**Estado Actual Proyecto**

Analizador Lexico Completo es lo que se tiene hasta el momento:

/\*\*

\* JavaCC template file created by SF JavaCC plugin 1.5.28+ wizard for JavaCC 1.5.0+

\*/

**options**

**{**

**static** = **true**;

**}**

**PARSER\_BEGIN(Analizador)**

**package** co.edu.uniquindio.compiladores.frontend.lexico;

**public** **class** Analizador

{

**static** **int** numero = 0;

**private** **static** **void** grabarLexema (**int** n, String token, String lexema, **int** nL, **int** nC) {

System.out.println(n + ") Linea:\t"+nL+"\tColumna:\t"+nC+"\tLexema:\t\""+lexema+"\"\t"+"Token:\t"+token);

}

**private** **static** **void** grabarDatosPieza(**int** n, String token, Token pieza) {

grabarLexema(n, token, pieza.image, pieza.beginLine, pieza.beginColumn );

}

**public** **static** **void** main(String args []) **throws** ParseException

{

**try**

{

System.out.println("Ingrese el código a analizar:");

Analizador parser = **new** Analizador(System.in);

parser.TokenList();

System.out.println("Analisis terminado:");

System.out.println("No se han hallado errores léxicos");

}**catch**(TokenMgrError te)

{

System.out.println("Se han encontrado errores lexicos.");

System.out.println(te.getMessage());

}

**catch**(ParseException e)

{

System.out.println("Analizador: Se han encontrado errores en el analisis.");

System.out.println(e.getMessage());

}

}

}

**PARSER\_END(Analizador)**

/\* Espacio en blanco, saltos de linea, tabulares \*/

**SKIP** :

**{**

" "

**|** "\t"

**|** "\n"

**|** "\r"

**|** "\f"

**}**

**TOKEN** : /\*PALABRA RESERVADA LEER\*/

**{**

**<** PALABRA\_RESERVADA\_LEER: "Leer" **>**

**}**

**TOKEN** : /\*PALABRA RESERVADA ESCRIBIR\*/

**{**

**<** PALABRA\_RESERVADA\_ESCRIBIR: "Escribir" **>**

**}**

**TOKEN** : /\*PALABRA RESERVADA SI\*/

**{**

**<** PALABRA\_RESERVADA\_SI: "Si" **>**

**}**

**TOKEN** : /\*PALABRA RESERVADA SINO\*/

**{**

**<** PALABRA\_RESERVADA\_SINO: "Sino" **>**

**}**

**TOKEN** : /\*PALABRA RESERVADA FINSI\*/

**{**

**<** PALABRA\_RESERVADA\_FINSI: "Finsi" **>**

**}**

**TOKEN** : /\*PALABRA RESERVADA ENTONCES\*/

**{**

**<** PALABRA\_RESERVADA\_ENTONCES: "Entonces" **>**

**}**

**TOKEN** : /\*PALABRA RESERVADA MIENTRAS\*/

**{**

**<** PALABRA\_RESERVADA\_MIENTRAS: "Mientras" **>**

**}**

**TOKEN** : /\*PALABRA RESERVADA RETORNAR\*/

**{**

**<** PALABRA\_RESERVADA\_RETORNAR: "Retornar" **>**

**}**

**TOKEN** : /\*PALABRA RESERVADA ENTERO\*/

**{**

**<** PALABRA\_RESERVADA\_ENTERO: "Entero" **>**

**}**

**TOKEN** : /\*PALABRA RESERVADA LOGICO\*/

**{**

**<** PALABRA\_RESERVADA\_LOGICO: "Logico" **>**

**}**

**TOKEN** : /\*PALABRA RESERVADA CADENA\*/

**{**

**<** PALABRA\_RESERVADA\_CADENA: "Cadena" **>**

**}**

**TOKEN** : /\*PALABRA RESERVADA IMPRIMIR\*/

**{**

**<** PALABRA\_RESERVADA\_IMPRIMIR: "Imprimir" **>**

**}**

**TOKEN** : /\* ASIGNACION\*/

**{**

**<** ASIGNACION : "=" **>**

**}**

**TOKEN** : /\*NUMEROS EJEMPLO 1, 1234, 987\*/

**{**

**<** ENTERO : **(**< DIGIT >**)+** **|** "-"**(**< DIGIT >**)+** **>**

**|** **<** #DIGIT : **[** "0"**-**"9" **]** **>**

**}**

**TOKEN** : /\*CONCATENAR\*/

**{**

**<** CONCATENAR : "@" **>**

**}**

**TOKEN** : /\* SEPARADOR\*/

**{**

**<** SEPARADOR : "," **>**

**}**

**TOKEN** : /\*IDENTIFICADORES EJEMPLO a, casa, clase1\*/

**{**

**<** IDENTIFICADOR :**[**"a"**-**"z","A"**-**"Z"**]** **(** **[**"a"**-**"z","A"**-**"Z","0"**-**"9","\_"**]** **)\*** **>**

**}**

**TOKEN** : /\* FIN DE LINEA\*/

**{**

**<** FIN\_LINEA : "." **>**

**}**

**TOKEN** : /\* ABRIR CORCHETE\*/

**{**

**<** ABRIR\_CORCHETE : "[" **>**

**}**

**TOKEN** : /\* CERRAR CORCHETE\*/

**{**

**<** CERRAR\_CORCHETE : "]" **>**

**}**

**TOKEN** : /\* ABRIR LLAVE\*/

**{**

**<** ABRIR\_LLAVE : "{" **>**

**}**

**TOKEN** : /\* CERRAR LLAVE\*/

**{**

**<** CERRAR\_LLAVE : "}" **>**

**}**

**TOKEN** : /\* OPERATORS SUSTRACCION\*/

**{**

**<** SUMA : "+" **>**

**|** **<** MULTIPLICACION : "\*" **>**

**|** **<** MODULO : "%" **>**

**|** **<** SUSTRACCION : "-" **>**

**|** **<** DIVISION : "/" **>**

**}**

**TOKEN** : /\*LOGICOS\*/

**{**

**<** LOGICOS :"&"**|**"|" **>**

**}**

**TOKEN** : /\*RELACIONALES\*/

**{**

**<** RELACIONALES :"<"**|**">"**|**" >="**|**"<="**|**"<>"**|**"=="**|**"!=" **>**

**}**

**TOKEN** : /\*COMENTARIO\*/

**{**

**<** COMENTARIO :"##"**(~[**"#"**])\***"##" **>**

**}**

**TOKEN** : /\*CADENA\*/

**{**

**<** CADENA :"\""**(~[**"\""**])\***"\"" **>**

**}**

/\*\*

\* Reconoce el contenido completo de un archivo

\*

\* TokenList -> (TokenElement)\*

\*/

**void** TokenList():

**{}**

**{**

**(** TokenElement() **)\***

**}**

**void** TokenElement():

**{**Token pieza; **}**

**{**

pieza = <PALABRA\_RESERVADA\_LEER> **{** numero++; grabarDatosPieza(numero, "PALABRA RESERVADA LEER", pieza); **}**

**|** pieza = <PALABRA\_RESERVADA\_ESCRIBIR> **{** numero++; grabarDatosPieza(numero, "PALABRA RESERVADA ESCRIBIR", pieza); **}**

**|** pieza = <PALABRA\_RESERVADA\_SI> **{** numero++; grabarDatosPieza(numero, "PALABRA RESERVADA SI", pieza); **}**

**|** pieza = <PALABRA\_RESERVADA\_SINO> **{** numero++; grabarDatosPieza(numero, "PALABRA RESERVADA SINO", pieza); **}**

**|** pieza = <PALABRA\_RESERVADA\_FINSI> **{** numero++; grabarDatosPieza(numero, "PALABRA RESERVADA FINSI", pieza); **}**

**|** pieza = <PALABRA\_RESERVADA\_ENTONCES> **{** numero++; grabarDatosPieza(numero, "PALABRA RESERVADA ENTONCES", pieza); **}**

**|** pieza = <PALABRA\_RESERVADA\_MIENTRAS> **{** numero++; grabarDatosPieza(numero, "PALABRA RESERVADA MIENTRAS", pieza); **}**

**|** pieza = <PALABRA\_RESERVADA\_RETORNAR> **{** numero++; grabarDatosPieza(numero, "PALABRA RESERVADA RETORNAR", pieza); **}**

**|** pieza = <PALABRA\_RESERVADA\_ENTERO> **{** numero++; grabarDatosPieza(numero, "PALABRA RESERVADA ENTERO", pieza); **}**

**|** pieza = <PALABRA\_RESERVADA\_LOGICO> **{** numero++; grabarDatosPieza(numero, "PALABRA RESERVADA LOGICO", pieza); **}**

**|** pieza = <PALABRA\_RESERVADA\_CADENA> **{** numero++; grabarDatosPieza(numero, "PALABRA RESERVADA CADENA", pieza); **}**

**|** pieza = <PALABRA\_RESERVADA\_IMPRIMIR> **{** numero++; grabarDatosPieza(numero, "PALABRA RESERVADA IMPRIMIR", pieza); **}**

**|** pieza = <ASIGNACION> **{** numero++; grabarDatosPieza(numero, "ASIGNACION", pieza); **}**

**|** pieza = <ENTERO> **{** numero++; grabarDatosPieza(numero, "ENTERO", pieza); **}**

**|** pieza = <CONCATENAR> **{** numero++; grabarDatosPieza(numero, "CONCATENAR", pieza); **}**

**|** pieza = <SEPARADOR> **{** numero++; grabarDatosPieza(numero, "SEPARADOR", pieza); **}**

**|** pieza = <IDENTIFICADOR> **{** numero++; grabarDatosPieza(numero, "IDENTIFICADOR", pieza); **}**

**|** pieza = <FIN\_LINEA> **{** numero++; grabarDatosPieza(numero, "FIN LINEA", pieza); **}**

**|** pieza = <ABRIR\_CORCHETE> **{** numero++; grabarDatosPieza(numero, "ABRIR CORCHETE", pieza); **}**

**|** pieza = <CERRAR\_CORCHETE> **{** numero++; grabarDatosPieza(numero, "CERRAR CORCHETE", pieza); **}**

**|** pieza = <ABRIR\_LLAVE> **{** numero++; grabarDatosPieza(numero, "ABRIR LLAVE", pieza); **}**

**|** pieza = <CERRAR\_LLAVE> **{** numero++; grabarDatosPieza(numero, "CERRAR LLAVE", pieza); **}**

**|** pieza = <SUMA> **{** numero++; grabarDatosPieza(numero, "SUMA" , pieza); **}**

**|** pieza = <MULTIPLICACION> **{** numero++; grabarDatosPieza(numero, "MULTIPLICACION", pieza); **}**

**|** pieza = <MODULO> **{** numero++; grabarDatosPieza(numero, "MODULO", pieza); **}**

**|** pieza = <SUSTRACCION> **{** numero++; grabarDatosPieza(numero, "SUSTRACCION", pieza); **}**

**|** pieza = <DIVISION> **{** numero++; grabarDatosPieza(numero, "DIVISION", pieza); **}**

**|** pieza = <LOGICOS> **{** numero++; grabarDatosPieza(numero, "LOGICO", pieza); **}**

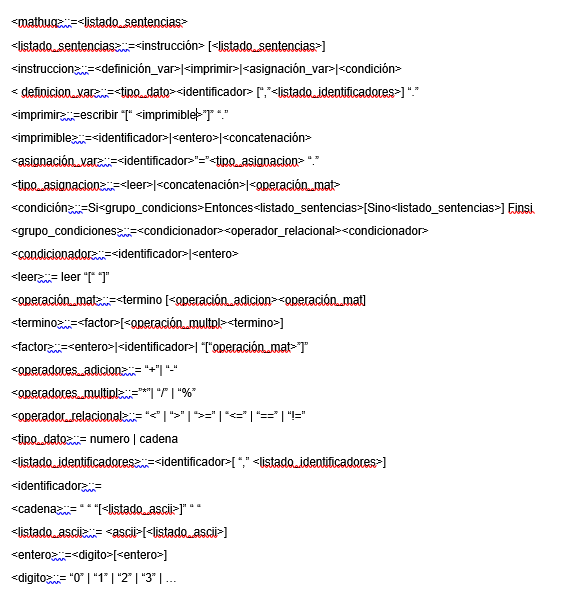
**|** pieza = <RELACIONALES> **{** numero++; grabarDatosPieza(numero, "RELACIONALES", pieza); **}**

**|** pieza = <COMENTARIO> **{** numero++; grabarDatosPieza(numero, "COMENTARIO", pieza); **}**

**|** pieza = <CADENA> **{** numero++; grabarDatosPieza(numero, "CADENA", pieza); **}**

**}**

BNF

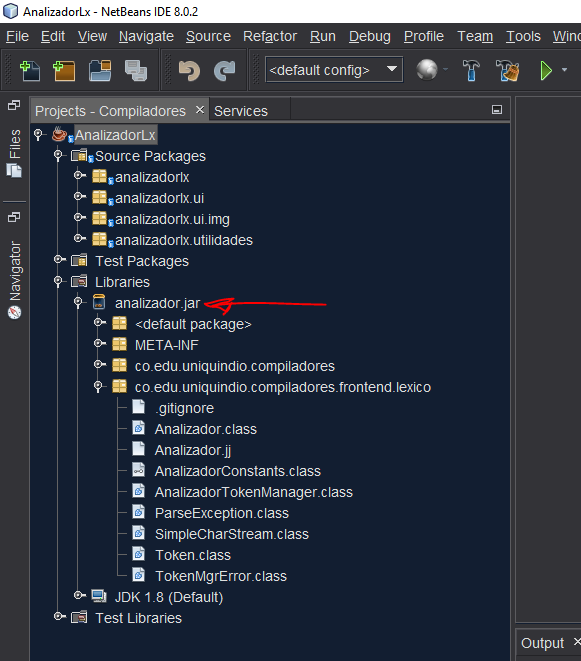


**Planeado para realizar en Clase**

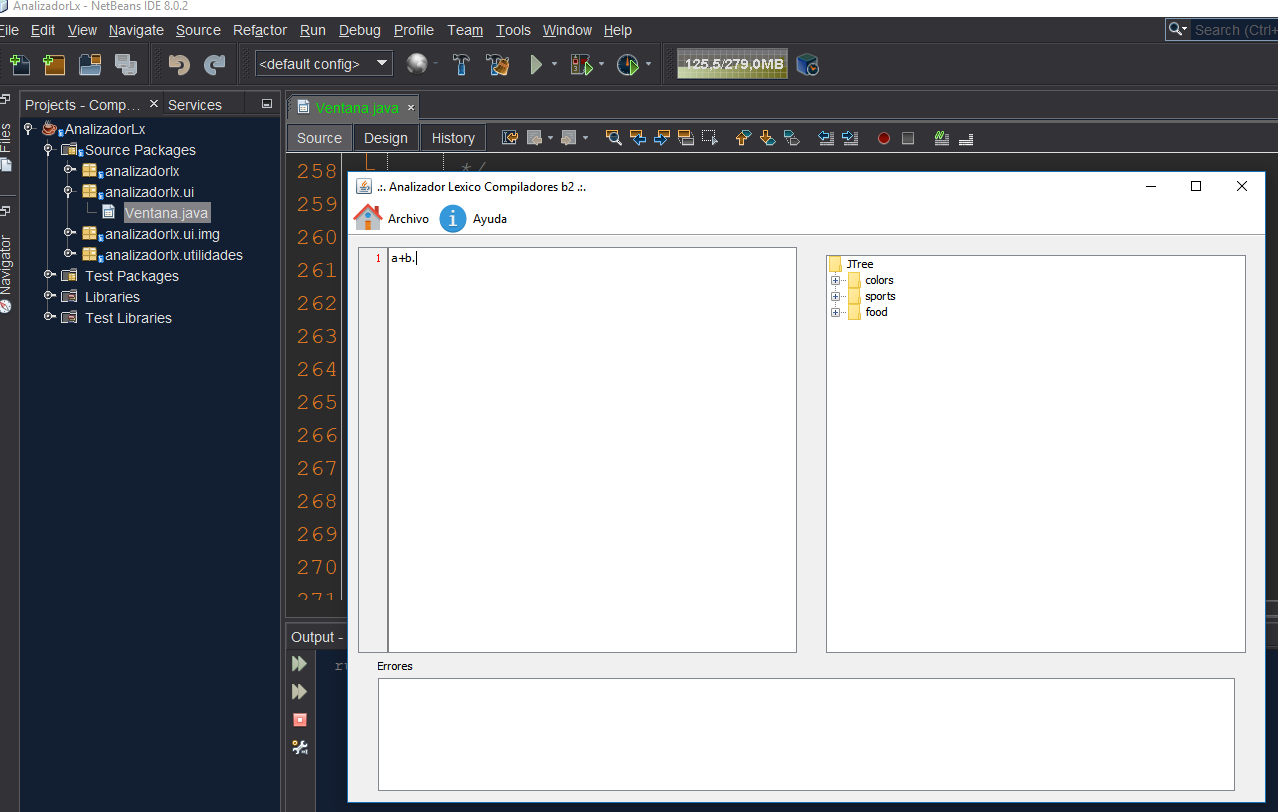
* Investigación Manejo y Recuperación de Errores
* Implementar el Analizador Sintáctico.
* Implementar el Árbol de Derivación.
* Investigación de integración javaCC como librería en Netbeans.
* Inicio de Implementación Interfaz Gráfica.

**Desarrollado de acuerdo a plazo 23:59**

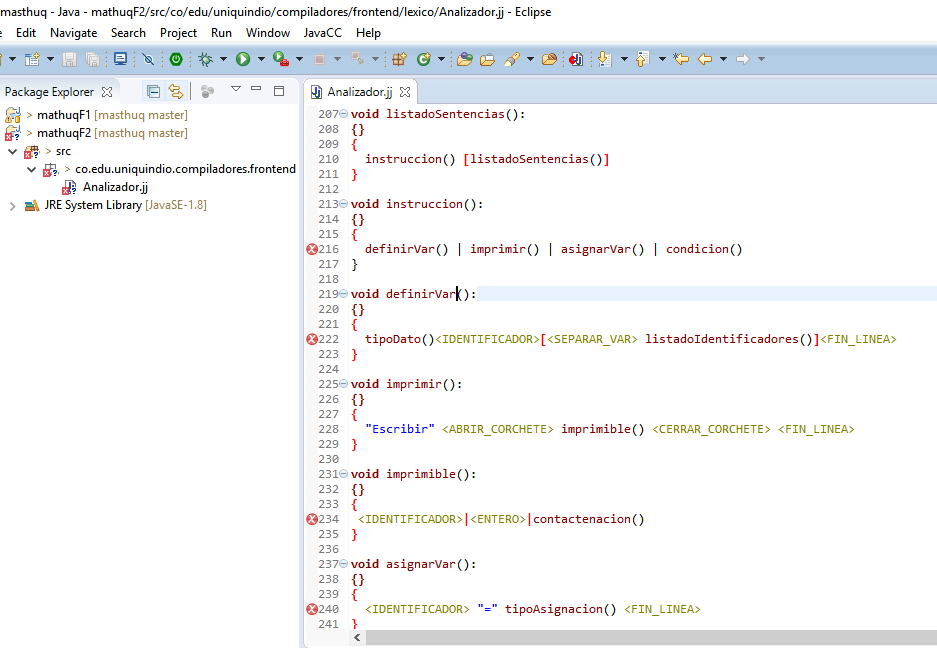
* **Investigación de integración javaCC como librería en Netbeans**: Se logra portar como librería el trabajo realizado con JavaCC de Eclipse a Netbeans para de esta forma agilizar el diseño de la interfaz.



* **Inicio de Implementación Interfaz Gráfica**: Se genera la primera versión de la interfaz gráfica donde se poder ver funcionando correctamente la lectura de los lexemas y la comprobación de los tokens.



* **Implementar el Analizador Sintáctico**: Se inicia transcripción del papel a JavaCC del BNF; no se completa todo, pero se avanza.



* **Investigación Manejo y Recuperación de Errores**: Se consulta en Internet sobre recuperación de errores y se valida lo investigado con la profesora determinando un camino de cómo hacerlo.

