



Documento anónimo

**3.pdf**

*Exámenes 1 Parcial Resueltos*



**3º Procesadores de Lenguajes**



**Grado en Ingeniería Informática**



**Escuela Técnica Superior de Ingenieros Informáticos  
Universidad Politécnica de Madrid**

**-50€ OFF**

**¡Viaje sorpresa en camper  
con tus amigos!**

Descubre tu destino 2 días antes

Código: WAYNABOXSTUDENT



**waynabox**  
www.waynabox.com

# Examen Análisis Léxico Final 12 Enero 2017

Enunciado:

## PROCESADORES DE LENGUAJES Y COMPILADORES

12 de enero de 2017

**Observaciones:** 1. Las calificaciones se publicarán hacia el 23 de enero.  
2. La revisión será hacia el 25 de enero.  
3. En la web se avisará de las fechas exactas.  
4. La duración de este examen es de 40 minutos por pregunta.  
5. Todos los ejercicios tienen la misma puntuación.

1. Una sentencia simplificada de un lenguaje tiene la siguiente estructura en EBNF:

```
S::= SELECT [DISTINCT] columna {, columna} FROM tabla {, tabla} [WHERE Cond] [ORDER BY columna]
Cond::= E OpRel E
E::= núm_entero | núm_real | id (E) | id | (S)
OpRel::= > | < | <= | >= | <>
```

Donde:

- id, columna y tabla son identificadores que comienzan por una letra que puede ir seguida de letras, dígitos o dólares (\$), pero no puede haber dos o más dólares seguidos.
- El lenguaje diferencia entre minúsculas y mayúsculas.
- Las palabras reservadas del lenguaje van en mayúsculas.
- Los elementos del lenguaje pueden ir separados por blancos, tabuladores y saltos de línea
- Los números pueden llevar parte decimal (798.37) o no (873); en caso de llevar parte decimal, ésta va separada con un punto. En un número con parte decimal se puede omitir la parte entera (.578), pero no la decimal (88. es incorrecto).
- En la notación EBNF, los corchetes ([ ]) y las llaves ({ }) no forman parte del lenguaje, e indican, respectivamente, opcionalidad y que el contenido puede aparecer de 0 a varias veces.

Se pide diseñar un **Analizador Léxico** para este fragmento de lenguaje (indicando una gramática regular, *tokens* completos, autómata finito determinista, acciones semánticas y errores), que introduzca toda la información posible en la **Tabla de Símbolos**.



Educación 3.0

Videos, Apuntes, Clases online, Tutorías

# Aprende desde casa, como si estuvieras en el aula. Cursos on-line, trato personalizado a distancia.

Contacto personalizado, material actualizado, videos explicativos, sesiones de dudas y tutorías.

Especializados en estudios de ingeniería informática. Computación, Software, Videojuegos. Dobles grados en ADE y Matemáticas.

Solución:

EXAMEN ANÁLISIS LÉXICO, 12 ENERO DE 2017

**Paso 1: gramática regular:**

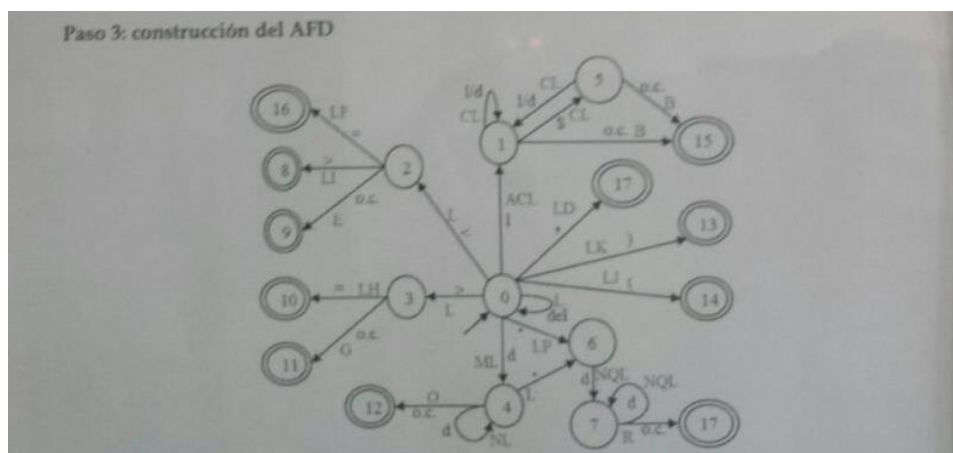
$$\begin{aligned} A &\rightarrow (B) \mid \cdot C \mid \cdot D \mid d E \mid \cdot G \mid ( \mid ) \mid \text{del } A \\ B &\rightarrow (B \mid d B \mid 5 F \mid \lambda \\ C &\rightarrow = \mid \times \mid \lambda \\ D &\rightarrow = \mid \lambda \\ E &\rightarrow d E \mid \lambda \mid \cdot G \\ F &\rightarrow (B \mid d B \mid \lambda \\ G &\rightarrow d H \\ H &\rightarrow d H \mid \lambda \end{aligned}$$

Donde: d := dígitos      del := delimitador      E := letras

**Paso 2: identificación de tokens**

Los tokens más adecuados para este lenguaje son:

- <ID, ptr(15)>: engloba a las palabras reservadas, los identificadores, las columnas y las tablas. El atributo será un puntero a la tabla de símbolos donde se encuentre el elemento. Se podría también diseñar un token distinto para las palabras reservadas.
- <COMA, ->: representa a la coma; no es necesario atributo
- <OPREL, n>: representa a los operadores relacionales. El atributo es un entero con el siguiente significado: 1: >; 2: >=; 3: <; 4: <=; 5: <>.
- <PAR, n>: representa a los paréntesis, siendo 1 el atributo cuando sea el paréntesis cerrado y 2 cuando sea el abierto.
- <INT, val>: representa a las constantes enteras, siendo el atributo el valor del número.
- <REAL, val>: representa a las constantes reales, siendo el atributo el valor del número.



#### Paso 4: acciones semánticas

Las funciones y variables indicadas tienen el siguiente significado:

- **BuscaTS**: función que busca la palabra en la Tabla de Símbolos, devolviendo la posición donde se encuentra o null si no se encuentra. La búsqueda se realiza teniendo en cuenta mayúsculas y minúsculas. La Tabla de Símbolos se encuentra inicializada con las palabras reservadas.
- **Concat**: concatena en la variable palabra el carácter leído.
- **GenToken**: genera el token formado por sus dos parámetros.
- **InsertaTS**: función que introduce la palabra en la Tabla de Símbolos, devolviendo la posición donde se ha introducido.
- **Leer**: realiza la lectura del siguiente carácter del fichero de entrada.
- **p**: variable donde se guarda la entrada de la Tabla de Símbolos.
- **palabra**: variable donde se concatenan los caracteres que se leen.

Todas las transiciones no indicadas, producirían una acción de Error.

| Nombre | Descripción  |
|--------|--|
| A      | palabra:= $\emptyset$  |
| B      | p:=BuscaTS (palabra)<br>if (p=null)<br>then p:=InsertaTS (palabra)<br>GenToken (ID, p) |
| C      | Concat (palabra)   |
| D      | GenToken (COMA, -)   |
| E      | GenToken (OPREL, 3)  |
| F      | GenToken (OPREL, 4)  |
| G      | GenToken (OPREL, 1)  |
| H      | GenToken (OPREL, 2)  |
| I      | GenToken (OPREL, 5)  |
| J      | GenToken (PAR, 1)  |
| K      | GenToken (PAR, 2)  |
| L      | Leer   |
| M      | núm:=d; dec:=0   |
| N      | núm:=núm*10+d  |
| O      | GenToken (ENT, núm)  |
| P      | núm:=0; dec:=0   |
| Q      | dec:=dec+1   |
| R      | GenToken (REAL, núm*10 <sup>-dec</sup> )   |