Expresiones regulares ("Regular Expresion" o "Regex")

Ya antes hablamos brevemente (o más bien utilizamos para nuestros propósitos) la expresiones regulares. En este artículo explicaremos de la manera más sencilla posible este quebradero de cabeza para muchos de nosotros.

Actualizado el miércoles 29 de abril de 2020.

Publicado el martes 13 de agosto de 2019.

https://twitter.com/nixcraft/status/1164606713858678789
«Me siento atacado.»

¿Qué son las expresiones regulares?

> Una expresión regular es una secuencia de caracteres que definen (o más bien *conforman*) un patrón de búsqueda.

Ejemplo:

Si nuestro patrón de búsqueda es "perro" y *lo aplicamos* a un archivo de texto con la siguiente lista:

```
1.-Tortuga.
2.-Autobús.
3.-Perro.
4.-Nutria gigante o perro de agua.
5.-Zamuro.
6.-Can (perro).
7.-Cinco céntimos de peseta (perra chica).
8.-Pera (fruto del peral).
```

obtendremos el siguiente resultado (filtrado de líneas):

```
4.-Nutria gigante o perro de agua.
6.-Can (perro).
```

Obvio que se trata de devolver y mostrar el contexto donde se encuentre la cadena de búsqueda (línea u oración) sin embargo veremos que es sumamente útil para localizar y sustituir rápidamente por otra cadena de texto. Emplearemos acá el color de fuente en rojo para resaltar las expresiones regulares y en anaranjado sus "resultados".

En nuestro ejemplo no resulta seleccionada la línea número 3 ni la número 7, veamos por qué.

https://twitter.com/nixcraft/status/1168379673018957824
Algunas personas, cuando confrontan un problema, piensan que "Lo se, usaré expresiones regulares." Ahora ellos tienen dos problemas.

Trabajando con letras, no con palabras

Fíjense que empleamos las comillas para denotar que es exactamente eso lo que buscamos, *todo en minúsculas* y la línea seis no cumple a cabalidad con lo que ordenamos buscar. Incluso las siguientes palabras:

```
p erro
perro-s
p e r r o
```

tampoco son las que buscamos porque ya bien tienen espacios o guiones, sin embargo:

```
-perro, perro albarraniego.
```

sí que cumplen pues los guiones o espacios están fuera y no "en el medio". Es decir, estamos buscando una letra pe seguida de una letra e, luego dos erres y la vocal o, *todo junto ("sin*

espacios"), en ese mismo orden, en minúsculas todos los caracteres.

El punto y el asterisco

Como vimos, nosotros los seres humanos tenemos una inteligencia superior y cuando buscamos algo también aceptamos sus equivalentes, así no estén explícitamente en la orden que dimos. **Esto resulta totalmente desconcertante a nuestras computadoras** y muchos científicos, matemáticos y programadores luchan todos los días para que las máquinas entiendan si quiera algo de lo que les pedimos (no hablemos ya de que piensen por sí mismas, todas los ordenadores del mundo juntos no tienen más inteligencia que un insecto cualquiera).

Queremos decir: que para nosotros buscar "perro" y que en el listado se encuentre la palabra "perra" y no haya sido seleccionada... ¡pues que a nosotros los humanos eso nos llama mucho la atención!.

Como dijimos estamos buscando carácter por carácter, letra por letra y "perra" no se corresponde exactamente y acá es donde comenzamos a utilizar un *comodín (por favor no confundir con* ***metacarácter,*** *no)*.

Para ello utilizaremos el punto para indicar que allí queremos buscar **una sola letra o carácter, sin importar cual sea.** Por eso ahora nuestra búsqueda sería "perr." y nuestro resultado sería el siguiente:

```
4.-Nutria gigante o perro de agua.
5.-Zamuro.
6.-Can (perro).
7.-Cinco céntimos de peseta (perra chica).
```

Ahora bien, *si tuviéramos en la* lista, palabras como "**perri**" o **"perru"** (del todo ilógicas) -o incluso **"perr_"-** también las devolvería como resultado...

¿Para qué nos serviría esto? Imaginen que programamos un corrector ortográfico: este toma palabra por palabra *(cadena de búsqueda)* de un texto que estemos escribiendo y la busca en una lista de palabras "bien escritas" y si no la consigue asume que es un posible error ortográfico (decimos *posible* porque puede ser una palabra nueva o muy especializada que no tengamos en nuestro diccionario).

Pero vamos más allá, si vamos de derecha a izquierda sustituyendo una letra por un punto podremos sugerir al usuario un candidato a corrección, para eso sirve nuestro comodín. Ahora veamos un metacarácter que modifica el comportamiento del comodín.

El asterisco funciona modificando el comportamiento del comodín punto: *sirve para indicar que no importa cuántas veces se repita el carácter, sin importar la longitud de cadena.* Si queremos que devuelva también resultados como "perrera" o "perreda" tendremos que buscar por "perr.*a"; o si queremos resultados como "perraje" o "perrengue" deberemos usar "perr.*e". Pero debemos tener cuidado porque si buscamos "perr.*e" y tenemos la frase "el perrengue ese" *nos devolverá el siguiente resultado:* "el perrengue ese".

> El asterisco funciona sin importar su longitud, la cual incluso **puede ser cero.** Exacto: cuando buscamos "perr.*a" también obtendríamos el resultado "perra", ¡sorprendente! ¿cierto?

Ejercicio: también podemos buscar "per*.", aplíquenla a la lista y verán el resultado.

Mantengamos la mente abierta porque en realidad las expresiones regulares funcionan de una manera un tanto más complicada

¿Cómo funciona?

Vamos a utilizar una noticia cualquiera del diario nacional venezolano «*Últimas Noticias*» y usaremos una línea de ella que contiene la palabra "oro":

```
Fortalezcamos el bolívar, resp
aldémoslo en oro
, deslindémoslo del
 dólar, aumentemos las reservas en oro
, que al estar en las bóvedas del BCV no podrán ni robarlas, ni bloquearl
as.
```

Buscar la *expresión regular* "oro" va de la siguiente manera (pseudocódigo):

1. Tomamos la primera letra de la expresión regular, "o".
2. De izquierda a derecha, tomamos la primera letra de la línea, y la comparamos.
3. Si no coincide tomamos la siguiente letra de la línea.
4. Si coincide tomamos *la siguiente letra de la línea* y la comparamos con la segunda letra de la expresión regular.
5. Si no coincide (y vemos claro que no porque: "Fortalezcamos ...") continuamos de nuevo con el paso número 3.
6. Como sospechamos, llegará un momento que coincidirán las tres letras de la expresión regular contra las tres letras en análisis de la línea, lo cual devuelve positivo, pero esperen, aún hay más.
7. Se debe seguir revisando ¿hasta el final de la línea? Pues no porque la expresión regular tiene tres caracteres, así que deberemos restar (en este ejemplo, 180 - 3 =177, hasta ese número de carácter).
8. Lo del paso anterior es importante (ver si hay más coincidencias) si estamos por sustituir texto, generalmente las expresiones regulares son muy utilizadas en ese propósito, hasta el final de una línea, párrafo o documento completo (fichero) .

https://twitter.com/garabatokid/status/1147063121678389253

Caracteres, comodines, juegos de caracteres y anclas

https://twitter.com/kvlly/status/1161838538045960192
«Digo, vean esta obra de arte»

Antes de tocar este punto, definiremos de nuevo como un concepto que "**devolver un resultado**" es tomar el hecho positivo de haber encontrado *la expresión regular en una cadena de caracteres (texto, archivo, línea, párrafo, etcétera).* De nuevo recuerden que la expresión regular la denotamos en color rojo y "sus resultados" en color anaranjado.

Ahora de manera muy básica y con fines didácticos podemos decir que las expresiones regulares constan de al menos uno o más de los siguientes componentes, de menor a mayor jerarquía:

- Caracteres en sí mismos.

- Comodines, osea, caracteres que se comportan como comodines (ya vimos el punto).
- *Metacaracteres* que modifican el comportamiento de los comodines (ya vimos el asterisco): ¿qué tal si necesitamos buscar el punto para así separar (o contar) las oraciones de un párrafo? Pues para ello necesitamos el *metacarácter* "\ " (barra inversa o barra izquierda), así: "\."; al final veremos muchas otras combinaciones, por ejemplo el espacio como es "invisible" lo buscamos como "\s" -metacarácter acompañado de un comodín, la letra ese-.
- Juegos de caracteres:
    - Implícitos:
        - Un solo carácter es el mínimo juego de caracteres posible y que devuelve como resultado un carácter (el mismo carácter, obvio).
        - Un comodín por sí solo es un juego de caracteres mínimo que puede representar un solo carácter como resultado (el caso del punto, por ejemplo).
    - Explícitos:
        -
        - Van encerrados o delimitados entre dos *metacaracteres:* para ello utilizamos el par de corchete rectos "**[]**" y dentro colocaremos uno o varios de los caracteres, *pero con la particularidad*

Fuentes consultadas

En idioma castellano

-

En idioma inglés

- «Regex For Noobs (like me!) - An Illustrated Guide».
- «Regular expressions quick reference».
- «Regex».
- «Regexone» practicing on line, regex exercise!
- «https://regexcrossword.com/».
- «Regex tutorial — A **quick** cheatsheet by examples» Jonny Fox»
- «Using Grep & Regular Expressions to Search for Text Patterns in Linux» by Justin Ellingwood
- «What the Regex?! A Practical Guide to Regular Expressions»

- «https://www.digitalocean.com/community/tutorials/using-grep-regular-expressions-to-search-for-text-patterns-in-linux»

https://twitter.com/ks7000/status/1072328512344940545