

# Procesadores de Lenguajes

## Primera Fase

Lidia Concepción Echeverría      Juan Ramón del Caño Vega  
[lidiacion@ucm.es](mailto:lidiacion@ucm.es)      [jdelcano@ucm.es](mailto:jdelcano@ucm.es)

3 de marzo de 2018

## Índice

<b>1. Clases léxicas</b>	<b>2</b>
<b>2. Especificación formal</b>	<b>4</b>

## 1. Clases léxicas

Dispondremos de las siguientes clases léxicas:

1. Nombres de tipo: serán palabras reservadas en nuestro analizador léxico.
  - *num*: palabra reservada para tipar las variables numéricas.
  - *bool*: palabra reservada para tipar las variables booleanas.
2. Nombres de variable: *iden* serán los identificadores de las variables que declaremos. Deberán empezar por una letra, y pueden contener letras, dígitos o subrayados (\_).
3. Separadores: dispondremos de dos tipos, uno para separar la sección de declaraciones de la sección de instrucciones, y otro para separar las declaraciones o instrucciones entre ellas.
  - *pcoma*: será el separador de declaraciones o instrucciones entre ellas mismas (;).
  - *end*: indicará el final de la sección de declaraciones y el comienzo de la sección de instrucciones (&&).
4. Expresiones: las usaremos para trabajar en la sección de instrucciones. Tenemos dos tipos.
  - a) Expresiones lógicas: de nuevo serán palabras reservadas.
    - *true*: valor lógico de verdad (true).
    - *false*: (false) contrario de *true*.
  - b) Expresiones numéricas: *numero*, podrán comenzar por un signo (+ o -), seguido de uno o más dígitos. Si se tratase de números reales, aparecería un punto (.) seguido de uno o más dígitos. Finalmente, para ambos casos, podría aparecer una parte exponencial, con una *e* o *E* seguida opcionalmente de un signo (+ o -) y uno o más dígitos.
5. Operadores: para la sección de instrucciones dispondremos de los siguientes operadores.
  - a) Operador de asignación: *igual* (=) entre una variable y una expresión.
  - b) Operadores aritméticos:
    - *mas*: operador de suma (+).
    - *menos*: operador de resta y menos unario (-).
    - *por*: operador de multiplicación (\*).

- *div*: operador de división (/).

c) Operadores lógicos: también serán palabras reservadas.

- *and*: operador de conjunción.
- *or*: operador de disyunción.
- *not*: operador de negación.

d) Operadores relacionales.

- *mayor*: comprueba si una expresión es estrictamente mayor a otra (>).
- *menor*: comprueba si una expresión es estrictamente menor a otra (<).
- *mayorIgual*: comprueba si una expresión es mayor o igual a otra ( $\geq$ ).
- *menorIgual*: comprueba si una expresión es menor o igual a otra ( $\leq$ ).
- *equiv*: comprueba si dos expresiones tienen el mismo valor ( $\equiv$ ).
- *noEquiv*: comprueba si dos expresiones tienen distinto valor ( $\neq$ ).

e) Paréntesis: modifican la prioridad de las operaciones entre *parAb* ( ( ) y *parCe* ( ) ).

## 2. Especificación formal

- Definiciones auxiliares:

$$\begin{aligned}
 letra &\longrightarrow a|...|z|A|...|Z \\
 digitoPositivo &\longrightarrow 1|...|9 \\
 digito &\longrightarrow 0|digitoPositivo \\
 parteDecimal &\longrightarrow .digito^+ \\
 parteExponencial &\longrightarrow (e|E)[+|-]digito^+
 \end{aligned}$$

- Definiciones de cadenas ignorables:

$$separador \longrightarrow \text{SP}|\text{TAB}|\text{NL}$$

- Definiciones léxicas:

$$\begin{aligned}
 num &\longrightarrow \mathbf{num} \\
 bool &\longrightarrow \mathbf{bool} \\
 iden &\longrightarrow letra(letra|digito|_)^* \\
 pcoma &\longrightarrow ; \\
 end &\longrightarrow \&\& \\
 true &\longrightarrow \mathbf{true} \\
 false &\longrightarrow \mathbf{false} \\
 numero &\longrightarrow [+|-]digito^+[parteDecimal][parteExponencial] \\
 mas &\longrightarrow \backslash+ \\
 menos &\longrightarrow - \\
 por &\longrightarrow \backslash* \\
 div &\longrightarrow / \\
 and &\longrightarrow \mathbf{and} \\
 or &\longrightarrow \mathbf{or} \\
 not &\longrightarrow \mathbf{not} \\
 mayor &\longrightarrow > \\
 menor &\longrightarrow < \\
 mayorIgual &\longrightarrow >= \\
 menorIgual &\longrightarrow <= \\
 equiv &\longrightarrow == \\
 noEquiv &\longrightarrow != \\
 parAb &\longrightarrow \backslash( \\
 parCe &\longrightarrow \backslash)
 \end{aligned}$$