

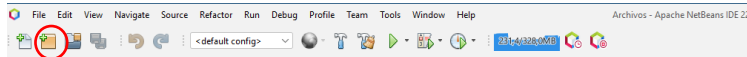
# EJEMPLO DE USO DE NETBEANS

## Contenido

Crear un proyecto .....	2
Guardar proyecto.....	2
Versionar.....	3
Recuperar desde el repositorio.....	5
Instalar <i>plugins</i> .....	5
Desarrollar.....	5
Compilar .....	6
Ejecutar .....	6
Probar (manual y automática) .....	7
Documentar .....	7
Generar documentación .....	8
Modelar .....	9
Depurar .....	9
Inspección de variables .....	11
Ejecución paso a paso .....	11
Comparar código .....	11
Refactorizar .....	12
Generar código.....	13
Ingeniería inversa .....	13
Administrar bases de datos.....	13
Ejecutar <i>script</i> sobre la base de datos.....	13

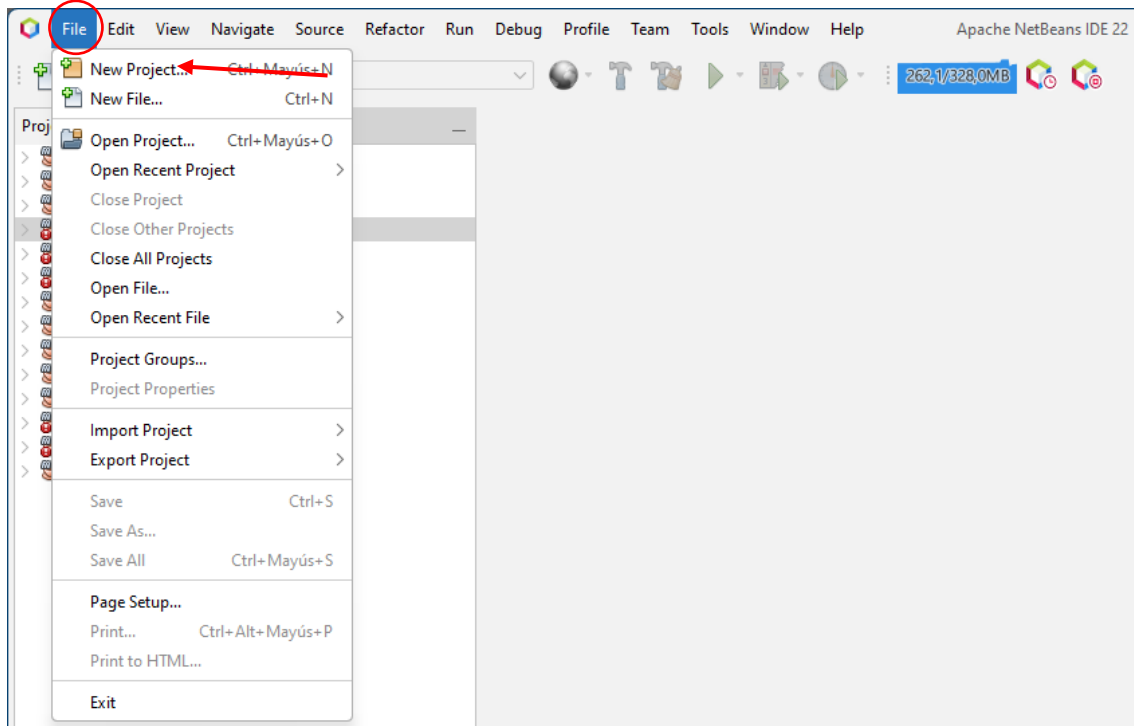
## Crear un proyecto

Para crear un proyecto, tendremos que seleccionar el siguiente icono



También se podría hacer desde el apartado File

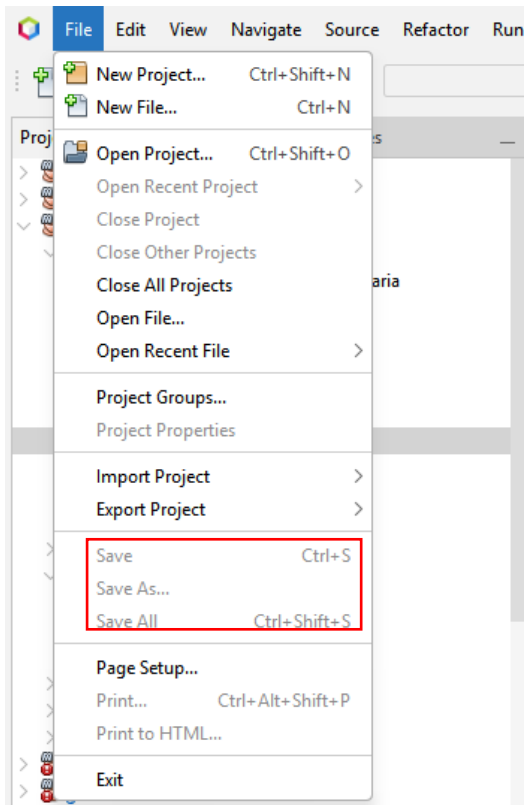
(O con los atajos de teclado: CTRL + Mayúscula + N)



## Guardar proyecto.

Hay 3 maneras:

- *Save* (Guarda el archivo)
- *Save as* (Guardar como)
- *Save all* (Guarda todo el proyecto)

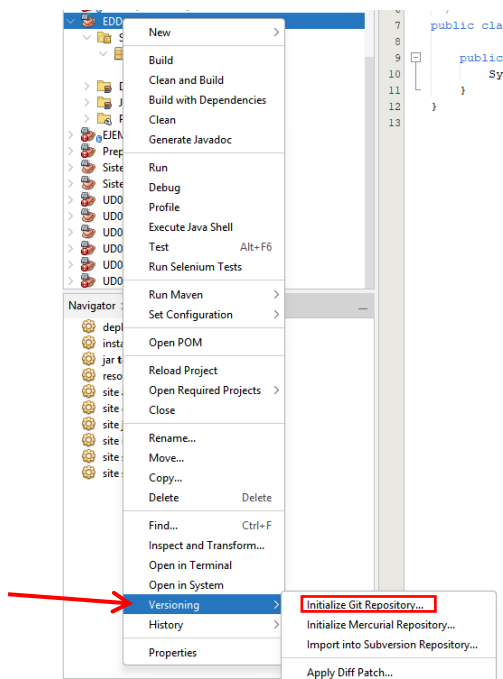


O también se podría desde el icono que aparece en el menú (Ahora mismo aparece de color gris ya que no hay nada que guardar, pero en el momento que hagamos cambios, aparecerá de color)

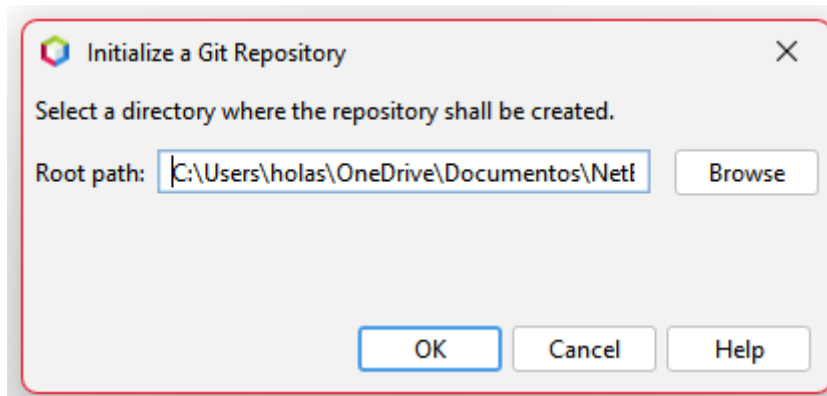


## Versionar.

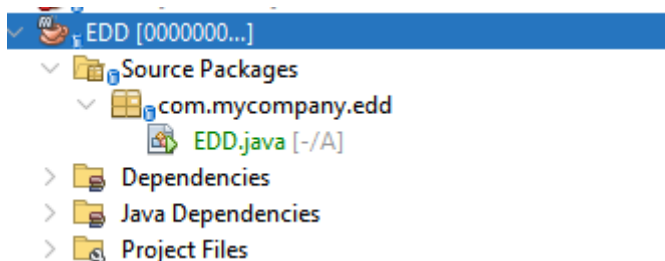
Click derecho sobre el proyecto que queramos versionar > Versioning > Initialize Git Repository



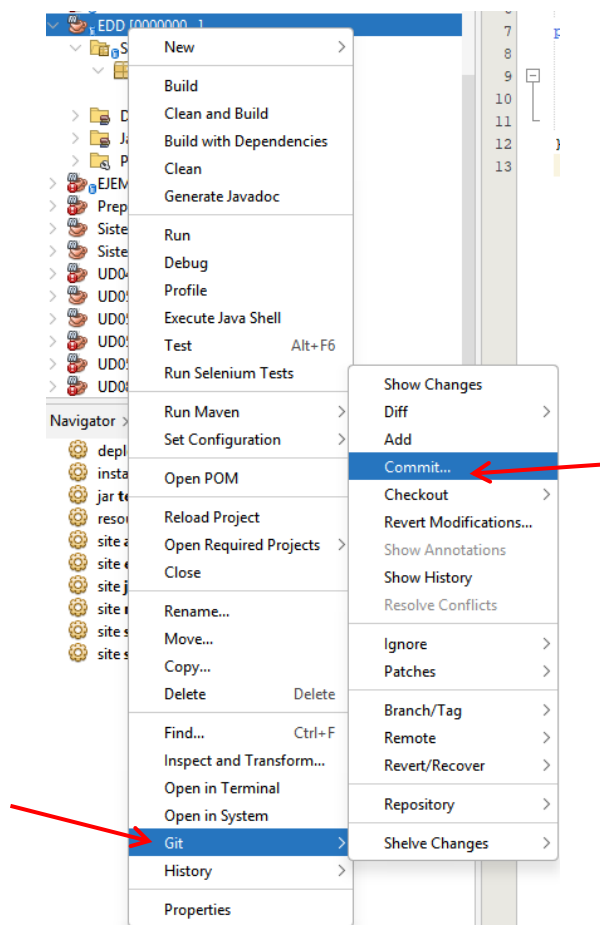
Seleccionamos la ruta del directorio que queremos versionar, en este caso, la ruta es la misma que donde está ubicado el proyecto. Le damos a OK para continuar



Esto significa que está correctamente inicializado



Para guardar la versión actual, hacemos *commit*

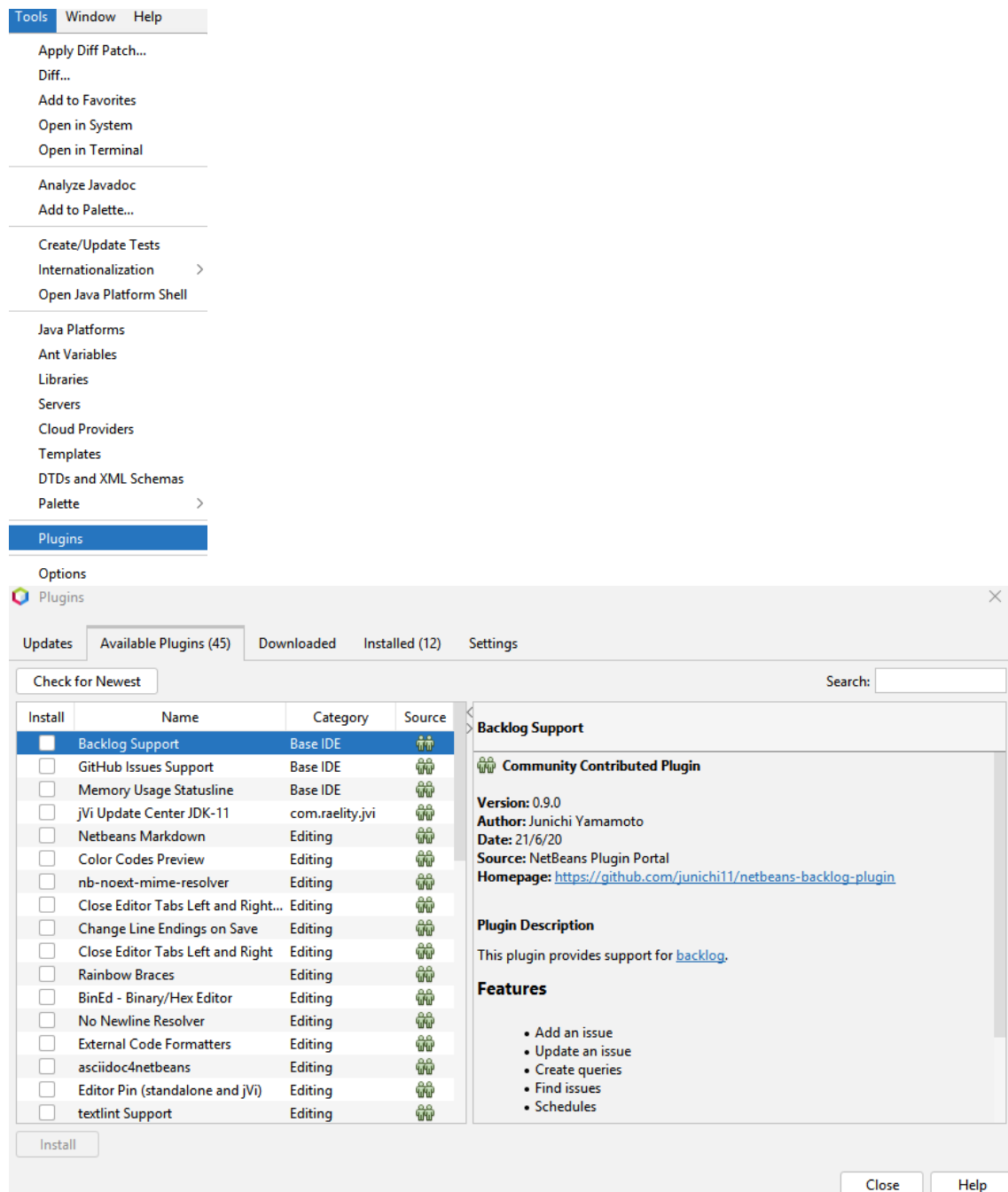


## Recuperar desde el repositorio.

**No sé hacer esto**

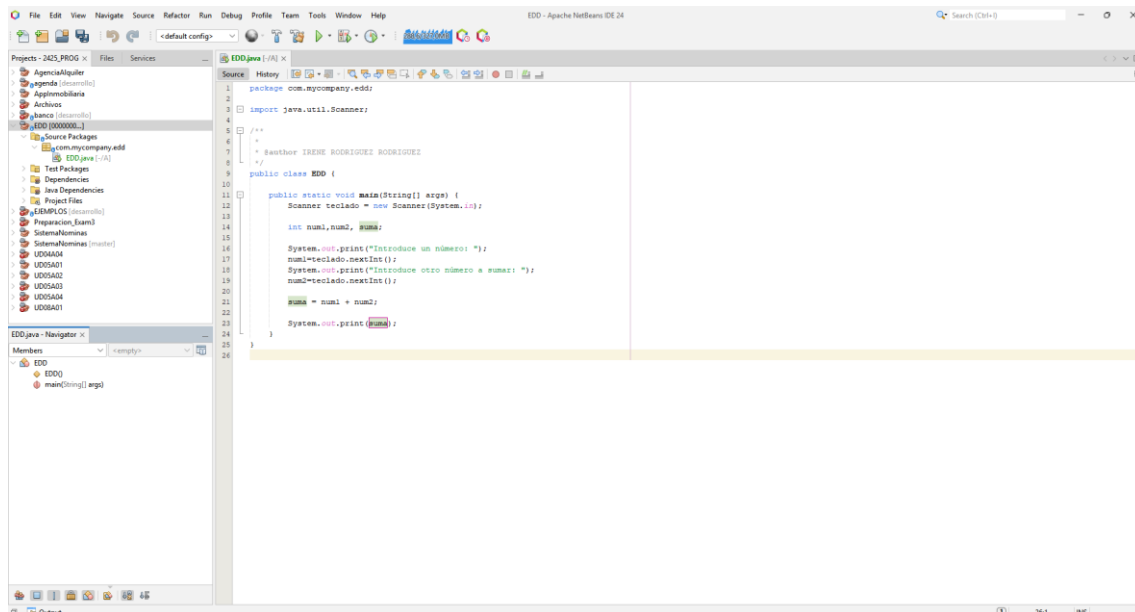
## Instalar *plugins*

*Tools > Plugins > Available plugins* (Y seleccionamos el que necesitamos)



## Desarrollar

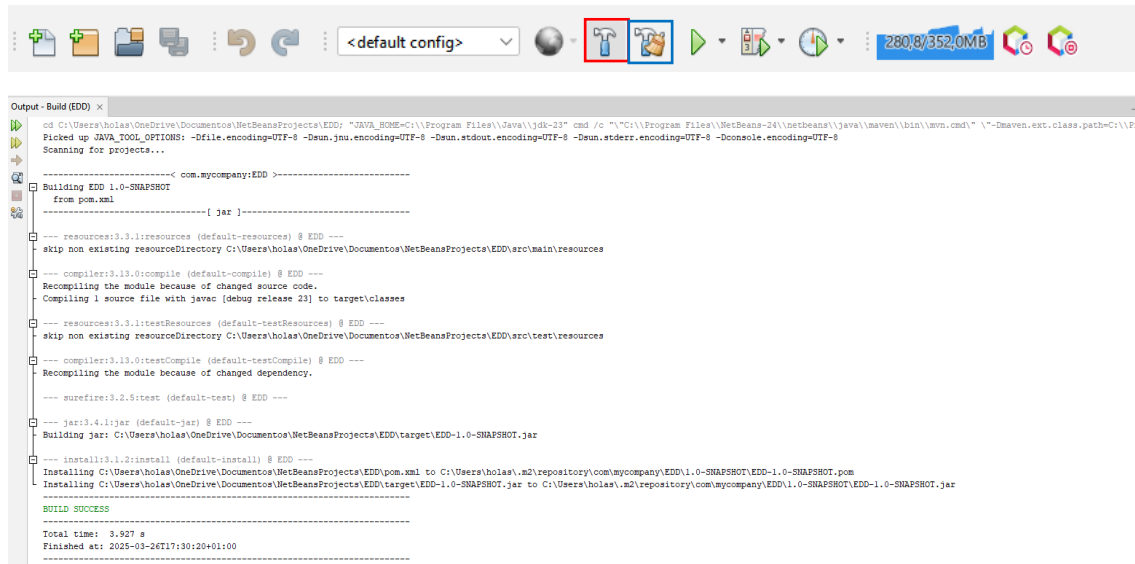
Generar código



## Compilar

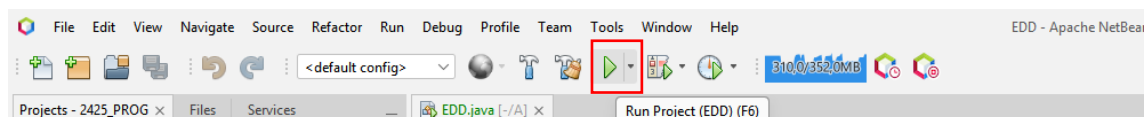
Hay dos maneras:

- **Build:** Compila el proyecto aprovechando los archivos generados anteriormente para ahorrar tiempo. (Es más eficiente que el siguiente). **Cuadrado rojo**
- **Clean and build:** Elimina todos los archivos generados previamente y recompila desde cero para garantizar una construcción limpia. **Cuadrado azul**



## Ejecutar

En caso de tener solo un *main* se haría con el siguiente botón, sino *click* derecho sobre el archivo que queremos ejecutar (OJO: tiene que ser una clase *main*)



```

Output - Run (EDD) x
Nothing to compile - all classes are up to date.

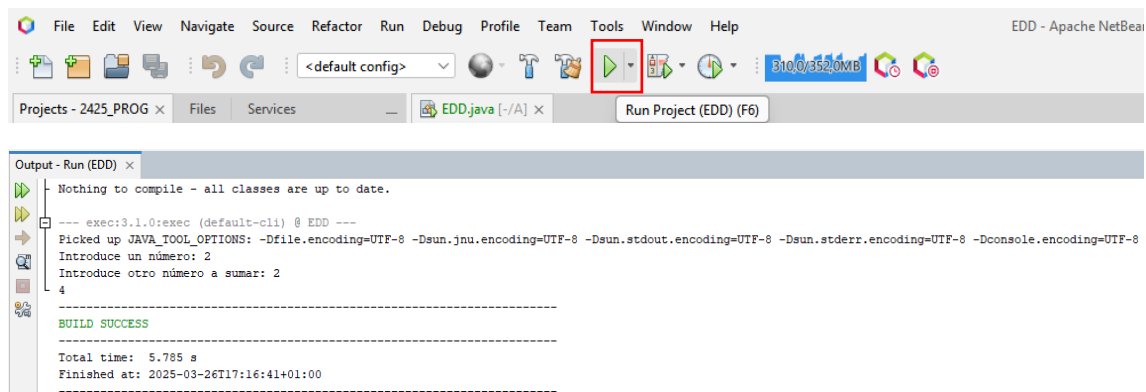
--- exec:3.1.0:exec (default-cli) @ EDD ---
Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding=UTF-8 -Dsun.jnu.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -Dconsole.encoding=UTF-8
Introduce un número: 2
Introduce otro número a sumar: 2
4

BUILD SUCCESS

Total time: 5.785 s
Finished at: 2025-03-26T17:16:41+01:00
    
```

## Probar (manual y automática)

En caso de tener solo un *main* se haría con el siguiente botón, sino *click* derecho sobre el archivo que queramos ejecutar (OJO: tiene que ser una clase *main*)



## Documentar

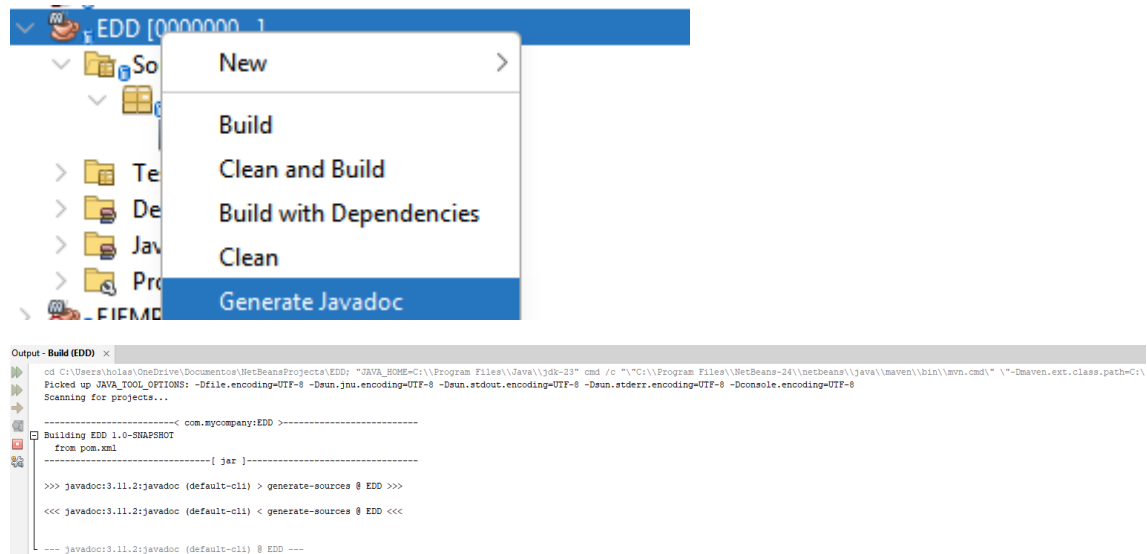
Consiste en agregar comentarios que ayudan a la hora de generar el JavaDoc sobre el funcionamiento de cada uno de los componentes del proyecto

```

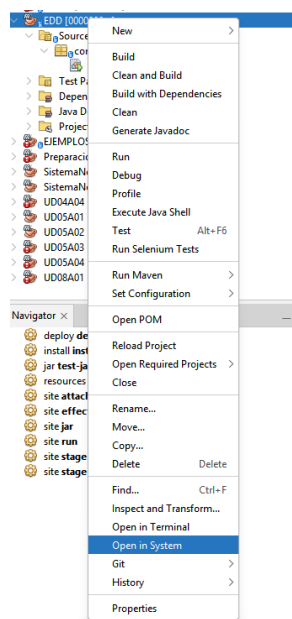
EDD.java [-/A] x
Source History
1 package com.mycompany.edd;
2
3 import java.util.Scanner;
4
5 /**
6  * Clase principal que realiza una suma de dos números ingresados por el usuario.
7  *
8  * Importa la clase Scanner para capturar la entrada del usuario desde la consola.
9  *
10  * @author IRENE RODRIGUEZ RODRIGUEZ
11  * @version 1.0
12  */
13 public class EDD {
14
15     /**
16      * Método principal del programa.
17      * Este método solicita dos números al usuario, calcula su suma
18      * y muestra el resultado en la consola.
19      *
20      * @param args Argumentos de línea de comandos, no utilizados en este programa.
21      */
22
23     public static void main(String[] args) {
24         Scanner teclado = new Scanner(System.in);
25
26         int num1, num2, suma;
27
28         System.out.print("Introduce un número: ");
29         num1=teclado.nextInt();
30         System.out.print("Introduce otro número a sumar: ");
31         num2=teclado.nextInt();
32
33         suma = num1 + num2;
34
35         System.out.print(suma);
36     }
37 }
38
    
```

## Generar documentación

Se da *click* derecho sobre el proyecto y seleccionamos *Generate Javadoc*



Para ver lo generado hacemos los siguiente



Se abrirá en nuestro sistema, y navegaremos hasta la siguiente ruta (`target\reports\apidocs`)

Nombre	Estado	Fecha de modificación	Tipo	Tamaño
.git	✓	26/03/2025 16:32	Carpeta de archivos	
src	✓	26/03/2025 17:09	Carpeta de archivos	
target	☁	26/03/2025 17:32	Carpeta de archivos	
.gitignore	✓	26/03/2025 16:37	Archivo de origen ...	1 KB
pom	✓	26/03/2025 16:26	Microsoft Edge H...	1 KB



Y este es el *JavaDoc* que se ha generado

Nombre	Estado	Fecha de modificación	Tipo	Tamaño
com	✓	26/03/2025 17:32	Carpeta de archivos	
legal	✓	26/03/2025 17:32	Carpeta de archivos	
resource-files	✓	26/03/2025 17:32	Carpeta de archivos	
script-files	✓	26/03/2025 17:32	Carpeta de archivos	
allclasses-index	✓	26/03/2025 17:32	Microsoft Edge H...	4 KB
allpackages-index	✓	26/03/2025 17:32	Microsoft Edge H...	4 KB
element-list	✓	26/03/2025 17:32	Archivo	1 KB
help-doc	✓	26/03/2025 17:32	Microsoft Edge H...	11 KB
index	✓	26/03/2025 17:32	Microsoft Edge H...	2 KB
index-all	✓	26/03/2025 17:32	Microsoft Edge H...	5 KB
member-search-index.js	✓	26/03/2025 17:32	JSFile	1 KB
module-search-index.js	✓	26/03/2025 17:32	JSFile	1 KB
overview-tree	✓	26/03/2025 17:32	Microsoft Edge H...	4 KB
package-search-index.js	✓	26/03/2025 17:32	JSFile	1 KB
search	✓	26/03/2025 17:32	Microsoft Edge H...	4 KB
tag-search-index.js	✓	26/03/2025 17:32	JSFile	1 KB
type-search-index.js	✓	26/03/2025 17:32	JSFile	1 KB

## Modelar

**No sé hacer esto**

## Depurar

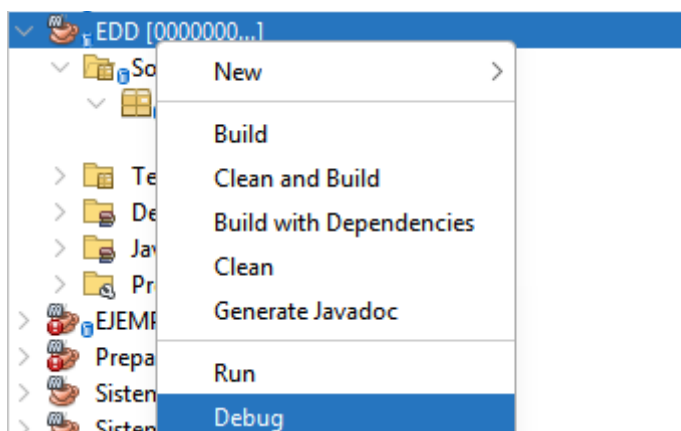
Añadimos puntos en los que se parará la aplicación mientras la probamos

```

1 package com.mycompany.edd;
2
3 import java.util.Scanner;
4
5 /**
6  * Clase principal que realiza una suma de dos números ingresados por el usuario.
7  *
8  * Importa la clase Scanner para capturar la entrada del usuario desde la consola.
9  *
10  * @author IRENE RODRIGUEZ RODRIGUEZ
11  * @version 1.0
12  */
13 public class EDD {
14
15     /**
16      * Método principal del programa.
17      * Este método solicita dos números al usuario, calcula su suma
18      * y muestra el resultado en la consola.
19      *
20      * @param args Argumentos de línea de comandos, no utilizados en este programa.
21      */
22
23     public static void main(String[] args) {
24         Scanner teclado = new Scanner(System.in);
25
26         int num1, num2, suma;
27
28         System.out.print("Introduce un número: ");
29         num1=teclado.nextInt();
30         System.out.print("Introduce otro número a sumar: ");
31         num2=teclado.nextInt();
32
33         suma = num1 + num2;
34
35         System.out.print(suma);
36     }
37 }
38

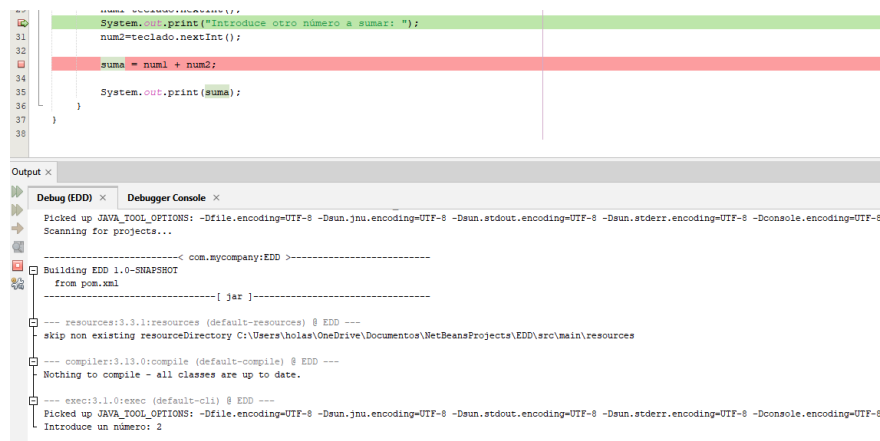
```

Iniciamos la depuración



Cambiará el color cuando lleguemos a un punto y saldrá un menú nuevo arriba





## Inspección de variables

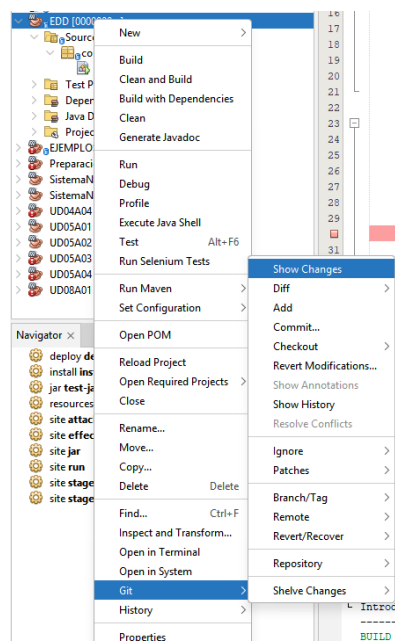
No sé hacer esto

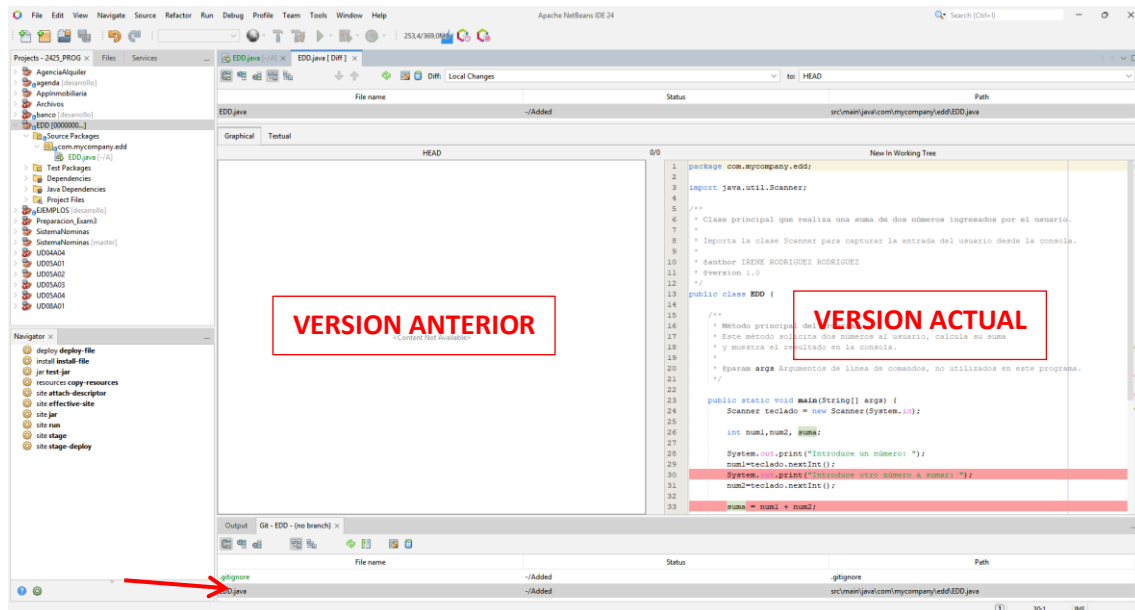
## Ejecución paso a paso

No sé hacer esto

## Comparar código

Click derecho > *Git* > *Show Changes* (Seleccionas un archivo a comparar y saldrá la versión que había antes y la actual. En este caso no sale nada porque no hay versiones anteriores a esta)

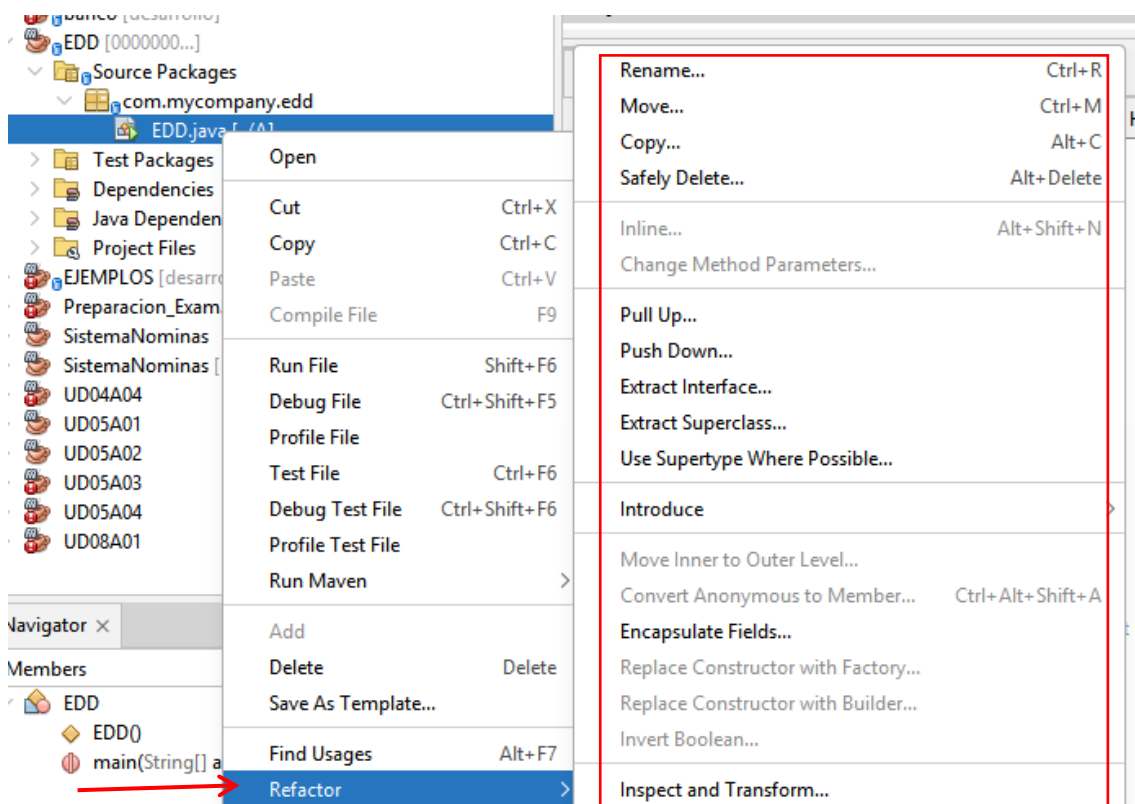




## Refactorizar

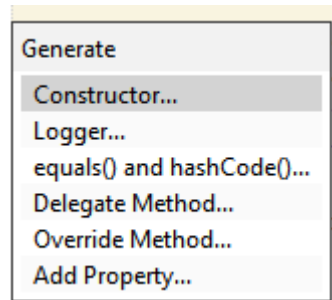
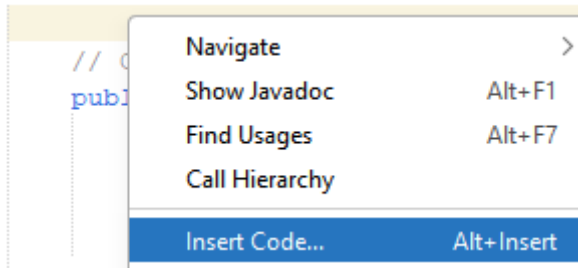
Click derecho en el archivo que estamos trabajando > *Refactor* > Seleccionamos lo que necesitamos. Sólo he usado los siguientes:

- *Rename* (Cambio de nombre al archivo)
- *Safely delete* (Borra un archivo sin afectar al proyecto entero)



## Generar código

Click derecho dentro del código > *Insert code*



## Ingeniería inversa

**No sé hacer esto**

## Administrar bases de datos

**No sé hacer esto**

## Ejecutar *script* sobre la base de datos

**No sé hacer esto**