

EXPRESIONES REGULARES

Son patrones que se utilizan para hacer coincidir combinaciones de caracteres en cadenas.
Se utilizan para buscar y reemplazar patrones en cadenas de texto

SÍMBOLOS

.	Cualquier carácter
^	La cadena debe contener la expresión al principio
\$	La cadena debe contener la expresión al final
[...]	Definición de conjunto
[...][...]	Un conjunto seguido de otro
[^...]	Indica negación, el conjunto debe excluir la cadena
[a-z1-9]	Indica rango, el conjunto debe tener de a-z y 1-9
A B	Indica OR : o
AB	Indica concatenación

META CARACTERES

\d	Dígito, equivale a [0-9]
\D	No dígito, equivale a [^0-9]
\s	Espacio en blanco
\S	No es espacio en blanco, equivale a [^\s]
\w	Palabra, equivale a [a-zA-Z0-9_]
\W	No palabra, equivale a [^\w]
\b	Límite de una palabra

CUANTIFICADORES

{X}	Indica lo que va antes de las llaves y se repite X
{X,Y}	Se repite mínimo X y máximo Y
{X,}	Se repite mínimo X y sin límite máximo
*	Se repite 0 o más, {0,}
+	Se repite 1 o más, {1,}
?	Se repite 0 o 1, {0,1}

CLASES

PATTERN: Se utiliza para crear una representación compilada de expresiones regulares

MÉTODOS: compile() = compila una expresión regular en un objeto Pattern

matcher() = crea un objeto Matcher que se puede usar para realizar operaciones de coincidencia en una secuencia de caracteres

split() = divide una cadena en sub cadenas utilizando una expresión regular como delimitador

MATCHER: Realiza operaciones de coincidencia en una secuencia de caracteres interpretando un patrón.
Se implementa con la interfaz "MatchResult"

MÉTODOS : find() = busca la subsecuencia de entrada que coincide con el patrón

group() = devuelve la subsecuencia de entrada capturada en la última coincidencia

matches() = intenta hacer coincidir toda la secuencia de entrada contra el patrón

replaceAll() = reemplaza todas las subsecuencias de entrada que coinciden con el patrón de cadena específica