

# RegExp

java, API, PTE!

/via: <https://blog.desdelinux.net/con-el-terminal-uso-de-expresiones-regulares/>

## básico

- `.` : cualquier caracter (pero solo 1)
- `*` : cualquier ristra de caracteres (desde 0)
- `?` : cualquier caracter, pero de manera opcional (1 o 0) → regexp extendidas
- `[abc]` : a || b || c
- `[^ac]` : NO a || NO c
- `[0-9]` : rango
- `<palabra1>|<palabra2>` palabra1 o palabra2 → regexp extendidas
- `\<palabra` : palabra sin que tenga nada a la izquierda (signo < escapado)
- `palabra\>` : palabra sin que tenga nada a la derecha (signo > escapado)
- `+` : repetido al menos una vez → regexp extendidas

```
grep -E ':[0-9]+' # haría match sobre los minutos y segundos -y dos puntos -  
de una hora (por ejemplo, 03:32:39)
```

- `\{n\}` : repetido n veces (signos { y } escapados en regexp básicas, no en extendidas)
- `\{n,m\}` : repetido entre n y m veces

## subexpresiones (backreference)

todas las mayúsculas se meten entre ( y )

```
sed 's/\([A-Z]\)/(\1)/g' frases
```

- en la primera parte de la expresión, se escapan ( y ) pq forman parte de la subexpresión
- en la segunda parte, simplemente se usan

otra expresión de reemplazo sin subexpresiones:

```
sed 's/.*/"&"/g' frases
```

## referencias/ejemplos

- biblioteca de regexp: <http://regexlib.com/>
- comprobador online, da información: <http://gskinner.com/RegExr/>
- add-on firefox: <https://addons.mozilla.org/en-US/firefox/addon/rext/>
- [sed](#)
- [grep](#)

## otros ejemplos

extraer direcciones de correo de un fichero:

```
grep -E '\<[A-Za-z0-9._%+- ]+@[A-Za-z0-9.- ]+\.[A-Za-z]{2,4}\>' FICHERO
```

From:

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

Permanent link:

<https://matebcn.eu/development:java:regex>

Last update: **13/12/2019 16:18**

