

UNIVERSIDAD DE VALLADOLID

INGENIERÍA INFORMÁTICA

GRAMÁTICAS Y LENGUAJES FORMALES

---

# Analizador léxico - Ficheros Java

---

*Autor:*

Vicente Martínez Franco

28 de abril de 2017

## 1. Introducción

En este documento explicaremos la resolución de la práctica correspondiente al siguiente enunciado.

Deberá elaborar un programa LeX, asignándole el nombre `jest.l`. Situe todos los programas (en LeX, C, ...) y documentos en una carpeta llamada GLF-P02, que entregará comprimida con el nombre `App1App2.zip` (`GomezSanz.zip`, por ejemplo). El programa que se pide tomará como argumentos de línea de órdenes los nombres de cero o más ficheros de extensión `.java` (deberá ignorarlos en caso contrario) o leer de la entrada estándar si no se especifica ningún nombre de fichero con extensión `.java` en la línea de órdenes. Para cada uno de los ficheros, el programa deberá presentar:

- La relación de clases que se definen en el mismo, indicando fila y columna del fichero en que aparece la definición.
- Para cada clase que se declare, la relación de métodos y propiedades, señalando así mismo la línea y columna en que aparece la declaración del método o propiedad.
- La relación de identificadores que aparezcan declarados en el fichero, indicando también la fila y columna en que aparece la declaración.
- En la relación anterior deberán ignorarse los contenidos que aparezcan dentro de comentarios válidos en Java.

## 2. Solución

Para analizar el fichero utilizo el siguiente conjunto de estados:

- `NCLASE` : Para coger el nombre de la clase encontrado su inicio
- `CLASE` : Mientras estamos en el interior de una clase
- `METATR` : cuando hemos encontrado un método o atributo
- `METODO` : Mientras estamos en el interior de un método
- `COM` : Cuando nos encontramos dentro de un comentario multilinea
- `COM_CLAS`: Cuando nos encontramos dentro de un comentario multilinea en una clase
- `COM_MET` : Cuando nos encontramos dentro de un comentario multilinea en un método

Para ignorar los comentarios he introducido dos expresiones regulares distintas correspondientes a los dos tipos de comentarios. Por un lado los comentarios de una línea que empiezan por `//` contienen un numero indefinido de caracteres y acaban con un salto de línea. Por otra parte están los comentarios de multiples líneas que empiezan por `/*` y acaban por `*/`. Al encontrar el comienzo de un comentario de múltiples líneas pasa al estado correspondiente según su ubicación ignorando las clases metodos y atributos que encuentre hasta que termina al encontrar los caracteres finales.

Cuando encuentra el comienzo de una clase comienza el estado `NCLASE` en el que coje el nombre de la clase, imprime el nombre fila y columna y pasa al estado `CLASE`. Dentro de este estado busca el comienzo de un método o atributo (cominezan igual) y cuando lo encuentra pasa al estado `METATR`. En el caso de que se trate de un atributo imprime su nombre fila y columna y regresa a `CLASE`. Si es un método también imprime su nombre fila y columna y pasa al estado `METODO` en el que espera hasta que encuentra una llave de cierre que da final al método, momento en el que vuelve a `CLASE`. De igual manera cuando encuentra una llave de cierre estando en `CLASE` vuelve al estado 0 (el básico).

Además para contar líneas y columnas cada vez que encontramos una entrada sumamos su longitud a columnas y cada vez que encontramos un salto de linea aumentamos el numero de lineas en 1 y lo ponemos a 0.