

# SEGUNDA FASE: ANALIZADOR SINTÁCTICO TINY(1)

*Javier Gómez Moraleda  
Mario Quiñones Pérez  
Grupo 18*

## 1. ESPECIFICACIÓN SINTÁCTICA PARA TINY(1).

Definición de la gramática	Definiciones auxiliares
PROGRAMA $\rightarrow$ DECLARACIONES && INSTRUCCIONES PROGRAMA $\rightarrow$ INSTRUCCIONES	
DECLARACIONES $\rightarrow$ DECLARACION ; DECLARACIONES DECLARACIONES $\rightarrow$ DECLARACION	
DECLARACION $\rightarrow$ DECVAR DECLARACION $\rightarrow$ DECTIPO DECLARACION $\rightarrow$ DECPROC	
DECVAR $\rightarrow$ VAR <b>TIPO</b> ID	
DECTIPO $\rightarrow$ TYPE <b>TIPO</b> ID	
DECPROC $\rightarrow$ PROC ID <b>PFORMALES</b> <b>BLOQUE</b>	PFORMALES $\rightarrow$ ( <b>LISTA_PFORMALES</b> ) PFORMALES $\rightarrow$ ( )  LISTA_PFORMALES $\rightarrow$ <b>PFORMAL</b> , LISTA_PFORMALES LISTA_PFORMALES $\rightarrow$ <b>PFORMAL</b>  PFORMAL $\rightarrow$ <b>TIPO</b> & ID PFORMAL $\rightarrow$ <b>TIPO</b> ID
TIPO $\rightarrow$ <b>TIPO_BASICO</b>  TIPO $\rightarrow$ ID  TIPO $\rightarrow$ ARRAY CAP LIT_ENT CCIERRE OF TIPO  TIPO $\rightarrow$ RECORD LLAP <b>LISTA_CAMPOS</b> LLCIERRE  TIPO $\rightarrow$ POINTER TIPO	TIPO_BASICO $\rightarrow$ INT TIPO_BASICO $\rightarrow$ REAL TIPO_BASICO $\rightarrow$ BOOL TIPO_BASICO $\rightarrow$ STRING  LISTA_CAMPOS $\rightarrow$ CAMPO ; LISTA_CAMPOS LISTA_CAMPOS $\rightarrow$ CAMPO  CAMPO $\rightarrow$ TIPO ID
INSTRUCCIONES $\rightarrow$ INSTRUCCION ; INSTRUCCIONES	

INSTRUCCIONES $\rightarrow$ INSTRUCCION	
INSTRUCCION $\rightarrow$ <b>EXPRESION</b> = <b>EXPRESION</b>	
INSTRUCCION $\rightarrow$ IF <b>EXPRESION</b> THEN <b>LISTA_INST_OP</b> ENDIF	LISTA_INST_OP $\rightarrow$ <b>LISTA_INST</b> LISTA_INST_OP $\rightarrow \epsilon$  LISTA_INST $\rightarrow$ INSTRUCCION ; LISTA_INST LISTA_INST $\rightarrow$ INSTRUCCION
INSTRUCCION $\rightarrow$ IF <b>EXPRESION</b> THEN <b>LISTA_INST_OP</b> ELSE <b>LISTA_INST</b> ENDIF	
INSTRUCCION $\rightarrow$ WHILE <b>EXPRESION</b> DO <b>LISTA_INST_OP</b> ENDWHILE	
INSTRUCCION $\rightarrow$ READ <b>EXPRESION</b>	
INSTRUCCION $\rightarrow$ WRITE <b>EXPRESION</b>	
INSTRUCCION $\rightarrow$ NL	
INSTRUCCION $\rightarrow$ NEW <b>EXPRESION</b>	
INSTRUCCION $\rightarrow$ DELETE <b>EXPRESION</b>	
INSTRUCCION $\rightarrow$ CALL ID PAP <b>LISTA_EXPR_OP</b> PCIERRE	LISTA_EXPR_OP $\rightarrow$ <b>LISTA_EXPR</b> LISTA_EXPR_OP $\rightarrow \epsilon$  LISTA_EXPR $\rightarrow$ <b>EXPRESION</b> , LISTA_EXPR LISTA_EXPR $\rightarrow$ <b>EXPRESION</b>
INSTRUCCION $\rightarrow$ BLOQUE	BLOQUE $\rightarrow$ { PROGRAMA } BLOQUE $\rightarrow$ { $\epsilon$ }
EXPRESION $\rightarrow$ E0	EXPRESION_BASICA $\rightarrow$ true EXPRESION_BASICA $\rightarrow$ false EXPRESION_BASICA $\rightarrow$ LIT_ENT EXPRESION_BASICA $\rightarrow$ LIT_REAL EXPRESION_BASICA $\rightarrow$ LIT_CAD EXPRESION_BASICA $\rightarrow$ ID EXPRESION_BASICA $\rightarrow$ NULL
E0 $\rightarrow$ E1 + E0 E0 $\rightarrow$ E1 - E1 E0 $\rightarrow$ E1	

$E1 \rightarrow E1 \text{ OPBN1 } E2$ $E1 \rightarrow E2$	$OPBN1 \rightarrow \text{and}$ $OPBN1 \rightarrow \text{or}$
$E2 \rightarrow E2 \text{ OPBN2 } E3$ $E2 \rightarrow E3$	$OPBN2 \rightarrow <$ $OPBN2 \rightarrow >$ $OPBN2 \rightarrow <=$ $OPBN2 \rightarrow >=$ $OPBN2 \rightarrow ==$ $OPBN2 \rightarrow !=$
$E3 \rightarrow E4 \text{ OPBN3 } E4$ $E3 \rightarrow E4$	$OPBN3 \rightarrow *$ $OPBN3 \rightarrow /$ $OPBN3 \rightarrow \%$
$E4 \rightarrow - E5$ $E4 \rightarrow \text{not } E4$ $E4 \rightarrow E5$	
$E5 \rightarrow E5 \text{ OPUN5}$ $E5 \rightarrow E6$	$OPUN5 \rightarrow [ \text{EXPRESION} ]$ $OPUN5 \rightarrow OP\_ACCESO$  $OP\_ACCESO \rightarrow . ID$ $OP\_ACCESO \rightarrow -> ID$
$E6 \rightarrow * E6$ $E6 \rightarrow E7$	
$E7 \rightarrow \text{EXPRESION\_BASICA}$ $E7 \rightarrow ( E0 )$	

## 2. ACONDICIONAMIENTO DE LA GRAMÁTICA PARA LA IMPLEMENTACIÓN DE UN ANALIZADOR SINTÁCTICO DESCENDENTE PREDICTIVO RECURSIVO.

Definición de la gramática	Definiciones auxiliares
PROGRAMA $\rightarrow$ DECLARACIONES && INSTRUCCIONES PROGRAMA $\rightarrow$ INSTRUCCIONES	
DECLARACIONES $\rightarrow$ DECLARACION <b>RDECLARACIONES</b>	RDECLARACIONES $\rightarrow$ ; DECLARACION RDECLARACIONES RDECLARACIONES $\rightarrow \epsilon$
DECLARACION $\rightarrow$ DECVAR DECLARACION $\rightarrow$ DECTIPO DECLARACION $\rightarrow$ DECPROC	
DECVAR $\rightarrow$ VAR <b>TIPO</b> ID	
DECTIPO $\rightarrow$ TYPE <b>TIPO</b> ID	
DECPROC $\rightarrow$ PROC ID <b>PFORMALES</b> <b>BLOQUE</b>	PFORMALES $\rightarrow$ ( <b>RPFORMALES</b>  RPFORMALES $\rightarrow$ <b>LISTA_PFORMALES</b> ) RPFORMALES $\rightarrow$ )  LISTA_PFORMALES $\rightarrow$ <b>PFORMAL</b> <b>RLISTA_PFORMALES</b>  RLISTA_PFORMALES $\rightarrow$ , <b>PFORMAL</b> RLISTA_PFORMALES RLISTA_PFORMALES $\rightarrow \epsilon$  PFORMAL $\rightarrow$ <b>TIPO RPFORMAL</b>  RPFORMAL $\rightarrow$ & ID RPFORMAL $\rightarrow$ ID
TIPO $\rightarrow$ <b>TIPO_BASICO</b>  TIPO $\rightarrow$ ID  TIPO $\rightarrow$ ARRAY CAP LIT_ENT CCIERRE OF TIPO	TIPO_BASICO $\rightarrow$ INT TIPO_BASICO $\rightarrow$ REAL TIPO_BASICO $\rightarrow$ BOOL TIPO_BASICO $\rightarrow$ STRING  LISTA_CAMPOS $\rightarrow$ CAMPO

TIPO $\rightarrow$ RECORD LLAP <b>LISTA_CAMPOS</b> LLCIERRE  TIPO $\rightarrow$ POINTER TIPO	RLISTA_CAMPOS  RLISTA_CAMPOS $\rightarrow$ ; CAMPO RLISTA_CAMPOS RLISTA_CAMPOS $\rightarrow \epsilon$  CAMPO $\rightarrow$ TIPO ID
INSTRUCCIONES $\rightarrow$ INSTRUCCION <b>RINST</b>	RINST $\rightarrow$ ; INSTRUCCION RINST RINST $\rightarrow \epsilon$
INSTRUCCION $\rightarrow$ <b>EXPRESION = EXPRESION</b>	
INSTRUCCION $\rightarrow$ IF <b>EXPRESION</b> THEN <b>LISTA_INST_OP RIFTHEN</b> ENDIF	LISTA_INST_OP $\rightarrow$ LISTA_INST LISTA_INST_OP $\rightarrow \epsilon$  LISTA_INST $\rightarrow$ INSTRUCCION <b>RLISTA_INST</b>  RLISTA_INST $\rightarrow$ ; INSTRUCCION RLISTA_INST RLISTA_INST $\rightarrow \epsilon$  RIFTHEN $\rightarrow$ ELSE <b>LISTA_INST</b> RIFTHEN $\rightarrow \epsilon$
INSTRUCCION $\rightarrow$ WHILE <b>EXPRESION</b> DO <b>LISTA_INST_OP</b> ENDWHILE	
INSTRUCCION $\rightarrow$ READ <b>EXPRESION</b>	
INSTRUCCION $\rightarrow$ WRITE <b>EXPRESION</b>	
INSTRUCCION $\rightarrow$ NL	
INSTRUCCION $\rightarrow$ NEW <b>EXPRESION</b>	
INSTRUCCION $\rightarrow$ DELETE <b>EXPRESION</b>	
INSTRUCCION $\rightarrow$ CALL ID PAP <b>LISTA_EXPR_OP</b> PCIERRE	LISTA_EXPR_OP $\rightarrow$ <b>LISTA_EXPR</b> LISTA_EXPR_OP $\rightarrow \epsilon$  LISTA_EXPR $\rightarrow$ <b>EXPRESION</b> <b>RLISTA_EXPR</b>

	$RLISTA\_EXPR \rightarrow , LISTA\_EXPR$ $RLISTA\_EXPR$ $RLISTA\_EXPR \rightarrow \varepsilon$
$INSTRUCCION \rightarrow BLOQUE$	$BLOQUE \rightarrow \{ \textbf{RBLOQUE}$ $RBLOQUE \rightarrow PROGRAMA \}$ $RBLOQUE \rightarrow \}$
$EXPRESION \rightarrow E0$	$EXPRESION\_BASICA \rightarrow \text{true}$ $EXPRESION\_BASICA \rightarrow \text{false}$ $EXPRESION\_BASICA \rightarrow LIT\_ENT$ $EXPRESION\_BASICA \rightarrow LIT\_REAL$ $EXPRESION\_BASICA \rightarrow LIT\_CAD$ $EXPRESION\_BASICA \rightarrow ID$ $EXPRESION\_BASICA \rightarrow \text{NULL}$
$E0 \rightarrow E1 RE0$ $RE0 \rightarrow + E0$ $RE0 \rightarrow - E1$ $RE0 \rightarrow \varepsilon$	
$E1 \rightarrow E2 RE1$ $RE1 \rightarrow \textbf{OPBN1} E2 RE1$ $RE1 \rightarrow \varepsilon$	$OPBN1 \rightarrow \text{and}$ $OPBN1 \rightarrow \text{or}$
$E2 \rightarrow E3 RE2$ $RE2 \rightarrow \textbf{OPBN2} E3 RE2$ $RE2 \rightarrow \varepsilon$	$OPBN2 \rightarrow <$ $OPBN2 \rightarrow >$ $OPBN2 \rightarrow <=$ $OPBN2 \rightarrow >=$ $OPBN2 \rightarrow ==$ $OPBN2 \rightarrow !=$
$E3 \rightarrow E4 RE3$ $RE3 \rightarrow \textbf{OPBN3} E4$ $RE3 \rightarrow \varepsilon$	$OPBN3 \rightarrow *$ $OPBN3 \rightarrow /$ $OPBN3 \rightarrow \%$
$E4 \rightarrow \textbf{not} E4$ $E4 \rightarrow - E5$ $E4 \rightarrow E5$	
$E5 \rightarrow E6 RE5$	$OPUN5 \rightarrow [ EXPRESION ]$

RE5 $\rightarrow$ <b>OPUN5</b> RE5 RE5 $\rightarrow \epsilon$	OPUN5 $\rightarrow$ OP_ACCESO OP_ACCESO $\rightarrow$ . ID OP_ACCESO $\rightarrow$ -> ID
E6 $\rightarrow$ * E6 E6 $\rightarrow$ E7	
E7 $\rightarrow$ <b>EXPRESION_BASICA</b> E7 $\rightarrow$ ( E0 )	