

FORMULARIOS DINÁMICOS – PROYECTO 1

MANUAL TÉCNICO

**Allen RománPROGRAMA REALIZADO POR EL ALUMNO
ALLEN GIAN KARLO ROMÁN VÁSQUEZ CARNET 202004745
PARA EL CURSO LENGUAJES FORMALES Y DE
PROGRAMACIÓN DE LA ESCUELA DE CIENCIAS Y
SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA
UNIVERSIDAD SAN CARLOS DE GUATEMALA**

Programa Utilizado

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.



Lenguaje Utilizado

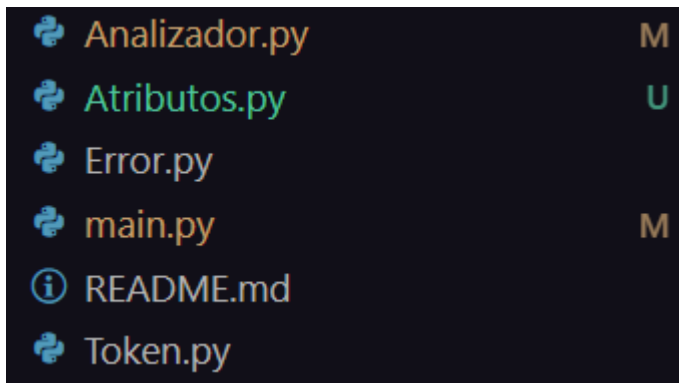
Es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo, ejemplos: Instagram, Netflix, Panda 3D, entre otros.



Código

Clases

El programa cuenta con cinco clases, una que es analizador, atributos en la que se encuentra los componentes del formulario, error que es el método constructor para la lista de errores, el main que es donde se ejecuta la aplicación y token donde está el constructor de la lista de tokens.



Interfaz gráfica

La interfaz gráfica es lo primero que se ejecuta en la aplicación, la cual sus botones, y listas tienen la siguiente forma.

```
VentanaPrincipal = Tk()
VentanaPrincipal.geometry("1200x600")
VentanaPrincipal.title("Formularios USAC")
VentanaPrincipal.config(bg="SkyBlue1")

botonCargar = Button(VentanaPrincipal, text="Cargar Archivo", width=15, height=1, font=("Arial", 12, "italic"), command= CargarArchivo)
botonAnalizar = Button(VentanaPrincipal, text="Analizar Archivo", width=15, height=1, font=("Arial", 12, "italic"), command= AnalizarArchivo)
Titulo = Label(VentanaPrincipal, text = "Generador de Formularios USAC", font=("Arial", 16, "italic"), bg="SkyBlue1")
TituloReportes = Label(VentanaPrincipal, text = "Reportes", font=("Arial", 16, "italic"), bg="SkyBlue1")
listasReportes = ttk.Combobox(VentanaPrincipal, font=("Arial", 12, "italic"), width=20, height=5, state="readonly", values=["Reporte de Tokens", "R
listasReportes.current(0)
listasReportes.bind("<<ComboboxSelected>>", callbackFunc)
Entrada = Text(VentanaPrincipal, font=("Arial", 12, "italic"))
BarraVertical = Scrollbar(Entrada)
BarraHorizontal = Scrollbar(Entrada, orient=HORIZONTAL)
Entrada.config(yscrollcommand=BarraVertical.set, wrap = "none", xscrollcommand=BarraHorizontal.set)
BarraHorizontal.pack(side = BOTTOM, fill = X)
BarraHorizontal.config(command=Entrada.xview)
BarraVertical.pack(side = RIGHT, fill = Y)
BarraVertical.config(command=Entrada.yview)

Titulo.place(x=300,y=25,width=600)
TituloReportes.place(x=1000,y=0,width=100)
botonCargar.place(x=0, y=0)
botonAnalizar.place(x=0, y=45)
listasReportes.place(x=975,y=30)
Entrada.place(x= 25,y=90,width=1150,height=500 )

VentanaPrincipal.mainloop()
```

Botón Reportes

Funciones que son utilizadas en la pestaña de reportes.

```
def callbackFunc(event):
    global ComprobacionCarga
    lexico = AnalizadorLexico()
    if listasReportes.get() == "Reporte de Tokens":
        if ComprobacionCarga == True:
            lexico.reporteTokens()
        else:
            messagebox.showinfo(message="No ha analizado el archivo", title="Error")

    elif listasReportes.get() == "Reporte de Errores":
        if ComprobacionCarga == True:
            lexico.reporteErrores()
        else:
            messagebox.showinfo(message="No ha analizado el archivo", title="Error")

    elif listasReportes.get() == "Manual de Usuario":
        startfile('./Documentación/ManualUsuario.pdf')

    elif listasReportes.get() == "Manual Técnico":
        startfile('./Documentación/ManualTecnico.pdf')
```

Tabla de tokens

Descripción	Patrón	Expresión Regular
guionOndulado	Un carácter '~'	~
CorcheteIzquierdo	Un carácter '['	[
CorcheteDerecho	Un carácter ']']
mayorQue	Un carácter '>'	>
menorQue	Un carácter '<'	<
dosPuntos	Un carácter ':'	:
coma	Un carácter ','	,
Palabra reservada tipo	tipo	tipo
Palabra reservada nombre	nombre	nombre
Palabra reservada valor	valor	valor
Palabra reservada valores	valores	valores
Palabra reservada evento	evento	evento
Palabra reservada fondo	fondo	fondo
Palabra reservada formulario	formulario	formulario

Palabra reservada entrada	entrada	entrada
Palabra reservada info	info	info
Cadena de texto	Una letra entre comillas	$([\backslash"][\wedge\backslash n\backslash"]+[\backslash"])([\backslash"][\wedge\backslash n\backslash"]+[\backslash"])$
Cadena de texto 2	Una letra entre comillas simples	$([\backslash"][\wedge\backslash n\backslash"]+[\backslash"])([\backslash"][\wedge\backslash n\backslash"]+[\backslash"])$

Autómata FND

