Analizador Sintáctico

Materia: Compiladores

Integrantes:

- Juan Pablo Enríquez Pedroza
- Ulises Gallardo Rodríguez
- Sara Carolina Gómez Delgado
- Luis Eduardo Robles Jiménez

a. **Las palabras reservadas que corresponda al lenguaje y sus respectivos Tokens, ya con los ajustes realizados.**

Lexema	Token
var	Var
int	Int
float	Float
bool	Bool
string	String
if	If
else	Else
\n	Linea
=	Igual
+	Suma
-	Resta
%	Modulo
*	Multiplicación
/	Division
(	Parentesis_a
)	Parentecis_c
{	Llave_a
}	Llave_c
[	Corchete_a
]	Corchete_c
++, --	Op_incremento
==, !=, <, <=, >, >=	Op_relacional
&&, ||	Op_logico
!	Op_logico_negacion
+, -, *,/ ,%	Op_aritmetico

+=	Op_asignacion
:=	Op_inferencia
<<, >>, &, |, ^, &^	Op_bits
func	Func
return	Return
for	For
continue	Continue
range	Range
switch	Switch
case	Case
default	Default
break	Break
“[^"]*”	Cadena
“	Comillas
;	Punto_c
{L}({L}|{D})*	Identificador
("(-"{D}+")")|{D}+	Numero
("(-"{D}*"."{D}*")")|({D}*"."{D}*)	Real
int, uint, uint8, float, float32, complex64, string, bool, byte	T_dato
,	Coma
…	Tres_puntos
break, case, chan, const, continue, default, defer, else, fallthrough, for, func, go, goto, if, import, interface, map, package, range, return, select, struct, switch, type, var, return, else	Reservadas
Error	Error

b. Las reglas de sintaxis, incluye terminales y no terminales.

```java
terminal Func, Var, If, Else, Linea, Igual, Suma, Resta, Modulo, Multiplicacion, Division, Parentesis_a, Parentesis_c,Llave_a, Llave_c, Corchete_a, Corchete_c, Op_booleano, Op_incremento, Op_relacional, Op_logico, Op_logico_negacion, Op_aritmetico, Op_asignacion, Op_atribucion, Op_inferencia, Op_bits, Return, Others, For, Continue, Range, Switch, Case, Default, Break, Cadena, Comillas, Main, Identificador, Numero, Real, T_dato, Coma, Punto_c,Tres_puntos, Reservadas,Error;
```

```java
non terminal INICIO, SENTENCIA, DECLARACION, DECLARACION_VARIABLE, DECLARACION_ARREGLO, INICIALIZACION_ARREGLO, LISTA_EXPRESION, EXPRESION, DECLARACION_FUNCION, NOMBRE_FUNCION, SENTENCIA_FUNCION, CUERPO_FUNCION, LISTA_FUNCIONES, LISTA_PARAMETROS, PARAMETROS, DECLARACION_PARAMETRO, RESULTADO, LISTA_T_dato, LLAMAR_FUNCION,SENTENCIA_ASIGNACION, BLOQUE, RESULTADO_RETURN, BLOQUE_RETURN, CUERPO_FUNCION_RETURN,SENTENCIA_FUNCION_RETURN, POSIBLE_PARAMETROS, PASAR_LISTA_PARAMETROS,DECLARACION_FOR, IF, IF_ELSE, FOR, VALOR, SENTENCIA_BOOLEANA,SENTENCIA_COMPARACION, SENTENCIA_ARITMETICA, SENTENCIA_FOR;
```

```java
start with INICIO;

INICIO ::=
        DECLARACION_FUNCION Func Main Parentesis_a Parentesis_c Llave_a SENTENCIA Llave_c |
    Func Main Parentesis_a Parentesis_c Llave_a SENTENCIA Llave_c
;

LISTA_FUNCIONES ::=
    DECLARACION_FUNCION |
    DECLARACION_FUNCION LISTA_FUNCIONES
;

SENTENCIA ::=
    SENTENCIA DECLARACION |
    DECLARACION |
```

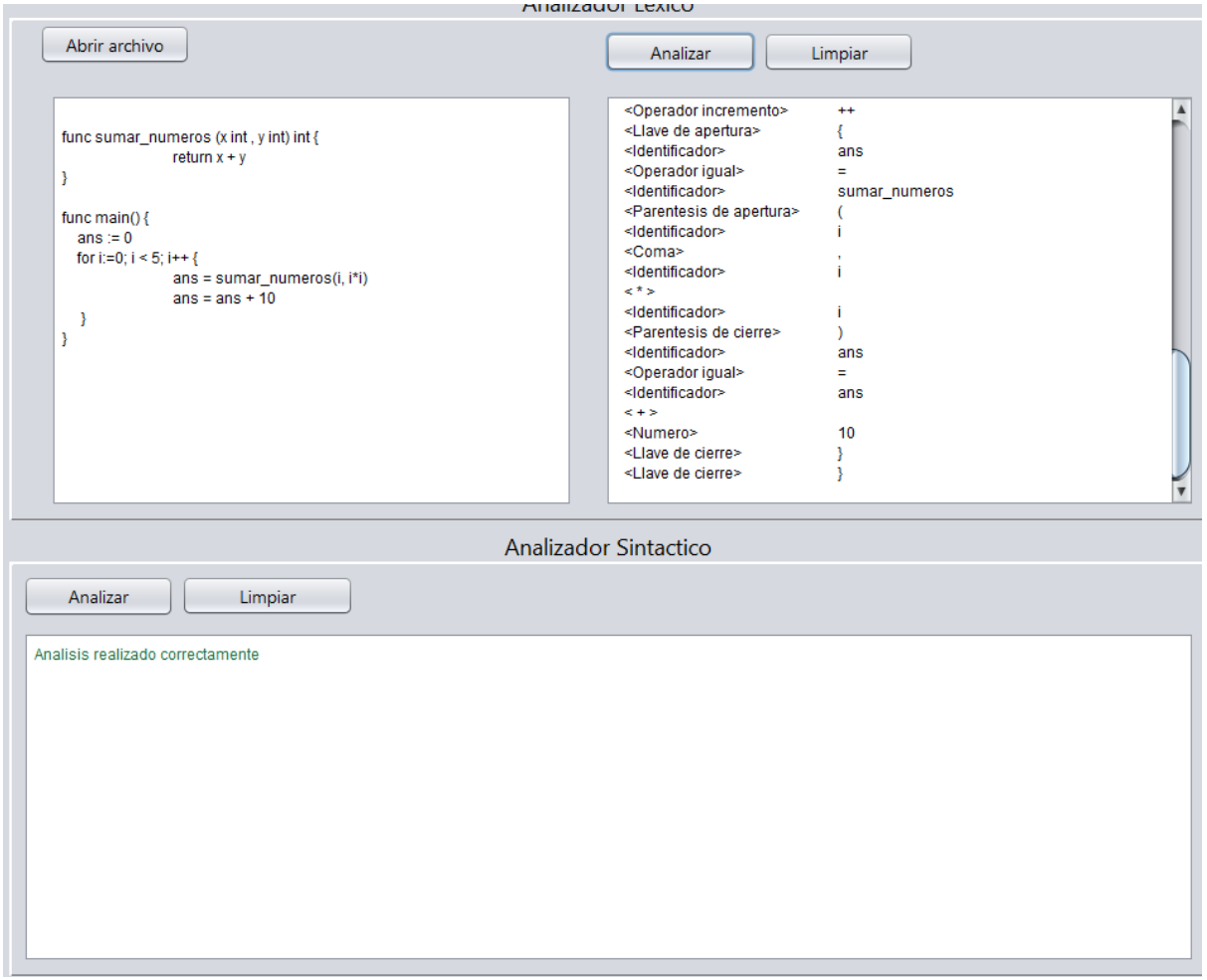
```
    SENTENCIA IF |
    IF |
    SENTENCIA IF_ELSE |
    IF_ELSE |
    SENTENCIA FOR |
    FOR |
    SENTENCIA_ASIGNACION |
    SENTENCIA SENTENCIA_ASIGNACION
;

DECLARACION ::=
    DECLARACION_VARIABLE |
    DECLARACION_FUNCION
;

DECLARACION_VARIABLE ::=
    Var Identificador |
    Var Identificador T_dato |
    Var Identificador T_dato Igual LLAMAR_FUNCION |
    Var Identificador T_dato Igual Numero |
    Var Identificador T_dato Igual Real |
    Var Identificador T_dato Igual Op_booleano |
    Var Identificador T_dato Igual Cadena |
    Identificador Op_inferencia Numero |
    Identificador Op_inferencia Real |
    Identificador Op_inferencia Op_booleano |
    Identificador Op_inferencia Cadena |
    Identificador Op_inferencia LLAMAR_FUNCION |
    Var Identificador Igual DECLARACION_ARREGLO |
    Identificador Op_inferencia DECLARACION_ARREGLO
;



DECLARACION_ARREGLO ::=
    Corchete_a Numero Corchete_c T_dato  INICIALIZACION_ARREGLO |
    Corchete_a Tres_puntos Corchete_c T_dato INICIALIZACION_ARREGLO |
    Corchete_a Corchete_c T_dato INICIALIZACION_ARREGLO
;

INICIALIZACION_ARREGLO ::= Llave_a LISTA_EXPRESION Llave_c;

LISTA_EXPRESION ::=
    EXPRESION |
    EXPRESION Coma LISTA_EXPRESION
;

EXPRESION ::=
```

```
    Numero |
    Real |
    Op_booleano |
    Cadena
;

IF ::= If SENTENCIA_BOOLEANA Llave_a SENTENCIA Llave_c
;

SENTENCIA_BOOLEANA ::=
    Op_booleano |
    SENTENCIA_COMPARACION |
    SENTENCIA_ARITMETICA |
    Op_booleano Op_logico SENTENCIA_BOOLEANA |
    SENTENCIA_COMPARACION Op_logico SENTENCIA_BOOLEANA |
    SENTENCIA_ARITMETICA Op_logico SENTENCIA_BOOLEANA |
    Op_logico_negacion Parentesis_a SENTENCIA_BOOLEANA Parentesis_c
;

SENTENCIA_COMPARACION ::=
    Identificador Op_relacional Op_booleano |
    Identificador Op_relacional Numero |
    Identificador Op_relacional Real |
    Identificador Op_relacional Identificador |
    Identificador Op_relacional Cadena
;

SENTENCIA_ARITMETICA ::=
    Identificador Op_aritmetico Identificador |
    Identificador Op_aritmetico Numero |
    Identificador Op_aritmetico Real |
    Identificador Op_aritmetico Cadena |
    Identificador Op_bits Numero
;

SENTENCIA_ASIGNACION ::=
    Identificador Igual LLAMAR_FUNCION |
    Identificador Igual Numero |
    Identificador Igual Real |
    Identificador Igual Op_booleano |
    Identificador Igual Cadena |
    Identificador Igual SENTENCIA_ARITMETICA
;

IF_ELSE  ::=  If  SENTENCIA_BOOLEANA  Llave_a  SENTENCIA  Llave_c  Else  Llave_a
SENTENCIA Llave_c
;
```

```
FOR ::= For SENTENCIA_FOR Llave_a SENTENCIA Llave_c
;

SENTENCIA_FOR ::=
      Identificador  Op_inferencia  VALOR  Punto_c  SENTENCIA_BOOLEANA  Punto_c
DECLARACION_FOR
;

VALOR ::=
    Numero |
    Real
;

DECLARACION_FOR ::=
    Identificador Op_atribucion Numero |
    Identificador Op_incremento
;


DECLARACION_FUNCION ::=
    Func NOMBRE_FUNCION SENTENCIA_FUNCION CUERPO_FUNCION |
    Func NOMBRE_FUNCION SENTENCIA_FUNCION_RETURN CUERPO_FUNCION_RETURN
;

LLAMAR_FUNCION ::=
    Identificador Parentesis_a PASAR_LISTA_PARAMETROS Parentesis_c |
    Identificador Parentesis_a Parentesis_c
;

NOMBRE_FUNCION ::= Identificador
;

SENTENCIA_FUNCION ::= LISTA_PARAMETROS
;

SENTENCIA_FUNCION_RETURN ::= LISTA_PARAMETROS RESULTADO
;

LISTA_PARAMETROS ::=
    Parentesis_a PARAMETROS Parentesis_c |
    Parentesis_a Parentesis_c
;

PARAMETROS ::=
    DECLARACION_PARAMETRO |
    DECLARACION_PARAMETRO Coma PARAMETROS
;
```

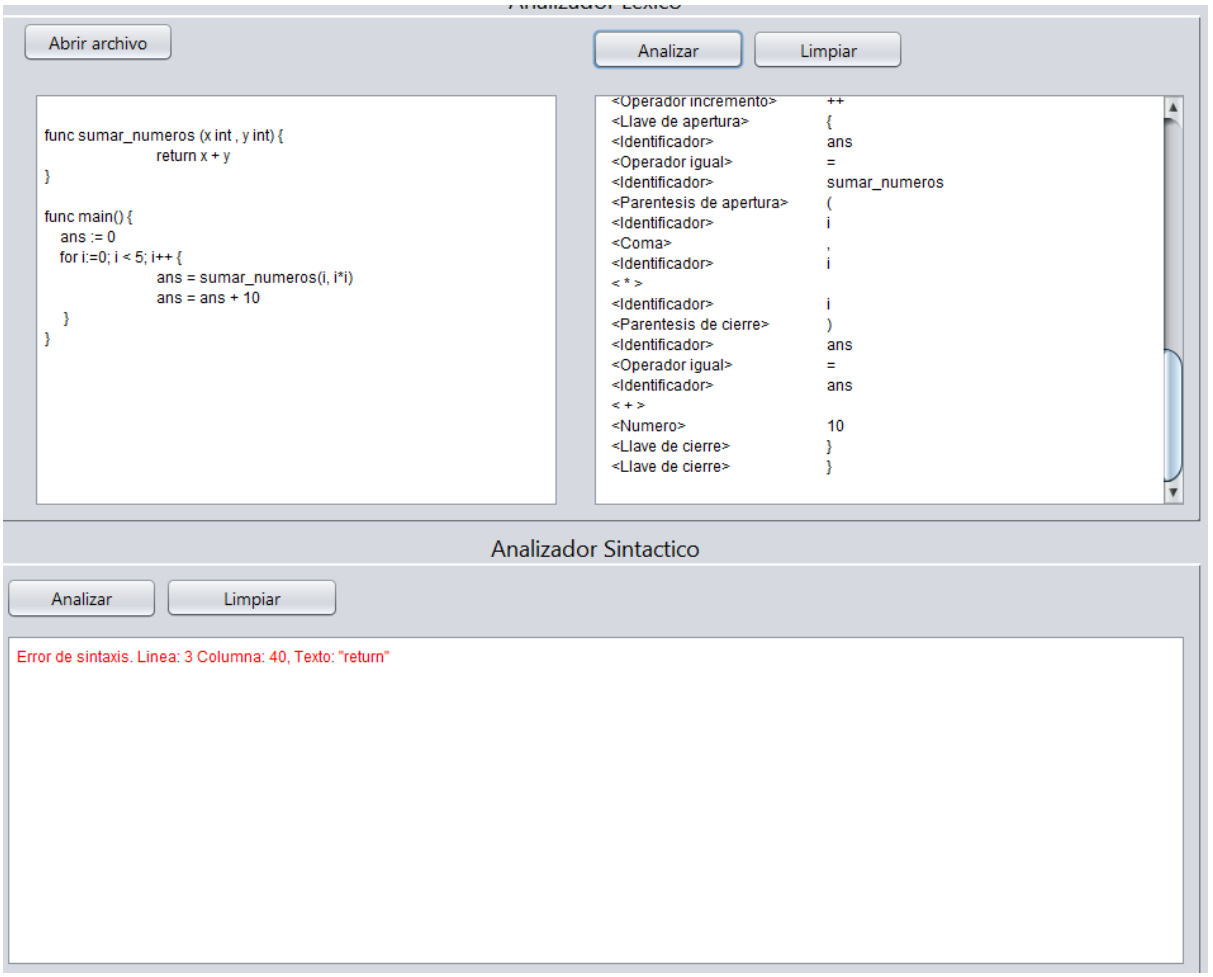
```
DECLARACION_PARAMETRO ::=
    Identificador T_dato |
    Identificador Tres_puntos T_dato
;

PASAR_LISTA_PARAMETROS ::=
    POSIBLE_PARAMETROS |
    POSIBLE_PARAMETROS Coma PASAR_LISTA_PARAMETROS
;

POSIBLE_PARAMETROS ::=
    Identificador |
    Numero |
    Cadena |
    SENTENCIA_BOOLEANA
;

RESULTADO ::=
    T_dato |
    Parentesis_a LISTA_T_dato Parentesis_c
;

LISTA_T_dato ::=
    T_dato |
    T_dato Coma LISTA_T_dato
;

CUERPO_FUNCION ::= BLOQUE
;

CUERPO_FUNCION_RETURN ::= BLOQUE_RETURN
;

BLOQUE ::= Llave_a SENTENCIA Llave_c
;

BLOQUE_RETURN ::=
    Llave_a SENTENCIA Return RESULTADO_RETURN Llave_c  |
    Llave_a Return RESULTADO_RETURN Llave_c
;

RESULTADO_RETURN ::=
    Numero |
    Real |
    Cadena |
    SENTENCIA_BOOLEANA |
    Identificador
;
```

c. Agrega dos screenshots con un código de mínimo 10 líneas que funcione y otro código también de 10 líneas que marque errores.



```java
func sumar_numeros (x int , y int) int {
      return x + y
}

func main() {
     ans := 0
     for i:=0; i < 5; i++ {
       ans = sumar_numeros(i, i*i)
       ans = ans + 10
      }
}
```

Código error:



```Java
func sumar_numeros (x int , y int) {
      return x + y
}

func main() {
    ans := 0
    for i:=0; i < 5; i++ {
      ans = sumar_numeros(i, i*i)
      ans = ans + 10
     }
}
```

d. Describe al menos 3 hallazgos que ubicaste durante la elaboración de esta etapa serán de utilidad para la exposición final.

- Entendimiento de BNF, sistema de archivos y RegEx para la generación de los analizadores léxico y sintácticos.
- Comprensión y análisis de GoLang; Notamos las grandes similitudes entre dos lenguajes altamente conocidos, C++ y GoLang.
- Aprendimos también que la tokenización de un lenguaje de programación es un trabajo arduo y extenso en tiempo.