

Universidad De San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

Lenguajes Formales y de Programación
Sección "B-"



MANUAL TÉCNICO:

Reportes lfp

Estudiante: José Ernesto Pajoc Raymundo

Carné: 201115455

Segundo semestre del 2,021

Presentación del proyecto: 28/10/2021

Objetivos

General:

Presentar las clases, métodos, funciones, librerías y diagramas que conforman al proyecto para el funcionamiento adecuado del software.

Específicos:

- Mostrar la descripción sobre la plataforma en la cual fue desarrollado el software.
- Identificar las librerías, funciones, métodos y atributos que conforman la parte principal del programa.
- Identificar las clases que utiliza el software para generar reportes.

Introducción

El programa corresponde al proyecto 2 del curso Lenguajes Formales y de Programación, el cual consiste en desarrollar una aplicación en el lenguaje de programación Python, con el cual se carga un archivo de texto plano con extensión “.lfp” a un editor de texto para luego ser analizado por un autómata finito implementado por el programador, obteniendo de esa forma los parámetros necesarios para poder generar los reportes y ejecutar instrucciones que podrán ser visualizadas en la consola según información solicitada por el usuario.

Información técnica

El software se desarrolló utilizando las siguientes tecnologías:

- Sistema operativo: Windows 10 (64 bits).
- Lenguaje utilizado: Python 3.8
- IDE: Visual Code, Versión: 1.56.2
- Navegador web: Google Chrome Versión 92.0.4515.159

Descripción del problema

El software permite analizar y realizar reportes de datos para la toma de decisiones futuras, además se pueda aplicar a cualquier tipo de negocio, generando información interesante para pequeñas empresas, en el cual los datos a cargar son claves y registros para luego poder ejecutar cualquiera de las siguientes instrucciones:

- Imprimir mensajes
- Realizar conteo de registros.
- Contar registros que cumplan un criterio.
- Visualizar los datos.
- Sumar campos numéricos.
- Obtener el valor máximo de un campo numérico.
- Obtener el valor mínimo de un campo numérico.
- Exportar reportes en archivos HTML.

Descripción de la Solución

Para poder desarrollar el software se analizó primero la información y el tipo de archivo que se debe cargar al sistema, con ello se crearon expresiones regulares las cuales permiten extraer los datos de las claves, registros y funciones para continuar con el proceso de la creación del árbol binario y luego dar paso a la elaboración de un autómata finito determinista el cual es programado dentro de la aplicación.

Donde:

$L = [a-zA-ZñÑ]$

$D = [0-9]$

$I = [\text{carácter ASCII imprimible en los rangos de } 33 - 39, 42 - 47, 128 - 239, 58 - 60, 62 - 63]$

Tabla I

Tokens implementados.

Token	Expresión regular
Claves	$L+=[“(L D I)+”(,“(L D I)”)*]$
Registros	$L+=[(\{ (L D)+ (, (L D))^* \})+]$
Conteo	$L+();$
Promedio	$L+("L+");$
Contarsi	$L+("L+", (L D)+);$
Datos	$L+();$
Max	$L+("L+");$
Min	$L+("L+");$
Sumar	$L+("L+");$
ExportarReporte	$L+("(L D I)+");$
Imprimir	$L+("(L D I)+");$
ImprimirLn	$L+("(L D I)+");$

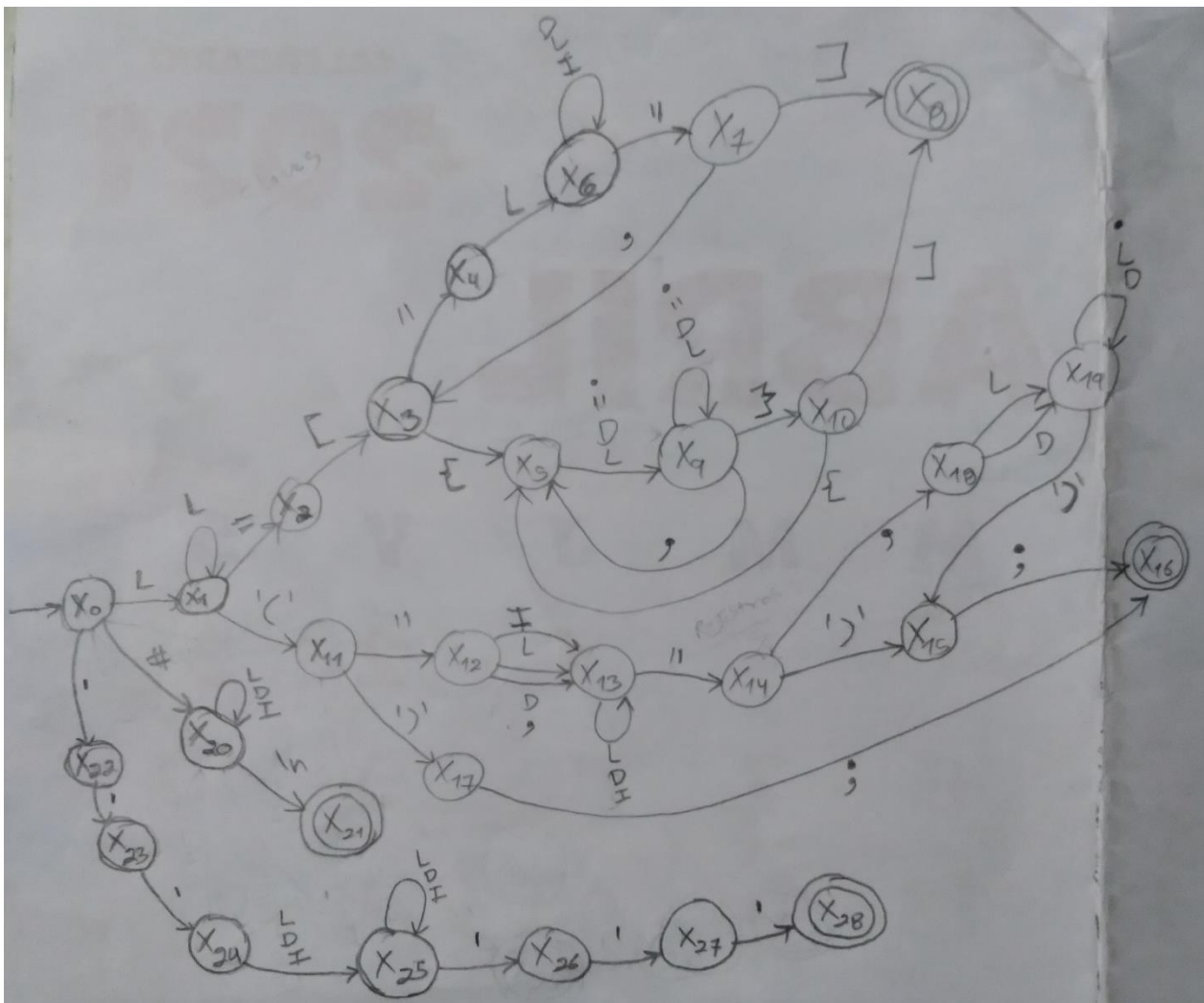
Fuente: elaboración propia.

Las expresiones regulares comparten el mismo inicio y varias características, es por ello que se unieron en una sola expresión regular para crear el árbol binario.

Expresión regular unificada:

$$(L+([(“(L|D|I)+”(,“(L|D|I)”)* | (\{ (L|D)+ (, (L|D))^* \})+]) | (); | (“L+”); | (“L”, (L|D)+); | (“(L|D|I)+”);))\$$$

Autómata



Luego de tener claro el autómata, se utilizó dos archivos dentro del proyecto para poder manipular los datos, los archivos se describen a continuación:

Tabla II

Componentes del archivo reportes.

reportes.py	
Librerías utilizadas	datetime: esta librería permite obtener la fecha y hora del sistema.
Clase	reporte: se utiliza para crear los objetos de tipo reporte con los atributos claves, registros y ahora. Los métodos que posee son verDatos, crearReporte los cuales son utilizados para crear archivos HTML con la información que fue analizada.

Fuente: elaboración propia.

En el archivo principal se crea la interfaz gráfica, esta cuenta con tres botones principales, un cuadro de selección, un área de texto para editar el código y una consola la cual muestra los resultados de la ejecución que no podrán ser editados.

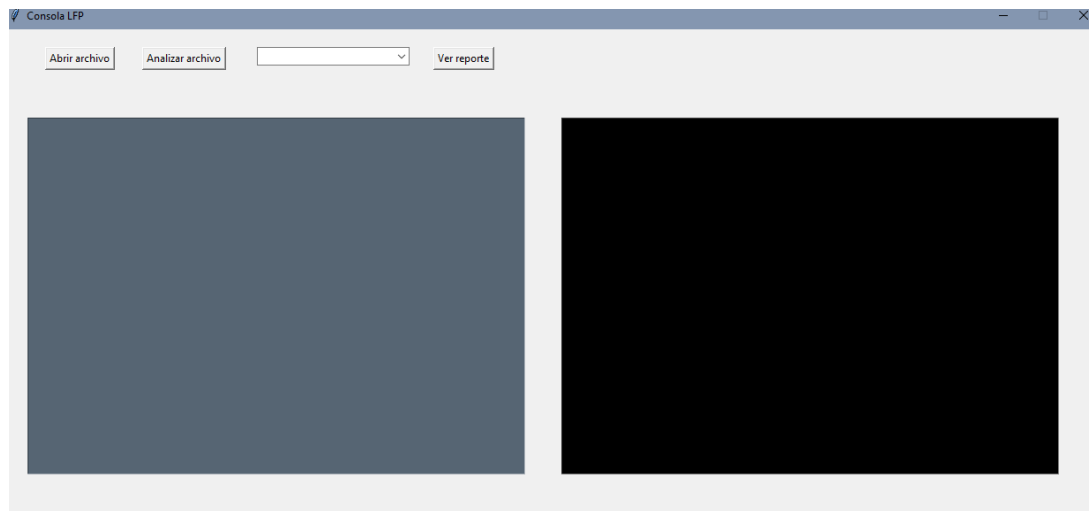


Figura 1: Interfaz gráfica del usuario.

Tabla III

Componentes del archivo main.

main.py	
Librerías utilizadas	<p>tkinter: esta librería permite crear interfaz gráfica, con la cual el usuario puede interactuar a través de marcos, paneles, botones, etiquetas, listas desplegables y cuadros de mensaje.</p> <p>re: el uso de esta librería es exclusivamente para verificar la extensión del archivo de texto plano, la cual debe ser únicamente “.lfp”.</p> <p>reportes: se utiliza para crear objetos de tipo reporte.</p>
Variables globales	<p>archivoCargado: almacena la información del archivo de texto plano que es cargado al sistema.</p> <p>cargarArch: se utiliza como variable booleana para saber si se cargó el archivo de texto plano.</p> <p>reservadas: posee una lista con todas las palabras claves que utiliza el sistema.</p> <p>claves: lista la cual almacena todos los campos.</p> <p>registros: lista la cual almacena todos los registros.</p>

	<p>clavesCargadas: se utiliza como variable booleana para saber si se cargó las claves al sistema.</p> <p>registrosCargados: se utiliza como variable booleana para saber si se cargó los registros al sistema.</p>
Funciones	<p>esLetra: convierten un carácter a su equivalente entero ASCII y verifica si se encuentra entre el rango de 65 y 90, o 97 y 122, o es igual a 165 o 164, retorna TRUE o FALSE según el caso.</p> <p>esNumero: convierten un carácter a su equivalente entero ASCII y verifica si se encuentra entre el rango de 48 y 57, retorna TRUE o FALSE según el caso.</p> <p>imprimible: convierten un carácter a su equivalente entero ASCII y verifica si se encuentra en los rangos de 33 – 39, 42 – 47, 128 – 239, 58 – 60, 62 - 63, retorna TRUE o FALSE según el caso.</p> <p>esComa: verifica si el carácter es una coma, retorna TRUE o FALSE según el caso.</p> <p>datos: muestra en la consola las claves y registros que se encuentran almacenadas en el sistema.</p> <p>promedio_suma: muestra por consola el promedio o la suma de un campo según el criterio solicitado por el usuario y la instrucción.</p> <p>funcion_max: muestra por consola el valor máximo de un campo numérico.</p> <p>funcion_min: muestra por consola el valor mínimo de un campo numérico.</p> <p>contar_si: muestra por consola la cantidad de elementos que existen en un campo, el usuario debe ingresar un criterio y un valor para poder utilizar la función.</p> <p>exportar_reporte: crea un nuevo archivo HTML el cual posee una tabla con todos los datos del sistema, para ello se necesita que el usuario ingrese un criterio que será utilizado como título del archivo y de la tabla.</p> <p>analizar: recorre carácter a carácter toda la información cargada en el editor de texto de la interfaz gráfica, donde se hace uso del autómata con sus 19 estados, donde los estados de aceptación son el estado 8 y el estado 16, en estos estado se valida que el lexema sea válido y se eliminan los títulos de las cadenas para solo dejar los datos de las claves y los registros para luego enviarlos a sus respectivas listas, pero en el caso de las instrucciones solicitadas por el usuario se hace llamado a la función respectiva.</p>

	<p>abrirArchivo: verifica que el archivo que selecciono el usuario sea con extensión “.lfp”, si esto es correcto entonces extrae la información y la muestra en el editor de texto de la interfaz gráfica.</p> <p>leerCodigo: extrae la información del editor de texto, para concatenar al final un símbolo que representa el final del contenido, luego hace llamado a la función de analizar y envía como parámetro los datos.</p> <p>ventana_inicial: crea una nueva ventana la cual será utiliza como interfaz gráfica para el usuario, en ella se inserta el panel principal, botones, cuadro de selección y áreas de texto.</p>
--	--

Fuente: elaboración propia.

Observaciones

- La carpeta “reporte html” no deben ser eliminada, ya que es utilizada por el software.
- Los reportes HTML se generan dentro de la carpeta con el nombre “reporte html”.
- La función contarsi necesita dos parámetros, el primero representa el nombre del campo, este desde ir encerrado entre comillas, en el caso del segundo parámetro el cual es utilizado para el criterio, si este es una cadena de texto debe ir encerrado entre comillas pero si el criterio es un valor numérico no es necesario encerrarlo entre comillas.