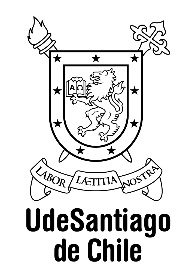
|  |  |
| --- | --- |
| **UNIVERSIDAD DE SANTIAGO DE CHILE**  **FACULTAD DE INGENIERÍA**  **DEPARTAMENTO DE INGENIERÍA INFORMÁTICA** |  |

**MANUAL DE USUARIO**

**SISTEMA DE RECUPERACION DE INFORMACIÓN**

**Realizado por: Jorge Luis Plaza Contreras**

**Programado en Scheme**

**Santiago de Chile**

**2-2017**

El objetivo de este manual es explicar la manera de llevar a cabo una correcta ejecución del programa, complementando con ejemplos para lograr una buena explicación en su desarrollo.

Este sistema de información consiste en lograr satisfacer la necesidad de información de parte del usuario. El cual, mediante una consulta del usuario, mostrara los mejores resultados para poder cumplir con su objetivo.

El primer paso es abrir el programa, se necesita tener DrRacket para poder ejecutarlo, este programa permite realizar programas bajo el lenguaje de programación Scheme.

Para descargar este programa, tiene que acceder al siguiente link, <https://racket-lang.org/>. Al ingresar en la página, selecciona la opción Download, y luego seleccionando la opción correcta según su sistema operativo puede descargar el programa.

Con el programa ya instalado, se procede a abrir el archivo con extensión racket mediante archivo->open, y seleccionando el archivo en la carpeta correspondiente.

Al abrir el archivo, en la cabeza del programa, se encuentran dos funciones que definen la entrada. Estas funciones se ven de la siguiente forma:

*Figura 1: StopWords*

*Figura 2: Lista de documentos*

En el caso de la figura 1, usted puede ingresar en cada entre comillas (““) las palabras que no quiere considerar al momento de recuperar la información, es decir, si ingresa la palabra “hola” el programa se encarga de ignorarla en los procesos de recuperación. Lo ideal es que ingrese una cantidad no mayor a mil palabras para no caer en procesos de error.

Para la figura 2, se va a ingresar los documentos de los cuales se requiere recuperar la información. Para que el programa tenga una correcta ejecución se debe seguir el formato mostrado en la figura. La función es una lista de listas, donde que cada lista se va a ingresar, el número de documento, el título, autor, bibliografía y finalmente el texto a ingresar. Luego para cada lista se debe seguir el mismo proceso hasta llenar con los documentos que requiera, es de importancia no ingresar una cantidad desorbitada de documentos para no caer en problemas de memoria.

Para ejecutar el programa se debe presionar en “ejecutar” o “run”, ubicado en la esquina superior derecha del programa.



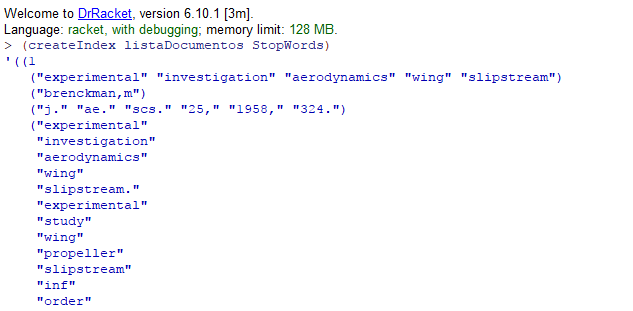
*Figura 3: Ejecutar programa*

Ahora el programa se encuentra listo para proceder a realizar el proceso de recuperación. Para realizarlo, se tienen distintas funcionalidades que se aplicaran escribiendo ciertas líneas en el programa. Estas funcionalidades son las siguientes.

1. createIndex: Esta funcionalidad permite crear un índice a partir de los parámetros ingresados anteriormente, se debe aplicar de la siguiente forma:

*Figura 4: Crear índice*

Al escribir esta línea en el programa y presionar “enter”, el programa procede a mostrar el índice creado por el programa.



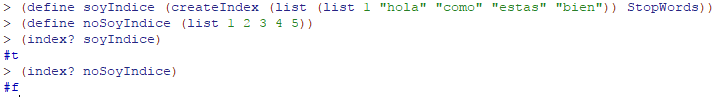
*Figura 5: Ilustración createIndex*

1. TDA índice: La estructura de índice representada en este programa, permite tener ciertas funcionalidades que son particularmente para la estructura. Estas funciones son: index?, que permite saber si algo es un índice. getTitulo, getAutor, getBibliografia y getTexto, que permiten obtener los parámetros título, autor, bibliografía y texto de un documento en particular. Para aplicar se tiene que escribir lo siguiente:

*Figura 6: Función pertenencia*

*Figura 7: Función selectora*

En la figura 6, se aplica la función de pertenencia que permite saber si una estructura es un índice. El programa muestra los valores #t, que indica que verdadero, y, #f que indica que la estructura no es un índice.



*Figura 8: Ilustración TDA*

En la figura 7, se aplica la función de pertenencia, getTitulo, la forma de aplicar para los demás es exactamente igual. Para aplicarla se necesita como parámetro el índice definido y el número del documento del cual quiere obtener el título. El programa mostrar el parámetro requerido.

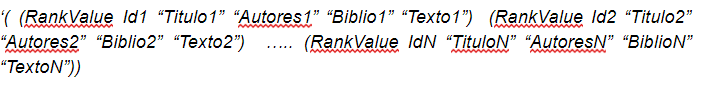
1. termQuery: Ya esta funcionalidad permite realizar una consulta al programa que devolverá lo pedido por el usuario. La forma de aplicar es la siguiente:

*Figura 8: Función termQuery*

En la figura 8 se puede observar la línea que se debe escribir para aplicar esta funcionalidad del programa. En el parámetro index se debe ingresar el índice definido anteriormente, el parámetro term es la consulta que el usuario va a ingresar para recuperar. Esta consulta term debe ser escrita en una sola palabra y mediante entre comillas (“”), finalmente el parámetro documents es la lista de documentos definida en la cabeza del programa (listaDocumentos). Por lo tanto, si se requiere realizar la consulta mediante la palabra “hola”. Se debe hacer de la siguiente forma:

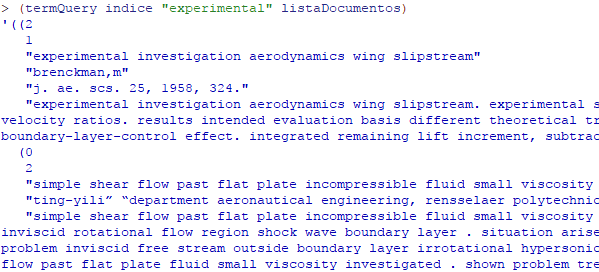
*Figura 9: Consulta “hola”*

Al escribir esta línea, el programa devolverá una estructura de la siguiente forma:



*Figura 10: Aplicación termQuery*

Lo mostrado por el programa es similar a un índice, pero con un valor adicional llamado RankValue. Este valor es la cantidad de veces que se encuentra la palabra en una sección del documento independientemente de la cantidad de veces que aparezca la palabra en la sección. Si la palabra “hola” se encuentra en el título y texto de un documento, 5 y 4 veces respectivamente, el valor de RankValue será de dos ya que la palabra se encuentra en dos secciones distintas.



*Figura 11: Ilustración termQuery*

1. phraseQuery: Esta funcionalidad es similar a termQuery. En este caso, se puede realizar una consulta pero mediante una frase ingresada por el usuario, la frase ingresada será filtrada mediante la lista de stopwords para quitarlos. La forma de aplicarla es la siguiente:

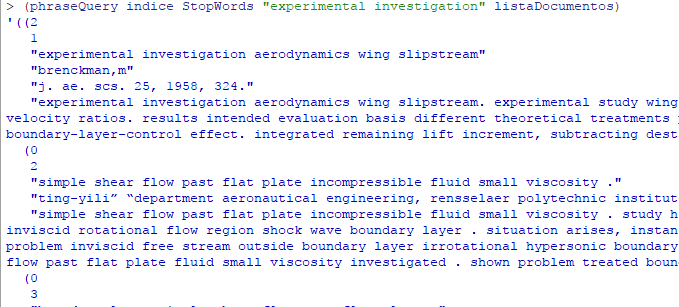
*Figura 11: Funcionalidad phraseQuery*

La función recibe 4 parámetros. El primero de ellos es el índice, el segundo la lista de stopwords, el tercero es la frase ingresada entre comillas y el cuarto es la lista de documentos.

Si se desea realizar una consulta, como por ejemplo, “hola como estas”. Se debe realizar de la siguiente forma:

*Figura 12: Consulta “hola como estas”*

Al realizar la consulta mediante la función phraseQuery se mostraran los resultados de la misma forma que en la función termQuery.



*Figura 13: Ilustración phraseQuery*

1. Ranking: Esta función se encarga de ordenar de forma ascendente o descendente los resultados obtenidos en las funciones termQuery o phraseQuery. Se aplica de la siguiente forma:

*Figura 13: Funcionalidad ranking*

orderType es un parámetro que representa un número, este parámetro debe ser 1 para ordena los resultados de forma ascendente o 2, que ordena los resultados de forma descendente. Al realizar la funcionalidad, el programa mostrara la lista de resultados ordenada según lo desee el usuario.

1. Results->String: Esta funcionalidad permite mostrar los resultados obtenidos de manera que sea legible para el usuario. Se aplica de la siguiente forma:

*Figura 14: Aplicación Results->String*

El parámetro resultado es un resultado definido mediante las funcionalidades termQuery o phraseQuery.

Los casos de errores del programa se pueden dar en un mal ingreso de los parámetros, por ejemplo, si se ingresa una estructura que no sea un índice a la función createIndex, lanzara un error.

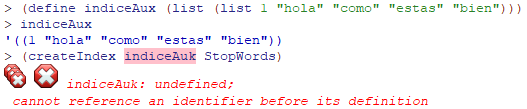


*Figura 15: Error parámetro*

En la figura 15 se comete el error de ingresar una lista en lugar de un índice, esto lanzara el error mostrado. Para solucionar el error, ingresar un índice adecuado a la función.

El error de ingresar mal los parámetros puede ser muy común a la hora de aplicar las funcionalidades del programa. Si se llega a caer en un error, revisar en primer lugar si los parámetros ingresados son correctos.

También puede ocurrir un error si el parámetro está mal escrito.



*Figura 16: Error de escritura*

En caso de que ocurra este error, volver a escribir nuevamente la línea de la funcionalidad con el parámetro correctamente escrito.

Otro error común es no seguir la sintaxis correcta del lenguaje.

C:\Users\Youta\Desktop\Imagenes Manual Scheme\Error.png

*Figura 17: Error sintaxis*

En este manual se explicaron paso a paso cada funcionalidad del programa con algunos posibles casos de error, en caso de existir otro tipo de error, contactar con el diseñador del programa.