

### **Parser.py:**

- *Es especificado por los atributos y las acciones semánticas*
- *no terminales pueden tener cualquier valor del output mientras los terminales no tienen valores explícitos, mientras los tokens que fueron tomados por el scanner tienen la información para ver los atributos.*
- *las acciones semánticas tienen declaraciones escritas en el lenguaje que fue generado el parser.*
- *los lookahead sirven si la gramática no es LL*
- la estructura general del parser es la siguiente:
  - scanner , error de la corriente, último token reconocido, lookahead token, parser, pasear y errores semánticos
- Pragmas - manejo de tokens que pueden ocurrir en cualquier área del input stream.
- obtener posición del parser
- sincronizar errores
- buscar errores semánticos
- string del look ahead
- get maneja los tokens con los lookahead tokens llamando el scanner y revisando que el lookahead token sea menor que el maxT del parser
- reglas semánticas: detectan un patrón lingüístico, siendo un lenguaje simplificado.

### **ManageProductions.py**

Separa terminales y no terminales y pone los no terminales en la izquierda para crear la gramática.

**First:** Sí el primer símbolo del espacio de trabajo es un terminal lo compara con el siguiente token de entrada y si el primer símbolo del espacio de trabajo es un no terminal predice con qué producción reemplazar ese terminal por lo que se basa en la terminal actual y el siguiente token que producción usar.

**Follow:** hace que un no terminal se desvanezca si es necesario para generar la cadena del árbol de análisis.

Para que se utilizan? Estos se utilizaron para que se pueda aplicar la regla necesaria en la posición correcta.

**CreateNewFunctions:** crea las funciones para cada elemento de la producción en el nuevo parser.