

### SELECTORES... para elegir qué contenido se comprueba

```
.           // Vale por cualquier cosa
[ABD]       // Grupo Opcional: o bien A, o B o C
[A-M]       // Rango : todas las letras de la A a la M
[A-Mn-z]    // RangosAgrupados : todas las letras de la A a la M , y de la n a la z
A|M        // alternativa : o bien una A o una M
\w          // Una letra mayúscula o minúscula, un dígito o el carácter Equivale a [a-zA-Z0-9_]
\d          // Un Dígito. Equivale a [0-9]
\s          // Un espacio en blanco
\n \t \r    // Un char de nueva línea, o un tabulador, o un retorno de carro
[^AB]       // Exclusión : no puede tener ni A ni M (no cuenta si se usa ^ como comienzo de regla)
\.          // Escape, indica que lo que sigue se ha de tomar literalmente: un punto
            (esto es, aquí el punto significa "un punto", no "vale por cualquier cosa")
```

### CUANTIFICADORES ... para decir cuántas veces pues aparecer algo

```
[A-Z]*      // una mayúscula, 0 o más veces
[A-Z]+      // una mayúscula, 1 o más veces
[A-Z]?      // una mayúscula, 0 o 1 vez
(?:porra)   // exclusion, no puede aparecer la palabra porra
```

### MULTIPLICADORES ... para multiplicar varias veces lo que va delante

```
[A-Z]{2}    // una mayúscula, 2 veces (ni mas ni menos)
[A-Z]{3,8}  // una mayúscula, de 3 a 8 veces
[A-Z]{5,}   // una mayúscula, 5 veces o más
```

### SEÑALES DE INICIO Y FIN DE CADENA

```
^Hola       // La cadena debe empezar por Hola
adios$      // La cadena debe terminar por adios
```

### Símbolos básicos

La union de dos elementos sin nada que los separe indica Concatenación .

El patron "AB" cumple si el String contiene A seguida de B

. Un punto indica cualquier carácter

El símbolo ^ **sin corchetes** indica el principio del String. Se usa delante de la cadena que evalúa.

El patron "^expresion" cumple si el String contiene expresión al principio, sin nada por delante.

El símbolo ^ **dentro de los corchetes** indica negación.

El patron "[^abc]" cumple si el String contiene cualquier carácter excepto a ó b ó c.

El símbolo \$ indica el final del String. Se usa detrás de la cadena que evalúa.

El patron "expresion \$" cumple si el String contiene expresión al final

[ ] Los corchetes representan una definición de conjunto.

El patron "[abc]" cumple si el String contiene las letras a ó b ó c.

- Indice un Rango.

El patrón "[a-z1-9]" cumple si el String contiene letras minúsculas desde la a hasta la z (ambas incluidas) y los dígitos desde el 1 hasta el 9 (ambos incluidos)

| El carácter | es un OR .

El patrón "A|B" cumple si el String contiene A ó B

( ) Agrupar (como en matematicas) para aplicar modificadores o cuantificadores

Una barra invertida **antes de un carácter no especial** indica que éste debe ser interpretado como un carácter especial y no de forma literal. Por ejemplo, una 's' sin '\' precediéndole coincidirá con cualquier 's' minúscula en la cadena, sin embargo, '\s' no coincidirá con algún carácter en específico; representará un espacio en blanco.

Una barra invertida **antes de un carácter especial** indica que éste deberá ser interpretado literalmente, esto es, como un carácter simple y no como un carácter especial. A esto se le denomina escapado. Por ejemplo para indicar que se desea la aparición de un signo + varias veces, no puedo usar (+)\* pues el + tiene significado como cuantificador. Hemos de usar (\\+)\*.

En un String de Java, se ha de usar \\ en vez de \, pues hay que "escapar" la propia barra invertida, ya que la \ es un carácter especial (precisamente, el de escapado en cadenas)

## Cuantificadores básicos

<b>{X}</b>	Indica que lo que va justo antes de las llaves se repite X veces
<b>{X,Y}</b>	Indica que lo que va justo antes de las llaves se repite mínimo X veces y máximo Y veces. También podemos poner <b>{X, }</b> indicando que se repite un mínimo de X veces sin límite máximo.
<b>*</b>	Indica 0 ó más veces (el elemento que haya delante). Equivale a {0,}
<b>+</b>	Indica 1 ó más veces (el elemento que haya delante). Equivale a {1,}
<b>?</b>	Indica 0 ó 1 veces (el elemento que haya delante). Equivale a {0,1}
<b>!</b>	<p>Se utiliza para realizar una "búsqueda anticipada negativa", algo así como "excluir esto". La construcción de la expresión regular es un par de paréntesis con este formato:</p> <p><i>paréntesis de apertura, seguido de un signo de interrogación, seguido de un signo de exclamación, y este seguido de lo que excluimos, para acabar cerrando el paréntesis.</i></p> <p>Por ejemplo, para una expresión que no pueda tener la palabra "porra", se usará <b>.*(?!porra).*</b></p> <p>No se usa tanto para validar una cadena, sino en la extracción de elementos que cumplan un patrón, otro de los usos de las expresiones regulares que no se tratan en este capítulo</p>

## Metacaracteres básicos

<b>\d</b>	Dígito. Equivale a [0-9]
<b>\D</b>	No dígito. Equivale a [^0-9]
<b>\s</b>	Espacio en blanco.
<b>\S</b>	No espacio en blanco. Equivale a [^\s]
<b>\w</b>	Una letra mayúscula o minúscula, un dígito o el carácter _ Equivale a [a-zA-Z0-9_]
<b>\W</b>	Algo que no es "Una letra mayúscula o minúscula, un dígito o el carácter _" Equivale a [^\w]
<b>\b</b>	Límite de una palabra.