

# Análisis Léxico con Flex – Apuntes

---

## Gramática base

\$t ightarrow ext{entero}\$

| rutina  
| principal  
| si  
| entonces  
| otro  
| mientras  
| haz  
| regresa  
| \$((l \mid d))\$  
/ \$d+\$  
/ \$"C"\$

Todas las cadenas se guardan en **yytext**.

La longitud se guarda en **yylen**.

---

## Identificadores léxicos

| ;  
| ,  
| (  
| )  
| {  
| }  
| [  
| ]  
| >=  
| >  
| <=  
| <  
| =  
| !=  
| ==  
| +  
| -  
| \*  
| /

---

## Identificadores de cadenas

```
| $((l \mid d))$  
/ $d+$  
/ $"C"$
```

## Definiciones léxicas

\$\$ delim \rightarrow (x\_0 \mid CR)^\* \$\$

\$x\_0\$: espacio en blanco

\$CR\$: salto de línea

\$\$ l \rightarrow [a - z \cup A - Z] \$\$

\$\$ d \rightarrow [0 - 9] \$\$

\$\$ \Sigma = \{ \text{ext\{ascii\}} \} \$\$

## Atributos adicionales requeridos

- **Números:** variable global `Valor_num`
- **Id:** variable global `ap_id`, apuntador a un nodo en la tabla de símbolos
- **Cadena:** variable global `pos_cad` (índice en la tabla de cadenas)

## Tabla de Símbolos

TS → [estructura][...][estructura][...]

↑  
ap\_id

## Archivo de entrada para Flex

```
%{  
    // Declaración de símbolos  
%}  
  
%%  
// Reglas de traducción  
%%  
  
%%  
// Función  
%%
```

Este archivo es procesado por **Flex**, que genera **Lex.yy.c**.

## Código generado en C

```
FILE *yyin = NULL;

int yylex() {
    ...
    return ...;
}

int yywrap() {
    return 1;
}
```

**yylex()** realiza las acciones asociadas a los patrones.  
**yywrap()** indica fin del análisis léxico.

## Programa principal

```
#include "Lex.yy.c"

struct Nodo { ap_id = ... };
int Valor_num = 0;
int pos_cad = 0;

int main() {
    printf("Prop. archivo prueba");
    scanf("%s", Nombre);

    yyin = fopen(Nombre, "r");
    Cod = yylex();
    fclose(yyin);
}
```

**yyin** se declara en **lex.flex** y se utiliza para leer el archivo de prueba.  
**ap\_id**: apuntador a una entrada en la tabla de símbolos  
**yyin**: archivo de entrada que se declara en **lex.flex** **yylex()**: analiza los tokens según las reglas definidas  
**yywrap()**: indica fin del archivo

## Entregables

1. Archivo de especificación de entrada para Flex.

2. **Archivos del programa principal y funciones auxiliares.**

3. **Pruebas con salidas del programa** (archivo con formato de tabla):

| <b>Código token</b> | <b>yytext</b> | <b>Valor del atributo</b> |
|---------------------|---------------|---------------------------|
|---------------------|---------------|---------------------------|

4. **Archivo de prueba** que contenga todos los tokens delimitadores y **secuencias inválidas**.

5. **Tabla de símbolos y cadenas.**

6. **Archivo de errores** que contenga:

- Número de línea donde ocurrió el error
- Cadena inválida
- Mensaje describiendo el error

El archivo de prueba debe contener todos los tokens válidos y al menos una secuencia inválida para probar el analizador léxico.