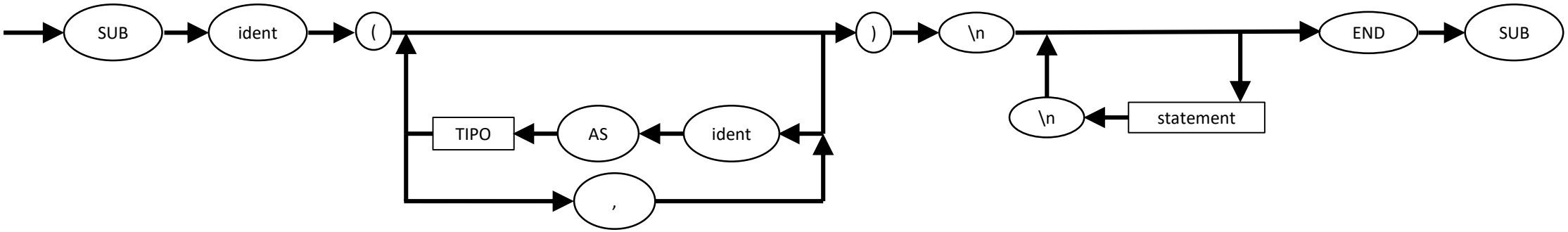
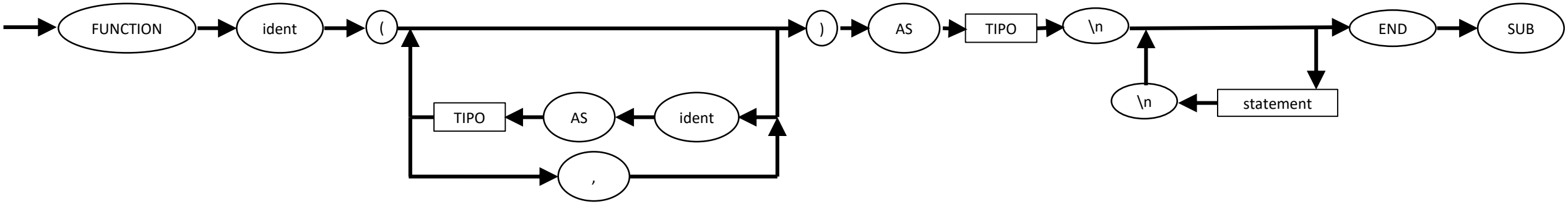
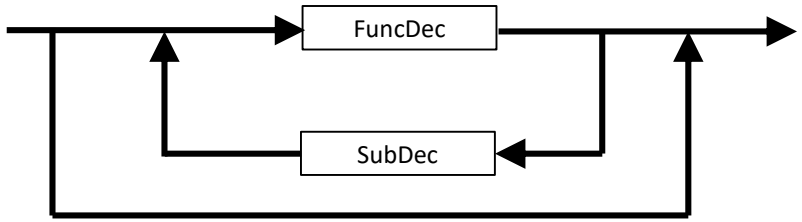
SubDec



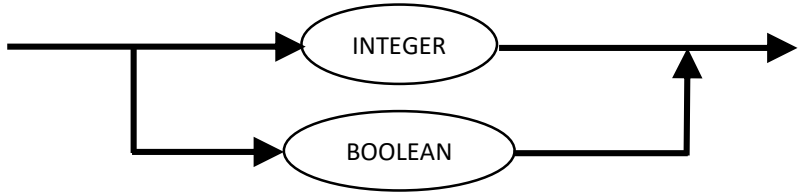
FuncDec



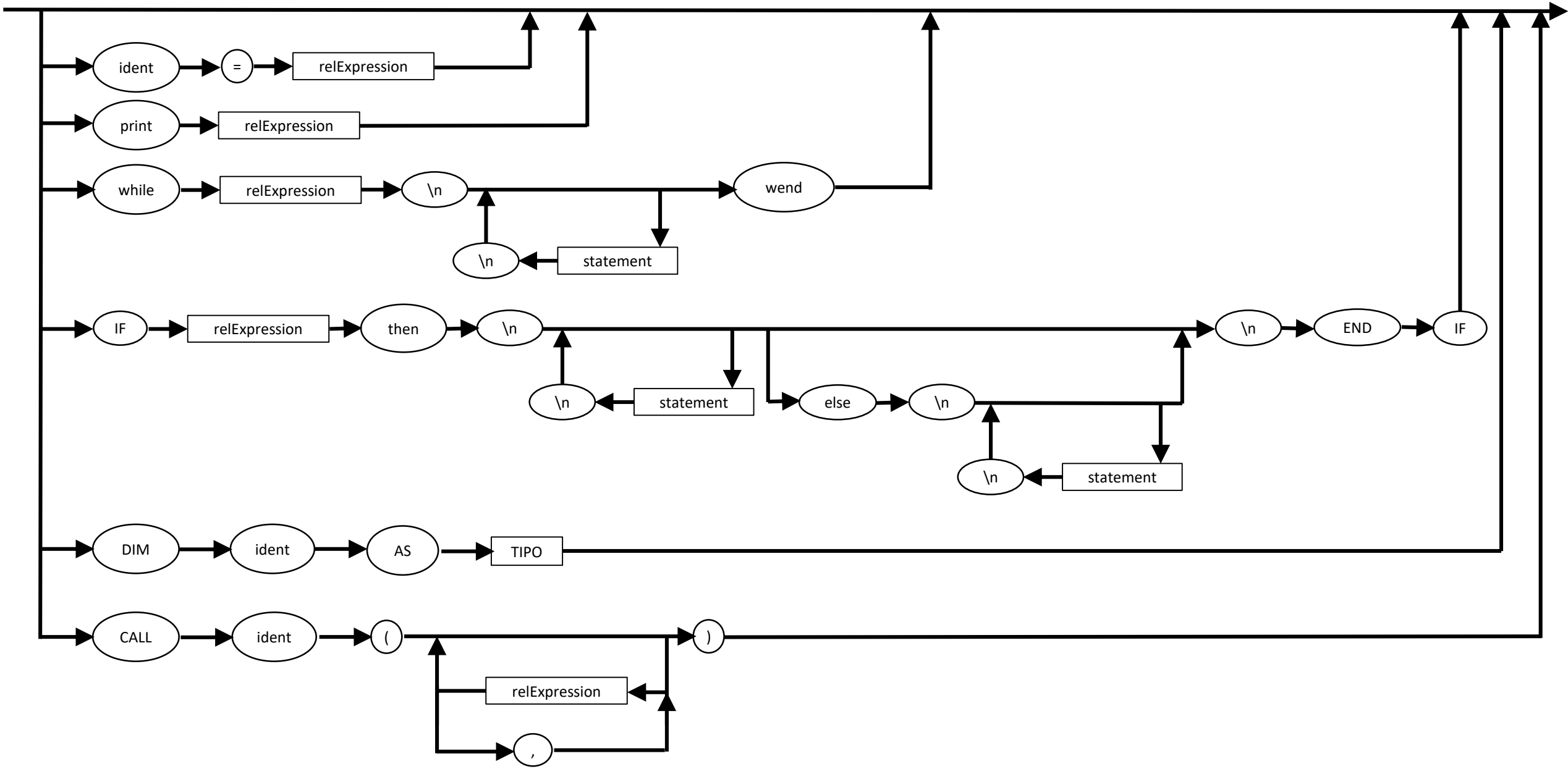
Program



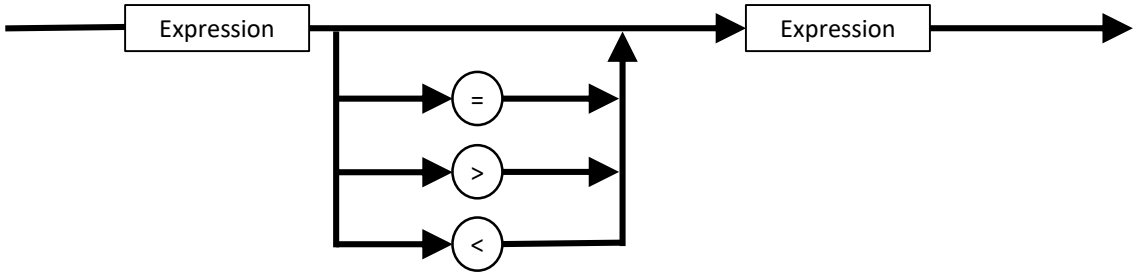
Tipo



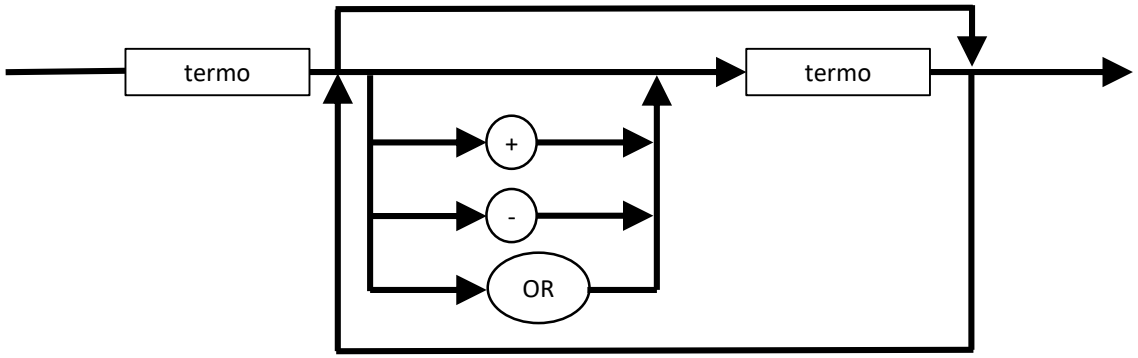
Statement


ident
=
relExpression
print
relExpression
while
relExpression
\n
wend
\n
statement
IF
relExpression
then
\n
\n
END
IF
\n
statement
else
\n
\n
statement
DIM
ident
AS
TIPO
CALL
ident
(
)
relExpression
,

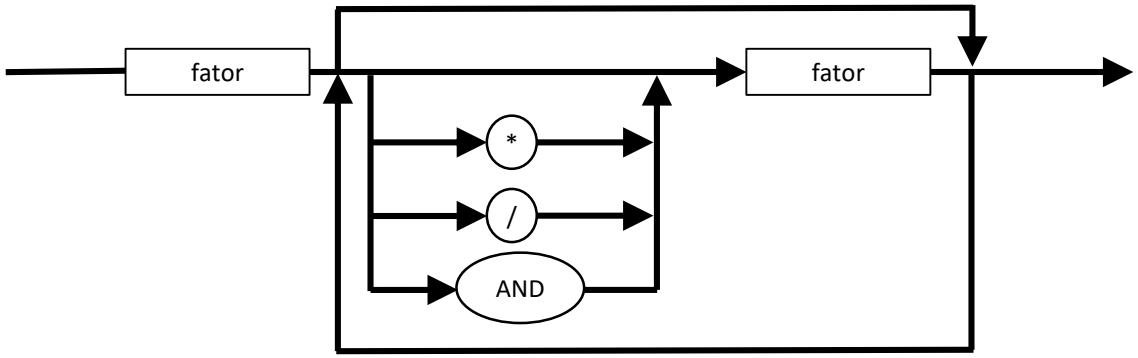
RelExpression



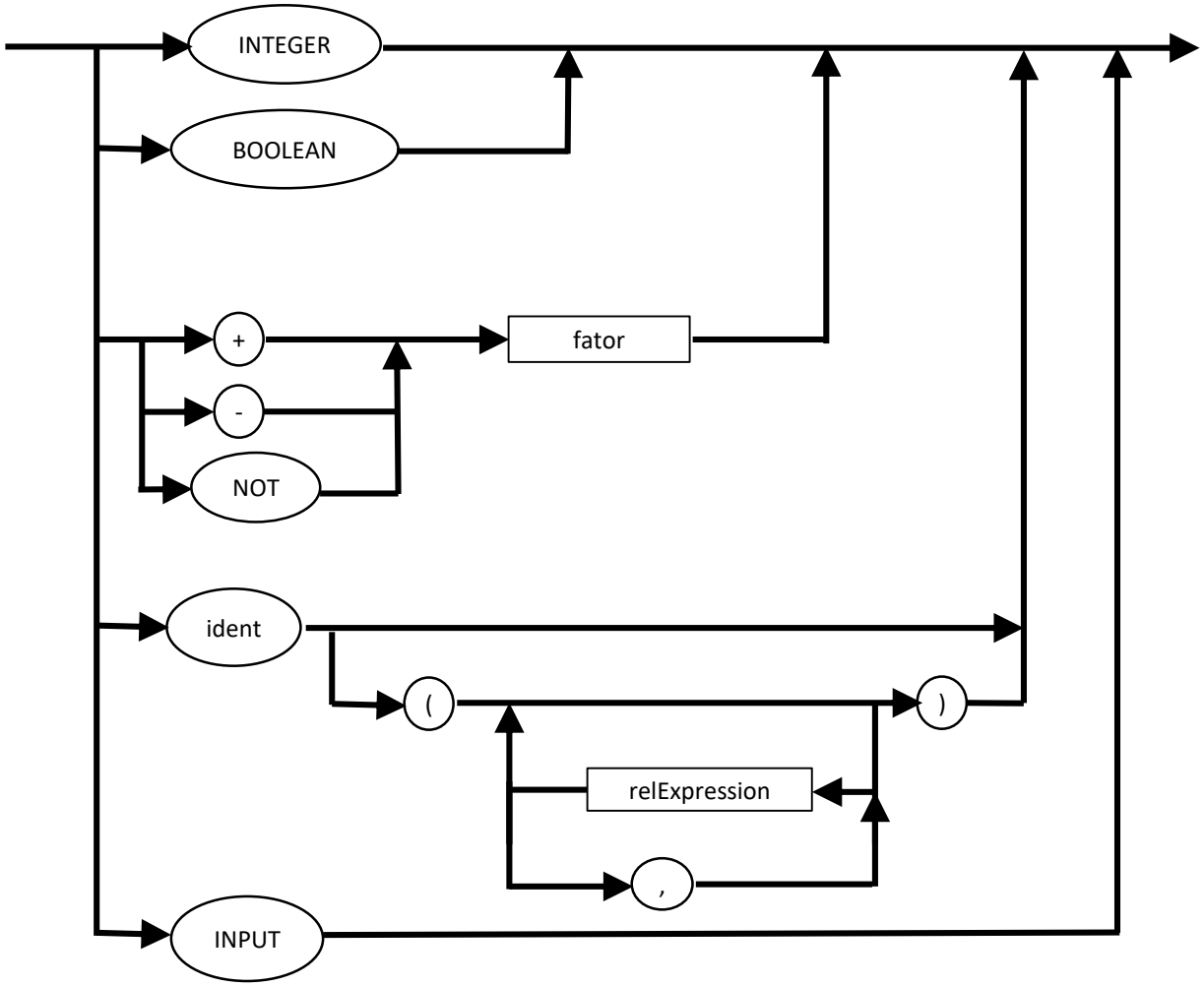
Expression



Termo



Fator



- Program = {(SubDec | FuncDec)}
- SubDec = `SUB`, `identifier`, `(`, {Null | (`identifier`, `AS`, TIPO)}, `)`, `\n`, { Null | (Statement, `\n`)}, `END`, `SUB`;
- FuncDec = `FUNCTION`, `identifier`, `(`, {Null | (`identifier`, `AS`, Tipo)}, `)`, `AS`, Tipo, `\n`, { | (Statement, `\n`)}, `END`, `FUNCTION`;
- Tipo = (`INTEGER`| `BOOLEAN`)
- Statement = Null | (`identifier`, `=`, RelExpression)
  | (`PRINT`, RelExpression)
  | (`DIM`, `identifier`, `AS`, Tipo)
  |(`WHILE`, RelExpression, `\n` {(Null | Statement, `\n`)} `WEND`)
  |(`IF`, RelExpression, `THEN`, `\n`, {Null | (Statement, `\n`), { Null | (`ELSE`, `\n`, {| (Statement, `\n`)}}, `END`, `IF`)
  | (`CALL`, `identifier`, `(`, {Null | {RelExpression, {| `,`}});
- RelExpression = Expression, {Null | (`=` | `>` | `<`)}, Expression;
- Expression = Termo, {Null | (`+` | `-` | `OR`), Termo | ;
- Termo = Fator, {Null | (`*` | `/` | `AND`), Fator} | ;
- Fator = `integer`
  | `boolean`
  | (`identifier`,{Null | {`(`{(Null | RelExpression | `,`)}`)`}})
  | {(`+` | `-` | `NOT`), Fator}
  | `INPUT`;