

**UNIVERSIDAD DON BOSCO.**



**FACULTAD DE INGENIERÍA  
ESCUELA DE COMPUTACIÓN.**

**Asignatura:  
Desarrollo de Software para Móviles**

**Actividad:  
Guía de laboratorio 01**

**Grupo de laboratorio:  
03L**

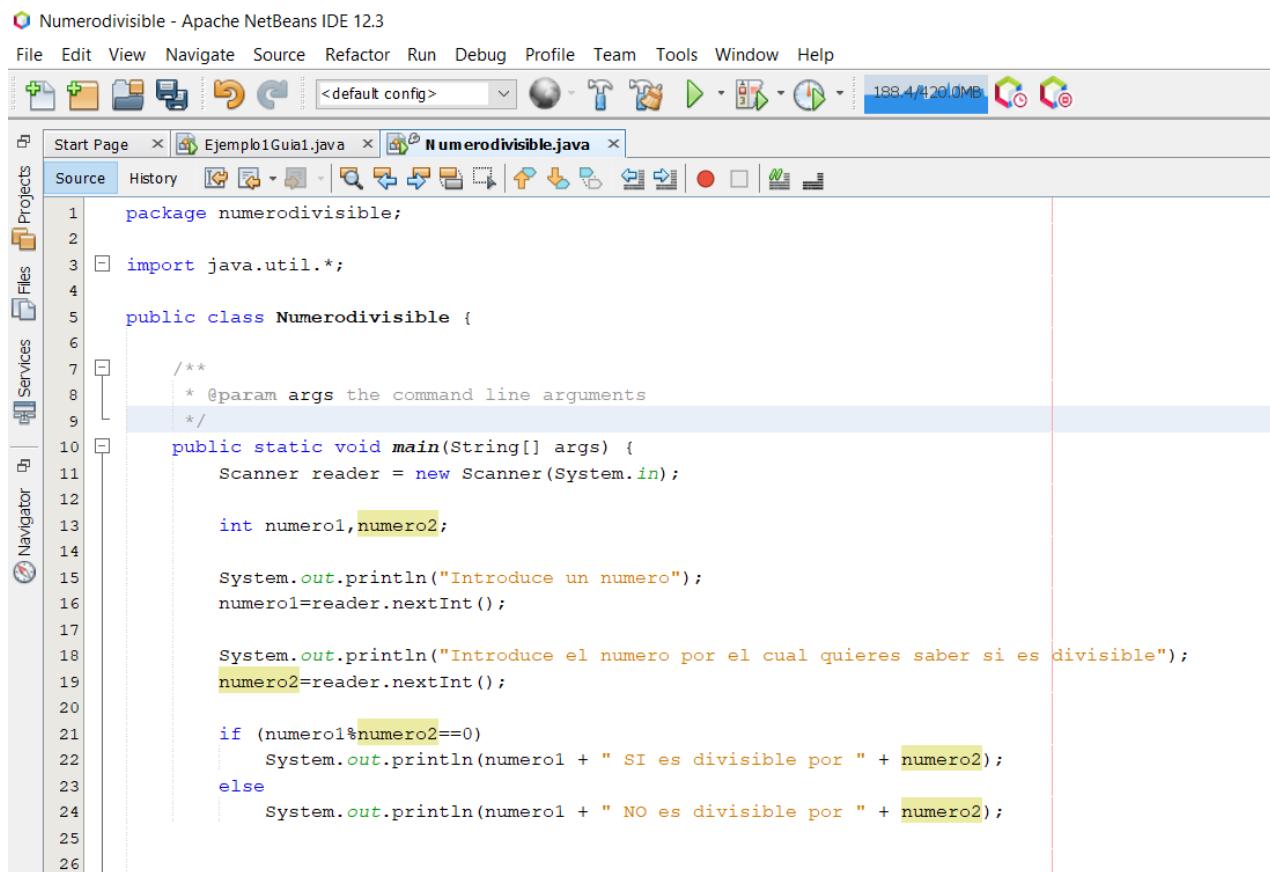
**Alumna:** Angulo Moreno, Diana Yaneth      **Carné** AM170125

## **Contenido**

Ejercicio 1 .....	3
Ejercicio 2 .....	4
Ejercicio 3 .....	6
Ejercicio 4 .....	8
Creación de emuladores .....	11
Ejercicio 1 .....	11
Ejercicio 2 .....	15
Ejercicio 3 .....	17

## Ejercicio 1

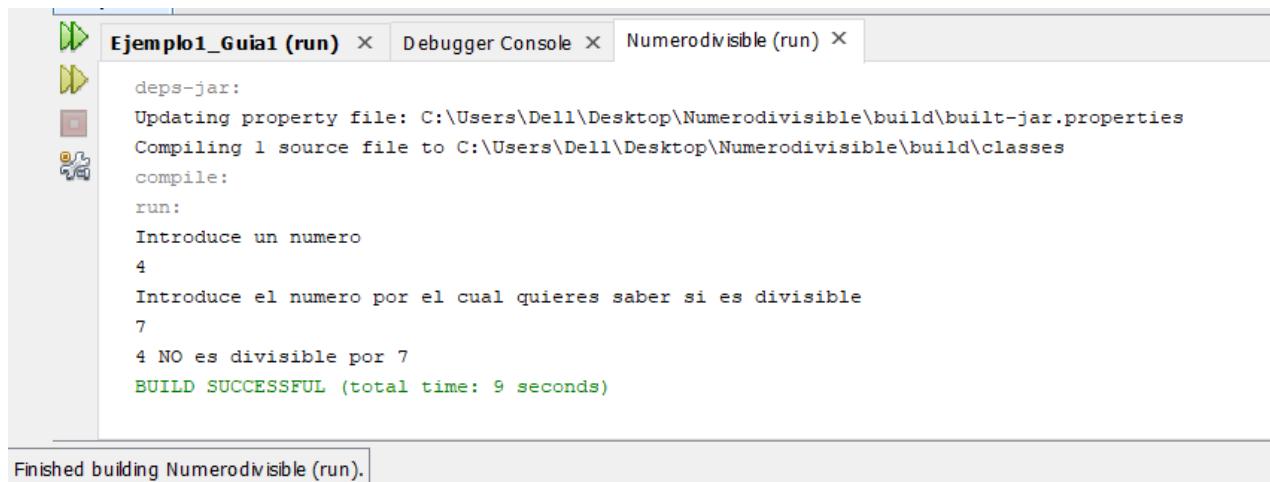
### Código



The screenshot shows the Apache NetBeans IDE interface. The title bar reads "Numerodivisible - Apache NetBeans IDE 12.3". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar has various icons for file operations like New, Open, Save, and Build. The status bar at the bottom right shows "188.4/420.0MB" and some system icons. The central workspace shows two tabs: "Ejemplo1Guia1.java" and "Numerodivisible.java". The "Numerodivisible.java" tab is active, displaying the following Java code:

```
1 package numerodivisible;
2
3 import java.util.*;
4
5 public class Numerodivisible {
6
7     /**
8      * @param args the command line arguments
9     */
10    public static void main(String[] args) {
11        Scanner reader = new Scanner(System.in);
12
13        int numero1,numero2;
14
15        System.out.println("Introduce un numero");
16        numero1=reader.nextInt();
17
18        System.out.println("Introduce el numero por el cual quieras saber si es divisible");
19        numero2=reader.nextInt();
20
21        if (numero1%numero2==0)
22            System.out.println(numero1 + " SI es divisible por " + numero2);
23        else
24            System.out.println(numero1 + " NO es divisible por " + numero2);
25
26    }
27}
```

### Resultado



The screenshot shows the "Run" tab of the NetBeans IDE. The title bar has three tabs: "Ejemplo1\_Guia1 (run)", "Debugger Console", and "Numerodivisible (run)". The "Numerodivisible (run)" tab is active and displays the following output:

```
deps-jar:
Updating property file: C:\Users\DELL\Desktop\Numerodivisible\build\built-jar.properties
Compiling 1 source file to C:\Users\DELL\Desktop\Numerodivisible\build\classes
compile:
run:
Introduce un numero
4
Introduce el numero por el cual quieras saber si es divisible
7
4 NO es divisible por 7
BUILD SUCCESSFUL (total time: 9 seconds)
```

At the bottom of the window, a message box says "Finished building Numerodivisible (run)."

Ejemplo1\_Guia1 (run) × Debugger Console × Numerodivisible (run) ×

```

ant -f C:\\Users\\Dell\\Desktop\\Numerodivisible -Dnb.internal.action.name=run run
init:
Deleting: C:\\Users\\Dell\\Desktop\\Numerodivisible\\build\\built-jar.properties
deps-jar:
Updating property file: C:\\Users\\Dell\\Desktop\\Numerodivisible\\build\\built-jar.properties
Compiling 1 source file to C:\\Users\\Dell\\Desktop\\Numerodivisible\\build\\classes
compile:
run:
Introduce un numero
4
Introduce el numero por el cual quieras saber si es divisible
2
4 SI es divisible por 2
BUILD SUCCESSFUL (total time: 5 seconds)
|
```

## Ejercicio 2

### Código

evaluadordenotas - Apache NetBeans IDE 12.3

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Start Page × Ejemplo1Guia1.java × Numerodivisible.java × Evaluadordenotas.java ×

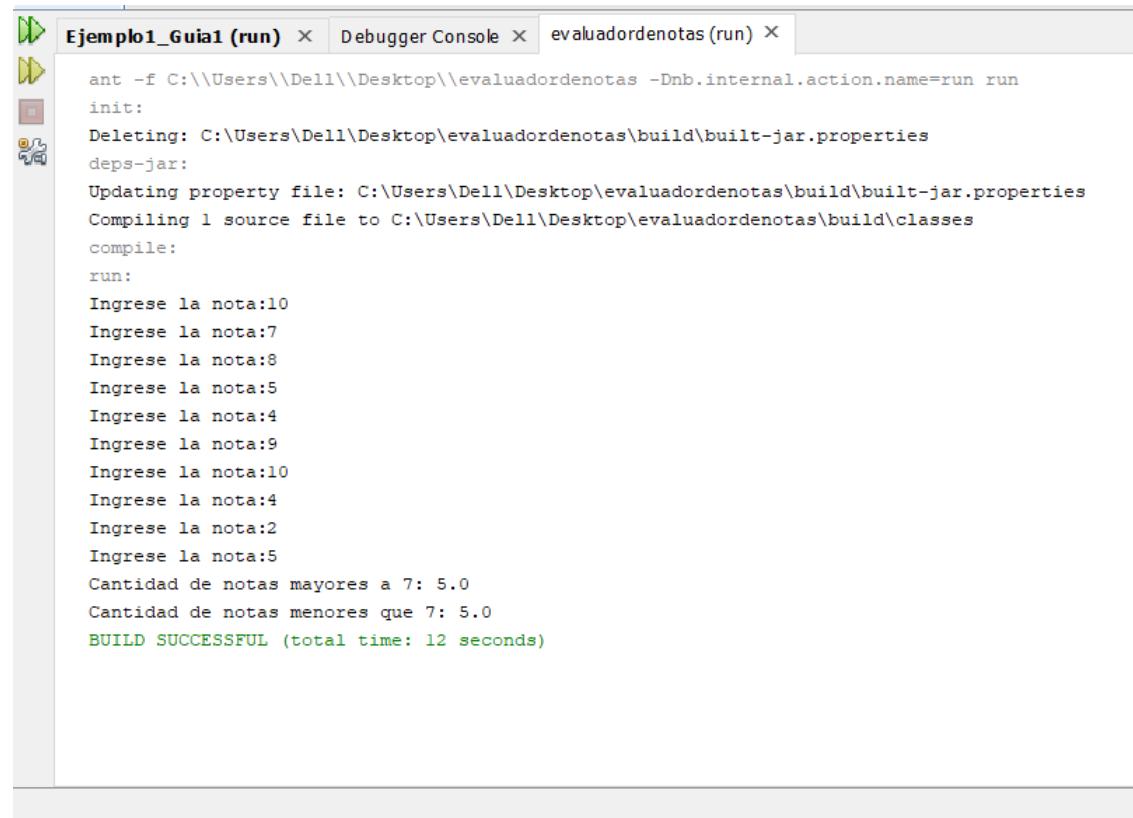
Projects Files Services Navigator

Source History |

```

9  import java.util.*;
10 
11 /**
12  * @author Dell
13  */
14 public class Evaluadordenotas {
15 
16     /**
17      * @param args the command line arguments
18     */
19     public static void main(String[] args) {
20         Scanner teclado=new Scanner(System.in);
21         float mayores,menores,fnota;
22         mayores=0;
23         menores=0;
24         for(f=1;f<=10;f++) {
25             System.out.print("Ingrese la nota:");
26             nota=teclado.nextInt();
27             if (nota>=7) {
28                 mayores=mayores+1;
29             } else {
30                 menores=menores+1;
31             }
32         }
33         System.out.println("Cantidad de notas mayores a 7: " + mayores);
34         System.out.println("Cantidad de notas menores que 7: " + menores);
35     }
36 }
37 }
38 }
39 }
```

## Resultado

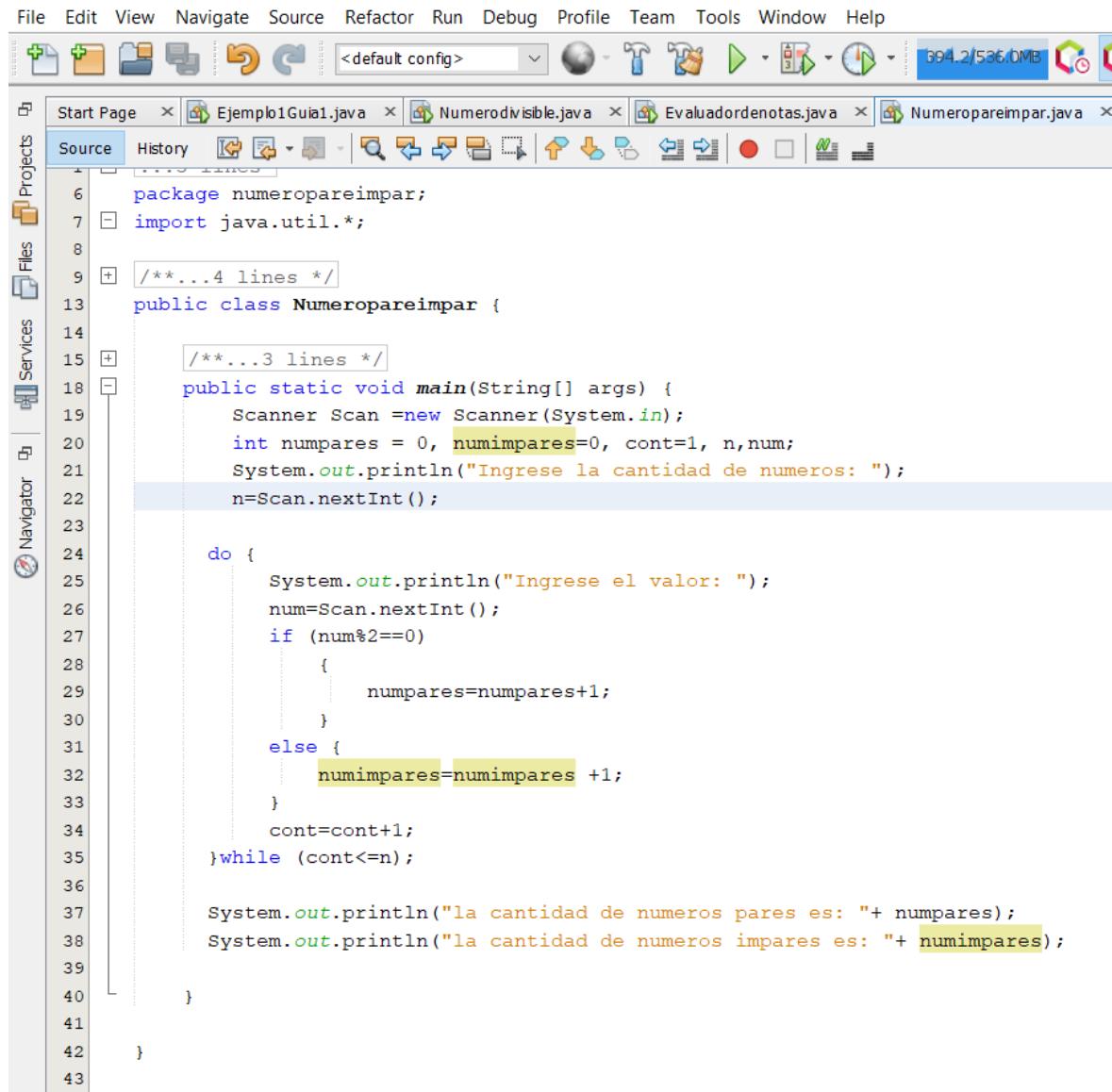


The screenshot shows a Java IDE interface with three tabs at the top: 'Ejemplo1\_Guia1 (run)', 'Debugger Console', and 'evaluadordenotas (run)'. The 'evaluadordenotas (run)' tab is active and displays the following build log:

```
ant -f C:\\Users\\Dell\\Desktop\\evaluadordenotas -Dnb.internal.action.name=run run
init:
Deleting: C:\\Users\\Dell\\Desktop\\evaluadordenotas\\build\\built-jar.properties
deps-jar:
Updating property file: C:\\Users\\Dell\\Desktop\\evaluadordenotas\\build\\built-jar.properties
Compiling 1 source file to C:\\Users\\Dell\\Desktop\\evaluadordenotas\\build\\classes
compile:
run:
Ingrese la nota:10
Ingrese la nota:7
Ingrese la nota:8
Ingrese la nota:5
Ingrese la nota:4
Ingrese la nota:9
Ingrese la nota:10
Ingrese la nota:4
Ingrese la nota:2
Ingrese la nota:5
Cantidad de notas mayores a 7: 5.0
Cantidad de notas menores que 7: 5.0
BUILD SUCCESSFUL (total time: 12 seconds)
```

## Ejercicio 3

### Código



The screenshot shows a Java development environment with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Toolbar:** Standard icons for file operations like Open, Save, Print, and a search icon.
- Project Bar:** Shows multiple open projects: Start Page, Ejemplo1Guia1.java, Numerodivisible.java, Evaluadordenotas.java, and Numeropareimpar.java.
- Sidebar:** Projects, Files, Services, and Navigator.
- Code Editor:** Displays the source code for the `Numeropareimpar` class. The code is as follows:

```
6 package numeropareimpar;
7 import java.util.*;
8
9 /**
10  * ...4 lines
11 */
12 public class Numeropareimpar {
13
14     /**
15      * ...3 lines
16     */
17     public static void main(String[] args) {
18         Scanner Scan =new Scanner(System.in);
19         int numpares = 0, numimpares=0, cont=1, n,num;
20         System.out.println("Ingrese la cantidad de numeros: ");
21         n=Scan.nextInt();
22
23         do {
24             System.out.println("Ingrese el valor: ");
25             num=Scan.nextInt();
26             if (num%2==0)
27                 {
28                     numpares=numpares+1;
29                 }
30             else {
31                 numimpares=numimpares +1;
32             }
33             cont=cont+1;
34         }while (cont<=n);
35
36         System.out.println("la cantidad de numeros pares es: "+ numpares);
37         System.out.println("la cantidad de numeros impares es: "+ numimpares);
38
39     }
40 }
41
42 }
43 }
```

## Resultados

The screenshot shows the 'Output - numeropareimpar (run)' tab in an IDE. The left sidebar includes 'Services' and 'Navigator' icons. The output window displays the following text:

```
ant -f C:\\Users\\Dell\\Desktop\\numeropareimpar -Dnb.internal.action.name=run run
init:
deps-jar:
    Created dir: C:\\Users\\Dell\\Desktop\\numeropareimpar\\build
    Updating property file: C:\\Users\\Dell\\Desktop\\numeropareimpar\\build\\built-jar.properties
    Created dir: C:\\Users\\Dell\\Desktop\\numeropareimpar\\build\\classes
    Created dir: C:\\Users\\Dell\\Desktop\\numeropareimpar\\build\\empty
    Created dir: C:\\Users\\Dell\\Desktop\\numeropareimpar\\build\\generated-sources\\ap-source-output
    Compiling 1 source file to C:\\Users\\Dell\\Desktop\\numeropareimpar\\build\\classes
compile:
run:
Ingrrese la cantidad de numeros:
6
Ingrrese el valor:
3
Ingrrese el valor:
4
Ingrrese el valor:
6
Ingrrese el valor:
7
Ingrrese el valor:
10
Ingrrese el valor:
20
la cantidad de numeros pares es: 4
la cantidad de numeros impares es: 2
BUILD SUCCESSFUL (total time: 12 seconds)
```

## Ejercicio 4

## Código

The screenshot shows the Apache NetBeans IDE 12.3 interface. The title bar reads "coordenadas - Apache NetBeans IDE 12.3". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar contains various icons for file operations like Open, Save, and Build. The status bar at the bottom right shows "B13.2/536.0MB". The left sidebar has tabs for Projects, Files, and Services. The main area displays Java code for a class named Coordenadas. The code uses Scanner to read input from the console and counts points based on their coordinates (x, y). The code is as follows:

```
1 ...5 lines
2 package coordenadas;
3
4 import java.util.*;
5
6 /**
7  * @param args the command line arguments
8 */
9
10 public class Coordenadas {
11
12     /**
13      * @param args the command line arguments
14     */
15
16     public static void main(String[] args) {
17         Scanner teclado=new Scanner(System.in);
18         int n,f,x,y,cant1,cant2,cant3,cant4;
19         cant1=0;
20         cant2=0;
21         cant3=0;
22         cant4=0;
23
24         System.out.print("Cantidad de puntos:");
25         n=teclado.nextInt();
26         for(f=1;f<=n;f++) {
27             System.out.print("Ingrese coordenada x:");
28             x=teclado.nextInt();
29             System.out.print("Ingrese coordenada y:");
30             y=teclado.nextInt();
31             if (x>0 && y>0) {
32                 cant1++;
33             } else {
34                 if (x<0 && y>0) {
35                     cant2++;
36                 } else {
37                     if (x<0 && y<0) {
38                         cant3++;
39                     } else {
40                         if (x>0 && y<0) {
41                             cant4++;
42                         }
43                     }
44                 }
45             }
46         }
47     }
48 }
```

coordenadas - Apache NetBeans IDE 12.3

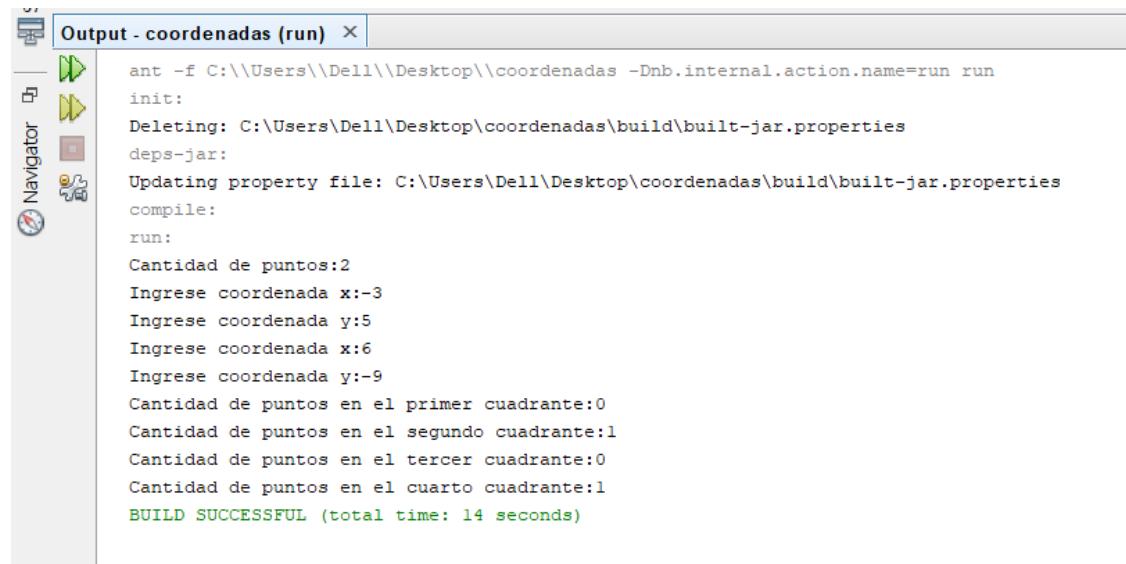
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Start Page Ejemplo1Guia1.java Numerodivisible.java Evaluadordenotas.java Numeropareimpar.java

Source History

```
26 System.out.print("Cantidad de puntos:");
27 n=teclado.nextInt();
28 for(f=1;f<=n;f++) {
29     System.out.print("Ingrese coordenada x:");
30     x=teclado.nextInt();
31     System.out.print("Ingrese coordenada y:");
32     y=teclado.nextInt();
33     if (x>0 && y>0) {
34         cant1++;
35     } else {
36         if (x<0 && y>0) {
37             cant2++;
38         } else {
39             if (x<0 && y<0) {
40                 cant3++;
41             } else {
42                 if (x>0 && y<0) {
43                     cant4++;
44                 }
45             }
46         }
47     }
48 }
49 System.out.print("Cantidad de puntos en el primer cuadrante:");
50 System.out.println(cant1);
51 System.out.print("Cantidad de puntos en el segundo cuadrante:");
52 System.out.println(cant2);
53 System.out.print("Cantidad de puntos en el tercer cuadrante:");
54 System.out.println(cant3);
55 System.out.print("Cantidad de puntos en el cuarto cuadrante:");
56 System.out.println(cant4);
57 }
58 }
59 }
```

## Resultados

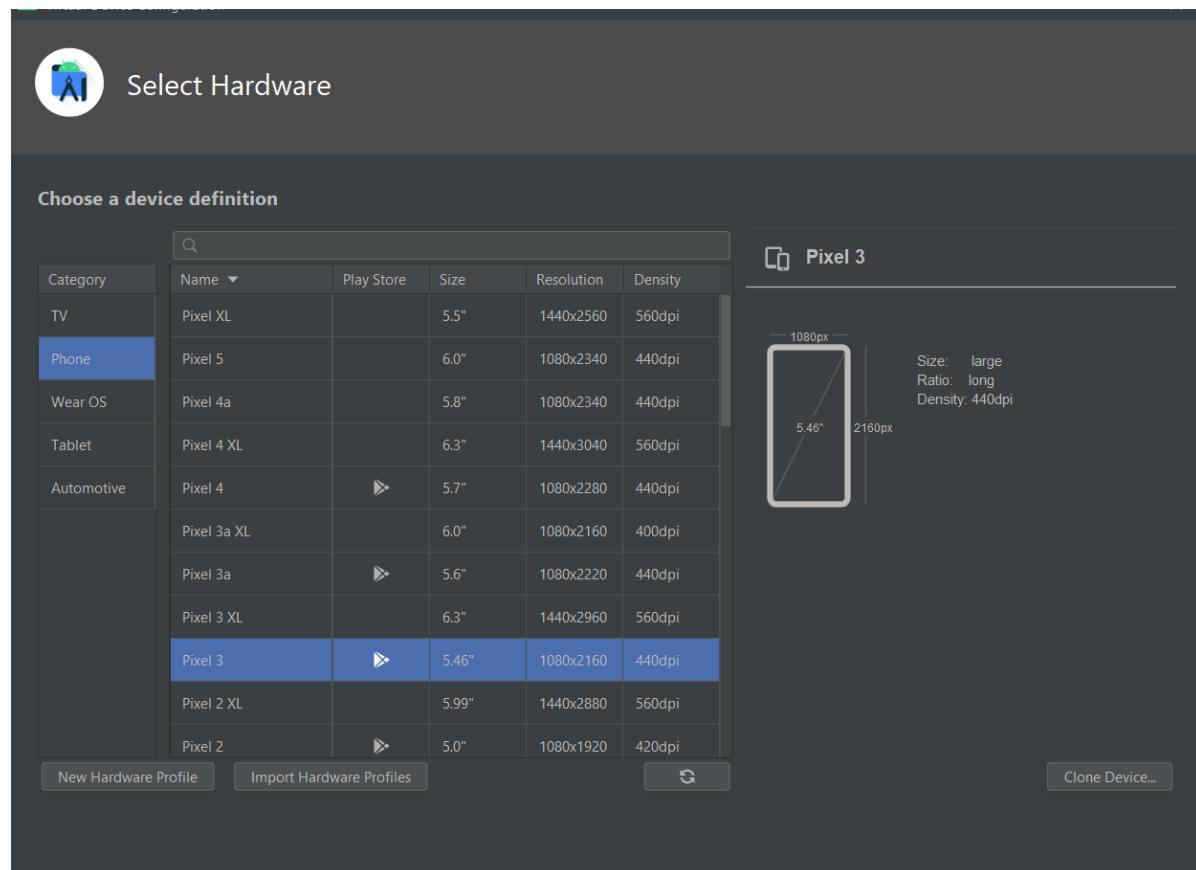


The screenshot shows the 'Output - coordenadas (run)' window in an IDE. The left sidebar has icons for Navigator, Run, Stop, and Help. The main area displays the following text:

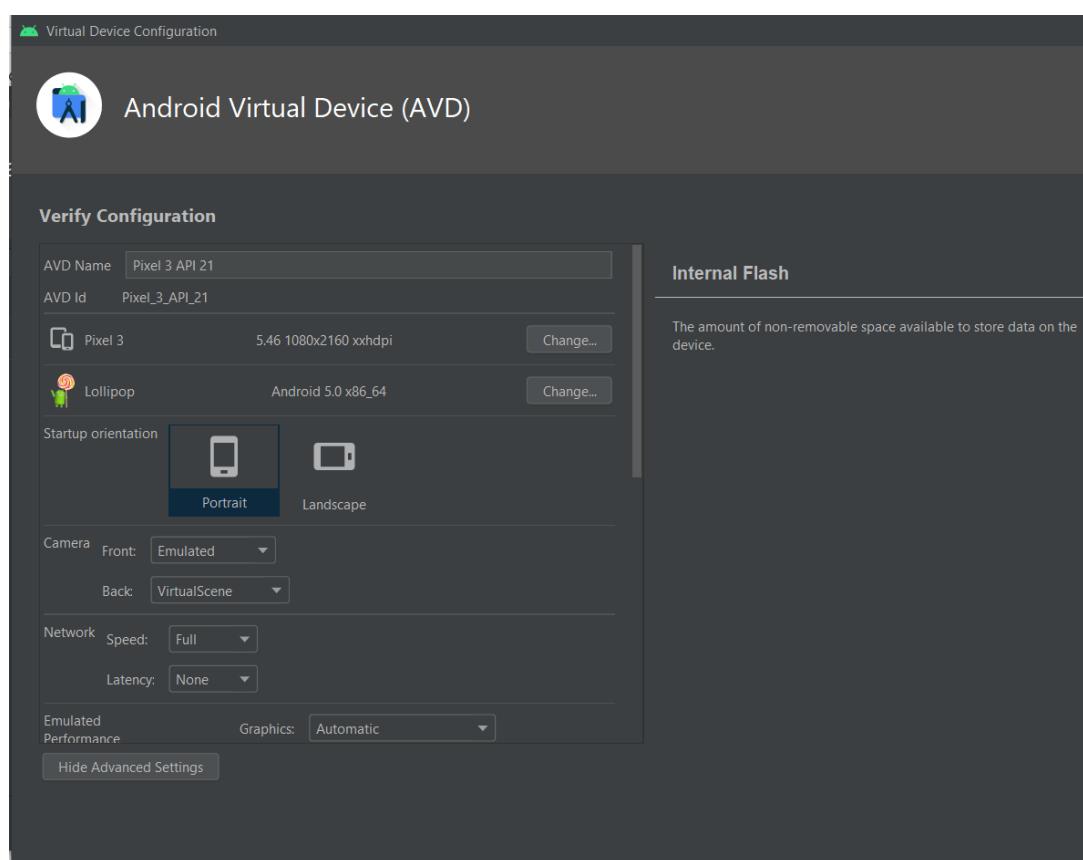
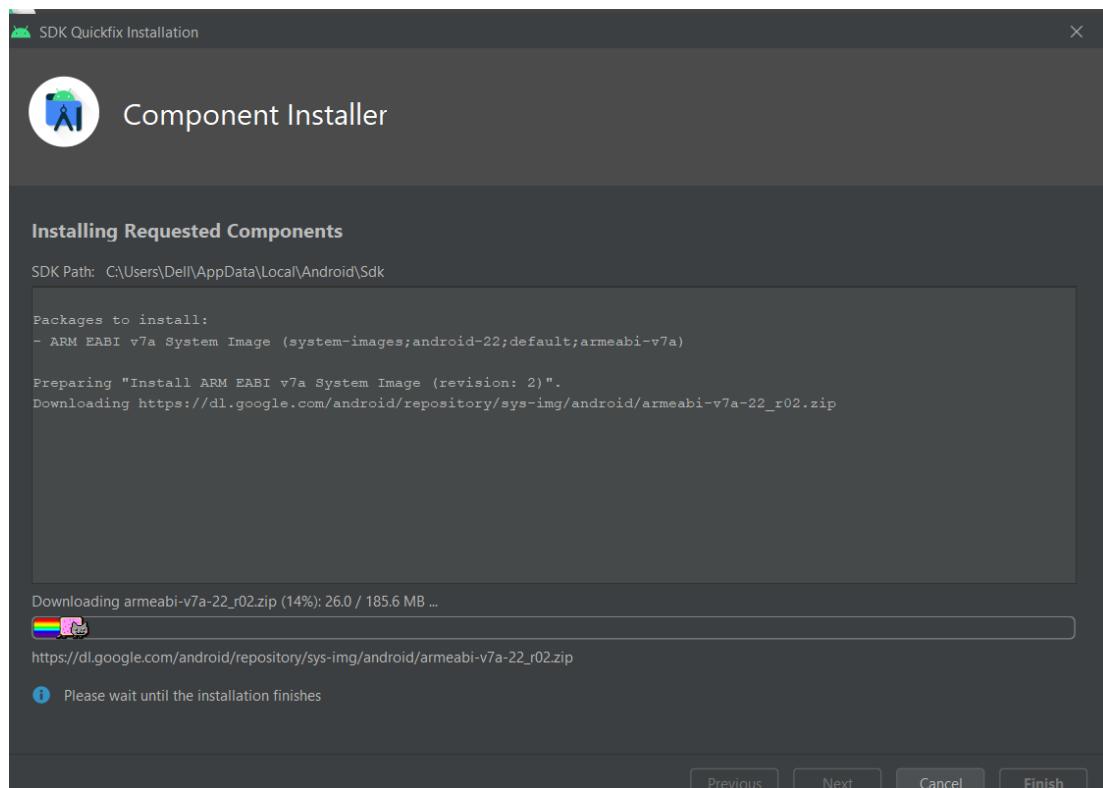
```
ant -f C:\\Users\\Dell\\Desktop\\coordenadas -Dnb.internal.action.name=run run
init:
Deleting: C:\\Users\\Dell\\Desktop\\coordenadas\\build\\built-jar.properties
deps-jar:
Updating property file: C:\\Users\\Dell\\Desktop\\coordenadas\\build\\built-jar.properties
compile:
run:
Cantidad de puntos:2
Ingrese coordenada x:-3
Ingrese coordenada y:5
Ingrese coordenada x:6
Ingrese coordenada y:-9
Cantidad de puntos en el primer cuadrante:0
Cantidad de puntos en el segundo cuadrante:1
Cantidad de puntos en el tercer cuadrante:0
Cantidad de puntos en el cuarto cuadrante:1
BUILD SUCCESSFUL (total time: 14 seconds)
```

## Creación de emuladores

### Ejercicio 1



## Descargando la version de android a usar



Virtual Device Configuration

## Android Virtual Device (AVD)

### Verify Configuration

Choose from snapshot  
(no snapshots) ▾

Multi-Core CPU 4

---

Memory and Storage

RAM:	500	MB
VM heap:	256	MB
Internal Storage:	7500	MB
SD card:	<input checked="" type="radio"/> Studio-managed 512	MB
	<input type="radio"/> External file	...
	<input type="radio"/> No SDCard	

---

Device Frame  Enable Device Frame

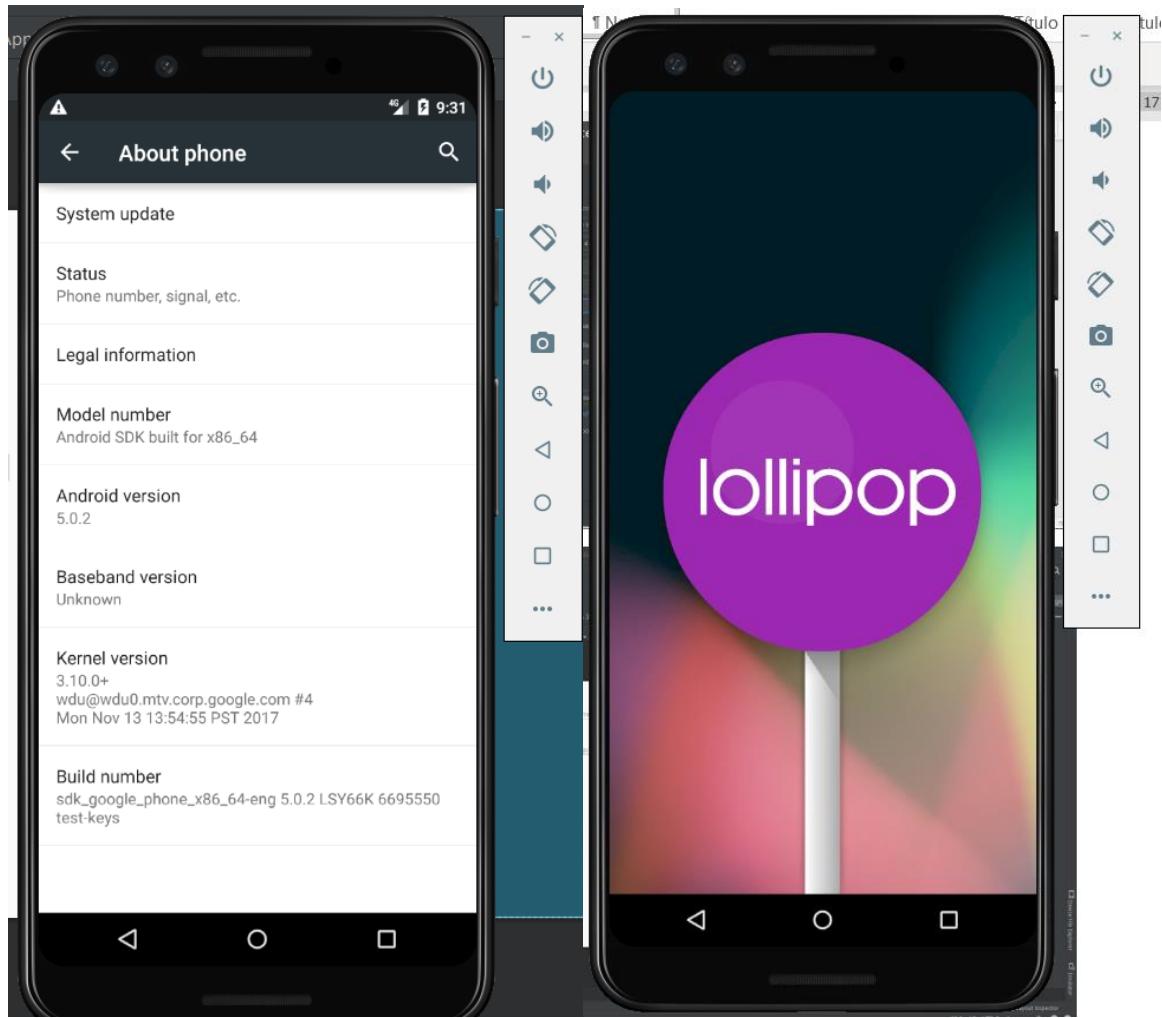
Custom skin definition pixel\_3 ...

How do I create a custom hardware skin? ↗

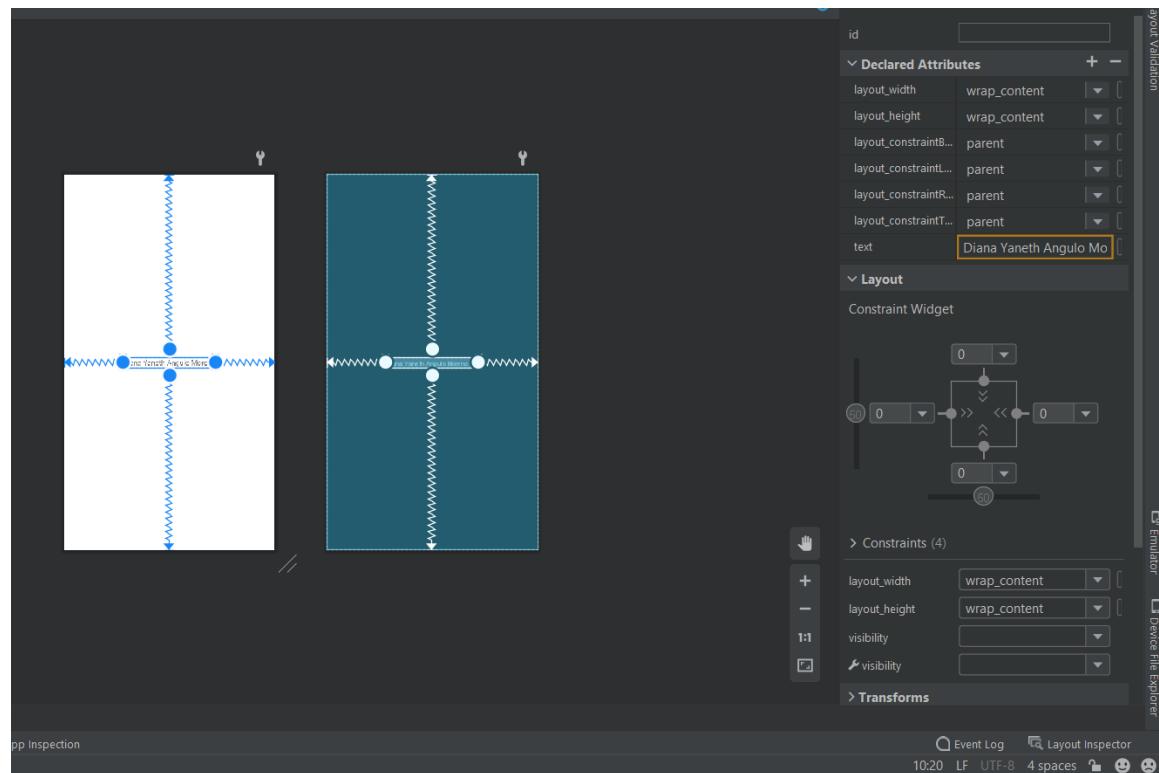
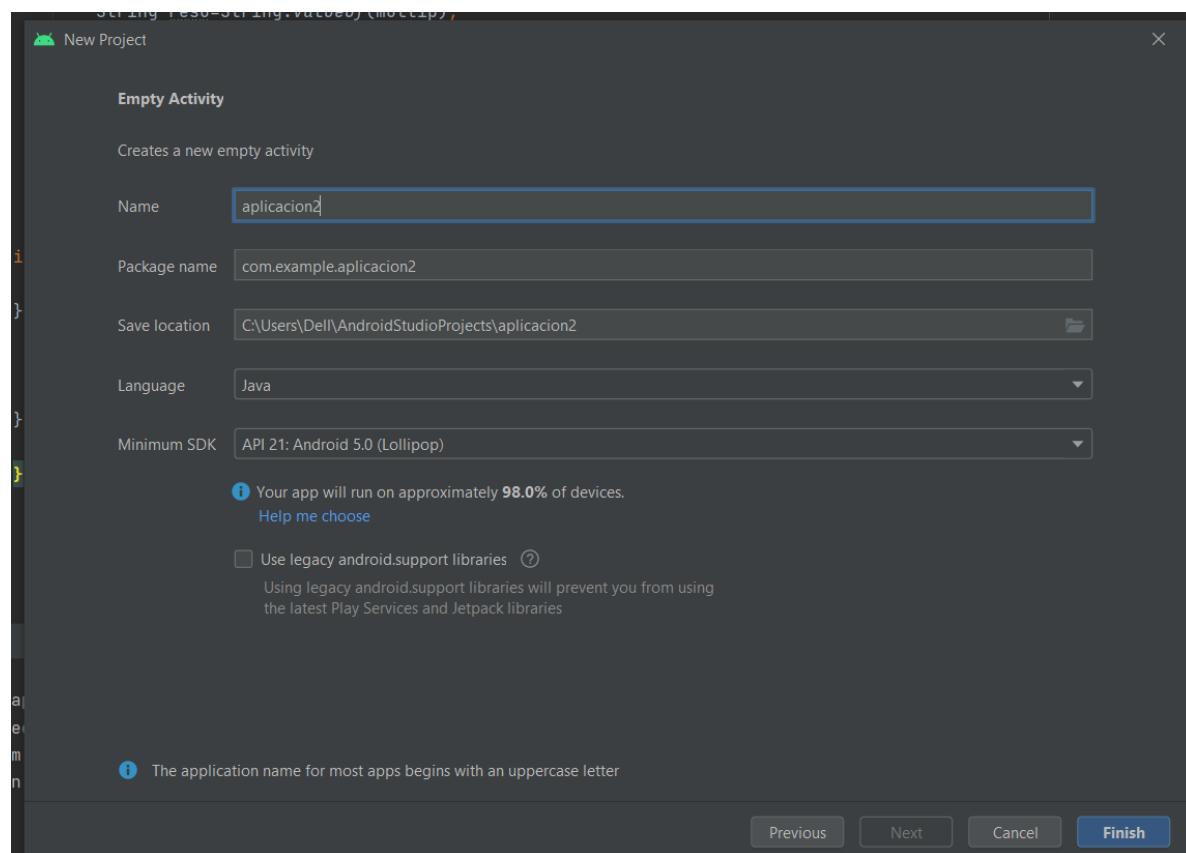
---

Keyboard  Enable keyboard input

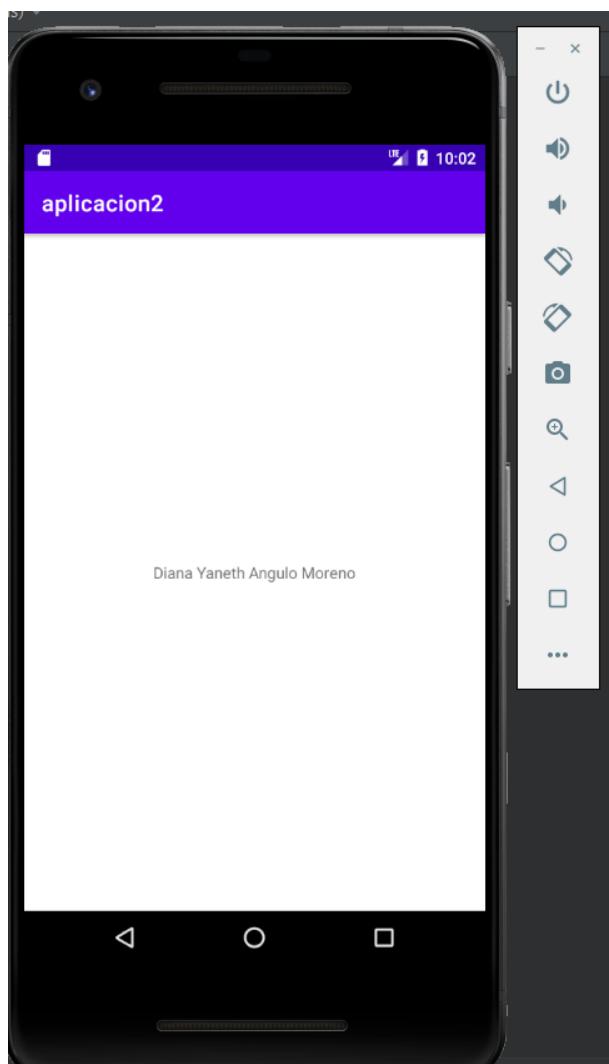
[Hide Advanced Settings](#)



## Ejercicio 2



## Resultado



## Ejercicio 3

### Código

The screenshot shows the Android Studio interface with the code editor open to the `MainActivity.java` file. The code implements a simple addition application.

```
package com.example.myapplication;
import ...;

public class MainActivity extends AppCompatActivity {
    private EditText et1;
    private EditText et2;
    private TextView tv1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        et1=findViewById(R.id.et1);
        et2=findViewById(R.id.et2);
        tv1=findViewById(R.id.tv1);
    }

    public void sumar(View view){
        String valor1=et1.getText().toString();
        String valor2=et2.getText().toString();

        int nro1=Integer.parseInt(valor1);
        int nro2=Integer.parseInt(valor2);

        int suma=nro1+nro2;

        String resu=String.valueOf(suma);
        tv1.setText(resu);
    }
}
```

The code defines a `MainActivity` that extends `AppCompatActivity`. It contains three fields: `et1`, `et2`, and `tv1`. The `onCreate` method sets the content view to `activity_main` and initializes the views. The `sumar` method retrieves the text from `et1` and `et2`, converts it to integers, adds them, and then sets the result to `tv1`.

```
public class MainActivity extends AppCompatActivity {

    public void restart(View view){
        String valor1=et1.getText().toString();
        String valor2=et2.getText().toString();
        int nro1=Integer.parseInt(valor1);
        int nro2=Integer.parseInt(valor2);
        int resta=nro1-nro2;
        String resu=String.valueOf(resta);
        tv1.setText(resu);

    }

    public void multiplicar(View view){
        String valor1=et1.getText().toString();
        String valor2=et2.getText().toString();
        int nro1=Integer.parseInt(valor1);
        int nro2=Integer.parseInt(valor2);
        int multip=nro1*nro2;
        String resu=String.valueOf(multip);
        tv1.setText(resu);

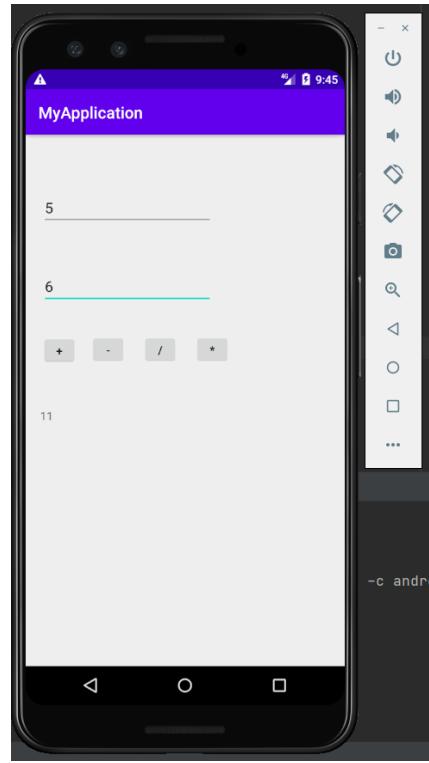
    }

    public void dividir(View view){
        String valor1=et1.getText().toString();
        String valor2=et2.getText().toString();
        int nro1=Integer.parseInt(valor1);
        int nro2=Integer.parseInt(valor2);
    if (nro2==0) {
        System.out.print("Ingrese un valor diferente a 0");
    } else {
        int div=nro1/nro2;
        String resu=String.valueOf(div);
        tv1.setText(resu);
    }
}

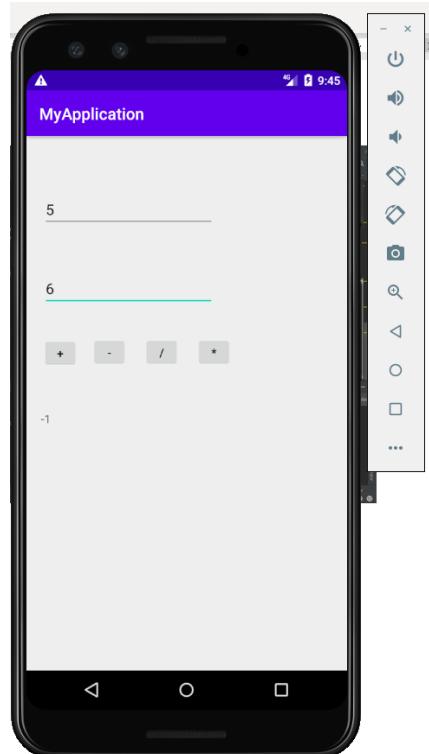
}
```

## Resultados

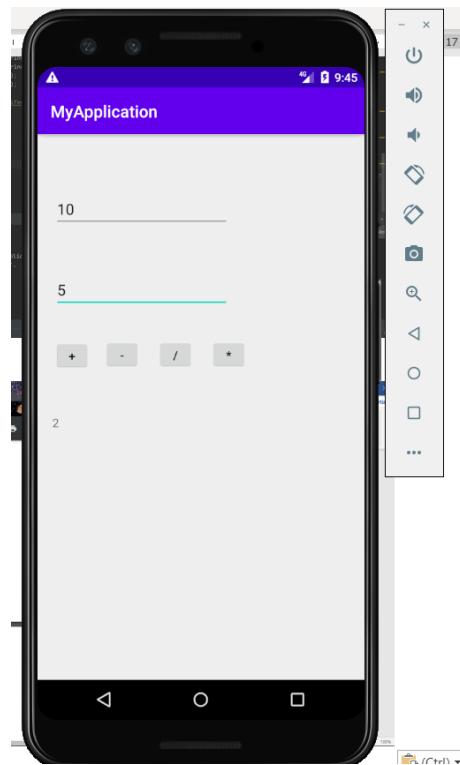
### Suma



### Resta



Division



Multiplicación

