Proyecto en Assembler

Organización de Computadoras, Segundo Cuatrimestre 2018

Dpto. de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur

Vega, Maximiliano Nicolás – Zanardi, Franco Iván

1. Estructura y contenido

1.1 Definiciones y especificación de requerimientos.

1.1.1 Definición general del proyecto de software

El sistema desarrollado consiste en calcular las métricas de un texto, teniendo la posibilidad de ingresarlo por consola o cargarlo desde un archivo, además también también se puede cargar el texto desde un archivo y guardar el resultado en otro.

Por calcular las métricas se entiende contar la cantidad de letras, palabras, saltos de líneas y párrafos a partir de un texto ingresado por el usuario.

Fue desarrollado como segundo proyecto de la materia Organización de Computadoras, dictada por el Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional de Sur, durante el segundo cuatrimestre del año 2018.

El software junto con el presente informe están dirigidos a la cátedra de la asignatura.

1.1.2 **Especificación de requerimientos del proyecto**

El software desarrollado es original y no forma parte de desarrollos previos, se implementó en lenguaje ensamblador para la arquitectura i386 y hace uso de las llamadas al sistema provistas por el sistema operativo GNU/LINUX.

El programa cuenta con una limitación a la hora de ingresar texto por consola, se dispone de 1024 líneas como máximo, donde cada línea puede contener a lo sumo 1024 caracteres.

Todo texto ingresado al programa, para su correcto funcionamiento, debe contener caracteres pertenecientes al código ASCII básico.

Con respecto a los requisitos establecidos, el sistema cumple con ellos de manera satisfactoria, teniendo en cuenta las limitaciones y las pautas mencionadas a continuación.

Podemos definir formalmente qué se debe interpretar por *letra*, *delimitador, palabra*, *linea* y *párrafo:*

- *Letra*: son todas las letras del abecedeario inglés, tanto en mayúscula como en minúscula.
- *Delimitador*: se considera un delimitador a los siguientes caracteres:
    - Espacio, Tabulación, Salto de línea
    - Punto, Coma, Punto y Coma, Dos Puntos
    - Signo de exclamación que cierra, Signo de interrogación que cierra.
    - Paréntesis que cierra
    - Comilla simple, comilla doble.
- *Palabra:* se denomina palabra a toda secuencia de letras seguidas por un *delimitador*.

- *Párrafo*: se considera un párrafo a toda linea que contenga *al menos* una palabra.

- *Linea*: se considera como linea a cada aparición del caracter '\n'.

1.1.3 **Procedimiento de instalación y prueba**

El programa fue desarrollado en Linux Debian 32 bits, por lo que para su utilización se requiere dicho sistema operativo.

Para su desarrollo se hicieron uso de diversas herramientas, las cuales son:

- *GitHub*: Se utilizó para un desarrollo más controlado, para poder trabajar de manera separada y que los cambios puedan ser solventados de manera sencilla.

- *Vim*: Esta herramienta se utilizó para escribir el código fuente.

- *DDD*: Debugger utilizado para verificar el funcionamiento correcto del programa.

La estrategia empleada para la realización de los algoritmos del programa fue:

- En primera instancia se decidió de qué manera se iba a realizar el cálculo de las métricas, optando finalmente por un autómata finito determinista para procesar el texto.

- Se diseñó el autómata de forma tal que pueda ser utilizado para realizar el conteo de las métricas.

- Luego procedimos a implementar lo necesario para la escritura y lectura de archivos.

- Finalmente implementamos el autómata diseñado.

La obtención del software desarrollado se puede realizar a través del siguiente repositorio de *GitHub*: enlace*.*

Primero se debe clonar el repositorio utilizando una tecnología Git, luego habrá que compilar el código fuente y enlazar (vincular) el código objeto resultante. Para ello se debe ejecutar la terminal y estando ubicado en la carpeta clonada, ingresar las siguientes instrucciones:

yasm –f elf metricas.asm

ld –o metricas metricas.o

Luego ya se podrá ejecutar el programa bajo la siguiente instrucción:

./metricas <parámetros>

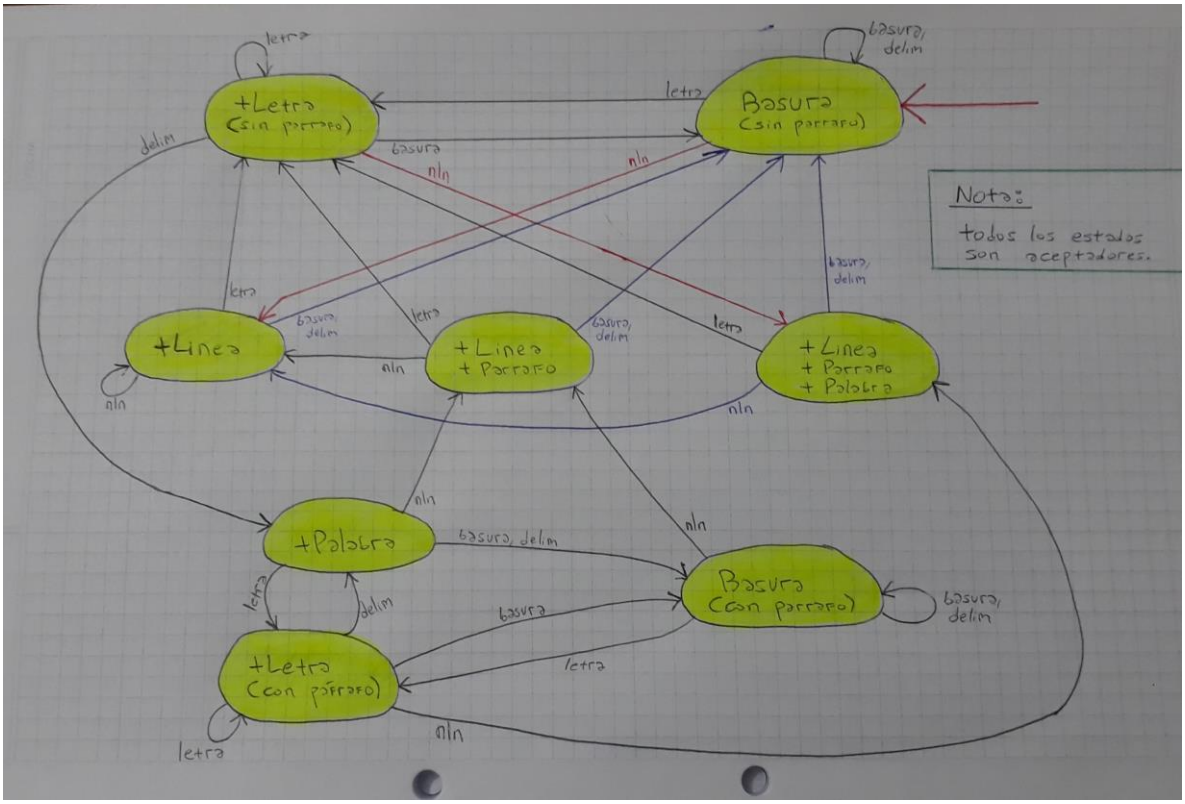
Se recomienda ingresar el argumento *–h* luego del nombre del programa en la ejecución, de esta manera podrá ver una guía rápida de ayuda que el mismo ofrece para su correcto uso.

1.2 **Arquitectura del sistema:**

El sistema está compuesto por un único módulo en lenguaje Assembly llamado *'metricas.asm'*. El cual es descripto a lo largo de todo el informe. No requiere librerías y hace uso únicamente de servicios provistos por el sistema operativo Linux 32 bits.

1.3 **Diseño de modelos de datos:**

El siguiente autómata modela el algoritmo para *calcular las métricas* de un texto (datos de entrada), utilizandolo podemos obtener los resultados de los conteos (datos de salida).



Aclaraciónes:

- El único fin en que algunas flechas tengan distinto color, es para distinguirlas mejor en zonas donde hay muchas intersecciones.
- Todos los estados son aceptadores.

1.4 **Descripción de procesos y servicios ofrecidos por el sistema:**

El sistema, como ya mencionamos, se encarga de calcular métricas, más precisamente contar la cantidad de letras, palabras, párrafos y líneas.

Ofrece distintas alternativas de entrada y salida, las cuales se pueden ver al invocar el programa con el argumento *–h* (ayuda).

A continuación se detalla las posibles invocaciones del programa.

- *metricas –h*: Muestra una ayuda sobre las diferentes alternativas de invocación que tiene el programa.
- *metricas:* Prepara la consola para que el usuario pueda ingresar el texto. Luego muestra por consola los resultados.
- *metricas archivo_entrada:* Calcula las métricas del texto que contiene el archivo *'archivo_entrada'* y muestra el resultado por la consola.
- *metricas archivo_entrada archivo_salida:* Calcula las métricas del archivo *'archivo_entrada'* y escribe el resultado en el archivo *'archivo_salida'*. Si este último no existe, lo crea con anterioridad.

1.5 **Documentación técnica – Especificación API:**

La documentación y especificación de cada procedimiento se encuentra en el código fuente del programa, exactamente encima de cada procedimiento, para facilitarle el trabajo el lector.

Cada procedimiento cuenta con:

- Una descripción de lo que hace
- Los *input* que que recibe
- Los *output* que produce.
- Junto con alguna nota u observación en caso que sea necesaria.