DOCUMENTACIÓN DEL CÓDIGO EN JAVA CON JAVADOC

INTRODUCCIÓN

La documentación del código es el proceso de escribir descripciones detalladas y claras sobre las funciones, clases, métodos y otros componentes del software. Este proceso ayuda a otros desarrolladores (y a ti mismo en el futuro) a entender qué hace el código, cómo interactuar con él y cómo mantenerlo.

¿Por qué es Importante Documentar el Código?

1. Facilita el Mantenimiento: La documentación clara permite que otros desarrolladores (o tú mismo en el futuro) comprendan rápidamente el propósito y el funcionamiento del código, lo que facilita la corrección de errores y la implementación de mejoras.

2. Mejora la Colaboración: En equipos de desarrollo, una buena documentación es esencial para que todos los miembros puedan entender y trabajar con el código sin confusiones.

3. Aumenta la Reusabilidad: El código bien documentado puede ser reutilizado más fácilmente en otros proyectos porque es más comprensible.

4. Ayuda en la Formación: Los nuevos miembros del equipo pueden aprender más rápido con una buena documentación, reduciendo el tiempo de formación y aumentando la productividad.

INTRODUCCIÓN A JAVADOC

Javadoc es una herramienta incluida en el JDK de Java que permite generar documentación en formato HTML a partir de comentarios en el código fuente. Estos comentarios deben seguir una sintaxis específica que Javadoc entiende y procesa.

Comentarios Javadoc

Los comentarios Javadoc se escriben entre `/** ... */` y se colocan justo antes de la declaración de clases, métodos, o campos. Dentro de estos comentarios, se utilizan etiquetas específicas como

- `@author`, autor/es del código.
- `@param`, para especificar los parámetros del método.
- `@return`, para especificar que es lo que retorna el método.
- `@throws`o @exception`, para describir las excepciones que lanza el método.
- `@version`, versión del código.
- `@deprecated`, indica que el método ha sido “descatalogado”
- `@since`, nos indica desde que versión está disponible el método.

Ejemplo Práctico con Javadoc en IntelliJ IDEA

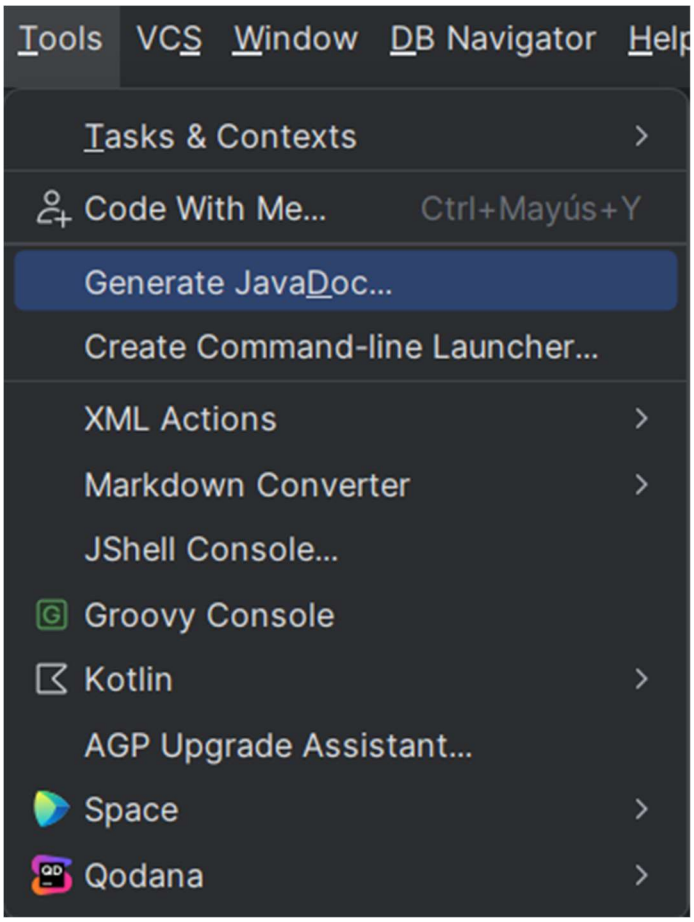
Vamos a ver un ejemplo de cómo documentar una clase simple en Java usando Javadoc y cómo generar la documentación en IntelliJ IDEA.

Paso 1: Crear la Clase Java (con los comentarios tal y como hemos dicho anteriormente, a cada uno de las clases y métodos)

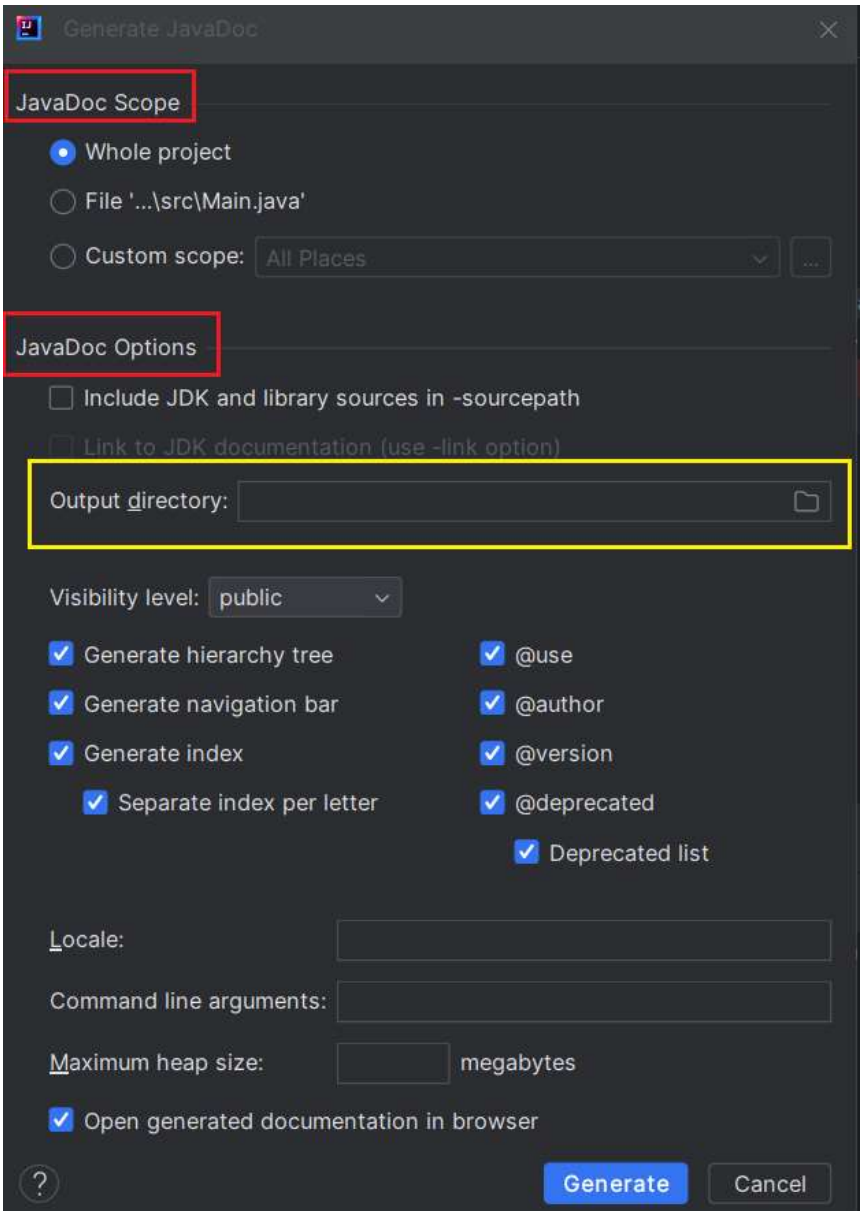
```java
public class Calculadora {
    /**
     * Suma dos números enteros.
     *
     * @param a el primer número a sumar
     * @param b el segundo número a sumar
     * @return la suma de a y b
     */
    public int sumar(int a, int b) {
        return a + b;
    }

    /**
     * Resta dos números enteros.
     *
     * @param a el número del que se resta
     * @param b el número que se resta
     * @return la diferencia de a y b
     */
    public int restar(int a, int b) {
        return a - b;
    }
}
```

Paso 2: Generar la Documentación con Javadoc en IntelliJ IDEA

- Ve al menú **Tools** - **Herramientas**.
- Selecciona **Generate Javadoc... - Generar Javadoc...**

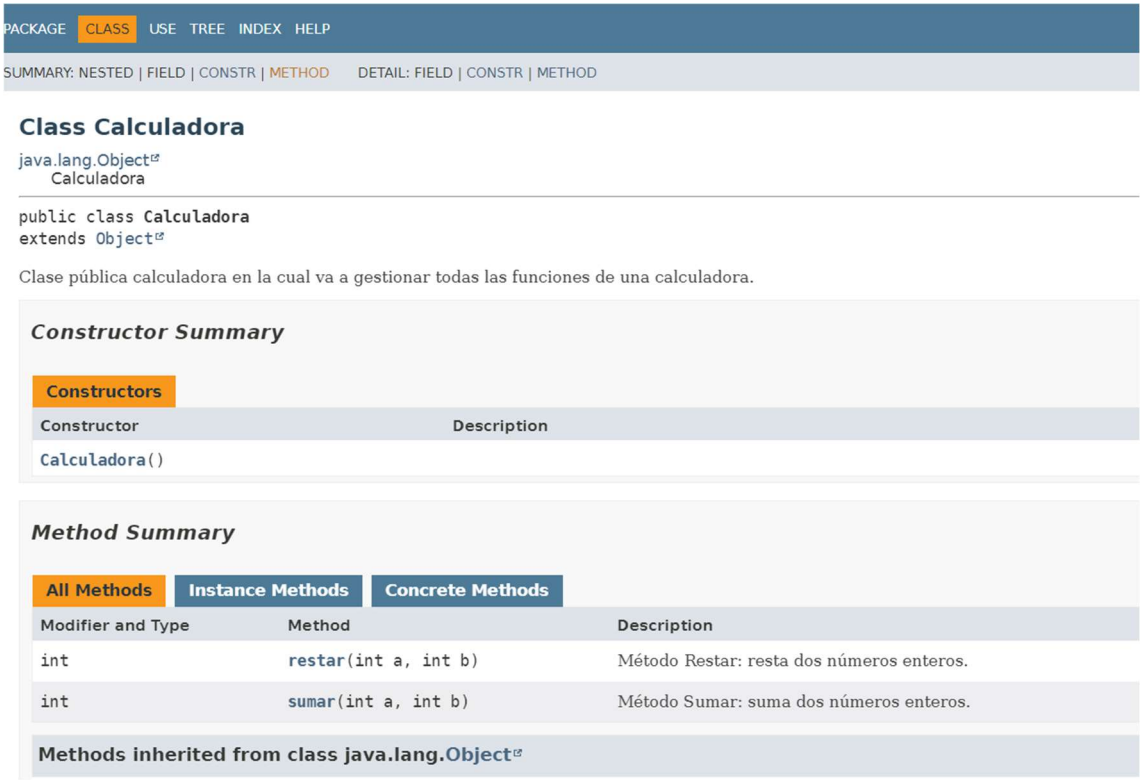


- En la ventana que aparece, configura las opciones según tus necesidades:
    - Output Directory – Directorio de salida: Selecciona el directorio donde quieres que se guarden los archivos HTML generados.
    - Scope - Ámbito: Selecciona el alcance (por ejemplo, solo el código de producción o también el de pruebas).
    - Other Options – Otras opciones: Puedes añadir opciones adicionales como generar documentación privada o incluir enlaces a la documentación estándar de Java.



3. Ejecutar la Generación: haz clic en OK para generar la documentación. IntelliJ IDEA ejecutará Javadoc y creará los archivos HTML en el directorio especificado.

4.-Ver la Documentación: Navega al directorio de salida especificado y abre el archivo `index.html` en tu navegador web para ver la documentación generada.



PACKAGE CLASS USE TREE INDEX HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class Calculadora

java.lang.Object
Calculadora

```
public class Calculadora
extends Object
```

Clase pública calculadora en la cual va a gestionar todas las funciones de una calculadora.

Constructor Summary

Constructors

Constructor	Description
Calculadora()	

Method Summary

All Methods **Instance Methods** **Concrete Methods**

Modifier and Type	Method	Description
int	restar(int a, int b)	Método Restar: resta dos números enteros.
int	sumar(int a, int b)	Método Sumar: suma dos números enteros.

Methods inherited from class java.lang.Object

CONCLUSIÓN

Documentar el código es una práctica esencial para cualquier desarrollador. Utilizar herramientas como Javadoc no solo facilita la creación de documentación detallada y profesional, sino que también mejora la mantenibilidad y la colaboración dentro del equipo de desarrollo. IntelliJ IDEA proporciona una integración sencilla para generar esta documentación, haciendo el proceso aún más eficiente.

Documentar tu código es una inversión en el futuro de tu proyecto y en la calidad de tu trabajo como desarrollador.