

PROGRAMA -> DECL PROGRAMA | PROCEDIMENT PROGRAMA | MAIN.

DECL -> DECL\_TUPLA ';' | DECL\_TAUOLA ';' | DECL\_TIPUS ';' .

MODIF -> MODIF\_TUPLA ';' | REM\_TUPLA ';' | AFG\_TUPLA ';' | MODIF\_TAUOLA ';' |  
REDEF\_TAUOLA ';' | REDEF\_TUPLA ';' | MODIF\_TIPUS ';' .

SENTENCIES -> DECL | MODIF | MENTRE | IF | SWITCH | FMENTRE | .

PROCEDIMENT -> metode ID '(' ARGS ')' '{' SENTENCIES '}' | metode TIPUS ID '(' ARGS ')' '{'  
SENTENCIES RETURN'}' .

RETURN -> return VALOR.

CRIDA\_PROC -> ID '(' ARGS ')' ';' .

MAIN -> metode main '(' cad arguments '[' ']' ')' '{' M\_1 '}' .

M\_1 -> SENTENCIES M\_1 | CRIDA\_PROC M\_1 | SENTENCIES.

ARGS -> ARGS\_1 | .

ARGS\_1 -> ARG ',' ARGS\_1 | ARG.

ARG -> TIPUS ID.

MENTRE -> mentre '(' L ')' fer '{' SENTENCIES '}' .

IF -> if '(' VALOR ')' '{' SENTENCIES '}' IF\_1.

IF\_1 -> else if '(' VALOR ')' '{' SENTENCIES '}' IF\_1 | else '{' SENTENCIES '}' | .

FMENTRE -> fer '(' SENTENCIES ')' mentre '(' L ')' ';' .

SWITCH -> switch '(' VALOR ')' '{' SWITCH\_1 '}' .

SWITCH\_1 -> case VALOR ':' SENTENCIES break ';' SWITCH\_1 | default ':' SENTENCIES  
break ';' | .

DECL\_TIPUS -> TIPUS A | TIPUS A\_1 .

MODIF\_TIPUS -> ID I\_1 .

TIPUS -> ent | logic | decimal | cadena | car .

A -> A\_1 I.

A\_1 -> ID ',' A\_1 | ID.

I -> '=' VALOR | '=' MODIF\_TIPUS | '=' MODIF\_TUPLA | '=' MODIF\_TAUOLA.

I\_1 -> '++' | '--' | '+=' VALOR | '-=' VALOR | I.

DECL\_TAUOLA -> TIPUS ID '[' ']' '=' NOVA\_TAUOLA | TIPUS ID '[' ']' .

NOVA\_TAUOLA -> nou ID '[' enter ']' | '{' VALORS '}' .

VALORS -> VALOR | VALOR ',' VALORS.

VALOR -> L.

MODIF\_TAULA -> ID '[' enter ']' I\_1 .  
REDEF\_TAULA -> ID '=' NOVA\_TAULA.

```
DECL_TUPLA -> tupla ID '(' ')' NOVA_TUPLA | tupla ID '(' ')' .
NOVA_TUPLA -> '=' '(' VALORS ')' .
```

```
MODIF_TUPLA -> ID '('enter')' I_1.
REM_TUPLA -> ID 'remove('VALOR')'.
AFG_TUPLA -> ID 'append('VALOR')'.
REDEF_TUPLA -> ID '=' NOVA_TUPLA.
```

E -> F OP.  
OP -> '+' F OP | '-' F OP | '\*' F OP | '/' F OP | '%' F OP | .

```
F -> ('E') | venter | vdecimal | vcharacter | '-' venter | '+' venter | '-' vdecimal | '+' vdecimal | '-'
vcharacter | '+' vcharacter | cadena | booleana | id.
```

$$L \rightarrow \text{EXPRESIO} \mid E \mid E \text{ COMP\_LOG } L \mid \text{EXPRESIO COMP\_LOG } L \mid '(' E \text{ COMP\_LOG } L ')' \mid '(' \text{EXPRESIO COMP\_LOG } L ')'$$

COMP\_LOG -> or | and .  
EXPRESIO -> NOT '('(EXPRESIO)' | '('(EXPRESIO)' | EXPRESIO\_1.

EXPRESIO\_1 -> E COMP E | NOT E COMP E | E COMP NOT E | NOT E COMP NOT E .

COMP -> '<=' .

### Prova: (Actualment Implementat)

VALOR -> L | E | EXPRESIO | '('L')'.

L -> F\_1 OPP.  
OPP -> or F\_1 OPP | or F\_1.

$$F_1 \rightarrow ('L') \mid E \mid \text{EXPRESIO.}$$

EXPRESIO -> NOT '('EXPRESIO')' | '('EXPRESIO')' | EXPRESIO\_1.

EXPRESIO\_1 -> E COMP E | NOT E COMP E | E COMP NOT E | NOT E COMP NOT E .

COMP -> '<='.

$$\begin{aligned} E &\rightarrow F \text{ OP.} \\ \text{OP} &\rightarrow '+' F \text{ OP} \mid . \end{aligned}$$
$$F \rightarrow '(\text{'E'})' \mid \text{id.}$$