

ARCHIVO DE GRAMÁTICAS DE “SYSCOMPILER”

ini

```
:general EOF{  
    return $1;  
}
```

;

general

```
:general cuerpo  
|cuerpo
```

;

cuerpo

```
:COMENTARIO  
|funcionMetodo  
|start with ID ( ) ;  
|start with ID ( expresion ) ;  
|decl
```

;

funcionMetodo

```
:tipo ID ( parametros ) { statement }  
|tipo ID ( ) { statement }
```

```
|DynamicList < tipo > ID ( parametros ) { statement }  
|DynamicList < tipo > ID ( ) { statement }
```

```
|tipo ID [ ] ( parametros ) { statement }  
|tipo ID [ ] ( ) { statement }
```

```
|void ID ( parametros ) { statement }  
|void ID ( ) { statement }
```

;

parametros

```
:parametros , parametros  
|tipo ID  
|DynamicList < tipo > ID  
|tipo ID [ ]
```

;

```
cuerpoFunciones
    :cuerpoFunciones declaraciones
    |declaraciones
;
;
```

```
declaraciones
    :sentencias
    |WriteLine ( listaExpresiones ) ;
    |break ;
    |continue ;
    |return expresion ;
    |return ;
    |COMENTARIO
    |decl
;
;
```

```
decl
    :variable
    |vectores
    |listas
    |error ;
;
;
```

```
sentencias
    :if
    |switch
    |while
    |for
    |dowhile
;
;
```

```
dowhile
    :do { statement } while ( expresion ) ;
;
;
```

```
for
    :for ( declaracionFor ; expresion ; declaracionFor ) { statement }
    |for ( declaracionFor ; expresion ; ID + + ) { statement }
    |for ( declaracionFor ; expresion ; ID - - ) { statement }
```

;

declaracionFor

:tipo ID = expresion

|ID = expresion

|tipo ID = casteos

|ID = casteos

;

switch

:switch (expresion) { casos }

;

if

:if (expresion) { statement } else

;

else

:else { statement }

|else if

|

;

while

:while (expresion) { statement }

;

statement

:cuerpoFunciones

|

;

casos

:casos caso

|caso

;

caso

:case expresion : statement

|default : statement

;

variable

```
:tipo identificadores ;  
|tipo identificadores = expresion ;  
|identificadores = expresion ;  
|tipo identificadores = casteos  
|identificadores = casteos  
|ID + + ;  
|ID - - ;  
|ID ( listaExpresiones ) ;  
|ID ( ) ;
```

;

vectores

```
:tipo ID [ ] = new tipo [ expresion ] ;  
|tipo ID [ ] = { listaVectores } ;  
|ID [ expresion ] = expresion ;  
|ID [ ] = new tipo [ expresion ] ;  
|ID [ ] = { listaVectores } ;
```

;

listas

```
:DynamicList < tipo > ID = new DynamicList < tipo > ;  
|append ( ID , expresion ) ;  
|setValue ( ID , aritmeticos , expresion ) ;  
|DynamicList < tipo > ID = toCharArray ( expresion ) ;
```

;

listaVectores

```
:listaVectores , listaVectores  
|expresion
```

;

listaExpresiones

```
:listaExpresiones , expresion  
|expresion
```

;

aritmeticos

```

:ID
|aritmeticos + aritmeticos
|aritmeticos - aritmeticos
|aritmeticos * aritmeticos
|aritmeticos / aritmeticos
|aritmeticos % aritmeticos
|aritmeticos ^ aritmeticos
|ENTERO
|DECIMAL
|ID [ expresion ]
|getValue ( ID , expresion )
|( aritmeticos )
;

casteos
:( tipo ) expresion ;
;

identificadores
: identificadores , identificadores
| ID
;

tipo
:int
|boolean
|double
|char
|string
;

expresion
:- expresion %prec UMENOS
|! expresion
|expresion && expresion
|expresion || expresion
|expresion + expresion
|expresion - expresion
|expresion * expresion
|expresion / expresion

```

|expression % expression
|expression ^ expression
|(expression)
|expression == expression
|expression != expression
|expression >= expression
|expression <= expression
|expression > expression
|expression < expression
|expression + +
|expression - -
|ID

|ID [expression]
|getValue (ID , expression)

|ID (listaExpresiones)
|ID ()

|toLower (expression)
|toUpper (expression)
|length (expression)
|truncate (expression)
|round (expression)
|typeof (expression)
|toString (expression)
|toCharArray (expression)

|ENTERO
|DECIMAL
|true
|false
|CHARACTER
|FRASE
|expression ? expression : expression

;