Universidad de las Ciencias Informáticas

Facultad 10

"Análisis y Diseño de un Sistema de Recuperación de Información asociada al proceso de desarrollo de software de la Universidad de las Ciencias Informáticas"

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

Autores:

Susell Cruz Reyes.

Angela Virgen Pérez Gómez.

Tutor:

Ing. Yandry Alberto Terry.

Ciudad Habana. Cuba

"Año 52 de la Revolución"

Declaración de autoría

Por este medio declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los __ días del mes de __ del 2010.

Autores

_____ _____

Angela Virgen Pérez Gómez Susell Cruz Reyes

Tutor

_____

Ing. Yandry Alberto Terry

"... Si los jóvenes fallan, todo fallará. Es mi más profunda convicción que la juventud cubana luchará por impedirlo. Creo en ustedes…"

Fidel Castro Ruz

Agradecimientos

A mis padres por todo su amor, apoyo, confianza y dedicación.

A mis tíos Joaquina y Rolando, que son mis otros padres, así como mis otros tíos que también me han apoyado mucho.

A toda mi familia que de una manera u otra me han ayudado a convertirme en quien soy.

A Deli y a Lily porque sé que me quieren mucho y que siempre han estado ahí para mí.

A mi novio por su apoyo y a su familia por preocuparse por mí.

A mi compañera Susell por confiar en mí y darme aliento para que todo saliera bien.

A mi tutor por guiarnos y además lidiar con mi insistencia.

A mis amistades que han estado ahí en los momentos más difíciles que he vivido en la universidad en especial Yanet, Loty, Mayrel, Luisi y todos aquellos que han sabido ser amigos de verdad.

A Dianelys por su ayuda para terminar este trabajo.

A todos aquellos que de una manera u otra han formado parte de mi vida, que aunque no ponga todos los nombres los quiero mucho y les estoy eternamente agradecida.

A la Revolución y a Fidel por la oportunidad que me han dado de convertirme en Ingeniera.

De Angela

Agradecimientos

A toda mi familia y a mi novio por toda su confianza, a mis vecinos en especial a Mari, María, Normita, Flores, Félix e Ismary por darme ánimo siempre.

A toda la familia de mi amor Alejandro, que ya es mi familia, por darme su apoyo, su cariño y su confianza para sentirme como uno más entre ellos.

A mi tutor, Yandry por guiarnos durante todo el desarrollo de la tesis.

A mi compañera de tesis Ángela por entendernos y tratar de que todo saliera bien.

A mi tío Juan Amador, por ser un gran apoyo para mí aquí en la universidad.

A todos los que de una forma u otra estuvieron pendientes todo el tiempo de mis logros dándome su apoyo y optimismo, en especial a mis amistades, Omay, Darianne, Eglis, Yilena, Yisel, Medina por ayudarme con el documento, a todos los muchachos del apto 136 106 por bridarme la máquina para que trabajara en mi tesis, a la Revolución y la Universidad de las Ciencias Informáticas por permitirme que me forjara como Ingeniero en Ciencias Informáticas.

De Susell

Dedicatoria:

A Olga Edilia y a mi padrastro Nicolás, los más buenos, comprensivos y bonitos del mundo entero, por su cariño, su ejemplo, sus grandes expectativas sobre mi carrera, su apoyo moral e incondicional, por todo su sacrificio y la confianza que siempre depositaron en mí, para que mis sueños se hicieran realidad.
A Saelis, mi hermanita linda, que luche fuerte para que encuentre sus sueños, que nunca se dé por vencida y que sepa que siempre me tendrá allí para ella.
A mi novio Alejandro, por su cariño, su confianza, su amor y su apoyo en todos los momentos buenos y malos que hemos pasado juntos.
A mis primos, Erlis y Ernesto que forman parte de mi vida y que los quiero mucho.
A mis tíos Amarilis y Medardo, por su cariño y apoyo, los quiero mucho.
A mi abuelo Gerardo, por ser para mi más que un abuelo; un padre por su amor eterno hacia mí, por su ejemplo de seguir luchando en la vida. Ser abuelo no significa vejez sino sabiduría teñida de blanco, no significa una voz lenta sino la fuerza del saber, ser abuelo no significa ausencia sino el viaje a la sabiduría que hoy cultiva.

A la memoria de mi abuela Aida Sánchez, por ser para mí la mujer más buena y dedicada, la abuela más especial, y aunque ya no se encuentre físicamente junto a mí, sé que siempre está a mi lado, esta es una dedicatoria especial por el gran amor y apoyo que siempre tuve de mi abuelita.

"Abuela, sé que me escucharás porque no te has ido y nunca te irás porque estás en cada latido, en cada lágrima, en cada suspiro. Ahora mismo estás viva, pues tu esencia sigue, tu recuerdo, tu ejemplo, tu valor y tu esfuerzo han quedado plasmados en nuestra memoria y escritos con letras doradas en el corazón. Abuela, mujer admirable, no has muerto, y nunca lo harás porque no se muere cuando el corazón deja de latir, se muere cuando en los recuerdos se deja de existir y tú estás presente, estas aquí, estás viva, para todos y para mí."

De Susell

Dedicatoria

Este es un momento muy especial para mí, estoy cumpliendo no solo mi sueño, sino el de mis padres, para demostrarles lo orgullosa que estoy de ellos y decirles que lo hicieron muy bien por eso les dedico este trabajo de manera muy especial.

A mi madre, por ser más que eso, por estar a mi lado siempre transmitiéndome su espíritu y su amor incondicional. A mi padre porque lo admiro mucho y sé que está orgulloso de mi.

A mi hermano porque lo adoro y es lo único que tengo, espero que encuentre su camino igual que yo encontré el mío.

A mi abuelita Agustina y mi tía Ramona, que son las personas que más profundo llevo en el corazón. Por ser aquellas personas llenas de amor y que siempre han estado ahí para mí como un manantial inagotable de comprensión.

A mis tíos porque han sido como mis padres, porque me han ayudado tanto y sé que lo seguirán haciendo. Sé que aunque no estén aquí están sintiendo este momento tanto como yo, porque son así.

Simplemente le dedico este trabajo a todo el que en algún momento ha confiado en mí. Gracias por su cariño y por todas las alegrías que me han dado. Los quiero mucho.

De Angela

Resumen

La constante búsqueda y recuperación de información por los usuarios constituye en la Universidad de las Ciencias Informáticas un problema real y que necesita especial atención dada las nuevas características de la institución orientadas a la producción de software y las limitaciones del acceso a internet. En el presente trabajo se realiza el análisis y diseño de un sistema de recuperación de información que servirá de apoyo vital al proceso productivo.

Se realizó un estudio de los principales sistemas de recuperación existentes a nivel internacional, nacional y dentro de la universidad, llegando a la conclusión de que no se cuenta con un sistema que se ajuste a las características tanto del país como del centro, tomando además de dichos sistemas los aspectos para el diseño de la propuesta. Se estudiaron además, herramientas, la notación para el modelado, metodologías de desarrollo y el lenguaje de programación, así como consideraciones para una arquitectura que se ajustara a las características del sistema.

En el trabajo se muestran todos los diagramas, figuras, tablas y prototipos de interfaz de usuarios que posibilitan el diseño del sistema, explicándose todas las características principales del mismo, con lo cual se deja listo el camino para las futuras actividades de implementación.

Palabras clave:

Sistema de Recuperación de Información, BPMN, análisis y diseño, prototipo de interfaz de usuario.

Índice

Introducción

El desarrollo de la tecnología y las comunicaciones, ha traído consigo una revolución de información que cualquiera en cualquier lugar del mundo puede publicar o simplemente acceder a informaciones publicadas por otros. El volumen de información que se maneja por la red crece considerablemente, dada la necesidad y avidez de los usuarios por aprender y satisfacer su curiosidad.

Desde la aparición de la escritura, la necesidad de almacenar y transmitir la información se convirtió en algo primordial, por lo que se hace imprescindible un sistema organizativo que posibilite la localización de la misma en cualquier momento. En los inicios cuando el volumen de información que se gestionaba era al menos manejable se hacía mediante la tabla de contenidos de un libro, luego a medida que el volumen fue creciendo se sustituyeron por estructuras algo más complejas, pero la evolución lógica de la tabla de contenidos fue el índice, lo que constituye el núcleo de los Sistemas de Recuperación actuales. Dada la explosión y el acelerado desarrollo de las tecnologías, la necesidad de recuperación no puede ser afrontada sino con nuevas técnicas y herramientas de almacenamiento, acceso, consulta y uso de esa información.

Por lo que la mejor manera de realizar una búsqueda de cualquier tema es utilizando un Sistema de Recuperación de Información (SRI) que puede entenderse en forma simple como aquel sistema que almacena y recupera información, que está compuesto por varios componentes que interactúan entre sí para lograr un objetivo: recuperar información, aunque en realidad su valor va a depender de la capacidad para localizar y recuperar los documentos de una manera veloz y económica.

En la actualidad existen muchas formas de recuperar información aunque la más usada son los llamados motores de búsqueda, entre ellos el que tiene la corona en el tema de recuperación es Google, una poderosa herramienta que brinda al usuario servicios eficientes, además de la excelencia en cuanto a rapidez y volumen de información se refiere, a pesar de que en ocasiones los resultados no sean los que el usuario desea.

La Universidad de Ciencias Informáticas (UCI), entidad dedicada al desarrollo de soluciones informáticas maneja un gran volumen de datos en este proceso. Cuba es uno de los países que tiene bloqueado el

pleno acceso a la información digital de Internet, por tal motivo el acceso a la información es controlado y la UCI no está exenta de estas regulaciones. Para solucionar la demanda del servicio de Internet, la Dirección de Seguridad Informática y Redes de la UCI brinda dos servicios fundamentales, los mismos son Internet limitada y FTP. Estos servicios están orientados al proceso docente de los estudiantes, por tal motivo no suple la demanda de información que necesitan la red de centros de desarrollo de la UCI. Los estudiantes y profesores que se encuentran involucrados en el proceso de desarrollo de software se demoran mucho buscando información en el FTP y la cuota limitada de internet no les alcanza para poder desarrollar las soluciones informáticas en el tiempo requerido además de que en internet no siempre se encuentra la información necesaria, por tales motivos es que existe un retraso en la búsqueda de información relativa al proceso de desarrollo de soluciones informáticas en la UCI.

Por lo que el presente trabajo plantea el siguiente **problema científico**: ¿Cómo agilizar el proceso de búsqueda y recuperación de información asociada al proceso de desarrollo de software en la UCI?

Su **objeto de estudio** son los procesos de búsqueda y recuperación de información automática y el **campo de acción** estaría enmarcado en los procesos de búsqueda de información en la red de los centros de desarrollo de la UCI.

De acuerdo al problema planteado anteriormente se propone como **objetivo general** Diseñar un sistema de recuperación de información que agilice la búsqueda de información asociada al proceso de desarrollo de software en la UCI.

Idea a defender

Con el análisis y diseño del Sistema de Recuperación de Información se logrará obtener una arquitectura robusta para el almacenamiento de la información así como un algoritmo matemático eficiente para la implementación del sistema además de preparar las bases para la futura implementación del sistema.

Objetivos específicos:

- Realizar estudio del estado del arte de los Sistemas de Recuperación de Información en la UCI y en el ámbito mundial.
- Proponer la arquitectura del Sistema de Recuperación de Información.
- Proponer el diseño del Sistema de Recuperación de Información de la Vice-rectoría de Producción de la Universidad de las Ciencias Informáticas.

Para el cumplimento de los objetivos se plantean las siguientes **tareas**:

- Análisis de la estructura de la información asociada al proceso de desarrollo de software en la Universidad de las Ciencias Informáticas.
- Definición de la información necesaria para el proceso de desarrollo de software en la Universidad de las Ciencias Informáticas.
- Análisis y comparación de los principales Sistemas de Recuperación de Información actuales.
- Realización de un estudio de los principales modelos matemáticos que utilizan los Sistemas de Recuperación de Información actuales.
- Evaluación y selección de los modelos matemáticos utilizados por los Sistemas de Recuperación de Información actuales.
- Análisis de las arquitecturas para el almacenamiento de la información.
- Elaboración de la propuesta de la arquitectura para el almacenamiento de la información que se ajuste a las características del proceso de desarrollo de software.
- Diseño del sistema de recuperación de información de la Vice-rectoría de Producción de la Universidad de las Ciencias Informáticas.

Para el cumplimiento de estas tareas se plantea la utilización de los siguientes métodos:

Analítico–sintético: Con la valoración que se obtiene de los Sistemas de Recuperación de Información existentes, así como de los modelos matemáticos y la forma de almacenar la información en la universidad, se puede seleccionar el mejor modelo matemático así como utilizar las mejores características de los Sistemas de Recuperación de Información existentes.

Histórico-Lógico: Permite que se analice el desarrollo histórico de los Sistemas de Recuperación de Información y facilita diferentes características y datos sobre los sistemas de recuperación más utilizados, así como cambios a que han sido sometidos los mismos a través de los años y las nuevas soluciones, mejoras y aportes a los ya existentes.
Modelado: Con la utilización de este método se selecciona y se elabora la propuesta de diseño del Sistema de Recuperación de Información.

Para una mejor organización del contenido, el documento está estructurado por tres capítulos, descritos a continuación:
Capítulo 1: Fundamentación teórica: Se realiza un estudio del estado del arte de los Sistemas de Búsqueda y Recuperación de Información a nivel internacional y nacional, se estudian lo métodos y modelos matemáticos. Su lectura le ofrecerá información referente a los principales conceptos tratados, y por último podrá conocer las herramientas, la metodología y los lenguajes a utilizar para dar solución al problema.
Capítulo 2: Características del Sistema: En este capítulo se describe el mapa de procesos de negocio, el modelo conceptual, la descripción de los requisitos, así como los prototipos de interfaz de usuario.

Capítulo 3: Análisis y Diseño del Sistema: Contiene los patrones utilizados, los diagrama de clases del diseño, el modelo de la arquitectura para el sistema y los resultados obtenidos con el diseño del mismo.

Capítulo 1 Fundamentación teórica.

En este capítulo primeramente se analizan los conceptos básicos a tratar durante el desarrollo del trabajo de diploma, luego el estado del arte de los principales Sistemas de Recuperación de Información que existen en el mundo. Se realiza además un estudio de las herramientas, lenguaje de modelado y metodología de desarrollo de software a utilizar durante el desarrollo práctico del trabajo.

1.1 Conceptos básicos

Proceso de Desarrollo del Software

Un proceso define “quién” está haciendo “qué”, “cuándo” y “cómo” para alcanzar un determinado objetivo. Un Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto.

Sistemas de Recuperación de Información

Baeza–Yates (1999): Parte de la informática que estudia la recuperación de la información (no datos) de una colección de documentos escritos. Los documentos recuperados pueden satisfacer una necesidad de información de un usuario expresada normalmente en lenguaje natural.

Korfhage (1997): La localización y presentación a un usuario de información relevante a una necesidad de información expresada como una pregunta.

Salton (1989): Un Sistema de Recuperación de Información procesa archivos de registros y peticiones de información e identifica y recupera de los archivos ciertos registros en respuesta a las peticiones de información.

Analizando estos conceptos finalmente se coincide con el concepto ofrecido por María Pinto Molina[1] "Proceso donde se accede a una información previamente almacenada, mediante herramientas informáticas que permiten establecer ecuaciones de búsqueda específicas. Dicha información ha debido de ser estructurada previamente a su almacenamiento."

Sistemas de Recuperación de Información en la Web

Los sistemas de recuperación en la web utilizan generalmente el Modelo de Espacio Vectorial para el almacenamiento de los documentos. Dos formas básicas de buscar información en la web son los motores de búsqueda y los directorios.

Ambas formas manejan grandes bases de datos que contienen principalmente direcciones e información de páginas.

Los motores de búsquedas

Son sofistificados programas que realizan la búsqueda de información en la web de forma automática, mediante los robots de búsqueda.

Los directorios

Son aplicaciones controladas por humanos que manejan subdirectorios de categorías temáticas con enlaces a páginas referenciadas.

Modelos de Recuperación de Información:

Existen muchos modelos, según diferentes autores, los agrupan en distintos grupos de acuerdo con sus conocimientos y nivel de estudio de los mismos (16).

[1] Catedrática de Universidad, Licenciada en Filosofía y Letras. Universidad de Granada, 1979. Doctora en Filosofía y Letras. Universidad de Granada, 1984.

Según Dominich estos tienen la siguiente clasificación

Modelo	Descripción
Clásicos	Este grupo incluye los tres más comunes: Lógico o Booleano, Probabilístico y del Espacio Vectorial.
Alternativos	Estos modelos están basados en la Lógica Fuzzy.
Lógicos	Están basados en la Lógica Formal y la recuperación de información se lleva a cabo por medio de un proceso inferencial.
Basados en la interactividad	Incluyen posibilidades de expansión del alcance de la búsqueda y emplean la retroalimentación por relevancia de los documentos recuperados.
Basados en la Inteligencia Artificial	Bases de conocimiento, redes neuronales, algoritmos genéticos y procesamiento del Lenguaje Natural.

Tabla 1 Clasificación de los Modelos de Recuperación de Información.

Baeza-Yates, divide a los modelos basados en la recuperación en dos grupos: clásicos y estructurados.

Modelos clásicos

Modelo Lógico o Booleano: Es un modelo de recuperación simple, basado en la teoría de conjuntos y el álgebra booleana.

El Modelo Booleano es uno de los primeros modelos y el más utilizado de los SRI. En este modelo, un documento se encuentra representado por un conjunto de palabras clave (palabras con un valor semántico), las cuales pueden ser extraídas de un documento, de una parte de éste o de sus metadatos. Igualmente, la consulta es un grupo de palabras clave. Generalmente se utilizan archivos inversos para

almacenar las palabras clave. Los archivos inversos contienen los siguientes campos: palabra clave o término índice (describe al documento), un identificador de documento (debe ser único para cada documento) y un identificador de campo (donde se encuentra la palabra clave). En un sistema booleano las consultas de los usuarios contienen operadores lógicos (Y, O, NO), y así un motor de búsqueda regresa aquellos documentos que cumplen con los aspectos lógicos de la consulta.

En un SRI hay dos instancias: el almacenamiento de los documentos, y la recuperación de información desde la solicitud del usuario. En la figura se ilustran las dos instancias del proceso de almacenamiento y recuperación basado en el Modelo Booleano.

a) Desde el punto de vista del almacenamiento del documento en el SRI van a ocurrir los siguientes procesos:

1. A cada documento que entra se le asigna un Identificador.
2. Se identifican las palabras contenidas en el documento.
3. Se excluyen las palabras vacías.
4. Se «cortan» las palabras, es decir, se extraen las raíces de las palabras.
5. Se establece un peso de ponderación para cada raíz.
6. Finalmente las raíces debidamente ponderadas se introducen en la base de datos.

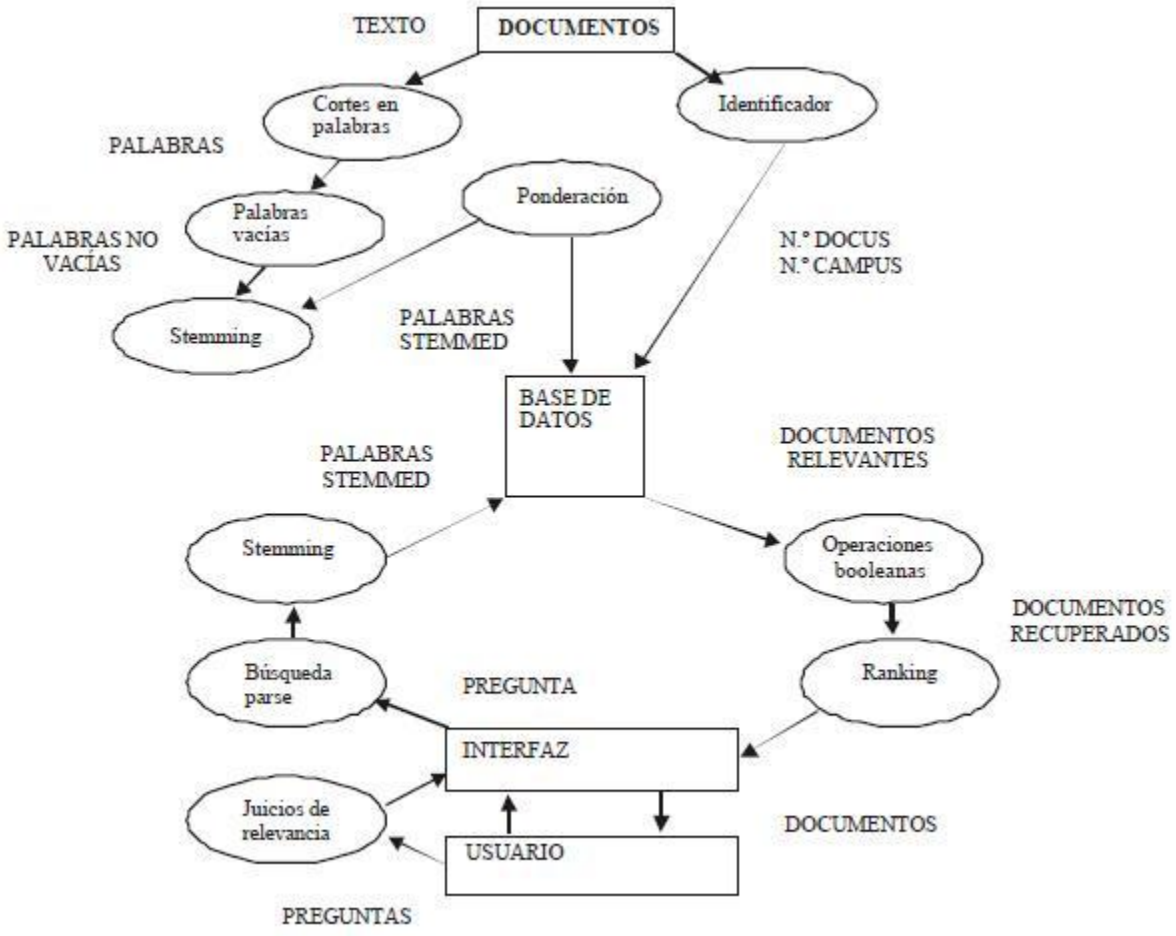


Figura 1 Vista Funcional del modelo Booleano

Modelo Espacio Vectorial: El modelo de recuperación vectorial o de espacio vectorial propone un marco en el que es posible el emparejamiento parcial, asignando pesos no binarios a los términos índice de las preguntas y de los documentos. Estos pesos de los términos se usan para computar el grado de similitud entre cada documento guardado en el sistema y la pregunta del usuario.

Según este modelo, cada expresión del lenguaje natural puede representarse como un vector de pesos de términos, en donde un término es la unidad mínima de información, por ejemplo una palabra o la raíz sintáctica de una palabra. La asignación de pesos a los términos indica su presencia o importancia en el

documento o en la colección de documentos. Habiendo varias técnicas para asignar pesos, una de ellas es la frecuencia del término, es decir, el número de veces que aparece el término en un documento.

Arquitectura

En la figura se presenta la gráfica de la vista funcional del modelo, en donde se realizan las siguientes tareas:

1. Se analizan los documentos y se transforman a una representación interna de cada uno.
2. Se analiza la consulta y se transforma a una representación interna.
3. A partir de las representaciones obtenidas en los pasos anteriores se calcula el grado de similitud entre cada documento y la consulta.
4. Se recuperan los documentos que guardan mayor similitud con la consulta del usuario. (1)

Figura 2 Vista funcional del modelo Espacio Vectorial

Modelos estructurados:

Entre los que destacan listas no sobrepuestas y el método de los nodos proximales.

Lucene:

Para hablar de Recuperación de Información hoy en día es necesario hablar de una de las más flexibles librerías del mercado para crear aplicaciones de este tipo: Lucene, que ha tenido gran éxito y escalabilidad en muchas aplicaciones. Lucene vio la luz por Doug Cutting, y puesta como software abierto (Open

Source) en marzo de 2000 a través de SourceForge. En septiembre de 2001 se unió a la Apache Software Foundation, en concreto a la familia Jakarta de productos escritos en Java.
Lucene es una novedosa herramienta que permite tanto la indexación cómo la búsqueda de documentos. Creada bajo una metodología orientada a objetos e implementada no solamente en Java sino también en C++, Python, C# y PHP, no se trata de una aplicación que pueda ser descargada, instalada y ejecutada sino de una API flexible, muy potente y realmente fácil de utilizar, a través de la cual se pueden añadir, con pocos esfuerzos de programación, capacidades de indexación y búsqueda a cualquier sistema que se esté desarrollando (17).

1.2 Sistemas de Recuperación de Información conocidos mundialmente

El estudio de los SRI también nos permite plantear el análisis y evaluación de cuatro sistemas de recuperación que sobresalen: KARPANTA, SISA, DIALOG y SMART, y finalmente del SRI más utilizado en la actualidad, el cual muchos consideran simplemente un motor de búsqueda sin embargo es un poderoso sistema de recuperación, en otras palabras estamos hablando de Google.

KARPANTA: Es un SRI basado en el Modelo de Espacio Vectorial desarrollado en la Universidad de Salamanca, España, los objetivos prioritarios de este trabajo no eran el de construir un motor de búsqueda operacional sino una herramienta para la docencia y la investigación, sin embargo el resultado del proyecto fue robusto como para ser utilizado con éxito en algunos entornos documentales.

En cuanto a su arquitectura, Karpanta se apoya en dos módulos, uno de indización, que construye los vectores de documentos y consultas, y otro de búsqueda, que obtiene los documentos más similares a una consulta dada. La construcción de los mismos se ha realizado utilizando el SGBD Microsoft Access, por su facilidad de uso, transparencia del sistema y posibilidades docentes, a pesar del descrédito que los sistemas de bases de datos relacionales han tenido en entornos documentales.

SISA: (Sistema para la indización Semiautomática) es un sistema de indexación desarrollado en la Universidad Politécnica de Valencia, España implementado en Java para el análisis de artículos científicos de Biblioteconomía y Documentación (2).

SMART: Diseñado en 1964 por Salton[2] fue concebido como una herramienta experimental de la evaluación de la efectividad de muchos tipos de análisis y procedimientos de búsqueda. Se distingue de los SRI convencionales en cuatro aspectos fundamentales:

- Usa métodos de indización automática.
- Agrupa documentos relacionados dentro de clases comunes de materias.
- Identifica los documentos a recuperar por similitud con la pregunta realizada por el usuario.
- Incluye procedimientos automáticos para generar mejores ecuaciones de búsqueda.

SMART incorpora tres procedimientos diferentes de análisis del lenguaje, conocidos como palabra, lema y tesauro. El primero de estos métodos emplea palabras comunes reducidas a su forma singular a las que se les asigna un peso. El segundo método extrae la base de la palabra, desprendiéndola de los sufijos, de manera que se agrupan varias palabras en un mismo lema, al cual se le asigna el peso. Con el tesauro se asignan los términos descriptores que mejor representan a los conceptos de los documentos y se les asigna un peso.

Estos sistemas fueron los pioneros en el caso de SISA solo indizaba los documentos en formato *.txt, además de que tenía una interfaz compleja. SMART, fue el primer SRI en utilizar métodos de indización automática, sirviendo de base para los futuros sistemas. En el caso de Karpanta considerado como un motor de búsqueda experimental, ya brindaba algunas ventajas como su implementación siguiendo el Modelo de Espacio Vectorial; consultas en lenguaje natural; almacenamiento de documentos de tamaño indefinido; devolución de resultados ordenados por similaridad e incorporando conocimiento lingüístico, así como facilidad en el código y en su utilidad en la docencia. Por estas características dieron paso al surgimiento de los sistemas que hoy conocemos.

[2] Gerard Salton (Nuremberg, 8 de marzo de 1927 - Nueva York, 28 de agosto de 1995) fue un informático y documentalista científico estadounidense de origen alemán. Especialista en Recuperación de información y en procesamiento del lenguaje natural.

Las grandes empresas desarrolladoras de software e incluso personas con intereses personales se han dedicado a la producción de herramientas que facilitan la recuperación de información, por lo que respecto a este tema se pueden mencionar diversos motores de búsqueda y directorios. De ellos el más conocido es el gigantesco motor de búsqueda Google y el directorio más antiguo Yahoo, aunque han surgido muchos otros los cuales han sido comprados por los propietarios de Yahoo o de Google.

A lo largo de todo el país se han desarrollado algunos esfuerzos por realizar Sistemas de Recuperación de Información, pero los resultados han sido poco alentadores, dado que la red nacional está aún creciendo y el volumen de información cada vez es mayor, aparejado a esto que cada institución necesita informaciones distintas y en la mayoría de los casos dicha información no se encuentra disponible en la red. Por lo que dichos sistemas apenas y se conocen y finalmente cada institución -de acuerdo a sus necesidades - ha tratado de suplir su necesidad de información mediante otros recursos.

En la Universidad se han desarrollado algunas herramientas para la recuperación de información como son SearchEngine y el proyecto Orion, la primera fue un trabajo de diploma que jamás vio su aplicación y la segunda es un nuevo proyecto que traerá grandes beneficios a la comunidad universitaria, pero ninguna de ellas ha sido puesta en funcionamiento, además ninguna ha estado orientada al proceso productivo, y dado el proceso de mejoras se requiere de un sistema que sea capaz de brindar toda la información que se necesita sin que esto represente un problema.

1.3 Arquitectura de software

Una Arquitectura de Software, también denominada Arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información. La Arquitectura de Software establece los fundamentos para que analistas, diseñadores, programadores, etc. trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades.

Una arquitectura de software se selecciona y diseña con base en objetivos y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como la mantenibilidad, auditabilidad, flexibilidad e interacción con otros sistemas de información. Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para

implementar sistemas de información. Unas arquitecturas son más recomendables de implementar con ciertas tecnologías mientras que otras tecnologías no son aptas para determinadas arquitecturas.

La arquitectura en capas soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite a los implementadores la partición de un problema complejo en una secuencia de pasos incrementales. Además, proporciona amplia reutilización y soporta fácilmente la evolución del sistema puesto que los cambios sólo afectan a las capas vecinas. Una arquitectura en capas permite cambiar las implementaciones respetando las interfaces con las capas adyacentes.

Generalmente se definen 3 capas: capa de presentación, capa de negocio y la capa de datos. A continuación la explicación de cada una de ellas:

Capa de presentación.

Esta capa es la que ve el usuario, presenta el sistema, le comunica la información y captura la información en un mínimo de proceso. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable".

Capa de negocio.

Aquí es donde se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

Capa de Datos.

Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. (3)

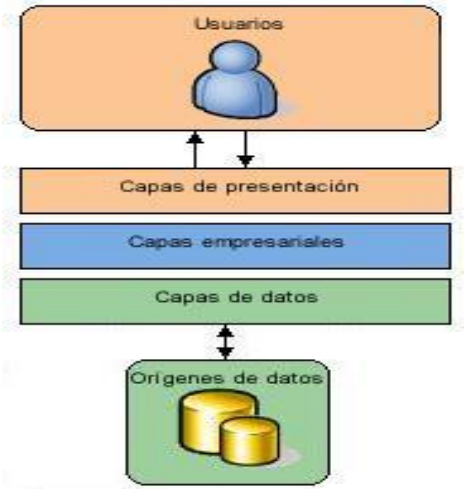


Figura 3 Arquitectura en Capas.

En este caso la capa empresarial representa la capa de negocio.

1.4 Metodologías de Desarrollo de Software.

En los inicios de la historia del software no existían metodologías para su desarrollo, ya que el software no tenía gran importancia en esos momentos. A medida que fue evolucionando el desarrollo de la informática y los pedidos de los clientes fue cada vez mayor, comenzaron a desarrollarse diferentes metodologías para el trabajo de los especialistas. Estas metodologías son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software.

1.4.1 Rational Unified Process (RUP)

RUP es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organización, diferentes niveles de aptitud y diferentes tamaños de proyecto. Está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes de software interconectados a través de interfaces bien definidas y utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) como soporte a la metodología. El Proceso Unificado de Rational está dirigido por casos de uso, centrado en la arquitectura y es iterativo e incremental. (4)

RUP se caracteriza por dividir el ciclo de vida de la producción del software en 4 fases:

1. Inicio: es donde se determina la visión del proyecto, o sea se comprende el entorno y se determina el alcance del producto.

2. Elaboración: en esta etapa se determinan los cimientos de la arquitectura y se analiza el dominio del problema.

3. Construcción: en esta fase se obtiene la capacidad operacional inicial del producto.

4. Transición: se obtiene el release o liberación del producto y se pone en manos de los usuarios finales.

1.4.2 Las metodologías ágiles: Programación Extrema -Extreme Programming (XP)- y SCRUM.

1.4.2.1 Programación Extrema

Esta metodología es utilizada en proyectos de corto plazo, equipo reducido, y cuyo plazo de entrega es muy corto por lo que requieren de un grupo de programadores pequeño, donde la comunicación sea factible. La metodología consiste en una programación rápida, cuya particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto, es por esto que la comunicación es un punto fundamental en este tipo de metodología ya que debe realizarse entre los programadores, los jefes de proyecto y los usuarios. (5).

Se basa en:

Las pruebas Unitarias que son las pruebas realizadas a los principales procesos para ver las fallas que pudieran ocurrir.

En la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

En la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo.

XP tiene cuatro variables para cualquier proyecto software: tiempo, costo, calidad y alcance. Su principal objetivo es satisfacer al cliente dándole el software que necesita.

1.4.2.2 SCRUM

Es una metodología de desarrollo ágil o ligero desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle, indicada para proyectos con un rápido cambio de requisitos. Las principales características que la definen es que el desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento del producto que se muestra al cliente. El marco para la gestión de proyectos que define se ha utilizado con éxito durante los últimos 10 años.

Otra cuestión peculiar de Scrum son las reuniones a lo largo del proyecto. De todas ellas la más importante se efectúa diariamente durante 15 minutos por parte del equipo de desarrollo para coordinar e integrar el trabajo. Su práctica es trabajar con un solo representante, el dueño del producto final, aunque últimamente se estila crear un grupo de clientes finales para darle agilidad al proceso. (5)

1.5 Lenguaje de modelado.

El surgimiento de numerosos lenguajes de modelado está dado por la necesidad de la comunidad del software de comunicar sus modelos.

Jacobson plantea los objetivos de los mismos:

- Captar y enumerar exhaustivamente los requisitos y el dominio de conocimiento.
- Pensar en el diseño de un sistema.
- Capturar decisiones del diseño a partir de los requisitos.
- Generar productos aprovechables para el trabajo.
- Organizar, encontrar, filtrar, recuperar, examinar y corregir la información en grandes sistemas.
- Explorar económicamente múltiples soluciones.
- Moderar los sistemas complejos.

A continuación los principales lenguajes de modelado:

1.5.1 Lenguaje Unificado de Modelación (Unified Modelling Language, UML)

Es un lenguaje gráfico para visualizar, especificar, construir, documentar y comunicar los artefactos de un sistema de software. UML permite el desarrollo de distintos tipos de diagramas, cada uno de los cuales representa el sistema a especificar, analizar o diseñar desde distintas perspectivas (6). Es utilizado para modelar la información del sistema basado en concepto de objetos.

Las funciones principales de UML son:

- Visualizar: Permite expresar de una forma gráfica un sistema de forma que otro lo pueda entender.
- Especificar: Permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

1.5.2 BPMN (Business Process Modeling Notation)

La Notación para el Modelado de Procesos de Negocio -Business Process Modeling Notation (BPMN)- es una notación gráfica estandarizada que permite el modelado de procesos de negocio en un formato de flujo de trabajo (workflow).

Su principal objetivo es proveer una notación estándar que sea fácilmente legible y entendible por parte de todos los involucrados e interesados del negocio (stakeholders). Entre estos interesados están los analistas de negocio, los desarrolladores técnicos y los gerentes y administradores del negocio.
En síntesis BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación. (7).

El modelado en BPMN se realiza mediante diagramas muy simples con un conjunto de elementos gráficos. Con esto se busca que para los usuarios del negocio y los desarrolladores técnicos sea fácil entender el flujo y el proceso. Las cuatro categorías básicas de elementos son:

- Objetos de flujo: Eventos, Actividades, Rombos de control de flujo (Gateways)
- Objetos de conexión: Flujo de Secuencia, Flujo de Mensaje, Asociación

- Swimlanes (Carriles de piscina): Pool, Lane
- Artefactos: Objetos de Datos, Grupo, Anotación

1.5.3 AxureRP Profesional

Axure RP es una aplicación ideal para crear prototipos y especificaciones muy precisas para páginas web. Se trata de una herramienta especializada en la tarea, así que cuenta con todo lo que se puede necesitar para crear los prototipos de forma más eficiente. Axure RP te permite componer la página web visualmente, añadiendo, quitando y modificando los elementos con suma facilidad.

1.6 Herramientas Case

Ingeniería de Software Asistida por Computadora (Computer Aided Software Engineering, CASE) es un tipo de ingeniería de software en la que se intenta aumentar la eficacia de sus procesos, al soportar la realización de las tareas con el uso de tecnologías. (8)

Las herramientas CASE son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas permiten a los desarrolladores modelar y documentar sus artefactos, cubriendo el ciclo de vida del proceso de desarrollo de software.

Actualmente, cerca de 450 a 500 herramientas CASE están presentes en el mercado, y cada año penetran más dentro de las organizaciones, debido a los beneficios que representa el uso de estas en el soporte de las diferentes etapas del ciclo de vida de los Sistemas de Información. (9)

1.6.1 Rational Rose:

Es una de las más poderosas herramientas de modelado visual basado en UML para el análisis y diseño de sistemas orientados a objetos. Es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros del equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Otras de las ventajas es que los diseñadores pueden modelar sus componentes e interfaces de forma individual y luego unirlos con otros componentes del proyecto. Ayuda a los desarrolladores de software a construir mejores productos en menor tiempo, da un excelente soporte en el

manejo de cambios durante el ciclo de vida del proyecto y mejora la comunicación entre los miembros del equipo, solo que no todos tienen acceso a la misma dado que es una herramienta privada (10), (11).

1.6.2 Visual Paradigm

Visual Paradigm es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite la creación de diagramas, código inverso, generar código desde diagramas y generar documentación. Soporta un conjunto de lenguajes, tanto en la generación de código e ingeniería inversa sobre Java, C + +, PHP, XML Schema, entre otros. Tiene la capacidad de integrarse con Eclipse, NetBeans IDE/Sun™ ONE, IntelliJ IDEA™y otros.

Visual Paradigm ofrece distintas funcionalidades como:

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio generando un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa en su versión profesional, e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDE.
- Disponibilidad en múltiples plataformas (Windows, Linux, etc.)

1.7 Lenguajes de Programación

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina.

Aunque muchas veces se usan los términos 'lenguaje de programación' y 'lenguaje informático' como si fuesen sinónimos, no tiene por qué ser así, ya que los lenguajes informáticos engloban a los lenguajes de programación y a otros más. Un lenguaje de programación permite a uno o más programadores

especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias.

1.7.1 Java

Java es un lenguaje de programación orientado a objetos, toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Los programas escritos en el lenguaje Java pueden ejecutarse en cualquier tipo de hardware. La recolección de basura de Java es un proceso prácticamente invisible al desarrollador, es decir, el programador no tiene conciencia de cuándo la recolección de basura tendrá lugar, ya que ésta no tiene necesariamente que guardar relación con las acciones que realiza el código fuente.

1.8 Sistemas de Gestión de Base de Datos (SGBD).

Un Sistema Gestión de Bases de Datos –SGBD- (Data Base Management System DBMS), es un software que permite la utilización y/o actualización de los datos almacenados en una o varias Bases de Datos, desde diferentes puntos de vista a la vez, por uno o varios usuarios.

Su objetivo fundamental consiste en suministrar al usuario herramientas que le permitan manipular, en términos abstractos y de una forma práctica y eficiente, los datos, de manera que no le sea necesario conocer el modo de almacenamiento de la información, ni el método de acceso empleado. Está compuesto por un lenguaje de definición de datos (DDL), un lenguaje de manipulación de datos (DML) y un lenguaje de consulta (SQL).

1.8.1 PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD.

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo

commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

1.9 Fundamentación de la herramienta, lenguajes y tecnologías.

La herramienta de desarrollo de software a utilizar en la propuesta de solución, es Visual Paradigm. Se toma esta elección principalmente por ser una herramienta robusta, multiplataforma, de fácil uso y que brinda la posibilidad de exportar documentos, además que permite el modelado de procesos de negocio con BPMN y diseñar con UML.

Se selecciona Java pues los programas escritos en este lenguaje pueden ejecutarse en cualquier tipo de hardware. Es un lenguaje portable, robusto, estable y sencillo de aprender. La notación de modelado de procesos de negocio BPMN ya que es gráficamente muy rica y permite modelar el negocio con buena claridad. La arquitectura de tres capas, ya que permite una gran organización entre los niveles y de ocurrir algún cambio en uno de ellos no afecta a los demás.

El gestor de base de datos debe estar basado en las licencias de software libre, presentar una estabilidad muy alta y gran seguridad de los datos. Como consecuencia de esto, se seleccionó PostgreSQL como gestor de base de datos a utilizar.

En el capítulo se exponen de manera detallada los principales conceptos que van a manejarse en el desarrollo de la investigación. Luego de explicar cada una de las metodologías, herramientas y lenguajes se plantea la justificación de la selección de cada una de ellas, siendo esto de vital importancia para el lector cuando vaya a avanzar hacia el segundo capítulo.

Capítulo 2 Características del Sistema.

En el capítulo se muestran las características que tendrá el sistema, partiendo de la descripción del negocio, la definición de los requerimientos del sistema, el modelo conceptual y la especificación de los requisitos de software.

2.1 Información asociada al proceso de desarrollo de software en la Universidad de las Ciencias Informáticas.

El sistema manejará un volumen de información bastante elevado, por lo que tiene que ser altamente eficiente y consumir escasos recursos. La información a manejar se estima que crezca a medida que el proceso de desarrollo en la UCI alcance las mejoras propuestas, por la importancia del sistema se considera imprescindible el conocimiento del tan amplio proceso de desarrollo de software, sus principales metodologías, lenguajes de programación, lenguajes de modelado, estándares de código, sistemas gestores de bases de datos y por último el modelo de calidad empleado.

La UCI con su estructura en centros productivos y como principal desarrolladora de software en el país se rige por los estándares internacionales de desarrollo de software, por lo que en su proceso de desarrollo priman los lineamientos de calidad con fines a alcanzar este año la certificación de nivel dos del modelo CMMI (Capability Madurity Model Integration) - modelo que contiene cinco niveles de madurez- convirtiéndose en la primera empresa cubana en alcanzar dicha certificación y una de las pocas del área del Caribe.

El proceso de desarrollo de software en la UCI se desarrolla utilizando dos sistemas operativos Windows y Linux, ambos en diferentes versiones, utiliza mayormente la metodología RUP, aunque últimamente luego de un estudio realizado a las metodologías ágiles ha comenzado a utilizarse tanto Programación extrema como SCRUM, solo que RUP sigue siendo la más usada por ser más robusta y factible para procesos de larga duración.

Muchos son los lenguajes utilizados en la universidad, de ellos para aplicaciones de desktop el más usado es Java, aunque el C# se utiliza bastante. En lo referente a gestores de Bases de Datos, se emplean varios aunque se propone usar PostgreSQL por estar basado en licencias de software libre además de ser bastante estable.

De manera general la universidad está inmersa en un proceso de mejoras, se espera no solo alcanzar un nivel 2 del modelo CMMI sino que finalmente se creen estándares en cuanto a metodologías, lenguajes de programación – tanto de desktop como web- , sistemas gestores de bases de datos, para lograr una uniformidad en los productos desarrollados en la universidad, en fin definir los productos con un sello único de desarrollo de software ajustando lo ya existente al proceso real de desarrollo de software. En los diferentes centros deben adoptarse medidas para lograr resultados relevantes y dejar siempre material que otros puedan utilizar e información que se pueda recuperar para avanzar más en el desarrollo de un software a la altura de un centro como la UCI.

2.2 Modelo matemático para el diseño del Sistema de Recuperación de la Información.

El modelo seleccionado para el diseño de la propuesta es el utilizado por el poderoso sistema Google, pues hasta el momento es un algoritmo robusto y eficiente, que utiliza los modelos matemáticos más usados explicados en el capítulo anterior: modelo de espacio vectorial y modelo booleano, sólo que reciben algunos aportes para formar un nuevo modelo el cual será analizado a continuación.

Para comenzar:

1-Cuestiones importantes a la hora de diseñar un buscador en la red:

- cómo almacenar la información;
- cómo actualizarla;
- cómo manejar/responder a peticiones;
- cómo buscar en las bases de datos.

2-Resultados de una búsqueda: ¿cómo mostrar los resultados y en qué orden?

Criterio de ordenación, una asignación de importancias a cada sitio de la red:

Sitios $-\rightarrow$ P1,. . . . , Pn

Importancias $-\rightarrow$ x1,. . . . , xn

Nota: no es lo mismo que un cierto término aparezca en una página en el título, en negrita, en un tipo de letra pequeña. Para búsquedas combinadas, tampoco es lo mismo que los términos buscados aparezcan "cerca" o "lejos" unos de otros.

2.2.1 Algoritmo (Page-Rank)

Se utiliza para ordenar los resultados de las búsquedas. Dicho algoritmo actúa sobre un grafo web de gran tamaño y basándose en ello, a cada página le asigna un valor en función del número de enlaces de otras páginas que le apuntan, el valor de esas páginas y otros criterios no públicos. El siguiente gráfico nos da una idea del problema:

La fórmula matemática que determina el Page-Rank de una página web es la siguiente:

$$PR(A) = (1 - d) + d * \left(\frac{PR(T_1)}{C(T_1)} + \ldots + \frac{PR(T_n)}{C(T_n)} \right)$$

Donde:

- PR(A) es el Page-Rank de la página A,
- PR(Ti) es el Page-Rank de las páginas Ti que enlazan a A,
- C(Ti) es el número de enlaces salientes de la página Ti,
- d es un factor de amortiguación que tiene un valor entre 0 y 1.

El algoritmo para el cálculo de las cantidades anteriores es complejo.
Sin entrar en mucho detalle podríamos decir que dicho algoritmo usa matrices de adyacencia entre páginas web, operaciones de diagonalización, el método de las potencias, el teorema de Perron-Frobënius aplicado a sistemas dinámicos.

Este algoritmo se selecciona para la futura implementación del sistema porque utiliza los modelos matemáticos más usados por todos los Sistemas de Recuperación de Información, y dichos modelos son los más sencillos de implementar, además de que los resultados que arroja son bastante satisfactorios.

2.3 Modelación del negocio

La modelación del negocio es utilizada para comprender el conjunto de procesos de negocio que tienen lugar dentro de una organización como paso previo para establecer los requisitos del sistema a desarrollar.

Tiene como objetivos:

- Entender la estructura y dinámica de la organización.
- Asegurar que los clientes, usuarios finales y desarrolladores tienen un entendimiento común de la organización.
- Derivar los requisitos del software necesarios para soportar la organización.

Sobre este último, Roger S. Pressman en su libro “Ingeniería del Software: Un enfoque práctico” plantea que se debe desarrollar un modelo de negocio y derivar los requisitos del sistema a partir de este. Esto garantiza que el software que se desarrolle responda a las necesidades y las condiciones de la organización. (12).

2.3.1 Descripción del negocio

Para comprender mejor el sistema es necesario comenzar por la modelación del negocio, en este caso se modelara por procesos. Este modelado consiste en describir la realidad de manera que pueda ser entendida y de ser necesario modificada con el fin de incorporarle mejoras. Es importante contar con una notación que permita modelar con la mayor claridad posible la esencia del negocio, por lo que se utiliza la notación BPMN.

En los Sistemas de Recuperación de Información los trabajadores están involucrados en uno o más procesos. El administrador es el responsable de la planeación, organización, dirección y control de las actividades. Estos trabajadores pueden convertirse en usuarios del sistema.

El proceso que se lleva a cabo en la Universidad de las Ciencias Informáticas en la recuperación de información asociada al proceso de desarrollo de software comienza una vez que el usuario solicita la información mediante una consulta. Este es el personal que recibe todos los beneficios de los procesos del negocio pues logran satisfacer su necesidad de información. El sistema es el encargado de procesar estas consultas, dándole al usuario una respuesta y permitiéndole realizar más búsquedas.

Los procesos fundamentales para el diseño del sistema de recuperación de información son los siguientes:

- Proceso de Indexado.
- Proceso de Búsqueda ó Consulta.
- Proceso de Recuperación.
- Proceso de Solicitud de Archivos.

Actores del negocio

Actores: Es cualquier individuo, organización, sistema o máquina que recibe un beneficio de los procesos asociados al *software.* Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio, enviando y recibiendo mensajes.

Usuario: Es el actor que solicita la información, beneficiándose de los procesos del negocio.

Trabajadores del Negocio

Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos. Se encuentran dentro de las fronteras del negocio y pueden convertirse en usuarios del sistema.

Administrador: Es la persona encargada de la planeación, organización, dirección y control de las actividades. Responsable de actualizar toda la información que se almacena en la base de datos del sistema.

Sistema: Es el responsable de procesar la solicitud y mostrar un resultado.

2.4 Mapa de Procesos

El objetivo de este mapa es mostrar los procesos de negocio que ocurren en la recuperación de la información asociada al proceso de desarrollo de software.

Un proceso de negocio es un conjunto de tareas relacionadas lógicamente para lograr un resultado de negocio definido. Cada proceso tiene sus entradas, funciones y salidas. Las entradas son requisitos que deben tenerse antes de que una función pueda ser aplicada. Cuando una función es aplicada a las entradas de un método, tendremos ciertas salidas resultantes.

Los procesos describen el trabajo, estos se caracterizan por ser mejorables, repetitivos, medibles y observables.

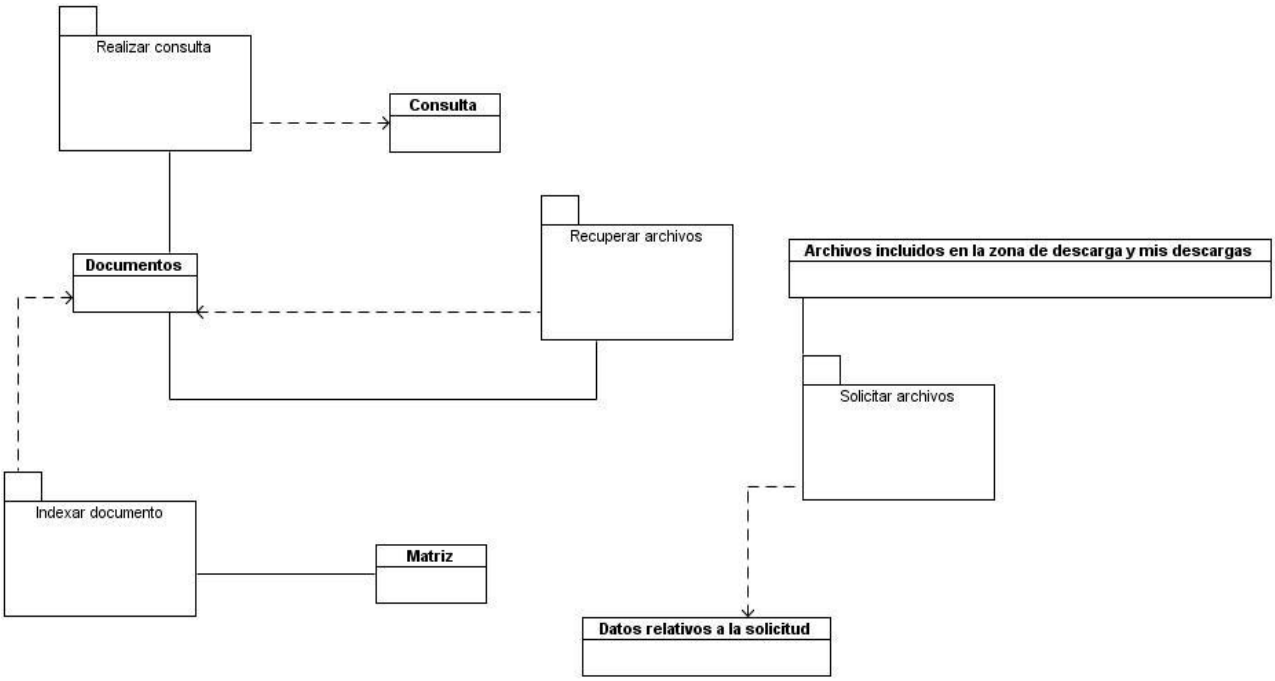


Figura 4 Mapa de Procesos del Negocio.

2.5 Especificación de los Procesos del Negocio

Descripción del proceso: Indexado

Objetivos: Indexar documentos.

Evento(s) que lo generan: Se inicia el proceso cuando se quiere incluir en la base de conocimientos un nuevo documento.

Precondiciones: Documento.

Pos condiciones: Se indexa un documento y se actualiza la base de conocimiento.

Clientes internos: Administrador y Sistema.

Clientes externos: Usuarios.

Entradas: Documentos.

Salidas: Una matriz.

Diagrama: Anexo # 1.

Descripción del proceso: Búsqueda ó Consulta.

Objetivos: Obtener la información buscada.

Evento(s) que lo generan: El proceso inicia cuando se introduce la consulta.

Precondiciones: Consulta.

Pos condiciones: Se ha realizado la consulta mostrando como respuesta un listado de documentos.

Clientes internos: Usuario y Sistema.

Clientes externos: Administrador.

Entradas: Consulta.

Salidas: Documentos.

Diagrama: Anexo # 2.

Descripción del proceso: Recuperación.

Objetivos: Realizar la recuperación de documentos.

Evento(s) que lo generan: El proceso se inicia con la inclusión del documento en la base de conocimientos.

Precondiciones: Documento.

Pos condiciones: Se ha realizado la ubicación de los archivos en el servicio de descarga a través de FTP ó HTTP.

Clientes internos: Administrador y Sistema.

Clientes externos: Usuarios.

Entradas: Documentos.

Salidas: Documentos.

Diagrama: Anexo # 3.

Descripción del proceso: Solicitud de Archivo.

Objetivos: Solicitar los archivos deseados.

Evento(s) que lo generan: El proceso se inicia cuando el usuario introduce los datos relativos a la solicitud.

Precondiciones: Datos.

Pos condiciones: Se ha realizado la solicitud, incluyendo los archivos en la zona de descarga y mis descargas.

Clientes internos: Usuario y Sistema.

Clientes externos: Administrador.

Entradas: Datos relativos a la solicitud.

Salidas: Archivos incluidos en la zona de descarga y mis descargas.

Diagrama: Anexo4.

2.6 Requisitos del software

"Los requisitos funcionales son los que definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Es importante que se describa el ¿Qué? y no el ¿Cómo? se deben hacer esas transformaciones. Estos requisitos al tiempo que avanza el proyecto de software se convierten en los algoritmos, la lógica y gran parte del código del sistema." (13)

2.6.1 Los requisitos funcionales son los siguientes:

RF1 Indexar Documento.

RF1.1 Permitir buscar información en base de conocimientos desde interfaz Web.

RF2 Realizar Consulta.

RF2.1 Mostrar parte ó resumen del archivo encontrado.

RF2.2 Permitir buscar por tipo de archivo, fecha de publicación, entre otros datos.

RF3 Recuperar Documento.

RF3.1 Permitir recuperar información relativa al resultado de la búsqueda.

RF3.2 Ordenar los resultados en dependencia del grado de similitud.

RF3.3 Acceder a las descargas desde FTP y HTTP.

RF4 Solicitar Archivos.

RF4.1 Permitir solicitar un documento no encontrado en la base de conocimiento.

RF4.2 Poder llenar control de mis solicitudes de descarga.

RF4.3 Recibir notificación de los cambios de estado de la solicitud.

2.6.2 Especificación de requisitos:

RF1 Indexar Documento

RF1.1 Permitir buscar información en base de conocimientos desde interfaz Web.

Conceptos tratados	**Conceptos**	**Atributos**
	Base de conocimiento, interfaz Web.	Acentos, token, palabras especiales
Precondiciones	**Precondiciones**	**Pre-requisito**
	Tener el documento que se quiere indexar.	Tener el documento
Descripción	1. El administrador tiene el documento que se quiere incluir en la base de conocimiento. 2. El sistema separa el documento en estructura y texto. 3. El sistema indexa el documento colocando la matriz en archivo *.XML. Si no se conoce la estructura y el texto ir al **Flujo Alterno 1.1**	
Validaciones	El sistema valida los datos.	
Post-condiciones	El sistema indexa documentos y permite buscar información desde interfaz Web.	
Post-requisito	Documento indexado.	

Flujo Alterno 1.1:

Descripción	1. El sistema elimina las palabras especiales, incluyendo acentos y caracteres extraños.

	2. El sistema realiza la separación de cada palabra en token. 3. El sistema realiza la representación de la matriz colocando cada token en una casilla de la misma. Volver al paso 3 del **RF1.1 Permitir buscar información en base de conocimientos desde interfaz Web.**

Tabla 2 Descripción Indexar Documento.

RF2 Realizar Consulta.

RF2.1 Mostrar parte ó resumen del archivo encontrado.

Conceptos tratados	**Conceptos**	**Atributos**
	Matriz, Ranking page, motor de búsqueda, consulta.	Interfaz, token.
Precondiciones	**Precondiciones**	**Pre-requisito**
	Realizar consulta.	Introducir consulta.
Descripción	1. El usuario introduce la consulta en lenguaje natural. 2. El sistema realiza el procesamiento de texto de la consulta en token. 3. El motor de búsqueda realiza la comparación entre la consulta con el archivo de indexación. 4. El sistema realiza la ubicación de los resultados en prioridades en forma matricial. 5. El sistema realiza la recuperación de archivos ubicando los resultados para realizar descargas. 6. El sistema muestra en la interfaz de usuario la lista de resultados.	
Validaciones	El sistema valida los datos.	
Post-condiciones	El sistema muestra los resultados.	
Post-requisito	Consulta introducida.	

Tabla 3 Descripción Mostrar parte ó resumen del archivo encontrado.

RF2 Realizar Consulta.

RF2.2 Permitir buscar por tipo de archivo, fecha de publicación, entre otros datos.

Conceptos tratados	**Conceptos**	**Atributos**
	Consulta, motor de búsqueda, búsqueda avanzada.	Token.
Precondiciones	**Precondiciones**	**Pre-requisito**
	Consulta.	Introducir consulta.
Descripción	1. El usuario introduce la consulta. 2. El sistema procesa el texto de consulta. 3. El sistema muestra la opción de consulta, introduciendo opciones avanzadas de búsqueda mediantes criterios: **Tipo de archivo, fecha de publicación, entre otros datos.** 4. El motor de búsqueda realiza la comparación entre la consulta con el archivo de indexación. 5. El sistema realiza la ubicación de los resultados en prioridades en forma matricial. 6. El sistema realiza la recuperación de archivos ubicando los resultados para realizar descargas. 7. El sistema muestra en la interfaz de usuario la lista de resultados.	
Validaciones	El sistema valida los datos.	
Post-condiciones	El sistema permite realizar opciones avanzadas de búsquedas.	
Post-requisito	Se realizó la consulta.	

Tabla 4 Descripción Permitir buscar por tipo de archivo, fecha de publicación, entre otros datos.

RF3 Recuperar Documento.

Conceptos tratados	**Conceptos**	**Atributos**
	Documento.	Archivos, fecha, código.
Precondiciones	**Precondiciones**	**Pre-requisito**
	Documento.	Documentos.
Descripción	1. El administrador introduce el nuevo documento en la base de conocimiento. 2. El sistema realiza el proceso de indexar documento y genera el nuevo archivo de indexación. 3. El sistema renombra los documentos, asigna código y fecha y colecciona los datos relativos al archivo. 4. El sistema inserta los datos relativos al archivo. 5. El sistema publica los archivos en el servicio de descarga a través de FTTP y HTTP.	
Validaciones	El sistema valida los datos.	
Post-condiciones	El sistema permite acceder a las descargas.	
Post-requisito	Documento recuperado.	

Tabla 5 Descripción Recuperar Documento.

RF4 Solicitar Archivos.

Conceptos tratados	**Conceptos**	**Atributos**
	Archivos.	Nombre, tipo documento.
Precondiciones	**Precondiciones**	**Pre-requisito**
	Insertar datos.	Solicitar archivo.
Descripción	1. El usuario introduce los datos relativos a la solicitud mediante los criterios: URL, nombre de archivo, tipo de documento. Si el archivo existe: ir al **Flujo Alterno 4.1** 2. El sistema inserta datos en la base de datos en caso de no encontrarse en la base de conocimiento. 3. El sistema inserta y muestra los datos en mis descargas con estado de ejecución de la misma. 4. El sistema realiza revisión y pone en descarga al archivo actualizando el estado de ejecución. 5. El sistema le notifica al usuario mediante correo que solicitó la descarga. 6. El sistema incluye el archivo en la zona de descarga y mis descargas.	
Validaciones	El sistema valida los datos.	
Post-condiciones	El sistema permite descargar archivos, solicitar documentos no encontrados en la base de conocimiento y recibir notificación de los cambios de la solicitud.	
Post-requisito	Se solicitaron los archivos incluyéndolos en la zona de descarga y mis descargas.	

Flujo Alterno 4.1:

Descripción	1. El sistema le presenta al usuario el archivo y algunas descargas. 2. El sistema realiza la presentación de información relativa, colocando la información en la interfaz de usuario de resultados. 3. Volver al paso 2 **RF4 Solicitar Archivos.**

Tabla 6 Descripción Solicitar Archivos.

2.6.3 Requisitos no funcionales

Los Requerimientos No Funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable.

Son los aspectos del sistema visibles para el usuario, que no están relacionados de forma directa con el comportamiento funcional del sistema. Los requerimientos no funcionales forman una parte significativa de la especificación.

Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, esto puede marcar la diferencia entre un producto bien aceptado y uno con poca aceptación (13).

RNF1 De disponibilidad.

RNF1.1 Los usuarios del sistema deben tener acceso (según sus permisos) en todo momento a la información solicitada.

RNF2 De Seguridad.

RNF2.1 La información será almacenada en bases de datos, dejando registro de todas las operaciones realizadas.

RNF2.2 El uso y manejo del sistema estará controlado. Toda la información podrá ser consultada solamente por el personal autorizado.

RNF3 De confiabilidad.

RNF3.1 Todas las salidas del sistema tienen que tener 100% de veracidad y precisión. RNF3.2 Toda la información está protegida del acceso no autorizado. Solo el personal acreditado podrá administrar la información solicitada.

RNF4 De ayuda y documentación en línea.

RNF4.1 El Sistema debe proporcionar en todo momento la documentación necesaria para que los usuarios puedan acceder a la misma en caso de algún inconveniente.

RNF5 De rendimiento.

RNF5.1 El sistema con el uso de las tecnologías Web debe tener un tiempo de respuesta rápido y eficiente, inferior a 10 segundos.

RNF5.2 Después de instalado el software, el mismo debe ser capaz de soportar gran cantidad de usuarios conectados simultáneamente.

RNF6 De hardware.

RNF6.1 Se necesita una PC que posea un procesador Pentium III o superior y 512 megabytes (MB) de memoria RAM como mínimo.

RNF7 De apariencia o interfaz externa.

RNF7.1 La interfaz en su totalidad debe ser amigable, sugerente, intuitiva e interactiva para que pueda captar la atención del usuario.

RNF7.2 Debe contener un diseño sencillo, con pocas imágenes y gráficos para acelerar la velocidad de respuesta del Sistema.

RNF7.3 La interfaz debe presentar solamente las funcionalidades del rol que esté utilizando el Sistema, para lograr la concentración del usuario en las actividades que esté realizando.

RNF8 De software.

RNF8.1 Las estaciones de trabajo clientes deben contar con el Sistema Operativo Windows XP, Linux NOVA u otra distribución.

RNF8.2 El Sistema utilizará para el almacenamiento de la información el gestor de base de datos PostgreSQL.

RNF9 De usabilidad.

RNF9.1 La interfaz debe ser fácil de usar para los diversos tipos de usuarios que interactúan con ella.

RNF9.2 La aplicación debe estar bien documentada con el fin de lograr el mejor uso de los servicios que la misma ofrecerá.

RNF10 De soporte.

RNF10.1 La instalación del Sistema debe ser lo más rápida y fácil posible.

RNF10.2 Es preciso disponer de una documentación apropiada del Sistema para agilizar su Mantenimiento y Configuración.

2.7 Modelo Conceptual

El objetivo de crear el modelo conceptual es aumentar la comprensión del problema y contribuir a esclarecer la terminología o nomenclatura del dominio. Puede verse como un modelo que comunica a los interesados cuáles son los términos importantes y cómo se relacionan entre sí. Este modelo es una representación de conceptos del mundo real y no de componentes del software y se representa mediante un diagrama de clases.

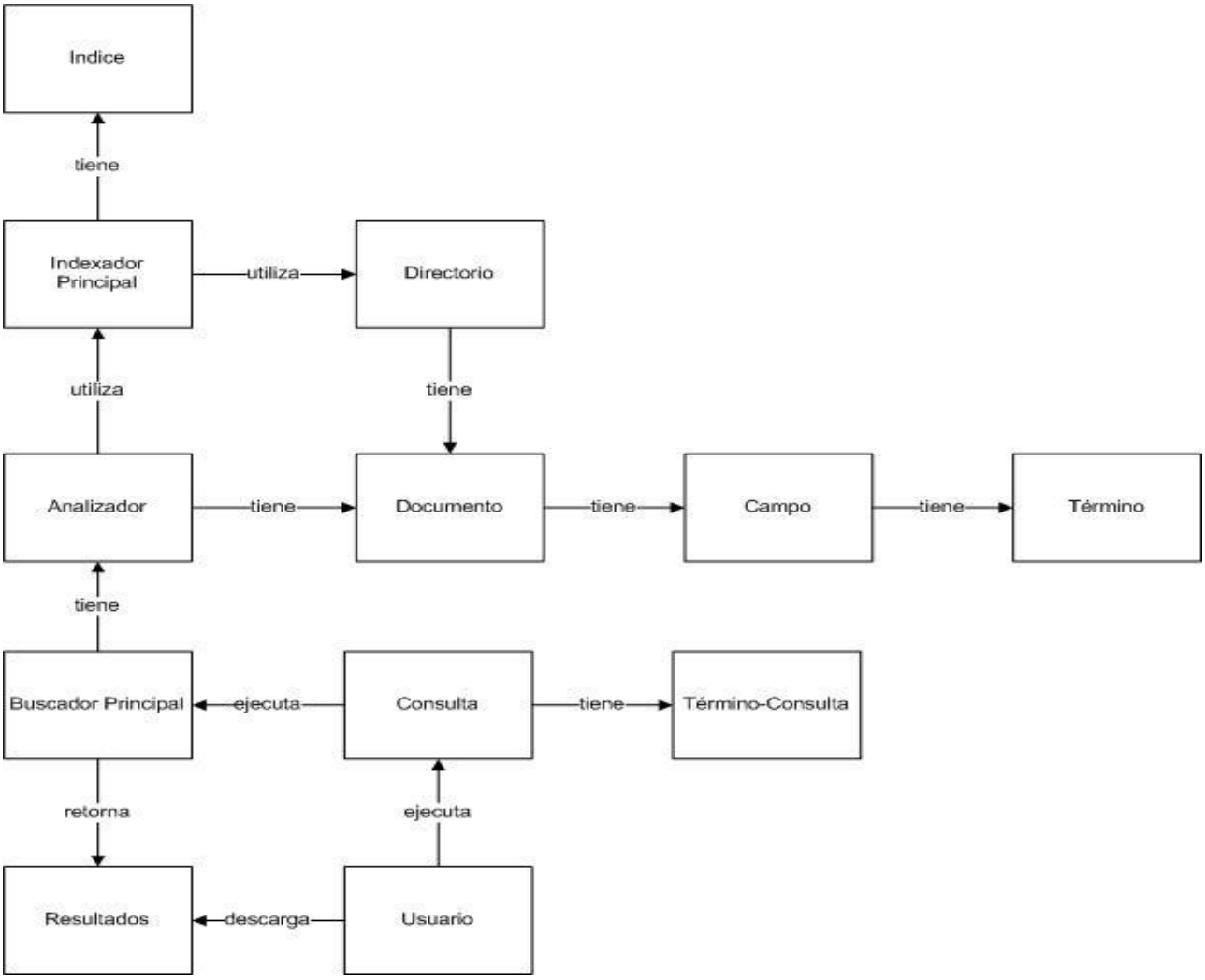


Figura 5 Modelo Conceptual.

2.8 Prototipo de Interfaz de usuario

La interfaz de usuario es el vínculo entre el usuario y el programa de computadora. La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso. Los prototipos de interfaz de usuario son una propuesta que presenta el equipo de desarrollo a los especialistas funcionales para que los mismos validen que la interfaz contempla las necesidades reales de los clientes y usuarios finales. Además son la base para que los desarrolladores implementen las

interfaces de usuario finales. Los prototipos de interfaz de usuario expuestos en el presente trabajo fueron validados por un especialista considerando que la interfaz final con la que va a interactuar el usuario debe ser sencilla, cumpliendo de manera eficiente con los requisitos funcionales propuestos y con la arquitectura seleccionada por el equipo de desarrollo. Los mismos fueron diseñados por la herramienta de modelado AxureRP Profesional. Los prototipos se muestran en los anexos (5, 6, 7,8)

En este capítulo se ha realizado la modelación de los procesos del negocio mediante la identificación de los procesos que se llevan a cabo en la recuperación de la información asociada al proceso de desarrollo de software, representándose en el mapa de procesos y descritos en los diagramas de procesos del negocio, confección de un glosario de un términos para mejor entendimiento. Se realizó la especificación de los requisitos.

A partir de este momento se puede comenzar a trabajar en el diseño del sistema de recuperación de información teniendo en cuenta que cumpla con todos los requisitos planteados en el capítulo.

Capítulo 3 Análisis y Diseño del Sistema.

En este capítulo se realizará el diseño del sistema. Se transforman los requisitos al diseño del futuro sistema utilizando una arquitectura correcta, se elaborarán los diagramas de clases del diseño y el de componentes.

3.1 Patrones.

Un patrón de diseño describe objetos y clases que se comunican entre si para resolver un problema de diseño. Estos patrones identifican: instancias, clases, roles, colaboraciones y la distribución de responsabilidades.

3.1.1 Patrones de asignación de responsabilidades.

Los patrones GRASP (General Responsibility Assignment Software Patterns, Patrones Generales de Software para Asignación de Responsabilidades), tienen como objetivo la descripción de los principios fundamentales del diseño de objetos para la asignación general de responsabilidades.
Los patrones utilizados fueron:

- Alta cohesión

Una clase con mucha cohesión es útil porque es fácil entenderla, darle mantenimiento y reutilizarla. Si alguien asume demasiadas responsabilidades, sobre todo las que debería delegar, no sería eficiente.

- Bajo acoplamiento.

El patrón bajo acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios y también más reutilizables, estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento.

- Experto.

Asigna una responsabilidad a la clase que tiene la información necesaria para cumplirla.

- Creador.

Guía la asignación de responsabilidades.

3.1.2 Patrón GOF.

Permite minimizar las dependencias y reducir la complejidad. Este patrón favorece a un Bajo Acoplamiento entre los clientes y los subsistemas, respondiendo a uno de los patrones GRASP, permiten variar las clases internas, de manera transparente a los clientes que las usan. (14)

Dentro de los patrones estructurales se encuentran:

- Fachada, propone implementar las clases con interfaces sencillas siendo más fáciles de usar.
- CamelCase, este patrón consiste en varias palabras unidas sin espacios entre ellas, mezclando minúscula y mayúsculas.

3.1.3 Patrón de arquitectura Modelo Vista Controlador.

El patrón de arquitectura conocido como Modelo-Vista-Controlador (MVC), separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario; es decir separa en tres capas diferentes los datos de una aplicación, la interfaz de usuario, y la lógica de control:

Modelo: Esta capa administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

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

Controlador: Esta capa controla el flujo de datos entre la vista y el modelo; es decir que responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. (15)

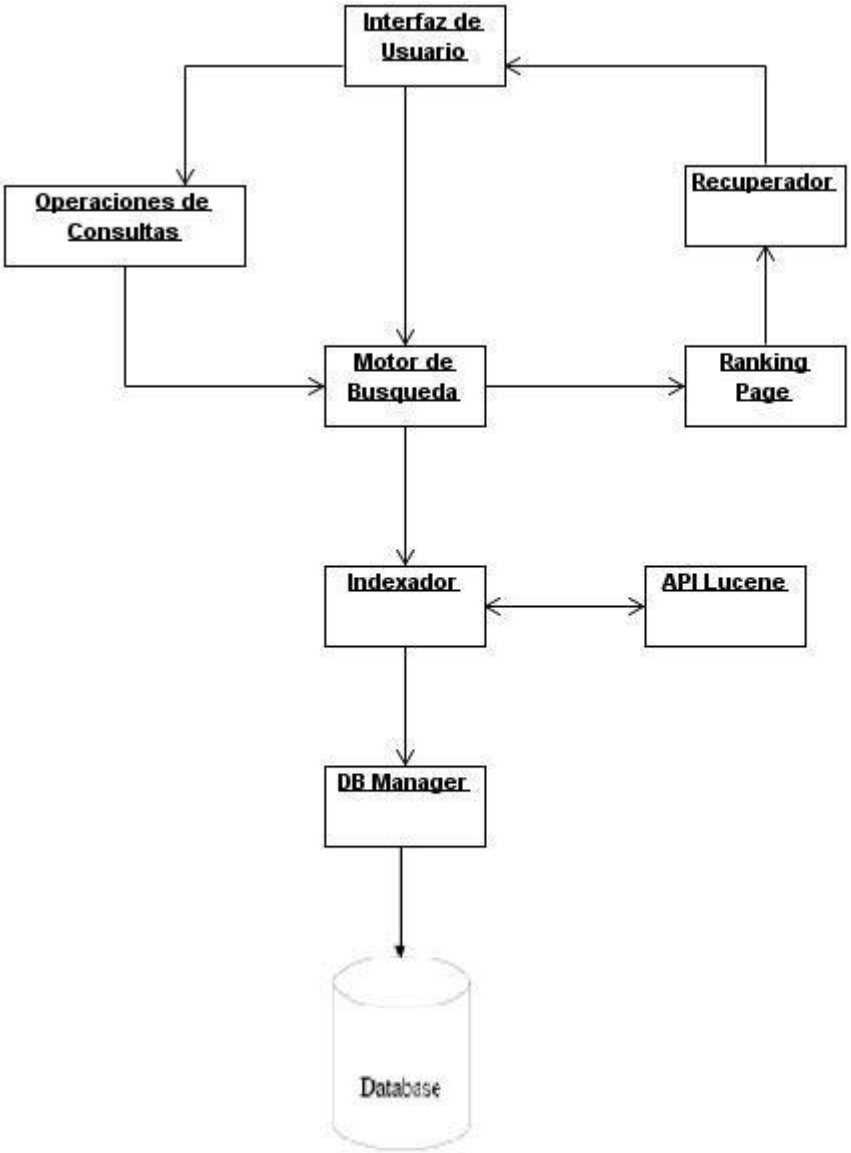


Figura 6 Arquitectura del SRI.

3.2 Diagramas de clases del diseño.

En el diseño se encuentra la forma del sistema. Además, el modelo de diseño es utilizado como entrada fundamental de las actividades de implementación.

Los siguientes diagramas muestran los diagramas de clases del diseño:

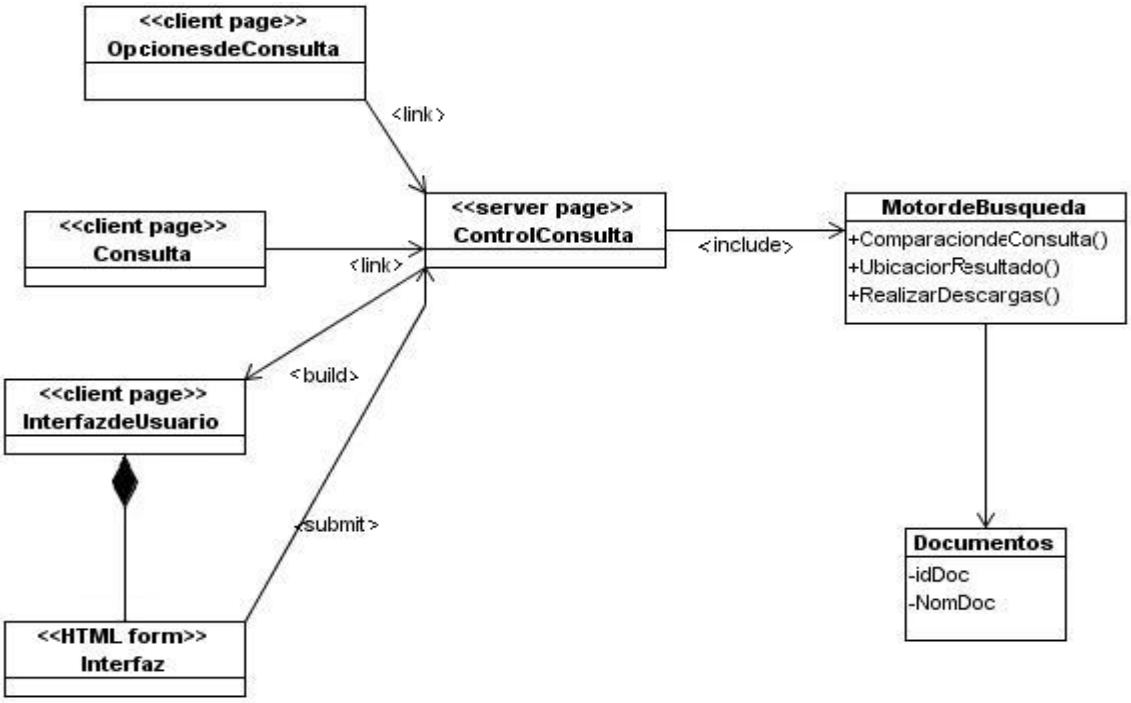


Figura 7 Diagrama de clase de diseño Consulta.

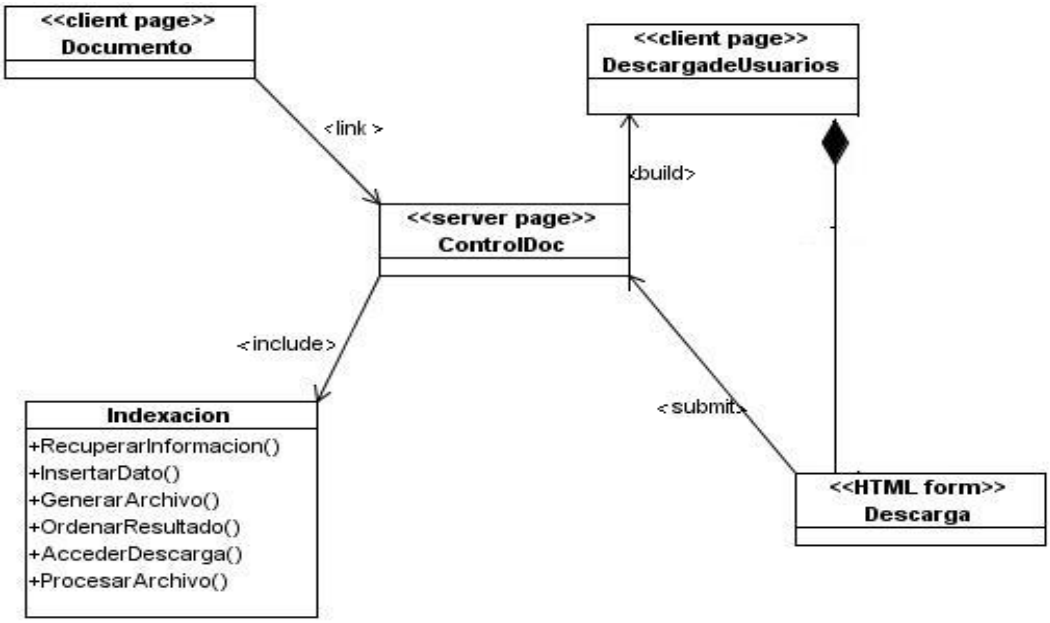


Figura 8 Diagrama de clases de diseño Recuperación.

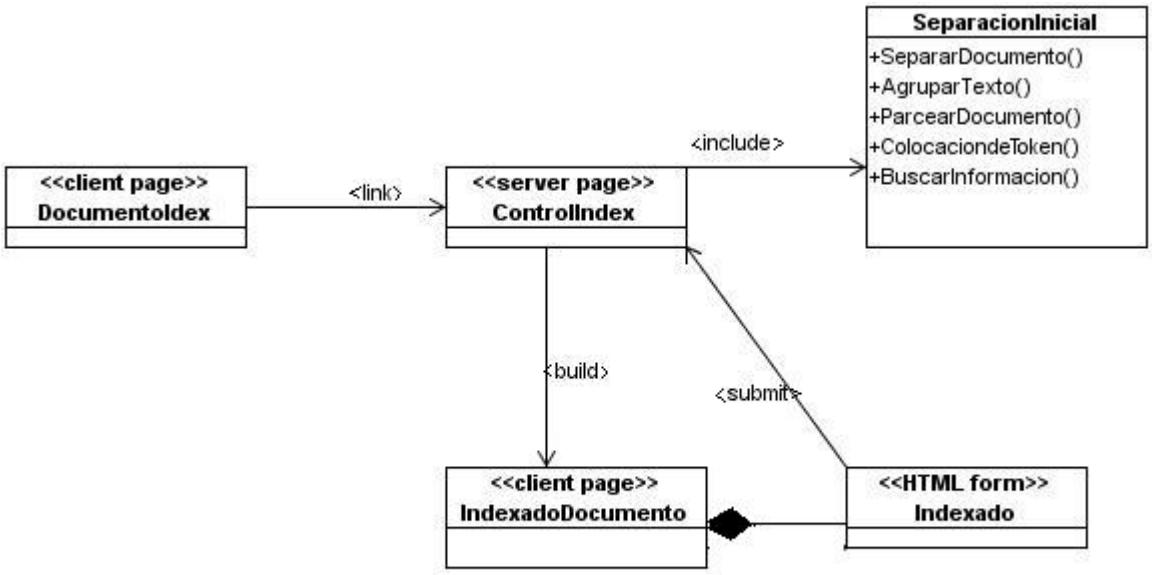


Figura 9 Diagrama de clases de diseño Indexado.

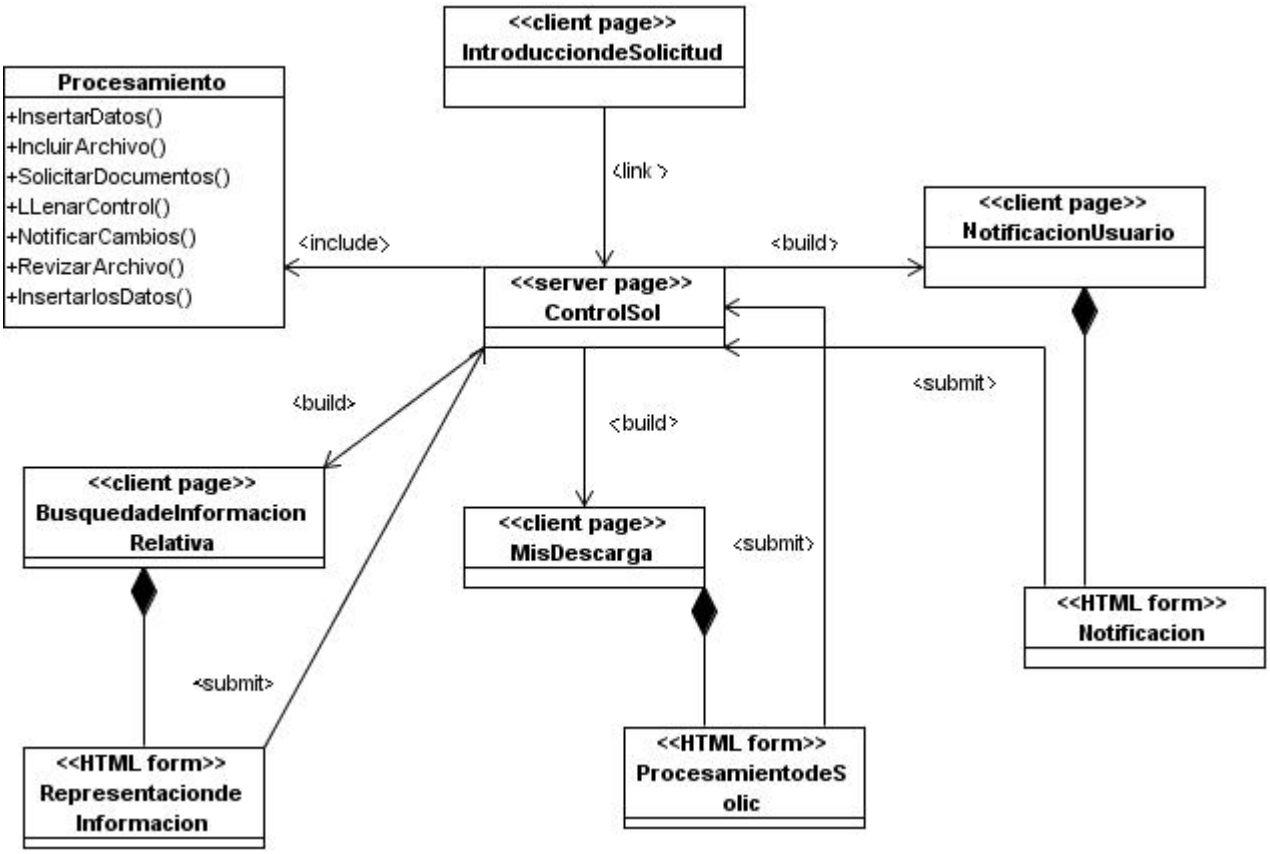


Figura 10 Diagrama de clase de diseño Solicitud de Archivo.

3.3 Descripciones de las clases de diseño.

Tabla 7 Descripción de la clase Consulta.

Nombre: Consulta	
Tipo de clase: Interfaz	
Descripción:	Introduce la consulta en lenguaje natural.

Tabla 8 Descripción de la clase ControlConsulta.

Nombre: ControlConsulta.	
Tipo de clase: Controladora.	
Para cada responsabilidad:	
Nombre:	ComparaciondeConsultas().
Descripción:	Se compara la consulta con el archivo de indexación.
Nombre:	UbicacionResultados().
Descripción:	Se ubican los resultados en prioridades en forma matricial.
Nombre:	RealizarDescargas().
Descripción:	Se ubican los resultados para realizar descargas.

Tabla 9 Descripción de la clase InterfazdeUsuario.

Nombre: InterfazdeUsuario.	
Tipo de clase: Interfaz.	
Descripción:	Se muestra la lista de resultados en la interfaz de usuario.

Tabla 10 Descripción de la clase OpcionesdeConsulta.

Nombre: OpcionesdeConsulta.	
Tipo de clase: Interfaz	
Descripción:	Se introducen las opciones avanzadas de búsqueda.

Tabla 11 Descripción de la clase Documento.

Nombre: Documento.	
Tipo de clase: Modelo(negocio)	
Descripción:	Inclusión del nuevo documento en la base de conocimientos.

Tabla 12 Descripción de la clase ControlDoc.

Nombre: ControlDoc.	
Tipo de clase: Controladora.	
Para cada responsabilidad:	
Nombre:	RecuperarInformacion().
Descripción:	Permite buscar información relativa al resultado de la búsqueda.
Nombre:	InsertarDato().
Descripción:	Se insertan los datos relativos al archivo.
Nombre:	GenerarArchivo().
Descripción:	Se genera el nuevo archivo de indexación.
Nombre:	ProcesarArchivo().
Descripción:	Se renombran los documentos, se asigna código y fecha y se coleccionan datos relativos al archivo.
Nombre:	OrdenarResultado().
Descripción:	Se ordenan los resultados en dependencia del grado de similitud.
Nombre:	AccederDescarga().
Descripción:	Se acceden a las descargas desde FTP y HTTP.

Tabla 13 Descripción de la clase DescargadeUsuarios.

Nombre: DescargadeUsuarios.	
Tipo de clase: Modelo(negocio)	
Descripción:	Se publica el archivo en el servicio de descarga a través de FTP ó HTTP.

Tabla 14 Descripción de la clase DocumentoIndex.

Nombre: DocumentoIndex.	
Tipo de clase: Modelo(negocio)	
Descripción:	Documento nuevo que se desea incluir en la base de conocimiento.

Tabla 15 Descripción de la clase ControlIndex.

Nombre: ControlIndex.	
Tipo de clase: Controladora.	
Para cada responsabilidad:	
Nombre:	SepararDocumento().
Descripción:	Se separa el documento en estructura y texto.
Nombre:	AgruparTexto().
Descripción:	Se separa cada palabra en un token.
Nombre:	ParsearDocumento().
Descripción:	Se eliminan las palabras especiales, incluyendo acentos caracteres extraños.
Nombre:	ColocaciondeToken().
Descripción:	Se coloca cada token en una casilla de la matriz de token.
Nombre:	BuscarInformacion().

Descripción:	Se permite buscar información en base de conocimientos desde interfaz web.

Tabla 16 Descripción de la clase IndexadoDocumento.

Nombre: IndexadoDocumento.	
Tipo de clase: Modelo(negocio)	
Descripción:	Se coloca la matriz en archivo *.xml.

Tabla 17 Descripción de la clase IntroduccióndeSolicitud.

Nombre: **IntroduccióndeSolicitud.**	
Tipo de clase: Interfaz	
Descripción:	El usuario introduce a través de la interfaz todos los datos relativos a la solicitud, URL, nombre de archivo y tipo de documento.

Tabla 18 Descripción de la clase ControlSol.

Nombre: **ControlSol.**	
Tipo de clase: Controladora.	
Para cada responsabilidad:	
Nombre:	InsertarDatos().
Descripción:	Se insertan los datos en la base de datos de no encontrarse en la base de conocimientos.
Nombre:	IncluirArchivo().
Descripción:	Se incluye el archivo en la zona de descarga y mis descargas.
Nombre:	SolicitarDocumentos().
Descripción:	Se permite solicitar un documento no encontrado en la base de conocimientos.
Nombre:	LLenarControl().
Descripción:	Se puede llenar control de mis solicitudes de descarga.

Nombre:	NotificarCambios().
Descripción:	Se recibe notificación de los cambios de estado de la solicitud.
Nombre:	RevizarArchivo().
Descripción:	Se realiza la revisión y puesta en descarga del archivo y se actualiza el estado de ejecución.
Nombre:	InsertarlosDatos().
Descripción:	Se insertan los datos en mis descargas con estado de ejecución de la misma.

Tabla 19 Descripción de la clase BusquedadeInformacionRelativa.

Nombre: BusquedadeInformacionRelativa.	
Tipo de clase: Modelo(negocio)	
Descripción:	Se le muestra al usuario si el archivo existe y algunos ya descargados.

Tabla 20 Descripción de la clase MisDescarga.

Nombre: **MisDescarga.**	
Tipo de clase: Modelo(negocio)	
Descripción:	Se muestran los datos en mis descargas con estado de ejecución de la misma.

Tabla 21 Descripción de la clase NotificacionUsuario.

Nombre: NotificacionUsuario.	
Tipo de clase: Modelo(negocio)	
Descripción:	Se le notifica al usuario mediante un correo que solicito la descarga.

En este capítulo se realizó el diseño del sistema cumpliendo los objetivos trazados, así como la realización de los diagramas de clases del diseño correspondientes con sus descripciones, obteniéndose los resultados esperados.

Conclusiones:

En el desarrollo del presente trabajo de diploma se realizó un estudio de las herramientas, metodologías, tecnologías, notación de modelado y lenguajes, estos se ajustan a las características necesarias para llevar a cabo su implementación. Así se definieron, como lenguaje de programación: Java, como gestor de base de datos: PostgreSQL, como notación de modelado del negocio: BPMN y como herramienta de modelado Visual Paradigm para las clases de diseño y AxureRP Profesional para los prototipos de interfaz de usuario. Se concluye que:

Se logró proponer el análisis y diseño del sistema, obteniéndose:

- Un algoritmo eficiente para la devolución de los resultados.
- Una arquitectura única para el almacenamiento de la información asociada al proceso de desarrollo de software.

Recomendaciones

Durante todo el desarrollo del trabajo han surgido algunas ideas que podrían ser desarrolladas en un futuro, con el objetivo de fortalecer el sistema propuesto, se recomienda lo siguiente:

- Implementar para la primera versión los procesos del Sistema propuestos en el trabajo de diploma.
- Continuar con el estudio de los Sistemas de Recuperación de Información, con vistas a lograr una mayor experiencia para las futuras mejoras a la propuesta.

Referencias Bibliográficas

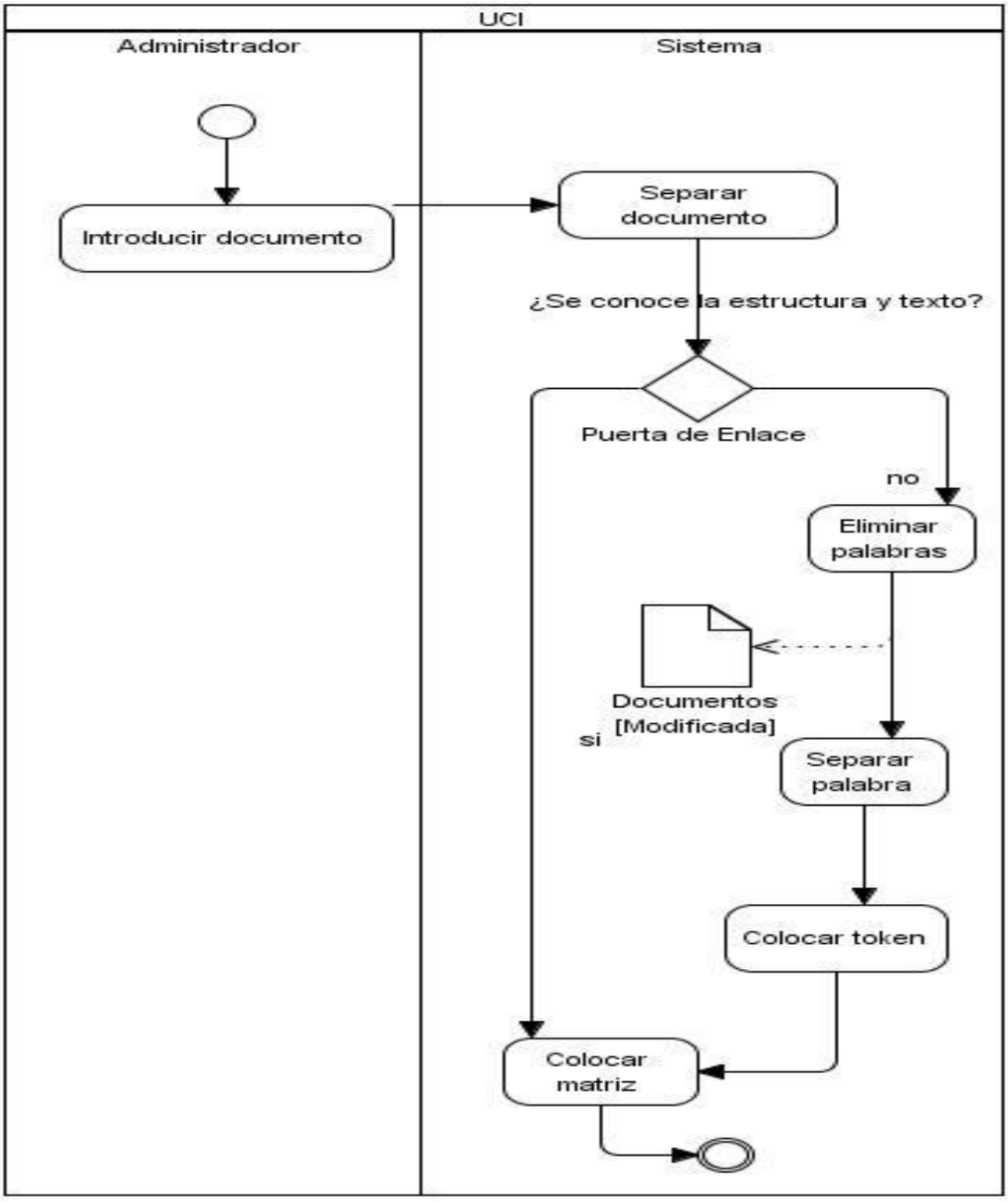
1. **Docentes de la Facultad de Ingeniería de Sistemas e Informática, Universidad Nacional Mayor de San Marcos, Lima-Perú.** sisbib. *sisbib.* [En línea] 2004. [Citado el: 6 de 1 de 2010.] http://sisbib.unmsm.edu.pe/bibvirtualdata/publicaciones/risi/N1_2004/a07 .pdf.
2. **GIL-LEIVA, Isidoro**. *Sistema para la Indización Semi-Automática (SISA) de Artículos de Revista de Biblioteconomía y Documentación*. *II Jornadas de Tratamiento y Recuperación de Información, septiembre 2003,* Leganés (Madrid), p. 228-232.
3. **SlideShare Inc.** SlideShare . *SlideShare .* [En línea] SlideShare Inc, 2009 . [Citado el: 8 de Enero de 2010.] http://www.slideshare.net.
4. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Addison Wesley, 2000. ISBN 84-7829-036-2.
5. **Larman, Craig.** *Agile and iterative development: a manager's guide.* Reino Unido (Inglaterra) : Addison-Wesley, 2003. ISBN: 978-0-13-111155-4.
6. **Boggs, Michael y Boggs, Wendy.** *Mastering UML with Rational Rose 2002.* USA : SYBEX Inc, 2002. ISBN:0782140173
7. **Stephen A, White.** *IBM Corporation. Introduction to BPMN.*
8. *Modelo Sistémico para la adopción tecnológica: caso herramientas CASE.* **Mendoza Morales, Luis Eduardo y Pérez, María Angélica.** Venezuela : s.n., 2004. ISSN 0001-5504.
9. **Hoffer, Jeffrey A.** *Modern systems analysis and design.* Upper Saddle River, N.J. : Pearson Prentice Hall, 2004. ISBN.
10. **Sánchez Freyre, Ana Lis y Campo, Isel.** *Análisis y Diseño del Proceso de Administración y Control de la Información para la Sección Sindical Vicerrectoría Primera.* Ciudad de la Habana : s.n., 2009.
11. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico.* s.l. : Vol. I.1-Ingeniería de Software 1. Conferencia #4. Fase de Inicio. Flujo de trabajo de requerimientos. Modelo de Diseño. Curso 2008-2009, 1998.
12. **2008.** Scribd. *UML - Analisis del negocio.* [Online] 04 18, 2008. [Cited: 01 13, 2010.] http://www.scribd.com/doc/2568110/UML-Analisis-del-negocio.

13. **Fernández, Yanisleivi Valdés; Costilla, Yoanis.** *PROPUESTA INICIAL DE UN PROCEDIMIENTO, PARA EL MODELADO DE NEGOCIO Y LA GESTIÓN DE REQUISITOS DE PROYECTOS PRODUCTIVOS.*
14. **Larman, Craig.** *UML y Patrones.* 1999.
15. *Arquitectura y Patrones de diseño.* **Colectivo de autores.** 2008-2009.
16. **Dominich, Sandor.** *A unified mathematical definition of classical information retrieval.* New York : John Wiley & Sons, Inc, 2000. ISSN:0002-8231 .
17. **The Apache Software Foundation.** Lucene. *Lucene.* [En línea] Lucid Imagination, 9 de Mayo de 2010. [Citado el: 20 de Marzo de 2010.] http://lucene.apache.org/.
18. **Baeza-Yates, Ricardo y Ribeiro-Neto, Berthier.** *Modern Information Retrieval .* Wokingham, UK : Addison-Wesley (ACM Press series) , 1999. ISBN.
19. **Hernández Sampieri, Roberto, Fernández Collado, Carlos y Baptista Lucio, Pilar.** *Metodología de la investigación.* Mexico : McGRAW-HILL INTERAMERICANA EDITORES, S. A, 1998. ISBN 970-10-1899-0 .
20. **Fernández Gallardo, Pablo.** UAM. *UAM.* [En línea] 4 de Mayo de 2004. [Citado el: 10 de Febrero de 2010.] http://www.uam.es.
21. **Pinto Molina, María.** mariapinto. *mariapinto.* [En línea] Maria Pinto Molina, 15 de Octubre de 2004. [Citado el: 15 de Diciembre de 2009.] http://www.mariapinto.es.
22. **Object Management Group, Inc.** BPMN. *BPMN.* [En línea] Object Management Group, Inc., 1997. [Citado el: 20 de Enero de 2010.] http://www.bpmn.org/.

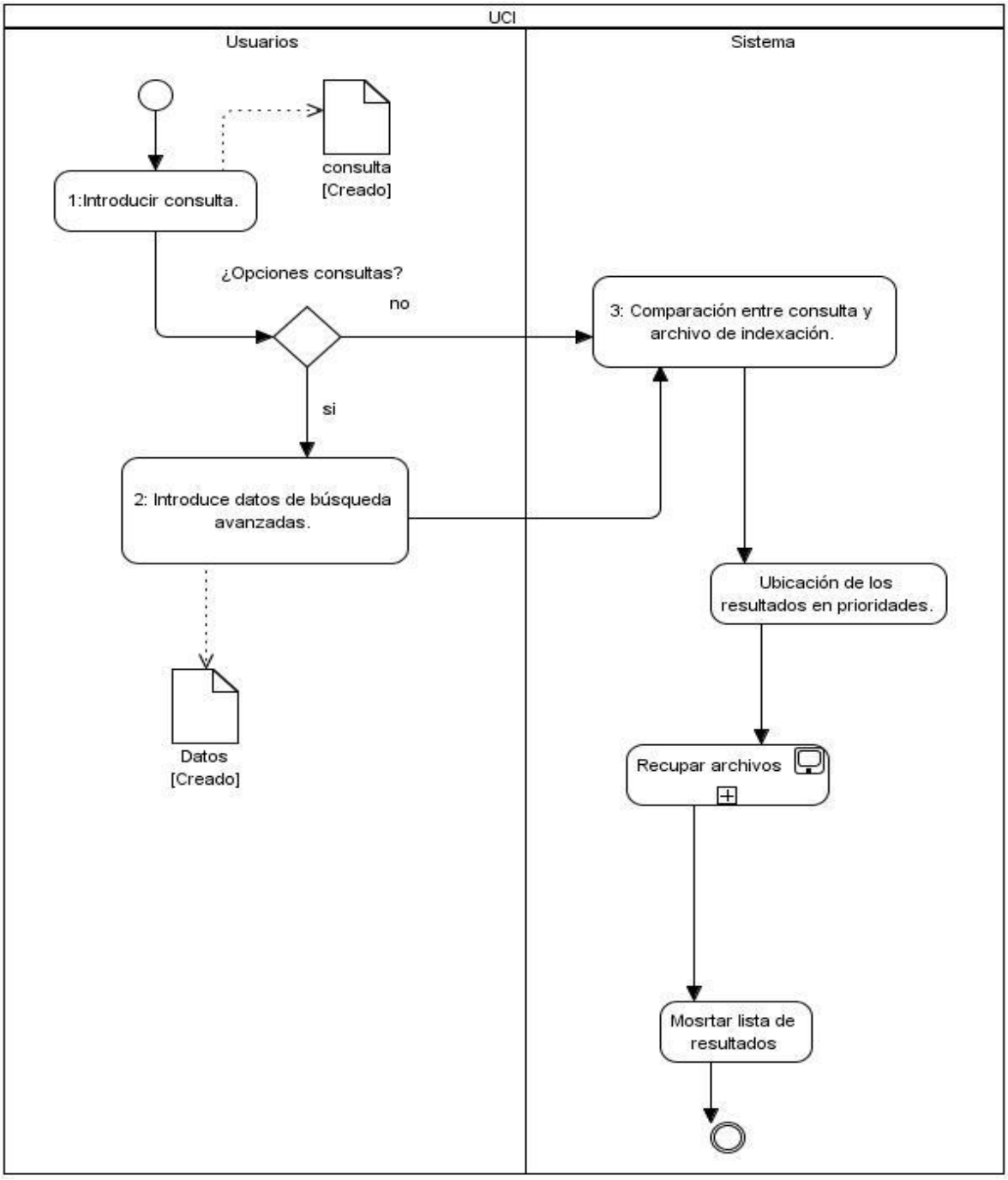
Bibliografía

1. **Docentes de la Facultad de Ingeniería de Sistemas e Informática, Universidad Nacional Mayor de San Marcos, Lima-Perú.** sisbib. *sisbib.* [En línea] 2004. [Citado el: 6 de 1 de 2010.] http://sisbib.unmsm.edu.pe/bibvirtualdata/publicaciones/risi/N1_2004/a07 .pdf.
2. **GIL-LEIVA, Isidoro**. *Sistema para la Indización Semi-Automática (SISA) de Artículos de Revista de Biblioteconomía y Documentación*. *II Jornadas de Tratamiento y Recuperación de Información, septiembre 2003,* Leganés (Madrid), p. 228-232.
3. **SlideShare Inc.** SlideShare . *SlideShare .* [En línea] SlideShare Inc, 2009 . [Citado el: 8 de Enero de 2010.] http://www.slideshare.net.
4. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Addison Wesley, 2000. ISBN 84-7829-036-2.
5. **Larman, Craig.** *Agile and iterative development: a manager's guide.* Reino Unido (Inglaterra) : Addison-Wesley, 2003. ISBN: 978-0-13-111155-4.
6. **Boggs, Michael y Boggs, Wendy.** *Mastering UML with Rational Rose 2002.* USA : SYBEX Inc, 2002. ISBN:0782140173
7. **Stephen A, White.** *IBM Corporation. Introduction to BPMN.*
8. *Modelo Sistémico para la adopción tecnológica: caso herramientas CASE.* **Mendoza Morales, Luis Eduardo y Pérez, María Angélica.** Venezuela : s.n., 2004. ISSN 0001-5504.
9. **Hoffer, Jeffrey A.** *Modern systems analysis and design.* Upper Saddle River, N.J. : Pearson Prentice Hall, 2004. ISBN.
10. **Sánchez Freyre, Ana Lis y Campo, Isel.** *Análisis y Diseño del Proceso de Administración y Control de la Información para la Sección Sindical Vicerrectoría Primera.* Ciudad de la Habana : s.n., 2009.
11. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico.* s.l. : Vol. I.1-Ingeniería de Software 1. Conferencia #4. Fase de Inicio. Flujo de trabajo de requerimientos. Modelo de Diseño. Curso 2008-2009, 1998.
12. **2008.** Scribd. *UML - Analisis del negocio.* [Online] 04 18, 2008. [Cited: 01 13, 2010.] http://www.scribd.com/doc/2568110/UML-Analisis-del-negocio.

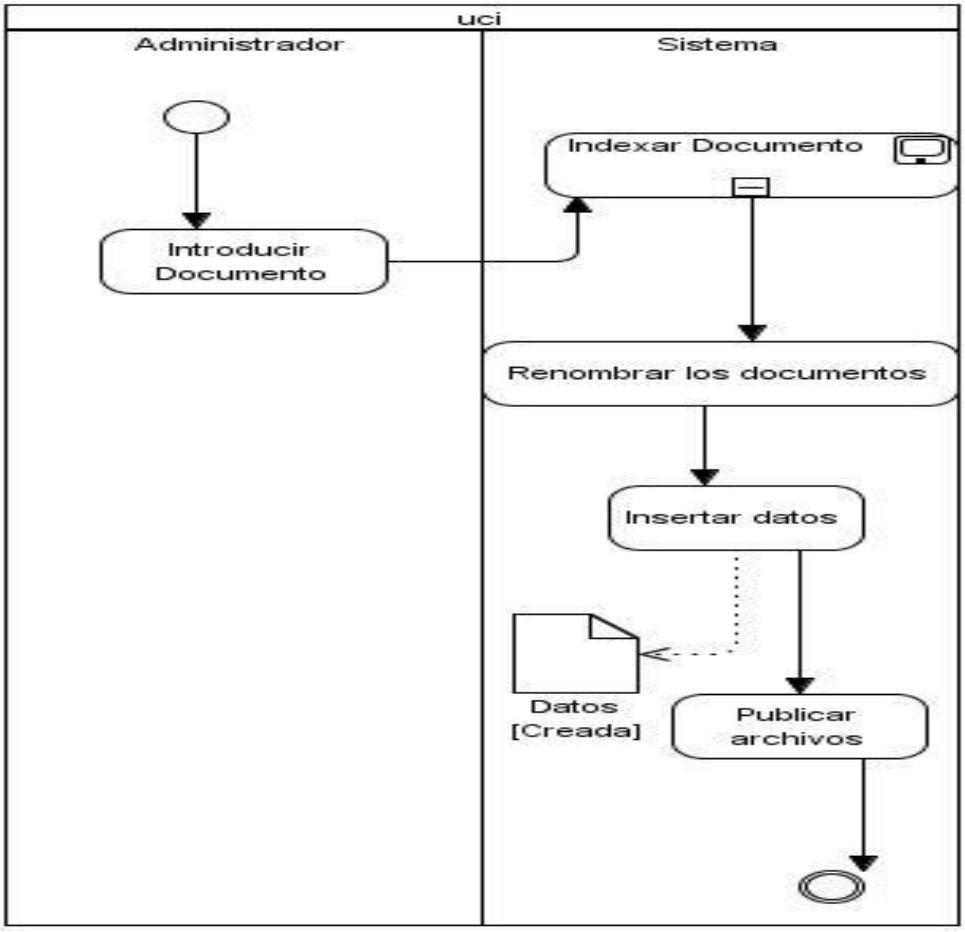
Anexo # 1 Descripción del proceso Indexar documentos



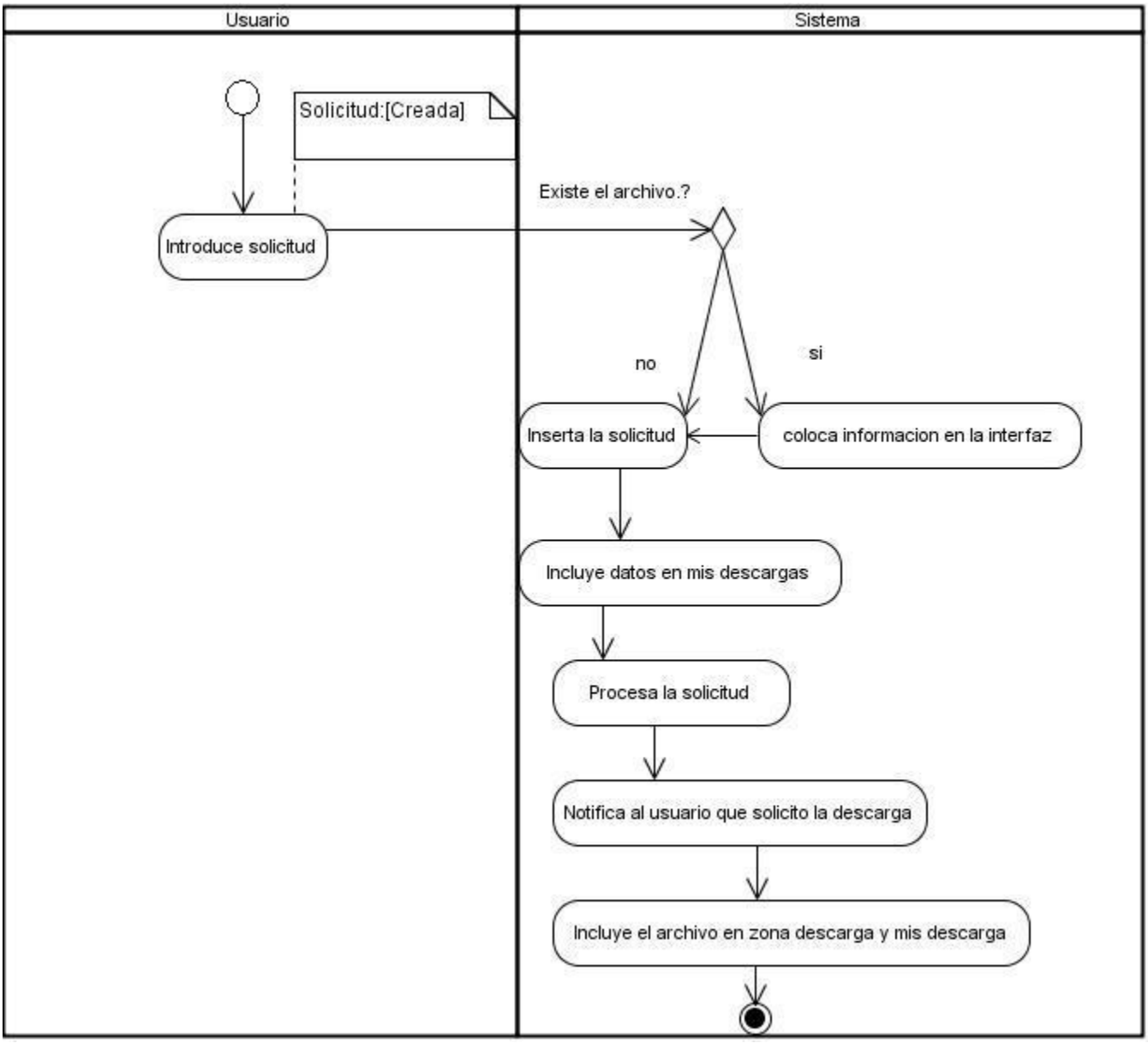
Anexo # 2 Descripción del proceso Búsqueda ó Consulta.



Anexo # 3 Descripción del proceso Recuperación.

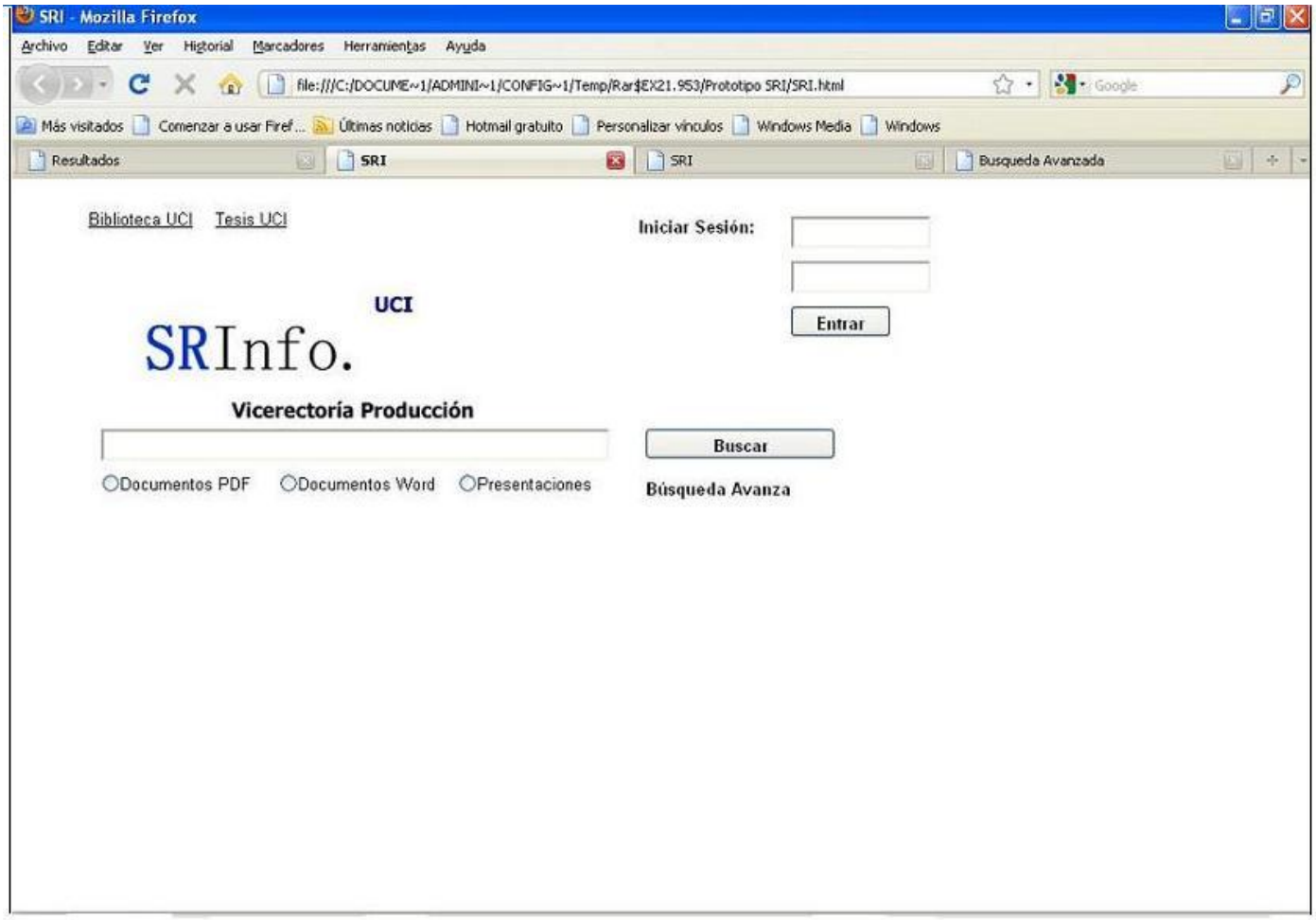


Anexo # 4 Descripción del proceso Recuperación.

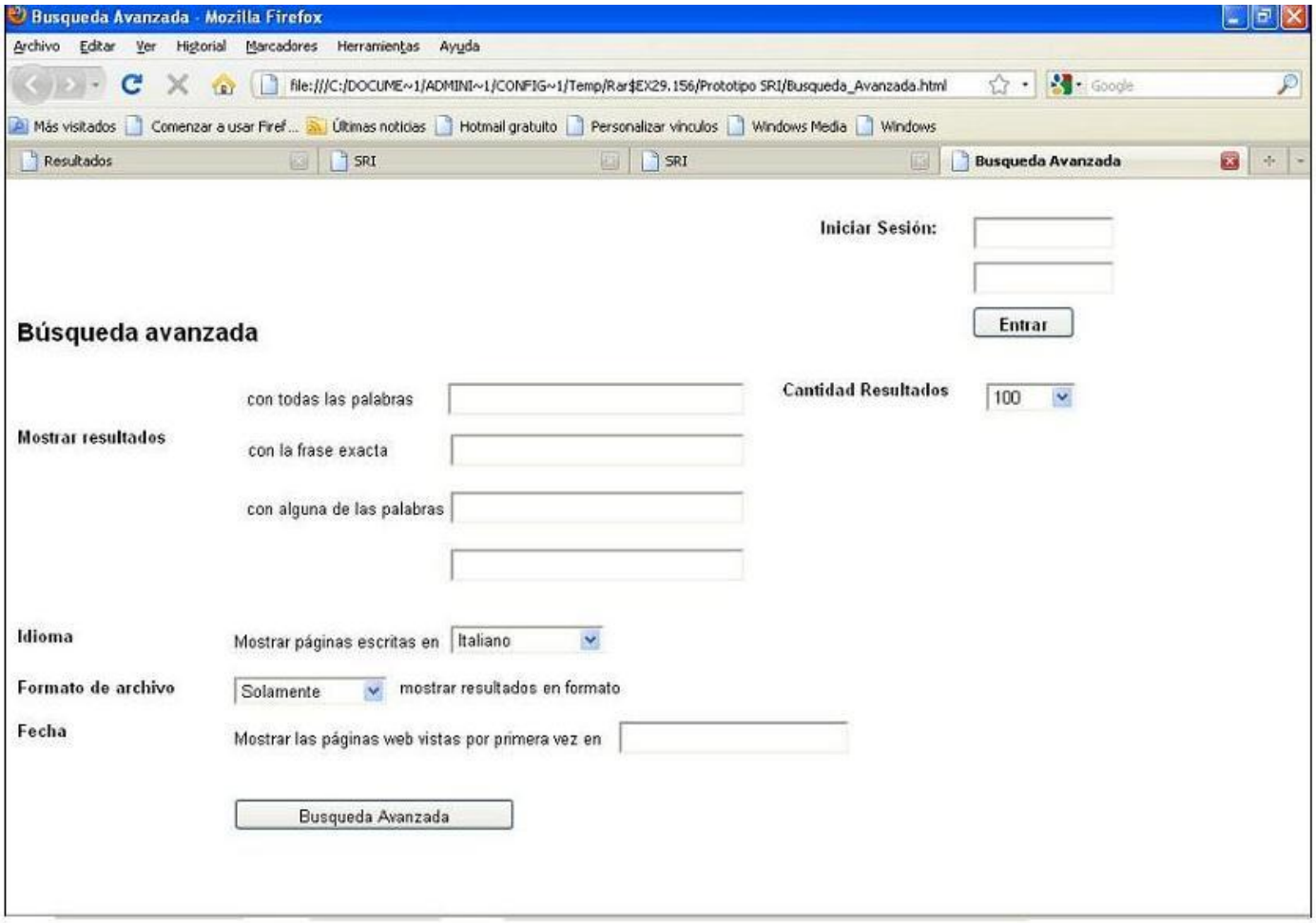


Anexo # 5 Prototipo de interfaz de usuario

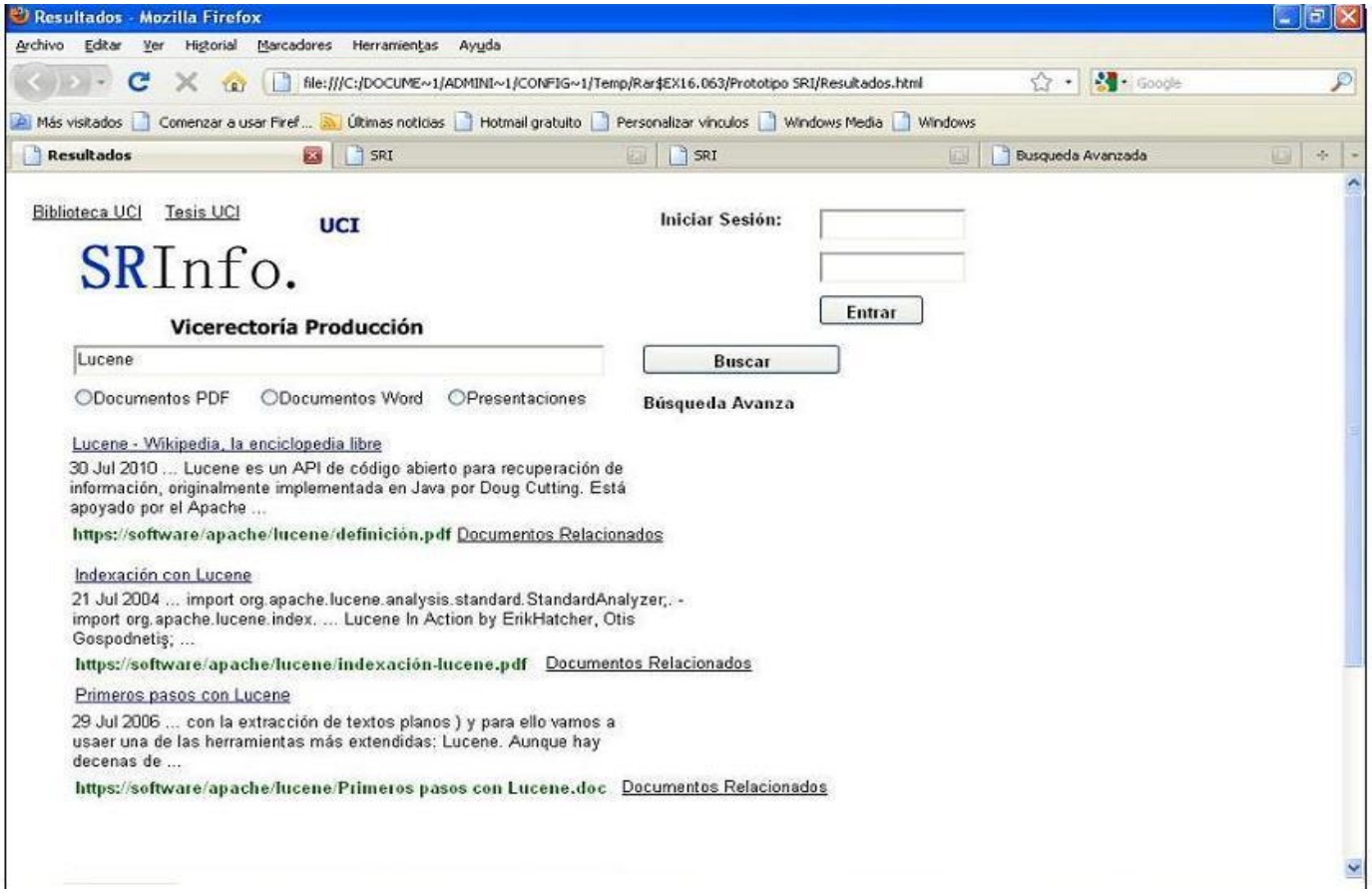
Anexo # 6 Prototipo de interfaz de usuario.

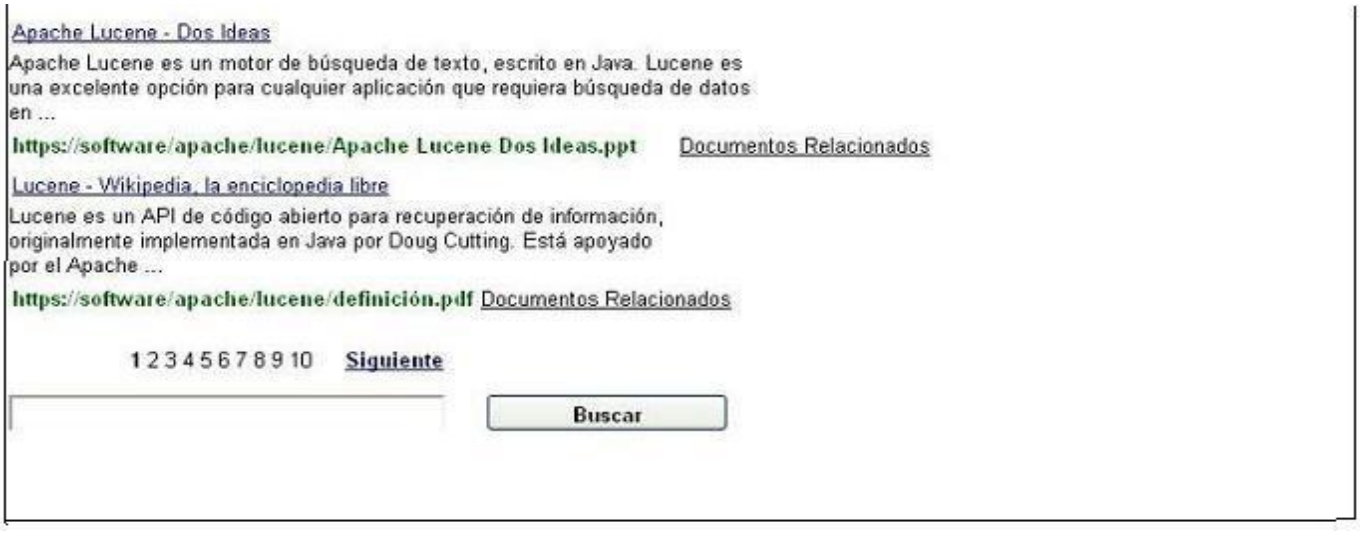


Anexo # 7 Prototipo de interfaz de usuario opciones avanzadas.



Anexo # 8 Prototipo de interfaz de usuario mostrando los resultados.


Resultados - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda
file:///C:/DOCUME~1/ADMINI~1/CONFIG~1/Temp/Rar$EX16.063/Prototipo SRI/Resultados.html
Más visitados Comenzar a usar Firef... Últimas noticias Hotmail gratuito Personalizar vínculos Windows Media Windows
Resultados
SRI
SRI
Busqueda Avanzada
Biblioteca UCI Tesis UCI
UCI
Iniciar Sesión:
Entrar
SRInfo.
Vicerectoría Producción
Lucene
Buscar
Documentos PDF Documentos Word Presentaciones
Búsqueda Avanza
Lucene - Wikipedia, la enciclopedia libre
30 Jul 2010 ... Lucene es un API de código abierto para recuperación de información, originalmente implementada en Java por Doug Cutting. Está apoyado por el Apache ...
https://software/apache/lucene/definición.pdf Documentos Relacionados
Indexación con Lucene
21 Jul 2004 ... import org.apache.lucene.analysis.standard.StandardAnalyzer;. - import org.apache.lucene.index. ... Lucene In Action by ErikHatcher, Otis Gospodnetiş; ...
https://software/apache/lucene/indexación-lucene.pdf Documentos Relacionados
Primeros pasos con Lucene
29 Jul 2006 ... con la extracción de textos planos) y para ello vamos a usaer una de las herramientas más extendidas: Lucene. Aunque hay decenas de ...
https://software/apache/lucene/Primeros pasos con Lucene.doc Documentos Relacionados


Apache Lucene - Dos Ideas
Apache Lucene es un motor de búsqueda de texto, escrito en Java. Lucene es una excelente opción para cualquier aplicación que requiera búsqueda de datos en ...
https://software/apache/lucene/Apache Lucene Dos Ideas.ppt Documentos Relacionados
Lucene - Wikipedia, la enciclopedia libre
Lucene es un API de código abierto para recuperación de información, originalmente implementada en Java por Doug Cutting. Está apoyado por el Apache ...
https://software/apache/lucene/definición.pdf Documentos Relacionados
1 2 3 4 5 6 7 8 9 10 Siguiente
Buscar

Anexo8.

Glosario de términos

BPMN (Business Process Modeling Notation): Es una notación que modela los procesos de negocio, basada en diagramas de flujo fácil de entender.

Patrones de diseño: (*Design patterns*) son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño es una solución a un problema de diseño.

Procesos del negocio: Conjunto de tareas relacionadas lógicamente para lograr un resultado de negocio definido.

Requisito: Condición o capacidad que debe cumplir un sistema.

Sistema gestor de base de datos: Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.

SQL (Structured Query Language): Conjunto estándar de comandos para gestionar bases de datos relacionales por sus mismas características relacionales.

SRI*:* Sistemas de Recuperación de Información.

UML (Unified Modeling Language): Lenguaje gráfico que brinda un vocabulario y reglas para especificar, construir, visualizar y documentar los artefactos de un sistema utilizando el enfoque orientado a objetos.