

REFLEXIÓN SOBRE EL USO DE JAVADOC

1. ¿Me resultó fácil usar Javadoc?

Sí, fue súper fácil, especialmente porque NetBeans hace todo el trabajo pesado.

- Lo más sencillo: Simplemente tienes que escribir los comentarios especiales (barra-asterisco-asterisco) justo encima del método.
- La magia: Al darle a "Generate Javadoc", el IDE transforma ese texto en una página HTML bonita y navegable, lista para que cualquiera la entienda. Es como si el IDE creará el "manual de usuario" de tu código por ti.

2. ¿Qué ventajas y desventajas he encontrado?

Ventajas:

- Evita el "¿Qué hace esta función?": En un proyecto web grande, tienes un montón de funciones (`getUsers()`, `formatDate()`, `calculateTax()`). La documentación Javadoc te permite saber exactamente qué hace cada una, qué datos necesita (`@param`) y qué devuelve (`@return`), sin tener que leer línea por línea el código de otra persona.
- Ahorro de tiempo brutal: Si un frontend necesita usar una API backend escrita en Java, Javadoc le da el endpoint claro sin necesidad de mandar 50 mensajes de Slack.
- Mantiene el orden: Impone una sola forma de documentar, lo cual es oro cuando trabajas con un equipo de 5 personas.

Desventajas:

- Dependencia del Formato: Javadoc solo genera HTML nativamente. La conversión a otros formatos solicitados (PDF y Markdown) requiere pasos adicionales (impresión a PDF y uso de conversores online).
- La subida a GitHub es fastidiosa: Para que la documentación HTML se visualice correctamente online (como en GitHub Pages), no basta con subir el archivo `.html`; tienes que subir un montón de carpetas (`resource-files`, `script-files`) que solo contienen CSS y JavaScript, lo cual ensucia un poco la carpeta del repositorio.

3. ¿Recomiendo usarlo en proyectos colaborativos?

¡Absolutamente sí, 100%!

En el desarrollo web colaborativo, donde la gente usa diferentes lenguajes y frameworks:

- Facilita la legibilidad y el trabajo en equipo: Si tu código está bien documentado, el desarrollador que venga después de ti (o incluso tú mismo dentro de seis meses) podrá entenderlo sin estrés. Es un seguro de vida contra el código "indescifrable".
- Es el estándar: Usar Javadoc te enseña a documentar profesionalmente, una habilidad que se traduce directamente a otras tecnologías de documentación como TSDoc (en TypeScript) o JsDoc (en JavaScript).

Conclusión: Documentar es tan importante como escribir el código. Javadoc es una herramienta simple y potente para garantizar que tu trabajo sea escalable y fácil de mantener en equipo.