Laboratorio 1: Estructuras de datos y algoritmos

Integrantes: Hugo Rojas e Isidora González.

Profesor: Yerko Ortiz.

Profesor de laboratorio: Mauricio Hidalgo.

En el siguiente informe se realizó una lectura y organización de datos desde un archivo CSV. el cual significa “Comma Separated Values” en español “valores separados por coma” , donde fueron entregados 3 códigos para poder ser revisados, corregidos y optimizados. El primer código “**SplitStringCSV**” era requerido para separar por comas y comillas los datos entregados, el segundo código “**ReadCSV**” se encarga de leer los datos sin hacer alguna separación de estos y por último el tercer código “**ReadAndSplitCSV**” el cual junta ambos procesos para leer y separarlos en un arreglo multidimensional.

Una vez descargados los códigos fueron analizados individualmente, el primer código a analizar fue “**SplitStringCSV**”; Como fue mencionado anteriormente, en este código había que separar los datos según las comas y comillas que tenía y al compilar el código entregaba bastantes comas o comillas extra en su “**Print**”, por lo que se realizaron cambios para simplificar la lectura de datos, principalmente desde la línea diecisiete.

* Se cambia el segundo “**if**” por un “**else if** ” debido a la redundancia de la condición “**if**”, con sólo colocar un “**else if**” y acortar las condiciones basta.
* El segundo cambio se presenta dentro del segundo “**if**”, más específicamente en la línea 22, donde se realizan modificaciones para cumplir con la tipografía solicitada debido a que en el código original se presentan errores al compilar, el cambio a realizar se encuentra específicamente en la línea que contiene “**newLines.add(string.substring(start,i)**”, donde se elimina el “**+1**” presente en el código, al solucionar este error se presenta por pantalla lo solicitado y se modifica específicamente la condición de término “**del** **substring**”, donde en el código inicial corta la frase una posición más adelante de donde debería ser la condición más uno y se arregla.
* El tercer cambio se presenta en la **línea 24**, donde se iguala la variable “**start**” a “**i-1**”, principalmente se modifica la condición que resta uno ”; aparte se agrega un “**if**”, el cual revisa la oración buscando una coma, en el caso que la encuentre la variable “**start**” se igualará a “**i+2**”.
* Como cuarto cambio se mueve la declaración de la variable “**inQuotes = false**” a la línea 27 como condición final del while y además se agrega una igualación de la variable “**start = 0**” debido a que esta variable es la que cuenta el tamaño de la fila y si no se reinicia se presentaría un error.

Luego de haber realizado los cambios en el primer código se procede a revisar “**ReadCSV**” para leer de manera correcta los datos solicitados; En este código se realizan principalmente cambios en la sintaxis del programa debido a que este no compila.

* El primer cambio se realiza en la línea ocho donde se elimina el parámetro arraylist del “**new ArrayList<ArrayList<String>>()”** debido a que se puede abreviar.
* El segundo cambio realizado es en la línea catorce, específicamente al **string[i]** al cual se le elimina el **[i]** y se le agrega un “**.charAt(i)**” debido a que el método utilizado en el código original no es el correcto.
* El tercer cambio realizado es en la línea 22, donde se modifica la forma de guardado de los datos en el dataset, eliminando la variable counter y el contador, ya que no se llegaba a utilizar esa variable en todo el código, por lo tanto, se agrega un método para guardar los datos ya editados en el arraylist “**newLines**” para posteriormente agregar los datos de “**newLines**” al dataset.
* El cuarto cambio se realiza en la línea veintisiete donde se agrega un return dataset debido a que la función pide que retorne el dataset.
* El último cambio se presenta en la línea de código treinta y cuatro en el cual se edita “**dataset[0]”**, se elimina el “**[0]”** y se agrega después del dataset un **“.get(0)”** por el error de sintaxis.

Código ReadAndSplitCSV:

En este código podemos presenciar los dos códigos anteriores juntos excepto que éste no compila y tiene bastantes errores, por lo tanto, al momento de editar los códigos anteriores por separado y luego reemplazarlos en el código principal se logra obtener un buen resultado; dadas las mejoras mencionadas anteriormente, los datos mostrados por la consola del compilador mostrarán el archivo CSV corregido con un total de 23.147 filas y 8 columnas; dada la complejidad del código se decide simplificar sintaxis de manera que el programa completo sea más simple, expresivo y eficiente, de tal manera que se logró encontrar la forma más óptima y simple para poder leer todos los datos, ya que en un principio el programa rescataba sólo una pequeña parte de estos o daba errores.