

# Práctica 1 - JavaDocs

Genera documentación de **JavaDocs** sobre alguno de los proyectos que hemos trabajado en esta asignatura. La documentación ha de tener:

- Documentar cada clase utilizando **@author**, **@version** y **@see** para referenciar y explicar otras clases o métodos que estén relacionados.
- Documentar uno o varios de los métodos más importantes de cada clase. Importante utilizar **@param** y **@return**.

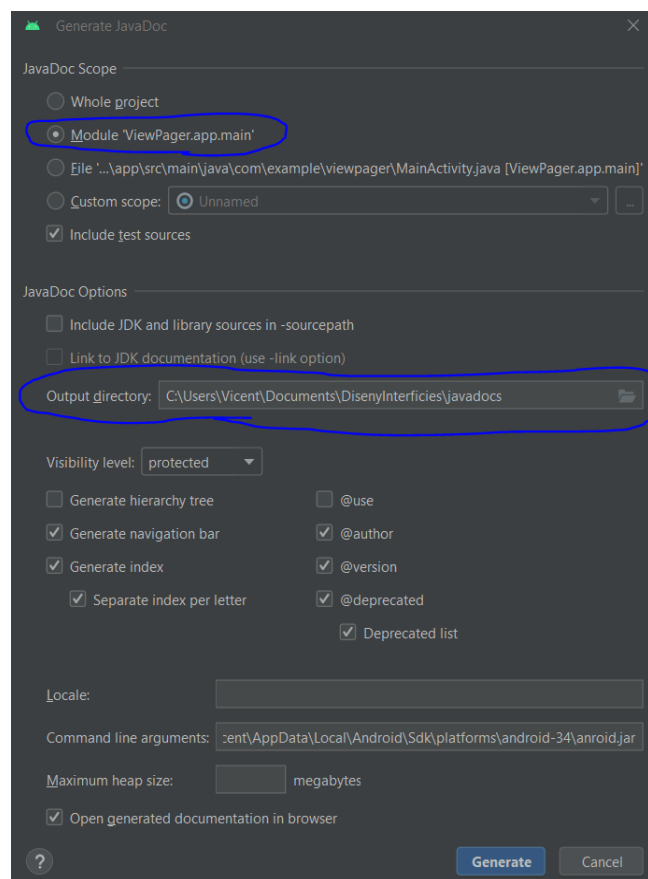
**Al finalizar entregad los archivos de la documentación comprimidos en un .zip**

## Ejemplo de utilización de JavaDoc:

Para añadir información a nuestra documentación escribiremos comentarios utilizando la siguiente sintaxis:

```
/**
 * MainActivity representa la actividad principal que utiliza un ViewPager2 y un TabLayout
 * para mostrar pestañas y contenido asociado.
 *
 * @see SectionsPagerAdapter Adaptador para gestionar las secciones dentro del ViewPager2.
 * @see PlaceholderFragment
 * @author Vicent Borja Roca
 * @version 2024-01-26
 */
2 usages
public class MainActivity extends AppCompatActivity {
```

Importante indicar el Scope(Proyecto entero o solo un paquete) y el directorio donde se guardará la documentación.



## Resultado:

PACKAGE

CLASS

INDEX

HELP

PACKAGE: DESCRIPTION | RELATED PACKAGES | CLASSES AND INTERFACES

SEARCH:

Package com.example.viewpager

package com.example.viewpager

Classes

Class	Description
MainActivity	MainActivity representa la actividad principal que utiliza un ViewPager2 y un TabLayout para mostrar pestañas y contenido asociado.
PlaceholderFragment	PlaceholderFragment es una clase que representa un fragmento dentro de SectionsPagerAdapter para mostrar contenido asociado a una pestaña específica.
SectionsPagerAdapter	

MainActivity representa la actividad principal que utiliza un ViewPager2 y un TabLayout para mostrar pestañas y contenido asociado.

Version:  
2024-01-26

Author:  
Vicent Borja Roca

See Also:

Adaptador para gestionar las secciones dentro del ViewPager2.,  
PlaceholderFragment

Abrimos con `/**` un comentario encima de la clase o método que queramos y **escribimos la descripción**.

Luego podemos indicar otros datos mediante etiquetas:

- **@author:** Indica el autor del código. Sin embargo, no es crítico en la generación de la especificación de la API y, por lo tanto, no siempre se incluye.
- **@version:** Proporciona información sobre la versión del código. La cadena SCCS proporciona detalles como el número de versión y la fecha de modificación.
- **@see nombre-clase:** Agregar referencias a otros elementos dentro de la documentación
- **@param nombre-param:** Documenta un parámetro del método. Incluye el nombre del parámetro seguido de una descripción que generalmente comienza con el tipo de datos del parámetro.
- **@return:** Documenta el valor de retorno de un método. Se omite para métodos que devuelven void y constructores, pero se incluye para todos los demás métodos.
- **@throws nombre-clase:** Documenta excepciones que puede arrojar un método. Debe incluirse para excepciones comprobadas.

