

Expresiones regulares y filtros.

Una expresión regular es un patrón que define a un conjunto de cadenas de caracteres. Las expresiones regulares se construyen de forma análoga a las expresiones aritméticas. Existe la posibilidad de combinar expresiones simples; para ello, debemos emplear distintos operadores.

Los bloques básicos de construcción son las expresiones regulares que referencian un único carácter. La mayoría de los caracteres, incluyendo todas las letras y dígitos, son expresiones regulares que se definen a si mismos. Cualquier metacarácter con significado especial debe ser precedido del símbolo `\` para que pierda su significado especial.

Una lista de caracteres encerrados dentro de `[]` referencia cualquier carácter sencillo de esa lista. Si el primer carácter es un `^` entonces estaremos haciendo referencia a los caracteres que no aparecen en la lista.

Los caracteres `^` y `$` son metacaracteres que representan una cadena vacía al principio y al final de la línea, respectivamente. Los símbolos `\<` y `\>` representan una cadena vacía al principio y al final de una palabra.

Una expresión regular que representa un carácter sencillo puede ser continuada con uno o varios caracteres de repetición.

- ? El elemento precedente es opcional y debe coincidir al menos una vez.
- * El elemento precedente debe coincidir cero o más veces
- {n} El elemento precedente debe coincidir exactamente n veces
- + El elemento precedente debe coincidir una o más veces.
- {,m} El elemento precedente es opcional y debe coincidir al menos m veces.

- {n,m} El elemento precedente debe coincidir al menos n veces pero no más de m veces.

Dos expresiones regulares pueden unirse con el operador `|`. La expresión resultante representa cualquier cadena que responda a uno de los dos patrones.

Ejemplos:

Patrón	Qué representa
pablo	La cadena Pablo
^pablo	La cadena pablo al comienzo de una línea.
pablo\$	La cadena pablo al final de una línea.
^pablo\$	La cadena pablo formando una única línea.
niñ[oa]	La cadena niño o niña
ni[^aeiou]o	La tercera letra no es una vocal minúscula.
ga.o	La tercera letra es cualquier carácter.
^....\$	Cualquier línea que contenga 4 caracteres.
^\.	Cualquier línea que comienza por punto.
^[^.]	Cualquier línea que no comienza por punto.
niños*	niño, niños, niñoss, niñosss, etc
"niño"	niño entre comillas dobles.
"*niño"*	niño con o sin comillas dobles.
[a-z][a-z]*	una o más letras minúsculas.
[a-z]+	una o más letras minúsculas (sólo válido en algunas aplicaciones).
[^0-9A-Z]	cualquier caracter que no sea ni número ni letra mayúscula.
[a-zA-Z]	cualquier letra sea mayúscula o minúscula.
[Ax5]	cualquier carácter que sea A, x o 5.
niño niña nada	una de las tres palabras.
(s arb)usto	la palabra susto o arbusto.
ga?t[oa]	gato, gata, gasto, gaita, etc.
\<ga	cualquier palabra que empiece por ga.
ño\>	cualquier palabra que termine por ño
\<niño\>	la palabra niño
o\{2,\}	dos o más oes en una misma fila.

Siempre que empleemos expresiones regulares con **grep**, usaremos la opción **-E**, para que entienda todas las expresiones regulares.

Existen algunas clases de caracteres predefinidas:

- **[[:alnum:]]** -> Carácteres alfanuméricos, también se puede representar por **\w**, y **\W** representa lo contrario.
- **[[:alpha:]]** -> Carácteres alfabéticos
- **[[:cntrl:]]** -> Códigos de control
- **[[:digit:]]** -> Dígitos
- **[[:graph:]]** -> Carácteres gráficos
- **[[:lower:]]** -> Letra minúscula
- **[[:print:]]** -> Carácter imprimible
- **[[:punct:]]** -> Símbolo de puntuación
- **[[:space:]]** -> Espacios y tabuladores
- **[[:upper:]]** -> Letra mayúscula
- **[[:xdigit:]]** -> Dígito hexadecimal