

Universidad de las Ciencias Informáticas
Facultad 2



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

Título: *Sistema de descubrimiento de bibliografía*
científica

Autores:

Marelys Martínez Moreira

Alberto Boza García

Tutor: Ing. Vladimir Milián Núñez

Co-tutor: Ing. Roberto A. Infante Milanés

La Habana, junio 2016

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 firmamos la presente a los ____ días del mes de _____ del _____.

Marelys Martínez Moreira

Firma del autor

Ing. Vladimir Milián Núñez

Firma del tutor

Alberto Boza García

Firma del autor

Ing. Roberto A. Infante Milanés

Firma del co-tutor



“El único modo de hacer un gran trabajo es amar lo que haces”

Steve Jobs

Dedicatoria

A mi mamita linda Niurka.

A mi super papá Alberto.

A mi superultramega hermano Jorgito.

A mis abuelas hermosas y a mis abuelitos.

A todos los presentes, los que creyeron, no me abandonaron y confiaron en mí.

Muchas gracias.

Alberto

A mis padres por su cariño, apoyo incondicional y por ser los promotores de todos mis logros.

Marelys

Agradecimientos

A mi compañera de tesis por escucharme siempre, por ser tan recalcitrante y por estar presente en todo momento para lograr llegar a esta tan deseada meta.

A mi mamá, la persona más importante en mi vida, por estar siempre a mi lado, en los buenos y malos momentos, por apoyarme y ayudarme, es evidente que no podría expresar todo lo que siento por ti y todo lo que tengo que agradecerte en tan cortas líneas.

A mi papá, por ser un luchador incansable contra todo lo que me pueda perjudicar, por estar siempre al pendiente de lo que pueda pasarme, alertarme, cuidarme y preocuparse por mí, realmente gracias papá por todo lo que haces por mí, eres un ejemplo para mí, te quiero mucho papá, gracias.

A mi hermano, por sacar siempre lo mejor de mí, por siempre hacer esa broma que me hace reír cada vez que estoy triste, te quiero mi hermanito lindo.

A Darvis, por ser más que un decano y un profesor, un amigo, por confiar en mí y darme su apoyo cuando más lo necesitaba, gracias Darvis, y a Mirian Nicado, por creer en mí, en mi palabra, por preocuparse por mí como una madre por su hijo, créame, aún no he conocido una persona más humana, Ud. también es como una madre para mí, jamás voy a olvidar lo que ha hecho Ud. por mí, aunque me diga mil veces que no, yo si siento que le debo mucho, gracias, muchas gracias y mil veces gracias por todo. Te quiero mucho.

A mis abuelas, Alla y Caty, gracias por preocuparse por mí, por sus consejos, son por las cuales me guio, a mis abuelitos, que aunque no están presentes, los quiero y los extraño mucho, quiero dedicarles este trabajo.

A mis niñas, Elaine y Mayara, realmente las amo con todas las fuerzas de Mi Corazón, gracias por estar siempre conmigo, apoyándome en todo, por estar conmigo en los buenos y los malos momentos, por no fallarme ni abandonarme, las quiero mucho. Mayara siempre serás la mamá, sabes de que hablo.

A mis hermanos del alma, Asney, Eric, Manuel y Yordi, Uds. han sido lo más grande que he tenido aquí en estos 6 años de carrera, gracias por todo, realmente los quiero un montón, nunca los olvidare.

A Ramón, un amigo, más que un amigo un hermano, gracias por estar conmigo, gracias por las noches sin dormir, por tu apoyo incondicional, de veras gracias, gran parte de esto te lo debo a ti, gracias mi hermano.

A todas esas personas que formaron parte de mi vida en la universidad, a Maite, mi adorada novia Yilian, a mis colegas Liony, los Leos, Pablo, Yisel, Rosi, Stephany, gracias por todo, por su apoyo, por creer, por no abandonarme y confiar en mí, por su amistad y por estar siempre conmigo, Uds., mis niñas y mis hermanos del alma forman parte de mi familia también, los quiero.

A mis tutores, Robe y Vlade, siempre pendientes y siempre ausentes, pero apoyándonos y defendiéndonos en todo, gracias de veras.

Alberto

A mis padres por ser las personas más maravillosas y humanas que he conocido. Gracias a ambos por su apoyo incondicional, incluso en los momentos en que fui criticada. Gracias por confiar en mí y cuidarme todos estos años, prometo recompensar todo el amor y cariño que me han brindado.

A mi hermanito por demostrarme su amor detrás de cada celo, después de cada pelea.

A mis abuelitos por inculcarme el carácter responsable de una persona.

A mis abuelitas por intentar incansablemente hacer de mí una persona alegre y fuerte.

A mis tíos y tías, ya sean los de sangre y los que no, gracias a todos por cuidarme como si fuera su propia hija.

A mi amiga y hermana Antu, por brindarme siempre su apoyo, aún en la distancia. Gracias amiga por cada momento de alegría, por cada consejo, por cada regaño, por cada lágrima que derramamos juntas.

A mi novio por ser paciente, apoyarme y quererme en los momentos buenos y difíciles. Gracias por soportar mis malcriadeces y querer hacer de mí una mejor mujer.

A mis compañeras de cuarto, Lazarita, Aray y Lupita. Gracias Lazarita por ser una amiga incondicional todos estos años, y a ti Lisi por demostrarme que no se requiere de mucho tiempo para conocer a una verdadera amiga.

A todas las personas que de una forma u otra han pasado por mi vida, haciendo de mí una mujer fuerte y con ambiciones.

A mis tutores, Vlade y Robe por confiar en nosotros y apoyarnos incansablemente.

A las personas que me enseñaron que con un poco de paciencia, esperanza y lucha, al final de todo terminamos realizando nuestros sueños más deseados.

Agradecer por último, y no por eso menos importante, a mi compañero de tesis, por haberme escogido para realizar juntos este sueño. Gracias por darme ánimos y fuerza en los momentos que lo necesitaba, por ser mi amigo y mi bastón en el camino hasta esta anhelada meta.

Gracias a todos.

Marelys

Resumen

Actualmente, el crecimiento de la web ha permitido que las personas tengan a su disposición una determinada cantidad de información que en ocasiones es mayor a la que pueden analizar. Para solucionar el problema que constituye la sobrecarga de información, surgen los sistemas de recomendación. Este tipo de sistemas son herramientas que brindan a los usuarios sugerencias de contenidos interesantes de acuerdo al objetivo de búsqueda.

El objetivo de la presente investigación es desarrollar un componente para el sistema web "Representación de Bibliografía Científica" desarrollado en la Universidad de las Ciencias Informáticas. Este componente permitirá obtener bibliografías científicas similares a partir de un documento subido por el usuario en el sistema. De esta manera la aplicación podrá realizar recomendaciones de publicaciones científicas utilizando el método por similitud de objetos. Estas recomendaciones le brindan al usuario la posibilidad de obtener bibliografía realmente relevante. Un proceso importante en el desarrollo del componente es el proceso de minería de texto, donde se seleccionó el algoritmo Agrupamiento Espacial Basado en Densidad de Aplicaciones con Ruido, para agrupar los documentos similares al que posee el usuario. El componente se desarrolló usando el marco de desarrollo Django 1.6, mediante el lenguaje de programación Python 2.7, haciendo uso de las bibliotecas NLTK, Sklearn y Psycopg2. Además se utilizó Pycharm como entorno de desarrollo integrado y PostgreSQL como sistema gestor de base de datos.

Palabras claves: científica, componente, documentos, minería de texto, similitud, sistemas de recomendación

Índice de Contenido

Introducción.....	13
Capítulo 1: Fundamentación Teórica	17
1.1. Introducción.....	17
1.2. Sistemas de Recomendación	17
1.3. Análisis de los sistemas existentes de recomendación bibliográfica por similitud de objetos	18
1.4. Conceptos asociados al objeto de estudio.....	20
1.5. Metodologías de desarrollo	26
1.6. Tecnologías y herramientas de desarrollo	29
1.7. Conclusiones parciales.....	32
Capítulo 2: Características del sistema	33
2.1. Introducción.....	33
1.3. Modelo conceptual	33
2.3. Propuesta de solución	34
2.4. Especificación de los requisitos del sistema	38
2.5. Definición de los actores.....	40
2.6. Diagrama de casos de uso del sistema	41
2.7. Conclusiones parciales.....	49
Capítulo 3: Arquitectura y diseño.....	50
3.1. Introducción.....	50
3.2. Diseño de la arquitectura.....	50
3.3. Etapa de Diseño	51
3.4. Modelo físico de los datos	55
3.5. Patrones de diseño.....	56
3.6. Conclusiones Parciales	58
Capítulo 4: Implementación y Prueba	59
4.1. Introducción.....	59
4.2. Etapa de implementación	59
4.3. Estándares de codificación empleados.....	60
4.4. Diagrama de despliegue.....	60
4.5. Pruebas.....	62
4.6. Conclusiones parciales.....	68
Conclusiones Generales	69

Recomendaciones..... 70
Referencias bibliográficas 71

Índice de Tablas

Tabla 1: Requisitos funcionales del sistema	39
Tabla 2: Actor relacionado con el sistema	41
Tabla 3: Caso de prueba de partición equivalente del RF2: Mostrar documentos similares.	66
Tabla 4: Resultados de las pruebas de Caja negra	68

Índice de Figuras

Ilustración 1: Fases de CRISP-DM	27
Ilustración 2: Modelo conceptual	34
Ilustración 3: Flujo de la propuesta de solución	34
Ilustración 4: Preparación de los datos	35
Ilustración 5: Aplicar minería.....	36
Ilustración 6: Procedimiento del algoritmo DBSCAN.....	37
Ilustración 7: Fórmula para calcular la similitud entre documentos	37
Ilustración 8: Variable Épsilon.....	38
Ilustración 9: Diagrama de casos de uso del sistema	41
Ilustración 10: Arquitectura Cliente – Servidor.	50
Ilustración 11: Funcionamiento del MTV en Django.....	51
Ilustración 12: Diagrama de paquetes	52
Ilustración 13: Diagrama de clases del diseño Subir documento	53
Ilustración 14: Diagrama de clases del diseño Mostrar documentos similares	53
Ilustración 15: Diagrama de clases del diseño Revisar y Descargar documentos obtenidos	54
Ilustración 16: Diagrama de colaboración de los RF: Subir documento y Mostrar documentos similares	55
Ilustración 17: Diagrama de colaboración de los RF: Revisar documentos y Descargar documento.....	55
Ilustración 18: Modelo físico de la base de datos.....	56
Ilustración 19: Diagrama de componentes del sistema	59
Ilustración 20: Diagrama de Despliegue	61
Ilustración 21: Pruebas unitarias.....	63
Ilustración 22: Resultado de las pruebas unitarias.....	64
Ilustración 23: Gráfico correspondiente al caso de prueba del RF1: Subir documento.....	65
Ilustración 24: Gráfico correspondiente al caso de prueba del RF3: Revisar documentos ...	66
Ilustración 25: Gráfico correspondiente al caso de prueba del RF4: Descargar documento	66
Ilustración 26: Prueba de carga - Obtener documentos similares	67

Introducción

En los últimos años, el número de tecnologías con acceso a la web ha tenido una tendencia al aumento, debido fundamentalmente a la posibilidad de acceder a Internet desde los dispositivos móviles al igual que desde las estaciones de trabajo. Sobre este escenario Internet se convierte en un medio para la obtención de información, que permite mediante una simple conexión realizar búsquedas en varias fuentes de datos que están en constante crecimiento. De manera general estas búsquedas generan un volumen de resultados demasiado grandes, lo que provoca que los usuarios se sobrecarguen de información.

Una solución a la sobrecarga de información, ha sido el empleo de los sistemas de recomendación. Estos sistemas realizan la recuperación y/o filtrado de información, con el objetivo de ofrecerle a los usuario contenidos interesantes.

No siempre la información obtenida del tema buscado es la más relevante y/o esperada. En gran medida esto depende de la forma en la que se organiza la información en las bases de datos y del método de recuperación de información que se utilice, siendo el más común la búsqueda por palabras claves. Algunas veces los resultados devueltos no son satisfactorios debido al volumen de datos y la falta de coherencia entre el objeto de búsqueda y los resultados. Por solo mencionar un ejemplo, si se introduce en el panel de búsqueda la palabra Python se obtiene información sobre el lenguaje de programación y la serpiente Python de manera simultánea. Como primeros efectos esto provoca el desvío del curso de la búsqueda y la pérdida de tiempo en la selección de la bibliografía realmente relevante.

En el campo de la informática, la búsqueda de información es fundamental en las fases iniciales de un proyecto de investigación. En ocasiones se cuenta con una documentación base, pero la misma no es suficiente ya que se debe profundizar en diferentes bibliografías para lograr un marco teórico sólido en el área de estudio. Por este motivo se hace necesaria la obtención de documentos similares a partir de un documento base. Pese a esto, las

herramientas de búsquedas actuales solo permiten realizar este proceso de forma manual, como por ejemplo Google¹, Bing², Yahoo³, Ask⁴, entre otros.

En el curso 2014-2015 en la Universidad de las Ciencias Informáticas (UCI), como resultado del trabajo de diploma (Montero, y otros, 2015), se desarrolló el sistema web de recomendación “Representación de Bibliografía Científica” (RBC). Esta herramienta le brinda al usuario la posibilidad de introducir un criterio de búsqueda por palabras claves y obtener publicaciones científicas en el área de las Ciencias Básicas y la Cibernética. A pesar de esto RBC no permite la obtención de documentos científicos similares a uno que posea el usuario, utilizando el método de recomendación por similitud de objetos. Esto provoca que la selección de la documentación más relevante deba realizarse manualmente, leyendo y analizando la información obtenida. Algunos de los efectos negativos que trae consigo este método son la selección de información poco relevante y el desvío del objetivo de la búsqueda, así como la pérdida de tiempo que conlleva analizar el volumen de información obtenido.

El **problema a resolver** se centra en la incapacidad del sistema RBC de realizar la búsqueda de bibliografía científica mediante la técnica de similitud de objetos, lo que afecta negativamente la obtención de documentos científicos relevantes que sean similares a uno que posea el usuario. Teniendo en cuenta lo expuesto anteriormente, la presente investigación tiene como **objetivo general** desarrollar un componente para el sistema RBC que permita la búsqueda de bibliografías científicas utilizando la recomendación por similitud de objetos.

El **objeto de estudio** abarca los sistemas de recomendación de bibliografía científica que utilizan la técnica de similitud de objetos y el **campo de acción** se centra en la recomendación de bibliografía científica en el sistema RBC.

Para dar cumplimiento al objetivo general se definen los siguientes **objetivos específicos**:

1. Realizar la fundamentación teórica de la investigación para alcanzar un dominio sobre el tema abordado.

¹ www.google.com

² www.bing.com

³ www.yahoo.com

⁴ www.ask.com

2. Elaborar el análisis y diseño de una propuesta de solución que cumpla con las funcionalidades necesarias para resolver la problemática planteada.
3. Implementar la propuesta de solución a la problemática planteada.
4. Validar la solución desarrollada aplicando los métodos de prueba.

Para dar cumplimiento a los objetivos específicos definidos anteriormente y solucionar el problemática planteada, se detallan las siguientes **tareas de la investigación**:

- Revisión de bibliografías relacionadas con el objeto de estudio y el campo de acción.
- Análisis de sistemas existentes similares a la propuesta de solución a desarrollar.
- Descripción de las herramientas de desarrollo, tecnologías, metodología y lenguaje de programación a utilizar en el análisis, diseño e implementación de las nuevas funcionalidades del sistema RBC.
- Definición de las nuevas funcionalidades y características principales que tendrá la aplicación web RBC.
- Implementación de las nuevas funcionalidades al sistema web RBC.
- Realización de pruebas a las nuevas funcionalidades del sistema RBC para verificar el correcto funcionamiento de las mismas.

Para la realización de la actual investigación se emplearon varios métodos científicos de investigación, los cuales fueron:

Métodos teóricos:

- **Histórico – Lógico:** Se evidencia en el estudio realizado a bibliografías de trabajos investigativos, artículos científicos, revistas publicadas sobre los sistemas de recomendación bibliográficos que utilizan el método por similitud de objetos.
- **Analítico – Sintético:** Se evidencia en el análisis del comportamiento de los sistemas de recomendación bibliográficos que utilizan el método por similitud de objetos, identificando sus características y funcionalidades, permitiendo la extracción de los elementos más significativos y el arribo a conclusiones teóricas y prácticas bien definidas.
- **Modelación:** Utilizado para modelar la arquitectura, crear los artefactos, diagramas y modelos a utilizar en el desarrollo de las nuevas funcionalidades del sistema RBC.

Métodos empíricos:

- **Entrevista:** Se empleó para conocer las necesidades del cliente y definir los requisitos y características de la solución de la propuesta.
- **Observación:** Se empleó para definir el valor de la variable ϵ a emplear en el algoritmo de agrupamiento seleccionado para agrupar los documentos similares al que introduce el usuario al sistema.

Para una mejor comprensión, el presente documento consta de 4 capítulos:

Capítulo 1: “Fundamentación Teórica”, permite encontrar los principales conceptos que se manejan a lo largo del trabajo investigativo; la metodología de desarrollo, las herramientas y lenguajes de programación que apoyan el desarrollo de la solución del problema planteado.

Capítulo 2: “Características del sistema”, expone los elementos que forman parte de las características de la propuesta de solución y se explica todo el proceso de minería de texto para el agrupamiento de documentos similares al que posee el usuario. Además son definidos los requisitos funcionales y no funcionales del sistema, así como los actores relacionados con la aplicación. Se establecen los artefactos requeridos en la planificación definidos por la metodología AUP.

Capítulo 3: “Arquitectura y diseño”, propone el diseño del sistema RBC, a partir de los Diagramas de Clases del Diseño empleando estereotipos web, los diagramas de colaboración, el diagrama de paquetes y el diseño de la base de datos. Se define el estilo arquitectónico y los patrones de diseño utilizados en el desarrollo de las nuevas funcionalidades del sistema RBC.

Capítulo 4: “Implementación y prueba”, describe el proceso de implementación de las nuevas funcionalidades del sistema RBC y los principales resultados obtenidos en la etapa de pruebas, para garantizar su correcto funcionamiento y su cumplimiento con los requisitos definidos por el cliente.

Capítulo 1: Fundamentación Teórica

1.1. Introducción

En el presente capítulo se abordan los elementos teóricos necesarios para darle soporte a la presente investigación. Se definen los conceptos de: sistemas de recomendación bibliográfica, minería de texto, procesamiento del lenguaje natural, preprocesamiento de los datos y el clustering⁵ de documentos. Se realiza un estudio del impacto social que tienen actualmente los sistemas que emplean la recomendación por similitud de objetos. Además, se muestra una descripción de las herramientas, tecnologías, metodologías y lenguaje de programación a emplear en el desarrollo de las nuevas funcionalidades del sistema RBC.

1.2. Sistemas de Recomendación

Los Sistemas de Recomendación (SR) “...ayudan al usuario a escoger elementos de una gran cantidad de opciones. El volumen de la información es la razón principal del surgimiento de estos sistemas”, según (Mizhquero Cañar, y otros, 2009).

(Torres, 2014) plantea que los SR son “...toda técnica de deducción de la información ofrecida por un usuario a unos temas concretos y conocidos por el sistema. Posteriormente el sistema compara la información deducida del usuario con otra información deducida de la misma forma de otras fuentes y ofrece al usuario una ponderación de qué fuente es la más afín al usuario”.

Para un mejor entendimiento se puede resumir conceptualmente un SR como una herramienta encargada de ayudar a las personas a encontrar información dentro del amplio volumen de datos disponible en Internet, a partir de las preferencias de los usuarios y de comportamientos similares.

Definir el tipo de recomendación en el momento de crear un SR, conlleva a especificar la técnica de filtrado de información a utilizar. Para la selección del tipo de recomendación a emplear en el presente trabajo se consideran los definidos por (Mizhquero Cañar, y otros, 2009), estos son:

⁵ División de los datos en grupos de objetos similares.

- *La recomendación por similitud de objetos se caracteriza porque el perfil creado para representar al usuario se basa en el análisis del contenido de los objetos, con el fin de recomendar objetos similares a los usuarios.*
- *La recomendación social es aquella que busca características sociales similares entre los usuarios para recomendar ítems⁶.*
- *En la recomendación basada en historia algunos sistemas mantienen una lista de compras, el historial de navegación en la World Wide Web⁷ o el contexto de correos electrónicos como un perfil de usuario.*

Para el desarrollo del componente a implementar para el sistema RBC, se decide emplear la recomendación por similitud de objetos como técnica para el filtrado de la información, debido a que el sistema debe ser capaz de buscar en una base de datos documentos similares al que posee inicialmente el usuario y que es introducido en el sistema.

1.3. Análisis de los sistemas existentes de recomendación bibliográfica por similitud de objetos

Como parte del estudio de las tendencias actuales de los SR bibliográfica que emplean la técnica de similitud de objetos, se analizaron los sistemas que se describen en este epígrafe. Debido a que no se encontraron referencias de la existencia de un sistema de este tipo en Cuba, los que se presentan a continuación, constituyen aplicaciones desarrolladas a nivel internacional.

En el estudio de estas herramientas se definieron algunas variables que permitieran evaluar si estos sistemas constituyen una solución al problema planteado. Los elementos que se tuvieron en cuenta fueron:

- Que fueran aplicaciones web.
- Que para la recomendación de bibliografía, utilizarán la similitud entre documentos.

⁶ Se usa para hacer distinción de artículos o capítulos en un escrito.

⁷ Telaraña de alcance mundial, o simplemente la Web. Forma de ver toda la información disponible en Internet. Sistema de distribución de información tipo revista, en la red quedan almacenadas lo que se llaman Páginas Web que no son más que páginas de texto con gráficos o fotos.

- Que permitieran obtener bibliografía científica similar a un documento que introduzca el usuario al sistema.

ClusterDoc: esta aplicación web fue resultado de la investigación (Giugni, y otros, 2011). ClusterDoc está dirigido a los investigadores de una comunidad científica. La herramienta divide el conjunto de documentos almacenados en pequeños grupos con características comunes, lo cual permiten minimizar el espacio de búsqueda y proporcionar información adaptada a los intereses del usuario. La recomendación se utiliza realizándole al investigador, en este caso el usuario, una encuesta, donde se mide en una escala del 1 al 5 el nivel de interés del investigador en una dominio de investigación determinado, y de esta manera brindarle una plan de lectura acorde a sus intereses.

WCopyfind: herramienta desktop para el sistema operativo Windows. Este sistema examina una colección de archivos de documentos, en busca de similitudes. Cuando encuentra dos documentos que comparten suficientes palabras en común, WCopyfind genera archivos de informes HTML. Estos informes contienen el texto del documento con las frases que emparejan subrayado. Esta herramienta no permite la búsqueda de documentos en la web o de Internet.

Recomendador de Bibliografía (RB): esta herramienta fue diseñada e implementada en el 2014, la misma fue resultado del proyecto Sistema de descubrimiento de bibliografía científica, cuyo autor es Sergio Santamaría Torres⁸. Esta aplicación fue desarrollada para desktop⁹ y tiene como objetivo descubrir nuevos documentos científicos que se asemejen a los que el usuario introduce en el sistema. El usuario al introducir uno o varios documentos a la aplicación, obtiene documentos que ofrece la búsqueda en Internet y posteriormente obtiene de esos resultados los que son realmente similares al documento subido por él en el sistema.

Como resultado del estudio realizado, se pudo constatar que:

- El sistema *ClusterDoc* utiliza para la recomendación de artículos científicos la similitud entre los documentos almacenados, pero no permite que el usuario suba un documento al sistema y obtenga de este, publicaciones científicas similares.

⁸ Ingeniero informático en la Faculta de Informática de Barcelona.

⁹ Equipo de escritorio.

- La función del sistema *Recomendador de Bibliografía* corresponde con la del componente a desarrollar en la presente investigación, ya que permite obtener documentos similares a uno introducido por el usuario en el sistema. Pese a esto al igual que el sistema *WCopyfind* son herramientas desktop, por lo que traen consigo las siguientes desventajas:
 - Acceso limitado al computador donde se haya instalado.
 - Solo funcionan en el sistema operativo para el cual fue desarrollado.
 - La capacidad de usuarios es baja, ya que se centran en un único usuario local.
 - La actualización se debe realizar en cada computador donde este instalada.

Teniendo en cuenta los elementos antes mencionados se puede concluir que ninguno de los sistemas analizados constituye una solución al problema planteado anteriormente, por lo que se hace necesario adicionarle a la herramienta RBC nuevas funcionalidades. Estas funcionalidades deben permitir la obtención de bibliografías científicas a partir de un documento subido al sistema por el usuario, utilizando la técnica de recomendación por similitud de objetos.

1.4. Conceptos asociados al objeto de estudio

1.4.1. Minería de texto

(Torres, 2014) plantea que: *“la particularidad de la recomendación de documentos es decidir qué es lo más relevante para categorizar de un texto, ya que textos diferentes pueden estar expresando exactamente lo mismo. Para poder categorizar y recomendar textos es necesario entender que entre documentos el concepto de similitud se refiere a parecido semántico, de significado, y no una similitud textual”*. Para obtener las relaciones semánticas entre documentos se estudia los precedentes desarrollados en el campo de la minería de textos.

El término minería de texto varía en algunas de las bibliografías consultadas. Algunos de los conceptos encontrados son los siguientes:

- La minería de textos es *“el tratamiento que se debe realizar sobre textos escritos en lenguaje natural para poder buscar y encontrar información relevante en ellos”*, según (Torres, 2014).
- (Hernández, 2014) define la minería de textos como *“una tecnología emergente cuyo objeto es la búsqueda de conocimiento en grandes colecciones de documentos no estructurados”*.

Conceptualmente se puede resumir la minería de texto como el área que se encarga del estudio de la información digital textual, con el objetivo de descubrir patrones y tendencias para obtener conocimiento oculto, útil y aplicable.

Un proceso de minería de textos consta de varias etapas:

1. **Pre-procesamiento:** *“Como ya se sabe que el texto no presenta una estructura para aplicar fácilmente las técnicas de minería de texto de forma directa, es necesario realizar operaciones o transformaciones sobre el texto, en algún tipo de presentación estructurada o semi-estructurada que facilite sin posterior análisis”* (Hernández, 2014).
2. **Minería de texto:** *“Etapa de descubrimiento donde las representaciones intermedias se analizan con el objetivo de descubrir patrones o conocimiento nuevo en las representaciones intermedias. En esta etapa se emplean técnicas de minería de texto como la categorización y clasificación de textos, descubrimiento de asociaciones, detección de desviaciones, análisis de tendencias, entre muchas más”* (Hernández, 2014).
3. **Visualización de los resultados:** *“En esta etapa, una vez que han aplicado la técnica de minería de texto, se escoge la representación de los textos; la cual podría ser por medio de palabras, términos llaves, características, conceptos, sugerencias de textos, etc”* (Hernández, 2014).

Algunos de los problemas más habituales resueltos por la minería de textos son los siguientes (Torres, 2014):

- Clasificación automática de documentos
- Extracción de información relevante
- Resumen automático de textos
- Filtrado de e-mails de spam¹⁰
- Posicionamiento web en buscadores de internet
- Traducción de documentos en diferentes idiomas

Existen dos vías para aplicar la minería de textos a cada problema concreto, expresa (Torres, 2014):

¹⁰ Correo electrónico no solicitado que se envía a un gran número de destinatarios con fines publicitarios o comerciales.

- Aprendizaje basado en el lenguaje natural
- Aprendizaje basado en ejemplos anteriores

El presente trabajo de investigación se centrará en el problema de la extracción de información relevante, debido a la necesidad de recuperar dicha información de los documentos científicos para obtener la similitud entre los mismos. Como vía a emplear para solucionar el problema antes mencionado se utilizará el aprendizaje basado en el lenguaje natural, siendo esta una de las técnicas más utilizadas para la transformación de documentos.

1.4.2. Procesamiento del lenguaje natural

El lenguaje natural (LN) *“es el medio que utilizamos de manera cotidiana para establecer nuestra comunicación con las demás personas. El LN ha venido perfeccionándose a partir de la experiencia a tal punto que puede ser utilizado para analizar situaciones altamente complejas y razonar muy sutilmente. Los lenguajes naturales tienen un gran poder expresivo y su función y valor como una herramienta para razonamiento”* (Cortez Vásquez, y otros, 2009).

El procesamiento del lenguaje natural consiste en la *“utilización de un lenguaje natural para comunicarnos con la computadora, debiendo ésta entender las oraciones que le sean proporcionadas, el uso de estos lenguajes naturales, facilita el desarrollo de programas que realicen tareas relacionadas con el lenguaje o bien, desarrollar modelos que ayuden a comprender los mecanismos humanos relacionados con el lenguaje”* (Cortez Vásquez, y otros, 2009).

(Hernández, 2014) define que *“el procesamiento del lenguaje natural es una disciplina que relaciona directamente la computación y la lingüística, la cual tiene como principal objetivo conseguir que el lenguaje humano pueda utilizarse como entrada en un proceso automatizado”*.

Para extraer información relevante de textos no estructurados, es decir, textos escritos en lenguaje natural, el presente trabajo se centrará en el algoritmo de procesamiento de lenguaje natural **Parts Of Speech** (POS, siglas en inglés). (Torres, 2014) expresa que: *“mediante este algoritmo se analiza el texto de forma de que una vez comprendido el texto se puedan deducir los conceptos sobre los que trata”*.

Para un mejor entendimiento se puede resumir que el algoritmo POS permite obtener los conceptos más relevantes de los que trata un documento. A continuación se explica el preprocesamiento que debe aplicarse a los datos obtenidos para que sean de utilidad en tiempo de ejecución.

1.4.3. Preprocesamiento de los datos

“El objetivo fundamental del preprocesamiento de datos es manipular y transformar los datos en bruto de modo que el contenido de la información envuelto en el conjunto de datos puede ser expuesto o accesibles con mayor facilidad. (Pyle, 1999).

En el preprocesamiento de los datos se realiza “...la selección, la limpieza, el enriquecimiento, la reducción y la transformación de las bases de datos” (Sinnexus, 2007). Existen dos pasos para realizar este proceso: la estandarización de textos y la extracción de características de los textos.

Estandarización de textos

En el método de estandarización se realizan las siguientes actividades:

1. Eliminación de caracteres extraños y signos de puntuación.
2. Eliminación de mayúsculas.
3. Bag of words (bolsa de palabras) del texto: separa el texto dividiéndolo por palabras y formando una lista con las mismas.
4. Eliminación de stopwords¹¹.
5. Stemming¹² del texto.

Se puede concluir que en la etapa de estandarización del texto, se obtiene como resultado una lista de palabras con repeticiones, es decir, una misma palabra puede estar repetida varias veces.

Extracción de características de los textos

Muchas veces después de la estandarización de los textos se obtiene una gran cantidad de datos. El objetivo de la etapa de extracción de características es reducir el tamaño de los

¹¹ Palabras carentes de significado, por ejemplo, un, y para, por, de, el, uno, dos, entre otros.

¹² Reducción de cada palabra a su raíz lexicográfica o lexema.

datos obtenidos en la etapa anterior, formando una serie de modelos que permitan deducir conceptos de similitud o clasificación de textos. La determinación de estos modelos se logran utilizando algoritmos de minería de texto.

A continuación, se realiza un estudio de las técnicas de minería de texto y los algoritmos utilizados en cada una de ellas.

1.4.4. Técnicas de Minería de texto

*“Las técnicas de minería de texto se pueden clasificar en descriptivas y predictivas. Las **descriptivas** caracterizan las propiedades generales de los datos que se encuentran en las formas intermedias y, por el contrario, las **predictivas** realizan inferencias en los datos para poder realizar predicciones. Hay que tener en cuenta lo que se desea obtener para determinar cuál de los dos enfoques se utiliza”* (Hernández, 2014).

1.4.4.1. Técnicas de tipo descriptivas (no supervisadas)

Análisis de varianza: *“Mediante el mismo se evalúa la existencia de diferencias significativas entre las medias de una o más variables en poblaciones distintas”* (Vallejos, 2006).

Análisis discriminante: *“Permite la clasificación de individuos en grupos que previamente se han establecido, permite encontrar la regla de clasificación de los elementos de estos grupos, y por tanto una mejor identificación con las variables que definan la pertenencia al grupo”* (Vallejos, 2006).

Agrupamiento: *“Es un procedimiento de agrupación de una serie de vectores según criterios habitualmente de distancia; se tratará de disponer los vectores de entrada de forma que estén más cercanos aquellos que tengan características comunes de tal manera que se maximice la similitud entre los vectores de un mismo grupo y se minimice la similitud entre los grupos, además esta técnica puede ser combinada fácilmente con cualquier otra”* (Garre M., 2005).

1.4.4.2. Técnicas de tipo predictivas (supervisadas)

Redes neuronales: *“Es una técnica que modela computacionalmente el aprendizaje humano llevado a cabo a través de las neuronas del cerebro. Las redes de neuronas constituyen una nueva forma de analizar la información con una diferencia fundamental respecto a las técnicas tradicionales: son capaces de detectar y aprender complejos patrones y características dentro de los datos”* (Garre M., 2005)

Clasificación: *“Es el proceso de dividir un conjunto de datos en grupos mutuamente excluyentes de tal manera que cada miembro de un grupo esté lo "más cercano" posible a otro, y grupos diferentes estén lo "más lejos" posible uno del otro, donde la distancia está medida con respecto a variable(s) específica(s) las cuales se están tratando de predecir”* (Vallejos, 2006).

Árbol de decisión: *“Los árboles de decisión son una técnica que permite analizar decisiones secuenciales basada en el uso de resultados y probabilidades asociadas. Cada nodo interno denota una prueba sobre un atributo. Cada rama representa el resultado de una prueba. Las hojas denotan las clases”* (Garre M., 2005).

1.4.4.3. Selección de la técnica de Minería de texto

Clustering o agrupamiento es la técnica seleccionada en la investigación, ya que se ajusta a la solución del objetivo general debido a la necesidad de agrupar los documentos similares al que sube el usuario al sistema.

El clustering de documentos es *“el proceso de buscar un agrupamiento natural en un conjunto de datos en base a su similitud”*, expresa (Godoy, 2015).

Dentro de las familias de algoritmos de clustering más utilizados se encuentran:

- Basado en particiones: crean particiones sucesivas del conjunto de datos.
- Jerárquicos: descomposición jerárquica del conjunto de objetos.
- Basados en densidades: funciones densidad y conectividad.

El tipo de algoritmo de clustering a utilizar en el presente trabajo es el agrupamiento basado en densidades. Dentro de estos algoritmos se encuentran:

- Agrupamiento Espacial Basado en Densidad de Aplicaciones con Ruido (DBSCAN, siglas en inglés)
- Ordenando Puntos para Identificar la Estructura de la Agrupación (OPTICS, siglas en inglés)
- Agrupación basado en densidad (DENCLUE, siglas en inglés)
- Agrupación Compartida de Vecinos más Próximos (SNN Clustering, siglas en inglés)

El algoritmo de agrupamiento seleccionado para agrupar los documentos similares al que introduce el usuario en el sistema, es el Agrupamiento Espacial basado en Densidad de

Aplicaciones con Ruido (DBSCAN, siglas en inglés). Este algoritmo puede adaptar a diferentes escenarios y trabaja bien frente a problemas como la redundancia y el ruido.

1.5. Metodologías de desarrollo

(Moine, 2013) define una metodología de desarrollo como *“un conjunto de actividades organizadas que tienen por objetivo la realización de un trabajo. Para cada actividad se define, además de las entradas y salidas, la forma en la que debe llevarse a cabo”*. En el presente trabajo se hace necesario el empleo de una metodología para guiar el trabajo con los datos y otra para el desarrollo de las nuevas funcionalidades.

Metodología para guiar el proceso de minería de texto

Existen varias metodologías para orientar el proceso de minería de texto, entre las cuales se encuentran Semma, CRISP-DM, Catalist, entre otras. Se decide utilizar la metodología Cross Industry Standard Process for Data Mining (CRISP-DM, siglas en inglés), teniendo en cuenta lo planteado por (Arancibia, 2009), cuando refiere que *“CRISP-DM es la guía de referencia más ampliamente utilizada en el desarrollo de proyectos de Data Mining”*.

La metodología CRISP-DM propone 6 fases para guiar el desarrollo del proceso de minería de texto. A continuación, se muestra una imagen donde se evidencia cada una de estas fases y el proceso jerárquico que existe entre ellas.

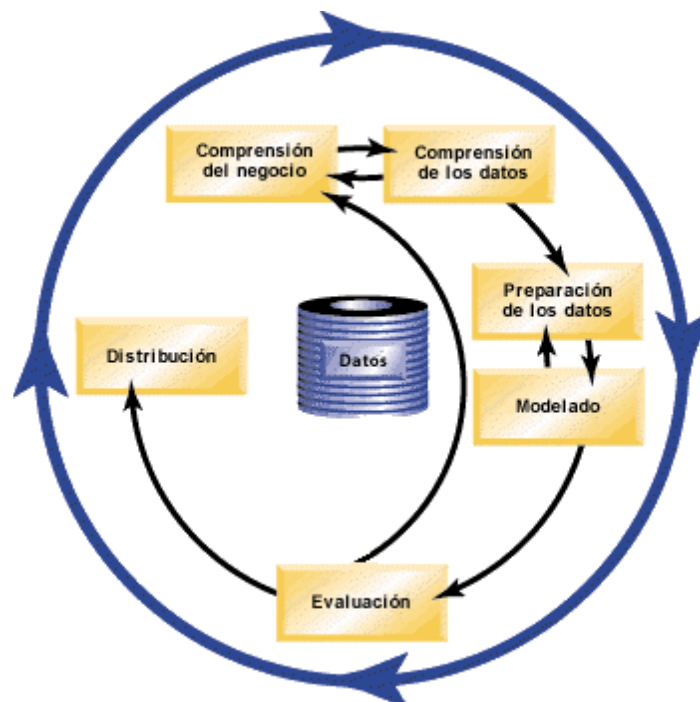


Ilustración 1: Fases de CRISP-DM

Comprensión del negocio: en esta fase determinan los objetivos y requerimientos del proyecto desde una perspectiva del negocio, definiendo el problema de minería y el plan de trabajo.

Comprensión de los datos: fase que consiste en la recolección de datos que se utilizarán en el proyecto y la familiarización con los mismos.

Preparación de los datos: comprende aquellas actividades de tratamiento de los datos para construir la vista minable o conjunto de datos final sobre el cual se aplicarán las técnicas de minería.

Modelado: en esta fase se aplican las diversas técnicas y algoritmos de minería sobre el conjunto de datos para obtener la información oculta y los patrones implícitos en ellos.

Evaluación: fase en la que se analizan los patrones obtenidos en función de los objetivos organizacionales. En esta etapa se debería determinar si se ha omitido algún objetivo importante del negocio y si el nuevo conocimiento será implementado, es decir, si se pasará a la próxima etapa.

Implementación: consiste en la comunicación e implementación del nuevo conocimiento, el cual debe ser representado de forma entendible para el usuario. (Moine, 2013)

CRISP-DM es flexible y se puede personalizar fácilmente, es decir, permite crear un modelo de minería de datos que se adapte a necesidades concretas. Además, profundiza en mayor detalle sobre las tareas y actividades a ejecutar en cada etapa del proceso de minería de datos.

Metodología para guiar el desarrollo del componente de búsqueda

Para el desarrollo del sistema RBC se utilizó la metodología XP, la misma realiza la descripción de los requisitos a través de Historia de usuario¹³ y las Tarjetas CRC¹⁴. Se decide no emplear esta metodología para guiar el desarrollo del componente de búsqueda propuesto

¹³ Representación de un requisito escrito en una o dos frases utilizando el lenguaje común del usuario.

¹⁴ Técnica de diseño orientado a objetos, que realiza un inventario de las clases necesarias para implementar el sistema y la forma en que van a interactuar.

en la presente investigación, ya que no brinda técnicas que permitan lograr una concepción detallada del negocio para poder definir la propuesta de solución. Por este motivo, en sustitución de la metodología XP, se decide utilizar la metodología Proceso Unificado Ágil (AUP, siglas en inglés), ya que a través de los artefactos que propone, permite lograr una clara comprensión del negocio y realizar una descripción más detallada de los requerimientos del componente a desarrollar.

Para seleccionar esta metodología se tuvo en cuenta que además:

- El equipo de desarrollo es pequeño (2 integrantes)
- Se cuenta con poco tiempo para el desarrollo de la herramienta
- Existe un intercambio directo con el cliente

AUP es una metodología de desarrollo ágil que heredera de otros paradigmas como la programación extrema (XP) y Rational Unified Process (RUP). Consta de principios y prácticas influyentes en la construcción del software en armonía con la documentación esencial de entregables específicos para el entendimiento de la solución. Entre sus objetivos destaca la reducción del costo del cambio en el proyecto en base a procedimientos iterativos (característica propia de RUP) donde la codificación y pruebas del software se llevan a cabo paralelamente (según XP) (Galindo, 2012).

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva (Flores, y otros):

Concepción: permite obtener una comprensión común sobre el alcance del nuevo sistema, entre el cliente y el equipo de desarrollo, definiendo una o varias arquitecturas candidatas para el mismo.

Elaboración: el objetivo es que al equipo de desarrollo se le facilite la comprensión de los requisitos del sistema y pueda validar la arquitectura definida.

Construcción: es la fase en la que el sistema es desarrollado y probado por completo en el entorno de desarrollo.

Transición: el sistema es llevado a entornos de preproducción donde se somete a determinadas pruebas, tanto de validación como de aceptación, siendo desplegado finalmente en los sistemas de producción.

Una de las ventajas de AUP radica en la forma de planificar el proyecto y la estimación de tiempo, factor importante ya que se cuenta con poco tiempo para el desarrollo de las nuevas funcionalidades. AUP en relación con la metodología RUP simplifica la documentación de algunos procesos de desarrollo, documenta solo aquellos que son necesarios.

La metodología AUP propone 4 escenarios para la disciplina Requisitos.

Escenario 1: Proyectos que modelen el negocio con Caso de Uso del Negocio (CUN) solo pueden modelar el sistema con Caso de Uso del Sistema (CUS).

$CUN + Modelo\ Conceptual\ (MC) = CUS$

Escenario 2: Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.

$MC = CUS$

Escenario 3: Proyectos que modelen el negocio con Diagrama de Procesos del Negocio (DPN) solo pueden modelar el sistema con Descripción de Requisitos por Proceso (DRP).

$DPN + MC = DRP$

Escenario 4: Proyectos que no modelen el negocio solo pueden modelar el sistema con Historia de Usuario (HU).

El escenario sobre el cual se trabajará es el Escenario 2, ya que luego de evaluar el negocio se llegó a la conclusión de que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades. Además el objetivo principal del presente trabajo corresponde con lo que propone el escenario 2, la gestión y presentación de la información, por la importancia que requiere los resultados obtenidos en la búsqueda de bibliografía científica.

1.6. Tecnologías y herramientas de desarrollo

Como se mencionó anteriormente, el componente de búsqueda propuesto por la presente investigación se integrará con el sistema RBC, por lo que se consideró utilizar las mismas herramientas con las cuales se implementó el sistema RBC. A continuación, se muestra una breve descripción de las mismas.

Lenguaje de Programación

Un lenguaje de programación permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis.

Para la solución propuesta se utiliza el lenguaje de programación Python en su versión 2.7, permitiendo el desarrollo de aplicaciones web rápidas y fáciles. Además de la adaptabilidad con el sistema operativo Windows 8.1, en el cual se está desarrollando la herramienta, y la integración de numerosas bibliotecas.

Lenguaje de modelado

El Lenguaje de Modelado Unificado (UML, siglas en inglés) “...es una de las herramientas más utilizadas en el mundo actual del desarrollo de software, esto se debe a que permite a los desarrolladores crear diseños que engloben sus propósitos de manera sencilla y fácil de comprender para otras personas. El UML está compuesto por diversos elementos gráficos que se combinan para formar diagramas. Proporciona características que permiten organizar y extender los diagramas. Es necesario resaltar que UML indica **qué** es lo que supuestamente hará el sistema, pero no **cómo** lo hará”. (Schmuller)

Framework utilizado

Un framework es un ambiente de trabajo que contiene librerías de códigos y módulos que pueden ser reutilizados para el rápido desarrollo de aplicaciones. Se definió para el desarrollo del sistema la utilización de Django en su versión 1.6.

Django es un framework que utiliza Python y que permite el desarrollo rápido de aplicaciones web. Usa una modificación de la arquitectura Modelo-Vista-Controlador (MVC), llamada MTV (Model-Template-View), que sería Modelo-Plantilla-Vista, esta forma de trabajar permite que sea pragmático¹⁵.

Bibliotecas empleadas para el desarrollo del componente de búsqueda

1. *NLTK*: se usa para el trabajo con texto, preprocesamiento y limpieza de palabras, como, por ejemplo:
 - Reducir todas las palabras a su raíz lexicográfica (Ejemplo: implementando - implementar).

¹⁵ Actúa dando prioridad a las consideraciones prácticas.

- Eliminar los caracteres raros, los signos de puntuación y dejar todo el texto en minúscula.
 - Elimina las palabras comunes de la lengua en la que está escrito (Ejemplo: el, para, de, por, y, un, entre otras).
2. *SKLEARN*: se utiliza para el trabajo de minitextos y con algunos algoritmos lógicos.
 3. *PSYCOPG*: se usa para la conexión del sistema con la base de datos.

Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado o IDE (siglas provenientes del inglés Integrated Development Environment), es un programa informático que brinda un conjunto de componentes que hacen más fácil la programación. Conforman un ambiente favorable para los desarrolladores. Como entorno de desarrollo integrado se utiliza Pycharm en su versión 4.0.4, el cual permite la integración con el framework Django y soporta intérpretes de Python 2.7.

Herramienta CASE

Se selecciona Visual Paradigm for UML como herramienta CASE¹⁶ para el modelado de la propuesta de solución. Esta herramienta permite representar todo tipo de diagramas en el ciclo de vida del desarrollo de software.

Sistema Gestor de Base de Datos

Un Sistema de Gestor de Base de Datos (SGBD) es un conjunto de programas que permiten la creación de base de datos y proporciona herramientas para añadir, borrar, modificar y eliminar datos de la base de datos, además de mantener la integridad, confidencialidad y seguridad de los datos.

PostgreSQL 9.2.4 es un SGBD objeto-relacional de código abierto, el cual puede ser ejecutado sobre la mayoría de los sistemas operativos que existen en la actualidad. El sistema es usado para manejar grandes cantidades de información y se basa en el modelo relacional, aunque incorpora conceptos del modelado orientado a objeto. Se destaca por ser robusto y

¹⁶ Ingeniería de Software Asistida por Computadora (CASE, siglas en inglés). Las herramientas CASE permiten la implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras.

cumplir con los estándares SQL. Es soportado ampliamente por una gran comunidad a nivel mundial (PostgreSQL, 2013).

Se utiliza PgAdmin en su versión 1.16 como herramienta de código abierto con el propósito general de diseñar, mantener, y administrar las bases de datos de PostgreSQL.

1.7. Conclusiones parciales

En este capítulo se analizaron algunos sistemas existentes para la recomendación de bibliografía científica por similitud de objetos. Se identificaron las características de los sistemas analizados y se detallaron conceptos relacionados con estos. Se definió el uso de dos metodologías, AUP como metodología para guiar el desarrollo de la herramienta web, ya que el equipo es pequeño y se cuenta con poco tiempo de desarrollo. Se selecciona la metodología CRISP-DM orientada al trabajo con los datos, ya que es la guía de referencia más ampliamente utilizada en el desarrollo de proyectos de minería de datos.

Capítulo 2: Características del sistema

2.1. Introducción

En el presente capítulo se exponen los elementos que permiten describir la propuesta de solución. Se abordan los procesos de negocio, diagramas y descripciones asociados al proceso de minería de texto para el clustering de documentos. Además son definidos los requisitos funcionales y no funcionales del sistema, así como los actores relacionados con la aplicación. Se establecen los artefactos requeridos en la planificación definidos por la metodología AUP.

1.3. Modelo conceptual

Un modelo conceptual tiene como objetivo *“identificar y explicar los conceptos significativos en un dominio de problema, identificando los atributos y las asociaciones existentes entre ellos”*. (Software, 2005). En este modelo se definen cuáles son y cómo se relacionan los conceptos relevantes en la descripción del problema, en este caso describe los relacionados con el negocio del sistema RBC.

A continuación se describen los conceptos fundamentales que componen el modelo conceptual del negocio:

Usuario: Persona que por medio de un ordenador puede acceder a la herramienta para realizar la búsqueda.

Cliente: Es un usuario del sistema que además, es el responsable de la gestión de documentos y otros usuarios.

Criterio de búsqueda: Comprende la palabra o el conjunto de palabras que introduce el usuario para buscar información.

Resultados de la búsqueda: Es un espacio donde se mostrarán los resultados de la búsqueda definida por el usuario.

Administración: Es un módulo del sistema, donde se podrán gestionar todas las funcionalidades.

Gestionar Documento: Es la funcionalidad donde se podrán gestionar todos los datos referentes a la gestión de documentos.

Gestionar Usuario: Es la funcionalidad donde se podrán gestionar todos los datos referentes a los usuarios autorizados a acceder a la administración.

Base de datos de publicaciones científicas: Es el espacio donde se encontrarán todos los datos referentes a los documentos, los autores y donde se comprobarán los datos de acceso a la administración.

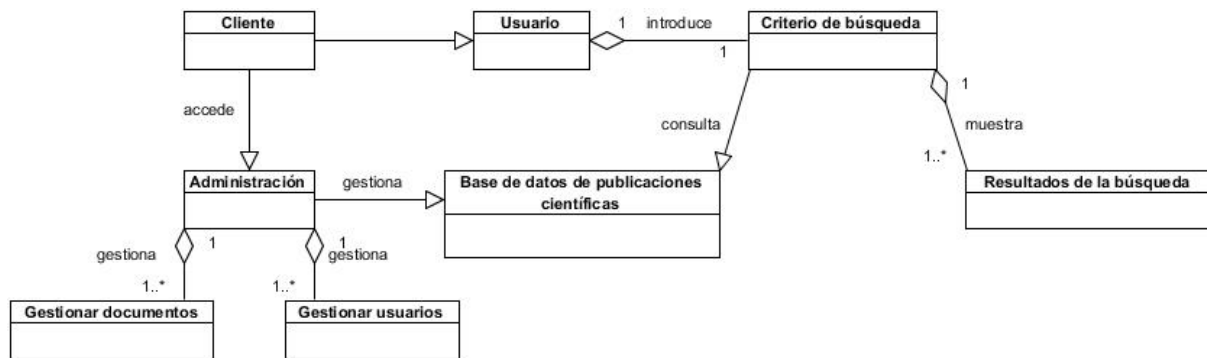


Ilustración 2: Modelo conceptual

2.3. Propuesta de solución

El presente trabajo propone adicionar nuevas funcionalidades al sistema RBC que le permitan al usuario introducir un documento al sistema y de este obtener los documentos similares existentes en la base de datos. Como resultado de la búsqueda se mostrará el autor y un resumen de los documentos obtenidos. El sistema también le permitirá al usuario analizar los documentos adquiridos y descargarlos.



Ilustración 3: Flujo de la propuesta de solución

El proceso de Minería de Texto es fundamental para el desarrollo de la solución propuesta, debido a la importancia que requiere el trabajo con datos, en este caso, con los documentos almacenados en la base de datos de publicaciones científicas y el documento subido por el usuario al sistema. El proceso permitirá obtener información relevante de cada uno de los

documentos, para de esta forma poder utilizarlos en tiempo de ejecución y definir similitud entre ellos. De esta forma el usuario podrá obtener los documentos similares al que posee.

A continuación se explica el proceso de minería de texto teniendo en cuenta las etapas que propone la metodología CRISP-DM:

Comprensión del negocio: en esta etapa se realiza una entrevista al cliente para conocer los requerimientos del componente a desarrollar desde una perspectiva del proceso de minería. Además se lleva a cabo un estudio de la herramienta RBC.

Comprensión de los datos: se obtienen los documentos científicos de fuentes de información, en este caso se selecciona el sitio web *Revista Cubana de Ciencias Informáticas* (disponible en <http://rcci.uci.cu>). Además se realiza una exploración de los documentos seleccionados y se comprueba la calidad de los mismos.

Preparación de los datos

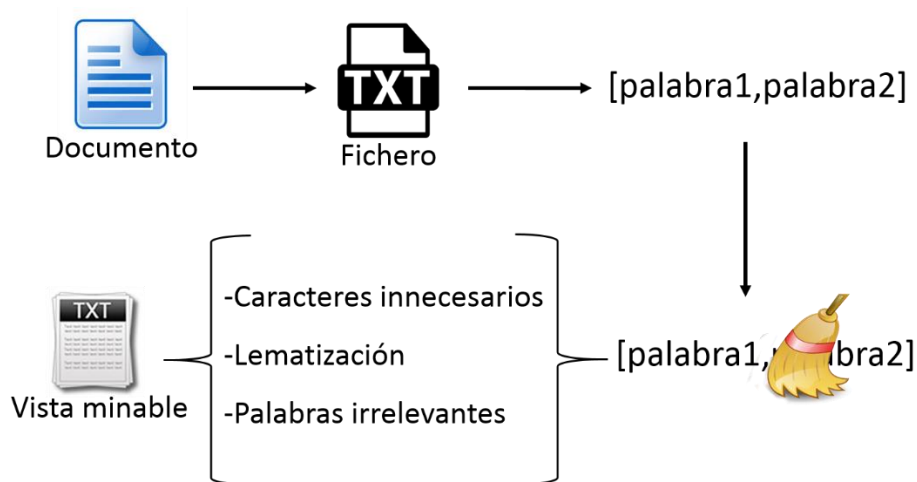


Ilustración 4: Preparación de los datos

Como muestra la imagen anterior, lo primero que se realiza en esta etapa es convertir el documento subido por el usuario en un fichero txt. A este fichero se le aplica un procesamiento del lenguaje natural, donde se tokeniza obteniendo un vector de palabras.

Al vector de palabras obtenido, se le realiza una limpieza del texto, donde se eliminan:

- Tildes (Ejemplo: técnicas = tecnicas).
- Mayúsculas (Ejemplo: Algoritmo = algoritmo).

- Sinónimos (Ejemplo: implementar = realizar).
- Caracteres innecesarios (Ejemplo: ¿, *, /, entre otros).
- Se eliminan palabras irrelevantes, que son las palabras que no aparecen en el diccionario (Ejemplo: los artículos, preposiciones, conjunciones, etc).
- Además se lematiza la palabra, donde (Ejemplo: desarrollando = desarrollar)

Al finalizar el procesamiento del lenguaje natural, se obtiene la vista minable del documento subido por el usuario, el cual no es más que un fichero txt limpio y preparado para la aplicación del algoritmo de agrupamiento seleccionado. Es importante destacar que este mismo procesamiento del lenguaje natural se aplicará a cada uno de los documentos almacenados en la base de datos.

Modelado

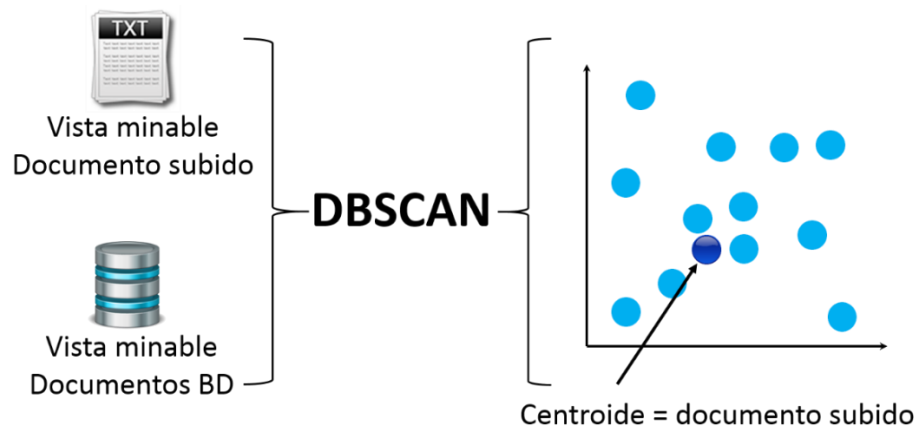


Ilustración 5: Aplicar minería

Una vez que se tiene la vista minable del documento subido por el usuario y la vista minable de cada uno de los documentos almacenados en la base de datos, se procede a aplicar el algoritmo seleccionado, en este caso el algoritmo de agrupamiento DBSCAN. Para agrupar los documentos por similitud, se toma como centroide¹⁷ el documento subido por el usuario.

¹⁷ Centro geométrico del cluster.

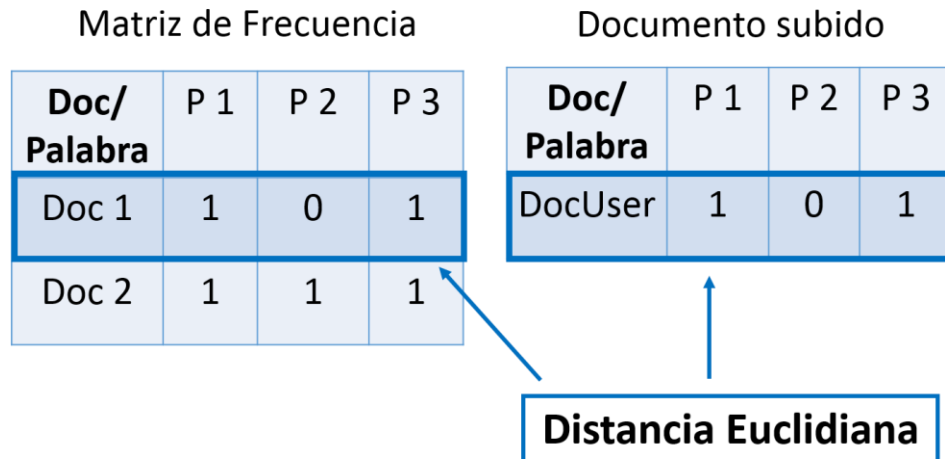


Ilustración 6: Procedimiento del algoritmo DBSCAN

El algoritmo DBSCAN trabaja con una matriz de frecuencia documento – término, donde se muestra en cada una de las filas los distintos documentos X, y en cada una de las columnas las palabras Y del documento subido por el usuario. En cada posición X, Y se encuentra la relevancia de la palabra Y en el documento X. Con esta misma estructura se encuentra el documento subido por el usuario.

Como se observa en la *Ilustración 6*, cada documento contiene un vector, por lo que se puede definir la distancia entre el vector del documento subido por el usuario con respecto a cada uno de los vectores de los documentos almacenados en la base de datos, haciendo uso de la Distancia Euclidiana.

A continuación, se muestra la fórmula matemática para calcular la distancia (similitud) entre dos documentos, donde:

- P y Q son los documentos a los cuales se les halla la similitud que existe entre ellos.
- p_n y q_n es la relevancia de la palabra n en los documentos P y Q.

$$d_E(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

Ilustración 7: Fórmula para calcular la similitud entre documentos

De esta manera se obtiene un cluster donde se define la similitud del documento subido por el usuario con respecto a cada uno de los documentos almacenados en la base de datos.

Evaluación

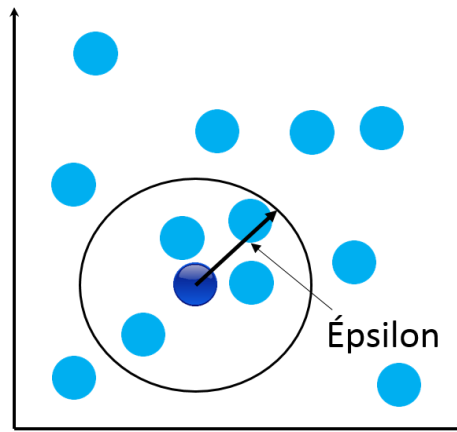


Ilustración 8: Variable Épsilon

Obtenido el cluster de documentos agrupados por similitud, se define la variable ϵ para determinar el conjunto de documentos similares al subido por el usuario al sistema. Para la selección del valor ϵ se emplea la Observación como método Empírico. Al aplicar este método, se tienen en cuenta los resultados obtenidos para diferentes valores de esta variable, comparándolos de forma tal que se pueda seleccionar aquel valor de la variable ϵ que permita obtener los documentos que más se asemejen al subido por el usuario al sistema.

Implementación: finalmente en esta etapa se presenta los documentos obtenidos de forma entendible para el usuario. Además, se analiza las mejoras que se puedan realizar a la herramienta y la posibilidad de incorporarle nuevas funcionalidades que permitan el crecimiento del sistema RBC.

2.4. Especificación de los requisitos del sistema

(Sommerville, 2005) plantea que: *“los requisitos del sistema especifican qué es lo que debe hacer (sus funciones) y sus propiedades esenciales y deseables”*. En esta etapa el principal objetivo es identificar las necesidades del cliente, con el fin de permitir que el equipo de desarrollo pueda trabajar según los requerimientos especificados y logre un producto terminado con calidad.

A continuación se especifican los requisitos funcionales y no funcionales del componente de búsqueda.

2.4.1. Requisitos Funcionales del sistema

Los requisitos funcionales de un software definen las funciones de la aplicación, es decir, los servicios que debe proporcionar el sistema.

Número	Nombre	Descripción
RF1	Subir documento	El sistema permitirá que el usuario pueda subir un documento.
RF2	Mostrar documentos similares	El sistema permitirá que el usuario obtenga los documentos similares al subido por él inicialmente.
RF3	Revisar documentos	El sistema permitirá que el usuario visualice el autor y el resumen de los documentos obtenidos.
RF4	Descargar documento	El sistema permitirá que el usuario descargue los documentos obtenidos.

Tabla 1: Requisitos funcionales del sistema

2.4.2. Requisitos no Funcionales

Los requisitos no funcionales describen las características del funcionamiento del sistema. Son características que permiten que la aplicación sea atractiva, usable, rápida y confiable.

Usabilidad

- La interfaz de la aplicación debe permitirle al usuario sin experiencia adaptarse rápidamente, para de esta forma poder interactuar fácilmente con el sistema.

Disponibilidad

- El sistema debe estar disponible para el usuario en el momento que lo necesite.
- Debe mantener su funcionamiento con la menor afectación posible en caso de que se presente algún error.

Rendimiento

- El funcionamiento del sistema debe ser estable.

Apariencia e Interfaz

- Todos los textos y mensajes en pantalla aparecerán en idioma español.
- La aplicación deberá poseer una interfaz fácil de usar por los usuarios.

Hardware

- Para las PC clientes:
 1. Requerimientos mínimos 512MB de RAM recomendada o superior.
 2. Tarjeta de red para establecer la conexión.
- Para el servidor:
 1. Computador con procesador Intel Xeon que es el tipo de microprocesador que utilizan los servidores de la Universidad de las Ciencias Informáticas (UCI), 4 GB de memoria RAM, 1 Tb de disco duro.
 2. Tarjeta de red para establecer la conexión.

Software

- Para las PC clientes:
 1. Sistema Operativo Windows XP, Windows 7 o Windows 8 y Linux con interfaz gráfica y soporte para conectarse a la red.
 2. Para la utilización del sistema se requerirá el uso de un navegador web Mozilla Firefox 30 o superior.
- Para el servidor:
 1. Se requiere el gestor de base de datos PostgreSQL 1.14.1, el marco de trabajo Django 1.6, las bibliotecas necesarias como nltk, gensim, sklearn, entre otras y Sistema Operativo Windows 7, Windows 8 o Linux.

2.5. Definición de los actores

Un actor es *“un agente, alguien o algo que solicita un servicio al sistema o actúa como catalizador para que ocurra algo. Un actor representa un rol que es jugado por una persona, un dispositivo hardware, incluso otro sistema.”* (Merseguer, 2010)

Actor	Objetivo
Usuario	El usuario podrá subir un documento al sistema, revisar si el documento subido es realmente el deseado, realizar la búsqueda y observar los resultados con

	documentos similares al documento subido. Además podrá descargar los documentos de su interés.
--	--

Tabla 2: Actor relacionado con el sistema

2.6. Diagrama de casos de uso del sistema

El diagrama de casos de uso permite describir la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema.

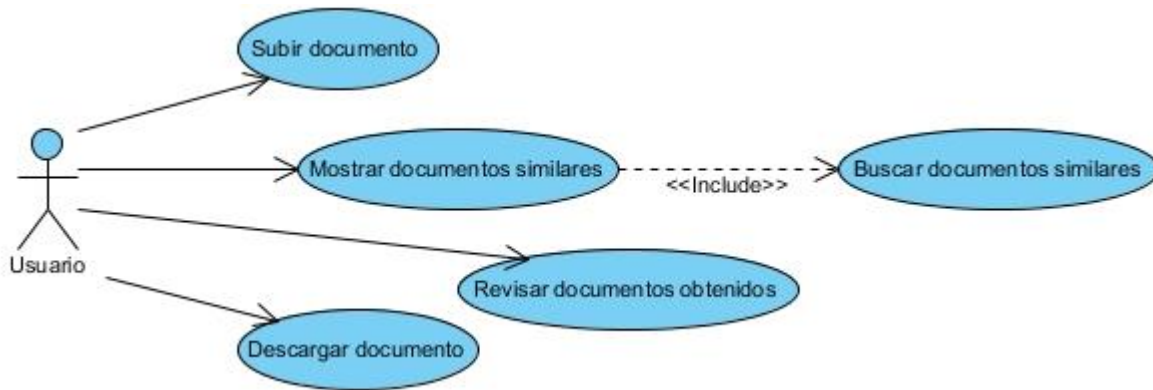


Ilustración 9: Diagrama de casos de uso del sistema

2.6.1. Descripción de los casos de uso del sistema

CU1: Subir documento

Objetivo	Subir un documento para obtener los documentos similares a él.
Actores	Usuario
Resumen	El caso de uso inicia cuando el usuario desea subir un documento y selecciona la opción Buscar Documentos Similares.
Complejidad	Baja
Prioridad	Alta
Precondiciones	Seleccionar el documento a subir.
Postcondiciones	Se sube el documento y queda almacenado en la carpeta Carga ubicada en el servidor.
Flujo de eventos	
Flujo básico Subir documento	
Actor	Sistema

1. Selecciona la opción Buscar Documentos Similares.	
2.	Brinda la opción Examinar.
3. Selecciona la opción Examinar.	
4.	Brinda la opción de seleccionar el documento que se desea subir.
5. Selecciona el documento y da click en el botón Abrir.	

Prototipo elemental de interfaz gráfica de usuario de la funcionalidad Subir documento

The screenshot shows a web interface with a navigation bar at the top containing 'Administración' and 'Información'. The main content area features a logo for 'RBC' (Física, Matemática, Computación) and the title 'REPRESENTACIÓN DE BIBLIOGRAFÍAS CIENTÍFICAS'. Below the title is a search bar with a 'Filtrar' button and a magnifying glass icon. A yellow button labeled 'Buscar Documentos Similares' is positioned below the search bar. A tooltip box below the button contains the text: 'Esta opción permite subir un documento y obtener documentos similares'.

REPRESENTACIÓN DE BIBLIOGRAFÍAS CIENTÍFICAS

Seleccione un Documento (doc, docx, pdf):

Examinar... No se ha seleccionado ningún archivo.

Buscar Documentos Similares

Carga de archivos

Nombre	Fecha de modifica...	Tipo	Tamaño
Convertidos	24/05/2016 1:59	Carpeta de archivos	
PID4	24/05/2016 3:12	Carpeta de archivos	
13275929_181250538941725_2101402678_n	25/05/2016 2:54	Archivo JPG	65 KB
13281979_181251638941615_730891343_n	25/05/2016 2:55	Archivo JPG	82 KB
13285585_181251425608303_1737345700_n	25/05/2016 2:55	Archivo JPG	87 KB
13288892_181264042273708_624911065_n	25/05/2016 3:31	Archivo JPG	107 KB
13292820_181262972273815_523216292_n	25/05/2016 3:31	Archivo JPG	79 KB
clase	26/05/2016 21:39	Presentación Ope...	324 KB
Comparación de metaheurísticas para obtener predicados difusos un caso curioso	24/05/2016 1:16	Documento de tex...	1 KB
Comparacion_de_metaheurísticas_para_obtener_predicados_difusos_un_caso_curioso	24/05/2016 1:16	Archivo PDF	745 KB
Computación con palabras en la toma de decisiones mediante mapas cognitivos difusos	24/05/2016 1:12	Archivo PDF	569 KB
Computación con palabras en la toma de decisiones mediante mapas cognitivos difusos	24/05/2016 1:13	Documento de tex...	2 KB
Desarrollo de un driver Linux para sistemas de adquisición de datos embebidos	24/05/2016 1:14	Archivo PDF	379 KB
Desarrollo de un driver Linux para sistemas de adquisición de datos embebidos	24/05/2016 1:14	Documento de tex...	1 KB
Directrices prácticas y métricas de calidad en la modelación de procesos de negocio un caso de estudio	24/05/2016 1:11	Archivo PDF	850 KB
Directrices prácticas y métricas de calidad en la modelación de procesos de negocio un caso de estudio	24/05/2016 1:12	Documento de tex...	2 KB
Framework basado en MDA y ontologías para la representación y validación de modelos de componentes	24/05/2016 1:17	Archivo PDF	409 KB
Framework basado en MDA y ontologías para la representación y validación de modelos de componentes	24/05/2016 1:17	Documento de tex...	1 KB
Intérprete de programas de usuario para el cálculo de parámetros petrofísicos	24/05/2016 1:18	Archivo PDF	312 KB
Intérprete de programas de usuario para el cálculo de parámetros petrofísicos	24/05/2016 1:19	Documento de tex...	2 KB
La toma de decisiones en los Sistemas Tutoriales Inteligentes utilizando el agrupamiento conceptual	24/05/2016 1:10	Archivo PDF	315 KB
La toma de decisiones en los Sistemas Tutoriales Inteligentes utilizando el agrupamiento conceptual	24/05/2016 1:11	Documento de tex...	2 KB
limpiarwikis	26/05/2016 21:39	JetBrains PyCharm	2 KB
Metodología para la localización del disco óptico	24/05/2016 1:17	Archivo PDF	312 KB
Metodología para la localización del disco óptico	24/05/2016 1:16	Documento de tex...	3 KB

CU2: Mostrar documentos similares

Objetivo	Obtener los documentos similares al subido al sistema.
Actores	Usuario
Resumen	El caso de uso inicia cuando el usuario selecciona la opción Buscar Documentos Similares.

Complejidad	Baja
Prioridad	Alta
Precondiciones	Haber seleccionado la opción Buscar Documentos Similares.
Postcondiciones	Muestra una página con el resultado de la búsqueda.
Flujo de eventos	
Flujo básico Mostrar documentos similares	
Actor	Sistema
1. Ejecuta el <u>CU4: Buscar documentos similares.</u>	
2.	Muestra una página con el resultado de la búsqueda.
Flujos alternos	
2a. En caso de no encontrarse documentos similares.	
Actor	Sistema
1.	Muestra el mensaje “No se encontraron Documentos Similares”.
Prototipo elemental de interfaz gráfica de usuario de la funcionalidad Mostrar documentos similares	

The screenshot shows a web interface with a search bar and a download button. The search bar contains the text "Comparacion de metaheurísticas para obtener predicados difusos un caso curioso" and a "Descargar" button. The interface also features a logo for "RBC" (Física, Matemáticas, Computación) and a "Filtrar" button.

CU3: Buscar documentos similares

Objetivo	Buscar los documentos similares al subido al sistema.	
Actores	Usuario	
Resumen	El caso de uso inicia cuando el usuario selecciona la opción Buscar Documentos Similares.	
Complejidad	Baja	
Prioridad	Alta	
Precondiciones	Haber subido un documento al sistema.	
Postcondiciones	Redirecciona a la página con los resultados de la búsqueda.	
Flujo de eventos		
Flujo básico Buscar documentos similares		
Actor	Sistema	
1. Da click sobre el botón Buscar Documentos Similares.		
2.	Redirecciona a la página con los resultados de la búsqueda.	

Flujos alternos

2a. En caso de dar click en el botón Buscar Documentos Similares y no haber subido anteriormente un documento.

Actor	Sistema
1.	Muestra el mensaje de error "Este campo es obligatorio".

2b. En caso de dar click en el botón Buscar Documentos Similares y haber subido un archivo que no es de formato doc, docx o pdf.

Actor	Sistema
1.	Muestra el mensaje de error "El formato del documento no es válido".

Prototipo elemental de interfaz gráfica de usuario de la funcionalidad Buscar documentos similares





REPRESENTACIÓN DE BIBLIOGRAFÍAS CIENTÍFICAS

- Este campo es obligatorio.

Seleccione un Documento (doc, docx, pdf):

Examinar... No se ha seleccionado ningún archivo.

Buscar Documentos Similares



REPRESENTACIÓN DE BIBLIOGRAFÍAS CIENTÍFICAS

- El formato del documento no es valido

Seleccione un Documento (doc, docx, pdf):

Examinar... No se ha seleccionado ningún archivo.

Buscar Documentos Similares

CU4: Revisar documentos

Objetivo	Revisar los documentos obtenidos en la búsqueda a través del autor y el resumen del documento.
Actores	Usuario
Resumen	El caso de uso inicia cuando el usuario da click sobre un documento obtenido.


Complejidad	Baja
Prioridad	Media
Precondiciones	Haber obtenidos al menos un documento en la búsqueda.
Postcondiciones	Resultan revisados los documentos obtenidos.


Flujo de eventos

Flujo básico Revisar documentos

Actor	Sistema
1. Da click sobre un documento obtenido.	
2.	Despliega el autor y el resumen de documento seleccionado.

Prototipo elemental de interfaz gráfica de usuario de la funcionalidad Revisar documentos



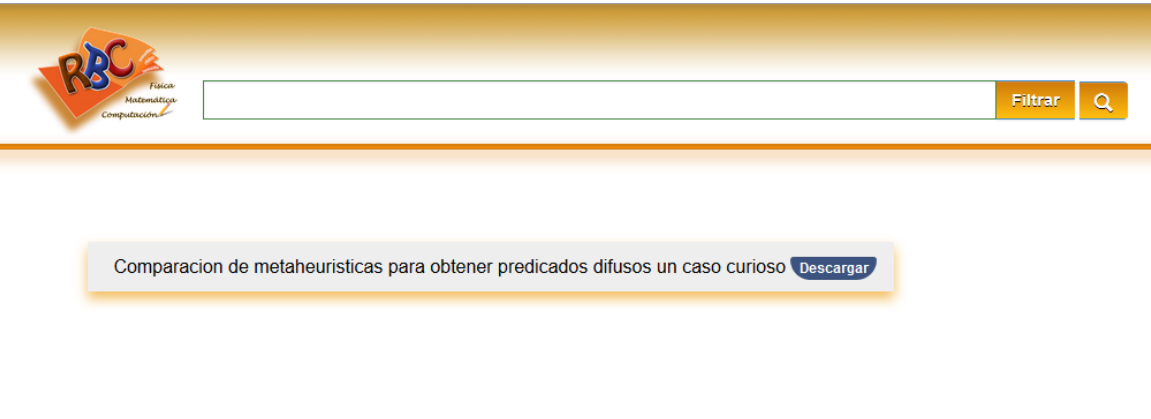
Filtrar 

Comparacion de metaheurísticas para obtener predicados difusos un caso curioso [Descargar](#)

- o Autor :Taymi Ceruto-Cordovés
- o Resumen :En este trabajo se presenta un estudio comparativo de tres metaheurísticas en el problema de obtener predicados difusos con alto valor de verdad. Según el Teorema No Free Lunch (NFL) no se puede establecer la superioridad general de ninguna metaheurística sobre las otras. Este trabajo demuestra que incluso dentro del mismo tipo de problema, puede ser difícil de establecer la superioridad de una metaheurística. En este caso, cada metaheurística logra ser la mejor en al menos en una de las cuatros variantes de operador difuso empleado y de forma normal del predicado obtenido. Este curioso caso, revela la importancia de la comparación experimental de metaheurísticas, antes de asumir la superioridad de una sobre las otras.

CU4: Descargar documento

Objetivo	Descargar los documentos obtenidos.
Actores	Usuario
Resumen	El caso de uso inicia cuando el usuario selecciona la opción Descargar.
Complejidad	Baja

Prioridad	Media	
Precondiciones	Haber obtenido al menos un documento en la búsqueda.	
Postcondiciones	Se descargan los documentos deseados por el usuario.	
Flujo de eventos		
Flujo básico Descargar documento		
Actor	Sistema	
1. Selecciona la opción Descargar.		
2.	Permite guardar e documento.	
Prototipo elemental de interfaz gráfica de usuario de la funcionalidad Descargar documento		
		

2.7. Conclusiones parciales

En este capítulo se definieron las características de la propuesta de solución. Se especificó el modelo conceptual, que permitió definir los conceptos fundamentales del negocio. De esta forma se logró obtener una visión más clara del entorno sobre el cual se sitúa el problema a resolver. Se definieron los requisitos funcionales y los requisitos no funcionales del sistema. Se modeló el diagrama de casos de uso del sistema y se realizó la especificación de cada caso de uso, permitiendo una mejor comprensión del flujo básico de estos.

Capítulo 3: Arquitectura y diseño

3.1. Introducción

En este capítulo se describe el diseño del componente propuesto para el sistema RBC. Se muestran los diagramas de clases del diseño empleando estereotipos web, los diagramas de colaboración, el diagrama de paquetes y el diseño de la base de datos. Se define el estilo arquitectónico y los patrones de diseño utilizados.

3.2. Diseño de la arquitectura

Una arquitectura de software *“constituye un modelo comprensible de cómo está estructurado el sistema y cómo trabajan juntos sus componentes.”* (Cervantes, 2010)

Para el desarrollo del sistema se empleará la arquitectura cliente-servidor ya que la aplicación web estará alojada en el servidor de aplicaciones y el usuario podrá acceder a la misma desde cualquier PC cliente.

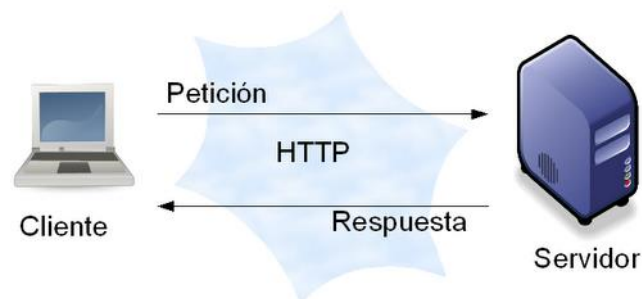


Ilustración 10: Arquitectura Cliente – Servidor.

El framework Django define una modificación de la arquitectura Modelo-Vista-Controlador (MVC), llamada MTV (Model-Template-View), que sería Modelo-Plantilla-Vista. El modelo en Django sigue siendo Modelo, la vista se llama plantilla (Template) y el controlador se llama Vista.

Capa Vista (views): Determina qué datos serán visualizados sin encargarse del estilo de la presentación de los mismos.

Capa Modelo (models): Define los datos almacenados, se encuentra en forma de clases de Python, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos

parámetros, posee métodos también. Todo esto permite indicar y controlar el comportamiento de los datos.

Capa Plantilla (templates): La plantilla recibe los datos de la vista y luego los organiza para la presentación al navegador web.

La imagen siguiente muestra el funcionamiento del patrón arquitectónico MTV en Django.

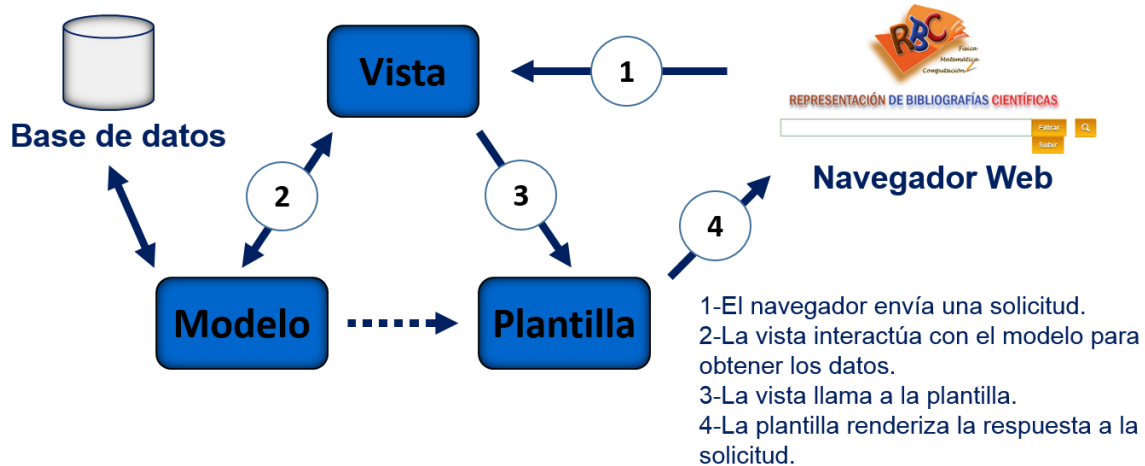


Ilustración 11: Funcionamiento del MTV en Django.

3.3. Etapa de Diseño

En esta etapa se especifican los elementos necesarios para lograr la correcta implementación de la solución propuesta, como son la arquitectura del sistema y la especificación de patrones de su diseño. También se definen los estándares de codificación a utilizar durante el desarrollo para describir las clases y funcionalidades. Se muestra el diagrama de despliegue y el diagrama de clases del diseño.

3.3.1. Diagrama de paquetes

El diagrama de paquetes permite un mejor entendimiento del sistema, organizando el mismo a través de paquetes y sus relaciones, conformando así una estructura lógica del sistema.

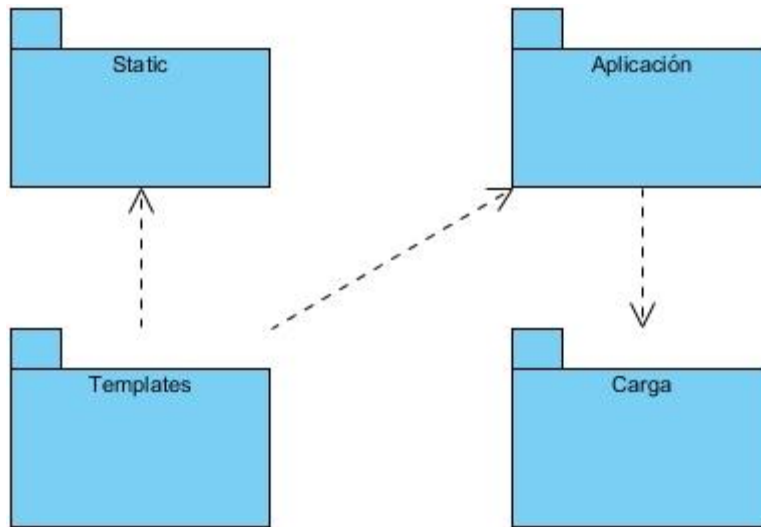


Ilustración 12: Diagrama de paquetes

- **Static:** este paquete contiene los archivos CSS, JavaScript y las imágenes utilizadas para el diseño y estilo de las plantillas HTML.
- **Templates:** contiene todas las plantillas HTML.
- **Aplicación:** este paquete contiene las nuevas clases y funcionalidades implementadas.
- **Carga:** contiene dos subdirectorios, uno llamado Documentos donde se almacena todos los documentos en distintos formatos que se encuentran en la base de datos. El otro subdirectorios es Carga el cual contiene el documento subido por el usuario.

3.3.2. Diagrama de clases del diseño utilizando estereotipos web

Los diagramas de clases son una estructura estática donde la representación de los requisitos se lleva a cabo a través de las clases del sistema y sus interrelaciones.

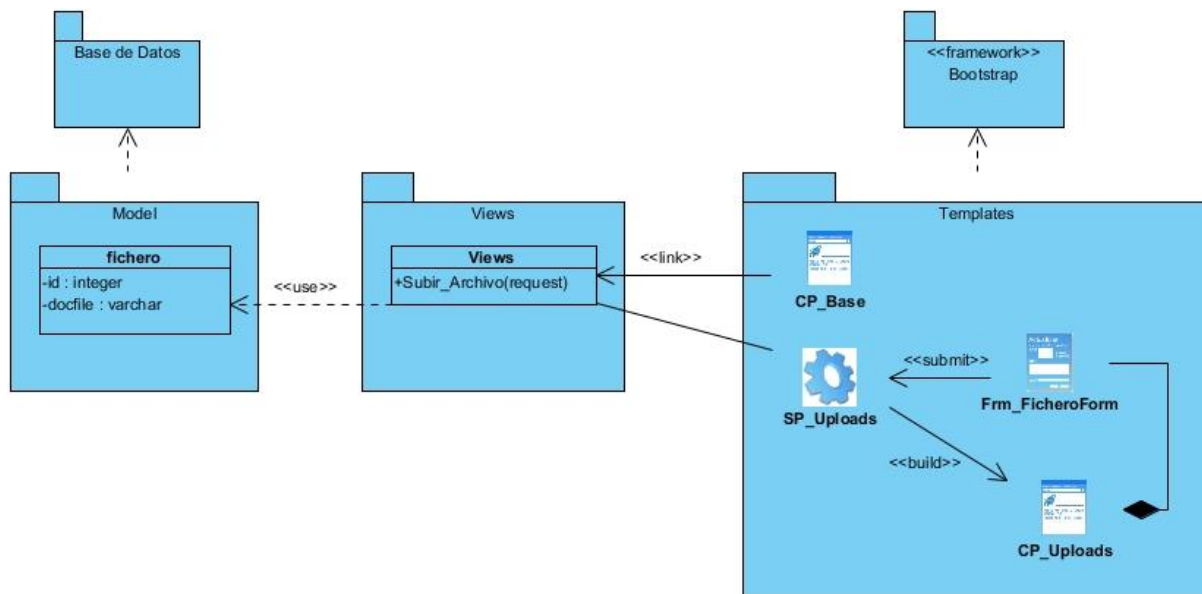


Ilustración 13: Diagrama de clases del diseño Subir documento

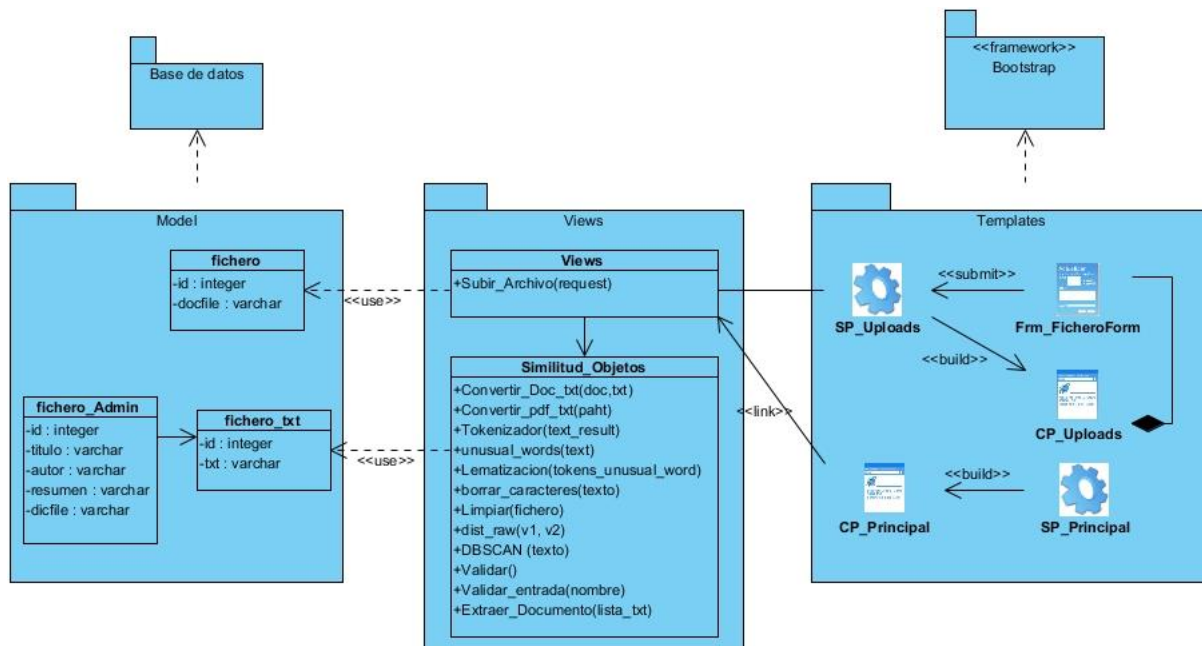


Ilustración 14: Diagrama de clases del diseño Mostrar documentos similares

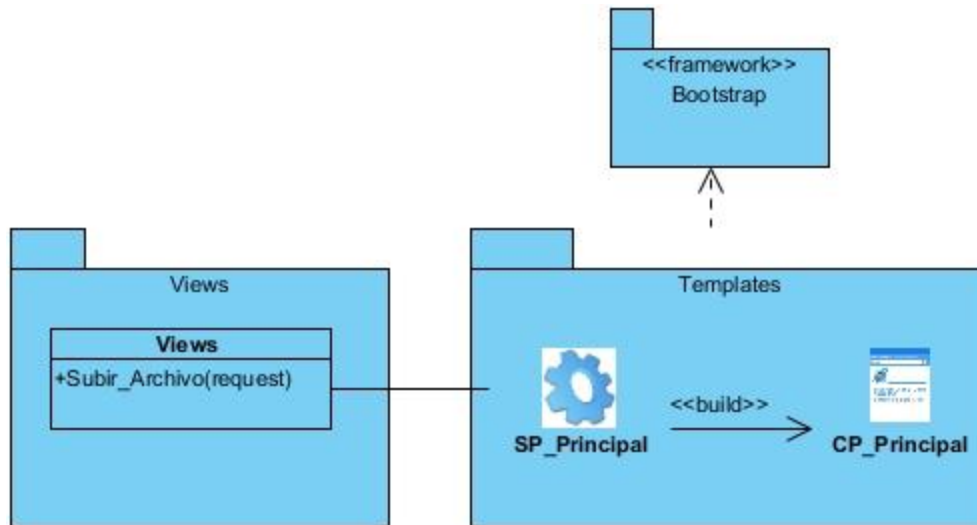


Ilustración 15: Diagrama de clases del diseño Revisar y Descargar documentos obtenidos

3.3.3. Diagrama de Interacción

El diagrama de interacción “*proporciona la información para entender la dinámica del modo en el que se conectan y comunican los objetos entre las capas. Ilustran los escenarios más significativos desde el punto de vista de la arquitectura.*” (Larman, 1999)

El UML define dos tipos de estos diagramas: diagrama de colaboración y de secuencia.

Diagramas Colaboración: “*Con el uso de un diagrama de colaboración se muestra la implementación de una operación. La colaboración muestra los parámetros y las variables locales de la operación, así como asociaciones más permanentes*” (Rumbaugh, y otros).

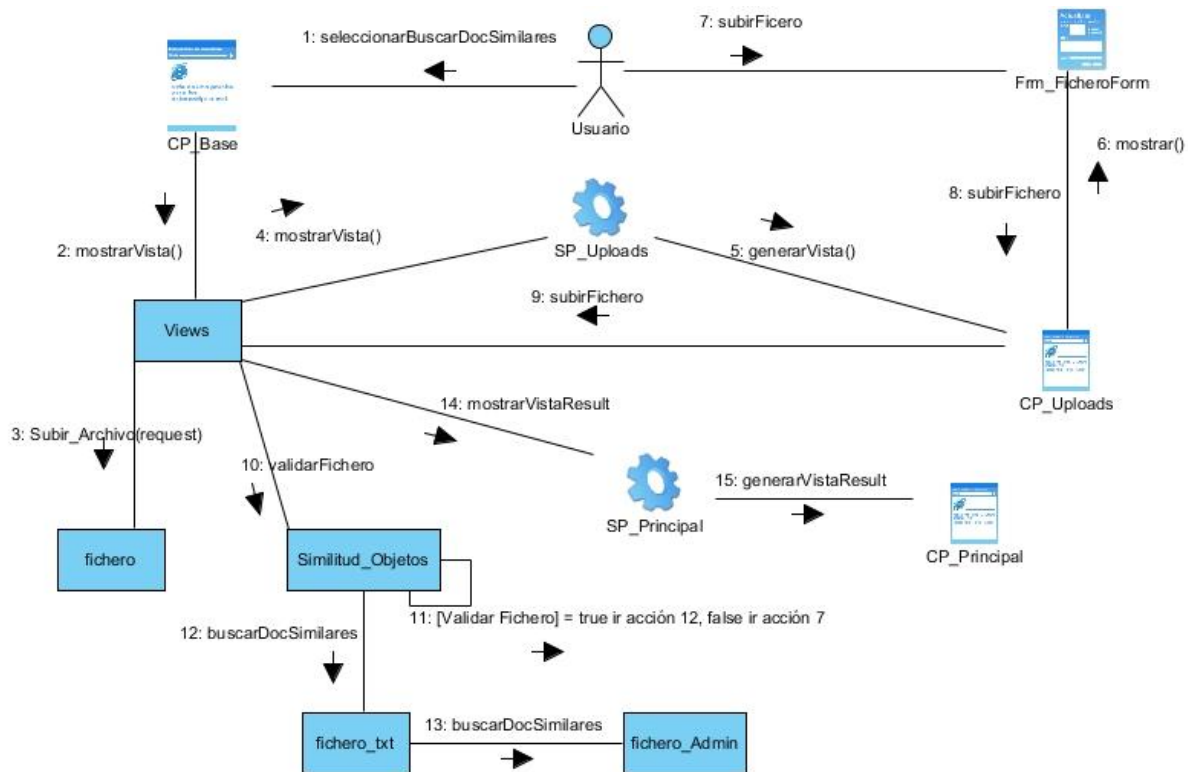


Ilustración 16: Diagrama de colaboración de los RF: Subir documento y Mostrar documentos similares

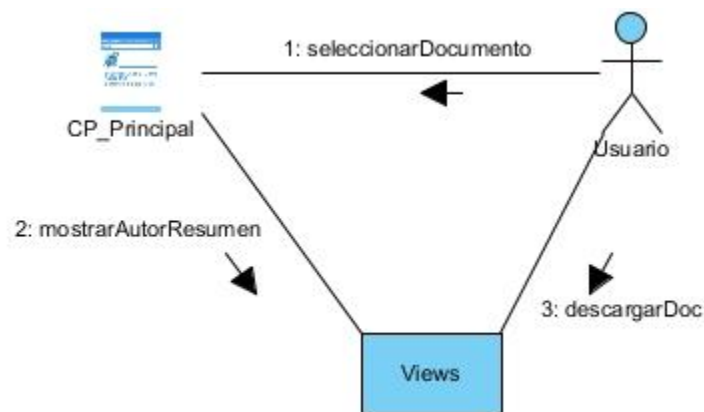


Ilustración 17: Diagrama de colaboración de los RF: Revisar documentos y Descargar documento

3.4. Modelo físico de los datos

Describe las representaciones físicas de los datos utilizados en la herramienta RBC y que serán almacenados en base de datos. Los elementos esenciales del diagrama son las entidades, los atributos y las relaciones entre las entidades.

- Entidades: objetos que el sistema necesita guardar su información, en este caso, las entidades son los documentos y los vectores.
- Atributos: características de las entidades, las cuales se clasifican en obligatorias, opcionales, claves foráneas y claves primarias.
- Relaciones: muestra como dos entidades se relaciona, en este caso, a cada documento lo identifica un vector.

Para el desarrollo de las nuevas funcionalidades propuestas se utiliza la tabla *fichero_Admin* de la base de datos que posee el sistema RBC inicialmente, además se crea la tabla *fichero_txt*.

fichero_Admin: contiene toda la información referente a los documentos almacenados que serán comparados con el documento que suba el usuario.

fichero_txt: en esta tabla contiene información de cada documento almacenado pero en formato txt.

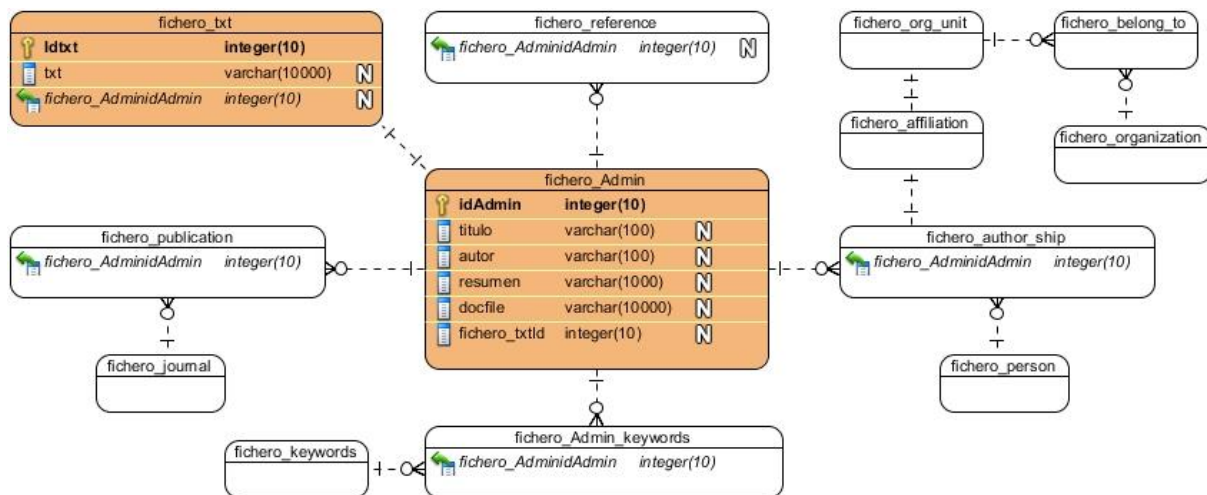


Ilustración 18: Modelo físico de la base de datos

3.5. Patrones de diseño

Los patrones de diseño “brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares.” (Tedeschi, 2010)

Patrones GRASP

Patrones Generales de Software para Asignación de Responsabilidades (GRASP¹⁸, siglas en inglés). Los patrones GRASP “*describen los principios fundamentales de la asignación de responsabilidades a objetos expresados en forma de patrones*”. (Larman, 1999).

Existen 9 patrones GRASP: experto, creador, controlador, bajo acoplamiento, alta cohesión, polimorfismo, fabricación pura, indirección y variaciones protegidas. Django que es el framework que se utiliza para el desarrollo de las nuevas funcionalidades del sistema RBC, implementa 3 de los patrones antes mencionados (experto, bajo acoplamiento y alta cohesión).

Experto: “*Es el encargado de asignar la responsabilidad de la creación de un objeto o la implementación de un método a una clase que contenga toda la información necesaria para cumplir con dicha responsabilidad*”. (Pressman, 2005)

- Ejemplo: el *models.py* se encarga de la estructuración de la base de datos y la lógica de la misma.

Bajo acoplamiento: “*El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con qué las conoce y con qué recurre a ellas. En tal sentido, el término bajo acoplamiento significa que una clase no depende de muchas clases*”. (Pressman, 2005)

- Ejemplo: Las URLs llaman a funciones y métodos que implementan las Vistas, pero algún cambio que se realice a una función o método, no afecta la URL.

Alta cohesión: “*Es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realizan un trabajo enorme*”. (Pressman, 2005)

- Ejemplo: las clases del modelo describen la estructura de las tablas de la base de datos de manera coherente y precisa.

¹⁸ GRASP, acrónimo de General Responsibility Assignment Software Patterns.

Otro patrón que se evidencia en el desarrollo de las nuevas funcionalidades, es el patrón **Creador**, el mismo tiene como objetivo *“asignar a la clase B la responsabilidad de crear una instancia de clase A”* (Larman, 1999).

- Ejemplo: el formulario FicheroForm es responsable de la creación de objetos de tipo formulario.

3.6. Conclusiones Parciales

En el presente capítulo se planteó el diseño de la herramienta RBC y el patrón arquitectónico MTV (Modelo-Plantilla-Vista). Se realizó una breve descripción de los patrones de diseño utilizados. Se modelaron los diagramas de clases del diseño con estereotipos web, de colaboración, de paquete, y el modelo físico de la base de datos, todo esto para lograr un mejor entendimiento del funcionamiento del sistema RBC.

Capítulo 4: Implementación y Prueba

4.1. Introducción

En este capítulo se describen aspectos relacionados con la implementación y validación del componente desarrollado. Se exponen los principales resultados obtenidos durante la etapa de pruebas, para garantizar su correcto funcionamiento y el cumplimiento con los requisitos definidos por el cliente.

4.2. Etapa de implementación

En esta etapa “...se realiza la codificación, se realiza la programación de la solución diseñada, en el lenguaje de programación y plataforma elegida a tal efecto teniendo en cuenta las restricciones obtenidas en la etapa de análisis”. (González, 2007)

4.2.1. Diagrama de componente

Un componente “...representa un módulo físico de código o paquete, puede ser relacionado en un diagrama de componentes. Un diagrama de componentes muestra varios componentes de un sistema, describiendo sus elementos físicos y sus dependencias”. (Jacobson, y otros, 2000)

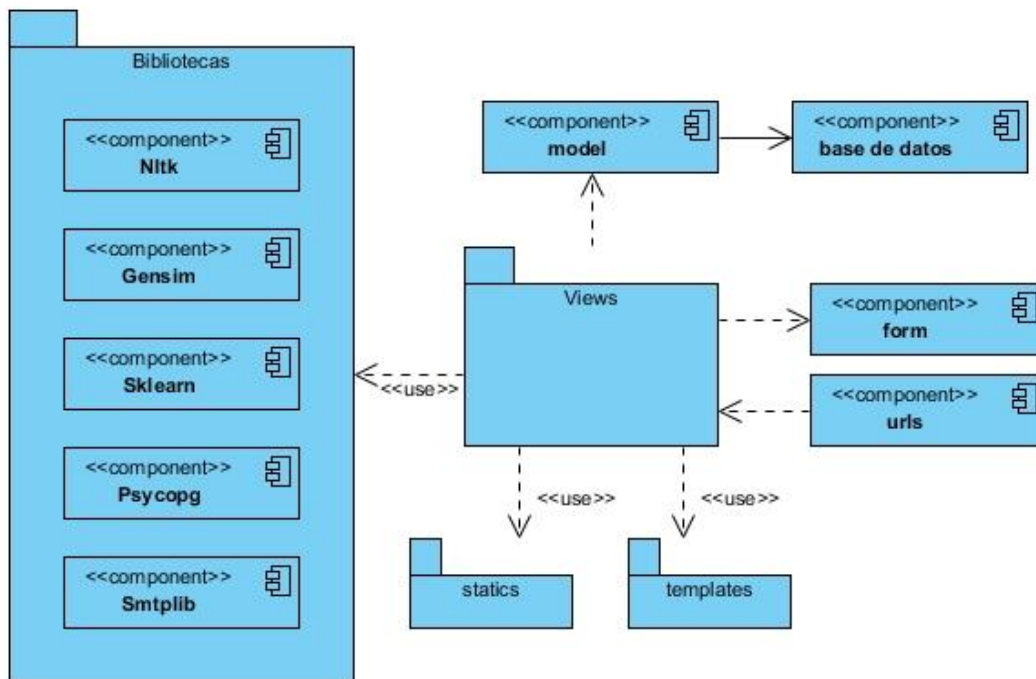


Ilustración 19: Diagrama de componentes del sistema

4.3. Estándares de codificación empleados

Los estándares de codificación permiten establecer una forma de programación única que haga más entendible el código fuente para otros desarrolladores, facilitando el mantenimiento posterior de las aplicaciones o software desarrollados. Estos estándares definen entre otras, la manera en que se debe indentar¹⁹, declarar las variables y funciones, así como las formas en que se van a escribir los comentarios, etc. Con el objetivo de alcanzar una uniformidad en el código, a continuación se muestran algunos estándares de codificación utilizados.

- Iniciar el nombre de las variables de las clases con letra inicial mayúscula y, en caso de ser un nombre compuesto se utiliza underscore (_).
- Al inicio de cada clase se realiza un comentario explicando el objetivo de la misma. Cada línea de un comentario empieza con # (numeral).
- Antes y después de la declaración de una clase o de una estructura y de la implementación de una función se deja una línea en blanco.
- Entre operadores lógicos y aritméticos se utiliza espacios en blanco.

4.4. Diagrama de despliegue

“Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Este muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos” (SOLUS S.A., 2000-2016). Este diagrama modela la topología del hardware del entorno donde se debe ejecutar el sistema, el software necesario para su funcionamiento y los protocolos de comunicación.

¹⁹ Notación secundaria utilizada para mejorar la legibilidad del código fuente por parte de los programadores, teniendo en cuenta que los compiladores o intérpretes raramente consideran los espacios en blanco entre las sentencias de un programa.

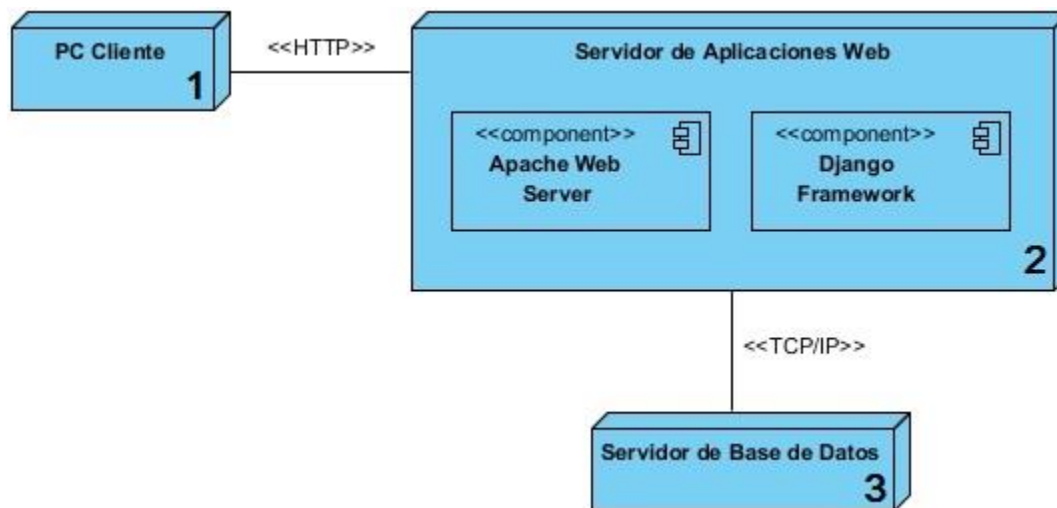


Ilustración 20: Diagrama de Despliegue

1. **PC Cliente:** representa las PC clientes mediante las que se accederá al sistema disponible en el servidor de aplicaciones.
2. **Servidor de Aplicaciones Web:** es el servidor donde estará disponible el sistema, además debe estar instalado el lenguaje de programación Python 2.7, el framework Django 1.6 y el servidor web Apache.
3. **Servidor de Base de Datos:** almacena los datos con los que interactúa el sistema.

Descripción de los protocolos de comunicación:

HTTP: El Protocolo de Transferencia de Hipertexto (HTTP, siglas en inglés) es un sencillo protocolo cliente - servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP (Guijarro, 2012). HTTP se basa en sencillas operaciones de solicitud/respuesta. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado (Guijarro, 2012).

TCP/IP: El Protocolo de Control de Transmisión/Protocolo de Internet (TCP/IP, siglas en inglés) se utiliza para enlazar computadoras que usan sistemas operativos similares o diferentes, incluyendo pc, minicomputadoras y computadoras centrales sobre redes de área local (LAN). En el caso del sistema DBC se utiliza para interconectarlo con el servidor de base de datos.

4.5. Pruebas

Luego de la implementación de la solución, se realizan un conjunto de actividades para verificar la calidad del producto y su cumplimiento con los requisitos definidos. El objetivo de esta fase es detectar y solucionar los errores que presenta el componente desarrollado y perfeccionar la solución implementada.

Las pruebas son *“un componente importante de calidad del software..., proceso de ejecutar un programa para detectar errores”*. (Jovanović, 2008)

4.5.1. Métodos de prueba

Los métodos de pruebas definen estrategias para descubrir fallos en el sistema. (Pressman, 2005) propone dos métodos de pruebas: caja blanca y caja negra.

Pruebas de caja blanca

“Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que: garanticen que se ejercite por lo menos una vez todos los caminos independientes de cada módulo; ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa; ejecuten todos los bucles en sus límites y con sus límites operacionales, y que se ejerciten las estructuras internas de datos para asegurar su validez” (Pressman, 2010). Estas pruebas se realizan al código fuente para asegurar que la operación interna se ajuste a las especificaciones.

Pruebas de caja negra

“Las pruebas de caja negra también conocidas como pruebas funcionales o pruebas de entrada y salida, son las que se ejecutan sobre la interfaz del software. Mediante el uso de estas se examinan todas las funcionalidades. Estas tienen poca relación con el comportamiento interno del software” (Pressman, 2010). Estas pruebas permiten demostrar que las funciones del sistema sean operativas, que la entrada se acepte de forma adecuada y que se produce una salida correcta.

4.5.2. Estrategia de prueba seguida

La estrategia seguida para la realización de las pruebas al componente de búsqueda desarrollado para el sistema RBC, comprende pruebas de los 4 niveles propuestos por Pressman en la sexta edición del libro *“Ingeniería de software. Un enfoque práctico”*:

Pruebas unitarias: son pruebas de caja blanca que se realizan con el objetivo de detectar errores de implementación en el componente desarrollado. Además se identifica errores de entrada o salida de datos.

Pruebas de integración: verifican que cada componente desarrollado no presente errores cuando se integre con los demás.

Pruebas de validación: son pruebas de caja negra que se enfocan en la satisfacción de las necesidades del cliente, verificado las acciones que el usuario realiza en el sistema y la correcta entrada y salida de datos.

Pruebas del sistema: son pruebas que confirman el correcto funcionamiento de las funciones desarrolladas.

4.5.2.1. Pruebas unitarias

Para comprobar el correcto funcionamiento de la implementación del componente desarrollado, se realizaron pruebas unitarias. Se utilizó la biblioteca de Python (*unit testing*) y la definición de las pruebas se realizaron dentro del archivo *tests.py*, ya creado por el marco de trabajo Django. Para ejecutar los test o iniciar el servidor de prueba se hace uso del comando “python manage.py test”.

Se realizaron pruebas unitarias en **models** a la clase *fichero* y *fichero_txt*, y en el **views** en los métodos *vectores* y *borrar_caracteres*.

```
class Test_models(unittest.TestCase):
+   def test_fichero(self):...
+   def test_Ficheros_TXT(self):...

class Test_views(unittest.TestCase):
+   def test_vectores(self):...
+   def test_borrar_caracteres(self):...
```

Ilustración 21: Pruebas unitarias

```
Ran 4 tests in 0.014s
OK
Destroying test database for alias 'default'...
```

Ilustración 22: Resultado de las pruebas unitarias

4.5.2.2. Pruebas de integración

Se realizaron pruebas de integración para comprobar que el buscador por palabras claves y el buscador por similitud de objetos desarrollado en la presente investigación, funcionan correctamente y transfieren los datos correctos en el tiempo preciso a través de sus interfaces.

4.5.2.3. Pruebas de validación

Las pruebas de validación se realizaron a través de casos de prueba, los cuales son “...un conjunto de acciones con resultados y salidas previstas basadas en los requisitos de especificación del sistema”. (Aristegui O, 2010). Las técnicas utilizadas para comprobar el funcionamiento del sistema son **partición equivalente** y **gráfico de prueba**.

“La *partición equivalente* es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba”. (Ruiz Tenorio, 2010). Esta técnica se utilizó para representar los diferentes estados posibles en cada condición de entrada, calificando cada estado en válidos o inválidos.

Otra de las técnicas empleadas en el desarrollo de las pruebas de caja negra es el gráfico de prueba, basada en “una colección de nodos que representan objetos, enlaces que representan la relación entre objetos, pesos de nodo que describen las propiedades de un nodo (como un valor de datos o un comportamiento de estado específico) y pesos de enlace que describen algunas características de un enlace”. (Pressman, 2005). Su utilización permite validar funcionalidades que no requieren entrada de datos.

A continuación, se muestra los casos de prueba realizados para verificar el cumplimiento de los requisitos funcionales, utilizando las técnicas antes mencionadas.

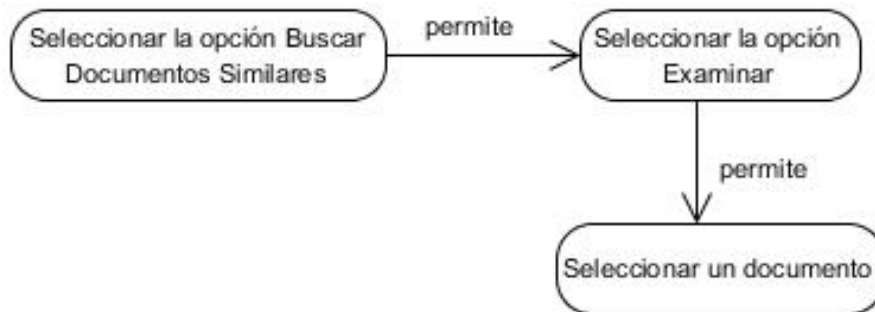


Ilustración 23: Gráfico correspondiente al caso de prueba del RF1: Subir documento

Escenario	Descripción	Documento	Respuesta del sistema	Flujo central
EC 2.1 Seleccionar la opción Buscar Documentos Similares y haber subir un documento de formato permitido	El usuario sube un documento con un formato permitido y selecciona la opción Buscar Documentos Similares.	V documento.doc V documento.docx V documento.pdf	El documento subido queda almacenado en la carpeta Carga ubicada en el servidor y se muestra una página con el resultado de la búsqueda.	El usuario sube un documento y selecciona la opción Buscar Documentos Similares.
EC 2.2 Seleccionar la opción Buscar Documentos Similares y haber subir un documento de formato no permitido	El usuario sube un documento con un formato no permitido y selecciona la opción Buscar Documentos Similares.	I documento.ppt	Muestra un el mensaje “El formato del documento no es válido”.	

EC	2.3	El usuario no sube un documento y selecciona la opción Buscar Documentos Similares.	N/A	Muestra el mensaje “Este campo es obligatorio”.	El usuario no sube un documento y selecciona la opción Buscar Documentos Similares.
----	-----	---	-----	---	---

Tabla 3: Caso de prueba de partición equivalente del RF2: Mostrar documentos similares

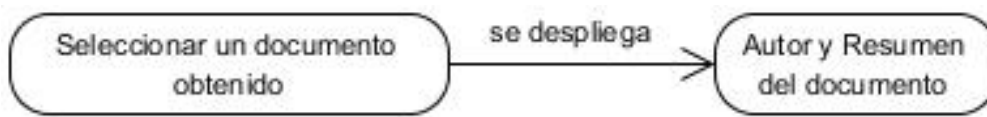


Ilustración 24: Gráfico correspondiente al caso de prueba del RF3: Revisar documentos

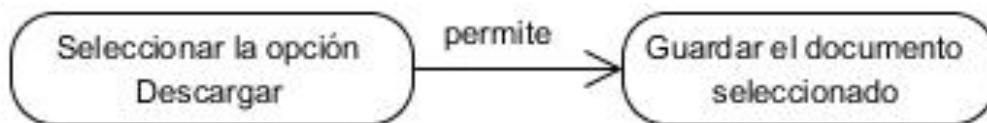


Ilustración 25: Gráfico correspondiente al caso de prueba del RF4: Descargar documento

4.5.2.4. Pruebas del sistema

Se realizaron pruebas de carga para 60 usuarios concurrentemente en un periodo de subida de 1 segundo. En este caso se observa que las pruebas se han realizado con un 0.00% de errores.

Las pruebas fueron realizadas en una PC con requerimientos de 1GB de RAM y un procesador Intel CORE i3 de segunda generación. A continuación, se muestran los reportes de las pruebas aplicadas con la herramienta Jmeter de la simulación de 60 usuarios en un período de subida de 1 segundo:

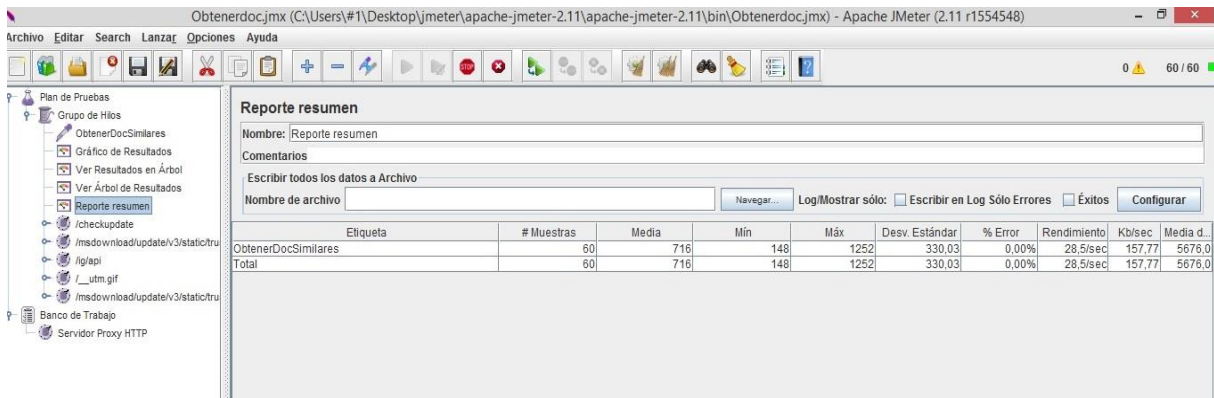


Ilustración 26: Prueba de carga - Obtener documentos similares

4.5.3. Resultados de las pruebas

El sistema se sometió a una serie de pruebas de validación que permitieron comprobar el correcto funcionamiento de los requisitos implementados mediante la utilización del método de Caja Negra. Las técnicas empleadas fueron la partición equivalente y gráfico de prueba para cada requisito funcional. Se desarrollaron 3 iteraciones de prueba arrojando un total de 3 no conformidades. Al finalizar las pruebas se obtuvo un 100% de no conformidades resueltas, lo que define que el componente desarrollado para el sistema RBC cumple con los RF. Entre los resultados de las pruebas están los siguientes:

Pruebas realizadas	Iteración	No conformidades
Subir documento	Iteración 1	Desarrollada satisfactoriamente
Mostrar documentos similares	Iteración 1	Cuando el usuario subía al sistema un documento que no era de formato doc, docx o pdf y daba click en la opción Buscar Documentos Similares, el sistema no mostraba el mensaje "El formato del documento no es válido".
	Iteración 2	Desarrollada satisfactoriamente
Revisar documentos	Iteración 1	Cuando el usuario daba click sobre un documento obtenido no se

		desplegaba el autor ni el resumen del documento.
	Iteración 2	Cuando el usuario daba click sobre un documento obtenido se desplegaba el autor y el resumen pero vacíos.
	Iteración 3	Desarrollada satisfactoriamente
Descargar documento	Iteración 1	Desarrollada satisfactoriamente

Tabla 4: Resultados de las pruebas de Caja negra

4.6. Conclusiones parciales

En este capítulo se explicó la implementación de las nuevas funcionalidades del sistema RBC, a través de los diagramas de componente del sistema y el diagrama de despliegue. Se definieron los estándares de codificación empleados en el código, para lograr un mejor entendimiento del mismo. Se documentó el resultado de estrategias que garantizan la calidad del componente desarrollado para el sistema RBC y el cumplimiento de los requisitos definidos por el cliente.

Conclusiones Generales

Una vez culmina la presente investigación y el desarrollo del componente de búsqueda para la aplicación web RBC, se puede arribar a las siguientes conclusiones:

- El estudio de las principales tendencias relacionadas con los SR que utilizan la técnica por similitud de objetos, permitió determinar que las mismas no solucionaban el problema planteado.
- La selección de las herramientas, tecnologías y metodologías de desarrollo contribuyó a agilizar la implementación del componente de búsqueda para el sistema RBC.
- Se desarrolló un componente de búsqueda para aplicación web RBC que permite la recomendación de bibliografías científicas similares a un documento que suba el usuario al sistema.
- Las pruebas realizadas arrojaron resultados satisfactorios, pues se identificaron un grupo de no conformidades que fueron corregidas, lo que posibilitó la verificación y validación de las funcionalidades del componente desarrollado al sistema RBC.

Recomendaciones

Una vez concluido el desarrollo del componente de búsqueda para el sistema RBC, utilizando la recomendación por similitud de objetos, y luego de haber cumplido los objetivos de la presente investigación, se recomienda:

- Ampliar la gama de extensiones permitidas de los documentos subidos por los usuarios al sistema.
- No limitar la búsqueda de documentos científicos a los que se encuentran en la base de datos, sino también que el sistema sea capaz de buscar publicaciones científicas en otras fuentes.

Referencias bibliográficas

Arancibia, José Alberto Gallardo. 2009. *Metodología para la Definición de Requisitos en Proyectos de Data Mining (ER-DM).*

Aristegui O, José Luis. 2010. Test cases in software test.

Cervantes, Dr.Humberto. 2010. Arquitectura de Software. Arquitectura. [En línea] [Citado el: 4 de abril de 2016.] <http://sg.com.mx/content/view/922>.

Chapman, Pete, y otros. 2000. *Guía paso a paso de Minería de Datos.*

Cortez Vásquez, Augusto, Vega Huerta, Hugo y Pariona Quispe, Jaime . 2009 Procesamiento de lenguaje natural. Universidad Nacional Mayor de San Marcos : s.n.

Flores, Ervin y Cordero, Jorge Luis. Metodologías Ágiles, Proceso Unificado Ágil (AUP). Bolivia : s.n.

Galindo, Raúl Miguel Romero. 2012. Análisis, diseño e implementación de un sistema de información aplicado a la gestión educativa en centros de educación especial. Perú : s.n.

Garre M., Cuadrado J.J., Sicilia,M.A., Charro M, Rodríguez D. 2005. *Seg-mented Parametric Software Esti-mation Models: Using the EM algo-rithm with the ISBSG 8 database,Information Technology Interfaces.* Croacia : s.n.

Giugni, Marylin y León, Luis. 2011. ClusterDoc un sistema de recuperación y recomendación de documentos basados en algoritmos de agrupamiento. Vol. 15, 60.

Godoy, Daniela. 2015. Minería de Datos Web. <http://www.exa.unicen.edu.ar/catedras/ageinweb/>. [En línea]

González, Carlos Caballero. 2007. *Desarrollo de software con calidad para una empresa.*

Guijarro, Álvaro Primo. 2012. Protocolo HTTP.

Hernández, Francisco Refugio Zavala. 2014. *Buscador de artículos científicos aplicando Minería de datos .* Mexico : s.n.

Jacobson, Ivar , Booch , Grady y Rumbaugh, James . 2000. *El Proceso Unificado De Desarrollo de Software.* Madrid : Addison Wesley, 2000. ISBN 84-7829-036-2.

Jorin, Ing.Michaell González. 2007. Tesis de maestría.Proceso de pruebas para la liberación de productos software. Ciudad de La Habana : s.n.

Jovanović, Irena. 2008. *Software testing methods and techniques.*

Larman, Craig. 1999. *UML y Patrones. Introducción al análisis y diseño orientado a objetos. Primera edición.* México : s.n. ISBN 970-1 7-0261-1.

Merseguer, José. 2010. Diagramas de casos de uso. Zaragoza : s.n.

Mizhuero Cañar, Katty y Barrera Heredia, Jorge. 2009. *Ánalysis, diseño e implementación de un sistema adaptivo de recomendación de información basado en Mashups.* Ecuador : s.n.

Moine, Juan Miguel. 2013. Metodologías para el descubrimiento de conocimiento en bases de datos: un estudio comparativo.

Montero, Ivelisse, Montero Jimenez, Ivelisse y Barroso Marquez, Yeneisy. 2015. Representación de datos bibliográficos mediante técnicas de visualización. La Habana, Cuba : s.n.

Oré, Alexander. Calidad y Software. [En línea] [Citado el: 20 de Mayo de 2016.] http://www.calidadyssoftware.com/testing/pruebas_unitarias1.php.

PostgreSQL. 2013. *The world's most advanced open source database.*

Pressman, Roger S. 2005. *Ingeniería del software. Un enfoque práctico. Sexta edición.*

Pressman, Roger S. 2010. *Software Engineering. A practitioner's Approach.* Madrid : McGraw-Hill : 7 Edición, 2010. ISBN 978-0-07-557597-7.

Pyle, Dorian. 1999. *Data Preparation for Data Mining.*

Rubén Dario. 2014. Metodologías de Desarrollo Ágiles Vs. Metodologías Tradicionales. [En línea] 20 de marzo de 2014. [Citado el: 18 de abril de 2016.] <http://rdsoporteymantenimientodepc.blogspot.com/2014/03/metodologias-de-desarrollo-agiles-vs.html>.

Ruiz Tenorio, Roberto. 2010. *Las pruebas de software y su importancia en las organizaciones.* Universidad Veracruzana : s.n.

Rumbaugh, James, Jacobson, Ivar y Booch, Grady. *El Lenguaje Unificado de Modelado.*

Schmuller, Joseph. *Aprendiendo UML en 24 horas.* México : Editorial División Computación.

Sinnexus. 2007. Sinnexus. [En línea] 2007. [Citado el: 30 de mayo de 2016.] <http://www.sinnexus.com/empresa/index.aspx>.

Software, Ingeniería del. 2005. Introducción al Modelo Conceptual.

SOLUS S.A. 2000-2016. Spark Systems. [En línea] 2000-2016. [Citado el: 5 de mayo de 2016.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.

Sommerville, Ian. 2005. Ingeniería del Software Séptima edición. Madrid : s.n., 2005. 84-7829-074-5.

Tedeschi, Nicolás. 2010. ¿Qué es un patrón de diseño? [En línea] 2010. [Citado el: 5 de Abril de 2016.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.

Torres, Sergio Santamaria. 2014. *Sistema de descubrimiento de bibliografía científica.*

Vallejos, Sofia J. 2006. *Diseño y Administración de Datos.* Argentina : s.n.