

Universidad San Carlos de Guatemala
Organización de Lenguajes y Computadores 1

Archivo de gramáticas

Proyecto 1

Nataly Saraí Guzmán Duarte

Carnet: 202001570

Gramática Para el Archivo Json:

Terminales:

CADENA, CADEN, COMA, DOUBLEN, LLAVE_A, LLAVE_C, DOSPUNTOS, COMENLI, COMENMULTILI, NOMBREAR.

No Terminales:

Inicio, asignación, lista_instr, instrucción, expresión, instrucjon, instrucg.

Producción:

inicio ::= lista_instr

;

lista_instr ::= lista_instr instruccion

| instruccion

;

instruccion ::= LLAVE_A instrucjon LLAVE_C

| COMENMULTILI

| COMENLI

;

instrucjon ::= instrucjon COMA asignacion

| asignacion

;

asignacion ::= CADENA:nombre DOSPUNTOS expresion:var

{:Devuelve los datos para guardarlos en otra clase :}

;

expresion ::= DOUBLEN:val {: Devuelve valor :}

| CADEN:val {: Devuelve valor:}

| CADENA:val {: Devuelve valor:}

;

Gramática Para el Archivo SP:

Terminales:

ENTERO, POR, MAS,MENOS,DIVISION, PARENTESIS_A, PARENTESIS_C, LLAVE_A, LLAVE_C,IGUAL, MOSTRAR, IF,ELSE,FOR,WHILE, PUNTOY-COMA,COMA, IDENTI,INT,DOUBLE,CHAR,BOOL,STRING, DOUBLEN,TRUE,FALSE,CADENA,COMENLI,COMENMULTILI, VOID, MAIN, MAYOR,MENOR,MAYORIGUAL,MENORIGUAL,IGUALIGUAL,CERO, AND, OR, NOT,DO,SWITCH,CASE,BREAK,DOSPUNTOS,DEFAULT, DEFIGLO,DOLAR,NEWVALOR,GRAFIB, TITULO,TITULOX,TITULOY,EJEX,EJEY,GRAFIP, CORCHETE,VALORES.

No Terminales:

Inicio, LinkedList lista_instr, instrucción,expresion, expresionf, instr_mostrar, if,else,for,while,declara,declaraig,comentarios,comentarios, declaradou ,declaraigdou, declaracha ,declaraigcha,declarabol ,declaraigbol, declaracade ,declaraigcade,declararfas, grafibarras, stringb,maind,do,funcion,declaraigdounew, ejex,stringtitux,stringtituy,grafipie, switch,case,break,default, expres,valores.

Precendencia:

left MAS, left POR, left MENOS,left DIVISION, left MAYOR, left MENOR, left MAYORIGUAL, left MENORIGUAL,left IGUALIGUAL,left AND,left OR, right NOT.

Producción:

inicio ::= lista_instr:lista { : Devuelve lista; :}

;

lista_instr ::= lista_instr instruccion

| instruccion

;

instruccion ::= instr_mostrar

| if

| declara

| declaraig

| declaradou

| declaraigdou

| declaracha

- | declaraigcha
- | declarabol
- | declaraigbol
- | declaracade
- | declaraigcade
- | comentarios
- | comentariosl
- | maind
- | else
- | while
- | for
- | declararfas
- | do
- | funcion
- | declaraigdounew
- | grafibarras
- | stringb
- | ejex
- | stringtitux
- | stringtituy
- | grafipie
- | switch
- | case
- | break
- | default
- | expres
- | valores

;

valores ::= DOUBLE CORCHETE VALORES IGUAL LLAVE_A lista_instr
 LLAVE_C PUNTOYCOMA ;

expres ::= CADENA COMA

- | CADENA
- | IDENTI COMA
- | IDENTI
- | DOUBLEN COMA
- | DOUBLEN
- | DOLAR LLAVE_A NEWVALOR COMA expresion COMA expresion LLAVE_C
COMA
- | DOLAR LLAVE_A NEWVALOR COMA expresion COMA expresion LLAVE_C

 ;

ejex ::= STRING CORCHETE EJEX IGUAL LLAVE_A lista_instr LLAVE_C PUN-
 TOYCOMA ;

default ::= DEFAULT DOSPUNTOS;

break ::= BREAK PUNTOYCOMA;

case ::= CASE expresion DOSPUNTOS;

switch ::= SWITCH PARENTESIS_A expresion PARENTESIS_C LLAVE_A
 lista_instr LLAVE_C;

grafibarras ::= VOID GRAFIB PARENTESIS_A PARENTESIS_C LLAVE_A
 lista_instr LLAVE_C;

grafipie ::= VOID GRAFIP PARENTESIS_A PARENTESIS_C LLAVE_A lista_instr
 LLAVE_C;

stringb ::= STRING TITULO IGUAL expresion PUNTOYCOMA ;

stringtitux ::= STRING TITULOX IGUAL expresion PUNTOYCOMA;

stringtituy ::= STRING TITULOY IGUAL expresion PUNTOYCOMA;

funcion ::= VOID DEFIGLO PARENTESIS_A PARENTESIS_C LLAVE_A
 lista_instr LLAVE_C;

declaraigdownew ::= DOUBLE expresion IGUAL DOLAR LLAVE_A NEWVALOR
 COMA expresion COMA expresion LLAVE_C PUNTOYCOMA;

```

maind ::= VOID MAIN PARENTESIS_A PARENTESIS_C LLAVE_A lista_instr
LLAVE_C;

comentarios ::= COMENMULTILI;

comentariosl ::= COMENLI;

if ::= IF PARENTESIS_A expresion PARENTESIS_C LLAVE_A lista_instr
LLAVE_C;

while ::= WHILE PARENTESIS_A expresion PARENTESIS_C LLAVE_A lista_instr
LLAVE_C;

do ::= INT expresion IGUAL expresion PUNTOYCOMA DO LLAVE_A lis-ta_instr
LLAVE_C WHILE PARENTESIS_A expresion PARENTESIS_C PUNTOYCOMA;

else ::= ELSE IF PARENTESIS_A expresion PARENTESIS_C LLAVE_A lista_instr
LLAVE_C;

instr_mostrar ::= MOSTRAR PARENTESIS_A expresion PARENTESIS_C PUN-
TOYCOMA;

declara ::= INT expresion PUNTOYCOMA;

declararfas ::= expresion IGUAL expresion PUNTOYCOMA ;

declaraig ::= INT expresion IGUAL expresion PUNTOYCOMA ;

declaradou ::= DOUBLE expresion PUNTOYCOMA;

declaraigdou ::= DOUBLE expresion IGUAL expresion PUNTOYCOMA;

declaracha ::= CHAR expresion PUNTOYCOMA ;

declaraigcha ::= CHAR expresion IGUAL expresion PUNTOYCOMA;

declarabol ::= BOOL expresion PUNTOYCOMA ;

declaraigbol ::= BOOL expresion IGUAL expresion PUNTOYCOMA;

declaracade ::= STRING expresion PUNTOYCOMA ;

declaraigcade ::= STRING expresion IGUAL expresion PUNTOYCOMA;

expresion ::= expresion POR expresion
| expresion MAS expresion
| expresion MENOS expresion
| expresion DIVISION expresion
| expresion MAYOR expresion

```

- | expresion MENOR expresion
- | expresion MAYORIGUAL expresion
- | expresion MENORIGUAL expresion
- | expresion IGUALIGUAL expresion
- | expresion NOT IGUAL expresion
- | expresion AND expresion
- | expresion OR expresion
- | expresion NOT expresion
- | CADENA MAS expresion
- | ENTERO
- | IDENTI
- | DOUBLEN
- | TRUE
- | FALSE
- | CADENA

;

for ::= FOR PARENTESIS_A INT expresionf expresionf PUNTOYCOMA expresion
MAS MAS PARENTESIS_C LLAVE_A lista_instr LLAVE_C ;

expresionf ::= IGUAL CERO PUNTOYCOMA expresionf

- | IGUAL expresionf PUNTOYCOMA expresionf
- | INT expresionf IGUAL expresionf
- | ENTERO
- | IDENTI
- | expresionf MENOR expresionf
- | expresionf MENORIGUAL expresionf

;