

Expresiones regulares

Resumen de las expresiones más frecuentes

Nicolas Vaughan
Universidad de los Andes, Bogotá
n.vaughan@uniandes.edu.co

20 de agosto de 2020

1 Clases de caracteres

[abc]	coge a, b o c
[^abc]	coge todo <i>menos</i> a, b o c
[a-c]	coge el rango desde a hasta a
[1-3]	coge el rango desde 1 hasta 3
[^1-3]	lo que <i>no</i> esté en rango desde 1 hasta 3
[1-3[4-6]]	unión de dos clases: [1-6]
[1-3&&[3-6]]	intersección de dos clases: 3

2 Clases predefinidas

\n	salto de línea (en Linux y Mac)
\r\n	salto de línea (en Windows)
.	cualquier caracter (excepto \n)
\p{L}	cualquier caracter Unicode (excepto \n)
\d	dígito: [0-9]
\D	no-dígito: [^0-9]
\t	tabulador
\s	<i>whitespace</i> : [\r\n\t]
\S	no- <i>whitespace</i> : [^s]
\w	alfanumérico: [a-zA-Z0-9_]
\W	no-alfanumérico: [^w]

3 Caracteres reservados

. * ? + < > - & ^ \$ { } [] () \$

Todos estos tienen un significado especial en Regex. Por eso, si se quieren buscar,¹ es necesario ‘escaparlos’ precediéndolos con un \. (\\$ para buscar el signo \$, etc.)

¹En algunos lenguajes/editores es necesario usarlos también para *reemplazar*.

4 Límites (anclas)

<code>^</code>	principio de línea
<code>\$</code>	fin de línea
<code>\b</code>	límite de palabra
<code>\B</code>	<code>[^\b]</code>

5 Cuantificadores

CODICIOSO	PEREZOSO	
<code>?</code>	<code>??</code>	cero o una sola vez
<code>*</code>	<code>*?</code>	cero o más veces
<code>+</code>	<code>+?</code>	una o más veces
<code>{n}</code>	<code>{n}?</code>	exactamente n veces
<code>{n,m}</code>	<code>{n,m}?</code>	de n a m veces
<code>{n,}</code>	<code>{n,}?</code>	por lo menos n veces
<code>{,m}</code>	<code>{,m}?</code>	máximo m veces

‘Codicioso’ (*greedy*) significa que el cuantificador coge la cadena más extensa posible. ‘Perezoso’ (*lazy*) significa que coge la cadena más pequeña posible (es decir, termina apenas alcanza la mínima que satisfaga la búsqueda).

6 Grupos

<code>()</code>	captura en un grupo el texto buscado
<code>\1, \2, \3, etc.</code>	cada uno de los grupos capturados ²
<code>(?:)</code>	agrupa pero no captura

7 Lookarounds: *lookahead* y *lookbehind*

Son *aserciones* (falsas o verdaderas) usadas para buscar un texto sin capturarlo.

<code>(?=)</code>	<i>lookahead</i> positivo
<code>(?!)</code>	<i>lookahead</i> negativo
<code>(?<=)</code>	<i>lookbehind</i> positivo
<code>(?<!)</code>	<i>lookbehind</i> negativo

²Algunos lenguajes/editores usan en su lugar: `$1`, `$2`, `$3`. etc. **VsCode requiere `\1`, etc. en las búsquedas, y `$1`, etc. en los reemplazos, para usar grupos capturados.**