



DESPLIEGUE DE APLICACIONES WEB CURSO: 2º DAW

Autor: Javier Gómez-Comino Comino



3 DE OCTUBRE DE 2025

Contenido

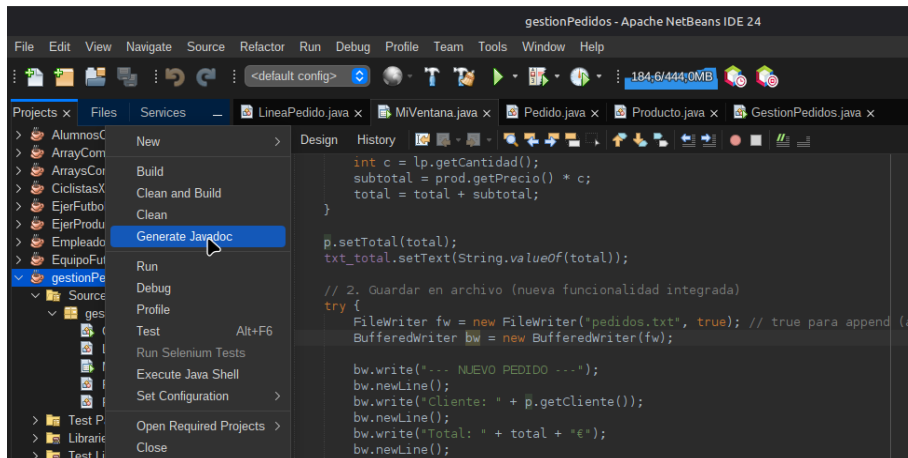
1.	Comparativa de Herramientas de Documentación	2
2.	Aplicación práctica en el programa del curso anterior	2
3.	Guardar el JavaDoc en HTML	4
4.	Reflexión Personal	5

1. Comparativa de Herramientas de Documentación

Característica	Doxygen	Sphinx	MkDocs
Enfoque Principal	Documentación de código fuente (cubre la mayoría de los lenguajes).	Documentación de proyectos y manuales (nació para Python).	Documentación orientada a proyectos, simple y rápida.
Facilidad de Uso	Media. Requiere un archivo de configuración (Doxyfile) extenso.	Media a Difícil. Necesita conocimiento de reStructuredText (reST).	Alta. Usa Markdown simple, y la configuración es mínima (mkdocs.yml).
Integración con Proyectos	Excelente. Se integra con CMake, Makefiles, y es muy personalizable.	Excelente. Integración nativa con Python (setup.py, pip), pero funciona bien con otros.	Buena. Se integra fácilmente con sistemas de control de versiones como Git y GitHub Pages.
Formatos de Salida	HTML, PDF (vía LATEX), RTF, XML, man pages.	HTML, PDF (vía LATEX), ePub, Texinfo, man pages.	HTML (estático), PDF (a través de plugins o impresión en navegador).
Comunidad	Muy grande y madura. Es un estándar histórico en C++, C, y otros lenguajes.	Grande, especialmente en el ecosistema Python y para documentación técnica general.	Creciente y activa. Popular por su simplicidad y el uso de Markdown.
Curva de Aprendizaje	Media. Aprender la sintaxis de marcado y la configuración del Doxyfile.	Media/Alta. Aprender reStructuredText y el sistema de extensiones.	Baja. Dominar la sintaxis de Markdown y la simple configuración.
Uso Ideal	Documentar código fuente en detalle, especialmente librerías y APIs en lenguajes como C++, Java, o PHP.	Documentar proyectos grandes, tutoriales, y manuales, especialmente en Python.	Documentar proyectos sencillos o medianos, creando documentación rápida y bien presentada.

2. Aplicación práctica en el programa del curso anterior

Selecciono un proyecto de java del año pasado que hice en netbeans de gestionar pedidos para poder realizar en mi caso el JavaDoc con exito:



Añado comentarios dentro del código de java para saber qué hace cada instrucción:

```

public class Producto {

    /**
     * Constructor por defecto que inicializa un producto sin parámetros.
     */
    public Producto() {
    }

    /**
     * Constructor que inicializa un producto con todos sus atributos.
     *
     * @param nombre El nombre del producto
     * @param categoria La categoría a la que pertenece el producto
     * @param precio El precio unitario del producto
     * @param stock La cantidad disponible en inventario
     */
    public Producto(String nombre, String categoria, double precio, int stock) {
        this.nombre = nombre;
        this.categoria = categoria;
        this.precio = precio;
        this.stock = stock;
    }

    /**
     * Obtiene el nombre del producto.
     *
     * @return El nombre del producto
     */
    public String getNombre() {
        return nombre;
    }

    /**
     * Establece el nombre del producto.
     *
     * @param nombre El nuevo nombre del producto
     */
}

```

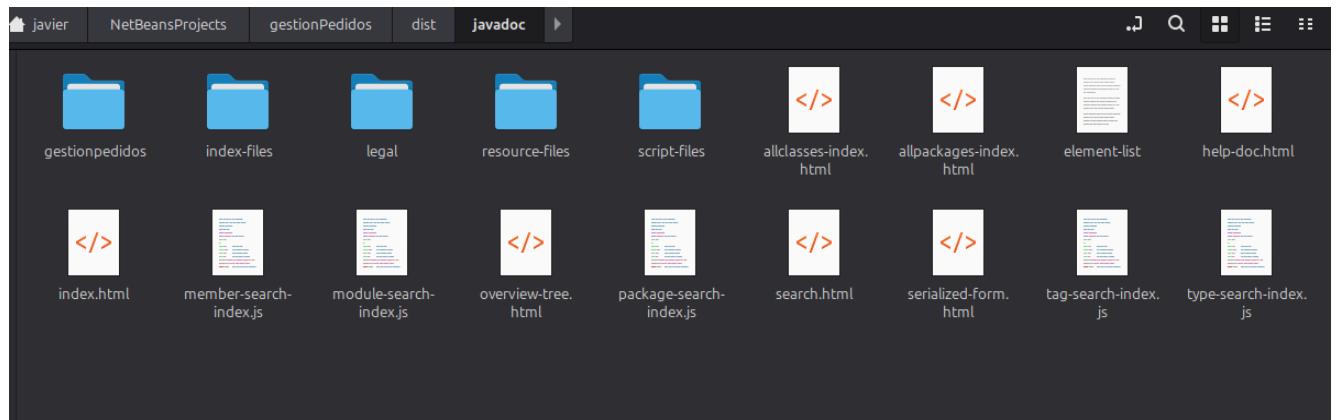
2.2 Generar el JavaDoc en Java

Para hacer el “Java doc” hay que darle clic derecho en el proyecto de java en el lado izquierdo y le daremos al botón que dice “JavaDoc”, nos manda a una página con todos los métodos y clases que hay:



3. Guardar el JavaDoc en HTML

Aquí lo que hago es identificar donde está el html dentro de la carpeta de Java Doc.



Y hará también un java doc en la página web que se genera.



4. Reflexión Personal

¿Qué tan fácil fue usar la herramienta?

La experiencia de utilizar Javadoc fue relativamente sencilla, ya que esta herramienta viene integrada dentro del JDK y no requiere instalaciones adicionales. La forma de uso es intuitiva: basta con añadir comentarios especiales en el código fuente siguiendo el formato `/** ... */` y después ejecutar el comando `javadoc` para generar automáticamente la documentación. Esto hace que el proceso sea muy accesible para cualquier programador que ya esté trabajando en proyectos Java.

¿Qué ventajas/desventajas ?

Entre las principales ventajas encontradas está el hecho de que la documentación se genera directamente a partir de los comentarios en el código, lo que asegura que se mantenga actualizada mientras el equipo documente adecuadamente. Además, el resultado en formato HTML es claro, navegable y profesional, lo que facilita la consulta de clases, métodos y atributos. Otra ventaja es que Javadoc es un estándar muy utilizado en la comunidad Java, por lo que existen numerosos recursos y guías para aprender a sacarle provecho.

Como **desventajas**, se puede mencionar que el formato de salida es principalmente HTML, por lo que si se requieren otros formatos (como PDF o Markdown) es necesario recurrir a herramientas adicionales o convertidores. Además, si no se documenta el código desde el inicio, el proceso de añadir comentarios a posteriori puede resultar tedioso y consumir tiempo extra.

¿La recomendarían para un proyecto colaborativo?

En conclusión, recomendaría Javadoc para proyectos colaborativos en Java, ya que fomenta buenas prácticas de documentación, mejora la comprensión del código y facilita la incorporación de nuevos miembros al equipo.