

### **Detalles generales:**

A grandes rasgos, el objetivo de este programa es realizar un escrutinio sobre el texto que el usuario desea, luego del cuál el programa le dirá al usuario cuantas letras, palabras, líneas y cuantos párrafos posee el texto. Este programa fue primeramente representado mediante un autómata reconocedor finito, luego esto se vio volcado en el programa utilizando las herramientas brindadas por el lenguaje Assembly para la arquitectura i386 y la compilación mediante el compilador yasm, junto con el debugging necesario realizado con el DD Debugger. A continuación más detalles:

El programa presenta algunas limitaciones a considerar:

- El archivo a escrudiñar debe ser de extensión .txt para el correcto funcionamiento del programa, en caso de optar por analizar un archivo existente.
- En caso de optar por escudriñar un texto ingresado por consola de debe presionar Ctrl + D dos veces si se desea finalizar el input en una letra o carácter distinto del salto de línea, o una sola vez en caso contrario.
- Será ejecutable únicamente en una computadora que cumpla con la arquitectura antes nombrada.
- Al haber hecho el programa considerando únicamente el código ASCII no extendido debemos tener en cuenta que las siguientes letras del alfabeto castellano: ñ, Ñ, á, Á, í, Í, ú, Ú, é, É, ó, Ó, así como las demás letras de cualquier alfabeto que no se encuentre en su totalidad representado en el código ASCII no extendido, serán consideradas como separadores.

Dentro de los menesteres propios de la codificación del programa podríamos nombrar:

- La utilización de etiquetas para la simulación de los estados pertenecientes al autómata finito antes mencionado.
- La utilización de numerosas etiquetas denotando tanto la simulación de funciones como de variables.
- La utilización de un único buffer que se encargaba de leer carácter a carácter del input para realizar el escrutinio del texto.
- La repetición de un segmento de código que nos fue imposible modularizar, esto se debe a que en cada etiqueta se debía realizar la misma verificación, revisando que el carácter leído fuera un separador, una letra, un salto de línea o el finalizador de archivo pero que dependiendo de en cual etiqueta se estaba haciendo la revisión, se debía dirigir a etiquetas distintas.

### **Comentarios personales:**

Si bien la dificultad del proyecto no era ni exageradamente fácil ni difícil, dada la complejidad del lenguaje objetivo y los escasos ejemplos específicos en internet el proyecto involucró un considerable número de horas de investigación para lograr alcanzar las soluciones a los distintos problemas surgidos durante la codificación, por ejemplo que el programa no leyera bien los archivos o que ciclara infinitamente por un error al intentar leer el fin de archivo como un carácter y no mediante la comparación del valor de `eax`. Un problema que no logramos resolver a pesar de probar diferentes cosas fue el escribir sobre el archivo de salida, al finalizar el programa este continúa completamente vacío sin haber sufrido modificación alguna; obviando ese error fuimos capaces de completar todas las demás metas propuestas por el proyecto.

Creemos innecesario añadir una sección de ayuda dentro del manual ya que el programa mismo brinda una sección de ayuda que se puede consultar mediante la siguiente invocación: `./metricas -h`.

Debido al tamaño del autómata no fue posible añadir una imagen del mismo al presente documento, sin embargo, mediante el siguiente link se puede acceder a ella y a la explicación que la acompaña:

<https://drive.google.com/open?id=1yHYdUz3P2jjgxMWSLQYJhYqKArj1x3Je>