|  |
| --- |
| **MANUAL DE USUARIO**  **SISTEMA DE RECUPERACIÓN DE INFORMACIÓN**  **Programado en prolog.** |
|  |
| **Realizado por: Jorge Luis Plaza Contreras** |

|  |  |  |
| --- | --- | --- |
|  | Santiago de Chile |  |
|  | 2 - 2017 |  |

Tabla de Contenidos

[Tabla de Contenidos I](#_Toc497849643)

[Índice de Figuras I](#_Toc497849644)

[CAPÍTULO 1. Introducción 1](#_Toc497849645)

[1.1 Procesos de compilación y Ejecución 1](#_Toc497849646)

[CAPÍTULO 2. FUNCIONALIDADES 3](#_Toc497849647)

[2.1 Consultas 3](#_Toc497849648)

[2.2 Otras funcionalidades 4](#_Toc497849649)

[CAPÍTULO 3. Ilustracion exitos o fallos 5](#_Toc497849650)

Índice de Figuras

[Figura 1: Programa abierto 2](#_Toc382443674)

[Figura 2: Consulta mediante singleTermQuery 3](#_Toc382443674)

[Figura 3: Consulta mediante bestMatch 3](#_Toc382443674)

[Figura 4: Consulta mediante exactMatch 4](#_Toc382443674)

[Figura 5: Funcionalidad documents 4](#_Toc382443674)

[Figura 6: Funcionalidad numDocuments 4](#_Toc382443674)

[Figura 7: Funcionalidad termTotalCount 5](#_Toc382443674)

[Figura 8: Funcionalidad maxFrequency 5](#_Toc382443674)

[Figura 9: Éxito SingleTermQuery 5](#_Toc382443674)

[Figura 10: Fallo singleTermQuery 6](#_Toc382443674)

[Figura 11: Éxito bestMatch 6](#_Toc382443674)

[Figura 12: Fallo bestMatch 6](#_Toc382443674)

[Figura 13: Éxito documents 7](#_Toc382443674)

[Figura 14: Fallo documents 7](#_Toc382443674)

# Introducción

Un sistema de recuperación de información consiste en la búsqueda de información en documentos o cualquier tipo de colección de documento digital.

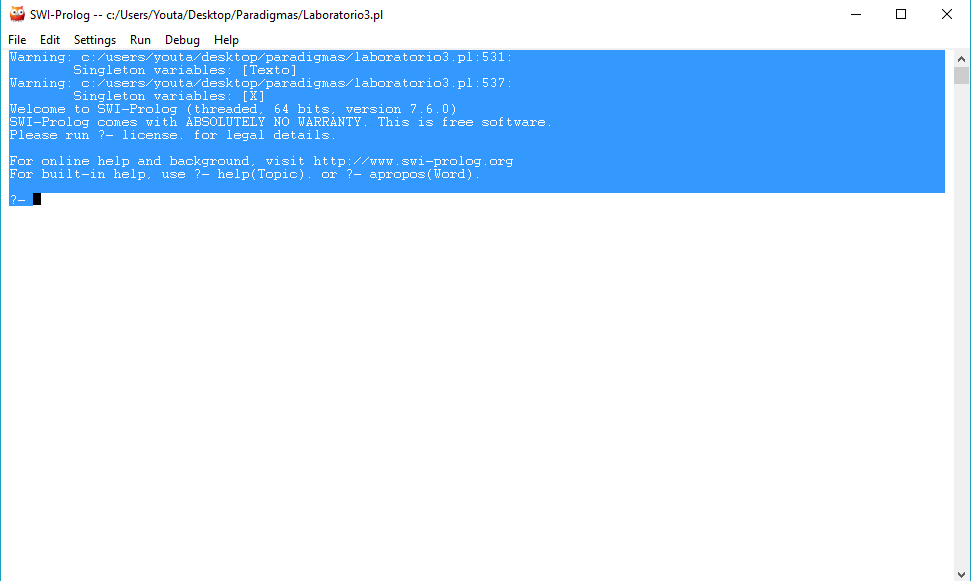
La idea es, mediante una consulta del usuario, que el sistema responda de la manera más rápida y eficiente lo pedido para satisfacer las necesidades del cliente.

El sistema también puede facilitar otras funcionalidades que pueden ser de interés para el cliente, como la obtención de extractos más largos de texto o cantidad de documentos inmersos en el sistema, entre otros.

Procesos de compilación y Ejecución

Para que se pueda utilizar el programa de forma satisfactoria, es necesario realizar unos pasos previos importantes. Estos pasos llevan por nombre compilación y ejecución. Estos pasos previos permiten al usuario comenzar a trabajar con el programa.

1. Como primer paso, se necesita descargar SWI-Prolog. Programa que permite al usuario abrir el programa para trabajar en él. Para descargarlo, es necesario ingresar a la siguiente página: <http://www.swi-prolog.org/download/stable>.
2. Luego de ingresar a la página, es necesario escoger debidamente el sistema operativo en el cual se está trabajando, para posteriormente, comenzar con la descarga.
3. Al descargar el programa, comenzar el proceso de instalación simplemente haciendo click en el instalador. Al abrirlo seguir los pasos correspondientes de la instalación.
4. Al instalarlo, el archivo del programa que viene pasara directamente a abrirse automáticamente con SWI-prolog. Por lo tanto, solo se necesita abrir directamente el programa.
5. Ya con el programa instalado, se puede observar la consola donde podrá utilizar las funcionalidades del programa.

****

*Figura 1: Programa abierto.*

# FUNCIONALIDADES

2.1 Consultas

Las funcionalidades más importantes del programa es la posibilidad de realizar consultas con respecto a los documentos en el código. Se pueden realizar 3 tipos distintos de consultas:

* singleTermQuery: La forma más simple de realizar una consulta, es entregar solo una palabra al programa. Para hacerlo, se necesita escribir en la consola lo siguiente:

Figura 2: Consulta mediante singleTermQuery

En la figura 2, se ilustra una consulta mediante la palabra consulta y una variable que permite al programa entregar los posibles resultados. La primera palabra será la consulta que quiera realizar y la segunda tiene que ser escrita si o si con mayúscula como letra inicial. Puede ser cualquier letra o palabra siempre y cuando la primera sea mayúscula.

* bestMatch: Otra forma de realizar una consulta es hacerlo escribiendo múltiples palabras. El programa entregara los resultados correspondientes. Para realizarlo se debe escribir lo siguiente:

Figura 3: Consulta mediante bestMatch

En la figura 3, se ilustra una consulta mediante múltiples palabras, donde la consulta debe ir entre corchetes y cada palabra separadas por una coma. Al igual que en singleTermQuery, la segunda palabra debe ser con la primera letra en mayúsculas para que prolog pueda contestar correctamente a la consulta. Prolog contestara los identificadores de los documentos en los que se encuentra cada palabra entre corchetes. Sin embargo, se detendrá en cada búsqueda, para que termine la consulta se debe hacer un punto y coma (“;”). De esta forma, prolog dará todos los posibles resultados de la consulta ingresada.

* exactMatch: La tercera y última forma de realizar una consulta, cambia en la forma en que responde prolog. Ya que la funcionalidad permite saber si un documento tiene exactamente todas las palabras ingresadas en las consulta. La forma de operar es igual a bestMatch.

Figura 4: Consulta mediante exactMatch.

2.2 Otras funcionalidades

El programa entrega otras 4 funcionalidades que permite mejorar la experiencia al usuario. Las funcionalidades son las siguientes:

* Documents: Esta funcionalidad permite extraer los títulos de los documentos que el usuario quiera. Para utilizar esta funcionalidad se escribe lo siguiente:

Figura 5: Funcionalidad documents

La figura 5, ilustra un ejemplo donde se piden los títulos de los documentos 1 y 2 respectivamente. Para hacerlo se colocan los números separados por coma y encerrado mediante corchetes. Nuevamente la segunda variable debe iniciar con mayúscula.

* numDocuments: Esta funcionalidad permite saber cuántos documentos hay en un listado ingresado por el usuario. La forma de aplicar es la siguiente:

Figura 6: Funcionalidad numDocuments

En la figura 6 se aprecia el uso de la funcionalidad, donde los documentos van encerrados entre corchete y con comas. La segunda palabra debe comenzar con mayúsculas para que prolog pueda responder correctamente.

termTotalCount: Esta funcionalidad entrega al usuario en la cantidad de documentos que se encuentra cierta palabra. Para aplicar la funcionalidad se escribe lo siguiente:

Figura 7: funcionalidad termTotalCount

En la figura 7 se muestra el ejemplo con la palabra consulta. A través de Frequency, prolog entregara el resultado correspondiente. Nuevamente la segunda variable debe tener mayúscula como letra inicial.

maxFrequency: Funcionalidad que entrega las palabras que se encuentran en todos los documentos.

Figura 8: funcionalidad maxFrequency

Solo se ingresa una variable. Tiene que ser con mayúscula inicial para que prolog entregue el resultado correctamente.

# Ilustracion exitos o fallos

Aquí se ilustraran caso de éxito o fallos para algunas de las funcionalidades presentes en el programa. Cada una vendrá acompañado de una figura para un mejor entendimiento del usuario.

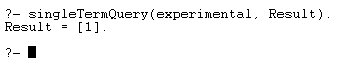
.

Figura 9: Éxito singleTermQuery

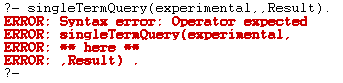


Figura 10: Fallo singleTermQuery

La figura 9 ilustra un caso de éxito para singleTermQuery, escribiendo tal y como se explica en el manual se obtiene como resultado que la palabra “experimental” se encuentra solo en el documento numero 1 según prolog.

La figura 10 muestra un típico error al escribir un predicado. Se escribieron dos comas, lo que deriva en un error de sintaxis. En caso de que ocurra este error, volver a escribir nuevamente la sentencia completa.

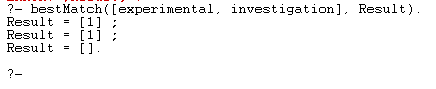


Figura 11: Éxito bestMatch

C:\Users\Youta\Desktop\Imagenes prolog\fallo best mAtch.png

Figura 12: Fallo bestMatch

La figura 11 ilustra un caso de éxito de bestMatch siguiendo las instrucciones del manual. En la figura 12 prolog no lanza un error como tal pero si es un mal uso de la funcionalidad que puede derivar en confusiones. Aquí no se ingresa la consulta entre corchetes y con comas, si no que se hace como en singleTermQuery. En caso de que ocurra esto, volver a escribir la sentencia nuevamente.

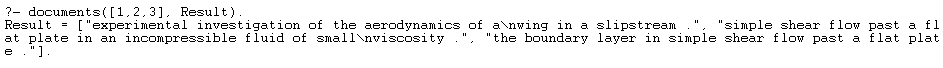


Figura 13: Éxito documents

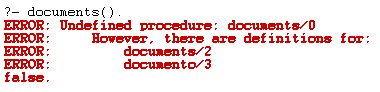


Figura 14: Fallo documents

En la figura 13 se ilustra un caso de éxito de la funcionalidad documents. Donde prolog contesta mediante una lista los títulos de los documentos ingresados según su identificador.

La figura 14 muestra un caso de error. Solo colocando la sentencia sin sus respectivas variables deriva en un problema de procedimiento. Para corregir el error, escribir nuevamente la sentencia con su respectivas variables.



Como se ha podido dar cuenta, la mayoría de los casos de error del programa se deben a un mal ingreso de las funcionalidades presentes. Procurar siempre seguir paso a paso el manual de usuario para poder tener una experiencia completa y satisfactoria.

Si se detecta cualquier otro tipo de error, contactar directamente con el programador y seguir sus instrucciones.