

TERCERA TAREA

PROGRAMADA

Lenguajes de Programación - Prof. Andrey
Fuentes L.

Adrián Díaz Meza / John Largaespada Pérez / Alonso Vargas
Astua

Tabla de Contenidos

<i>Contenido</i>	<i>Pag.</i>
<i>Portada</i>	1
<i>Tabla de Contenidos</i>	2
<i>Descripción del Problema</i>	3
<i>Diseño del Programa</i>	4
<i>Librerías Usadas</i>	5
<i>Análisis de Resultados</i>	6
<i>Manual de Usuario</i>	7
<i>Conclusión</i>	10

Descripción del Problema

La siguiente descripción hace referencia a la problemática presente todos los días. La cuál nos ha sido asignada la tarea de solucionarlo de una manera nueva, rápida y sencilla (para el usuario), a continuación la descripción del problema planteado:

“En una investigación típica, se pueden consultar decenas de documentos con el fin de recopilar toda la información relevante. Algunos de esos documentos están disponibles en Internet, mientras que otros son descargados a las computadoras en formatos como el pdf. Ahora, la administración de los pdfs descargados suele ser una tarea bastante tediosa, ya que si se quiere buscar un documento específico, se tiene que abrir y revisar la información del documento, lo cual es un proceso sumamente lento.”

Por esta razón es que se nos solicita el desarrollo de un programa con el lenguaje de programación LISP, se pide un sistema muy eficiente con el fin de realizar la búsqueda de documentos de una manera más rápida y sencilla.

Además, el sistema trabajará mediante la búsqueda de metadatos, de manera que al tener esta información de los documentos (nombre, autor, tamaño, asunto, palabras clave...) situados en un directorio especificado por el usuario.

Por otra parte, se debe crear en memoria una base de datos que mantenga los metadatos de los pdf's, ésta debe estar definida detalladamente. Una vez que se tenga la información almacenada se debe de realizar una serie de consultas para el usuario, entre ellas se encuentran:

1. Es necesario mostrar toda la información de todos los pdf's usando un formato de tabla con las siguientes columnas: nombre del archivo, título, autor, asunto, palabras clave y fecha de creación.
2. También se debe mostrar la información de los documentos introduciendo datos específicos como el nombre del autor, títulos, palabras claves (cabe rescatar que un documento puede tener varias palabras claves) o fechas de creación. Estos datos deben mostrarse en un formato igual a los del punto anterior.

Además para la creación de dicho proyecto es necesaria la investigación de cómo leer la estructura de los metadatos en los pdfs, para poder extraer la información y usarla a la hora de hacer búsquedas de documentos.

No obstante se debe investigar cómo leer archivos y de recorrer todos los archivos de cierto tipo dentro de un directorio especificado, para poder ir extrayendo la información de metadatos de cada uno de los archivos de dicho directorio.

Diseño del Programa

Durante el diseño procuramos crear primero la base de datos en la que almacenarán los metadatos, definiendo cada una de los slots necesarios. Luego procedemos a definir la función agrega, que introduce un dato nuevo a la BD esto porque se debe ingresar nuevos datos constantemente a ella. Llegamos a la conclusión de que deberíamos almacenar la dirección de los ficheros los cuáles queríamos investigar, para lo cual creamos la variable "tablas_ficheros".

La función busca_label trabaja en el momento que se desea buscar los datos en específico de los metadatos, por ejemplo el título o el autor, retornan la búsqueda del metadato (título, autor...).

A continuación se encuentra la función extraer_metadatos, que se enfatiza a simplemente extraer los datos del pdf. Después llegamos a las funciones "select..." estas se encargan de hacer consultas a la base de datos con un parámetro y retornan las filas con las que hay coincidencias. Se realiza de esta manera ya que vimos que es más fácil extraer todos los metadatos, almacenarlos en nuestra memoria y luego hacer las búsquedas e imprimir la información.

Para que esto trabaje de la mejor manera hacemos uso de las funciones itera que recorre el contenido de la tabla hash que contiene los directorios y los envía a las funciones que extraen los metadatos. Antes de eso hacemos uso de la función recorre que se encarga de extraer todos los archivos de tipo pdf que existen en un directorio y los almacena en una tabla hash.

Por último trabajamos con la función inicio que recibe un directorio y lo manda a las funciones que extraen los nombres de los ficheros y metadatos para crear la base de datos, en conjunto con esta función trabaja otra función llamada menú que se encarga de mostrar las opciones de consulta que el usuario puede realizar durante el uso del programa.

Librerías Usadas

Debido a la serie de diagramas y las facilidades que posee lisp, para este trabajo no se utilizó ningún tipo de librería. Lo cual facilitó el desarrollo del programa en gran medida, y nos hizo sentir un trabajo más directo a la hora de desarrollar el software.

Análisis de resultados

Documentación Interna: Se puede ver una definición o comentario en las funciones que se presentan en la tarea programada con la finalidad de brindar a las demás personas una idea de lo que realizan las funciones de dicha tarea.

Documentación Externa: Se puede observar la totalidad de los puntos a evaluar estipulados en el enunciado que entrega el profesor, entre ellas se encuentran la tabla de contenidos, descripción del problema, diseño del programa, librerías usadas, análisis de resultados, manual de usuario además de la conclusión personal.

Búsqueda de Ficheros: La búsqueda de los archivos de tipo PDF en un determinado directorio se realizó con éxito, las direcciones de los archivos PDF de almacenan para después realizar las consultas.

Extracción de Metadatos: La extracción de metadatos se realizó con éxito, se pueden extraer los metadatos de los archivos PDF y almacenarlos en la Base de Datos, La funcionalidad del programa solo permite extraer metadatos de archivos PDF 1.5

Creación de la Base de Datos: Los metadatos de los archivos PDF son almacenados con éxito en la base de datos, cada metadato se guarda con su respectiva etiqueta por ejemplo: un metadato de tipo título se guarda con una etiqueta que indica que lo es.

Consultas: Todas las consultas son realizadas con éxito, esta parte del proyecto se realizó completamente a excepción de la forma en que se despliegan los datos ya que la especificación indica que se deben desplegar en forma de tabla y esta parte no se pudo lograr, los datos son desplegados en forma de lista.

Manual de usuario

Para empezar debemos de instalar un prompt de common list, en este caso instalaremos “clisp” de la siguiente manera:

1. Abrimos terminal (ctrl + alt + t).
2. Ejecutamos el comando para la instalación de clisp: `sudo apt-get install clisp`

```
john@john-laptop:~$ sudo apt-get install clisp
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
clisp ya está en su versión más reciente.
```

3. Una vez ya instalado el programa procedemos a acceder a la carpeta en donde se encontrará el código ejecutable además de los archivos pdf, que en mi caso es la siguiente dirección:

```
john@john-laptop: ~/Escritorio/Progra 3
Archivo Editar Ver Terminal Ayuda
john@john-laptop:~$ ls
Descargas  examples.desktop  Imágenes  Público  Vídeos
Documentos Holis          Música    Python-2.7.2
Escritorio Holis.c      Plantillas Python-2.7.2.tgz
john@john-laptop:~$ cd Escritorio
john@john-laptop:~/Escritorio$ ls
Gestión Del valor ganado.docx  lkn.png  Pantallazo-2.png
Instalador Ubuntu 10.04      Pantallazo-1.png  Progra 3
john@john-laptop:~/Escritorio$ cd Progra\ 3/
john@john-laptop:~/Escritorio/Progra 3$ ls
BD_Lleccion_16_2012_NormalizacionEjerciciosSolucion.pdf  todoA.lisp  todox.lisp
todol.lisp  todo.lisp
```

4. Una vez que se encuentra en la carpeta con todos los archivos necesario se puede abrir clisp digitando lo siguiente: “**clisp**”.

Hecho esto aparecerá la siguiente codificación en la pantalla, que nos muestra el momento en que inicio lisp en funcionamiento.

```
john@john-laptop:~/Escritorio/Progra 3$ clisp
i i i i i i i i      00000 0      0000000 00000 00000
I I I I I I I I      8 8 8      8 8 0 8 8
I \ \ '+' / I      8 8 8      8 8 8
      \ '+' /      8 8 8      8 8
      | | | |      8 8 8      8 8
      | | | |      8 0 8      8 0 8 8
      | | | |      00000 8000000 0008000 00000 8
-----+-----

Welcome to GNU CLISP 2.44.1 (2008-02-23) <http://clisp.cons.org/>

Copyright (c) Bruno Haible, Michael Stoll 1992, 1993
Copyright (c) Bruno Haible, Marcus Daniels 1994-1997
Copyright (c) Bruno Haible, Pierpaolo Bernardi, Sam Steingold 1998
Copyright (c) Bruno Haible, Sam Steingold 1999-2000
Copyright (c) Sam Steingold, Bruno Haible 2001-2008

Type :h and hit Enter for context help.
```

- Ahora aparece el “puntero de lisp” entonces se debe cargar el programa que deseamos correr de la siguiente manera: se escribe (load “nombre archivo”) como se ve a continuación.

```
[1]> (load "todox")
;; Loading file /home/john/Escritorio/Progra 3/todox.lisp ...
;; Loaded file /home/john/Escritorio/Progra 3/todox.lisp
T
```

- Una vez cargado el programa se debe llamar al menú del programa escribiendo la palabra inicio entre paréntesis (esto para que cargue la base de datos). Luego nos aparece la solicitud de ingresar la dirección de los documentos que quieren ser analizados por el programa este dato se da entre comillas "", de la siguiente manera:

```
[2]> (inicio)

ESCRIBA LA DIRECCION DEL DIRECTORIO
"/home/john/Escritorio/Progra 3/"
```

- A continuación el programa se encarga de cargar la dirección del directorio, por lo cual tiene a durar algunos segundos “sin hacer nada”, luego aparece un menú como el presente en la siguiente imagen:

```
CONSULTAS
1 = Titulo
2 = Autor
3 = Asunto
4 = Palabra Clave
5 = fecha
6 = todo
7 = volver
```

- Una vez que aparece el menú, se presiona uno de los números según el tipo de consulta que se desea realizar o si desea volver al menú en caso que la dirección de memoria no se ingresara de la manera correcta. Para este ejemplo consultaremos el “Título” del documento por lo que ingresamos el numero 1 (el ingreso de estos datos se deben hacer sin las comillas).
- Ya definido en ámbito sobre el cual se va a hacer la consulta, se le solicita al usuario que escriba la palabra con la cual se desea buscar los documentos entre paréntesis, en este

caso el nombre del Título del libro.

```
1
Palabra a consultar
"NORMALIZACION:"
```

10. Al ingresar esta información procederá a imprimir todos los datos de los libros que dispongan de esa misma información entre sus metadatos, de la manera que veremos a continuación:

```
"NORMALIZACION:"
((:TITULO "NORMALIZACION:" :AUTOR "JOSE A. STRADI G." :ASUNTO "actica"
:PALABRAS_CLAVE "fina" :FECHA_CREACION "20120604"))
```

Nota: Este procedimiento se realiza con cada una de las opciones del menú, entonces el procedimiento es el mismo para todas. El menú nos permite verificar datos del título, autor, asunto, palabra clave, fecha en que se realizó en documento además de la opción de ver todo lo que se encuentra en la Base de Datos. Si después de realizar una consulta se requiere implementar otra, se puede hacer escribiendo la palabra menú entre paréntesis como se puede ver a continuación:

```
Break 1 [4]> (menu)

CONSULTAS
1 = Titulo
2 = Autor
3 = Asunto
4 = Palabra Clave
5 = fecha
6 = todo
7 = volver
```

Conclusión

Se considera que la realización de este proyecto ha sido de gran provecho, ya que han aprendido algunos aspectos que consideramos muy importantes en el ámbito de la programación. Cuando hacemos referencia a un aspecto muy importante nos referimos a aspectos como por ejemplo leer todos los archivos .pdf de un directorio especificado por un usuario, lo cual la consideramos sumamente importante ya que primero se debió investigar como leer este tipo de archivos sin utilizar Lisp, esto para poder entender cómo se maneja desde el terminal este tipo de tareas.

Además y quizá el aspecto aprendido más importante fue el trabajar con meta-datos de los pdf, ya que se tuvo que llevar a cabo una extensa investigación. Con esta investigación conocimos como manejan los pdf los metadatos, como extraerlos, además diversas maneras para modificarlos y convertir pdf de formato de versiones anteriores a la versión pdf-1.5 a versión pdf-1.5.

Finalmente podemos decir que Lisp es un lenguaje de programación sumamente difícil para su manejo. Pero con la realización de este proyecto hemos puesto en practica tanto los contenidos vistos en clases como los contenidos que se obtuvieron con las diferentes investigaciones realizadas durante este trabajo, permitiéndonos agilizarlos o más bien familiarizarnos con este lenguaje de programación.