

# **MANUAL TECNICO**

Vinicio Manfredo López Pérez, Universidad de San Carlos de Guatemala

## PARADIGMA APLICADO DE PROGRAMACIÓN

Programación modular y Programación orientada a objetos.

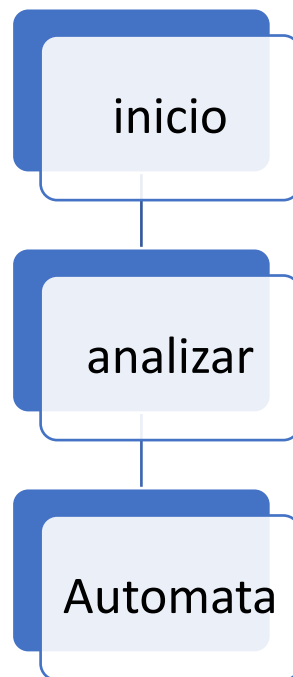
## CONVENCIONES DE NOMENCLATURA

### Método Limpiar

Este método limpia el text cada vez que se abre un nuevo documento ya que si no se limpia muestra de nuevo los valores al abrirlo.

```
def clearTextInput():  
    my_text.delete("1.0", "end")
```

## DIAGRAMA DE CLASES

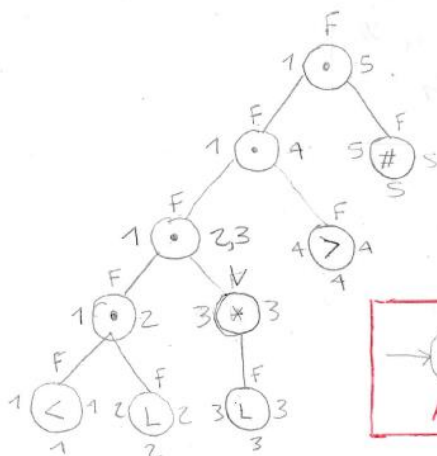


## TABLA DE TOKENS Y PATRONES (EXPRESION REGULAR)

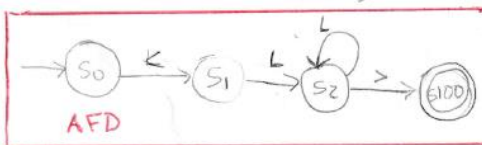
Token	Patrón (E.R.)
Etiqueta 1	$\langle L^+ \rangle$
Etiqueta 2	$\langle L^+ = L^+ \rangle$
Etiqueta 3	$\langle L^+ \rangle D^+ (. D^+ ) ? \langle L^+ \rangle$
Etiqueta 4	$\langle / L^+ \rangle$
Etiqueta 5	$\langle L^+ \rangle L^+ \langle L^+ \rangle$
Etiqueta 6	$\langle L^+ \rangle [ L^+ ] \langle L^+ \rangle$
Etiqueta 7	$\langle L^+ = L^+ = D^+ ( /   \mathcal{E} ) \rangle$
Etiqueta 8	$L^+ ( .   ( , L^+ ( .   \mathcal{E} ) ) ^* )$

## Proyecto 1 - Automatas Finitos deterministas

1)  $\langle L^+ \rangle$  para  $\langle \text{tipo} \rangle \rightarrow \langle \text{texto} \rangle \rightarrow \langle \text{estilo} \rangle$

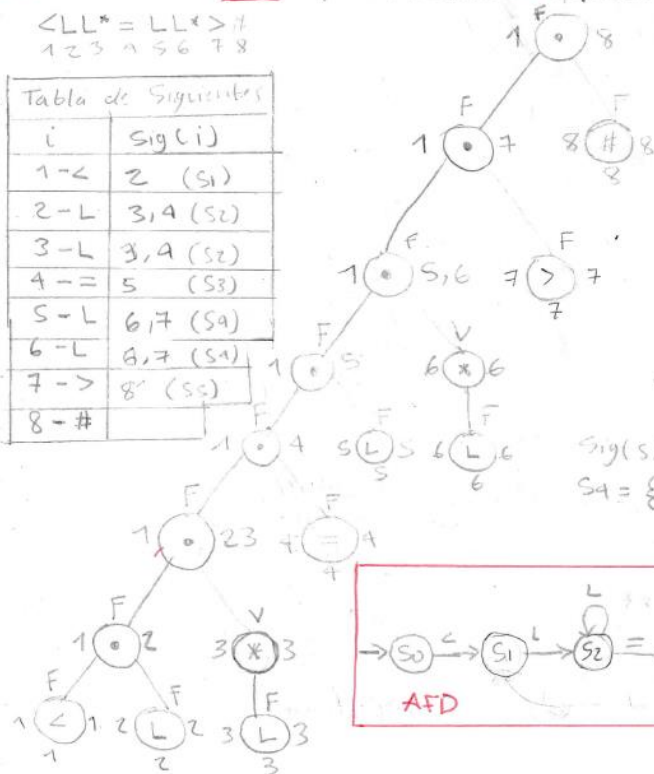
$$\begin{matrix} <LL^* > \# \\ 123 & 45 \end{matrix}$$


i	Sig (i)
1 - <	2 {S1}
2 - L	3, 4 {S2}
3 - L	3, 4 {S2}
4 - >	5 {S3}
5 - #	

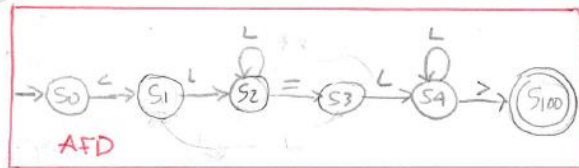
$$S_0 = \{1, \frac{1}{2}\}$$
$$\text{Sig}(4) = \text{Sig}(1) = \{2\} = S_1$$
$$S_1 = \{2\}$$
$$\text{Sig}(L) = \text{sig}(z) = \{3, 4\} = S_2$$
$$S_2 = \{3, 4\}$$
$$\text{Sig}(L) = \text{Sig}(3) = \{3, 4\} = S_2$$
$$\text{sig}(+) = \text{sig}(-) = \{5\} = 5$$


2)  $\langle L^* = L^* \rangle$  Par  $\langle \text{operation} = \text{division} \rangle \rightarrow \langle \text{funcion} = \text{describir} \rangle$   
 $\langle \text{operation} = \text{suma} \rangle \rightarrow \langle \text{operation} = \text{resta} \rangle \rightarrow \langle \text{operation} = \text{multiplicacion} \rangle$   
 $\langle LL^* = LL^* \rangle$   
 1 2 3 4 5 6 7 8

Tabla de Sigüientes	
i	sig(i)
1 - <	2 (S1)
2 - L	3, 4 (S2)
3 - L	3, 4 (S2)
4 - =	5 (S3)
5 - L	6, 7 (S4)
6 - L	6, 7 (S4)
7 - >	8 (S5)
8 - #	



$S_0 = \{1\}$   
 $\text{sig}(<) = \text{sig}(1) = \{2\} = S_1$   
 $S_1 = \{2\}$   
 $\text{sig}(L) = \text{sig}(2) = \{3, 4\} = S_2$   
 $S_2 = \{3, 4\}$   
 $L =$   
 $\text{sig}(L) = \text{sig}(3) = \{3, 4\} = S_2$   
 $\text{sig}(=) = \text{sig}(4) = \{5\} = S_3$   
 $S_3 = \{5\}$   
 $L =$   
 $\text{sig}(L) = \text{sig}(5) = \{6, 7\} = S_4$   
 $S_4 = \{6, 7\}$   
 $L =$   
 $\text{sig}(L) = \text{sig}(6) = \{6, 7\} = S_4$   
 $\text{sig}(>) = \text{sig}(7) = \{8\} = S_5$

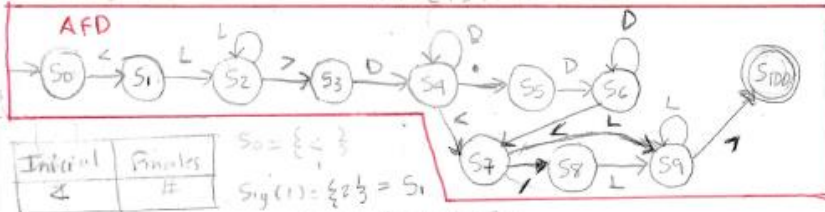


AFD

3)  $\langle L^+ \rangle D^+ (\cdot D^+)? \langle /L^+ \rangle$  para  $\langle \text{numero} \rangle 4.5 \langle / \text{numero} \rangle$

$\langle LL^* \rangle DD^* (\cdot DD^* | \epsilon) \langle (/ \epsilon) LL^* \rangle \#$

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15



Initial	Final
$\epsilon$	$\#$

$S_0 = \{ \epsilon \}$

$Sig(1) = \{ 2 \} = S_1$

$Sig(2) = \{ 3, 4 \} = S_2$

$Sig(5) = \{ 6, 7, 10 \} = S_4$

$Sig(3) = \{ 3, 4 \} = S_2$

$Sig(6) = \{ 6, 7, 10 \} = S_4$

$Sig(4) = \{ 5 \} = S_3$

$Sig(7) = \{ 8 \} = S_5$

$Sig(8) = \{ 9, 10 \} = S_6$

$Sig(10) = \{ 11 \} = S_7$

$Sig(9) = \{ 9, 10 \} = S_6$

$Sig(11) = S_8$

$Sig(10) = \{ 11, 12 \} = S_7$

$Sig(12) = S_9$

$Sig(11) = \{ 12, 13 \} = S_8$

$Sig(12) = \{ 13, 14 \} = S_9$

$Sig(13) = \{ 15 \} = S_{10}$

$Sig(14) = \{ 15 \} = S_{10}$

$Sig(15) = \{ 15 \} = S_{10}$

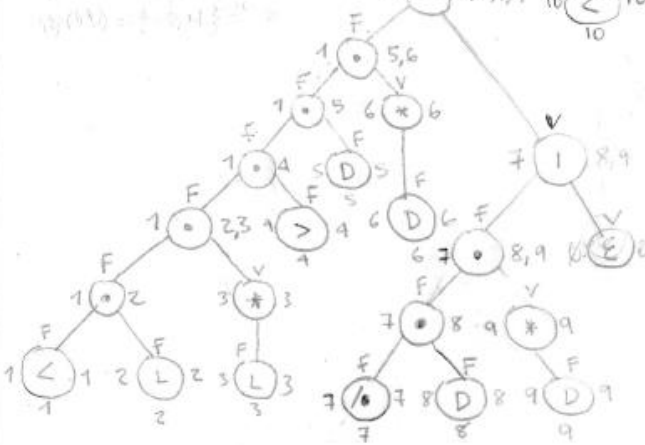
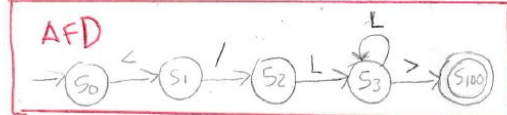


Tabla de Sigüientes

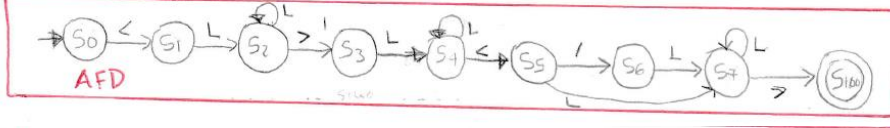
i	Sig(i)
1 - <	2 (S1)
2 - L	3, 4 (S2)
3 - L	3, 4 (S2)
4 - >	5 (S3)
5 - D	6, 7, 10 (S4)
6 - D	6, 7, 10 (S4)
7 - .	8 (S5)
8 - D	9, 10 (S6)
9 - D	9, 10 (S6)
10 - <	11, 12 (S7)
11 - /	12, 12 (S8)
12 - L	13, 14, 14 (S9)
13 - L	13, 14 (S9)
14 - >	15 (S10)
15 - #	

4)  $\langle /L^+ \rangle$  para  $\langle / \text{operacion} \rangle \rightarrow \langle / \text{tipo} \rangle \rightarrow \langle / \text{texto} \rangle \rightarrow \langle / \text{funcion} \rangle \rightarrow \langle / \text{estilo} \rangle$

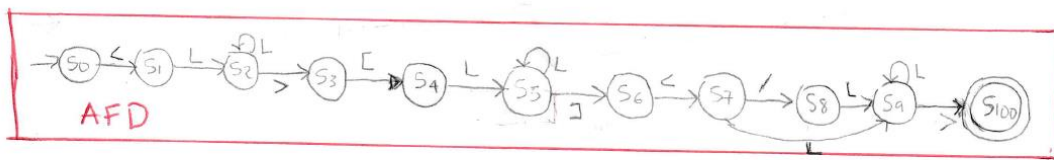
$\langle /LL^* \rangle \#$



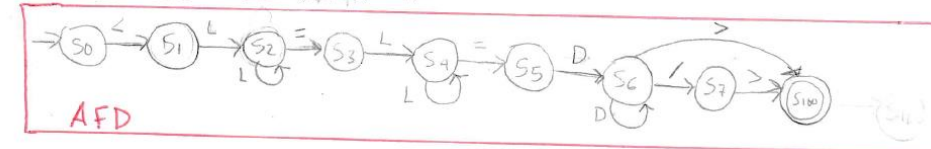
5)  $\langle L^+ \rangle L^+ \langle /L^+ \rangle$  para  $\langle \text{titulo} \rangle$  operaciones  $\langle / \text{titulo} \rangle$   
 $\langle LL^* \rangle LL^* \langle /LL^* \rangle \#$



6)  $\langle L^+ \rangle [L^+] \langle /L^+ \rangle$  para  $\langle \text{descripcion} \rangle [\text{texto}] \langle / \text{descripcion} \rangle$   
 $\langle LL^* \rangle [LL^*] \langle /LL^* \rangle \#$   $\rightarrow \langle \text{contenido} \rangle [\text{tipo}] \langle / \text{contenido} \rangle$



7)  $\langle L^+ = L^+ = D^+ (/IE) \rangle$   
 $\langle LL^* = LL^* = DD^* (/IE) \rangle \#$



8)  $L^+ ( ( \circ | ( \circ L^+ ( \circ IE ) )^* | IE )$  para  $\text{paréntesis de suma}$ ...

$LL^* ( ( \circ | ( \circ LL^* ( \circ IE ) )^* | IE ) \#$   
 $12 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8$   
 $S_0 = \{1\}$   
 $Sig(1) = \{2, 3, 4\} = S_1$   
 $S_1 = \{2, 3, 4\}$   
 $Sig(2) = S_1$   
 $Sig(3) = \{8\} = S_2$   
 $Sig(4) = \{5\} = S_3$   
 $S_3 = \{5\}$   
 $Sig(5) = \{6, 7, 8\} = S_4$   
 $Sig(6) = S_4$   
 $Sig(7) = S_2$

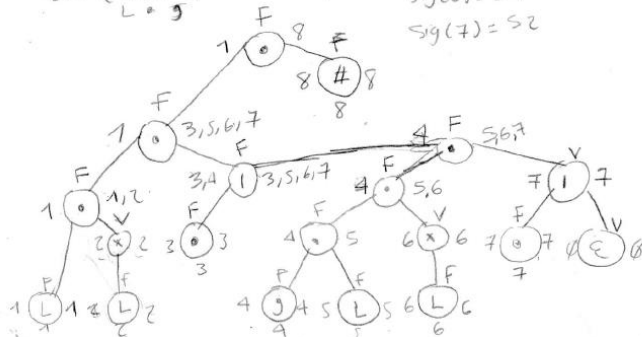
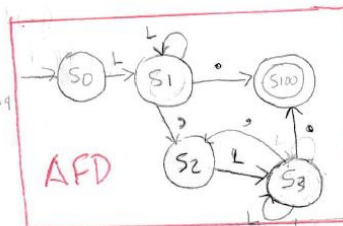


Tabla de Siguenza	
i	Sig(i)
1-L	2, 3, 4
2-L	2, 3, 4
3-L	8
4-L	S1
5-L	6, 7, 8
6-L	6, 7, 8
7-L	8
8-L	#



```

        self.errors.append(i)
        return False
    elif self.transicion=="/":
        self.estado=2
    elif self.transicion==".":
        self.estado=100
    elif self.transicion==",":
        self.estado=3
    else:
        self.errors.append(self.transicion)
        self.errors.append(i)
        return False
##### ESTADO 2 #####
elif self.estado==2:
    if str.isalpha(self.transicion):
        if str.isalpha(self.cadena[1]):
            if str.isalpha(self.cadena[0]):
                self.estado=3
            elif self.cadena[0]=="<":
                self.estado=2
            else:
                self.errors.append(self.transicion)
                self.errors.append(i)
                return False
        elif self.cadena[1]=="/":
            self.estado=3
        else:
            self.errors.append(self.transicion)
            self.errors.append(i)
            return False
    elif self.transicion==">":
        if len(self.cadena) in range(6,9):
            self.estado=100
        elif len(self.cadena)>8:
            self.estado=3
        else:
            self.errors.append(self.transicion)
            self.errors.append(i)
            return False
    elif self.transicion=="=":
        self.estado=3
    else:
        self.errors.append(self.transicion)
        self.errors.append(i)
        return False

```



```

##### ESTADO 3 #####
elif self.estado==3:
    if str.isalpha(self.transicion):
        if "<titulo>" in self.cadena or "color" in
self.cadena:
            self.estado=4
        else:
            self.estado=3
    elif self.transicion==">":
        self.estado=100
        if "operacion" in self.cadena:
            self.operaciones.append(self.cadena)
    elif str.isdigit(self.transicion):
        self.estado=4
    elif self.transicion=="." :
        self.estado=100
    elif self.transicion=="," :
        self.estado=2
    elif self.transicion=="[" :
        self.estado=4
    else:
        self.errors.append(self.transicion)
        self.errors.append(i)
        return False
##### ESTADO 4 #####
elif self.estado==4:
    if str.isdigit(self.transicion):
        self.estado=4
    elif str.isalpha(self.transicion):
        if "<descripcion>" in self.cadena or "<contenido>"
in self.cadena:
            self.estado=5
        else:
            self.estado=4
    elif self.transicion=="." :
        self.estado=5
    elif self.transicion=="=" :
        self.estado=5
    elif self.transicion=="<" :
        self.estado=7
    else:
        self.errors.append(self.transicion)
        self.errors.append(i)
        return False
##### ESTADO 5 #####

```

```

elif self.estado==5:
    if str.isdigit(self.transicion):
        self.estado=6
    elif str.isalpha(self.transicion):
        self.estado=5
    elif self.transicion=="]":
        self.estado=6
    else:
        self.errors.append(self.transicion)
        self.errors.append(i)
        return False
##### ESTADO 6 #####
elif self.estado==6:
    if str.isdigit(self.transicion):
        self.estado=6
    elif self.transicion=="<" or self.transicion=="/":
        self.estado=7
    elif self.transicion==">":
        self.estado=100
    else:
        self.errors.append(self.transicion)
        self.errors.append(i)
        return False
##### ESTADO 7 #####
elif self.estado==7:
    if self.transicion=="/":
        self.estado=8
    elif str.isalpha(self.transicion):
        self.estado=9
    elif self.transicion==">":
        self.estado=100
    else:
        self.errors.append(self.transicion)
        self.errors.append(i)
        return False
##### ESTADO 8 #####
elif self.estado==8:
    if str.isalpha(self.transicion):
        self.estado=9
    else:
        self.errors.append(self.transicion)
        self.errors.append(i)
        return False
##### ESTADO 9 #####
elif self.estado==9:

```

```

        if str.isalpha(self.transicion):
            self.estado=9
        elif self.transicion==">":
            self.estado=100
            if "numero" in self.cadena:
                self.operaciones.append(self.cadena)
            else:
                self.errors.append(self.transicion)
                self.errors.append(i)
                return False
        else:
            self.errors.append(self.transicion)
            self.errors.append(i)
            return False
    if self.estado==100:
        self.validos.append(self.contador)

        self.estado=0
    else:
        self.errors.append(self.transicion)
        self.errors.append(i)
        return False
    global crack
    crack=self.validos
    if len(self.matriz)==self.contador:
        return True

```

## MÉTODOS PRINCIPALES

### Método open\_txt

Este método se usa cuando se presiona el botón abrir archivo, abre un archivo de texto y lo inserta en un text área, luego permite leerlo y editarlo.

```

def open_txt():
    clearTextInput()
    global text_file1
    text_file1 = filedialog.askopenfilename(title="Open Text File",
filetypes=(("Text Files", "*.txt"),("all files","*.*")))
    try:
        miArchivo = open(text_file1, "r",encoding = "ISO-8859-1")
        global lectura
        lectura = miArchivo.read()
        tkinter.messagebox.showinfo("Informacion","Archivo cargado
correctamente")

```

```

except ValueError:
    tkinter.messagebox.showerror("Error","Formato incorrecto")
    return None
except FileNotFoundError:
    tkinter.messagebox.showerror("Error","Archivo dañado \n o no
seleccionado")
    return None
name = text_file1
name = name.replace("C:/gui/", "")
name = name.replace(".txt", "")
text_file1 = open(text_file1, 'r',encoding = "ISO-8859-1")
stuff = text_file1.read()
my_text.insert(END, stuff)
text_file1.close()

my_text.place(x=340, y=0, height=150, width=360)
text_scroll1.grid(column=3, row=0, rowspan=6,sticky=N+S+W)
l1.grid(row=0,column=2, sticky=W+E, padx=5)
text_scroll2.grid(column=2, row=6,sticky=S+E+W)

```

### Método save\_txt

Este método guarda lo que hayamos editado del archivo de entrada al presionar el botón guardar.

```

def save_txt():
    try:
        text_file =text_file1.name
        text_file = open(text_file, 'w', encoding = "ISO-8859-1")
        text_file.write(my_text.get(1.0, END))
        tkinter.messagebox.showinfo("Informacion","Archivo guardado
correctamente.")
    except:
        tkinter.messagebox.showerror("Error","No se pudo guardar el
archivo")
    if my_text.place_info() != {}:
        my_text.place_forget()
        text_scroll1.grid_forget()
        text_scroll2.grid_forget()
        l1.grid_forget()

```

### Método save\_as

Este método guarda lo que se encuentra dentro del objeto text con el nombre y tipo de archivo que se desee haciendo Clic en el botón Guardar como.

```
def save_as():
    file=filedialog.asksaveasfile()
    if file is None:
        return
    filetext=str(my_text.get(1.0,END))
    file.write(filetext)
    file.close()
```

## Método info

Este método al presionar el botón ocultar, oculta todos los objetos tkinter que contienen la información del creador del programa.

```
def info():
    if l2.grid_info() != {}:
        l2.grid_forget()
        l3.grid_forget()
        l4.grid_forget()
        l5.grid_forget()
        l6.grid_forget()
        l7.grid_forget()
        l8.grid_forget()
        l9.grid_forget()
    else:
        l2.grid(row=0,column=2,sticky=W+E, columnspan=2)
        l3.grid(row=1,column=2,sticky=W,columnspan=2)
        l4.grid(row=2,column=2,sticky=W,columnspan=2)
        l5.grid(row=3,column=2,sticky=W,columnspan=2)
        l6.grid(row=4,column=2,sticky=W,columnspan=2)
        l7.grid(row=5,column=2,sticky=W,columnspan=2)
        l8.grid(row=6,column=2,sticky=W,columnspan=2)
        l9.grid(row=6,column=3,sticky=E)
```

## Método analizar

Este método instancia la clase main que contiene los métodos para analizar el programa y los muestra en un objeto text de tkinter, además de crear el archivo de errores con formato html con el primer error que se encuentre en el archivo.

```
def analizar():
    from analizar import main,errores
    main()
    errores()
    from analizar import cadenaf
    clearTextInput()
```

```

my_text.insert(END, cadenaf)
my_text.place(x=340, y=0, height=150, width=360)
text_scroll1.grid(column=3, row=0, rowspan=6, sticky=N+S+W)
l1.grid(row=0, column=2, sticky=W+E, padx=5)
text_scroll2.grid(column=2, row=6, sticky=S+E+W)
l10.grid(row=7, column=2, sticky=E)

```

## Método errr

Este metodo se acciona al presionar el botón errores en la interfaz y muestra los errores del último archivo que se creo en el método analizar.

```

def errr():
    webbrowser.open('file://' +
os.path.realpath("errores/ERRORES_202002912.html"))

```

## Método manual\_t

Este método se acciona al presionar el botón manual técnico en la interfaz y abre el manual técnico.

```

def manual():
    path = 'Manual Tecnico.pdf'
    subprocess.Popen([path], shell=True)

```

## Creación de la interfaz

Dentro del archivo de inicio se crea la interfaz que contiene todos los botones, los labels y el objeto texto de tkinter con sus respectivas posiciones dentro del frame.

```

#Creacion de interfaz
ventana=Tk()
ventana.geometry("+800+400")
aplicacion=Menu(ventana)
ventana.title("Proyecto 1")
principal=Frame(ventana)
principal.grid(row=0, column=0, columnspan=3, pady=0)
#Labels
Label(principal, borderwidth = 2, relief="groove",
text="Archivo", bg="BLACK", fg="WHITE").grid(row=0, column=0, sticky=W+E)
Label(principal, borderwidth = 2, relief="groove",
text="Ayuda", bg="BLACK", fg="WHITE").grid(row=0, column=1, sticky=W+E)
l1 = Label(principal, text="", padx=175)
l1.grid(row=0, column=2, sticky=W+E, padx=5)
l2 = Label(principal, borderwidth = 2, relief="groove", text="Informacion del
Creador", bg="BLACK", fg="WHITE")

```

```

12.grid(row=0,column=2,sticky=W+E, columnspan=2)
13 = Label(principal, text="Nombre: Vinicio Manfredo López Pérez")
13.grid(row=1,column=2,sticky=W, columnspan=2)
14 = Label(principal, text="Carné: 202002912")
14.grid(row=2,column=2, sticky=W, columnspan=2)
15 = Label(principal, text="Lugar: Guatemala")
15.grid(row=3,column=2, sticky=W,columnspan=2)
16 = Label(principal, text="Curso: Lenguajes Formales de Programación")
16.grid(row=4,column=2,sticky=W,columnspan=2)
17 = Label(principal, text="Año de Creacion: 2022")
17.grid(row=5,column=2,sticky=W,columnspan=2)
18 = Label(principal, text="Sección: A+")
18.grid(row=6,column=2,sticky=W)
#Botones
ttk.Button(principal,text="Abrir",command=open_txt).grid(row=1,column=0,stick
y=W+E)
ttk.Button(principal,text="          Guardar          ",command=save_txt).grid
(row=2,column=0,sticky=W+E)
ttk.Button(principal,text="Guardar
Como",command=save_as).grid(row=3,column=0,sticky=W+E)
ttk.Button(principal,text="Analizar",command=analizar).grid(row=4,column=0,s
ticky=W+E)
ttk.Button(principal,text="Errores",
command=errr).grid(row=5,column=0,sticky=W+E)
ttk.Button(principal,text="Salir",command=ventana.destroy).grid(row=6,column
=0,sticky=W+E)
ttk.Button(principal,text="Manual de
Usuario").grid(row=1,column=1,sticky=W+E)
ttk.Button(principal,text="          Manual
Técnico          ",command>manual_t).grid(row=2,column=1,sticky=W+E)
ttk.Button(principal,text="Temas de
Ayuda",command=info).grid(row=3,column=1,sticky=W+E)
19 = ttk.Button(principal,text="Ocultar",command=info)
19.grid(row=6,column=3,sticky=E)
110 = ttk.Button(principal,text="Ocultar",command=info1)
110.grid(row=7,column=2,sticky=E)
# Crear scrollbar
text_scroll1 = Scrollbar(principal)
text_scroll1.grid(column=3, row=0, rowspan=6,sticky=N+S+W)
text_scroll2 = Scrollbar(principal, orient=HORIZONTAL)
text_scroll2.grid(column=2, row=6,sticky=S+E+W)
my_text =
Text(principal,wrap=NONE,yscrollcommand=text_scroll1.set,xscrollcommand=text
_scroll2.set)

```

```

my_text.place(x=340, y=0, height=150, width=360)
# Configurar scrollbar
text_scroll1.config(command=my_text.yview)
text_scroll2.config(command=my_text.xview)
#Ocultar al iniciar el programa
my_text.place_forget()
text_scroll1.grid_forget()
text_scroll2.grid_forget()
l1.grid_forget()
l2.grid_forget()
l3.grid_forget()
l4.grid_forget()
l5.grid_forget()
l6.grid_forget()
l7.grid_forget()
l8.grid_forget()
l9.grid_forget()
l10.grid_forget()
#Ejecucion de la ventana
ventana.mainloop()

```

## Método gestión

Este solo se ejecuta si el archivo de entrada tiene el formato correcto. Si es así este método por medio de listas, cadenas y validaciones lee las operaciones que contiene el archivo de entrada y las realiza para luego insertarlas en un objeto text de tkinter que simula ser la consola con las respuestas de las operaciones.

```

def gestion(self):
    crack=[]
    global cadenaf
    cadenaf="Generacion de Archivo HTML\n"
    for linea in self.operaciones:
        linea=str(linea).replace("numero","")
        linea=str(linea).replace("/","")
        linea=str(linea).replace("<","")
        linea=str(linea).replace(">","")
        crack.append(linea)
    cadena=""
    try:
        for line in range(len(crack)):
            if len(crack[line])>9 and not any(chr.isdigit() for chr in
str(crack[line])):
                crack[line]=str(crack[line]).replace("="," ")
                cadena+="\n"+crack[line]+":\n"

```



```

        elif crack[line]=="operacion" and not any(chr.isdigit() for
chr in str(crack[line])):
            cadena+="\n"+crack[line)+"\n"
        else:
            if any(chr.isalpha() for chr in str(crack[line+1])):
                if crack[line+1]=="operacion":
                    cadena+=crack[line]
                else:
                    cadena+=crack[line]+","
            else:
                cadena+=crack[line]+","

    lista_cadena=cadena.split("\n")
    matriz_2 = [elemento_lista for elemento_lista in lista_cadena if
elemento_lista != ""]
    final1=[]
    count=0
except:
    cadenaf+="Error en la linea: "+self.matriz[len(crack)]
for line1 in range(len(matriz_2)):
    if any(chr.isdigit() for chr in str(matriz_2[line1])):
        if len(matriz_2[line1+1])>9:
            count+=1
    if ":" in matriz_2[line1] and ":" in matriz_2[line1+1]:
        count+=1

    if matriz_2[line1]=="operacion suma:":
        matriz_2[line1+1]=matriz_2[line1+1].replace(",","+")
    elif matriz_2[line1]=="operacion resta:":
        matriz_2[line1+1]=matriz_2[line1+1].replace(",","-")
    elif matriz_2[line1]=="operacion multiplicacion:":
        matriz_2[line1+1]=matriz_2[line1+1].replace(",","*")
    elif matriz_2[line1]=="operacion division:":
        matriz_2[line1+1]=matriz_2[line1+1].replace(",","/")
    elif matriz_2[line1]=="operacion potencia:":
        matriz_2[line1+1]=matriz_2[line1+1].replace(",","**")
    elif matriz_2[line1]=="operacion raiz:":
        matriz_2[line1+1]=matriz_2[line1+1].replace(",","**(1/(")
    elif matriz_2[line1]=="operacion mod:":
        matriz_2[line1+1]=matriz_2[line1+1].replace(",","%")

    final1.append(matriz_2[line1])
final2=[]
for line2 in range(len(final1)-count):
    if ':' in final1[line2] and ':' in final1[line2+1]:

```

```

        final1.pop(line2+1)
    if any(chr.isdigit() for chr in str(final1[line2])):
        if len(final1[line2+1])>9:
            final1.pop(line2+1)
            if any(chr.isdigit() for chr in str(final1[line2+1])):
                final1[line2+1]="("+str(final1[line2+1])+")"
        else:
            line2=line2
    final2.append(final1[line2])
print(final2)
final3=[]
for line2 in range(len(final2)):
    if len(final2[line2])>9 and not any(chr.isdigit() for chr in
str(final2[line2])):
        if any(chr.isdigit() for chr in str(final2[line2+1])):
            if any(chr.isdigit() for chr in str(final2[line2+2])):
                final2[line2]="operacion compleja:"
        final3.append(final2[line2])
matriz_3 = [elemento_lista for elemento_lista in final3 if
elemento_lista != "operacion"]
cadena1=""
for line in range(len(matriz_3)):
    if any(chr.isalpha() for chr in str(matriz_3[line])):
        cadena1+=","+str(matriz_3[line])+","
    else:
        cadena1+=str(matriz_3[line])
lista_cadena1=cadena1.split(",")
matriz_4 = [elemento_lista for elemento_lista in lista_cadena1 if
elemento_lista != ""]

cadena2=""
for line in range(len(matriz_4)):
    if "(1/" in matriz_4[line]:
        matriz_4[line]=str(matriz_4[line])+"))"
    elif "inverso" in matriz_4[line]:
        matriz_4[line+1]="("+str(matriz_4[line+1])+"))*(-1)"
    elif "seno" in matriz_4[line]:
        matriz_4[line+1]="math.sin("+str(matriz_4[line+1])+"))"
    elif "coseno" in matriz_4[line]:
        matriz_4[line+1]="math.cos("+str(matriz_4[line+1])+"))"
    elif "tangente" in matriz_4[line]:
        matriz_4[line+1]="math.tan("+str(matriz_4[line+1])+"))"
    if any(chr.isdigit() for chr in str(matriz_4[line])):
        respuesta=eval(str(matriz_4[line]))
        respuesta1=round(respuesta, 2)

```

```

        if ":" in str(matriz_4[line]):
            cadena2+=","+str(matriz_4[line])+","
        else:
            cadena2+=str(matriz_4[line])+"="+str(respuesta1)
        lista_cadena2=cadena2.split(",")
        matriz_final = [elemento_lista for elemento_lista in lista_cadena2
if elemento_lista != ""]
        print(matriz_final)
        for linea in matriz_final:
            cadenaf+=linea+"\n"
        print(cadenaf)

```

## Método errores

Este método se ejecuta si existen errores en el archivo de entrada y crea un archivo html que contiene una tabla con los errores.

```

def errores():
    filein=open('Plantilla/Plantilla.html')
    src=Template(filein.read())
    dic={"numero":1, "lexema":trac1[0],"tipo":"Error",
"columna":trac1[1],"fila":len(crack)}
    resultt=src.substitute(dic)
    try:
        os.mkdir("errores")
        filein2=open("Errores/"+"ERRORES_202002912"+"*.html","w")
        filein2.writelines(resultt)
    except OSError:
        if os.path.exists("errores"):
            filein2=open("Errores/"+"ERRORES_202002912"+"*.html","w")
            filein2.writelines(resultt)

```

## Método main

Este método se ejecuta al presionar Analizar en la interfaz grafica y lee el archivo de entrada línea por línea para pasar como parámetro a la clase autómeta la matriz, espera un false o un true si es true ejecuta el método gestión, si por el contrario es false ejecuta el método errores.

```

def main():
    matriz=[]
    text_file1 = filedialog.askopenfilename(title="Open Text File",
filetypes=(("Text Files", "*.txt"),("all files","*.*")))

```

```

try:
    miArchivo = open(text_file1, "r",encoding = "ISO-8859-1")
    global lectura
    lectura = miArchivo.read()
    tkinter.messagebox.showinfo("Informacion","Archivo cargado
correctamente")

except ValueError:
    tkinter.messagebox.showerror("Error","Formato incorrecto")
    return None
except FileNotFoundError:
    tkinter.messagebox.showerror("Error","Archivo dañado \n o no
seleccionado")
    return None

print("LECTURA LINEA POR LINEA")
texto = open(text_file1, "r",encoding = "ISO-8859-1")
lineas = texto.readlines()
for Linea in lineas:
    Linea = Linea.replace(" ", "")
    Linea = Linea.replace("\n", "")
    Linea = Linea.replace("\t", "")
    Linea = Linea.replace("Ã³", "o")
    Linea = Linea.replace("Ã¡", "a")
    Linea = Linea.replace("Ã\xad", "i")
    Linea = Linea.replace("Ã©", "e")
    Linea = Linea.replace("Ãº", "u")
    line =Linea.lower()
    matriz.append(line)
matriz_1 = [elemento_lista for elemento_lista in matriz if
elemento_lista != ""]
# Esta sección es siempre ejecutada
texto.close()
#llamada a clase automata
AFD=Automata(matriz_1)
if AFD.automata() == True:
    print("Archivo con Formato Valido\n")
    AFD.gestion()
else:
    list1=["<",">","/",".",",","="]
    global cadenaf
    cadenaf=("Archivo contiene errores. \n")+ "Error en la linea:
"+str(len(crack))+ "\n"+matriz_1[len(crack)]+ "\n"
    if trac1[0] in list1 or str(trac1[0]).isalpha():

```

```
        cadenaf+="Syntax Error -> "+trac1[0]+" | En la columna ->
"+str(trac1[1])
    else:
        cadenaf+="Error Lexico -> "+trac1[0]+" | En la columna ->
"+str(trac1[1])
```

## DESCRIPCIÓN

Tkinter: se eligió esta herramienta ya que cuenta con muchas opciones para el diseño de interfaces graficas.

Math: se elegio esta herramienta ya que es la más utilizada en Python para la realización de operaciones.

Subprocess: se eligió esta herramienta ya que es la mas idónea para abrir archivos pdf.

Webbrowser: se eligio esta herramienta ya que es la que mas resalta para abrir archivos html o incluso paginas web.

Os: se eligio esta herramienta ya que ejecuta las acciones del sistema por medio del cmd.

## IMPORTACIONES

```
import subprocess
import webbrowser
import os
from tkinter import *
from tkinter import ttk
from tkinter import filedialog
```