

**Universidad de las Ciencias Informáticas**

**Facultad 3**



**Herramienta para la evaluación de la originalidad en  
documentos de textos digitales**

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

**Autor(es):** Martha de la Caridad Cardonne Fong.

Ariel Tomás Sánchez Pérez.

**Tutor:** MsC. Yarina Amoroso Fernández.

**Co-Tutor:** Ing. Reinier Fernández Coello.

*Ciudad de La Habana, Junio 2013*

*“Año 54 de la Revolución”*

*"... aquí está una de las tareas de la juventud: empujar, dirigir con el ejemplo la producción del hombre de mañana. Y en esta producción, en esta dirección, está comprendida la producción de sí mismos..."*

*Ernesto Che Guevara*

## **DECLARACIÓN DE AUTORÍA**

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

\_\_\_\_\_

Martha Cardonne Fong

Firma del Autor

\_\_\_\_\_

Ariel Tomás Sánchez Pérez

Firma del Autor

\_\_\_\_\_

MsC. Yarina Amoroso Fernández

Firma del Tutor

\_\_\_\_\_

Ing. Reinier Fernández Coello

Firma del Co-Tutor

## **DATOS DE CONTACTO**

### **Autores:**

Martha Cardonne Fong.

**Correo Electrónico:** [mlcardonne@estudiantes.uci.cu](mailto:mlcardonne@estudiantes.uci.cu)

Ariel Tomás Sánchez Pérez.

**Correo Electrónico:** [atsanchez@estudiantes.uci.cu](mailto:atsanchez@estudiantes.uci.cu)

### **Tutor:**

MsC Yarina Amoroso Fernández.

**Correo Electrónico:** [yarina@uci.cu](mailto:yarina@uci.cu)

### **Co-Tutor(es):**

Ing. Reinier Fernández Coello.

**Correo Electrónico:** [rfoello@uci.cu](mailto:rfoello@uci.cu)

## AGRADECIMIENTOS

A mis padres Félix Cardonne Fong y Adriana Fong Freitas que han sido mi inspiración durante toda mi vida. Gracias por ser los mejores padres del mundo, por apoyarme en todo momento, por depositar su confianza en mí, por su dedicación, amor, ternura y comprensión. Gracias por su paciencia y entrega, por estar siempre a mi lado, los quiero mucho.

A mi hermano Rodolfo Hadfeg Fong, por su apoyo y amor incondicional. Gracias por ser el mejor hermano, gracias por tus consejos y cariño, por estar siempre a mi lado.

A mi abuelo Jacinto Fong Álvarez, por sus consejos, apoyo, dedicación y amor. Gracias por guiarme hacia el camino del éxito y convertirme en la persona que soy hoy. Te quiero mucho y siempre te llevaré en mi corazón y presente en mi pensamiento.

A mis abuelos Xiomara Freitas Manzano, Miriam Fong Leyva y Félix Cardonne Molina

por su dedicación, amor y ternura, los quiero mucho.

A mis otros padres Liliana Hung Vargas y Santiago Renté Gómez por sus consejos y apoyo incondicional.

A mis amigas Greysis Donatién Galano y Taimé Pacheco Ruiz por su apoyo, por sus consejos, por todos los momentos de alegría que compartieron conmigo y por tenerme presente en cada momento de sus vidas.

A mis tutores Reinier Coello Fernández y Yarina Amoroso Fernández quienes con su paciencia y entrega me han guiado en la elaboración de este trabajo.

A Maricela, Laritza y a la cangreja Ailyn por ser quienes han sido hasta ahora, personas en las que se puede confiar, que siempre voy a querer y respetar por ayudarme y acompañarme en todo momento.

A todos ustedes muchas gracias.

**Martha Cardonne Fong.**

A mis padres por su dedicación hacia mí y el apoyo que me han dado durante estos 5 años de carrera, por confiar en todo momento a pesar de las circunstancias.

*A mi hermano que fue el primero en mi familia en alcanzar nivel superior y de esta manera se convirtió en un ejemplo y una meta para mí.*

*A Ismary que ha sido de gran apoyo para mí durante estos 2 años, gracias por tu dedicación y cariño, que me han ayudado a seguir adelante y romper cualquier obstáculo que se ha puesto en el camino, gracias por no dejarme ir, te quiero.*

*A mis amigos y compañeros de aula que me ayudaron mucho en esta trayectoria sin ellos*

*esto no hubiera sido posible especialmente a Raydel que es como mi hermano.*

*A Martha la mejor compañera de tesis que he podido tener gracias por ayudarme tanto.*

*A mis tutores Reinier Fernández Coello, y Yarina Amoroso Fernández.*

*A este país y a la revolución cubana que me dio la posibilidad de tener una carrera universitaria.*

**Ariel Tomás Sánchez Pérez**

## DEDICATORIA

*A mis padres Adriana Fong Freites y Félix Cardonne Fong quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento, depositando su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento en mi inteligencia y capacidad. Es por ello que he llegado a ser lo que soy ahora.*

**Martha Cardonne Fong.**

*A mi familia, y en especial a mi mamá y a mi papá por todo el apoyo, dedicación y confianza que siempre han depositado en mí. Sin ustedes no hubiera podido lograr lo que he sido hasta hoy.*

**Ariel Tomás Sánchez Pérez**

## RESUMEN

En la Universidad de las Ciencias Informáticas se realizan disímiles eventos científicos, en los cuales se generan un gran cúmulo de documentos en las investigaciones realizadas por parte de los estudiantes. Estos en ciertas ocasiones llevan a cabo la copia sobre otros documentos para poder completar su investigación y no tienen en cuenta las citas o referencias una vez que copian ideas o textos de otros trabajos. Además presentan trabajos generados en cursos anteriores sobre los cuales realizan ciertas modificaciones, suplantando de esta forma la identidad del verdadero autor. En busca de originalidad a los profesores se les hace engorroso revisar manualmente cada documento.

Con el objetivo de desarrollar una herramienta que verifique la originalidad en los documentos de textos digitales generados en la esfera de investigación en la Universidad de Ciencias Informáticas, en la presente investigación se realiza un estudio del proceso de validación de la originalidad en los documentos de textos digitales. De igual forma se describen las herramientas, tecnologías y artefactos generados en el desarrollo de la solución.

Como resultado de la presente investigación se obtuvo una herramienta para la evaluación de la originalidad en documentos de textos digitales que facilitará la determinación del grado de originalidad en los documentos emitidos en la esfera de investigación en la Universidad de las Ciencias Informáticas.

### **Palabras claves:**

Documentos de textos, evaluación, originalidad, plagio.



## ÍNDICE DE CONTENIDOS

Introducción .....	1
Capítulo 1: Fundamentación teórica.....	5
1.1 Detección del plagio .....	5
1.1.1 Análisis de las soluciones existentes para la evaluación de la originalidad en documentos de texto digitales.....	6
1.1.2 Procedimientos asociados con la determinación del grado de originalidad en los documentos de textos .....	8
1.1.3 Características asociadas a los sistemas que evalúan la originalidad en los documentos de textos .....	8
1.1.4 Algoritmos para la comparación de textos.....	9
1.2 Metodología de desarrollo .....	11
1.3 Lenguaje de programación .....	13
1.4 Plataforma de desarrollo.....	15
1.5 Herramienta Case .....	15
1.6 Herramienta para el modelado de prototipos web.....	16
1.7 Gestor de base de datos .....	17
1.8 Servidor Web.....	18
1.9 Sistema Gestor de Contenido.....	18
1.10 Conclusiones parciales.....	20
Capítulo 2: Planificación, diseño y desarrollo .....	21
2.1 Descripción del sistema.....	21
2.2 Planificación .....	21
2.2.1 Requisitos del sistema .....	22
2.2.2 Historias de usuario .....	24
2.2.3 Plan de iteraciones.....	27
2.3.4 Plan de entregas .....	28
2.3 Diseño .....	29
2.3.1 Modelo del Dominio .....	29
2.3.2 Diagrama de clases .....	30
2.3.3 Patrones de diseño .....	31
2.3.4 Tarjetas CRC .....	35

---

2.3.5 Modelo de datos.....	36
2.4 Implementación .....	37
2.4.1 Modelo de implementación .....	37
2.4.2 Tareas de ingeniería por Historia de Usuarios .....	38
2.5 Conclusiones parciales.....	42
Capítulo 3: Validación de la solución.....	43
3.1 Validación de los requisitos .....	43
3.2 Validación del diseño.....	46
3.3 Pruebas de Caja Blanca .....	50
3.4 Pruebas de Caja Negra .....	52
3.5 Conclusiones parciales.....	56
Conclusiones generales.....	57
Recomendaciones .....	58
Referencias bibliográficas .....	59

## ÍNDICE DE TABLAS

Tabla 1. Comparación entre los software que evalúan la originalidad en documentos.....	7
Tabla 2. HU1: Autenticar usuario. ....	25
Tabla 3. HU2: Cargar documento en el sistema.....	25
Tabla 4. HU3: Analizar documentos. ....	26
Tabla 5. HU4: Mostrar resultados de la evaluación. ....	26
Tabla 6. HU5: Guardar documento. ....	27
Tabla 7. HU6: Generar reporte.....	27
Tabla 8. HU7: Gestionar documento. ....	27
Tabla 9. Plan de Entregas.....	29
Tabla 10. Tarjeta CRC: SimilitudDocumental. ....	36
Tabla 11. Distribución de las tareas por cada Historia de Usuario.....	40
Tabla 12. Descripción de la tarea de ingeniería 7. ....	40
Tabla 13. Descripción de la tarea de ingeniería 8. ....	40
Tabla 14. Descripción de la tarea de ingeniería 9. ....	41
Tabla 15. Descripción de la tarea de ingeniería 10. ....	41
Tabla 16. Descripción de la tarea de ingeniería 11. ....	42
Tabla 17. Descripción del Caso de Prueba de Aceptación 3.....	53
Tabla 18. Descripción del caso de Prueba de Aceptación 4.....	54

## ÍNDICE DE FIGURAS

Figura 1. Modelo del Dominio. ....	30
Figura 2. Diagrama de clases. ....	31
Figura 3. Patrón arquitectónico Modelo Vista Controlador. ....	32
Figura 4. Clase experta en información referente a los documentos. ....	33
Figura 5. Clase controladora del módulo SimilitudDocumental.....	34
Figura 6. Modelo de la base de datos. ....	36
Figura 7. Diagrama de despliegue. ....	38
Figura 8. Prototipo de interfaz de usuario de la funcionalidad Adicionar documento. ....	44
Figura 9. Representación de la cantidad de clases por cantidad de procedimientos que contienen. .....	47
Figura 10. Resultados de la evaluación de la métrica TOC en los atributos Responsabilidad, Comportamiento y Reutilización respectivamente. ....	47
Figura 11. Representación de la cantidad de clases por cantidad de relaciones de usos que poseen.....	48
Figura 12. Resultados de la evaluación de la métrica RC en los atributos Acoplamiento y Complejidad respectivamente. ....	49
Figura 13. Resultados de la evaluación de la métrica RC en los atributos Reutilización y Pruebas respectivamente.....	49
Figura 14. Código fuente del método get_tabla_resultado. ....	51
Figura 15. Grafo de flujo asociado al algoritmo. ....	51

## INTRODUCCIÓN

Desde los últimos años el uso de la informática en la educación es una de las aplicaciones más comunes e importantes; lo que ha permitido contar con métodos novedosos, sencillos y eficaces para el aprendizaje en cualquier nivel educacional, además proporcionando herramientas que facilitan la búsqueda de información haciéndola más rápida y abarcadora. Pero no todo ha sido beneficioso, pues este ingenioso método ha proporcionado a su vez la facilidad para plagiar, disminuyendo la motivación por la investigación y afectando la originalidad en los documentos que se desean crear.

La originalidad es la cualidad de las obras creadas que las hace ser nuevas o novedosas, y que las distingue de las copias, las falsificaciones y el plagio. Una obra original no deriva de otras obras ni es una copia realizada sobre otra, que sería su origen (1).

El plagio es definido como «acción de copiar en lo sustancial obras ajenas, dándolas como propias» (1). Es un tema debatido en las disímiles universidades pues se considera como una alteración de la originalidad al constituir una actividad mecánica y poco creativa por parte de los estudiantes. La apropiación de ideas, afirmaciones y textos de forma digital, es cada vez más habitual y creciente en el ámbito universitario, lo cual constituye un hecho delictivo por copiarlos y hacerlos pasar por su propia autoría.

En Cuba el plagio no deja de ser un tema importante en el área educacional. En la UCI se generan un gran cúmulo de trabajos en las investigaciones realizadas, por lo que a los profesores se les hace casi imposible revisar manualmente cada uno de los trabajos, evaluando la originalidad de los mismos como principal indicador del plagio. Para la realización de las tareas orientadas en la esfera de investigación, a los estudiantes se les ofrecen diversas bibliografías, lo que trae consigo en la mayoría de los casos que se generen documentos de textos con similitudes al llevar a cabo la copia total o parcial de estos materiales, sin referenciar las ideas o fragmentos de textos copiados. En muchas ocasiones los estudiantes interactúan con los mismos trabajos generados en cursos anteriores, ya sean trabajos realizados en eventos como: Fórum de Historia, Seminario Juvenil Martiano y Jornada Científica Estudiantil. En situaciones como esta se suplanta la identidad del verdadero autor del trabajo.

El plagio académico se extiende como una plaga y constituye una tarea de vital importancia que los estudiantes reflejen su espíritu investigador a través del desarrollo de sus habilidades

(sintetizar, redactar e interpretar textos) y la creación de documentos de textos con originalidad.  
(2)

Partiendo de lo expuesto anteriormente se propone el siguiente problema a resolver: ¿Cómo verificar la originalidad en los documentos de textos digitales generados en la esfera de investigación en la Universidad de las Ciencias Informáticas?

Teniendo como **objeto de estudio**: el proceso de validación de la originalidad en los documentos de textos digitales.

Para resolver el problema planteado en el trabajo de diploma, se propone como **objetivo general**: desarrollar una herramienta que verifique la originalidad en los documentos de textos digitales generados en la esfera de investigación en la Universidad de Ciencias Informáticas.

Enfocado en el **campo de acción**: determinación del grado de originalidad en los documentos de textos digitales generados en la esfera de investigación en la Universidad de las Ciencias Informáticas.

Teniendo como referencia el problema a resolver y el objetivo general se plantea la siguiente **idea a defender**: si en la Universidad de las Ciencias Informáticas se cuenta con un sistema que evalúe el grado de originalidad en los documentos emitidos en la esfera de investigación, entonces se detectará el plagio en dichos documentos.

Con la finalidad de dar cumplimiento al objetivo trazado se proponen los siguientes **objetivos específicos**:

1. Elaborar el marco teórico de la investigación.
2. Elaborar la propuesta de solución.
3. Validar la solución propuesta.

Las **tareas de investigación** trazadas quedaron estructuradas de la siguiente forma:

1. Valoración de las herramientas existentes para la evaluación de la originalidad en documentos de texto a nivel nacional e internacional.
2. Valoración de los algoritmos de comparación de texto.

3. Selección de la metodología de desarrollo del software aplicable para la solución informática al problema planteado.
4. Selección del lenguaje de programación y las tecnologías para el desarrollo de la herramienta.
5. Definición de los requisitos del sistema.
6. Diseño e implementación de la herramienta para verificar la originalidad en los documentos de textos digitales.
7. Validación de los resultados obtenidos.

La realización de dichas tareas se sustenta en los siguientes **métodos de investigación**:

**Métodos teóricos:**

- ✓ Histórico – Lógico: permitió realizar un estudio sobre las principales metodologías y herramientas empleadas en el desarrollo de aplicaciones web, las tendencias del uso actual de las mismas a nivel mundial, así como en los centros productivos de la universidad, con el fin de seleccionar las más apropiadas para darle cumplimiento al objetivo general de la presente investigación.
- ✓ Analítico – Sintético: permitió realizar el estudio teórico de la investigación facilitando el análisis de diversas bibliografías y la extracción de los elementos más importantes acerca del proceso de determinación del grado de originalidad en documentos de textos.

**Métodos empíricos:**

- ✓ Medición: permitió medir la calidad de la especificación de los requisitos y el grado de ambigüedad de estos, además de obtener una medida de la calidad del diseño para su validación.

El presente documento se estructura en 3 capítulos.

**CAPÍTULO 1.** Fundamentación Teórica: Contiene la base teórica para comprender el problema. Se describen los aspectos y conceptos asociados al dominio del problema a resolver, constituyendo estos, puntos esenciales para entender el entorno del mismo. Se presenta el

estado del arte de las herramientas que analizan la originalidad en documentos de textos y el conjunto de técnicas, herramientas, lenguajes y metodología involucrados en el desarrollo de la propuesta, así como la justificación de las seleccionadas para la solución del problema.

**CAPÍTULO 2.** Planificación, diseño e implementación: Se detallan elementos relacionados con la descripción y análisis del sistema informático para la evaluación de la originalidad en documentos de textos: los requisitos del software a través de las historias de usuario. Se describen los principales artefactos generados en estas fases como: Plan de iteraciones, Plan de entrega, Modelo de datos, Tarjetas CRC y las Tareas de ingeniería.

**CAPÍTULO 3.** Validación de la solución propuesta: se exponen los resultados de las pruebas y validaciones aplicadas al sistema para comprobar la calidad y fiabilidad de la solución obtenida, asegurando su correcto funcionamiento.



## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

En este capítulo se realiza un estudio del estado del arte de los sistemas existentes para la determinación del grado de originalidad en los documentos de textos digitales, y para ello se tienen en cuenta las características asociadas a los sistemas de detección del plagio y los algoritmos existentes para la comparación de textos. Además se expone la metodología que guiará todo el proceso de desarrollo del software, así como las herramientas y los métodos empleados en el desarrollo de una herramienta que verifique la originalidad en los documentos de textos digitales.

### **1.1 Detección del plagio**

El plagio constituye una de las prácticas anti éticas que realizan hoy en día los estudiantes universitarios en la elaboración de sus trabajos. Cuando Internet no existían o no estaba tan difundido, los estudiantes para realizar una investigación debían acudir a la biblioteca en busca de bibliografías, lo que exigía horas de búsqueda entre las páginas de diversas obras para encontrar las ideas esenciales que les sirviera para el desarrollo de sus trabajos. Al igual que resulta más fácil copiar, también es más sencillo determinar el grado de originalidad en los documentos debido al avanzado desarrollo de nuevas tecnologías que colaboran en la identificación de la copia. (2)

#### **Tipos de plagio de acuerdo al propósito**

Muchos son los propósitos por los cuales los estudiantes cometen el plagio, ya sea para mejorar sus calificaciones en la esfera investigativa o para obtener méritos presentando un trabajo que fue copiado. Sin importar los motivos, el plagio puede producirse ya sea de forma intencional o accidental. (3)

- ✓ **Plagio intencionado:** Se considera como un engaño consciente, pues es cuando se presenta un trabajo como propio no siendo realmente de esta forma. (3)
- ✓ **Plagio no intencionado:** Es cuando se comete el plagio de forma accidental al olvidar referenciar el texto copiado o al parafrasear un texto que ya había consultado. (3)

### **1.1.1 Análisis de las soluciones existentes para la evaluación de la originalidad en documentos de textos digitales**

Uno de los instrumentos que se utilizan para la evaluación de la originalidad en los documentos de textos son los programas informáticos creados con esta finalidad. Muchos de estos software son utilizados en las universidades a nivel mundial debido a la gran cantidad de documentos que se generan en las investigaciones realizadas.

Con el objetivo de adquirir experiencias en cuánto al funcionamiento de la herramienta para la evaluación de la originalidad en los documentos de textos, se realiza un estudio de los sistemas creados con esta finalidad a nivel nacional e internacional, los cuales de forma general tienen como objetivo determinar el grado de similitud en determinados documentos. A continuación se exponen las características de cada uno de ellos.

**The Plagiarism Checker:** Es una aplicación online, gratuita, la cual fue creada y es mantenida por la Universidad de Maryland. La búsqueda de documentos se hace en Google, es decir, compara con los millones de webs que indexa el buscador. Entre las desventajas que presenta se puede mencionar que no determina el porcentaje de similitud respecto a los sitios sobre los cuales encontró coincidencias; además no determina el tipo de copia realizada (parcial o total). Solo puede usarse desde computadoras con acceso a Internet. (4)

**Article Checker:** Es una herramienta gratuita, la cual analiza los posibles casos de copia entre los textos introducidos en la propia aplicación y las fuentes de documentos en Internet. Entre las desventajas que presenta se puede mencionar que no determina el porcentaje de similitud, así como el tipo de copia realizada. Además muestra como resultado la cantidad de artículos con los cuales tuvo similitud, pero no visualiza el nombre o dirección donde se pueden encontrar dichos artículos. Solo puede usarse desde computadoras con acceso a Internet. (5)

**Turnitin:** Es una herramienta que se puede usar desde cualquier navegador, la cual se integra en las plataformas de aprendizaje: Blackboard, WebCT, Angel o Moodle. Este sistema analiza los posibles casos de copia entre las fuentes de documentos de Internet y los documentos de referencia que se encuentran almacenados en una base de datos interna. Tiene la desventaja de que es una herramienta de software privado por lo que hay que pagar una licencia o su uso en la web. (6)

**Sistema para la detección del plagio en los documentos científicos emitidos en el Centro de Información Científico Técnica del Instituto Superior Minero Metalúrgico de Moa (ISMMM):** Es una herramienta gratuita que se ejecuta en los sistemas operativos Linux y Windows XP. El proceso de la evaluación de la originalidad en los documentos de textos se lleva a cabo entre un documento que se carga en el propio sistema con los restantes que se encuentran en la base de datos, buscando similitudes entre ellos y exportando los resultados encontrados en un documento en formato .pdf. Este sistema tiene la desventaja que no analiza los documentos en formato .docx y .odt (solo en formato .doc). Por otra parte no muestra el porcentaje de similitud del texto copiado, así como no detecta si la copia realizada fue parcial o íntegra, por lo que la solución que ofrece este sistema no es la más adecuada para su uso en la UCI. (7)

A continuación se muestra una comparación entre los software analizados anteriormente:

<b>Software</b> <b>Indicadores</b>	<b>Article Checker</b>	<b>The Plagiarism Checker</b>	<b>Turnitin</b>	<b>Sistema de detección del plagio en el ISMMM</b>
Gratuito	x	x		x
Acceso a Internet.	x	x	x	
Porcentaje de similitud.			x	
Tipo de copia.			x	
Comparación en Internet.	x	x		
Comparación con una base de datos centralizada.			x	x

**Tabla 1. Comparación entre los software que evalúan la originalidad en documentos.**

Estos software no son los más apropiados para su utilización en la UCI ya que, de manera general, son privativos; además de que en su mayoría no siempre están disponibles en Internet. Por otra parte son herramientas que comparan textos introducidos en un momento determinado haciendo comparaciones en Internet, pues no son sistemas centralizados sobre una base de datos, por lo que solo evalúan el grado de originalidad de un documento en Internet, no de

manera local en una universidad determinada. Además no detallan el porcentaje de similitud del texto copiado, así como no determinan el tipo de copia realizada respecto al texto original (copia parcial o íntegra). Solo pueden usarse desde computadoras con acceso a Internet.

### **1.1.2 Procedimientos asociados con la determinación del grado de originalidad en los documentos de textos**

Entre los procedimientos asociados con la determinación del grado de originalidad en los documentos de textos se encuentran:

- ✓ Determinación del grado de originalidad en documentos con un corpus de referencia.

En este caso se realiza la comparación del documento sospechoso de plagio con un conjunto de documentos de referencias que se encuentran en la base de datos, buscando coincidencias entre los mismos. Este esquema es muy utilizado, pero tiene la desventaja de que si la fuente de la cual se plagió no se encuentra en la base de datos de documentos de referencias, el plagio no será detectado, por lo que la base de datos va a ser un poco extensa. (8)

- ✓ Determinación del grado de originalidad en documentos sin corpus de referencias.

En este caso no es necesario tener una base de datos con documentos de referencias, ya que para determinar el grado de originalidad se basa en analizar exclusivamente el documento sospechoso de plagio, pues su objetivo es detectar cambios bruscos en el estilo o estructura de dicho documento. Para proceder de esta manera se procesa el texto analizando la sintaxis, así como la estructura de las frases y párrafos que contiene el documento, analizando para ello su longitud, la cantidad de signos de interrogación y de exclamación. (8)

Para llevar a cabo el desarrollo de la solución propuesta se tendrá en cuenta el procedimiento para la determinación del plagio con corpus de referencia, mediante comparación de documentos.

### **1.1.3 Características asociadas a los sistemas que evalúan la originalidad en los documentos de textos**

Un sistema que evalúe la originalidad en los documentos de textos debe tener ciertas características, independientemente del tipo de plagio que detecte (10):

- ✓ No debe ser sensible a los espacios en blanco, signos de puntuación o mayúsculas: Cuando se realiza una comparación entre dos fragmentos de determinados textos, esta no debe afectarse por la existencia de espacios en blanco extras, signos de puntuación o letras en mayúsculas.
- ✓ Supresión del ruido: los pasajes considerados deben tener un cierto tamaño, de forma tal de estar seguros de que se trata de un plagio y descartar coincidencias producidas por expresiones idiomáticas comunes. Se deben eliminar aquellas palabras y caracteres que no contengan información relevante.
- ✓ Independencia de la posición: La permutación, borrado o agregado de contenido (de cierto tamaño, párrafos o pasajes más extensos) no debe afectar el proceso de determinación del grado de originalidad en los documentos.

#### **1.1.4 Algoritmos para la comparación de textos**

- ✓ **Greedy String Tiling (conocido como Mosaico Cadena):** Es un algoritmo Greedy (glotón) el cual basa su funcionamiento en el cálculo de subsecuencias comunes entre documentos. Entre las desventajas que presenta se puede citar que solo detecta copias exactas por lo que la permutación, borrado o agregado de contenido en un determinado documento afectaría el proceso de comparación de textos en busca de similitudes. Por otra parte su funcionamiento es lento y requiere gran espacio en memoria para almacenar las estructuras de datos usadas para la determinación del grado de originalidad, por lo que se ve afectado cuando el corpus de referencias de documentos es muy extenso. Este algoritmo no constituye el más adecuado para su implementación debido a las desventajas que posee. (9)
- ✓ **FingerPrint (conocido como Huella Digital):** Es un algoritmo rápido, el cual requiere poco espacio de almacenamiento. Este basa su funcionamiento en la determinación de un patrón que va a representar el contenido del documento para la búsqueda del mismo en una base de datos de patrones y detectar posibles similitudes. Es un algoritmo susceptible a pequeños

cambios en la estructura de las oraciones de un determinado texto, por lo que el cambio de un carácter por otro, el empleo de sinónimos, o el agregado o borrado de texto afectaría al proceso de la determinación del grado de originalidad en los documentos. (9)

- ✓ **Vector Space Model (VSM):** Es un algoritmo rápido el cual basa su funcionamiento en la representación del contenido de los documentos en términos de vectores y posteriormente mediante fórmulas matemáticas se arrojan los resultados de las similitudes. Dentro de este algoritmo se encuentra la variante: Similitud coseno. (9)

**Similitud coseno:** La medida de similitud coseno es un algoritmo VSM ampliamente utilizado en los sistemas que evalúan el grado de originalidad en determinados documentos, donde cada documento es representado como un vector de n-dimensiones en el espacio. Cada dimensión del vector representa un término de todo el vocabulario que provee la colección de documentos. Cada documento  $d_j$  es representado como un vector de pesos de n-dimensiones, siendo  $d_j = \{w_{1j}, w_{2j}, \dots, w_{nj}\}$  donde  $w_{ij}$  es el peso para el término  $t_k$  (término perteneciente al diccionario que se obtiene de todos los documentos) en el documento  $d_j$ . (9)

Con el fin de obtener valores de pesos en el intervalo [0,1] y los documentos sean representados por vectores de igual longitud, se determina el peso normalmente a partir de la normalización del coseno (9):

$$w_{k,j} = \frac{TF - IDF(t_{k,d_j})}{\sqrt{\sum_{s=1}^{|T|} (TF - IDF(t_{s,d_s}))^2}}$$

Donde:

- ✓ TF (por sus siglas en inglés Term Frequency) se determina como:

$$TF(t_k, d_j) = \frac{f_{k,j}}{\max_z f_{z,j}}$$

Siendo  $f_{k,j}$  la frecuencia del término  $k$  en el documento  $j$  y la expresión  $\max_z f_{z,j}$  representa el término de mayor frecuencia sobre todos los términos del vocabulario que aparecen en el documento  $d_j$ .

- ✓ IDF (por sus siglas en inglés Inverse Document Frequency) se determina como:

$$IDF(t_k) = \log \frac{N}{n_k}$$

Siendo  $N$  el número total de documentos y  $n_k$  el número de documentos donde el término  $t_k$  ocurre al menos una vez.

De manera general se puede decir entonces que:

$$TF - IDF = TF(t_k, d_j) \cdot IDF(t_k)$$

Luego se determina la cercanía entre los documentos a través de la similitud del coseno donde a menor distancia entre los vectores, mayor similitud existe entre los documentos:

$$sim(d_i, d_j) = \frac{\sum_k w_{k,i} * w_{k,j}}{\sqrt{\sum_k w_{k,i}^2} * \sqrt{\sum_k w_{k,j}^2}}$$

Como algoritmo para la comparación de textos se seleccionó este último debido a que cumple con las tres características fundamentales que debe tener un sistema para la evaluación de la originalidad en los documentos, permitiendo de esta manera que los usuarios realicen la búsqueda de documentos de manera rápida utilizando los índices invertidos, ya que las consultas sobre el índice se realizan por cada término del texto sospechoso, obteniendo así las posibles formas de aparición de estos textos en el conjunto de documentos de referencias.

## 1.2 Metodología de desarrollo

Todo desarrollo de software es difícil de controlar, por lo que es necesario tener en cuenta una metodología de desarrollo que nos ayude a estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

Para el desarrollo de la investigación, se necesita una metodología para equipos de trabajo pequeños (menos de 10 integrantes), que sea adaptable, configurable, de poca documentación, que se realicen pruebas constantemente para que se pueda obtener en poco tiempo productos funcionales, en aras de satisfacer sus necesidades y que permita que estos productos sean fáciles y simples de modificar, por lo que se propone el uso de la metodología ágil y no la

tradicional. Dentro de las metodologías ágiles más utilizadas se encuentran: XP (Extreme Programming).

### **Extreme Programming (XP)**

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo.

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo y equipos de trabajo pequeño (menos de 10 integrantes). La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. El ciclo de trabajo de esta metodología consta de 4 fases (11):

- ✓ Planificación: Es la fase donde se crean las Historias de Usuario (HU), el Plan de iteraciones y el Plan de entrega.
- ✓ Diseño: En esta fase se crean las Tarjetas CRC las cuales definen una clase expresando sus funcionalidades y las otras clases con las que colabora.
- ✓ Desarrollo: Es donde se definen las tareas de ingeniería para que los desarrolladores tengan una guía para implementar todas las HU.
- ✓ Pruebas: En esta fase se realizan las pruebas de aceptación a cada una de las HU para probar que cumplen con las funcionalidades que desea el cliente.

Algunas de las características que presenta XP son las siguientes (11):

- ✓ Desarrollo iterativo e incremental: Se logran pequeñas mejoras consecutivamente.
- ✓ Pruebas unitarias continuas: Son las pruebas que se realizan frecuentemente a los principales procesos para obtener en un futuro las fallas que pudieran ocurrir, es decir, los posibles errores. La calidad de los sistemas basados en esta metodología tiende a ser superior respecto a otros, debido a la gran cantidad de pruebas que se realizan durante todo el proyecto.



- ✓ Corrección de todos los errores: Se realiza antes de añadir nuevas funcionalidades.
- ✓ Simplicidad en el código: Es la mejor forma de que las cosas funcionen correctamente garantizando así la opción de añadir alguna nueva funcionalidad si fuera necesario.
- ✓ Refactorización del código: Consiste en reescribir algunas partes del código donde su comprensión no sea clara, para aumentar su legibilidad y mantenibilidad sin modificar su comportamiento.
- ✓ Programación en pares: Se basa en que un desarrollador que participe en un proyecto en una misma estación de trabajo que otro desarrollador lleve a cabo la acción que el otro no está haciendo es ese momento.

Lo fundamental en este tipo de metodología es la comunicación entre los usuarios y los desarrolladores, la simplicidad al desarrollar y codificar los módulos del sistema y la retroalimentación concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

La metodología XP tiene principalmente dos objetivos: lograr una satisfacción con el cliente tratando de darle un software que verdaderamente necesita y en el momento que lo requiera, además de potenciar al máximo el trabajo en grupo identificando a cada miembro del equipo de desarrollo con el producto entregable al cliente (11).

### **1.3 Lenguaje de programación**

Los lenguajes de programación constituyen idiomas artificiales para expresar procesos llevados a cabo por la computadora, los cuales son usados para crear programas que lleven a cabo el control físico y lógico de una máquina, así como también para expresar algoritmos con precisión o como modo de comunicación humana.

#### **PHP 5.3**

PHP (PHP siglas de Hypertext Preprocessor en inglés) es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde

una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica (12).

Es un lenguaje multiplataforma, el cual tiene capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destacando su conectividad con el gestor MySQL. Además posee capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones). Todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. Además permite las técnicas de Programación Orientada a Objetos y tiene una biblioteca nativa de funciones sumamente amplia e incluida. No requiere definición de tipos de variables y tiene manejo de excepciones.

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun estando dirigido alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

Entre las principales ventajas que presenta PHP se encuentra (12):

- ✓ Rendimiento: El motor de PHP fue rediseñado completamente con un administrador de memoria optimizado para mejorar el rendimiento.
- ✓ Portabilidad: Está disponible para sistemas operativos como UNIX, Windows, Mac OS y OS/2. Los programas escritos en este lenguaje son portables entre plataformas.
- ✓ Facilidad de uso: Su sintaxis es clara y consistente, cuenta además con documentación exhaustiva para todas las funciones de su núcleo.
- ✓ Código abierto: El lenguaje es desarrollado por una comunidad de programadores voluntarios que publican su código libremente en la Web y puede ser usado sin el pago de licencias o inversiones en hardware costoso. Esto reduce el costo de la producción del software sin afectar la flexibilidad o confiabilidad.
- ✓ Contiene librerías para manejar imágenes y archivos en formato pdf, lo que en otros lenguajes es necesario la instalación de componentes adicionales.

#### **1.4 Plataforma de desarrollo**

Para el desarrollo de la aplicación además de un potente lenguaje de programación, es necesario un potente IDE (IDE siglas de Integrated Development Environment en inglés) de desarrollo.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDE pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios. (13)

#### **NetBeans IDE 7.3**

NetBeans es un proyecto de código abierto para desarrolladores de software. Puede obtener todas las herramientas que necesite para crear aplicaciones profesionales para el escritorio, la empresa, la web y equipos móviles con el lenguaje Java, C/C++, y Ruby. Es fácil de instalar y de uso instantáneo y se ejecuta en varias plataformas incluyendo Windows, Linux y Mac OS X y Solaris. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. (14)

Para el desarrollo de la herramienta para la evaluación de la originalidad en los documentos de textos se determinó utilizar NetBeans 7.3 como plataforma de desarrollo, por ser de código abierto y poseer funcionalidades que son extensibles mediante la instalación de paquetes, además de tener un potente motor para la conexión a la base de datos. Por otra parte, el resaltado de líneas, el completamiento de código y la adaptabilidad hacia los estándares de codificación definidos es más usable desde NetBeans que desde Eclipse.

#### **1.5 Herramienta CASE**

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas que tienen como objetivo aumentar la productividad en el desarrollo de software, reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software, en tareas como: el proceso de realizar un diseño del proyecto, cálculo de costes, implementación automática de parte del código con el diseño dado, compilación automática, documentación o detección de errores entre otras. (15)

### **Visual Paradigm 8.0**

Es una herramienta CASE profesional que soporta el ciclo de vida de un proyecto, proporcionando un conjunto de herramientas necesarias para realizar diferentes tareas como son: la captura de requisitos, modelado de clases, modelado de datos, entre otros. Entre las principales características que soporta se encuentran las siguientes (16):

- ✓ Es un software multiplataforma (Windows, Linux y Mac).
- ✓ Exporta los diagramas y diseños en formatos: JPEG, PDF, HTML y Microsoft Word.
- ✓ Permite generar código en lenguaje PHP, C++, CORBA, IDL, Java, XML Schema, Ada, Python, C#, VB .NET, ODL, Flash Action Script, Delphi, Perl y Rugby.
- ✓ Permite la generación de base de datos, es decir, convierte los diagramas de entidad-relación a tablas de base de datos, mapeo de objetos y relaciones.

Por todas las características antes expuestas se ha seleccionado la herramienta Visual Paradigm 8.0 para realizar el diseño de los diagramas necesarios durante el desarrollo del sistema.

### **1.6 Herramienta para el modelado de prototipos web.**

#### **Axure 5.5**

Es una herramienta que permite crear prototipos amigables, con calidad visual, la cual brinda un conjunto de librerías y la posibilidad de crearlas nuevas para satisfacer las necesidades del modelado. Permite generar los diseños de prototipos a HTML permitiendo la interacción de cualquier miembro del equipo de desarrollo o las partes interesadas con estos sin necesidad de

instalar Axure; de esta forma se puede comprobar si el sistema de navegación de la aplicación web es el deseado y el cliente podrá tener una idea clara y precisa de cómo será la aplicación en un futuro. Se pueden crear fácilmente elementos de diseño reutilizables como encabezados, pies de página y plantillas. (17)

### **1.7 Gestor de base de datos**

Es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Es un software que permite introducir, organizar y recuperar la información de las bases de datos, en general, administrarlas. (17)

Una base de datos se puede definir como un conjunto de información relacionada o conjuntos de datos relacionados entre sí, que se encuentran agrupados o estructurados. Es un sistema formado por un conjunto de datos almacenados en discos duros que permiten el acceso directo a ellos y un conjunto de programas que manipulen dichos datos. (17)

### **MySQL 5.5**

Es un sistema gestor de bases de datos relacionales, multiplataforma y multiusuario. Tiene un diseño multihilos lo que le permite soportar una gran carga de forma muy eficiente. Presenta gran rapidez así como facilidad de uso, es de fácil configuración e instalación y la velocidad operacional es baja. Consume poco de memoria y es poco probable que corrompa los datos. El objetivo que persigue MySQL es cumplir con el estándar SQL pero sin sacrificar velocidad, fiabilidad o usabilidad. Su propósito general es lograr mayor velocidad y robustez, así como garantizar que cada base de datos cuente con tres archivos: uno de estructura, uno de datos y otro de índice. Además soporta mensajes de error en distintos idiomas. (17)

Entre las características que presenta se encuentran las que se mencionan a continuación (17):

- ✓ Disponibilidad en gran cantidad de plataformas y sistemas.
- ✓ Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica, transacciones, entre otras.
- ✓ Conectividad segura.

Teniendo en cuenta las características potenciales del gestor MySQL, estabilidad y rendimiento se selecciona, además de que se requiere de almacenamiento y procesamiento de grandes volúmenes de información así como la estabilidad en dichas operaciones.

### **1.8 Servidor Web**

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales, síncronas o asíncronas con el cliente, generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. Son los que garantizan los servicios que muchas empresas brindan a todo el mundo, además de que permiten la interacción de muchas personas en la red. (19)

#### **Apache Tomcat 7.0.23**

Apache Tomcat es uno de los servidores web más utilizados actualmente para el despliegue de las aplicaciones desarrolladas por la UCI, porque es gratis y además fácil de instalar. Además se ejecuta en máquinas más pequeñas y es compatible con las API más recientes de PHP. Por otra parte es muy fiable por la solidez que le brinda a los desarrolladores que contribuyen a su estabilidad.

Apache Tomcat es el servidor web más utilizado a la hora de trabajar con PHP en entornos web; dispone de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la Apache Software Licence 2.0. Tomcat es más rápido que sus homólogos contenedores de servlet: JBoss y Glass Fish, es muy conveniente ya que se integra fácilmente con el IDE NetBeans, dando la oportunidad de depurar aplicaciones que están corriendo en el servidor, así como desplegar y replegar aplicaciones hacia y desde el servidor respectivamente. Es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. (18)

### **1.9 Sistema Gestor de Contenido**

Un Sistema Gestor de Contenidos (CMS, por sus siglas en inglés Content Management System) constituye un programa empleado para la gestión y publicación de un sitio web dinámico con actualizaciones periódicas, uniforme y diseño centrado en el usuario. (19)

Los CMS permiten la interacción y colaboración entre los usuarios con el uso de herramientas como foros, blogs, comentarios a contenidos o chats, que pueden estar instalados en el núcleo del CMS o ser incorporados posteriormente. Cuenta con una gran comunidad de desarrolladores que trabajan constantemente en el mantenimiento y extensión de sus funcionalidades.

El principal objetivo de los CMS es proveer al desarrollador de una herramienta para la construcción de aplicaciones web que manipulen contenidos de forma dinámica, minimizando la necesidad de conocimientos técnicos en cuanto a determinados lenguajes de programación.

### **Drupal 7**

Drupal es un CMS modular, multipropósito y configurable que permite publicar artículos, imágenes u otros archivos, además de los servicios añadidos como: foros, encuestas, votaciones, blogs y administración de usuarios y permisos. Es un sistema dinámico, pues almacena el contenido textual de las páginas y otras configuraciones en una base de datos y no en los archivos del sistema de ficheros del servidor. Es un CMS basado en el lenguaje de programación PHP y base de datos MySQL. Está maquettato con hojas de estilo CSS, con los cuales es posible construir sitios web totalmente accesibles. (20)

Entre las características que presenta se encuentran las siguientes (20):

- ✓ Los usuarios pueden acceder a la sesión de forma local o utilizando sistemas como un directorio activo.
- ✓ No es necesario que los administradores creen permisos a cada usuario. Drupal brinda la posibilidad de agrupar a los usuarios por roles y el administrador puede establecer los permisos por roles.
- ✓ Posibilita el seguimiento de las actualizaciones de los contenidos. Ofrece la opción de hacer comentarios sobre los cambios realizados, como también permite recuperar la versión anterior desechando los cambios.

- ✓ Posee aspectos importantes como el diseño de contenidos, pues en él se pueden crear y configurar los contenidos que el usuario desee, agregando a cada uno de ellos la información que estime conveniente, incluyendo la validación.

Teniendo en cuenta las características de la aplicación a desarrollar, la versatilidad, seguridad, extensibilidad para aplicaciones informáticas escalables, se utiliza como CMS Drupal 7 para implementar las funcionalidades de la herramienta para la evaluación de la originalidad en documentos de textos.

### **1.10 Conclusiones parciales**

En este capítulo se expusieron los conceptos necesarios para la comprensión de la solución informática que da origen a esta investigación. También se realizó un estudio sobre la determinación del grado de originalidad en los documentos de textos, demostrando la necesidad de contar con una herramienta que evalúe el grado de originalidad en los documentos de textos digitales emitidos en la esfera de investigación en la UCI. Se realizó un estudio de las tecnologías, herramientas y tendencias del desarrollo de software actuales, exponiendo las principales características y ventajas, lo que ayudó a seleccionar las herramientas para el desarrollo del sistema, teniendo como resultado:

- ✓ **Extreme Programming:** como metodología de desarrollo.
- ✓ **Visual Paradigm 8.0:** como herramienta CASE.
- ✓ **MySQL 5.5:** como servidor de Base Datos.
- ✓ **NetBeans 7.3:** como entorno de desarrollo.
- ✓ **PHP 5.3:** como lenguaje de programación.
- ✓ **Drupal 7:** como Sistema Gestor de Contenido.
- ✓ **Apache Tomcat 7.0.23:** como Servidor Web.
- ✓ **Axure 5.5:** como herramienta para modelado de prototipos web.



## **CAPÍTULO 2: PLANIFICACIÓN, DISEÑO Y DESARROLLO**

En este capítulo se justifica la obtención de un conjunto de artefactos de la metodología XP en su aplicación al desarrollo del sistema informático, para las fases: Planificación, Diseño y Desarrollo, donde se documentan los elementos significativos para que se obtenga como resultado un producto con la calidad requerida por el cliente.

### **2.1 Descripción del sistema**

El sistema propuesto se basa en una herramienta para determinar el grado de originalidad en los documentos de textos digitales emitidos en la UCI en la esfera de investigación, ya sean documentos de tesis o los documentos generados en eventos como: Jornada Científica Estudiantil, Fórum de Historia y Evento Juvenil Martiano. Para realizar la comparación entre documentos, se tendrá una base de datos la cual contendrá los documentos evaluados como original, al poseer un porcentaje de similitud bajo respecto a otros.

Para acceder al sistema se han definido los siguientes roles:

- ✓ Administrador: Podrá realizar cualquier operación en el sistema. Es el único usuario que tiene otorgado todos los privilegios. Es el encargado de gestionar los permisos a los roles que serán asignados a los usuarios.
- ✓ Profesores y estudiantes: Podrán acceder a la aplicación y determinar el grado de originalidad en los documentos de textos, además de visualizar los resultados obtenidos durante este proceso. No están autorizados a eliminar los documentos de referencia que se encuentran en la base de datos.

### **2.2 Planificación**

La metodología de desarrollo de software XP muestra la planificación como un permanente diálogo entre el usuario final y el equipo de desarrollo, definiéndose lo que el software tiene que resolver para que genere algún valor. [27]

En esta fase el cliente establece la prioridad de cada HU y los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Además se identifica el número y tamaño de las iteraciones, al igual que se plantean ajustes necesarios a la metodología según las características del proyecto. [28]

### **2.2.1 Requisitos del sistema**

Los requisitos funcionales son capacidades o cualidades que el sistema debe tener para cubrir las necesidades de los usuarios. En cambio, los requisitos no funcionales van a constituir las propiedades o cualidades que el producto debe tener para que llegue a ser atractivo, confiable, usable y rápido. (24)

De acuerdo a las clasificaciones y descripciones antes mencionadas se han definido un conjunto de requisitos.

#### **Requisitos funcionales del sistema**

##### **HU: Autenticar usuario.**

RF1: Autenticar usuario.

##### **HU: Cargar documento en el sistema.**

RF2: Cargar documento en el sistema.

##### **HU: Analizar documentos.**

RF3: Analizar documentos.

##### **HU: Mostrar resultados de la evaluación.**

RF4: Mostrar el tipo de copia realizada (copia parcial o íntegra).

RF5: Mostrar el porcentaje de la copia que se realizó.

RF6: Mostrar el nombre del documento con el cual tuvo similitud.

##### **HU: Guardar documento.**

RF7: Guardar los datos del documento.

RF8: Guardar documento.

**HU: Generar reporte.**

RF9: Generar reporte con los resultados de la evaluación.

**HU: Gestionar documento.**

RF10: Insertar documento.

RF11: Eliminar documento.

RF12: Buscar documento.

### **Requisitos no funcionales del sistema**

#### **Requisitos de Software:**

- ✓ Para la utilización del sistema se requerirá Internet Explorer 6 o superior, o Firefox Mozilla 1.5 o superior.

#### **Requisitos de Hardware:**

- ✓ Las computadoras de los clientes solo requerirán memoria RAM de 512MB.
- ✓ El servidor deberá contar con una memoria RAM de 1GB o más.
- ✓ El servidor deberá contar un disco duro de 320GB o más.

#### **Requisitos en el diseño:**

- ✓ El sistema tiene que ofrecer una interfaz amigable, fácil de operar.
- ✓ Las interfaces tienen que ostentar un diseño sencillo, con pocas entradas, permitiendo un balance adecuado entre funcionalidad y simplicidad de tal manera que no se haga difícil para los usuarios la utilización del sistema.
- ✓ Los campos obligatorios serán señalados con un asterisco (\*).
- ✓ El sistema mostrará el nombre del usuario que está autenticado en el sistema en un momento determinado.

- ✓ El idioma de todas las interfaces de la aplicación será el español.

**Integridad:**

- ✓ El administrador es el único que tiene el control total del sistema.

**Disponibilidad:**

- ✓ El sistema podrá ser usado en cualquier momento por todos los usuarios autorizados.

**2.2.2 Historias de usuario**

Las HU constituyen una representación de un requisito funcional de software. Representan una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar un gran cúmulo de documentos y sin requerir de mucho tiempo para administrarlos. (22)

Las HU deben ser:

- ✓ Valoradas por los clientes o usuarios: Los intereses de los clientes y de los usuarios no siempre coinciden, pero en todo caso, cada HU debe ser importante para alguno de ellos más que para el desarrollador.
- ✓ Estimables: Un resultado de la discusión de una HU es la estimación del tiempo que tomará completarla. Esto permite estimar el tiempo total del proyecto.
- ✓ Pequeñas: Las HU muy largas son difíciles de estimar e imponen restricciones sobre la planificación de un desarrollo iterativo. Generalmente se recomienda la consolidación de HU muy cortas en una sola HU.
- ✓ Verificables: Las HU cubren requisitos funcionales, por lo que generalmente son verificables. Cuando sea posible, la verificación debe automatizarse, de manera que pueda ser verificada en cada entrega del proyecto.

Durante la fase de planificación se identificaron 7 HU, representando cada una las funcionalidades del sistema. A continuación se muestra la descripción de las HU identificadas.

<b>Historia de Usuario</b>	
<b>Número:</b> 1	<b>Usuario:</b> Todos los usuarios.

<b>Nombre de la HU:</b> Autenticar usuario.	
<b>Prioridad en negocio:</b> Alta.	<b>Riesgo en desarrollo:</b> Baja.
<b>Puntos estimados:</b> 1 semana.	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Martha Cardonne Fong.	
<b>Descripción:</b> Para acceder a las funcionalidades que brinda el sistema, el usuario debe especificar el nombre de usuario y contraseña del dominio UCI. En caso contrario el sistema debe mostrar un mensaje de error y notificarle al usuario.	

**Tabla 2. HU1: Autenticar usuario.**

<b>Historia de Usuario</b>	
<b>Número:</b> 2.	<b>Usuario:</b> Todos los usuarios.
<b>Nombre de la HU:</b> Cargar documento en el sistema	
<b>Prioridad en negocio:</b> Alta.	<b>Riesgo en desarrollo:</b> Alto.
<b>Puntos estimados:</b> 1 semanas.	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Martha Cardonne Fong.	
<b>Descripción:</b> El sistema debe permitir al usuario cargar un documento para ser analizado con posterioridad. Se debe mostrar un mensaje de error en caso de que no se halla cargado un documento o se halla subido cualquier archivo fuera del formato especificado (.doc, .docx, .odt). Se le debe dar la posibilidad al usuario de volver a realizar esta operación en caso de no haber podido realizar esta operación.	

**Tabla 3. HU2: Cargar documento en el sistema.**

<b>Historia de Usuario</b>	
<b>Número:</b> 3	<b>Usuario:</b> Todos los usuarios.
<b>Nombre de la HU:</b> Analizar documentos.	
<b>Prioridad en negocio:</b> Alta.	<b>Riesgo en desarrollo:</b> Alta.

<b>Puntos estimados:</b> 3 semanas.	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Ariel Tomás Sánchez Pérez.	
<b>Descripción:</b> El sistema debe permitir al usuario determinar el grado de originalidad de un documento cargado en el sistema. Para ello busca similitudes entre dicho documento y los restantes que se encuentran en la base de datos.	

**Tabla 4. HU3: Analizar documentos.**

<b>Historia de Usuario</b>	
<b>Número:</b> 4	<b>Usuario:</b> Todos los usuarios.
<b>Nombre de la HU:</b> Mostrar resultados de la evaluación.	
<b>Prioridad en negocio:</b> Media.	<b>Riesgo en desarrollo:</b> Baja.
<b>Puntos estimados:</b> 2 semanas.	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Martha Cardonne Fong.	
<b>Descripción:</b> El sistema debe mostrar al usuario el nombre de los documentos con el cual tuvo similitud el documento cargado en el sistema. Además debe mostrar el porcentaje de similitud asociado a cada documento así como el tipo de copia realizada (parcial o total).	

**Tabla 5. HU4: Mostrar resultados de la evaluación.**

<b>Historia de Usuario</b>	
<b>Número:</b> 5	<b>Usuario:</b> Todos los usuarios.
<b>Nombre de la HU:</b> Guardar documento.	
<b>Prioridad en negocio:</b> Alta.	<b>Riesgo en desarrollo:</b> Alta.
<b>Puntos estimados:</b> 2 semanas.	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Martha Cardonne Fong.	

**Descripción:** El sistema debe permitir al usuario guardar los datos correspondientes del documento cargado en el sistema, en caso de que haya sido evaluado como original. Además debe permitir guardar el documento en la base de datos para futuras comparaciones.

**Tabla 6. HU5: Guardar documento.**

<b>Historia de Usuario</b>	
<b>Número:</b> 6	<b>Usuario:</b> Todos los usuarios.
<b>Nombre de la HU:</b> Generar reporte.	
<b>Prioridad en negocio:</b> Media.	<b>Riesgo en desarrollo:</b> Baja.
<b>Puntos estimados:</b> 1 semanas.	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Ariel Tomás Sánchez Pérez.	
<b>Descripción:</b> El sistema debe permitir al usuario exportar los resultados de la evaluación de la originalidad en un documento en formato pdf.	

**Tabla 7. HU6: Generar reporte.**

<b>Historia de Usuario</b>	
<b>Número:</b> 7	<b>Usuario:</b> Administrador.
<b>Nombre de la HU:</b> Gestionar documento.	
<b>Prioridad en negocio:</b> Alta.	<b>Riesgo en desarrollo:</b> Media.
<b>Puntos estimados:</b> 2 semanas.	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Martha Cardonne Fong.	
<b>Descripción:</b> El usuario podrá adicionar, eliminar y buscar un documento en la base de datos en dependencia de los permisos otorgados a cada rol.	

**Tabla 8. HU7: Gestionar documento.**

### 2.2.3 Plan de iteraciones

Este plan incluye varias iteraciones sobre el sistema antes de ser entregado. Los elementos que deben tenerse en cuenta durante su elaboración son: HU no abordadas, pruebas de aceptación no superadas y tareas no terminadas en la iteración efectuada con anterioridad. Este plan define exactamente cuáles HU serán implementadas en cada iteración. Tomando como base cada una de las HU y el esfuerzo que se requiere para el desarrollo de estas, se procede a fragmentar el trabajo en dos iteraciones, obteniendo un trabajo incremental donde la piedra angular es la comunicación entre el equipo de desarrollo y el cliente. (23)

A continuación se describen cada una de las iteraciones propuestas, donde la duración total de iteraciones en días, se obtiene a partir del esfuerzo en días estimado por el desarrollador para implementar cada HU:

**Iteración # 1:** Se tuvieron en cuenta aquellas HU de mayor importancia en cuanto a la funcionalidad que describe cada una, es decir, aquellas HU con prioridad en negocio alta, entre las cuales se encuentran:

HU Autenticar usuario.

HU Gestionar documento.

HU Cargar documento en el sistema.

HU Analizar documentos.

HU Guardar documento.

**Iteración # 2:** Se tuvieron en cuenta aquellas funcionalidades del sistema con menos prioridad, es decir, aquellas HU identificadas con prioridad en negocio media o baja. Por otra parte en esta iteración se corregirán los errores encontrados en la iteración anterior, obteniéndose una nueva versión del sistema. Entre las HU a desarrollar en esta iteración se encuentran:

HU Mostrar resultados de la evaluación.

HU Generar reporte.

#### **2.3.4 Plan de entregas**



Una vez identificadas por el cliente las HU y confeccionado por los desarrolladores el plan de iteración, se procede a la confección del plan de entrega, con la intención de que los clientes obtengan una estimación real del tiempo que conllevará la implementación de las funcionalidades que describe cada HU.

A continuación se presenta el plan de entrega elaborado por el equipo de desarrollo donde se reflejan las fechas de entregas para las primeras versiones de las HU implementadas:

<b>Nro. de iteración.</b>	<b>Historia de usuario a implementar.</b>	<b>Fecha de entrega.</b>
1	Autenticar usuario. Gestionar documento. Cargar documento en el sistema. Analizar documentos. Guardar documento.	20 /04/ 2013
2	Mostrar resultados de la evaluación. Generar reporte.	20/05/2013

**Tabla 9. Plan de Entregas.**

## **2.3 Diseño**

La metodología de desarrollo de software XP sugiere que se realicen diseños sencillos para de esta forma lograr que sean de fácil entendimiento en la fase de desarrollo, lo que le costará menos tiempo al desarrollador llevar a cabo esta tarea. En esta fase el sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo, la cual describe como debería funcionar el sistema a través del diseño de las tarjetas CRC (Class, Responsibilities and Collaboration por sus siglas en inglés). (23)

### **2.3.1 Modelo del Dominio**

Cuando en el entorno del negocio los procesos tienen un bajo nivel de estructuración y no existen flujos de información interconectados y bien definidos, se genera la imposibilidad de llevar a cabo un Modelo de Negocio, haciéndose necesario la realización de un Modelo de Dominio. El Modelo de Dominio, permite mostrar los principales conceptos que se manejan para el dominio del sistema en desarrollo. El diagrama del Modelo de Dominio se presenta en forma de diagrama de clases donde figuran los principales conceptos y roles del sistema propuesto. (27)

La metodología XP no especifica que se realice el diseño del Modelo del Dominio entre los artefactos que generan durante el ciclo de vida que propone, pero para un mejor entendimiento del problema se llevó a cabo el diseño del mismo.

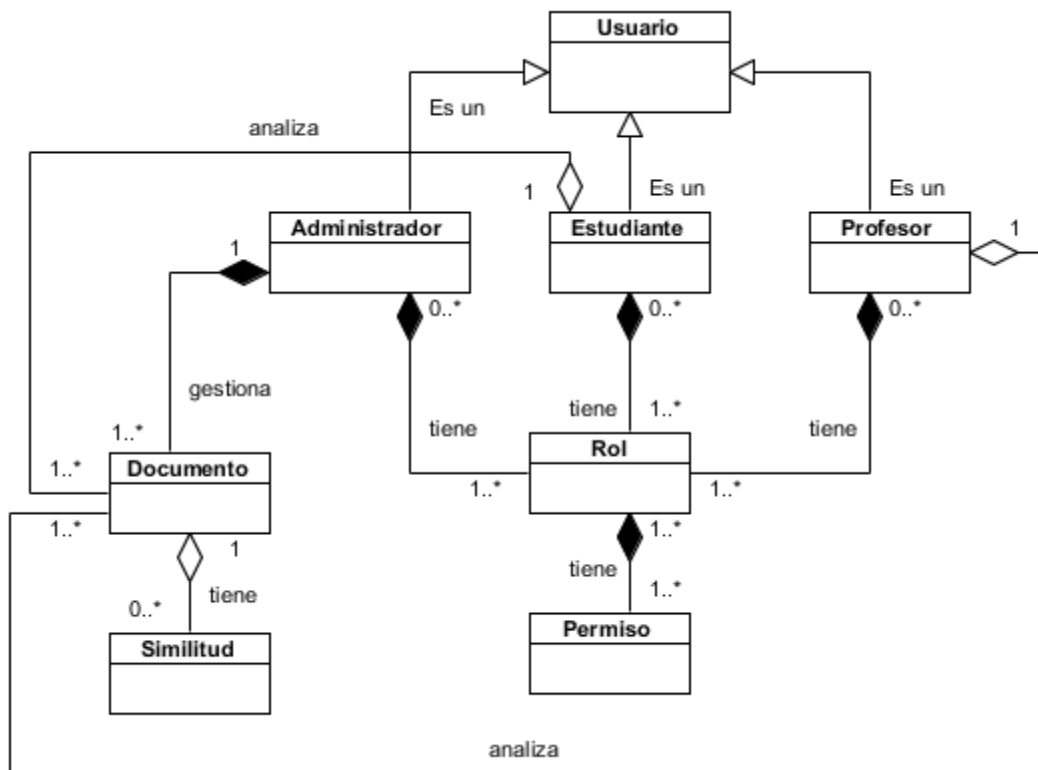


Figura 1. Modelo del Dominio.

### 2.3.2 Diagrama de clases

Un diagrama de clases del diseño representa las especificaciones de las clases en un sistema. En él se contemplan las clases, sus atributos, funcionalidades, las relaciones entre ellas,

navegabilidad, dependencia, entre otros elementos. Cada clase describe algún elemento del dominio del problema, con enfoque en los aspectos del problema visibles para el usuario o el cliente. (24)

A continuación se muestra el diagrama de clases del diseño, donde se especifica para cada clase los atributos que las componen y las principales funcionalidades que realizan:

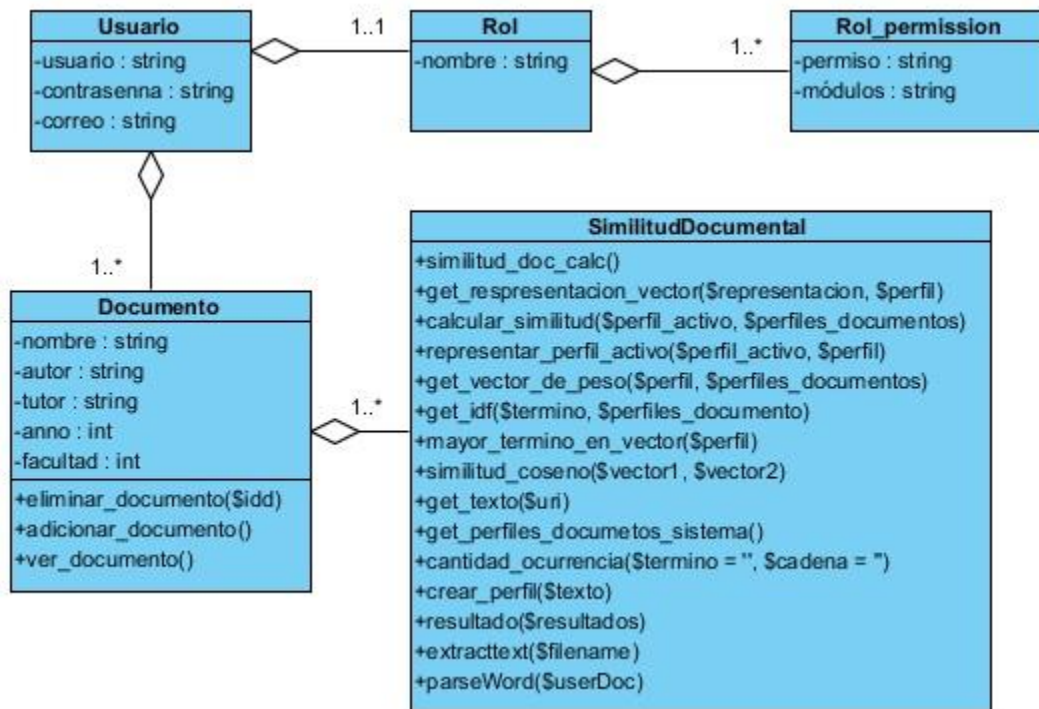


Figura 2. Diagrama de clases.

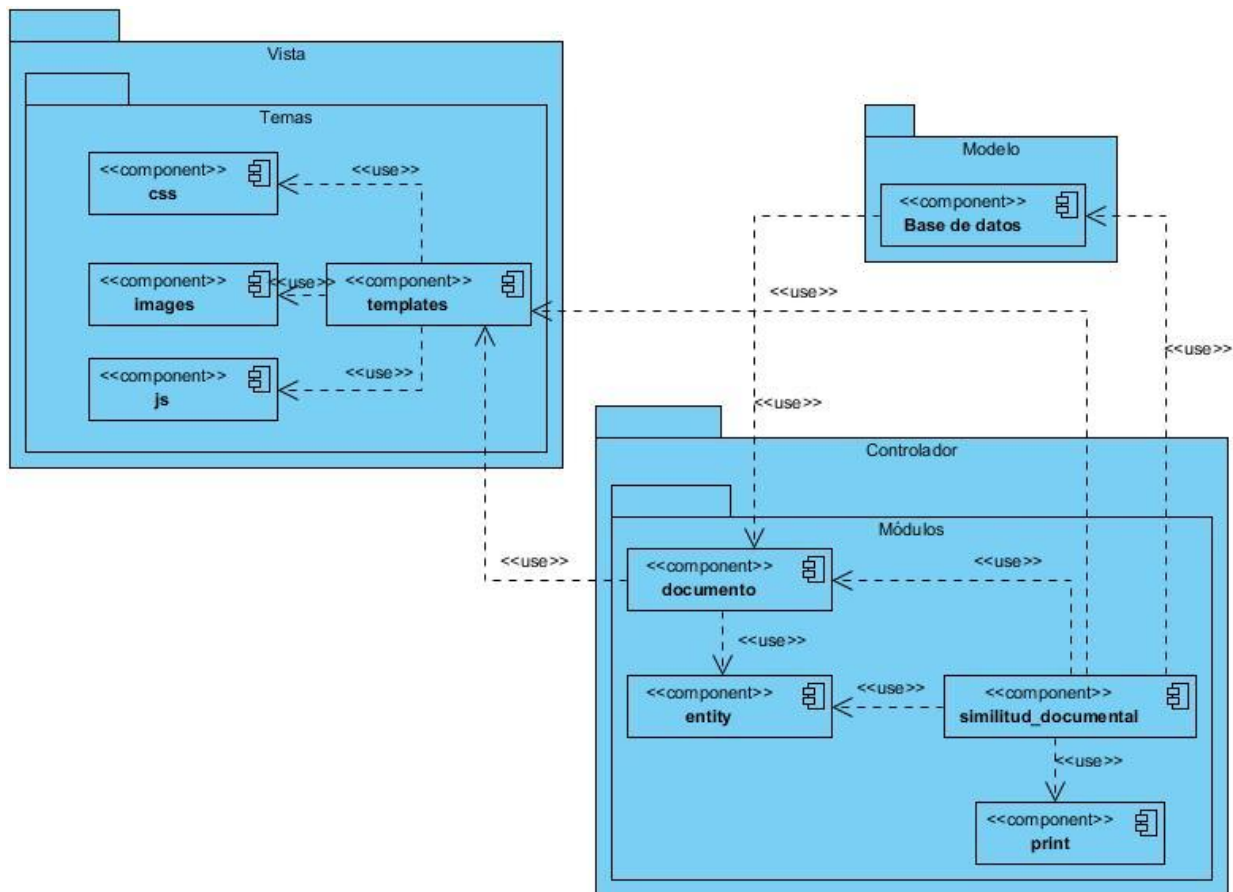
### 2.3.3 Patrones de diseño

El CMS empleado promueve las buenas prácticas del diseño y la implementación haciendo uso de patrones de diseño. A continuación se describen los patrones usados en el diseño de la herramienta para la evaluación de la originalidad en documentos de textos digitales.

#### Patrón arquitectónico

Se emplea como patrón arquitectónico el Modelo Vista Controlador (MVC) el cual divide la aplicación en tres partes lógicas: el modelo, la vista y el controlador. El modelo contiene toda la

información con la que opera la aplicación (Base de datos). Por su parte, la vista es la encargada de mostrarle la información al usuario a través de una página web que le permite al mismo interactuar con ella (temas de Drupal), mientras que el controlador se ocupa de procesar las interacciones del usuario y realizar los cambios apropiados en el modelo o en la vista (módulos de Drupal), sirviendo de intermediaria entre ambas partes.

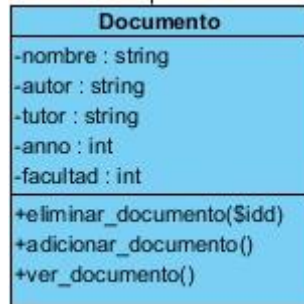


**Figura 3. Patrón arquitectónico Modelo Vista Controlador.**

### Patrones de diseño GRASP.

En el diseño del sistema se emplearon los patrones de asignación de responsabilidades o GRASP (Patrones Generales de Software para Asignar Responsabilidades), los cuales se describen a continuación.

- ✓ **Experto en información:** Se pone en práctica con el uso de clases que poseen responsabilidades específicas a cumplir, de acuerdo con la información que manejan. Este patrón es aplicado en la clase Documento la cual contiene la información necesaria para cumplir con las responsabilidades que le fueron asignadas.



**Figura 4. Clase experta en información referente a los documentos.**

- ✓ **Creador:** el patrón se evidencia en las clases controladoras que, para cada uno de los módulos o funcionalidades del sistema, son las encargadas de crear las instancias de los objetos que manejan, favoreciendo así la reutilización y el bajo acoplamiento.

```
SimilitudDocumental
+similitud_doc_calc()
+get_representacion_vector($representacion, $perfil)
+calcular_similitud($perfil_activo, $perfiles_documentos)
+representar_perfil_activo($perfil_activo, $perfil)
+get_vector_de_peso($perfil, $perfiles_documentos)
+get_idf($termino, $perfiles_documento)
+mayor_termino_en_vector($perfil)
+similitud_coseno($vector1, $vector2)
+get_texto($uri)
+get_perfiles_documentos_sistema()
+cantidad_ocurrencia($termino = "", $cadena = "")
+crear_perfil($texto)
+resultado($resultados)
+extracttext($filename)
+parseWord($userDoc)
+similitud_documental_permission()
+similitud_documental_menu()
+similitud_doc_formulario()
+similitud_doc_formulario_submit($form, $form_state)
+resultado_formulario($form, $form_state, $resultados)
+resultado_formulario_submit($form, $form_state)
+get_tabla_resultado($documentos)
```

Figura 5. Clase controladora del módulo SimilitudDocumental.

- ✓ **Alta Cohesión:** el patrón se evidencia debido a que a cada una de las clases se le asignaron responsabilidades de tal forma, que estén estrechamente relacionadas entre sí y no realicen un trabajo excesivo (ver diagrama de clases).
- ✓ **Bajo Acoplamiento:** se proporciona un bajo acoplamiento en el diseño debido a que las clases existentes tienen asignadas responsabilidades de tal forma que estas no dependan en gran medida de otras, permitiendo así tener sistemas más robustos y de fácil mantenimiento.

### Patrón de diseño en la base de datos.

En el diseño de la base de datos del sistema se empleó el siguiente patrón:

- ✓ **Control de Acceso Basado en Roles (RBAC, pos sus siglas en inglés):** este patrón se implementa mediante la asignación de roles a los diferentes usuarios del sistema, a través de los cuales dichos usuarios obtienen los permisos para realizar determinadas funciones en la

aplicación, lo que simplifica la gestión de los usuarios debido a que el manejo de los permisos se reduce solo a asignarle el rol apropiado a determinado usuario.

#### **2.3.4 Tarjetas CRC**

Las tarjetas CRC (Class, Responsibilities and Collaboration por sus siglas en inglés) sirven para diseñar el sistema en conjunto entre todo el equipo de desarrollo. Estas tarjetas representan objetos, para los cuales se especifican la clase a la que pertenece dicho objeto, las responsabilidades u objetivos que debe cumplir y las clases que colaboran con cada responsabilidad. (25)

A continuación se muestra el diseño de la tarjeta CRC asociada a la clase SimilitudDocumental.

<b>Responsabilidad</b>	<b>Clases relacionadas</b>
Extraer el texto en los documentos en formato .doc. Extraer el texto en los documentos en formato .docx. Extraer el texto en los documentos en formato .odt. Obtener cantidad de ocurrencia de una palabra en un documento. Crear el perfil del documento activo. Crear el perfil de los documentos en la base de datos. Representar los documentos en términos de vectores. Obtener el valor del IDF. Obtener el vector de peso. Obtener el mayor término en vector. Calcular el porcentaje de similitud. Mostrar el tipo de copia realizada.	Documento

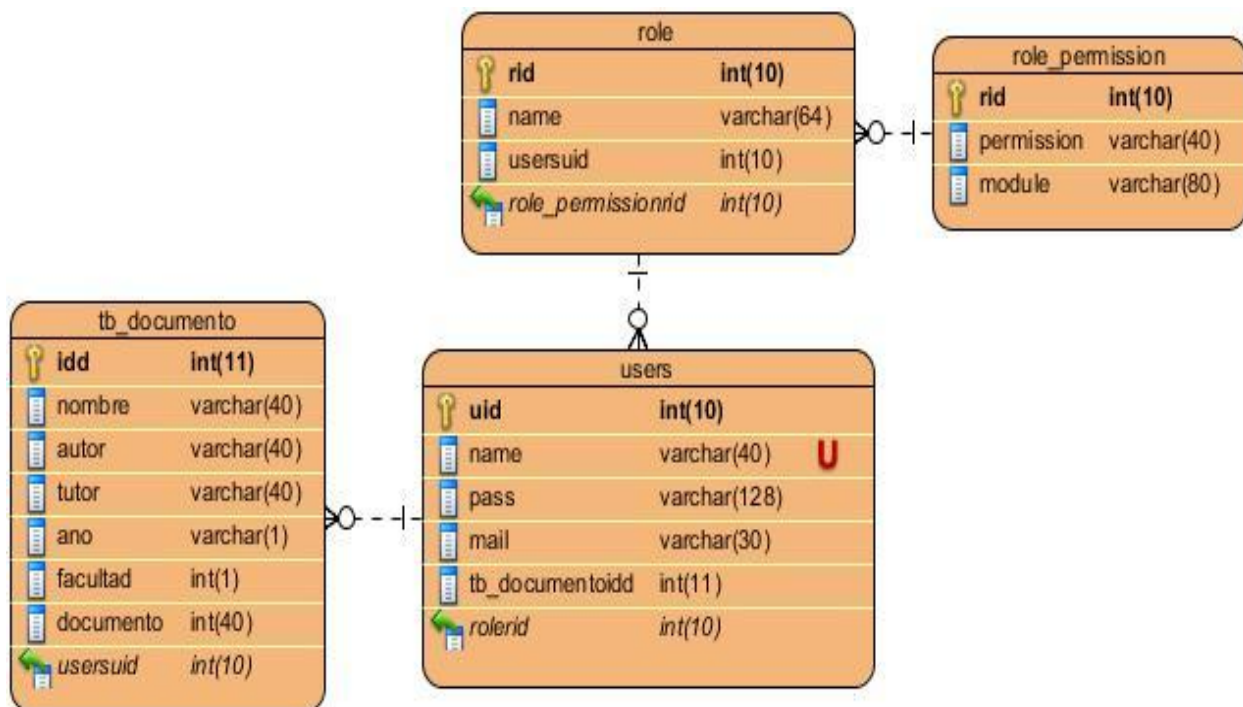
Mostrar el porcentaje de similitud.	
Mostrar el(los) nombre(s) de documento con similitudes.	
Exportar los resultados en formato .pdf.	

**Tabla 10. Tarjeta CRC: SimilitudDocumental.**

### 2.3.5 Modelo de datos

Un modelo de datos es un conjunto de conceptos que permiten describir a distintos niveles de abstracción, la estructura de una base de datos. Se trata de una colección de herramientas conceptuales que se emplean para especificar datos, las relaciones entre ellos, su semántica asociada y las restricciones de integridad. (26)

A continuación se muestra el diseño del modelo de datos asociado al sistema.



**Figura 6. Modelo de la base de datos.**



El modelo de datos cuenta con un total de cuatro tablas, de las cuales son de uso común para el sistema las tablas: *users*, *role* y *role\_permission*, donde se almacena información acerca de los usuarios con acceso a la aplicación, los roles asignados a cada usuario así como los permisos otorgados a cada rol respectivamente. Además cuenta con la tabla *Documento*, en la cual se almacenan los datos de los documentos evaluados como original que se encuentran en la base de datos.

## **2.4 Implementación**

La implementación del sistema es la parte más importante dentro del desarrollo del software en la metodología XP. Esta propone una serie de prácticas que sirven para el desarrollo exitoso del proyecto que se desee realizar. Se tiene en cuenta el desarrollo de las iteraciones según el Plan de iteraciones definido, la entrega del sistema en iteraciones pequeñas, un diseño simple y poco redundante del código con las funcionalidades necesarias.

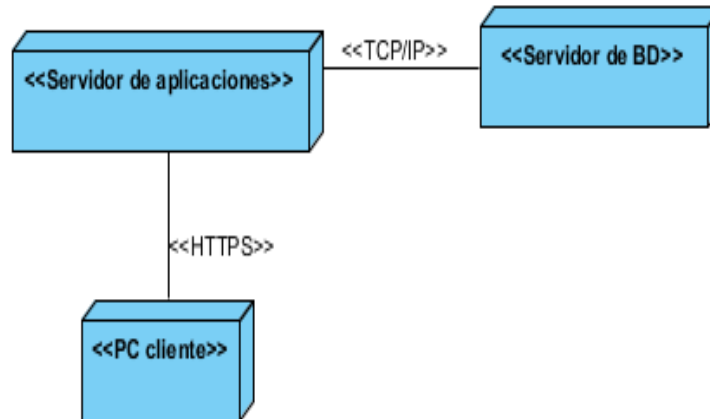
### **2.4.1 Modelo de implementación**

El modelo de implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Este modelo especifica cómo los elementos del modelo de diseño, fundamentalmente las clases, se implementan en términos de componentes como: ficheros de código fuente, ejecutables, librerías, entre otros. Además describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación utilizado. (27)

### **Diagrama de despliegue**

El diagrama de despliegue muestra las relaciones físicas entre componentes de hardware y software en el sistema, es decir, permite apreciar de forma visual, cómo se encuentran relacionados físicamente los componentes de la aplicación (28). XP no plantea el modelado de

este artefacto, pero para un mejor entendimiento del problema propuesto, se decide su realización.



**Figura 7. Diagrama de despliegue.**

El modelo de despliegue muestra la distribución de los nodos necesarios para el despliegue del sistema a nivel local. Para ello se contará con computadoras clientes, un servidor web y un servidor de base de datos. La comunicación entre las computadoras clientes y el servidor de aplicaciones se realizará mediante el protocolo HTTPS y TCP/IP entre el servidor de aplicaciones y el servidor de base de datos.

### 2.4.2 Tareas de ingeniería por Historia de Usuarios

Las tareas de ingeniería son actividades que los programadores conocen que el sistema debe hacer. Estas tareas se derivan directamente de las HU y deben ser implementadas entre uno y tres días ideales. Se usan para describir las tareas que se realizan sobre el proyecto, las cuales pueden ser: desarrollo, corrección y mejora.(29)

Historias de Usuario	Tareas de ingeniería
Autenticar usuario.	Crear interfaz para autenticar usuario. Permitir que el usuario se autentifique en el dominio UCI. Validar que la entrada que se someta a iniciar la sesión sea la correcta.

	Permitir o denegar el acceso a la aplicación.
Cargar documento.	Crear interfaz para cargar un documento. Cargar un documento en el sistema.
Analizar documentos.	Permitir la lectura del documento cargado en el sistema. Permitir que se extraiga el texto del documento (en formato: .doc, .docx, .odt) cargado en el sistema. Permitir que se compare el texto escrito entre el documento cargado en el sistema y los restantes que se encuentran en la base de datos.
Mostrar resultados de la evaluación.	Crear interfaz para mostrar los resultados de la evaluación de la originalidad. Permitir que se muestren los resultados de la evaluación.
Guardar documento.	Permitir que se guarde un documento evaluado como original en la base de datos.
Generar reporte.	Permitir que se exporten los resultados en un documento en formato pdf.
Gestionar documento.	Crear interfaz para agregar un documento a la base de datos. Permitir que se guarde un documento en la base de datos. Validar que la entrada de los datos del documento sea correcta. Crear interfaz para eliminar un documento de la base de datos. Permitir que se elimine un documento de la base de datos.

**Tabla 11. Distribución de las tareas por cada Historia de Usuario.**

A continuación se muestra la descripción de las tareas de ingeniería asociadas a las HU *Analizar documentos* y *Mostrar resultados de la evaluación*.

**HU Analizar documentos.**

<b>Tarea de ingeniería</b>	
<b>Número Tarea:</b> 7.	<b>Número Historia de Usuario:</b> 3.
<b>Nombre Tarea:</b> Permitir la lectura del documento cargado en el sistema	
<b>Tipo Tarea:</b> Desarrollo	<b>Puntos estimados (días):</b> 1.
<b>Fecha inicio:</b> 19/02/2013	<b>Fecha fin:</b> 19/02/2013
<b>Programador responsable:</b> Ariel Tomás Sánchez Pérez.	
<b>Descripción:</b> Se abre el documento cargado en el sistema y se lee para posteriormente extraer el texto que contiene el mismo para la realización de las comparaciones.	

**Tabla 12. Descripción de la tarea de ingeniería 7.**

<b>Tarea de ingeniería</b>	
<b>Número Tarea:</b> 8.	<b>Número Historia de Usuario:</b> 3.
<b>Nombre Tarea:</b> Permitir que se extraiga el texto del documento (en formato: .doc, .docx, .odt) cargado en el sistema.	
<b>Tipo Tarea:</b> Desarrollo	<b>Puntos estimados (días):</b> 2.
<b>Fecha inicio:</b> 20/02/2013	<b>Fecha fin:</b> 22/02/2013
<b>Programador responsable:</b> Ariel Tomás Sánchez Pérez.	
<b>Descripción:</b> Se abre el documento cargado en el sistema y se extrae el texto para ser comparado posteriormente con los restantes documentos que se encuentran en la base de datos.	

**Tabla 13. Descripción de la tarea de ingeniería 8.**

Tarea de ingeniería	
<b>Número Tarea:</b> 9.	<b>Número Historia de Usuario:</b> 3.
<b>Nombre Tarea:</b> Permitir que se compare el texto escrito entre el documento cargado en el sistema y los restantes que se encuentran en la base de datos.	
<b>Tipo Tarea:</b> Desarrollo	<b>Puntos estimados (días):</b> 15.
<b>Fecha inicio:</b> 5/04/2013	<b>Fecha fin:</b> 20/04/2013
<b>Programador responsable:</b> Ariel Tomás Sánchez Pérez.	
<b>Descripción:</b> Se compara el texto extraído entre el documento cargado en el sistema y los restantes que se encuentran en la base de datos, utilizando para ello el algoritmo seleccionado (Modelo Vectorial).	

Tabla 14. Descripción de la tarea de ingeniería 9.

**HU Mostrar resultados de la evaluación.**

Tarea de ingeniería	
<b>Número Tarea:</b> 10.	<b>Número Historia de Usuario:</b> 4.
<b>Nombre Tarea:</b> Crear interfaz para mostrar los resultados de la evaluación de la originalidad.	
<b>Tipo Tarea:</b> Desarrollo	<b>Puntos estimados (días):</b> 1.
<b>Fecha inicio:</b> 06/02/2013	<b>Fecha fin:</b> 06/02/2013
<b>Programador responsable:</b> Martha Cardonne Fong.	
<b>Descripción:</b> Se plasma mediante el código los elementos necesarios para que se muestren los resultados de la evaluación de la originalidad.	

Tabla 15. Descripción de la tarea de ingeniería 10.

Tarea de ingeniería	
<b>Número Tarea:</b> 11.	<b>Número Historia de Usuario:</b> 4.

<b>Nombre Tarea:</b> Mostrar los resultados de la evaluación.	
<b>Tipo Tarea:</b> Desarrollo	<b>Puntos estimados (días):</b> 5.
<b>Fecha inicio:</b> 18/04/2013	<b>Fecha fin:</b> 23/04/2013
<b>Programador responsable:</b> Ariel Tomás Sánchez Pérez.	
<b>Descripción:</b> Una vez analizado el documento cargado en el sistema con los restantes de la base de datos, se muestra al usuario el nombre de los documentos con el cual tuvo similitud el documento cargado en el sistema. Además se muestra el porcentaje de similitud asociado a cada documento así como el tipo de copia realizada.	

**Tabla 16. Descripción de la tarea de ingeniería 11.**

## 2.5 Conclusiones parciales

En este capítulo se describieron las tres primeras fases del ciclo de vida de la metodología XP. En la fase de Planificación se generaron las historias de usuario que caracterizan al sistema, así como la estimación del esfuerzo necesario para la implementación de las mismas. Además se construyó el plan de iteraciones el cual permitió conocer las historias de usuario que serán implementadas en cada iteración, así como el orden de prioridad de las mismas durante su desarrollo. Por otra parte se elaboró el plan de entregas, el cual permitió conocer la fecha aproximada en que serán entregadas las iteraciones definidas como partes funcionales del sistema. Igualmente se obtuvieron los elementos esenciales durante la fase de Diseño. A partir de las especificaciones de los requisitos funcionales, se realizó el diseño del diagrama de clases en el cual se evidencia el uso de los patrones de diseño utilizados para la obtención de un producto funcional. Además se diseñaron las tarjetas CRC así como el modelo de datos, representando este los datos que se manejan en el sistema. Por último, en la fase de Desarrollo se realizó el diseño del diagrama de despliegue, el cual permitió conocer los componentes necesarios para el despliegue del sistema a nivel local; además se definieron un conjunto de tareas de ingeniería con el objetivo de darle cumplimiento al desarrollo de las historias de usuario identificadas. De este modo se obtuvo una propuesta de solución que respondió a las necesidades del desarrollo de la herramienta para la evaluación de la originalidad en documentos de textos digitales.

## **CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN**

En el presente capítulo se muestran los resultados obtenidos de la aplicación de las métricas y técnicas empleadas para validar los requisitos y el diseño del sistema. Además se realizan las pruebas de caja blanca y caja negra para validar la solución propuesta.

### **3.1 Validación de los requisitos**

Una vez definidos los requisitos del sistema, estos deben ser validados para asegurar que el análisis de los mismos y los resultados obtenidos son correctos. La validación de los requisitos tiene como misión demostrar que su definición especifica realmente el sistema que el cliente desea.

Durante este proceso se examinan las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones; que los errores detectados hayan sido corregidos y que el resultado del trabajo se ajuste a los estándares establecidos para el proceso, el proyecto y el producto. (30)

#### **Técnica aplicada a los requisitos**

Para verificar que los requisitos identificados definen el sistema esperado por el cliente, se llevó a cabo un proceso de validación de los mismos, donde se empleó la técnica: Construcción de prototipos evolutivos. Mediante los prototipos se le mostró al cliente un modelo ejecutable de la herramienta, que le permitió tener una visión preliminar de cómo sería el sistema. Esto le permitió comprobar si satisfacía sus necesidades y dar su aprobación respecto al mismo. A continuación se muestra el prototipo de interfaz diseñado para la funcionalidad *Adicionar documento*.

Adicionar documento

Nombre:

Autor(es):

Tutor(es):

Año:

Facultad:

Documento:

Examinar <sup>1</sup> Subido al servidor <sup>2</sup>

Adicionar <sup>3</sup>

Detailed description: The image shows a web form titled 'Adicionar documento'. It contains six text input fields labeled 'Nombre:', 'Autor(es):', 'Tutor(es):', 'Año:', 'Facultad:', and 'Documento:'. Below these fields are three buttons: 'Examinar' (with a yellow callout box containing the number 1), 'Subido al servidor' (with a yellow callout box containing the number 2), and 'Adicionar' (with a yellow callout box containing the number 3). The buttons are arranged in a row, with 'Adicionar' positioned below the 'Documento:' field.

**Figura 8. Prototipo de interfaz de usuario de la funcionalidad Adicionar documento.**

### **Métricas aplicadas a los requisitos.**

Para medir la estabilidad de los requisitos, se aplicó la métrica “Estabilidad de requisitos” de la siguiente manera:

Se identificaron un total de 12 requisitos funcionales, de los cuales ninguno fue sometido a modificaciones, por tanto:

$$ETR = \left[ \frac{RT - RM}{RT} \right] * 100$$

**Donde:**

**RT:** Total de requisitos definidos.

**RM:** Cantidad de requisitos modificados.

Como resultado de la sustitución de los valores, se obtiene:

$$ETR = [(12 - 0) / 12 * 100 = \mathbf{100}.$$

Al medir la estabilidad de los requisitos se llegó a la conclusión que los requisitos definidos son estables, por lo tanto, es confiable trabajar el análisis y diseño sobre los mismos.

Para cuantificar la especificidad de los requisitos se aplicó la métrica “Especificidad de requisitos” de la siguiente manera:



$$Q_1 = \frac{n_{ui}}{n_r}$$

**Donde:**

**Q1:** Especificidad de los requisitos.

**Nr:** Total de requisitos definidos.

**Nui:** Total de requisitos donde los revisores tuvieron interpretaciones idénticas.

Como resultado de la sustitución de los valores, se obtiene:

$$Q_1 = 12/12 = 1$$

Por lo tanto se puede inferir que la especificación de los requisitos no posee ambigüedad, lo que asegura mayor calidad en el proceso de especificación.

Para medir que los requisitos identificados con el cliente fueron correctos, se aplicó la métrica "Grado de validación" de la siguiente manera:

$$Q_3 = \frac{n_c}{(n_c + n_{nv})}$$

**Donde:**

**Q3:** Grado de validación de requisitos.

**Nc:** Total de requisitos validados correctamente.

**Nnv:** Total de requisitos no validados.

Como resultado de la sustitución de los valores, se obtiene:

$$Q_3 = 12/(12+0) = 12/12 = 1$$

De acuerdo al resultado obtenido se concluye que la definición de los requisitos es correcta.

### 3.2 Validación del diseño

Las métricas de diseño permiten medir de forma cuantitativa la calidad de los atributos internos del software, permitiendo evaluar la calidad durante el desarrollo del sistema. Se centran en cuantificar tanto la complejidad, como la funcionalidad y eficiencia inmersa en el proceso de desarrollo de software. Sus objetivos están inclinados a mejorar la comprensión de la calidad del producto, a estimar la efectividad del proceso y mejorar la calidad del trabajo. (31)

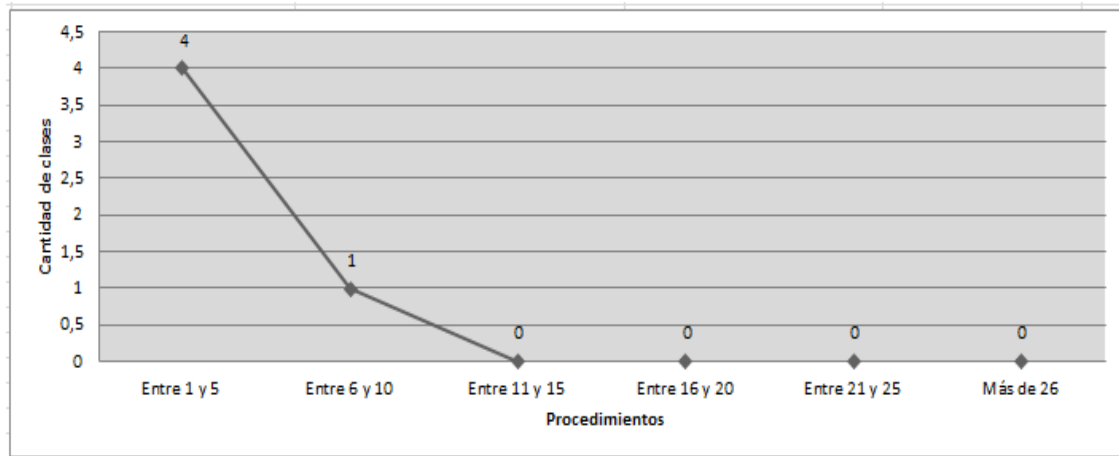
A continuación se muestran los resultados de la aplicación de dichas métricas.

#### **Tamaño Operacional de Clases (TOC):**

La métrica TOC fue aplicada a cada una de las clases con el objetivo de medir la calidad de las mismas con respecto a su grado de responsabilidad, complejidad y reutilización (31).

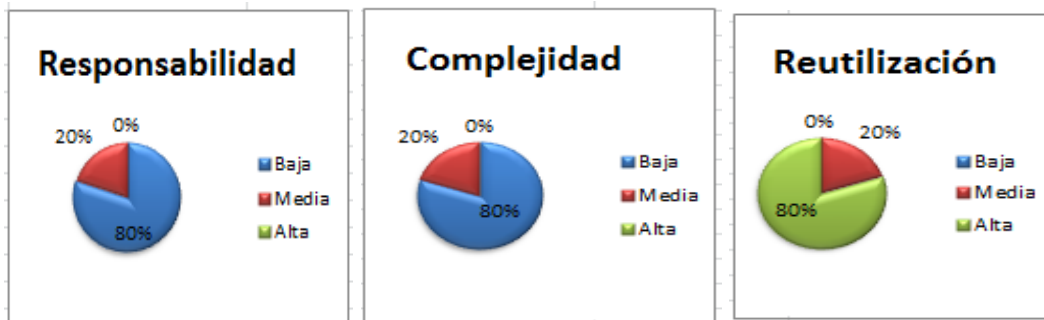
- ✓ **Responsabilidad:** Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto de la problemática propuesta. Un aumento del TOC implica un aumento de la responsabilidad asignada a una clase determinada. Si existen valores grandes de TOC significa que una clase tiene demasiada responsabilidad, lo cual reduciría la reutilización de la clase y hará complicada la implementación. De forma contraria sucede si los valores de TOC son pequeños.
- ✓ **Complejidad de implementación:** Consiste en el grado de dificultad que tiene implementar un diseño de clases determinado. Un aumento del TOC implica un aumento de la complejidad de implementación de una determinada clase.
- ✓ **Reutilización:** Consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software. Un aumento del TOC implica una disminución del grado de reutilización de una determinada clase.

En la siguiente gráfica se muestran los resultados obtenidos una vez aplicada dicha métrica.



**Figura 9. Representación de la cantidad de clases por cantidad de procedimientos que contienen.**

A continuación se muestra el porcentaje de las clases afectadas según los atributos de calidad establecidos por la métrica TOC.



**Figura 10. Resultados de la evaluación de la métrica TOC en los atributos Responsabilidad, Comportamiento y Reutilización respectivamente.**

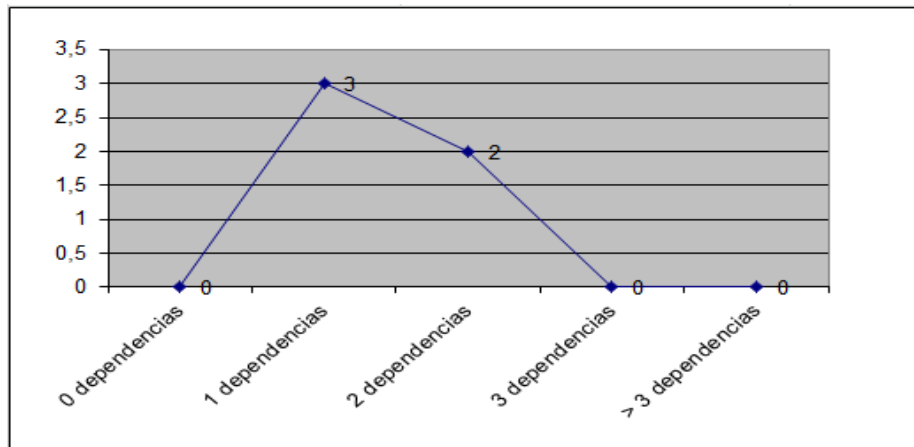
Las clases, de manera general, no están sobrecargadas en cuanto a responsabilidades, el nivel de complejidad de las mismas es baja lo que favorece en gran medida la reutilización de dichas clases. De esta manera el diseño de las clases favorecen el bajo acoplamiento y la alta cohesión.

### **Relaciones entre clases (RC):**

La métrica RC fue aplicada a cada una de las clases con el fin de medir el grado de acoplamiento, complejidad de mantenimiento, reutilización y la cantidad de pruebas necesarias asociadas a cada clase. (31)

- ✓ **Acoplamiento:** Consiste en el grado de dependencia o interconexión de una clase o estructura de clase con otras.
- ✓ **Complejidad del mantenimiento:** Consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de un determinado software. Puede influir indirectamente, pero fuertemente en los costes y la planificación del proyecto.
- ✓ **Reutilización:** Consiste en el grado de reutilización presente en una clase o estructura de clase dentro de un diseño de software. Un aumento del TOC implica una disminución del grado de reutilización de una determinada clase.
- ✓ **Cantidad de pruebas:** Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad del producto (componente, módulo, clase, conjunto de clases, etc.) diseñado.

En la siguiente gráfica se muestran los resultados obtenidos una vez aplicada dicha métrica.



**Figura 11. Representación de la cantidad de clases por cantidad de relaciones de usos que poseen.**

A continuación se muestra el porcentaje de las clases afectadas según los atributos de calidad establecidos por la métrica RC.

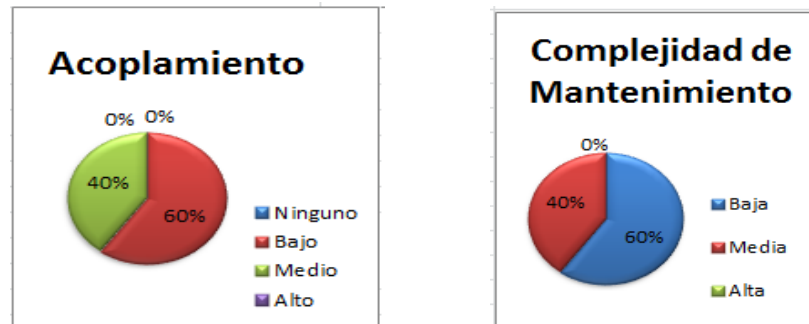


Figura 12. Resultados de la evaluación de la métrica RC en los atributos Acoplamiento y Complejidad respectivamente.

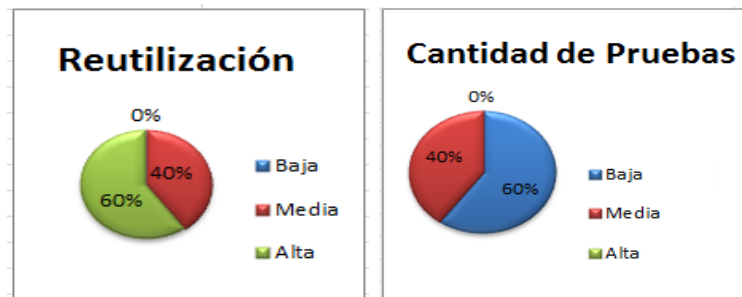


Figura 13. Resultados de la evaluación de la métrica RC en los atributos Reutilización y Pruebas respectivamente.

Luego de aplicar la métrica RC se tiene como resultado que las clases presentan valores de RC pequeños, es decir, un bajo acoplamiento, lo que garantiza menor grado de complejidad de mantenimiento de las mismas. Además las clases son reutilizables y necesitan pocas pruebas para ser probadas.

#### Resultados obtenidos:

En sentido general, los resultados obtenidos de la aplicación de las métricas TOC y RC demuestran que el diseño de la herramienta no es complejo, que las clases presentan bajo acoplamiento, una alta cohesión y un alto grado de reutilización.

### 3.3 Pruebas de Caja Blanca

La prueba de Caja Blanca del software se encarga de comprobar los caminos lógicos del sistema proponiendo casos de prueba para que se ejecuten conjuntos específicos de condiciones y/o bucles. (32)

Para llevar a cabo se utilizará la técnica de camino básico, la cual permite al equipo de desarrollo obtener una medida de la complejidad lógica del código implementado. Posteriormente el uso de esta medida servirá de guía para la definición de un conjunto de caminos independientes de ejecución, lo que garantiza que durante la prueba se ejecuten por lo menos una vez, cada sentencia del sistema desarrollado.

La complejidad ciclomática es una métrica de software que proporciona una medición cuantitativa de la complejidad lógica de un programa. Cuando se usa en el contexto de las pruebas, el cálculo de la complejidad ciclomática representa el número de caminos independientes del conjunto básico de un programa. (32)

Esta técnica está basada en la teoría de grafos, calculándose de tres formas (32):

1. El número de regiones del grafo de flujo coincide con la complejidad ciclomática.
2. La complejidad ciclomática,  $V(G)$ , de un grafo de flujo  $G$  se define como  $V(G) = A - N + 2$ , donde  $A$  es el número de aristas del grafo de flujo y  $N$  es el número de nodos del mismo.
3. La complejidad ciclomática,  $V(G)$ , de un grafo de flujo  $G$  también se define como  $V(G) = P + 1$ , donde  $P$  es el número de nodos predicado contenidos en el grafo de flujo  $G$ .

A continuación se enumeran las sentencias del código asociado al método `get_tabla_resultado`, el cual tiene como función una vez realizada la evaluación de la originalidad de un determinado documento, mostrar los resultados de la evaluación.

```

8 function get_tabla_resultado($documentos){
9     $header = array(
10         array('data' => t('Documento')),
11         array('data' => t('Similitud')),
12         array('data' => t('%')),
13     );
14     $rows = array();
15     foreach ($documentos as $key => $value){
16         if($key != 'doc_analizado' && $key != 'fid'){
17             $output->nombre = l($key, 'sites/default/files/documentos/' . $key);
18             $output->similitud = $value;
19             $porcentaje = $value*100;
20             $output->porcentaje = round($porcentaje*100)/100;
21             $rows[] = array('data' => (array) $output);
22         }
23     }
24     $tabla = array(
25         '#theme' => 'table',
26         '#header' => $header,
27         '#rows' => $rows
28     );
29     return $tabla;
30 }
    
```

Figura 14. Código fuente del método `get_tabla_resultado`.

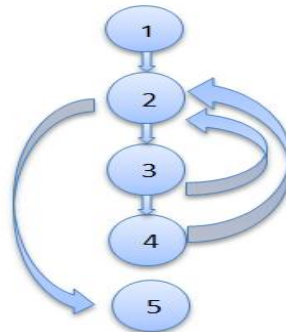


Figura 15. Grafo de flujo asociado al algoritmo.

### Cálculo de la Complejidad Ciclomática

1. El número de regiones del grafo de flujo coincide con la complejidad ciclomática.

2. El número de regiones del grafo del flujo definido es 5 lo que deviene que se obtenga el siguiente resultado:

$$V(G) = A - N + 2 = 6 - 5 + 2 = 3$$

Una vez calculada la complejidad ciclomática por cada una de sus variantes se define como límite superior cinco pruebas a realizar para que todo el código se encuentre probado.

**Caminos independientes establecidos:**

Camino1: 1-2-5

Camino2: 1-2-3-2-5

Camino3: 1-2-3-4-2-5

La realización de las pruebas de caja blanca permitió demostrar que las funciones internas se ajustan a las especificaciones y que los componentes internos funcionan correctamente.

### **3.4 Pruebas de Caja Negra**

Estas pruebas son llevadas a cabo sobre la interfaz del software, proporcionando entradas y estudiando las salidas para ver si son o no las esperadas. Estas pruebas se realizan para demostrar que cada función es operativa y al mismo tiempo se busquen errores en cada una de ellas.

A continuación se muestran los resultados obtenidos durante la realización de las pruebas de caja negra a través de las pruebas de aceptación.

#### **Pruebas funcionales o de aceptación**

Las pruebas funcionales o de aceptación, constituyen pruebas basadas en la ejecución, revisión y retroalimentación de las funcionalidades que han sido diseñadas para el software. Para ello, estas pruebas se realizan a través de modelos de prueba conocidos como casos de prueba, los cuales buscan evaluar cada una de las opciones con la que cuenta el sistema informático, enfocando su atención a las respuestas del sistema de acuerdo a los datos de entrada y su resultado en los datos de salida.



A través de estas pruebas se valida si los requisitos que fueron definidos por el cliente cumplen su funcionalidad adecuada en el sistema, buscando de esta forma reducir los fallos en el software y permitiendo la corrección en etapas tempranas.

A continuación se muestra la prueba de aceptación aplicada a las HU: *Analizar documentos* y *Mostrar resultados de la evaluación*.

Caso de Prueba de Aceptación	
<b>Código:</b> 3.	<b>Historia de Usuario:</b> Analizar documentos.
<b>Nombre:</b> Comprobar que la comparación entre los documentos se realice correctamente.	
<b>Descripción:</b> Prueba para la funcionalidad: Analizar documentos.	
<b>Fecha:</b> 23/04/2013.	
<b>Tipo de prueba:</b> Funcional	
<b>Condiciones de Ejecución:</b> El usuario tiene que estar autenticado. Se debe haber cargado un documento en el sistema.	
<b>Entrada / Pasos de ejecución:</b> Se carga un documento en el sistema en formato .doc, .docx, .odt. Se selecciona la opción <i>Analizar</i> para que dicho documento cargado en el sistema sea analizado con los restantes que se encuentran en la base de datos.	
<b>Resultado esperado:</b> El documento cargado en el sistema se compara correctamente con los restantes que se encuentran en la base de datos de referencia.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 17. Descripción del Caso de Prueba de Aceptación 3.

Caso de Prueba de Aceptación	
<b>Código:</b> 4.	<b>Historia de Usuario:</b> Mostrar resultados de la evaluación.
<b>Nombre:</b> Mostrar resultados de la comparación entre documentos.	

<b>Descripción:</b> Prueba para la funcionalidad: Mostrar resultados de la evaluación de la originalidad.
<b>Fecha:</b> 23/04/2013.
<b>Tipo de prueba:</b> Funcional
<b>Condiciones de Ejecución:</b> El usuario tiene que estar autenticado. Se debe haber cargado un documento en el sistema. Se debe haber realizado la comparación de este documento con los restantes que se encuentran en la base de datos.
<b>Entrada / Pasos de ejecución:</b> Se carga un documento en el sistema en formato .doc, .docx, .odt. Se selecciona la opción <i>Analizar</i> para que dicho documento cargado en el sistema sea analizado con los restantes que se encuentran en la base de datos.
<b>Resultado esperado:</b> Una vez finalizado el proceso de comparación entre documentos se muestran los resultados de la evaluación de la originalidad al usuario.
<b>Evaluación de la prueba:</b> Prueba satisfactoria.

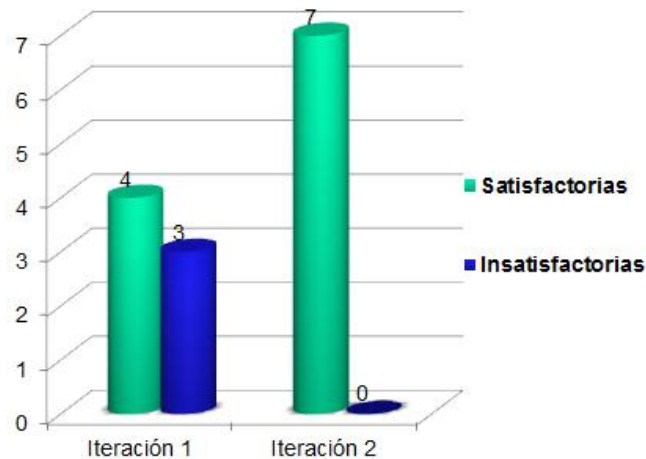
**Tabla 18. Descripción del caso de Prueba de Aceptación 4.**

### **Resultados obtenidos:**

Para validar que la salida emitida por el sistema informático fuese el resultado esperado por el cliente se diseñaron 7 pruebas funcionales. Se realizaron un total de dos iteraciones para lograr alcanzar los resultados satisfactorios desde el punto de vista funcional de la herramienta.

En la primera iteración, 4 coincidieron con los resultados esperados por el cliente representando un 75%, mientras que 3 pruebas resultaron fallidas representando un 25% de las pruebas insatisfactorias. En una segunda iteración se arrojaron a resultados satisfactorios, pues para el desarrollo de 7 pruebas funcionales se obtuvieron un total de 7 pruebas exitosas para un 100% de pruebas satisfactorias, mientras que ninguna de las pruebas realizadas resultaron fallidas representando un 0%.

En la siguiente gráfica se muestran los resultados por cada una de las iteraciones de las pruebas funcionales realizadas el sistema.



**Figura 8. Resultados de las pruebas de aceptación por iteraciones.**

.A continuación se realiza una descripción de las no conformidades detectadas durante la ejecución de las pruebas de aceptación.

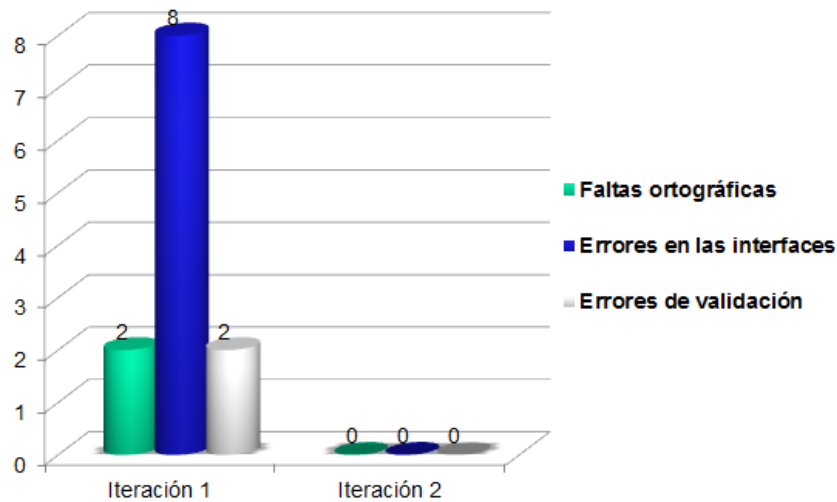
En la primera iteración se encontraron un total de 12 no conformidades (NC):

- ✓ Faltas ortográficas: 2 NC.
- ✓ Errores en las interfaces: 8 NC.
- ✓ Errores de validación: 2 NC.

En la segunda iteración no se encontraron conformidades (NC):

- ✓ Faltas ortográficas: 0 NC.
- ✓ Errores en las interfaces: 0 NC.
- ✓ Errores de validación: 0 NC.

La gráfica que se muestra a continuación representa el resultado de estas pruebas según las NC detectadas durante su ejecución:



**Figura 9. Resultados de las NC detectadas por iteraciones.**

La realización de las pruebas de caja negra permitió demostrar que las funciones son operativas, que las entradas al sistema se aceptan correctamente y que las salidas producidas por el mismo son las esperadas por el cliente.

### 3.5 Conclusiones parciales

Con la realización de este capítulo se evidenció la importancia y necesidad de realizar pruebas para validar la solución obtenida, ya que estas ayudaron a evaluar las clases diseñadas y permitieron descubrir errores una vez implementadas. La aplicación de técnicas de validación de requisitos permitió ratificar que los requisitos obtenidos estaban en correspondencia con las solicitudes del cliente. Se aplicaron además métricas de calidad a estas clases donde se evaluaron una serie de atributos de calidad dentro de los que se encuentran: Reutilización, Acoplamiento, Complejidad de Implementación, obteniéndose en todos los casos resultados satisfactorios. Además se realizaron pruebas de caja negra para validar la implementación, las cuales fueron arrojando resultados cada vez más positivos a medida que se fueron realizando las diferentes iteraciones, identificándose no conformidades tales como errores de validación de campos de entrada, errores ortográficos en algunas interfaces, entre otros, los cuales fueron corregidos eficazmente para lograr el correcto funcionamiento de la solución desarrollada.

## **CONCLUSIONES GENERALES**

Al término de esta investigación se concluye que:

- ✓ El estudio realizado sobre las diferentes tecnologías de desarrollo permitió seleccionar las herramientas y metodologías más adecuadas a utilizar.
- ✓ El estudio de los principales algoritmos de comparación de textos utilizados en los sistemas de detección del plagio a nivel nacional y mundial sirvió de base para la selección del algoritmo utilizado en la herramienta.
- ✓ Con el diseño e implementación de la herramienta para la evaluación de la originalidad en documentos de textos se resolvieron las limitantes planteadas en el problema.
- ✓ La aplicación de las pruebas tanto al código como a las interfaces permitió corregir las no conformidades de manera satisfactoria arrojando los resultados esperados por el equipo de desarrollo.
- ✓ Se obtuvo una herramienta para la evaluación de la originalidad en los documentos de textos digitales, que facilitará la determinación del grado de originalidad en los documentos emitidos en la esfera de investigación en la Universidad de las Ciencias Informáticas.

## **RECOMENDACIONES**

Una vez concluido el presente trabajo se recomienda:

1. Continuar con el desarrollo del sistema implementando un módulo de comparación de los documentos buscando similitudes en la Enciclopedia Cubana Ecured.
2. Integrar el sistema con la base de datos de la biblioteca UCI, para que realice comparaciones con los documentos almacenados en la misma.
3. Continuar con el desarrollo del sistema implementando un módulo que determine, durante el proceso de evaluación, la existencia de textos referenciados en los documentos para que posteriormente no se tengan en cuenta durante su análisis.

## REFERENCIAS BIBLIOGRÁFICAS

1. Glosario de la Organización Mundial de la Propiedad Intelectual. Ginebra, Suiza. 1980.
2. Rodríguez, Armando Soto. [En línea] 2 de diciembre de 2012. <http://revistaebci.ucr.ac.cr/volumenes/2/2-1/2-1-2/2-1-2.pdf>.
3. Software antiplagio detector de copiones. Software antiplagio detector de copiones. [En línea] [Citado el: 1 de diciembre de 2012.] <http://noticias.universia.es/vida-universitaria/reportaje/2009/06/09/647951/software-antiplagio-detector-copiones.pdf>.
4. Plagiarism Checker. Plagiarism Checker. [En línea] [Citado el: 5 de enero de 2013.] <http://www.dustball.com/cs/plagiarism.checker/>.
5. Article Checker. Article Checker. [En línea] [Citado el: 5 de enero de 2013.] <http://www.articlechecker.com/>.
6. IParadigms. . Turnitin. . [En línea] [Citado el: 5 de enero de 2013.] [Online] <http://www.turnitin.com..>
7. Pérez, Yurisel Aguirre. Sistema para la detección de plagio y validación de estructura en los documentos científicos emitidos en el Centro de Información Científico Técnica (CICT) del ISMMM. Sistema para la detección de plagio y validación de estructura en los documentos científicos emitidos en el Centro de Información Científico Técnica (CICT) del ISMMM. [En línea] Revista Caribeña de Ciencias Sociales. [Citado el: 5 de enero de 2013.] <http://caribeña.eumed.net/sistema-para-la-deteccion-de-plagio-y-validacion-de-estructura-en-los-documentos-cientificos-emitidos-en-el-centro-de-informacion-cientifico-tecnica-cict-del-ismmm/>.
8. Cedeño, Luis Alberto Barrón. Detección automática de plagio en texto. [En línea] noviembre de 2008. [Citado el: 12 de febrero de 2013.] <http://users.dsic.upv.es/~proso/resources/BarronCede%C3%B1oMSc.pdf>.
9. Diego Rodríguez, José Martín. Detección de plagio en documentos. Sistema externo monolingüe de altas prestaciones basado en n-gramas contextuales. Huelva : s.n., 2010.
10. Elizalde, Victoria. Estudio y desarrollo de nuevos algoritmos de detección del plagio. Universidad de Buenos Aires : s.n., 2011.
11. Solís, Manuel Caldero. Una explicación de la programación extrema (XP). Una explicación de la programación extrema (XP). Una explicación de la programación extrema (XP). Una explicación de la programación extrema (XP). [En línea] [Citado el: 5 de enero de 2013.] [http://www.apolosoftware.com/..](http://www.apolosoftware.com/)
12. PHP. 2001-2011. Hypertext Preprocessor. . PHP. 2001-2011. Hypertext Preprocessor. . [En línea] [Citado el: 5 de enero de 2013.] <http://www.php.net/>.

13. Qué es un entorno de desarrollo integrado, IDE. Programación Desarrollo. Qué es un entorno de desarrollo integrado, IDE. Programación Desarrollo. [En línea] Editorbfb, 2011. [Citado el: 5 de enero de 2013.] <http://programaciondesarrollo.es/que-es-un-entorno-de-desarrollo-integrado-ide/>.
14. NetBeans IDE 7.3 Release Information. NetBeans IDE 7.3 Release Information. [En línea] [Citado el: 5 de enero de 2013.] <https://netbeans.org/community/releases/73/>.
15. Herramienta CASE. [En línea] [Citado el: 5 de enero de 2013.] [www.case.org/](http://www.case.org/)
16. Sommerville, Ian. Ingeniería del software. Madrid. España : Pearson Educación : Séptima Edición, 2005. S. A., 2005. 84-7829-074-5..
17. Software de prototipado para la arquitectura de la información: funcionalidad y evaluación. [En línea] [Citado el: 12 de febrero de 2013.] [http://www.academia.edu/314808/Software\\_de\\_prototipado\\_para\\_la\\_arquitectura\\_de\\_la\\_informacion\\_funcionalidad\\_y\\_evaluacion](http://www.academia.edu/314808/Software_de_prototipado_para_la_arquitectura_de_la_informacion_funcionalidad_y_evaluacion).
18. Sistemas de Gestión de Bases de datos y SIG. [En línea] [Citado el: 6 de febrero de 2013.] [http://www.um.es/geograf/sigmur/sigpdf/temario\\_9.pdf](http://www.um.es/geograf/sigmur/sigpdf/temario_9.pdf).
19. La biblia de MySQL.
20. Servidor Web y de aplicaciones. [En línea] [Citado el: 6 de febrero de 2013.] <http://trevinca.ei.uvigo.es/~txapi/espanol/proyecto/superior/memoria/node19.html>.
21. Apache Tomcat. The Apache Software Foundation, 1999-2012. [En línea] <http://tomcat.apache.org/>.
22. Conociendo las siglas: CMS. Conociendo las siglas: CMS. [En línea] <http://programaciondesarrollo.es/conociendo-siglas-cms/>.
23. 27 Características del CMS Drupal. Drupal: CMS Drupal, el futuro es ahora. 27 Características del CMS Drupal. Drupal: CMS Drupal, el futuro es ahora. [En línea] Pilos, 2009. [Citado el: 5 de enero de 2013.] [http://www.pilos.com.co/drupal/27-86-caracteristicas-de-drupal/..](http://www.pilos.com.co/drupal/27-86-caracteristicas-de-drupal/)
24. Pressman. Cap\_07\_Ingenieria\_de\_Requisitos\_Parte\_1.
25. Emilio A. Sánchez Patricio, Letelier José, H. Canós. Mejorando la gestión de historias de usuario en eXtreme Programming. Mejorando la gestión de historias de usuario en eXtreme Programming. [En línea] Departamento de Sistemas Informáticos y Computación. [Citado el: 12 de enero de 2013.] <http://issi.dsic.upv.es/publications/archives/f-1068918773758/JISBD03SanchezLetelier.pdf>.
26. Ciclo de vida de un proyecto XP. Ciclo de vida de un proyecto XP. [En línea] [Citado el: 12 de enero de 2013.] <http://oness.sourceforge.net/proyecto/html/ch05s02.html>.



27. Modelo del dominio. [En línea] [Citado el: 10 de junio de 2013.] <http://lsi.ugr.es/~ig1/isoo/larman/Modelo%20del%20dominio.pdf>.
28. Modelo de clases. [En línea] <http://users.dcc.uchile.cl/~psalinas/uml/modelo.html>.
29. Descubre las tarjetas CRC. [En línea] Isidro Fuentes Hermoso. [Citado el: 15 de marzo de 2013.] [www.bubok.es/libros/208759/Descubre-las-tarjetas-CRC](http://www.bubok.es/libros/208759/Descubre-las-tarjetas-CRC).
30. Pantaleón, Marta E. Zorrilla. [En línea] Universidad de Cantabria. Universidad de Cantabria. [Citado el: 15 de marzo de 2013.] <http://personales.unican.es/zorrillm/PDFs/Docencia/Master/02%20-%20Modelos%20de%20datos%20ER-UML-relacional.pdf..>
31. Pressman, Roger. Ingeniería de Software, un enfoque práctico. 2005.
32. Diagrama de despliegue. [En línea] [Citado el: 15 de febrero de 2013.] [lsi.ugr.es/~is2/Teoria/.../l.Diagrama%20de%20%20despliegue-0910red.pdf](http://lsi.ugr.es/~is2/Teoria/.../l.Diagrama%20de%20%20despliegue-0910red.pdf).
33. Beck., K. Extreme Programming Explained. 1999.
34. Técnicas de validación de requisitos. Técnicas de validación de requisitos. [En línea] [Citado el: 20 de febrero de 2013.] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/419>.
35. Maite Sánchez Fornaris, Dayanis Alcantara Rabí. Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica. 2010.
36. Pruebas de software. Pruebas de software. [En línea] [Citado el: 20 de marzo de 2013.] <http://www.cc.uah.es/drg/docencia/Pruebas/Pruebas4x1.pdf>.