Primera Clase

```java
package ejercicios;

import java.util.Random;
import java.util.Scanner;

public class sumaFilasTotal {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Ingrese el número de filas");
        int f = sc.nextInt();
        System.out.println("Ingrese el número de columnas");
        int c = sc.nextInt();

        int [][] matriz = llenarMatriz(f,c);
        int [] sumas = sumarFilasTotal(matriz);
        int total = sumarTotal(sumas);

        System.out.println("Matriz          >> Sumas por filas");
        for (int i = 0; i < f; i++) {
            for (int j = 0; j < c; j++) {
                System.out.print(matriz[i][j] + " ");
            }
            System.out.println("        >> " + sumas[i]);
        }
/*
        System.out.println("Sumas por Filas");
        for (int xd: sumas) {
            System.out.println(">> " + xd);
        }
 */

        System.out.println("Suma total:" + total);
    }

    public static int[] [] llenarMatriz(int f, int c) {
        Random r = new Random();
        int [][] matriz = new int[f][c];
        for (int i = 0; i < f; i++) {
            for (int j = 0; j < c; j++) {
                matriz[i][j] = r.nextInt(9) + 1;
            }
        }

        return matriz;
    }

    public static int[] sumarFilasTotal(int[][] matriz) {
        int[] sumasFilas = new int[matriz.length];

        for (int i = 0; i < matriz.length; i++) {
            for (int j = 0; j < matriz[0].length; j++) {
                sumasFilas[i] += matriz[i][j];
            }
        }

        return sumasFilas;
    }

    public static int sumarTotal(int[] vector) {
        int total = 0;
```

```
        for (int i = 0; i < vector.length; i++) {
            total += vector[i];
        }

        return total;
    }
}
```

Segunda Clase

```
package ejercicios;

import java.util.Scanner;

public class sumaFilasTotalHilos2 implements Runnable {
    private int[] row;
    private static int[] sumaTotal;
    private int index = 0;

    public sumaFilasTotalHilos2(int[] row, int index) {
        this.row = row;
        this.index = index;
    }

    public int[] getRow() {
        return row;
    }

    public static int[] getSumaTotal() {
        return sumaTotal;
    }

    @Override
    public void run() {
        int sum = 0;
        for (int num : row) {
            sum += num;
        }

        /* ESTO ES PARA SINCRONIZAR EL ACCESO AL VECTOR Y EVITAR
INCONSISTENCIA
        synchronized (sumaFilasTotalHilos2.class) {
            sumaTotal[index] = sum;
        }
         */
        sumaTotal[index] = sum;
        System.out.printf("%s trabajando. Total: %d\n",
Thread.currentThread().getName(), sum);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Ingrese el número de filas");
        int f = sc.nextInt();
        System.out.println("Ingrese el número de columnas");
        int c = sc.nextInt();

        int[][] matriz = sumaFilasTotal.llenarMatriz(f, c);

        sumaTotal = new int[f]; // para no estar inicializando el suma total
por cada hilo
```

```
        // simplemente se hace que este sumaTotal sea static y se lo
inicializa con el tamaño de filas
        // que también será la cantidad de hilos generados

        Thread[] threads = new Thread[f];
        for (int i = 0; i < f; i++) {
            threads[i] = new Thread(new sumaFilasTotalHilos2(matriz[i], i));
// mando el índice para evitar
            // crear un contador static e ir sumándolo
            threads[i].start();
        }

        for (int i = 0; i < f; i++) {
            try {
                threads[i].join();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        int sT = 0;
        for (int xd : getSumaTotal()) {
            sT += xd;
        }

        System.out.println("Suma total: " + sT);
        presentarMatriz(matriz);
    }

    public static void presentarMatriz(int[][] matriz) {
        System.out.println(">>> PRESENTAR MATRIZ <<<");
        for (int i = 0; i < matriz.length; i++) {
            for (int j = 0; j < matriz[0].length; j++) {
                System.out.print(matriz[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Capturas:

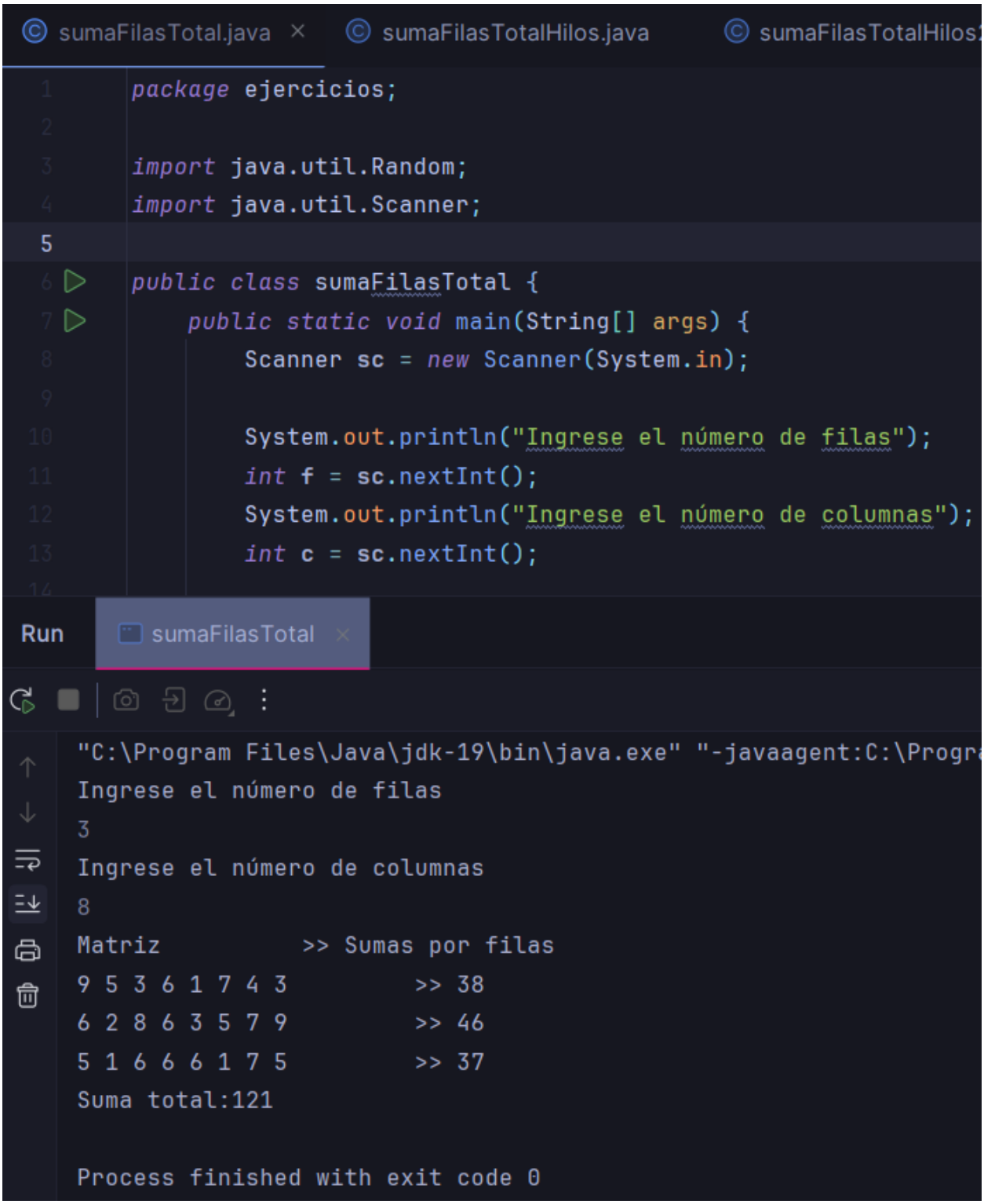
Primera clase

sumaFilasTotal.java × sumaFilasTotalHilos.java sumaFilasTotalHilos2

```java
package ejercicios;

import java.util.Random;
import java.util.Scanner;

public class sumaFilasTotal {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Ingrese el número de filas");
        int f = sc.nextInt();
        System.out.println("Ingrese el número de columnas");
        int c = sc.nextInt();
```

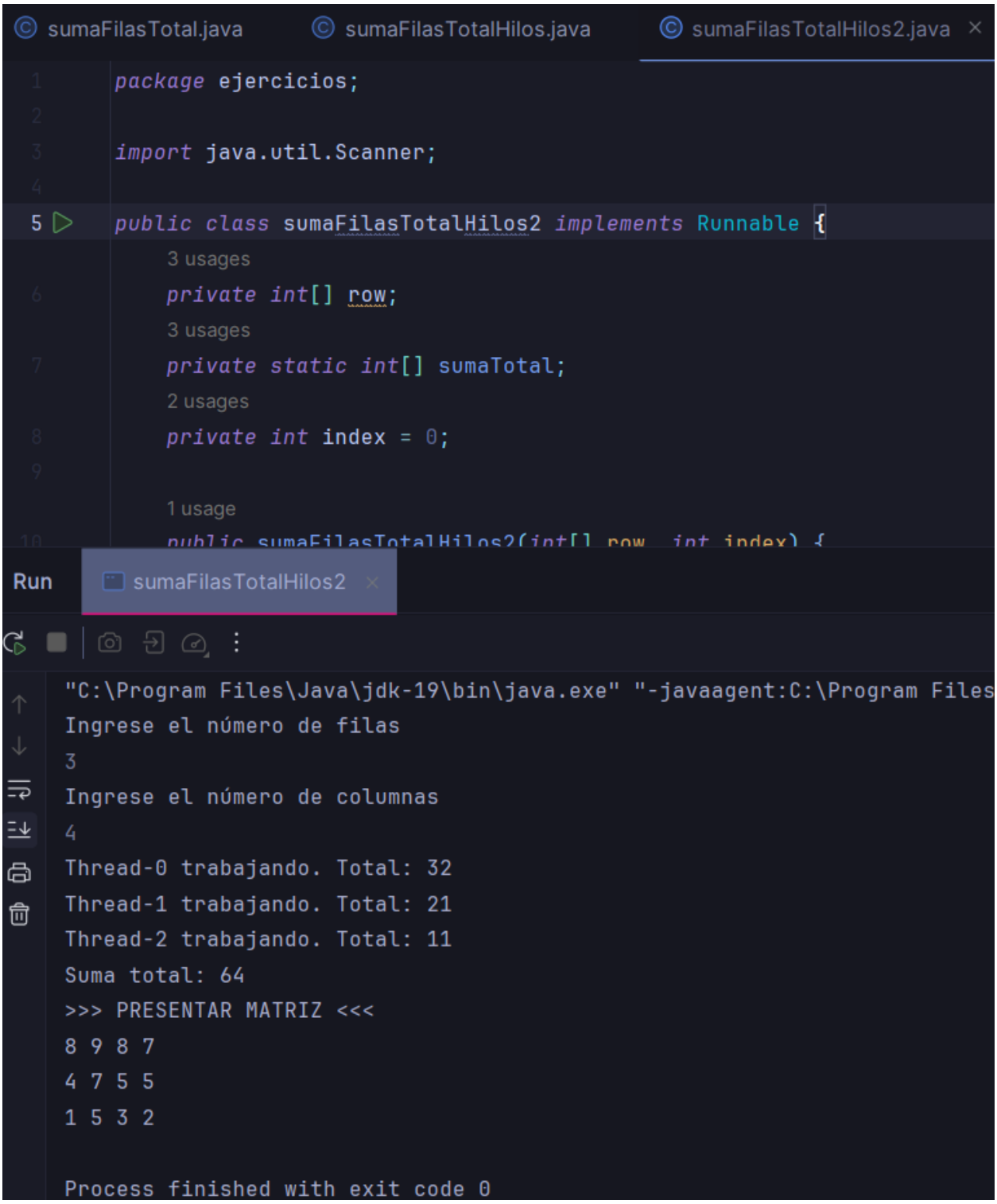
Run sumaFilasTotal ×

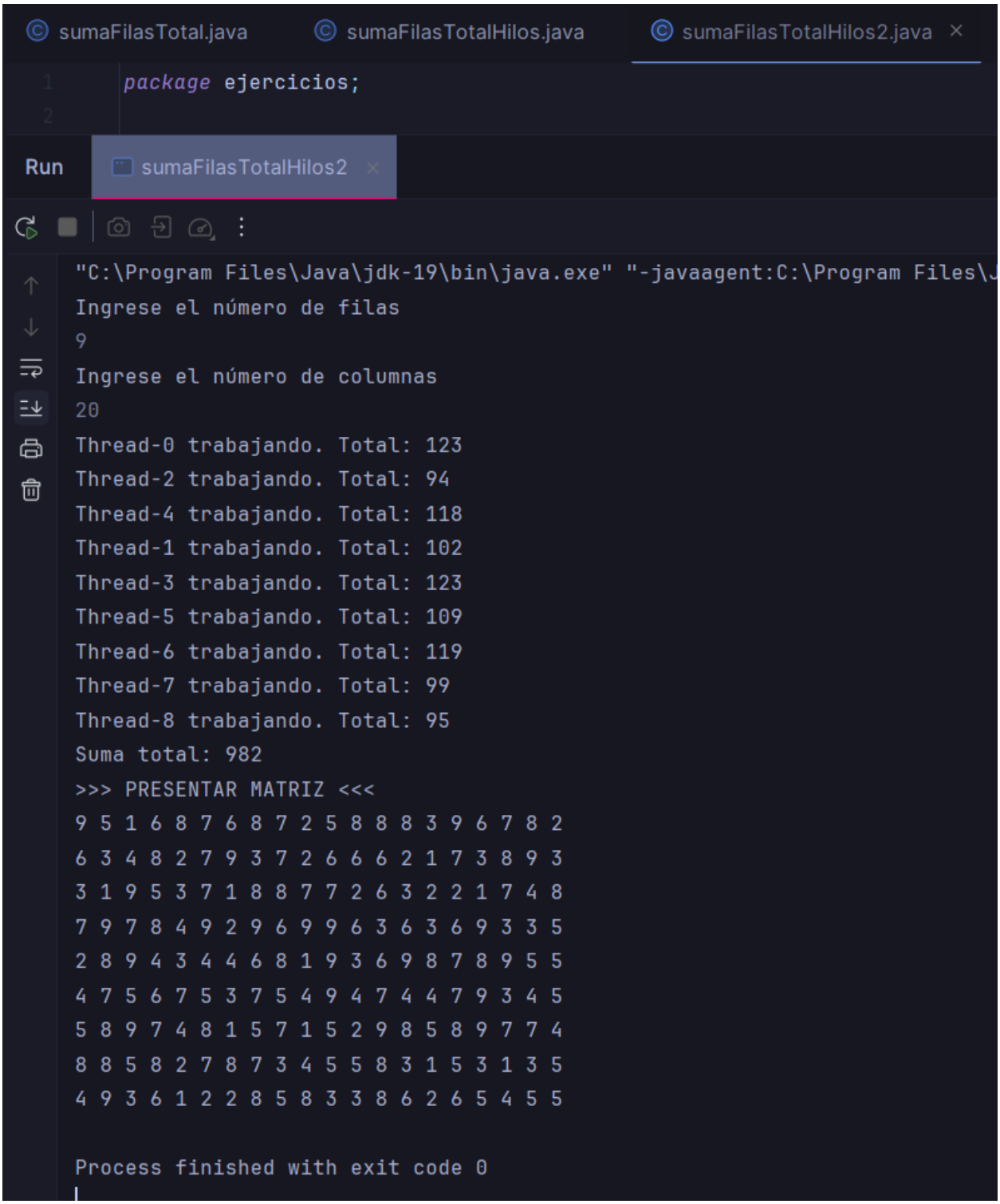
```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Progra
Ingrese el número de filas
3
Ingrese el número de columnas
5
Matriz          >> Sumas por filas
2 1 2 8 7         >> 20
5 9 8 3 5         >> 30
2 1 1 1 7         >> 12
Suma total:62

Process finished with exit code 0
```



Segunda clase

Matriz Secuencial




sumaFilasTotal.java
sumaFilasTotalHilos.java
sumaFilasTotalHilos2.java
package ejercicios;
Run
sumaFilasTotalHilos2
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\J
Ingrese el número de filas
9
Ingrese el número de columnas
20
Thread-0 trabajando. Total: 123
Thread-2 trabajando. Total: 94
Thread-4 trabajando. Total: 118
Thread-1 trabajando. Total: 102
Thread-3 trabajando. Total: 123
Thread-5 trabajando. Total: 109
Thread-6 trabajando. Total: 119
Thread-7 trabajando. Total: 99
Thread-8 trabajando. Total: 95
Suma total: 982
>>> PRESENTAR MATRIZ <<<
9 5 1 6 8 7 6 8 7 2 5 8 8 8 3 9 6 7 8 2
6 3 4 8 2 7 9 3 7 2 6 6 6 2 1 7 3 8 9 3
3 1 9 5 3 7 1 8 8 7 7 2 6 3 2 2 1 7 4 8
7 9 7 8 4 9 2 9 6 9 9 6 3 6 3 6 9 3 3 5
2 8 9 4 3 4 4 6 8 1 9 3 6 9 8 7 8 9 5 5
4 7 5 6 7 5 3 7 5 4 9 4 7 4 4 7 9 3 4 5
5 8 9 7 4 8 1 5 7 1 5 2 9 8 5 8 9 7 7 4
8 8 5 8 2 7 8 7 3 4 5 5 8 3 1 5 3 1 3 5
4 9 3 6 1 2 2 8 5 8 3 3 8 6 2 6 5 4 5 5
Process finished with exit code 0