DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

TRABAJO DE TITULACIÓN PREVIO LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS E INFORMÁTICA

TEMA: "DESARROLLO DE UN WEB CRAWLER PARA EL SISTEMA DE RECOMENDACIÓN ESCOLÁSTICO AQUINO BASADO EN PERFILES PARA LOS POSTULANTES DE LAS BECAS DEL SENESCYT"

AUTORES: NATALIA PAOLA CERÓN ARMAS
MIREYA ESTEFANÍA CHILLÁN CUSI

DIRECTOR: ING. MAURICIO CAMPAÑA

SANGOLQUÍ, NOVIEMBRE 2015

CERTIFICACIÓN

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, "***DESARROLLO DE UN WEB CRAWLER PARA EL SISTEMA DE RECOMENDACIÓN ESCOLÁSTICO AQUINO BASADO EN PERFILES PARA LOS POSTULANTES A BECAS DEL SENESCYT***" realizado por las señoritas ***CERÓN ARMAS, NATALIA PAOLA Y CHILLAN CUSI, MIREYA ESTEFANÍA***, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar a las señoritas ***CERÓN ARMAS, NATALIA PAOLA Y CHILLAN CUSI, MIREYA ESTEFANÍA*** para que lo sustente públicamente.

Sangolquí, 15 de noviembre del 2015

ING. MAURICIO EDUARDO CAMPAÑA ORTEGA

DIRECTOR

AUTORÍA

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORÍA DE RESPONSABILIDAD

Nosotras, ***CERÓN ARMAS, NATALIA PAOLA***, con cédula de identidad N° 1722754692, y ***CHILLAN CUSI, MIREYA ESTEFANÍA***, con cédula de identidad N° 1720628872, declaramos que este trabajo de titulación "***DESARROLLO DE UN WEB CRAWLER PARA EL SISTEMA DE RECOMENDACIÓN ESCOLÁSTICO AQUINO BASADO EN PERFILES PARA LOS POSTULANTES A BECAS DEL SENESCYT***" ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaramos que este trabajo es de nuestra autoría, en virtud de ello nos declaramos responsables del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 15 de noviembre del 2015

NATALIA PAOLA CERÓN ARMAS
C.C. 1722754692

MIREYA ESTEFANÍA CHILLAN CUSI
C.C. 1720628872

AUTORIZACIÓN

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Nosotras, ***CERÓN ARMAS, NATALIA PAOLA*** y ***CHILLAN CUSI, MIREYA ESTEFANÍA***, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación "***DESARROLLO DE UN WEB CRAWLER PARA EL SISTEMA DE RECOMENDACIÓN ESCOLÁSTICO AQUINO BASADO EN PERFILES PARA LOS POSTULANTES A BECAS DEL SENESCYT***" cuyo contenido, ideas y criterios son de nuestra autoría y responsabilidad.

Sangolquí, 15 de noviembre del 2015

NATALIA PAOLA CERÓN ARMAS
C.C. 1722754692

MIREYA ESTEFANÍA CHILLAN CUSI
C.C. 1720628872

DEDICATORIA

"Nuestra recompensa se encuentra en el esfuerzo y no en el resultado, un esfuerzo total es una victoria completa" Mahatma Gandhi

Al creador de todas las cosas, el que me ha dado fortaleza para continuar cuando he estado a punto de caer, por ello, con todo el amor de mi corazón dedico primeramente mi trabajo a Dios.

De igual forma, dedico este trabajo a mis padres, Jorge y Laura por su infinito amor, trabajo y sacrificios en estos años, gracias a ustedes he logrado llegar hasta aquí y convertirme en lo que soy, siempre será un privilegio ser su hija porque son los mejores padres. A mi hermana Kimberly que siempre ha estado junto a mí, brindándome su apoyo incondicional e impulsándome para ser el mejor ejemplo. Infinitamente gracias a todos por estar presente no solo en esta etapa tan importante de mi vida, sino en todo momento ofreciéndome lo mejor y buscando lo mejor para mi persona.

Natalia Cerón Armas

"La diferencia entre una persona exitosa y los demás no es la falta de fuerza ni de conocimiento, sino la falta de voluntad" Vince Lombadi

A mis padres Norma y Modesto que sin ellos nunca hubiese podido hacer realidad este sueño y culminar la tesis con éxito. A mis hermanos Wilson y Miriam quienes me han apoyado y cuidado para poder llegar a esta instancia de mis estudios. A mis sobrinos Lady, Javier, y Naomi por ser la inspiración para superarme cada día y les sirva como ejemplo para su buen futuro. Sobre todo gracias a mi pequeño ángel que desde arriba me cuida y está dentro de mi corazón su recuerdo.

Mireya Chillán Cusi

AGRADECIMIENTO

"El agradecimiento es la memoria del corazón"

Agradezco a Dios por ser mi guía y fortaleza. A mis padres y hermana por apoyarme en cada decisión y estar junto a mí en esas noches de desvelo durante estos cinco años de carrera. A mis ingenieros que cada uno aporto un granito de arena para mi aprendizaje, no solo profesionalmente sino compartiendo lecciones de vida y forjando una gran amistad. A mi mejor amigo y novio, Stalin por impulsarme a terminar este trabajo y apoyarme incondicionalmente desde que empezamos el largo viaje de la universidad.

A mis compañeros de clase y amigos, con los que he compartido grandes momentos, en especial a mis compañeros de tesis Mireya y Ramiro con quienes en estos últimos meses nos hemos esforzado para cumplir este sueño que es el primero de muchos que vendrán, éxitos mis colegas. Agradezco a todos mis familiares, mis amigos de la academia y en general a todos aquellos que siguen estando cerca de mí y que le regalan a mi vida algo de ellos, lo valoro mucho.

Natalia Cerón Armas

"Nunca es demasiado el agradecimiento, a quien no te abandono en tus peores momentos"

A Dios y a todas las personas que colaboraron en este proyecto de tesis, a todos los profesores que supieron transmitir sus conocimientos durante los cinco años de carrera. A mis padres y hermanos, por su amor, trabajo, sacrificio y por ser el mejor ejemplo de constancia, fortaleza y responsabilidad. A todos mis amigos, de manera muy especial a mis compañeros Ramiro y Natalia con quienes trabajamos arduamente para que este proyecto de tesis culmine satisfactoriamente. A mis amigos que desde la niñez me brindaron su sincera amistad con altas y bajas permanecemos juntos.

Mireya Chillan Cusi

AGRADECIMIENTO ESPECIAL

"La disciplina es el puente que une las metas con los logros"

Agradecemos al PhD. Filipe Mota Prometeo del Departamento de Ciencias de la Computación de la Universidad de las Fuerzas Armadas – ESPE y Asesor Técnico de la SENESCYT por la confianza brindada a nosotros y a su vez por la colaboración siendo el vínculo con La Secretaria de Educación Superior, Ciencia, Tecnología e Innovación quien nos otorgó el auspicio para el desarrollo de este proyecto de investigación.

Al Msc. Mauricio Campaña por aceptar este reto siendo nuestro Director de Proyecto y habernos guiado durante el proceso del desarrollo escrito del mismo; además de acompañarnos durante nuestro proceso formativo en la Carrera de Ingeniería de Sistemas e Informática

Al Ing. Henry Coral por su constante apoyo y guía para el desarrollo práctico del proyecto, compartiendo su experiencia y conocimiento no solo profesionales sino también lecciones de vida y forjando una gran amistad.

Ramiro Andrade

Natalia Cerón

Mireya Chillán

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

RESUMEN

La Secretaria Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT), concede becas a los estudiantes universitarios que desean estudiar en el extranjero. Para la aplicación de la mencionada beca, los estudiantes deben buscar por si solos información de las universidades con las cuales el estado ecuatoriano tiene convenio, pero al realizar esta búsqueda se encuentran con la gran pregunta ¿Qué voy a estudiar?, ¿Cuál es la carrera que debo elegir?, esta ha sido la razón para implementar un sistema que ayude a la selección, búsqueda de información y recomendación de carreras para los postulantes de becas del Senescyt. El presente artículo detalla la implementación de un sistema de Búsqueda Documental genérico, el cual puede ejecutar de forma independiente búsquedas de texto de páginas web y archivos PDF de los cuales su contenido se encuentra almacenado en una base de datos documental. El producto final desarrollado y que es la parte central de este proyecto de investigación es el desarrollo de un web crawler conjuntamente con un sistema de búsqueda e indexación de documentos que integran el uso de las librerías Apache Tika y Apache Lucene, dando como resultados las funcionalidades de recolección de información, extracción de contenido en texto plano, indexación de documentos y búsqueda para facilitar la integración con el sistema web AQUINO.

PALABRAS CLAVE:
WEB CRAWLER
BÚSQUEDA
TIKA
LUCENE
DOCUMENTAL
INDEXACIÓN
REPOSITORIOS
BASE DE DATOS NOSQL.

ABSTRACT

The Ministry of Higher Education, Science, Technology and Innovation (SENESCYT) award scholarships to college students who wish to study abroad. For the application of that scholarship , students must find their own information of the universities with which the Ecuadorian state has an agreement , but to perform this search are the big question What will I study? , What is the career I choose?, this was the reason for implementing a system to help the selection, information search and recommendation racing for applicants Senescyt scholarship. This article details the implementation of a generic document search system, which can run independently text searches of web pages and PDF files which content is stored in a document database. The final product developed and the central part of this research project is the development of a web crawler conjunction with a search system and indexing of documents that integrate the use of libraries Apache Lucene and Apache Tika, giving as a result the data collection capabilities, content extraction in plain text, indexing and searching documents to facilitate integration with web system AQUINO.

KEYWORDS:
WEB CRAWLER
SEARCH
TIKA
LUCENE
DOCUMENTARY
INDEXING
REPOSITORIES
NoSQL DATABASE.

CAPÍTULO 1
INTRODUCCIÓN

1.1 ANTECEDENTES

La formación de los estudiantes universitarios en el Ecuador se fortalece gracias a la Ley Orgánica de Educación Superior según el Art.4 que plantea: “El derecho a la educación superior consiste en el ejercicio efectivo de igualdad de oportunidades, en función de los méritos respectivos, a fin de acceder a una formación académica y profesional con producción de conocimiento pertinente y de excelencia”.

La Secretaria Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT), en la actualidad tiene más de ocho mil becados alrededor del mundo y ha dado más de diez mil becas alrededor del mundo desde el 2010.

Para los becarios del Ecuador la SENESCYT cuenta con un sistema de atribución de becas basado en procesos administrativos, los cuales no poseen un sistema informático que ayude a los postulantes a la selección de la carrera o programa, debido a la escasez de información que posee la SENESCYT acerca de las universidades del mundo. Las posibilidades generadas con respecto a los programas de becas no satisfacen completamente las expectativas de los postulantes, por tal motivo al regresar los becarios no se encuentran motivados y satisfechos con la elección que realizaron, generando así un bajo nivel de productividad científica para el desarrollo del país.

1.2 PROBLEMÁTICA

La Educación Superior es un derecho de los ecuatorianos, sin importar el nivel social-económico, cualquier persona tiene la oportunidad de cursar sus estudios dentro o fuera del país; el problema radica en la selección de la carrera en la que realizarán sus estudios de pregrado o postgrado, la falta de información o desconocimiento y poco desarrollo de las habilidades de cada uno de los estudiantes.

Se plantea desarrollar un sistema de recomendación escolástico basado en los datos que se pueden recoger de la URL de cada universidad y en perfiles de los estudiantes para orientarlos a elegir la carrera más idónea, de esta manera se optimiza la inversión de los recursos en la Educación Superior del País.

1.3 JUSTIFICACIÓN

Senescyt como institución gestora de fondos para atribución de becas ha publicado un listado de Universidades a los que los ecuatorianos podrían aplicar (universidades reconocidas) con el fin de acceder a una beca, el mismo que solo contiene la dirección web de cada una de estas universidades como información adicional. Actualmente, el postulante o posible becario, debe ingresar a los sitios web de las universidades con el fin de obtener información acerca del programa de estudios en el que está interesado, pero tiene la desventaja de no poder acceder a dicha información de forma centralizada, es decir, que no puede determinar exactamente los programas de carreras según su perfil en el listado de las universidades que tienen convenio con la Senescyt.

El presente proyecto consiste en proveer a los postulantes Ecuatorianos de una plataforma que les ayude a escoger la universidad y la carrera a la que podrían aplicar, recomendando un grupo de opciones pertinentes basada en los requerimientos individuales del mismo. Este proyecto en forma global contempla las siguientes fases de desarrollo: base de datos de postulantes, base de datos de oferta académica mundial y sistema de recomendación accesible mediante internet.

1.4 OBJETIVOS

1.4.1 Objetivo General

Diseñar e implementar una base de datos que contenga la información correspondiente a los programas de Ciencias de la Computación de las universidades que se encuentran en el listado de la Senescyt mediante el uso de web crawlers.

1.4.2 Objetivos Específicos

- Estudiar y aplicar algoritmos de búsqueda que permitirán la implementación del web crawler.
- Implementar un buscador (web crawler) que obtenga información de páginas web pertenecientes a las universidades que se encuentran en el listado de la Senescyt.
- Implementar un repositorio de datos que sirva como base para el motor de búsqueda y recomendación.

1.5 ALCANCE

Se desarrollará un prototipo dentro de la comunidad de estudiantes de la Facultad de Ciencias de la Computación de la Universidad de las Fuerzas Armadas–ESPE y el alcance inicial contempla búsquedas de datos solamente con respecto a programas de Ciencias de la Computación en el listado del Senescyt, en idioma Español.

Figura 1. Estructura General del Sistema AQUINO – Web Crawler

Las tareas a desarrollarse se relacionan con la obtención de información sobre las universidades registradas por el Senescyt, determinación de perfiles de estudiantes y registro en bases de datos para posteriormente enfocarse en el web crawler y que permitan encontrar perfiles de líneas de investigación en el listado de programas de maestría del Senescyt, tal como lo ilustra la Figura 1.

CAPÍTULO 2
MARCO TEÓRICO

2.1 BECAS SENESCYT

Las becas SENESCYT son financiamientos que otorga el estado por medio de las Secretaria Nacional de Educación Superior, Ciencia y Tecnología a los estudiantes ecuatorianos para realizar sus estudios a nivel Pregrado o Postgrado en las universidades del Ecuador o a su vez en universidades extranjeras que tengan convenio con el país.

2.1.1 Requisitos

Los estudiantes que deseen participar en las convocatorias para las becas del Senescyt deben cumplir los requisitos:

Tabla 1

Requisitos formales y documentales de respaldo para la postulación

Nº	Requisitos Formales para la Postulación	Documentación de Respaldo
REQUISITOS GENERALES		
1.	Ser ciudadano/a ecuatoriano/a	a. Fotocopia a color o archivo digital de la cédula de ciudadanía, o del pasaporte; y, certificado de votación vigentes.
2.	Al momento del cierre del período de postulación de la convocatoria: **Para el componente general:** Tener hasta 35 años de edad cumplidos, para programas de maestría; y, hasta 45 años de edad cumplidos, al momento de la postulación, para aplicar a programas de doctorados. Para especialidades médicas no existe límite de edad. **Para el resto de componentes:** Tener hasta 55 años de edad cumplidos, al momento de la postulación, para programas de maestría y doctorados. Para especialidades médicas no existe límite de edad.	

Continúa

3.	Contar con título profesional o grado académico habilitante al momento de la postulación, este título debe constituir el requisito previo al programa de estudios aplicado de igual manera debe estar debidamente registrado en el SNIESE de la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación.	a. Estos documentos se verificarán en el sistema de la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación. b. En caso de títulos nacionales no registrados adjuntar certificación de la Universidad en la cual se exprese que el mismo está en trámite de registro. c. En caso de títulos extranjeros sin registro deberán adjuntar copia del título o certificación emitida por la IES extranjera notariada o apostillada. Se aceptarán sólo aquellos que provengan de universidades que se hallen en el listado de reconocimiento automático de títulos.
4.	Encontrarse en proceso de aplicación a un programa de estudios de maestría, doctorado; o, especialidad médica en una universidad o centros de estudios en el extranjero.	a. Carta de aceptación, pre-aceptación u otro documento que pruebe el inicio del contacto o aplicación a un programa de estudios en universidades o centros de estudios en el extranjero, en el cual se demuestre que es elegible para cursar el programa. b. Información específica del programa de estudios a llevar a cabo, en el extranjero, que incluya modalidad, número de créditos, duración del programa de estudios y título que otorga el programa de estudios.
5.	Hoja de vida	a. Presentar Hoja de Vida, según formato
REQUISITOS ESPECÍFICOS DE CONFORMIDAD AL COMPONENTE APLICADO		
Para quienes pertenecen a pueblos o nacionalidades		
1	Pertenecer a un pueblo o nacionalidad ecuatoriana	a. Certificado de pertenecer a un pueblo o nacionalidad ecuatoriana emitido por las organizaciones de base de Pueblos y Nacionalidades legalmente constituidas o avalado por la CODENPE, CODEPMOC o CODAE, según corresponda
Para personas con discapacidad		
1	En caso de tener capacidades especiales calificadas por la entidad competente.	a. Fotocopia a color o archivo digital del documento oficial expedido por la entidad competente.
Para convulsiones sociales, políticas, económicas; desastres naturales o antropogénicos:		

Continúa

1	Convulsiones sociales, económicas, políticas; desastres naturales y antropogénicos.	a. Certificados o documentos que demuestren la situación especial o de vulnerabilidad por la que se encuentre atravesando el/la solicitante o su familia en primer grado de consanguineidad y cónyuge, tales como: ✓ Informes institucionales, policiales o judiciales ✓ Declaratorias de zonas de emergencia. ✓ Documentos emitidos por instituciones públicas respecto a convulsiones sociales, políticas, económicas y desastres naturales.

REQUISITOS ADICIONALES SOLO PARA EL COMPONENTE GENERAL[1]		
Nro.	**Requisito**	**Documento de respaldo**
1.	Información socio-económica	1. Ficha socio-económica a. Mecanizado del IESS de miembros que aportan al grupo familiar. (Este requisito será opcional para los/as postulantes que residan en el extranjero). b. Certificado de no aportar al IESS de miembros del grupo familiar menores de 25 años. (Este requisito será opcional para los/as que residan en el extranjero). c. Planilla de un servicio básico actualizada donde conste la dirección de la residencia de el/a solicitante. (Este requisito será opcional para los/as que residan en el extranjero). d. En caso de vivienda arrendada copia del contrato de arrendamiento o factura/comprobante de pago donde se detalle el canon de arrendamiento. . (Este requisito será opcional para los/as postulantes)

Fuente: (SENESCYT, 2014)

Para postular a una beca dentro de este programa, los interesados/as deberán cumplir con las siguientes condiciones:

a. No podrán mantener obligaciones vencidas o glosas con instituciones del sector público, tanto como deudor y/o garante.

[1] Reforma realizada mediante Acta Nro. 010-2014 aprobada por el Comité Institucional de Becas y Ayudas Económicas el 27 de mayo de 2014.

b. No podrán ser contratistas incumplidos/as o adjudicatarios/as fallido del Estado.
c. No podrán adeudar más de dos pensiones alimenticias, lo cual deberán declarar dentro del sistema de postulación en línea del programa.
d. No podrán percibir beca o ayuda económica otorgada por alguna institución pública o privada ecuatoriana, o de la cooperación internacional receptada por el Estado Ecuatoriano; para el mismo fin o para el mismo concepto.

(SENESCYT, 2011)

2.1.2 Programas

La Senescyt como política de Estado ha creado desde el año 2013 tres programas de otorgación de becas:

1) **Becas en el exterior**
   a) "UNIVERSIDADES DE EXCELENCIA 2015": programa para estudiar nivel técnico – tecnológico superior, tercer o cuarto nivel en los mejores centros de educación superior del mundo.
   b) "CONVOCATORIA ABIERTA 2015 – PRIMERA FASE": estudios de cuarto nivel en universidades y centros de educación de excelencia académica en el extranjero.
   c) "Docentes de Universidades y Escuelas Politécnicas": otorga becas para estudios de Doctorado (PHD) en universidades y centros de estudio en el extranjero.
   d) "Becas Investigadores": financia estudios de cuarto nivel en universidades y centros de estudio en el extranjero a personas que sean investigadores de las Instituciones Públicas de País y aquellos que deseen formar parte de los proyectos de investigación de las Universidades y Escuelas Politécnicas. (SENESCYT)
2) **Becas nacionales**
   a) "Programa Nacional de Becas": proporciona financiamiento de carreras en nivel técnico – tecnológico superior, en Instituciones de Educación Superior del país (IES). (SENESCYT)

3) Otros programas

a) GAR Cuarto Nivel: grupo de excelencia de alto rendimiento.

b) Ayudas Económicas: apoyo económico con fines educativos a estudiantes, docentes, investigadores, y profesionales para que puedan cubrir rubros inherentes a su formación superior.

c) Becas de Reforzamiento: dirigido a adjudicatarios y adjudicatarias de una beca SENESCYT residentes en el país; su principal objetivo es fortalecer las habilidades del idioma, especialmente inglés, y las destrezas necesarias para afrontar los retos académicos que implica un programa de estudios de cuarto nivel en el exterior

d) Becas para Posdoctorado: becas de formación profesional de Cuarto Nivel para Pueblos y Nacionalidades, que posteriormente puedan participar activamente en la solución de las problemáticas sociales, culturales, económicas y políticas que enfrentan los distintos Pueblos y Nacionalidades del Ecuador.

e) Becas Vamos Yachay: financiar la capacitación intensiva en idioma inglés, orientado a la prestación de servicios turísticos y de soporte, que se realizará en el extranjero; dirigido a las personas que residen en el área de influencia del proyecto “Ciudad del Conocimiento YACHAY”

f) Becas para Fortalecimiento del Talento Humano en Salud: potenciar los mecanismos necesarios para la incorporación de personal altamente capacitado, que responda a la necesidad del sector salud con la implementación de un programa de becas que permita la formación de talento humano altamente calificado para cubrir con las necesidades del primer, segundo y tercer nivel de atención primaria en salud, del Sistema Nacional de Salud.

g) Becas SENESCYT – Embajada de Francia 2013: conceder becas de maestría para la formación de profesionales ecuatorianos, en los establecimientos de enseñanzas superiores públicas o privadas reconocidos por el Estado francés.

h) Becas del Plan de Contingencia: garantizar la continuidad de estudios de las y los estudiantes de las catorce universidades y escuelas politécnicas

suspendidas en forma definitiva en el mes de abril del 2012 por el Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de Educación Superior, el Consejo de Educación Superior y la SENESCYT.

2.1.3 Áreas de Estudio

La SENESCYT cuenta con diferentes áreas de estudio para la otorgación de becas tanto a nivel nacional e internacional así como de Pregrado y Postgrado.

Tabla 2

Descripción de las áreas de estudio

Área	Campo Amplio	Campo Detallado	Descripción de formación por área
1	Educación	• Ciencias de la educación. • Formación para docentes para todos los niveles de educación • Formación para docentes sin asignatura de especialización. • Formación para docentes con asignatura de especialización. • Gestión educativa	• Docencia para: educación preescolar, escolar, educación media y superior, con o sin especialización. • Educación de adultos • Desarrollo educativo • Diseño y evaluación de modelos educativos • Diseño y gestión de proyectos socio educativos • Desarrollo curricular • Modelos y sistemas de educación • Educación especial • Educación intercultural • Educación y desarrollo del pensamiento • Educación y desarrollo social • Evaluación de conocimientos, pruebas y mediciones • Gerencia educativa • Gestión académica • Informática educativa • Investigación educativa • Orientación educativa vocacional y profesional • Pensamiento estratégico y prospectiva para la educación • Planificación, evaluación y acreditación de la educación • Psicología educativa • Tecnologías de la información y de la comunicación aplicadas a la educación • Tratamiento de dificultades de aprendizaje
2	Artes	• Bellas artes • Artesanías • Música y artes escénicas. • Técnicas audiovisuales y producción para medios de comunicación.	• Dibujo y pintura • Escultura • Música • Arte dramático • Danza • Artes gráficas y audiovisuales • Fotografía • Cinematografía y producción cinematográfica • Producción musical • Producción de radio y televisión • Actuación • Anatomía artística • Ciencias de la conservación, restauración y patrimonio • Producción de arte contemporáneo • Crítica, curaduría y teorías de arte • Grabado • Investigación artística • Museología y museografía • Teoría e historia del arte

Continúa

3	**Ciencias naturales, matemáticas y estadística**	• Biología. • Bioquímica. • Ciencias del medio ambiente. • Medio ambientes naturales y vida silvestre. • Química. • Ciencias de la tierra. • Física. • Matemáticas. • Estadística.	• Biología • Botánica • Bacteriología • Toxicología • Microbiología • Zoología • Entomología • Ornitología • Genética • Bioquímica • Biofísica. • Astronomía y ciencias espaciales • Física y afines • Química y afines • Geología • Geofísica • Mineralogía, • Antropología Física • Geografía y afines • Meteorología y demás ciencias de la atmósfera • Ciencias Marinas • Vulcanología • Paleoecología • Matemáticas • Ciencias actuariales • Estadística • Auditoría ambiental • Biotecnología • Ciencias ambientales • Ciencias del manejo y conservación de los recursos naturales • Ecología • Econometría
4	**Tecnologías de la información y la comunicación (TIC´S)**	• Uso de computadores. • Diseño y administración de redes y bases de datos. • Desarrollo y análisis de software y aplicaciones	• Informática • Programación • Procesamiento de datos • Sistemas operativos. • Conectividad y redes telecomunicaciones • Desarrollo y análisis de software y aplicaciones • Diseño de redes y bases de datos • Ingeniería de e-learning • Ingeniería de sistemas • Seguridad informática aplicada • Análisis de sistemas computacionales • Telemática • Inteligencia artificial
5	**Ingeniería, industria y construcción**	• Ingeniería y procesos químicos. • Tecnología de protección del medio ambiente. Electricidad y energía. • Electrónica y automatización. • Mecánica y profesiones afines a la metalistería Vehículos, barcos y aeronaves motorizadas. • Procesamiento de alimentos. • Materiales (vidrio, papel, plástico y madera). Productos textiles (ropa, calzado y artículos de cuero). • Minería y extracción. • Arquitectura y construcción. • Construcción e ingeniería civil	• Mecánica • Electricidad • Electrónica • Automatización • Telecomunicaciones • Topografía • Procesamiento de alimentos y bebidas • Minería e industrias extractivas. • Arquitectura y urbanismo • Arquitectura Estructural • Arquitectura Paisajística • Planificación Comunitaria • Ingeniería Civil. • Industria y producción • Cadenas productivas agroindustriales • Diseño arquitectónico • Diseño mecánico • Diseño industrial • Ingeniería ambiental • Ingeniería automática y de control • Ingeniería biónica • Ingeniería cibernética

Continúa

			• Ingeniería de minas • Ingeniería del petróleo • Ingeniería eléctrica • Ingeniería electromecánica • Ingeniería electrónica • Ingeniería hidráulica • Ingeniería mecatrónica • Ingeniería vial • Ingeniería de barcos y aeronaves motorizadas • Ingeniería automotriz • Ingeniería industrial • Ingeniería y procesos químicos • Materiales (vidrio, papel, plástico y madera) • Productos textiles (ropa, calzado y artículos de cuero) • Tecnología de protección del medio ambiente • Transportes • Ingeniería de energías renovables.
6	**Agricultura, silvicultura, pesca y veterinaria**	• Producción agrícola y ganadera. • Horticultura • Silvicultura • Pesca • Veterinaria	• Agricultura • Producción agropecuaria • Agronomía • Ganadería • Horticultura y jardinería • Silvicultura y técnicas forestales, • Parques naturales • Flora y fauna • Ciencia y tecnología pesqueras. • Veterinaria. • Acuacultura • Agroecología • Agricultura tropical • Agroindustria • Avicultura • Etología • Taxonomía
7	**Salud y bienestar**	• Odontología. • Medicina. • Enfermería y partería. • Tecnología de diagnóstico y tratamiento médico. Terapia y rehabilitación Farmacia. • Medicina y terapia tradicional y complementaria Bienestar. • Asistencia a adultos mayores y discapacitados Asistencia a la infancia y servicios para jóvenes. Trabajo social y orientación. • Arte terapia	• Medicina: alergología, anatomía, epidemiología, citología, fisiología, inmunología e inmunohematología, patología, anestesiología, pediatría, obstetricia y ginecología, medicina interna, cirugía, neurología, psiquiatría, radiología, oftalmología, angiología y cirugía vascular, aparato digestivo o gastroenterología, bioquímica clínica, cardiología, traumatología, endocrinología, epidemiología, y otras especialidades médicas. • Farmacia y farmacología • Terapéutica, rehabilitación, prótesis • Nutrición • Enfermería • Odontología. • Asistencia a personas con capacidades especiales • Asistencia a la infancia • Servicios de gerontología; • Orientación y asistencia social. • Administración de instituciones de salud • Deficiencia mental y trastornos del aprendizaje • Salud pública • Citología

Fuente: (SENESCYT, 2014)

Es importante acotar, que cuando se requiera una beca de estudios, tanto a nivel nacional como en el exterior; el SENESCYT no financiará programas de estudio en las áreas de:

- Administración de empresas
- Ciencias Sociales
- Negocios y afines
- Marketing y afines
- Mercadotecnia
- Diseño de interiores y afines
- Diseño de modas y afines
- Recursos humanos
- Especialidades médicas relacionadas con la estética (Becas SENESCYT)

2.2 MÉTODOS DE BÚSQUEDA

Un algoritmo de búsqueda es aquel que está diseñado para localizar un elemento con ciertas propiedades dentro de una estructura de datos. Consiste en solucionar un problema de existencia o no, de un elemento determinado en un conjunto finito de elementos. Un algoritmo de búsqueda tiene como parámetros la profundidad y anchura. (Wikipedia) (Viñals, pág. 26)

Existen diferentes métodos de búsqueda, todos parten de un esquema iterativo, en el cual se trata una parte o la totalidad de la estructura de datos sobre la que se busca. El tipo de estructura condiciona el proceso de búsqueda debido a los diferentes modos de acceso a los datos. (INACAP)

2.2.1 Búsqueda primero en anchura (BFS)

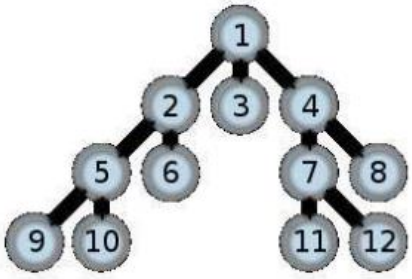

Figura 2. Orden en el que se despliegan los nodos (BFS)

Fuente: (Viñals, pág. 27)

BFS es una estrategia de búsqueda en grafos donde la búsqueda está limitada a dos operaciones: visitar e inspeccionar un nodo del grafo y obtener acceso para visitar los nodos vecinos del nodo siendo visitado. La búsqueda primero en anchura comienza en un nodo inicial, o raíz, y desde ahí inspecciona todos sus vecinos; después, para cada uno de esos vecinos, inspecciona cada uno de sus correspondientes vecinos. (Viñals, pág. 27)

BFS es un algoritmo de búsqueda no informada que intenta expandir todos los nodos de un grafo, buscando sistemáticamente cada una de las posibles soluciones. Estos algoritmos pueden ser utilizados para resolver múltiples problemas en la teoría de grafos, pero el más importante en el campo de la robótica es que es capaz de encontrar siempre la distancia más corta entre dos nodos dados. Es decir, BFS siempre devuelve el camino óptimo, aunque no puede garantizarse que encuentre la solución en un tiempo finito. (Viñals, pág. 27)

2.2.2 Búsqueda primero en profundidad (DFS)

DFS es un algoritmo de búsqueda no informada que progresa al expandir el primer “hijo” del nodo actual y repitiendo recursivamente esta operación hasta encontrar una solución o un nodo “hoja”, es decir, un nodo sin ”hijos”. En este caso, retrocederá, también recursivamente, hasta encontrar el primer nodo anterior con “hijos” que expandir. En el ámbito de la robótica, los algoritmos de búsqueda en profundidad, así como sus variantes informadas, tienen multitud de aplicaciones

como la búsqueda de nodos conectados (que nos permite determinar si un determinado problema tiene solución) y, fundamentalmente, encontrando el primer camino que une dos nodos. (Viñals, pág. 28)

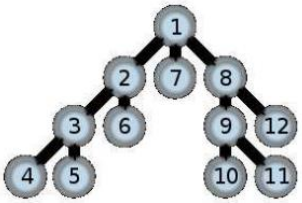

Figura 3. Orden en el que se despliegan los nodos (DFS)

Fuente: (Viñals, pág. 28)

2.2.3 Búsqueda informada. Algoritmos A*

Si un algoritmo de búsqueda tiene un cierto conocimiento a priori del problema a resolver, y es capaz de usar esta información para otorgar una cierta prioridad a los nodos que ha de expandir, decimos que se trata de un algoritmo de búsqueda informada (Algoritmo estrella). Un algoritmo A* es un algoritmo de búsqueda informado con una heurística optimista, esto nos garantiza que una búsqueda A* expandirá el menor número de nodos de todos los algoritmos de búsqueda y siempre encontrará una solución, aunque esta no tiene por qué ser necesariamente la solución óptima. (Viñals, pág. 28)

2.2.4 Algoritmo de Fuerza Bruta

Se alinea la primera posición del patrón con la primera posición del texto, y se comparan los caracteres uno a uno hasta que se acabe el patrón, esto es, se encontró una ocurrencia del patrón en el texto, o hasta que se encuentre una discrepancia. (INACAP, pág. 89)

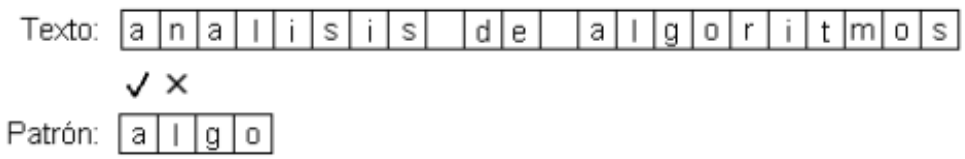

Figura 4. Algoritmo de Fuerza Bruta 1

Fuente: (INACAP)

Si se detiene la búsqueda por una discrepancia, se desliza el patrón en una posición hacia la derecha y se intenta calzar el patrón nuevamente. En el peor caso este algoritmo realiza O(m.n)comparaciones de caracteres. (INACAP, pág. 89)

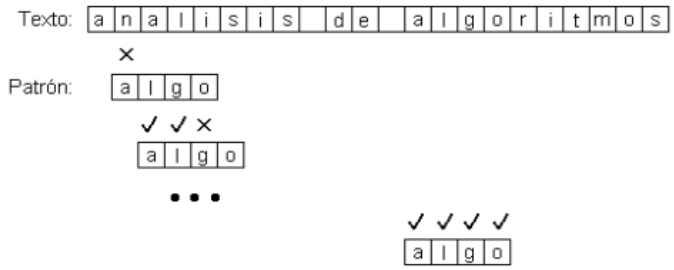

Figura 5. Algoritmo de Fuerza Bruta 2

Fuente: (INACAP)

2.2.5 Otros Métodos de Búsqueda

Existe un gran número de algoritmos de búsqueda, otros comúnmente utilizados son los siguientes:

- Búsqueda Lineal
- Búsqueda Binaria
- Árboles de Búsqueda
- Búsqueda por Transformación de Claves (Hashing)
- Búsqueda en Textos: Algoritmo de Fuerza Bruta, Algoritmo Knuth-MorrisPratt, Algoritmo de Boyer-Moore.

Se considera importante explicar más a fondo los algoritmos de búsqueda basados en textos, ya que estos nos serán de utilidad para la realización de este proyecto de investigación.

La búsqueda de patrones en un texto es un problema muy importante en la práctica. Sus aplicaciones en computación son variadas, como por ejemplo la búsqueda de una palabra en un archivo de texto, el cual requiere buscar patrones en donde ocurren alteraciones con cierta probabilidad, esto es, la búsqueda no es exacta. (INACAP, pág. 88)

2.3 WEB CRAWLER

Un web crawler es un software o script programado que explora la World Wide Web (WWW) en forma sistemática y automatizada. La estructura de la WWW es una estructura gráfica, es decir, los enlaces de una página web pueden ser utilizados para abrir otras páginas web, por lo tanto la operación de búsqueda de un web crawler puede resumirse como un proceso dirigido para atravesar varias páginas, recuperarlas y guardarlas en un repositorio local (Trupti V. , Ravindra D., & Rajesh C., 2014). En general, un web crawler comienza visitando una lista específica de urls, e identifica los hiperenlaces en dichas páginas y los añade a la lista de urls visitadas de manera recurrente de acuerdo a determinado conjunto de reglas; el proceso es repetitivo hasta encontrar la información solicitada. (Del Coso Santos, 2009)

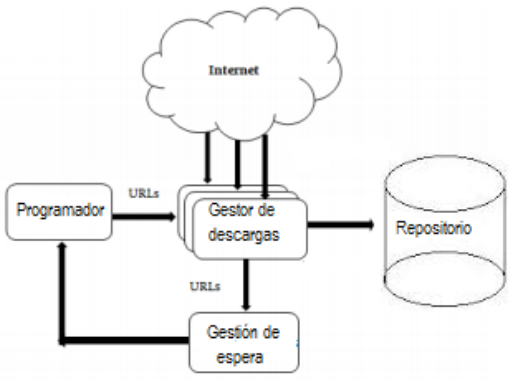

Figura 6. Arquitectura básica de un web crawler

Fuente: (Trupti V. , Ravindra D., & Rajesh C., 2014)

2.3.1 Funcionamiento de un Web Crawler

Un web crawler en un inicio dispone de un conjunto inicial de urls, conocidas como semillas. Entonces el web crawler va descargando las páginas web asociadas o enlazadas a las semillas y buscando dentro de éstas otras urls. Cada nueva URL encontrada se añade a la lista de urls que el crawler debe visitar. A este proceso se le denomina recolección de urls. (Del Coso Santos, 2009)

De cada página descargada se obtiene su contenido (paginas HTML) y se crea un documento con sus metadatos y los almacena en un repositorio. A su vez las nuevas urls encontradas son enviadas al gestor de espera para su procesamiento posterior. Posteriormente se encuentra el modulo llamado "Programador", que se encarga realizar un nuevo proceso con las urls que se encuentra en cola, llamado barrido de segundo nivel. (Camargo Sarmiento, 2013)

Cuando el crawler parsea[2] una página web, procesa la información y decide que partes de esta son de utilidad. Es decir aplica algún tipo de algoritmos para conseguir el objetivo establecido.

2.3.2 Políticas de búsqueda

El comportamiento de un web crawler es el resultado de una combinación de políticas:

- **Política de selección**

  Es un hecho que el tamaño actual de la Web es inmensamente grande, por lo que los motores de búsqueda solo pueden abarcar una pequeña porción de Internet. Por lo tanto la política de selección indica que páginas van a ser descargadas.

- **Política de revisitado**

  Es probable que en los sitios ya visitados se produzcan modificaciones como pueden ser actualización, inserción o borrado de contenidos, la política de revisitado establece cuando el crawler comprobara dichos cambios de acuerdo al parámetro frescura y edad de una página web.

[2] Parsea: Proceso de analizar una secuencia de símbolos a fin de determinar su estructura gramatical con respecto a una gramática formal dada. Formalmente es llamado análisis de sintaxis.

- **Política de cortesía**

Los web crawlers requieren un gran ancho de banda agotando así los recursos de la red, para evitar lo mencionado la política de cortesía aplica el protocolo de exclusión de robots que es un estándar para que los administradores de un sitio indiquen a los robots a que partes del servidor pueden acceder.

- **Política de paralelización**

Un web crawler múltiple es un robot que ejecuta múltiples procesos en paralelo para maximizar la tasa de descarga y evitar la descarga de una misma página varias veces. Para evitar este último inconveniente se precisa de dos tipos de políticas:

1. Asignación dinámica: en la que existe un servidor central que se encarga de asignar las urls a visitar, lo que permite balancear la carga que tienen los robots.
2. Asignación estática: en la que hay una regla definida desde el comienzo de la ejecución que define como asignar las nuevas urls. (Del Coso Santos, 2009)

2.4 ALMACENAMIENTO DE DATOS EN MONGODB

MongoDB es un sistema de bases de datos no relacionales, multiplataforma y orientado a documentos, su nombre proviene del término inglés “humongous” que significa enorme. Está liberada bajo licencia de software libre, específicamente GNU AGPL 3.0. (Graterol)

MongoDB usa el formato BSON (JSON Compilado) para guardar los datos en documentos tipo JSON con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida. (Ávila, 2012)

El desarrollo de MongoDB empezó en octubre de 2007 por la compañía de software 10gen; este motor de bases de datos es uno de los más conocidos y usados pudiéndolo comparar en popularidad con MySQL en el caso de las bases de datos relacionales; (Ávila, 2012)

2.4.1 Características Principales de MongoDB

- **Alto Rendimiento**

MongoDB proporciona la persistencia de datos de alto rendimiento, en particular:

- Soporta modelos de datos incrustados reduciendo la actividad de I/O en el sistema de base de datos.
- Reduce la actividad de I/O en el sistema de base de datos.
- Los índices apoyan para realizar consultas más rápidas y pueden incluir claves en documentos o arreglos embebidos. (MongoDB)

- **Alta Disponibilidad**

MongoDB proporciona una alta disponibilidad facilitando la replicación, a través de:

- Conmutación automática.
- Redundancia de datos. Un conjunto de réplicas es un grupo de servidores MongoDB que mantienen el mismo conjunto de datos, proporcionando redundancia e incremento de la disponibilidad de datos. (MongoDB)

- **Escalamiento Automático**

MongoDB ofrece escalabilidad horizontal como parte de su funcionalidad:

- Distribución automática de datos a través de un conjunto de máquinas (clúster).
- Conjuntos de réplicas que proporcionan lecturas consistentes para implementaciones de alto rendimiento y baja latencia. (MongoDB)

2.4.2 Términos básicos entorno a MongoDB

2.4.2.1 JSON-JavaScript Object Notation

JSON es un formato de representación de objetos independiente del lenguaje, actualmente se usa JSON en gran cantidad de sistemas para intercambiar información por su simplicidad en comparación con XML.

Este formato soporta gran cantidad de tipos de datos, lo que lo hace atractivo para un uso generalizado, y cada vez más lenguajes de programación dan soporte a este formato.

```
mongodb => {
        _id : 1 ,
        nombre : "MongoDB" ,
        u r l : " h t t p : / /www. mongodb . o r g " ,
        t i p o : " Documental "
}
```

Figura 7. Ejemplo de documento en JSON

Fuente: (Graterol)

2.4.2.2 Documento

Un documento es un conjunto de datos estructurados (mas no con un esquema estricto), que contiene pares clave/valor, y se usa BSON (JSON Binario) como formato para almacenar los documentos. Un documento puede ser comparado con una fila o registro en una base de datos relacional. (Graterol)

2.4.2.3 Colección

Una colección es un conjunto de documentos, similar a una tabla en las bases de datos relacionales. (Graterol)

2.5 INDEXACIÓN

El proceso de indexación consiste en analizar y extraer de entre toda la información disponible, la verdaderamente relevante. Posteriormente, con esa información se crea el índice a partir del cual se realizarán las búsquedas. El índice es una estructura de datos que permite acceso rápido a la información, algo similar a lo que podría ser el índice de un libro. (LC)

Figura 8. Proceso de Indexación

2.5.1 Índices Invertidos

Muchos motores de búsqueda incorporan un índice invertido en la evaluación de una consulta de búsqueda para localizar rápidamente los documentos que contengan las palabras en una consulta y luego clasificar los documentos por fecha. Debido a que el índice invertido almacena una lista de los documentos que contienen cada

palabra, el motor de búsqueda puede utilizar el acceso directo para encontrar los documentos asociados a cada palabra de la consulta con el fin de recuperar los documentos que coinciden con rapidez. (Balarezo, 2014)

La siguiente tabla es una ilustración simplificada de un índice invertido:

Tabla 3

Índice Invertido

Palabra	Documentos
Los	Documento 1, Documento 2
Carros	Documento 3, Documento 1

Este índice sólo puede determinar si existe una palabra dentro de un documento en particular, ya que no almacena información con respecto a la frecuencia y la posición de la palabra, por lo que se considera que es un índice booleano. Este índice determina qué documentos coincide con una consulta pero no clasifica los documentos coincidentes. En algunos diseños el índice incluye información adicional, como la frecuencia de cada palabra en cada documento o las posiciones de una palabra en cada documento. (Balarezo, 2014)

La información de posición permite que el algoritmo de búsqueda identifique la palabra más próxima para apoyar la búsqueda de frases; la frecuencia se puede utilizar para ayudar en el ranking de la relevancia de los documentos a la consulta. Tales temas son el foco central de la investigación de la recuperación de información. (Balarezo, 2014)

El índice invertido es una matriz dispersa, ya que no todas las palabras están presentes en cada documento. Para reducir los requisitos de memoria de almacenamiento de ordenador, se almacena de manera diferente a partir de una matriz de dos dimensiones. El índice es similar a las matrices de documentos empleados por el análisis semántico latente. (Balarezo, 2014)

2.5.2 Fusión de Índices

El índice invertido se llena a través de una fusión o reconstrucción. La reconstrucción es similar a una fusión, pero elimina primero el contenido del índice invertido. La arquitectura puede ser diseñada para apoyar la indexación gradual, donde una combinación identifica el documento o los documentos que se agregan o se actualizan y luego analiza cada documento en palabras. Para mayor precisión técnica, una combinación fusiona documentos recientemente indexados, por lo general reside en la memoria virtual, con el cache de índice que reside en los discos duros de una o más computadoras. (Balarezo, 2014)

Después del análisis, el indexador agrega el documento referenciado a la lista de documentos para las palabras adecuadas. En un motor de búsqueda más grande, el proceso de búsqueda de cada palabra en el índice invertido (con el fin de informar de que se produjo dentro de un documento) puede ser demasiado lento, por lo este proceso se suele dividir en dos partes, el desarrollo de un índice hacia adelante y un proceso que ordena el contenido del índice hacia adelante en el índice invertido. El índice invertido se llama así porque es una inversión del índice hacia adelante. (Balarezo, 2014)

2.5.3 Índice hacia adelante

El índice hacia adelante almacena una lista de palabras para cada documento. La siguiente tabla es una forma simplificada del índice hacia adelante:

Tabla 4

Índice hacia adelante

Documento	Palabras
Documento 1	Los, perros, ladran
Documento 2	El, gato, con, botas
Documento 3	Los, planetas

El fundamento de la elaboración de un índice hacia adelante es que a medida que los documentos se están analizando, es mejor almacenar de inmediato las palabras por documento. La delimitación permite el procesamiento del sistema asíncrono, lo que evita en parte el cuello de botella de actualización del índice invertido. El índice de avance está ordenado para transformarlo a un índice invertido. El índice hacia adelante es esencialmente una lista de pares formados por un documento y una palabra, recopilado por el documento. Convertir el índice de interés con un índice invertido es sólo una cuestión de la clasificación de los pares de las palabras. En este sentido, el índice invertido es un índice de avance de palabras ordenados. (Balarezo, 2014)

2.6 LUCENE

Lucene es una herramienta que permite tanto la indexación como la búsqueda de documentos. Creada bajo la metodología orientada a objetos e implementada completamente en Java, no se trata de una aplicación que se descarga, instala y ejecuta sino de una API flexible, a través de la cual se añaden, con esfuerzos de programación, capacidades de indexación y búsqueda a cualquier sistema que se esté desarrollando. (Ramos Hernández & Alor Hernández, 2015)

Existen otras herramientas, aparte de Lucene, que permiten realizar la indización y búsqueda de documentos pero dichas herramientas se utilizan para usos concretos, lo que implica que el intentar adaptarlas a un proyecto específico sea una tarea realmente difícil. La idea que engloba Lucene es completamente diferente, ya que su principal ventaja es su flexibilidad, permite su utilización en cualquier sistema que lleve a cabo procesos de indización o búsqueda. (Ramos Hernández & Alor Hernández, 2015)

2.6.1 Características de Lucene

A continuación se detallan algunas características que hacen de Lucene una herramienta flexible y adaptable: (Ramos Hernández & Alor Hernández, 2015)

- Lucene es un API de desarrollo para indización y búsquedas, escrita en Java.
- Está disponible en C++, Perl, C# y Ruby.
- Multiplataforma.
- Permite indización incremental.
- Algoritmos de búsquedas fiables y confiables.
- Permite ordenar resultados por relevancia.
- Búsqueda por campos, rangos de fecha, entre otras.
- Ordenación por cualquier campo.
- Permite búsqueda mientras se actualiza el índice.
- Lucene soporta la indización de documentos con formato: TXT, PDF, DOC, RFT, XML, PPT y HTML.

2.6.2 Funcionalidad Básica de Lucene

2.6.2.1 Concepto de Indexación

Cuando se requiere hacer uso de búsquedas dentro de una aplicación, rápido se viene a la mente crear un programa que haga esto, es decir, que busque en todos los archivos palabras o frases relacionadas, esto tendría fallas en el caso de archivos muy grandes. Por eso es importante crear los índices, transformar el texto en un formato donde la búsqueda sea más rápida, eliminando el proceso de exploración lento. Este proceso de conversión es llamado indización y al archivo resultante se le llama índice. Un índice separa las palabras del documento en campos y permite el acceso rápido a los datos que fueron almacenados en el proceso indizado. (Otis Gospodnetic, 2005)

2.6.2.2 Concepto de Búsqueda

La búsqueda es el proceso de entrar al índice y buscar palabras relacionadas, para encontrar documentos donde aparezca. Es importante para la búsqueda tomar en cuenta dos factores: la destitución y la precisión; la destitución se encarga de indicar

que documentos son relevantes a la búsqueda mientras que la precisión se encarga del filtrado de datos. (Otis Gospodnetic, 2005)

2.6.3 Integración de Lucene

Lucene permite añadir capacidades de indexación y búsqueda en sus aplicaciones. También puede indexar y consultar los datos que se pueden convertir a un formato de texto. A continuación se muestra en la figura la integración con Lucene: (Balarezo, 2014)

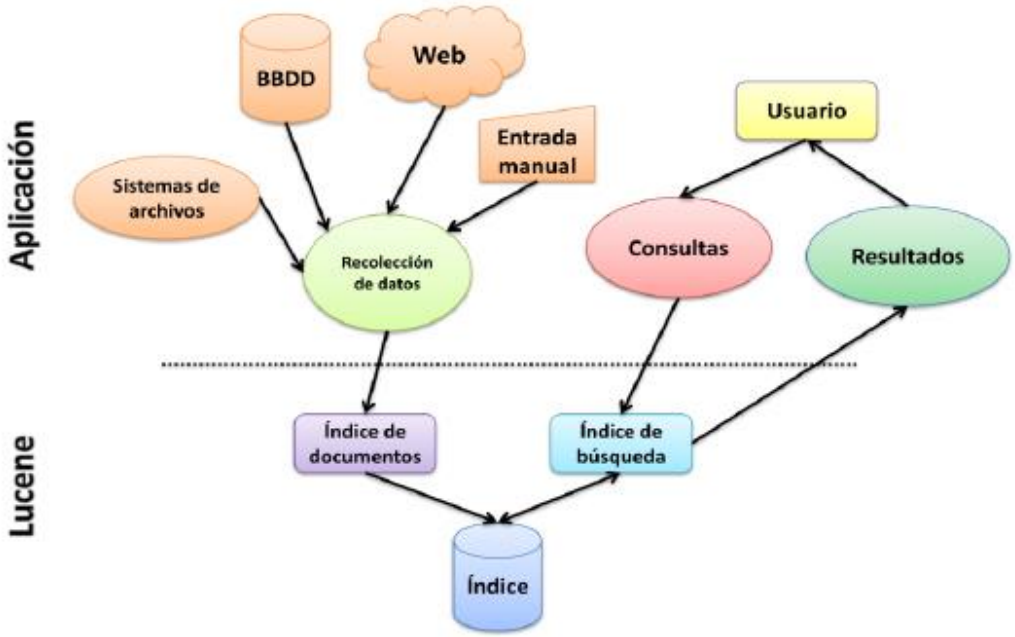

Figura 9. Integración con Lucene

Fuente: (Balarezo, 2014)

2.6.3.1 Componentes de la Indexación

Un índice es como una estructura de datos que permite un rápido acceso aleatorio a palabras almacenadas en su interior. El concepto detrás de esto es análogo a un índice al final de un libro, que le permite localizar rápidamente las páginas que tratan sobre ciertos temas. En el caso de Lucene, un índice es una estructura de datos

especialmente diseñado, típicamente almacenado en el sistema de archivos como un conjunto de archivos de índice. (Balarezo, 2014)

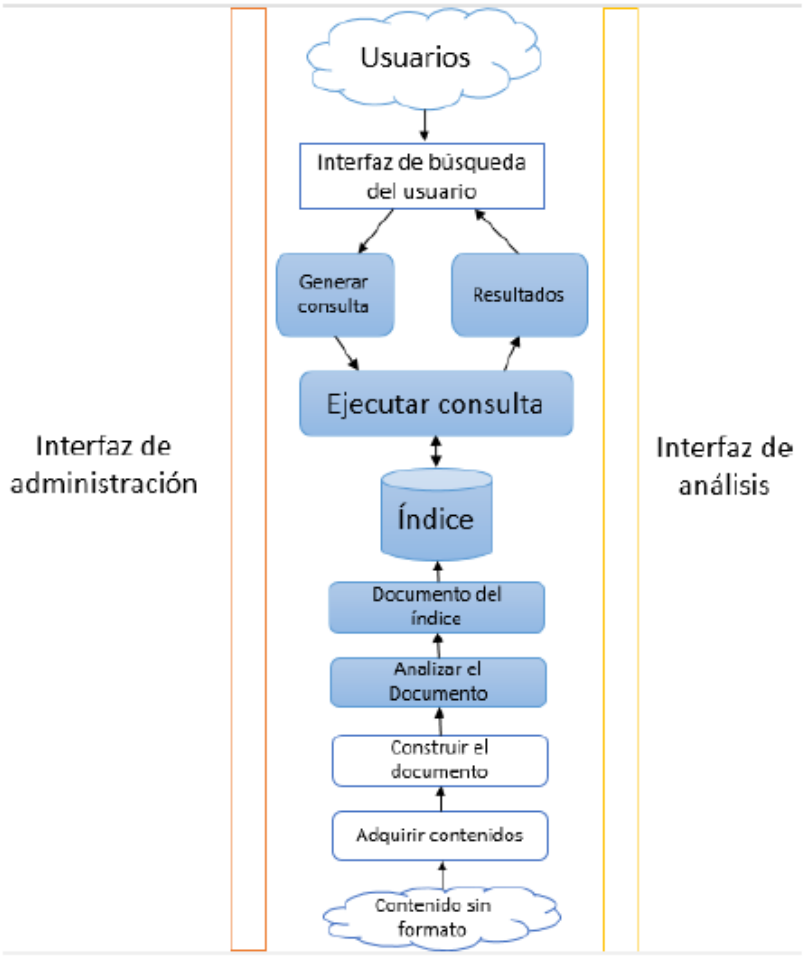

Figura 10. Componentes típicos de la Indexación

Fuente: (Balarezo, 2014)

2.6.3.1.1 Adquirir contenido (Acquire Content)

El primer paso, en la parte inferior de la figura 10, es adquirir contenido. Este proceso, que se refiere a menudo como un crawler (buscador) o araña, recoge el contenido que necesita ser indexado. Eso puede ser trivial, por ejemplo, si desea indexar un conjunto de archivos XML que residen en un directorio específico del sistema de archivos o si todo su contenido reside en una base datos bien organizado. Puede ser terriblemente complejo y desordenado, si el contenido se esparce en todo

tipo de lugares (sistemas de archivos, sistemas de gestión de contenidos, Microsoft Exchange, varios sitios web, bases de datos, archivos XML locales, etc). (Balarezo, 2014)

Lucene, al ser una biblioteca de búsqueda de núcleo, no proporciona ninguna funcionalidad para apoyar la adquisición de contenidos. Esto es totalmente fuera de su aplicación, o una pieza separada de software. (Balarezo, 2014)

2.6.3.1.2 Construir Documento (Build Document)

Una vez que se obtenga el contenido en bruto que necesita ser indexado, debe traducir el contenido a las "unidades" (generalmente llamados "documentos") utilizadas por el motor de búsqueda. El documento consiste típicamente en varios campos denominados por separado con los valores, por ejemplo, el título, cuerpo, resumen, autor, url, etc. (Balarezo, 2014)

2.6.3.1.3 Analizar Documento (Analyze Document)

No hay índices de los motores de búsqueda de texto directamente: más bien, el texto debe ser dividido en una serie de elementos atómicos individuales llamados tokens. Esto es lo que sucede durante el paso de "Analyze document". Cada ficha corresponde aproximadamente a una "palabra" en el idioma, y este paso determina cómo los campos de texto en el documento se dividen en una serie de tokens. (Balarezo, 2014)

2.6.3.1.4 Índice de Documentos (Index Document)

Durante el paso de la indexación, se añade el documento al índice. Lucene, ofrece todo lo necesario para este paso, y trabaja con un poco de magia bajo un sorprendentemente simple API. Es importante recordar que la indexación es algo así como un "mal necesario" que se debe realizar con el fin de proporcionar una buena experiencia de búsqueda: se debe diseñar y personalizar el proceso de indexación en

la medida en que mejora la experiencia de búsqueda de los usuarios. (Balarezo, 2014)

2.6.3.2 Clases Básicas en la Indexación

Las clases básicas e importantes para el proceso de indexación son:

- **IndexWriter**

  Es el componente central del proceso de indexado. Esta clase permite crear un nuevo índice o usar uno ya creado y añadirle documentos. Da permisos de escritura en el índice pero no de lectura o búsqueda.

  Para crear un índice, lo primero que se debe hacer es crear un objeto IndexWriter. El objeto IndexWriter se utiliza para crear el índice y para agregar nuevas entradas de índice. Se puede crear un IndexWriter de la siguiente manera:

```
StandardAnalyzer analyzer = new StandardAnalyzer (Version.LUCENE_XX);
FSDirectory dir = FSDirectory.open(new File("indexpath"));
IndexWriterConfig config = new IndexWriterConfig(Version.LUCENE_XX, analyzer);
IndexWriter writer = new IndexWriter (dir, config);
```

  El primer parámetro es el directorio en el que se creará el índice de Lucene, en este caso es el índice de directorio C:\\.

  El segundo parámetro especifica el “document parser” o “document analyzer”, que se utilizará cuando Lucene indexa sus datos.

  El tercer parámetro indica a Lucene crear un nuevo índice, si el índice no se ha creado en el directorio todavía. (Balarezo, 2014)

- **Directory**

  La clase Directory representa la ubicación de un índice en Lucene. Esta a su vez utiliza subclases FSDirectory para guardar los índices en el sistema de archivos. Esta es la clase que más se usa para el

almacenamiento de índices. La clase IndexWriter hace uso de FSDirectory cuando necesita recibir como parámetro el directorio donde se almacenarán los índices. Otras subclase llamada RAMDirectory, a diferencia de FSDirectory, estas se usa para almacenar los índices en memoria, es recomendable cuando se crean índices pequeños o si se realizan pruebas de indexación o búsqueda. (Otis Gospodnetic, 2005)

- **Analyzer**

  El analizador especificado en el constructor del IndexWriter es el encargado de extraer los tokens del texto que van a ser indexados y eliminar el resto. Hay muchas implementaciones de esta clase que realizan distintos filtros.

  El trabajo del Analyzer es "analizar" cada campo de datos en "tokens" indexables o palabras clave. La siguiente Tabla 5 muestra algunos de los más interesantes: (Balarezo, 2014)

Tabla 5

Analizadores de Lucene

Analyzer	Descripción
StandardAnalyzer	Un sofisticado analizador de propósito general.
WhitespaceAnalyzer	Un analizador muy sencillo que simplemente separa tokens utilizando el espacio en blanco.
StopAnalyzer	Elimina las palabras comunes inglesas que no son generalmente útiles para la indexación.
SnowballAnalyzer	Un interesante analizador experimental que trabaja en las raíces de palabras (Por ejemplo una búsqueda sobre lluvia también debe devolver entradas con llover, llovió, y así sucesivamente).

Existe una serie de analizadores específicos del idioma, incluyendo analizadores para alemán, holandés ruso, francés y otros. Cuando se crea un IndexWriter, se debe especificar qué Analyzer se usará para el índice. (Balarezo, 2014)

- **Document**

  Representa una colección de campos. De algún modo es un documento virtual que se quiere hacer recuperable. Los campos almacenan toda la información del documento, por ejemplo: autor, título tema son indexados y almacenados como campos separados de un documento. (Otis Gospodnetic, 2005)

- **Field**

  Corresponden a porciones de información que van a ser requeridas por el índice durante la búsqueda. Consisten en pares de nombre y valor y pueden ser de 4 tipos: (Balarezo, 2014)

  - **Keyword:** Nos es analizado pero es indexado y almacenado. Se utiliza para valores que deben ser preservados como URLs, rutas de archivos o fechas.
  - **UnIndexed:** No es analizado ni indexado pero si almacenado y se utiliza para valores que se quieren mostrar cuando se realiza una búsqueda pero que nunca son buscados directamente.
  - **UnStored:** El contrario de “UnIndexed”. Este campo es analizado e indexado pero no almacenado en el índice. Se utiliza para grandes cantidades de texto que no deben ser recuperadas en su forma original como los cuerpos de páginas web.
  - **Text:** Es analizado e indexado y es contra lo que se busca.

  Todo campo tiene un nombre y valor, que son pasados como argumentos al definir el tipo de campo a crear. Estos son algunos ejemplos de cómo utilizar la clase Field y, según su tipo si cada campo es analizado, indexado o guardado en el momento de crear el índice. (Balarezo, 2014)

Tabla 6

Tipos de Campos en Lucene

Método Field	Analizado	Indexado	Guardado	Ejemplos
Field.Keyword(String, String) Field.Keyword(String, Date)		**X**	**X**	Teléfonos y números de seguridad social, URLs, nombres de personas, fechas.
Field.UnIndexed(String ,String)			**X**	Tipo de documento (PDF, HTML, etc.), si no son usados como un criterio de búsqueda.
Field.UnStored(String, String)	**X**	**X**		Títulos de documentos y contenidos.
Field.Text(String, String)	**X**	**X**	**X**	
Field.Text(String, Reader)	**X**	**X**		

Fuente: (Otis Gospodnetic, 2005)

2.6.3.3 Clases para la Búsqueda

Para la búsqueda se debe conocer las siguientes clases básicas:

- **IndexSearcher**

  IndexSearcher es en la búsqueda lo que IndexWriter es en la indexación. Es la clase principal que abre el índice para buscar en él y ofrece varios métodos de búsqueda, lo que hace esta clase es pasar como parámetro el "query" o consulta y regresar un objeto hits. (Balarezo, 2014)

- **Term**

  Un término es la unidad básica para la búsqueda. Similar al objeto Field, consiste de un par de elementos: el nombre del campo y su valor. (Balarezo, 2014)

- **Query**

  Lucene tiene diferentes subclases de Query, la más utilizada es "TermQuery" por los métodos que ella contiene. (Balarezo, 2014)

- **QueryParser**

  La clase QueryParser es utilizada para construir un analizador que puede buscar a través de un índice. Un Query es una serie de cláusulas. Una cláusula puede ser un término que indica todos los documentos que contienen un término en particular o de una consulta anidada entre paréntesis. Una consulta anidada se puede utilizar con (+) o (-) de prefijo para requerir cualquier de un conjunto de términos. (Balarezo, 2014)

- **TermQuery**

  Es el tipo de Query más básico soportado por Lucene, se utiliza para hacer coincidir documentos que tienen valores específicos. (Balarezo, 2014)

- **Hits**

  La clase Hits almacena los puntos de referencia a los resultados de la búsqueda, es decir todos los documentos encontrados que se relacionan con el Query. (Balarezo, 2014)

2.6.4 Estructura de un Índice en Lucene

Un índice de Lucene se almacena en un único directorio del sistema de archivos en un disco duro. Los elementos básicos de un índice de Lucene son segmentos, documentos, campos y términos. Un índice de Lucene se compone de uno o más segmentos. Cada segmento contiene uno o más documentos. Cada documento tiene uno o más campos, y cada campo contiene uno o más términos. Cada término es un par de cadenas que representan un nombre de campo y valor.

Un segmento consta de una serie de archivos como se puede ver en la Figura 11. (Balarezo, 2014)

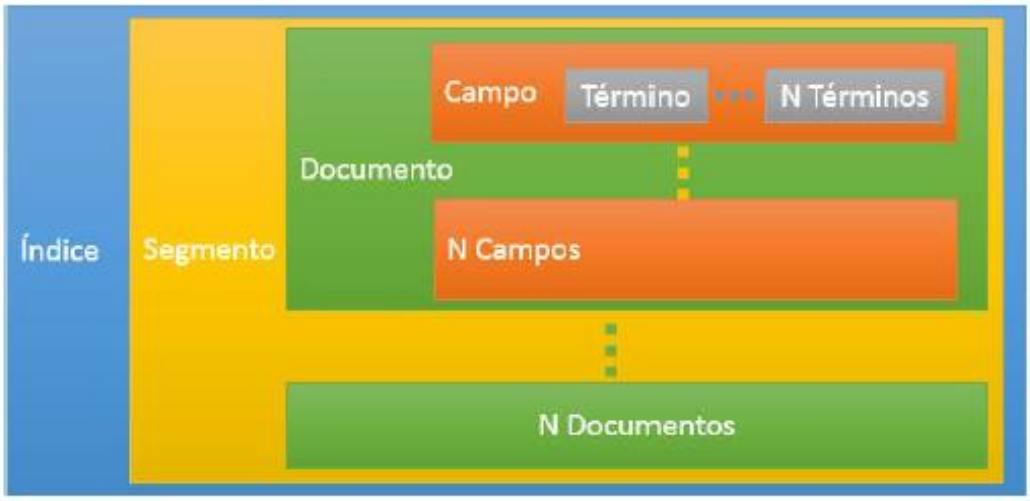

Figura 11. Estructura de un Índice

Fuente: (Balarezo, 2014)

2.6.5 Opciones de Campo

El campo es tal vez la clase más importante al indexar documentos: es la clase real que tiene cada valor para ser indexados. Cuando se crea un campo, hay numerosas opciones especificadas para controlar exactamente lo que Lucene debe hacer con ese campo una vez añadido el documento al índice. Las opciones se dividen en varias categorías independientes, que se cubren en cada subsección siguiente: indexación, almacenamiento y vectores de términos. (Balarezo, 2014)

A continuación se enumeran las combinaciones comunes de opciones de campo: (Balarezo, 2014)

- **Index.ANALYZED:** Se utiliza el analizador para romper el valor del campo en una secuencia de símbolos separados y hacer que cada una de las secuencias se pueden buscar.
- **Index.NOT_ANALYZED:** Indexar el campo, pero no analizar la cadena. En su lugar, tratar el valor entero del campo como un solo símbolo y hacer ese símbolo de búsqueda. Esto es útil para los campos que desea buscar, pero no debe ser dividido, como URLs, rutas del sistema de archivos, fechas, nombres de personas, números de seguro

social, números de teléfono, etc. Esto es especialmente útil para permitir "coincidencia excava" de la búsqueda. Se tiene indexado la ruta del sistema de archivos en el índice usando esta opción.

- **Index.ANALYZED_NO_NORMS:** Una variante avanzada de Index.ANALYZED no almacena información de normas en el índice. Las normas son un registro de información de aumento en el índice, pero puede ser consumida la memoria cuando se busca.
- **Index.NO:** No se tiene el valor de este campo disponible para la búsqueda en todos.

2.7 APACHE TIKA

Apache Tika es una librería que detecta y extrae los metadatos y el contenido de texto estructurado de documentos en diferentes formatos (Microsoft Word, Excel, Power Point, PDF, txt, etc) mediante bibliotecas analizadoras existentes. Tika es un proyecto de la Fundación de Software Apache, y anteriormente fue un subproyecto de Apache Lucene. (Mattmann, 2012)

Figura 12. Extracción de datos con Tika

Fuente: (Tutorialspoint)

2.7.1 Formatos de Documentos Admitidos

A continuación se muestra los formatos de documentos más comunes soportados por Apache Tika y la forma en que se analiza por Tika.

Tabla 7

Formatos de Documentos admitidos por Tika

Tipos de Documentos	Descripción
Lenguaje de Marcas de Hipertexto (HTML)	El lenguaje de marcas de hipertexto HTML es la lengua franca de la web. Tika utiliza la biblioteca "TagSoup" para soportar prácticamente cualquier tipo de HTML que se encuentra en la web.
XML y formatos derivados	El formato de lenguaje de marcado extensible (XML) es un formato genérico que puede ser utilizado para todo tipo de contenido. Tika utiliza la clase DcXMLParser por defecto para extraer el contenido del texto del documento y hace caso omiso de cualquier estructura XML.
Formatos de documentos de Microsoft Office	Las clases "OfficeParser" y "OOXMLParser" utilizan las bibliotecas Apache POI para que admita texto y los metadatos de extracción de ambos documentos OOXML y OLE2.
Formato OpenDocument	El formato OpenDocument (ODF) se utiliza como el formato por defecto de la suite ofimática OpenOffice.org. La clase OpenDocumentParser soporta este formato.
Formato de Documento Portátil	La clase PDFParser analiza los documentos del Formato de Documento Portátil (PDF) usando la librería Apache PDFBox.
Formatos de texto	La clase TXTParser utiliza la codificación de código de detección del proyecto UCI para detectar automáticamente la codificación de caracteres de un documento de texto.
Formatos de audio	Tika puede detectar varios formatos comunes de audio y extraer metadatos de ellos; se utilizan las clases AudioParser y MidiParser.
Formatos de imagen	La clase ImageParser utiliza la

Continúa

	característica javax.imageio estándar para extraer metadatos sencilla de formatos de imagen compatibles con la plataforma Java.
Formatos de video	Actualmente Tika sólo soporta el formato de video Flash mediante un algoritmo de análisis sencillo implementado en la clase FLVParser. La familia MP4 de formatos de video (MP4, QuickTime, 3GPP, etc) es apoyado por la clase MP4Parser, que extrae los metadatos en el video, junto con el flujo de audio.

Fuente: (Balarezo, 2014)

2.8 METODOLOGÍA DE DESARROLLO SCRUM

Es una metodología para la gestión de proyectos, expuesta por Hirotaka Tackeuchi e Ikujiro Nonaka, en las que pone de manifiesto que, el mercado exige ciclos de desarrollo más cortos. Scrum es un proceso que incluye un conjunto de prácticas y roles predefinidos, un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final, que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite. (2.0, 2014)

La metodología SCRUM aplicada al desarrollo de software, está basado en el modelo de las metodologías ágiles, incrementales, basadas en iteraciones y revisiones continuas. Se caracteriza por: (Toapanta Chancusi, Vergara Ordoñez, & Campaña Ortega)

- Eleva al máximo la productividad del equipo de desarrollo.
- Reduce al máximo las actividades no orientadas a producir software funcional.
- Produce resultados en periodos cortos de tiempo.

2.8.1 Fases de la Metodología SCRUM

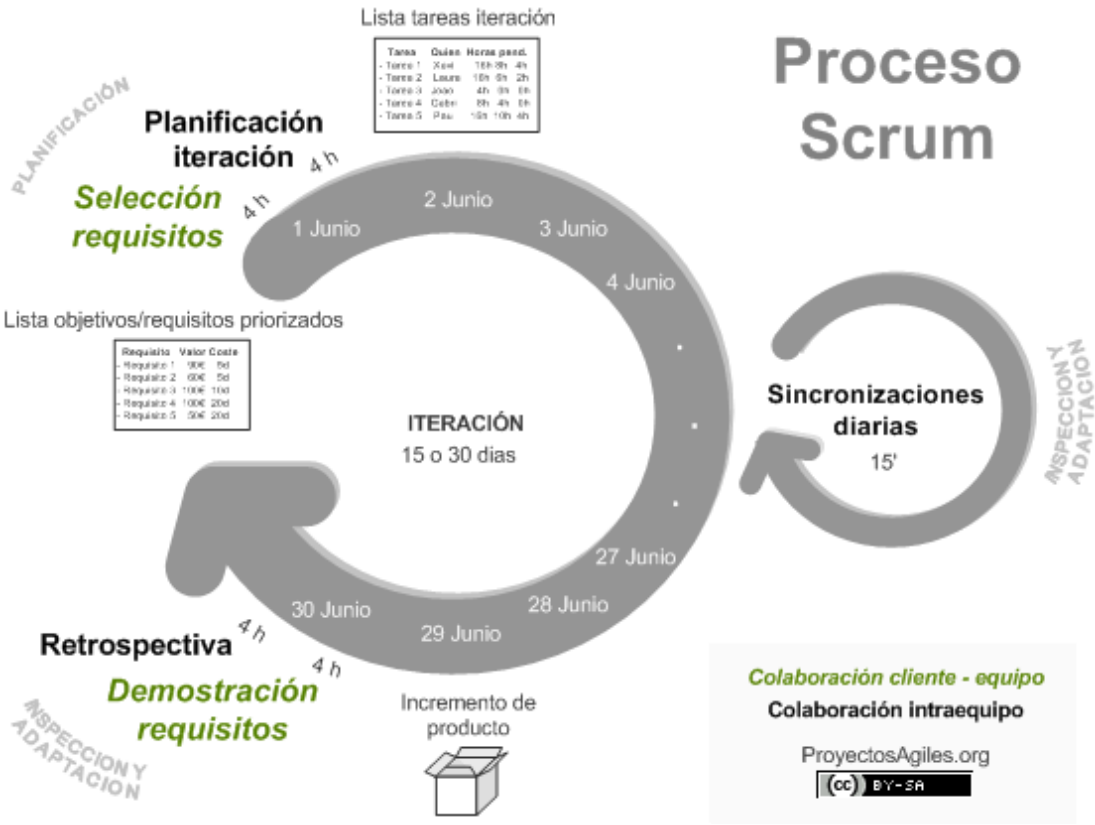

Figura 13. Fases de la Metodología SCRUM

Fuente: (2.0, 2014)

Las actividades que se llevan a cabo en Scrum son las siguientes:

2.8.1.1 Planificación de la iteración

El primer día de la iteración se realiza la reunión de planificación de la iteración. Consta de dos partes:

- **Selección de requisitos** (4 horas máximo): El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.

- **Planificación de la iteración** (4 horas máximo): El equipo elabora la lista de tareas de la iteración, necesarias para desarrollar los requisitos a los que se ha comprometido. La estimación de esfuerzos, se hace de manera conjunta y los miembros del equipo se auto-asignan las tareas. (2.0, 2014)

2.8.1.2 Ejecución de la iteración

Cada día el equipo realiza una reunión de sincronización (15 minutos máximos). Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido. En la reunión cada miembro del equipo responde a tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener?

Durante la iteración el facilitador se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad, a partir de las siguientes tareas:

- Elimina los obstáculos que el equipo no puede resolver por sí mismo.
- Protege al equipo de interrupciones externas que puedan afectar su compromiso o su productividad. (2.0, 2014)

2.8.1.3 Inspección y adaptación

El último día de la iteración se realiza la reunión de revisión de la iteración. Esta revisión, se divide en dos partes:

- **Demostración** (4 horas máximo): El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados

mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, se replanifica el proyecto.

- **Retrospectiva** (4 horas máximo): El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continúa su productividad. El facilitador se encargará de ir eliminando los obstáculos identificados. (2.0, 2014)

CAPÍTULO 3
ANÁLISIS Y DISEÑO DEL PROYECTO

3.1 PLANIFICACIÓN

El sistema de indexación y búsqueda en relación con los requerimientos del sistema se desarrollará de la siguiente manera:

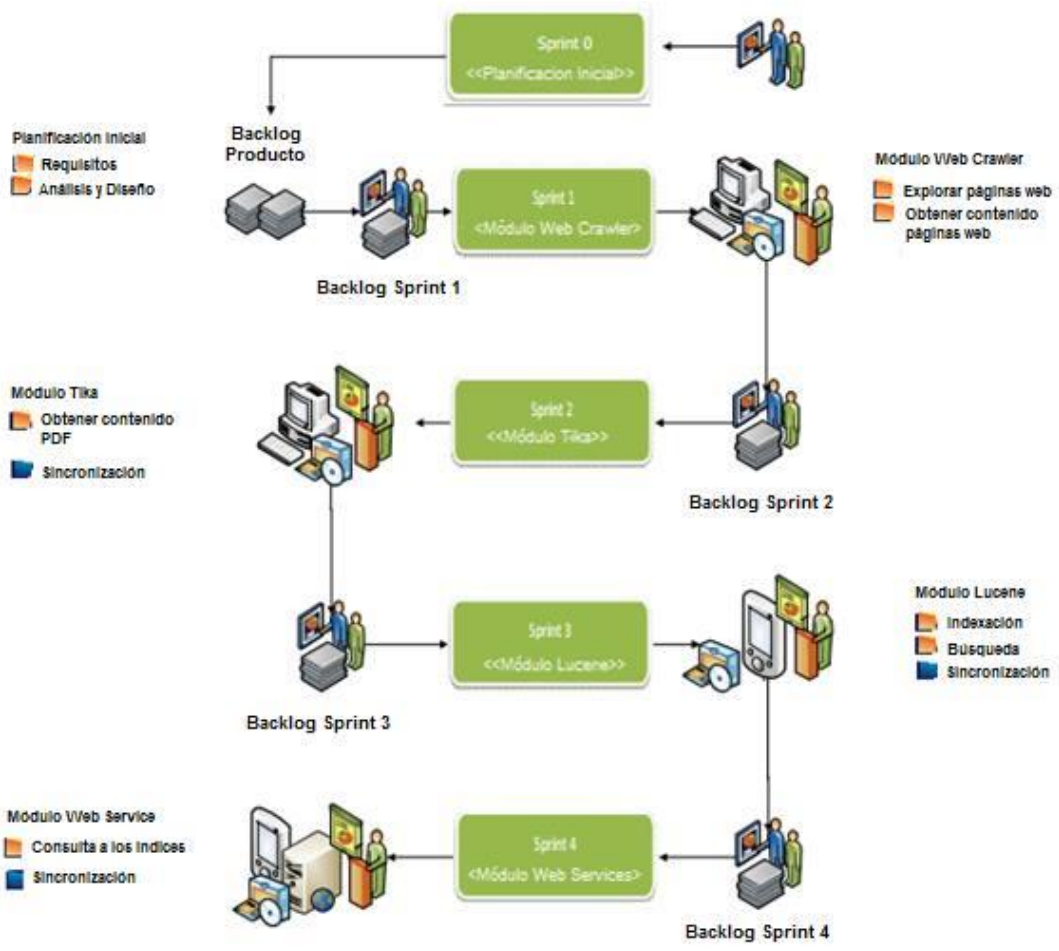

Figura 14. Planificación del Proyecto

3.1.1 Análisis de Procesos

El sistema para cumplir sus objetivos y alcance, implementará como parte de la ejecución del proyecto los siguientes procesos.

- Proceso – Búsqueda de información por medio del Web Crawler: Proceso mediante el cual se recolecta información correspondiente a las páginas web del listado de las Universidades del Senescyt.
- Proceso – Indexación de documentos: Proceso por el cual se indexan las búsquedas almacenadas en la base de datos y los archivos descargados PDF descargados.
- Proceso – Retorno de las consultas solicitas: Proceso mediante el cual se busca la información solicitada en la indexación.

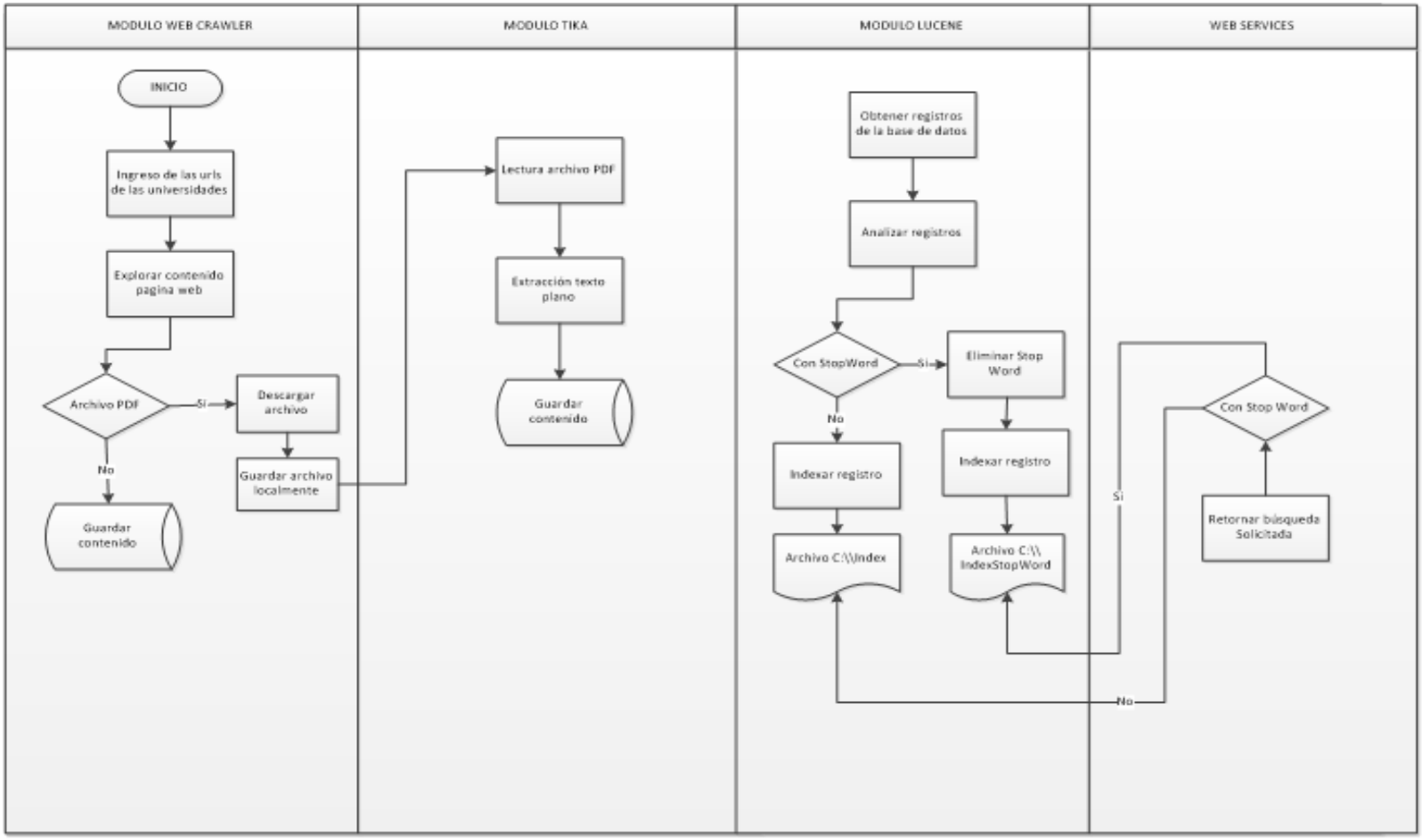

Figura 15. Procesos del Proyecto

3.1.2 Especificación de Requerimientos

De los procesos descritos y el respectivo diagrama, se han identificado los requisitos funcionales y no funcionales con los cuales debe cumplir el sistema a implementar.

3.1.2.1 Requerimientos Funcionales

Tabla 8

Resumen de Requerimientos Funcionales

Referencia	Requerimiento Funcional	Descripción
RE01	Búsqueda y almacenamiento de información	Permite buscar y almacenar información relacionada con las Carreras de Ciencias de la Computación de los Niveles Postgrado y Doctorado en el listado de universidades del Senescyt.
RE02	Búsqueda y almacenamiento de archivos PDF	Permite buscar y almacenar los documentos PDF relacionados con las Carreras de Ciencias de la Computación de los Niveles Postgrado y Doctorado en el listado de universidades del Senescyt.
RE03	Lectura de archivo PDF y extracción de texto	Permite leer y extraer el contenido de los PDF para almacenarlo en la base de datos.
RE04	Creación de índices e Indexación de documentos	Se crea los índices de búsqueda, para su posterior indexación de los documentos es decir de los registros almacenados y filtrados de la base de datos.
RE05	Envió de información solicitada	El indexador Lucene busca la información solicitada por el Sistema AQUINO, y envía para su posterior proceso.

3.1.2.2 Requerimientos No Funcionales

3.1.2.2.1 Interfaces de Software

- Plataforma JEE6
- Lenguaje de programación Python
- Lenguaje de programación Java.
- Base de datos NoSQL MongoDB
- Indexador Lucene Apache.

3.1.2.2.2 Interfaces de Comunicación

El Web Crawler se comunica con la base de datos NoSQL para el almacenamiento de los datos, a su vez el indexador Lucene se comunica con la base de datos para obtener datos requeridos por el FrontEnd del sistema AQUINO, los cuales son enviados por medio de un Web Service.

3.1.2.3 Interfaces del Usuario

El aplicativo no presenta interfaces gráficas al usuario, ya que se trata del BackEnd del sistema web AQUINO con el que va a ser integrado.

3.1.2.4 Interfaces de Hardware

Tabla 9

Requerimientos Mínimos de Hardware

	Aplicativo (Escritorio)	Servidor DBM
Procesador	1.8 GHz o mayor	1.8 GHz o mayor
Memoria RAM	512 MB	512 MB
Monitor	SVGA de 1024x768 px	SVGA de 1024x768 px
	Teclado estándar	Teclado estándar
	Ratón	Ratón

Tabla 10

Requerimientos Óptimos de Hardware

	Aplicativo (Escritorio)	Servidor DBM
Procesador	Intel Core i7-2.40 GHz	Intel Core i7-2.40 GHz
Memoria RAM	6 GB	6GB
Monitor	AMD Radeon HD 7600M Series	AMD Radeon HD 7600M Series
	Teclado estándar y ratón	Teclado estándar y ratón

3.2 ALCANCE DEL SOFTWARE

De acuerdo al análisis de los procesos y las funcionalidades identificadas, el software a desarrollarse tendrá los siguientes alcances.

- Módulo Web Crawler, que permitirá:
  - Buscar, explorar las páginas web del Listado de Universidades del Senescyt y a su vez descargar su contenido para su posterior análisis en el idioma español.
  - Descargar archivos PDF relacionados con información del Listado de Universidades del Senescyt.
- Módulo Lucene, que permitirá:
  - Obtener la información de la base de datos para la indexación de documentos los cuales servirán en la búsqueda posterior.
  - Búsqueda de información de acuerdo a criterios específicos de búsqueda.
- Módulo Tika
  - Lectura de los archivos descargados para su posterior análisis e indexación.
- Módulo Web Services, que permitirá :
  - Retorna información solicitada por el Front End del sistema AQUINO para mostrar al usuario.

3.3 CONFORMACIÓN DEL EQUIPO DE TRABAJO

Al igual que en todo tipo de proyecto, es necesario conocer el equipo humano con el cual se trabajara en el proyecto.

El equipo de trabajo para llevar a cabo el sistema para la búsqueda e indexación de información relacionada con las páginas web del Listado de Universidades del Senescyt estará conformado según lo descrito en la tabla 11.

Tabla 11

Equipo de Trabajo y Roles

ROL	Persona	Área
Product Owner	Ing. Ernesto Nieto	Becas – Senescyt
SCRUM Manager	Dr. Felipe Mota	
Equipo	Natalia Cerón Mireya Chillán Ramiro Andrade	Desarrollo Desarrollo Desarrollo

3.4 DEFINICIÓN DEL BACKLOG DEL PRODUCTO

El Backlog del producto contiene toda la funcionalidad que el producto final deberá tener lo que indica la metodología. Para el presente proyecto se ha elaborado el Backlog del producto. Identificando las funcionalidades, realizando una estimación del tiempo requerido y a su vez priorizando cada una de ellas.

De acuerdo al estudio de los procesos y los requerimientos identificados durante la etapa de planificación, el Backlog del producto para el presente proyecto se encuentra definido en la siguiente tabla:

Tabla 12

Backlog del Producto

ID	Nombre	Importancia	Tiempo estimado (semanas)	Comentarios
1	Módulo Web Crawler	80	3	Funcionalidad para la búsqueda de información
2	Módulo Lucene	90	3	Funcionalidad para la indexación de información
3	Módulo Tika	70	1	Funcionalidad para lectura e indexación de archivos PDF
4	Módulo Web Services	100	1	Funcionalidad para el retorno de información solicitada.

Nota: La importancia está estimada de acuerdo, a las necesidades del Propietario del producto y está cuantificada con números enteros entre el 0(Ninguna importancia) y 100 (alta importancia).

3.5 DISEÑO

3.5.1 Modelo de Datos

El sistema almacena grandes cantidades de información por lo cual se ha implementado una base de datos No SQL documental.

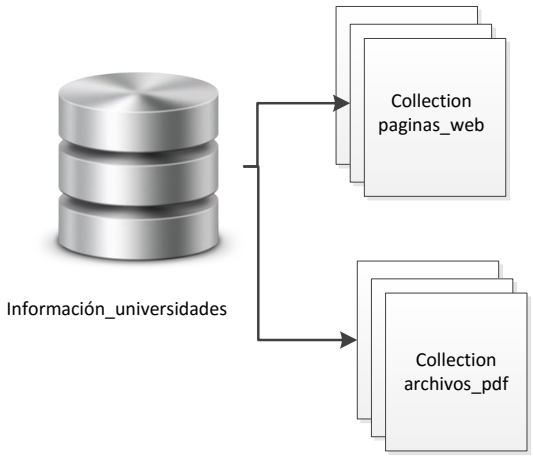

Figura 16. Base de datos NoSQL documental

3.5.2 Arquitectura General

Para la implementación del sistema Búsqueda e indexación de información relacionado con el Listado de Universidades del Senescyt, se utilizada un modelo N capas, distribuidas de la siguiente manera.

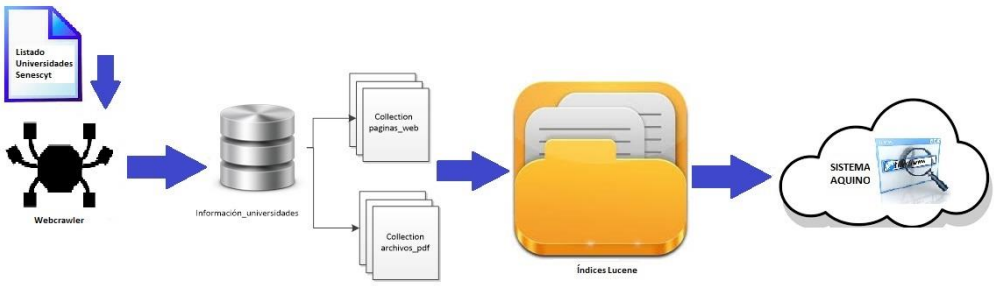

Figura 17. Estructura General del Sistema AQUINO – Web Crawler

3.6 SPRINT 1 "MÓDULO WEB CRAWLER"

El primer sprint tiene como objetivo implementar las funcionalidades requeridas para el desarrollo del web crawler, esto es: búsqueda de información y descarga de archivos relacionados con las páginas web del listado de universidades de la Senescyt.

3.6.1 Planificación

Para la planificación del Sprint1, se llevó a cabo una reunión con el Propietario del Producto. En esta reunión se realizó el análisis de las funcionalidades relacionadas con el Web Crawler que serán implementadas.

De los Requerimientos Funcionales y No Funcionales – Sprint0 se selecciona las funcionalidades correspondientes al módulo que serán el objetivo del Sprint1.

Tabla 13

Requerimiento Funcional 1: Búsqueda y almacenamiento de información

Id. Requerimiento	RE01 Búsqueda y almacenamiento de información
Descripción	Permite buscar y almacenar información relacionada con las Carreras de Ciencias de la Computación de los Niveles Postgrado y Doctorado en el listado de universidades del Senescyt.
Entradas	URLs del listado de las universidades del Senescyt.
Salidas	Contenido de las páginas web sin etiquetas HTML y su respectiva URL, almacenados en la base de datos.
Proceso	El Web Crawler obtiene el contenido de las páginas web de las universidades de listado del Senescyt, las analiza y guarda aquellas que tengan información relacionada con Carreras de Ciencias de la Computación.
Precondiciones	Conexión a internet
Postcondiciones	Se actualizará la tabla de almacenamiento de información
Efectos Colaterales	La base de datos se actualizará el 1ero de cada mes, ejecutándose el web crawler para recopilar nueva información.
Prioridad	Alta
Rol que lo ejecuta	Proceso Bach

Tabla 14

Requerimiento Funcional 2: Búsqueda y almacenamiento de archivos PDF

Id. Requerimiento	RE02 Búsqueda y almacenamiento de archivos PDF
Descripción	Permite buscar y almacenar los documentos PDF relacionados con las Carreras de Ciencias de la Computación de los Niveles Postgrado y Doctorado en el listado de universidades del Senescyt.
Entradas	URLs del listado de las universidades del Senescyt.
Salidas	Documentos en formato PDF almacenados en un directorio local.
Proceso	El Web Crawler obtiene el contenido de las páginas web de las universidades de listado del Senescyt, las analiza y guarda localmente los archivos PDF que tengan información relacionada con Carreras de Ciencias de la Computación.
Precondiciones	Conexión a internet
Postcondiciones	Se descarga los archivos PDF y se guardan en un directorio local.
Efectos Colaterales	La descarga de archivos PDF se actualizará el 1ero de cada mes, ejecutándose el web crawler para recopilar nueva información.
Prioridad	Alta
Rol que lo ejecuta	Proceso Bach

De acuerdo a las funcionalidades identificadas para el módulo del Web Crawler, se identifican las historias de usuario para el Sprint1.

Tabla 15

Historia de Usuario-Sprint 1

ID	Historia de usuario	Importancia Propietario del Producto	Importancia Técnica	Descripción
1	Definición urls de las páginas web de listado de universidades de la Senescyt	100		Es necesario filtrar las páginas web en el idioma español.

Continúa

2	Desarrollo del web crawler	90	100	Consiste en obtener el código fuente de las páginas web
3	Descarga de archivos	80	90	El sistema requiere de información de archivos PDF, por lo tanto aquellos links que tengas información relacionada a las universidades se descargan para su posterior análisis.

Nota: La importancia está estimada de acuerdo a las necesidades del Propietario del Producto y está cuantificada con números enteros entre 0 y 100. La importancia técnica está basada en las necesidades no funcionales desde el punto de vista del usuario, pero necesarias para un funcionamiento de la aplicación desde el punto de vista técnico.

3.6.1.1 Definición del Equipo de Trabajo

El equipo de trabajo para la implementación de las funcionalidades del módulo del Web Crawler esta descrito en la siguiente tabla:

Tabla 16

Equipo de Trabajo-Sprint 1

ROL		Persona	Descripción/ Tareas
Product Owner		Ing. Ernesto Nieto	Administra el proyecto desde la perspectiva del negocio.
SCRUM Master		Dr. Felipe Mota	Asegurar que el proceso SCRUM se lleve a cabo.
Team	Codificación	Mireya Chillan	Codificación de las funcionalidades identificadas
	Pruebas	Natalia Cerón	Pruebas

3.6.2 Análisis

Una vez identificadas las historias de usuario, se identifican los actores y se diagrama los casos de uso identificados para la implementación del Sprint 1.

3.6.2.1 Actores del Sistema

La siguiente tabla describe los actores que participan en los casos de uso identificados para el modulo Web Crawler.

Tabla 17

Actores del Sistema-Sprint 1

Actor	Descripción
Administrador	Usuario administrador del sistema que puede gestionar los links de las universidades así como los archivos descargados

3.6.2.2 Diagramas de Casos de Uso del Módulo Web Crawler

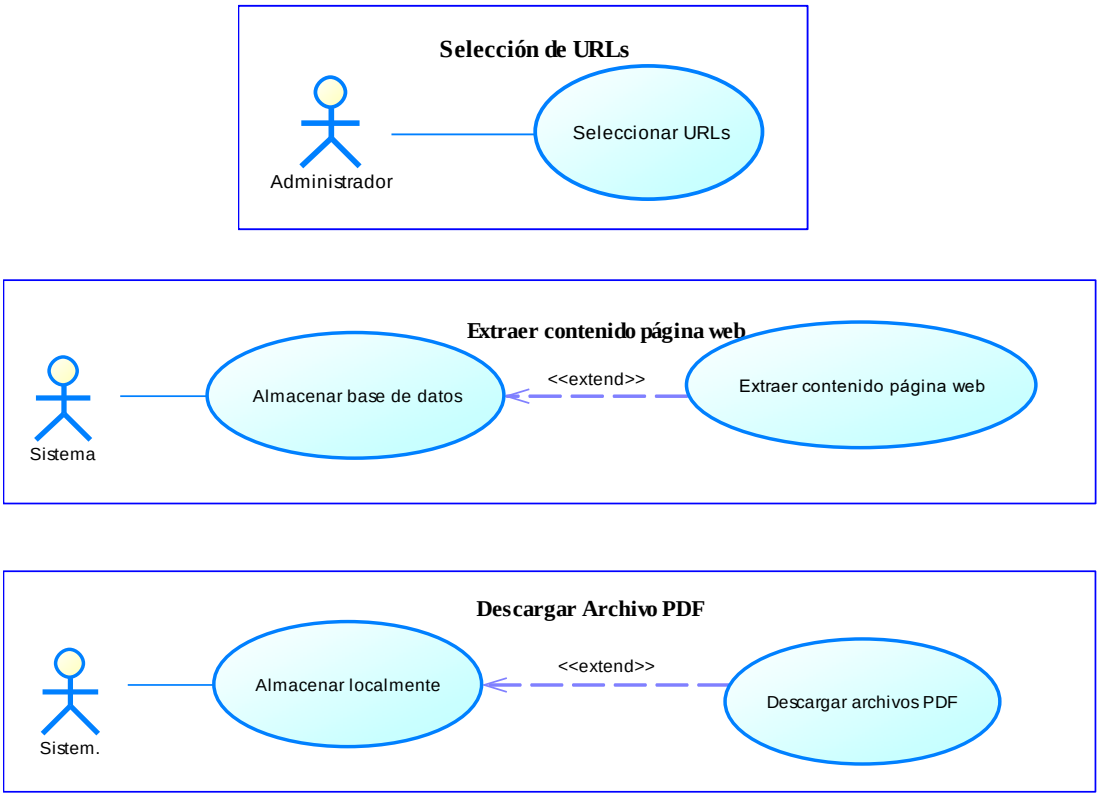

Figura 18. Casos de Uso-Módulo Web Crawler

3.6.2.3 Especificación de Casos de Uso del Módulo Web Crawler

Tabla 18

Especificación del Caso de Uso: Selección de URLs

ID	RF-01
Descripción	Proporcionar las URL para el funcionamiento del Web Crawler
Precondición	Listado de universidades del Senescyt
Postcondición	
Flujo Normal	1. Revisar lista de universidades del Senescyt 2. Comprobar validez de las urls.
Flujo Alterno	Si la URL no es válida borrar de la lista.
Excepciones	
Notas	Las páginas web deben estar en idioma español.

Tabla 19

Especificación del Caso de Uso: Extraer contenido página web

ID	RF-02
Descripción	Extraer el contenido de la página web de las universidades del listado de universidades del Senescyt.
Precondición	RF – 01
Postcondición	Información actualizada
Flujo Normal	1. Explorar la página web y sus links 2. Analizar el contenido de las paginas 3. Guarda contenido en la base de datos
Flujo Alterno	Si el url pertenece a un archivo PDF se ejecuta el RF-03
Excepciones	Si la página no contiene información relacionada con educación no se almacena
Notas	

Tabla 20

Especificación del Caso de Uso: Extraer contenido página web

ID	RF-03
Descripción	Descargar el archivo PDF encontrado y guardarlo localmente en el computador.
Precondición	
Postcondición	Información actualizada
Flujo Normal	1. Explorar link perteneciente a un archivo PDF 2. Descargar archivo PDF 3. Guardar localmente el archivo
Flujo Alterno	Si el url no es un archivo PDF se ejecuta el RF-03
Excepciones	Si el archivo no contiene información relacionada con educación no se almacena
Notas	

3.6.3 Diseño

3.6.3.1 Modelo de Datos

Del análisis de la especificación de los casos de uso, se determina la necesidad de utilizar un modelo de datos no relacional descrito en el siguiente diagrama.

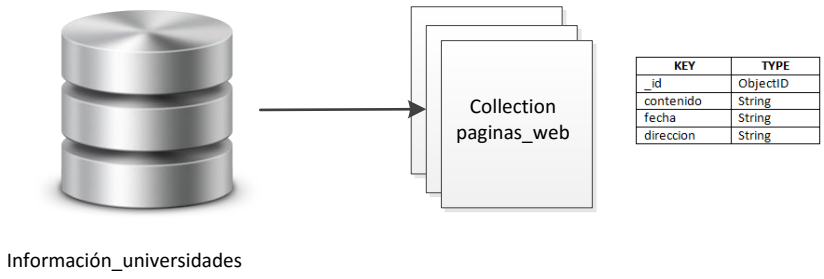

Figura 19. Modelo de datos

3.6.4 Arquitectura

El siguiente diagrama describe de manera detallada como se implementó el Modulo Web Crawler.

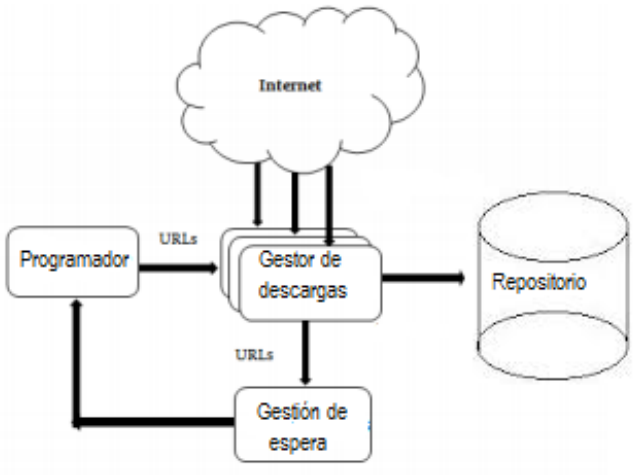

Figura 20. Arquitectura del Módulo Web Crawler

3.6.5 Construcción y Pruebas

3.6.5.1 Backlog del Sprint

Una vez identificadas las historias de usuario, las cuales se encuentran descritas en los casos de uso y sus respectivas especificaciones, estás serán implementadas en el primer sprint; para ello es necesario definir el Backlog del sprint es decir la pila de tareas que permitan implementar las funcionalidades especificadas en la etapa de análisis. La pila de tareas para el Sprint1 se muestra en la siguiente tabla:

Tabla 21

Backlog Sprint 1 "Módulo Web Crawler"

Story ID	Task #	Story Name, Task Name	Assigned 1	Estimado (horas)
1		Desarrollo Web Crawler		
	1	Codificación para extraer el contenido de una pagina web.	Mireya Chillan	16
	2	Almacenamiento base de datos	Mireya Chillan	4
	3	Pruebas de desarrollo	Natalia Cerón	2
	4	Pruebas unitarias	Natalia Cerón	1
2		Descarga archivos PDF	Mireya Chillan	4

Continúa

	1	Almacenamiento local de archivos	Mireya Chillan	4
	2	Pruebas de desarrollo	Natalia Cerón	2
	3	Pruebas unitarias	Natalia Cerón	1

3.6.5.2 Pruebas

Las pruebas que se realizaron dependieron del módulo y componentes de los mismos. Los resultados de las pruebas son adjuntados como anexos al presente documento.

3.7 SPRINT 2 "MÓDULO TIKA"

El segundo sprint tiene como objetivo implementar las funcionalidades requeridas para los archivos descargados por el Web Crawler, estas son: lectura y almacenamiento en la base de datos.

3.7.1 Planificación

De los requerimientos Funcionales / No Funcionales obtenidos durante el Sprint0, se selecciona las funcionalidades correspondientes al módulo de gestión en este Sprint.

Tabla 22

Requerimiento Funcional 3: Lectura de archivo PDF y extracción de texto

Id. Requerimiento	RE03 Lectura de archivo PDF y extracción de texto
Descripción	Permite leer y extraer el contenido de los PDF para almacenarlo en la base de datos.
Entradas	Archivos PDF.
Salidas	Contenido de los PDF en formato texto plano, almacenados en la base de datos.
Proceso	Usando la librería Apache Tika de Java se obtiene el contenido de los PDF en formato texto y se lo almacena en la base de datos.
Precondiciones	Documentos PDF descargados en directorio local.

Continúa

Postcondiciones	Se actualizará la tabla de almacenamiento de información.
Efectos Colaterales	La base de datos se actualizará el 1ero de cada mes, ejecutándose el web crawler para recopilar nueva información.
Prioridad	Alta
Rol que lo ejecuta	Proceso Bach

Una vez identificados los requerimientos funcionales y no funcionales se plantean las historias de usuario que deben ser implementadas durante el desarrollo del Sprint2.

Tabla 23

Historia de Usuario-Sprint 2

Id	Historia de usuario	Importancia Product Owner	Importancia Técnica	Descripción
1	Lectura de archivos PDF	90	100	Consiste en abrir cada uno de los archivos y recolectar solo texto, para su procesamiento.
2	Almacenamiento base de datos	100	100	Permite guardar los datos de las lecturas realizadas a los archivos para su posterior gestión.

Nota: La importancia está estimada de acuerdo a las necesidades del Product Owner y está cuantificada con números enteros entre 0 y 100. La importancia técnica está basada en las necesidades no funcionales desde el punto de vista del usuario, pero necesarias para el funcionamiento de la aplicación desde el punto de vista técnico.

3.7.1.1 Definición del Equipo de Trabajo

El equipo de trabajo para la implementación de las funcionalidades del módulo del Web Crawler esta descrito en la siguiente tabla:

Tabla 24

Equipo de Trabajo-Sprint 2

ROL		Persona	Descripción/ Tareas
Product Owner		Ing. Ernesto Nieto	Administra el proyecto desde la perspectiva del negocio.
SCRUM Master		Dr. Felipe Mota	Asegurar que el proceso SCRUM se lleve a cabo.
Team	Codificación	Natalia Cerón	Codificación de las funcionalidades identificadas
	Pruebas	Mireya Chillan	Pruebas

3.7.2 Análisis

Una vez identificadas las historias de usuario, se identifican los actores y se diagrama los casos de uso identificados para la implementación del Sprint 2.

3.7.2.1 Actores del Sistema

La siguiente tabla describe los actores que participan en los casos de uso identificados para el módulo Tika.

Tabla 25

Actores del Sistema-Sprint 2

Actor	Descripción
Administrador	El administrador especifica la ruta de descarga y almacenamiento local para los archivos PDF.

3.7.2.2 Diagramas de Casos de Uso del Módulo Tika

Figura 21. Casos de Uso-Módulo Tika

3.7.2.3 Especificación de Casos de Uso del Módulo Tika

Tabla 26

Especificación del Caso de Uso: Lectura de archivos PDF y almacenamiento del contenido en la base de datos

ID	RF-04
Descripción	Leer los archivos PDF descargado y almacenar su contenido en la base de datos.
Precondición	RF – 03
Postcondición	Base de datos actualizada
Flujo Normal	1. Leer el archivo PDF 2. Extraer el texto plano 3. Almacenar en la base de datos
Flujo Alterno	
Excepciones	Se almacena solo texto plano
Notas	Los archivos PDF pueden estar en idioma español.

3.7.3 Diseño

3.7.3.1 Modelo de Datos

Del análisis de la especificación de los casos de uso, se determina la necesidad de utilizar un modelo de datos no relacional descrito en el siguiente diagrama.

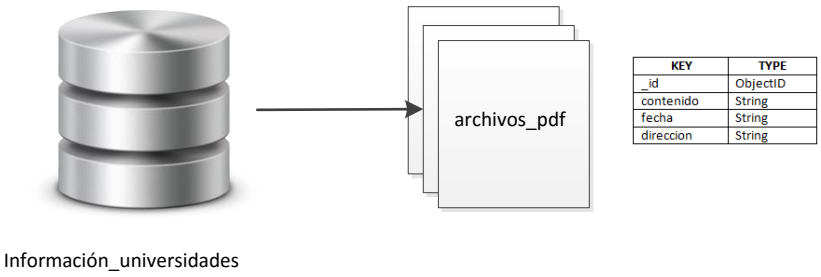

Figura 22. Modelo de datos PDF

3.7.4 Arquitectura

El siguiente diagrama describe de manera detallada como se implementó el Modulo Tika para la lectura y extracción de texto de los archivos PDF.

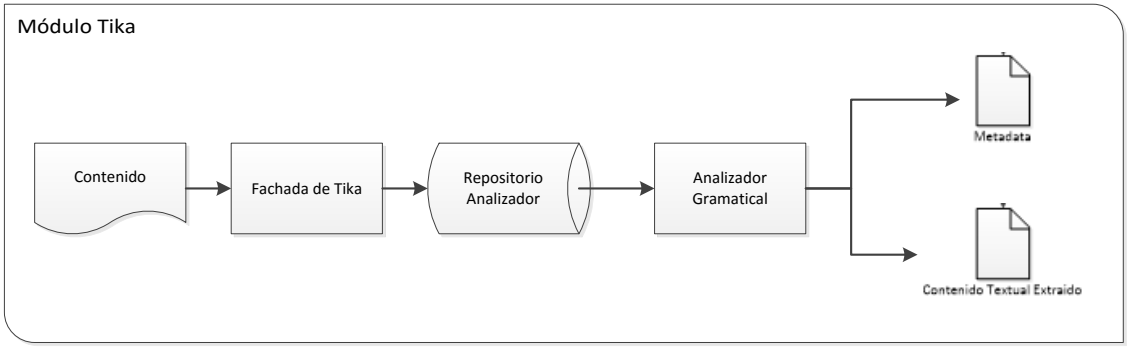

Figura 23. Arquitectura del Módulo Tika

3.7.5 Construcción y Pruebas

3.7.5.1 Backlog del Sprint

Una vez identificadas las historias de usuario, las cuales se encuentran descritas en los casos de uso y sus respectivas especificaciones, estás serán implementadas en el segundo sprint; para ello es necesario definir el Backlog del sprint es decir la pila de tareas que permitan implementar las funcionalidades especificadas en la etapa de análisis. La pila de tareas para el Sprint 2 se muestra en la siguiente tabla:

Tabla 27

Backlog Sprint 2 "Módulo Tika"

Story ID	Task #	Story Name, Task Name	Assigned 1	Estimado (horas)
1		Desarrollo de Apache Tika		
	1	Codificación de Apache Tika	Natalia Cerón	16
	2	Almacenamiento base de datos	Natalia Cerón	4
	3	Pruebas de desarrollo	Mireya Chillán	2
	4	Pruebas unitarias	Mireya Chillán	1

3.7.5.2 Pruebas

Las pruebas que se realizaron dependieron del módulo y componentes de los mismos. Los resultados de las pruebas son adjuntados como anexos al presente documento.

3.8 SPRINT 3 "MÓDULO LUCENE"

El tercer sprint tiene como objetivo implementar las funcionalidades requeridas para la creación de índices e indexación de documentos.

3.8.1 Planificación

De los Requerimientos Funcionales y No Funcionales – Sprint0 se selecciona las funcionalidades correspondientes al módulo que serán el objetivo del Sprint 3.

Tabla 28

Requerimiento Funcional 4: Creación de índices e Indexación de documentos

Id. Requerimiento	RE04 Creación de índices e indexación de documentos
Descripción	Se crea los índices de búsqueda, para su posterior indexación de los documentos es decir de los registros almacenados y filtrados de la base de datos.
Entradas	Registros de la base de datos.
Salidas	Índices generados de acuerdo a las Carreas de las Ciencias de la Computación.
Proceso	El indexador Lucene crea los índices establecidos para la

	indexación de los documentos generados de acuerdo a los registros filtrados de la base de datos.
Precondiciones	Acceso a la base de datos.
Postcondiciones	Índices creados con documentos indexados.
Efectos Colaterales	Ninguno
Prioridad	Alta
Rol que lo ejecuta	Automático

De acuerdo a las funcionalidades identificadas para el módulo de Lucene, se identifican las historias de usuario para el Sprint 3.

Tabla 29

Historia de Usuario-Sprint 3

ID	Historia de usuario	Importancia Propietario del Producto	Importancia Técnica	Descripción
1	Obtener registros de la base de datos	90		Es necesario obtener los registros de la base de datos para poder crear los índices y los documentos respectivos.
2	Creación de índices	100		Es necesario crear los índices (directorios) para poder indexar documentos.
3	Indexación de documentos	90	100	Consiste en indexar los documentos que se encuentran almacenados en la base de datos (información de las universidades y archivos PDF) y guardarlos según el parámetro de las stopwords para facilitar la posterior búsqueda de información.

Continúa

Nota: La importancia está estimada de acuerdo a las necesidades del Propietario del Producto y está cuantificada con números enteros entre 0 y 100. La importancia técnica está basada en las necesidades no funcionales desde el punto de vista del usuario, pero necesarias para un funcionamiento de la aplicación desde el punto de vista técnico.

3.8.1.1 Definición del Equipo de Trabajo

El equipo de trabajo para la implementación de las funcionalidades del módulo de Lucene esta descrito en la siguiente tabla:

Tabla 30

Equipo de Trabajo-Sprint 3

ROL		Persona	Descripción/ Tareas
Product Owner		Ing. Ernesto Nieto	Administra el proyecto desde la perspectiva del negocio.
SCRUM Master		Dr. Felipe Mota	Asegurar que el proceso SCRUM se lleve a cabo.
Team	Codificación	Natalia Cerón	Codificación de las funcionalidades identificadas
	Pruebas	Mireya Chillan	Pruebas

3.8.2 Análisis

Una vez identificadas las historias de usuario, se identifican los actores y se diagrama los casos de uso identificados para la implementación del Sprint 3.

3.8.2.1 Actores del Sistema

La siguiente tabla describe los actores que participan en los casos de uso identificados para el módulo Lucene.

Tabla 31

Actores del Sistema-Sprint 3

Actor	Descripción

Administrador	El administrador puede gestionar la creación de índices y la indexación de documentos.

3.8.2.2 Diagramas de Casos de Uso del Módulo Lucene

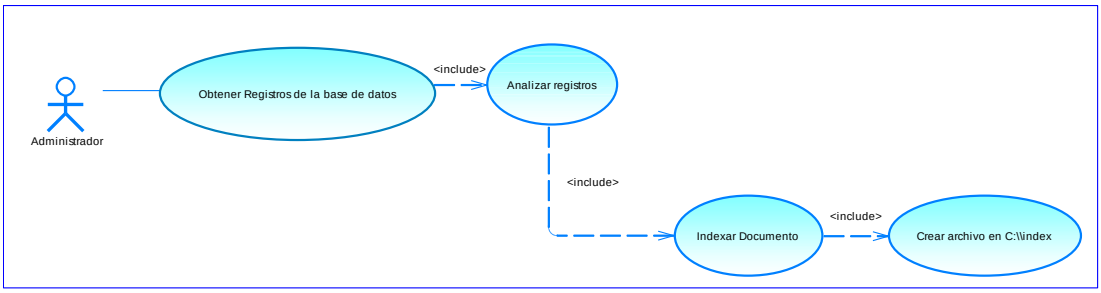

Figura 24. Casos de Uso-Módulo Lucene Indexación normal

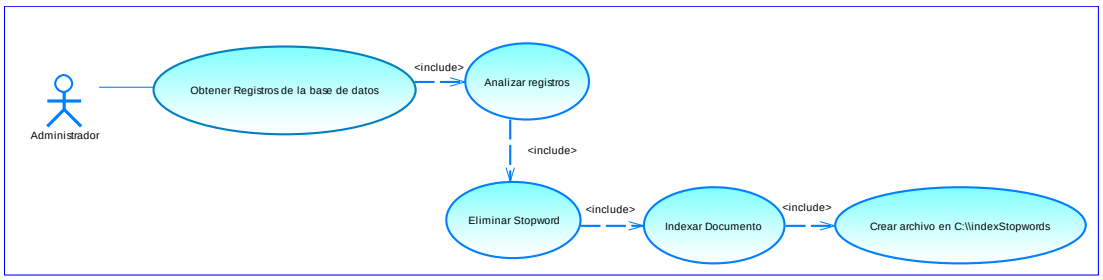

Figura 25. Casos de Uso-Módulo Lucene Indexación sin stopwords

3.8.2.3 Especificación de Casos de Uso del Módulo Lucene

Los casos de uso de las Figuras 24 y 25 han sido especificados en la siguiente tabla:

Tabla 32

Especificación del Caso de Uso: Crear índices e Indexar documentos

ID	RF-05
Descripción	Se crea los índices de búsqueda, para su posterior indexación de los documentos es decir de los registros almacenados y filtrados de la base de datos.
Precondición	Acceso a la base de datos.

Postcondición	Índices creados con documentos indexados.
Flujo Normal	1. El usuario administrador accede a los registros de la base de datos. 2. Se analizan los registros. 3. El indexador Lucene crea el índice. 4. Se indexan los documentos en el índice establecido (C:\\index)
Flujo Alterno	2.1. Se eliminan las stopwords de los registros de la base de datos. 3.1. El indexador Lucene crea el índice (C:\\indexStopwords). 4.1. Se indexan los documentos en el índice establecido (C:\\indexStopwords)
Excepciones	
Notas	

3.8.3 Arquitectura

La figura 26, muestra la arquitectura de Lucene, la cual está compuesta principalmente por dos módulos; en este caso se explica el primero que se encarga del análisis gramatical del texto (descomposición en palabras), generación de índices y el almacenamiento de estos.

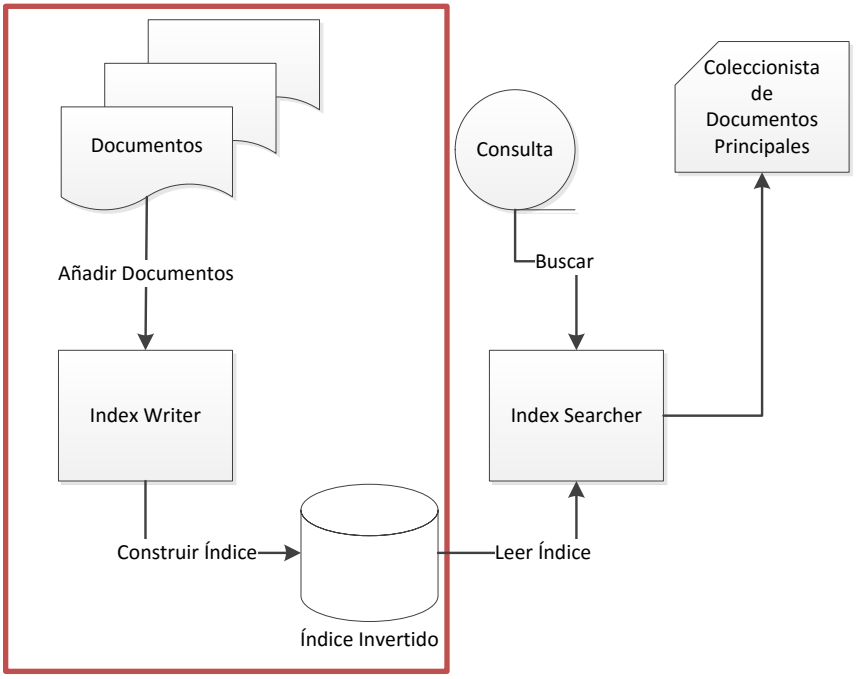

Figura 26. Arquitectura de Lucene

Fuente: (Coral Coral & Vaca Herrera, 2014)

3.8.4 Construcción y Pruebas

3.8.4.1 Backlog del Sprint

Una vez identificadas las historias de usuario, las cuales se encuentran descritas en los casos de uso y sus respectivas especificaciones, estás serán implementadas en el tercer sprint; para ello es necesario definir el Backlog del sprint es decir la pila de tareas que permitan implementar las funcionalidades especificadas en la etapa de análisis. La pila de tareas para el Sprint3 se muestra en la siguiente tabla:

Tabla 33

Backlog Sprint 3 "Módulo Lucene"

Story ID	Task #	Story Name, Task Name	Assigned 1	Estimado (horas)
1		Creación de índices		
	1	Codificación para extraer la información de la base de datos	Natalia Cerón	16
	2	Codificación para crear los índices	Natalia Cerón	4
	3	Pruebas de desarrollo	Mireya Chillán	2
	4	Pruebas unitarias	Mireya Chillán	1
2		Indexación de documentos	Mireya Chillan	4
	1	Codificación para la indexación de documentos	Mireya Chillan	4
		Almacenamiento de documentos indexados en los índices respectivos		
	2	Pruebas de desarrollo	Natalia Cerón	2
	3	Pruebas unitarias	Natalia Cerón	1

Continúa

3.8.4.2 Pruebas

Las pruebas que se realizaron dependieron del módulo y componentes de los mismos. Los resultados de las pruebas son adjuntados como anexos al presente documento.

3.9 SPRINT 4 "MÓDULO WEB SERVICES"

El cuarto sprint tiene como objetivo implementar las funcionalidades requeridas para la búsqueda y envío de información solicitada mediante la utilización de webservices para comunicarse con el sistema web AQUINO.

3.9.1 Planificación

De los Requerimientos Funcionales y No Funcionales – Sprint0 se selecciona las funcionalidades correspondientes al módulo que serán el objetivo del Sprint 4.

Tabla 34

Requerimiento Funcional 5: Envió de información solicitada

Id. Requerimiento	RE05 Envío de información solicitada
Descripción	El indexador Lucene busca la información solicitada por el Sistema AQUINO, y envía para su posterior proceso.
Entradas	Palabra clave para la búsqueda.
Salidas	Información solicitada por el Sistema AQUINO.
Proceso	La palabra clave es usada para ejecutar el RE02 y luego devolver la información filtrada y solicitada por el Sistema AQUINO para su posterior proceso.
Precondiciones	Comunicación con el Sistema AQUINO por medio del Web Service.
Postcondiciones	Retorno de la información solicitada.
Efectos Colaterales	Ninguno
Prioridad	Alta
Rol que lo ejecuta	Automático

Continúa

De acuerdo a las funcionalidades identificadas para el módulo del Web Services, se identifican las historias de usuario para el Sprint 4.

Tabla 35

Historia de Usuario-Sprint 4

ID	Historia de usuario	Importancia Propietario del Producto	Importancia Técnica	Descripción
1	Buscar en los índices respectivos según el criterio de búsqueda	100		Consiste en buscar según los criterios de búsqueda en los índices generados en el módulo Lucene.
2	Retornar la información solicitada	100	100	Consiste en retornar la información para que sea mostrada por el sistema web AQUINO al usuario.

Nota: La importancia está estimada de acuerdo a las necesidades del Propietario del Producto y está cuantificada con números enteros entre 0 y 100. La importancia técnica está basada en las necesidades no funcionales desde el punto de vista del usuario, pero necesarias para un funcionamiento de la aplicación desde el punto de vista técnico.

3.9.1.1 Definición del Equipo de Trabajo

El equipo de trabajo para la implementación de las funcionalidades del módulo del Web Services esta descrito en la siguiente tabla:

Tabla 36

Equipo de Trabajo-Sprint 4

ROL		Persona	Descripción/ Tareas
Product Owner		Ing. Ernesto Nieto	Administra el proyecto desde la perspectiva del negocio.
SCRUM Master		Dr. Felipe Mota	Asegurar que el proceso SCRUM se lleve a cabo.
Team	Codificación	Mireya Chillan Natalia Cerón	Codificación de las funcionalidades identificadas
	Pruebas	Ramiro Andrade	Pruebas

3.9.2 Análisis

Una vez identificadas las historias de usuario, se identifican los actores y se diagrama los casos de uso identificados para la implementación del Sprint 4.

3.9.2.1 Actores del Sistema

La siguiente tabla describe los actores que participan en los casos de uso identificados para el módulo Web Services.

Tabla 37

Actores del Sistema-Sprint 4

Actor	Descripción
AQUINO	El sistema AQUINO envía el parámetro de búsqueda.
Sistema de Búsqueda e Indexación	El sistema realiza la búsqueda según parámetros definidos y retorna la información solicitada.

3.9.2.2 Diagramas de Casos de Uso del Módulo Web Services

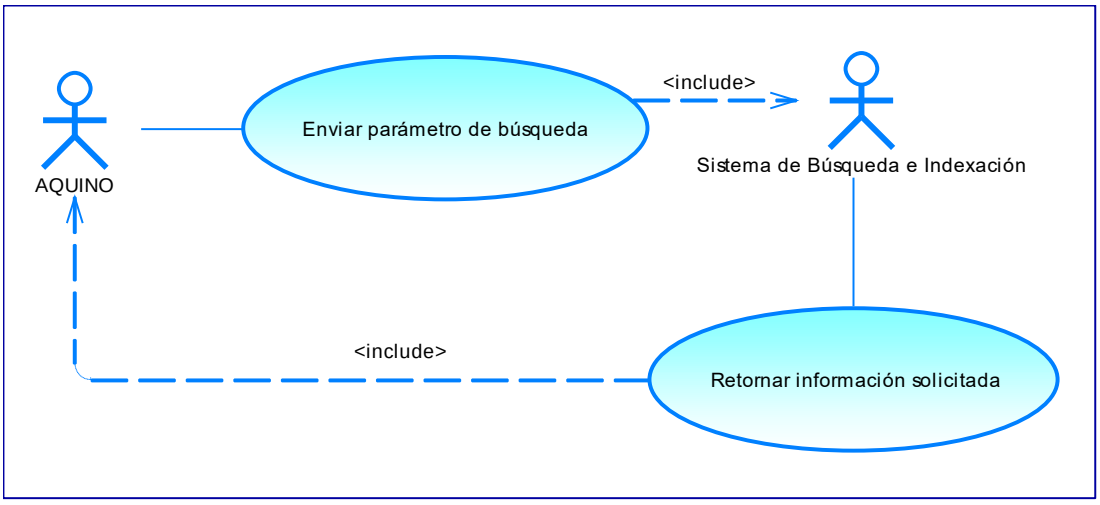

Figura 27. Casos de Uso-Módulo Web Services

3.9.2.3 Especificación de Casos de Uso del Módulo Web Services

Tabla 38

Especificación del Caso de Uso: Enviar información solicitada

ID	RF-06
Descripción	El indexador Lucene busca la información solicitada por el Sistema AQUINO, y envía para su posterior proceso.
Precondición	Comunicación con el Sistema AQUINO por medio del Web Service (palabra clave de búsqueda).
Postcondición	Retorno de la información solicitada.
Flujo Normal	1. El sistema AQUINO envía la palabra de búsqueda. 2. El sistema de búsqueda e indexación realiza la búsqueda respectiva. 3. El sistema de búsqueda e indexación retorna la información solicitada. 4. AQUINO muestra los resultados en su sistema web.
Flujo Alterno	3.1. El sistema de búsqueda e indexación no retorna resultados.
Excepciones	

3.9.3 Arquitectura

La figura 27, muestra la arquitectura de Lucene, la cual está compuesta principalmente por dos módulos; en este caso se explica el segundo que se encarga de ejecutar las búsquedas y retornar los resultados de acuerdo a los términos ingresados.

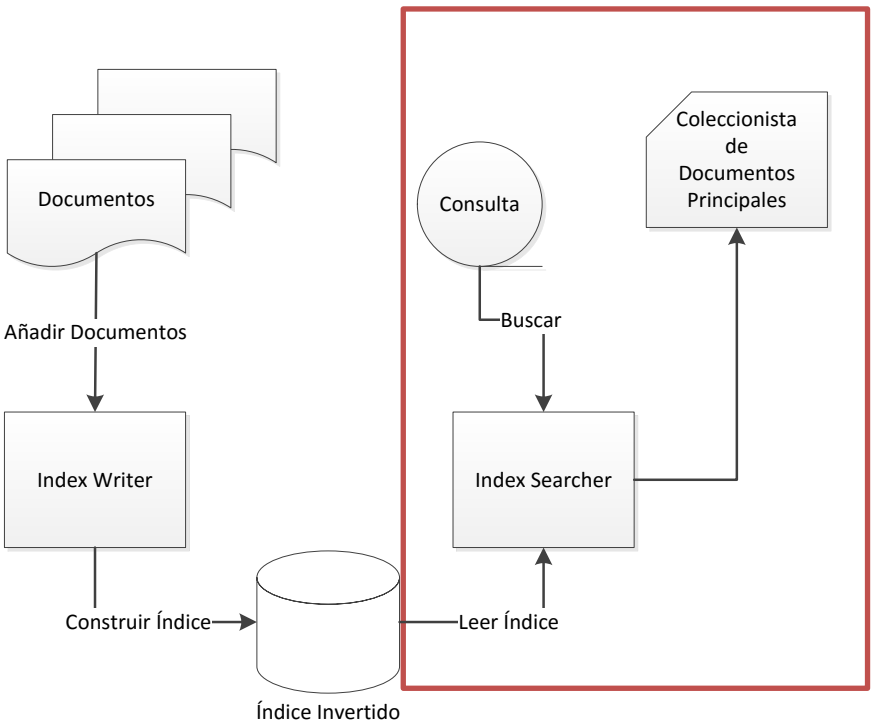

Figura 28. Arquitectura Lucene-Módulo de búsqueda

3.9.4 Construcción y Pruebas

3.9.4.1 Backlog del Sprint

Una vez identificadas las historias de usuario, las cuales se encuentran descritas en los casos de uso y sus respectivas especificaciones, estás serán implementadas en el cuarto sprint; para ello es necesario definir el Backlog del sprint es decir la pila de tareas que permitan implementar las funcionalidades especificadas en la etapa de análisis. La pila de tareas para el Sprint4 se muestra en la siguiente tabla:

Tabla 39

Backlog Sprint 4 "Módulo Web Services"

Story ID	Task #	Story Name, Task Name	Assigned 1	Estimado (horas)
1		Desarrollo del Módulo Lucene Búsqueda		
	1	Codificación para buscar en los documentos indexados.	Mireya Chillan	16
	2	Pruebas de desarrollo	Natalia Cerón	2

	3	Pruebas unitarias	Natalia Cerón	1
2		Desarrollo del Web Services	Mireya Chillan	4
	1	Codificación del Web Services	Mireya Chillan Natalia Cerón	4
	2	Pruebas de desarrollo	Ramiro Andrade	2
	3	Pruebas unitarias	Ramiro Andrade	1

3.9.4.2 Pruebas

Las pruebas que se realizaron dependieron del módulo y componentes de los mismos. Los resultados de las pruebas son adjuntados como anexos al presente documento.

CAPÍTULO 4
CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Las páginas web en su mayoría tienen una estructura jerárquica, por lo tanto para la implementación del web crawler se utilizó la búsqueda en profundidad obteniendo de esta manera una exploración y extracción de su contenido; a la vez de archivos PDF relacionados con los programas de Ciencias de la Computación.

- La implementación de Apache Tika permitió la lectura y obtención de texto de los archivos PDF descargados por el web crawler, facilitando la indexación de documentos y el almacenamiento en la base de datos para su posterior búsqueda.

- Al implementar el repositorio de datos se optimizó el proceso de búsqueda en el sistema, ya que se cuenta con una base offline que trabaja conjuntamente con el indexador Lucene para retornar resultados eficazmente.

- Terminado el desarrollo del Sistema de Búsqueda e Indexación utilizando las librerías Apache Lucene y Tika y su integración con el sistema web AQUINO se ejecutaron pruebas en las que se obtuvieron buenos resultados en tiempo de búsqueda, menores a 2 segundos utilizando un conjunto de 1000 archivos aproximadamente.

- El Web Services esta implementado en un estilo de arquitectura REST, ya que este se recomienda para sistemas que utilicen directamente HTTP para obtener datos o indicar la ejecución de operaciones sobre los datos.

RECOMENDACIONES

- Por la rapidez con la que se realiza la indexación y búsqueda, es factible implementar el sistema para todas las Carreras del listado de Universidades que tiene convenio con la Senescyt; mejorando la aplicación de Becas por parte de los estudiantes del Ecuador.

- Para la implementación del Sistema de Búsqueda e Indexación con Lucene y Tika no se debe adquirir otro equipo dedicado a este propósito, ya que estará integrado en el mismo sistema y así facilitará su manejo.

- Se debe tener en cuenta las nuevas versiones de Apache Lucene y Tika para que la aplicación del Sistema de Búsqueda e Indexación no quede obsoleto, esté actualizado y se puedan desarrollar nuevas funcionalidades.

- Los servicios del Web Services deben ser implementados en arquitectura REST por su principio fundamental se basa en separar el API en recursos lógicos que son manipulados usando peticiones HTTP.

REFERENCIAS BIBLIOGRÁFICAS

2.0, W. (18 de Febrero de 2014). *Fase de planificación del proyecto, KLC y metodología SCRUM*. Recuperado el 31 de Octubre de 2015, de http://inteligenciadenegociosval.blogspot.com/2014/02/fase-de-planificacion-del-proyecto-klc.html

Ávila, L. M. (02 de 10 de 2012). *Mongodb.* Recuperado el 12 de 10 de 2015, de http://www.buenastareas.com/ensayos/Mongodb/5618608.html

Balarezo, J. (Septiembre de 2014). *Análisis de factibilidad y selección de un framework de búsqueda global para su implementación en el sistema gestor fiducia fondos JEE de la Empresa Gestorincsa S.A.* Recuperado el 2015 de Octubre de 2015, de http://repositorio.espe.edu.ec/bitstream/21000/8968/1/T-ESPE-048198.pdf

Becas SENESCYT. (s.f.). *Areas del Conocimiento*. Obtenido de http://programasbecas.educacionsuperior.gob.ec/areas-del-conocimiento-3/

Camargo Sarmiento, F. I. (29 de Noviembre de 2013). *Evolución y tendencias actuales de los Web crawlers.* Obtenido de Dialnet-EvolucionYTendenciasActualesDeLosWebCrawlers-4797426.pdf

Coral Coral, H., & Vaca Herrera, A. (Septiembre de 2014). *Extracción de Información de Documentos Binarios Almacenados en Repositorios de Sistemas Transaccionales.*

Del Coso Santos, A. (Octubre de 2009). *Desarrollo de Infraestruturas para el modelado de usuarios.* Obtenido de http://e-archivo.uc3m.es/bitstream/handle/10016/8544/PFC_Ana_Coso_Santos.pdf?sequence=1

Graterol, Y. (s.f.). *MongoDB en español.* Recuperado el 12 de 10 de 2015, de http://collection.openlibra.com.s3.amazonaws.com/pdf/MongoDB-El-principio.pdf?AWSAccessKeyId=AKIAIGY5Y2YOT7GYM5UQ&Signature=Cq543XCn2SB%2ByPhw9Vk%2FRsptsJA%3D&Expires=1444666280

INACAP. (s.f.). *Manual Análisis de Algoritmos*. Recuperado el 20 de 06 de 2015, de http://colabora.inacap.cl/sedes/ssur/Asignatura%20Indtroduccion%20a%20la%20Programacn/An%C3%A1lisis%20de%20Algoritmo/Manual-Analisis%20de%20Algoritmos_v1.pdf

LC, J. (s.f.). *Archivo de la etiqueta: indexación*. Recuperado el 15 de Octubre de 2015, de http://jesuslc.com/tag/indexacion/

Mattmann, C. (2012). *Tika in Action.* NY: Manning Publications Co.

MongoDB. (s.f.). *MongoDB Documentation.* Recuperado el 15 de Octubre de 2015, de http://docs.mongodb.org/master/MongoDB-manual.pdf

Otis Gospodnetic. (2005). *Lucene in action.* Manning.

Ramos Hernández, J. P., & Alor Hernández, G. (2015). *Indización y Búsqueda a través de Lucene.* Recuperado el 18 de Octubre de 2015, de http://es.slideshare.net/lucreciamvc/3013167-indizacionybusquedaatravesdelucene

SENESCYT. (14 de Agosto de 2011). *Becas Secretaría de Educación Superior, Ciencia, Tecnología e Innovación.* Obtenido de http://programasbecas.educacionsuperior.gob.ec/requisitos-2014/

SENESCYT. (junio de 2014). *Areas de Estudio.* Obtenido de http://programasbecas.educacionsuperior.gob.ec/areas-de-estudio-10/

SENESCYT. (agosto de 2014). *Requisitos.* Obtenido de http://programasbecas.educacionsuperior.gob.ec/requisitos-2014/

SENESCYT. (s.f.). *Becas en el Exterior*. Obtenido de http://programasbecas.educacionsuperior.gob.ec/becas-en-el-exterior/

SENESCYT. (s.f.). *Becas Nacionales.* Obtenido de Programa Nacional de Becas: http://www.fomentoacademico.gob.ec/becas_iece/becas-nacionales#informaci%C3%B3n-adicional

SENESCYT. (s.f.). *Otros Programas*. Obtenido de http://programasbecas.educacionsuperior.gob.ec/otros-programas/

Toapanta Chancusi, K., Vergara Ordoñez, M., & Campaña Ortega, M. (s.f.). *Método Ágil SCRUM, aplicado a la implantación de un Sistema Informático para el proceso de recolección masiva de información con tecnología móvil*. Recuperado el 31 de Octubre de 2015, de http://repositorio.espe.edu.ec/bitstream/21000/5899/1/AC-SIS-ESPE-034427.pdf

Trupti V. , U., Ravindra D., K., & Rajesh C., D. (Febrero de 2014). *Study Of Web Crawler and its Different Types.* Obtenido de http://files.figshare.com/1717744/A016160105.pdf

Tutorialspoint. (s.f.). *Apache Tika*. Recuperado el 31 de Octubre de 2015, de http://www.tutorialspoint.com/tika/tika_overview.htm

Viñals, J. (s.f.). *Localización y generación de mapas del entorno (SLAM) de un robot por medio de una Kinect*. Recuperado el 20 de 06 de 2015, de https://riunet.upv.es/bitstream/handle/10251/17544/Memoria.pdf?sequence=1

Wikipedia. (s.f.). *Algoritmo de Búsqueda*. Recuperado el 20 de 06 de 2015, de https://es.wikipedia.org/wiki/Algoritmo_de_b%C3%BAsqueda

Anexo 1: Casos de Prueba del Sistema de Búsqueda e Indexación

Noviembre 2015

Introducción

Un aspecto crucial en desarrollo de software son las pruebas y, dentro de estas, las pruebas funcionales, en las cuales se hace una verificación dinámica del comportamiento de un sistema, basada en la observación de un conjunto seleccionado de ejecuciones controladas o casos de prueba.

Para hacer pruebas funcionales se requiere una planificación, la cual consiste en definir los aspectos a revisar y la forma de verificar su correcto funcionamiento.

En este documento se describen los casos de prueba funcional de los Módulos Web Crawler, Tika, Lucene, Web Services que conforman el sistema de búsqueda e indexación.

1. CASO DE PRUEBA MÓDULO WEB CRAWLER

Caso de Prueba 01	
Código de Identificación:	Sistema.Busq.Index 01
Nombre Caso de Prueba:	Funcionamiento Web Crawler
Descripción	El administrador podrá mirar el registro del contenido de las páginas web almacenados en la base de datos con el gestor de MongoDB llamado MongoChef y a la vez comprobar la descarga de los archivos en el directorio local C:\\guardarPDF
Variables de Entrada (Inputs):	direccionesPostgrados
Flujo normal del evento	1. Lee el arreglo de la variable direccionesPostgrado. 2. Examina el contenido de cada página web con sus respectivos links enlazados. 3. Extrae el contenido y lo guarda en la base de datos.
Resultado esperado:	Se visualiza los registros en la base de datos.
Flujo alterno	El link pertenece a un archivo PDF 1. El sistema guarda el archivo en la carpeta local del computador.
Resultado alternativo esperado:	Archivos PDF descargados en el directorio C:\\guardarPDF

2. CASO DE PRUEBA MÓDULO TIKA

Caso de Prueba 02	
Código de Identificación:	Sistema.Busq.Index 02
Nombre Caso de Prueba	Lectura de archivos PDF y almacenamiento de su texto plano.
Descripción	Se determina la ruta de los archivos descargados para su posterior lectura y almacenamiento de su texto plano en la base de datos.
Precondiciones	Caso de prueba Sistema.Busq.Index 01 flujo alterno
Flujo normal del evento	1. Apertura del directorio C:\\guardarPDF 2. Lectura de los archivos PDF 3. Análisis de los archivos PDF 4. Extracción del texto plano de los archivos PDF 5. Almacenamiento del contenido en la base de datos.
Resultado esperado	Registros almacenados en la base de datos
Comentarios	Sin comentarios

3. CASO DE PRUEBA MÓDULO LUCENE

Caso de Prueba 03	
Código de Identificación:	Sistema.Busq.Index 03
Nombre Caso de Prueba:	Indexación de documentos
Descripción	El sistema recupera los registros de la base de datos y los analiza para indexarlos.
Variables de Entrada (Inputs):	Direcciones
Flujo normal del evento	1. Recuperación de registros de la base de datos 2. Análisis del contenido 3. Indexación de documentos sin Stop Word 4. Indexación de documentos con Stop Word 5. Índices se guarda localmente en el directorio C:\\index e C:\\indexStopWord
Resultado esperado:	Directorios con archivos .cfs que contiene los índices
Flujo alterno	Indexar los registros de la base de datos que contienen el texto de los archivos PDF
Resultado alternativo esperado:	Directorio C:\\indexPDF con archivos .cfs que contiene los índices.

4. CASO DE PRUEBA MÓDULO WEB SERVICES

Caso de Prueba 04	
Código de Identificación:	Sistema.Busq.Index 04
Nombre Caso de Prueba	Web Services
Descripción	El web services retorna la información que el Sistema AQUINO complementario solicita para mostrar en su Front End.
Precondiciones	Servicio Rest levantado
Flujo normal del evento	1. El sistema AQUINO envía la palabra de búsqueda. 2. El sistema de búsqueda e indexación realiza la búsqueda respectiva en los documentos indexados (.cfs). 3. El sistema de búsqueda e indexación retorna la información solicitada a través del webservices. 4. AQUINO muestra los resultados en su sistema web.
Resultado esperado	El sistema web AQUINO despliega los resultados encontrados.
Comentarios	

5. RESULTADOS

- **Tiempo de Búsqueda de resultados**

Terminado el desarrollo del Sistema de Búsqueda e Indexación utilizando las librerías Apache Lucene y Tika y su integración con el sistema web AQUINO se ejecutaron pruebas en las que se obtuvieron buenos resultados en tiempo de búsqueda, aproximadamente 2 segundos utilizando un conjunto de alrededor 1000 archivos.

A continuación se presentan los resultados de la prueba realizada:

- Búsqueda manual por parte del usuario

Universidad	Tiempo de Búsqueda(min)	Carrera Encontrada
Unc	6	Pregrado Analista de Sistema e Informática
Uniandes	13	Postgrado Maestría en Ingeniería de Software
		Postgrado en Seguridad de la Información
TOTAL TIEMPO DE PRUEBA	**19**	
Promedio	10 min x pag	
Total links universidades con convenio con la SENESCYT	200	
TOTAL TIEMPO DE BÚSQUEDA	**2000 min=44 horas**	

- Búsqueda utilizando el sistema AQUINO

Universidad	Tiempo de Búsqueda(seg)	Carrera Encontrada
Unc	2	Pregrado Analista de Sistema e Informática
Uniandes	6	Postgrado Maestría en Ingeniería de Software
		Postgrado en Seguridad de la Información
TOTAL TIEMPO DE PRUEBA	**8**	
Promedio	4 seg x pag	
Total links universidades filtradas con webcrawler	48	
TOTAL TIEMPO DE BÚSQUEDA	**192 seg=4 horas**