Análisis Léxico

Para el análisis léxico, se utilizó un algoritmo el cual se encarga de recorrer carácter por carácter el código ingresado, manejando estados por medio de un switch se logra interpretar cada palabra como un respectivo token, así como cada símbolo. Para poder identificar a que lexema pertenece cada palabra de manera más sencilla, se hizo uso de expresiones regulares las cuales se listan a continuación:

- Comentario de una linea
- Comentario multilínea
- Identificador
- Cadena
- Carácter
- Entero
- Html
- Boleano
- Doble

El uso de palabras reservadas también hizo que el análisis fuera mas efectivo, dichas palabras se listan a continuación:

- Void
- Main
- If
- Else
- Case
- Default
- For
- While
- Do
- Console
- Write
- Int
- Double
- Char
- Bool
- String
- Return
- Break
- Continue

Para poder almacenar datos de manera temporal, se utilizaron arreglos. Los arreglos almacenan tokens, errores y comentarios.

Los tokens son generados por una clase Token la cual cuenta con un constructor solicitando la siguiente información:

- Nombre
- Lexema
- Linea
- Columna

Los errores son generados por una clase Error la cual cuenta con un constructor solicitando la siguiente información:

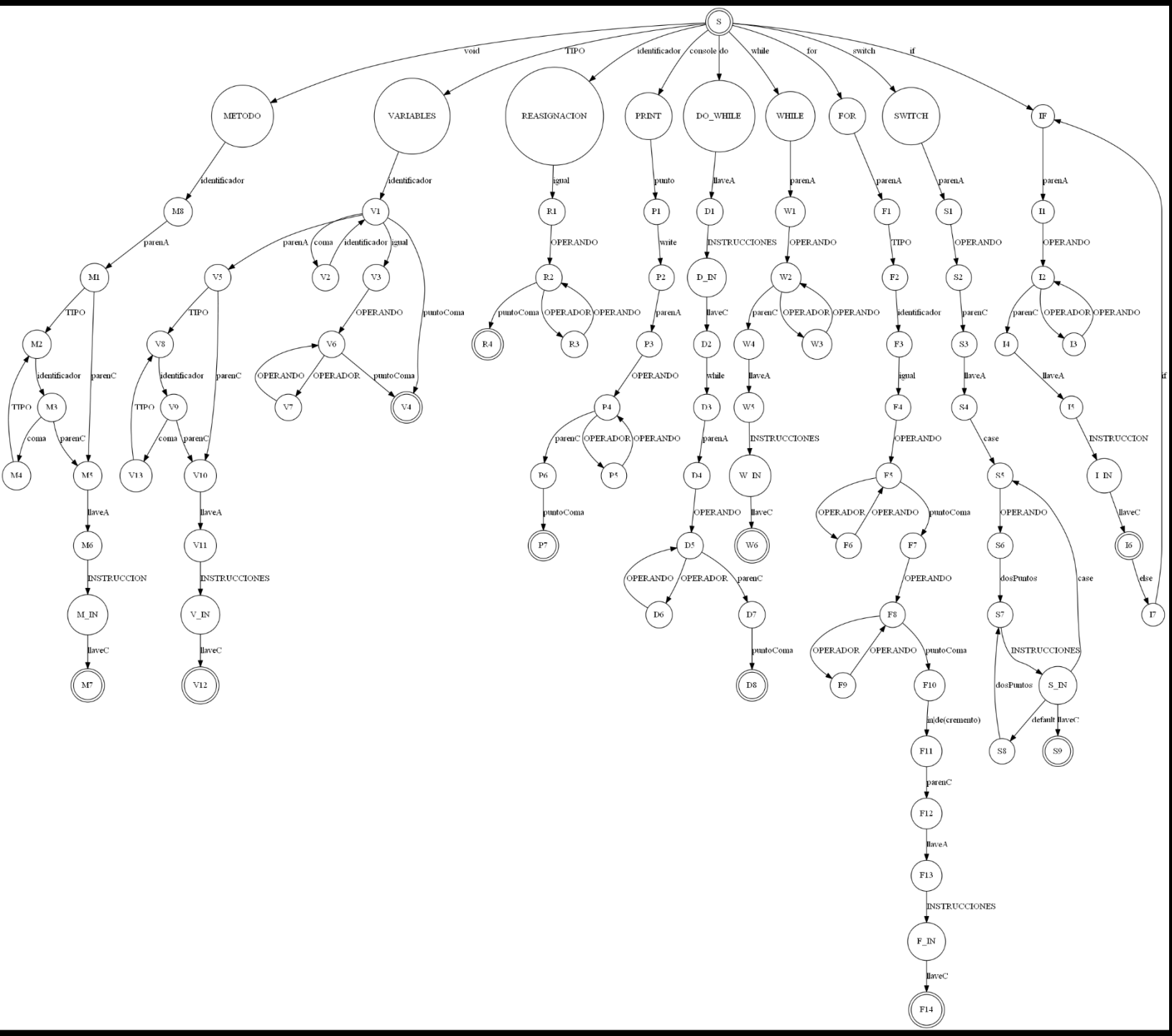
- Tipo
- Lexema
- Linea
- Columna
- Mensaje

Análisis Sintáctico

Para el análisis sintáctico se definieron arreglos en los cuales se almacenaban tipos, operandos y operadores.

Se hace uso de una clase NodoArbol para almacenar todos los tokens de manera ordenada y con lógica, esto da paso a la utilización de un AST para poder generar una traducción correcta y en el mismo orden.

Para poder analizar de manera sintáctica, se utilizo el siguiente esquema el cual consiste en la parte lógica que requiere cada instrucción que se podrá utilizar en el código a ingresar.



El análisis sintáctico inicia recorriendo la cola de tokens generada en el análisis léxico, al momento de tener una ocurrencia de cualquier carácter de inicio perteneciente al esquema anterior, se muda a una función específico para operar dicha instrucción. Cada función devuelve la posición en la que se encontró el final de la instrucción, así como un nodo con toda la información recopilada dentro de la instrucción. Este análisis devuelve la raíz del AST para poder realizar su traducción.

Traducción Python

Este método recibe como parámetro un nodo padre y un texto de tabulación, la razón de recibir una tabulación es porque en Python no se manejan las funciones como if, switch, While, entre otros, ya que, para estas funciones, cada instrucción que contiene viene dentro de un par de llaves, pero en Python se debe manejar por medio de una tabulación. El método de traducción devuelve una cadena en la cual se encuentra la porción de código traducida.

La traducción funciona al recorrer cada nodo de el AST y buscar ocurrencias con palabras reservadas.