : Universidad Carlos III de Madrid, 2007, bid. Disponible en: <http://bid.ub.edu/19marza2.htm>. 1575-5886.
16. *EL BLOG EN EL AULA: Experiencias, Ventajas e Inconvenientes*. **MARTÍNEZ, s.** 4, PaidereX : s.n., 2011, Vol. 2. 2172-6639.
17. *WEBLOGS EN EDUCACIÓN*. **CONTRERAS, F.** 10, s.l. : Digital Universitaria, 2004, Vol. 5. Disponible en: [http://www.revista.unam.mx/vol.5/num10/art65/nov\\_art65.pdf](http://www.revista.unam.mx/vol.5/num10/art65/nov_art65.pdf). 1067-6079.
18. *Plataformas virtuales como recurso para la enseñanza en la universidad: análisis, evaluación y propuesta de integración de moodle con herramientas de la web 2.0*. **Meléndez Tamayo, Carlos Fernando**. 20466, Madrid : Universidad Complutense de Madrid, 2013.
19. **Oromí Farre, Ramon**. *Herramientas para la creación y gestión de blogs. Xarxa Punt TIC*. [En línea] 2013. [Citado el: 20 de 3 de 2015.] [http://moodle.punttic.cat/file.php/198/modul\\_4/modulo4.pdf](http://moodle.punttic.cat/file.php/198/modul_4/modulo4.pdf). 198.
20. **Admin**. *Los 'weblogs' o bitácoras permiten a usuarios sin experiencia publicar contenidos en Internet*. **EROSKI CONSUMER**. [En línea] 2014. [Citado el: 20 de 3 de 2015.] <http://revista.consumer.es/web/es/20020601/internet/.20020601>.

21. **LÓPEZ, BERTO**. ¿Qué diferencias hay entre WordPress.com, Blogger y WordPress.org y cuál es para mi? *wordpress*. [En línea] 2014. [Citado el: 20 de 3 de 2015.] <https://manueldiazvassolo.wordpress.com/2014/10/16/que-diferencias-hay-entre-wordpress-com-blogger-y-wordpress-org-y-cual-es-para-mi/>.
22. **Viñas, Meritxell** . Tumblr: Creando un blog de aula interactivo en segundos. *TotemGuard*. [En línea] 2011. [Citado el: 20 de 3 de 2015.] <http://www.totemguard.com/aulatotem/2011/12/tumblr-creando-un-blog-de-aula-interactivo-en-segundos/>.
23. **Condori, Ticona** . tallerinf281. *Metodologías tradicionales, ágiles, para juegos y para aplicaciones móviles*. [En línea] 2012. [Citado el: 22 de 2 de 2015.] <http://tallerinf281.wikispaces.com/file/view/METODOLOG%C3%8DAS+TRADICIONALES.pdf>.
24. **autores, Colectivo de**. EcuRed. *Proceso Unificado de Desarrollo (RUP)*. [En línea] 2012. [Citado el: 23 de 2 de 2015.] [http://www.ecured.cu/index.php/Proceso\\_Unificado\\_de\\_Desarrollo](http://www.ecured.cu/index.php/Proceso_Unificado_de_Desarrollo).
25. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. **Letelier, Patricio y Penadés, Carmen** . 26, Buenos Aires - Argentina : Ciencia y Técnica Administrativa, 2006, Vol. 05. Disponible en: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>. 1666-1680.
26. Programación en castellano. *Programación en castellano*. [En línea] 2008. [Citado el: 20 de 1 de 2015.] <http://www.programacion.net/php>.
27. **Admin**. Lenguaje unificado de modelado. *Wikipedia*. [En línea] 2015. [Citado el: 20 de 3 de 2015.] [http://es.wikipedia.org/wiki/Lenguaje\\_unificado\\_de\\_modelado](http://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado).
28. **Carlos**. UNAM. *¿Qué es UML?* [En línea] 2010. [Citado el: 5 de 4 de 2015.] <http://profesores.fi-b.unam.mx/carlos/aydoo/uml.html>.
29. **Gutiérrez, Javier J**. [En línea] 2010. [Citado el: 20 de 1 de 2015.] [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf).
30. **Torrealday, Gustavo F**. Torrealday Infomática. *Torrealday Infomática*. [En línea] 2010. [Citado el: 21 de 1 de 2015.] <http://www.torrealday.com.ar/articulos/articulo006.htm>.
31. **españa, Ministerio de Educación**. ITE. *ITE*. [En línea] 2010. [Citado el: 23 de 1 de 2015.] <http://ares.cnice.mec.es/>.
32. **Admin, Somos Libres**. Somos Libres org. *Somos Libres org*. [En línea] 2010. [Citado el: 23 de 1 de 2015.] <http://www.somoslibres.org/modules.php?name=News&file=article&sid=3468>.
33. **Symfony**. Symfony.es. *Symfony.es*. [En línea] 2010. [Citado el: 21 de 1 de 2015.] <http://www.symfony.es/que-es-symfony>.
34. **Pérez Mata, Manel**. TecnoRetales. *TecnoRetales*. [En línea] 2009. [Citado el: 22 de 1 de 2015.] <http://www.tecnoretas.com/programacion/que-es-doctrine-orm>.
35. **Toledo Alma, Enríquez, y otros, y otros**. ¿Qué es MySQL? *Grid Morelos*. [En línea] 2013. [Citado el: 20 de 3 de 2015.] <http://www.gridmorelos.uaem.mx/~mcruz/cursos/miic/MySQL.pdf>.
36. **codigo, Editores de**. PhpStorm. *Editores de codigo*. [En línea] 2014. [Citado el: 25 de 4 de 2015.] <http://www.editoresdecodigo.com/2014/06/descargar-phpstorm-full-ide-para-php-y-mas.html>.
37. **Monagas Santana , Jose Juan**. Slim Framework. *Scoop*. [En línea] 2014. [Citado el: 20 de 4 de 2015.] <http://www.scoop.it/t/agile-programming-by-jose-juan-monagas-santana/p/4014208664/2014/01/15/slim-framework-webcodeonline>.
38. **Llamaret Heredia, Maikel**. phpMyAdmin 4.2.10, última versión del administrador de bases de datos MySQL. *Joven Club*. [En línea] 2014. [Citado el: 20 de 4 de 2015.] <http://gutl.jovenclub.cu/phpmyadmin-4-2-10-ultima-version-del-administrador-de-bases-de-datos-mysql/>.
39. **Kabir, Mohammed**. Intercambios virtuales. *La Bliblia Server Apache 2*. [En línea] 2003. [Citado el: 24 de 2 de 2015.] <http://www.intercambiosvirtuales.org/libros-manuales/la-biblia-del-servidor-apache-2-anaya-multimedia-mohammed-j-kabir>.
40. **Admin**. Apache Friends. *¿Qué es XAMPP?* [En línea] 2015. [Citado el: 10 de 3 de 2015.] <https://www.apachefriends.org/es/index.html>.
41. **Equipo de desarrollo** . Visual SVN. *Visual SVN*. [En línea] 2014. [Citado el: 20 de 4 de 2015.] <https://www.visualsvn.com/server/>.
42. **Admin**. wikipedia. *Modelo relacional*. [En línea] 2015. [Citado el: 20 de 4 de 2015.] [http://es.wikipedia.org/wiki/Modelo\\_relacional](http://es.wikipedia.org/wiki/Modelo_relacional).
43. —. Blogs en el sector educativo. *Greenti México*. [En línea] 5 de 2015. [Citado el: 23 de 2 de 2015.] <http://www.greentimexico.com/2015/05/blogs-en-el-sector-educativo/>.



## Anexos

### Anexo 1: Historias de usuarios

Tabla 47: Insertar administradores

Historias de Usuario	
Número: 5	Usuario: Administrador
Nombre Historia: Insertar administradores	
Prioridad en el negocio: Media	Nivel de complejidad: Medio
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: El sistema debe permitir insertar administradores, mediante de un formulario con los campos usuario, contraseña, nombre y correo.	

Tabla 48: Modificar administradores

Historias de Usuario	
Número: 6	Usuario: Administrador
Nombre Historia: Modificar administradores	
Prioridad en el negocio: Media	Nivel de complejidad: Medio
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: El sistema debe permitir a un administrador actualizar sus datos, mediante de un formulario con los campos contraseña, nombre y correo.	

Tabla 49: Eliminar administradores

Historias de Usuario	
----------------------	--

Número: 7	Usuario: Administrador
Nombre Historia: Eliminar administradores	
Prioridad en el negocio: Media	Nivel de complejidad: Medio
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: El sistema debe permitir eliminar administradores, mediante un listado de los administradores existente y un botón para seleccionar la opción eliminar.	

Tabla 50: Mostrar administradores

Historias de Usuario	
Número: 8	Usuario: Todos
Nombre Historia: Mostrar administradores	
Prioridad en el negocio: Media	Nivel de complejidad: Medio
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir mostrar los administradores, mediante un listado de los administradores existente.	

Tabla 51: Enviar notificación al usuario al crearse un blog.

Historias de Usuario	
Número: 36	Usuario: Todos
Nombre Historia: Enviar notificación al usuario al crearse un blog	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 2

Descripción: El sistema debe permitir notificar al usuario que se cree un blog vía correo electrónico.

Tabla 52: Enviar notificación al usuario al recibir un comentario en su blog.

Historias de Usuario	
Número: 37	Usuario: Autor
Nombre Historia: Enviar notificación al usuario al recibir un comentario en su blog	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: El sistema debe permitir notificar al autor que se ha recibido un comentario en su blog vía correo electrónico.	

Tabla 53: Enviar notificación de una nueva entrada a los seguidores de un blog.

Historias de Usuario	
Número: 38	Usuario: Todos
Nombre Historia: Enviar notificación de una nueva entrada a los seguidores de un blog.	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: El sistema debe permitir notificar los usuarios seguidores de un blog que se creó una nueva entrada vía correo electrónico.	

Tabla 54: Enviar notificación a los seguidores de un blog cuando un blog sea eliminado.

Historias de Usuario	
Número: 39	Usuario: Todos

Nombre Historia: Enviar notificación a los seguidores de un blog cuando un blog sea eliminado.	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: El sistema debe permitir notificar los usuarios seguidores de un blog que el blog ha sido eliminado vía correo electrónico.	

Tabla 55: Invitar a uno o varios usuarios a leer y/o comentar una entrada.

Historias de Usuario	
Número: 41	Usuario: Autor
Nombre Historia: Invitar a uno o varios usuarios a leer y/o comentar una entrada.	
Prioridad en el negocio: Baja	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 3
Descripción: El sistema debe permitir invitar a uno o varios usuarios a leer y/o comentar una entrada.	

Tabla 56: Compartir blog

Historias de Usuario	
Número: 21	Usuario: Todos
Nombre Historia: Compartir blog	
Prioridad en el negocio: Alta	Nivel de complejidad: Medio
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir al usuario compartir blog, mediante de un formulario	

con el campo correo a quien quieres compartir el blog.

Tabla 57: Imprimir entrada

Historias de Usuario	
Número: 22	Usuario: Todos
Nombre Historia: Imprimir entrada	
Prioridad en el negocio: Baja	Nivel de complejidad: Medio
Tiempo de estimación: 0.5	Iteración asignada: 3
Descripción: El sistema debe permitir al usuario imprimir entradas. Seleccionando el botón imprimir.	

Tabla 58: Listar entradas similares al crear una entrada en un blog.

Historias de Usuario	
Número: 25	Usuario: Autor y coautor
Nombre Historia: Listar entradas similares al crear una entrada en un blog.	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: El sistema debe permitir listar entradas similares al crear entrada en un blog.	

Tabla 59: Crear servicio de autenticación a través de una Api de servicio

Historias de Usuario	
Número: 26	Usuario: Todos
Nombre Historia: Crear servicio de autenticación a través de una Api de servicio.	

Prioridad en el negocio: Baja	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 3
Descripción: El sistema debe permitir servicio de autenticación a cualquier sistema externo.	

Tabla 60: Listar blogs relacionados de acuerdo a las categorías.

Historias de Usuario	
Número: 27	Usuario: Todos
Nombre Historia: Listar blogs relacionados de acuerdo a las categorías.	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir listar blogs relacionados de acuerdo a las categorías.	

Tabla 61: Listar entradas de acuerdo a sus categorías.

Historias de Usuario	
Número: 29	Usuario: Todos
Nombre Historia: Listar entradas de acuerdo a sus categorías.	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir listar entradas de acuerdo a sus categorías en el blog correspondiente.	

Tabla 62: Listar entradas de un blog.

Historias de Usuario	
Número: 30	Usuario: Todos
Nombre Historia: Listar entradas de un blog.	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir listar entradas de un blog determinado.	

Tabla 63: Listar entradas ordenadas por fecha en un blog.

Historias de Usuario	
Número: 31	Usuario: Todos
Nombre Historia: Listar entradas ordenadas por fecha en un blog.	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir listar entradas ordenadas por fecha en un blog determinado.	

Tabla 64: Calcular número de visitas de un blog.

Historias de Usuario	
Número: 32	Usuario: Todos
Nombre Historia: Calcular número de visitas de un blog.	
Prioridad en el negocio: Alta	Nivel de complejidad: Medio

Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir calcular número de visitas de un blog determinado.	

Tabla 65: Seleccionar si le gusta un blog.

Historias de Usuario	
Número: 33	Usuario: Todos
Nombre Historia: Seleccionar si le gusta un blog.	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir al usuario seleccionar si le gusta un blog, haciendo clic en un botón.	

Tabla 66: Crear seguimiento de un blog.

Historias de Usuario	
Número: 34	Usuario: Todos
Nombre Historia: Crear seguimiento de un blog.	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir al usuario ser seguidor de un blog, mediante de un formulario con los campos nombre y correo del usuario que quiere seguir al blog.	

Tabla 67: Mostrar seguidores de un blog

Historias de Usuario
----------------------



Número: 35	Usuario: Todos
Nombre Historia: Mostrar seguidores de un blog	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe mostrar en un blog los seguidores de este.	

Tabla 68: Aceptar solicitud de coautor de un blog.

Historias de Usuario	
Número: 43	Usuario: Autor
Nombre Historia: Aceptar solicitud de coautor de un blog.	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: Permitir al autor aceptar la solicitud de coautor.	

Tabla 69: Denegar solicitud de coautor de un blog.

Historias de Usuario	
Número: 44	Usuario: Autor
Nombre Historia: Denegar solicitud de coautor de un blog.	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: Permitir al autor denegar la solicitud de coautor.	

Tabla 70: Enviar notificación de que se ha denegado o aceptado la solicitud de coautor.

Historias de Usuario	
Número: 45	Usuario: Todos
Nombre Historia: Enviar notificación de que se ha denegado o aceptado la solicitud de coautor.	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: El sistema debe permitir notificar que se ha denegado o aceptado la solicitud de coautor.	

Tabla 71: Aceptar o denegar categoría sugerida

Historias de Usuario	
Número: 51	Usuario: Administrador
Nombre Historia: Aceptar o denegar categoría sugerida	
Prioridad en el negocio: Media	Nivel de complejidad: Medio
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: El sistema debe permitir al administrador aceptar o denegar categorías sugeridas, mediante una lista de categorías sugeridas con las opciones de aceptar o denegar.	

Tabla 72: Notificar a un usuario si la categoría sugerida fue aceptada o denegada.

Historias de Usuario	
Número: 52	Usuario: Todos

Nombre Historia: Notificar a un usuario si la categoría sugerida fue aceptada o denegada.	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: El sistema debe notificar si la categoría sugerida por el usuario fue aceptada o denegada vía correo electrónico.	

Tabla 73: Listar los blogs más visitados

Historias de Usuario	
Número: 53	Usuario: Todos
Nombre Historia: Listar los blogs más visitados	
Prioridad en el negocio: Alta	Nivel de complejidad: Medio
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir listar blogs más visitados del sistema.	

Tabla 74: Listar los blogs más comentados

Historias de Usuario	
Número: 54	Usuario: Todos
Nombre Historia: Listar los blogs más comentados.	
Prioridad en el negocio: Alta	Nivel de complejidad: Medio
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir listar los blogs más comentados del sistema.	

Tabla 75: Listar los blogs más seguidos

Historias de Usuario	
Número: 55	Usuario: Todos
Nombre Historia: Listar los blogs más seguidos	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir listar los blogs más seguidos del sistema.	

Tabla 76: Listar los blogs más gustados

Historias de Usuario	
Número: 56	Usuario: Todos
Nombre Historia: Listar los blogs más gustados	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir listar los blogs más gustados del sistema.	

Tabla 77: Listar los blogs más recientes

Historias de Usuario	
Número: 57	Usuario: Todos
Nombre Historia: Listar los blogs más recientes	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto

Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir listar los blogs más recientes del sistema.	

Tabla 78: Mostrar las categorías más usadas en la plataforma.

Historias de Usuario	
Número: 58	Usuario: Todos
Nombre Historia: Mostrar las categorías más seguidas en la plataforma.	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir mostrar las categorías más usadas en la plataforma.	

Tabla 79: Mostrar las categorías más seguidas en la plataforma.

Historias de Usuario	
Número: 59	Usuario: Todos
Nombre Historia: Mostrar las categorías más seguidas en la plataforma.	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir mostrar las categorías más seguidas en la plataforma.	

Tabla 80: Insertar autores

Historias de Usuario	
Número: 1	Usuario: todos

Nombre Historia: Insertar autores	
Prioridad en el negocio: Alta	Nivel de complejidad: Medio
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: Permitir al usuario registrarse en el sistema para poder crearse uno o varios blogs, mediante un formulario que contenga usuario y contraseña, nombre, correo, categoría e imagen de perfil.	

Tabla 81: Modificar autores

Historias de Usuario	
Número: 2	Usuario: Administrador
Nombre Historia: Modificar autores	
Prioridad en el negocio: Media	Nivel de complejidad: Medio
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: Permitir al usuario actualizar los datos de perfil de usuario en el sistema, mediante un formulario que contenga los datos actuales del usuario para que elija los datos que quiere actualizar.	

Tabla 82: Eliminar autores

Historias de Usuario	
Número: 3	Usuario: Administrador
Nombre Historia: Eliminar autores	
Prioridad en el negocio: Media	Nivel de complejidad: Medio
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: El sistema debe permitir al administrador eliminar los usuarios existentes en	

la plataforma. Para ello se muestran todos los usuarios almacenados, dando la opción de seleccionar el que desea eliminar.

Tabla 83: Mostrar autores

Historias de Usuario	
Número: 4	Usuario: Todos
Nombre Historia: Mostrar autores	
Prioridad en el negocio: Media	Nivel de complejidad: Medio
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: El sistema debe permitir mostrar el perfil del autor con todos los datos del autor.	

Tabla 84: Eliminar comentarios

Historias de Usuario	
Número: 15	Usuario: Administrador
Nombre Historia: Eliminar comentario	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 2
Descripción: El sistema debe permitir al administrador eliminar los comentarios existentes en la plataforma. Para ello se muestran todos los comentarios almacenados, dando la opción de seleccionar el que desea eliminar.	

Tabla 85: Mostrar comentarios

Historias de Usuario	
Número: 16	Usuario: Todos

Nombre Historia: Mostrar comentarios	
Prioridad en el negocio: Alta	Nivel de complejidad: Medio
Tiempo de estimación: 1	Iteración asignada: 1
Descripción: El sistema debe permitir mostrar los comentarios de los usuarios en el blog correspondiente.	

Tabla 86: Modificar blog

Historias de Usuario	
Número: 11	Usuario: Autor del blog
Nombre Historia: Modificar blog	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 1
Descripción: Permitir al autor del blog actualizar los datos de su blog en el sistema, mediante un formulario que contenga los datos actuales del blog correspondiente a ese autor, para que elija los datos que quiere actualizar.	

Tabla 87: Invitar a un usuario a ser coautor de un blog creado.

Historias de Usuario	
Número: 46	Usuario: autor
Nombre Historia: Invitar a un usuario a ser coautor de un blog creado.	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 2
Descripción: El sistema debe permitir invitar a un usuario a ser coautor de un blog creado.	



Tabla 88: Aceptar la invitación de coautor.

Historias de Usuario	
Número: 47	Usuario: Todos
Nombre Historia: Aceptar la invitación de coautor.	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 2
Descripción: El sistema debe permitir a un usuario aceptar la invitación a ser coautor de un blog.	

Tabla 89: Eliminar coautor de un blog.

Historias de Usuario	
Número: 48	Usuario: Autor
Nombre Historia: Eliminar coautor de un blog.	
Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: El sistema debe permitir al autor eliminar los coautores de un blog.	

Tabla 90: Seleccionar entradas similares como referencia al crear una entrada en un blog.

Historias de Usuario	
Número: 49	Usuario: Autor y coautor
Nombre Historia: Seleccionar entradas similares como referencia al crear una entrada en un blog.	

Prioridad en el negocio: Media	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 2
Descripción: El sistema debe permitir seleccionar entradas similares al crear entrada en un blog.	

Tabla 91: Eliminar entrada de un blog

Historias de Usuario	
Número: 19	Usuario: Autor y coautor
Nombre Historia: Eliminar entrada de un blog	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 1
Descripción: El sistema debe permitir al autor y coautor eliminar entradas creadas por ellos, mediante una tabla donde se listan todas las entradas del blog perteneciente al autor; y si es coautor solo las creadas por él, con un botón para seleccionar la opción de eliminar.	

Tabla 92: Mostrar entrada de un blog

Historias de Usuario	
Número: 20	Usuario: Todos
Nombre Historia: Mostrar entrada de un blog.	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Tiempo de estimación: 1	Iteración asignada: 1
Descripción: El sistema debe permitir mostrar las entradas de un blog.	

Prototipo:

## Anexo 2: Tarjetas CRC

Tabla 93 : Tarjeta CRC modelo\_administradores

Tarjeta CRC	
Clase: modelo_administradores	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Crear un Administrador del sistema.</li> <li>• Actualizar los datos de un Administrador del sistema.</li> <li>• Eliminar un Administrador del sistema</li> <li>• Obtener los datos de los administradores del sistema</li> </ul>	<p>Conexión</p>

Tabla 94: Tarjeta CRC modelo\_control

Tarjeta CRC	
Clase: modelo_control	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Iniciar control de un blog</li> <li>• Actualizar la cantidad de visitas de un blog.</li> <li>• Actualizar la cantidad de me gustas de un blog.</li> <li>• Actualizar la cantidad de comentarios de un blog.</li> <li>• Actualizar la cantidad de seguidores de un blog.</li> </ul>	<p>Conexión</p>

<ul style="list-style-type: none"> <li>• Seleccionar el control de un blog.</li> </ul>	
--	--

Tabla 95: Tarjeta CRC modelo\_seguidores

Tarjeta CRC	
Clase: modelo_seguidores	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Crear un seguidor de un blog.</li> <li>• Seleccionar los seguidores de un blog.</li> </ul>	Conexión

Tabla 96: Tarjeta CRC modelo\_categoria

Tarjeta CRC	
Clase: modelo_categoria	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Crear categoría por parte del administrador.</li> <li>• Sugerir categoría por cualquier usuario.</li> <li>• Aceptar categoría sugerida por parte del administrador.</li> <li>• Actualizar la cantidad de uso de una categoría.</li> <li>• Actualizar la cantidad de seguidores de una categoría.</li> </ul>	Conexión

<ul style="list-style-type: none"> <li>• Seleccionar las categorías aprobadas.</li> <li>• Seleccionar las categorías sin aprobar.</li> <li>• Seleccionar los datos de una categoría.</li> <li>• Verificar si la categoría ya existe.</li> <li>• Seleccionar las categorías más usadas en la plataforma.</li> <li>• Seleccionar las categorías más seguidas en la plataforma.</li> </ul>	
---	--

Tabla 97: Tarjeta CRC modelo\_buscarBlog

Tarjeta CRC	
Clase: modelo_buscarBlog	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Seleccionar los blogs más visitados.</li> <li>• Seleccionar los blogs más comentados.</li> <li>• Seleccionar los blogs más seguidos.</li> <li>• Seleccionar los blogs más gustados.</li> <li>• Seleccionar los blogs más recientes.</li> <li>• Seleccionar los blogs de un determinado autor.</li> <li>• Seleccionar los blogs relacionados.</li> <li>• Seleccionar los blogs de un determinado coautor.</li> <li>• Seleccionar los blogueros más seguidos.</li> </ul>	Conexión

<ul style="list-style-type: none"> <li>• Seleccionar los datos del autor de un blog.</li> <li>• Buscar blogs con filtro.</li> <li>• Buscar blog sin filtro</li> </ul>	
---	--

Tabla 98: Tarjeta CRC modelo\_usuarios

Tarjeta CRC	
Clase: modelo_autores	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Crear un autor.</li> <li>• Actualizar los datos de un autor.</li> <li>• Seleccionar los datos de un autor.</li> <li>• Verificar usuario de un autor.</li> <li>• Verificar usuario y contraseña de un autor.</li> <li>• Eliminar un autor.</li> </ul>	Conexión,

### Anexo 3: Tareas de ingeniería

Tabla 99: Tarea 1 de HU 46

Tarea de Ingeniería	
<b>Número de tarea:</b> 1	<b>Número de Historia de Usuario:</b> 46
<b>Nombre de la tarea:</b> Implementar el formulario encargado de permitir al autor de un blog invitar a una persona a ser coautor de su blog.	

<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 13/4/2015	<b>Fecha Fin:</b> 13/4/2015
<b>Descripción:</b> Se diseña e implementa el formulario con los campos necesarios para que el autor del blog pueda invitar a una persona a ser coautor de su blog.	

Tabla 100: Tarea 2 de HU 46

Tarea de Ingeniería	
<b>Número de tarea:</b> 2	<b>Número de Historia de Usuario:</b> 46
<b>Nombre de la tarea:</b> Implementar el controlador encargado de procesar la información entrada por el autor para invitar a ser coautor.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 13/4/2015	<b>Fecha Fin:</b> 13/4/2015
<b>Descripción:</b> Se implementa el controlador solicitudCoautor.php el cual se encarga de realizar una correcta validación a los datos para garantizar la seguridad de la información, y una vez pasada la validación se le da paso al modelo para que sea almacenada la información en la base de datos, luego se notifica la persona invitada, y una vez concluido este proceso el controlador se encarga de enviar la respuesta a la vista para que el autor sea informado si el proceso se realizó con éxito o si hubo errores.	

Tabla 101: Tarea 3 de HU 46

Tarea de Ingeniería	
<b>Número de tarea:</b> 3	<b>Número de Historia de Usuario:</b> 46
<b>Nombre de la tarea:</b> Insertar los valores en la entidad 'coautores' de la base de datos de la plataforma.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 13/4/2015	<b>Fecha Fin:</b> 13/4/2015
<b>Descripción:</b> Se implementa en el modelo_coautores.php la función que se encarga de almacenar en la entidad 'coautores' de la base de datos de la plataforma la información referente a un coautor suministrada por el controlador de este proceso.	

Tabla 102: Tarea 4 de HU 46

Tarea de Ingeniería	
<b>Número de tarea:</b> 4	<b>Número de Historia de Usuario:</b> 46
<b>Nombre de la tarea:</b> Implementar la función que notificará a la persona invitada.	

<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 13/4/2015	<b>Fecha Fin:</b> 13/4/2015
<b>Descripción:</b> Se implementa en el la clase notificaciones.php la función que se encarga de notificar a una persona que ha sido invitada por el autor de un blog a ser coautor del mismo.	

Tabla 103: Tarea 1 de HU 50

Tarea de Ingeniería	
<b>Número de tarea:</b> 1	<b>Número de Historia de Usuario:</b> 50
<b>Nombre de la tarea:</b> Implementar el formulario encargado de permitir al usuario sugerir una categoría.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha Inicio:</b> 20/4/2015	<b>Fecha Fin:</b> 20/4/2015
<b>Descripción:</b> Se diseña e implementa el formulario con los campos necesarios para que un usuario pueda sugerir una categoría	

Tabla 104: Tarea 2 de HU 50

Tarea de Ingeniería	
<b>Número de tarea:</b> 2	<b>Número de Historia de Usuario:</b> 50
<b>Nombre de la tarea:</b> Implementar el controlador encargado de procesar la información entrada por el usuario para sugerir una categoría.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 20/4/2015	<b>Fecha Fin:</b> 20/4/2015
<b>Descripción:</b> Se implementa el controlador sugerirCategorias.php el cual se encarga de realizar una correcta validación a los datos para garantizar la seguridad de la información, y una vez pasada la validación se le da paso al modelo para que sea almacenada la información en la base de datos, y una vez concluido este proceso el controlador se encarga de enviar la respuesta a la vista para que el usuario sea informado si el proceso se realizó con éxito o si hubo errores.	

Tabla 105: Tarea 3 de HU 50

Tarea de Ingeniería	
<b>Número de tarea:</b> 3	<b>Número de Historia de Usuario:</b> 50
<b>Nombre de la tarea:</b> Insertar los valores en la entidad 'categorías' de la base de datos de la plataforma.	



<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha Inicio:</b> 20/4/2015	<b>Fecha Fin:</b> 20/4/2015
<b>Descripción:</b> Se implementa en el modelo_categoria.php la función que se encarga de almacenar en la entidad 'categorías' de la base de datos de la plataforma la información referente a una categoría suministrada por el controlador de este proceso.	

Tabla 106: Tarea 1 de HU 42

Tarea de Ingeniería	
<b>Número de tarea:</b> 1	<b>Número de Historia de Usuario:</b> 42
<b>Nombre de la tarea:</b> Implementar el formulario encargado de permitir al usuario solicitar ser coautor de un blog.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 11/4/2015	<b>Fecha Fin:</b> 11/4/2015
<b>Descripción:</b> Se diseña e implementa el formulario con los campos necesarios para que un usuario pueda solicitar ser coautor de un blog.	

Tabla 107: Tarea 2 de HU 42

Tarea de Ingeniería	
<b>Número de tarea:</b> 2	<b>Número de Historia de Usuario:</b> 42
<b>Nombre de la tarea:</b> Implementar el controlador encargado de procesar la información entrada por el usuario para solicitar ser coautor.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha Inicio:</b> 11/4/2015	<b>Fecha Fin:</b> 11/4/2015
<b>Descripción:</b> Se implementa el controlador solicitudCoautor.php el cual se encarga de realizar una correcta validación a los datos para garantizar la seguridad de la información, y una vez pasada la validación se le da paso al modelo para que sea almacenada la información en la base de datos, y una vez concluido este proceso el controlador se encarga de enviar la respuesta a la vista para que el usuario sea informado si el proceso se realizó con éxito o si hubo errores.	

Tabla 108: Tarea 3 de HU 42

Tarea de Ingeniería	
<b>Número de tarea:</b> 3	<b>Número de Historia de Usuario:</b> 42

<b>Nombre de la tarea:</b> Insertar los valores en la entidad 'coautores' de la base de datos de la plataforma.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha Inicio:</b> 11/4/2015	<b>Fecha Fin:</b> 11/4/2015
<b>Descripción:</b> Se implementa en el modelo_coautores.php la función que se encarga de almacenar en la entidad 'coautores' de la base de datos de la plataforma la información referente a un coautor suministrada por el controlador de este proceso.	

#### Anexo 4: Pruebas de caja blanca

<p><b>Método: Actualizar blog</b></p> <pre> public function ActualizarBlog() {     \$consulta = "UPDATE `blogs` SET `titulo`=?, `descripcion`=? WHERE `id`=? ;"; (1)     \$params array(\$this-&gt;titulo, \$this-&gt;desc, \$this-&gt;id ); (1)     \$sql = \$this-&gt;conexion-&gt;consulta_base_de_datos(\$consulta, \$params); (1)     if (!\$sql) (2)     return false; (3)     else (4)     return true; } (5)         </pre>	<p><b>Grafo resultante:</b></p> <pre> graph TD     1((1)) --&gt; 2((2))     2 --&gt; 3((3))     2 --&gt; 4((4))     3 --&gt; 5((5))     4 --&gt; 5         </pre>
<p><b>Complejidad Ciclomática:</b></p> <p><math>V(G) = \# \text{ de regiones} = 2</math></p> <p><math>V(G) = A - N + 2 = 5 - 5 + 2 = 2</math></p> <p><math>V(G) = P + 1 = 1 + 1 = 2</math></p>	<p><b>Caminos básicos:</b></p> <p>{1235}, {1245}</p>

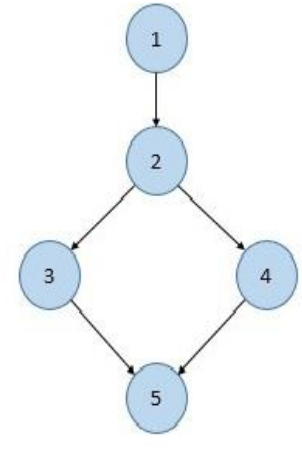
<p><b>Método: Seleccionar blogs más seguidos</b></p> <pre> public function SelectBlogMasSeguidos() {     \$consulta = "Select usuario.img_usuario, usuario.nombre, blogs.titulo, blogs.descripcion,     blogs.id, control.cant_comentarios, control.cant_visitas, control.cant_like,     control.cant_seguidores from usuario join blogs join control Where         </pre>	<p><b>Grafo resultante:</b></p>
--	---------------------------------

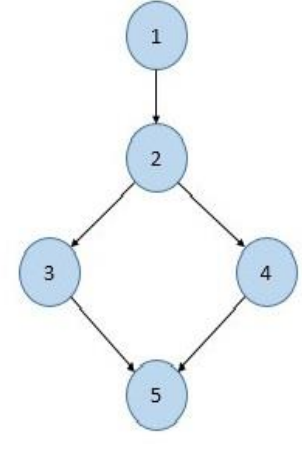
<pre> usuario.usuario=blogs.autor_user and blogs.id=control.blogs_id order by control.cant_seguidores DESC limit 10;"; (1)  \$sql = \$this-&gt;conexion-&gt;Obtener_Resultados (\$consulta); (1)  if (!\$sql) (2)  return false; (3)  else (4)  return true;  } (5)         </pre>	<pre> graph TD     1((1)) --&gt; 2((2))     2 --&gt; 3((3))     2 --&gt; 4((4))     3 --&gt; 5((5))     4 --&gt; 5         </pre>
--	---

<p><b>Complejidad Ciclomática:</b></p> <p><math>V(G) = \# \text{ de regiones} = 2</math></p> <p><math>V(G) = A - N + 2 = 5 - 5 + 2 = 2</math></p> <p><math>V(G) = P + 1 = 1 + 1 = 2</math></p>	<p><b>Caminos básicos:</b></p> <p>{1235}, {1245}</p>
--	--

<p><b>Método: Seleccionar blogs más comentados</b></p> <pre> public function SelectBlogMasComentados() { \$sql = "Select usuario.img_usuario,usuario.nombre, blogs.titulo, blogs.descripcion, blogs.id, control.cant_comentarios,control.cant_visitas, control.cant_like, control.cant_seguidores from usuario join blogs join control Where usuario.usuario=blogs.autor_user and blogs.id=control.blogs_id order by control.cant_comentarios DESC limit 10;"; (1)  \$sql = \$this-&gt;conexion-&gt;Obtener_Resultados (\$consulta); (1)  if (!\$sql) (2)  return false; (3)  else (4)  return true;  } (5)         </pre>	<p><b>Grafo resultante:</b></p> <pre> graph TD     1((1)) --&gt; 2((2))     2 --&gt; 3((3))     2 --&gt; 4((4))     3 --&gt; 5((5))     4 --&gt; 5         </pre>
--	---

<p><b>Complejidad Ciclomática:</b></p> <p><math>V(G) = \# \text{ de regiones} = 2</math></p> <p><math>V(G) = A - N + 2 = 5 - 5 + 2 = 2</math></p> <p><math>V(G) = P + 1 = 1 + 1 = 2</math></p>	<p><b>Caminos básicos:</b></p> <p>{1235}, {1245}</p>
--	--

<p><b>Método: Seleccionar blogs más visitados</b></p> <pre> public function SelectBlogMasVisitados() {     \$consulta = "Select usuario.img_usuario,usuario.nombre, blogs.titulo, blogs.descripcion,     blogs.id, control.cant_comentarios,control.cant_visitas, control.cant_like,     control.cant_seguidores from usuario join blogs join control Where     usuario.usuario=blogs.autor_user and blogs.id=control.blogs_id order by     control.cant_visitas DESC limit 10;";     \$sql = \$this-&gt;conexion-&gt;Obtener_Resultados (\$consulta);     if (!\$sql)     return false;     else     return true; }                 </pre> <p>(1) (1) (2) (3) (4) (5)</p>	<p><b>Grafo resultante:</b></p> 
<p><b>Complejidad Ciclomática:</b></p> <p><math>V(G) = \# \text{ de regiones} = 2</math></p> <p><math>V(G) = A - N + 2 = 5 - 5 + 2 = 2</math></p> <p><math>V(G) = P + 1 = 1 + 1 = 2</math></p>	<p><b>Caminos básicos:</b></p> <p>{1235}, {1245}</p>

<p><b>Método: Seleccionar blogs más gustados</b></p> <pre> public function SelectBlogMasGustados() {     \$consulta = "Select usuario.img_usuario,usuario.nombre, blogs.titulo, blogs.descripcion,     blogs.id, control.cant_comentarios,control.cant_visitas, control.cant_like,     control.cant_seguidores from usuario join blogs join control Where     usuario.usuario=blogs.autor_user and blogs.id=control.blogs_id order by control.cant_like     DESC limit 10;";     \$sql = \$this-&gt;conexion-&gt;Obtener_Resultados (\$consulta);     if (!\$sql)     return false;     else     return true; }                 </pre> <p>(1) (1) (2) (3) (4) (5)</p>	<p><b>Grafo resultante:</b></p> 
---	---

<p><b>Complejidad Ciclomática:</b></p> <p><math>V(G) = \# \text{ de regiones} = 2</math></p> <p><math>V(G) = A - N + 2 = 5 - 5 + 2 = 2</math></p> <p><math>V(G) = P + 1 = 1 + 1 = 2</math></p>	<p><b>Caminos básicos:</b></p> <p>{1235}, {1245}</p>
--	--

## Anexo 5: Interfaces principales de UCIBlogs

The screenshot displays the UCIBlogs platform interface. At the top, there is a navigation bar with 'UCIBlogs', 'Crear Blogs', and 'Sugerir Categorías'. To the right are input fields for 'Usuario' and 'Contraseña', and an 'Autenticar' button. Below the navigation bar is a large welcome message: 'Bienvenid@ a la Plataforma UCIBlogs' followed by a paragraph: 'Te brindamos la posibilidad con solo unos click de compartir tu conocimiento y contribuir con el proceso de Enseñanza y Aprendizaje de nuestra universidad. No lo dejes para después, crea tu propio blog educativo.'

Below the welcome message, there is a section titled 'Visita los 10 blogs más destacados en la plataforma.' This section contains a grid of featured blogs. The first blog is 'Mi blog de php' by Rigoberto Jiménez Pérez, with 4 followers, 0 comments, 10 visits, and 1 like. The second blog is 'El HTML y el CSS en la WEB' by Ana María Domínguez, with 0 followers, 0 comments, 4 visits, and 1 like. To the right of the featured blogs are three sidebars: 'Blogs recientes' (listing 'El HTML y el CSS en la WEB' and 'Mi blog de php'), 'Blogueros más seguidos' (listing Rigoberto Jiménez Pérez and Ana María Domínguez), and 'Categorías más usadas' (listing 'Css', 'Html', and 'Php', each with a count of 1).

Figura 9: Interfaz inicial de UCIBlogs.

UCIBlogs [Crear Blogs](#) [Sugerir Categorías](#) [Solicitar ser coautor](#)

# Mi blog de php

En este blog se publicarán las genialidades de php

UCIBlogs > Mi blog de php   167 4 4 5 Me gusta | Seguir | Compartir

**Entradas por categorías**

**Php**

Procesar JSON en petición ajax a PHP con jQuery

Como evitar inyección SQL limpiando las variables con PHP

Como eliminar un directorio con subdirectorios con php

**Blogs Relacionados**

**Php**

No hay blogs

**Procesar JSON en petición ajax a PHP con jQuery**

Procesar JSON en petición ajax a PHP con jQuery... [Leer más](#)

**Como evitar inyección SQL limpiando las variables con PHP**

Como evitar inyección SQL limpiando las variables con PHP... [Leer más](#)

**Como eliminar un directorio con subdirectorios con php**

Como eliminar un directorio con subdirectorios con php... [Leer más](#)

**Mi Perfil**

**Nombre :** Rigoberto Jiménez Perez

**Correo :** rperez@estudiantes uci cu

**Categoría :** Estudiante

**Mis blogs**

Mi blog de php 167

**Entradas ordenadas por fechas**

2015

Figura 10: Interfaz principal de un blog.

UCIBlogs [Crear Blogs](#) [Sugerir Categorías](#) [Solicitar ser coautor](#)

UCIBlogs > La Ingeniería de software > Patrones de diseño 24 | 2 |

**Entradas por categorías**

**3d**

Patrones de diseño

**Entradas Relacionadas**

No hay entradas

**Escrito por:** Yaima

**3d**

*Patrones de diseño*

Los patrones te ayudan mucho a la hora de programar Adjunto les dejo unos documentos muy buenos para que se los estudien, cualquier duda no duden en comentar.

Espero que les ayude un poco con el tema de los patrones

**Adjuntos**

zip

**Mi Perfil**

**Nombre :** Yaima

**Correo :** yfiallo@uci cu **Categoría :** Profesor

**Mis blogs**

La Ingeniería de software 18

**Entradas ordenadas por fechas**

2015

29/5 Patrones de diseño

No hay seguidores

**Entradas referenciadas por el autor**

↩ Esta entrada no tiene referencias

**Comentarios**

entre el nombre


**Rigo dice:** 2015-05-30 03:40:47

Muchas gracias profe, nos sirvió mucho

**Juan dice:** 2015-05-29 13:08:10

jaja

Figura 11: Interfaz de una entrada.

UCIBlogs [Crear Blogs](#) [Sugerir Categorías](#) [Administrar Blogs](#) [Ver mis Blogs](#)  Ana María Domínguez [Salir](#)

[Blog](#) / [Admin](#) / Entradas Publicadas

**Entradas Publicadas**

- Crear entrada
- Actualizar datos del blog
- Aceptar o denegar coautores ▼
- » Eliminar coautores
- » Invitar a ser coautor
- Actualizar Perfil
- Invitar a leer el blog

Título	Autor	Comentarios	Fecha de creación			
Mi primer truco	Ana Maria Dominguez	0	2015-05-24 18:45:03	◀	<a href="#">Editar</a>	<a href="#">Eliminar</a>
HTML5 y sus cosas	Ana Maria Dominguez	0	2015-05-24 18:43:36	◀	<a href="#">Editar</a>	<a href="#">Eliminar</a>

Figura 12: Interfaz de la administración de un blog.

UCIBlogs [Crear Blogs](#) Rigoberto [Salir](#)

UCIBlog / [Admin](#) / Eliminar usuarios

Gestión de Categorías ▼

- » Crear Categorías
- » Eliminar categorías
- » Aprobar o denegar categorías sugeridas
- Habilitar o inhabilitar usuarios**
- Eliminar blogs
- Eliminar entradas
- Eliminar comentarios
- Gestión de Administradores ▼
- » Crear administrador
- » Eliminar y actualizar administradores
- Enviar Correo a los autores de UCIBlogs

Habilitar o inhabilitar usuarios

Nombre	Usuario	Categoría	
Ana Maria Dominguez	rigo	Estudiante	<a href="#">Inhabilitar</a>

Figura 13: Interfaz de la administración de UCIBlogs.