

Expresiones Regulares

- existen de 2 tipos: básicas y extendidas
- reglas avanzadas de comparación
 - **[a,b,c,d,1,2,3]** : entre corchetes
 - **1-9, a-z** : rangos
 - **.** : (punto) == cualquier caracter, excepto salto de línea
 - **^ \$** : marcan principio y final de línea respectivamente
 - *** + ?** : indica (*) cero o más coincidencias, (+) una o más coincidencias y (?) cero o una coincidencias respectivamente
 - **|** : O lógica, 2 expresiones que puede ser cualquiera de ellas
 - **()** : preferencia de ejecución de la REGEXP, subexpresiones
 - **** : caracter de escape (para signos que tienen algún significado como REGEXP)

sed

- `sed -i <reg_exp> fichero` : realiza los cambios sobre el fichero directamente
 - `-i.bak` : crea un fichero .bak del original antes de modificar

ejemplos

- sustituir una palabra en medio de una cadena:

```
echo "mi mama me mima" | sed 's/\(mam\) [^ ]*/papa/'  
# -> mi papa me mima
```

grep

- `$ grep [opciones] EXP_REG [archivos]`
 - **-c ó -count** : cuenta las líneas en lugar de las coincidencias
 - **-f ó -file** : archivo de entrada que indica patrón de búsqueda
 - **-i ó -ignore-case**
 - **-r ó -recursive**
 - **-F ó -fixed-strings** : cambia de modo de búsqueda a modo cadena en lugar de REG-EXP
 - **-E ó -extended-regexp** : usa expresiones regulares extendidas
 - Ejemplos:

```
▪ $ grep eth[01] /etc/*
```

: busca cadenas «eth0» y «eth1» en los ficheros de /etc/

```
▪ $ grep -E "(twain\.example\.com|bronto\.pangea\.edu).*127" /etc/*
```

: busca twain.example.com, bronto.pangea.edu, seguido, en la misma línea, por el número 127

```
▪ $ ps aux | grep xterm
```

: filtra la salida del comando mostrando solo los que coinciden con la expresión buscada

From:

<https://matebcn.eu/> - **miguel angel torres egea**

Permanent link:

<https://matebcn.eu/info:cursos:pue:lpic1:103.7>

Last update: **15/08/2018 13:33**

