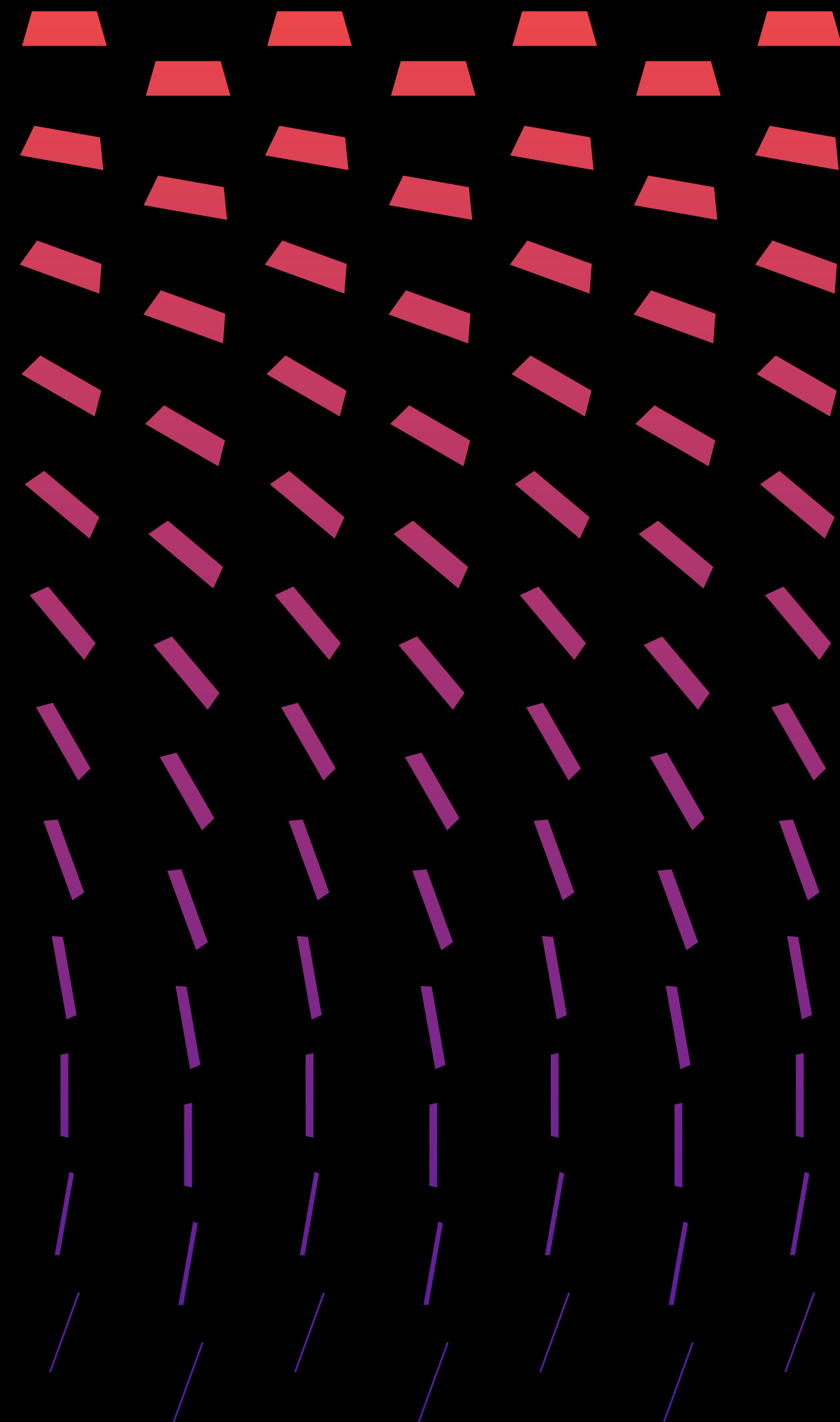


Estilos de codificación y documentación de código en Java (JavaDoc)

Octubre 2024

Taller de Documentación de SW



Contenidos

1 Introducción a JavaDoc

2 Estilo y Consistencia

3 Ejemplos Prácticos

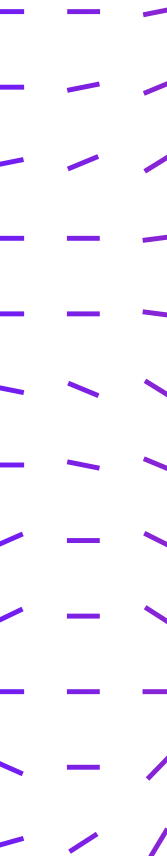
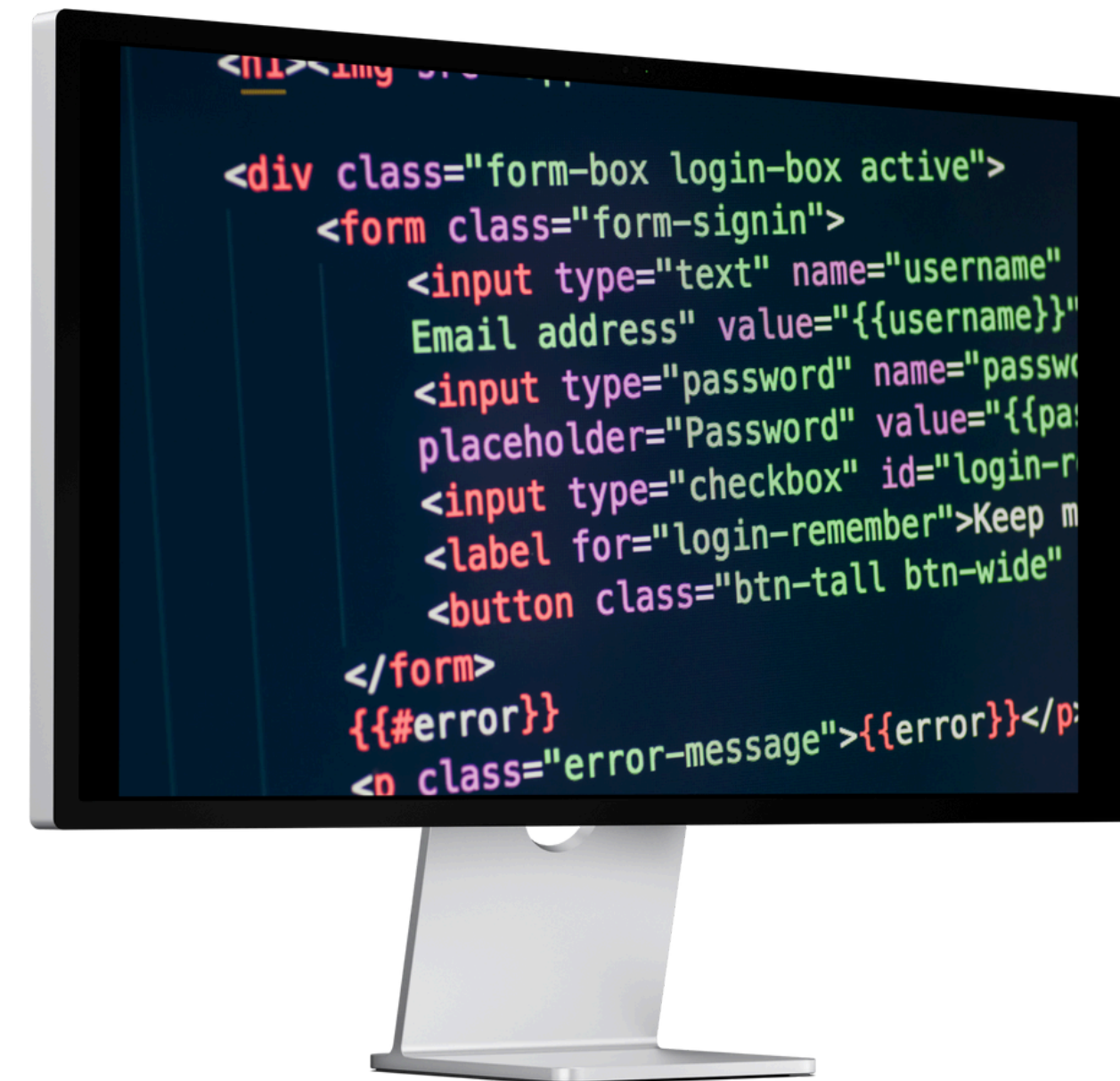
4 Extensiones y Complementos para IDEs

5 Conclusiones



JavaDoc

La herramienta JavaDoc es un generador de documentos que utiliza el lenguaje de programación Java para generar documentación estándar en formato HTML. Crea documentación de API. Analiza una serie de archivos fuente que contienen declaraciones y documentación de clases, métodos, constructores y campos.



¿Por qué documentar tu código?

- Puede ayudar a los desarrolladores a comprender el código con mayor rapidez, depurar y corregir errores con más eficacia, refactorizar y mejorar la calidad del código.
- Evitar duplicaciones e incoherencias, documentar decisiones lógicas y de diseño,
- Escribir pruebas y crear documentación externa para usuarios y clientes.



Beneficios de usar JavaDoc



Genera documentación comprensible en HTML directamente desde el código.



Facilita la comprensión del código tanto para los desarrolladores actuales como para futuros



Permite generar documentación en un formato estandarizado y fácil de leer



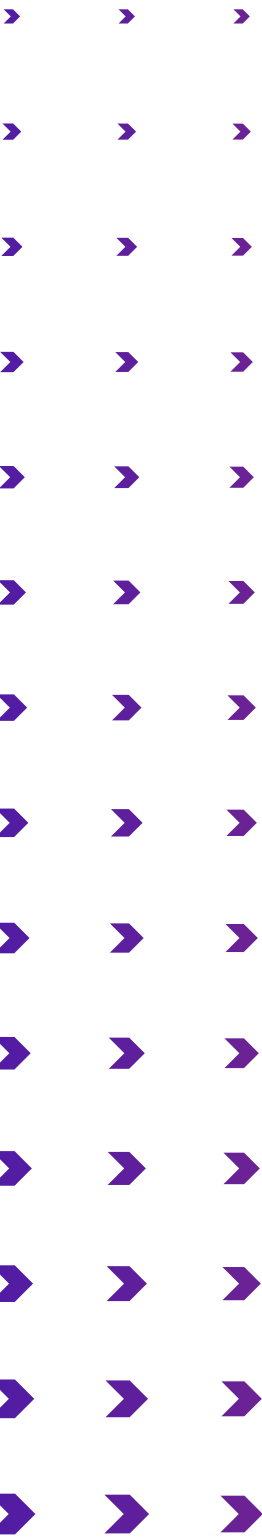


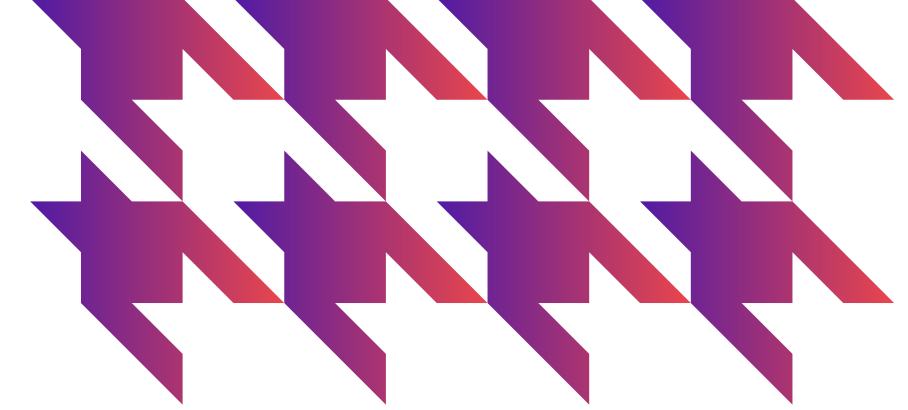
Estilo y Consistencia

Los **comentarios** Javadoc son esenciales, ya que proporcionan una forma estructurada de describir clases, métodos y campos.

Una buena documentación debe ser

- **Clara:** Explica lo que hace el código, cómo funciona y por qué.
- **Completa:** Cubrir todos los aspectos, incluyendo clases, métodos, campos, parámetros, valores de retorno y excepciones.
- **Coherente:** Mantener un estilo y una estructura uniformes para mejorar la legibilidad.





Comentarios de estilo JavaDoc

→ Cada comentario de documentación comienza con:

`"/**"`

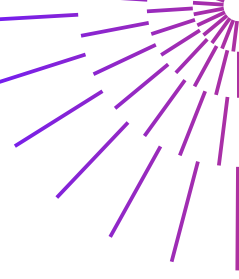
y termina con

`"*/"`

¿Cuándo hay que poner un comentario?

-
- Al principio de cada clase
 - Al principio de cada método
 - Ante cada variable de clase

Javadoc tags



Etiqueta	Descripción
@author	Describe un autor (Nombre del desarrollador)
@version	Proporciona la versión de la clase, metodo o interfaz
@see	Genera un enlace a otro elemento del documento
@param	Parámetro que acepta el método o constructor.
@deprecated	Permite a los usuarios saber que la clase o el método ya no se utilizan.
{@link}	Se utiliza para crear vínculos a otras clases o métodos.
@return	Lo que devuelve el método.
@throws	Tipo de excepción que produce el método.
@Override	Realiza una comprobación para ver si el método es una invalidación.

Ejemplos prácticos de Javadoc

```
/**
 * Calcula la suma de 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;
}
```

Proporciona una descripción concisa de lo que hace el método. Explica que su función principal es sumar dos números enteros, sin entrar en detalles adicionales.

@param a: Representa el primer número entero que será sumado.

@param b: Describe el segundo número entero que será sumado a a.

@return: Se retorna la suma de los dos números enteros (a + b).

Ejemplos prácticos de Javadoc

```
/**
 * Calcula el promedio de una lista de números enteros.
 *
 * @param numeros una lista de números enteros sobre los cuales se calculará el promedio
 * @return el promedio de los números en la lista, o 0 si la lista está vacía
 * @throws IllegalArgumentException si la lista es null
 */
public double calcularPromedio(List<Integer> numeros) {
    if (numeros == null) {
        throw new IllegalArgumentException("La lista de números no puede ser null");
    }
    if (numeros.isEmpty()) {
        return 0;
    }
    int suma = 0;
    for (int numero : numeros) {
        suma += numero;
    }
    return (double) suma / numeros.size();
}
```

La primera línea (Calcula el promedio de una lista de números enteros.) describe brevemente lo que hace el método, es decir, calcular el promedio de una lista de enteros.

@param numeros: Describe el parámetro numeros, que es una lista de números enteros. Se aclara que el método calcula el promedio de los elementos de esta lista.

@return: Explica que el método devuelve un número double que representa el promedio de los números en la lista.

@throws IllegalArgumentException: Se usa para indicar que el método lanzará una excepción si el parámetro numeros es null

Ejemplo Práctico con JavaDoc

```
public class ParImpar {  
  
    public boolean esPar(int numero) {  
        return numero % 2 == 0;  
    }  
  
    public boolean esImpar(int numero) {  
        return numero % 2 != 0;  
    }  
  
    public int[] contarParesImpares(int inicio, int fin) {  
        int[] conteo = new int[2];  
  
        for (int i = inicio; i <= fin; i++) {  
            if (esPar(i)) {  
                conteo[0]++;  
            } else {  
                conteo[1]++;  
            }  
        }  
  
        return conteo;  
    }  
}
```



Declaración de la Clase



Método esPar



Método esImpar



Método contarParesImpares



Bucle para Contar Pares e Impares:



Retorno del Resultado



Extenciones y complementos para IDEs

Visual Studio Code:

Complementos para mejorar la generación de
JavaDoc.

NetBeans:

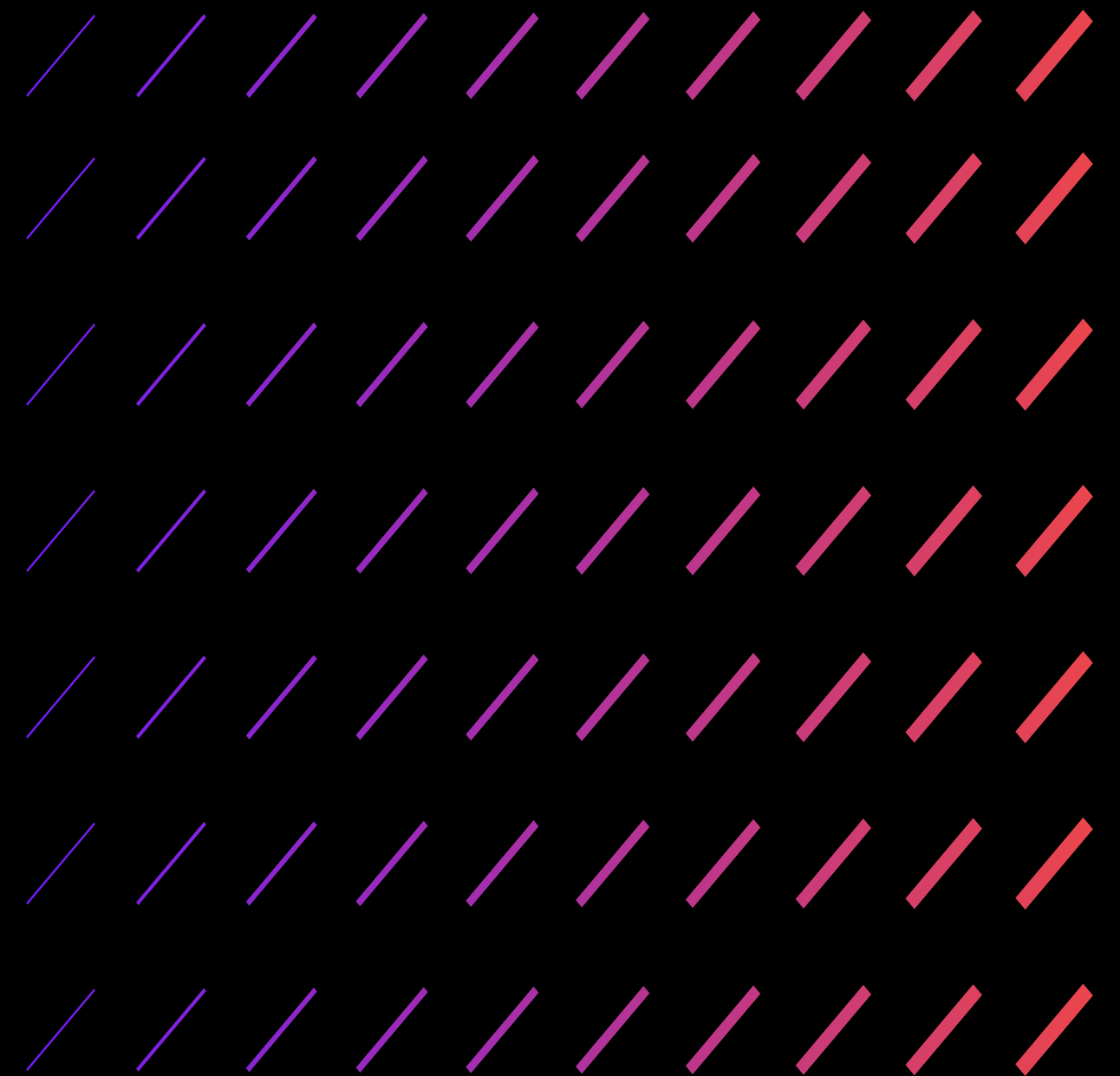
Generación automática y visualización en tiempo
real.

IntelliJ IDEA:

Herramientas integradas y complementos
adicionales (como SmartDoc).

Conclusión

En resumen, JavaDoc es una herramienta que genera documentación clara y estandarizada en HTML a partir del código Java. Ayuda a que los desarrolladores comprendan y mantengan el código más fácilmente. Usar JavaDoc mejora la calidad del software al evitar errores y facilitar la colaboración, con soporte para varios IDEs que simplifican su uso(JAVA).



Referencias

- What are the most important Java code documentation practices? (2024). Linkedin.com. https://www.linkedin.com/advice/O/what-most-important-java-code-documentation-practices-66due?utm_source=share&utm_medium=guest_desktop&utm_campaign=copy
- Documentación de código. (2024). Dit.upm.es. <https://www.dit.upm.es/~pepe/doc/adsw/base/doc/doc.htm>
- What is JavaDoc tool and how to use it? (2019, April 8). GeeksforGeeks. <https://www.geeksforgeeks.org/what-is-javadoc-tool-and-how-to-use-it/>
- Javadoc. (2022, December 12). Wikipedia. <https://en.wikipedia.org/wiki/Javadoc>
- Java coding standards and Javadoc style comments. (n.d.). Www2.Hawaii.edu. https://www2.hawaii.edu/~tp_200/ics111/material/codingStandards.html
- Java: Javadoc tags. (2015, January 1). I'd Rather Be Writing Blog and API Doc Course. <https://idratherbewriting.com/java-javadoc-tags/>
- Ramuglia, G. (2023, October 31). Javadoc Comments: Javadoc Usage Guide. Linux Dedicated Server Blog. <https://ioflood.com/blog/javadoc-comments/>