

Práctica 1 – Entornos de Desarrollo Integrados


En esta práctica aprenderemos a instalar y utilizar las funciones básicas de edición, compilación y ejecución de clases Java de tres de los IDEs más utilizados para el desarrollo de aplicaciones Java (además del Eclipse que utilizamos habitualmente) como son NetBeans, IntelliJ IDEA y Visual Studio Code.

Práctica

1. Realiza y documenta con capturas de pantalla y explicaciones la instalación del IDE **NetBeans** de Apache. Realiza y documenta con capturas de pantalla y explicaciones la creación un programa básico en Java mediante las herramientas de NetBeans que contenga al menos dos clases (una de ellas ejecutable, la otra con un método que invocarás desde el *main* de la clase ejecutable) que pertenezcan a dos paquetes diferentes. Realiza de manera documentada también la compilación y ejecución del programa mediante el IDE NetBeans, detallando qué ficheros se crean en nuestro disco duro en un proyecto de Java con este IDE y para qué sirven (por ejemplo, en Eclipse se crea la carpeta *src* para el código fuente, la carpeta *bin* para las clases compiladas y la carpeta *.settings* y ficheros *.classpath* y *.project* para configuraciones del Eclipse para ese proyecto)
2. Realiza todo lo indicado en el apartado 1 de la presente práctica, pero con la versión gratuita del IDE **IntelliJ IDEA** de JetBrains (la versión gratuita es IntelliJ IDEA Community Edition).
3. Realiza todo lo indicado en el apartado 1 de la presente práctica, pero con el IDE **Visual Studio Code** de Microsoft.

NetBeans

1

 netbeans.apache.org/front/main/download/nb23/

En primer lugar nos dirigiremos al enlace que se muestra en la imagen, que es la página oficial de los desarrolladores de NetBeans.

2

Installers and Packages:

- [Apache-NetBeans-23-bin-windows-x64.exe \(SHA-512, PGP ASC\)](#)

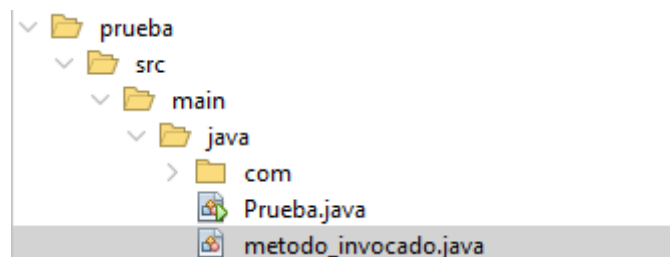
A continuación buscaremos en la página de descargas en enlace acorde a nuestro sistema operativo y tipo de archivo que queramos descargar ya sea un .exe o un .rar

3



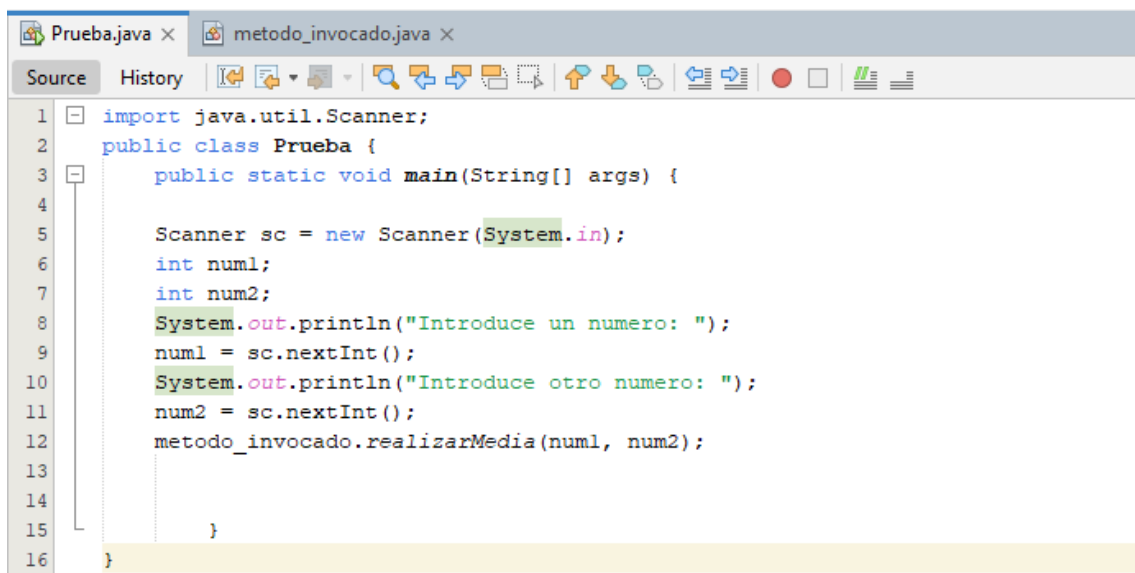
Después de la instalación y configuración de la aplicación a nuestro gusto, en el panel que nos encontraremos será como las imágenes posteriores. Los iconos que estás viendo sirven para crear clases y proyectos en los que trabajaremos nuestro código.

4



Cuando hayamos creado las 2 clases que necesitamos, veremos una difusión de las carpetas donde se guardan todos los archivos del programa (la gran mayoría que son configuraciones no se muestran en el lugar de la imagen porque son elementos no ejecutables y que no nos sirve que aparezcan donde de el entorno de desarrollo). En este caso yo he creado un paquete con el nombre “prueba” donde dentro tendremos la carpeta “src”, que es donde se guardaran todos los .java que creemos en el mismo paquete. La carpeta “java” se crea automáticamente al saber que lo que hemos creado con dos clases java, ya que podrían ser de otro lenguaje de programación y así poder diferenciarlos. Y por último, podemos observar las dos clases creadas, una llamada “prueba” que será el Main de nuestro programa, y “metodo_invocado” donde ira el método al que tendremos que llamar desde el Main para poder ejecutarlo.

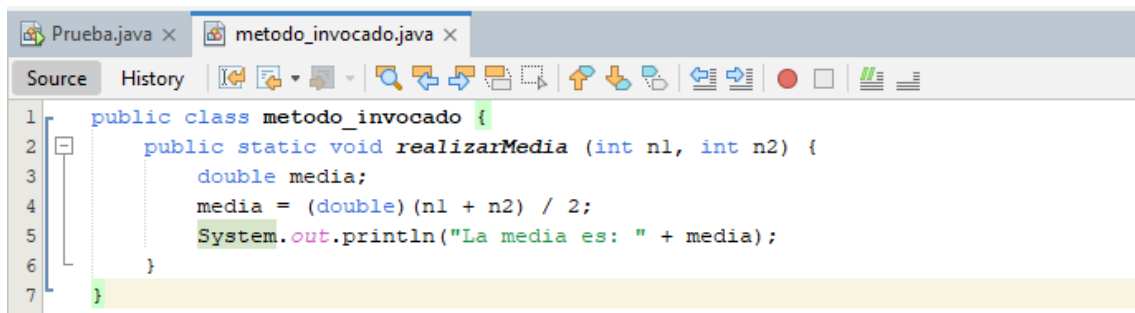
5



```
1 import java.util.Scanner;
2 public class Prueba {
3     public static void main(String[] args) {
4
5         Scanner sc = new Scanner(System.in);
6         int num1;
7         int num2;
8         System.out.println("Introduce un numero: ");
9         num1 = sc.nextInt();
10        System.out.println("Introduce otro numero: ");
11        num2 = sc.nextInt();
12        metodo_invocado.realizarMedia(num1, num2);
13
14    }
15 }
16 }
```

Aquí muestro el código del programa que lleva el Main, es decir el programa principal. Al final del código se puede apreciar que he llamado a el método que está en la otra clase anteriormente mencionada. Con el comando, poniendo el nombre de la clase donde se encuentra el método y posteriormente ponemos el nombre el método separado por un “.” (punto).

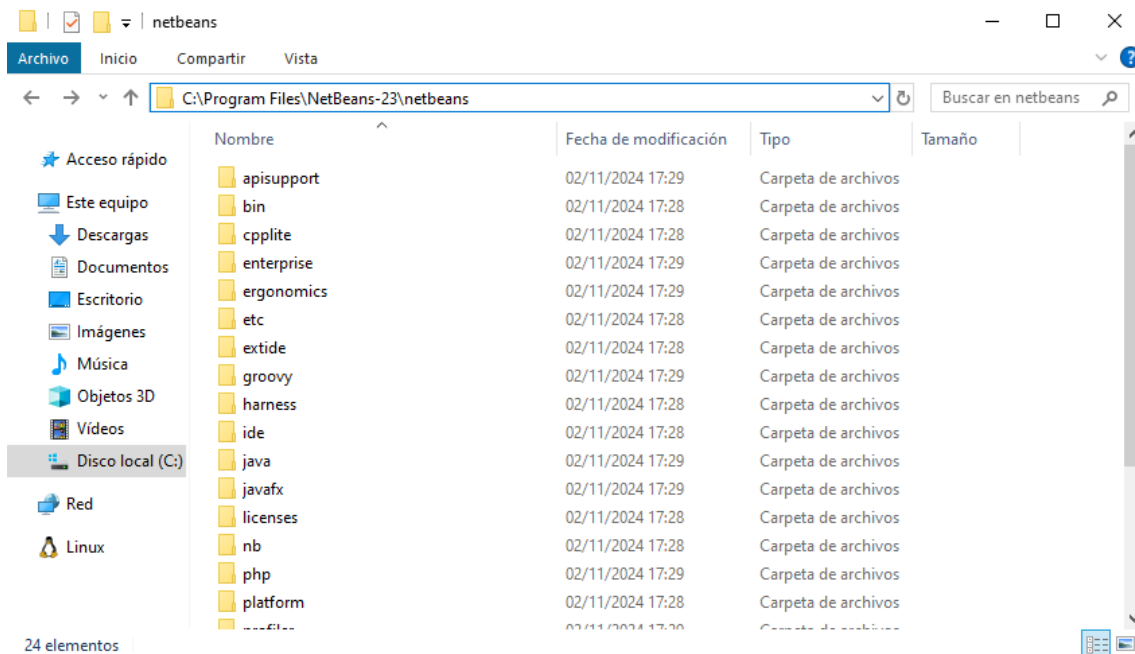
6



```
1 public class metodo_invocado {
2     public static void realizarMedia (int n1, int n2) {
3         double media;
4         media = (double) (n1 + n2) / 2;
5         System.out.println("La media es: " + media);
6     }
7 }
```

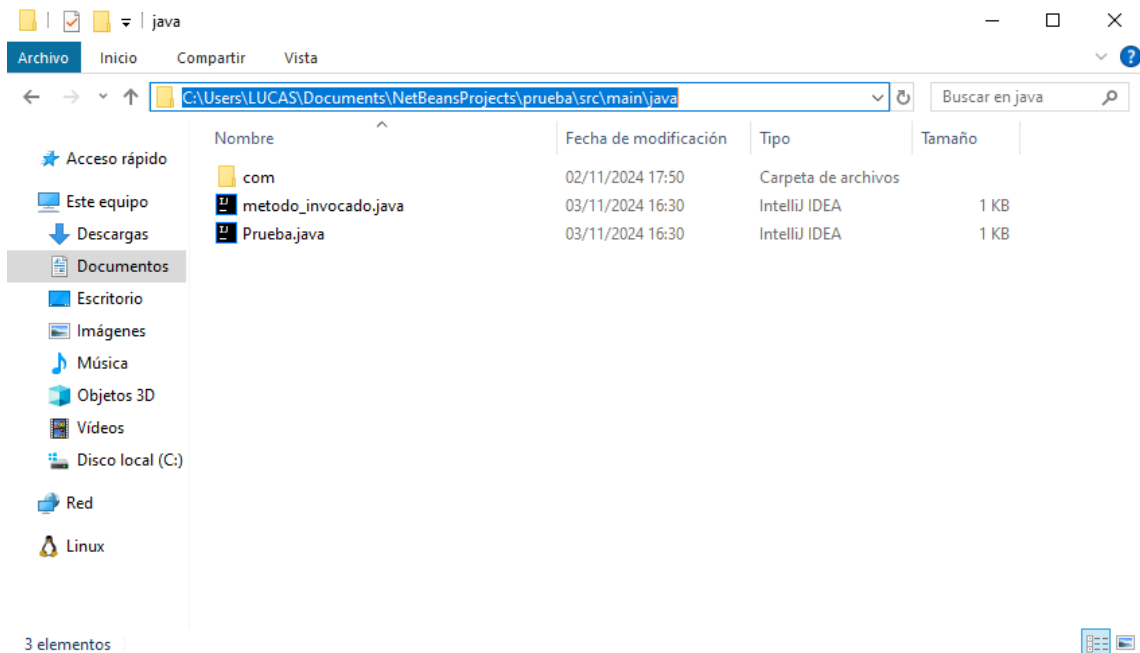
Aquí se puede apreciar la otra clase, donde se ve que únicamente tenemos un método guardado. Esto lo que realiza es que pide al usuario dos números enteros y realiza la media entre ellos.

7



Aquí se pueden ver todos los archivos que constituyen el programa con su ubicación.

8



Y aquí se ve la ubicación de los archivos .java que hemos creado.

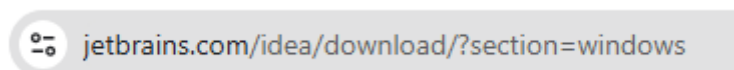
9



Por último, si queremos ejecutar el programa, solo deberemos de pulsar el botón que estás viendo y se abrirá una consola donde deberás de ver la ejecución del código.

IntelliJ IDEA

1



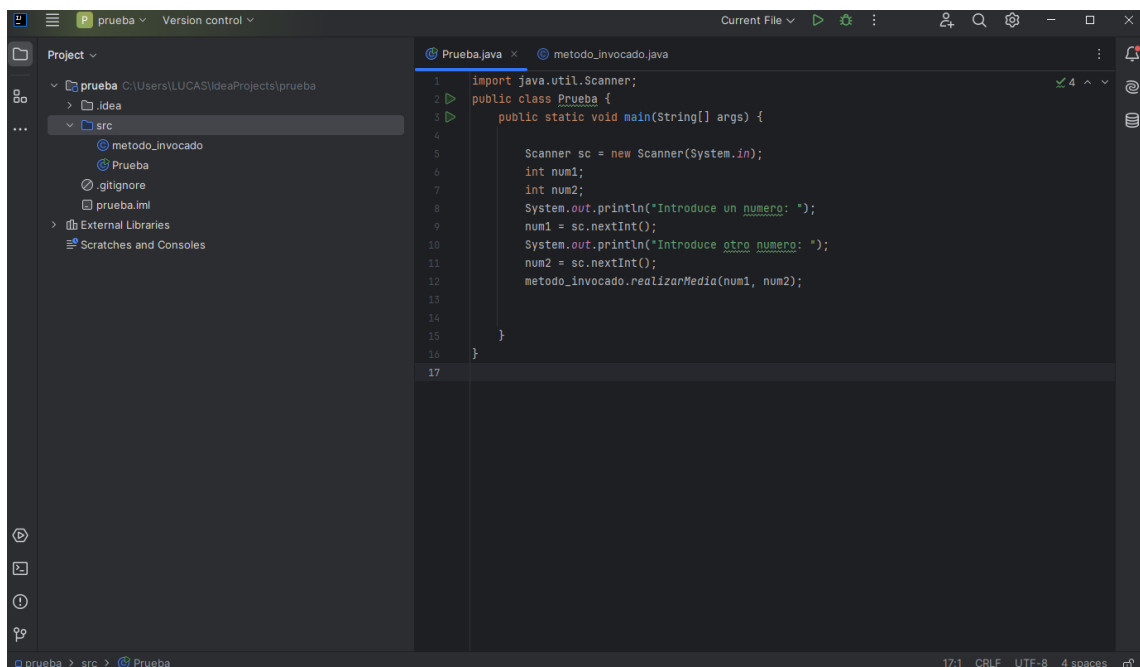
En primer lugar nos dirigiremos al enlace que se muestra en la imagen, que es la página oficial de los desarrolladores de IntelliJ IDEA.

2



A continuación buscaremos en la página de descargas en enlace acorde a nuestro sistema operativo y tipo de archivo que queramos descargar ya sea un .exe o un .rar

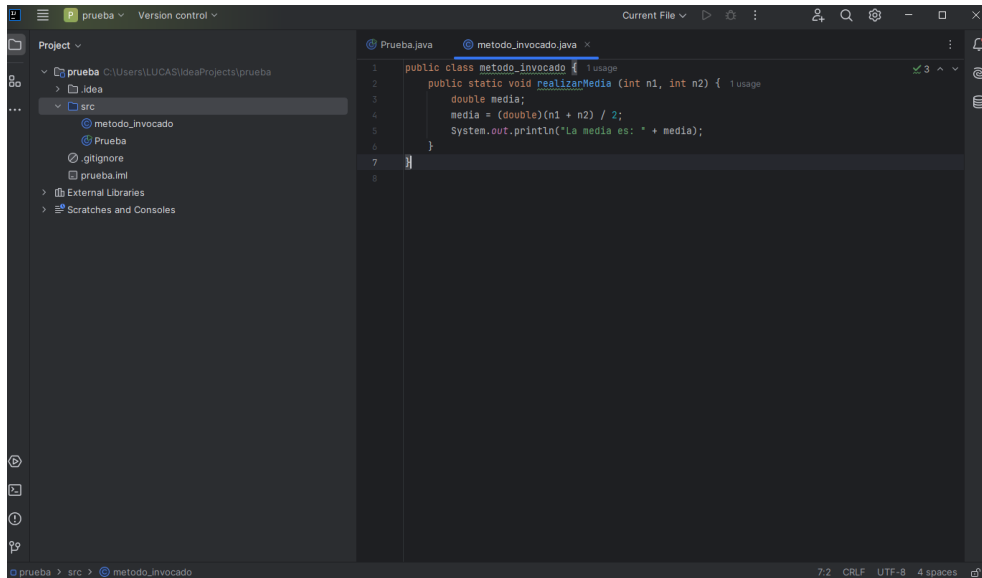
3



Cuando hayamos creado las 2 clases que necesitamos, veremos una difusión de las carpetas donde se guardan todos los archivos del programa (la gran mayoría que son configuraciones no se muestran en el lugar de la imagen porque son elementos no ejecutables y que no nos sirve que aparezcan donde del entorno de desarrollo). En este caso yo he creado un paquete con el nombre “prueba” donde dentro tendremos la carpeta “src”, que es donde se guardaran todos los .java que creemos en el mismo paquete Y por último, podemos observar las dos clases creadas, una llamada “prueba” que será el Main de nuestro programa, y “metodo_invocado” donde ira el método al que tendremos que llamar desde el Main para poder ejecutarlo.

Aquí muestro el código del programa que lleva el Main, es decir el programa principal. Al final del código se puede apreciar que he llamado a el método que está en la otra clase anteriormente mencionada. Con el comando, poniendo el nombre de la clase donde se encuentra el método y posteriormente ponemos el nombre el método separado por un “.” (punto).

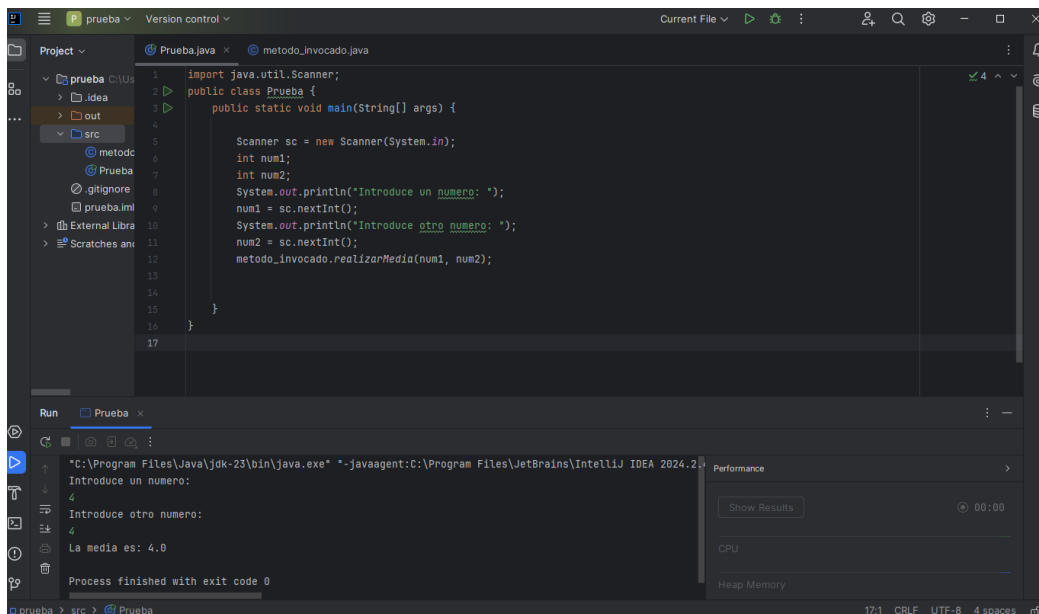
4



```
1 public class metodo_invocado {
2     public static void realizarMedia (int n1, int n2) {
3         double media;
4         media = (double)(n1 + n2) / 2;
5         System.out.println("La media es: " + media);
6     }
7 }
8
```

Aquí se puede apreciar la otra clase, donde se ve que únicamente tenemos un método guardado. Esto lo que realiza es que pide al usuario dos números enteros y realiza la media entre ellos.

5



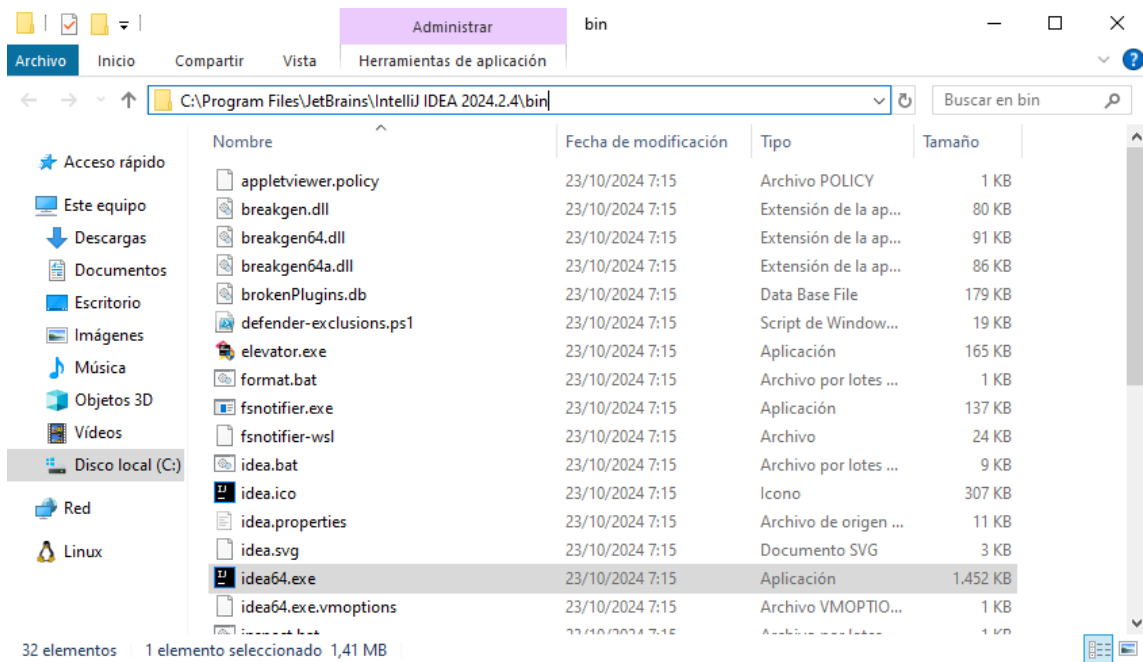
Aquí muestro la ejecución del programa, podemos ver que se muestra la consola con el código ejecutado, donde podemos ver que en ningún momento tenemos el método de que se realice la media y este escrito en la clase Main sino que ha sido llamada y hemos podido ejecutarla

6



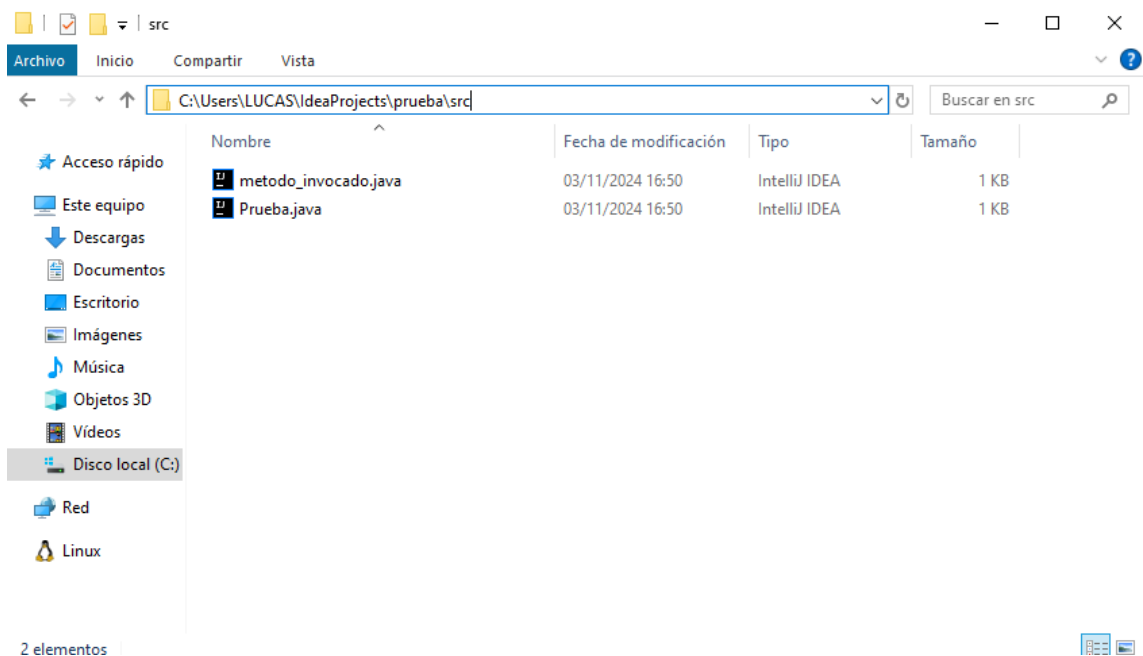
Estos son los iconos en donde se encuentra la función de ejecutar el código, pulsando el “play”/”ejecutar” para poder ejecutar el código. Se encuentra en la parte superior derecha del entorno de desarrollo y tienes esos iconos alrededor (los muestro para que sea más fácil de identificar/encontrar).

7



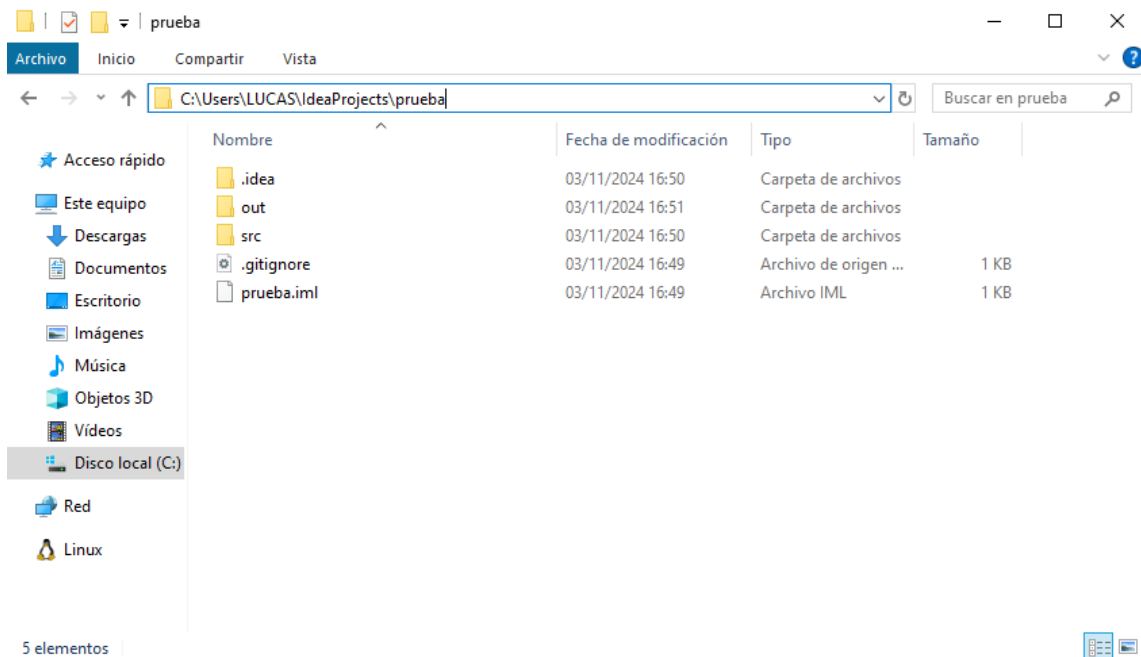
Aquí se pueden ver todos los archivos que constituyen el programa con su ubicación.

8



Y aquí se ve la ubicación de los archivos .java que hemos creado.

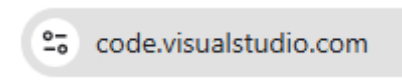
9



Aquí se la ubicación de los archivos que se crean en nuestro proyecto llamado “prueba”.

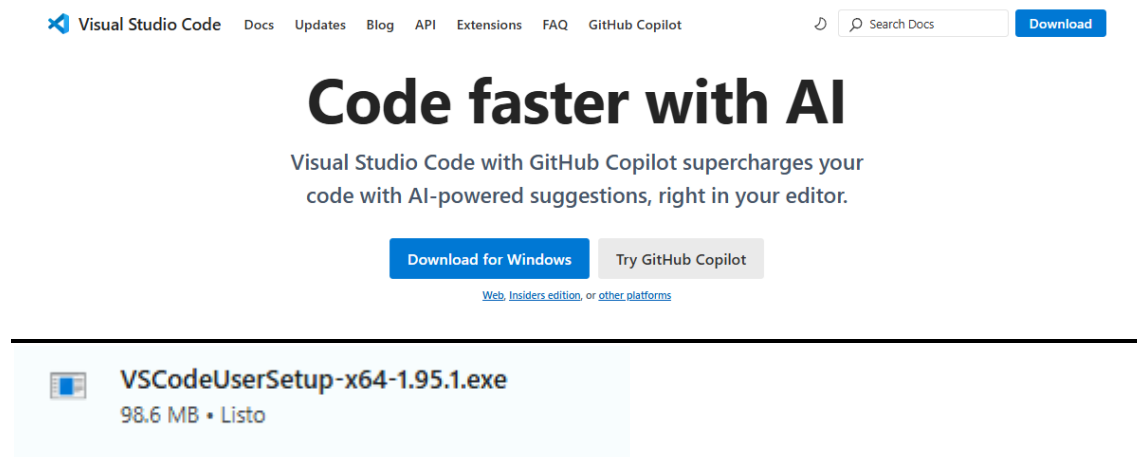
Visual Studio Code

1



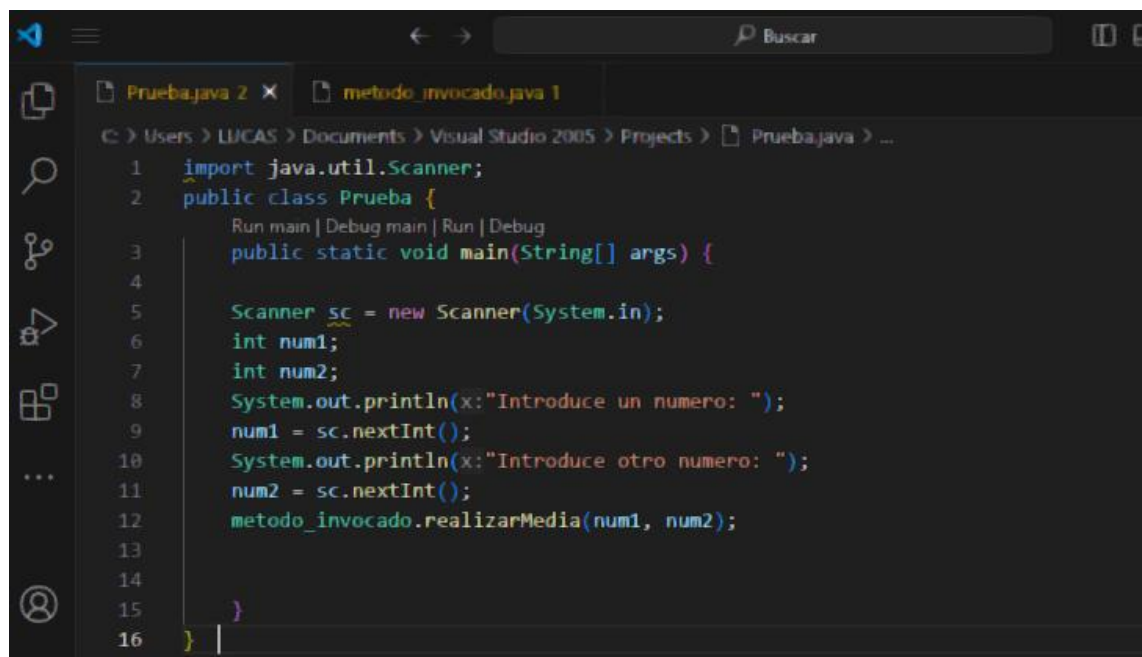
En primer lugar nos dirigiremos al enlace que se muestra en la imagen, que es la página oficial de los desarrolladores de Visual Studio Code.

2



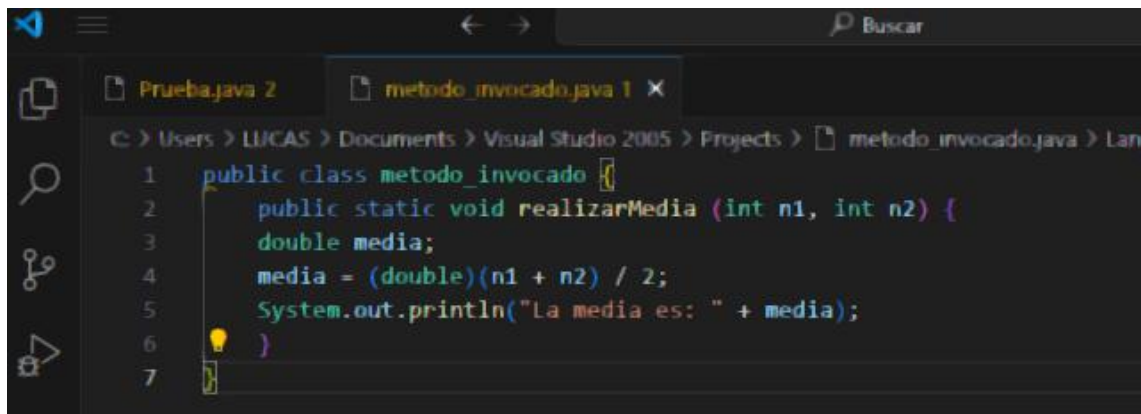
A continuación buscaremos en la página de descargas en enlace acorde a nuestro sistema operativo y tipo de archivo que queramos descargar ya sea un .exe o un .rar

3



Aquí muestro el código del programa que lleva el Main, es decir el programa principal. Al final del código se puede apreciar que he llamado a el método que está en la otra clase anteriormente mencionada. Con el comando, poniendo el nombre de la clase donde se encuentra el método y posteriormente ponemos el nombre el método separado por un "." (punto).

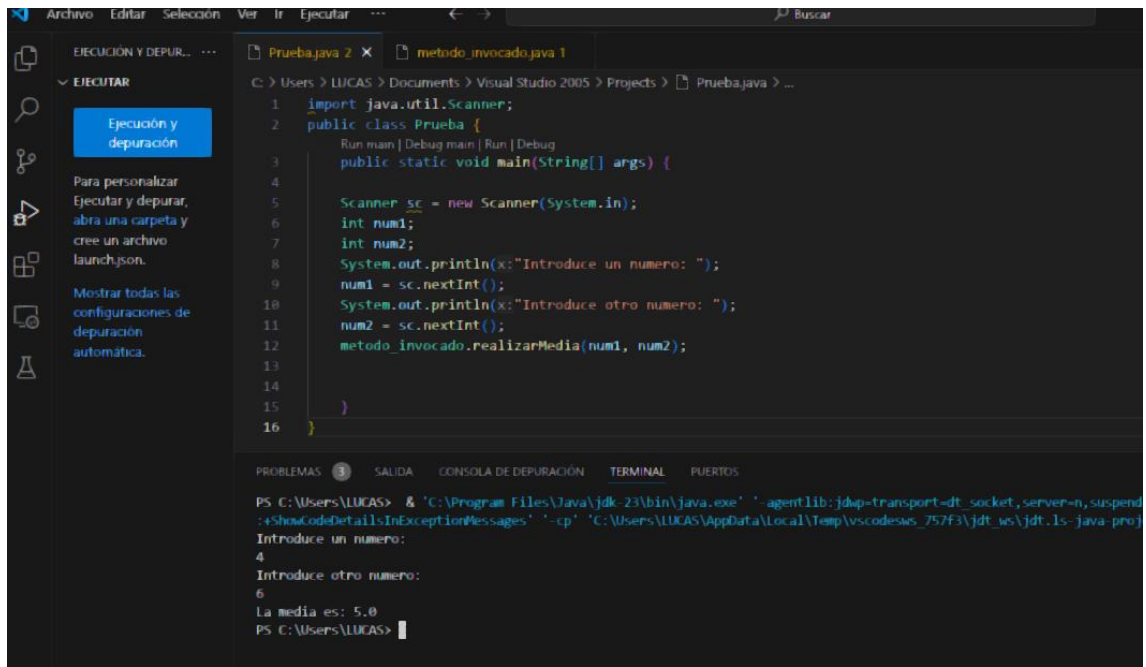
4



```
1 public class metodo_invocado {
2     public static void realizarMedia (int n1, int n2) {
3         double media;
4         media = (double)(n1 + n2) / 2;
5         System.out.println("La media es: " + media);
6     }
7 }
```

Aquí se puede apreciar la otra clase, donde se ve que únicamente tenemos un método guardado. Esto lo que realiza es que pide al usuario dos números enteros y realiza la media entre ellos.

5



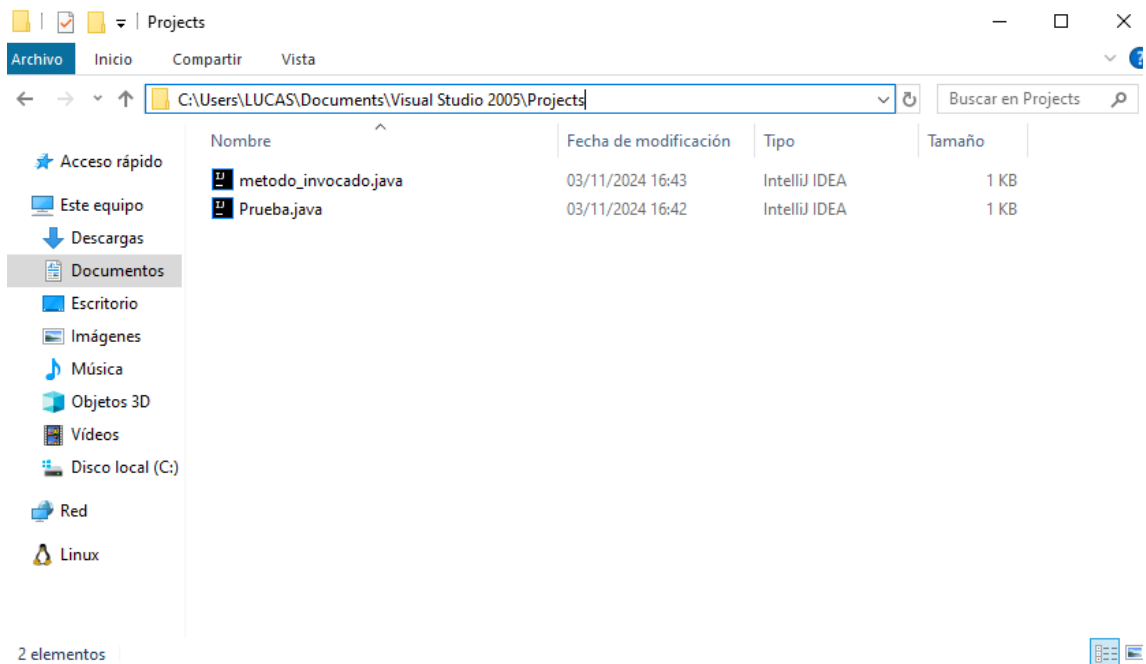
```
1 import java.util.Scanner;
2 public class Prueba {
3     public static void main(String[] args) {
4
5         Scanner sc = new Scanner(System.in);
6         int num1;
7         int num2;
8         System.out.println("Introduce un numero: ");
9         num1 = sc.nextInt();
10        System.out.println("Introduce otro numero: ");
11        num2 = sc.nextInt();
12        metodo_invocado.realizarMedia(num1, num2);
13    }
14 }
15
16 }
```

PROBLEMAS 3 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

```
PS C:\Users\LUCAS> & 'C:\Program Files\Java\jdk-23\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend
:4ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\LUCAS\AppData\Local\Temp\vscodeses_757f3\jdt_ws\jdt.ls-java-proj
Introduce un numero:
4
Introduce otro numero:
6
La media es: 5.0
PS C:\Users\LUCAS>
```

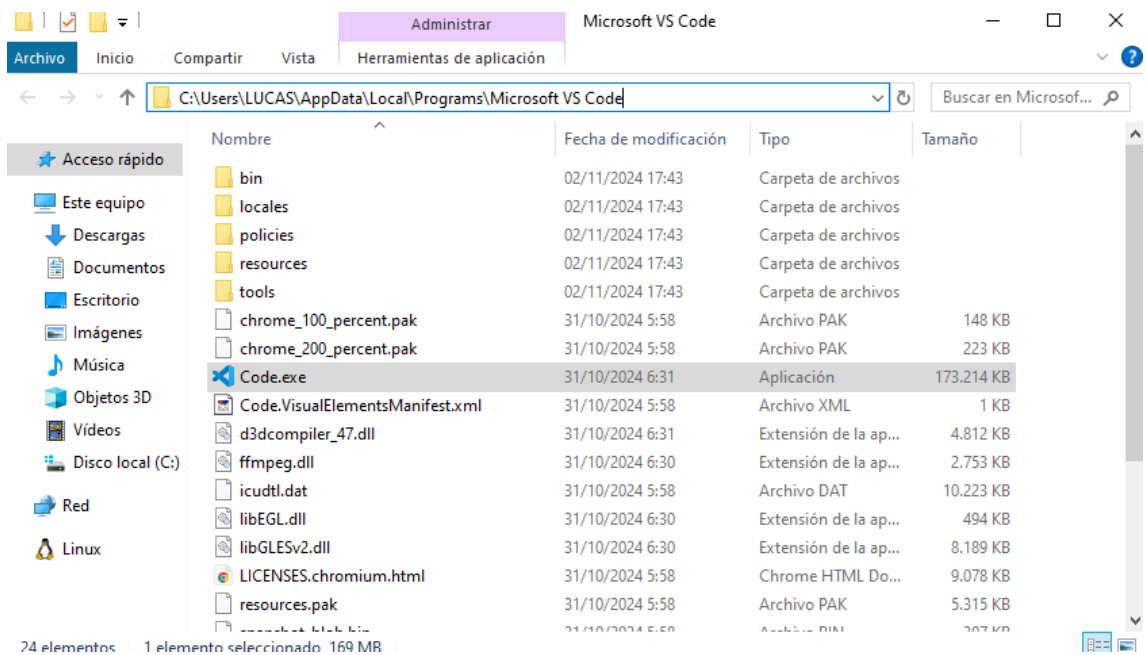
Aquí muestro la ejecución del programa, podemos ver que se muestra la consola con el código ejecutado, donde podemos ver que en ningún momento tenemos el método de que se realice la media y este escrito en la clase Main sino que ha sido llamada y hemos podido ejecutarla

6



Y aquí se ve la ubicación de los archivos .java que hemos creado.

7



Aquí se pueden ver todos los archivos que constituyen el programa con su ubicación.