

Javadoc

Etiquetas Javadoc en Java

Las etiquetas Javadoc permiten estructurar mejor la documentación generada a partir de los comentarios. A continuación, se detallan las principales **etiquetas** utilizadas:

1. Etiquetas para Información General

@author Nombre

- Indica el autor del código.
- Se usa generalmente en la documentación de clases.
- **Ejemplo:**

```
/**
 * Clase que representa a una persona.
 *
 * @author Profeprogramacion
 */
public class Persona{ }
```

@version número

- Especifica la versión del código o la clase.
- **Ejemplo:**

```
/**
 * @version 1.0
 */
```

@since versión

- Indica desde qué versión existe una funcionalidad.
- **Ejemplo:**

```
/**
 * @since 1.8
 */
```

2. Etiquetas para Métodos y Parámetros

@param nombre descripción

- Explica el uso de cada parámetro de un método.
- **Ejemplo:**

```
/**
 * Calcula la suma de dos números.
 *
 * @param a Primer número.
 * @param b Segundo número.
 * @return La suma de ambos números.
 */
public int sumar(int a, int b) {
    return a + b;
}
```

@return descripción

- Describe lo que devuelve un método.
- **Ejemplo:**

```
/**
 * Convierte un texto a mayúsculas.
 *
 * @param texto Texto de entrada.
 * @return Texto en mayúsculas.
 */
public String convertirMayusculas(String texto) {
    return texto.toUpperCase();
}
```

@throws excepción descripción

- Indica qué excepciones puede lanzar un método.
- **Ejemplo:**

```
/**
 * Divide dos números.
 *
 * @param a Dividendo.
 * @param b Divisor.
 * @return Resultado de la división.
 * @throws ArithmeticException Si b es 0.
 */
public double dividir(int a, int b) throws ArithmeticException {
    if (b == 0) {
        throw new ArithmeticException("No se puede dividir por cero.");
    }
    return (double) a / b;
}
```

3. Etiquetas para Referencias y Enlaces

@see referencia

- Agrega una referencia a otra clase, método o documentación externa.
- **Ejemplo:**

```
/**
 * Convierte una cadena en mayúsculas.
 *
 * @see String#toUpperCase()
 */
```

@link referencia

- Inserta un enlace dentro del texto sin interrumpir la lectura.
- **Ejemplo:**

```
/**
 * Usa el método {@link #convertirMayusculas(String)} para transforma
 * r el texto.
 */
```

@linkplain referencia

- Similar a `@link`, pero sin formato de código.
- **Ejemplo:**

```
/**
 * Para más información, consulta {@linkplain String#toLowerCase()}.
 */
```

4. Etiquetas para Advertencias y Notas

@deprecated

- Indica que un método o clase está obsoleto.
- Puede usarse con `@see` para sugerir alternativas.
- **Ejemplo:**

```
/**
 * @deprecated Usar {@link #nuevoMetodo()} en su lugar.
 */
@Deprecated
public void metodoAntiguo() { }
```

@todo descripción

- No es estándar, pero se usa para marcar tareas pendientes.
- **Ejemplo:**

```
/**
 * @todo Implementar la función de restar.
```

```
*/
```

5. Etiquetas para Código y Formato

@code {código}

- Inserta código en línea.
- **Ejemplo:**

```
/**
 * Ejemplo de uso: {@code System.out.println("Hola");}
 */
```

@literal texto

- Muestra texto sin interpretar caracteres especiales.
- **Ejemplo:**

```
/**
 * Usa {@literal <b>} para negritas en HTML.
 */
```

Resumen

Etiqueta	Descripción
@author	Autor del código
@version	Versión del código
@since	Desde qué versión está disponible
@param	Documenta los parámetros de un método
@return	Explica el valor devuelto por un método
@throws	Indica excepciones que puede lanzar un método
@see	Agrega una referencia a otra clase o método
@link	Inserta un enlace dentro del texto
@linkplain	Similar a @link , pero sin formato de código

@deprecated	Indica que un método o clase está obsoleto
@todo	Marca tareas pendientes (no estándar)
@code	Muestra código en línea
@literal	Muestra texto sin interpretar caracteres especiales

Cómo generar javadoc en IntelliJ

- Tools → generate javadoc → creamos y seleccionamos un directorio (javadoc en raíz) y generamos
- Se abre en el navegador el HTML con el código generado