

Universidad de San Carlos de Guatemala -USAC-

Facultad de Ingeniería

Ingeniería en Ciencias y Sistemas

Ing. Otto Amilcar Rodríguez Acosta

Aux. Carlos Esteban Godínez Delgado

Lenguajes Formales y de Programación A+

PROYECTO FINAL
COMPILADOR DE LENGUAJE ESPECÍFICO A
HTML
(Manual Técnico)

Robin Omar Buezo Díaz

Carné 201944994

Viernes 28 de octubre de 2022.

El presente documento tiene como finalidad mostrar al usuario la funcionalidad y construcción del software para que entienda su creación y pueda dar solución a cualquier error que pueda presentarse.

Se explican el flujo y las diferentes partes que constituyen el software y como debemos de interactuar con este para que el sistema nos sea de gran ayuda.

PARADIGMA DE PROGRAMACIÓN

Para la creación de este software se utilizó el paradigma de Programación Orientada a Objetos, ya que esto da una mejor facilidad a la hora de manejar el archivo que se está manipulando a lo largo de toda la ejecución, como también el poder encapsular los objetos y luego poder utilizar los mismos objetos en las diferentes ventanas que se utilizan a lo largo del programa.

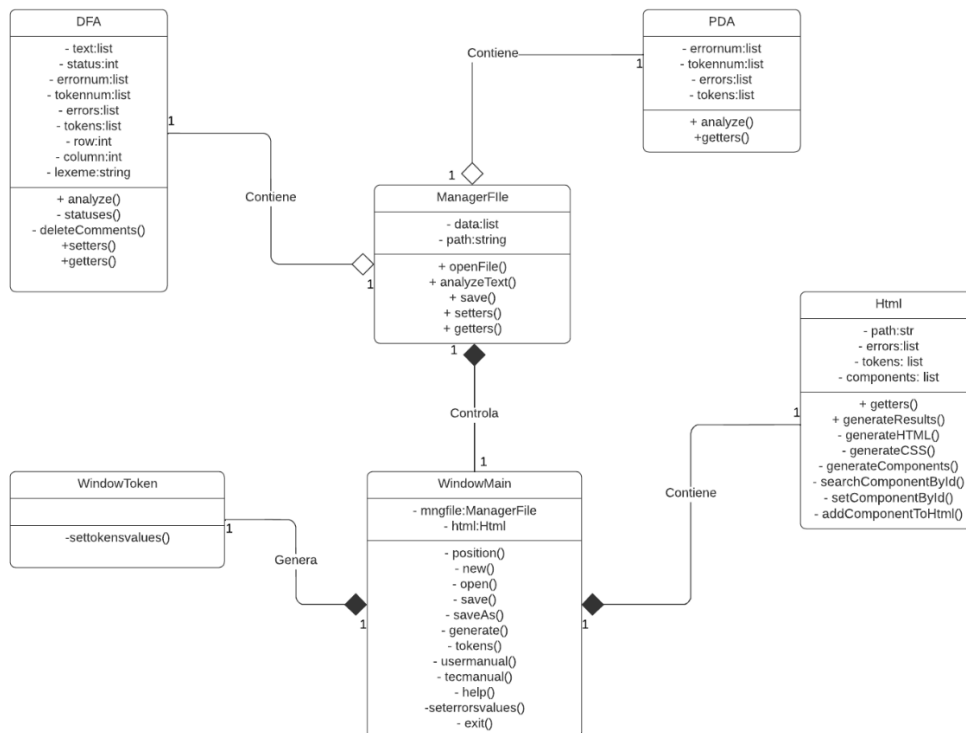
NOMENCLATURA

Identificador	Regla	Ejemplo
Clases	Sustantivos, primera letra mayúscula	class MyClass
Métodos	Verbos, primera letra minúscula y la del resto de palabras en mayúscula	ejecutar() cargarArchivo()
Variables	En minúsculas y deben empezar con una letra	variableuno variabledes

DIAGRAMA DE CLASES

Diagrama de Clases

ROBIN OMAR BUEZO DÍAZ | October 28, 2022



TOKENS

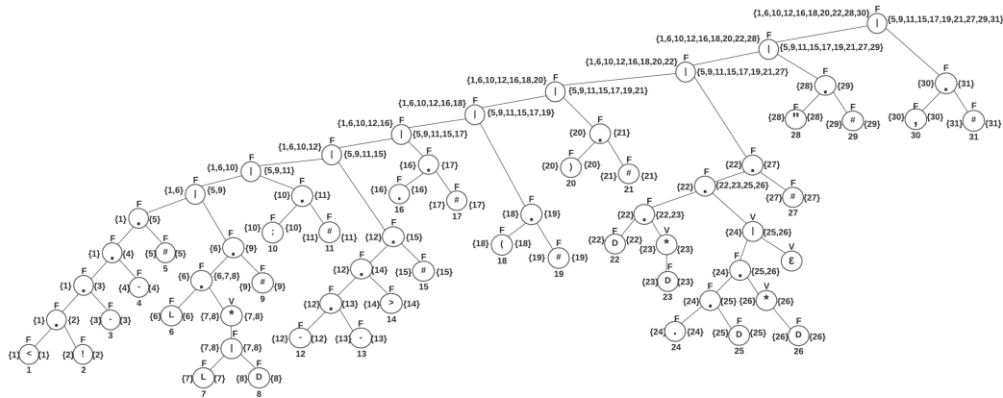
Tokens

- 1 = Apertura = <!--
- 2 = ID = L(L|D)*
- 3 = PuntoComa = ;
- 4 = Cierre = -->
- 5 = Punto = .
- 6 = ParentesisA = (
- 7 = ParentesisC =)
- 8 = Numero = DD*((.DD*)|ε)
- 9 = Comilla = "
- 10 = Coma = ,

Palabras Reservadas

- 11 = Controles
- 12 = Control
 - Contenedor
 - Boton
 - Clave
 - Etiqueta
 - Texto
 - Check
 - RadioBoton
 - AreaTexto
- 13 = Propiedades
- 14 = Colocacion

CONSTRUCCIÓN DE AFD



Nodo	Sig()
1, <	2
2, I	3
3, >	4
4, .	5
5, #	
6, L	7, 8, 9
7, L	7, 8, 9
8, D	7, 8, 9
9, #	
10, .	11
11, #	
12, >	13
13, >	14
14, >	15
15, #	
16, L	17
17, #	
18, L	19
19, #	
20, .	21
21, #	
22, D	23, 24, 27
23, D	23, 24, 27
24, .	25
25, D	26, 27
26, D	26, 27
27, #	
28, >	29
29, #	
30, .	31
31, #	

$S_0 = \{1, 6, 10, 12, 16, 18, 20, 22, 28, 30\} = \{<, L, ., >, ., D, ., .\}$
 $Sig(<) = Sig(1) = \{2\} = S_1$
 $Sig(L) = Sig(6) = \{7, 8, 9\} = S_5$
 $Sig(.) = Sig(10) = \{11\} = S_4$
 $Sig(>) = Sig(12) = \{13\} = S_7$
 $Sig(L) = Sig(16) = \{17\} = S_{10}$
 $Sig(.) = Sig(18) = \{19\} = S_{11}$
 $Sig(D) = Sig(20) = \{21\} = S_{12}$
 $Sig(D) = Sig(22) = \{23, 24, 27\} = S_{13}$
 $Sig(>) = Sig(28) = \{29\} = S_{16}$
 $Sig(.) = Sig(30) = \{31\} = S_{17}$

$S_1 = \{2\} = \{<\}$
 $Sig(<) = Sig(2) = \{3\} = S_2$

$S_2 = \{3\} = \{>\}$
 $Sig(>) = Sig(3) = \{4\} = S_3$

$S_3 = \{4\} = \{.\}$
 $Sig(.) = Sig(4) = \{5\} = S_4$

$S_5 = \{7, 8, 9\} = \{L, D, \# \}$
 $Sig(L) = Sig(7) = \{7, 8, 9\} = S_5$
 $Sig(D) = Sig(8) = \{7, 8, 9\} = S_5$

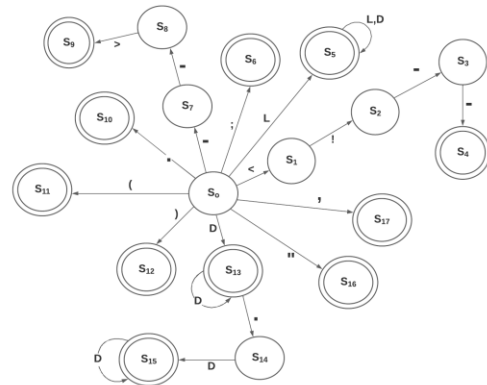
$S_7 = \{13\} = \{>\}$
 $Sig(>) = Sig(13) = \{14\} = S_8$

$S_8 = \{14\} = \{>\}$
 $Sig(>) = Sig(14) = \{15\} = S_9$

$S_{13} = \{23, 24, 27\} = \{D, ., \# \}$
 $Sig(D) = Sig(23) = \{23, 24, 27\} = S_{13}$
 $Sig(.) = Sig(24) = \{25\} = S_{14}$

$S_{14} = \{25\} = \{D\}$
 $Sig(D) = Sig(25) = \{26, 27\} = S_{15}$

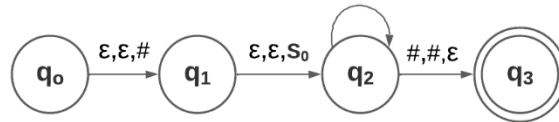
$S_{15} = \{26, 27\} = \{D, \# \}$
 $Sig(D) = Sig(26) = \{26, 27\} = S_{15}$



GRAMÁTICA Y PDA

$S_0 \rightarrow S_1 S_2 S_3$
 $S_1 \rightarrow 1 11 S_4 11 4$
 $S_2 \rightarrow 1 13 S_5 13 4$
 $S_3 \rightarrow 1 14 S_5 14 4$
 $S_4 \rightarrow 12 2 3 S_4$
 $\quad | \epsilon$
 $S_5 \rightarrow 2 5 2 6 S_6 7 3 S_5$
 $\quad | \epsilon$
 $S_6 \rightarrow 8 S_7$
 $\quad | 9 S_8 9$
 $\quad | 2$
 $S_7 \rightarrow 10 8 S_7$
 $\quad | \epsilon$
 $S_8 \rightarrow 2 S_8$
 $\quad | 8 S_8$
 $\quad | \epsilon$

$14, 14, \epsilon$
 $13, 13, \epsilon$
 $12, 12, \epsilon$
 $11, 11, \epsilon$
 $10, 10, \epsilon$
 $9, 9, \epsilon$
 $8, 8, \epsilon$
 $7, 7, \epsilon$
 $6, 6, \epsilon$
 $5, 5, \epsilon$
 $4, 4, \epsilon$
 $3, 3, \epsilon$
 $2, 2, \epsilon$
 $1, 1, \epsilon$
 $\epsilon, S_8, 2 S_8$
 $\epsilon, S_8, 8 S_8$
 ϵ, S_8, ϵ
 $\epsilon, S_7, 10 8 S_7$
 ϵ, S_7, ϵ
 $\epsilon, S_6, 8 S_7$
 $\epsilon, S_6, 9 S_8 9$
 $\epsilon, S_6, 2$
 $\epsilon, S_5, 2 5 2 6 S_6 7 3 S_5$
 ϵ, S_5, ϵ
 $\epsilon, S_4, 12 2 3 S_4$
 ϵ, S_4, ϵ
 $\epsilon, S_3, 1 14 S_5 14 4$
 $\epsilon, S_2, 1 13 S_5 13 4$
 $\epsilon, S_1, 1 11 S_4 11 4$
 $\epsilon, S_0, S_1 S_2 S_3$



METODOS PRINCIPALES

openFile:

Este método nos permite cargar el archivo que se quiere editar o manipular desde el programa.

generate:

Este método se encarga de correr el AFD y el PDA sobre el programa para poder determinar si el texto esta correctamente apegado al lenguaje y si es así poder ejecutar las instrucciones, también es el método que se encarga de encontrar los errores en nuestro código.

save:

Este método nos permite guardar nuestro código ya sea sobrescribiendo el mismo archivo o bien generando uno nuevo.

position:

Este método se encarga de devolvernos en todo momento nuestra posición dentro del editor de texto.

tokens:

Este método se encarga de generar la ventana con la tabla de tokens.

seterrorsvalues:

Este método se encarga de poblar la tabla de errores si en caso se encontraran en el análisis.

HERRAMIENTAS

Para poder dar solución a los requerimientos anteriores se utilizó el lenguaje de programación Python versión 3 y su documentación por su versatilidad y fácil programación.

Como herramienta de programación se utilizó el programa Visual Studio Code por su amplia funcionalidad y herramientas que brinda a los programadores a la hora de programar en cualquier lenguaje.

Para poder realizar los diagramas, el AFD y el DFA se utilizó la herramienta Lucidchart por sus plantillas que nos facilitan el hacer diagramas más profesionales y entendibles.

Por último, se utilizó la herramienta de versionamiento GitHub. Para poder tener un mejor control sobre los cambios que se iban realizando en nuestro código y no tener el problema de perder funcionalidad si en caso algún cambio ocasionaba erros.

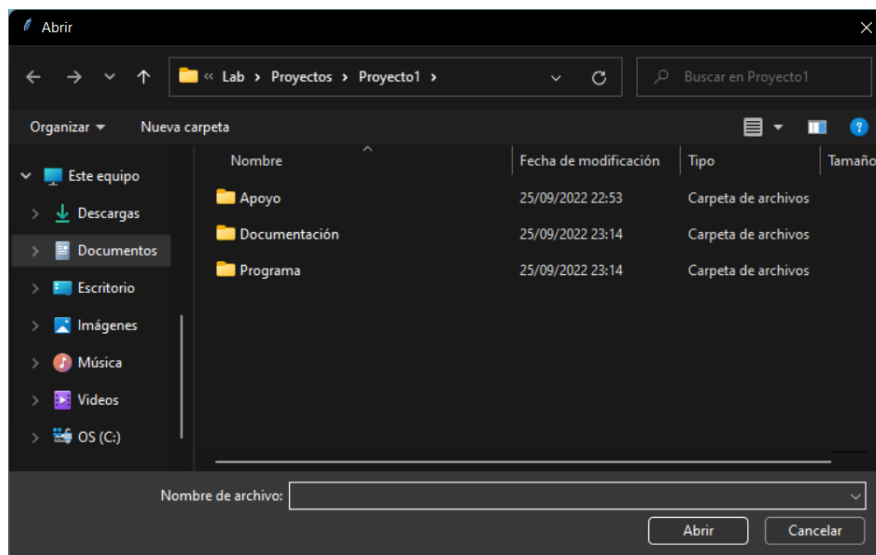
GUI PRINCIPALES

Practica 1 (Ventana Principal)

Esta es la ventana principal y en donde tendremos a nuestro editor de texto para poder manipular nuestro código, como también nuestra barra de menú para el resto de las opciones.

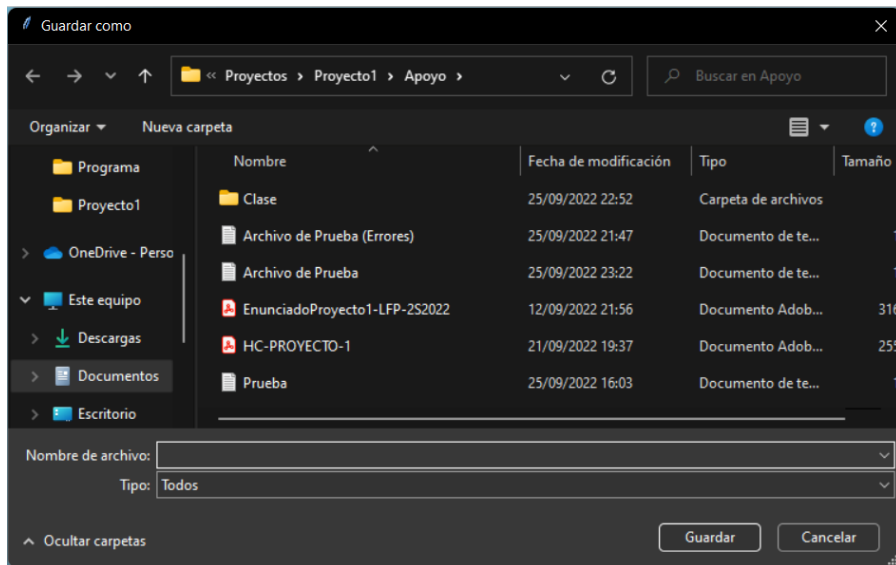
Abrir

Esta opción nos abrirá una ventana emergente desde donde podremos buscar y seleccionar el archivo que queremos cargar.



Guardar Como

Este botón nos permite guardar nuestro archivo, pero creando otro archivo sin modificar el archivo cargado originalmente. Para ello también tendremos una ventana emergente.



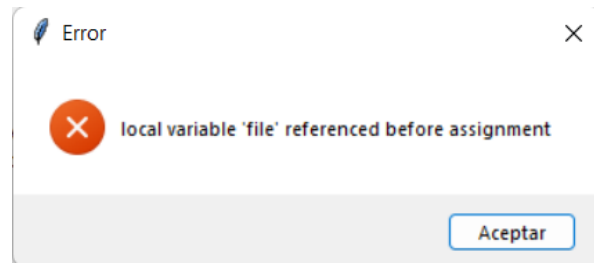
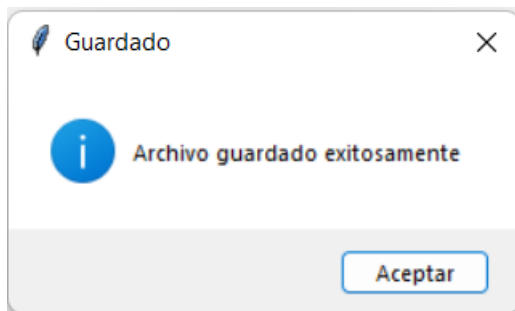
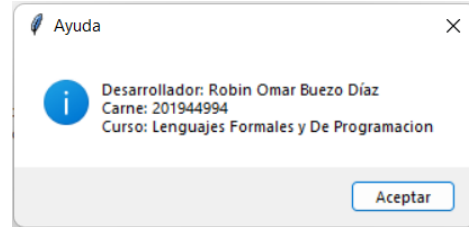
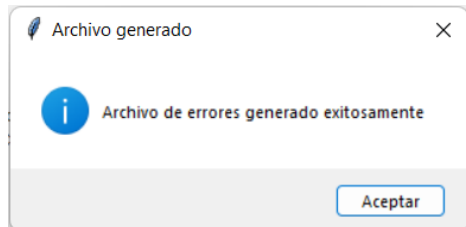
Tokens

Esta ventana nos muestra la tabla de los tokens que se encontraron en el análisis del código.

No.	Token	Numero	Lexema
1	Apertura	1	<!--
2	Controles	11	Controles
3	Control	12	Contenedor
4	ID	2	contlogin
5	PuntoComa	3	;
6	Control	12	Contenedor
7	ID	2	contFondo
8	PuntoComa	3	;
9	Control	12	Boton
10	ID	2	cmdIngresar
11	PuntoComa	3	;
12	Control	12	Clave
13	ID	2	pswClave
14	PuntoComa	3	;
15	Control	12	Etiqueta
16	ID	2	passw
17	PuntoComa	3	;
18	Control	12	Etiqueta

Cuadros de diálogo

Cabe destacar que nuestro sistema nos irá mostrando cuadros de diálogo a lo largo de la ejecución para poder darnos información o bien mostrarnos errores que puedan surgir.



GLOSARIO

HTML: Es un lenguaje de marcado que nos permite indicar la estructura de nuestro documento mediante etiquetas.

CSS: Es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML (en-US) (incluyendo varios lenguajes basados en XML como SVG, MathML o XHTML).

AFD (Autómata Finito Determinista): Un autómata finito determinista (abreviado AFD) es un autómata finito que además es un sistema determinista; es decir, para cada estado en que se encuentre el autómata, y con cualquier símbolo del alfabeto leído, existe siempre no más de una transición posible desde ese estado y con ese símbolo.

PDA (Pushdown Automata): Un autómata con pila, autómata a pila o autómata de pila es un modelo matemático de un sistema que recibe una cadena constituida por símbolos de un alfabeto y determina si esa cadena pertenece al lenguaje que el autómata reconoce. El lenguaje que reconoce un autómata con pila pertenece al grupo de los lenguajes libres de contexto en la clasificación de la Jerarquía de Chomsky.