

PROYECTO 1 - FORMULARIOS HTML

MANUAL TÉCNICO

FACULTAD DE INGENIERÍA, USAC

MARZO 2022

YENIFER ESTER YOC LARIOS, 201952336

OBJETIVOS

Objetivo General:

Brindar a todas aquellas personas que desean realizar formularios en formato HTML de manera fácil y rápida, un programa tiene la capacidad de crear la cantidad de campos que el usuario desee, así como muchas otras opciones que le serán de utilidad para obtener los datos de las personas a las que les dé acceso.

Objetivos Específicos:

Presentar una interfaz interactiva, que permita una interacción entre el usuario y el programa.

Implementar el análisis de un archivo de entrada

Generar el formulario con las especificaciones indicadas

Alcances:

Que pueda implementarse el análisis léxico en un programa que a su vez permita generar archivos en de extensión HTML

Especificación Técnica: Lenguaje Python

Requisitos del hardware:

- Un ordenador de escritorio o portátil.

Requisitos del Software:

- El programa presentado, fue desarrollado en un sistema operativo de 64 bits
- Para su correcta ejecución es necesario que se tenga instalado en el ordenador el IDE "Visual Studio Code".
- El presente programa no tiene retención de datos más que en la memoria RAM, luego de cerrado el programa, la memoria es volátil.
- El desarrollo del trabajo presentado fue bajo el procesador Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz

LÓGICA DEL PROGRAMA

Se realizó una interfaz gráfica por medio del lenguaje de programación Python, se crearon la clase Main

```
def main(): #METODO PRINCIPAL QUE INVOCA AL MENU2
    app = Todo()

if __name__ == "__main__":
    main()
```

El menú

```
def opcion_elegida(self,opcion):
    if(opcion == 1): ...

    if(opcion == 2): ...

    if(opcion == 3): ...

    if(opcion == 4): ...
```

Se creó la clase Token

```
class Token():
    lexema_valido = ''
    tipo = 0
    fila = 0
    columna = 0

    PALABRA_RESERVADA = 1
    CADENA = 2
    NUMERO = 3
    CURVA = 4
    MAYOR_QUE = 5
    MENOR_QUE = 6
    CORCHETE_ABRE = 7
    CORCHETE_CIERRA = 8
    COMA = 9
    DOS_PUNTOS = 25
    DESCONOCIDO = 10
    TIPO = 11
    VALOR = 12
    FONDO = 13
    VALORES = 14
    EVENTO = 15
    ENTRADA = 16
    INFO = 17
    LETRAS = 18

    def __init__(self,lexema,tipo,fila,columna): ...

    def getLexema(self): ...
```

La clase Analizador Léxico y sintáctico

```

from Token import Token
class Analizador_Lexico():
    lexema = ''
    tokens= []
    tokens_bien = []
    tokens_errores= []
    estado = 1
    fila = 1
    columna = 1
    generar = False

    def analisis(self,entrada): ...
    def AgregarToken(self,tipo): ...
    def RESERVADA(self): ...
    def Imprimir(self): ...
    def ImprimirErrores(self): ...

from Token import *
from Elemento import *
class Sintactico_form():
    nombre_mes = ''
    año_gra = 0
    arreglo_elementos = []
    lista_tokens = []
    arreglo_lleno = []
    datos_generales = []

    def analizar(self,tokens): ...
    def Buscar_elemento(self, tipo): ...
    def Buscar_pos_menorque(self, inicio, fin, tipo): ...
    def Buscar_atributo(self, inicio, fin, lexema): ...
    def Buscar_valores(self, inicio, fin, lexema): ...
    def printTokens(self): ...

```

Para la creación del formulario y su funcionamiento se utilizó JavaScript y su código se generó junto con el HTML

Estructura del Archivo

Formulario

El formulario dará inicio con la palabra reservada formulario seguida de los caracteres ~>> seguido de un arreglo de elementos encerrados con [] que contendrá el formulario.

- **Formulario** será una palabra reservada.

Elemento

La lista de elementos dentro del formulario tendrá 1 o muchos elementos separados por comas, cada elemento estará encerrado entre los caracteres < > y tendrán las siguientes propiedades.

1. **tipo**: Este será el tipo de elemento que se colocará en el formulario, podrá tomar los siguientes valores con su correspondencia en html

- **Tipo** será una palabra reservada

Sus valores pueden ser:

Etiqueta, Texto, Grupo radio. Grupo-option, botón

Serán de tipo cadena

2. **valor**: Es una palabra reservada del lenguaje. Su valor es una cadena, letras, seguido de más letras o números

3. **fondo:** Es una palabra reservada y su valor es de tipo cadena, empieza con comillas, seguido de letras o números
4. **valores:** Es una palabra reservada, su valor es un arreglo de valores de letras o números separadas por coma
5. **evento:** Es una palabra reservada, sus valores son palabras reservadas, INFO y Entrada

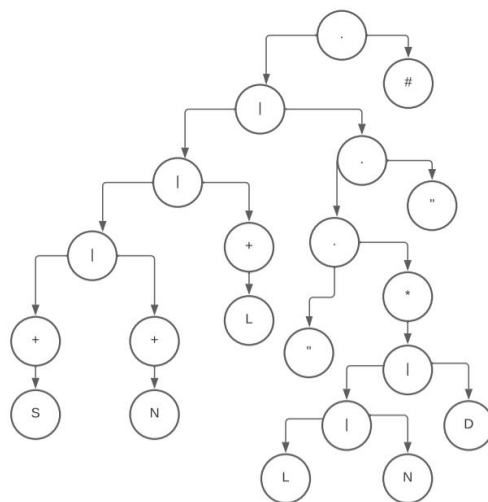
Expresiones Regulares:

Símbolos = [curva, mayor que, menor que, corchete abre, corchete cierra, coma, dos puntos]
Números = N = [0-9]+ = N+
Letras = L = [a-zA-Z]+ = L+
Desconocido = D = [Cualquiera]
Cadenas = "(L N D)*"

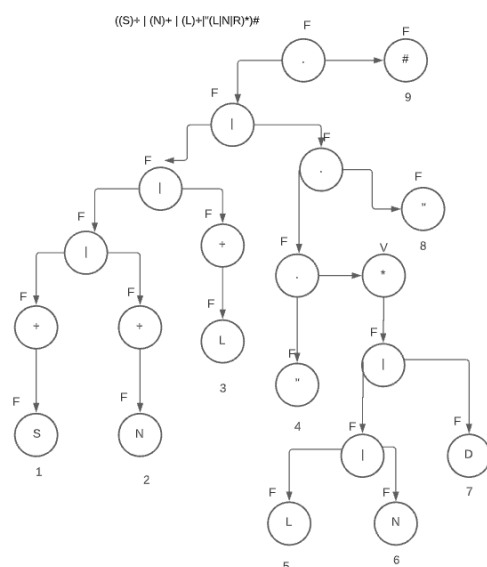
Expresión Regular

$((S)^+ | (N)^+ | (L)^+ | "(L|N|D)^+")\#$

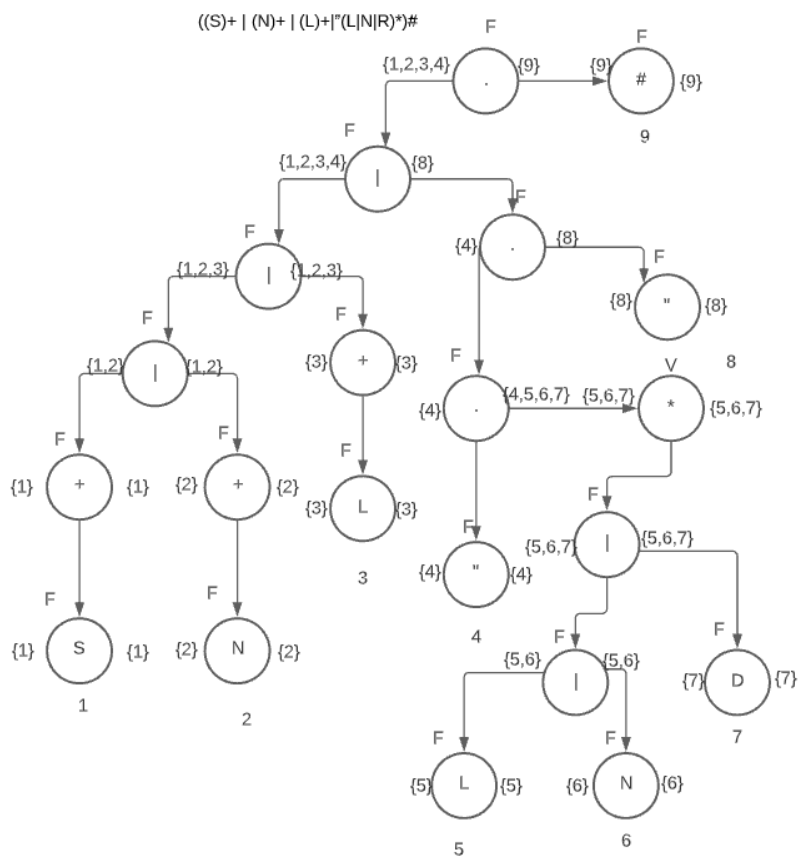
Crear árbol de sintaxis



ANULABLES



PRIMEROS Y ÚLTIMOS



SIGUIENTES

I	TERMINAL	SIGUIENTE
1	S	1,9
2	N	2,9
3	L	4,8
4	"	8,9
5	L	7
6	N	7
7	D	7
8	#	

TABLA

ESTADO	S	N	L	"	D	#
S1 = {1(S), 2(N), 3(L), 4(")}	S2	S3	S4	S5		
S2 = {1(L), 2	S2					
S3		S3				
S4			S4			
S5	S5	S5	S5	S6	S5	
S6						

AUTÓMATA

