UNIVERSIDAD DEL BÍO – BÍO

FACULTAD DE CIENCIAS EMPRESARIALES

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y TECNOLOGÍAS DE LA INFORMACIÓN

INGENIERÍA CIVIL EN INFORMÁTICA

Prototipo de una aplicación de apoyo a las Revisiones Sistemáticas de la Literatura

28 de mayo de 2016

Chillán – Chile

Alumno: Mauricio Moisés Sepúlveda Venegas

Profesor guía: Dra. María Angélica Caro Gutiérrez

Memoria para optar al título de Ingeniero Civil en Informática

Resumen

Este proyecto se presenta para dar conformidad a los requisitos exigidos por la Universidad del Bío-Bío en el proceso de titulación para la carrera de Ingeniería Civil en Informática. El proyecto se ha titulado "Prototipo de una aplicación de apoyo a las Revisiones Sistemáticas de la Literatura".

La cantidad de información que se puede encontrar en la web es cuantiosa. Cuando se realiza una búsqueda a través de internet, referida a artículos científicos, los resultados obtenidos no son vistos en su totalidad por el tiempo que significa su revisión. El propósito de este proyecto es generar un prototipo de software para ayudar al proceso de Revisión Sistemática de la Literatura, la principal característica es la obtención de resultados desde "Google Académico" y posterior almacenamiento. El proyecto se basa en una investigación previa de tecnologías y herramientas para el procesamiento de información dinámica contenida y generada a través de páginas web.

Abstract

This project appears to provide conformity to the requirements of the University of the Bío-Bío in the process of qualification for Civil Engineering Computer. The project entitled "Prototype of an application to support Systematic Review of Literature".

The amount of information that can be found on the Web is large. When a search is performed over the Internet, based on scientific articles, the results are not seen in full by the time it means review. The purpose of this project is to generate a software prototype to help the process of Systematic Literature Review, the main feature is the outcome from "Google Scholar" and subsequent storage. The project builds on previous research of technologies and tools for processing information from web pages.

Agradecimientos

Este fue un proyecto que requirió mucho esfuerzo y perseverancia para sacarlo adelante. Por ello quiero agradecer a Dios, a mi familia, compañeros y amigos, sin ellos este trabajo no hubiese sido posible.

Ellos me dieron el empuje y el aliento necesario para llevarlo a cabo, para seguir adelante enfrentando y superando las dificultades presentes en el camino, sé que este es el término de una etapa de mi vida que me da las herramientas para enfrentar la siguiente.

A todos ustedes muchas gracias.

Índice General

Índice Tablas

Índice Figuras

1 INTRODUCCIÓN

Una Revisión Sistemática de la Literatura (RSL) es una manera de evaluar e interpretar toda la investigación disponible, que sea relevante respecto de una interrogante de investigación particular, esta investigación disponible, está representada por estudios primarios que son buscados principalmente en internet en sitios dedicados, la RSL tiene diversas etapas, la primera esta es la planificación de la revisión, que contiene la etapa de definición de un protocolo de revisión, esta etapa da como resultado los términos claves a buscar y en que sitios de internet serán buscados esos términos junto con las actividades de la revisión disponible. La segunda Etapa es el Desarrollo de la revisión donde se realiza la búsqueda y se selecciona los estudios primarios, esta actividad se realiza de manera manual, seleccionando los artículos pertinentes a nuestra investigación y documentando nuestra apreciación de la literatura seleccionada y no seleccionada. La tercera etapa es la publicación de resultados a la comunidad científica.

Como primera parte, presenta una investigación sobre las tecnologías necesarias para extraer información en internet, abarca algunas herramientas en los distintos lenguajes de programación, para extraer información de los resultados entregados por Google Académico, extracción de información en documentos y páginas web, las fuentes de búsqueda son diversas. En este prototipo se ha elegido Google Académico, dado que presenta un amplio número de resultados y es transversal todas las áreas del conocimiento.

Como segunda parte, este proyecto ayuda en la problemática de revisión, selección y documentación en las etapas de búsqueda y selección, para ello se desarrollará un prototipo de sistema que realice las búsquedas por palabras claves obtenidas del protocolo de revisión, almacene las búsquedas, permita hacer una revisión preliminar de cada uno de los resultados, esta parte es desarrollada gracias a las herramientas obtenidas en la investigación.

Este documento se organiza de la siguiente manera: sección 1 de introducción, sección 2 Definición del problema (presenta la descripción del área de estudio y la problemática), sección 3 Definición del proyecto (se establece la metodología de trabajo, objetivos y conceptos relacionados), sección 4 Análisis de la tecnología disponible, sección 5 Desarrollo de prototipo, sección 6 Conclusiones y finalmente la sección 7 Bibliografía.

2 DEFINICIÓN DEL PROBLEMA

En esta sección se pone en contexto y se describe el área de estudio al cual apunta el desarrollo de esta memoria, da a conocer la problemática existente y donde se pretende intervenir para lograr una solución.

2.1 Descripción del área de estudio

Una Revisión Sistemática de la Literatura (RSL) es una manera de evaluar e interpretar toda la investigación disponible, que sea relevante respecto de una interrogante de investigación particular, en un área temática o fenómeno de interés (Kitchenham, 2004). Una RSL es una metodología que consta de varias etapas (ver Tabla 2.1), las cuales estructuran el proceso.

Tabla 2.1 Etapas de una Revisión Sistemática de la Literatura

Etapa 1 Planificar la Revisión
Identificación de la necesidad de revisión
Definición de un protocolo de revisión
Etapa 2 Desarrollo de la Revisión
Búsqueda de estudios primarios
Selección de estudios primarios
Etapa 3 Publicación de resultados

El principio de una RSL presenta tres etapas, la primera etapa *planificar la revisión*, esta etapa considera dos sub-etapas, la primera se *Identifica la necesidad de revisión* en la cual se determinan interrogantes de investigación, el objetivo de la revisión y donde se hará. Un aspecto importante de esta etapa es la definición de las palabras claves, estas palabras son combinadas de tal manera que generan frases que serán ingresadas al motor de búsqueda seleccionado según el protocolo de búsqueda, estas por ejemplo pueden ser “Data Quality” o “Calidad de Datos”, el desarrollo se realiza preferentemente a través de internet en las diversas fuentes que entregan información científica, en internet tenemos a “Google Académico”, “Scielo”, “Science Direct”, “IEEE”; y en revistas impresas.

La segunda sub etapa de planificación de la revisión, es *definir el protocolo de revisión*, este define principalmente como abordar los resultados obtenidos y hacer una revisión; el protocolo de revisión define donde debemos enfocarnos en el análisis de un artículo. Por ejemplo, la norma de revisión suele estar enfocada en el resumen, introducción, conclusión o palabras claves, para determinar si la información sirve a la investigación. Generalmente se lee el resumen y conclusión de los artículos, si el investigador descubre en su lectura que el artículo es pertinente para su investigación, lo lee completo, sino, es descartado, se deben definir registros de los resultados y los criterios de selección.

La segunda etapa es el *desarrollo de la revisión*, es aquí donde el trabajo es mayor, se debe buscar estudios primarios sobre las fuentes de información definidas en la etapa anterior en internet o en revistas científicas impresas, aplicando las palabras definidas y registrando resultados y evaluando de acuerdo con los criterios de selección del material científico. Por cada artículo pertinente a la investigación se debe extraer la información importante que contribuya a nuestro trabajo.

La tercera etapa y última de una RSL es dar a conocer el resultado de nuestra investigación como artículos de conferencias y revistas científicas, una RSL es un trabajo que requiere esfuerzo y debe ser realizado con la mayor prolijidad posible para validar los resultados obtenidos. Entonces la RSL consta de varias etapas en las cuales se desarrollan diversas actividades para hacer de la búsqueda en la literatura, un proceso estructurado.

2.2 Descripción de la problemática

En los últimos años, con la aparición de Internet, la búsqueda de la literatura en la etapa de Revisión, se realiza en gran medida, utilizando internet, en sitios dedicados a esta tarea como lo son "Google Académico", "Elsevier", "IEEE", entre otros. Sin embargo, al realizar una consulta en estos sitios dedicados, suelen arrojar gran cantidad de artículos científicos como resultados. Asimismo, estos resultados suelen contener material irrelevante, que solo aumenta el tiempo que se invierte en la revisión.

En el párrafo anterior aparecen diversas fuentes de búsqueda, algunas de ellas dedicadas a una sola área de estudio, se ha elegido para el desarrollo del prototipo Google Académico ya que presenta un amplio número de resultados y contiene información transversal de todas las áreas de estudio.

Si comenzamos el proceso de búsqueda, con la frase "Calidad de datos" a través del motor de búsqueda "Google Académico", obtendremos la Figura 2.1 donde se refleja la estructura que posee este buscador y las configuraciones que podemos realizar, con respecto a los resultados, estos resultados son ordenados de acuerdo a criterios del buscador. Realizar una búsqueda de acuerdo a intervalos específicos de fechas, filtrar por resultados en español, descartar patentes, citas y nos muestra la cantidad de resultados obtenidos, en este caso 1.440.000 resultados fueron los que el motor de búsqueda seleccionó.

Figura 2.1 Búsqueda en "Google Académico"

En los resultados se muestra el título con un rótulo, el cual especifica si es libro, archivo PDF, cita o patente. Luego aporta el autor, página web y año de publicación (solo en algunos), nos muestra un extracto del archivo, libro o página web donde se encuentran los resultados buscados, e información importante respecto al número de citaciones, *artículos relacionados*, las versiones del mismo artículo encontradas en la web, *citar* para obtener un texto con las referencias y si es necesario guardar, como requisito se debe tener una cuenta de correo electrónico "Google" para poder realizar esta acción. Entonces debemos ingresar a cada uno de los ítems del resultado para poder leer el artículo de acuerdo al protocolo de revisión, esta tarea es muy ardua ya que la cantidad de resultados es enorme, lo que significa que el

investigador realizará una depuración manual de las primeras cuatro o cinco hojas de resultado.

Para tener una aproximación a la cantidad real de resultados de "Google Académico" el motor entrega por cada hoja 10 resultados y para búsquedas que dan gran cantidad de información solo entrega como máximo 100 hojas de resultados, entonces la cantidad de resultados que entrega el motor como máximo son 1.000, cifra que está muy por debajo de los 1.440.000 resultados aproximados que manifiesta el buscador, cuando las búsquedas son acotadas a palabras estrictas la cantidad total de resultados desciende a 2.030 resultados reflejado en la Figura 2.2.

Figura 2.2 Búsqueda de la palabra "Calidad de datos"

En la Figura 2.3 se observa el fin de las hojas de resultado de la búsqueda estricta sobre "Calidad de datos", entonces si tenemos 95 hojas completa con resultados, la hoja 96 solo con 4 resultados suma un total de 954 resultados y no los 1.940 que da a conocer el motor.

Figura 2.3 Fin de los resultados de la búsqueda

En la Figura 2.4 vemos la hoja de resultados 97 vacía y esto se repite hasta la página 100, sin resultados disponibles. Esto significa que la cantidad de resultados real disponibles al usuario cuando consulta por todos, es diferente a la que se proporciona en un principio.

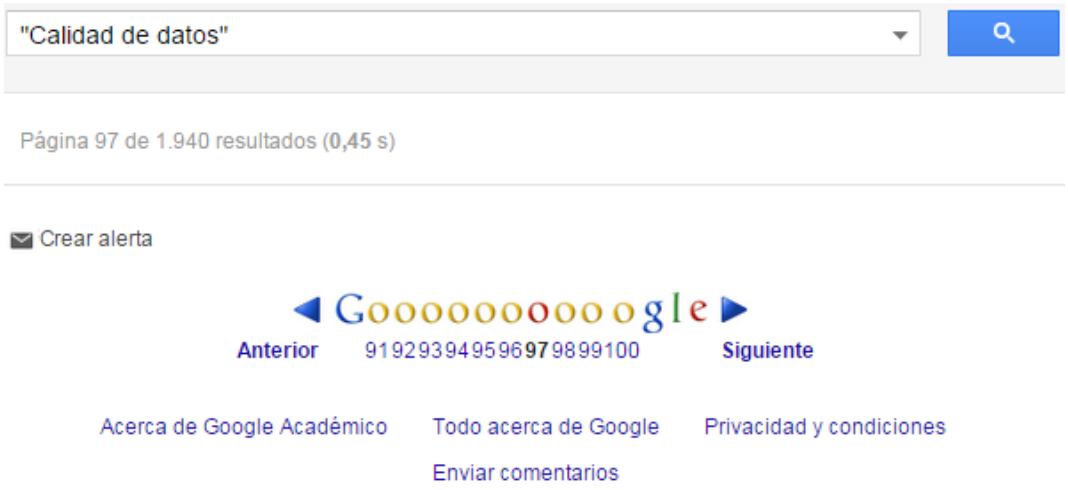

Figura 2.4 Hoja sin resultados de búsqueda

De la revisión de resultados entregados por "Google Académico" solo están disponibles como máximo 1000 resultados por búsqueda y no es necesariamente el número que entrega de referencia cuando se realiza la búsqueda. Si bien "Google Académico" no es claro en la entrega de resultados porque dice una cantidad de resultados que no corresponde a la realidad, solo entrega como máximo 1000 resultados cuando hay variada información de los términos buscados, cuando la palabra es estricta la cantidad disminuye y no entrega como máximo 1000 resultados sino que menos, es un defecto encontrado en el motor de búsqueda a la hora de buscar información científica, pero sigue siendo una herramienta útil a la hora de buscar literatura científica. Se evaluó este buscador, debido a que es el seleccionado como base para obtener artículos científicos por la cantidad y diversidad de los resultados de distintas páginas de internet.

3 DEFINICIÓN PROYECTO

En esta sección se logran establecer los objetivos del proyecto dando a conocer el plan de acción para enfrentar el proyecto y el proceso de investigación llevado a cabo.

3.1 Objetivos del proyecto

3.1.1 Objetivo general

- Desarrollar un prototipo de aplicación para apoyar el desarrollo de Revisiones Sistemáticas de la Literatura.

3.1.2 Objetivos específicos

- Identificar un método de extracción de datos de los resultados en las búsquedas con Google Scholar con el fin de ser procesados.

- Construir un prototipo de aplicación que muestre los resultados de la búsqueda depurados y pre-seleccionados. Este prototipo deberá considerar: Interfaz de usuario, conexión con un motor de búsqueda, almacenamiento de datos y estadísticas de las búsquedas.

3.2 Metodología de Trabajo

El desarrollo de este prototipo requiere de una investigación sobre herramientas para el procesamiento de página web. La investigación se realiza abordando una revisión de métodos e instrumentos, para facilitar el desarrollo del prototipo, específicamente, técnicas de extracción de información, a través de los lenguajes de programación. Esta investigación es realizada utilizando la metodología Investigación-Acción propuesta por Kurt Lewin (1951) para el área de las ciencias sociales (Myers & Avison, 2002). Por las características que tiene este método se ha llevado a la ingeniería de software (Acosta, López, & Espinoza, 2011).

La metodología Investigación-acción (IA) se clasifica como un método cualitativo y empírico, su propósito se basa principalmente en la prueba de metodologías, métodos y técnicas para

observar qué es lo que ocurre. El proceso de la IA consiste en proceso cíclico, este proceso se aprecia en la Figura 3.1 en la cual se desarrolla en cuatro etapas planificación, acción, observación y reflexión (Latorre, 2003).

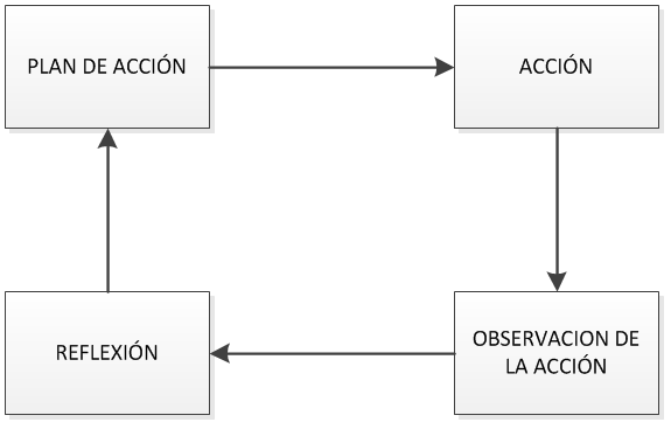

Figura 3.1 Ciclo de la Investigación Acción

En el plan de acción se ven los objetivos de la investigación, se definen los integrantes que participaran del proceso y el grupo crítico de referencia en la que se hará observación y posterior reflexión de los resultados obtenidos. En la acción se desarrollan las actividades necesarias para la investigación. En la observación de la acción, se ve el resultado de la actividad y como es llevada a cabo, en la última etapa de reflexión, se evalúa si es necesario tomar medidas correctivas o seguir en el plan original, además de definir si es necesario un nuevo ciclo o una nueva etapa de IA.

El desarrollo de este proyecto establece dos etapas o ciclos: la investigación de herramientas y el desarrollo del prototipo, la Figura 3.2 muestra el esquema del proceso desarrollado y las fases en cada etapa. En este caso el primer ciclo consiste en un plan de acción para la búsqueda de herramientas para la extracción de información de páginas web luego, en la etapa de acción esas herramientas se prueban, se observan los resultados obtenidos y se selecciona o descarta. El segundo ciclo está relacionado con el desarrollo del prototipo, con la ayuda de las herramientas seleccionadas en el primer ciclo, entonces se desarrolla la herramienta y se prueba su funcionamiento.

La primera etapa, presenta 4 fases para el desarrollo del proceso, desde la base de conocimientos se identificaron problemas relacionados en el desarrollo del proceso de RSL, específicamente, en la búsqueda y selección de artículos primarios en motores de búsqueda.

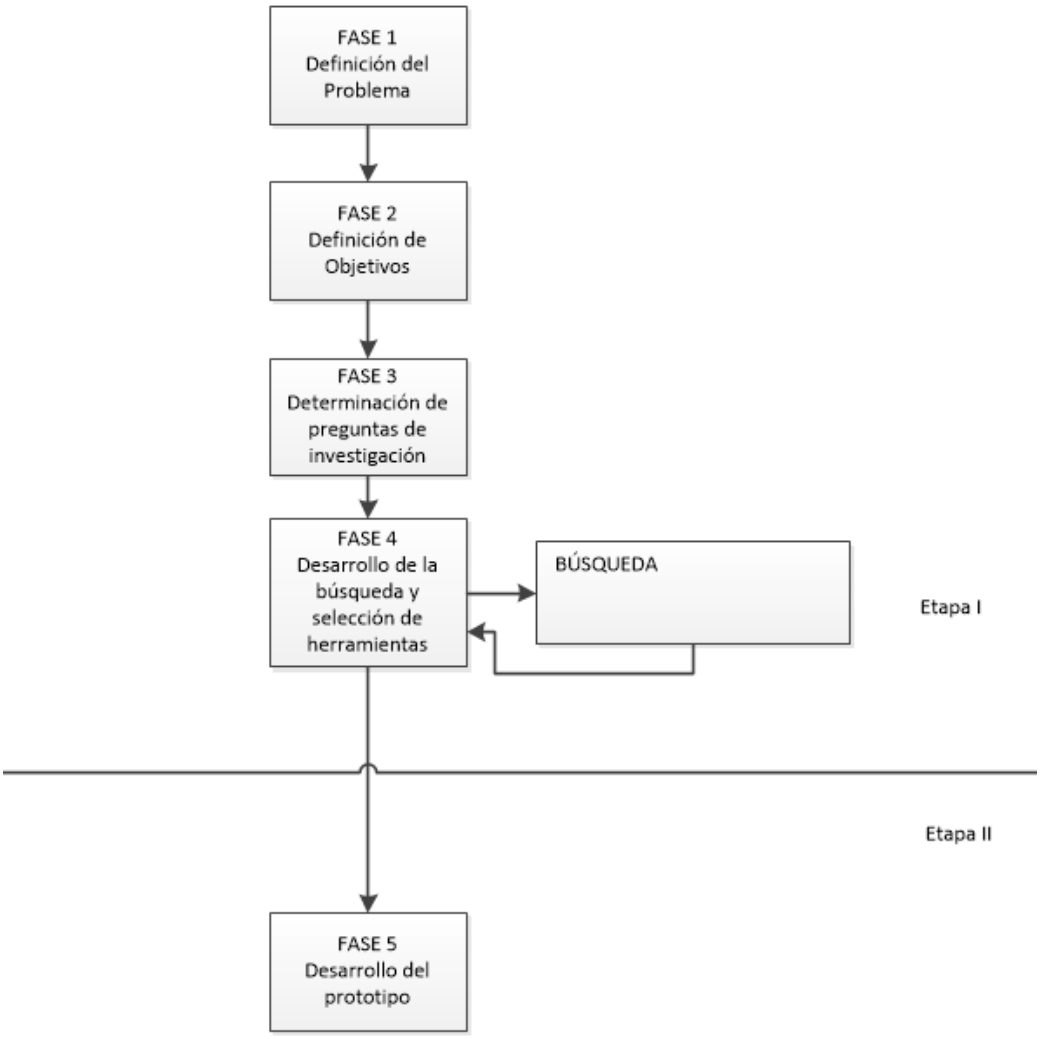

Figura 3.2 Esquema que representa el proceso de la investigación

En la fase 1 se define el problema relacionado con la gran cantidad de información disponible que entrega el motor de búsqueda, en la fase 2 se definen los objetivos relacionados a filtrar los resultados obtenidos de una búsqueda, en la fase 3 se definen las preguntas para la búsqueda de tecnología, metodologías y herramientas para abordar el problema de obtención, filtrado y presentación de resultados al investigador, para ello se hizo la selección del motor

de búsqueda con el cual se trabajará "Google Académico", el cual indexa gran variedad de artículos perteneciente a las revistas científicas y no se restringe a una sola revista o corporación.

En la Figura 3.3 se detalla el sub-proceso de búsqueda y selección de las herramientas. Esta búsqueda se realiza respecto de 3 tipos de herramientas, la primera sobre el procesamiento de páginas web, enfocadas a obtener información de la página, la segunda relativa a procesar documentos PDF para extraer información de los artículos científicos que se encuentren en este formato y la tercera la conexión de la aplicación con un motor de búsqueda, muy importantes, porque el motor de búsqueda proveerá los artículos científicos a analizar.

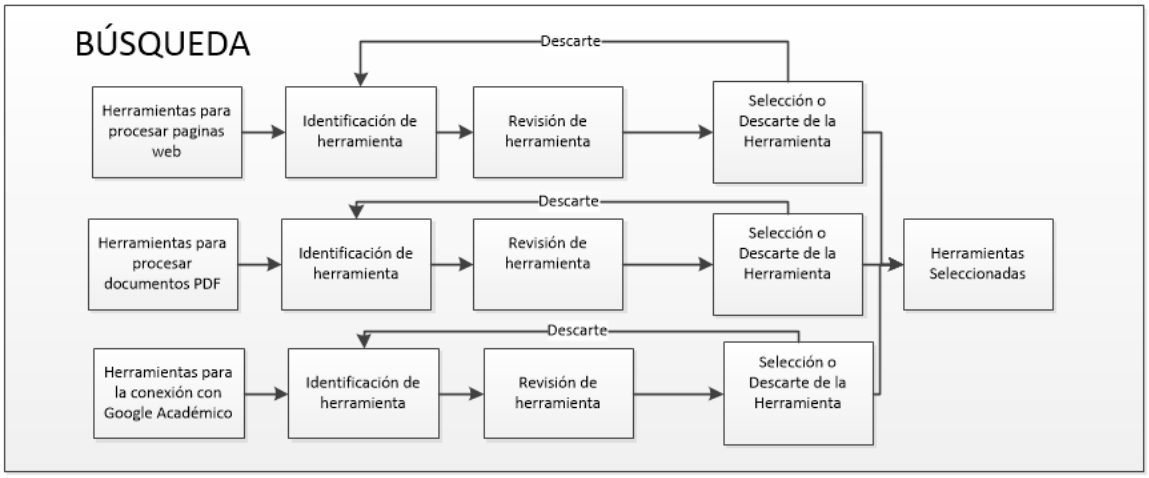

Figura 3.3 Esquema que detalla el proceso de búsqueda de herramientas

Entonces el sub-proceso de búsqueda consiste en que por cada tipo de herramienta encontrada, se hace una identificación, se desarrolla una prueba y se revisa su funcionamiento el cual genera un resultado, con ese resultado se selecciona o se descarta, una vez que se tienen todas las herramientas seleccionadas se verifica la compatibilidad para el desarrollo del prototipo, entonces el resultado de este sub proceso es la obtención de las herramientas compatibles que utiliza el prototipo de software.

3.2.1 Proceso de Investigación

El proceso de investigación acción hace necesario definir roles del proceso, los roles relativos a esta investigación se mantendrán hasta el final y serán definidos de la siguiente manera:

- Investigador: Alumno memorista
- Objetivo Investigador: Obtener metodologías, técnicas y/o herramientas para la extracción de información de páginas web y desarrollo de un prototipo.
- Grupo Crítico de Referencia (GCR): Profesor guía.
- Beneficiario: Individuos que desarrollen Revisiones Sistemáticas de la Literatura.

De acuerdo a la metodología de investigación seleccionada IA involucra actividades previas y dos grandes ciclos, el primero sobre la búsqueda, prueba y selección de las herramientas y el segundo el desarrollo del prototipo.

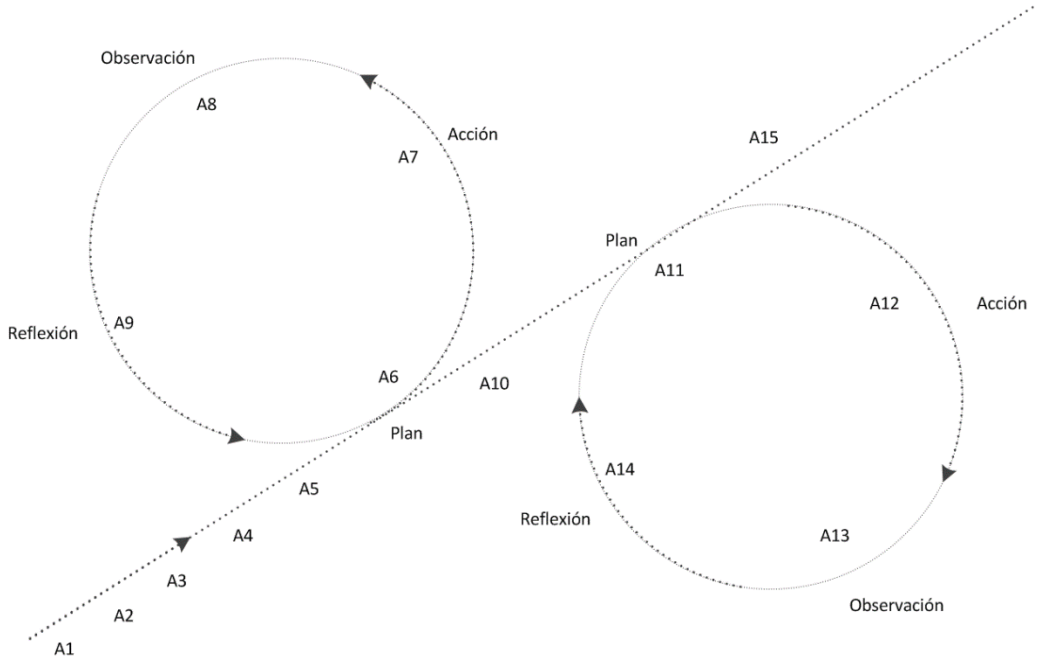

Figura 3.4 Ciclos de la Investigación Acción desarrollados

La Figura 3.4 muestra los hechos más importantes, en la primera etapa se logra identificar el problema y contextualizar la investigación, luego se desarrolla el prototipo de software, el detalle de cada una de las actividades y los participantes involucrados. Esto se refleja en la Tabla 3.1

Tabla 3.1 Investigación - Acción en la investigación de herramientas y desarrollo del prototipo

Aspecto	Actividad	Hecho	Roles	
			Grupo Critico de Referencia	Investigador
Contexto de investigación	A1	Identificación de los participantes	X	X
	A2	Establecimiento de roles		X
Problema	A3	Preparación de propuesta de investigación		X
	A4	Identificación del problema		X
Definición de objetivos	A5	Planeamiento de objetivos		X
	A6	Planeamiento de preguntas		X
Desarrollo de la búsqueda y selección	A7	Identificación de herramientas		X
	A8	Prueba de herramientas		X
	A9	Reflexión sobre la herramienta	X	X
	A10	Selección de todas las herramientas necesarias		X
Desarrollo del prototipo	A11	Identificación de requisitos previos	X	X
	A12	Aprender la tecnología		X
	A13	Desarrollo del prototipo de software		X
	A14	Reflexión y pruebas del prototipo		X
	A15	Presentación del prototipo	X	X

El detalle de la investigación se encuentra en los puntos siguientes, en los cuales se identifica todo lo necesario para el desarrollo de la investigación y posterior prototipo.

3.2.1.1 Identificación del problema

La gran cantidad de información que arroja una búsqueda de internet hace imposible una revisión de todos los resultados. Para poder revisar el contenido, se hace notar la necesidad de alguna herramienta que permita depurar los resultados y mostrar al usuario información clave para pre-evaluar el artículo científico.

La selección del motor de búsqueda para obtener los artículos fue "Google Académico", este buscador, no se centra solamente en artículos de una revista, como son los buscadores de "Science Direct" o "Scielo", sino que, además de incluir resultados de las revistas más

importantes, se incluye una diversidad de artículos que pudiesen ser relevantes para la investigación.

3.2.1.2 Preguntas de investigación

Las preguntas de investigación formuladas desde un inicio no fueron suficientes para obtener toda la información necesaria, lo que significó agregar preguntas posteriores al inicio de la investigación Tabla 3.1.

Tabla 3.2 Preguntas de Investigación Planteadas Inicialmente

Preguntas iniciales	PI1	¿Qué herramienta permite procesar información de páginas web?
	PI2	¿Google Académico provee una interfaz de comunicación para obtener el resultado de las búsquedas?

La búsqueda de las respuestas de las preguntas de la Tabla 3.1 generó nuevas preguntas con respecto a metodologías o técnicas más específicas, es ahí en donde aparecen nuevas preguntas de investigación las cuales se ven reflejadas en la Tabla 3.3.

Tabla 3.3 Preguntas de investigación obtenidas posteriormente

Preguntas posteriores	PI3	¿Qué herramientas permiten hacer "Scraping"[1]?
	PI4	¿Cómo procesar código HTML?
	PI5	¿Cómo realizar "Scraping" a Google Académico?
	PI6	¿Cómo procesar documentos PDF?
	PI8	¿Cómo hacer "Scraping" a documentos PDF?

Las preguntas iniciales fueron muy amplias, pero permitieron ir refinando la búsqueda he ir acotando los resultados con las preguntas posteriores. La respuesta a estas interrogantes es realizada a través del motor de búsqueda Google.

[1] Técnica en la cual un programa extrae la información de la página web.

3.3 Conceptos Relacionados

La metodología de desarrollo del software utilizado es por prototipo, esta metodología pertenece a los modelos evolutivos del desarrollo de software, es utilizado cuando los requisitos del software van cambiando de acuerdo a su avance en el desarrollo, se establece los requisitos generales del software, pero luego se van refinando hasta el desarrollo final (Pressman, 1997).

3.4 Definiciones, Siglas y Abreviaciones

La investigación arrojó como resultado diversos conceptos que son necesarios definir para poderla contextualizar, en la Tabla 3.4 se detallan los términos encontrados en la investigación.

Tabla 3.4 Definición de términos encontrados

Término	Definición
Scraping	Es una técnica en la cual un programa extrae la información de la página web (School of Data, n.d.).
Python	Python es un lenguaje de programación interpretado que permite trabajar de forma rápida e integrar los sistemas de manera más eficaz (Python ORG, n.d.)
Django	Django es un framework web de Python de alto nivel que fomenta el rápido desarrollo y el diseño limpio y pragmático (django project, 2005).
Framework	Marco de trabajo o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos. Esta ayuda a organizar y desarrollar software (Sánchez, 2006).
PDF	PDF es un formato de almacenamiento de documentos digitales independiente de plataformas de software o hardware (Adobe Systems Sofware, 1994)
HTML	Es el lenguaje de publicación de amplia red mundial (W3C, 2013)

4 ANÁLISIS DE LA TECNOLOGÍA DISPONIBLE

La tecnología encontrada en la investigación y las funciones que aborda cada una son plasmadas en esta sección, las cuales son agrupadas por procesamiento de páginas web, conexión con el motor de búsqueda y procesamiento de documento PDF. La Tabla 4.1 muestra las herramientas obtenidas de la búsqueda sobre procesamiento de páginas web en algunos de los lenguajes de programación existentes.

Tabla 4.1 Herramientas encontradas para el procesamiento de páginas web

Obtención y Procesamiento de páginas web		
Nombre	**Definición**	**Lenguaje de Programación**
Scrapy	Es de código abierto y un marco de colaboración para la extracción de los datos que necesita de los sitios web. En una forma rápida y simple, pero extensible (Scrapy ORG, n.d.)	Python
Beautiful Soup	Biblioteca diseñada para proyectos que están tratando de obtener datos desde una página web (Crummy, 2014)	
PHP DOM	Es un analizador HTML, permite manipular HTML a través de la extracción de la información contenida dentro de las etiquetas de su código fácilmente (Chen, 2010)	PHP
Jaunt	Es una biblioteca de java para la obtención de información y automatización web. (Jaunt, n.d.)	Java

En la Tabla 4.2 se visualiza la única herramienta encontrada para obtener resultados desde "Google Académico", está desarrollada en lenguaje de programación Python, cabe mencionar que este buscador no posee una interfaz oficial para obtener resultados y la herramienta construye la consulta, la envía y luego procesa el código HTML devuelto con los resultados.

Tabla 4.2 Herramienta encontrada para obtener resultados de "Google Académico"

Obtención de resultados y búsqueda en Google Académico		
Nombre	**Definición**	**Lenguaje de Programación**
Scholar.py	Módulo de Python que implementa un interrogador y analizador para la salida de Google Académico. (Kreibich, 2013)	Python

En la exploración de los resultados que entrega "Google Académico" muchos de los artículos científicos están como documentos PDF, lo que hace necesario herramientas que procesen este tipo de documentos para extraer información. En la Tabla 4.3 se visualizan los resultados encontrados en este tipo de herramientas.

Tabla 4.3 Herramientas encontradas para el procesamiento de documentos PDF

Obtención y Procesamiento de documentos PDF		
Nombre	**Definición**	**Lenguaje de Programación**
pdfquery	Está diseñado para extraer de forma fiable los datos de conjuntos de archivos PDF con el menor código posible (Cushman, 2010).	Python
pdfminer	Es una herramienta para extraer información de documentos PDF. Centrada exclusivamente en la obtención y el análisis de datos de texto (Shinyama, 2014)	Python
Pdfbox	Es una herramienta de fuente abierta Java para trabajar con documentos PDF. Este proyecto permite la creación de nuevos documentos PDF, manipulación de documentos existentes y la posibilidad de extraer el contenido de los documentos (The Apache Software Foundation, n.d.)	Java
PDF Parse	Librería PHP para analizar archivos PDF y extraer elementos como texto (Malot, n.d.)	PHP

4.1 Tecnología para obtener y procesar páginas web

El desarrollo de las pruebas, para la obtención de información de la página web son las siguientes:

- Prueba 1 consiste en obtener el código HTML de una página web

Los resultados son reflejados en la Tabla 4.4.

Tabla 4.4 Resultado de prueba obtención de código HTML

Herramienta	**Prueba 1**
Scrapy	Si
Beautiful Soup	Si
PHP	Si
Jaunt	Si

Cada herramienta satisface la prueba (Ver Anexo II Pruebas herramientas seleccionadas y Anexo III Prueba Herramientas no seleccionadas), a modo de ejemplo se muestra el resultado de la prueba de Beautiful Soup, su código es presentado en la Figura 4.1 en lenguaje de programación Python. El resultado fue el código HTML de la página web de la página "Google".

```
import urllib2
from bs4 import BeautifulSoup
import re
def parse_url(my_url):
    header = {
        'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:27.0)
Gecko/20100101 Firefox/27.0',
        'Accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
        'Accept-Charset': 'ISO-8859-1,utf-8;q=0.7,*;q=0.3',
        'Accept-Encoding': 'none',
        'Accept-Language': 'en-US,en;q=0.8',
        'Connection': 'keep-alive'}
    request = urllib2.Request(my_url, headers=header)
    my_content = urllib2.urlopen(request)
    my_soup = BeautifulSoup(my_content)

    print(my_soup.prettify())

def main():
    parse_url("http://www.google.cl/")
if __name__ == '__main__':
    main()
```

Figura 4.1 Código de ejemplo Beautiful Soup

A través de las funciones disponibles en la librería de Beautiful Soup se puede acceder a una dirección web y obtener su código HTML para ser analizado y obtener la información más importante.

4.2 Tecnología para obtener y procesar documentos PDF

Para la extracción de la información en archivos PDF la prueba es la siguiente:

- Prueba 2 obtener la transformación a texto de un archivo PDF.

Los resultados son reflejados en la Tabla 4.5 de acuerdo a las pruebas realizadas de cada herramienta (Ver Anexo II Pruebas herramientas seleccionadas y Anexo III Prueba Herramientas no seleccionadas).

Tabla 4.5 Resultado prueba procesamiento de PDFs

Herramienta	**Prueba 2**
Pdfquery	Si
Pdfminer	Si
Pdfbox	Si
PDF Parse	Si

A modo de ejemplo la prueba realizada con la herramienta Pdfminer funciona exportando documentos PDF a un archivo de código HTML o texto plano se ve reflejada en la Figura 4.2, el código es muy simple, se ejecuta a través de la terminal del sistema operativo previa instalación de la herramienta.

```
$ pdf2txt.py -o salida.html articulo.pdf
```

Figura 4.2 Código Ejemplo Pdfminer

La herramienta toma como entrada el "articulo.pdf" y realiza una conversión a un archivo (salida.html) con el que es más fácil trabajar.

4.3 Tecnología para obtención y procesamiento de resultados de "Google Académico"

La prueba del módulo de Python para la obtención de resultados y búsqueda en Google Académico "Scholar.py" fue analizado de una manera diferente, esta herramienta fue la única encontrada para hacer búsquedas y obtener sus resultados, se analiza su código fuente y se observa que cuenta con la librería Beautiful Soup, construye y envía la consulta a Google Académico, luego recibe los resultados, los analiza y descompone, entregando los resultados limpios al usuario solo a través de la consola de Python, no posee interfaz gráfica, aparecen citas, patentes y libros los que no son necesarios en una RSL, solo considera artículos de fuentes primarias, además de entregar solo los resultados de la primera hoja. Se debe entonces seleccionar herramientas compatibles con "Scholar.py" para el desarrollo del prototipo.

El código de ejemplo de esta herramienta lo podemos observar en la Figura 4.3, muestra la estructura del objeto diccionario que soporta un artículo de "Google Académico", con los atributos de título, dirección, año, link del PDF entre otros, este diccionario contiene claves las cuales pueden ser cadenas o números para acceder a sus atributos. La Figura 4.4 muestra como construye la consulta para ser enviada al motor de búsqueda, obtener los resultados y aplicar un filtro. Para obtener los resultados, esta estructura de consulta es específica del motor, por tanto no puede ser modificada, solo los parámetros que se envían van cambiando su valor de acuerdo a lo requerido por el usuario.

```
class ScholarArticle(object):
    def __init__(self):
        self.attrs = {
            'title':         [None, 'Title',          0],
            'url':           [None, 'URL',            1],
            'year':          [None, 'Year',           2],
            'num_citations': [0,    'Citations',      3],
            'num_versions':  [0,    'Versions',       4],
            'cluster_id':    [None, 'Cluster ID',     5],
            'url_pdf':       [None, 'PDF link',       6],
            'url_citations': [None, 'Citations list', 7],
            'url_versions':  [None, 'Versions list',  8],
            'url_citation':  [None, 'Citation link',  9],
            'excerpt':       [None, 'Excerpt',       10],
        }
```

Figura 4.3 Objeto que da soporte a un resultado de búsqueda

```
SCHOLAR_QUERY_URL = ScholarConf.SCHOLAR_SITE + '/scholar?' \
        + 'as_q=%(words)s' \
        + '&as_epq=%(phrase)s' \
        + '&as_oq=%(words_some)s' \
        + '&as_eq=%(words_none)s' \
        + '&as_occt=%(scope)s' \
        + '&as_sauthors=%(authors)s' \
        + '&as_publication=%(pub)s' \
        + '&as_ylo=%(ylo)s' \
        + '&as_yhi=%(yhi)s' \
        + '&as_sdt=%(patents)s%%2C5' \
        + '&as_vis=%(citations)s' \
        + '&btnG=&hl=en' \
        + '&num=%(num)s'
```

Figura 4.4 Formato de consulta a "Google Académico"

Esta herramienta Scholar.py crea y realiza la consulta con los parámetros definidos por el usuario, obtenido el resultado, es decir, el código HTML Scholar.py realiza un filtrado en el código, con lo cual logra obtener el resultado con sus atributos, en este caso los atributos del objeto ScholarArticle de la Figura 4.3.

4.4 Análisis y Conclusiones

Si bien se encontraron herramientas en varios lenguajes de programación, para la extracción de información web y dado que se trabaja con Google Académico, se debió seleccionar dentro de un conjunto de herramientas compatibles con “Scholar.py” de Python, esto permitió procesar las búsquedas de Google Académico, es decir, brindó parte de la solución, se deberá entonces escoger entre de las herramientas compatibles y desarrolladas bajo el lenguaje de programación Python para el desarrollo del prototipo final, modificar la herramienta “Scholar.py” porque algunos de los resultados que entrega no son relevantes para una investigación, estos son libros, patentes y citas, además deberá procesar no solo la primera página web de resultados del buscador, sino, también las páginas que se muestran a continuación de esta y otros relevantes que no están presentes como autor o resumen.

Tabla 4.6 Tecnologías y/o herramientas seleccionadas para el desarrollo del prototipo

Tecnología y/o herramienta	**Justificación de la selección**
Python	Lenguaje de programación que se encontró mayor información respecto a la extracción de información de páginas web.
Scholar.py	Módulo de Python que permite la conexión con el motor de búsqueda seleccionado "Google Académico" escrito en lenguaje Python.
Beautiful Soup	El módulo de Python Scholar.py hace uso de esta librería para procesar los resultados de la búsqueda a través del código html.
Pdfquery	Procesar el link de cada uno de los artículos científicos encontrados como documentos PDF por el buscador.
Django	Es un framework que permite desarrollar aplicaciones web en Python, ayudando al desarrollador en la construcción de la aplicación.

Cada una de las tecnologías y/o herramientas seleccionadas, dará paso a la construcción del prototipo de software que ayude en una RSL. El marco de trabajo Django, será útil para crear el proyecto de una manera más estructurada de acuerdo al lenguaje Python, se deberá aprender su funcionamiento, para utilizar al máximo las potencialidades que ofrece (Ver Anexo II Pruebas herramientas seleccionadas y Anexo III Prueba Herramientas no seleccionadas).

5 DESARROLLO DE PROTOTIPO

En esta sección se describen las etapas llevadas a cabo en la creación del prototipo así también como los problemas encontrados, que no fueron previstos en el análisis y selección de la tecnología disponible.

5.1 Alcances

Con el desarrollo de este prototipo de software se logrará ayudar al usuario (investigador) en el desarrollo de una RSL, específicamente en la segunda etapa de Desarrollo de la revisión, permite la búsqueda de literatura científica en Google Scholar, el usuario ingresa la consulta y el sistema hace la consulta, guarda, filtra los resultados (artículos científicos) y obtiene información adicional (resumen y/o palabras claves), sobre ellos el investigador revisa cada uno pudiendo seleccionar, descartar y/o hacer comentarios sobre los resultados, luego el usuario podrá ver estadísticas de su avance en la revisión de cada consulta. Las herramientas existentes, que ayudan en un RSL son gestores bibliográficos como Mendeley, refWorks, JabRef y EndNote, permiten la creación de una base de datos y búsqueda de bibliografía pero no brindan el soporte necesario para una RSL.

5.2 Objetivo del software

El sistema debe realizar una búsqueda en "Google Académico" con las palabras claves definidas en la RSL, procesar los resultados obtenidos y mostrar al usuario los resultados filtrados, mostrando información que ayude al investigador a seleccionar o descartar el resultado. El sistema almacena cada una de las búsquedas realizadas, manteniendo un historial de búsqueda por usuario y estadísticas de revisión.

5.3 Descripción Global del Producto

El prototipo de aplicación web que ayuda a personas en el análisis y revisión de literatura científica primaria, realiza una consulta sobre Google Académico, llamada RevTool, obteniendo las primeras 10 páginas de resultados, dejando fuera citas, patentes y libros, junto con la obtención del resumen de los artículos y palabras claves de un número acotado de páginas web como Elsevier, IEEE, Jstor entre otros.

5.3.1 Interfaz de usuario universal

El prototipo considera interfaces a usuarios universales de inicio de sesión, esto significa que no están registrados en la aplicación web dando la posibilidad de ser parte de la aplicación a través de un formulario de registro, a los usuarios registrados da la posibilidad de inicio de sesión y un formulario de contacto por si alguien requiere información adicional.

5.3.2 Interfaz de usuario registrado

En la interfaz de usuario registrado podrá:

- Desarrollar una búsqueda con los términos claves.
- El resultado de la búsqueda será almacenado y podrá tener acceso cuando estime conveniente a cada uno de los resultados a través de la revisión.
- Ver información del artículo, además de la posibilidad de visitar la página web donde es almacenado, realizar comentarios, y cambiar su estado de acuerdo a si es pertinente con su investigación o no.
- Estadísticas sobre las búsquedas realizadas y las revisiones hechas, dando a conocer al usuario la cantidad de resultados obtenidos, los que fueron revisados, los seleccionados y los que no fueron seleccionados.

5.3.3 Interfaz Software

El prototipo considera diversas interfaces, que son necesarias para la conexión con el motor de búsqueda y posterior procesamiento de los resultados, incluye el módulo de Pyhton "Scholar.py" el cual fue modificado para que estuviera acorde a este proyecto, este módulo necesita de Beautiful Soup, que es un analizador de código HTML y Django es un framework para desarrollo de aplicaciones web en Python, las especificaciones se visualizan en la Tabla 5.1.

Tabla 5.1 Herramientas de software necesarios

Nombre	Abreviación	Versión	Fuente
Scholar.py	Scholar	2.0	GitHub
Beautiful Soup	Beautiful Soup	4.3.2	Crummy
Pdfquery	Pdfquery	0.2.7	Python Sofware
Django	Django	1.8	Django Project

5.4 Requerimientos Específicos

5.4.1 Requerimientos Funcionales del prototipo

Los requisitos funcionales del prototipo están dados principalmente en la Tabla 5.2.

Tabla 5.2 Requisitos funcionales del prototipo

Id	Nombre	Descripción
00	RF-00	El sistema deberá disponer de un ingreso de palabras claves generadas en el protocolo de búsqueda de la RSL.
01	RF-01	El sistema deberá consultar al motor de búsqueda "Google Académico" las palabras claves de acuerdo a la RSL.
02	RF-02	El sistema deberá procesar los resultados de la consulta dejando fuera citas, patentes y libros.
03	RF-03	El sistema deberá almacenar los resultados obtenidos por el motor (título, autor, año, dirección web, dirección del archivo pdf).
04	RF-04	El sistema deberá extraer información de los artículos obtenidos (resumen y/o palabras claves).
05	RF-05	El sistema deberá proveer al usuario la capacidad de revisión de los artículos almacenados.
06	RF-06	El sistema deberá disponer de estadísticas por palabras claves buscadas (cantidad de artículos: seleccionados, no seleccionados y no revisados).
07	RF-07	El sistema deberá tener un módulo de registro e inicio de sesión de usuarios.

5.4.2 Requerimientos no funcionales del prototipo

Los requisitos no funcionales del prototipo están dados en la Tabla 5.3.

Tabla 5.3 Requisitos no funcionales del prototipo

Id	Nombre	Descripción
00	RNF-00	El sistema debe estar restringido y rechazar acceso o modificaciones no autorizadas.
01	RNF-01	El sistema proveerá de mensajes de error, cuando estos ocurran.

5.5 Diagrama de casos de uso

En este ítem se consideran los actores que hacen la interacción con el sistema, qué es lo que realizan y qué debe hacer el sistema con ellos.

5.5.1 Actores

El actor del sistema es el Usuario, la información de su rol y nivel de conocimientos técnicos requeridos para usar el sistema, además de sus privilegios en el mismo, son detallados en la Tabla 5.4

Tabla 5.4 Especificación de los actores del sistema

Rol	Nivel de Conocimiento Técnicos	Privilegios del Sistema
Usuario	Conocimiento de la metodología de RSL y conocimiento básico en internet.	Puede realizar búsquedas en el sistema, consultar historial y estadísticas de búsquedas.

5.5.2 Casos de uso y descripción

El diagrama de casos de uso generados, se pueden ver en la Figura 5.1, muestra la interacción que existe del participante con el sistema.

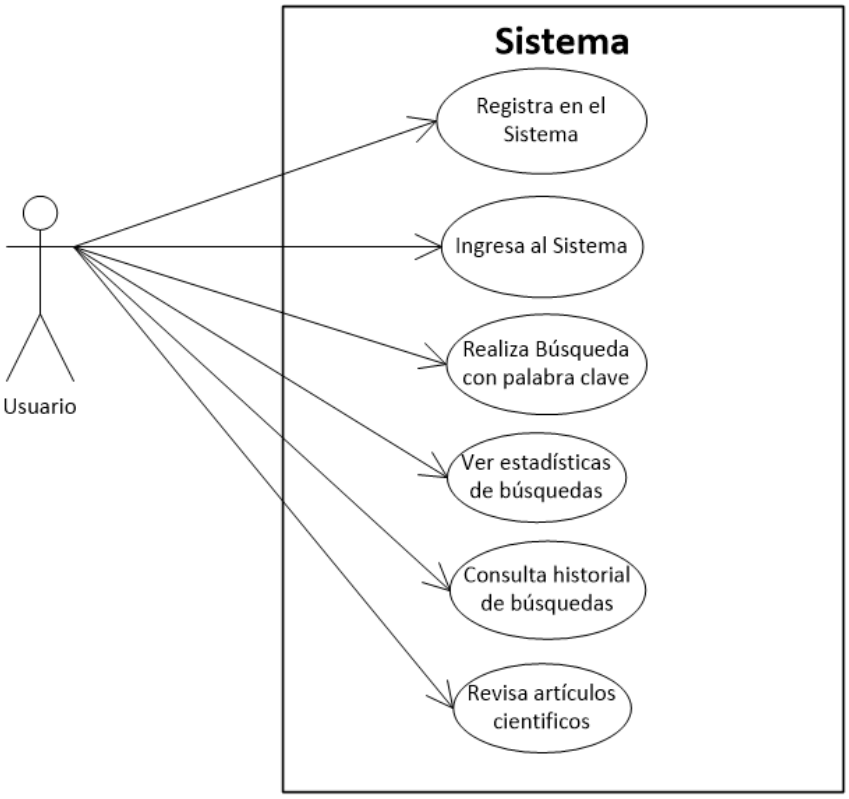

Figura 5.1 Diagrama de caso de uso del sistema

En el sistema aparece un rol de Usuario, este podrá obtener el rol una vez registrado, dentro de las principales funciones, se puede mencionar realizar la búsqueda con los términos claves definidos en la RSL, revisar cada uno de los artículos científicos obtenidos y consultar sobre su historial de búsquedas en el sistema.

5.5.3 Especificación de los Casos de Uso

En este apartado se describen los casos de uso especificados en la Figura 5.1 del apartado anterior.

5.5.3.1 Caso de Uso: Registra en el sistema

- Descripción: El usuario ingresa a la aplicación y realiza su registro de usuario del sistema, para poder realizar búsquedas, consultar el historial y ver estadísticas.
- Pre-Condiciones: Haber ingresado a la aplicación sin estar registrado.
- Flujo de Eventos Básicos: El usuario ingresará al menú de Registro, llenará el formulario con los datos requeridos y podrá iniciar sesión con éxito.

Al actor	El sistema
1 Ingresa a la página y hace clic en el menú de Registro.	2 Muestra página de registro a través de un formulario con los datos requeridos.
3 Ingresa los datos requeridos Usuario, Contraseña, Dirección de correo electrónico, Nombre, Apellido y hace clic en el botón Registrar.	4 El sistema deberá verificar los datos del formulario y realizara el registro correspondiente, llevando al usuario a la página de Iniciar Sesión.

- Flujo de Eventos Alternativo: Si el usuario ingresa datos del formulario erróneos.

Al actor	El sistema
3(a) Ingresa datos del formulario erróneos.	4(a) El sistema indicará al usuario que no fue registrado como usuario en el sistema.

- Post-Condiciones: El usuario queda registrado en el sistema.

5.5.3.2 Caso de Uso: Ingresa al sistema

- Descripción: El usuario ingresa al sistema como usuario
- Pre-Condiciones: Estar registrado en el sistema

  Flujo de Eventos Básicos: El usuario ingresará su nombre de usuario y contraseña, el sistema mostrará la página de inicio según corresponda al usuario.

Al actor	El sistema
1 Ingresa a la página y hace clic en Iniciar Sesión.	2 Muestra página de acceso a través de un usuario y contraseña.
3 Ingresa Usuario y Contraseña y hace clic en el botón ingresar.	4 El sistema deberá verificar el usuario y contraseña, una vez validadas se ingresará a la página principal del usuario registrado.

- Flujo de Eventos Alternativo: Si el usuario ingresa usuario y contraseña erróneos.

Al actor	El sistema
3(a) Ingresa usuario y contraseña erróneos.	4(a) El sistema indicará al usuario que no fueron validados su usuario y contraseña.

- Post-Condiciones: El usuario podrá usar el sistema.

5.5.3.3 Caso de Uso: Realizar búsqueda con palabra clave

- Descripción: El usuario hace la búsqueda de la palabra clave.
- Pre-Condiciones: El usuario debe haber ingresado la palabra clave en la pestaña de Buscar.
- Flujo de Eventos Básicos: El usuario una vez registrado e ingresada la palabra clave, debe presionar el botón buscar para hacer la búsqueda.

Al actor	El sistema
1 Ir a la barra de Buscar y hacer clic.	2 Muestra página de búsqueda donde aparece un formulario simple donde indica Ingresar palabra clave y el botón Buscar.
3 El usuario ingresa la palabra de búsqueda y hace clic en Enviar.	4 El sistema realizará la consulta a "Google Académico" y mostrará los resultados filtrados.

- Flujo de Eventos Alternativo: El sistema no mostrará resultados, entonces deberá realizar nuevamente la búsqueda.

Al actor	El sistema
1(a) El usuario presionará el botón Enviar.	2(a) El sistema no muestra resultados.
2(a) El usuario deberá realizar nuevamente la búsqueda.	

- Post-Condiciones: La búsqueda quedará almacenada en el sistema para posteriores consultas.

5.5.3.4 Caso de Uso: Ve estadísticas de búsqueda

- Descripción: El usuario consultará las estadísticas por palabra clave.
- Pre-Condiciones: Estar registrado en el sistema y haber realizado al menos una búsqueda.
- Flujo de Eventos Básicos: El usuario hará clic en la pestaña de Estadísticas mostrando la información de palabras claves.

Al actor	El sistema
1 Hacer clic en la pestaña de estadísticas.	2 El sistema mostrará a través de una lista las estadísticas del proceso de avance de la revisión por fecha.

- Flujo de Eventos Alternativo: Selecciona una pestaña errónea.

Al actor	El sistema
1(a) El ingresa a una pestaña errónea.	2(a) El sistema mantendrá la pestaña Estadísticas en la cual podrá ver la información por palabras claves históricas.

- Post-Condiciones: No existen Post-Condiciones en este caso de uso.

5.5.3.5 Caso de Uso: Consultar historial de búsqueda

- Descripción: El usuario consultará el historial de búsquedas realizadas
- Pre-Condiciones: Estar registrado en el sistema y haber realizado al menos una búsqueda.
- Flujo de Eventos Básicos: El usuario hará clic en la pestaña de Revisión mostrando la información histórica de palabras claves y seleccionará alguna, mostrando los resultados obtenidos, seleccionados, no revisados y no seleccionados.

Al actor	El sistema
1 Hacer clic en la pestaña de revisión.	2 El sistema mostrará a través de una lista las palabras claves por fecha.
3 El usuario seleccionará alguna de ellas, por el tipo y presionara el botón Ver resultados.	4 El sistema mostrará los resultados obtenidos de acuerdo a su tipo y mostrará la lista de ellos.

- Flujo de Eventos Alternativo: Se eliminara una búsqueda.

Al actor	El sistema
3(a) El usuario selecciona Borrar.	4(a) El sistema mostrará un mensaje de advertencia y confirmación antes de borrar la búsqueda con sus resultados.
5 El usuario hará clic en aceptará.	6 El sistema eliminará la búsqueda.

- Post-Condiciones: No existen Post-Condiciones en este caso de uso.

5.5.3.6 Caso de Uso: Revisa artículos científicos

- Descripción: El usuario revisará artículos científicos de acuerdo a los resultados obtenidos por palabra clave
- Pre-Condiciones: Estar registrado en el sistema y haber realizado al menos una búsqueda.
- Flujo de Eventos Básicos: El usuario hará clic en la pestaña de Revisión, seleccionará una palabra clave de búsqueda, luego seleccionará la lista que desee revisar, aparecerán los artículos y entrará al artículo que revisará.

Al actor	El sistema
1 Hacer clic en la pestaña de revisión.	2 El sistema mostrará a través de una lista los resultados por fecha y palabra clave.
3 El usuario seleccionará la búsqueda y la lista que desee.	4 El sistema mostrará la lista de resultados obtenidos y la fecha de búsqueda.
5 El usuario hará clic en el artículo que quiera revisar	6 El sistema mostrará la información del artículo,
7 El usuario podrá cambiar el estado del artículo y realizar comentario sobre el mismo y presionará en el botón Guardar.	8 El sistema guardará la información.

- Flujo de Eventos Alternativo: Se selecciona un artículo erróneo.

Al actor	El sistema
3(a) El usuario selecciona una artículo que no deseaba.	4(a) El sistema mantendrá la dirección donde se encuentra en la cual podrá volver a la página anterior.

- Post-Condiciones: No existen Post-Condiciones en este caso de uso.

5.6 Modelamiento de datos

El diseño de la base de datos considera un modelo sencillo capaz de soportar los datos requeridos en esta primera instancia del prototipo en la Figura 5.2 se han omitido los atributos para mejorar su legibilidad.

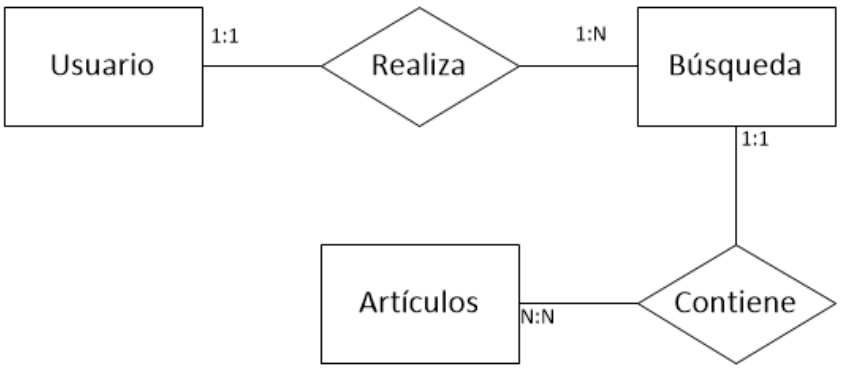

Figura 5.2 Modelo entidad relación del Prototipo

5.6.1 Entidades del sistema

Se explican a continuación cada una de las entidades de la Figura 5.2 Modelo entidad relación del prototipo.

5.6.1.1 Usuario

Almacena las principales características del usuario que se registra en el sistema, los atributos de esta entidad son:

- id_user: identificador único de la entidad.
- username: identificador del usuario.
- password: contraseña del usuario.
- email: correo electrónico del usuario.
- first_name: nombre del usuario.
- last_name: apellido del usuario.

5.6.1.2 Búsqueda

Almacena la característica de la búsqueda contiene los siguientes atributos:

- id_busqueda: corresponde al identificador único de la entidad.
- consulta: corresponde al término clave a buscar.
- fecha: es la fecha en que se realiza la búsqueda.
- nroResultados: es el número de resultados artículos obtenidos en la búsqueda.

5.6.1.3 Artículo

Almacena las características de los artículos resultantes de la búsqueda, los atributos correspondientes son:

- id_articulo: corresponde al identificador único de la entidad.
- p_clave: corresponde a las palabras claves del artículo.
- titulo: título del artículo.
- link: dirección del artículo.
- anio: año de publicación del artículo.
- num_citaciones: número de citaciones del artículo.
- num_versiones: número de versiones del artículo.
- cluster_id: número identificador de Google.
- link_pdf: dirección del archivo pdf del artículo.
- link_vesiones: dirección de las versiones del artículo.
- link_citaciones: dirección de las citaciones del artículo.
- resumen: resumen del artículo.
- autor: autor del artículo.
- estado: identifica el estado del artículo seleccionado, no revisado o no seleccionado.

5.7 Diseño Físico de la Base de datos

El diseño de la base de datos (ver Figura 5.3) este modelo solo incluye las tablas usadas por el sistema no así las que genera el marco de trabajo django (Ver Anexo 8 Base de datos).

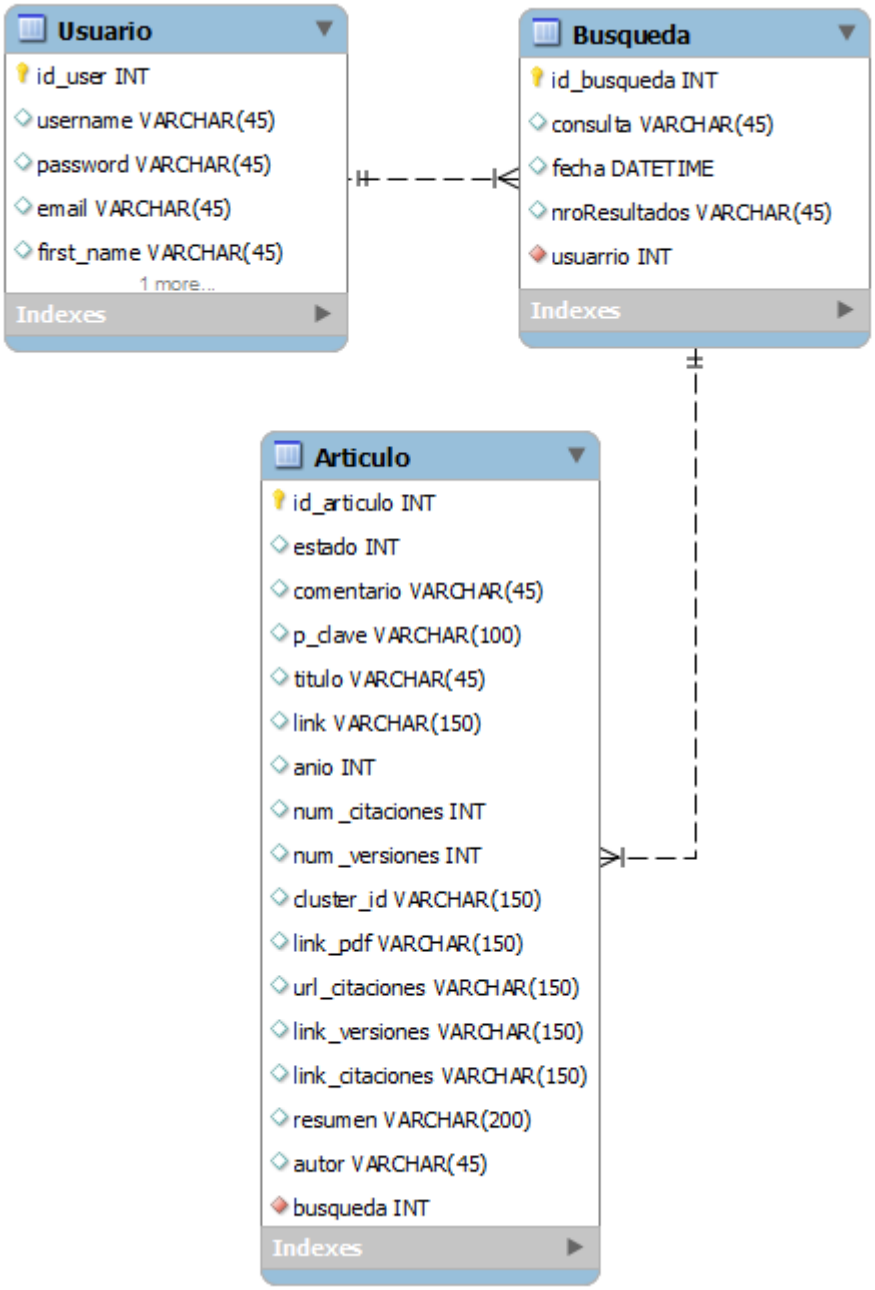

Figura 5.3 Modelo relacional

5.8 Diseño de arquitectura funcional

La arquitectura del prototipo (Ver Figura 5.4) considera la combinación de todas las tecnologías seleccionadas para el desarrollo, en el ámbito de las tecnologías django, es donde se desarrolla toda aplicación web, base de datos registro de usuario y almacenamiento de búsqueda, luego en el nivel de Beautiful Soup, Selenium y Scholar.py es donde se emula la consulta a Google académico y los resultados entregados por Google se procede a buscar información (como resumen y palabras claves si está disponible) adicional del artículo si es que su estructura está previamente disponible en el prototipo.

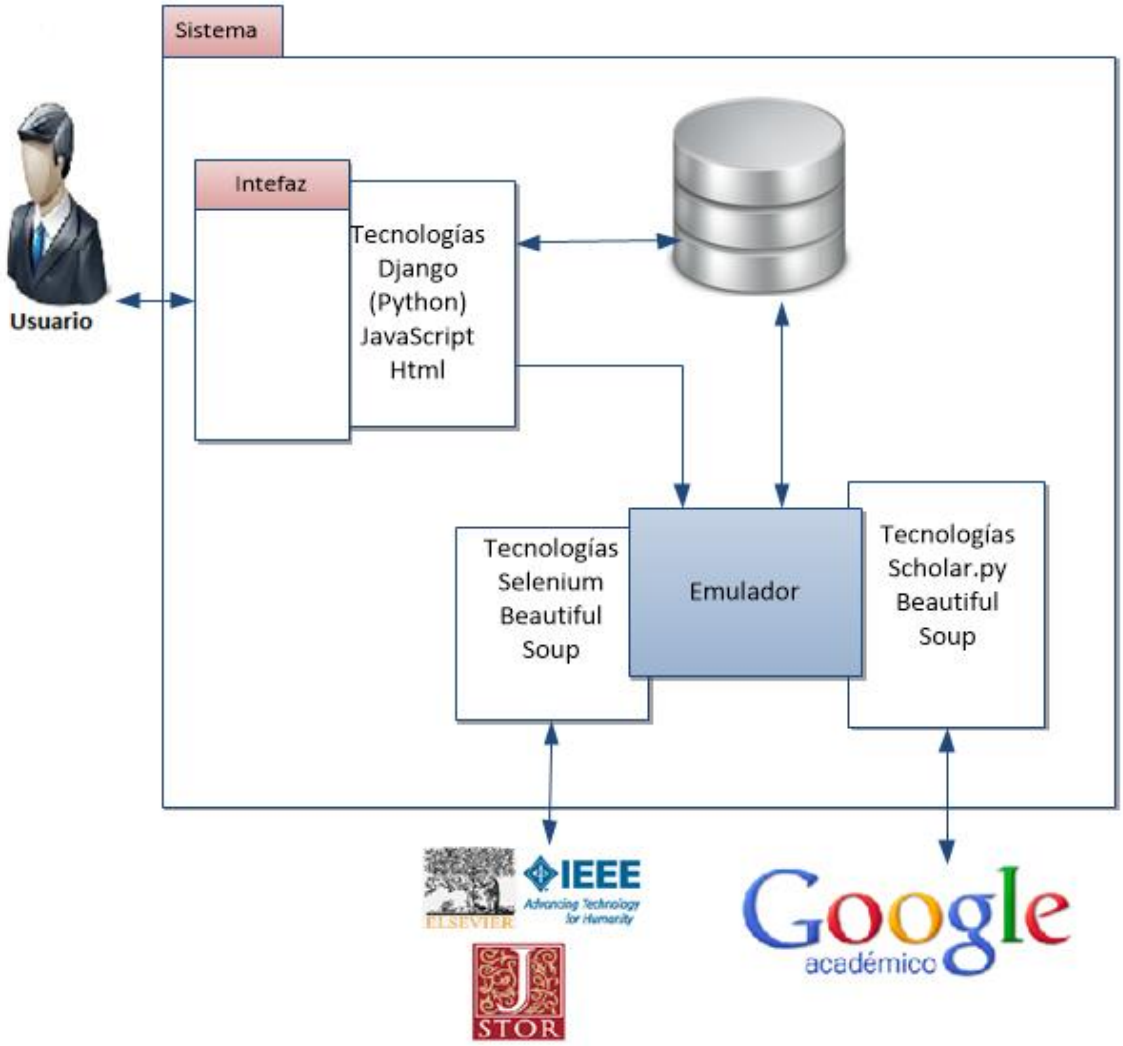

Figura 5.4 Interrelaciones existentes

5.8.1 Limitaciones, problemas y soluciones encontrados

El sistema es quien interactúa con el usuario y se comunica con el módulo Scholar.py, este módulo debió ser modificado para satisfacer los requerimientos del prototipo, inicialmente este un módulo fue operado por la consola del sistema a través de su línea de comandos, se desarrolló las siguientes modificaciones:

- Adaptación para el almacenamiento de los resultados en la base de datos y operación con el sistema.
- Filtro de resultados, sacando de los resultados los libros, patentes y citas.
- Agregar autor o autores del artículo.
- Agregar resumen o abstract como información adicional al artículo cuando la estructura de la página web fuente del artículo esté disponible en el prototipo
- Agregar palabras claves si están disponibles en donde la estructura esta previamente almacenada en el prototipo.
- Obtener número de resultados de Google Académico.
- Agregar estructura de páginas web más comunes para extraer información (resumen y/o palabras claves).

En el momento de obtener el código fuente para revisar la estructura de la página web del artículo se encontraron dificultades, estas dificultades están relacionadas con la tecnología JavaScript que solo es visible a través de un navegador, para ello fue necesario agregar Selenium, una herramienta para realizar pruebas a páginas web. Esta herramienta permite obtener el código de la página web con toda la información disponible ya que ejecuta la dirección de la página web a través de un navegador por consola y devuelve el código fuente con los JavaScript ejecutados para luego ser procesado con Beautiful Soup sacando la información ya sea palabras claves o resumen para tener mayor información por el investigador para hacer una selección o descarte.

Una de las limitaciones del sistema está en el análisis de documentos PDF a través de la web, lo que requiere la descarga del documento en formato PDF y posterior procesamiento, esto significa mucho tiempo en la descargar de este tipo de documentos, por ello, se optó por una

visualización previa del documento en el prototipo para realizar una selección o comentar el artículo. Otra limitación es el uso de la librería Scholar.py ya que esta no es una librería oficial de Google académico (Google académico no provee librería oficial para la conexión con su motor) y al obtener los resultados de manera automática Google se protege de la extracción de información poniendo barreras que impiden el acceso de manera automatizada a sus resultados, una manera de vulnerar estas barreras es cambiando de dirección IP cuando es usada de manera local o ingresado una persona desde el servidor de la aplicación a Google académico superando las pruebas impuestas por el motor.

5.8.2 Interfaz gráfica

En esta sección se presentan capturas de la aplicación, en la Figura 5.5 vemos la página donde el usuario registrado realiza las búsquedas por palabras claves, el usuario debe estar registrado en la aplicación para ver esta pantalla.

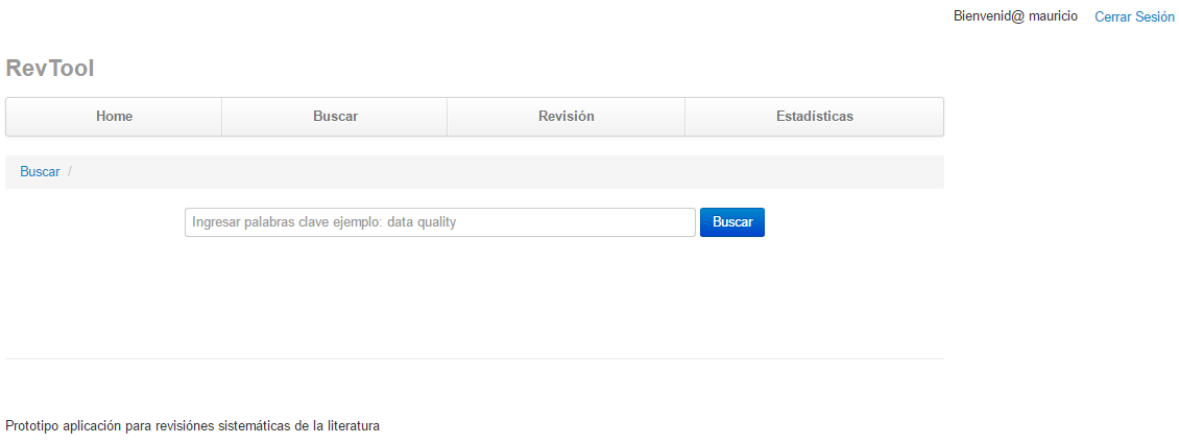

Figura 5.5 Página de Búsqueda

En la Figura 5.6 se observa cada palabra clave con sus fechas respectivas, la aplicación da la opción de borrar la búsqueda, junto con los artículos encontrados, el usuario puede filtrar por fecha y ver la totalidad de resultados o los resultados que no han sido revisados.

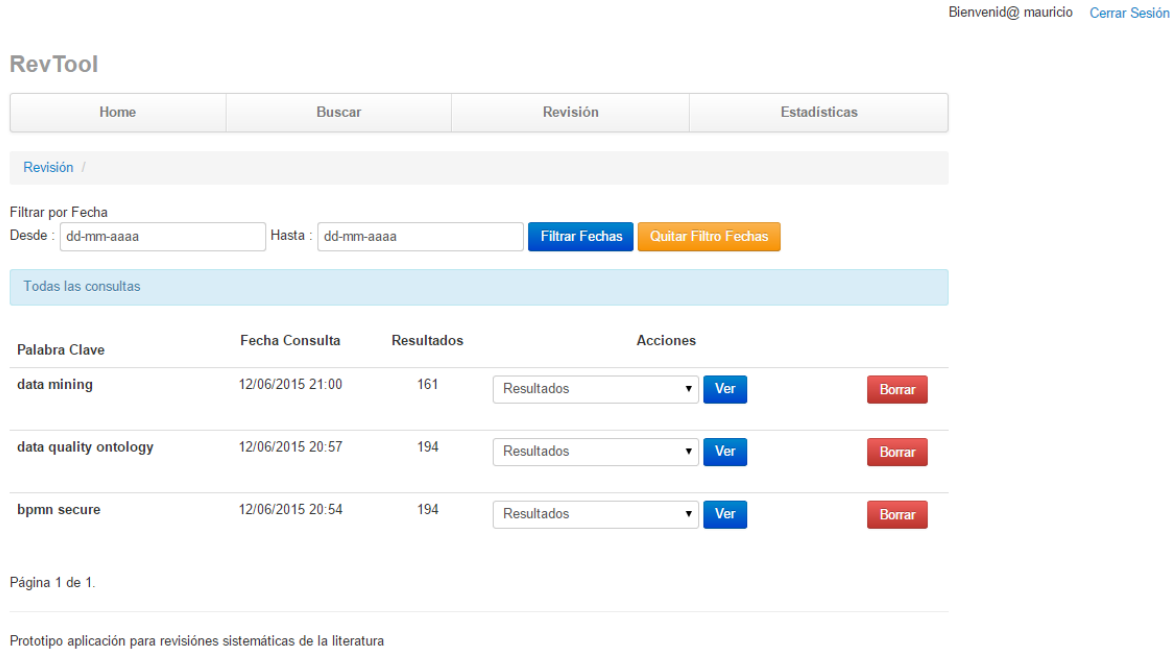

Figura 5.6 Revisión de búsquedas

En la Figura 5.7 está la estructura del artículo científico rescatado.

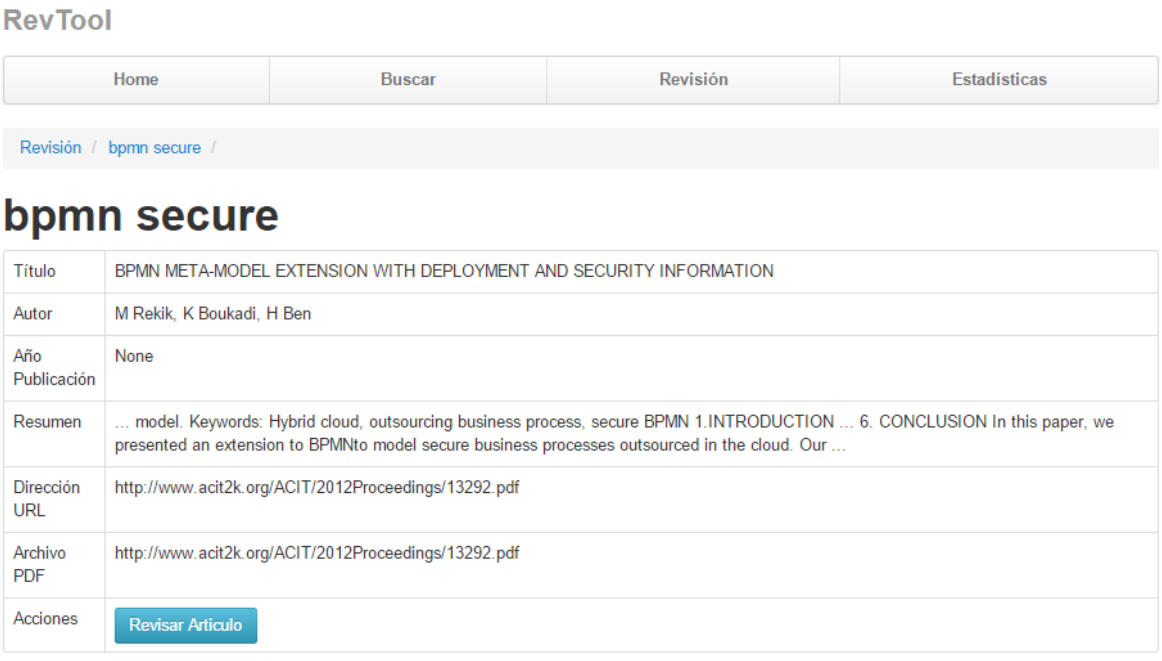

Figura 5.7 Resultado de búsqueda

En la Figura 5.8 se observa la página de revisión, donde el investigador de acuerdo a los datos y la visualización previa del artículo, se realiza una selección previa y comentarios.

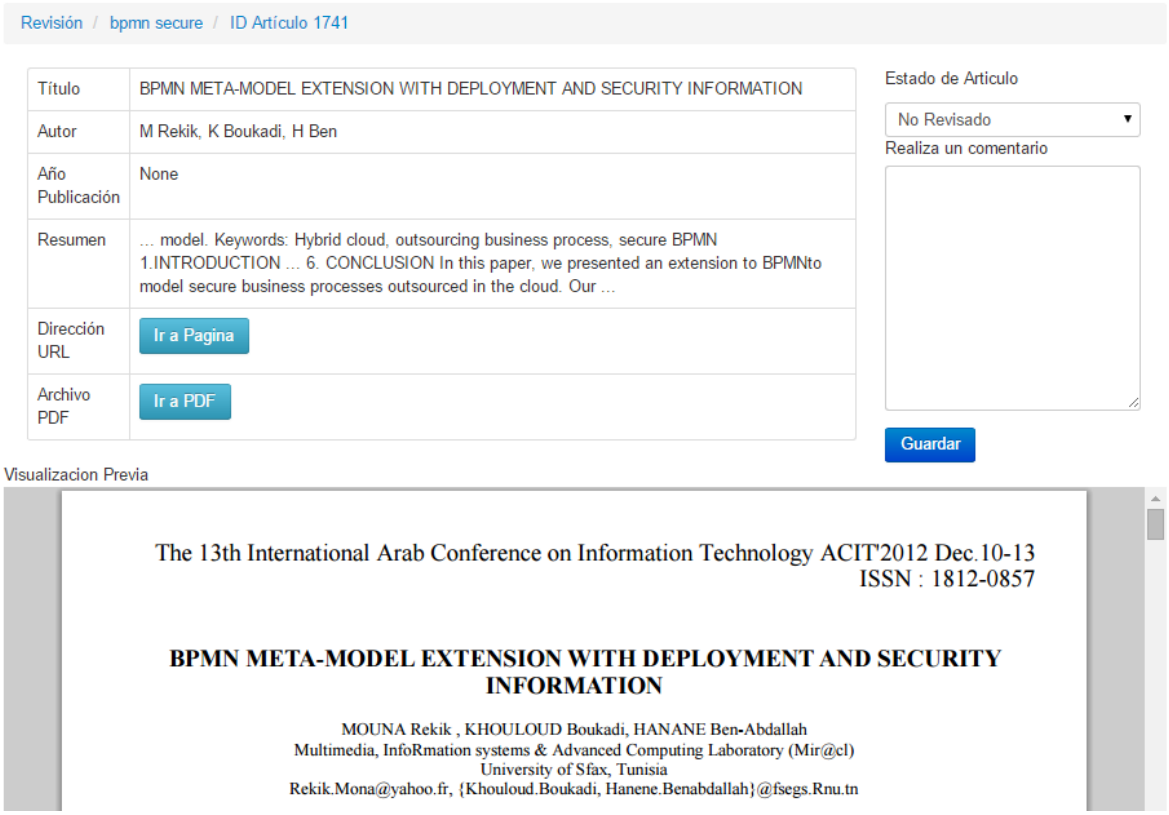

Figura 5.8 Revisión del artículo

En la Figura 5.9 muestra estadísticas de las búsquedas y revisiones realizadas.

Figura 5.9 Estadísticas sobre las revisiones

5.9 Pruebas

El desarrollo de las pruebas fue manual. Las cuales se centraron en ver el flujo de trabajo y los formularios. Estas pruebas principalmente fueron dadas por la conexión de la aplicación con Google Académico (tiempos de acceso y respuesta), formularios y navegación de la página.

5.9.1 Planificación

En esta sección se consideran los elementos a probar del sistema y sus criterios de cumplimiento.

5.9.1.1 Elementos de prueba

Los elementos que serán partícipes de las pruebas son:

- Los requisitos funcionales de la aplicación.
- Los requisitos no funcionales
- Desempeño del sistema (Tiempos de acceso y respuesta).

5.9.1.2 Especificación de las pruebas

Las características que fueron probadas, son los requisitos funcionales del sistema y formularios del sistema a través de las actividades detalladas en la Tabla 5.4 y 5.5, se determina su éxito de acuerdo a los criterios de cumplimiento establecidos para cada prueba.

Tabla 5.5 Especificación de las pruebas funcionales (1/2)

Código prueba	Nivel de prueba	Objetivo de la prueba	Actividades de prueba	Criterios de cumplimiento
RTP01	Sistema	Obtener resultados de Google Académico	1. Iniciar sesión. 2. Ingresar a pestaña de búsqueda. 3. Ingresar palabra clave. 4. Realizar clic en el botón buscar.	Deben aparecer en la pantalla el resultado de la búsqueda y por cada resultado el botón de revisar.

Tabla 5.6 Especificación de las pruebas funcionales (2/2)

Código prueba	Nivel de prueba	Objetivo de la prueba	Actividades de prueba	Criterios de cumplimiento
RTP02	Sistema	Revisar Artículo	1. Iniciar sesión. 2. Ingresar a pestaña de revisión. 3. Seleccionar una búsqueda presionando el botón de ver resultado búsqueda. 4. Realizar clic en el botón revisar artículo. 5. Cambiar estado del artículo, realizar comentario y presionar Guardar.	Debe aparecer el comentario y el estado nuevo del artículo.
RTP03	Sistema	Verificación de registro en la aplicación	1. Ingresar a la aplicación. 2. Ir a la pestaña de Registro. 3. Llenar los datos del formulario. 4. Presionar Registrar.	El sistema debe mostrar la pestaña inicio de sesión.
RTP04	Sistema	Ver estadísticas de las búsqueda y revisión	1. Iniciar Sesión. 2. Ir a la pestaña de estadísticas.	El sistema debe mostrar las una tabla con las estadísticas.

Tabla 5.7 Especificación de pruebas no funcionales

Código prueba	Nivel de prueba	Objetivo de la prueba	Actividades de prueba	Criterios de cumplimiento
RTP05	Sistema	Ingreso con usuarios no creados	1. Ir a la página. 2. Iniciar Sesión con usuario y contraseñas erróneos.	El sistema debe mostrar un mensaje de error al ingresar.
RTP06	Sistema	Llenar formularios con datos	1. Iniciar sesión. 2. Ir a formularios. 3. Ingresar los datos requeridos. 4. Presionar el botón Guardar o enviar.	El sistema debe mostrar un mensaje de error si los datos ingresados en el formulario son erróneos o un mensaje de éxito en llenar el formulario.

En la Tabla 5.8 vemos las especificaciones de las pruebas con respecto a tiempos de respuesta con respecto al acceso de la búsqueda y tiempos de acceso a la información de artículos científicos.

Tabla 5.8 Especificación de pruebas de desempeño

Código prueba	Nivel de prueba	Objetivo de la prueba	Actividades de prueba	Criterios de cumplimiento
RTP08	Sistema	Verificar tiempos de respuesta en la obtención de resultado	1. Iniciar Sesión, 2. Ir a la pestaña de búsqueda. 3. Ingresar a la aplicación palabra clave. 4. Buscar palabra clave.	El sistema debe mostrar a través de la consola de Python el tiempo de demora en el acceso a los resultados de Google académico hasta que los muestra al usuario.
RTP09	Sistema	Verificar tiempos de respuesta de acceso a resumen y palabras claves.	1. Iniciar Sesión. 2. Ir a la pestaña de revisión. 3. Seleccionar una búsqueda y presionar "Ver Resultados". 4. Seleccionar un artículo de los resultados e ingresar a revisión del artículo.	El sistema debe mostrar la información del artículo y el resultado del acceso a la página web del artículo. Junto con ello también deberá mostrar por la consola de Python el tiempo de acceso a la información del artículo.

En la tabla 5.9 vemos la matriz de trazabilidad en la cual quedan cubiertos los requisitos funcionales y no funcionales a través de las pruebas del prototipo.

Tabla 5.9 Trazabilidad pruebas – requisitos

Prueba\Requisito	RF-00	RF-01	RF-02	RF-03	RF-04	RF-05	RF-06	RF-07	RNF-00	RFN-01
RTP01	X	X	X	X						
RTP02					X	X				
RTP03								X		
RTP04							X			
RTP05								X	X	X
RTP06										X
RTP08		X								
RTP09						X				

5.9.2 Desarrollo de las pruebas

El desarrollo de las pruebas y su avance está dado por la Tabla 5.10 que muestra el porcentaje de aprobación en pruebas de acuerdo al avance en el desarrollo, pruebas funcionales y no funcionales, de acuerdo a las Tablas 5.5, 5.6 y 5.7.

Tabla 5.10 Avance de acuerdo al tiempo

Prueba\Requisito	Ene-15	Feb-15	Mar-15	Abr-15
RTP01	100%	100%	100%	100%
RTP02	20%	50%	70%	100%
RTP03	70%	70%	100%	100%
RTP04	50%	65%	80%	100%
RTP05	100%	100%	100%	100%
RTP06	70%	76%	90%	100%
RTP08	0%	0%	50%	100%
RTP09	0%	0%	50%	100%

El desarrollo de las pruebas de acuerdo al desempeño RTP08 (ver Tabla 5.8) sobre tiempos de respuesta en la obtención de resultados están en la Tabla 5.11 a través de una conexión de

banda ancha móvil (internet inalámbrico) y la Tabla 5.12 a través de una conexión por banda ancha por cable de red (internet alámbrico) en dependencias de la universidad.

Tabla 5.11 Obtención de resultados de Google académico a través de internet inalámbrico.

Palabra Clave	Resultados[artículos]	Tiempo [segundos]
data quality	185	109,9
sap crm	169	102,9
bpmn extensión	189	89,5
bpmn secure	194	92,3
data quality ontology	194	98,9
data mining	161	94,9
redes neuronales	191	109,3
data base structure	36	63,9
erp extensión	197	171
data cube	195	159
Promedio	**171,1**	**109,16**

Tabla 5.12 Obtención de resultados de Google académico a través de internet alámbrico.

Palabra Clave	Resultados[artículos]	Tiempo [segundos]
data quality	185	40,8
sap crm	169	36,5
bpmn extensión	189	33,7
bpmn secure	194	45,6
data quality ontology	194	34,6
data mining	161	35,5
redes neuronales	191	50,7
data base structure	36	15,6
erp extensión	197	64,8
data cube	195	54,8
Promedio	**171,1**	**41,26**

El tiempo de acceso del sistema para la extracción de información del artículo científico (ver Tabla 5.13 y 5.14) está dada por la prueba RTP09 (Tabla 5.8). Al igual que la prueba anterior se desarrolló a través de una banda ancha inalámbrica y una banda ancha por cable.

Tabla 5.13 Tiempo de acceso al artículo científico a través internet inalámbrico.

Página Web	Tiempo [segundos]
Science Direct	57,8
Science Direct	58,6
Science Direct	51,8
IEEE	48,2
IEEE	48,2
IEEE	68,9
Dialnet	57,6
Dialnet	50,5
Jstor	48,5
Jstor	53,2
Promedio	**54,3**

Tabla 5.14 Tiempo de acceso al artículo científico a través de internet alámbrico.

Página Web	Tiempo [segundos]
Science Direct	15,2
Science Direct	20,5
Science Direct	22,8
IEEE	10,0
IEEE	11,9
IEEE	19,2
Dialnet	20,4
Dialnet	30,7
Jstor	19,5
Jstor	41,2
Promedio	**21,14**

5.9.3 Conclusiones de las pruebas

Como conclusión del desarrollo de las pruebas podemos observar que, se ha cumplido con la totalidad de los requisitos, de acuerdo a los criterios de cumplimiento detallados en las especificaciones, este avance fue a medida del desarrollo del software de acuerdo a la Tabla 5.10 podemos obtener la Figura 5.5 que muestra el porcentaje de avance de acuerdo a los meses de desarrollo.

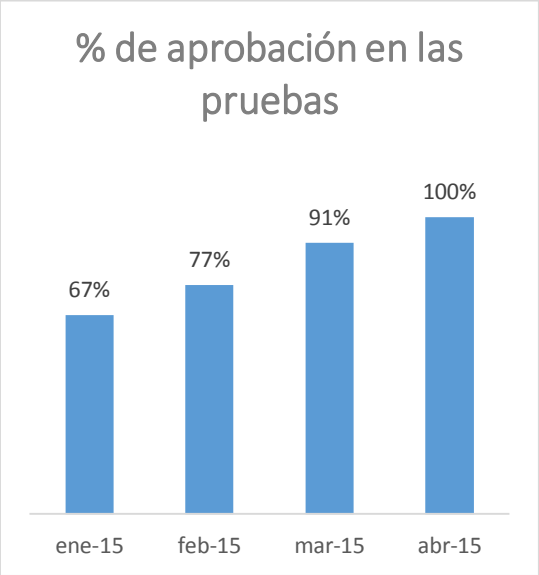

Figura 5.10 Aprobación de las pruebas por meses

Sobre el desempeño de la aplicación referente a las pruebas RTP 08 y 09 de la Tabla 5.8 podemos sacar conclusiones de los resultados obtenidos y el tiempo de acceso. La obtención de resultados de Google académico a través de una conexión a internet inalámbrica (ver Tabla 5.10) da un promedio de 1.56 artículos por segundo, mientras que a través de una conexión de red alámbrica (ver Tabla 5.11) está en el orden de 4,14 artículos por segundo. El acceso a la página fuente del artículo para la extracción de información (resumen y/o palabras claves) a través de una conexión de red inalámbrica (ver Tabla 5.12) está en el orden de 54 segundos por el acceso a esa información mientras que en una red alámbrica (ver Tabla 5.13) está en el orden de 21,14 segundos. De lo anterior debemos inferir que el acceso a la información y tiempos de respuesta están condicionados por la velocidad de conexión a internet, estas tienen una relevancia importante en realizar consultas y acceso a la información.

6 CONCLUSIONES

Con la investigación previa de las tecnologías disponibles, se ha descubierto la gran cantidad de herramientas que existen para la extracción de la información en la web, términos nuevos que surgen en este ámbito. Las herramientas disponibles para los investigadores para buscar información relacionada en sus trabajos, es poca, solo gestores de bibliografía y buscadores en internet que no ayudan lo suficiente para una revisión sistemática de la literatura.

Dentro de las dificultades enfrentadas durante el desarrollo se encuentra la tecnología, en la cual se desarrolló el prototipo lenguaje de programación Python y su marco de trabajo web Django, esta tecnología no era manejada y se presentaron dificultades a medida que se avanzaba en el desarrollo, una de las cosas importantes que se puede encontrar en la web, es la gran cantidad de documentación existente, ayudando a resolver los problemas presentados. La modificación del módulo Scholar.py se presentó como un desafío del cual su resultado fue favorable, para llevar a cabo esta actividad fue necesario revisar exhaustivamente su código para realizar las modificaciones correspondientes y entender su funcionamiento, con poca información disponible sobre el módulo en internet.

Otra dificultad encontrada, es la extracción de información de páginas web, muchas de ellas establecen limitaciones para aplicaciones que intentan recolectar información de sus sitios web, también nuevas tecnologías que se ejecutan por el lado del usuario o cliente, este es el caso de aplicaciones web que usan JavaScript, al realizar la petición a estos sitios web con librerías estándares como Beautiful Soup no se dispone de la totalidad del código y en muchos casos aparecen mensajes en el código HTML de la página sugiriendo habilitar la tecnología en el navegador, entonces se deben emular navegadores para poder ejecutar las funciones JavaScript y tener la totalidad del código HTML lo que significa que se puede extraer la totalidad de información del sitio web, con un costo de tiempo significativo.

Los objetivos del proyecto y del software fueron cumplidos, al identificar métodos y herramientas de extracción de información en la web, con ello se desarrolló el prototipo, en base a las encontradas y compatibles entre sí, dando cumplimiento a los requisitos planteados.

El procesar documentos con formato PDF también es una dificultad presente, no en si el procesamiento como texto del documento, sino de obtener el documento ya que para poder procesar su contenido, es necesario la descarga del archivo, esto toma bastante cantidad de tiempo si consideramos que muchos resultados en internet están en este formato, por ello se optó por no incluir esta funcionalidad en esta iteración del prototipo.

Al ser un desarrollo basado en prototipos, sienta una base para posteriores modificaciones para agregar nuevos motores de búsqueda, nuevas estructuras que obtengan mayor información y nuevas funcionalidades. La abundante documentación que existe en Python, ayuda a enfrentar estos desafíos y seguir mejorando en el desarrollo de esta aplicación, que sin duda tiene un gran potencial para el desarrollo de la investigación en nuestro país, ayudando así a su desarrollo.

Como trabajo futuro, se pretende agregar nuevos motores de búsqueda de literatura científica, debido a que un investigador no solo realiza búsquedas en Google Académico, porque existe una variada gama de buscadores que proveen revistas científicas como Elsevier o IEEE, además de enmarcar la búsqueda dentro de un proyecto de revisión sistemática de la literatura, para obtener estadísticas de avance por la revisión sistemática que se lleve a cabo, además ver si es posible disminuir el tiempo que se demora al hacer una búsqueda, desarrollar búsquedas avanzadas aprovechando al máximo Google Académico y buscar a través de dominios específicos.

Este prototipo significó un gran desafío en el área profesional, la investigación de herramientas entregó resultados favorables que permitieron su construcción.

7 BIBLIOGRAFÍA

Acosta, M., López, E., & Espinoza, E. (2011). Metodología de investigación en el desarrollo de software. *Iiis.org*. Retrieved from http://www.iiis.org/CDs2011/CD2011CSC/CISCI_2011/PapersPdf/CA918XJ.pdf

Adobe Systems Sofware. (1994). Acerca del formato PDF de Adobe. Retrieved from http://www.adobe.com/la/products/acrobat/adobepdf.html

Chen, S. C. (2010). PHP Simple HTML DOM Parse. Retrieved December 14, 2014, from http://simplehtmldom.sourceforge.net/

Crummy. (2014). Beautiful Soup. Retrieved December 14, 2014, from http://www.crummy.com/software/BeautifulSoup/

Cushman, J. (2010). Git Hub PDFquery. Retrieved December 14, 2014, from https://github.com/jcushman/pdfquery

django project. (2005). django. Retrieved December 14, 2014, from https://www.djangoproject.com/

Jaunt. (n.d.). Jaunt Java Web Scraping & Automation. Retrieved December 14, 2014, from http://jaunt-api.com/index.htm

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, *33*, 28. http://doi.org/10.1.1.122.3308

Kreibich, C. (2013). Scholar.py. Retrieved December 14, 2014, from https://github.com/ckreibich/scholar.py

Latorre, A. (2003). *Investigación acción*. Gra{ó}.

Malot, S. (n.d.). PDF Parse. Retrieved December 14, 2014, from http://www.pdfparser.org/

Myers, M. D., & Avison, D. E. (2002). *Qualitative research in information systems: a reader*. *Introducing Qualitative Methods Series* (p. 328). http://doi.org/10.1007/978-0-387-35309-8

Pressman, R. S. (1997). *Ingeniería del Software: Un enfoque práctico*. Mikel Angoar.

Python ORG. (n.d.). python. Retrieved December 14, 2014, from https://www.python.org/

Sánchez, J. (2006). ¿Que es un "Framework"? Retrieved December 16, 2014, from http://jordisan.net/blog/2006/que-es-un-framework/

School of Data. (n.d.). Introducción a la extracción de datos. Retrieved from http://es.schoolofdata.org/introduccion-a-la-extraccion-de-datos-de-sitios-web-scraping/

Scrapy ORG. (n.d.). Scrapy. Retrieved December 14, 2014, from http://scrapy.org/

Shinyama, Y. (2014). PDFminer. Retrieved December 14, 2014, from http://www.unixuser.org/~euske/python/pdfminer/

The Apache Software Foundation. (n.d.). Apache PDFbOX - A Java PDF Library. Retrieved December 14, 2014, from https://pdfbox.apache.org/

W3C. (2013). HTML. Retrieved December 14, 2014, from http://www.w3.org/html/

ANEXO I BASE DE DATOS

7.1 Base de datos

El presente anexo establece la construcción del modelo de la base de datos, el marco de trabajo Django genera la base de datos relacional a través de clases orientadas a objetos, es por ello que solo se desarrolla la clase con los atributos correspondientes y el marco de trabajo genera la base de datos relacional y se conecta al sistema de gestión de base de datos seleccionado, en este caso MySql:

Tabla 0.1 Diagrama de clases que representa la base de datos (1/2)

```
from django.db import models
from django.contrib.auth.models import User

class Busqueda(models.Model):
       usuario = models.ForeignKey(User)
       consulta = models.CharField(max_length = 150)
       fecha = models.DateTimeField(auto_now=True)
       nroResultados=models.IntegerField(default=0)

       def __unicode__(self):
             return (self.consulta)

class Articulo(models.Model):
      SELECCION = ( (0, 'No Revisado'), (1, 'No Seleccionado'), (2,
'Seleccionado'),)
      estado = models.IntegerField(default=0, choices=SELECCION)
      busqueda = models.ForeignKey(Busqueda)
      comentario = models.TextField(max_length=100,null=True,blank=True)
      p_clave = models.CharField(max_length=100,null=True,blank=True)
      titulo = models.CharField(max_length = 400,null=True,blank=True)
      link = models.CharField(max_length=500,null=True,blank=True)
```

Tabla 0.2 Diagrama de clases que representa la base de datos (2/2)

```
      anio = models.CharField(max_length=30,null=True,blank=True)
      num_citaciones =
models.CharField(max_length=30,null=True,blank=True)
      num_versiones = models.CharField(max_length=30,null=True,blank=True)
cluster_id = models.CharField(max_length=100,null=True,blank=True)
      link_pdf = models.CharField(max_length=500, null=True,blank=True)
      url_citaciones =
models.CharField(max_length=500,null=True,blank=True)
      link_versiones =
models.CharField(max_length=500,null=True,blank=True)
      link_citaciones =
models.CharField(max_length=500,null=True,blank=True)
      resumen = models.TextField(null=True,blank=True)
      autor = models.CharField(max_length=500, null =True, blank=True)

      def __unicode__(self):
          return (self.titulo)
```

ANEXO II PRUEBAS DE HERRAMIENTAS SELECCIONADAS

En el presente anexo se muestra las pruebas a las herramientas presentes en el prototipo

7.2 Scholar.py

Las pruebas de esta herramienta son hechas a través de la consola ejecutando el archivo y pasando a través de parámetros los términos de búsqueda (ver Tabla 10.1) generando la consulta y mostrando a través de la misma consola los resultados, pero solo de la primera hoja de resultados de Google Académico (ver Figura 10.1).

Tabla 0.1 Comando de ejecución del scholar.py

```
Python Scholar.py –phrase "data quality"
```

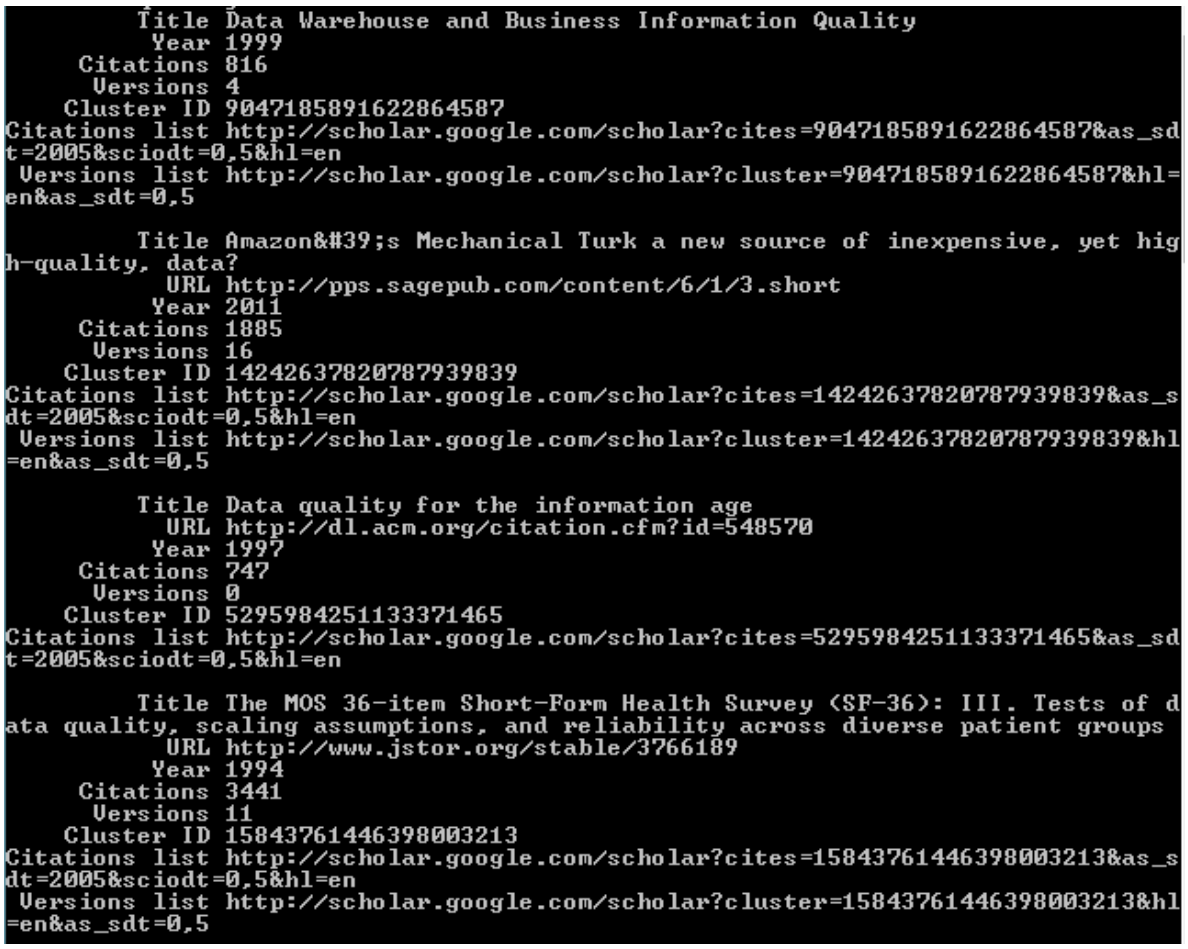

Figura 0.1 Resultados obtenidos

7.3 Selenium

Algunas de las páginas web utilizan JavaScript, esta herramienta ayuda a obtener el código fuente de la página porque utiliza un navegador para ello en la Tabla 10.2 se utiliza con el navegador Firefox pero al ejecutar el programa abre el navegador y obtiene el código, por ello se utilizó Selenium con PhantomJs que es un navegador sin interfaz gráfica.

Tabla 0.2 Programa ejemplo utilizando Selenium y Firefox

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

browser = webdriver.Firefox()

browser.get('http://scholar.google.cl')
assert 'Yahoo' in browser.title

elem = browser.find_element_by_name('p')  # Find the search box
elem.send_keys('seleniumhq' + Keys.RETURN)

browser.quit()
```

7.3.1 Phantomjs

Este navegador es utilizado por selenium para obtener los datos de la página web internamente, no posee interfaz gráfica y se puede utilizar a través de consola (ver Tabla 10.3).

Tabla 0.3 Programa ejemplo utilizando Selenium y Phantomjs

```
from  selenium  import webdriver
from  selenium.webdriver.common.keys import Keys
…
Driver                                                              =
webdriver.PhantomJS(executable_path='C:\python27\scripts\phantomjs.
exe')
driver.get(url)
html =  driver.page_source
print(html.encode("utf-8"))
```

7.4 Django

Este marco de trabajo web para Python provee de muchas herramientas para enfrentar el desarrollo web. Se trabajó a través de la consola, con comandos propios de django para crear el proyecto, dentro del proyecto general se crea la aplicación, luego se genera el modelo de la base de datos a través del paradigma orientado a objetos y luego el marco de trabajo genera la base de datos relacional de acuerdo a las clases (ver Anexo 8 Base de datos). Algunos de los comandos son dados a conocer en la Tabla 10.4.

Tabla 0.4 Comandos utilizado por Django

```
#Comenzar un nuevo proyecto
Python django-admin.py startproject MiProyecto
#Comenzar una nueva aplicación
Python manage.py startapp aplicacion
#Creación de la base de datos
Python manage.py syncdb
#Hace lo necesario antes de migrar cuando editas, borras o agregas
atributos o clases
Python manage.py makemigrations
#Hace la migración en la base de datos
Python manage.py migrate
#Habilita el proyecto en el servidor para ser accesado a través del
navegador
Python manage.py runserver
```

7.5 Beautiful Soup

Esta herramienta desarrollada en Python, permite extraer información de la página web, es fácil de utilizar, provee de mucha documentación y foros con mucha información sobre la extracción de diversos tipos de información. En la Tabla 10.5 se establece la extracción de links de acuerdo a la estructura de la página web que se necesita.

Tabla 0.5 Ejemplo de obtención de direcciones

```
soup = BeautifulSoup(html)
tag = soup.find(name='div', attrs={'id': 'gs_nml'})
link = productLinks = tag.findAll('a', attrs={'class' : 'gs_nma'})
    for links in link:
        a = links['href']
        print("link : "+a)
        self.link.append(a)
```

Figura 0.2 Código HTML donde se aplica Tabla 10.5

ANEXO III PRUEBA DE HERRAMIENTAS NO SELECCIONADAS

Este anexo considera la prueba de herramientas que no fueron seleccionadas por pertenecer al lenguaje de programación distinto a Python, debido a que al usar la librería Scholar.py (ver apartado Anexo 10.1) hace necesario herramientas compatibles con el lenguaje, si bien en este anexo existen herramientas desarrolladas en Python, las seleccionadas cumplían de mejor manera el propósito.

7.6 PHP DOM

PHP DOM es una librería de php que permite extraer información desde páginas web en ese lenguaje.

Tabla 0.1 Ejemplo PHP DOM

```
// Crea el DOM de la dirección URL
$html = file_get_html('https://scholar.google.cl/');

// Encuentra todas las imagines de Google Académico
foreach($html->find('img') as $element)
       echo $element->src . '<br>';

// Encuentra todos los link
foreach($html->find('a') as $element)
       echo $element->href . '<br>';

//Imprime el text plano de la pagina
echo file_get_html('http://www.google.com/')->plaintext;
```

7.7 Jaunt

Jaunt es una herramienta en java que permite la extracción de información de páginas web, se ve el ejemplo de utilización de esta herramienta en la Tabla 11.2.

Tabla 0.2 Ejemplo Jaunt

```
import com.jaunt.*;

public class GoogleScraperDemo{
  public static void main(String[] args) throws JauntException{
    UserAgent userAgent = new UserAgent();
    userAgent.settings.autoSaveAsHTML = true;
    userAgent.visit("http://scholar.google.com");
userAgent.doc.apply("butterflies");
    userAgent.doc.submit("Google Academico");

    Elements links = userAgent.doc.findEvery("<h3
class=r>").findEvery("<a>");
    for(Element link : links)
System.out.println(link.getAt("href"));
  }
}
```

7.8 Scrapy

Scrapy es una herramienta desarrollada en Python, que permite extraer los datos necesarios de las páginas web.

Tabla 0.3 Ejemplo Scrapy

```
from scrapy import Spider, Item, Field

class Post(Item):
    title = Field()

class BlogSpider(Spider):
    name, start_urls = 'GoogleAcademico',
['http://scholar.google.cl']

    def parse(self, response):
        return [Post(title=e.extract()) for e in response.css("h2
a::text")]
```

7.9 Pdfquery

Esta herramienta es necesaria siempre y cuando el documento es conocido ya que hace el tratamiento del documento como una imagen a través de coordenadas

Tabla 0.4 Ejemplo Pdfquery

```
pdf = pdfquery.PDFQuery("articulo.pdf")
pdf.load()
label = pdf.pq('LTTextLineHorizontal:contains("abstract")')
left_corner = float(label.attr('x0'))
bottom_corner = float(label.attr('y0'))
name = pdf.pq('LTTextLineHorizontal:in_bbox("%s, %s, %s, %s")' %
(left_corner, bottom_corner-30, left_corner+150,
bottom_corner)).text()
name
```

7.10 PDFMiner

Esta herramienta permite extraer información de documentos PDF a traves de la consola del sistema operativo transformando a texto el documento (ver Tabla 11.5).

Tabla 0.5 Ejemplo Pdf miner con la función PDF a texto

```
pdf2txt.py samples/simple1.pdf
```

7.11 Pdfbox

Pdfbox es una librería de java que permite trabajar con documentos PDF, permite crear y manipular documentos.

Tabla 0.6 Ejemplo Pdfbox

```
public class PDFReader{
    public static void main(String args[])
    {PDFTextStripper pdfStripper = null;
    PDDocument pdDoc = null;
    COSDocument cosDoc = null;
    File file = new File("C:/ariticulo.pdf");
    try {
        PDFParser parser = new PDFParser(new
FileInputStream(file));
        parser.parse();
        cosDoc = parser.getDocument();
        pdfStripper = new PDFTextStripper();
        pdDoc = new PDDocument(cosDoc);
        pdfStripper.setStartPage(1);
        pdfStripper.setEndPage(5);
        String parsedText = pdfStripper.getText(pdDoc);
        System.out.println(parsedText);
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        }
    }
}
```

7.12 PDF Parse

Esta es una librería para obtener elementos de archivos en PDF y transformarlo a texto, es para el lenguaje PHP.

Tabla 0.7 Ejemplo PDF Parser

```
<?php

include 'home/prueba.php';
$parser = new \Smalot\PdfParser\Parser();
$pdf    = $parser->parseFile('articulo.pdf');
$pages = $pdf->getPages();
foreach ($pages as $page) {
echo $page->getText();
}

?>
```