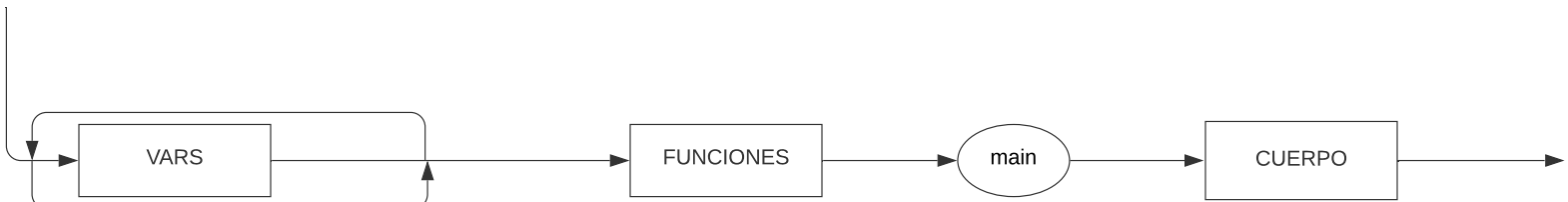
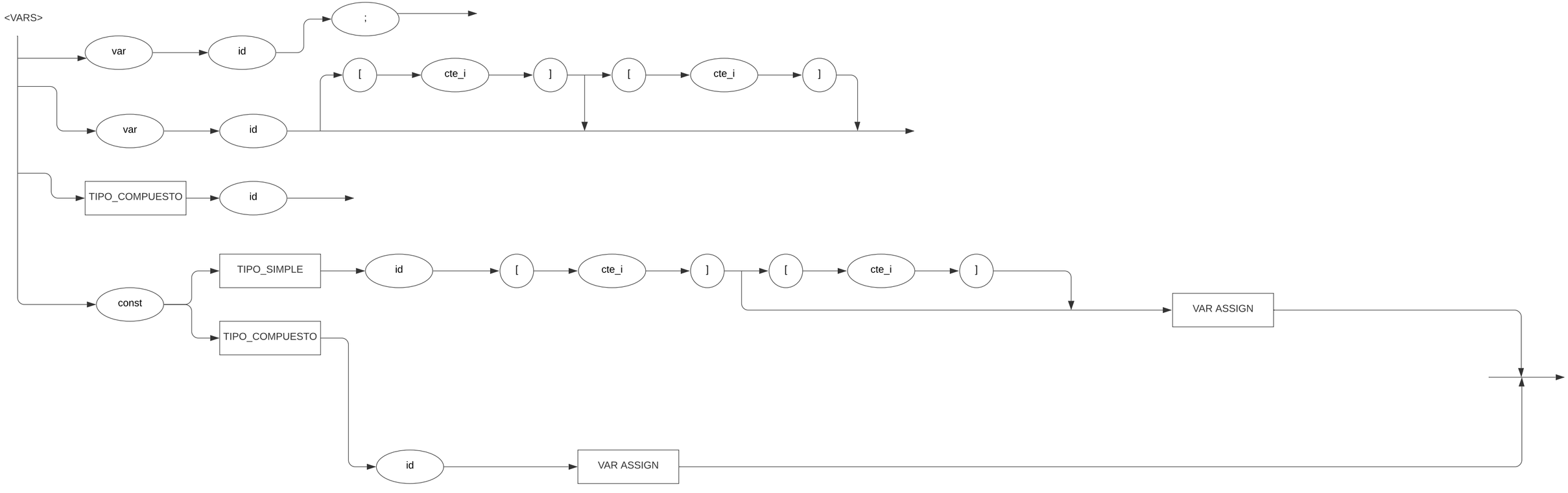
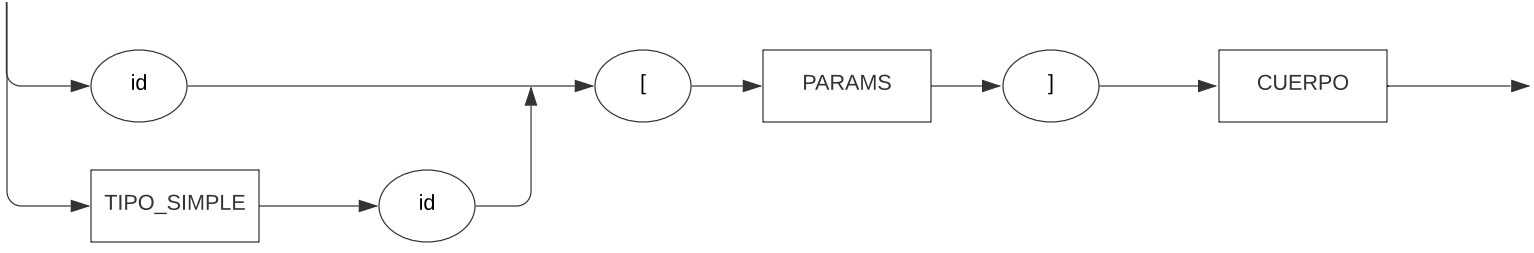
<PROGRAMA>

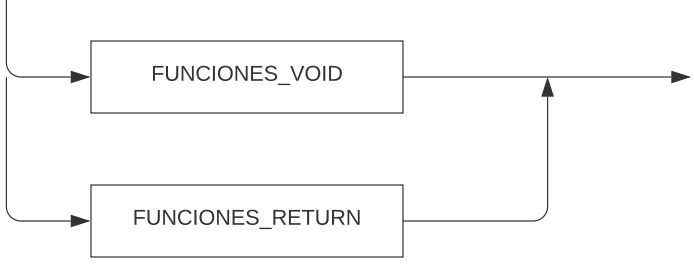


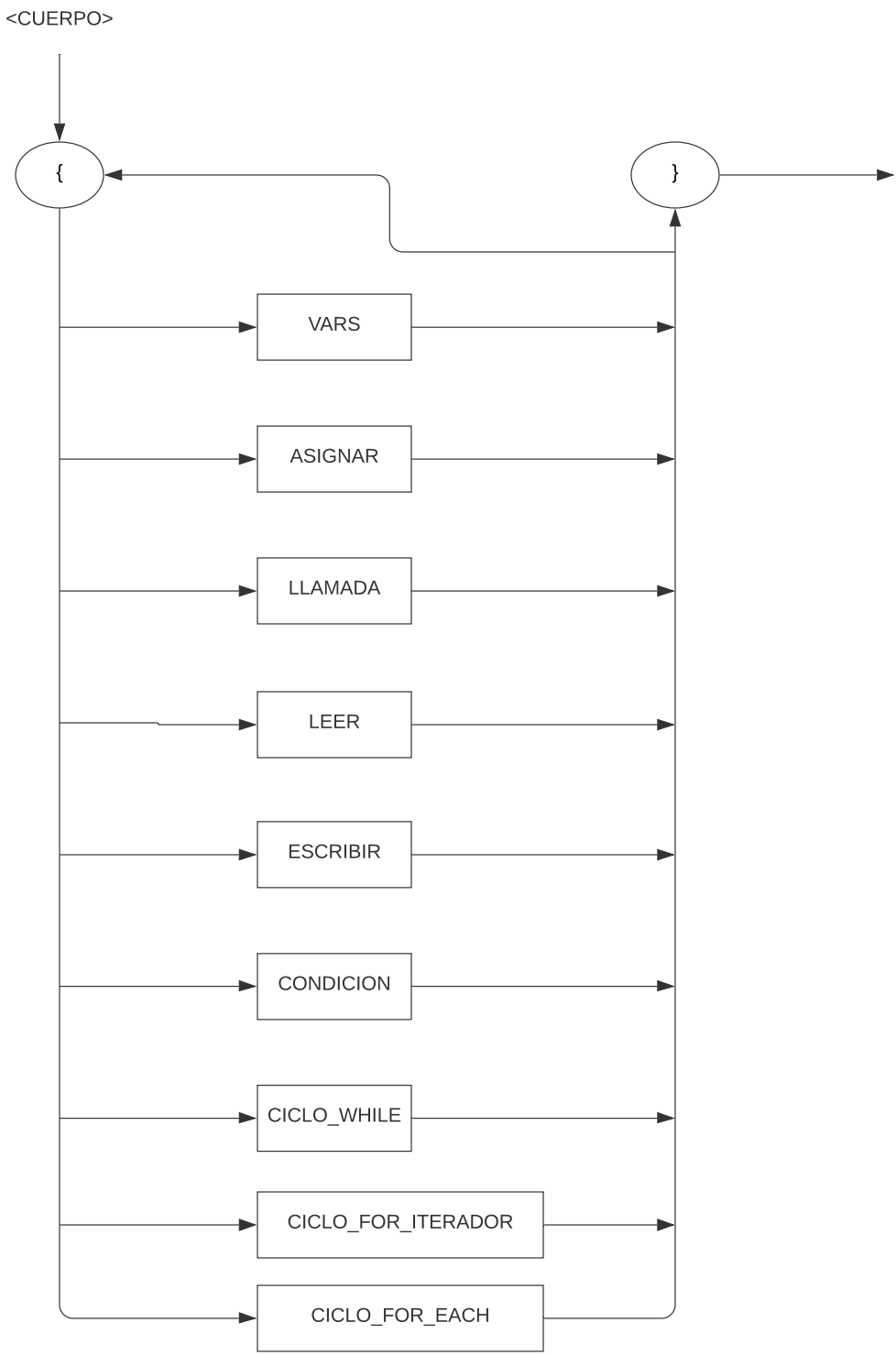

<VARS>
var
id
;
var
id
[
cte_i
]
[
cte_i
]
TIPO_COMPUESTO
id
const
TIPO_SIMPLE
id
[
cte_i
]
[
cte_i
]
TIPO_COMPUESTO
VAR ASSIGN
id
VAR ASSIGN

<FUNCION>

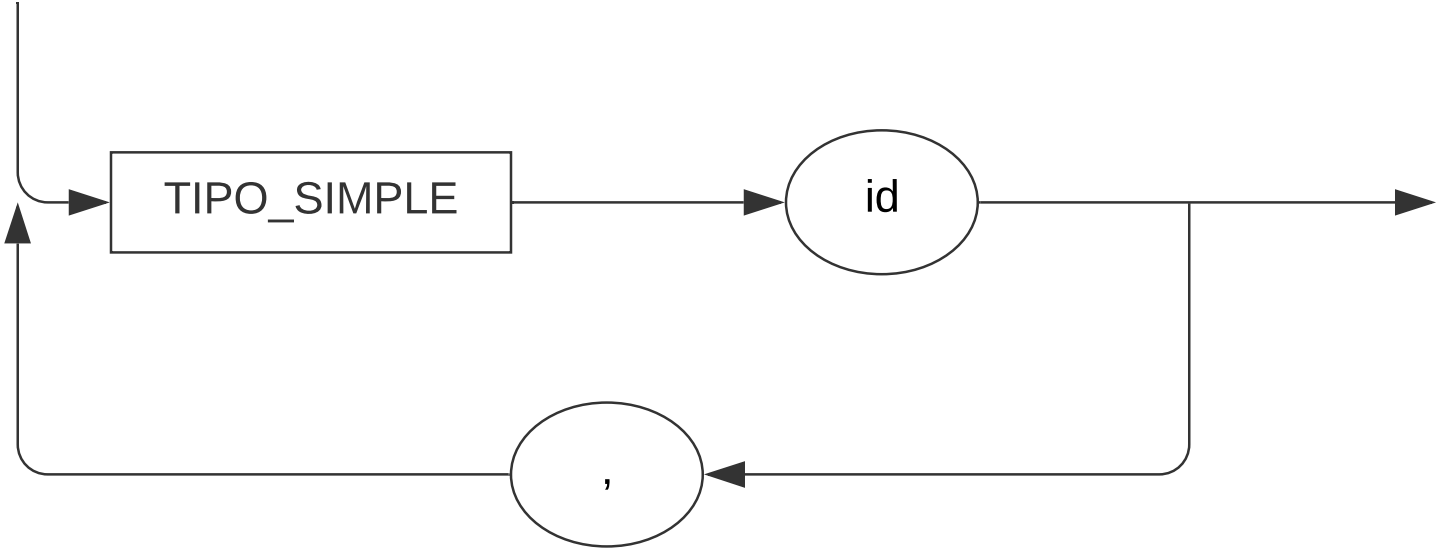


<FUNCIONES>



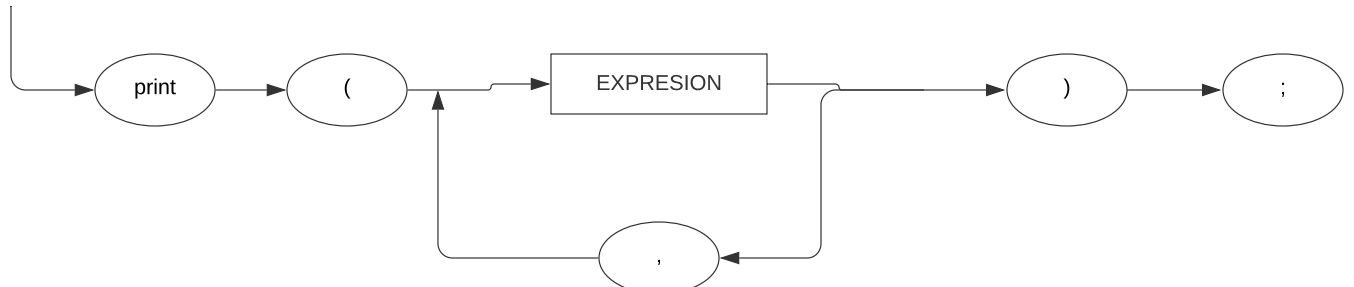

<CUERPO>
{
}
VARS
ASIGNAR
LLAMADA
LEER
ESCRIBIR
CONDICION
CICLO_WHILE
CICLO_FOR_ITERADOR
CICLO_FOR_EACH

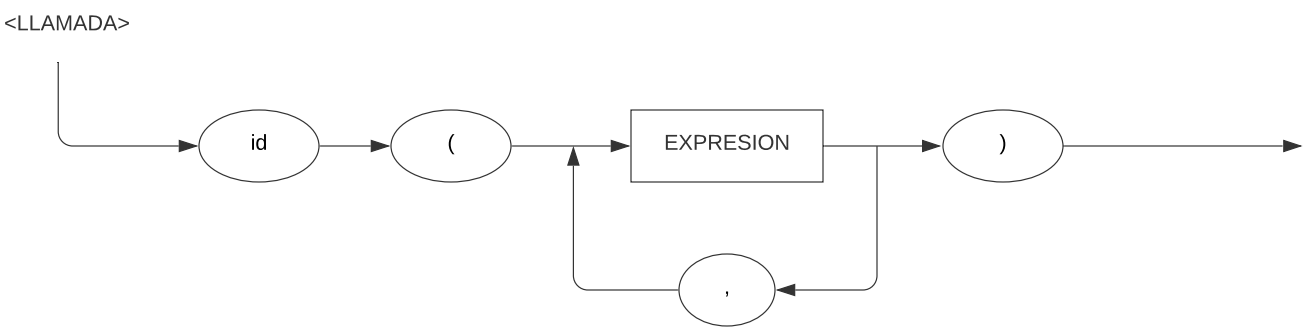
<PARAMS>



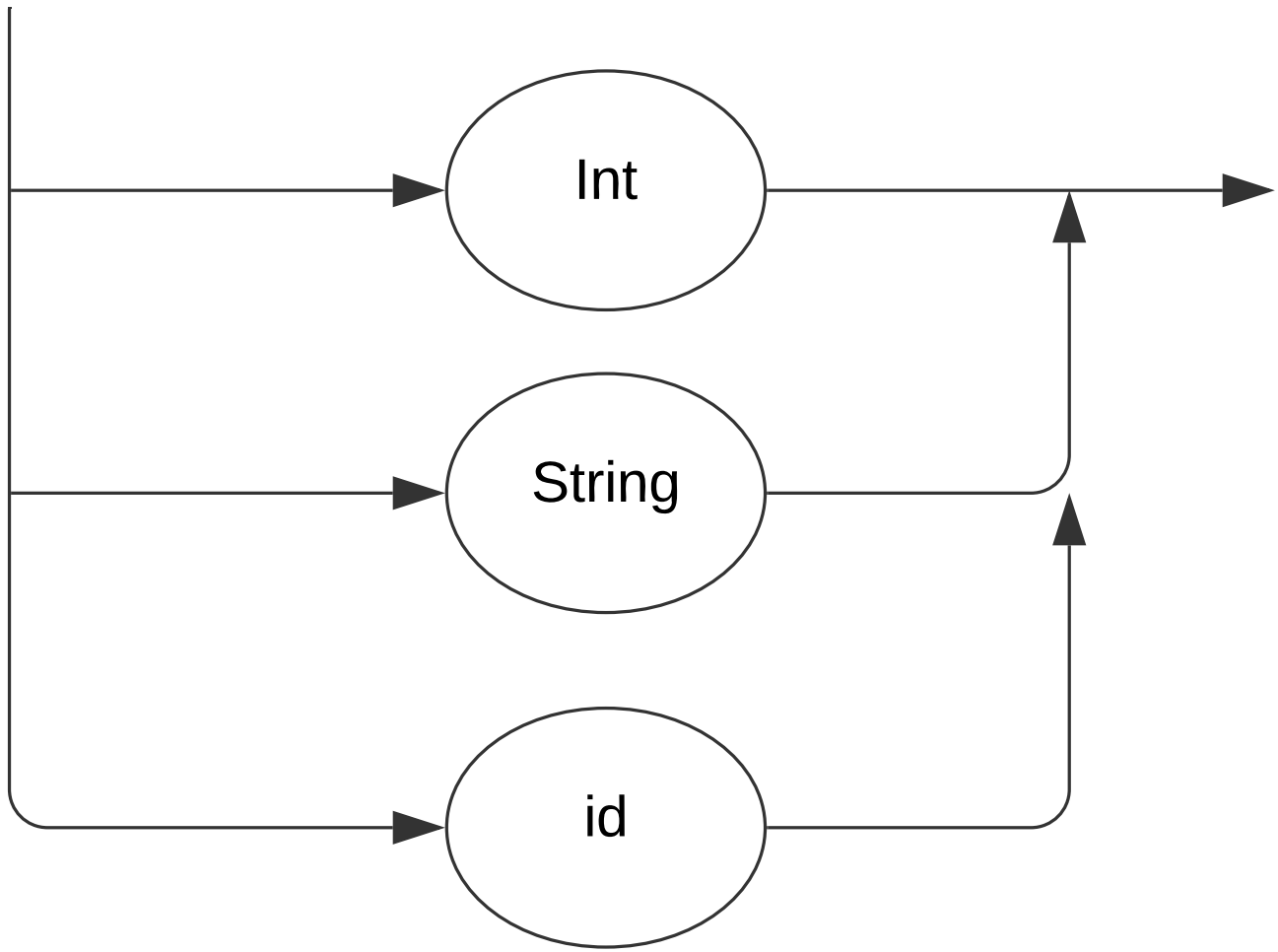

<ASIGNAR>
id
VAR ASSIGN
;

<ESCRIBIR>

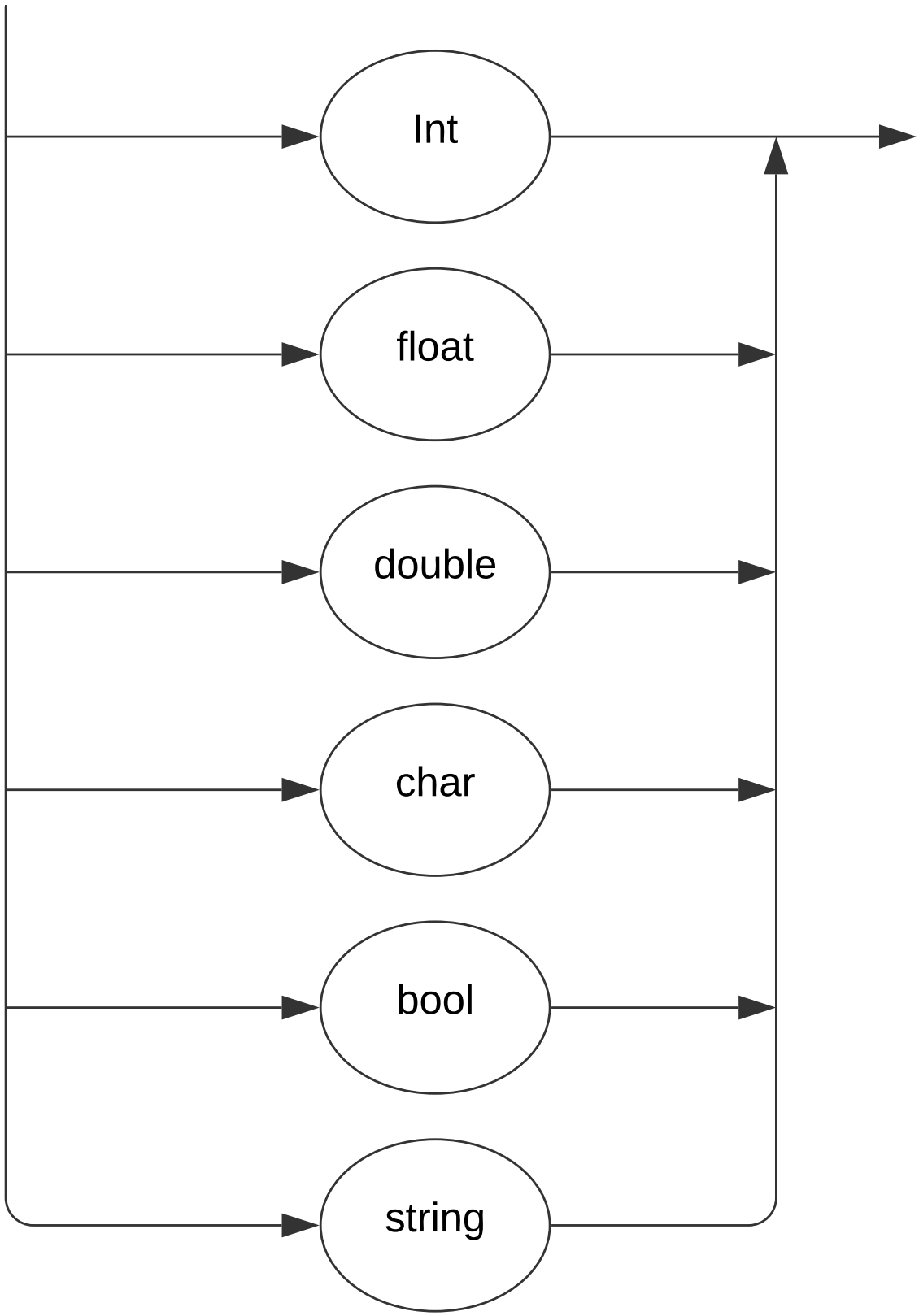


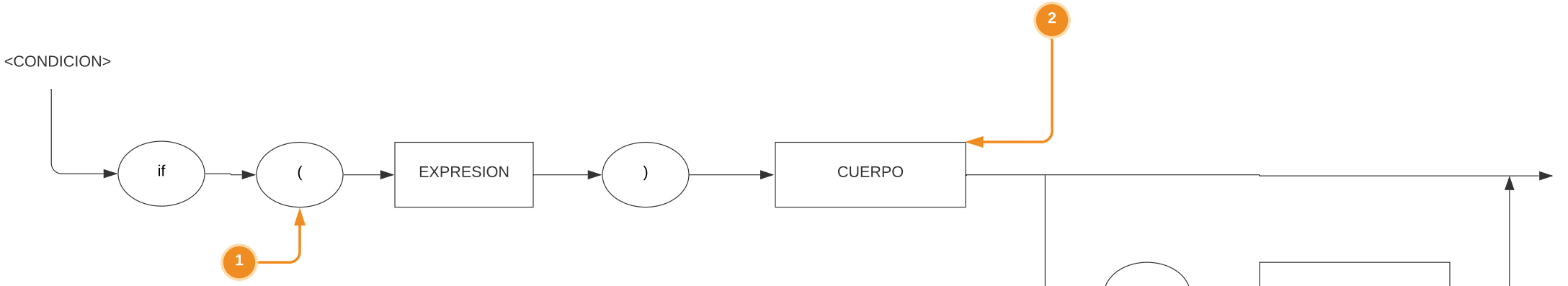

<LLAMADA>
id
(
EXPRESION
)
,

<TIPO_COMPUESTO>



<TIPO_SIMPLE>

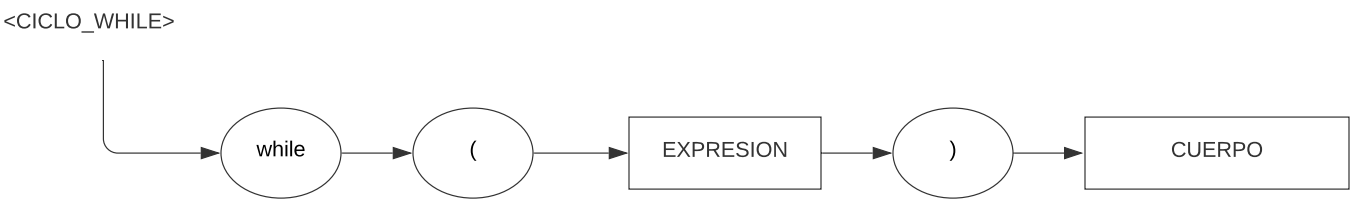


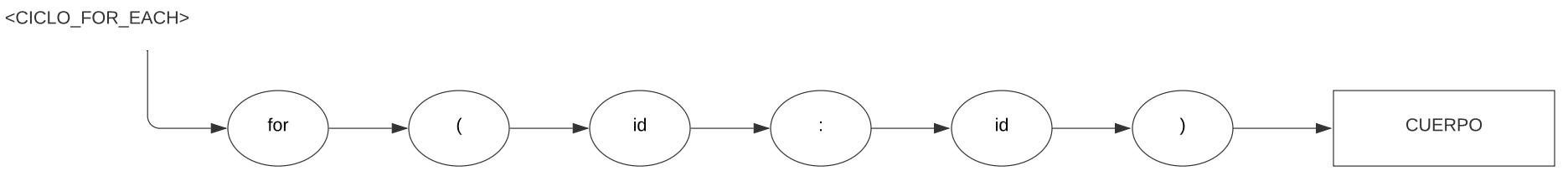


1.- exp_type = PTypes.Pop()
if (**exp_type** != bool) **ERROR**(Type-mismatch)
else
result = PilaO.Pop()
Generate quad: **GotoF, result , , ___**
PJumps.Push (cont-1)

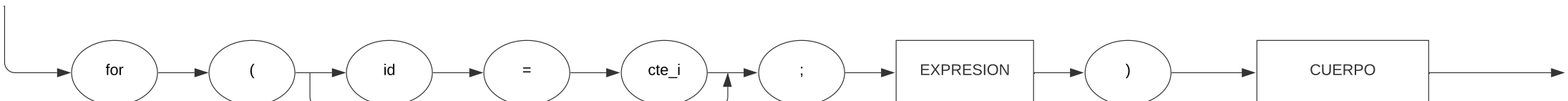
2.- **end**=PJumps.Pop()
FILL (end, cont)

3.- Generate quad: **GOTO ___**
false= PJumps.Pop()
PJumps.Push(**cont-1**)
FILL (false, cont)

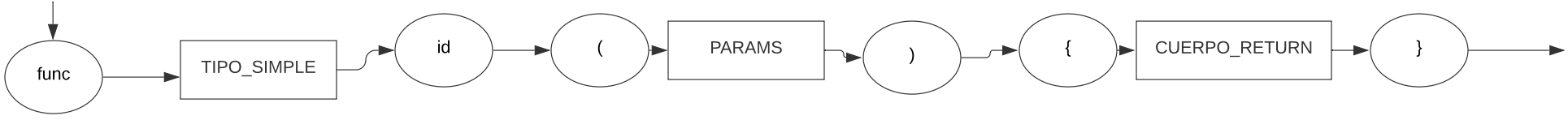

<CICLO_WHILE>
while
(
EXPRESION
)
CUERPO


<CICLO_FOR_EACH>
for
(
id
:
id
)
CUERPO

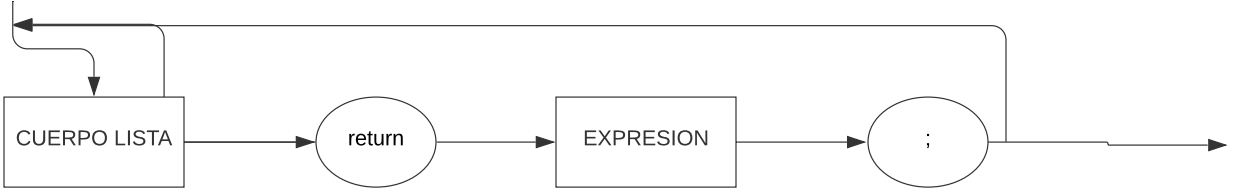
<CICLO_FOR_ITERADOR>

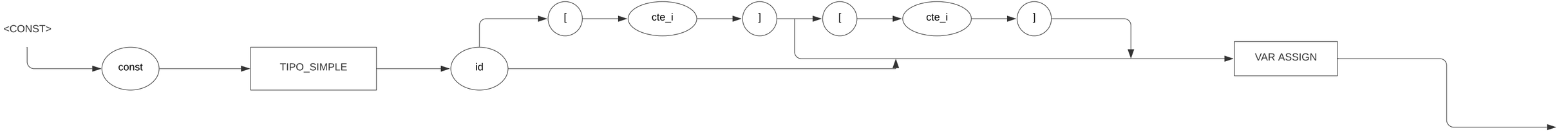


<FUNCIONES_RETURN>

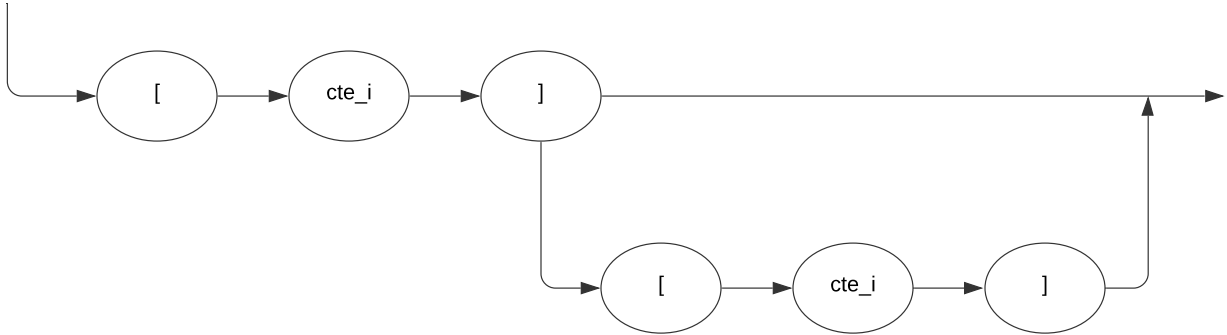


<CUERPO_RETURN>

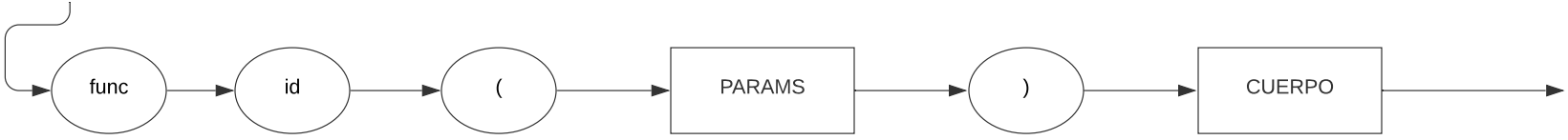



<CONST>
const
TIPO_SIMPLE
id
[
cte_i
]
[
cte_i
]
VAR ASSIGN

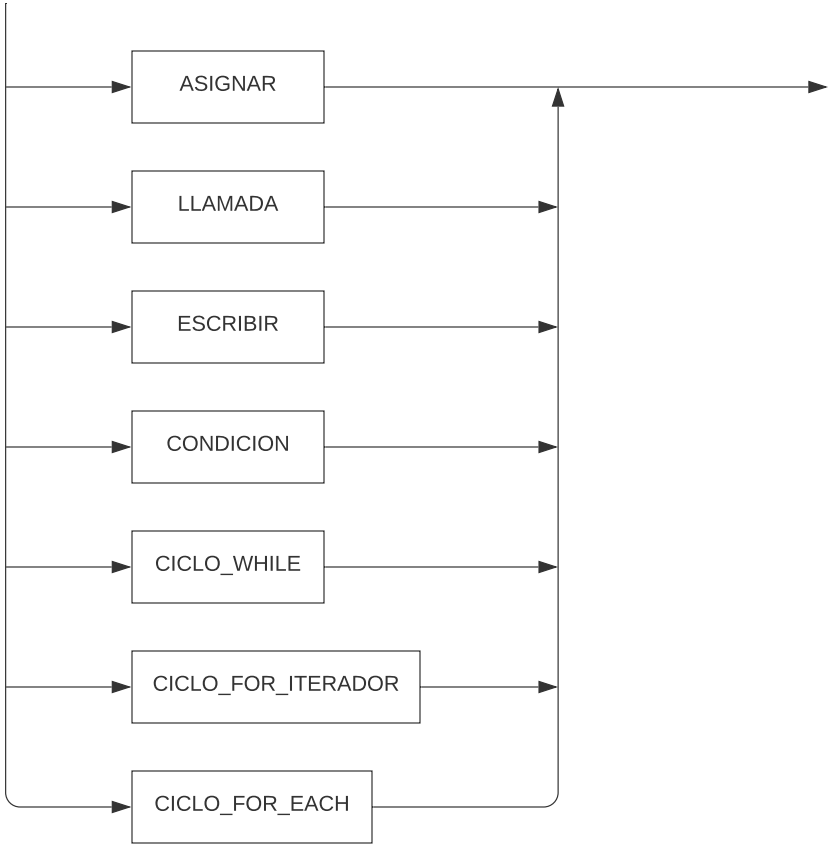
<ARRAY>



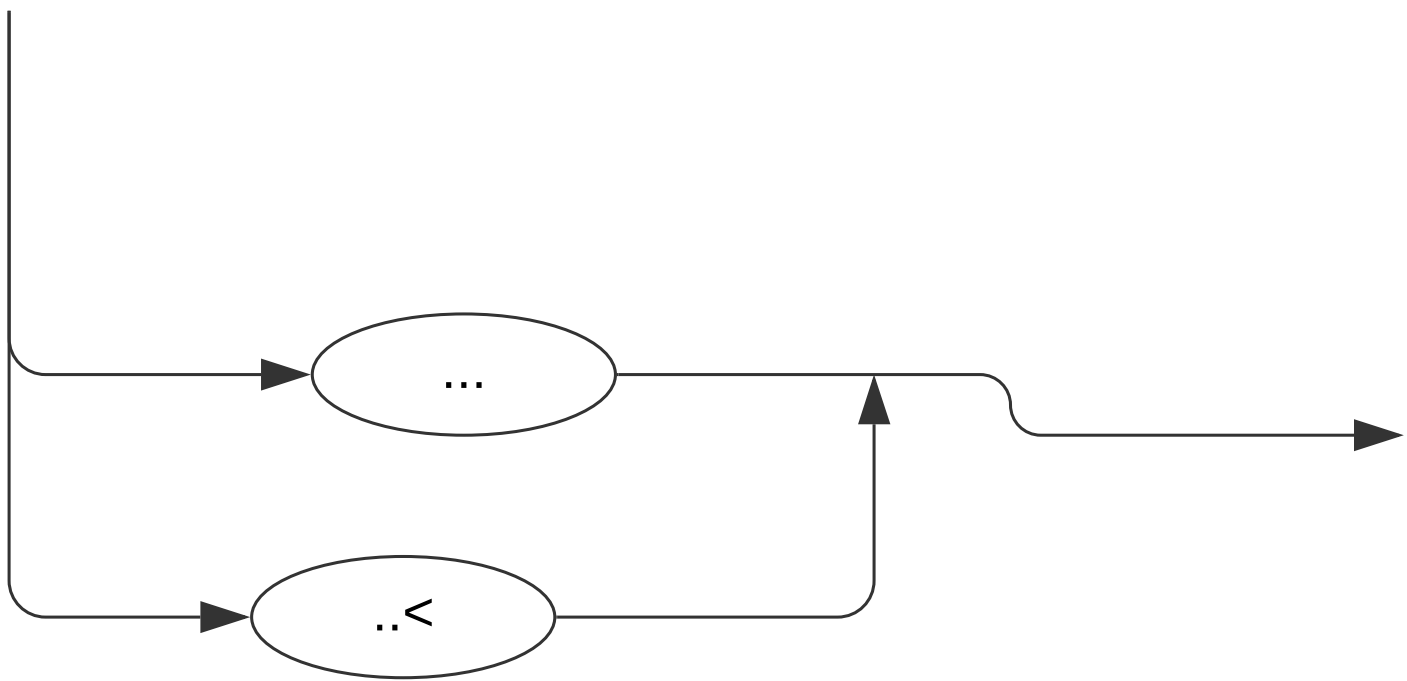
<FUNCIONES_VOID>



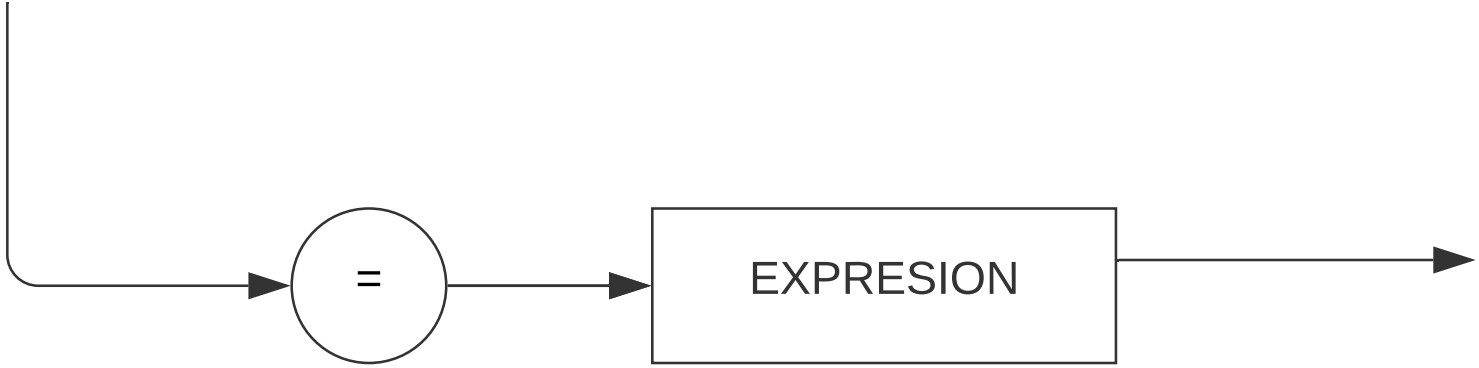
<FLUJO_BLOQUE>



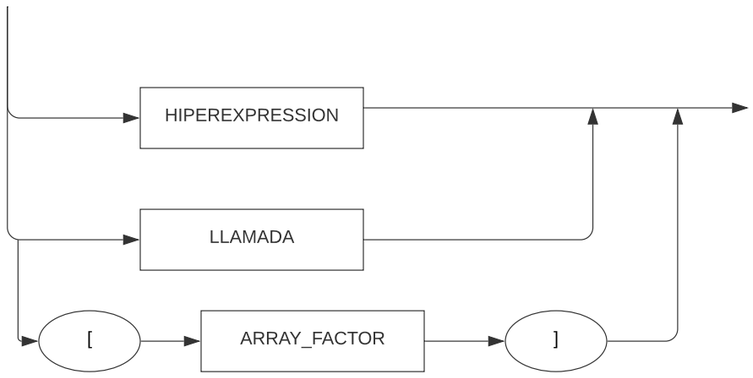
<RANGE>



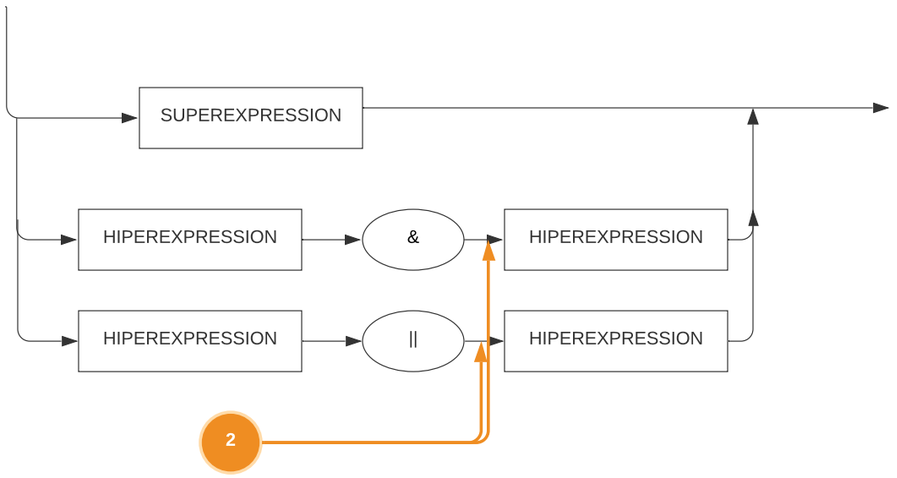
<VAR ASSIGN>



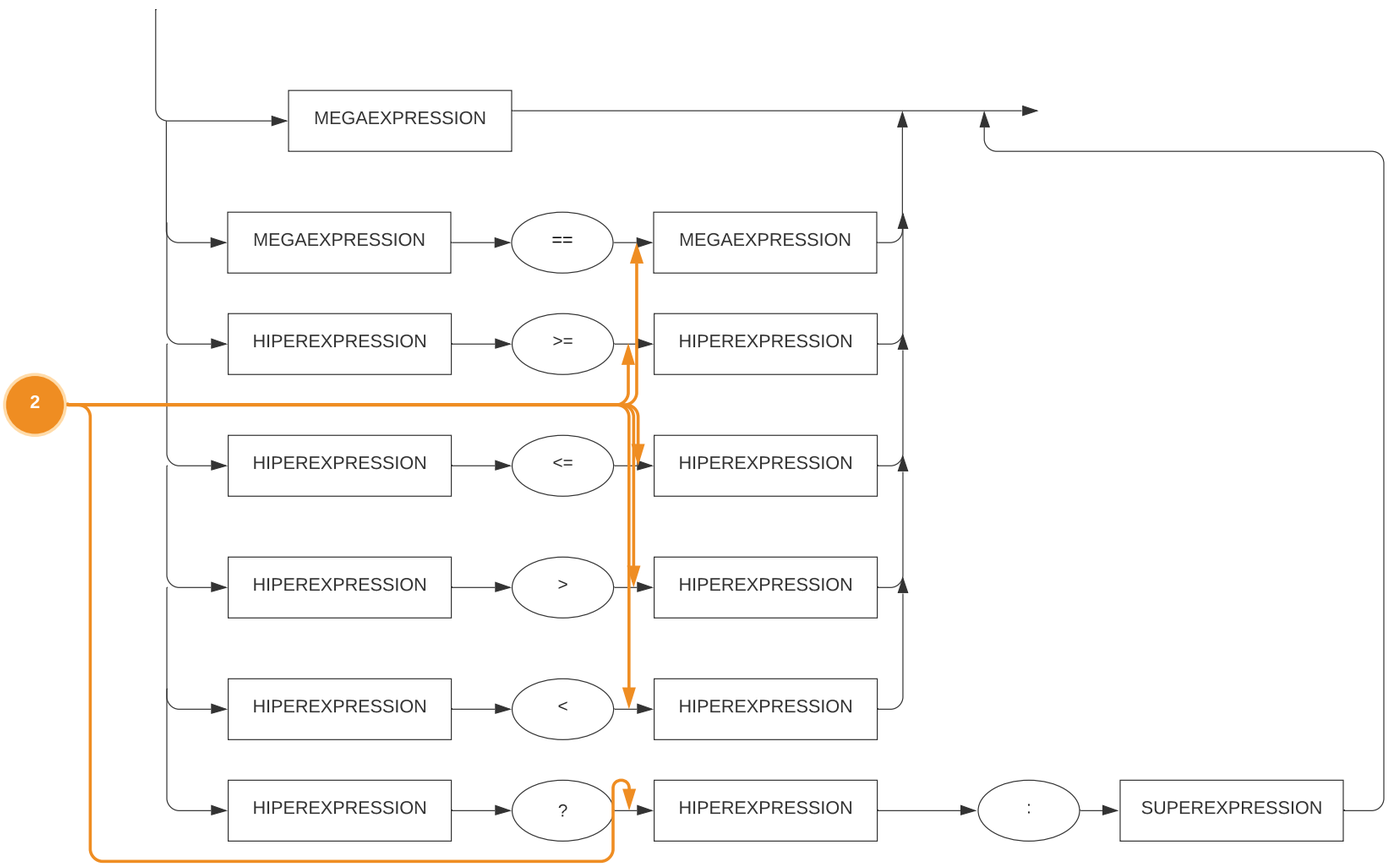
<Expression>



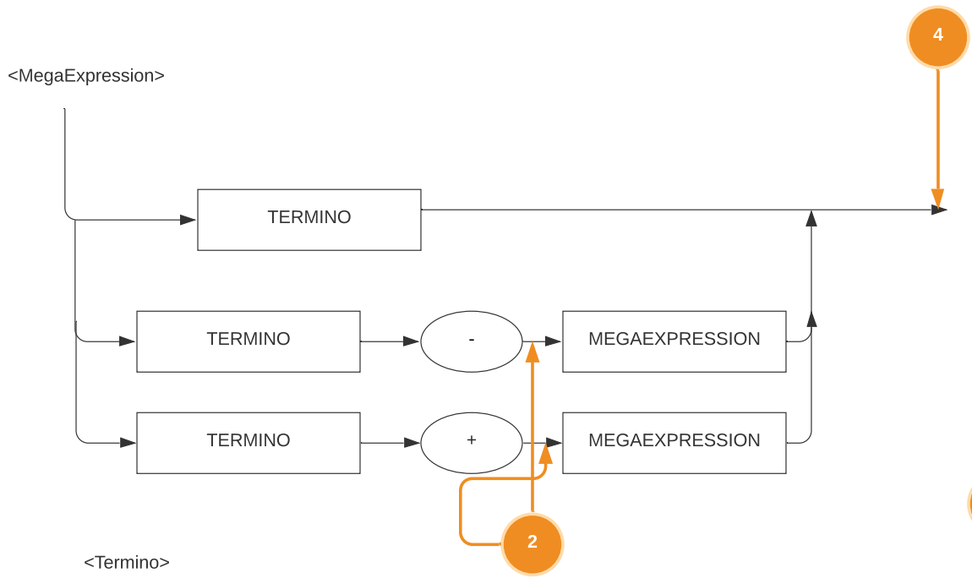
<HiperExpression>



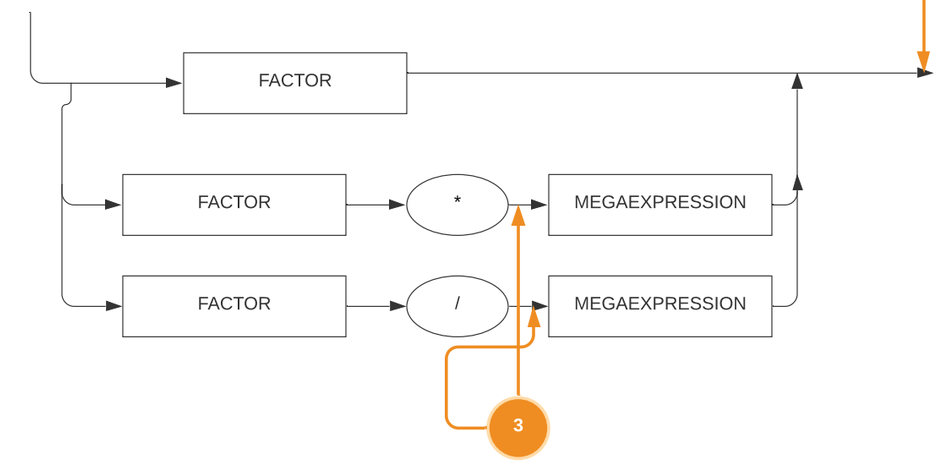
<SuperExpression>



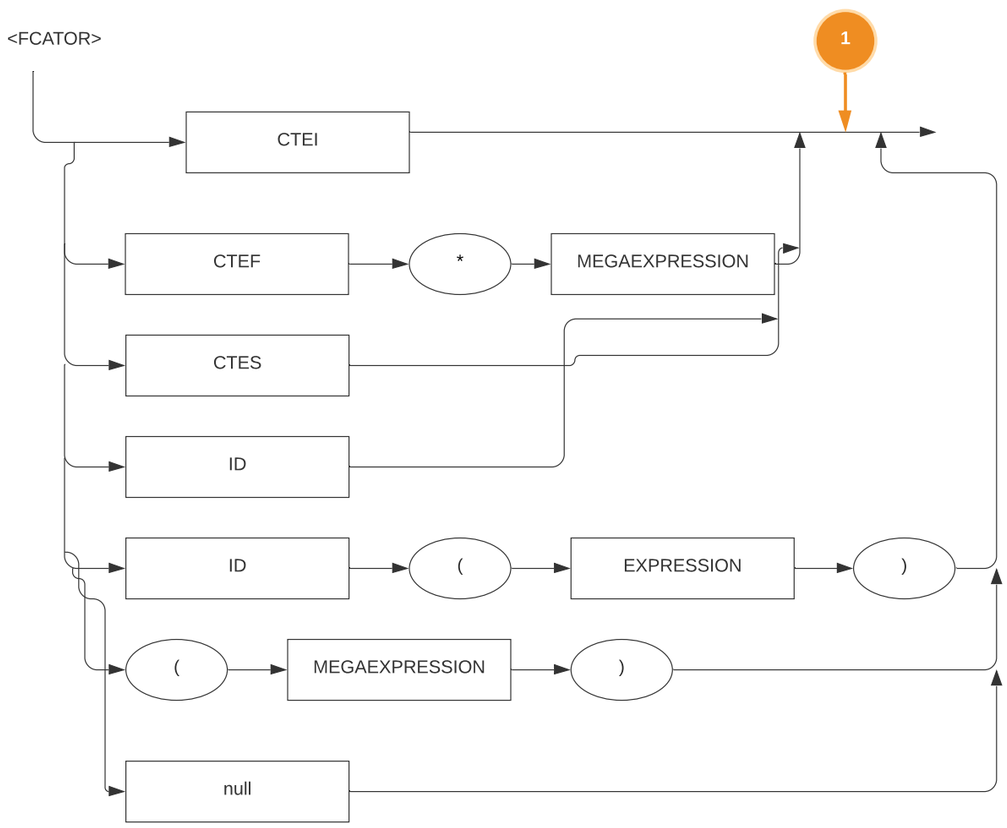
<MegaExpression>



<Termino>



<FCATOR>



1.- PilaO.Push(id.name) and PTypes.Push(id.type)
2.- POper.Push(+ or -)
3.- POper.Push(* or /)
4.- If POper.top() == '+' or '-' then
right_operand= PilaO.Pop() right_Type=PTypes.Pop()
left_operand=PilaO.Pop() let_Type=PTypes.Pop()
operator= POper.Pop()
result_Type= Semantics[left_Type,
right_Type, operator]
if (result_Type != ERROR)
result ←AVAIL.next()
generate quad
(operator, left_operand, right_operand, result)
Quad.Push(quad)
PilaO.Push(result) PTypes.Push(result_Type)
If any operand were a temporal space,
return it to AVAIL
Else
ERROR ("Type mismatch")
5.- If POper.top() == '*' or '/' then
= to #4 with *,/