

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMA

ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1

SECCIÓN A

AUX. MYNOR RENE RUIZ GUERRA



DANIEL ROLANDO SOTZ ALVARADO

201430496

1ER. SEMESTRE 2023

INTRODUCCION

El curso de Organización de Lenguajes y Compiladores 1, perteneciente a la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, ha quedado satisfecho con el programa EXREGAN (que previamente ha entregado), por lo que nuevamente se interesan en usted para generar el lenguaje TypeWise, que será un intérprete para que los estudiantes de Introducción a la Programación y Computación 1 utilicen para sus primeras prácticas.

Objetivo General

Aplicar los conocimientos sobre la fase de análisis léxico y sintáctico de un compilador para la construcción de un software que permita generar análisis por medio del método del árbol.

Objetivos Específicos:

- Reforzar los conocimientos de análisis léxico y sintáctico para la creación de un lenguaje de programación.
- Aplicar los conceptos de compiladores para implementar el proceso de interpretación de código de alto nivel.
- Aplicar los conceptos de compiladores para analizar un lenguaje de programación y producir las salidas esperadas.

Alcances del Proyecto

La intención del proyecto es que el encargado del desarrollo de esta aplicación aprenda a utilizar las herramientas que nos proveen la librería Jison, a la vez que se pone en práctica también los conceptos aprendidos en las clases magistrales de cursos anteriores como del curso actual organización de lenguajes y compiladores.

Especificación técnica

Los requerimientos mínimos para uso del programa son:

- Lenguajes implementados: Javascript, HTML, DOT
- Editor de Código: Visual Studio Code
- Sistema Operativo: Windows 10 Home (64 bits)
- Librerías implementadas: JISON
- Interfaz Gráfica: Node, Angular

Descripcion del Lenguaje

Este lenguaje permite las siguientes funcionalidades:

- Asignacion de variables
- Declaracion de variables
- Metodos con y sin parámetros
- Llamadas a metodos con y sin parámetros
- Imprimir en consola
- Flujo de control:
 - If
 - If Else
- Ciclos
 - While
 - Do While
 - For
 - Switch

Operaciones Permitidas

- Aritméticas
 - Suma
 - Resta
 - Multiplicacion
 - División
 - Potencia
 - Mod
- Relacionales
 - Mayor que
 - Menor que
 - Mayor o igual
 - Menor o igual
 - Idéntico
 - Diferente
- Lógicas
 - And
 - Or
 - Not
- Unarios
 - Incremento
 - Decremento

Tipos de Datos Soportados

- Enteros
- Decimales
- Booleanos
- Caracteres
- Cadenas

Configuración de Análisis Léxico

Definición y declaración algunos Tokens

```
"/".* // comentario simple
[/][*][^]*[*]+([/*][^]*[*]+)*[/] //comentario

"expression"    return 'tk_expresion';
"print"         return 'PRINT';
"toLowerCase"    return 'LOWER';
"toUpperCase"   return 'UPPER';
"length"        return 'LENGTH';
"truncate"      return 'TRUNCATE';
"round"         return 'ROUND';
"typeof"        return 'TYPEOF';
"toString"      return 'TOSTRING';
"toCharArray"   return 'CHARARRAY';
"exec"          return 'EXEC';
"new"           return 'NEW';

"int"           return 'RENTERO';
"double"        return 'RDOUBLE';
"boolean"       return 'RBOOLEANO';
"char"          return 'RCARACTER';
"string"        return 'RCADENA';

"\\n"           return 'SALTOLINEA';
"\\\\"          return 'BARINVERSA';
"\\'"          return 'COMILLASSIM';
"\\\""         return 'COMILLASDOB';
"\\t"          return 'TABULAR';
```

```
"while"         return 'WHILE';
"do"            return 'DO';
"if"            return 'IF';
"else"          return 'ELSE';
"for"           return 'FOR';
"switch"        return 'SWITCH';
"case"          return 'CASE';
"default"       return 'DEFAULT';

"break"         return 'BREAK';
"continue"      return 'CONTINUE';
"return"        return 'RETURN';

"void"          return 'VOID';
"main"          return 'MAIN';

":"            return 'DOSPTS';
","            return 'COMA';
";"            return 'PTCOMA';
"{"            return 'LLAVIZQ';
"}"            return 'LLAVDER';
"("            return 'PARIZQ';
")"            return 'PARDER';
"["            return 'CORIZQ';
"]"            return 'CORDER';

"+="           return 'O_MAS';
"_"            return 'O_MENOS';
"*="           return 'O_POR';
"/="           return 'O_DIVIDIDO';
```

Configuración de Análisis Sintactico

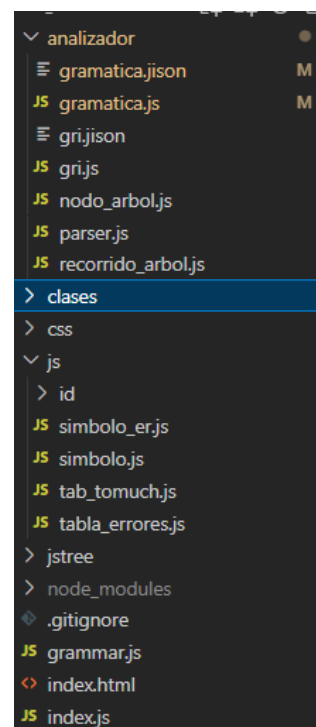
Parte inicial de la definición y declaración de la gramática en el archivo grammar.json

```
%left 'OR'
%left 'AND'
%right 'NOT'
%left 'DOBLEIGUAL' 'NOIGUAL' 'MENQUE' 'MAYQUE' 'MENIGQUE' 'MAYIGQUE'
%left 'MAS' 'MENOS'
%left 'POR' 'DIVIDIDO' 'MOD'
%nonassoc 'POTENCIA'
%right UMENOS

%start inicio
%%

inicio:
    instrucciones EOF { $$= new Nodo_Arbol("INICIO","");
                      $$.agregarHijo($1);
                      return $$; }
;
instrucciones:
    instrucciones instruccion { $$= new Nodo_Arbol("INSTRUCCIONES","");
                                $$.agregarHijo($1);
                                $$.agregarHijo($2);}
    | instruccion { $$= new Nodo_Arbol("INSTRUCCION","");
                  $$.agregarHijo($1); }
;
instruccion
: PRINT PARIZQ expresion PARDER PTCOMA { $$ = new Nodo_Arbol("PRINT","");
    $$.agregarHijo($3);
    $$.tipo=$3.tipo;
    var texto= $$.recorrer_print($3);
    texto = texto.replace(/Ex/g, "");
    document.getElementById("txtsalida1"+publico_id).value+=texto;
}
| DECLARACION { $$ = $1 }
```

Muestra de algunas de las clases, módulos, Herramientas y componentes utilizados en Parte del proyecto para recorrer la gramática y hacer las funciones y exportarlas a la interfaz gráfica.



GRAMATICA UTILIZADA

```
inicio: instrucciones EOF

instrucciones:
    instrucciones instruccion
    | instruccion

instruccion
    : PRINT PARIZQ expresion PARDER PTCOMA
    | DECLARACION
    | ASIGNACION
    | condIF
    | switchCASE
    | condWHILE
    | condDOWHILE
    | condFOR
    | FUNCION
    | LLAMADA
    | IDENTIFICADOR INCREMENTO PTCOMA
    | IDENTIFICADOR DECREMENTO PTCOMA
    | BREAK PTCOMA
    | CONTINUE PTCOMA
    | RETORNO
    | error

RETORNO
    : RETURN expresion PTCOMA
    | RETURN PTCOMA

LLAMADA
    : IDENTIFICADOR PARIZQ PARDER PTCOMA
    | IDENTIFICADOR PARIZQ L_exp PARDER PTCOMA
    | EXEC IDENTIFICADOR PARIZQ PARDER PTCOMA
    | EXEC IDENTIFICADOR PARIZQ L_exp PARDER PTCOMA

FUNCION
    : TIPO IDENTIFICADOR PARIZQ PARDER BLOQUE
    | VOID IDENTIFICADOR PARIZQ PARDER BLOQUE
    | TIPO IDENTIFICADOR PARIZQ PARAMETROS PARDER BLOQUE
    | VOID IDENTIFICADOR PARIZQ PARAMETROS PARDER BLOQUE

PARAMETROS
    : PARAMETROS COMA TIPO IDENTIFICADOR
    | TIPO IDENTIFICADOR
```

```

condFOR
    : FOR PARIZQ DECLARACION expresion PTCOMA refreshFOR PARDER BLOQUE
    | FOR PARIZQ ASIGNACION expresion PTCOMA refreshFOR PARDER BLOQUE

refreshFOR
    : IDENTIFICADOR INCREMENTO
    | IDENTIFICADOR DECREMENTO
    | ASIGNACION

condWHILE
    : WHILE expresion BLOQUE

condDOWHILE
    : DO BLOQUE WHILE expresion PTCOMA

switchCASE
    : SWITCH expresion LLAVIZQ ListCase LLAVDER
    | SWITCH expresion LLAVIZQ ListCase DEFAULT BLOQUECASE LLAVDER

ListCase
    : ListCase CASE expresion BLOQUECASE
    | CASE expresion BLOQUECASE
BLOQUECASE
    : DOSPTS
    | DOSPTS instrucciones

condIF
    : IF expresion BLOQUE
    | IF expresion BLOQUE ELSE condIF
    | IF expresion BLOQUE ELSE BLOQUE

BLOQUE
    : LLAVIZQ LLAVDER
    | LLAVIZQ instrucciones LLAVDER

ASIGNACION
    : IDENTIFICADOR IGUAL expresion PTCOMA

DECLARACION
    : TIPO IDENTIFICADOR IGUAL expresion PTCOMA
    | TIPO IDENTIFICADOR PTCOMA

TIPO
    : RENTERO
    | Rbooleano
    | RCADENA
    | RCARACTER
    | RDOUBLE

```

```
L_exp
: L_exp COMA expresion
| expresion

expresion
: expresion MAS expresion
| expresion MENOS expresion
| expresion POR expresion
| expresion DIVIDIDO expresion
| expresion POTENCIA expresion
| expresion MOD expresion
| expresion DOBLEIGUAL expresion
| expresion NOIGUAL expresion
| expresion MENQUE expresion
| expresion MAYQUE expresion
| expresion MENIGQUE expresion
| expresion MAYIGQUE expresion
| expresion AND expresion
| expresion NOT expresion
| expresion OR expresion
| PARIZQ expresion PARDER
| LOWER PARIZQ expresion PARDER
| UPPER PARIZQ expresion PARDER
| LENGTH PARIZQ expresion PARDER
| TRUNCATE PARIZQ expresion PARDER
| ROUND PARIZQ expresion PARDER
| TYPEOF PARIZQ expresion PARDER
| TOSTRING PARIZQ expresion PARDER
| NOT expresion
| MENOS expresion %prec Umenos
| TRUE
| FALSE
| CADENA
| CHARACTER
| DECIMAL
| ENTERO
| IDENTIFICADOR
| IDENTIFICADOR PARIZQ PARDER
| IDENTIFICADOR PARIZQ L_exp PARDER
```