Definición y especificación

La finalidad del sistema es contabilizar la cantidad de apariciones de cada palabra en un archivo compuesto por todo tipo de caracteres. Se considerará como palabra toda secuencia de caracteres S tal que:

S=<c1,…,cn>, n>0

Ci{a,…,z} {á,é,í,ó,ú}, (0<<=).

Este sistema esta apuntado a personas que necesitan una solución a la hora de contabilizar las apariciones de palabras en un archivo con el formato mencionado anteriormente, estos usuarios para poder usar el sistema deben tener conocimientos de como ejecutar un programa por consola y como ingresar los parámetros que sean adecuados.

El sistema cumple con los siguientes requerimientos:

(como maneja el programa el ingreso del archivo, etc)

* Para contabilizar la cantidad de palabras, se debe diseñar e implementar un algoritmo no recursivo, que utilice el TDA Trie implementado anteriormente. Este algoritmo, además de resolver el problema, debe respetar las siguientes consideraciones:
* (restricciones del programa)

El sistema es un desarrollo origina y fue desarrollado en parte con el IDE Code::Blocks ( www . codeblocks .org/), el editor de texto Komodo Edit (http://komodoide.com/komodo-edit/) y compilado con el compilador GCC ( https:// gcc .gnu.org/). (es igual?)

Para poder ejecutar el programa primero se debe descomprimir el archivo comprimido que contiene el código fuente del sistema, cuyo archivo principal es “evaluador.c”. Alternativamente se puede clonar el repositorio ubicado en el siguiente enlace: <https://github.com/emarod/ProyectoC>.

El programa debe ser compilado sobre un sistema operativo GNU/Linux con arquitectura x32 para garantizar su funcionamiento. Para ser invocado desde la línea de comandos:

$ evaluador<archivo\_texto>

El parámetro **archivo\_texto** , indica el archivo a partir del cual se contabilizara la frecuencia de aparición de las palabras que lo componen. En caso de ingresar un parámetro erróneo se mostrará un mensaje indicando el error y se finalizará la ejecución del programa.

Arquitectura del sistema

El sistema está conformado por tres módulos separados en base a su funcionalidad:

• Evaluador

• Trie

• Lista Ordenada

• Lista

Cada modulo contiene la siguiente descripción:

* **Evaluador**: Contiene el programa principal que se ejecutara, y por ende en él se encuentra toda la implementación concerniente al manejo de los datos de entrada/salida y a la resolución propiamente dicha del problema definido anteriormente. Este módulo depende de los otros tres para funcionar, así también como de la librería de entrada/salida estándar (stdio) y la librería estándar (stdlib), todas provistas por el lenguaje de programación.
* **Trie**: Empaqueta la estructura de datos Trie, cuya arquitectura está compuesta por nodos, con rótulos de tipo *caracter* (char), que almacenaran , junto con todas las operaciones previstas por este TDA. Esta estructura utiliza la librería de entrada/salida estándar (stdio) y la librería estándar (stdlib).
* **Lista Ordenada**: Aquí se encuentra almacenada la estructura Lista Ordenada la cual permitirá almacenar elementos en forma ordenada mediante un comparador diseñado específicamente para este problema. Esta estructura utiliza la librería de entrada/salida estándar (stdio) y la librería estándar (stdlib).
* **Lista**: Estructura conformada por Celdas que almacenara elementos de tipo genérico. Esta estructura utiliza la librería de entrada/salida estándar (stdio) y la librería estándar (stdlib).

Todo el sistema está programado usando el lenguaje de programación C y compilado bajo el sistema operativo GNU/Linux. (chequear)

Diseño del modelo de datos

El sistema esta diagramado de la siguiente forma: