

**Asignación:**

Tarea #1

**Curso:**

Compiladores e Intérpretes

**Grupo:**

60

**Profesor:**

Allan Rodriguez Davila

**Estudiantes:**

Kevin Lanzas Quen(2019160391)

Jirgort McCarty Vazquez(2019203402)

**Semestre 1 del 2022**

**Año:**

2022

**Gramática de lenguaje**

letra←[A-Z a-z]

digR←[1-9]

dig←[0-9]

id←letra(letra|digito)\*

cero←0

sigN←”-”

entero←sigN? digR dig\*

flotante←(sigN? digR dig\*)\punto dig+

punto←”.”

array← declaArray |obtValArray| modArray

char←letra

String←id

Boolean←”True” | “False”

Nulo ← “Ε”

=================P===================

inicioC:: inicio

inicio ← main func\*

inicio←func\* main func\*

==================A-O================

inicio←func\*

func ← tipo id “(“(tipo “parametro”)\*? “)” “{“ bloque “}”

parametro← (entero|flotante|String|bolean|char|nulo)

==================B==================

tipo ← “ (float |char | int | Ε | Boolean | String | ) ” id “ # ”

=====================C===============

declaArray←entero|Char (id [dim])”#”

obtValArray← id [entero] “#”

modArray← id[entero] “=” entero|char “#”

==================D==================

bloque←sentencia expresion return

sentencia← while\*| switch\*| expresion\*| if |

creaAsig← tipo id “=”expresion “#”

creaVar← tipo id “#”

AsigExpVar←crearVar“=”expresion “#”

ExpSinAsig←tipo id “(“expresion”)” “#”

==================E=================

expresion← creaAsig |creaVar|AsigExpVar|ExpSinAsig|expresionesAritUnarias|expresionAritBinaria |expresionesLogicas |array|break|return

==================G================

expresionAritBinaria ← (“int” id = (entero|cero) (“+”|”-”|”//”|”\*” |”~”|”^”)(entero|cero)) | (“float” id = (flotante|cero) (“+”|”-”|”/”|”\*” |”~”|”^”)(flotante|cero)) | (“float” id = (flotante|cero) (“+”|”-”|”/”|”\*” |”~”|”^”)(entero|cero)) | (“float” id = (entero|cero) (“+”|”-”|”/”|”\*” |”~”|”^”)(flotante|cero)) “return” flotante “#”

==================H================

expresionesAritUnarias←(((”-”?)(entero|cero)) | ((”-”?)(flotante|cero)) ) | (id (”++”?|”--”?))“#”

===================I================

expresionesRelaciones←(”int” id = (entero|cero) (“<”|”>”|”<=”|”>=”|”=”|”!=”)(entero|cero)) | (”float” id = (flotante|cero) (“<”|”>”|”<=”|”>=”|”=”|”! =”)(flotante|cero)) | (”flotante” id = (entero|cero) (“<”|”>”|”<=”|”>=”|”=”|”!=”)(flotante|cero)) | (”flotante” id = (flotante|cero) (“<”|”>”|”<=”|”>=”|”=”|”!=”)(entero|cero)) | (”boolean” id = (Boolean) (”=”,”!=”)(Boolean)) “return” Boolean“#”

===================J=================

expresionesLogicas←”boolean”id = (not?(“expresion”))| (not?)(“expresion”)(“And”|”or”)(not?)(“expresion”)”#”

====================M===============

if ← “if” ”(”expresión “)” (“{“ bloque”}” | “{“bloque”} “else” “{“ bloque ”}”)

while←”while”(“condicional”)({“sentencia” break?})

switch ← “switch”(dig) “{“case\* ”}”

case←sentencia\* break? “#”

break← “break””#”

return←”return” (“expresion”)”#”

====================N==========================

funcR← “read””(“entero|flotante”)” “#”

FuncP← “print””(”char|entero|flotante|id”)””#”

===================Q==========================

comentariosUnaLinea←“//” (letra(letra|digito)\* ) “#”

comentariosMultiple←”/\*” (“{”(letra|digito)\*”}) “\*/” ”#”