<Programa> → PROGRAMA id ; <decl var> <instrucciones> .

<decl var> → VAR <lista id> : <tipo> ; <decl v> |λ

<decl v> → <lista id> : <tipo> ; <decl v> |λ

<lista id> → id <resto listaid>

<resto listaid> → , <lista id> |λ

<Tipo> → <tipo std> | VECTOR [ num] de <Tipo std>

<Tipo std> → ENTERO | REAL | BOOLEANO

<instrucciones> → INICIO <lista inst> FIN

<lista inst> → <instrucción> ; <lista inst> |λ

<instrucción> → INICIO <lista instiFIN | <inst simple> | <inst e/s>

→ SI <expresion> ENTONCES <instruccion> SINO <instrucción>

→ MIENTRAS <expresión> HACER <instrucción>

<Inst simple> → id <resto instsimple>

<resto instsimple> → opasigna <expresión> | [ <expr simple> ] opasigna < expresión> |λ

<variable> → id <resto var>

<resto var> → [ <expr simplei] | λ

<inst e/s> → LEE ( id ) | ESCRIBE ( <expr simple>)

<expresión> → <expr simple> oprel <expr simple> | <expr simple>

<expr simple> → <término> <resto exsimple> | <signo> <término> <resto exsimple>

< resto exsimple> → opsuma <termino> <resto exsimple> | O <termino> <resto exsimple> |λ

<termino> → <factor> <resto term>

<resto term> → opmult <factor> <resto term> | Y <factor> <resto term> |λ

<factor> → <variable> | num | ( <expresión> ) | NO <factor> | CIERTO | FALSO