



EJERCICIO 1. ¿Qué ocurre si en la primera sección se quitan las llaves al nombre de la macro letra?

La primera sección le corresponde a las expresiones regulares, las cuales tienen la forma `<Expresión Regular>={<Acción Léxica>}`. La acción léxica se ejecuta cada vez que se encuentra una cadena que coincida con la expresión regular, por lo tanto, si quitamos las llaves en `palabra={letra}+`, entonces el programa solo podrá identificar números.

EJERCICIO 2. ¿Qué ocurre si en la segunda sección se quitan las llaves a las macros?

Si quitamos las llaves a las macros en esta sección, entonces tenemos que al ejecutar `java Lexer input.txt`, el resultado es el contenido de la entrada sin analizar, esto se debe a que estamos declarando las macros pero nunca las utilizamos (cosa que nos advierte el mismo compilador), por lo que las llaves resultan indispensables.

EJERCICIO 3. ¿Cómo se escribe un comentario en flex?

Flex admite comentarios de estilo C, es decir, cualquier cosa entre `'/*'` y `'*/'` se considera un comentario.

EJERCICIO 4. ¿Qué se guarda en `yytext`?

En `yytext` se guarda la cadena que pertenece a las expresiones regulares asignadas en las macros, para que podamos usarla en nuestras funciones auxiliares al momento del análisis léxico.

EJERCICIO 5. ¿Qué pasa al ejecutar el programa e introducir cadenas de caracteres y de dígitos en el archivo de entrada?

Si en el archivo de entrada hay cadenas de caracteres y dígitos, entonces nuestro programa analizará correctamente su contenido e imprimirá las palabras y los números que compongan dichos elementos.

EJERCICIO 6. ¿Qué ocurre si introducimos caracteres como `"**"` en el archivo de entrada?

Como este carácter no ha sido capturado por ninguna de nuestras reglas,

entonces el analizador léxico no hará más que ignorarlo, es decir, seguirá funcionando salvo con aquellos elementos que no hemos englobado con nuestras expresiones regulares.

EJERCICIO 7. Modificar al código anterior en un archivo nuevo, de tal manera que reconozca lo siguiente:

- La expresión regular para los hexadecimales en lenguaje Java.
- 5 palabras reservadas del lenguaje Java.
- Los identificadores válidos del lenguaje Java, con longitud de a lo más 32 caracteres.
- Los espacios en blanco.

```
maxonomar@fedora:~/Documents/Compiladores/Dexter/Prácticas/P1/src/Primer programa en Lex$
jflex archivoMod.lex
Reading "archivoMod.lex"
Constructing NFA : 116 states in NFA
Converting NFA to DFA :
.....
64 states before minimization, 59 states in minimized DFA
Writing code to "Lexer.java"
maxonomar@fedora:~/Documents/Compiladores/Dexter/Prácticas/P1/src/Primer programa en Lex$
javac Lexer.java
maxonomar@fedora:~/Documents/Compiladores/Dexter/Prácticas/P1/src/Primer programa en Lex$
java Lexer inputMod.txt
Encontré un identificador: variable
Encontré un espacio
Encontré un identificador: $var
Encontré un espacio
Encontré una palabra reservada: static
Encontré un espacio
Encontré una palabra reservada: void
Encontré un espacio
Encontré un hexadecimal: 0x1a2a
Encontré un espacio
Encontré un hexadecimal: 0xdeadbeee
maxonomar@fedora:~/Documents/Compiladores/Dexter/Prácticas/P1/src/Primer programa en Lex$
```