

MANUAL TECNICO



Kevin Estuardo Palacios
Quiñonez

LFP B+

ÍNDICE

I INTRODUCCION

II OBJETIVOS

III DIRIGIDO

IV ESPECIFICACION TÉCNICA

1. REQUISITOS DE HARDWARE
2. REQUISITOS DE SOFTWARE

V LÓGICA DEL PROGRAMA

I INTRODUCCIÓN

Este es un programa que mediante un archivo de extensión .gpw podemos analizar mediante el uso de un analizador léxico y sintáctico los posibles errores que se pueden encontrar del archivo antes de generar una página HTML.

II OBJETIVOS

El objetivo primordial de este manual es poder ayudar a distintos programadores y estudiantes de catedra de los cursos de ciencias de la computacion a informarse y guiarse de la comprensión utilizada para el desarollo de esta aplicación.-

III DIRIGIDO

Este manual esta orientado a todos los distintos programadores interesados en el desarollo de la aplicación , y a todos los auxiliares de catedra de este curso para que vean otra forma de poder llegar a la resolución de problema mediante el enunciado dado.

IV ESPECIFICACION TÉCNICA

1. REQUISITOS DE HARDWARE

- Computadora de Escritorio o Portatil.
- Mínimo 8GB de Memoria RAM.
- 20 GB disponibles de Disco duro.
- Procesador Core i3 o Superior.
- Procesador a 64 bits.
- Pantalla con Resolución Gráfica de 1024*768 píxeles.

2. REQUISITOS DE SOFTWARE

- Tener Instalado Windows 7 o superior
- Python 3.9.2
- Navegador Web para la lectura de Archivos HTML

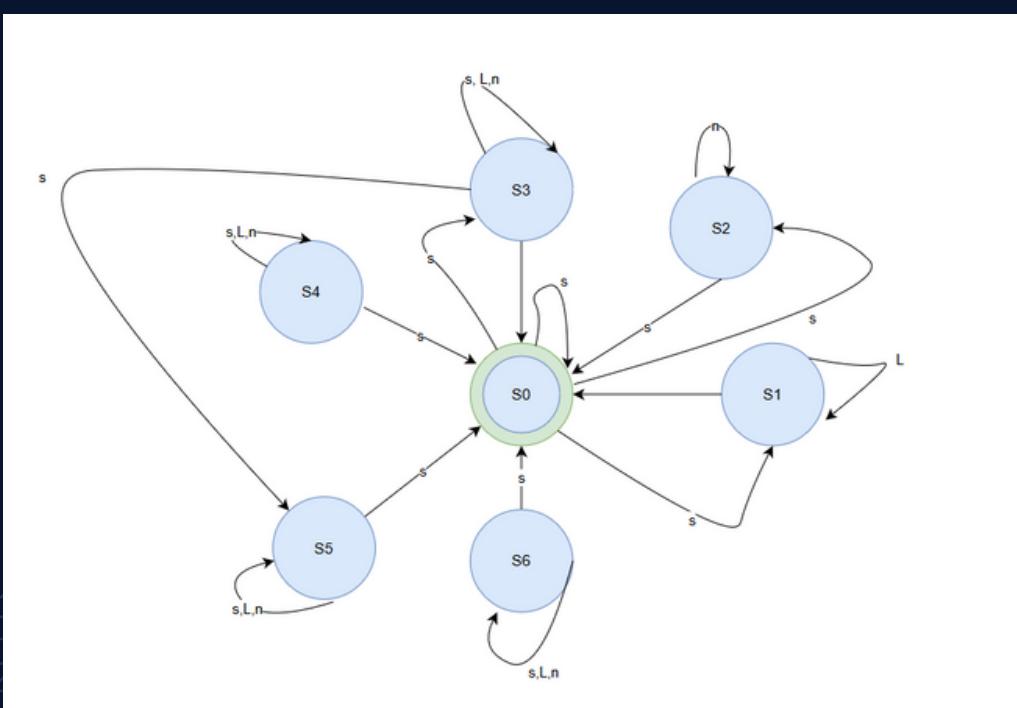
V LÓGICA DEL PROGRAMA

1. Descripción del Lenguaje .gpw

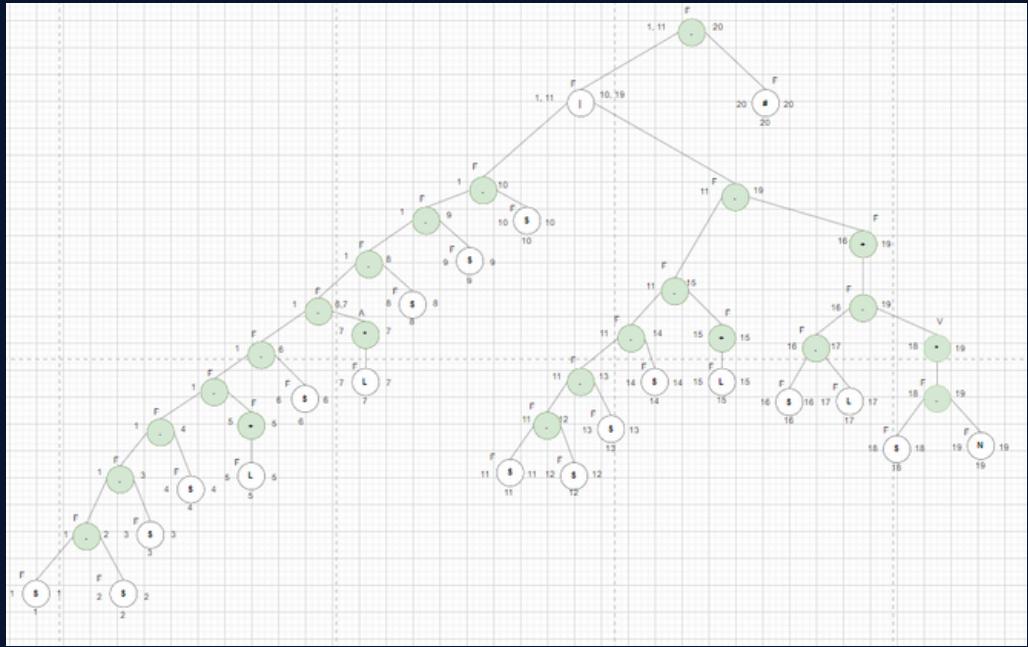
Token	Descripción	Simbología
<	T_MENORQUE	<
>	T_MAYORQUE	>
,	T_COMA	,
.	T_PUNTO	.
;	T_PUNTOCOMA	;
!	T_EXCLAMACION	!
-	T_GUION	-
(T_PARABRE	(
)	T_PARCIERRA)
Controles	T_CONTROLES	Controles
propiedades	T_PROPIEDADES	propiedades
Colocacion	T_COLOCACION	Colocacion
setColorLetra	T_SETCOLORLETRA	setColorLetra
setTexto	T_SETTEXTO	setTexto
setAlineacion	T_SETALINEACION	setAlineacion
setColorFondo	T_SETCOLORFONDO	setColorFondo
setMarcada	T_SETMARCADA	setMarcada
setGrupo	T_SETGRUPO	setGrupo

Token	Descripción	Simbología
setAncho	T_SETANCHO	setAncho
setAlto	T_SETALTO	setAlto
setPosition	T_SETPOSICION	setPosition
add	T_ADD	add
this	T_THIS	this
Centro	T_CENTRO	Centro
Izquierdo	TIZQUIERDO	Izquierdo
Derecho	T_DERECHO	Derecho
true	T_TRUE	true
false	T_FALSE	false
Etiqueta	T_ETIQUETA	Etiqueta
Boton	T_BOTON	Boton
Check	T_CHECK	Check
RadioButton	T_RADIOBUTTON	RadioButton
Texto	T_TEXTO	Texto
AreaTexto	T_AREATEXTO	AreaTexto
Clave	T_CLAVE	Clave
Contenedor	T_CONTENEDOR	Contenedor

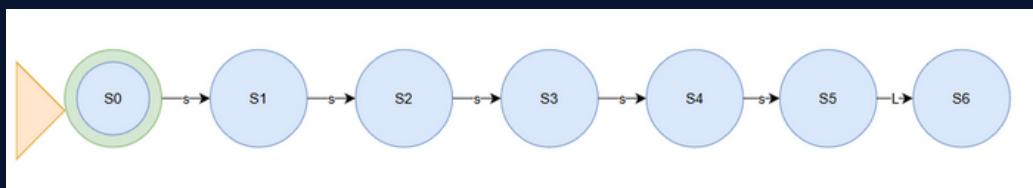
3. Autómata gráfico



2. Arbol que g  nero el automata



4. Estados de Transicion del Automata



3. Código Fuente

Método Ejecutar

Este se encuentra Ubicado en El Principal.py donde se llama al Analizador Léxico y luego añade a las tablas los errores

```
● ● ●  
def Ejecutar():  
    global hola  
    global textbox  
    global LE  
    global tv  
    hola=analizador_lexico()  
    hola.analizar(textbox.get(1.0, END))  
    LE=hola.ListaErrores  
    arreglo=[]  
    for e in LE:  
        arreglo.append((e.tipo,e.linea," ",e.lexema))  
    for ee in arreglo:  
        tv.insert("",tk.END,values=ee)  
    messagebox.showinfo("Succes",  
"Se Compilo con Éxito el programa")
```

Función Guardar

```
● ● ●  
def Guardar():  
    global RutaArchivo  
    global textbox  
    try:  
        ruta = open(RutaArchivo,"w",encoding="utf8")  
        ruta.write(textbox.get(1.0,END))  
        ruta.close()  
        messagebox.showinfo("Succes","Se Guardo el Archivo")  
    except:  
        messagebox.showwarning("Advertencia",  
"No se pudo guardar el Archivo, por favor cargue un archivo  
primero"  
)
```

Se encarga de Obtener el contenido que se encuentra en el textbox y lo guarda.

Función Guardar Como

```
● ● ●  
def GuardarComo():  
    global textbox  
    global ruta  
    try:  
        ruta = filedialog.asksaveasfile(title= "Guardar Archivo", filetypes = ((("Text files", "*.gpw*"), ("all files", "*.*")))  
        if ruta:  
            MiTexto = textbox.get(1.0,END)  
            ruta.write(MiTexto)  
            ruta.close()  
            messagebox.showinfo("Succes", "Se Guardo el Archivo")  
    except:  
        messagebox.showwarning("Advertencia", "No se pudo guardar el Archivo")
```

Se encarga de Obtener el contenido que se encuentra en el textbox y guarda un nuevo Archivo.

Nuevo

```
● ● ●

def Nuevo():
    global RutaArchivo
    global lblespacio
    if lblespacio != "":
        resultado=messagebox.askquestion("Nuevo Documento",
        , "¿Esta seguro?", icon="warning")
        if resultado=="yes":
            textbox.delete("1.0", "end")
            lblespacio.configure(text="Ruta: ")
            messagebox.showinfo("Succes",
            "Se creó un nuevo documento ")
        else:
            messagebox.showwarning("Advertencia",
            "No se creó nuevo documento")
```

Borra el contenido que se encuentra en Textbox y Ruta.

Abir

```
● ● ●

def Abrir():
    global textbox
    global material
    global RutaArchivo
    global lblespacio
    try:
        RutaArchivo = filedialog.askopenfilename(title=
"Cargar Archivo", filetypes = (("Text files", "*.gpw*"), (
"all files", "*.*")))
        ruta2 = open(RutaArchivo,"r",encoding="utf-8")
        material = ruta2.read()
        print(material)
        textbox.insert(END, str(material))
        lblespacio.configure(text="Ruta: "+RutaArchivo)
        messagebox.showinfo("Succes", "Se Cargó el Archivo")
    except:
        messagebox.showwarning("Advertencia",
        "No se pudo Cargar el Archivo")
```

Se encarga de Obtener el contenido del archivo.

VentanaPrincipal

```
● ● ●
def VentanaPrincipal():
    global ventanaprincipal
    global textbor
    global libespacio
    global libposicionx
    global libposiciony
    global lf
    global tv
    ventanaprincipal = tkinter.Tk()
    ventanaprincipal.title("Kevin")
    ventanaprincipal.geometry("1000x600")
    ventanaprincipal.configure(bg="white")
    ventanaprincipal.config(bg="white")
    ventanaprincipal.resizable(0,0)
    ventanaprincipal.iconbitmap("icono.ico")
    #ESTRUCTURA DE LA VENTANA
    tv = ttk.Treeview(ventanaprincipal,columns=("cole",
    "cole2","cole3"),height = 6)
    tv.column("#0",width=0)
    tv.column("cole",width=180, anchor=CENTER)
    tv.column("cole2",width=180, anchor=CENTER)
    tv.column("cole3",width=55, anchor=CENTER)
    tv.heading("cole", text="Tipo de Error", anchor=CENTER)
    tv.heading("cole2", text="Se esperaba", anchor=CENTER)
    tv.heading("cole3", text="Descripción de Error", anchor=CENTER)
    tv.pack()
    tv.place(x=0,y=665)
    #MENU
    menubar = Menu(ventanaprincipal)
    ventanaprincipal.config(menu=menubar)
    menubar.add_cascade(label="Archivo", command=Guardar)
    file_menu = Menubutton(menubar,tearoff=1)
    help_menu = Menubutton(menubar,tearoff=0)
    Run_menu = Menubutton(menubar,tearoff=0)
    Edit_menu = Menubutton(menubar,tearoff=0)
    View_menu = Menubutton(menubar,tearoff=0)
    Tool_menu = Menubutton(menubar,tearoff=0)
    Windows_menu = Menubutton(menubar,tearoff=0)
    #ITEMS DEL MENU
    file_menu.add_command(label="Abrir", command=abrir)
    file_menu.add_command(label="Nuevo", command=Nuevo)
    file_menu.add_command(label="Guardar", command=Guardar)
    file_menu.add_command(label="Guardar Como", command=GuardarComo)
    file_menu.add_command(label="Salir", command=ventanaprincipal.destroy)
    #ITENS DEL MENU AVAYA
    help_menu.add_command(label="Manual de Usuario", command=ManualDeUsuario)
    help_menu.add_command(label="Manual Técnico", command=ManualTecnico)
    #ITENS DEL MENU COMPIILAR
    Run_menu.add_command(label="Ejecutar", command=Ejecutar)
    #ITENS DEL MENU EDIT
    Edit_menu.add_command(label="Edición", command=Edicion)
    #ITENS DEL MENU VIEW
    View_menu.add_command(label="Vista", command=View)
    #ITENS DEL MENU TOOLS
    Tool_menu.add_command(label="Reporte de Tokens", command=Tokens)
    Tool_menu.add_command(label="Reporte de Errores", command=Erros)
    #ITENS DEL MENU WINDOWS
    Windows_menu.add_command(label="Windows", command=Windows)
    #AGREGAR ITENS A LA BARRA
    menubar.add_cascade(label="File",menu=file_menu)
    menubar.add_cascade(label="Edit",menu=Edit_menu)
    menubar.add_cascade(label="View",menu=View_menu)
    menubar.add_cascade(label="Tools",menu=Tool_menu)
    menubar.add_cascade(label="Help",menu=help_menu)
    menubar.add_cascade(label="Compiler",menu=Run_menu)
    #ETIQUETAS
    llabel1 = Label(ventanaprincipal,text="Ruta:",font=
    "Cambria 12", fg="gray97")
    libespacio.place(x=255,y=0)
    llabel1.place(x=10,y=635)
    llabel2 = Label(ventanaprincipal,text="Lineas:",font=
    "Cambria 12", fg="gray97")
    libespacio.place(x=255,y=635)
    llabel3 = Label(ventanaprincipal,text="Columna:",font=
    "Cambria 12", fg="gray97")
    libespacio.place(x=255,y=635)
    llabel4 = Label(ventanaprincipal,text="Error:",font=
    "Cambria 12", fg="gray97")
    libespacio.place(x=255,y=635)
    llabel5 = Label(ventanaprincipal,text="Status:",font=
    "Cambria 12", fg="gray97")
    libespacio.place(x=255,y=635)
    llabel6 = Label(ventanaprincipal,text="Contar:",font=
    "Cambria 12", fg="gray97")
    libespacio.place(x=255,y=635)
    #TEXTBOX
    textbor = scrolledtext.ScrolledText(ventanaprincipal,
    width=30, height=30, bg="lightgray")
    textbor.place(x=48,y=38)
    def callbackleft(event):
        libposicionx.configure(text="+" +str(event.x))
        libposiciony.configure(text="+" +str(event.y))
    def callbackright(event):
        libposicionx.configure(text="+" +str(event.x))
        libposiciony.configure(text="+" +str(event.y))
    def callbackleft(event):
        libposicionx.configure(text="+" +str(event.x))
        libposiciony.configure(text="+" +str(event.y))
    def callbackright(event):
        libposicionx.configure(text="+" +str(event.x))
        libposiciony.configure(text="+" +str(event.y))
    #BOTONES
    photo = PhotoImage(file = "C:/Users/kevin0
    neDrive/Documentos/FP-92-20390227/nuev.png")
    button.place(x=120,y=0)
    photo2 = PhotoImage(file = "C:/Users/kevin0
    neDrive/Documentos/FP-92-20390227/abrir.png")
    button2.place(x=140,y=0)
    photo3 = PhotoImage(file = "C:/Users/kevin0
    neDrive/Documentos/FP-92-20390227/save.png")
    button3.place(x=160,y=0)
    photo4 = PhotoImage(file = "C:/Users/kevin0
    neDrive/Documentos/FP-92-20390227/search2.png")
    button4.place(x=180,y=0)
    photo5 = PhotoImage(file = "C:/Users/kevin0
    neDrive/Documentos/FP-92-20390227/quit.png")
    button5.place(x=200,y=0)
    ventanaprincipal.mainloop()
```

Funcion donde se muestra Todo el contenido de la ventana.