

## ER en aplicaciones computacionales

Las expresiones regulares juegan un rol importante en aplicaciones computacionales que involucran texto. Su utilidad más obvia es la de describir un conjunto de cadenas, lo que resulta de utilidad en editores de texto y aplicaciones para buscar y manipular textos. Muchos lenguajes de programación admiten el uso de expresiones regulares con este fin. Por ejemplo, Perl<sup>1</sup> tiene un potente motor de expresiones regulares directamente incluido en su sintaxis. Las herramientas proporcionadas por las distribuciones de Unix (incluyendo el editor *sed* y el filtro *grep*) fueron las primeras en popularizar el concepto de expresión regular.

Las expresiones regulares están disponibles en casi cualquier lenguaje de programación, así como algunas herramientas y aplicaciones. Aunque su sintaxis es relativamente uniforme, ésta cambia según las herramientas y aplicaciones consideradas. A continuación se describen los caracteres más usados en expresiones regulares.

Caracter	Descripción
.	Coincide con cualquier caracter excepto con "\n". Por ejemplo la expresión regular "g.t" en la cadena "el gato de piedra en la gótica puerta de getisboro goot" el motor de búsqueda encontrará "gat", "gót" y por último "get". Nótese que el motor de búsqueda no encuentra "goot"; esto es porque el punto representa un solo caracter y únicamente uno.
*	Es el cierre de Kleen. Coincide cero o más veces con el caracter o subexpresión anterior. Por ejemplo, zo* coincide con "z", "zo," "zoo", etc. Utilizando la expresión "[a-zA-Z]\d*" será posible encontrar tanto "H" como "H1", "H01", "H100" y "H1000", es decir, una letra seguida de un número indefinido de dígitos
+	Es la clausura positiva. Coincide una o más veces con el caracter o subexpresión anterior. Por ejemplo, 'zo+' coincide con "zo", "zoo", etc, pero no coincide con "z". La expresión "[a-zA-Z]\d+" encontrará "H1" pero no encontrará "H".
?	Equivale a {0,1}. Coincide una vez o ninguna con el caracter o subexpresión anterior. Por ejemplo la expresión regular "ob?scuridad" permite encontrar tanto "oscuridad" como "obscuridad". En conjunto con los paréntesis permite especificar que un conjunto mayor de caracteres es opcional; por ejemplo "Nov(\. iembre ember)?" permite encontrar tanto "Nov" como "Nov.", "Noviembre" y "November". Equivale a {0,1}.
[]	La función de los corchetes en el lenguaje de las expresiones regulares es representar "clases de caracteres", o sea, agrupar caracteres en grupos o clases. Son útiles cuando es necesario buscar uno de un grupo de caracteres. Dentro de

<sup>1</sup> Es un lenguaje de programación diseñado por Larry Wall en 1987. Perl toma características del lenguaje C, del lenguaje interpretado bourne shell (sh), AWK, sed, Lisp y, en un grado inferior, de muchos otros lenguajes de programación. Estructuralmente, Perl está basado en un estilo de bloques como los del C o AWK, y fue ampliamente adoptado por su destreza en el procesado de texto y no tener ninguna de las limitaciones de los otros lenguajes de script

	<p>los corchetes es posible utilizar el guion "-" para especificar rangos de caracteres. La expresión regular "[A-Fa-f]" nos permite encontrar cualquier caracter de la "A" a la "F". Los corchetes nos permiten también encontrar palabras aún si están escritas de forma errónea, por ejemplo, la expresión regular "expresi[oó]n" permite encontrar en un texto la palabra "expresión" aunque se haya escrito con o sin tilde. Es necesario aclarar que sin importar cuantos caracteres se introduzcan dentro del grupo por medio de los corchetes, el grupo sólo le dice al motor de búsqueda que encuentre un solo caracter a la vez, es decir, que "expresi[oó]n" no encontrará "expresioon" o "expresioón".</p>
()	<p>De forma similar que los corchetes, los paréntesis sirven para agrupar caracteres, sin embargo existen varias diferencias fundamentales entre los grupos establecidos por medio de corchetes y los grupos establecidos por paréntesis:</p> <ul style="list-style-type: none"> <li>• Los caracteres especiales conservan su significado dentro de los paréntesis.</li> <li>• Los grupos establecidos con paréntesis establecen una "etiqueta" o "punto de referencia" para el motor de búsqueda que puede ser utilizada posteriormente (véase más abajo el caracter "\").</li> </ul>
{ }	<p>Comúnmente las llaves son caracteres literales cuando se utilizan por separado en una expresión regular. Para que adquieran su función de metacaracteres es necesario que encierren uno o varios números separados por coma y que estén colocados a la derecha de otra expresión regular de la siguiente forma: "\d{2}"; esta expresión le dice al motor de búsqueda que encuentre dos dígitos contiguos. En la siguiente expresión "\d{2,4}" se añade segundo número separado por una coma, el cual indica al motor de búsqueda que como mínimo debe aparecer 2 veces y como máximo debe aparecer 4 veces la expresión regular \d.</p>
	<p>Sirve para indicar una de varias opciones. Por ejemplo, la expresión regular "a e" encontrará cualquier "a" o "e" dentro del texto. La expresión regular "este oeste norte sur" permitirá encontrar cualquiera de los nombres de los puntos cardinales. La barra se utiliza comúnmente en conjunto con otros caracteres especiales.</p>
^	<p>Este caracter tiene una doble funcionalidad, que difiere cuando se utiliza individualmente y cuando se utiliza en conjunto con otros caracteres especiales. En primer lugar su funcionalidad como caracter individual: el caracter "^" representa el inicio de la cadena (de la misma forma que el signo de dólar "\$" representa el final de la cadena). Por tanto, si se utiliza la expresión regular "^[a-z]" el motor encontrará todos los párrafos que den inicio con una letra minúscula. Cuando se utiliza en conjunto con los corchetes de la siguiente forma "[^\\w ]" permite encontrar cualquier caracter que NO se encuentre dentro del grupo indicado. La expresión indicada permite encontrar, por ejemplo, cualquier caracter que no sea alfanumérico o un espacio, es decir, busca todos los símbolos de puntuación y demás caracteres especiales.</p>

\$	Representa el final de la cadena de caracteres o el final de la línea.
\	<p>Marca el siguiente caracter como un caracter especial o un literal. Por ejemplo al utilizarlo en combinación con el punto "\" este deja de tener su significado normal y se comporta como un caracter literal. Por ejemplo, 'n' coincide con el caracter "n" y '\n' coincide con un caracter de nueva línea. A continuación se presentan las combinaciones de caracteres especiales más usadas:</p> <ul style="list-style-type: none"> <li>• \t — Representa un tabulador.</li> <li>• \r — Representa el "retorno de carro" o "regreso al inicio".</li> <li>• \n — Representa la "nueva línea".</li> <li>• \d — Representa un dígito del 0 al 9.</li> <li>• \w — Representa cualquier caracter alfanumérico.</li> <li>• \s — Representa un espacio en blanco.</li> <li>• \D — Representa cualquier caracter que no sea un dígito del 0 al 9.</li> <li>• \W — Representa cualquier caracter no alfanumérico.</li> <li>• \S — Representa cualquier caracter que no sea un espacio en blanco.</li> </ul> <p>En algunos lenguajes de programación este caracter también se utiliza para recuperar “puntos de referencia”, es decir, la sintaxis \N con N = {1,2,3,...} indica que en la posición donde se encuentra la \N se deberá colocar el valor correcto correspondiente al N-avo paréntesis de la expresión en cuestión; los paréntesis se numeran contando de izquierda a derecha y numerando dichos paréntesis conforme los vas encontrando, comenzando en 1. Por ejemplo en la expresión “av(DFHTI)[0-9](wxz)\1qd\2” el valor correspondiente a \1 es el que corresponde al primer paréntesis encontrado de izquierda a derecha, es decir, el valor correcto de (DFHTI). Ahora, el valor que va en la posición donde encontramos el símbolo \2 será el caracter correcto del segundo paréntesis encontrado, es decir (wxz).</p>