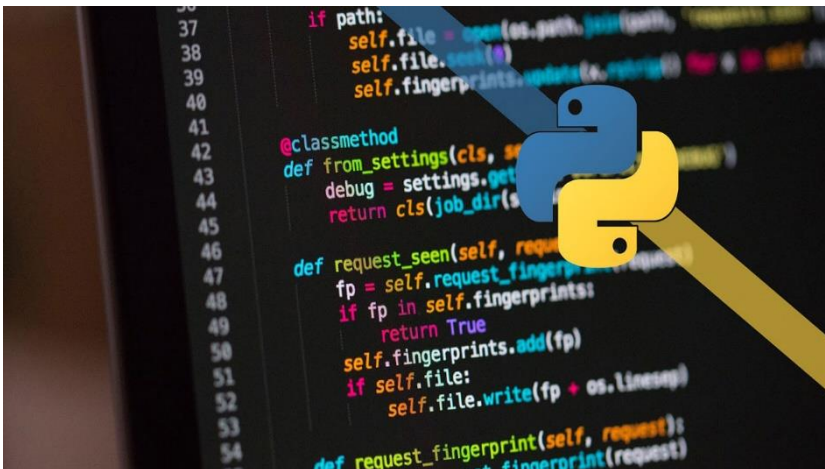


Proyecto 2

(Manual De Programador)



Universidad San Carlos De Guatemala.

Guatemala 28 de oct. de 21.

Wilber Steven Zúñiga Ruano.

Contenido

Introducción	3
Objetivos	3
Especificaciones Técnicas	3
Lógica del programa.	4
Instalación de aplicación	10

Introducción

En este manual será descrito de forma técnica el programa solicitado en el proyecto 2 del laboratorio de lenguajes formales de la programación. Se dará a conocer los aspectos más importantes para el programador sobre la aplicación, mostrando sus características y el uso de cada una de ellas.

Objetivos

- Guía completa para el programador sobre los datos técnicos del programa solicitado.
- Especificar funciones utilizadas y su uso específico.

Especificaciones Técnicas(Requerimientos)

La aplicación puede ser ejecutada en cualquier sistema operativo que cuente con los siguientes requisitos:

- Python 3
- Editor de texto o IDE (VScode)

Lógica del programa.

Clases utilizadas:

- **App.py:** Esta es la clase principal del programa, en esta tenemos los métodos principales que es la lectura del documento y pasarlo a un variable string, el método de análisis léxico, el análisis sintáctico, las presentaciones en consola y la generación de reportes.
- **Token.py:** Esta clase es la que permite almacenar en una lista los tokens que se obtienen mediante un archivo de entrada (.lfp) y así poder llevar un control del lexema, tipo, fila y columna.
- **analizadorLexico.py:** Esta clase es la más importante del programa ya que es el analizador léxico que lee el archivo y separa cada carácter dependiendo el token, encuentra los errores, genera reportes, etc.
- **analizadorSintactico.py:** Esta clase es importante para el programa ya que es la encargada de devolver cada carácter solicitado, como el máximo o mínimo de algún dato específico, un comentario, etc.
- **analizadorSintactico.py:** Esta clase es importante para poder llevar un buen manejo de los productos que almacenaremos.

Librerías utilizadas:

- **Tkinter:** Es la interfaz por defecto de Python para el toolkit de la GUI Tk, la utilizamos para desplegar la ventana para la selección del archivo a leer.
- **re:** Este módulo proporciona operaciones de coincidencia de expresiones regulares similares a las encontradas en Perl. Tanto los patrones como las cadenas de texto a buscar pueden ser cadenas de Unicode (str) así como cadenas de 8 bits (bytes).
- **PyQt5:** Es uno de los módulos más utilizados en la creación de aplicaciones con interfaces gráficas en Python y esto es debido a su simplicidad, como veras a continuación.

Métodos utilizados:

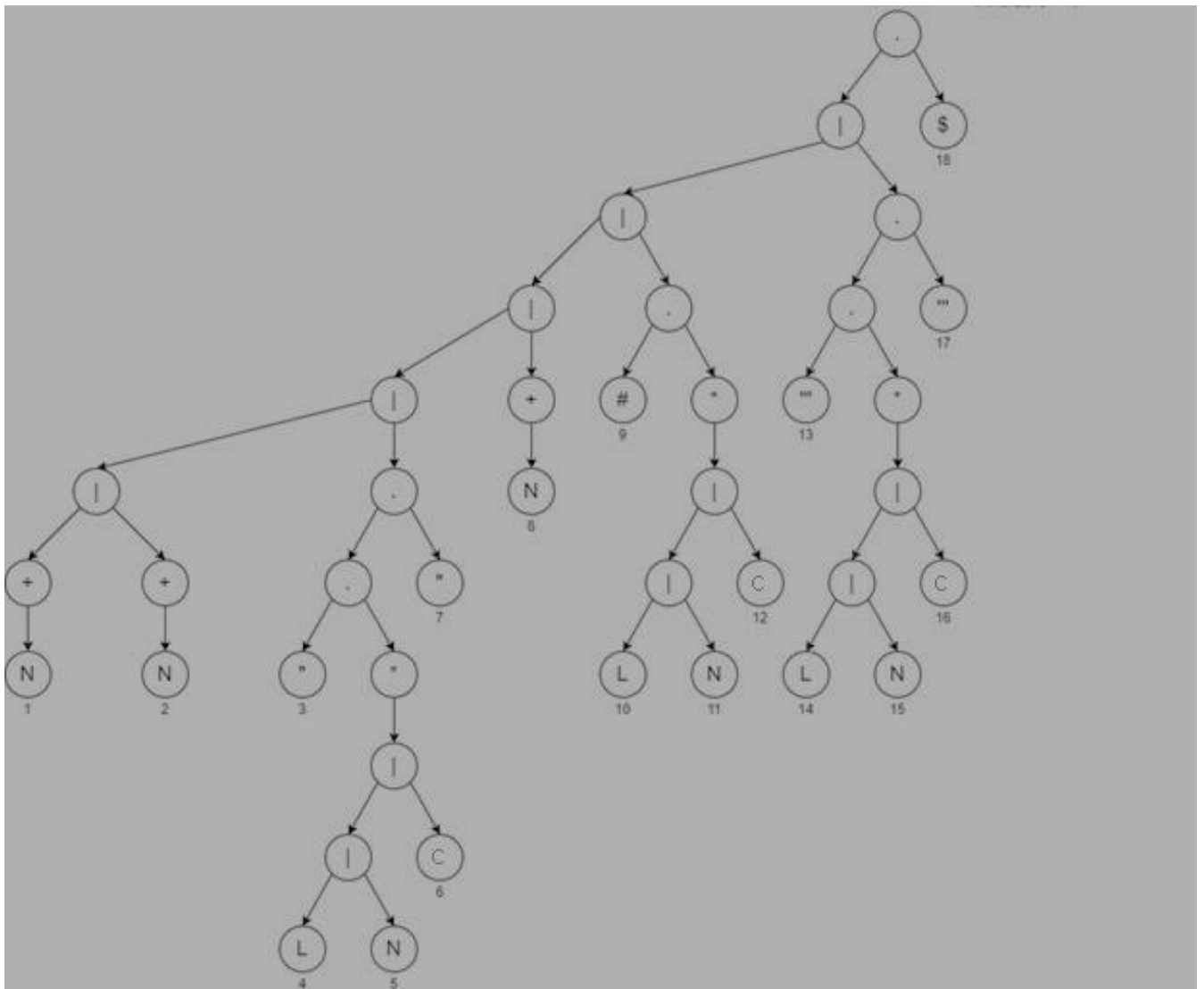
- **cargaArchivo():** Este método despliega la ventana para elegir tu archivo .pxla para poder cargar archivos, lo recibe y almacena todo el texto en una sola variable y la devuelve.
- **analizar ():** Este es el método principal, que recibe la variable con todo el texto y empieza a separar las variables a usar por medio de un autómata finito, analizando cada carácter y convirtiéndolo en un token para devolverlo, descubriendo si son palabras reservadas etc. Como también realiza el análisis sintáctico y devuelve las instrucciones solicitadas.
- **generarReporte ():** Este método genera el reporte en de los tokens encontrados y de los errores léxicos.

Expresiones Regulares

Letras – L = [A-Za-z] = L+
Signos – S = [=(){};,[]#“”]=S+
Cadena = “(L N C)*”
Numeros -N = [0-9]=N+
Cualquiera – C = [Cualquier Signo]
Comentario = #(L N C)*
Comentario = “”(L N C)*”

Expresión Regular Formada

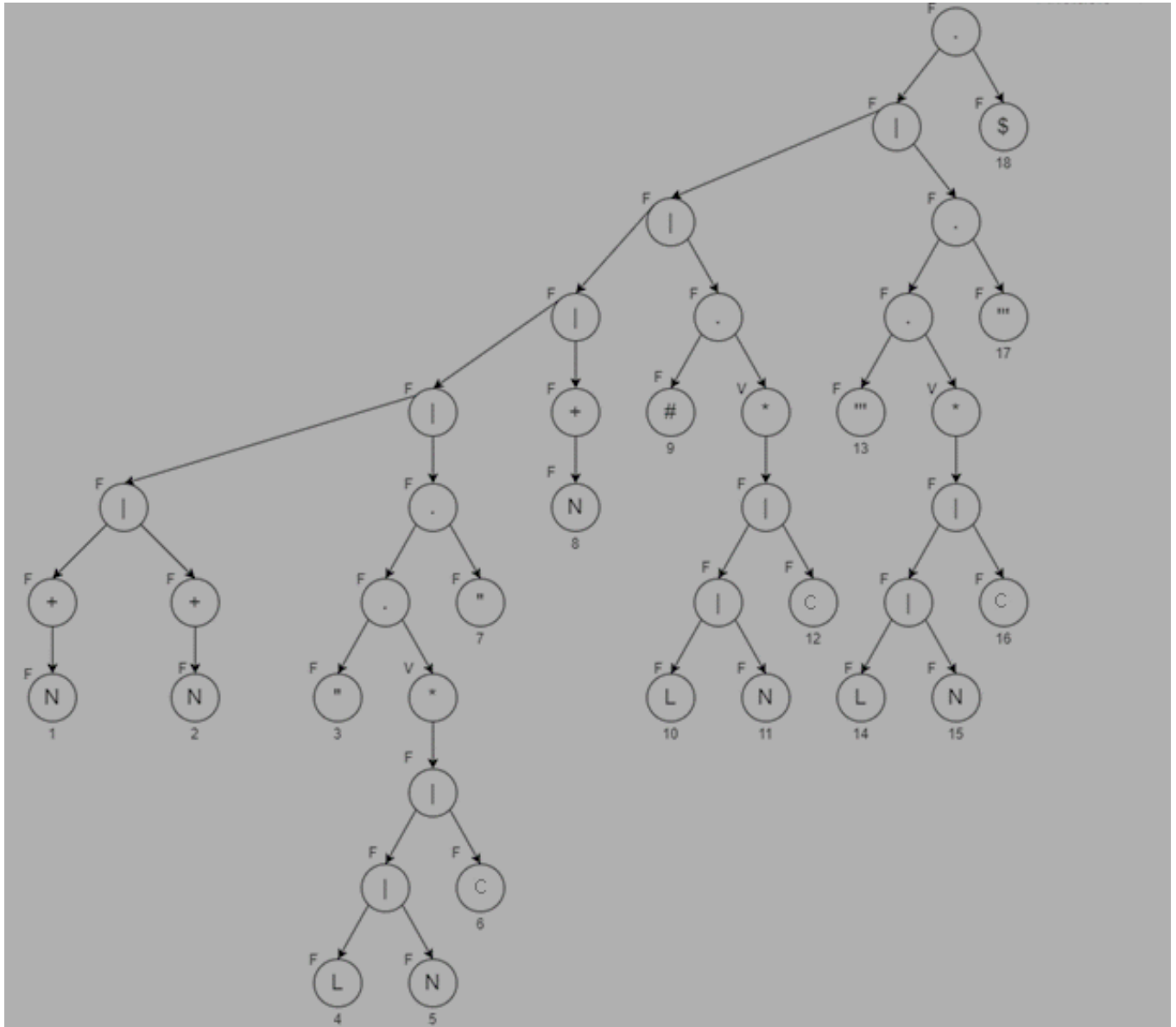
[(L)+|(S)+| “(L|N|C)*” |(N)+| #(L|N|R)* | “”(L|N|R)*”]\$



Anulables

V = Anulable

F = No Anulable



First y Post

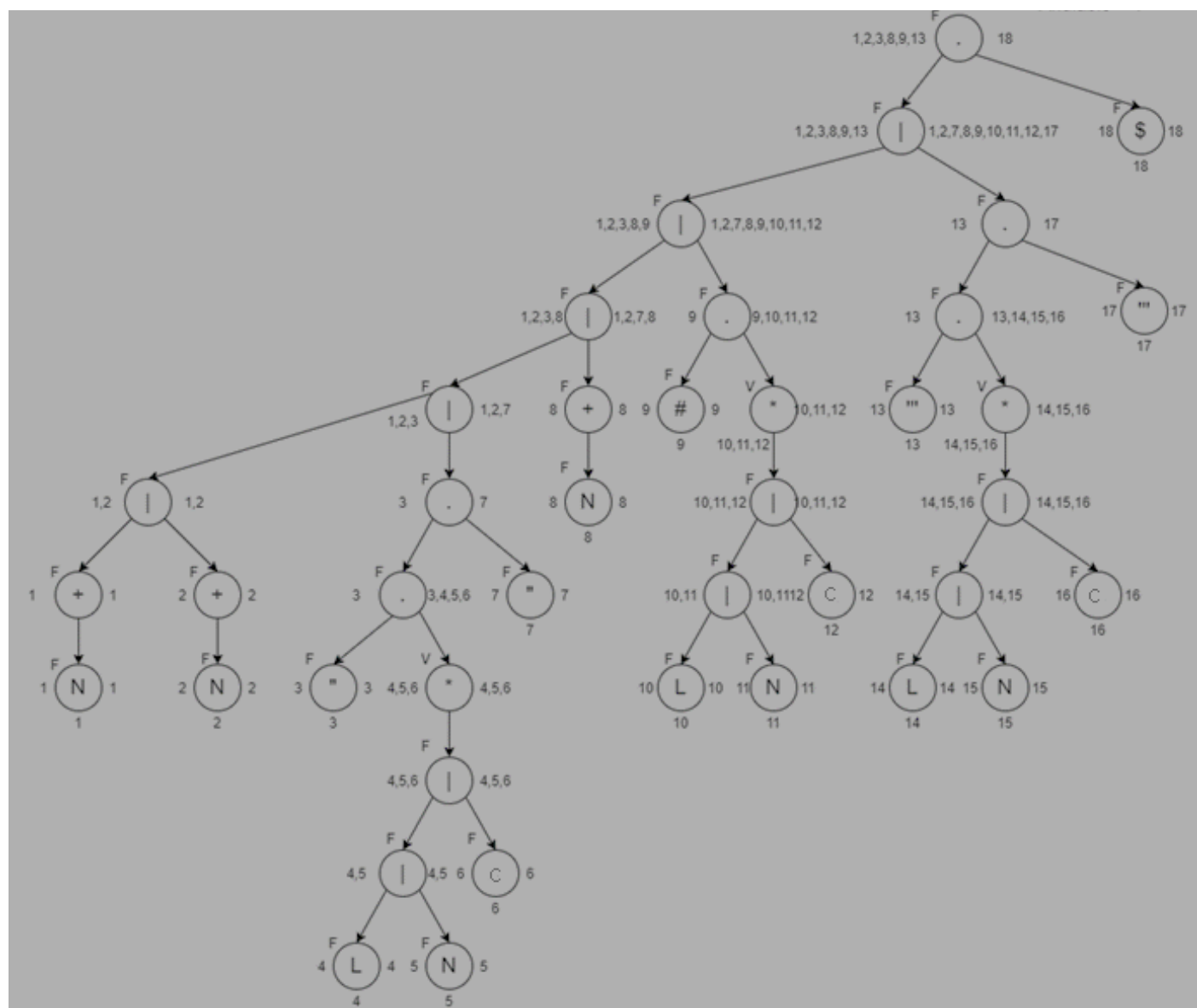


Tabla de transiciones.

NO.	TERMINAL	SIGUIENTE
1	L	1,18
2	S	2,18
3	"	4,5,6,7
4	L	4,5,6,7
5	N	4,5,6,7
6	C	4,5,6,7
7	"	18
8	N	8,18
9	#	10,11,12,18
10	L	10,11,12,18
11	N	10,11,12,18
12	C	10,11,12,18
13	'''	14,15,16,17
14	L	14,15,16,17
15	N	14,15,16,17
16	C	14,15,16,17
17	'''	18
18	\$	

Automata

<i>Estado</i>	L	S	"	N	C	#	'''	\$
S0 = 1[L],2[S],3["],8[N],9[#],13['']	S1	S2	S3	S5		S6	S7	
S1 = 1[L],18[\$]	S1							
S2 = 2[S],18[\$]	S2							
S3 = 4[L],5[N],6[C],7["]	S3		S4	S3	S3			
S4 = 18[\$]								
S5 = 8[N],18[\$]				S5				
S6 = 10[L],11[N],12[C],18[\$]	S6			S6	S6			
S7 = 14[L],15[N],16[C],17['']	S7			S7	S7			S4

Instalación de aplicación

No es necesaria una instalación como si, solo es necesario tener instalado Python y un ide como en nuestro caso visual studio code o trabajar en consola.

Solo necesitas la carpeta Principal, que contiene el programa.