



Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ciencias de la Computación e Informática

Recuperación de Información

Proyecto Programado

Profesor

Diego Villalba

Estudiante / Carnet

Ivannia Alvarado / B10273

Oscar Castro / B11616

Jenny Vásquez / B17016

Diciembre, 2015

Tabla de Contenidos

[Introducción](#)

[Análisis del Problema](#)

[Diseño](#)

[Pasos para replicar el proceso](#)

[Manual de usuario](#)

[Problemas sin resolver](#)

[Características adicionales](#)

1.Introducción

Los motores de búsqueda actualmente forman parte indispensable de nuestras vidas y de ahí que su estudio en un curso de Ciencias de la Computación sea un tema de suma importancia. Es así como este proyecto se trata de crear un motor de búsqueda, que cuente con alguna característica que lo haga particular que lo haga especial.

Para este proyecto se decidió utilizar Scrapy como araña y java como lenguaje de programación. La idea principal se enfocó en hacer una especialización en información sobre medicamentos, ofreciendo además sugerencias con respecto a las búsquedas más recientes, y para finalizar, se ofrece además un sistema de traducción que ayuda a mejorar la consulta.

2. Análisis del Problema

Se decidió abordar el problema de una forma modular, lo que implica la división del mismo en distintas partes que se comunican y complementan, pero actúan de forma autónoma. De esta forma, se tienen las siguientes partes identificadas en la aplicación:

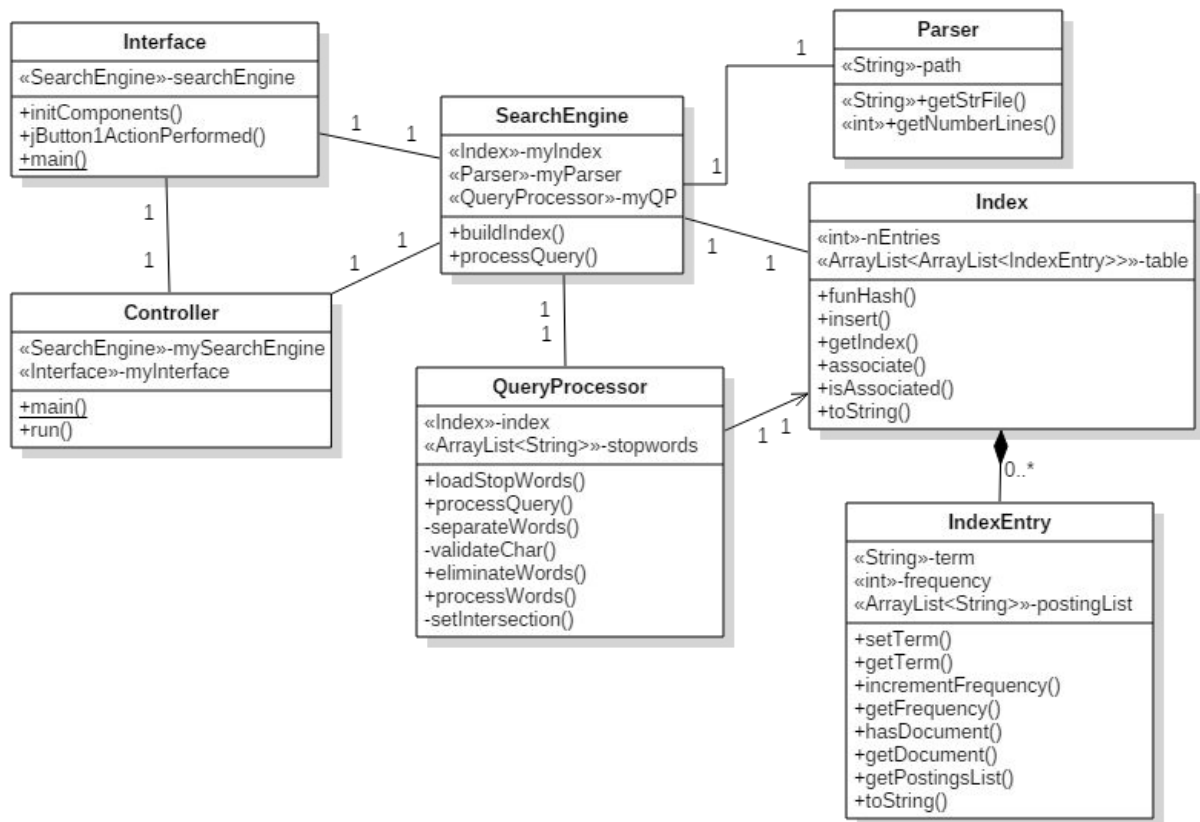
1. *Obtención de la colección:* esta es la parte que se encarga de conseguir la colección de documentos mediante la araña, además de organizarlos en un directorio y hacer la debida conversión al formato soportado por la aplicación.
2. *Procesamiento de archivos, obtención de tokens y obtención de la lista de postings:* esta es la parte que se encarga de tomar el directorio con la información de las páginas obtener los tokens de cada documento y procesarlos hasta obtener la lista de postings. Este será el que se guarde en un archivo en disco, pues sirve de base para construir el índice de búsqueda en la parte siguiente.
3. *Construcción del Índice de búsqueda:* en esta parte se toma el archivo guardado en disco con la lista de postings en formato .txt y se construye el índice de búsqueda en la aplicación. El algoritmo utilizado para la construcción del índice es BSBI y la estructura de datos es una Tabla Hash (Tabla de Dispersión).
4. *Procesamiento de consultas:* en esta parte, se hace el procesamiento de la consulta ingresada por el usuario y se busca mediante el índice de los archivos que satisfagan la misma. También esta parte es la encargada de procesar la respuesta y mostrarla al usuario. Además también se encarga de hacer el ordenamiento de los resultados de acuerdo con la similitud de de cosenos en el modelo del espacio vectorial.
5. *Traductor de consultas:* en esta parte, se realiza el reconocimiento del lenguaje que se utiliza en la consulta. Si la consulta se realiza en un idioma distinto al del dominio, se le ofrece al usuario la posibilidad de realizar la consulta en el dominio de la aplicación, que en nuestro caso corresponde a páginas en inglés, esto con el fin de obtener mejores resultados, si es que hay alguno. La traducción es simple, corresponde al uso de un servicio web ofrecido por Yandex.com, el cual permite traducir frases entre un par de idiomas dados por el usuario. Si el usuario, desea

utilizar esta sugerencia, se procede a traducir su consulta y verificar si se encontró algún documento con la traducción de la misma.

6. *Recomendador de búsquedas recientes*: en esta parte, se carga y posteriormente se almacenan (actualizados) los resultados de las últimas búsquedas, para que así el usuario tenga una idea de lo que se ha buscado más recientemente; claramente este criterio podría mejorarse, pero para efectos del proyecto no es nuestro enfoque. Mantenemos una lista de diez documentos que han sido recientemente recuperados por el buscador, en el caso de que la nueva búsqueda contenga alguno de ellos, la siguiente búsqueda tendrá ese o esos documentos de primero en la lista de sugerencias.

3. Diseño

La siguiente imagen muestra el diseño de clases de la aplicación y la relación entre las mismas.

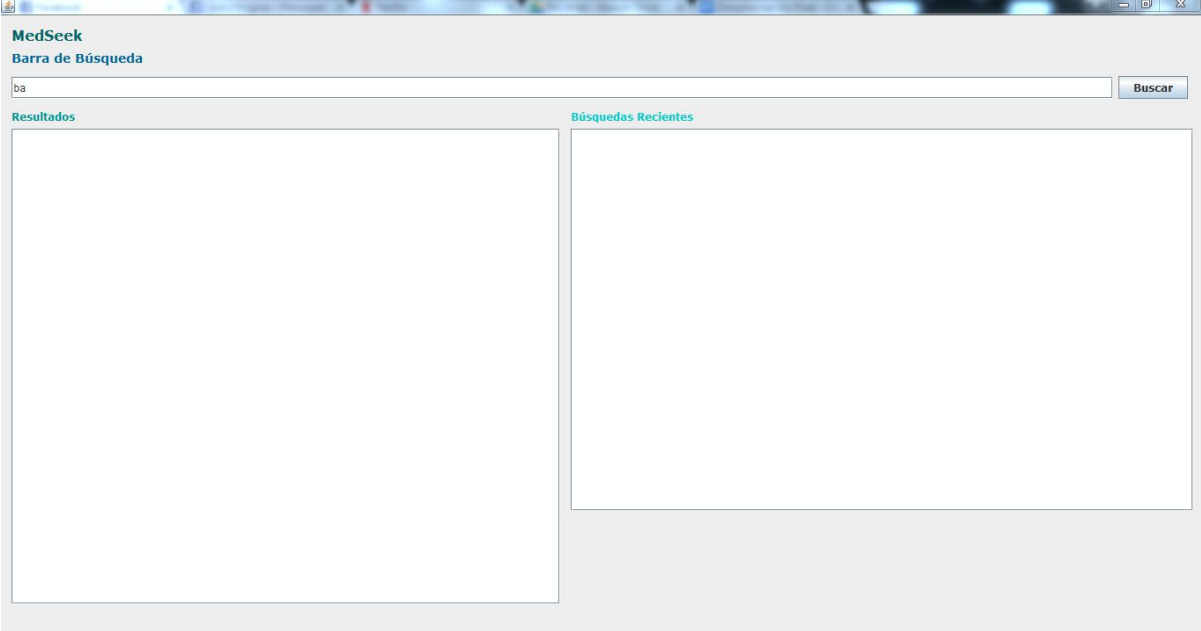


4. Pasos para replicar el proceso

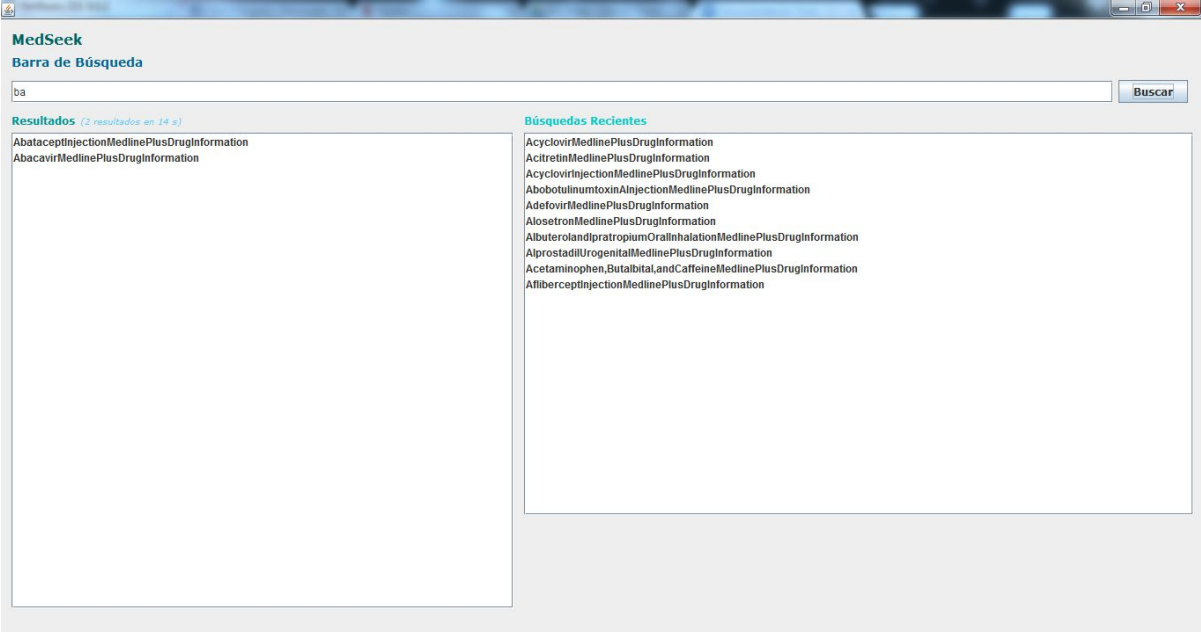
1. Instalar *python* y *pywin* (la arquitectura de *pywin* debe ser la misma que la de *python*).
2. Agregar ***python*** y ***python\Scripts*** a la variable de entorno ***path***.
3. En la consola de comandos, correr el comando ***pip install Scrapy***.
4. En el directorio ***arana***, correr el archivo ***correrArana.bat*** para que corra la araña que se realizó en Scrappy.
5. Correr el ejecutable del proyecto ***arreglarPaginas***.
6. Agregar a la variable de entorno ***CLASSPATH*** (si no existe crearla) la dirección ***carpetadelproyecto\stanford-parser-full-2015-04-20\stanford-parser.jar***.
7. Correr el archivo ***tokenizar.bat***.
8. Correr el ejecutable del proyecto ***NormalizarTokens***.
9. Correr el ejecutable del proyecto ***CrearDiccionario***.
10. Correr el ejecutable del proyecto ***CrearPostings***.
11. Agregar al proyecto ***SearchEngine*** la librería ***json_simple-1.1.jar*** (si no está incluida) ubicada en la carpeta del proyecto.
12. Correr el ejecutable del proyecto ***SearchEngine*** (verificar antes que se ejecutó bien el paso anterior y se generó el archivo *postings.txt*, además, tomar en cuenta que el programa dura arrancando aproximadamente 10min).
13. Probar el buscador.

5.Manual de usuario

a. Escriba su consulta y a continuación haga clic en “Buscar”:



The screenshot shows the MedSeek web application interface. At the top, there is a header with the text "MedSeek" and "Barra de Búsqueda". Below the header is a search bar containing the text "ba". To the right of the search bar is a button labeled "Buscar". Below the search bar, there are two main panels. The left panel is titled "Resultados" and is currently empty. The right panel is titled "Búsquedas Recientes" and is also empty.



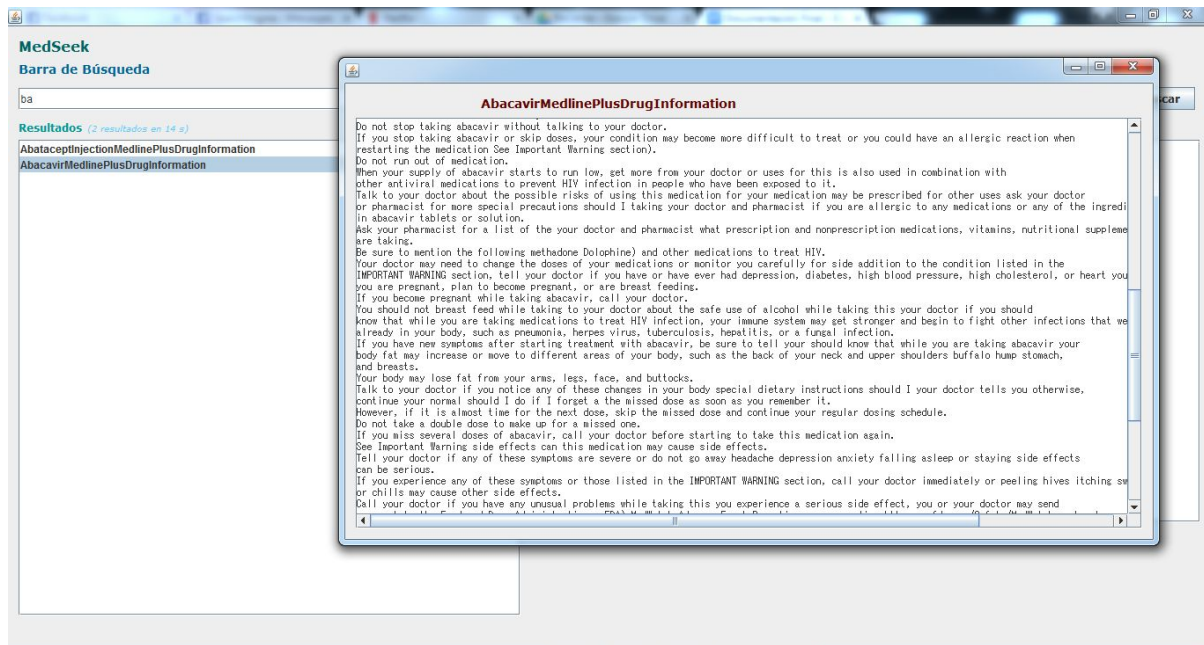
The screenshot shows the MedSeek web application interface after a search. The search bar still contains the text "ba". The "Resultados" panel now displays two search results:

- AbataceptInjectionMedlinePlusDrugInformation
- AbacavirMedlinePlusDrugInformation

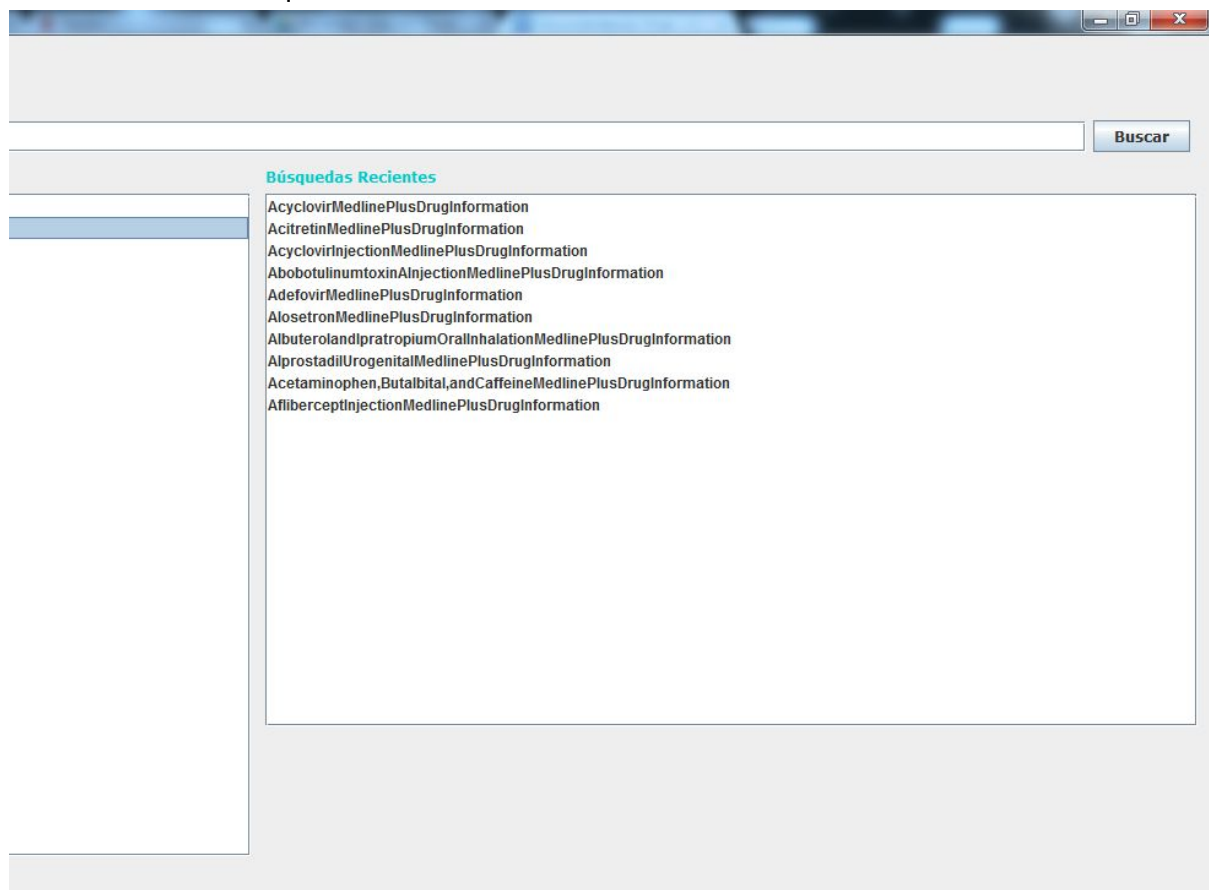
The "Búsquedas Recientes" panel now displays a list of recent searches:

- AcyclovirMedlinePlusDrugInformation
- AcitretinMedlinePlusDrugInformation
- AcyclovirInjectionMedlinePlusDrugInformation
- AbobotulinumtoxinAInjectionMedlinePlusDrugInformation
- AdefovirMedlinePlusDrugInformation
- AlosetronMedlinePlusDrugInformation
- AlbuterolandIpratropiumOralInhalationMedlinePlusDrugInformation
- AlprostadilUrogenitalMedlinePlusDrugInformation
- Acetaminophen,Butalbital,andCaffeineMedlinePlusDrugInformation
- AfliberceptInjectionMedlinePlusDrugInformation

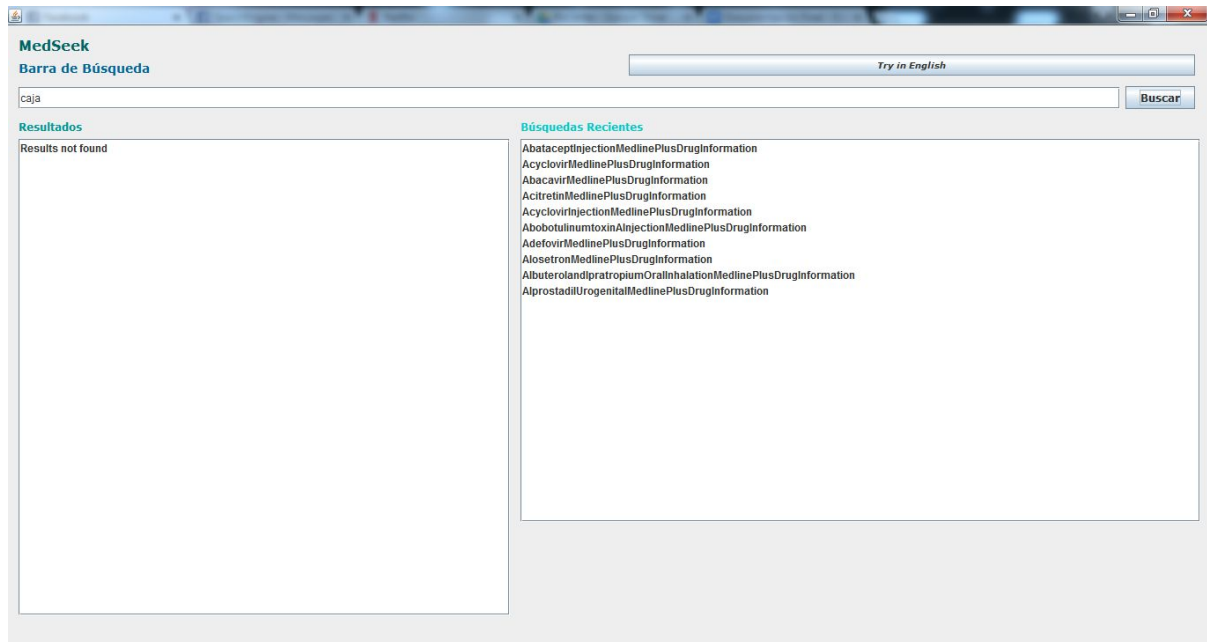
b. Abra los documentos (doble clic):



c. Revise sus búsquedas recientes



- d. Traduzca su búsqueda: en caso de que escriba una palabra, por ejemplo en idioma español, el motor de búsqueda hará una sugerencia de traducir la búsqueda a inglés.
- e.



6.Problemas sin resolver

No pudimos hacer que la lógica del programa fuera un servicio web porque el código que ya habíamos trabajado no tenía una forma fácilmente acoplable a un servicio web cuando se intentó crear.

7. Características adicionales

Sistema de Reconocimiento de Lenguajes y Traducción de Consulta

Decidimos tomar en consideración que dado que nuestro sistema, de momento, está planteado para un dominio de páginas web recuperadas que se encuentran en inglés, el usuario podría ingresar una consulta que esté en otro idioma nuestro sistema no tendría la más remota posibilidad de ofrecerle la información aunque sea que ésta esté en el idioma del dominio. Por eso planteamos y desarrollamos que nuestro sistema adhiriera una reinterpretación básica de la consulta. Para lograr esto, se aprovechó el servicio web que ofrece *Yandex.com* mediante un API para Java, este API permite la detección del idioma del texto y la traducción del mismo. La idea, resulta en considerar qué pasa si la búsqueda realizada en otro idioma, previamente detectado, se realizaría en el idioma del dominio; y de ser que la búsqueda en el idioma del dominio (inglés) produzca más resultados (o alguno, en caso de que la búsqueda original no ofrezca), se le ofrezca al usuario como sugerencia reinterpretar el idioma de la búsqueda, dándole como opción la traducción generada por el servicio web.

Evidentemente, la calidad del reconocimiento y traducción del texto afectará o beneficiará el resultado provisto por nuestro motor de búsqueda; sin embargo, *Yandex.com* ofrece una solución gratuita (a diferencia de Google) y de buena calidad, por esta razón decidimos recurrir al API de traducción de esta compañía.