Fase 2 del Compilador

Objetivo: Se deberán de tener implementado el mecanismo de análisis sintáctico que será capaz de identificar una secuencia de elementos de lenguaje y determinara si corresponden a una gramática valida de lenguaje. Es indispensable que el ambiente integral de desarrollo (IDE) se encuentre terminado ya que una de las pestañas del IDE mostrara la información relativa al análisis sintáctico. Recuerde que el compilador debe ser capaz de trabajar de manera independiente al IDE

• Análisis Sintáctico:

A continuación, se describe un lenguaje de programación denominado Tiny extended. La especificación del lenguaje se proporciona mediante una descripción en BNF de cada construcción del lenguaje.

Recuerde que deberá leer un archivo (programa) de entra desde la línea de comandos y generar como salida dos archivos, el primero con la información de identificación de cada uno de los tokens del lenguaje, y el segundo contendrá toda la información referente al árbol sintáctico. Es importante recordar que debido a la importancia del árbol sintáctico es indispensable presentarlo de manera gráfica para asegurar su correcta construcción.

La sintaxis del lenguaje deberá ser implementado mediante un algoritmo descendente recursiva.

```
programa → lista-declaración
lista-declaración → lista-declaración declaración | declaración
declaración → declaración-variable | lista-sentencias
declaración-variable → tipo identificador ; |
tipo → int | real | void
lista-sentencias → lista-sentencia sentencia | vació
sentencia → selección | iteración | repetición | sent-in | sent-out |
          asignación
asignación → := sent-expresión
sent-expresión → expresión; | ;
selección → if expresión sentencia end |
          if expresión sentencia else sentencia end
iteración → while expresión sentencia end
repetición → do sentencia until expresión
sent-in → cin identificador ;
sent-out → cout expresión ;
expresión → expresión-simple relación-op expresión-simple |
     expresión-simple
relacion-op → <= | < | > | >= | == | !=
expresión-simple → expresión-simple suma-op termino | termino
suma-op → + | -
termino → termino mult-op factor | factor
mult-op → * | /
factor → ( expresión ) | numero | identificador
```

Dentro de los operadores, no olvide agregar el ++, -- , %(modulo)

Recuerde que para la entrega deber estar su árbol expandido