MANUAL TÉCNICO

Allan Roynell González Hernández

202003708

28-10-2021

ESPECIFICACIÓN TÉCNICA

Requisitos de hardware:

- Procesador Intel preferiblemente superior a 4ta generación en caso de AMD superior a Opteron
- 2GB de Memoria RAM

Requisitos de Software:

- Contar con un sistema operativo compatible con Python, Windows, MacOS, distribuciones de Linux.
- Tener Python instalado en el sistema operativo, muy importante para poder ejecutar los procesos de cada uno de estos desde la consola de comandos o terminal.

LÓGICA DEL PROGRAMA

Clases Utilizadas:

Dentro de las clases utilizadas tenemos:

Analizador:

Esta clase maneja todo el orden lógico del programa, en este existen las diferentes instancias de los objetos y a su vez realiza el análisis léxico y sintáctico respectivo para el archivo de entrada.

Error_Sintactico:

Este se utilizó para poder almacenar errores de tipo objetos, para poder realizar un append dentro de la clase Analizador cada vez que se encuentre un error de este tipo

Registros:

Clase utilizada para objetos de los registros.

Tokens:

Clase utilizada para objetos de tokens utilizados dentro de la clase Analizador.

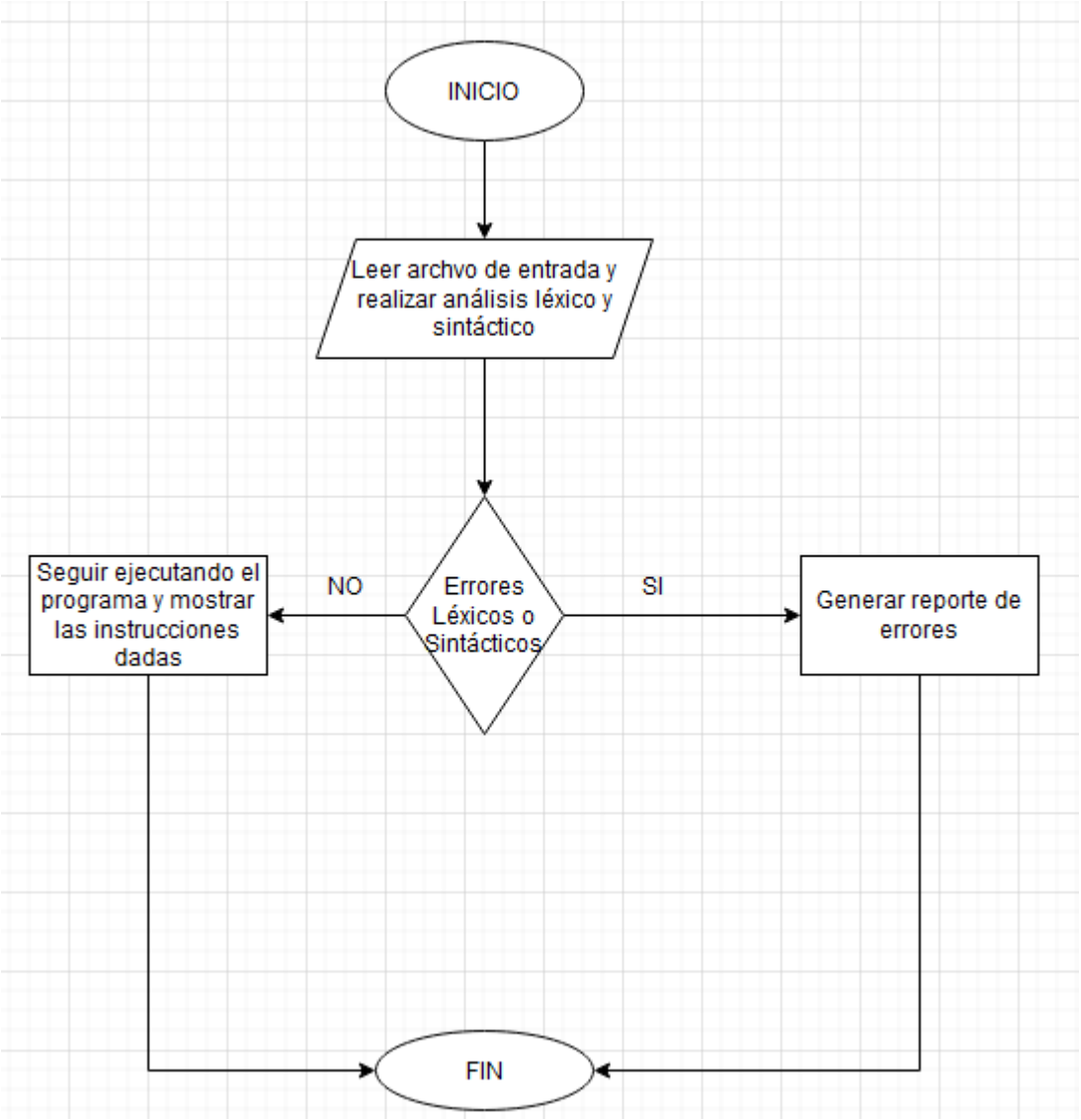
GUI:

Esta clase es utilizada manejar la interfaz con quien interactúa el usuario.

Flujo del Programa:

Se ingresa un archivo el cual tiene un conjunto de Claves, Registros e Instrucciones las cuales el programa va leyendo y a su vez verificando si existe algún tipo de error, dando a su vez un reporte si lo hay y de lo contrario sigue a la ejecución del programa en la interfaz de usuario.

DIAGRAMA DE FLUJO

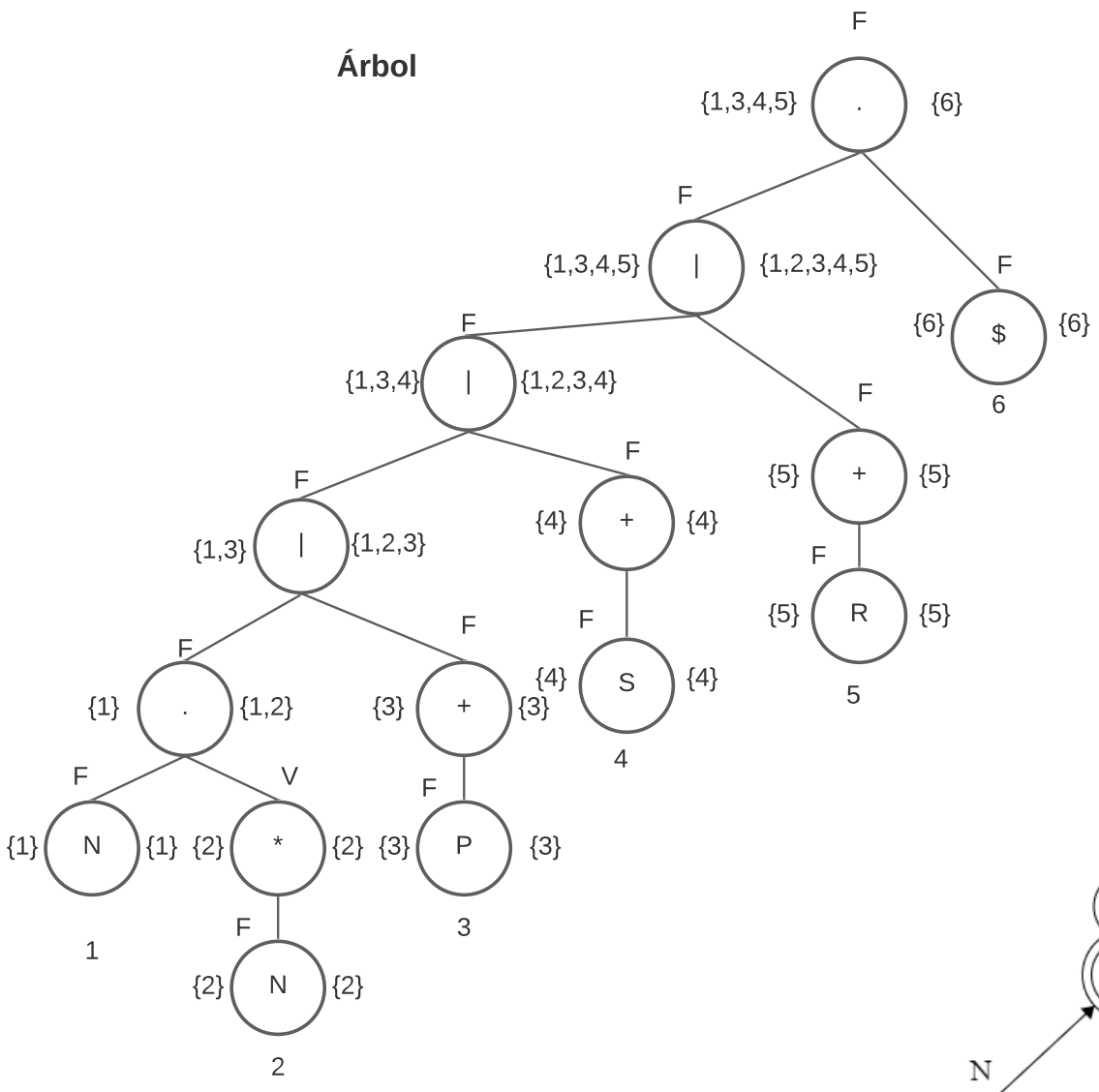


A continuación, se muestra la estructura y el comportamiento del programa mediante el método del Árbol y a su vez las expresiones regulares y gramática utilizada.

Expresión Regular

Números	N	[0-9]
Palabas Reservadas	P	[a-zA-Z]
Signos	S	[= ( ) { } , ; [ ] # ‴]
Resto de Signos	R	Signos random
Expresión Regular	[N.N* | P+ | S+ | R*]$	

Árbol



Siguientes y DFA

i	Sig(i)
1 - N	2,6
2 - N	2,6
3 - P	3,6
4 - S	4,6
5 - R	5,6
6 - $	

So = {1,2,4,5}
N N S R

Sig(N)USig(N) = Sig(1)USig(2) = {2,6} = S1

Sig(S) = Sig(4) = {4,6} = S2

Sig(R) = Sig(5) = {5,6} = S3

S1 = {2,6}
N $

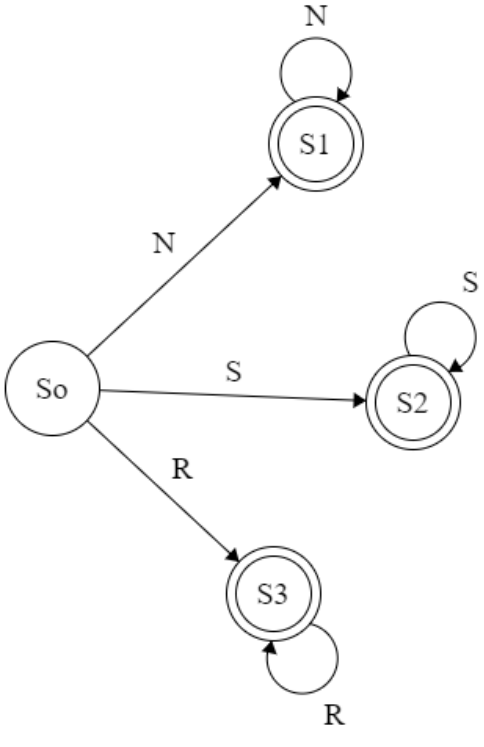
Sig(N) = Sig(2)= {2,6} = S1

S2 = {4,6}
S $

Sig(S) = Sig(4)= {4,6} = S2

S3 = {5,6}
R $

Sig(R) = Sig(5)= {5,6} = S3



Grafo de grado 3

Gramática para el Proyecto 2 LFP

<INICIO> ::= <Clave>

| <Registros>

| <Instrucciones>

<Clave> ::= <tk_Clave> <tk_igual> <tk_corcheteI> <BloqueClave> <tk_corcheteD>

<BloqueClave> ::= <CuerpoClave> <BloqueClave>

| <CuerpoClave>

<CuerpoClave> ::= <tk_llaveI> <tk_cadena><MasClaves> <tk_llaveD>

<MasClaves> ::= <tk_coma> <tk_cadena> <MasClaves>

| <tk_cadena>

<Registros> ::= <tk_Registro> <tk_igual> <tk_corcheteI> <BloqueRegistro> <tk_corcheteD>

<BloqueRegistro> ::= <CuerpoRegistro> <BloqueRegistro>

| <CuerpoRegistro>

<CuerpoRegistro> ::= <tk_llaveI> <tk_cadena> <MasRegistros> <tk_llaveD>

| <tk_llaveI> <tk_numero> <MasRegistros> <tk_llaveD>

<MasRegistros> ::= <tk_coma> <tk_cadena> <MasRegistros>

| <tk_coma> <tk_numero> <MasRegistros>

| <tk_cadena>

| <tk_numero>

<Instrucciones> :== <tk_imprimir> <tk_parentesisI> <tk_cadena> <tk_parentesisD> <tk_puntoycoma>

| <tk_imprimirln> <tk_parentesisI> <tk_cadena> <tk_parentesisD> <tk_puntoycoma>

| <tk_conteo> <tk_parentesisI> <tk_parentesisD> <tk_puntoycoma>

| <tk_promedio> <tk_parentesisI> <tk_cadena> <tk_parentesisD> <tk_puntoycoma>

| <tk_contarsi> <tk_parentesisI> <tk_cadena> <tk_coma> <tk_numero> <tk_parentesisD> <tk_puntoycoma>

| <tk_datos> <tk_parentesisI> <tk_parentesisD> <tk_puntoycoma>

| <tk_sumar> <tk_parentesisI> <tk_cadena> <tk_parentesisD> <tk_puntoycoma>

| <tk_max> <tk_parentesisI> <tk_cadena> <tk_parentesisD> <tk_puntoycoma>

| <tk_min> <tk_parentesisI> <tk_cadena> <tk_parentesisD> <tk_puntoycoma>

| <tk_exportarReporte> <tk_parentesisI> <tk_cadena> <tk_parentesisD> <tk_puntoycoma>