



Ingeniería de Sistemas

Procesos estocásticos y simulación

Proyecto Final Procesos Estocásticos y Simulación

Presentado por:  
Gabriel Jiménez Forero  
Cod:625455  
David Andrés Rubiano Venegas  
Cod:625457  
Leonardo Oliveros Coral  
Cod:624906

Presentado a:  
Roger Guzmán

25 de Noviembre del 2017

# Contenido

<b>1</b>	<b>Introducción</b>	<b>5</b>
1.1	Propósito y audiencia . . . . .	5
1.2	Contexto . . . . .	5
1.3	Organización del documento . . . . .	5
<b>2</b>	<b>Objetivos</b>	<b>6</b>
2.1	Objetivo general . . . . .	6
2.2	Objetivos específicos . . . . .	6
<b>3</b>	<b>Análisis de requerimientos</b>	<b>6</b>
3.1	Requerimientos funcionales . . . . .	6
3.2	Requerimientos no funcionales . . . . .	10
<b>4</b>	<b>Casos de uso</b>	<b>12</b>
<b>5</b>	<b>Diagrama de Clases</b>	<b>13</b>
<b>6</b>	<b>Modelo entidad - relación</b>	<b>13</b>
<b>7</b>	<b>Diagrama de componentes</b>	<b>14</b>
<b>8</b>	<b>Diseño de la arquitectura del sistema de recuperación de información</b>	<b>14</b>
<b>9</b>	<b>Metodología de desarrollo</b>	<b>15</b>
<b>10</b>	<b>Propuesta de la solución Computacional</b>	<b>21</b>
<b>11</b>	<b>Cálculo de la complejidad computacional usado para la recuperación de documentos</b>	<b>21</b>
<b>12</b>	<b>Manual de Usuario</b>	<b>22</b>
<b>13</b>	<b>Manual de Aministrador</b>	<b>24</b>
<b>14</b>	<b>Bibliografia</b>	<b>28</b>

## **Lista de figuras**

1	Diagrama caso de uso cliente . . . . .	12
2	Diagrama caso de uso sistema . . . . .	12
3	Diagrama de Clases . . . . .	13
4	Diagrama modelo entidad - relación . . . . .	13
5	Diagrama de componentes . . . . .	14
6	Diagrama de arquitectura del sistema . . . . .	14

## **Lista de tablas**

1	Tabla RFs-01 . . . . .	6
2	Tabla RFs-02 . . . . .	6
3	Tabla RFs-03 . . . . .	7
4	Tabla RFs-04 . . . . .	7
5	Tabla RFs-05 . . . . .	7
6	Tabla RFs-06 . . . . .	8
7	Tabla RFs-07 . . . . .	8
8	Tabla RFs-08 . . . . .	8
9	Tabla RFs-09 . . . . .	8
10	Tabla RFs-10 . . . . .	9
11	Tabla RFs-11 . . . . .	9
12	Tabla RFs-12 . . . . .	9
13	Tabla RNFs-01 . . . . .	10
14	Tabla RNFs-02 . . . . .	10
15	Tabla RNFs-03 . . . . .	10
16	Tabla RNFs-04 . . . . .	11
17	Tabla RNFs-05 . . . . .	11
18	Tabla RNFs-06 . . . . .	11
19	Tabla RNFs-07 . . . . .	11
20	Tabla tarea 01 . . . . .	19
21	Tabla tarea 02 . . . . .	19
22	Historia de usuario 1 . . . . .	20
23	Historia de usuario 12 . . . . .	20

# 1 Introducción

Grandes volúmenes de datos dan una idea clara de la información que se quiere transmitir, como por ejemplo una novela o un libro relacionado con astrología, pero uno de los principales problemas al tratar información es que a medida que se va almacenando más y más de ella, la búsqueda de información específica se complica, ya que a medida que el volumen de datos aumenta la búsqueda precisa se torna difusa. Entonces ¿qué se puede hacer para tener una claridad y o filtración de información de forma que lo que se lee es perfectamente lo que se interesa leer?

Es aquí donde la recuperación de información tendría un punto fuerte en la medida en que por medio de (“estructuras de información”) y metadatos se puede llegar a filtrar o restablecer haciendo un enfoque en las principales palabras claves de lo que se desea buscar. Teniendo una ventaja de lo que a material investigativo o de conocimiento se refiere.

## 1.1 Propósito y audiencia

El objetivo principal de este documento es plasmar de forma precisa las necesidades del cliente en términos del software de recuperación de la información, haciendo uso de diferentes tipos de vistas para describir los diferentes aspectos del sistema, y dejar la documentación lo más claro posible para que sea entendible por aquella audiencia externa a el desarrollo

El documento está dirigido a el usuario, al equipo de desarrollo y demás personas interesadas en el desarrollo del proyecto, Adicionalmente también puede ser usado por los usuarios que utilizarán el software y que necesiten definir nuevos requerimientos o ajustes retroalimentando así mejoras al desarrollo del sistema.

## 1.2 Contexto

Se desea desarrollar un sitio web en el que se pueda consultar la información de diferentes libros .txt previamente cargados en una base de datos, de tal forma que se facilite la recuperación de la información, debe ser desarrollado en HTML5, java y demás tipos de tecnologías que se requieran usar.

## 1.3 Organización del documento

El actual documento se encuentra dividido por diferentes secciones, En la primera encontraremos la introducción a este documento en el que se dará a entender sobre el tema a hablar a lo largo del documento, en la segunda sección se encontraran los objetivos en la cual se describe el objetivo general y los objetivos específicos, para la siguiente sección estará el análisis de los requerimientos allí se describirán los requerimientos funcionales y los requerimientos no funcionales, en la cuarta sección estarán los diagramas de casos de uso, posteriormente estarán los diagramas de clases, en la siguiente sección estará el diagrama modelo entidad relación de la base de datos, para la séptima sección estarán los diagramas de componentes después se encontrara el diseño de la arquitectura, luego se presentara el modelamiento matemático, ya para la decima sección estará descrita la metodología de desarrollo, finalmente en las dos últimas secciones se presenta la propuesta de la solución computacional y el cálculo de la complejidad computacional usado para la recuperación de los documentos.

## 2 Objetivos

### 2.1 Objetivo general

Diseñar un sistema que permita recuperar información por medio de un query de búsqueda para distintos libros en formato .txt que serán tratados y previamente almacenados en una base de datos.

### 2.2 Objetivos específicos

- Definir la arquitectura que mejor se ajuste a la solución planteada
- Generar un sistema amigable con el usuario
- Brindar una respuesta óptima de búsqueda al usuario con relación al query solicitado
- Analizar la efectividad de búsqueda respecto a grandes volúmenes de información

## 3 Análisis de requerimientos

En esta sección se presenta la identificación y especificación de los requerimientos tanto funcionales como no funcionales para el sistema de la recuperación de la información

### 3.1 Requerimientos funcionales

Tabla 1: Tabla RFs-01

FRs-01	Lectura de archivos
Dependencias:	N/A
Precondición:	Tener almacenados los libros que se desean leer en la carpeta "libros"
Descripción:	El sistema solo debe leer los archivos con extensión .txt guardados en una carpeta
Secuencia normal:	1- Lectura de ruta de documentos .txt
Poscondición:	Se leerá el documento para su tratamiento
Excepciones:	Solo leerá los archivos con formato .txt para otra extensión el sistema no lo evaluara
Comentarios:	Solo tomar los textos de la carpeta "Libros"

Tabla 2: Tabla RFs-02

FRs-02	Tokenizar libros
Dependencias:	- FRs-01 Lectura de archivos
Precondición:	Los documentos a tratar deben de estar en formato .txt
Descripción:	El sistema Almacenara los libros en formato .txt para su evaluación de filtros por tokens
Secuencia normal:	1- Lectura de cada uno de los libros 2- Aplicar técnica de tokenización
Poscondición:	Guardar las palabras de cada documento
Excepciones:	N/A
Comentarios:	Se debe realizar la tokenización para todo el texto de todos los libros

Tabla 3: Tabla RFs-03

FRs-03	Eliminación de StopWords
Dependencias:	- FRs-02 Tokenizar libros
Precondición:	Tener tokenizados todos los libros
Descripción:	El sistema contara con las palabras que son: Conectores, artículos, pronombres, preposiciones, conjunciones, Interjección.
Secuencia normal:	1- Comparar las palabras StopWords y las tokenizadas 2- Las coincidencias deben ser eliminadas
Poscondición:	Almacenar las palabras que no se encuentran en coincidencia
Excepciones:	N/A
Comentarios:	Los stopwords son aquellas palabras que no aplican en la búsqueda del sistema, como lo son Artículos, pronombres, preposiciones, conjunciones, Interjección

Tabla 4: Tabla RFs-04

FRs-04	Eliminación de mayúsculas
Dependencias:	- FRs-03 Eliminación de StopWords
Precondición:	Tener guardadas las palabras las cuales ya no tienen los StopWords
Descripción:	Se deben poner todas las palabras en minúscula
Secuencia normal:	1- Buscar en las palabras aquellas que tienen mayúsculas y volverla minuscula
Poscondición:	Almacenamiento de las palabras tratadas
Excepciones:	N/A
Comentarios:	N/A

Tabla 5: Tabla RFs-05

FRs-05	Eliminación de caracteres especiales
Dependencias:	- RFs-04 Eliminación de mayúsculas
Precondición:	El sistema debe tener precargadoas todos los caractares especiales
Descripción:	El sistema deberá eliminar los caracteres especiales de las palabras
Secuencia normal:	1- Buscar en las palabras caracteres especiales 2- Si encontró un carácter especial reemplazar por un espacio vacio (" ")
Poscondición:	Almacenar las nuevas palabras sin caracteres especiales
Excepciones:	Las palabras que no tienen caracteres especiales no se les realiza ningún proceso
Comentarios:	N/A

Tabla 6: Tabla RFs-06

FRs-06	Transformar palabras
Dependencias:	- FRs-05 Eliminación de caracteres especiales
Precondición:	Deben estar guardadas las palabras claves para la búsqueda
Descripción:	El sistema debe convertir las palabras tratadas a singular para luego ser almacenadas
Secuencia normal:	1- Buscar las palabras tratadas y singularizar las que se encuentran en plural
Poscondición:	Almacenar las palabras claves para búsqueda
Excepciones:	Las palabras que ya se encuentran en singular no se le debe realizar ningún proceso
Comentarios:	N/A

Tabla 7: Tabla RFs-07

FRs-07	Ordenar palabras alfabéticamente
Dependencias:	- FRs-06 Transformar palabras
Precondición:	Deben estar guardadas las palabras claves para la búsqueda
Descripción:	El sistema organizará las palabras alfabéticamente
Secuencia normal:	1- Se organizan alfabéticamente las palabras
Poscondición:	Se guardan las palabras ordenadas alfabéticamente
Excepciones:	N/A
Comentarios:	N/A

Tabla 8: Tabla RFs-08

FRs-08	Eliminación de palabras repetidas
Dependencias:	- FRs-07 Ordenar palabras alfabéticamente
Precondición:	Tener guardadas las palabras ordenadas alfabéticamente
Descripción:	Se deben eliminar las palabras que se repiten mas de una vez y dejar solamente un vez la palabra
Secuencia normal:	1- Verificar en las palabras aquellas que se encuentran repetidas y eliminarlas
Poscondición:	Guardar las palabras que ya no están repetidas
Excepciones:	N/A
Comentarios:	N/A

Tabla 9: Tabla RFs-09

FRs-09	Calcular frecuencia
Dependencias:	- FRs-08 Eliminación de palabras repetidas
Precondición:	Haber hecho el proceso del requerimiento FRs-08
Descripción:	Se debe calcular la frecuencia de aparición de cada palabra en el libro
Secuencia normal:	1- Escojer cada palabra y determinar cuál es la frecuencia en el libro
Poscondición:	Guardar la palabra la frecuencia y el libro al que pertenece
Excepciones:	N/A
Comentarios:	En el proceso no se debe perder ninguna palabra y deben aparecer las mismas que se encuentran en el RFs-08

Tabla 10: Tabla RFs-10

FRs-10	Almacenar Datos
Dependencias:	- FRs-09 Calcular frecuencia
Precondición:	Tener las palabras separadas con su frecuencia y relacionado con el nombre del libro
Descripción:	El sistema procederá a guardar los datos que se obtuvieron en el RFs-09, los títulos de cada texto y un fragmento del libro en una base de datos
Secuencia normal:	1- Realizar el insertado de los títulos en la base de datos 2- Realizar el insertado de los datos del RFs-09
Poscondición:	Todos los datos deben estar en la base de datos
Excepciones:	N/A
Comentarios:	En el proceso se insertado en la base de datos no se debe perder ningún solo dato y debe ser ingresada en el mismo orden en el que se encuentra

Tabla 11: Tabla RFs-11

FRs-11	Consultar información
Dependencias:	- FRs-10 Almacenar Datos
Precondición:	Tener la información guardada perfectamente en la base de datos del sistema
Descripción:	Permitir que el sistema recupere la información con respecto a una petición de búsqueda
Secuencia normal:	1- Abrir la página web del sistema 2- Digitar el query que se desea buscar 3- Dar clic sobre el botón de "buscar" 4- Petición a la base de datos para obtener coincidencias de palabras
Poscondición:	Mostrar el resultado de la búsqueda en el ranking de aparición junto con el nombre del libro, el fragmento de texto y un link en el título para abrir el texto completo, al final mostrar un botón para regresar
Excepciones:	Si la petición el query ingresado no se encuentra en la base de datos se debe mostrar un resultado vacío
Comentarios:	-No se debe poder realizar una búsqueda con el espacio vacío del query

Tabla 12: Tabla RFs-12

FRs-12	Consultar Libro
Dependencias:	- FRs-11 Consultar información
Precondición:	Se debe tener cargada la web con el resultado de la búsqueda
Descripción:	El sistema debe mostrar el texto del libro al realizar clic en el link del libro
Secuencia normal:	1- Se dará un clic sobre el link del libro del cual se desea obtener el texto
Poscondición:	Cargar la página web con todo el texto del libro
Excepciones:	N/A
Comentarios:	Para abrir el texto del libro solo se podrá dándole clic sobre el link

### 3.2 Requerimientos no funcionales

Tabla 13: Tabla RNFs-01

ID:	RNFs-01
Nombre:	Eficiencia
Tipo:	Atributo de calidad
Prioridad:	Alta
Descripción de RNFs	
Al momento de realizar una búsqueda con un query, el sistema debe ser capaz de mostrar con rapidez el ranking de los libros en donde se encuentra dicho query	
Criterios de aceptación	
El sistema debe mostrarar el ranking de los libros en un tiempo menor a 2 segundos	

Tabla 14: Tabla RNFs-02

ID:	RNFs-02
Nombre:	Eficiencia
Tipo:	Atributo de calidad
Prioridad:	Alta
Descripción de RNFs	
El sistema debe guardar todos los datos generados por el script en la base de datos	
Criterios de aceptación	
El sistema debe realizar el insertado de los datos en un tiempo promedio de 2 segundos por cada 500 kilobytes	

Tabla 15: Tabla RNFs-03

ID:	RNFs-03
Nombre:	Confiabilidad
Tipo:	atributo de calidad
Prioridad:	Alta
Descripción de RNFs	
El sistema debe ser capaz de realizar un filtro para solo recibir los libros con extensión de texto	
Criterios de aceptación	
El sistema debe admitir el 100% de los libros con extensión .txt	

Tabla 16: Tabla RNFs-04

ID:	RNFs-04
Nombre:	Capacidad
Tipo:	Atributo de calidad
Prioridad:	Alta
Descripción de RNFs	
El sistema debe ser capaz de recibir cualquier cantidad de textos de cualquier tamaño	
Criterios de aceptación	
Al haber una carga excesiva de libros en la carpeta "libros" el sistema debe estar en capacidad de realizar el proceso para el 100%	

Tabla 17: Tabla RNFs-05

ID:	RNFs-05
Nombre:	Escalabilidad
Tipo:	Atributo de calidad
Prioridad:	Alta
Descripción de RNFs	
El sistema debe soportar una gran cantidad de libros y debe aun mantener el servicio del mismo	
Criterios de aceptación	
Al haber bastantes libros en el sistema este debe mantenerse operando el 100% de la veces	

Tabla 18: Tabla RNFs-06

ID:	RNFs-06
Nombre:	Extensibilidad
Tipo:	Atributo de calidad
Prioridad:	Alta
Descripción de RNFs	
El sistema debe estar diseñado de manera que en el futuro se puedan realizar cambios	
Criterios de aceptación	
El código del sistema debe estar bien organizado de manera que a la hora de realizar un cambio sea de fácil entendimiento y sea más sencillo añadir nuevas mejoras o cambios	

Tabla 19: Tabla RNFs-07

ID:	RNFs-07
Nombre:	Usabilidad
Tipo:	Atributo de calidad
Prioridad:	Alta
Descripción de RNFs	
El sistema debe tener una interfaz gráfica de tal forma que el usuario esté cómodo al momento de usarla y pueda usar la página fácilmente.	
Criterios de aceptación	
Los usuarios pueden saber la ubicación de las características del sistema de manera fácil, para no tener confusiones en su uso.	

## 4 Casos de uso

Figura 1: Diagrama caso de uso cliente

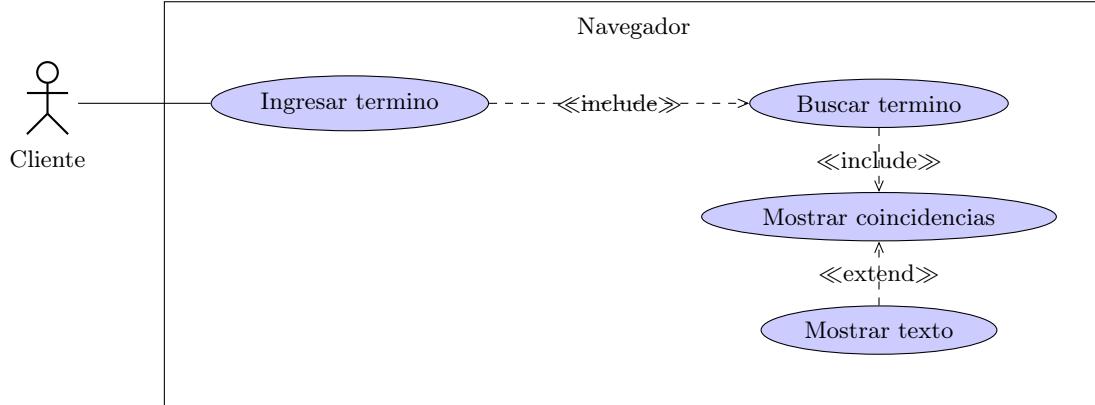
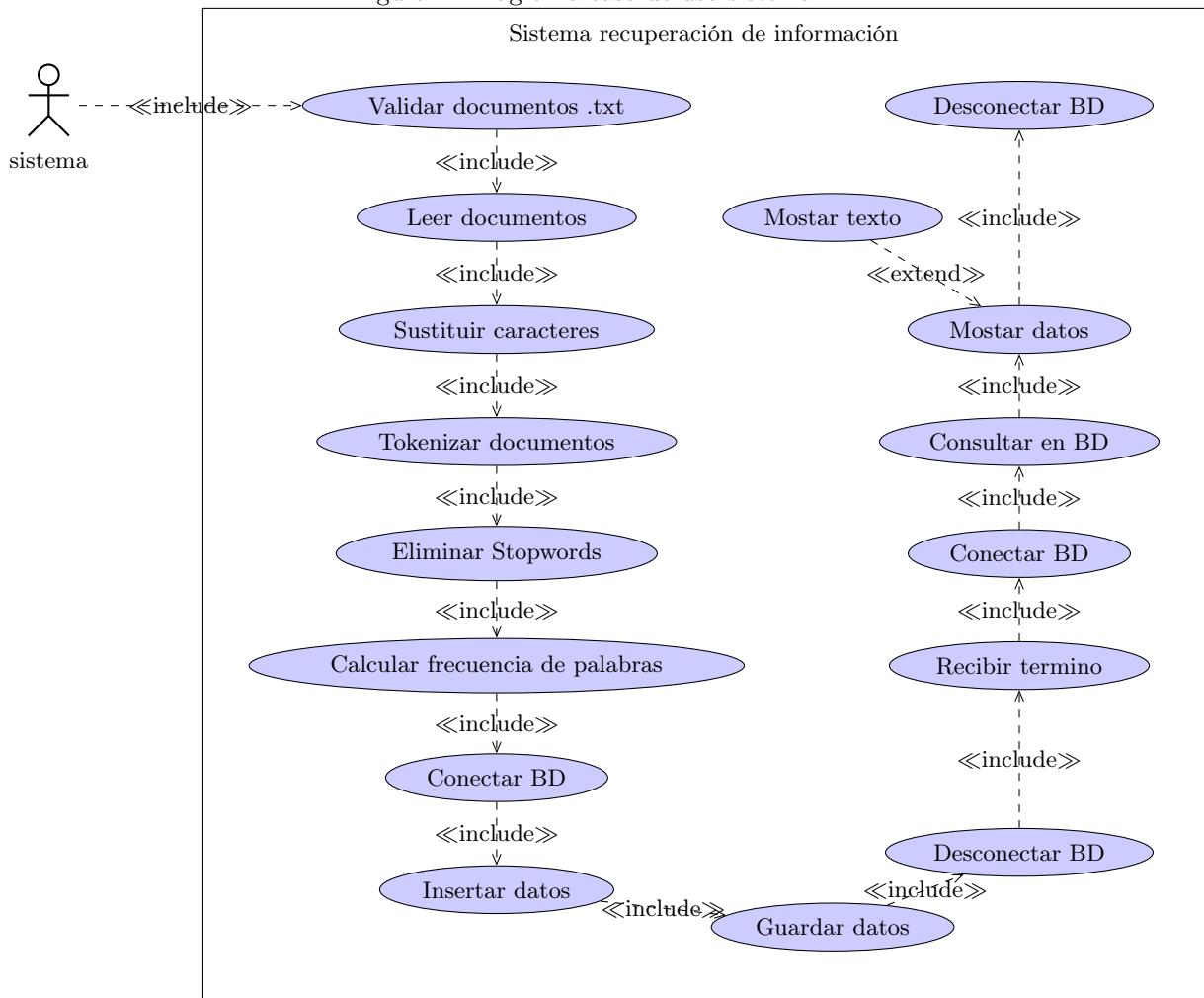
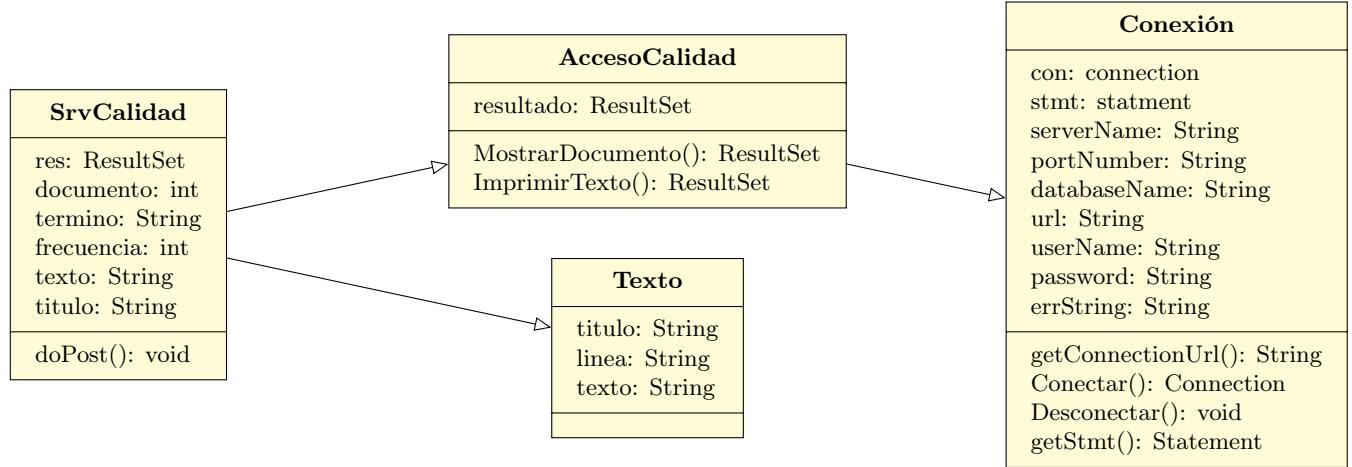


Figura 2: Diagrama caso de uso sistema



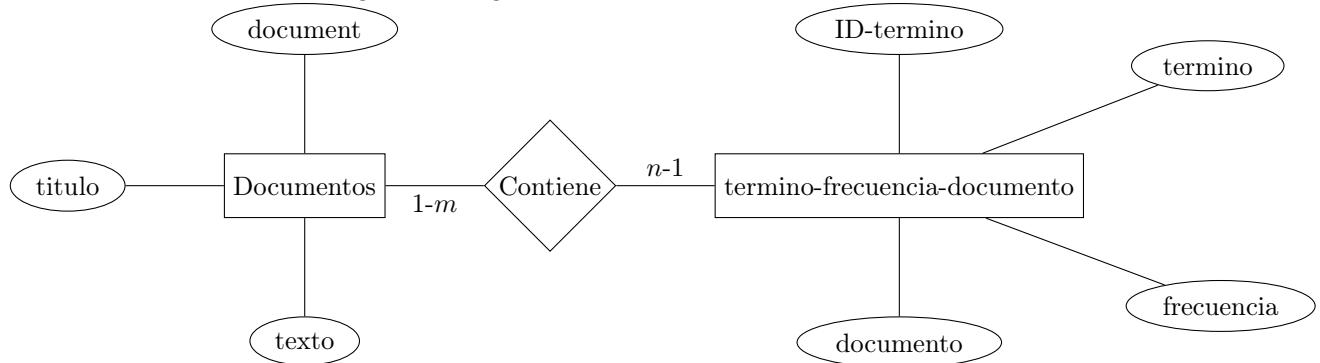
## 5 Diagrama de Clases

Figura 3: Diagrama de Clases



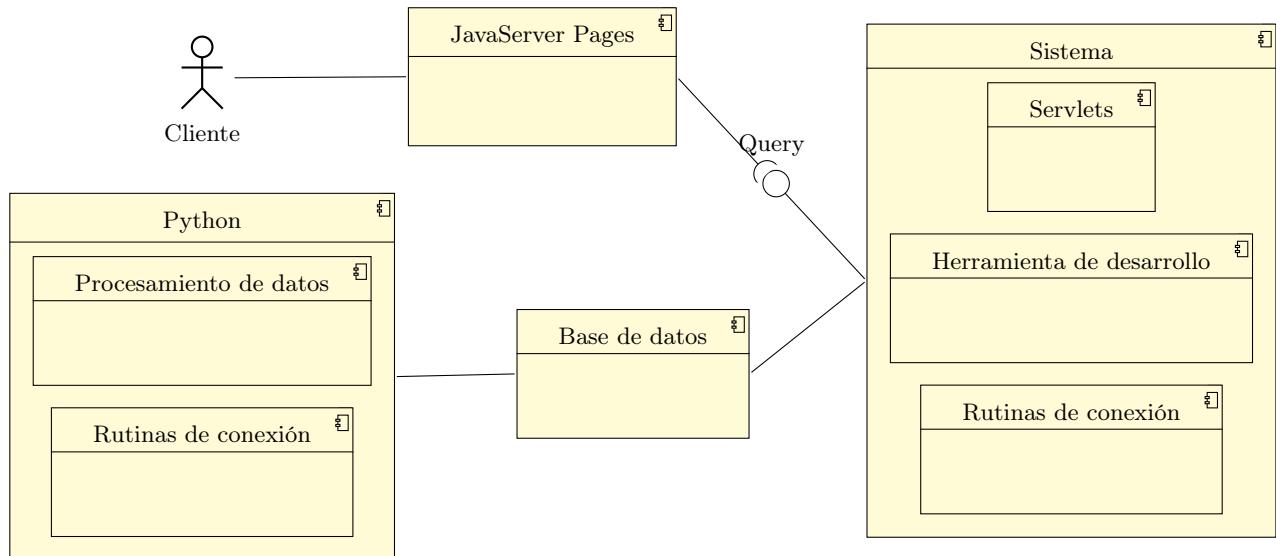
## 6 Modelo entidad - relación

Figura 4: Diagrama modelo entidad - relación



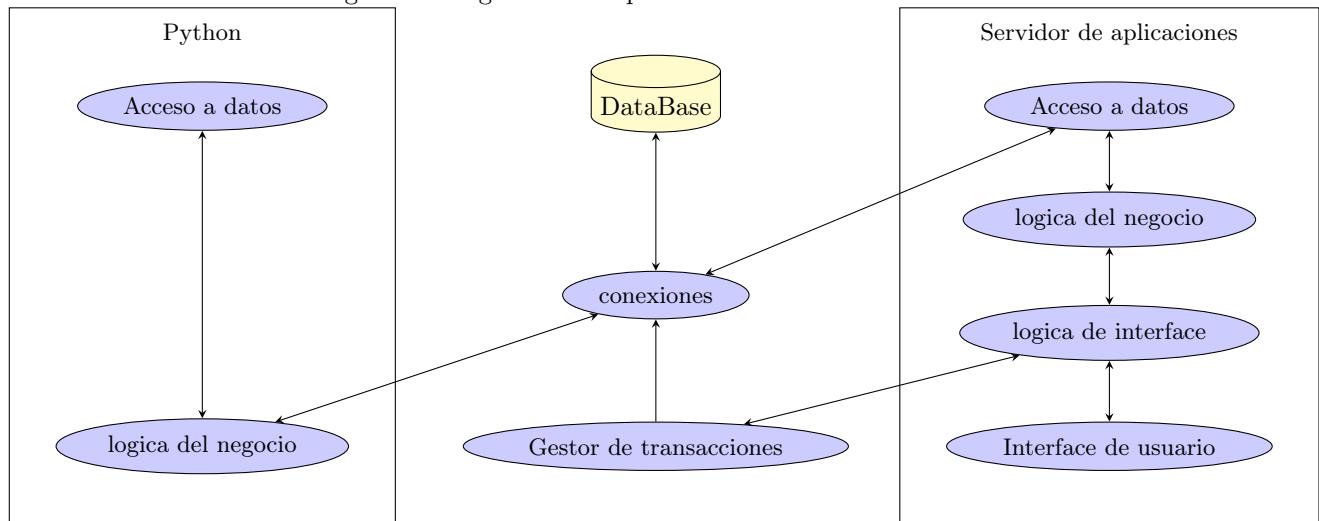
## 7 Diagrama de componentes

Figura 5: Diagrama de componentes



## 8 Diseño de la arquitectura del sistema de recuperación de información

Figura 6: Diagrama de arquitectura del sistema



## 9 Metodología de desarrollo

### Metodología Extreme programming

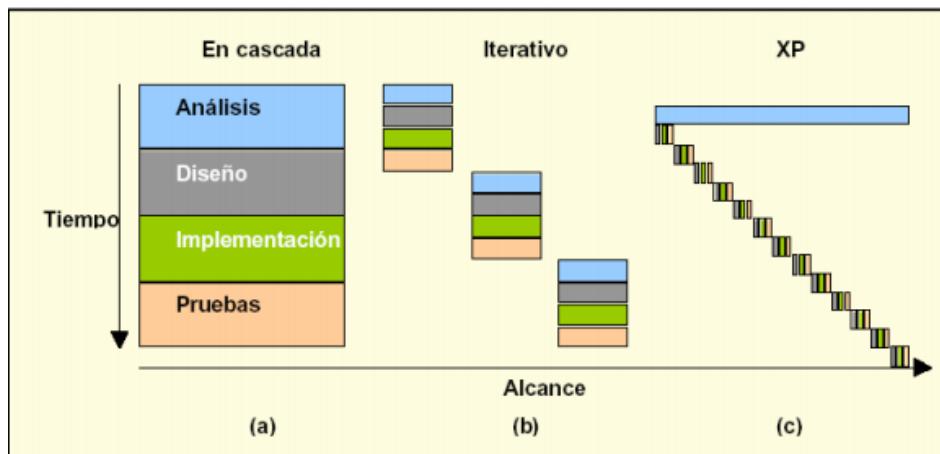
En el desarrollo del software para la recuperación e información es esencial jactarse de una metodología para su construcción a través de su planteamiento, en este caso la metodología escogida por su característica de compactación a través de su desarrollo es la programación extrema. Ya que consta de fases de desarrollo que aplica muy bien a la manera de trabajo de este equipo por medio de estructura organizacional.

Una de las principales características de la apropiación de esta metodología es la característica que aplica al equipo de trabajo, ya que está vinculada a proyectos de corto y mediano plazo y se ajusta a cambios ya que es un modelo adaptativo. Otro punto clave de la metodología es propósito de desarrollo, ya que permite su construcción de forma que su arquitectura, diseño y codificación permitan incorporar modificaciones sin demasiado impacto en la calidad del mismo.

El desarrollo XP brinda el espacio y la visualización de entregas tempranas para fijar un entorno futuro de entrega final, con lo que se busca una aceptación por parte del cliente con el fin de mostrar resultados y confianza. Buscando la aceptación y que cubra con sus expectativas y necesidades.

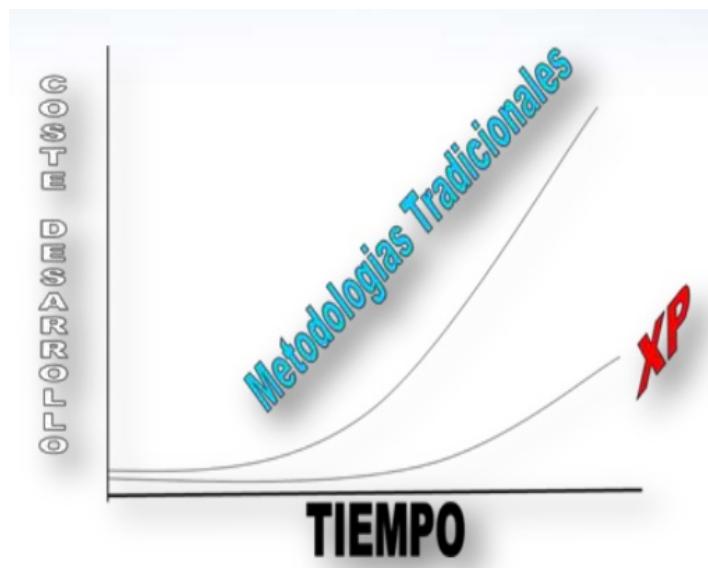
Algunas condiciones para optar por programación extrema son:

1. Interés sincero por todas las partes en que el proyecto tenga éxito.
2. El equipo de trabajo es pequeño.
3. El equipo dispone de una formación elevada y capacidad de aprender.



### Objetivos de la metodología

1. La satisfacción de la parte final (Cliente)
2. Potenciar trabajo en equipo
3. Minimizar riesgo actuando sobre las variables del proyecto:
  - Tiempo
  - Calidad
  - Alcance



### Exploración

El principal proyecto requerido para el curso se trata de un sistema de información el cual debe traer o recuperar información relevante a un libro previamente almacenado, con el fin de que al ingresar un query de búsqueda retorne el libro que tenga más relevancia con la palabra buscada, como una funcionalidad debe mostrar parte del texto al retornar la búsqueda de información para denotar una idea del texto analizado.

### Fases

#### 1. Planificación

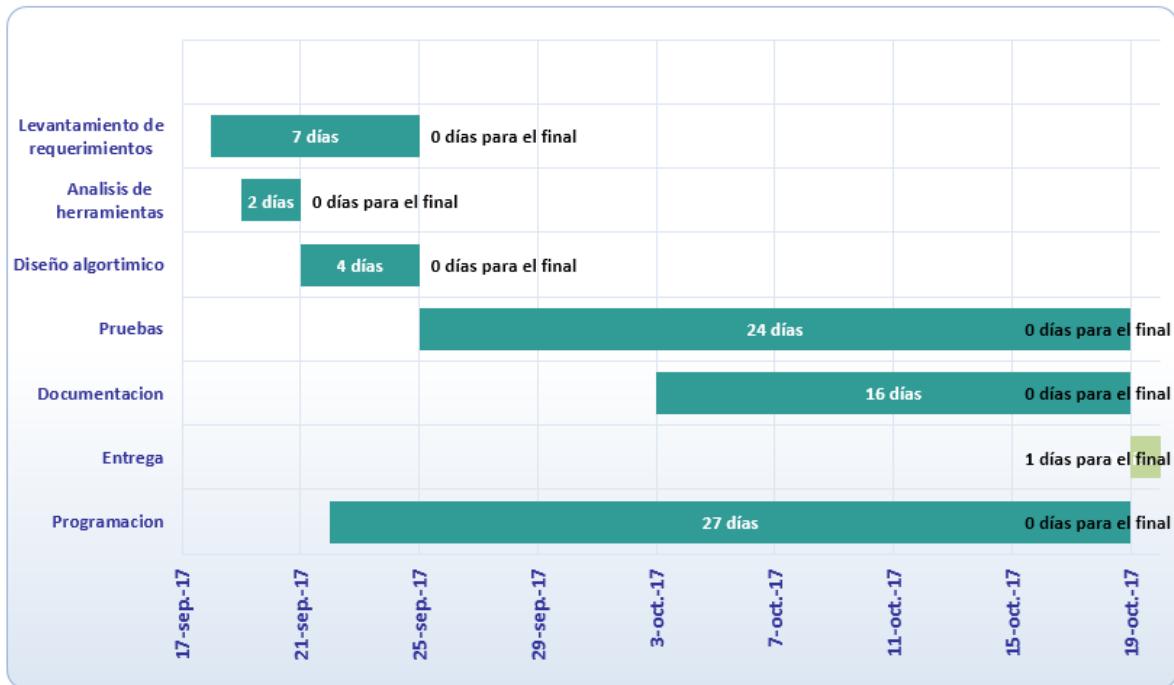
Como principal acercamiento a la idea de proyecto se evaluó la integración del sistema mediante un modelo vista controlador, de manera en que la interface de los requerimientos y la lógica del negocio pudiera estar estructurado de una manera más ordenada para con ello brindar la información de manera amena con el fin de posicionar el proyecto entre los demás. Se opta por entregas funcionales a medida que el sistema avanza, con el fin de tener clara la misión del proyecto y así mitigar los riesgos de abandono de metas en este.

#### Cronograma segundo corte

### recuperación de información

Tareas	Fecha inicio prevista	Dias trabajados	Fecha final prevista	Situación	Dias para el final
Programacion	22-sep.-17	27	19-oct.-17	En curso	0
Entrega	19-oct.-17	0	20-oct.-17	En curso	1
Documentacion	3-oct.-17	16	19-oct.-17	En curso	0
Pruebas	25-sep.-17	24	19-oct.-17	En curso	0
Diseño algoritmico	21-sep.-17	4	25-sep.-17	Terminado	0
Analisis de herramientas	19-sep.-17	2	21-sep.-17	Terminado	0
Levantamiento de requerimientos	18-sep.-17	7	25-sep.-17	Terminado	0

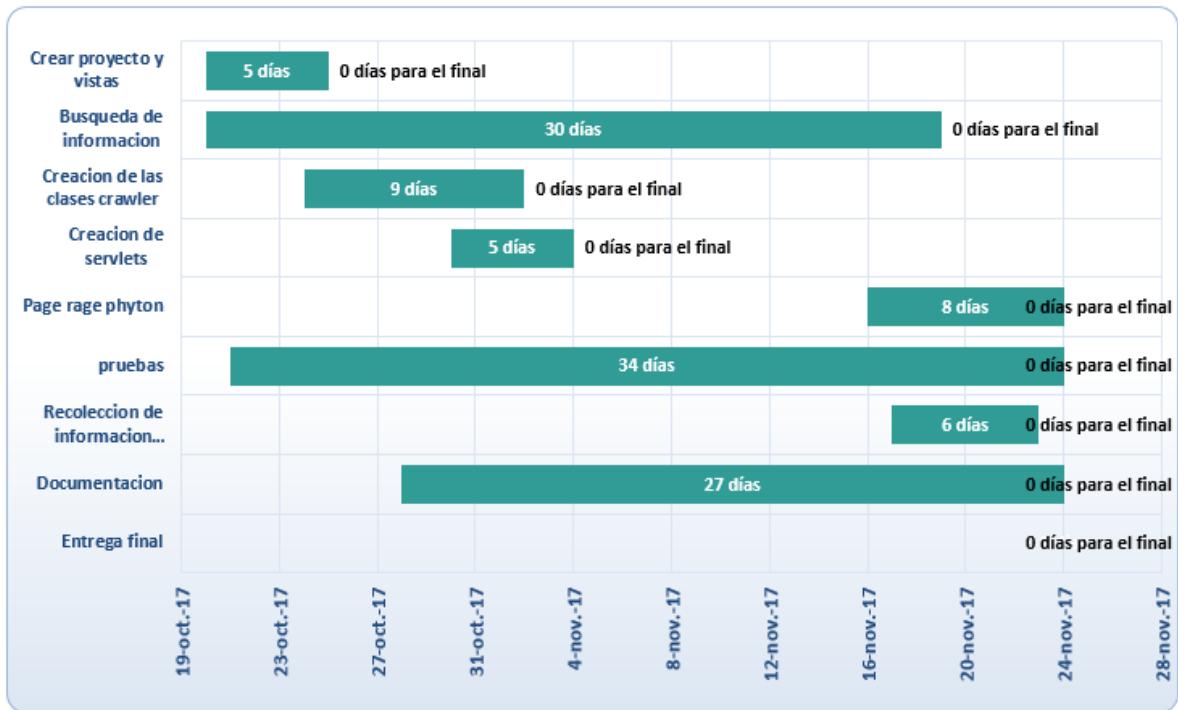
Completado Pendiente



Cronograma tercer corte

### recuperación de información

Tareas	Fecha inicio prevista	Dias trabajados	Fecha final prevista	Situación	Dias para el final
Entrega final	24-nov.-17	0	24-nov.-17	Terminado	0
Documentacion	28-oct.-17	27	24-nov.-17	Terminado	0
Recolección de información crawler	17-nov.-17	6	23-nov.-17	Terminado	0
pruebas	21-oct.-17	34	24-nov.-17	Terminado	0
Page rage python	16-nov.-17	8	24-nov.-17	Terminado	0
Creación de servlets	30-oct.-17	5	4-nov.-17	Terminado	0
Creación de las clases crawler	24-oct.-17	9	2-nov.-17	Terminado	0
Busqueda de información	20-oct.-17	30	19-nov.-17	Terminado	0
Crear proyecto y vistas	20-oct.-17	5	25-oct.-17	Terminado	0



## 2. Diseño

Como parte del diseño se propuso una herramienta web en donde se evidencie la funcionalidad obtenida a partir de la planeación del desarrollo.

Enfatizando su simplicidad como ítem de construcción nos permitirá tener el sistema más rápidamente en comparación de un sistema complejo por este media la metodología nos propone implementar el diseño más simple que funcione y a partir de este, comenzar con el desarrollo que implante los requerimientos visualizando la meta de sus funciones para con ello dar una buena funcionalidad al proyecto en gestión.

## 3. Desarrollo

Por parte del desarrollo del sistema de información nuestra solución vinculaba la programación por pares con el fin de tener claridad de las soluciones planteadas por parte de la lógica del negocio. Otro punto clave en la construcción del sistema es la eficacia de detección de errores y sus posibles soluciones a parte del apoyo con el grupo de trabajo. Evidenciando ahorro de tiempo que puedo ser visto como costo en el desarrollo del proyecto.

Una de las ventajas de la programación a pares es la vinculación del equipo de trabajo con el aplicativo ya que como todos hacen parte de la codificación del mismo, tendrán más claridad de la construcción y manejo de componentes del sistema de información. Teniendo como plus que el equipo de trabajo aprendan y afiancen sus capacidades de trabajar en grupo.

## 4. Pruebas

En término de pruebas se adoptó el testeo conforme se avanzaba en su desarrollo con pruebas unitarias. Partiendo de la revisión de código conforme se iba creando funcionalidades, esto con el fin de tener claridad de su buen desarrollo y seguir un camino firme en términos de construcción de software permitiendo comprobar que funcione la propiedad colectiva del código.

Cuando se encuentra un bug o falla de código este se corrige inmediatamente para dar cabida a una siguiente línea del algoritmo, permitiendo la confiabilidad y robustez del trabajo realizado

## Tarea de ingeniería

Tabla 20: Tabla tarea 01

Tarea	
Numero de tarea: 01	Numero de historia: 01
Nombre de tarea: estipular un sistema de recuperación de información	
	Puntos estimados:
Tipo tarea: Desarrollo	Escoger formato de archivos a tratar Elegir herramientas de desarrollo
Fecha Inicio: 17/09/2017	Fecha Fin: 20/09/2017
Programador responsable: David R, Gabriel J, Leonardo O	
Descripción: Se debe tener claridad y una visión del desarrollo del aplicativo	

Tabla 21: Tabla tarea 02

Tarea	
Numero de tarea: 02	Numero de historia: 01
Nombre de tarea: Funcionalidad vista	
	Puntos estimados:
Tipo tarea: Corrección	Se debe mostrar un indicio del documento Elegir herramientas de desarrollo
Fecha Inicio: 14/10/2017	Fecha Fin: 15/10/2017
Programador responsable: David R, Gabriel J, Leonardo O	
Se debe visualizar parte del documento al momento de retornar la búsqueda que se le da en el buscador	

## **Historias de usuario**

Tabla 22: Historia de usuario 1

Historia de usuario	
Numero: 1	Nombre: Consumo de gran cantidad de archivos
Usuario: Stakeholder	
Modificación de Historia Numero: 0	Iteración asignada:2
Prioridad en negocio: Alta	Puntos estimados: Cargar 200 libros en formato .txt
Riesgo en desarrollo: Media	Puntos reales: Se han cargado 2400 libros en formato .txt
Descripción: Al momento de almacenar los libros debe de tener una buena capacidad para que el sistema se alimente de palabras para su búsqueda y se vea bien referenciado, con esto su funcionalidad será bien vista.	

Tabla 23: Historia de usuario 12

Historia de usuario	
Numero: 2	Nombre: Agilidad al retornar información
Usuario: Stakeholder	
Modificación de Historia Numero: 0	Iteración asignada:2
Prioridad en negocio: Alta	Puntos estimados: Optimización al devolver el texto indexado por la palabra buscada
Riesgo en desarrollo: Alta	Puntos reales: Buscar la mejor solución de optimización al mostrar la información buscada
Descripción: El sistema retorna la información pero la espera es demasiado larga, lo que quiero es que se agilice los procesos para que al momento de darle clic a el vínculo del libro, la información se muestre en el menor tiempo posible.	

## 10 Propuesta de la solución Computacional

Para la propuesta de solución al problema de la recuperación de la información se va a utilizar un servidor de aplicaciones, una base de datos y dos lenguajes de programación con los respectivos programas que se van a usar:

### 1. Python 3.6(Spyder)

En este lenguaje se desarrollará un script el cual leerá cada texto tomando las palabras y en base a eso creará dos matrices una que contenga la palabra, su frecuencia y su número de documento y otra matriz que contenga el número de documento, su título y un segmento de su texto, por ultimo estas matrices se guardarán a una base de datos.

### 2. Java (Netbeans)

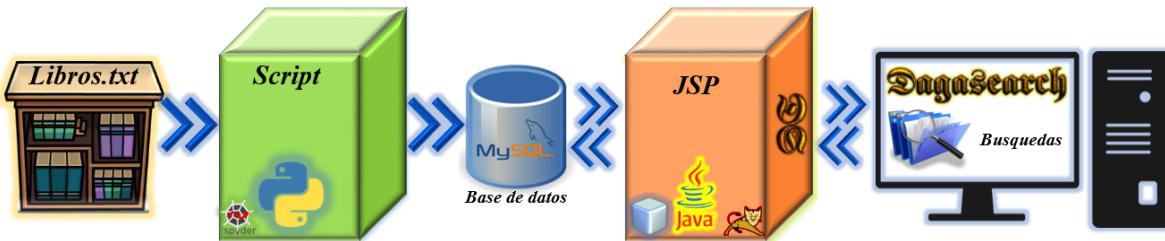
En este lenguaje se desarrollará un MVC (modelo vista controlador) en el cual tendrá un JSP como vista, Servlet como controlador y un conector para la base de datos como modelo, en el JSP se creará la interfaz para consultar una palabra, en los Servlet se llamarán los datos del conector y seguidamente se postearán y por último en el conector se tomarán los datos de la base de datos.

### 3. Apache Tomcat (Servidor Web)

Este servidor funcionará como un contenedor de Servlets y Jsp también nos permitirá la gestión de transacciones, almacenamiento temporal de ficheros y directorios.

### 4. Base de datos(MySQLWorkBench)

En esta base de datos se guardarán las matrices creadas por el script de Python en dos tablas llamadas "documentos" y "termino\_frecuencia\_documento".



## 11 Cálculo de la complejidad computacional usado para la recuperación de documentos

```
while (res.next()).....Log(n)
____termino = res.getString("termino").....1*n
____frecuencia = res.getInt("frecuencia").....1*n
____titulo =res.getString("titulo").....1*n
____texto = res.getString("texto").....1*n
```

$$Log(n) + 1 * n + 1 * n + 1 * n + 1 * n$$

$$Log(n) + 4n$$

$$Log(n) + n$$

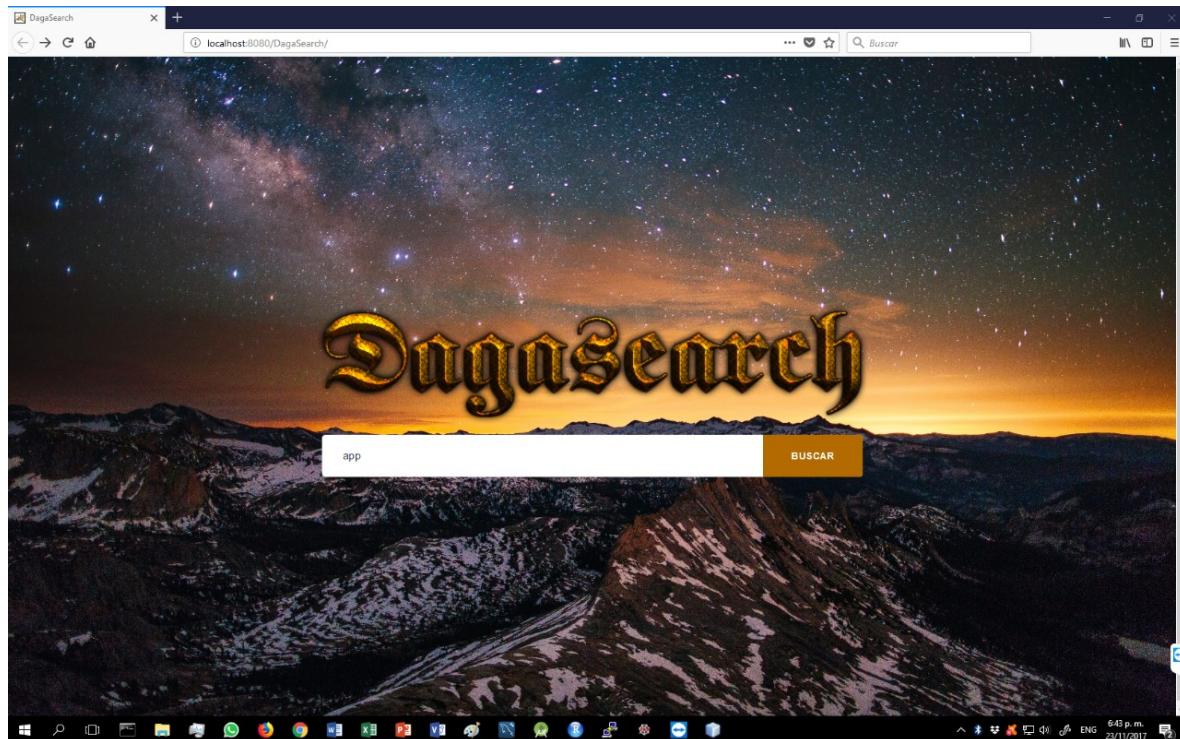
$$Log(n)$$

Por lo tanto la complejidad es  $O(Log(n))$

## 12 Manual de Usuario

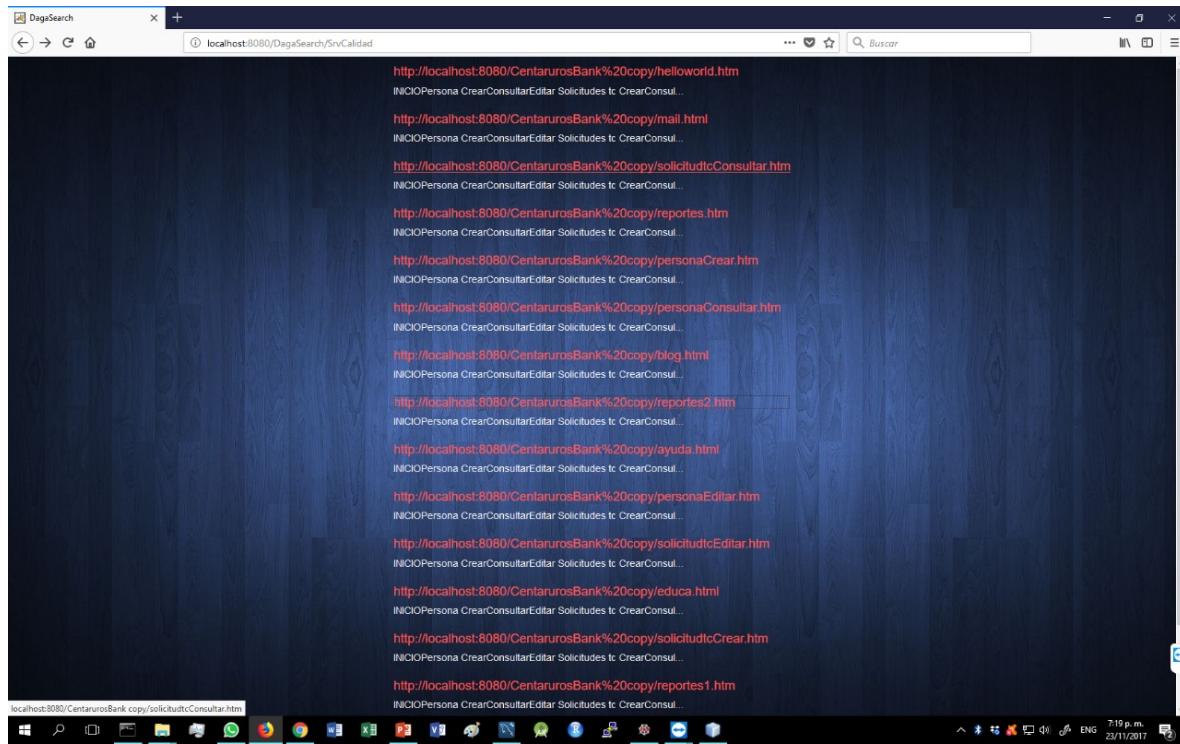
### Pantalla de ingreso de búsqueda

Al momento de abrir el buscador lanzara una ventana en donde dispone de una barra de búsqueda en donde se pondrá la palabra a buscar para posteriormente buscar la información solicitada  
Como ejemplo se busca información "app" en el buscador.



## Recolección de información buscada

Al darle click al botón buscar se abrirá otra ventana en donde mostrara las coincidencias con la palabra ingresada en la barra de búsqueda. En esta ventana podemos darle click a el link que trae para poder visualizar su contenido completo en esta ventana también podemos ver una corta descripción del texto relacionado con la búsqueda.



## Visualización de la búsqueda

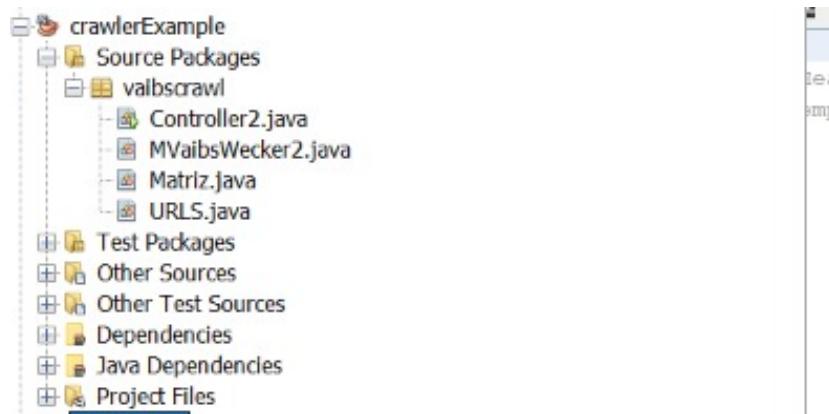
Cuando se lista la información relacionada con el query de búsqueda el usuario podrá seleccionar cualquier link de su interés dándole click sobre el mismo y será redirigido a ese link posteriormente.

## 13 Manual de Aministrador

### Crawler

Dentro del proyecto crawlerexample se tiene la estructura de extracción de información del dominio universidad nacional, con el cual se hará el análisis pertinente del page Range.

En este procedimiento se creará una matriz de adyacencia en donde estará estipulado el nodo padre y sus hijos (páginas secundarias).



### Proyecto de trabajo Dagasearch

En la estructura inicial contamos con 3 clases principales dentro del proyecto llamado Dagasearch, Con esta se hará la vinculación entre la capa de vista y la capa de modelo.

#### Conexión base de datos

Clase conexión que es la encargada de comunicar el entorno de desarrollo con la fuente de datos, en este caso la base de datos mysql. Se envían parámetros como usuario, contraseña, puerto, dirección y así poder vincular los sistemas.

```
public class Conexion {  
  
    protected Connection con;  
    protected Statement stat;  
    private String serverName= "localhost";  
    private String portNumber = "3306";  
    private String databaseName= "matriz";  
    private String url = "jdbc:mysql://localhost:3306/" + databaseName;  
    private String userName = "root";  
    private String password = "root"; // Indica al controlador que debe utilizar un cursor de servidor, // lo que permite más de una instrucción activa // en una conexión  
    //private String selectMethod = "cursor";  
    private String errString;  
  
    private String getConnectionUrl()  
    {  
        return url;  
    }  
  
    public Connection Conectar()  
    {  
        con=null;  
        try{  
            Class.forName("org.gjt.mm.mysql.Driver");  
            con = DriverManager.getConnection(getConnectionUrl(),userName,password);  
            stat=con.createStatement();  
            System.out.println("Conectado");  
        }catch(Exception e){  
            errString = "Error Mientras se conectaba a la Base de Datos";  
            System.out.println(errString);  
            return null;  
        }  
        return con;  
    }  
    public void Desconectar()  
    {  
        try{  
            stat.close();  
        }  
    }  
}
```

## Consulta a la base de datos

Esta clase es la encargada de retomar los valores almacenados en la base de datos al momento de pedirlos por la vista en la web, de esta manera se envía un query con el parámetro introducido en el buscador, para así buscar coincidencias y retornar la información solicitada.

```
public class AccesoCalidad extends Conexion{
    private ResultSet resultado;
    public AccesoCalidad()
    {
        Conectar();
    }
    public ResultSet Listado() throws Exception
    {
        try{
            getStmt();
            resultado= stmt.executeQuery("SELECT * FROM termino_frecuencia_documento order by frecuencia desc");
            return resultado;
        } catch (Exception ex){
            System.out.println("SQLException: " + ex.getMessage());
            return null;
        }
    }

    public ResultSet MostrarDocumento(String Término) throws Exception
    {
        try{
            getStmt();
            resultado= stmt.executeQuery("SELECT documento,termino,frecuencia,título,Texto FROM termino_frecuencia_documento INNER JOIN documentos ON termino_frecuencia_documento.documento = documentos.id_documento WHERE termino LIKE '%"+Término+"%'");
            return resultado;
        } catch (Exception ex){
            System.out.println("SQLException: " + ex.getMessage());
            return null;
        }
    }

    public ResultSet ImprimirTexto(String documento) throws Exception
    {
        try{
            getStmt();
            resultado= stmt.executeQuery("SELECT * FROM documentos where documento LIKE "+documento+"");
            return resultado;
        } catch (Exception ex){
            System.out.println("SQLException: " + ex.getMessage());
            return null;
        }
    }
}
```

## Método Dopost

Es el método encargado de revisar las coincidencias entre el valor enviado por el buscador y la información almacenada en la base de datos, retornando los links con la información pertinente.

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
    response.setContentType("text/html;charset=ISO-8859-1");
    PrintWriter out = response.getWriter();
    try {
        ResultSet res;
        AccesoCalidad calidad = new AccesoCalidad();
        texto tex =new texto();
        int documento = 0;
        String termino = "";
        int frecuencia=0;
        String texto="";
        String titulo="";
        res = calidad.Listado();
        if(request.getParameter("termino")!= ""){
            out.println("<html>");
            out.println("<head> <meta charset=\"ISO-8859-1\">");
            out.println("<title>Búsqueda</title>");
            out.println("<link rel=\"stylesheet\" href=\"//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css\"/> <link rel=\"stylesheet\" href=\"css/style.css\"");
            out.println("</head>");
            out.println("<body background='img/fondo.jpg'>");
            out.println("<table align=center border=0>");
            //out.println("<tr style='color:#FFFFFF' align=center ><td>Término</td><td>Título documento</td><td>Frecuencia</td>");
            res = calidad.MostrarDocumento(request.getParameter("termino"));

            while (res.next()) {
                termino = res.getString("termino");
                frecuencia = res.getInt("frecuencia");
                titulo = res.getString("título");
                texto = res.getString("Texto");
                //out.println("<tr style='color:#FFFFFF' align=center > <td> <form method='post' action='<%=texto%>'><button style='color:#FA5858' class='btn btn-link'> <input type='submit value='Borrar' style='color:#FA5858'></td> <td> <a href='<%=titulo%>' style='color:#FA5858'><h4>" + titulo + "</h4></a><h5 style='color:#FFFFFF'>" + texto + "</h5> </td> </tr>");
                out.println("<tr style='color:#FFFFFF' align=left > <td> <a href='<%=titulo%>' style='color:#FA5858'><h4>" + titulo + "</h4></a><h5 style='color:#FFFFFF'>" + texto + "</h5> </td> </tr>");
            }
            out.println("</table>");
            out.println("<center><a href='index.jsp'><img src='img/volver.png' width='15%' height='13%' /> </a></center> </body>");
        }
    } catch (Exception ex){
        System.out.println("SQLException: " + ex.getMessage());
    }
}
```

## Subir el servicio

Al momento que el algoritmo este implementado se procederá a subir al servidor (Tomcat este caso) el proyecto web asociado para tener acceso al directamente desde el buscador local.

The screenshot shows the Tomcat Manager Application interface. At the top, there is a message bar with "Mensaje: OK". Below it is a navigation bar with tabs: "Gestor" (selected), "Listar Aplicaciones", "Ayuda HTML de Gestor", "Ayuda de Gestor", and "Estado de Servidor". The main content area is titled "Aplicaciones" and lists several applications:

Trayectoria	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado	Welcome to Tomcat	true	0	Arrancar   Parar   Recargar   Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/DagaSearch	Ninguno especificado		true	0	Arrancar   Parar   Recargar   Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/docs	Ninguno especificado	Tomcat Documentation	true	0	Arrancar   Parar   Recargar   Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/examples	Ninguno especificado	Servlet and JSP Examples	true	0	Arrancar   Parar   Recargar   Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/host-manager	Ninguno especificado	Tomcat Host Manager Application	true	0	Arrancar   Parar   Recargar   Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/manager	Ninguno especificado	Tomcat Manager Application	true	1	Arrancar   Parar   Recargar   Replegar Expirar sesiones sin trabajar ≥ 30 minutos

At the bottom, there is a "Desplegar" section with the sub-instruction "Desplegar directorio o archivo WAR localizado en servidor".

## Page Rank python

Por medio de java se genera un texto específico con unos parámetros organizados, con el fin de enviarle ese dato como parámetro a Python, esto para al momento de compilarse se generen los números del page Rank para cada página analizada por el Crawler implementado. Estas entradas son, la página padre, sus hijos y la cadena de datos en donde especifica los vértices que tienen direccionamiento.

```

14 import time
15 inicio=time.time()
16
17 def step(p,s=0.85):#s es una probabilidad
18     n = size
19     v = np.matrix(np.zeros((n,1)))#vector de tamaño n de os
20     inner_product = sum([p[j] for j in dangling_pages.keys()])#suma de los valores de p de dangling
21     for j in range(n):
22         v[j] = s*sum([p[k]/number_out_links[k]
23                     for k in in_links[j]])+s*inner_product/n*(1-s)/n
24     #sum([p[k]/number_out_links[k]for k in in_links[j]])
25     #sumatoria del PageRank[k]/el # de Links que salen[j]
26     return v/np.sum(v)#divide el vector v sobre la suma del mismo y retorna el PR
27
28 def pagerank(s=0.85,tolerance=0.00001):#disminuyéndola para garantizar una tolerancia más apropiada
29     n = size
30     p = np.matrix(np.ones((n,1))/n#vector de tamaño n de 1/n
31     iteration = 1
32     change = 2
33     while change > tolerance:
34         #print ("Iteration: %s" % iteration)
35         new_p = step(p,s)#pageRank inicial
36         change = np.sum(np.abs(p-new_p))#valor absoluto de la resta de p-new_p y se suma los valores
37         #print ("Change: %s" % change)
38         p = new_p#pageRank final
39         iteration += 1
40     return p
41
42 print("---Resultado PageRank---")
43
44 archivo = open("Matriz.txt","r") # nombre del documento abierto

```

```

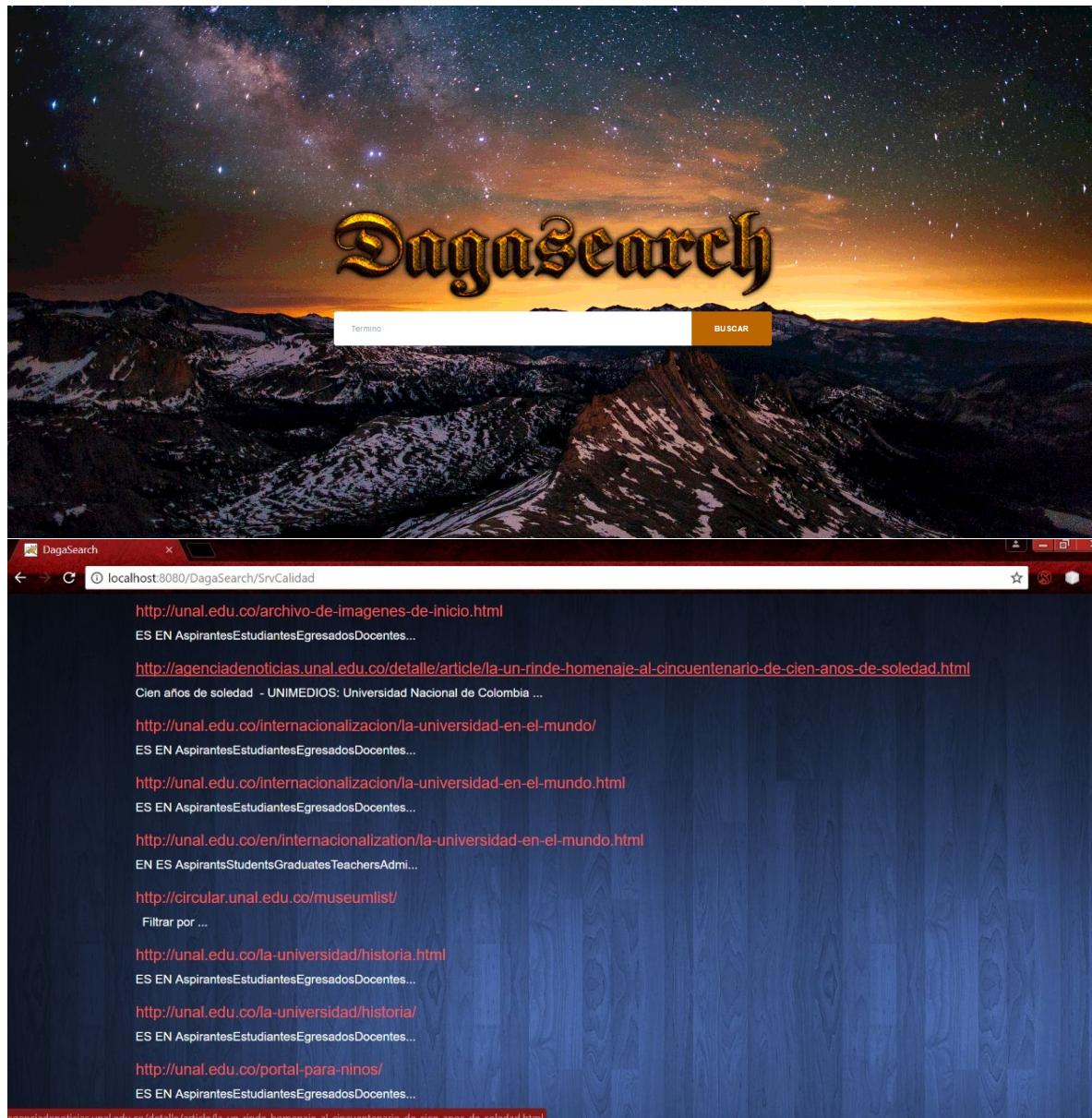
PAGERANK.py', wdir='C:/Users/David/Desktop/crawlerExample')
-->--Resultado PageRank--
[[ 1.20000000e+01  1.19592639e-03]
 [ 2.60000000e+01  1.19592639e-03]
 [ 2.70000000e+01  1.19592639e-03]
 [ 1.51000000e+02  1.19592639e-03]
 [ 1.53000000e+02  1.19592639e-03]
 [ 1.55000000e+02  1.19592639e-03]
 [ 1.58000000e+02  1.19592639e-03]
 [ 1.60000000e+02  1.19592639e-03]
 [ 1.61000000e+02  1.19592639e-03]
 [ 1.62000000e+02  1.19592639e-03]
 [ 1.63000000e+02  1.19592639e-03]
 [ 1.66000000e+02  1.19592639e-03]
 [ 1.68000000e+02  1.19592639e-03]
 [ 1.69000000e+02  1.19592639e-03]
 [ 1.70000000e+02  1.19592639e-03]
 [ 1.54000000e+02  1.23012952e-03]
 [ 9.60000000e+01  1.31365843e-03]
 [ 8.10000000e+01  1.31413717e-03]
 [ 8.20000000e+01  1.31413717e-03]
 [ 8.30000000e+01  1.31413717e-03]
 [ 8.50000000e+01  1.31413717e-03]
 [ 8.60000000e+01  1.31413717e-03]
 [ 8.70000000e+01  1.31413717e-03]
 [ 8.80000000e+01  1.31413717e-03]
 [ 8.90000000e+01  1.31413717e-03]
 [ 9.10000000e+01  1.31413717e-03]
 [ 1.64000000e+02  1.31591521e-03]
 [ 1.57000000e+02  1.31662813e-03]
 [ 9.30000000e+01  1.32832926e-03]

```

## Solución y búsqueda

Por parte de la solución se tiene una salida en donde traerá el link que tenga la mayor consecuencia de palabras relacionadas con la búsqueda, otorgando unos datos más precisos en términos de recolección de información.

Se ingresa el término a buscar



Al momento de dar click en un link de interés se abrirá la url con su contenido original, re direccionando de la base de datos a la web para una experiencia mayor.

The screenshot shows a news article titled "La U.N. rinde homenaje al cincuentenario de Cien años de soledad" (The U.N. pays tribute to the 50th anniversary of 'Cien años de soledad'). The article discusses the translation of the novel into 36 languages and its sales. Below the article are sharing options (Email, Compartir, Imprimir) and a sidebar with other news items from the Agencia de Noticias UN.

## 14 Bibliografia

- <https://www.iseebug.com/crawler4j-example-kickoff-with-crawler4j-crawler4j-with-maven/>
- <http://michaelnielsen.org/blog/using-your-laptop-to-compute-pagerank-for-millions-of-webpages/>
- <https://www.humanlevel.com/diccionario-marketing-online/pagerank-google>