

IES JUAN BOSCO
DESARROLLO DE APLICACIONES WEB



U.T. 1

Práctica: **Evaluación y uso de un generador de documentación**

DESPLIEGUE DE APLICACIONES WEB

Curso: 2º DAW

INDICE

1 - Investigación breve.....	3
1-1. Tabla comparativa	3
2 - Aplicación práctica en el programa del curso anterior	4
2-1. Seleccionar el proyecto desarrollado en el curso pasado.	4
2-2. Añadir comentarios en el código.....	4
2-3. Generar el JavaDoc en Java.....	5
2-4. Guardar el JavaDoc en HTML.....	6
3 - Reflexión	7

1 - Investigación breve

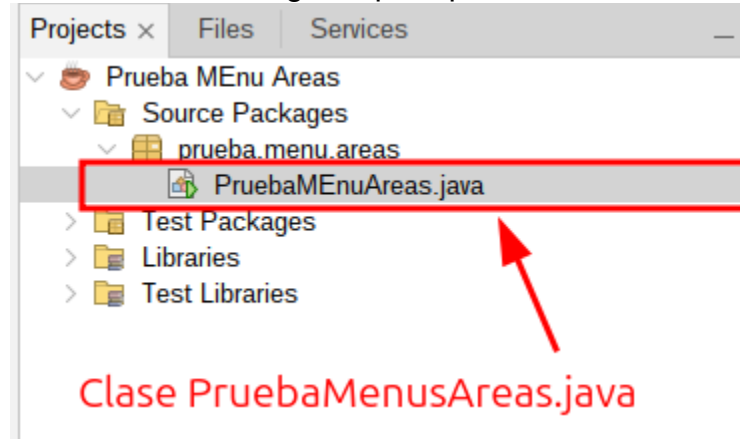
1-1. Tabla comparativa

	JavaDoc	JSDoc	Swagger
Facilidad de uso	Muy fácil: ya viene con Java y se genera con un comando.	Muy fácil: solo instalar con npm y comentar el código.	Fácil: necesita definir especificaciones en JSON/YAML.
Integración con proyectos	Excelente en proyectos Java, ya que está integrado en el JDK.	Buena integración en proyectos JavaScript y Node.js , soporta TypeScript.	Excelente para proyectos con APIs REST . Compatible con muchos lenguajes y frameworks.
Formatos de salida	Principalmente HTML . Se puede exportar a PDF con herramientas externas.	HTML principalmente, algunos plugins permiten Markdown/PDF.	HTML interactivo , JSON, YAML. Puede exportar a PDF con herramientas adicionales.
Comunidad	Muy amplia (usada en casi todos los proyectos Java).	Activa y en crecimiento, usada en la comunidad JS.	Muy grande, estándar en la industria para APIs.
Curva de aprendizaje	Baja: solo hay que aprender etiquetas básicas	Baja-media: requiere aprender anotaciones y configurar a veces plantillas.	Media: requiere aprender la especificación OpenAPI.

2 - Aplicación práctica en el programa del curso anterior

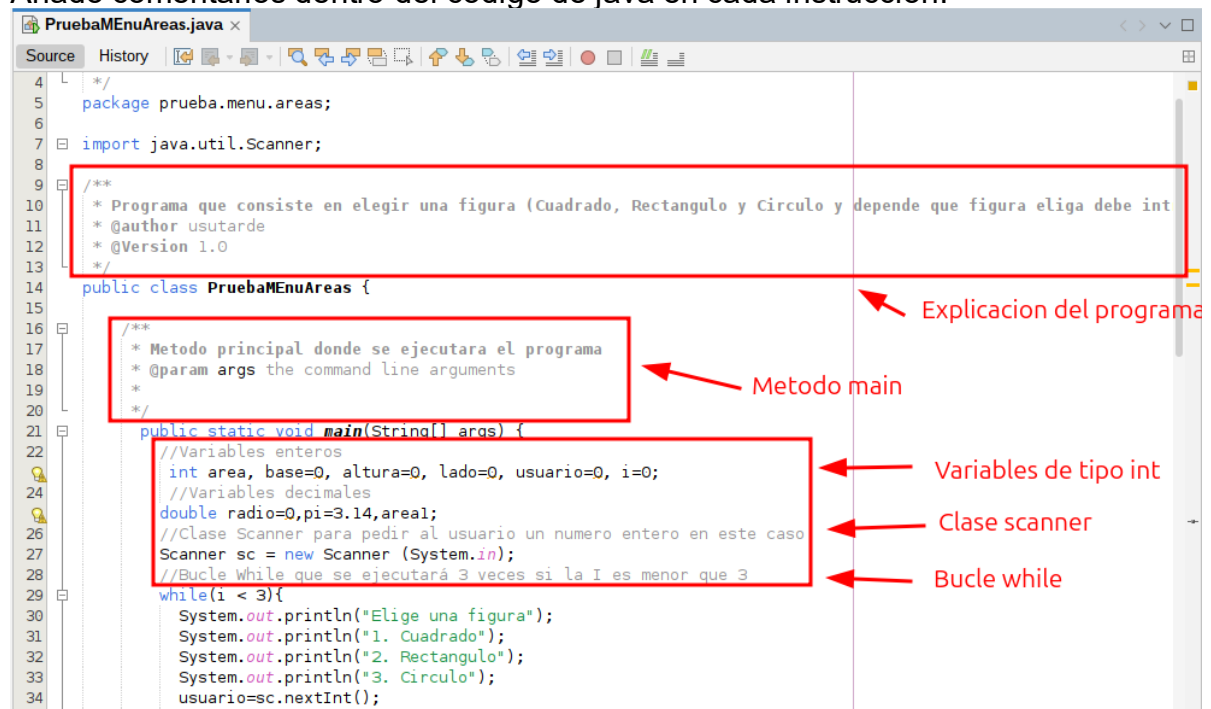
2-1. Seleccionar el proyecto desarrollado en el curso pasado.

Selecciono un proyecto de java del año pasado que hice en netbeans de calcular áreas en distintas figuras para poder realizar en mi caso el JavaDoc con éxito:



2-2. Añadir comentarios en el código

Añado comentarios dentro del código de java en cada instrucción:



```

//Condicion if en caso de elegir el cuadrado
if(usuario ==1){
    System.out.println("Has elegido el cuadrado");

    System.out.println("Introduce el lado del cuadrado");
    lado=sc.nextInt();
    area=lado * lado;
    System.out.println("El area del cuadrado es:"+ area);
}

//Condicion if en caso de elegir el rectangulo
if(usuario==2){
    System.out.println("Has elegido el rectangulo");

    System.out.println("Introduce la base del rectangulo");
    base = sc.nextInt();
    System.out.println("Introduce la altura del rectangulo");
    altura= sc.nextInt();
    area= base * altura;
    System.out.println("El area del rectangulo es: "+area);
}

//Condicion if en caso de elegir el circulo
if(usuario==3){
    System.out.println("Has elegido el circulo");

    System.out.println("Introduce el area del circulo");
    radio = sc.nextInt();
    areal= radio * radio * pi;
    System.out.println("El area del circulo es:"+areal);
}

i++;
}

```

IF cuadrado

IF Rectangulo

IF Circulo

2-3. Generar el JavaDoc en Java

Para ello hay que darle clic derecho en el proyecto de java en el lado izquierdo y nos dira JavaDoc, se crea automáticamente y nos manda a una pagina con todos los métodos y clases que hay:

PACKAGE CLASS USE TREE INDEX SEARCH HELP

prueba.menu.areas > PruebaMenuAreas

Contents Filter

Description

Constructor Summary

Method Summary

Constructor Details

PruebaMenuAreas()

Method Details

main(String[])

Class PruebaMenuAreas

java.lang.Object[®]
prueba.menu.areas.PruebaMenuAreas

public class PruebaMenuAreas
extends Object[®]

Programa que consiste en elegir una figura (Cuadrado, Rectangulo y Circulo y depende que figura elija debe introducir los datos necesarios para saber el area de la figura seleccionada

Constructor Summary

Constructor	Description
PruebaMenuAreas()	

Method Summary

All Methods Static Methods Concrete Methods

Modifier and Type	Method	Description
static void	main(String [®] [] args)	Metodo principal donde se ejecutara el programa

Methods Inherited from class java.lang.Object[®]

clone[®], equals[®], finalize[®], getClass[®], hashCode[®], notify[®], notifyAll[®], toString[®], wait[®], wait[®], wait[®]

Constructor Details

PACKAGE CLASS USE TREE INDEX SEARCH HELP

prueba.menu.areas > PruebaMEnuAreas

Contents Filter

Description

Constructor Summary

Method Summary

Constructor Details

PruebaMEnuAreas()

Method Details

main(String[])

All Methods Static Methods Concrete Methods

Modifier and Type	Method	Description
static void	main(String[] args)	Metodo principal donde se ejecutara el programa

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Details

PruebaMEnuAreas

public PruebaMEnuAreas()

Method Details

main

public static void main(String[] args)

Metodo principal donde se ejecutara el programa

Parameters:

args - the command line arguments

2-4. Guardar el JavaDoc en HTML

Aquí lo que he hecho es identificar donde esta el HTML dentro de la carpeta JavaDoc:

Basí 1 daw y 2SMR Programación Java Prueba MEnu Areas CON IF dist javadoc

index-files legal prueba resource-files script-files allclasses-index.html allpackages-index.html element-list help-doc.html

index.html member-search-index.js module-search-index.js overview-tree.html package-search-index.js search.html tag-search-index.js type-search-index.js

index.html del javadoc

Y ahora un javadoc en la pagina web de donde me ha generado:

DAW 2CFGSDAW DAW: Por x Atajo Flameshot Linux x PruebaMEnuAreas x PruebaMEnuAreas x M-index

file:///media/usuario/Basi/1 daw y 2SMR/Programación/Java/Prueba MEnu Areas CON IF/dist/javadoc/index-files/index.html

PACKAGE TREE INDEX SEARCH HELP

Index

M P
All Classes and Interfaces | All Packages

M

main(String[]) - Static method in class prueba.menu.areas.PruebaMEnuAreas
Metodo principal donde se ejecutara el programa

M P
All Classes and Interfaces | All Packages

17:47:40

3 - Reflexión

¿Qué tan fácil fue usar la herramienta?

Sinceramente el JavaDoc debo decir que lo di el año pasado junto a java y en mi entorno de trabajo que era NetBeans no he tenido ningún tipo de problemas respecto a la generalización del JavaDoc.

¿Qué ventajas/desventajas encontraron?

-En javadoc la **ventaja** desde mi punto de vista es que es demasiado cómodo para por ejemplo cuando otra persona tiene que parar a mirar tu código o por su claridad en la web para entender mejor el funcionamiento según tu el código que hayas comentado.

-La **Desventaja** que he tenido es que he tenido problemas al generar el Javadoc porque no he puesto el control de versiones ni tampoco el @param args y me saltaba 2 errores en la consola de NetBeans para generar el javadoc, sobre lo demás, no he encontrado alguna que otra desventaja ni le veo una desventaja, he usado poco esta herramienta para generar y sinceramente veo mas ventajas que desventajas.

¿La recomendarían para un proyecto colaborativo?

Como dije anteriormente, el generar el javadoc o poner los comentarios para luego un compañero de proyecto vea tu código puede ser una idea muy buena y si recomendaría mucho el uso de javadoc para un proyecto colaborativo.