

# Práctica Obligatoria

Primera entrega

David Moreno Majuelo – GII

Rubén Moreno Martín – GII + Mat.

Jaime Llanos Melchor – GII

## Introducción

En esta primera parte de la práctica se ha realizado tanto el analizador léxico como el sintáctico, dejando la traducción dirigida por sintaxis para la última entrega.

## Analizador Léxico

En primer lugar hemos desarrollado el .flex, para que imprimiera por pantalla los token según los fuera reconociendo, así fue como nos cercioramos del correcto funcionamiento del analizador léxico, para su posterior incorporación al sintáctico.

Posteriormente adaptamos el .flex para que se conectase con el analizador sintáctico añadiendo el “%cup”, las sentencias “%line” y “%column” que nos ayudarán a la hora de detectar los errores ya que nos sitúan en la línea y columna en la que se producen. Y por último cambiando lo que devuelve el analizador léxico, en vez de imprimírnos por pantalla vamos a devolver los tokens con un return al analizador sintáctico.

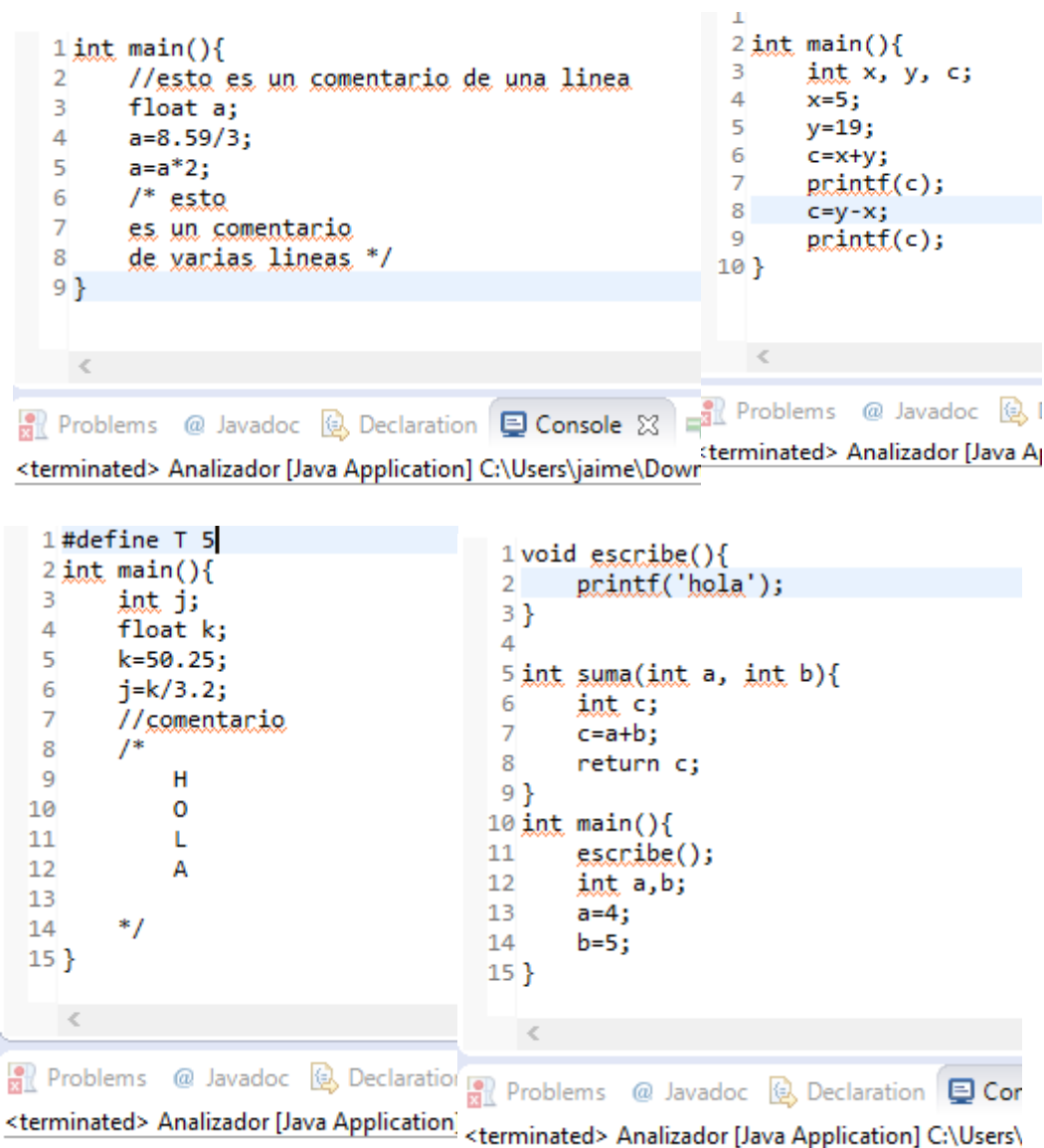
## Analizador Sintáctico

Hemos cogido la gramática que nos daba el enunciado y hemos declarado los elementos terminales que el .flex nos devuelve como tokens. También hemos definido dentro del .cup unos métodos para el tratamiento de errores sintácticos en la parte correspondiente a la clase “parser”.

Posteriormente hemos creado la clase Analizador.java, que actúa como “main” y que coge el fichero de texto que le pasamos como prueba. Con la ejecución de dicha clase, se nos mostrará por pantalla la columna y línea de nuestro error, el cual estará escrito en negro en el caso de ser Léxico y en rojo en caso de ser Sintáctico.

## Casos de Prueba

### -Casos correctos:



```
1 int main(){
2     //esto es un comentario de una linea
3     float a;
4     a=8.59/3;
5     a=a*2;
6     /* esto
7     es un comentario
8     de varias lineas */
9 }
```

```
1
2 int main(){
3     int x, y, c;
4     x=5;
5     y=19;
6     c=x+y;
7     printf(c);
8     c=y-x;
9     printf(c);
10 }
```

```
1 #define T 5
2 int main(){
3     int j;
4     float k;
5     k=50.25;
6     j=k/3.2;
7     //comentario
8     /*
9         H
10        O
11        L
12        A
13
14     */
15 }
```

```
1 void escribe(){
2     printf('hola');
3 }
4
5 int suma(int a, int b){
6     int c;
7     c=a+b;
8     return c;
9 }
10 int main(){
11     escribe();
12     int a,b;
13     a=4;
14     b=5;
15 }
```

Aquí tenemos 4 casos de prueba correctos. Son correctos porque al ejecutar el “Analizador.java”, no nos imprime ningún error por pantalla.

### -Casos erróneos:

```
1 int main( {
2     int c;
3 }
```

<terminated> Analizador [Java Application] C:\Users\jaime\  
Error de sintaxis en la línea 0 caracter 10

Error forzado quitando el “)” del main.

```
1 #define H=3;
2
3 int main() {
4     int c
5 }
```

<terminated> Analizador [Java Application] C:\Users\jaime\Downloads\EclipseComplete-win64\EclipseI  
Error en el analizador lexico: # en la línea 0 y el caracter de la posición 0  
Error de sintaxis en la línea 0 caracter 1

Error forzado poniendo mal el “#define”.

```
1 int main() {
2     int a, b;
3     a=9
4     b=1;
5 }
```

<terminated> Analizador [Java Application] C:\Users\jain  
Error de sintaxis en la línea 3 caracter 1


Error forzado quitando el “;” final de la línea nº3 (línea 2 para nuestro analizador, que empieza a contar en el 0). Por eso da error en la siguiente línea, porque ahí el analizador se espera un “;” y se encuentra un “ident”.

```
1 int main()
2     int suma = 1+1;
3 }
```

<terminated> Analizador [Java Application] C:\Users\jain  
Error de sintaxis en la línea 1 caracter 1

Error forzado no poniendo la llave abierta “{” en el main.

## -Pruebas en ejecutable

 Windows PowerShell

```
PS C:\Users\jaime\Desktop\FALTA SOLO MEMORIA\Ejecutable y pruebas> java -jar EjecutableAnalizador.jar
Escriba el nombre del fichero txt:
<<Asegúrese de que se encuentra en la misma ruta que el ejecutable)>>
correcto1.txt
PS C:\Users\jaime\Desktop\FALTA SOLO MEMORIA\Ejecutable y pruebas> java -jar EjecutableAnalizador.jar
Escriba el nombre del fichero txt:
<<Asegúrese de que se encuentra en la misma ruta que el ejecutable)>>
error1.txt
Error de sintaxis en la línea 0 carácter 10
PS C:\Users\jaime\Desktop\FALTA SOLO MEMORIA\Ejecutable y pruebas>
```

Aquí un ejemplo de las pruebas en el ejecutable.