

III Proyecto de Lenguajes de Programación

SML/NJ – Búsqueda y Validación de Fechas

Yaxiri;Esteban Azofeifa García

Esteban Benavides Cruz

30-05-2013

La tarea programada consiste en permitirle al usuario ingresar el nombre de un archivo de texto. El programa se encargará de abrir y leer el archivo de texto e identificar fechas con el formato dd/mm/yy ó dd/mm/yyyy. Luego se encargará de validar si las fechas son correctas y posteriormente desplegarlas como hileras y con el formato día del mes del año.

Tabla de contenido

Descripción del problema	2
Diseño del programa:	2
Decisiones de Diseño y Algoritmos utilizados	2
Lectura del texto:	2
Búsqueda de fechas:	3
Conversión de fechas a <i>Strings</i> :	3
Ordenamiento Ascendente	4
Análisis de resultados	4
Objetivos alcanzados	4
Manual de Usuario	4
Conclusión Grupal	8

Descripción del problema

La programación es el proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales. Para el Proyecto Programado III se debía crear un programa basado en el paradigma funcional y con ayuda del lenguaje Standard ML of New Jersey

El programa le deberá permitir al usuario ingresar el nombre de un archivo texto. Luego se deberá abrir y leer archivo y buscar fechas en él. Las fechas deberán tener el formato dd/mm/yyyy o dd/mm/yy de lo contrario no se tomarán en cuenta.

Una vez extraídas las fechas del archivo de texto, se deberá verificar si son fechas válidas: Si los meses y los días existen, si ese año era bisiesto, entre otros.

Cuando se hayan verificado si las fechas son reales o no, el programa de ordenarlas de manera ascendente y posteriormente retornarlas como Strings y con el formato día del mes del año.

Diseño del programa:

Decisiones de Diseño y Algoritmos utilizados

Lectura del texto:

Para la lectura del texto se diseñó una función que se encarga de abrir el texto, almacenarlo en una variable y mediante recursiones se verifica si hay saltos de líneas. De ser así, cada línea del texto se almacena en una lista, hasta que no haya más texto por leer.

Se decidió almacenar el texto en una lista para manipularlo de manera más fácil

Búsqueda de fechas:

La búsqueda de fechas está compuesta por varias funciones.

La primera de ellas es una función que toma el texto leído y almacenado en una lista y transforma la lista de Strings en una lista de Chars. Utilizamos la función *explode* de los Strings.

Se decidió transformar la lista de Strings en una lista de Chars para manipularlo de manera más fácil y así poder identificar de manera individual cada uno de los caracteres.

Luego, se diseñó la función principal de búsqueda, la cual se encarga de recorrer toda la lista de caracteres, mediante recursiones, y verificar si son números. Como son chars es más fácil identificar si son números con ayuda de la función *Char.isDigit* de la estructura de los chars.

Luego con ayuda de un contador, verificamos la posición de los números de acuerdo con el formato deseado (dd/mm/yy o dd/mm/yyyy), y nos apoyamos en variables temporales para almacenar de manera parcial los varoles. Al final almacenamos las fechas en listas, ya que a la hora de que se van agregando las fechas, se hace de manera inversa, por lo que al almacenarlas en una lista nos permite aplicar la función built-in *reverse* de la estructura lista, tomando en cuenta que estas son mutables.

Para poder realizar la inserción de las fechas utilizamos una función auxiliar que se encarga de convertir los chars a enteros.

Por último, se diseñaron dos funciones, una que se encarga de transformar una lista a una tupla y otra que se encarga de tomar la lista de fechas y aplicarles esta conversión a todas las listas internas.

Conversión de fechas a *Strings*:

Para esto se creó una función principal que verifica si se ingresaron fechas o no. Posteriormente, se encarga de tomar la lista de fechas, que en este caso cada uno de los valores son *int*, y con la ayuda de una función auxiliar, convertir las tuplas de *int* a una lista de *strings*, y por último almacenarlas en una lista.

La función que convierte los *int* a *strings* llama a otras funciones auxiliares que se encargan de darle el formato de hilera correspondiente (día del mes del año) y con la ayuda de otra función auxiliar se logra identificar el número del mes y cambiarlo por el nombre correspondiente. Los nombres de los meses están almacenados en una tupla aparte.

Ordenamiento Ascendente

Para lograr el ordenamiento de las fechas de manera ascendente, simplemente se tomaron las tuplas que se encuentran dentro de la lista y ya que son listas es mucho más fácil acceder por medio de índices. Posteriormente se compararon y se identificaron las fechas mayores.

Análisis de resultados

Al final se alcanzaron los resultados deseados y el programa cumple con todos los requerimientos funcionales planteados. Se logró generar todas las estructuras necesarias y no hubo problema en el paso de datos a través de las funciones. Aún se puede trabajar en una mayor optimización, sin embargo la modularidad de las funciones y la definición de funciones auxiliares permitió alcanzar todos los objetivos de una manera más sencilla.

Objetivos alcanzados

- ✓ Abrir y leer el archivo de texto
- ✓ Buscar las fechas que cumplan con el formato adecuado
- ✓ Almacenar las fechas
- ✓ Convertir las fechas a Hileras
- ✓ Darle el formato correspondiente a las fechas
- ✓ Retornar las fechas

Manual de Usuario

Esta tarea programada se trabajó en Ubuntu 12.10.

Inicialmente se debe instalar SML/NJ. El único paquete que se debe descargar es el config.tz <http://smlnj.cs.uchicago.edu/dist/working/110.75/config.tgz>

Se configura y se instala utilizando los comandos respectivamente:

```
$config/targets
```

```
$config/install.sh.
```

Luego se descargarán los datos faltantes de manera automática.

Otro método de instalación es escribiendo el comando:

```
$sudo apt-get install smlnj
```

Una vez instalado SML/NJ se debe instalar el entorno de desarrollo. En este caso se utilizará GNU/Emacs.

Para instalar emacs se escribe en consola el siguiente comando:

```
$sudo apt-get install emacs24
```

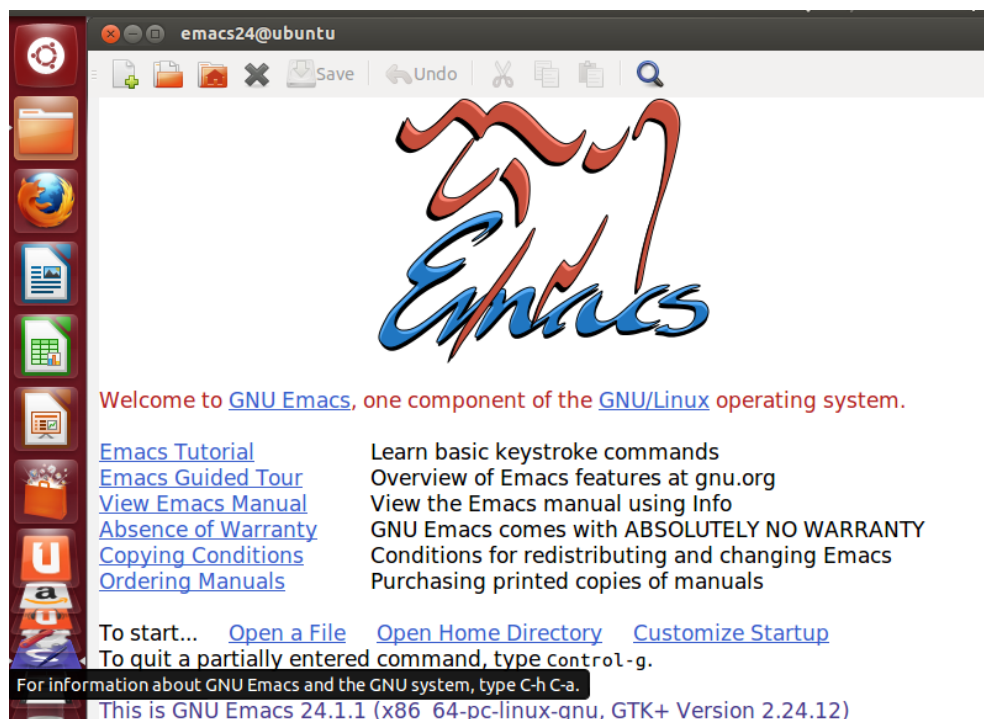
Luego automáticamente instala.

Dentro de emacs, ejecutar el comando Alt-x, y digitar el comando listpackages, digitar Enter.

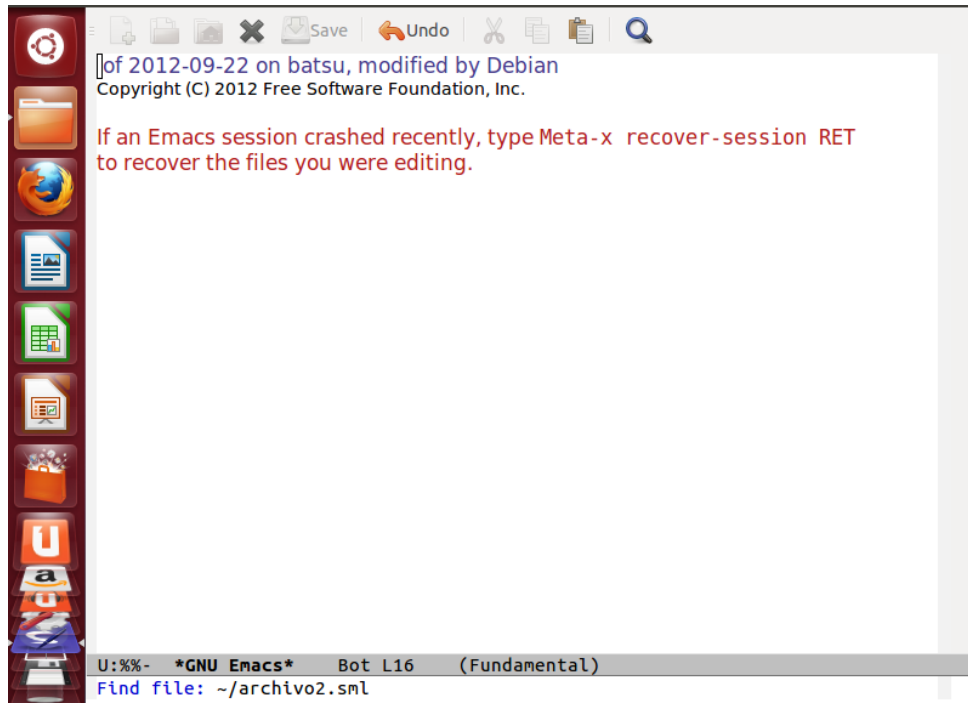
Dentro de los paquetes listados, buscar "sml-mode", darle click, y darle install.

Salir y reiniciar emacs

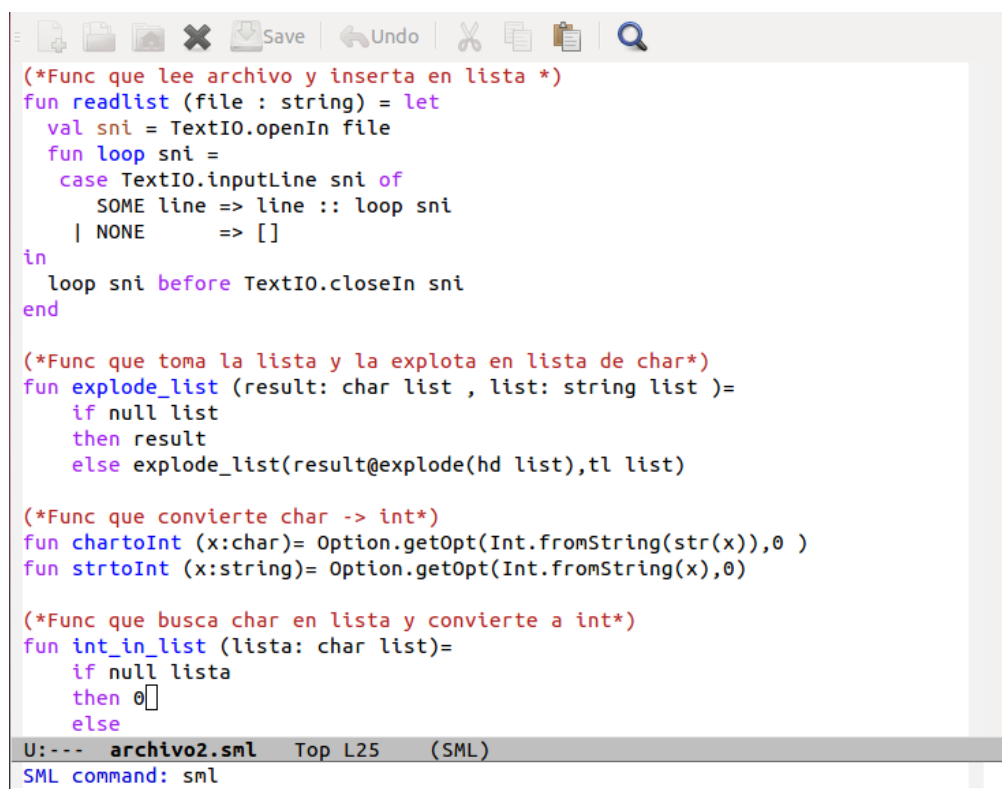
Se ejecuta Emacs



Luego ejecutar el comando Control X –F y escribir el nombre del archivo



Abrir el archivo y ejecutar el comando META C –L , presionar Enter



Escribir main(“dirección/del/archivo.txt”) y ejecutar el programa

Conclusión Grupal

A la hora de realizar este proyecto con base en el paradigma funcional se generó un mayor conocimiento en la utilización de funciones y en el funcionamiento de la recursividad. Por otro lado, se investigó mucho sobre cada una de las propiedades y funciones que poseen los diferentes tipos primitivos de SML ya que dicho conocimiento era necesario para la manipulación de muchos datos en el proyecto. Esto se logró con la ayuda de *“The Standard ML Basis Library”*. Se lograron alcanzar todos los objetivos satisfactoriamente.