

`LUIS@zend.com` //En mayúsculas
`luis@aupep.org` //Dos @

Las expresiones regulares son patrones de búsqueda dentro de cadenas. Estos patrones se construyen mediante caracteres especiales que cumplen unas reglas determinadas.

Tabla 6.2. Reglas de expresiones regulares.

Ejemplo Regla

<code>aabb</code>	Los caracteres no especiales, como un grupo de letras, se escriben como de costumbre.
<code>^</code>	Indica que hay que buscar el patrón desde el principio de la cadena.
<code>\$</code>	Este símbolo obliga al patrón a cumplirse hasta el final de la cadena.
<code>.</code>	Simboliza cualquier carácter.
<code>*</code>	Indica que puede haber 0 o más instancias de una expresión.
<code>+</code>	Indica que puede haber 1 o más instancias de una expresión.
<code>[ab]</code>	Indica que se puede encontrar el carácter a o el b y la expresión sería correcta. Lo podemos acompañar de <code>[ab]*</code> para evaluar si hay varias letras seguidas del tipo a o b.
<code>[a-z]</code>	Indica que se puede elegir entre un rango de caracteres que va desde la a hasta la z.
<code>\</code>	Si queremos utilizar cualquiera de los anteriores caracteres como parte del patrón tendremos que utilizar éste símbolo para indicarle a PHP que es un literal.

Como ejemplo vamos a intentar hacer un patrón que verifique si un correo electrónico está correctamente construido o no. Sería fácil pensar en una expresión parecida a la siguiente:

`[a-z]+@[a-z]+\.`org

La primera parte, `[a-z]+`, nos dice que aceptará una letra, o conjunto de letras, sin espacios y en minúsculas, que pueden corresponderse con el nombre de usuario de la cuenta de correo. Después aceptará un símbolo `@`, seguido de otro conjunto de caracteres, correspondientes al servidor de correo. La última parte de la expresión la componen los símbolos `\.`, que indican que tiene que aparecer un punto que separe la descripción del servidor del dominio y la cadena `org`, que obliga a que todos los correos sean de dominio no gubernamental.