Proyecto 1

Guatemala, 8 agosto del 2021

Índice de contenido

1 Definición de la solución

1.1 Objetivos

Que el estudiante:

- Desarrolle una solución de software implementando un analizador léxico
- mediante autómatas.
- Aplique los conocimientos adquiridos en el laboratorio, del lenguaje de
- programación Python
- Desarrolle una interfaz gráfica utilizando el lenguaje Python.

1.2 Alcances del proyecto

- Los entregables para este proyecto se adjunta en el repositorio de Git Hub con toda su documentación, este proyecto de sube a un repositorio privado, para que en un futuro los desarrolladores se le pueda dar mantenimiento al sistema, con toda su documentación.
- El sistema está diseñado para ser ejecutado en consola, utilizando archivos de entrada.
- El sistema permite realizar ciertas operaciones de interés, las cuales son:
    - MIRRORX: Este filtro gira horizontalmente la imagen original.
    - MIRRORY: Este filtro gira verticalmente la imagen original.
    - DOUBLEMIRROR: Este filtro gira horizontal y verticalmente la imagen original.
- La información es almacenada en cada ejecución en la memoria RAM.
- El sistema es capaz de generar reportes en HTML con todos los datos obtenidos.

1.3 Requerimientos del sistema

1.3.1 Requerimientos funcionales

- **Cargar archivo**: Muestra una ventana emergente que permite al usuario seleccionar un archivo LFP y carga el archivo a memoria, lo analiza y ejecuta el reporte.
- **Mostrar reporte en consola**: Muestra los datos obtenidos durante la ejecución del reporte. Se debe mostrar en pantalla el nombre del curso, el total de estudiantes que contiene el archivo y los datos propios del reporte que se solicite.
- **Exportar reporte**: Se debe de escribir un archivo HTML con los datos del reporte generado, la manera en que los datos son mostrados deben de ser agradables al usuario. Por lo menos, debe mostrar el nombre del curso, en una tabla la lista de estudiantes y su respectiva nota (en rojo las notas reprobadas y en azul las aprobadas) y al final los parámetros solicitados.

1.3.2 Descripción

Se le ha designado a usted estudiante de Ingeniería en Ciencias y Sistemas como colaborador en el área de reportes de notas en Control Académico en la Facultad de Ingeniería. La tarea que se le ha asignado es desarrollar una aplicación que permita la lectura de un archivo de texto plano con extensión lfp que contiene los datos de los estudiantes de un curso y la nota final que obtuvieron en dicho curso, finalmente el archivo contiene una palabra reservada que indica que tipo de reporte se quiere realizar.

1.3.3 Ejemplo de un archivo de entrada

```
TITULO="Pokebola";
ANCHO=300;
ALTO=300;
FILAS=12;
COLUMNAS=12;
CELDAS = {
        [0,0,FALSE,#000000],
        [0,1,FALSE,#000000],
        [3,3,FALSE,#000000],
        [3,4,TRUE,#000000],
        [3,5,TRUE,#000000],
        [3,6,TRUE,#000000],
        [3,7,TRUE,#000000],
        [4,1,FALSE,#000000]
    };
FILTROS = MIRRORX;

@@@@

TITULO="Estrella";
ANCHO=300;
ALTO=300;
FILAS=4;
COLUMNAS=4;
CELDAS = {
        [0,0,FALSE,#000000],
        [1,1,FALSE,#000000],
        [3,3,FALSE,#000000],
        [2,1,FALSE,#000000]
    };
FILTROS = MIRRORX,MIRRORY,DOUBLEMIRROR;
```

1.4 Componentes

1.4.1 Descripción de componentes

1.4.2.1 helpers

Este archivo contiene todas las utilidades para generar y mostrar una imagen a partir de un objeto ImageEntry, los métodos que podemos encontrar son:

- **parse_true_false_color**: convierte todos los pixeles con valor FALSE, en color "#FFFFFF" (blanco).
- **create_image:** recibe un objeto ImageEntry y un string con el parámetro de transformación, si recibe "NORMAL" no aplica ninguna transformación, de lo contrario aplica la transformación indicada.
- **rotate_x, rotate_y, rotate_xy:** aplican las transformaciones indicadas sobre la imagen.
- **process_file:** lee el archivo he identifica los tokens respectivos
- **define_geometry:** función auxiliar utilizada por la ventana de tkinter.

1.4.2.2 opciones

El programa lee un conjunto de transformaciones, las cuales pueden ser aplicadas individualmente a la imagen, estas transformaciones se muestran en un menú, es decir únicamente se podrán aplicar las transformaciones que previamente se hayan definido en el archivo de entrada.

1.4.2.3 Lectura de archivo

El programa contiene un conjunto de palabras reservadas, las cuales permite asignar ciertas propiedades al sistema, en forma de pares clave-valor, de la siguiente forma:

PALABRA_RESERVADA=VALOR;

Siendo PALABRA_RESERVADA, un lexema del siguiente conjunto:

- TITULO: contiene el título de la imagen.
- ANCHO: contiene el ancho en pixeles de la imagen.
- ALTO: contiene el alto en pixeles de la imagen.
- FILAS: contiene el numero de filas en los que se divide la cuadricula de pixeles del archivo.
- COLUMNAS: contiene el numero de columnas en los que se divide la cuadricula de pixeles del archivo.
- CELDAS: contiene un listado de las celdas, su valor tiene la siguiente forma: { [num, num, bool, color], …[num, num, bool, color]}
- FILTROS: contiene un listado de filtros aplicables a la imagen, separados por comas.

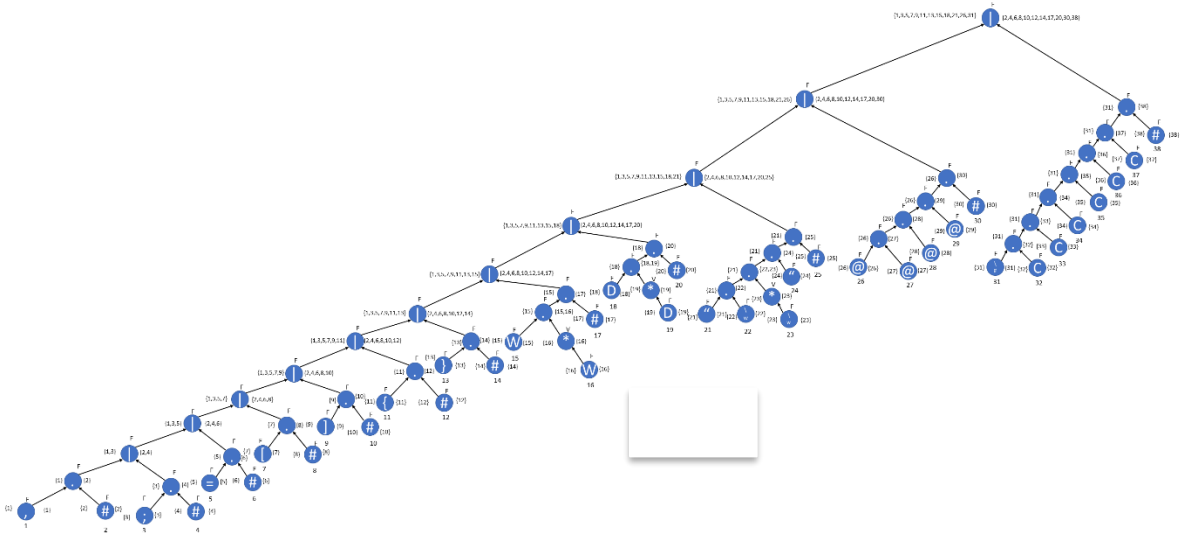
Cada celda esta formada por 2 números que indican las coordenadas, un valor booleano (TRUE, FALSE) que indica si esa celda debe ser dibujada o no y el color de la celda en cuestión además del color el cual tiene el formato: “#CCCCCC”, donde C es una expresión regular definida como: “[A-Z0-9]”.

1.5 Autómata finito determinista

1.5.1 Lista de Tokens

S=Símbolos=[,;={}\[\]]
W=Palabras Reservadas=[A-Z]
D=Dígitos=[0-9]
String=”\w+”
Separador=@@@@
Color=#[A-Z0-9]{6}

1.5.2 árbol de sintaxis



1.5.2 autómata finito determinista