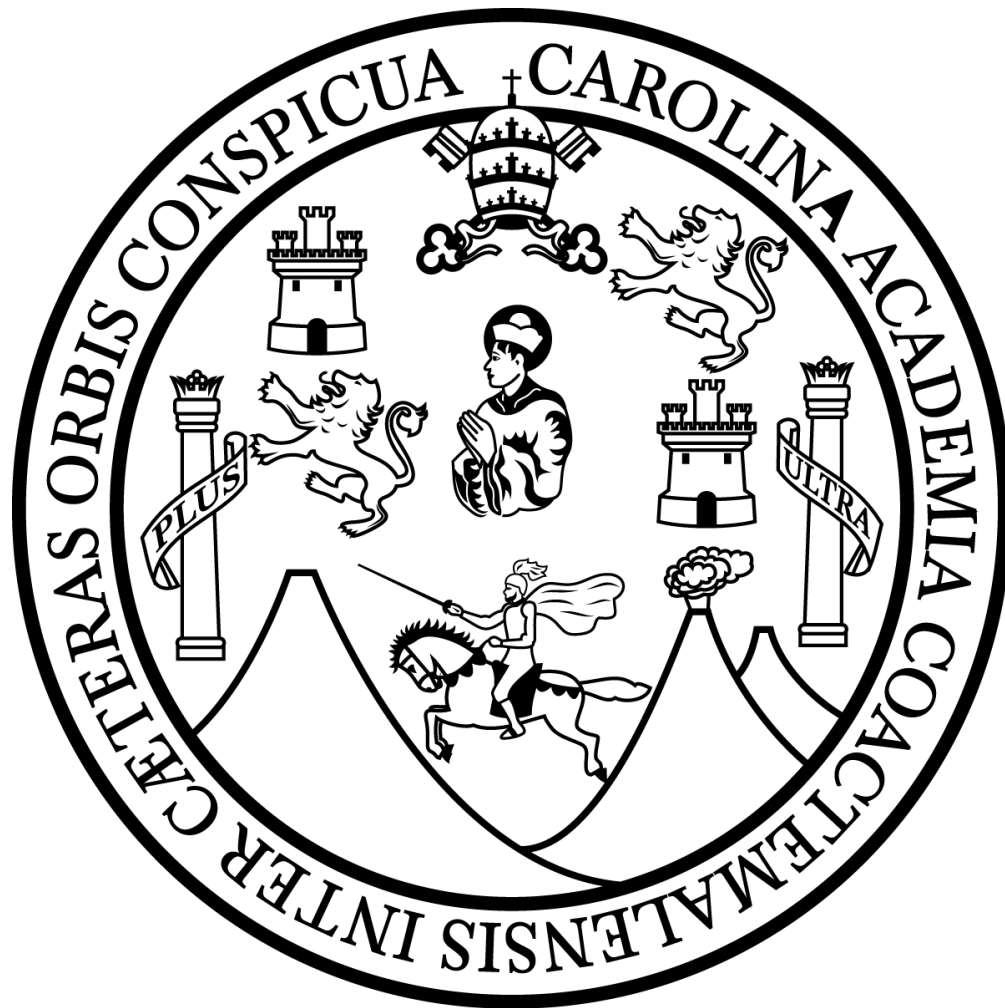


UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERIA EN CIENCIAS Y SISTEMAS  
LENGUAJES FORMALES DE PROGRAMACION  
SECCION "B+"  
PROYECTO #2



---

MANUAL DE TECNICO

---

JUAN PABLO GONZALEZ LEAL

201901374

30/04/2021

## Super autómeta

Este es un software que se realizó a través del lenguaje de programación Python, con la finalidad de profundizar en el tema de los lenguajes independientes del contexto y crear o recorrer un autómeta de pila. Para mostrar un diseño agradable al usuario se utilizó HTML Y Bootstrap.

```
1 from tabulate import tabulate
2 import sys
3 from tkinter import filedialog
4 import CargarArchivo as CargarArchivo
5 import Gramatica as Gramatica
6 import html_tabla as html_tabla
7 import Equivalente as Equivalente
8 from CargarArchivo import *
9 from Gramatica import *
10 import time
11 import os
12 DatosCargados="          Datos Cargados Correctamente          "
13 ruta=""
14 opcion=0
15 def Inicio():
16     table = [{"SUPER AUTOMATA"}, [{"JUAN PABLO GONZALEZ LEAL"}, [{"201901374"}]]
17     print(tabulate(table, tablefmt="pretty"))
18     conteo(6)
19
20
21 def conteo(valor):
22     for i in range(valor):
23         time.sleep(1)
24         valor -= 1
25         print(valor,end=" ")
26         if valor==0:
27             print("")
28             menu()
29
30 def menu():
31     global ruta
32     global opcion
33     while True:
34         print('1. Cargar archivo')
35         print('2. Mostrar informacion de la gramatica')
36         print('3. Generar automata de pila equivalente')
37         print('4. reporte de recorrido')
38         print('5. Reporte en tabla')
39         print('6. Salir')
40         valor=int(input('ELIGE UNA OPCION'))
41         if valor == 1:
42             eliminar()
43             ruta=filedialog.askopenfilename(title="abrir")
44             archivo=open(ruta)
45             for linea in archivo.readlines():
46                 CargarArchivo.analizar(linea)
47             archivo.close()
48             print(DatosCargados)
49             menu()
50         elif valor == 2:
51             tamaño=len(Titulos)
52             for i in range(0,tamaño):
53                 print(str(Cantidad1[i])+" Nombre:"+Titulos[i])
54
55             opcion=int(input('ELIJA UNA OPCION'))
56             Gramatica.analizar(opcion)
57             esperar()
58         elif valor == 3:
59             Equivalente.crear(opcion)
60             menu()
61         elif valor == 4:
62             print('posicion 4')
63             menu()
64         elif valor == 5:
65             html_tabla.crear(opcion)
66             menu()
67         elif valor == 6:
68             break
69
70 def esperar():
71     print('1. Desea Mostnar Menu')
72     print('2. Desea Salir del Progrma')
73     opcion=int(input('ELIJA UNA OPCION'))
74     if opcion==1:
75         for i in range(0,100):
76             print("")
77             menu()
78     else:
79         sys.exit()
```

### Archivo:

main.py

### Librerías:

Tkinter

Os

time

### Métodos utilizados:

1. Inicio()
  2. conteo(valor)
  3. menu()
  4. esperar()
  5. eliminar()
1. Se inicia el programa mostrando mensajes y pasando por parámetro el 6 al método valor.
  2. Al recibir el número se iniciará un conteo regresivo de 5 a 0, para mostrar el menú.
  3. Se muestran las diferentes opciones que para entrar a diferentes archivos y realizar tareas específicas.
  4. Este método muestra un submenú, si se desea regresar al menú principal o salir del software
  5. Este método elimina todo contenido de todas las listas dejándolas en blanco para poder ingresar una nueva ruta de archivo

```

81 def eliminar():
82     global ruta
83     global opcion
84     opcion=0
85     g_nombre=""
86     g_Noterminales=""
87     g_Terminales=""
88     g_TerminalInicial=""
89     Validar=False
90     ContadorCantidad=0
91     contador=0
92     ruta=""
93     Terminales.clear()
94     Noterminales.clear()
95     Terminalinicial.clear()
96     Titulos.clear()
97     Produccion.clear()
98     Cantidad1.clear()
99     Cantidad2.clear()
100    Cantidad3.clear()
101    Iteracion.clear()
102    Pila.clear()
103    Entrada.clear()
104    Transiciones.clear()
105    TransicionesGrafico.clear()
106
107 Inicio()

```

#### Archivo:

CargarArchivo.py

#### Librerías:

tkinter

#### Métodos:

1. Analizar(cadena)
2. Nombre()
3. Automatas()
4. Producciones()
5. Eliminar()

1. Este analiza la cadena con condiciones, dependiendo de esas condiciones enviara la cadena leída a diferente método.
2. Toma el nombre de la gramática y lo guarda en una lista, en otra vamos guardando que numero de título es.
3. El método autómatas separa la cadena leída a través de “;” e ir guardándolos en listas por separado de igual manera se guarda en una lista que numero de autómatas o alfabeto es.
4. Este método se encarga de ir analizando las producciones e ir guardándolas de igual manera se crear un lista a que numero de titulo pertenecen, e ir

```

1 from tkinter import *
2 from tkinter import messagebox as MessageBox
3 noterminales=""
4 terminales=""
5 terminalinicial=""
6 contador=0
7 Terminales=[]
8 Noterminales=[]
9 Terminalinicial=[]
10 Titulos=[]
11 Produccion=[]
12 Cantidad1=[]
13 Cantidad2=[]
14 Cantidad3=[]
15 Validos=[]
16 NoValidos=[]
17 ContadorCantidad=0
18 Validar=False
19 def analizar(cadena):
20     global contador
21     global ContadorCantidad
22     global Validar
23
24     inicio=cadena[0]
25     final=cadena[-1]
26     if inicio=="G":
27         ContadorCantidad += 1
28         Nombre(cadena)
29
30     paso2=";" in cadena
31     if paso2==True:
32         Automatas(cadena)
33
34     paso3=">" in cadena
35     if paso3==True:
36         Producciones(cadena)
37
38     paso4="*" in cadena
39     if paso4==True:
40         if Validar==True:
41             Validos.append(ContadorCantidad)
42         else:
43             Eliminar()
44             NoValidos.append(ContadorCantidad)
45     contador=0
46     Validar=False

```

validando si son gramáticas libres del contexto

5. La validación de producciones si es Falsa esta se eliminara toda gramática basándonos en el numero que se le fue asignado a que producción pertenecían

```
49 def Nombre(cadena):
50     global ContadorCantidad
51     Titulos.append(cadena)
52     Cantidad1.append(ContadorCantidad)
53
54 def Automatas(cadena):
55     global ContadorCantidad
56     global terminales
57     global notterminales
58     global terminalinicial
59     Alfabeto=cadena.split(";")
60     notterminales=Alfabeto[0]
61     terminales=Alfabeto[1]
62     terminalinicial=Alfabeto[2]
63     Terminales.append(terminales)
64     Notterminales.append(notterminales)
65     Terminalinicial.append(terminalinicial)
66     Cantidad2.append(ContadorCantidad)
67
68 def Producciones(cadena):
69     global ContadorCantidad
70     global contador
71     global terminales
72     global notterminales
73     global terminalinicial
74     global Validar
75     contadorterminales=0
76     contadornotterminales=0
77     paso1=True
78     paso2=False
79     Paso3=False
80
81     for i in cadena:
82         caracter=i
83         if paso1==True:
84             if caracter==">":
85                 paso2=True
86                 paso1=False
87             else:
88                 pass
89         elif paso2==True:
90             ter=caracter in terminales
91             noter=caracter in notterminales
92             if ter==True:
93                 contadorterminales += 1
94             elif noter==True:
95                 contadornotterminales += 1
96         if contadorterminales==2 and contadornotterminales==1:
97             Validar=True
98
99     Cantidad3.append(ContadorCantidad)
100     contador += 1
101     Produccion.append(str(contador)+": "+cadena)
102
103 def Eliminar():
104     global ContadorCantidad
105     ultimo=Titulos[-1]
106     MessageBox.showwarning("Alerta","La gramatica "+ultimo)
107     tamaño1=Cantidad1.count(ContadorCantidad)
108     for i in range(0,tamaño1):
109         Titulos.pop()
110         Cantidad1.pop()
111
112     tamaño2=Cantidad2.count(ContadorCantidad)
113     for i in range(0,tamaño2):
114         Notterminales.pop()
115         Terminalinicial.pop()
116         Terminales.pop()
117         Cantidad2.pop()
118
119     tamaño3=Cantidad3.count(ContadorCantidad)
120     for i in range(0,tamaño3):
121         Produccion.pop()
122         Cantidad3.pop()
123
124
```

```

1 from graphviz import Digraph
2 from graphviz import Graph
3 from CargarArchivo import *
4 from Gramatica import *
5 import os
6
7 def crear(opcion):
8     cadena=""
9     os.environ["PATH"] += os.pathsep + 'C:/Program Files (x86)/Graphviz2.38/bin
10     g = Digraph('finite_state_machine',format='png')
11     g.attr(rankdir='LR', size='8,5')
12     g.node('i', shape='circle', width='1')
13     g.node('p', shape='circle', width='1')
14     g.node('q', shape='circle', width='2')
15     g.node('f', shape='doublecircle', width='1')
16     g.edge('i','p', label='λ,λ;#', width='1')
17     tamaño=len(Cantidad1)
18     for i in range(0,tamaño):
19         igual=opcion==Cantidad1[i]
20         if igual==True:
21             TerInicial="λ,λ;#Terminales[i]"
22             Titulo=Titulos[i]
23             g.edge('p','q', label=TerInicial, width='1')
24
25     for i in TransicionesGrafico:
26         cadena=cadena+i+"\n"
27
28     g.edge('q','q',label=cadena, width='1')
29
30     g.edge('q','f', label='λ,λ;#', width='1')
31
32     g.render('Graficos/AP_'+Titulo.replace("\n",""),view=False)
33     crearhtml(opcion)
34
35 def crearhtml(opcion):
36     tamaño=len(Cantidad1)
37     for i in range(0,tamaño):
38         igual=opcion==Cantidad1[i]
39         if igual==True:
40             Titulo=Titulos[i]
41             terminales=Terminales[i]
42             alfabeto=Terminales[i]+" "+Noterminales[i]+"#"
43     crear=open("Graficos/AP_"+Titulo.replace("\n","").html,"w")
44     crear.write("<html lang='es'><head><meta charset='UTF-8'><meta
45     crear.write("<h1>Nombre:AP_"+Titulo+"</h1>")
46     crear.write("<h4>Terminales={terminales}</h4>")
47     crear.write("<h4>Alfabeto de pila={alfabeto}</h4>")
48     crear.write("<h4>Estados={ i,p,q,f}</h4>")
49     crear.write("")
50     crear.write("</div></div></div></div></body></html>")
51     crear.write("<h4>Estado inicial={i}</h4>")
52     crear.write("<h4>Estado de aceptacion={f}</h4>")
53     crear.close()
54     os.system("Graficos/AP_"+Titulo.replace("\n","").html")

```

## Archivo:

Gramtica.py

## Librerías:

Os

## Métodos:

1. analizar(opcion)
2. AutomataPila(opcion)
3. Conversión(palabra1,siguiente,a,contador, x1,x2,x3,x4,x5,cajon)
4. Imprimir(opcion)

1. Este método se encarga de analizar la opción que ingreso el usuario enviando por dos métodos mas
2. El AutomataPila de pila es el encargado de crear y procesar las producciones de la gramática que se eligió ingresando

## Archivo:

Equivalente.py

## Librerías:

graphviz

os

## Métodos:

1. Crear(opcion)
  2. Crearhtml(opcion)
1. Este crea el grafico a partir de la opción que eligió el usuario, validando varios if y comparando para mostrar los resultados correctos.
  2. En este se crea el HTML abriendo la imagen que se creo y se guardó en el método Crear(opción)

```

1 import os
2 from CargarArchivo import *
3 g_nombre=""
4 g_Terminales=""
5 g_Noterminales=""
6 g_TerminalesInicial=""
7 Iteracion=[]
8 Pila=[]
9 Entrada=[]
10 Transiciones=[]
11 TransicionesGrafico=[]
12 def analizar(opcion):
13     imprimir(opcion)
14     AutomataPila(opcion)
15
16
17 def AutomataPila(opcion):
18     global g_Terminales
19     global g_Noterminales
20     global g_TerminalesInicial
21     siguiente=""
22     cajon=""
23     y2=""
24     y3=""
25     x1=""
26     x2=""
27     x3=""
28     x4=""
29     x5=""
30     a=""
31     palabra1=""
32     palabra2=""
33     palabra3=""
34     contadorIter=0
35     Iteracion.append(contadorIter)
36     Pila.append("")
37     Entrada.append(a)
38     Transiciones.append(("x1","x2","x3","x4","x5"))
39     x1="p"
40     x4="q"
41     contadorIter += 1
42     Iteracion.append(contadorIter)
43     Pila.append(x5)
44     Entrada.append(a)
45     Transiciones.append(("x1","x2","x3","x4","x5"))
46     tamaño2=len(Cantidad3)
47     for i in range(0,tamaño2):
48         igual=opcion==Cantidad3[i]

```

todas estas a una lista llamada Transiciones y otra TransicionesGrafico ya que uno se encarga de guardar todo lo necesario para mostrar en el HTML de tabla y el otro para mostrar el grafico de equivalente.

3. Este método es parte de AutomataPila ya que envía a realizar una tarea específica cuando existen más de 2 terminales y 1 no terminal.
4. Esta imprime la producción que se eligió.

```

49 if igual==True:
50     paso1=True
51     paso2=False
52     paso3=False
53     paso4=False
54     contadorterminales=0
55     contadornotterminales=0
56     Texto=Produccion[i]
57     for k in Texto:
58         caracter=k
59         if paso1==True:
60             if caracter=="(":
61                 paso1=False
62                 paso2=True
63             else:
64                 pass
65         elif paso2==True:
66             if caracter==">":
67                 x4="q"
68                 x1="q"
69                 x3=palabra1
70                 y3=palabra1
71                 paso2=False
72                 paso3=True
73             elif caracter=="-":
74                 pass
75             else:
76                 if caracter==" ":
77                     pass
78                 else:
79                     palabra1=palabra1+caracter
80         elif paso3==True:
81             ter=caracter in g_Terminales
82             noter=caracter in g_Noterminales
83             if ter==True:
84                 palabra2=palabra2+caracter
85                 contadorterminales += 1
86                 if contadorterminales==2:
87                     siguiente=caracter
88             elif noter==True:
89                 palabra2=palabra2+caracter
90                 contadornotterminales += 1
91
92         if contadorterminales==2 and contadornotterminales==1:
93             ultimo palabra2
94             contadorIter += 1
95             Iteracion.append(contadorIter)
96             Pila.append(palabra1-cajon:x5)
97             Entrada.append(a)
98             Transiciones.append("("x1","x2","x3","x4","palabra2)")
99             TransicionesGrafico.append(y2","y3","palabra2)
100             conversio(palabra2,siguiente,a,contadorIter,x1,x2,x3,x4,x5,cajon)
101             cajon=siguiente-cajon
102             palabra1=""
103             palabra2=""
104
105         if contadorterminales==1 and contadornotterminales==0:
106             contadorIter += 1
107             if siguiente != None:
108                 ultimo palabra2
109                 Iteracion.append(contadorIter)
110                 Pila.append(palabra1-cajon:x5)
111                 Entrada.append(a)
112                 Transiciones.append("("x1","x2","x3","x4","palabra2)")
113                 TransicionesGrafico.append(y2","y3","palabra2)
114                 palabra1=""
115                 palabra2=""
116             else:
117                 Iteracion.append(contadorIter)
118                 Pila.append(palabra1-cajon:x5)
119                 Entrada.append(a)
120                 Transiciones.append("("x1","x2","x3","x4","palabra2)")
121                 TransicionesGrafico.append(y2","y3","palabra2)
122                 ultimo palabra2
123                 palabra1=""
124                 palabra2=""
125
126         if contadorterminales==0 and contadornotterminales==1:
127             contadorIter += 1
128             if siguiente != None:
129                 ultimo palabra2
130                 Iteracion.append(contadorIter)
131                 Pila.append(palabra1-cajon:x5)
132                 Entrada.append(a)
133                 Transiciones.append("("x1","x2","x3","x4","palabra2)")
134                 TransicionesGrafico.append(y2","y3","palabra2)
135                 palabra1=""
136                 palabra2=""
137             else:
138                 Iteracion.append(contadorIter)
139                 Pila.append(palabra1-cajon:x5)
140                 Entrada.append(a)
141                 Transiciones.append("("x1","x2","x3","x4","palabra2)")
142                 TransicionesGrafico.append(y2","y3","palabra2)
143                 ultimo palabra2
144                 palabra1=""
145                 palabra2=""
146
147         if contadorterminales==1 and contadornotterminales==1:
148             TransicionesGrafico.append(y2","y3","palabra2)
149             palabra1=""
150             palabra2=""
151
152     cajon=ultimo-cajon
153     if cajon != None:
154         for i in cajon:
155             contadorIter += 1
156             Iteracion.append(contadorIter)
157             Pila.append(cajon:x5)
158             Entrada.append(a)
159             Transiciones.append("("q","i","i","q","x2")")
160             TransicionesGrafico.append(i","i","y2)
161             cajon=cajon.replace(i,"")
162     contadorIter += 1
163     Iteracion.append(contadorIter)
164     Pila.append("q")
165     Entrada.append("lambda")
166     Transiciones.append("("q,lambda,,#f,lambda)")
167     contadorIter += 1
168     Iteracion.append(contadorIter)
169     Pila.append("lambda")
170     Entrada.append("lambda")
171     Transiciones.append("("")

```

	<pre> 173 174 175 def imprimir(opcion): 176     global g_Terminales 177     global g_Noterminales 178     global g_TerminalInicial 179     os.system("cls") 180     for i in range(0,100): 181         print("") 182     tamaño=len(Cantidad1) 183     for i in range(0,tamaño): 184         igual=opcion==Cantidad1[i] 185         if igual==True: 186             print('Nombre de la gramática: '+Titulos[i]) 187             print('No terminales: '+Noterminales[i]) 188             print('Terminales: '+Terminales[i]) 189             print('Terminal Inicial: '+TerminalInicial[i]) 190             g_nombre=Titulos[i] 191             g_Noterminales=Noterminales[i] 192             g_Terminales=Terminales[i] 193             g_TerminalInicial=TerminalInicial[i] 194     tamaño2=len(Cantidad3) 195     for i in range(0,tamaño2): 196         igual=opcion==Cantidad3[i] 197         if igual==True: 198             print('Produccion: '+Produccion[i]) 199 200 def conversion(palabra1,siguiente,a,contadorIter,x1,x2,x3,x4,x5,cajon): 201     contadorIter=contadorIter+1 202     iteracion.append(contadorIter) 203     Pila.append(palabra1-cajon*x5) 204     Entrada.append(a) 205     Transiciones.append(("&lt;"+x1+"&gt;"+siguiente+"&gt;"+siguiente+"&gt;"+x4+"&gt;"+x2+"&gt;")) 206 </pre>
<pre> 1 from Gramatica import * 2 from CargarArchivo import * 3 import os 4 def crear(opcion): 5     tamaño=len(Cantidad1) 6     for i in range(0,tamaño): 7         igual=opcion==Cantidad1[i] 8         if igual==True: 9             Titulo=Titulos[i] 10     crear=open("AP_"+Titulo.replace("\n","").+".html","w") 11     crear.write("&lt;html lang='es'&gt;") 12     crear.write("&lt;head&gt;&lt;meta charset='ISO-8859-1'&gt;&lt;link href='https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css' rel='stylesheet'&gt;&lt;/head&gt;&lt;div class='container'&gt;&lt;div class='row justify-content-center'&gt;&lt;div class='col-md-12'&gt;&lt;table class='table'&gt;&lt;thead&gt;&lt;tr class='table-dark'&gt;&lt;th scope='col'&gt;No/&lt;th scope='col'&gt;Transiciones&lt;/thead&gt;&lt;tbody&gt;&lt;tr&gt;&lt;td&gt;"+str(i)+"&lt;/td&gt;&lt;td&gt;"+Pila[i]+"&lt;/td&gt;&lt;td&gt;"+Entrada[i]+"&lt;/td&gt;&lt;/tr&gt;&lt;/tbody&gt;&lt;/table&gt;&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;&lt;/html&gt;") 13     crear.close() 14     os.system("AP_"+Titulo.replace("\n","").+".html") 15 </pre>	<p><b>Archivo:</b>          Html_tabla.py</p> <p><b>Librerías:</b>          OS</p> <p><b>Métodos:</b></p> <ol style="list-style-type: none"> <li>1. crear(opcion)</li> <li>1. El método crear(opcion) recibe el numero que indico el usuario al desear ver esa gramática</li> </ol>