



**E.T.S.I. Informáticos**  
**Universidad Politécnica de Madrid**



## **Procesadores de Lenguajes**

**Práctica**

---

# **Presentación de la Práctica** **Implementación del Analizador Léxico**

**José Luis Fuertes**

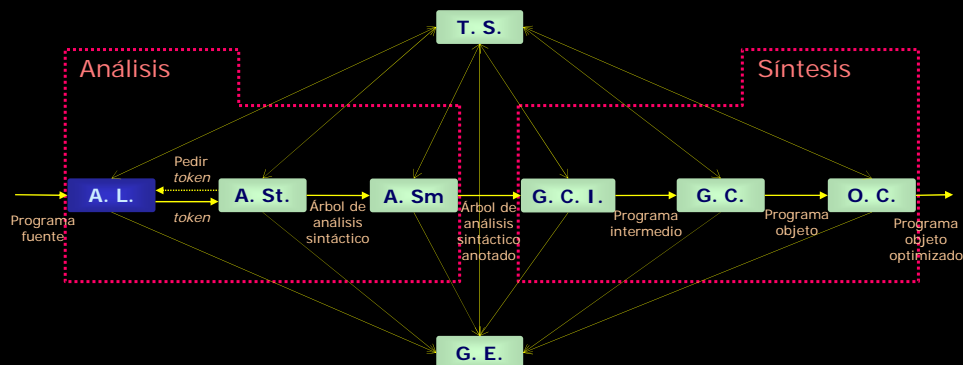
septiembre de 2022

## **Contenido**

---

1. Implementación de un Analizador Léxico
2. Ejemplo de implementación
3. Normas de la Práctica
4. Resumen del Lenguaje que hay que analizar

## Compilador



3

Análisis Léxico

## Implementación de un Analizador Léxico

- Implementar un tipo de datos para representar la matriz de transiciones del AFD

- Columnas:

- caracteres de entrada

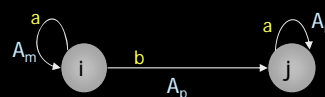
- Filas:

- estados

- Celdas:

- Estado al que se transita
- Acción semántica a ejecutar o código de error

	a	b	...	
...				
i	i	$A_m$	j	$A_p$
j	j	$A_n$		$E_1$
...				
				estado
				acción



<https://dlsiis.fi.upm.es/procesadores/Documentacion.html>

4

Práctica-AL

## Implementación de un Analizador Léxico

- Implementar el algoritmo de análisis con acceso a la matriz de transiciones

### Algoritmo

```

ALex ()
{
    estado:= 0;    /* estado inicial */
    LOOP UNTIL estado final
    {
        acción:= AFD.acción (estado, car);
        estado:= AFD.estado (estado, car);
        IF estado = null
            THEN error (acción);
            ELSE SWITCH (acción) /* ejecuta las acciones */
            {
                CASE Ai: Acción semántica i
                ...
            }
    } /* fin del LOOP */
}
    
```

5

Práctica-AL

## Ejemplo de implementación de un AL

- Sea el siguiente fragmento de un lenguaje de programación
  - Números enteros hexadecimales:
    - 0x seguido de al menos un dígito hexadecimal (0-9, A-F)
    - se representan con 4 bytes
  - Operadores + y -
  - Los elementos pueden ir separados por espacios
  - Los números enteros pueden ser positivos o negativos

6

Práctica-AL

## Ejemplo de implementación de un AL

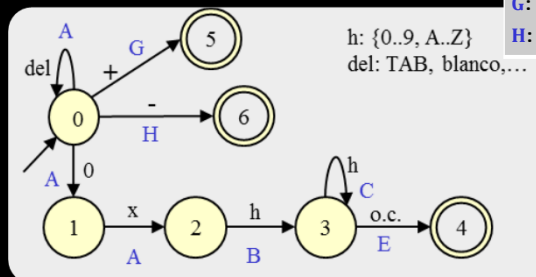
- Diseño del Analizador Léxico
  1. Identificación de los *tokens*
    - Números enteros: <Entero, valor>
    - Operadores aritméticos: <Más, -> y <Menos, ->
  2. Escritura de la gramática
    - $S \rightarrow 0 A \mid + \mid - \mid \text{del } S$
    - $A \rightarrow x B$
    - $B \rightarrow h C$
    - $C \rightarrow h C \mid \lambda$

7

Práctica-AL

## Ejemplo de implementación de un AL

- Diseño del Analizador Léxico
  3. Diseño del AFD
  4. Incorporación de acciones semánticas
  5. Errores



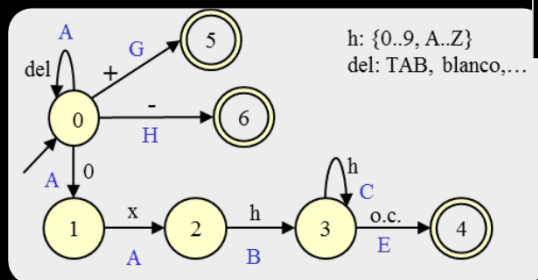
```
A: Lee
B: valor:= valorHexadecimal (h); Lee
C: valor:= valor * 16 + valorHexadecimal (h); Lee
E: if (valor < 231) Then GenToken (Entero, valor)
    Else Error (61)
G: Lee; GenToken (Más, -)
H: Lee; GenToken (Menos, -)
```

8

Práctica-AL

## Ejemplo de implementación de un AL

- Creación de la matriz de transiciones



	x	0	1..9 A..F	+	-	del
→0	50	1 A	51	5 G	6 H	0 A
1	2 A	52	53	54	55	56
2	57	3 B	3 B	58	59	60
3	4 E	3 C	3 C	4 E	4 E	4 E
4						
5						
6						
estado:						
acción:						

9

Práctica-AL

## Ejemplo de implementación de un AL

Implementación

```

Alex ()
{
    estado:= 0;
    LOOP UNTIL estado ≥ 4
    {
        acción:= AFD.acción (estado, car);
        estado:= AFD.estado (estado, car);
        IF estado = null THEN error (acción);
        ELSE SWITCH (acción) /*ejecuta la acción semántica que corresponda*/
        {
            CASE A: car:= leer ();
            CASE B: valor:= valorHexadecimal (car);
                     car:= leer ();
            CASE C: valor:= valor * 16 + valorHexadecimal (car);
                     car:= leer ();
            CASE E: IF valor < 231;
                     THEN Return (Entero, valor);
                     ELSE error (61);
            CASE G: car:= leer (); Return (Más, -);
            CASE H: car:= leer (); Return (Menos, -);
        }
    }
}
    
```

	x	0	1..9 A..F	+	-	del
→0	50	1 A	51	5 G	6 H	0 A
1	2 A	52	53	54	55	56
2	57	3 B	3 B	58	59	60
3	4 E	3 C	3 C	4 E	4 E	4 E

10

Práctica-AL

## Normas de la Práctica

- <https://dlsiis.fi.upm.es/procesadores/Practica.html>
- Crear grupos en la web
  - ♦ Máximo 3 estudiantes
- Funcionamiento
  - ♦ Entrada: fichero fuente
  - ♦ Salida: ficheros de *tokens*, tabla de símbolos y *parse*
    - Formato predefinido en la web
- Presentación
  - ♦ 2 entregas parciales
  - ♦ Presentación con la entrega final
- Procesador de JavaScript-PdL
  - ♦ Analizador Léxico, Tabla de Símbolos, Analizador Sintáctico, Analizador Semántico, Errores

## Lenguaje de la Práctica

- <https://dlsiis.fi.upm.es/procesadores/Practica.html>
  - ♦ Partes comunes a todos los grupos
    - Declaración de funciones
    - Tipos enteros, lógicos y cadenas
    - Variables y su declaración
    - Constantes enteras y cadenas
    - Sentencias: asignación, if, llamada y retorno de funciones, entrada/salida
    - Expresiones
    - 1 operador aritmético, 1 operando lógico, 1 operador relacional
  - ♦ Partes específicas de cada grupo
    - Comentarios
    - Cadenas
    - Sentencias compuestas
    - Operadores especiales
    - Técnica de análisis sintáctico

## Lenguaje de la Práctica

---

- <https://dlsiis.fi.upm.es/procesadores/IntroJavaScript.html>
  - ♦ Generalidades
  - ♦ Estructura de un Programa
  - ♦ Comentarios
  - ♦ Constantes
  - ♦ Operadores
  - ♦ Identificadores
  - ♦ Declaraciones
  - ♦ Tipos de Datos
  - ♦ Instrucciones de entrada/salida
  - ♦ Sentencias
  - ♦ Funciones