

# 1. Conceptos clave

- **Expresiones regulares** son una forma de buscar y manipular texto basándose en patrones.
  - **Metacaracteres principales:**
    - `.`: Representa **cualquier carácter** excepto un salto de línea.
    - `*`: Cero o más repeticiones del carácter o grupo anterior.
    - `+`: Una o más repeticiones del carácter o grupo anterior.
    - `?`: Cero o una repetición del carácter o grupo anterior.
    - `^`: Marca el **inicio de la cadena**.
    - `$`: Marca el **final de la cadena**.
    - `\\`: Escapa caracteres especiales como `.`, `*`, `+`.
- 

## 2. Patrones comunes

### Números y caracteres básicos

Patrón	Significado	Ejemplo de coincidencia
<code>[0-9]</code>	Un solo número (cifra).	1, 5, 9
<code>[a-z]</code>	Una letra minúscula.	a, m, z
<code>[A-Z]</code>	Una letra mayúscula.	A, M, Z
<code>[a-zA-Z]</code>	Una letra (mayúscula o minúscula).	A, z, m
<code>\d</code>	Un dígito (equivalente a <code>[0-9]</code> ).	3, 0, 7
<code>\w</code>	Un carácter alfanumérico (o guión bajo).	a, 3, _
<code>\s</code>	Un espacio en blanco (espacio, tab).	" " (espacio), <code>\t</code> (tab)

### Repeticiones

Patrón	Significado	Ejemplo de coincidencia
--------	-------------	-------------------------

<code>a*</code>	Cero o más <code>a</code> .	<code>`, a, aaaa</code>
<code>a+</code>	Una o más <code>a</code> .	<code>a, aa, aaaa</code>
<code>a?</code>	Cero o una <code>a</code> .	<code>`, a</code>
<code>a{3}</code>	Exactamente tres <code>a</code> .	<code>aaa</code>
<code>a{2,5}</code>	Entre dos y cinco <code>a</code> .	<code>aa, aaa, aaaaa</code>

## Ubicación

Patrón	Significado	Ejemplo de coincidencia
<code>^hola</code>	Cadena que empieza con <code>hola</code> .	<code>hola mundo</code> (pero no <code>mundo hola</code> )
<code>hola mundo\$</code>	Cadena que termina con <code>mundo</code> .	<code>hola mundo</code>
<code>^hola\$</code>	Cadena que es exactamente <code>hola</code> .	Solo <code>hola</code>

## 3. Ejemplos útiles de patrones

### Validación de datos

Tipo	Patrón	Ejemplo válido
Correo	<code>^[A-Za-z0-9+_.-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\$</code>	<code>user@example.com</code>
Teléfono	<code>^\d{9,}\$</code> (9+ números)	<code>123456789</code>
Fecha (dd/mm/yyyy)	<code>^[0-2][0-9]/[0-1][0-9]/[0-1][0-9][0-9]\$</code>	<code>3[0-1]/[01-9]</code>
URL	<code>^(http</code>	<code>https)://[\w.-]+\.[a-zA-Z]{2,}(/.)*?\$`</code>

## 4. Herramientas para practicar

## 4. Usar **Pattern** y **Matcher** en Java

Para usar expresiones regulares en Java, debes compilar un patrón con `Pattern.compile()` y luego usar un **Matcher** para comprobar si una cadena coincide con ese patrón.

### Métodos clave de **Pattern** y **Matcher**:

- **Pattern.compile(String regex)**: Compila la expresión regular en un patrón.
- **matcher.find()**: Busca una coincidencia en el texto (devuelve `true` si la encuentra).
- **matcher.matches()**: Comprueba si toda la cadena coincide con la expresión regular.
- **matcher.replaceAll(String replacement)**: Reemplaza todas las coincidencias con una cadena dada.
- **matcher.group()**: Obtiene el grupo de coincidencia actual.

### Ejemplo de uso de **matcher.find()**

java

Copiar código

```
1. import java.util.regex.*;
2.
3. public class Ejemplo {
4.     public static void main(String[] args) {
5.         // Ejemplo de texto con números de teléfono nacionales
        e internacionales
6.         String texto = "Mi número de teléfono es +34
        123456789, el otro es +1 987654321, y también tengo el
        612345678";
7.
8.         // Expresión regular para validar teléfonos con
        prefijo internacional o sin prefijo
9.         String patron = "(\\+\\d{1,3} \\d{9}|\\d{9})"; //
        Coincide con un número internacional con prefijo o un número
        nacional sin prefijo
10.
11.         Pattern pattern = Pattern.compile(patron);
12.         Matcher matcher = pattern.matcher(texto);
13.
14.         // Buscar y mostrar coincidencias
15.         while (matcher.find()) {
```

```
16.         System.out.println("Número encontrado: " +  
    matcher.group());  
17.     }  
18. }  
19. }  
20.  
21.
```

**22.**