



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS

LFP

MANUAL TECNICO

Nombre: Angel Francisco Sique Santos

Carnet:

2	0	2	0	–	1	2	0	3	9
---	---	---	---	---	---	---	---	---	---

Contenido

Contenido

Introducción	3
1. Objetivo.....	3
2. Requisitos del programa.	3
Pestaña principal	4
Archivo	4
Apariencia	4
Analizar.....	5
Reportes	5
Expresión regular	6
AFD	7

Introducción

Generar formularios dinámicos a través de un archivo de entrada generado por el departamento de informática de la Universidad de San Carlos y a partir de él generar formularios dinámicos en formato HTML.

1. Objetivo.

Implemente un analizador léxico utilizando los conceptos de alfabeto, tokens y sus propiedades.

2. Requisitos del programa.

- 200mb de disco duro.
- 512mb de RAM
- Windows 7
- Java 8
- Equipo Intel Pentium o superior

Opciones del programa

Pestaña principal

Para crear la interfaz se usó la librería PySimpleGUI.

```
from Analizador import Ana
import PySimpleGUI as sg
from tkinter import font
```

```
##----CONFIGURACION DE LA PANTALLA-----##

def main_window(settings):
    elem_width= 80
    menu_layout = [
        ['Archivo',['Nuevo','Abrir','Guardar','Guardar Como','---','Salir']],
        ['Apariencia',['Temas', settings['themes'],'Fuente','Tamaño de letras','Mostrar configuración']],
        ['Analizar',['Analizar']],
        ['Reportes',['Reporte de Tokens','Reporte de Errores','Reporte de Usuario','Reporte Tecnico']]

    col1 = sg.Column([[sg.Multiline(default_text=settings['body'], font=settings['font'], key='_BODY_', size=(elem_width,20))]])
    col2 = sg.Column([[sg.Multiline(default_text=settings['out'], font=settings['font'], key='_OUT_', autoscroll=True, size=(elem_width,8))]])

    window_layout = [
        [sg.Menu(menu_layout)],
        [sg.Text(settings['info'], key='_INFO_', font=('Consolas',11), size=(elem_width,1))],
        [sg.Pane([col1, col2])]]

    window = sg.Window('Analizador Lexico', window_layout, resizable=True, margins=(0,0), return_keyboard_events=True)
    return window
```

Se crea un layout donde se define el tamaño y las pestañas y opciones que incluye.

Archivo

La librería PySimpleGUI facilita la creación de la ventana y la manera de agregar opciones es la siguiente.

```
['Archivo',['Nuevo','Abrir','Guardar','Guardar Como','---','Salir']],
```

Luego de agregar las opciones se agrega los eventos al escoger cada opción.

```
if event in (None, 'Salir'):
    close_settings()
    break
if event in ('Nuevo','n:78'):
    new_file(window)
if event in ('Abrir','o:79'):
    open_file(window)
if event in ('Guardar','s:83'):
    save_file(window, values)
if event in ('Guardar Como',):
    save_file_as(window, values)
```

Apariencia

De igual manera que con la ventana archivos se crea la ventana Apariencia y la forma de definir las acciones a realizar es igual..

```
['Apariencia',['Temas', settings['themes'],'Fuente','Tamaño de letras','Mostrar configuración']],
```

```

if event in ('Fuente',):
    change_font(window)
if event in ('Tamaño de letras',):
    change_tabsize(window)
if event in ('Mostrar configuración',):
    show_settings()

```

Analizar

Esta ventana solo incluye la opción Analizar.

```

[ 'Apariencia', [ 'Temas', 'selec
[ 'Analizar', [ 'Analizar' ] ],

```

Esta opción inicializa el analizador léxico, el cual analiza el archivo .form ingresado y realiza un HTML con las opciones validas y que cumplen las reglas del lenguaje.

```

def run_module(): # F5
    countLabel = []
    cadena = open(settings.get('filename'),'r+').read()

    lexico = AnalizadorLexico()
    lexico.analizar(cadena)

    listaTokens = lexico.listaTokens

    strHtml = ''
    strInfo = '<div id="div1" style="visibility: hidden;">'
    strScriptInfoOption = ''
    strScriptInfo = ''
    strHtmlinicio = '<!DOCTYPE html>...'
    f = open('Formulario.html','w')

```

```

for i in range(0,len(listaTokens)):
    #llenado de etiquetas
    if listaTokens[i].tipo == 'dosPuntos' and listaTokens[i+1].lexema == 'etiqueta' and listaTokens[i-1].tipo == 'reservada_tipo' and listaTokens[i+2].tipo == 'coma':...
    #llenado de textos
    if listaTokens[i].tipo == 'dosPuntos' and listaTokens[i+1].lexema == 'texto' and listaTokens[i-1].tipo == 'reservada_tipo' and listaTokens[i+2].tipo == 'coma':...
    #llenado de group-radio
    if listaTokens[i].tipo == 'dosPuntos' and listaTokens[i+1].lexema == 'grupo-radio' and listaTokens[i-1].tipo == 'reservada_tipo' and listaTokens[i+2].tipo == 'coma':...
    #llenado de group-option
    if listaTokens[i].tipo == 'dosPuntos' and listaTokens[i+1].lexema == 'grupo-option' and listaTokens[i-1].tipo == 'reservada_tipo' and listaTokens[i+2].tipo == 'coma':...
    #llenado de botones
    if listaTokens[i].tipo == 'dosPuntos' and listaTokens[i+1].lexema == 'boton' and listaTokens[i-1].tipo == 'reservada_tipo' and listaTokens[i+2].tipo == 'coma':...

strHtml += '...'
f.write(strHtmlinicio+strHtml)
f.close()
webbrowser.open('Formulario.html')

```

Reportes

La creación de la ventana es igual que las demás, así como las acciones al seleccionar cada opción.

```

[ 'Reportes', [ 'Reporte de Tokens', 'Reporte de Errores', 'Reporte de Usuario', 'Reporte Tecnico' ] ]

```

```

if event in ('Reporte Tecnico',):
    show_ReporteTecnico()
if event in ('Reporte de Tokens',):
    cadena = open(settings.get('filename'),'r+').read()
    lexico.analizar(cadena)
    lexico.imprimirTokens()
if event in ('Reporte de Errores',):
    lexico = AnalizadorLexico()
    cadena = open(settings.get('filename'),'r+').read()
    lexico.analizar(cadena)
    lexico.imprimirErrores()
if event in ('Reporte de Usuario',):
    show_ReporteUsuario()

```

```

##---REPORTES-----##
v def show_ReporteUsuario():
    webbrowser.open('Manual de usuario.pdf')

v def show_ReporteTecnico():
    webbrowser.open('Manual tecnico.pdf')

```

Expresión regular

El analizador tiene 10 estados.

- En el estado 1 la expresión es: \d
- En el estado 2 la expresión es: \D
- En el estado 3 la expresión es: <
- En el estado 4 la expresión es: >
- En el estado 5 la expresión es: :
- En el estado 6 la expresión es: [
- En el estado 7 la expresión es:]
- En el estado 8 la expresión es: ,
- En el estado 9 la expresión es: '(d+|D+)'"(d+|D+)'"
- En el estado 10 la expresión es: ~

AFD

