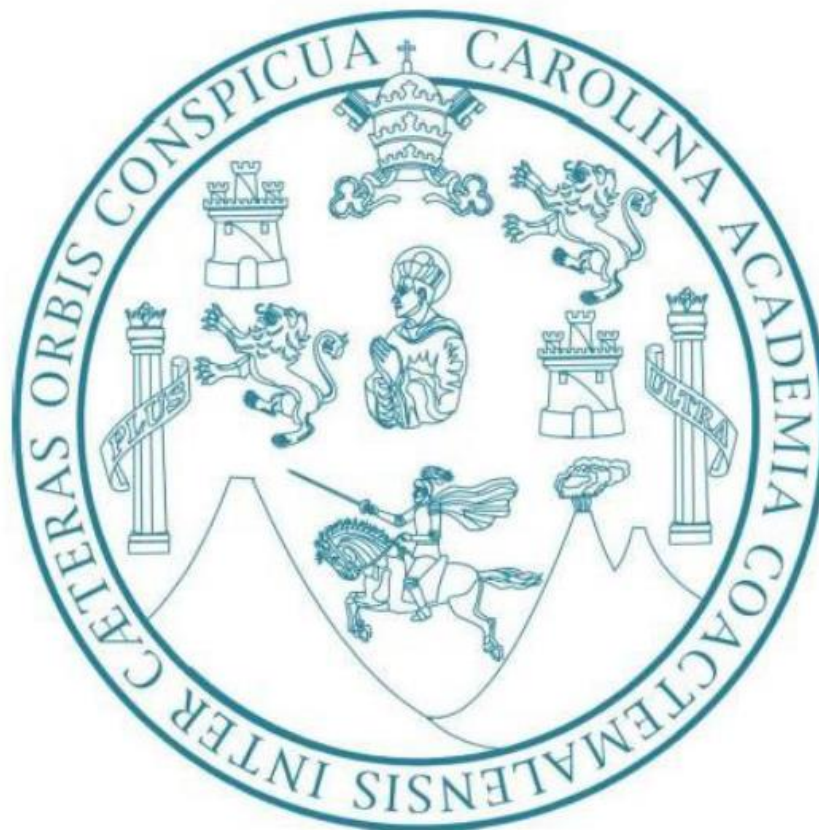


Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Lenguajes Formales y de Programación
Primer Semestre 2022
Catedrático: Viviana Campos
Tutor Académico: Cesar David Juárez



Manual Técnico

Nombre: Hesban Amílcar Argueta Aguilar
Carné: 201704787

Contenido

| | |
|---|----|
| INTRODUCCIÓN: | 3 |
| REQUERIMIENTOS DEL SISTEMA..... | 4 |
| Windows | 4 |
| Mac OS X..... | 4 |
| Linux | 4 |
| Menú | 5 |
| Cargar formulario..... | 6 |
| Generar tablas de tokens/errores | 8 |
| PARADIGMAS DE PROGRAMACIÓN UTILIZADAS | 9 |
| Programación estructurada..... | 10 |
| Programación procedimental | 11 |
| Programación modular | 12 |
| Programación Orientada a Objetos..... | 12 |
| CONCLUSIÓN: | 15 |

INTRODUCCIÓN:

Se realizó una aplicación, que permite analizar y procesar información de un archivo de texto dado con un formato, el cual contiene una estructura y en base a esa estructura se identifican los diferentes patrones para extraer la información que contiene y con eso se generan reportes, se hizo a través del lenguaje Python, se hizo uso de HTML y CSS para generar reportes web y también se hizo uso de librerías externas como Tkinter para mostrar un explorador de archivos.

REQUERIMIENTOS DEL SISTEMA

Windows:

- Python 3.8.1
- Windows 10 (8u51 y superiores)
- Windows 8.x (escritorio)
- Windows 7 SP1 • Windows Vista SP2

Mac OS X:

- Python 3.8.1
- Mac con Intel que ejecuta Mac OS X 10.8.3+, 10.9+
- Privilegios de administrador para la instalación

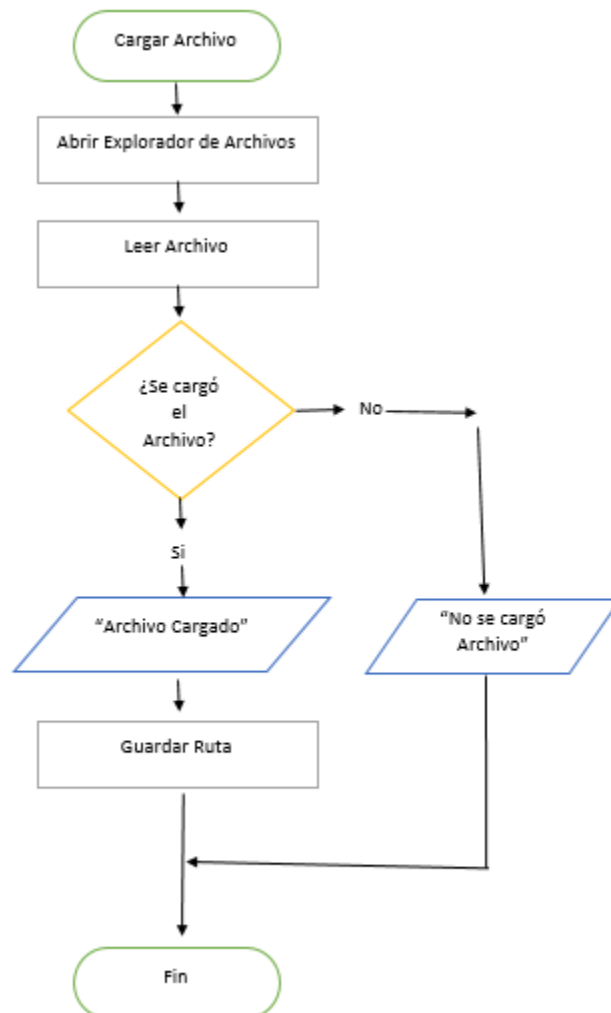
Linux:

- Python 3.8.1
- Cualquier distribución de Linux.

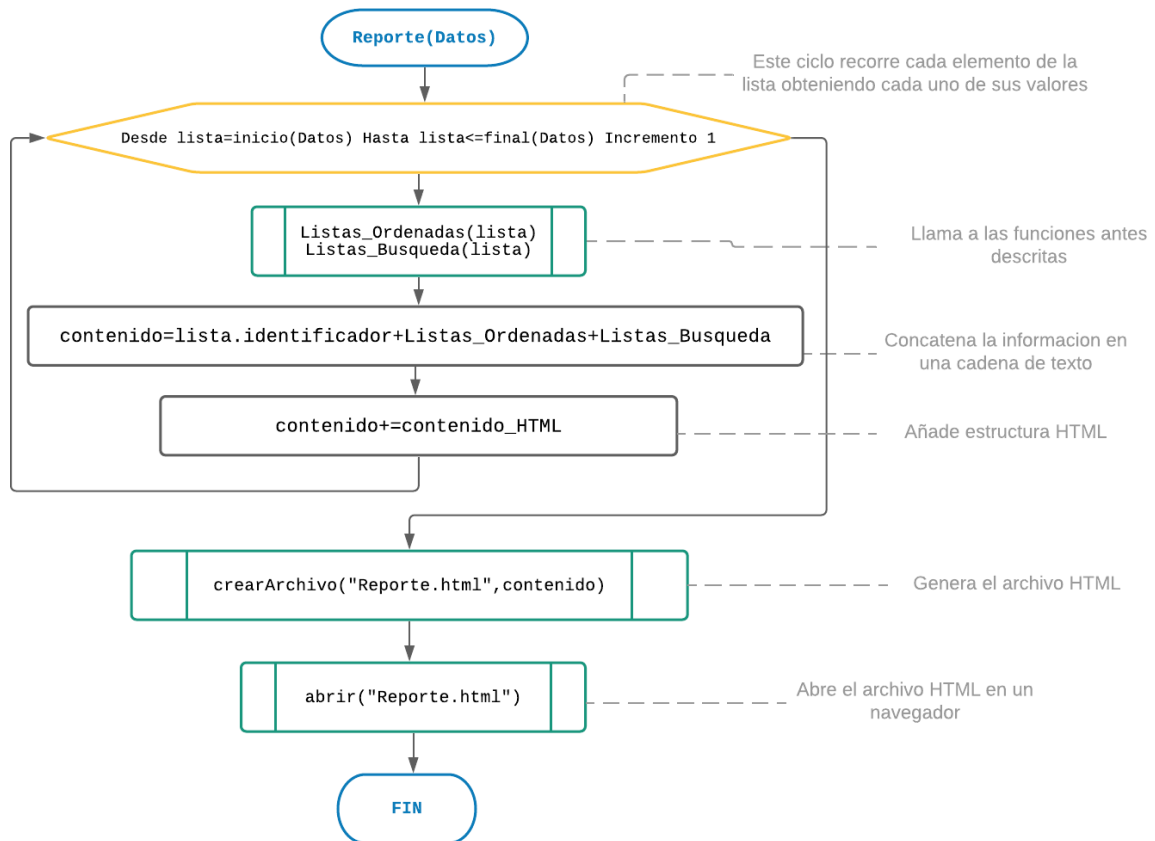
En cualquiera de los sistemas que vaya a ejecutar la aplicación se debe contar con un navegador que soporte HTML 5, para poder visualizar el reporte.

- Python en la versión 3.9 en adelante, puede descargarlo si no lo tiene desde su pagina oficial: <https://www.python.org/downloads/>.

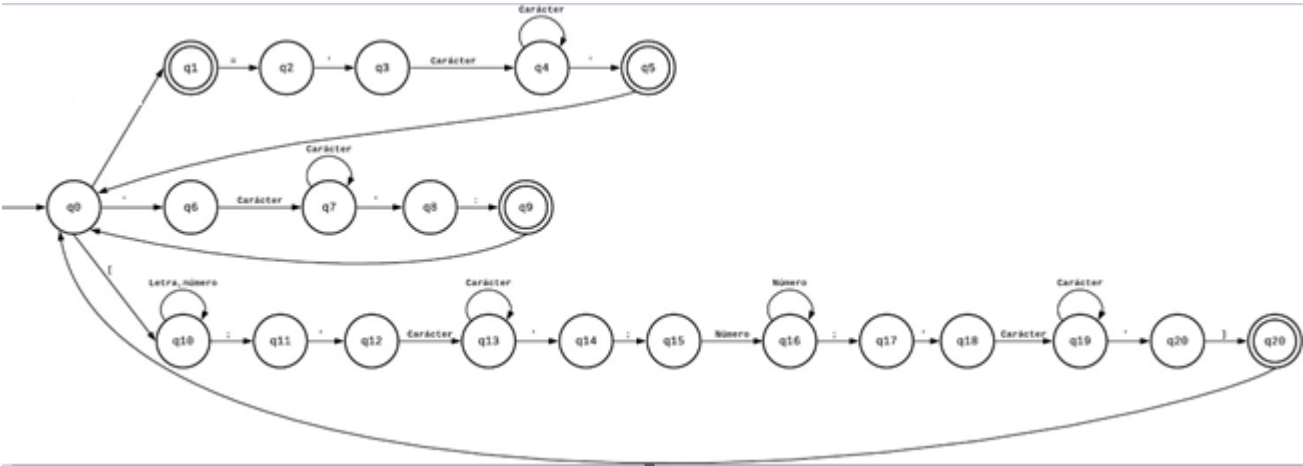
Cargar Formulario:



Generar tablas de tokens/errores:



AUTÓMATA FINITO DETERMINIS PARA LEER LA ESTRUCTURA DEL FORMULARIO



PARADIGMAS DE PROGRAMACIÓN UTILIZADAS

Programación estructurada

```
32     def analizar(self):
33         estado = 0
34         posicion = 0
35         columna = 1
36         string = ""
37         noIdentificados="~!<>!@#$$%^&*()\, _+-|/¿¡?{[]}"
38         noIdentificado2="~!<>!@#$$%^&*()\,+-|/¿¡?{[]}"
39         longitud = len(self.texto)
40         tipo1 = None
41         etiqueta = None
42         valor1 = None
43         nombre1 = None
44         while posicion<longitud:
45
46             caracter = self.texto[posicion]
47             if estado == 0:
48                 if caracter == "f":
49                     estado = 1
50                     string+= caracter
51                     posicion+= 1
52                     columna+= 1
53
54                 elif caracter == " ":
55                     posicion+= 1
56                     columna+=1
57
58                 elif caracter == '[':
59                     posicion+= 1
60                     columna+=1
61
```


Programación procedimental

```
#Funciones
def buscarPalabraReservada(self):
    encontrado = False
    for buscar in self.ListaTokens:
        if buscar.lexema == "formulario":
            encontrado = True

    if encontrado == False:
        aux = Error("formulario", "0", "0", "No existe la palabra reservada")
        self.ListaErrores.append(aux)

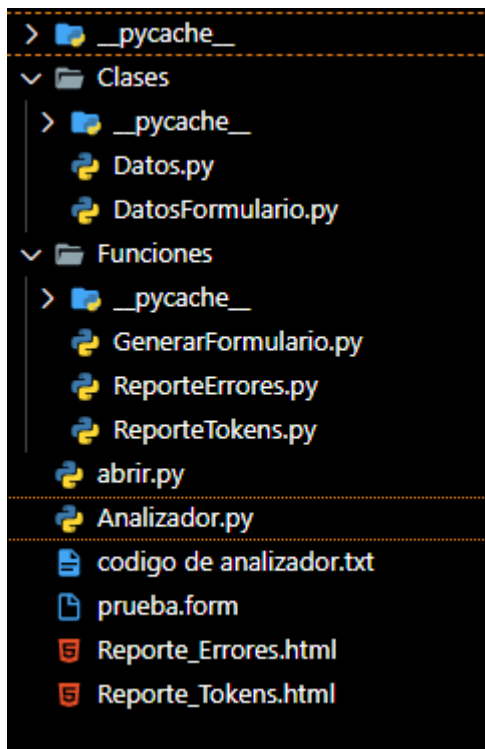
def buscarIdentificador(self, id):
    encontrado = False
    for buscar in self.ListaTokens:
        if buscar.lexema == id:
            encontrado = True

    return encontrado

def imprimirTokens(self):
    cont=1
    for tokens in self.ListaTokens:
        print(cont,tokens)
        cont+=1

def imprimirErrores(self):
    for errores in self.ListaErrores:
        print(errores)
```

Programación modular



Programación Orientada a Objetos

```
1 class Datos:
2     def __init__(self, lexema, linea, columna, token):
3         self.lexema = lexema
4         self.linea = linea
5         self.columna = columna
6         self.token = token
7
8     def __str__(self):
9         string = str("Lexema: ") + str(self.lexema) + str("Línea: ") + str(self.linea) + str("Columna: ") +
10         return string
11
12 class Error:
13     def __init__(self, lexema, linea, columna, descripcion):
14         self.lexema = lexema
15         self.linea = linea
16         self.columna = columna
17         self.descripcion = descripcion
18
19     def __str__(self):
20         string = str("Línea: ") + str(self.linea) + str("Columna: ") + str(self.columna) + str("Carácter: ")
21         return string
```

Estructura y formato de los archivos

Formulario

Nombre del formulario:

El nombre del formulario vendrá acompañado con los caracteres ~>>, después un corchete [.

```
1  formulario ~>> [
2  |  <
```

Opciones del formulario:

Las opciones del formulario que tiene cada sección (tipo, valor, fondo, evento, etcétera) vendrán entre el signo > uno en cada línea. Donde vendrá primero su tipo único, nombre, seguido de otros campos. Esta información vendrá separada por comas. El nombre y la descripción vendrán entre comillas simples.

```
2  |  <
3  |      tipo: "etiqueta",
4  |      valor: "Nombre:"
5  |  >,
```

Ejemplo de un archivo:

```
1  formulario ~>> [  
2      <  
3          tipo: "etiqueta",  
4          valor: "Nombre:"  
5      >,  
6      <  
7          tipo: "texto",  
8          valor: "Nombre:",  
9          fondo: "Ingrese nombre"  
10     >,  
11     <  
12         tipo: "grupo-radio",  
13         valor: "sexo",  
14         valores: ['Masculino' , 'Femenino']  
15     >,  
16     <  
17         tipo: "grupo-option",  
18         valor: "pais",  
19         valores: ['Guatemala', 'El Salvador', 'Honduras']  
20     >,  
21     <  
22         tipo: "boton",  
23         valor: "VALOR",  
24         evento: "Entrada"  
25     >  
26 ]
```

CONCLUSIÓN:

- Es necesario establecer un formato y estructura al archivo que se desea analizar y procesar ya que con ello se pueden crear algoritmos que ayudaran a realizar tales acciones.
- El explorador de archivos hace más fácil la búsqueda y carga de archivos.
- Los reportes permiten visualizar de forma detallada cierta información.

