

## Introducción al lenguaje lex

Un programa escrito en lex se compone de tres partes:

```
definiciones
%%
reglas de traducción
%%
rutinas auxiliares
```

En la primera parte se declaran las expresiones regulares que definen los componentes léxicos que el analizador reconocerá. Los operadores que se emplean en estas expresiones son:

`\ “ . ^ $ [ ] * + { } | ( ) /`

La siguiente tabla presenta en orden decreciente de precedencia, la conformación de expresiones regulares en lex. (*c* representa cualquier carácter simple, *r* una expresión regular, y *s* una cadena)

Expresión	Significado	Ejemplo
<i>c</i>	cualquier carácter <i>c</i> que no sea operador	a
<code>\c</code>	el carácter <i>c</i> literalmente	<code>\+</code>
<code>"s"</code>	la cadena <i>s</i> literalmente	<code>"adios..."</code>
<code>.</code>	cualquier carácter excepto de nueva línea	a.*b
<code>^</code>	el comienzo de línea	<code>^int</code>
<code>\$</code>	el fin de línea	a;\$
<code>[s]</code>	cualquier carácter en <i>s</i> excepto -	<code>[*+ab]</code>
<code>[^s]</code>	cualquier carácter que no esté en <i>s</i>	<code>[^1234567890]</code>
<i>r</i> *	cero o más ocurrencias de <i>r</i>	a*
<i>r</i> +	una o más ocurrencias de <i>r</i>	a+
<i>r</i> ?	cero o una <i>r</i>	a?
<i>r</i> { <i>m,n</i> }	<i>m</i> a <i>n</i> ocurrencias de <i>r</i>	a{1,8}
<i>r</i> <sub>1</sub> <i>r</i> <sub>2</sub>	<i>r</i> <sub>1</sub> y entonces <i>r</i> <sub>2</sub>	ab
<i>r</i> <sub>1</sub>   <i>r</i> <sub>2</sub>	<i>r</i> <sub>1</sub> o <i>r</i> <sub>2</sub>	a b
( <i>r</i> )	<i>r</i>	(a b)
<i>r</i> <sub>1</sub> / <i>r</i> <sub>2</sub>	<i>r</i> <sub>1</sub> cuando va seguida de <i>r</i> <sub>2</sub>	a/--

Forma de ejecutar un programa escrito en lex en linux:

```
$flex programa.l
$gcc lex.yy.c -lfl
```