**Trabajo Práctico Nro 2**

**Web Scrapping Financiero**

**Sintáxis y Semántica de los Lenguajes**

**Docente:** Ing. Pablo Damián Mendez

**Alumno:** Leandro Tomás Rouillet

**Legajo:** 16.3443-4

**Estrategia de resolución:**

Para este trabajo, opté por hacer un *parse* del archivo html dado a un formato csv. De esta manera, el manejo de la información sería mucho más sencillo. Una vez realizado el parseo, utilicé este csv generado para realizar los diferentes reportes ni bien se inicia el programa. De esta manera, el usuario tendrá precargados los reportes incluso antes de seleccionar cualquier opción de la interfaz.

Usé la función strtok\_r() para poder separar filas y columnas. No usé strtok porque tiene limitaciones a la hora de generar tokens anidados (generar un token a partir de otro), en cambio, a strtok\_r se le puede pasar una referencia al último token generado, por lo que se pueden anidar tokens.

Para el caso del reporte por consola y por csv, la estrategia implementada es similar. Se iteran las filas, y en cada una de ellas se obtiene la columna requerida y se la guarda en un buffer que luego será utilizado para imprimir en pantalla en caso del reporte por consola, o para escribir un archivo en caso del reporte en csv.

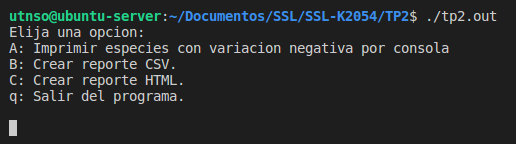
En el caso del reporte html, se iteró de igual manera que en los reportes anteriores, con la diferencia de que ahora se tenía que verificar que se cumpla cierta condición para que la fila sea de color verde o no. Para la verificación de que la variación sea negativa, use la función strchr(), la cual me indica si un carácter está presente en un string (en este caso, el carácter ‘-’). Para la lógica de pintado de la fila usé strcat en un buffer de tamaño fijo, ya que sabía cuáles eran los valores que podía tomar (“green” o “none”). Este buffer contiene el color de fondo que la fila debe tener, y se asigna con la función snprintf(), la cual imprime un template string en una variable dada (similar a printf(), pero con la diferencia de que se puede decidir dónde imprimir el string).

La interfaz de usuario es una máquina de estados (implementada con una sentencia switch) que loopea y espera que se le ingresen caracteres mediante scanf. En caso de que se ingrese el carácter ‘q’, el programa sale del while de la máquina de estados y finaliza su ejecución.

**Manual de usuario:**

El sistema Operativo utilizado es Lubuntu.

Cuando se ejecuta el programa, se mostrará el siguiente menú:



El usuario deberá entonces elegir una de las opciones escribiendo la letra correspondiente **en mayúscula**. Caso contrario, el sistema arrojará un mensaje de error “Opción incorrecta”. En el caso de los reportes csv y html, si el archivo no existe lo generará, y si existe, lo sobreescribirá.

Este proceso puede repetirse infinitamente hasta que el usuario seleccione la opción ‘q’, la cual terminará el programa.

**Casos de prueba:**

1. Generar reporte por terminal.
   1. Inicio el programa
   2. Selecciono la opción A
   3. Se imprime en pantalla el reporte de las especies cuya variación es negativa.
2. Generar reporte csv.
   1. Inicio el programa
   2. Selecciono la opción B
   3. Se genera un archivo .csv en el directorio donde se encuentra el ejecutable, con todas las entradas en formato .csv
3. Generar reporte html.
   1. Inicio el programa
   2. Selecciono la opción C
   3. Se genera un archivo .html en el directorio donde se encuentra el ejecutable, con todas las entradas cuya variación es negativa en formato .html. Si alguna entrada tiene un valor de apertura mayor que el de venta y el de compra, se marcará en color verde.
4. Salir del programa
   1. Inicio el programa.
   2. Selecciono la opción q
   3. El programa se termina con el mensaje “Hasta luego!”.