Gramatica Regular (Javascript)

# Terminales =

“{” “}” “+” “-” “\*” “/” “(” “)” “<” “>” “=” “>=” “<=” “==” “!=” “!” “;” “,” “//” “/\*” “\*/” “:” “.” “&&” “||” “&” ”|” “[” “]” “%” “^” “?” “int” “double” “boolean” “char” “string” “new” “dynamiclist” “append” “getvalue” “setvalue” “if” “else” “switch” “default” “case” “break” “while” “for” “do” “continue” “return” “void” “writeline” “tolower” “toupper” “length” “truncate” “round” “typeof” “tostring” “tochararray” “start” “with” “true” “false”

“Entero” = [0-9]+,

“Numero” = [0-9]+("."[0-9]+)?\b,

“ID” = ([a-zA-Z\_])[a-zA-Z0-9\_]\*,

“Cadenas” = \"[^\"]\*\"

“Carácter” = \'[^\"]?\'

“Carácterentero” = [0-9]

“CarácterAlfabeto” = ([a-zA-Z\_])

Todo = Todos los terminales anteriores

# No Teminales = S, SENTENCIAS, SENTENCIA, VARIABLES, VECTORES, LISTAS, FUNCIONES, METODOS, PARAMETROS, TIPOPARAMETRO, TIPO, LLAMADA, PARALLAMADA, COMENTARIOS, VECTORINT, VECINT, VALORESINT, VECTORSTRING, VECSTRING, VALORESSTRING, VECTORCHAR, VECCHAR, VALORESCHAR, VECTORBOOLEAN, VECBOOLEAN, VALORESBOOLEAN, VECTORDOUBLE, VECDOUBLE, VALORESDOUBLE, VARINT, VARIABLEINT, DECI, VARDOUBLE, VARIABLEDOUBLE, VARBOOLEAN, VARIABLEBOOLEAN, TRUEFALSE, VARSTRING, VARIABLESTRING, MAYMEN, TEXT, VARCHAR, VARIABLECHAR, DECLARACION, LISTAINT, LISTASTRING, LISTACHAR, LISTABOOLEAN, LISTADOUBLE, APPENDLISTA, IF, ELSE, SWITCH, BLOQUESWITCH, LISTCASE, BSW, BREAK, CONTINUE, RETURN, WHILE, FOR, ASIG\_DEC, INCRE\_DECRE, DO\_WHILE, BLOQUE, WRITE, START, VALUE, LISTVALUE, TERNARIO, DECLA\_VECTOR, GETVAL, SETVAL, EXP

No terminal Inicial = S

# Producciones:

S ::= SENTENCIAS EOF ;

SENTENCIAS ::= SENTENCIAS SENTENCIA

| SENTENCIA ;

SENTENCIA ::= VARIABLES

|COMENTARIOS

|VECTORES

|LISTAS

|IF

|SWITCH

|BREAK “;”

|WHILE

|FOR

|INCRE\_DECRE “;”

|DO\_WHILE “;”

|CONTINUE “;”

|RETURN “;”

|FUNCIONES

|METODOS

|WRITE “;”

|START “;”

|TERNARIO “;”

|DECLARACION “;”

|DECLA\_VECTOR “;”

|APPENDLISTA “;”

|GETVAL “;”

|SETVAL “;”

|LLAMADA “;” ;

VARIABLES ::= VARINT “;”

|VARDOUBLE “;”

|VARBOOLEAN “;”

|VARCHAR “;”

|VARSTRING “;” ;

VECTORES ::= VECTORINT “;”

|VECTORSTRING “;”

|VECTORCHAR “;”

|VECTORBOOLEAN “;”

|VECTORDOUBLE “;” ;

LISTAS ::= LISTAINT “;”

|LISTASTRING “;”

|LISTACHAR “;”

|LISTABOOLEAN “;”

|LISTADOUBLE “;” ;

FUNCIONES ::= “int” “ID” PARAMETROS BLOQUE

|”string” “ID” PARAMETROS BLOQUE

|”char” “ID” PARAMETROS BLOQUE

|”boolean” “ID” PARAMETROS BLOQUE

|”double” “ID” PARAMETROS BLOQUE ;

METODOS ::= “void” “ID” PARAMETROS BLOQUE ;

PARAMETROS ::= “(” TIPOPARAMETRO “)”

|”(” “)” ;

TIPOPARAMETRO ::= TIPOPARAMETRO “,” TIPO “ID”

|TIPO “ID” ;

TIPO ::= “int”

|”string”

|”char”

|”boolean”

|”double” ;

LLAMADA ::= “ID” “(“ “)”

|”ID” “(“ PARALLAMADA “)” ;

PARALLAMADA ::= PARALLAMADA “,” EXP

|EXP ;

COMENTARIOS ::= “/\*” “Todo” “\*/”;

VECTORINT ::= “int” “ID” “[” “]” “=” VECINT;

VECINT ::= “new” “int” “[” “Entero” “]”

|”{” VALORESINT “}”;

VALORESINT ::= VALORESINT “,” EXP

| EXP;

VECTORSTRING ::= “string” “ID” “[” “]” ”=” VECSTRING;

VECSTRING ::= “new” “string” “[” “Entero” “]”

|”{” VALORESSTRING “}” ;

VALORESSTRING ::= VALORESSTRING “,” EXP

| EXP ;

VECTORCHAR ::= “char” “ID” “[” “]” “=” VECCHAR;

VECCHAR ::= “new” “char” “[” “Entero” “]”

|”{” VALORESCHAR “}”;

VALORESCHAR ::= VALORESCHAR “,” EXP

| EXP ;

VECTORBOOLEAN ::= “boolean” “ID” “[” “]” “=” VECBOOLEAN;

VECBOOLEAN ::= “new” “boolean” “[” “Entero” “]”

|”{” VALORESBOOLEAN “}” ;

VALORESBOOLEAN ::= VALORESBOOLEAN “,” EXP

| EXP;

VECTORDOUBLE ::= “double” “ID” “[” “]” “=” VECDOUBLE;

VECDOUBLE ::= “new” “double” “[” “Entero” “]”

|”{” VALORESDOUBLE “}” ;

VALORESDOUBLE ::= VALORESDOUBLE “,” EXP

| EXP;

VARINT ::= “int” “ID” VARIABLEINT ;

VARIABLEINT ::= “,” “ID” VARIABLEINT

| “ID”

| “=” EXP

| “=” “(” “int” “)” EXP

| “=” “length” “(” EXP “)”

| “=” “truncate” “(” EXP “)” ;

DECI ::= “Numero”

|”ID” ;

VARDOUBLE ::= “double” “ID” VARIABLEDOUBLE ;

VARIABLEDOUBLE ::= “,” “ID” VARIABLEDOUBLE

| “ID”

| “=” EXP

| “=” “(” “double” “)” EXP

|”=” “round” “(” EXP “)”;

VARBOOLEAN ::= “boolean” “ID” VARIABLEBOOLEAN ;

VARIABLEBOOLEAN ::= “,” “ID” VARIABLEBOOLEAN

| “P\_ID”

| “=” EXP ;

TRUEFALSE ::= “true”

|”false” ;

VARSTRING ::= “string” “ID” VARIABLESTRING ;

VARIABLESTRING ::= “,” “ID” VARIABLESTRING

| “ID”

| “=” EXP

| “=” MAYMEN;

MAYMEN ::= “tolower” “(” EXP “)”

|”toupper” “(” EXP “)”

|”typeof” “(” EXP “)”

|”tostring” “(” EXP “)”;

TEXT ::= “Numero”

|TRUEFALSE ;

VARCHAR ::= “char” “ID” VARIABLECHAR ;

VARIABLECHAR ::= “,” “ID” VARIABLECHAR

| “ID”

| “=” EXP

| “=” “(” “char” “)” EXP;

DECLARACION ::= “ID” “=” EXP

|”ID” “=” “(” TIPO “)” EXP

|”ID” “,” DECLARACION

|”ID” “=” “length” “(” EXP “)”

|”ID” “=” “truncate” “(” EXP “)”

|”ID” “=” “round” “(” EXP “)”

|”ID” MAYMEN;

LISTAINT ::= “dynamiclist” “<” “int” “>” “ID” “=” “new” “dynamiclist” “<” “int” “>” ;

LISTASTRING: ::= “dynamiclist” “<” “string” “>” “ID” “=” “new” “dynamiclist” “<” “string” “>”;

LISTACHAR: ::= “dynamiclist” “<” “char” “>” “ID” “=” “new” “dynamiclist” “<” “char” “>”

|“dynamiclist” “<” “char” “>” “ID” “=” “tochararray” “(” EXP “)” ;

LISTABOOLEAN ::= “dynamiclist” “<” “boolean” “>” “ID” “=” “new” “dynamiclist” “<” “boolean” “>”;

LISTADOUBLE ::= “dynamiclist” “<” “double” “>” “ID” “=” “new” “dynamiclist” “<” “double” “>”;

APPENDLISTA ::= “append” “(” “ID” “,” EXP “)” ;

IF ::= “if” “(” EXP “)” BLOQUE

|”if” “(” EXP “)” BLOQUE “else” ELSE ;

ELSE ::= IF

|BLOQUE ;

SWITCH ::= “switch” “(” EXP “)” BLOQUESWITCH ;

BLOQUESWITCH ::= “{” “}”

|”{” LISTCASE “}” ;

LISTCASE ::= “case” EXP BSW LISTCASE

|”case” EXP BSW

|”default” BSW ;

BSW ::= “:” SENTENCIAS

|”:” ;

BREAK ::= “break” ;

CONTINUE ::= “continue” ;

RETURN ::= “return”

|”return” EXP ;

WHILE ::= “while” “(” EXP “)” BLOQUE ;

FOR ::= “for” “(” ASIG\_DEC “;” EXP “;” EXP “)” BLOQUE;

ASIG\_DEC ::= “int” “ID” “=” EXP

|”ID” “=” EXP ;

INCRE\_DECRE ::= “ID” “+” “+”

|”ID” “-” “-” ;

DO\_WHILE ::= “do” BLOQUE “while” “(” EXP “)” ;

BLOQUE ::= “{” SENTENCIAS “}”

|”{” “}” ;

WRITE ::= “writeline” “(” EXP “)” ;

START ::= “start” “with” “ID” VALUE ;

VALUE ::= “(” LISTVALUE “)”

|”(” “)” ;

LISTVALUE ::= LISTVALUE “,” EXP

|EXP ;

TERNARIO ::= “ID” “=” EXP “?” EXP “:” EXP ;

DECLA\_VECTOR ::= “ID” “[” “Entero” “]” “=” EXP

|”ID” “[” “Entero” “]” “=” “(” TIPO “)” EXP

|”ID” “[” “Entero” “]” “,” DECLA\_VECTOR ;

GETVAL ::= “getvalue” “(” “ID” “,” EXP “)”;

SETVAL ::= “setvalue” “(” “ID” “,” EXP “,” EXP “)” ;

EXP ::= EXP “+” EXP

|EXP “-” EXP

|EXP “\*” EXP

|EXP “/” EXP

|EXP “^” EXP

|EXP “%” EXP

|EXP “!=” EXP

|EXP “==” EXP

|EXP “☹” EXP

|EXP “>” EXP

|EXP “<” EXP

|EXP “>=”EXP

|EXP “<=” EXP

|EXP “&&”EXP

|EXP “||” EXP

|”!” EXP

|”(” EXP “)”

|”Cadenas”

|”Numero”

|”true”

|”false”

|”ID”

|”-” EXP

|EXP “+” “+”

|EXP “-” “-”

|”Caracter”

|”ID” “[” “Entero” “]”

|GETVAL

|LLAMADA;