

# **PRACTICE 4:**

## **VISUAL STUDIO CODE**

**Bruno David Meléndez Gómez**



## Indice

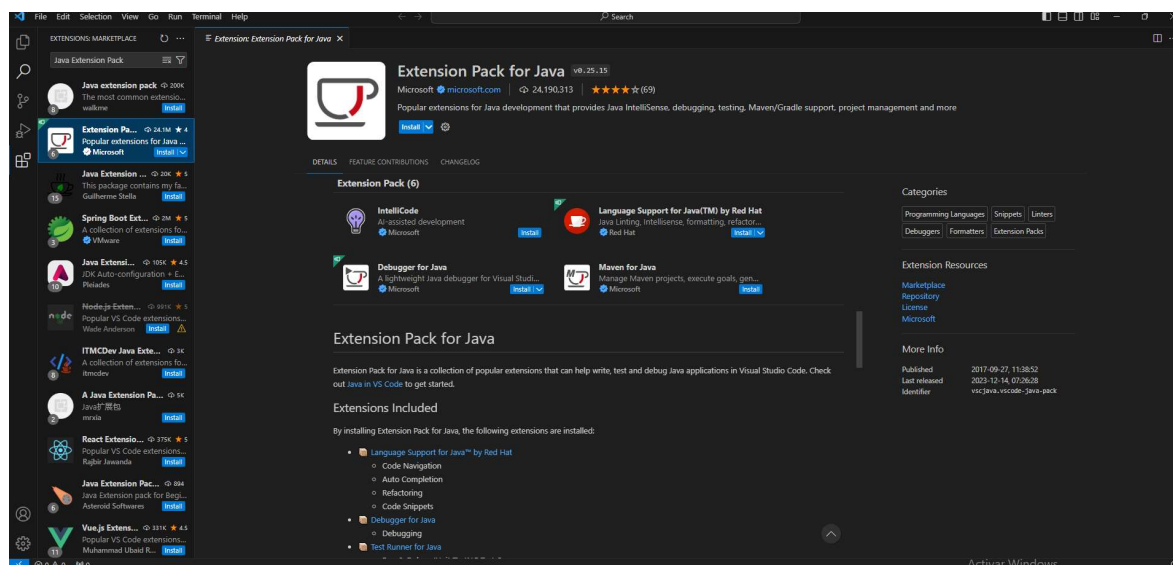
Tarea 1: Instalación de un IDE.....	3
Tarea 2: Instalación de paquetes.....	3
Tarea 3: Get Started.....	4
Tarea 4: Nuestro primer proyecto.....	5
Tarea 5: Importando un proyecto.....	6
Tarea 6: Trapecio.....	9
Biblografia.....	10

## Tarea 1: Instalación de un IDE

Nos descargamos el programa desde su pagina web oficial y los instalamos siguiendo los pasos comunes para cualquier tipo de instalacion.

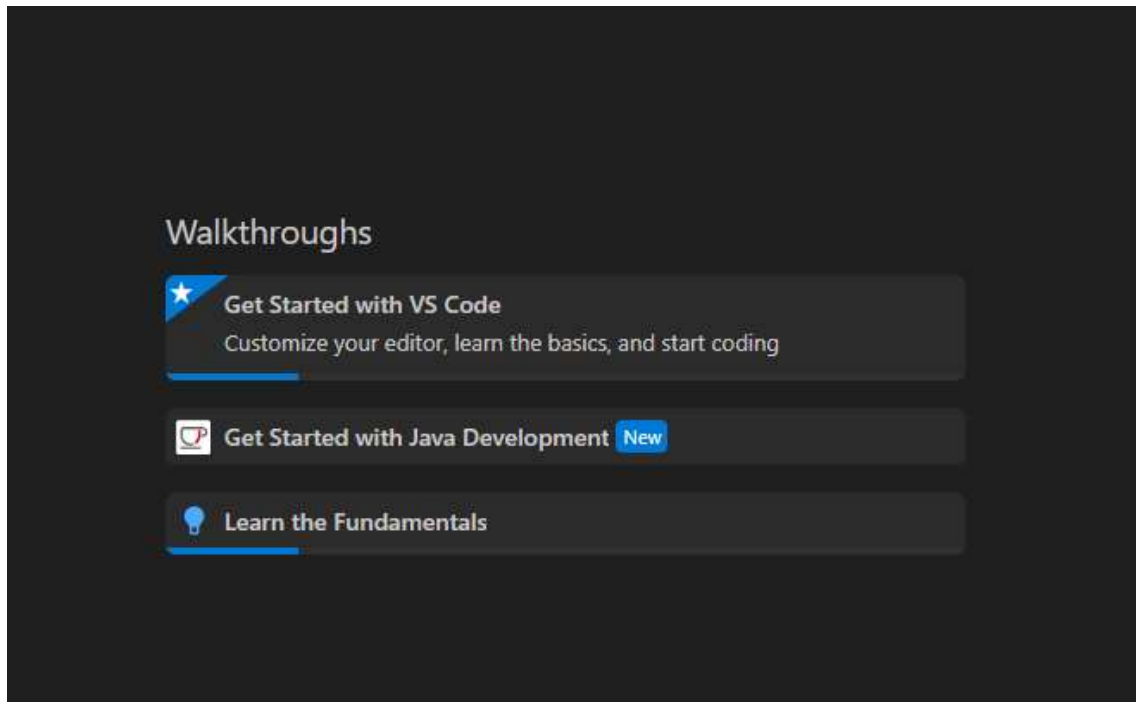


## Tarea 2: Instalación de paquetes

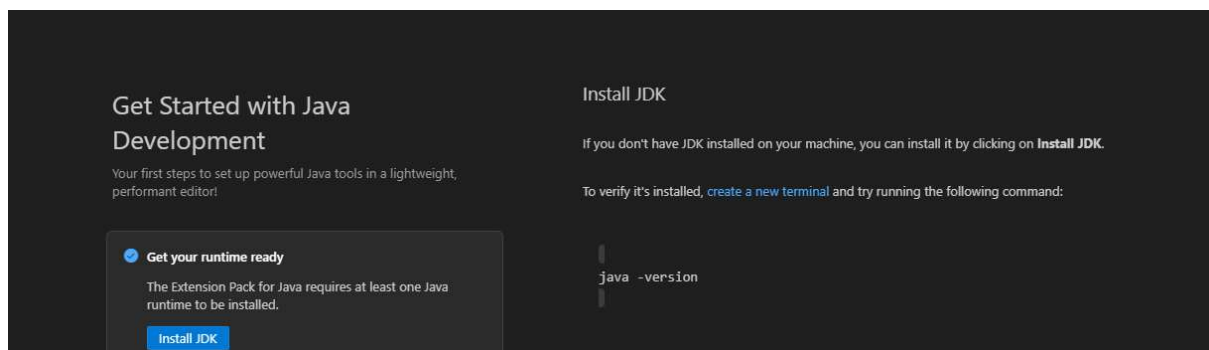


Ahora debemos de dirigirnos a la pestañas de extensiones y buscar Java Extension Pack para que asi nuestro IDE sea 100% compatible con JAVA, hay que tener cuidado y elegir el que sea de la compañía Microsoft , le damos a *Install* una vez lo hayamos localizado

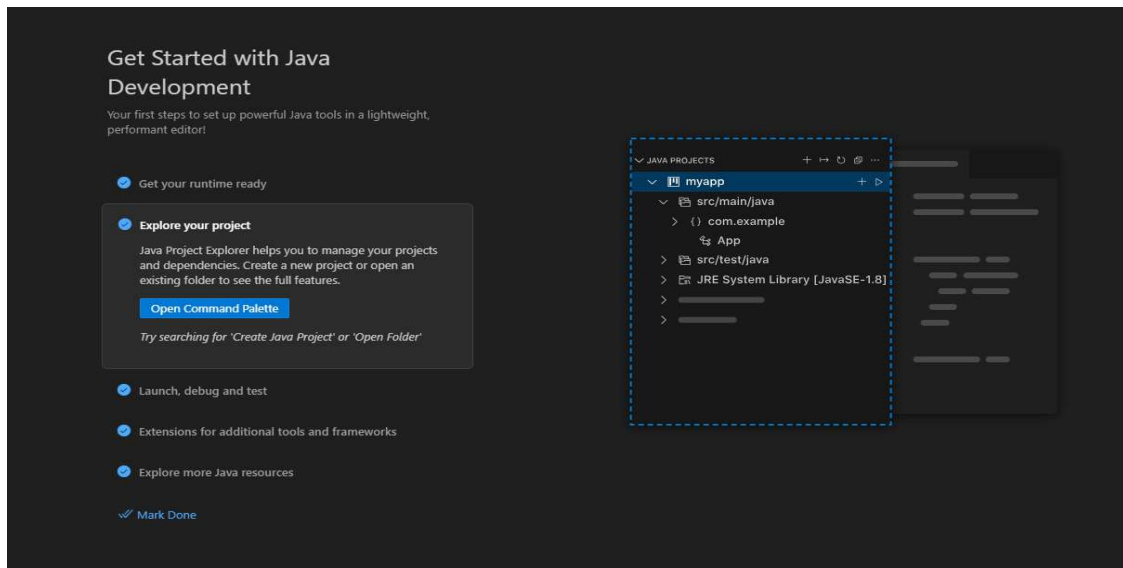
## Tarea 3: Get Started



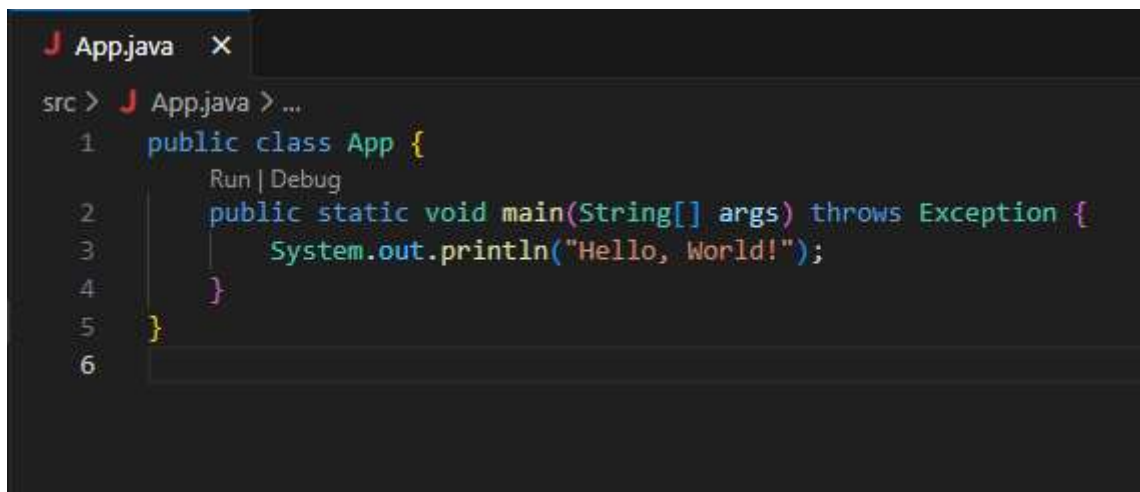
Una vez finalizada la instalacion reiniciamos nuestro VS Code y en la pestaña de bienvenida nos apareceran estas tres opciones le damos a la segunda *Get Started with Java Development* y el propio VS Code nos guiara de una forma sencilla para configurarlo, si ya tenemos el JDK instalado pasamos directamente al el paso 2



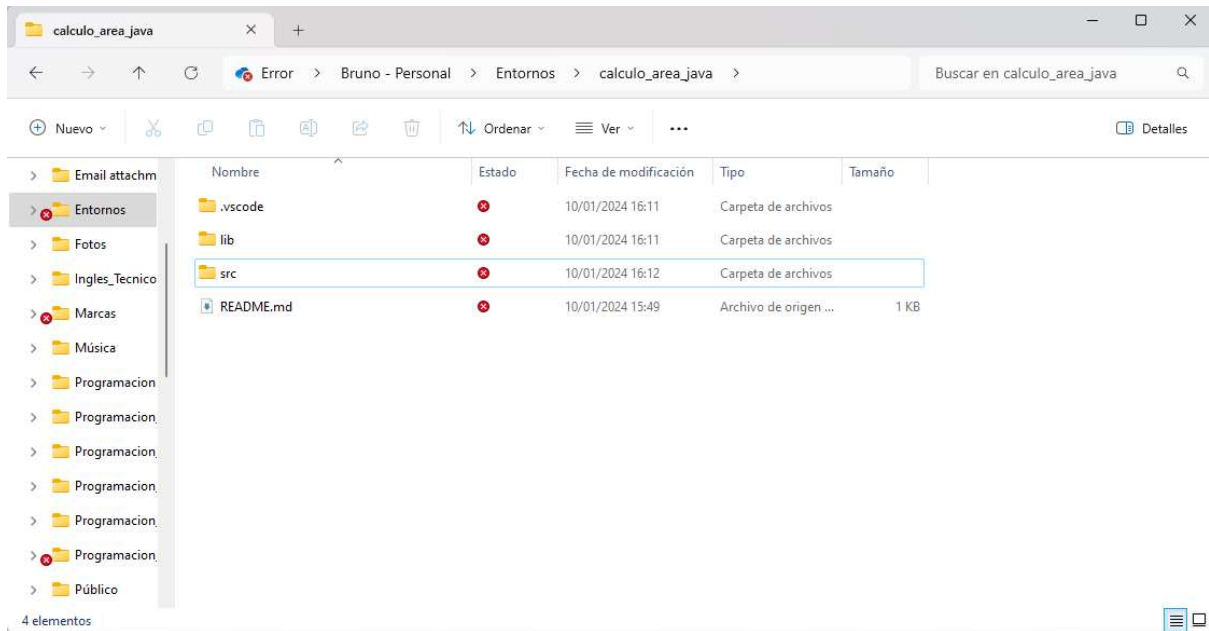
## Tarea 4: Nuestro primer proyecto



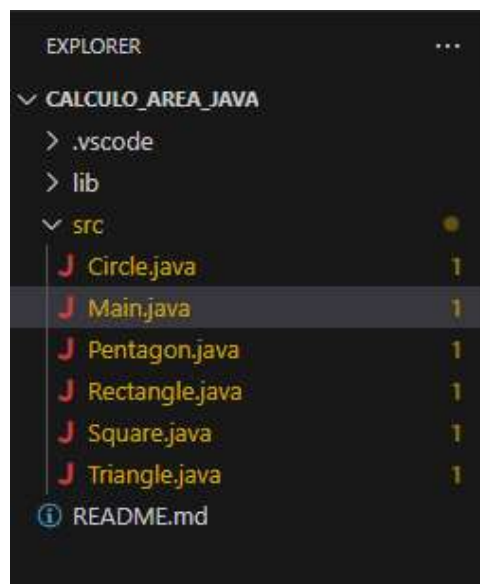
Pulsamos sobre *Open Command Palette* y luego en el desplegable pulsamos sobre *Java: Create Java Project* luego debemos de escoger la ruta donde queremos guardar nuestro proyecto y una vez aceptemos ya nos abrirá una nueva ventana que contendrá un archivo llamado *App.java* que será un programa con un método *main* y la salida por pantalla *Hello World*



## Tarea 5: Importando un proyecto



Para importar el proyecto lo descargamos y hacemos un nuevo proyecto de JAVA y en la carpeta `src` pegamos los archivos que nos hemos descargado previamente para que así podamos verlo en nuestro Visual Studio Code y dándole doble click sobre los archivos podemos acceder a el código fuente del programa y poder editarlo a nuestro antojo



```

src > J Circle.java  J Main.java x  J Pentagon.java  J Rectangle.java  J Square.java  J Triangle.java
1 // Main.java
2
3 import java.util.Scanner;
4
5 public class Main {
6     public static void main(String[] args) {
7         try {
8             Scanner scanner = new Scanner(System.in);
9             int option;
10            do {
11                System.out.println("Calculadora de Área - Seleccione una opción:");
12                System.out.println("1. Círculo");
13                System.out.println("2. Cuadrado");
14                System.out.println("3. Triángulo");
15                System.out.println("4. Rectángulo");
16                System.out.println("5. Pentágono");
17                System.out.println("0. Salir");
18                System.out.print("Opción: ");
19                option = scanner.nextInt();
20                switch (option) {
21                    case 1:
22                        Circle.calculateArea();
23
24                        break;
25                    case 2:
26                        Square.calculateArea();
27                        break;
28                    case 3:
29                        Triangle.calculateArea();
30                        break;
31                    case 4:
32                        Rectangle.calculateArea();
33                        break;
34                    case 5:
35                        Pentagon.calculateArea();
36                        break;
37                    case 0:
38                        System.out.println("Saliendo...");
39                        break;
40                    default:
41                        System.out.println("Opción no válida");
42                }
43            } while (option != 0);
44            Circle.closeScanner();
45            Square.closeScanner();
46            Triangle.closeScanner();
47            Rectangle.closeScanner();

```

Para poder arreglar los *Resource Leak* debemos de hacer lo siguiente: El problema es al crear las variables de tipo Scanner que se inicializan pero nunca se cierra el Scanner, para poder arreglarlo debemos de crear al final de cada clase un metodo para cerrar el Scanner y llamarlo al final del metodo main, ya que sino hacemos este procedimiento obtendremos una excepción una vez hallamos finalizado de realizar un operación.

```

// Pentagon.java

import java.util.Scanner;

public class Pentagon {
    private static Scanner scanner = new Scanner(System.in);
    public static void calculateArea() {
        System.out.print("Ingrese el lado del pentágono: ");
        double side = scanner.nextDouble();
        System.out.print("Ingrese la apotema del pentágono: ");
        double apothem = scanner.nextDouble();
        double area = 0.5 * 5 * side * apothem;
        System.out.println("El área del pentágono es: " + area);
    }

    public static void closeScanner(){
        scanner.close();
    }
}

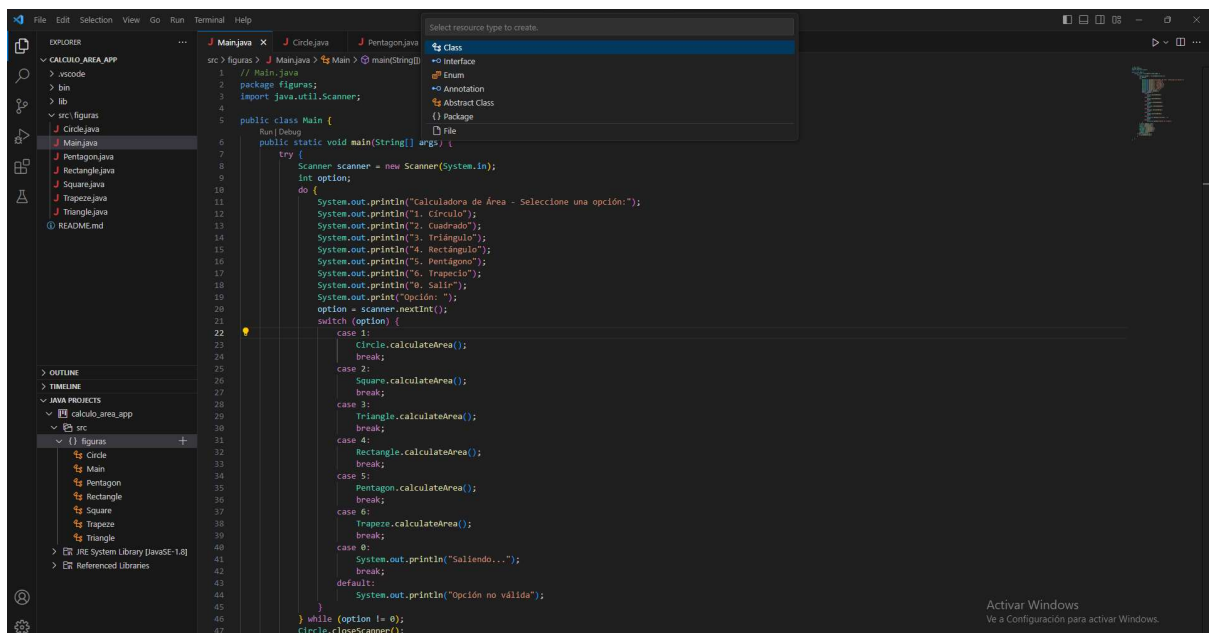
```

```
option = scanner.nextInt();
switch (option) {
    case 1:
        Circle.calculateArea();
        break;
    case 2:
        Square.calculateArea();
        break;
    case 3:
        Triangle.calculateArea();
        break;
    case 4:
        Rectangle.calculateArea();
        break;
    case 5:
        Pentagon.calculateArea();
        break;
    case 0:
        System.out.println("Saliendo...");
        break;
    default:
        System.out.println("Opción no válida");
}
} while (option != 0);
Circle.closeScanner();
Square.closeScanner();
Triangle.closeScanner();
Rectangle.closeScanner();
Pentagon.closeScanner();
scanner.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

De esta forma conseguimos arreglar los Resource Leak y que no se obtenga ninguna excepcion a la hora de ejecutar el programa.

**Nota:** Debemos de añadir a el mismo paquete los archivos .java para que asi compilen y que no tengamos ningun problema a la hora de ejecutar nuestro peroyecto , la forma de crear un paquete en VS Code es la siguiente:

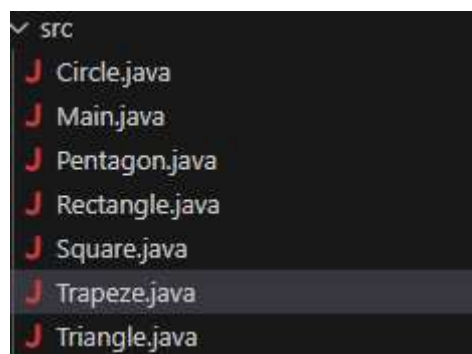




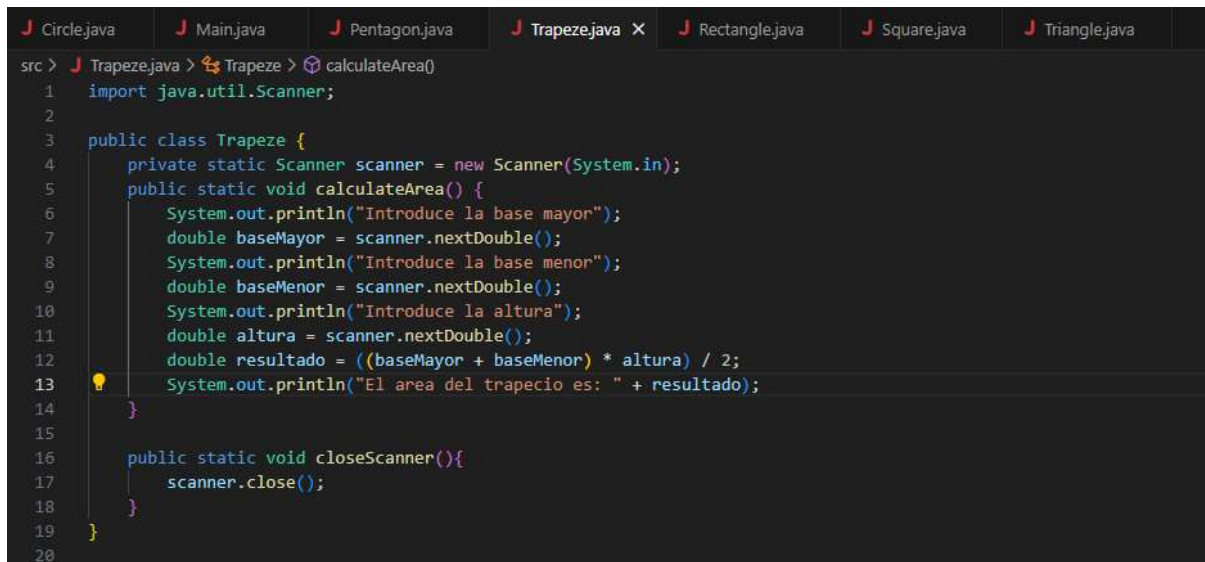
Dentro de la pestaña *Java Project* le damos a el botón + en la carpeta donde queremos añadir nuestro paquete Java, luego en las opciones que se nos muestran en el menu de arriba podemos seleccionar la opcion *Package* una vez le demos a essa opcion elegimos el nombre de nuestro paquete y ya estaria listo.

## Tarea 6: Trapecio

Lo primero que he hecho ha sido crear un archivo .java en el proyecto que he llamado Trapeze.java



Luego he implementado el código en dicho archivo de la siguiente manera para que asi el programa permita calcular el area del trapecio



```
src > J Trapeze.java > Trapeze > calculateArea()
1  import java.util.Scanner;
2
3  public class Trapeze {
4      private static Scanner scanner = new Scanner(System.in);
5      public static void calculateArea() {
6          System.out.println("Introduce la base mayor");
7          double baseMayor = scanner.nextDouble();
8          System.out.println("Introduce la base menor");
9          double baseMenor = scanner.nextDouble();
10         System.out.println("Introduce la altura");
11         double altura = scanner.nextDouble();
12         double resultado = ((baseMayor + baseMenor) * altura) / 2;
13         System.out.println("El area del trapecio es: " + resultado);
14     }
15
16     public static void closeScanner(){
17         scanner.close();
18     }
19 }
20
```

Ahora en la clase Main debemos añadir una línea al menú para que así salga la opción 6 como la del trapecio, lo he hecho de la siguiente manera:

```
System.out.println("6. Trapecio");
```

y dentro de la estructura de control del switch he hecho un nuevo case que le he asignado el número 6 y dentro he llamado al método estático *calculateArea* de la clase Trapeze

```
break;
case 6:
    Trapeze.calculateArea();
    break;
```

## Bibliografía