

```

inicio ::= funcS

funcS ::= func
funcS ::= funcS func
func ::= tipo separador id '(' ')' '{' bloque '}'
func ::= tipo separador id '(' parametros ')' '{' bloque '}'

parametros ::= param
parametros ::= parametros ',' param
param ::= tipo separador id

bloque ::= expr
bloque ::= bloque expr

expr ::= creacion | creacionAsign | asign | estructura | exprUni | retorno

estructura ::= if | for

if ::= 'if' '(' exprRelLog ')' '{' bloque '}'
if ::= 'if' '(' exprRelLog ')' '{' bloque '}' 'else' '{' bloque '}'

for ::= 'forRange' '(' paramRange ')' '{' bloque '}'
paramRange ::= intN //fin
paramRange ::= intN ',' intN //inicio,fin
paramRange ::= intN ',' intN ',' intN // inicio,fin,step

exprUni ::= exprP finExpr

retorno ::= 'return' finExpr
retorno ::= 'return' separador exprP finExpr

creacion ::= tipo id finExpr

creacionAsign ::= tipo id '=' exprP finExpr

asign ::= id '=' exprP finExpr

exprP ::= exprArit | exprRelLog

exprArit ::= operandoArit
exprArit ::= exprArit operadorArit operandoArit

exprRelLog ::= exprRel_I | exprLog_I

exprRel_I ::= operandoRel operadorRel operandoRel

exprLog_I ::= operandoLog
exprLog_I ::= exprLog_I operadorLog operandoLog

```

```

invocaFunc ::= id '(' ')'
invocaFunc ::= id '(' parametrosInv ')'
parametrosInv ::= paramInv
parametrosInv ::= parametrosInv ',' paramInv
paramInv ::= exprP

operandoArit ::= id | literal | invocaFunc
operandoRel ::= operandoArit | '(' exprArit ')'
operandoLog ::= id | literal | invocaFunc | '(' exprRel_I ')'

operadorArit ::= '+' | '-' | '/' | '*'
operadorRel ::= '<' | '>' | '==' | '!='
operadorLog ::= '&&' | '||'

tipo ::= 'int' | 'float' | 'char' | 'string' | 'bool'

id ::= [a-zA-Z_][a-zA-Z0-9_]*
literal ::= intN | floatN | charN | stringN | boolN
intN ::= (-?)[1-9][0-9]* | 0
floatN ::= (((-?)[1-9][0-9]*) | 0) . [0-9]+ | 0.0
charN ::= '"' [a-zA-Z]? '"'
stringN ::= '"' [^"]* '"'

boolN ::= 'true' | 'false'
separador ::= '#'
finExpr ::= ';'

```

Ejemplo:

```

int#main(){miFunc(hola, 'a');return;}

int # miFunc (int # dif, char # otra ) {
    int a;
    string str = "Hola $%&/#$&) mundo" ;
    int var = 34 + id ^ miFunc2 (func3(34,id));
    if ((var+23) > 45 && true || otra == dif){
        miV = (var+23) > 45 && true || otra == otrFunc() ;
    }
    forRange(5){print("Hola mundo");}
    return # 3+4;
}

```