

Contenido

1. Crear un Proyecto	2
3. Versionar (Git).....	3
4. Recuperar desde el Repositorio	6
5. Instalar Plugins (Extensiones)	7
6. Desarrollar (Escribir Código y Gestionar Archivos)	8
7. Compilar	8
8. Ejecutar Código	9
9. Probar (Manual y Automática)	10
10. Documentar	10
11. Generar Documentación (Javadoc, Doxygen, etc.)	10
12. Modelar (Diagramas UML y más)	11
13. Depurar (Debugging)	11
14. Inspección de Variables.....	12
15. Ejecución Paso a Paso	13
16. Comparar Código.....	14
17. Refactorizar	15
18. Generar Código Automáticamente	16
19. Ingeniería Inversa (Generar UML desde Código).....	16
20. Administrar Base de Datos.....	16
21. Ejecutar Script sobre la Base de Datos	16

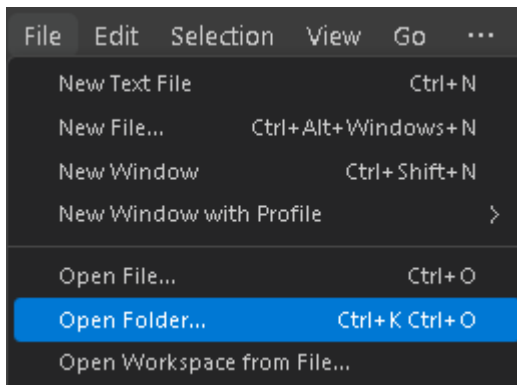
VISUAL STUDIO CODE

1. Crear un Proyecto

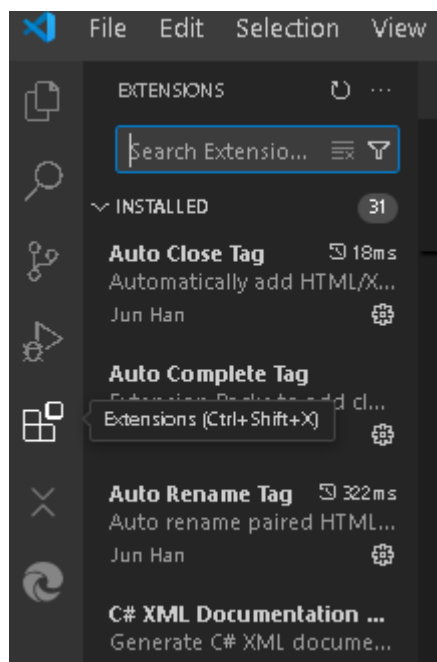
1. Abre **Visual Studio Code**.



2. Ve a **File** → **Open Folder** (Archivo → Abrir Carpeta).

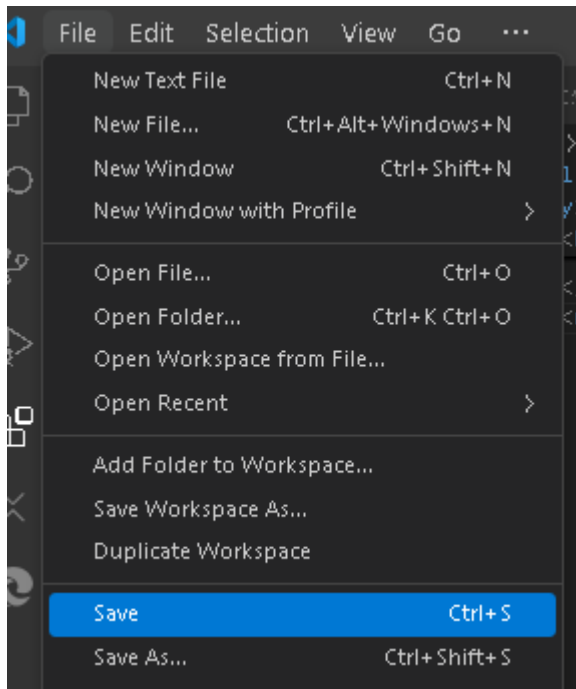


3. Selecciona o crea una carpeta para tu proyecto.
4. Si trabajas con un lenguaje específico (Java, Python, JavaScript, etc.), instala las extensiones necesarias desde la pestaña **Extensions** (Ctrl + Shift + X).



2. Guardar Proyecto

- VS Code guarda automáticamente los cambios en los archivos abiertos.
- Para forzar el guardado, presiona **Ctrl + S** o ve a **File** → **Save**.



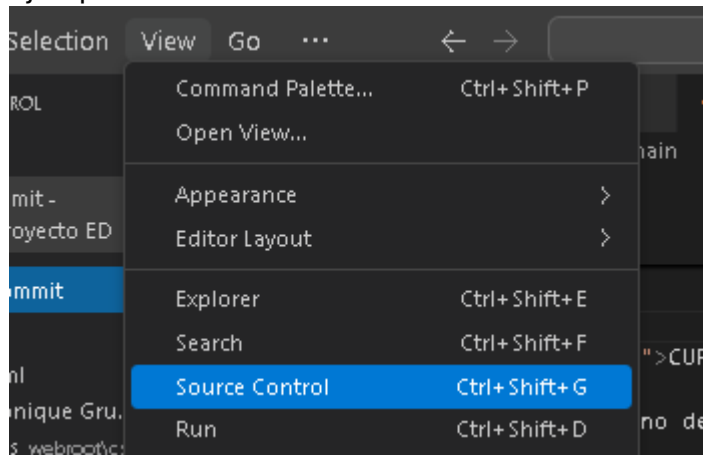
3. Versionar (Git)

1. desde la terminal de Visual Studio

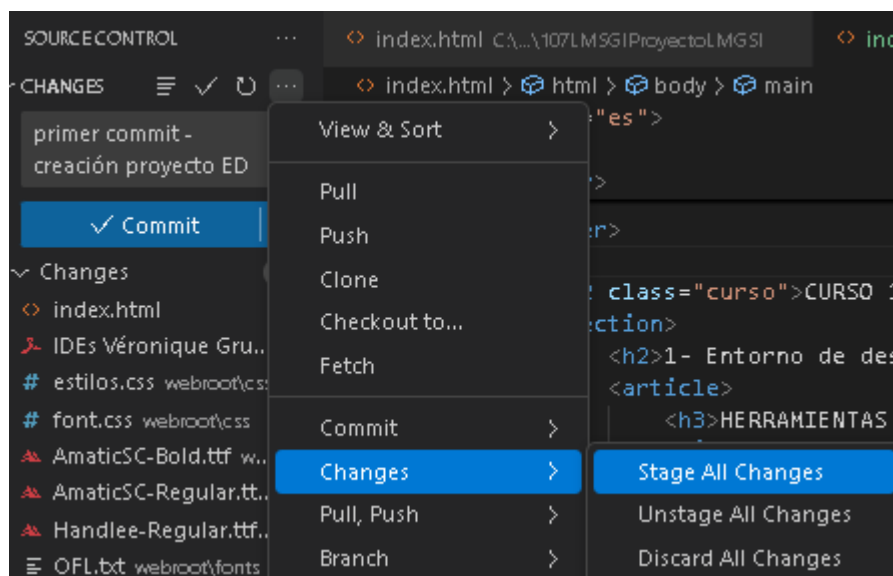
```
59 C:\hody>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Usuario\OneDrive - Educacyl\AA LENGUAJES\107EDProyectoED> git init
Initialized empty Git repository in C:/Users/Usuario/OneDrive - Educacyl/AA LENGUAJES/107EDProyectoED/.git/
PS C:\Users\Usuario\OneDrive - Educacyl\AA LENGUAJES\107EDProyectoED> git remote add origin https://github.com/verogmayo/107EDProyectoED
PS C:\Users\Usuario\OneDrive - Educacyl\AA LENGUAJES\107EDProyectoED>
```

0

Abre **View** → **Source Control** (Ctrl + Shift + G).



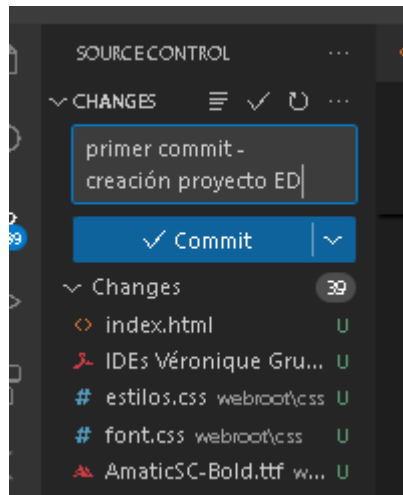
2. Si tu carpeta no está versionada, haz clic en **Initialize Repository**.
3. Preparar todos los archivos modificados para ser incluidos en el próximo commit.



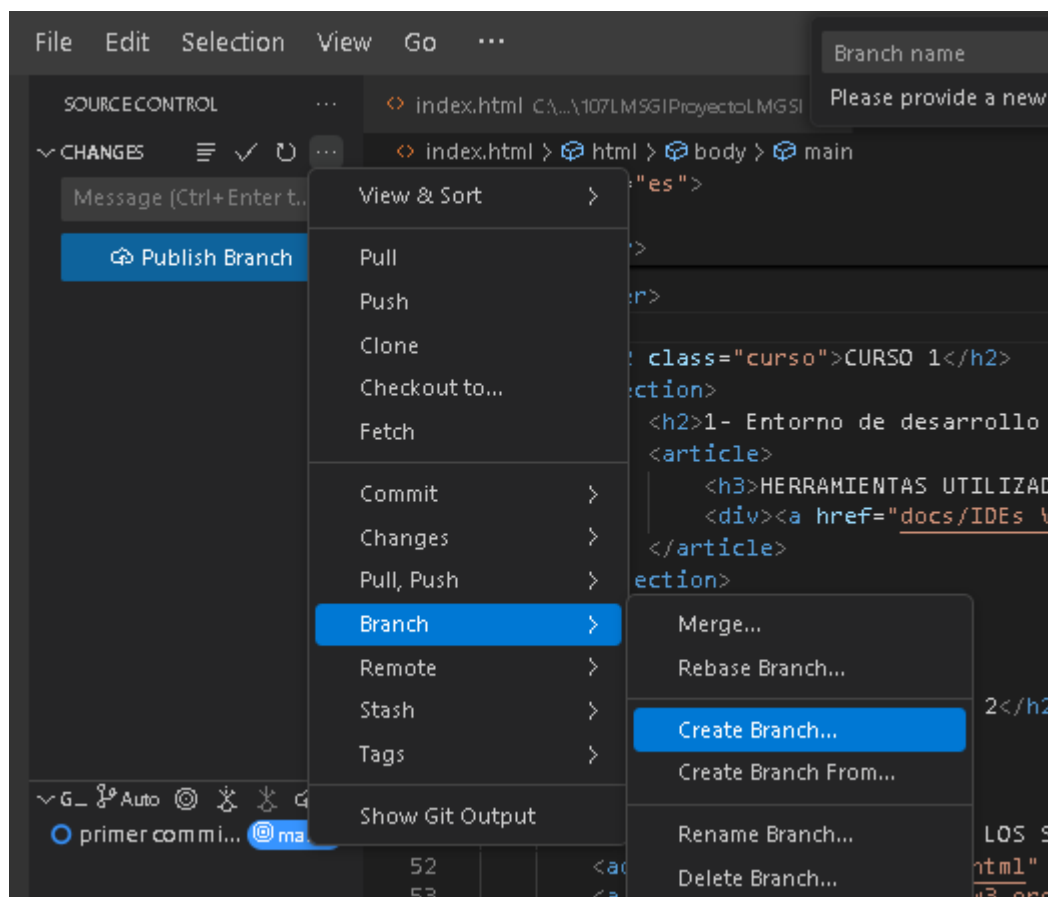
En la terminal será así

```
PS C:\Users\Usuario\OneDrive - Educacy1\AA LENGUAJES\107EDProyectoED> git add.
```

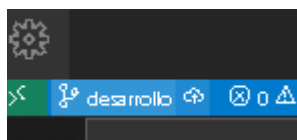
4. Escribe un mensaje en la caja de texto y presiona **Commit** (Ctrl + Enter).



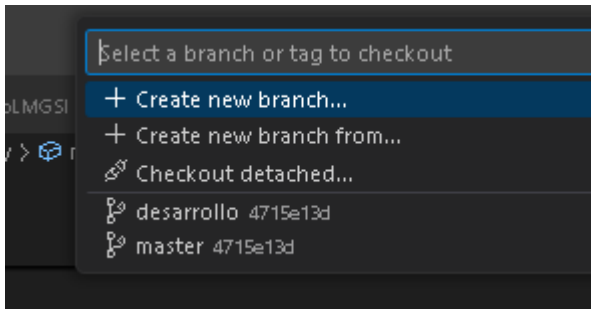
5. Crear una nueva rama .



6. Cambiar de rama. En la esquina inferior izquierda haz clic en la rama actual



7. Saldrá un menú arriba donde elegir la rama que quieras



8. Para subir cambios, haz clic en **Push** o usa git push en la terminal.

4. Recuperar desde el Repositorio

1. Abre **View** → **Source Control**.
2. Haz clic en **Clone Repository** y pega la URL del repositorio (GitHub, GitLab, etc.).

Create a new fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (*).

Owner * / Repository name *
verogmayo / 100LMSGIUnidad3
✓ 100LMSGIUnidad3 is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

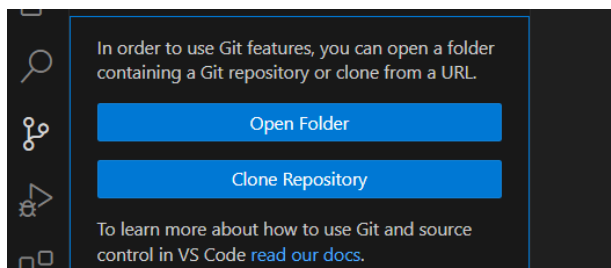
Description (optional)

Unidad 3 Manipulación de documentos web

☒ Copy the master branch only
Contribute back to arodriguezsaues/100LMSGIUnidad3 by adding your own branch. [Learn more.](#)

① You are creating a fork in your personal account.

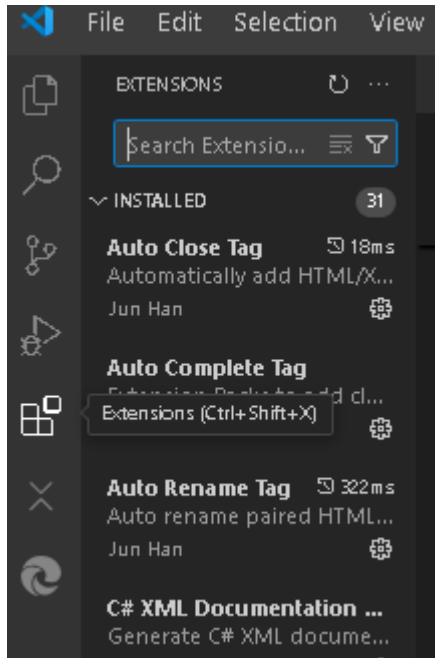
Create fork



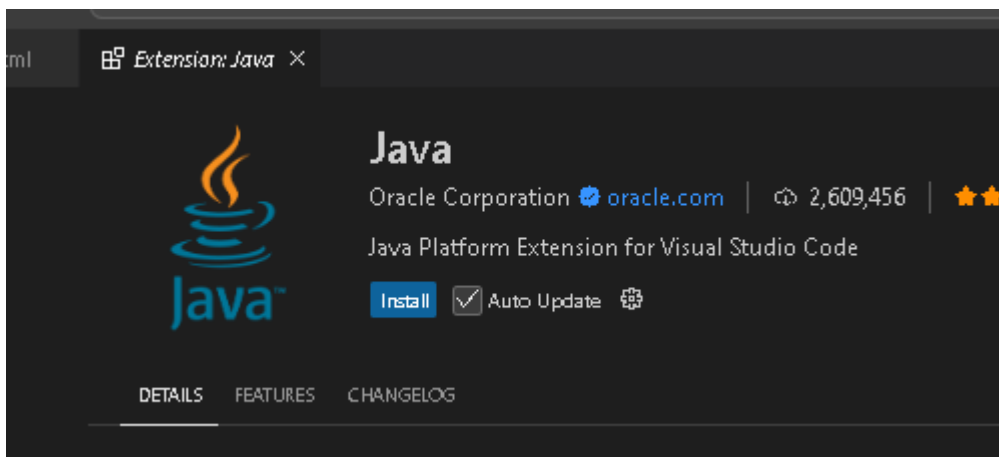
3. Selecciona la carpeta donde se clonará el proyecto.
4. Abre la carpeta en VS Code.

5. Instalar Plugins (Extensiones)

1. Ve a **View** → **Extensions** (Ctrl + Shift + X).

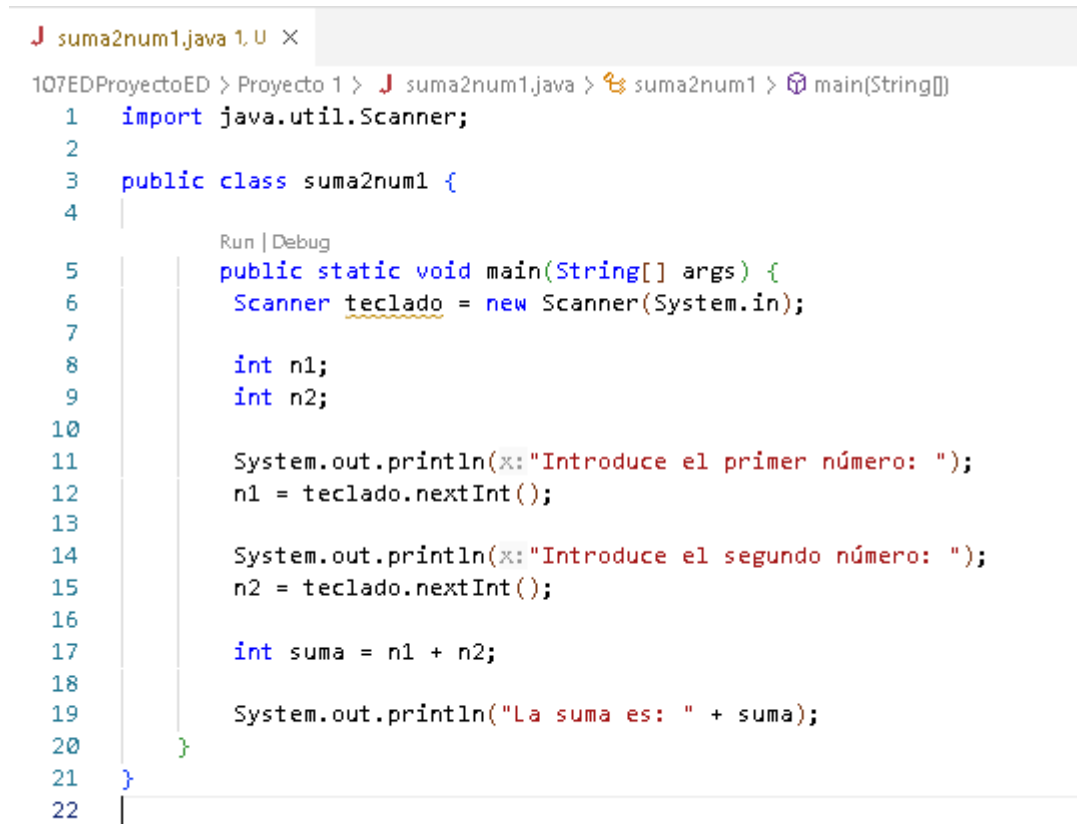


2. Busca el plugin necesario (ej. "Java Extension Pack" para Java).
3. Haz clic en **Install**.



6. Desarrollar (Escribir Código y Gestionar Archivos)

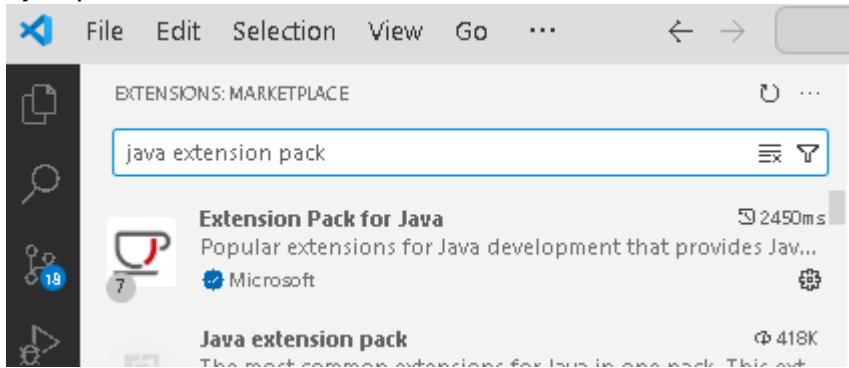
1. Usa el explorador de archivos para crear y abrir archivos.
2. Escribe tu código con soporte de autocompletado y resaltado de sintaxis.



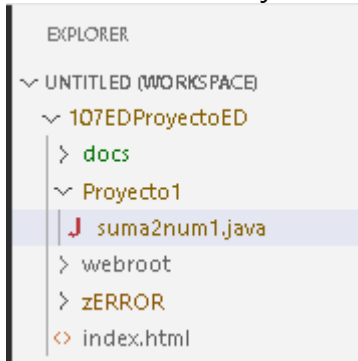
```
107EDProyectoED > Proyecto 1 > suma2num1.java > suma2num1 > main(String[])
1  import java.util.Scanner;
2
3  public class suma2num1 {
4
5      Run | Debug
6      public static void main(String[] args) {
7          Scanner teclado = new Scanner(System.in);
8
9          int n1;
10         int n2;
11
12         System.out.println(x: "Introduce el primer número: ");
13         n1 = teclado.nextInt();
14
15         System.out.println(x: "Introduce el segundo número: ");
16         n2 = teclado.nextInt();
17
18         int suma = n1 + n2;
19
20         System.out.println("La suma es: " + suma);
21     }
22 }
```

7. Compilar

- **Para Java:** Instala **Java Extension Pack**, luego compila con `javac MiClase.java`.



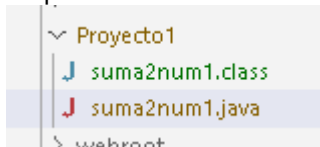
Este es el archivo de java sin compilar



Para compilar hay se escribe en la terminal el comando javac

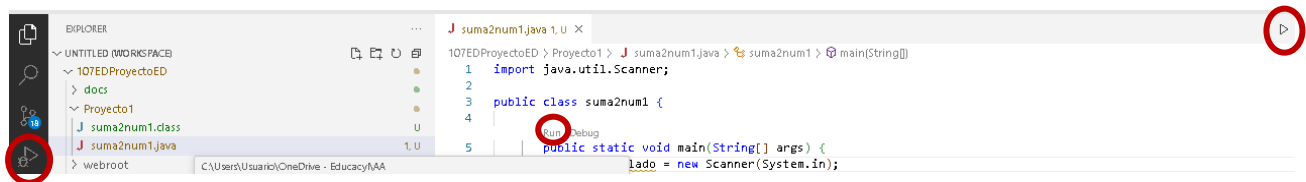
```
PS C:\Users\Usuario\OneDrive - Educacyl\AA LENGUAJES\107EDProyectoED\Proyecto1> javac suma2num1.java
PS C:\Users\Usuario\OneDrive - Educacyl\AA LENGUAJES\107EDProyectoED\Proyecto1>
```

Y aparecerá el archivo .class en la carpeta del proyecto

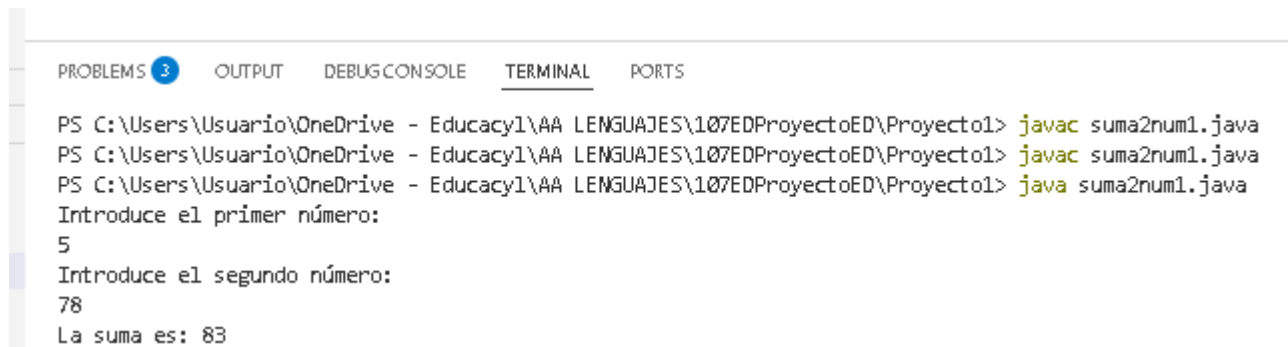


8. Ejecutar Código

1. Abre el archivo de código.
2. Se puede hacer clic en algunas de estas opciones



3. También puedes usar la terminal integrada para ejecutar comandos específicos según el lenguaje.



The screenshot shows the Visual Studio interface with the 'TERMINAL' tab selected. The terminal output shows the compilation and execution of a Java program named 'suma2num1.java'. The commands executed are 'javac suma2num1.java' and 'java suma2num1.java'. The program prompts for two numbers: 'Introduce el primer número:' and 'Introduce el segundo número:'. The user inputs '5' and '78' respectively. The final output is 'La suma es: 83'.

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Usuario\OneDrive - Educacy1\AA LENGUAJES\107EDProyectoED\Proyecto1> javac suma2num1.java
PS C:\Users\Usuario\OneDrive - Educacy1\AA LENGUAJES\107EDProyectoED\Proyecto1> javac suma2num1.java
PS C:\Users\Usuario\OneDrive - Educacy1\AA LENGUAJES\107EDProyectoED\Proyecto1> java suma2num1.java
Introduce el primer número:
5
Introduce el segundo número:
78
La suma es: 83
```

9. Probar (Manual y Automática)

- **Manual:** Ejecuta el programa y verifica la salida.

10. Documentar

- Usa comentarios en el código:

11. Generar Documentación (Javadoc, Doxygen, etc.)

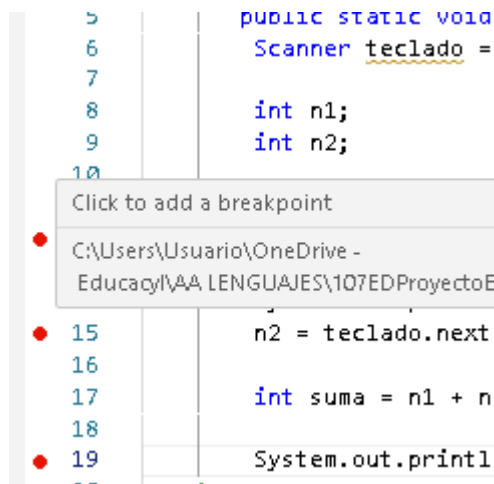
- **Java:** Usa `javadoc -d doc miarchivo.java` para generar documentación en HTML.
- **Python:** Usa `pydoc` o **Sphinx**.
- **C/C++:** Usa **Doxygen** (doxygen Doxyfile).

12. Modelar (Diagramas UML y más)

1. Instala la extensión **PlantUML**.
2. Crea un archivo .puml y dibuja el diagrama, por ejemplo:
3. Abre la vista previa con Alt + D.

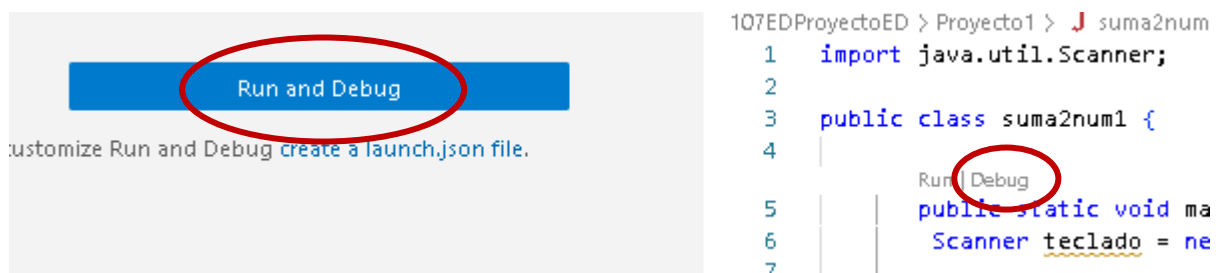
13. Depurar (Debugging)

1. Usar **breakpoints** (clic en el margen izquierdo del código), en los puntos que consideres apropiados.

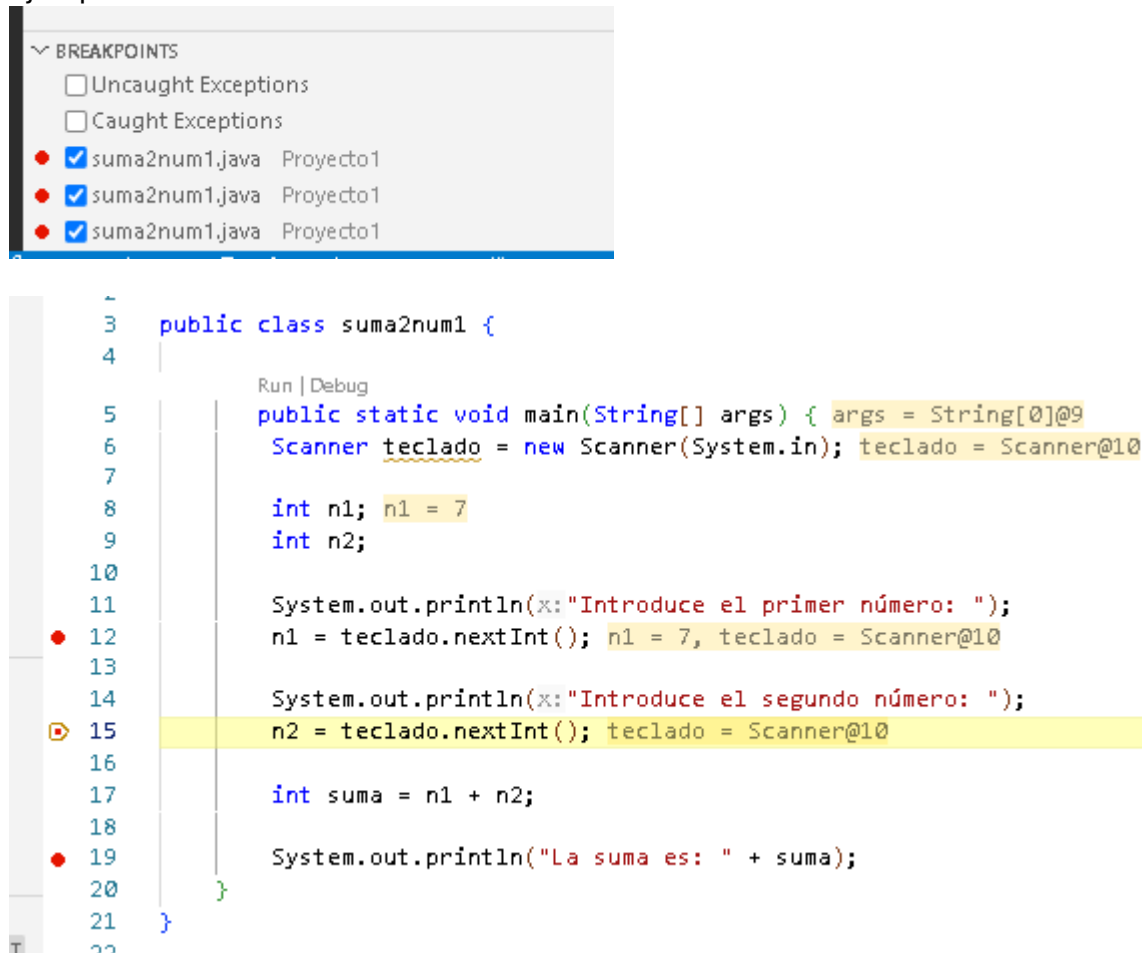


Se quitan volviendo a hacer clic en los puntos

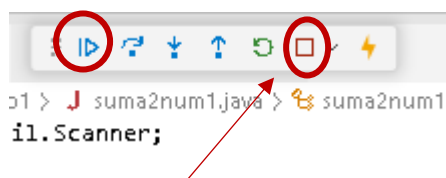
2. Hacer clic en algunos de estos puntos para iniciar el debug.



3. Se puede seleccionar en esta parte todos o parte de los puntos de break abajo a la izquierda.



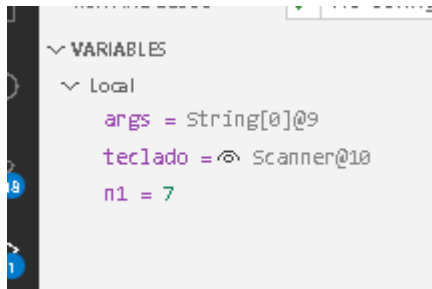
Para seguir ejecutando el código después del break, haz clic en continuar



Aquí para pararlo.

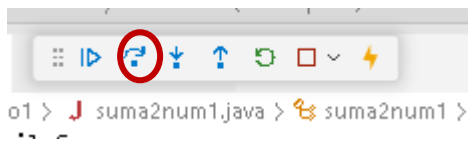
14. Inspección de Variables

1. Activa el modo de depuración (F5).
2. Usa la pestaña **Variables** en la vista de depuración.

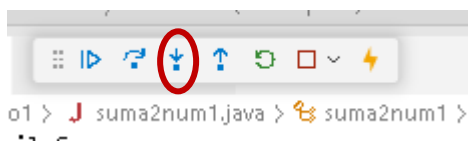


15. Ejecución Paso a Paso

- **F10:** Ejecutar línea sin entrar en funciones.



- **F11:** Entrar dentro de una función.

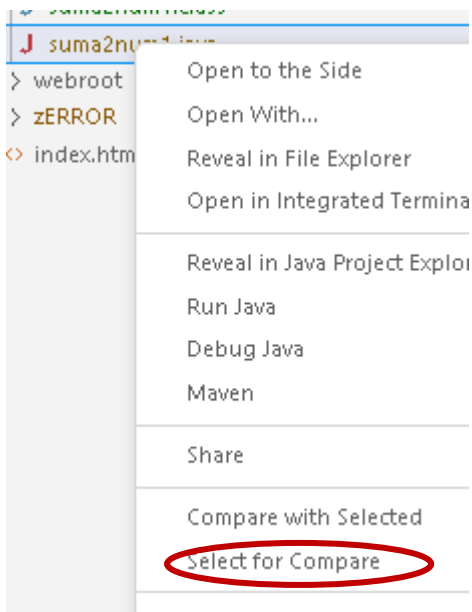


- **Shift + F11:** Salir de una función.

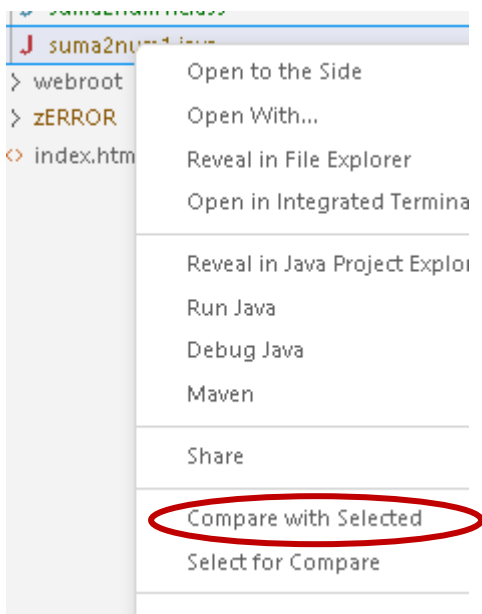


16. Comparar Código

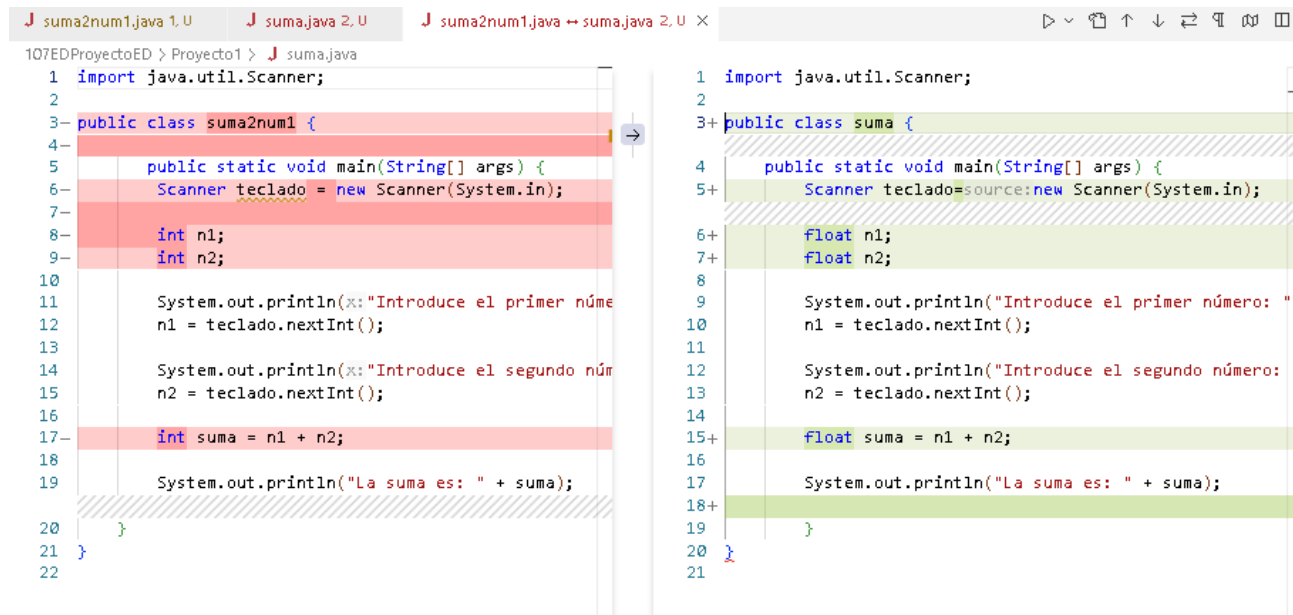
1. Haz clic derecho sobre un archivo en el **Explorador**.
2. Selecciona **Select for Compare**.



3. Luego, haz clic derecho sobre otro archivo y elige **Compare with Selected**.

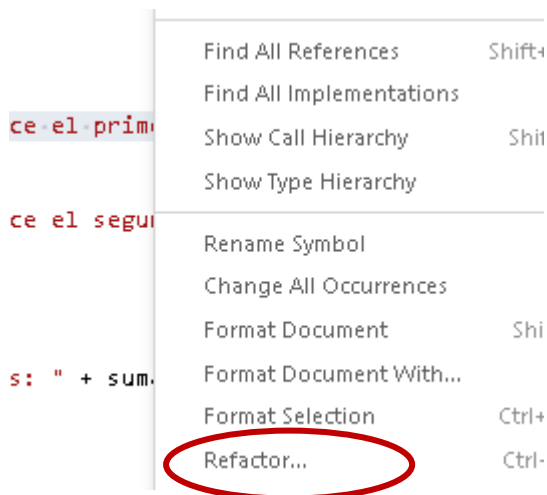


4. Puedes comparar los dos códigos



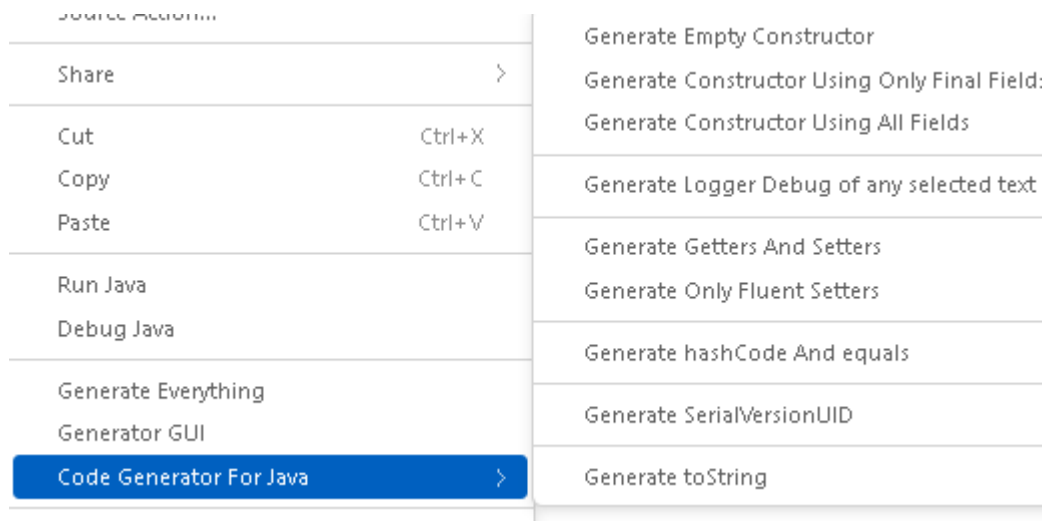
17. Refactorizar

1. Selecciona el código a refactorizar.
2. Usa el atajo **Ctrl + Shift + R** o haz clic derecho → **Refactor**.



18. Generar Código Automáticamente

- **Java:** Botón derecho en el Código para generar constructores, getters y setters....



19. Ingeniería Inversa (Generar UML desde Código)

1. Instala la extensión **Java Extension Pack** o **PlantUML**.
2. Usa la opción de **Diagrama UML** para visualizar clases automáticamente.

20. Administrar Base de Datos

1. Instala la extensión **SQL Server** o **MySQL** según la base de datos.
2. Usa el panel **View** → **SQL Explorer** para conectar bases de datos.

21. Ejecutar Script sobre la Base de Datos

1. Abre un archivo .sql.
2. Conéctalo a una base de datos desde el panel de extensiones.
3. Ejecuta el script con **Run Query** (Ctrl + Shift + E).