

## **Fase 2: Desarrollo de analizador sintáctico para Tiny(0) y Tiny**

G01

Esther Babon Arcauz

Pablo Campo Gómez

Claudia López-Mingo Moreno

José Antonio Ruiz Heredia

## Índice:

<b>1. Desarrollo manual de un analizador sintáctico para Tiny(0).</b>	<b>2</b>
1.1. Especificación sintáctica (gramática) para Tiny(0).	2
1.2. Acondicionamiento de la gramática para permitir la implementación de un analizador sintáctico descendente predictivo recursivo.	3
1.3. Directores de cada regla de la gramática acondicionada.	5
<b>2. Desarrollo de analizadores sintácticos descendentes y ascendentes para Tiny.</b>	<b>8</b>
2.1. Especificación sintáctica (gramática) para Tiny.	8
2.2. Acondicionamiento de la gramática para permitir la implementación de un analizador sintáctico descendente predictivo recursivo.	11

# 1. Desarrollo manual de un analizador sintáctico para Tiny(0).

## 1.1. Especificación sintáctica (gramática) para Tiny(0).

programa  $\rightarrow$  bloque EOF

bloque  $\rightarrow$  { declaraciones\_opt instrucciones\_opt }

declaraciones\_opt  $\rightarrow$  declaraciones &&

declaraciones\_opt  $\rightarrow \epsilon$

instrucciones\_opt  $\rightarrow$  instrucciones

instrucciones\_opt  $\rightarrow \epsilon$

declaraciones  $\rightarrow$  declaracion\_var ; declaraciones

declaraciones  $\rightarrow$  declaracion\_var

declaracion\_var  $\rightarrow$  tipo identificador

tipo  $\rightarrow$  int

tipo  $\rightarrow$  real

tipo  $\rightarrow$  bool

instrucciones  $\rightarrow$  instrucciones ; instruccion

instrucciones  $\rightarrow$  instruccion

instruccion  $\rightarrow$  @ E0

E0  $\rightarrow$  E1 = E0

E0  $\rightarrow$  E1

E1  $\rightarrow$  E1 op1 E2

E1  $\rightarrow$  E2

E2  $\rightarrow$  E2 + E3

E2  $\rightarrow$  E3 - E3

E2  $\rightarrow$  E3

E3  $\rightarrow$  E4 op3

E3  $\rightarrow$  E4

E4  $\rightarrow$  E4 op4 E5

E4  $\rightarrow$  E5

E5  $\rightarrow$  op5 E5

E5  $\rightarrow$  literalEntero

E5  $\rightarrow$  literalReal

E5  $\rightarrow$  literalBool

E5  $\rightarrow$  variable

E5  $\rightarrow$  ( E0 )

op1  $\rightarrow$  >

$op1 \rightarrow >=$   
 $op1 \rightarrow <$   
 $op1 \rightarrow <=$   
 $op1 \rightarrow ==$   
 $op1 \rightarrow !=$   
 $op3 \rightarrow \text{and } E3$   
 $op3 \rightarrow \text{or } E4$   
 $op4 \rightarrow *$   
 $op4 \rightarrow /$   
 $op5 \rightarrow -$   
 $op5 \rightarrow \text{not}$

## 1.2. Acondicionamiento de la gramática para permitir la implementación de un analizador sintáctico descendente predictivo recursivo.

Original	Acondicionado
programa $\rightarrow$ bloque EOF	
bloque $\rightarrow \{ \text{declaraciones\_opt instrucciones\_opt} \}$	
declaraciones_opt $\rightarrow$ declaraciones && declaraciones_opt $\rightarrow \epsilon$	
instrucciones_opt $\rightarrow$ instrucciones instrucciones_opt $\rightarrow \epsilon$	
declaraciones $\rightarrow$ declaracion_var ; declaraciones declaraciones $\rightarrow$ declaracion_var	declaraciones $\rightarrow$ declaracion_var restodec restodec $\rightarrow$ ; declaraciones restodec $\rightarrow \epsilon$
declaracion_var $\rightarrow$ tipo identificador	
tipo $\rightarrow$ int tipo $\rightarrow$ real tipo $\rightarrow$ bool	
instrucciones $\rightarrow$ instrucciones ; instruccion instrucciones $\rightarrow$ instruccion	Instrucciones $\rightarrow$ Instruccion Instrucciones' Instrucciones' $\rightarrow$ ; Instruccion Instrucciones' Instrucciones' $\rightarrow \epsilon$
instruccion $\rightarrow$ @ E0	
$E0 \rightarrow E1 = E0$	$E0 \rightarrow E1 RE0$

$E0 \rightarrow E1$	$RE0 \rightarrow = E0$ $RE0 \rightarrow \varepsilon$
$E1 \rightarrow E1 \text{ op1 } E2$ $E1 \rightarrow E2$	$E1 \rightarrow E2 RE1$ $RE1 \rightarrow \text{op1 } E2 RE1$ $RE1 \rightarrow \varepsilon$
$E2 \rightarrow E2 + E3$ $E2 \rightarrow E3 - E3$ $E2 \rightarrow E3$	$E2 \rightarrow E3 RE2 E2'$ $RE2 \rightarrow - E3$ $RE2 \rightarrow \varepsilon$ $E2' \rightarrow + E3 E2'$ $E2' \rightarrow \varepsilon$
$E3 \rightarrow E4 \text{ op3}$ $E3 \rightarrow E4$	$E3 \rightarrow E4 RE3$ $RE3 \rightarrow \text{op3}$ $RE3 \rightarrow \varepsilon$
$E4 \rightarrow E4 \text{ op4 } E5$ $E4 \rightarrow E5$	$E4 \rightarrow E5 RE4$ $RE4 \rightarrow \text{op4 } E5 RE4$ $RE4 \rightarrow \varepsilon$
$E5 \rightarrow \text{op5 } E5$ $E5 \rightarrow \text{literalEntero}$ $E5 \rightarrow \text{literalReal}$ $E5 \rightarrow \text{literalBool}$ $E5 \rightarrow \text{variable}$ $E5 \rightarrow ( E0 )$	
$\text{op1} \rightarrow >$ $\text{op1} \rightarrow >=$ $\text{op1} \rightarrow <$ $\text{op1} \rightarrow <=$ $\text{op1} \rightarrow ==$ $\text{op1} \rightarrow !=$ $\text{op3} \rightarrow \text{and } E3$ $\text{op3} \rightarrow \text{or } E4$ $\text{op4} \rightarrow *$ $\text{op4} \rightarrow /$ $\text{op5} \rightarrow -$ $\text{op5} \rightarrow \text{not}$	

### 1.3. Directores de cada regla de la gramática acondicionada.

programa  $\rightarrow$  bloque EOF

- Director: {

bloque  $\rightarrow$  { declaraciones\_opt instrucciones\_opt }

- Director: {

declaraciones\_opt  $\rightarrow$  declaraciones &&

- Dir: **int, real, bool**

declaraciones\_opt  $\rightarrow \epsilon$

- Dir:  $\emptyset$

instrucciones\_opt  $\rightarrow$  instrucciones

- Dir: **@**

instrucciones\_opt  $\rightarrow \epsilon$

- Dir:  $\emptyset$

declaraciones  $\rightarrow$  declaracion\_var restodec

- Dir: **int, real, bool**

restodec  $\rightarrow$  ; declaraciones

- Dir: ;

restodec  $\rightarrow \epsilon$

- Dir:  $\emptyset$

declaracion\_var  $\rightarrow$  tipo identificador

- Dir: **int, real, bool**

tipo  $\rightarrow$  int

- Dir: **int**

tipo  $\rightarrow$  real

- Dir: **real**

tipo  $\rightarrow$  bool

- Dir: **bool**

Instrucciones  $\rightarrow$  Instruccion Instrucciones'

- Dir: @

Instrucciones'  $\rightarrow$  ; Instruccion Instrucciones'

- Dir: ;

Instrucciones'  $\rightarrow \epsilon$

- Dir:  $\emptyset$

instruccion  $\rightarrow$  @ E0

- Director: @

E0  $\rightarrow$  E1 RE0

- Dir: -,not, literalEntero, literalReal, literalBool, variable, (

RE0  $\rightarrow$  = E0

- Dir: =

RE0  $\rightarrow \epsilon$

- Dir:  $\emptyset$

E1  $\rightarrow$  E2 RE1

- Dir: -,not, literalEntero, literalReal, literalBool, variable, (

RE1  $\rightarrow$  op1 E2 RE1

- Dir: >, >=, <, <=, ==, !=

RE1  $\rightarrow \epsilon$

- Dir:  $\emptyset$

E2  $\rightarrow$  E3 RE2 E2'

- Dir: -, not, literalEntero, literalReal, literalBool, variable, (

RE2  $\rightarrow$  - E3

- Dir: -

RE2  $\rightarrow \epsilon$

- Dir:  $\emptyset$

E2'  $\rightarrow$  + E3 E2'

- Dir: +

E2'  $\rightarrow \epsilon$

- Dir:  $\emptyset$

$E3 \rightarrow E4 RE3$

- Dir: -, **not**, **literalEntero**, **literalReal**, **literalBool**, **variable**, (

$RE3 \rightarrow op3$

- Dir: **and**, **or**

$RE3 \rightarrow \varepsilon$

- Dir:  $\emptyset$

$E4 \rightarrow E5 RE4$

- Dir: -, **not**, **literalEntero**, **literalReal**, **literalBool**, **variable**, (

$RE4 \rightarrow op4 E5 RE4$

- Dir: \*, /

$RE4 \rightarrow \varepsilon$

- Dir:  $\emptyset$

$E5 \rightarrow op5 E5$

- Dir: -, **not**

$E5 \rightarrow \text{literalEntero}$

- Dir: **literalEntero**

$E5 \rightarrow \text{literalReal}$

- Dir: **literalReal**

$E5 \rightarrow \text{literalBool}$

- Dir: **literalBool**

$E5 \rightarrow \text{variable}$

- Dir: **variable**

$E5 \rightarrow ( E0 )$

- Dir: (

$op1 \rightarrow >$

- Dir: >

$op1 \rightarrow >=$

- Dir: >=

$op1 \rightarrow <$

- Dir: <



op1  $\rightarrow$  <=

- Dir: <=

op1  $\rightarrow$  ==

- Dir: ==

op1  $\rightarrow$  !=

- Dir: !=

op3  $\rightarrow$  and E3

- Dir: **and**

op3  $\rightarrow$  or E4

- Dir: **or**

op4  $\rightarrow$  \*

- Dir: \*

op4  $\rightarrow$  /

- Dir: /

op5  $\rightarrow$  -

- Dir: -

op5  $\rightarrow$  not:

- Dir: **not**

## 2. Desarrollo de analizadores sintácticos descendentes y ascendentes para Tiny.

### 2.1. Especificación sintáctica (gramática) para Tiny.

programa  $\rightarrow$  bloque EOF

bloque  $\rightarrow$  { declaraciones\_opt instrucciones\_opt }

declaraciones\_opt  $\rightarrow$  declaraciones &&

declaraciones\_opt  $\rightarrow$   $\epsilon$

instrucciones\_opt  $\rightarrow$  instrucciones  
instrucciones\_opt  $\rightarrow \epsilon$

declaraciones  $\rightarrow$  declaracion ; declaraciones  
declaraciones  $\rightarrow$  declaracion

declaracion  $\rightarrow$  declaracion\_var  
declaracion  $\rightarrow$  declaracion\_tipo  
declaracion  $\rightarrow$  declaracion\_proc

declaracion\_var  $\rightarrow$  tipo identificador  
declaracion\_tipo  $\rightarrow$  type tipo identificador  
declaracion\_proc  $\rightarrow$  proc identificador ( parametros\_formales\_opt ) bloque

parametros\_formales\_opt  $\rightarrow$  parametros\_formales  
parametros\_formales\_opt  $\rightarrow \epsilon$

parametros\_formales  $\rightarrow$  parametros\_formales , parametro\_formal  
parametros\_formales  $\rightarrow$  parametro\_formal

parametro\_formal  $\rightarrow$  tipo and\_opt identificador  
and\_opt  $\rightarrow$  &  
and\_opt  $\rightarrow \epsilon$

tipo  $\rightarrow$  tipo [ literalEntero ]  
tipo  $\rightarrow$  tipo1  
tipo1  $\rightarrow$  ^ tipo1  
tipo1  $\rightarrow$  tipo2  
tipo2  $\rightarrow$  identificador  
tipo2  $\rightarrow$  struct { campos }  
tipo2  $\rightarrow$  int  
tipo2  $\rightarrow$  real  
tipo2  $\rightarrow$  bool  
tipo2  $\rightarrow$  string

campos  $\rightarrow$  campos , campo  
campos  $\rightarrow$  campo

campo  $\rightarrow$  tipo identificador

instrucciones  $\rightarrow$  instrucciones ; instruccion  
instrucciones  $\rightarrow$  instruccion

instruccion  $\rightarrow$  @ E0  
 instruccion  $\rightarrow$  if E0 bloque  
 instruccion  $\rightarrow$  if E0 bloque else bloque  
 instruccion  $\rightarrow$  while E0 bloque  
 instruccion  $\rightarrow$  read E0  
 instruccion  $\rightarrow$  write E0  
 instruccion  $\rightarrow$  nl  
 instruccion  $\rightarrow$  new E0  
 instruccion  $\rightarrow$  delete E0  
 instruccion  $\rightarrow$  call identificador ( parametros\_reales\_opt )  
 instruccion  $\rightarrow$  bloque

parametros\_reales\_opt  $\rightarrow$  parametros\_reales  
 parametros\_reales\_opt  $\rightarrow$   $\epsilon$

parametros\_reales  $\rightarrow$  parametros\_reales , E0  
 parametros\_reales  $\rightarrow$  E0

E0  $\rightarrow$  E1 = E0  
 E0  $\rightarrow$  E1  
 E1  $\rightarrow$  E1 op1 E2  
 E1  $\rightarrow$  E2  
 E2  $\rightarrow$  E2 + E3  
 E2  $\rightarrow$  E3 - E3  
 E2  $\rightarrow$  E3  
 E3  $\rightarrow$  E4 op3  
 E3  $\rightarrow$  E4  
 E4  $\rightarrow$  E4 op4 E5  
 E4  $\rightarrow$  E5  
 E5  $\rightarrow$  op5 E5  
 E5  $\rightarrow$  E6  
 E6  $\rightarrow$  E6 op6  
 E6  $\rightarrow$  literalEntero  
 E6  $\rightarrow$  literalReal  
 E6  $\rightarrow$  literalBool  
 E6  $\rightarrow$  literalCadena  
 E6  $\rightarrow$  identificador  
 E6  $\rightarrow$  null  
 E6  $\rightarrow$  ( E0 )  
 op1  $\rightarrow$  >  
 op1  $\rightarrow$  >=  
 op1  $\rightarrow$  <  
 op1  $\rightarrow$  <=

op1  $\rightarrow$  ==  
 op1  $\rightarrow$  !=  
 op3  $\rightarrow$  and E3  
 op3  $\rightarrow$  or E4  
 op4  $\rightarrow$  \*  
 op4  $\rightarrow$  /  
 op4  $\rightarrow$  %  
 op5  $\rightarrow$  -  
 op5  $\rightarrow$  not  
 op6  $\rightarrow$  [ literalEntero ]  
 op6  $\rightarrow$  .campo  
 op6  $\rightarrow$  ^

## 2.2. Acondicionamiento de la gramática para permitir la implementación de un analizador sintáctico descendente predictivo recursivo.

Original	Acondicionado
programa $\rightarrow$ bloque EOF	
bloque $\rightarrow$ { declaraciones_opt instrucciones_opt } declaraciones_opt $\rightarrow$ declaraciones && declaraciones_opt $\rightarrow$ $\epsilon$ instrucciones_opt $\rightarrow$ instrucciones instrucciones_opt $\rightarrow$ $\epsilon$	
declaraciones $\rightarrow$ declaracion ; declaraciones declaraciones $\rightarrow$ declaracion declaracion $\rightarrow$ declaracion_var declaracion $\rightarrow$ declaracion_tipo declaracion $\rightarrow$ declaracion_proc	declaraciones $\rightarrow$ declaracion declaracion_extra declaracion_extra $\rightarrow$ ; declaraciones declaracion_extra $\rightarrow$ $\epsilon$ declaracion $\rightarrow$ declaracion_var declaracion $\rightarrow$ declaracion_tipo declaracion $\rightarrow$ declaracion_proc
declaracion_var $\rightarrow$ tipo identificador declaracion_tipo $\rightarrow$ type tipo identificador declaracion_proc $\rightarrow$ proc identificador ( parametros_formales_opt ) bloque	
parametros_formales_opt $\rightarrow$ parametros_formales parametros_formales_opt $\rightarrow$ $\epsilon$	

parametros_formales → parametros_formales , parametro_formal parametros_formales → parametro_formal	parametros_formales → parametro_formal parametros_formales' parametros_formales' → , parametro_formal parametros_formales' parametros_formales' → ε
parametro_formal → tipo and_opt identificador and_opt → & and_opt → ε	
tipo → tipo [ literalEntero ] tipo → tipo1 tipo1 → ^ tipo1 tipo1 → tipo2 tipo2 → identificador tipo2 → struct { campos } tipo2 → int tipo2 → real tipo2 → bool tipo2 → string	tipo → tipo1 tipo' tipo' → [literalEntero] tipo' tipo' → ε tipo1 → ^ tipo1 tipo1 → tipo2 tipo2 → identificador tipo2 → struct { campos } tipo2 → int tipo2 → real tipo2 → bool tipo2 → string
campos → campos , campo campos → campo	campos → campo campos' campos' → , campo campos' campos' → ε
campo → tipo identificador	
Instrucciones → instrucciones ; instruccion Instrucciones → instruccion	Instrucciones → instruccion instrucciones' instrucciones' → ; instruccion instrucciones' instrucciones' → ε
instruccion → @ E0 instruccion → if E0 bloque instruccion → if E0 bloque else bloque instruccion → while E0 bloque instruccion → read E0 instruccion → write E0 instruccion → nl instruccion → new E0 instruccion → delete E0	instruccion → if E0 bloque if_select if_select → else bloque if_else → ε instruccion → @ E0 instruccion → while E0 bloque instruccion → read E0 instruccion → write E0 instruccion → nl instruccion → new E0 instruccion → delete E0

instruccion $\rightarrow$ call identificador ( parametros_reales_opt ) instruccion $\rightarrow$ bloque	instruccion $\rightarrow$ call identificador ( parametros_reales_opt ) instruccion $\rightarrow$ bloque
parametros_reales_opt $\rightarrow$ parametros_reales parametros_reales_opt $\rightarrow \epsilon$	
parametros_reales $\rightarrow$ parametros_reales , E0 parametros_reales $\rightarrow$ E0	parametros_reales $\rightarrow$ E0 parametros_reales' parametros_reales' $\rightarrow$ , E0 parametros_reales' parametros_reales' $\rightarrow \epsilon$
E0 $\rightarrow$ E1 = E0 E0 $\rightarrow$ E1	E0 $\rightarrow$ E1 RE0 RE0 $\rightarrow$ = E0 RE0 $\rightarrow \epsilon$
E1 $\rightarrow$ E1 op1 E2 E1 $\rightarrow$ E2	E1 $\rightarrow$ E2 RE1 RE1 $\rightarrow$ op1 E2 RE1 RE1 $\rightarrow \epsilon$
E2 $\rightarrow$ E2 + E3 E2 $\rightarrow$ E3 - E3 E2 $\rightarrow$ E3	E2 $\rightarrow$ E3 RE2 E2' RE2 $\rightarrow$ - E3 RE2 $\rightarrow \epsilon$ E2' $\rightarrow$ + E3 E2' E2' $\rightarrow \epsilon$
E3 $\rightarrow$ E4 and E3 E3 $\rightarrow$ E4 or E4 E3 $\rightarrow$ E4	E3 $\rightarrow$ E4 RE3 RE3 $\rightarrow$ op3 RE3 $\rightarrow \epsilon$
E4 $\rightarrow$ E4 op4 E5 E4 $\rightarrow$ E5	E4 $\rightarrow$ E5 RE4 RE4 $\rightarrow$ op4 E5 RE4 RE4 $\rightarrow \epsilon$
E5 $\rightarrow$ op5 E5 E5 $\rightarrow$ E6	
E6 $\rightarrow$ E6 op6 E6 $\rightarrow$ literalEntero E6 $\rightarrow$ literalReal E6 $\rightarrow$ literalBool E6 $\rightarrow$ literalCadena E6 $\rightarrow$ identificador E6 $\rightarrow$ null E6 $\rightarrow$ ( E0 )	E6 $\rightarrow$ RE6 E6' E6' $\rightarrow$ op6 E6' E6' $\rightarrow \epsilon$ RE6 $\rightarrow$ literalEntero RE6 $\rightarrow$ literalReal RE6 $\rightarrow$ literalBool RE6 $\rightarrow$ literalCadena RE6 $\rightarrow$ identificador

	RE6 → null RE6 → ( E0 )
op1 → > op1 → >= op1 → < op1 → <= op1 → == op1 → != op3 → and E3 op3 → or E4 op4 → * op4 → / op4 → % op5 → - op5 → not op6 → [ literalEntero ] op6 → .campo op6 → ^	