



Instituto Politécnico Nacional  
ESCOM  
“Escuela Superior de Cómputo”



Matemáticas Avanzadas para la Ingeniería

14/02/2022

Tarea 2

Números

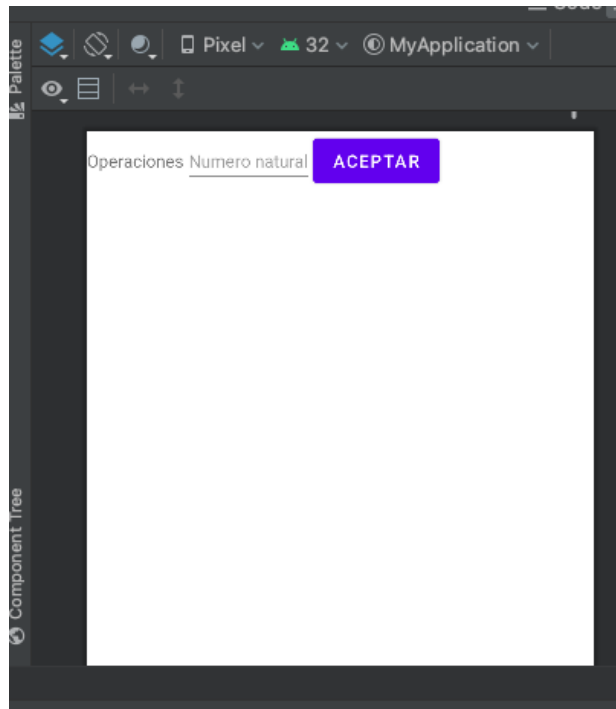
Profesor: Cifuentes Álvarez Alejandro

Sigfrido

Grupo: 3CM16

Nombre: Santiago Pérez Carlos Augusto

En este programa se muestran en pantalla un TextView, el cual indicará las operaciones a realizar por el usuario, un EditText en el cual el usuario podrá ingresar los números naturales que el quiera, un botón que tendrá como texto “Aceptar” y por último un EditText que le mostrará al usuario si el número es maravilloso, es primo y si pertenece a la serie Fibonacci. La interfaz desde Android Studio se ve de la siguiente manera.



Una vez declaradas las variables de texto como en la primera tarea obtendremos un resultado como la imagen.

En el archivo activity\_main.xml es importante declarar el LinearLayout.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

Declaramos el id en cada uno de los elementos que vamos a utilizar en nuestro MainActivity.java, los cuales son el EditText, Button y el segundo TextView.

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/operaciones" />

<EditText
    android:id="@+id/xet1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:autofillHints="no"
    android:hint="@string/natural"
    android:textSize="14sp" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/aceptar"
    android:id="@+id/xbn1" />

<TextView
    android:id="@+id/xtv2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

```

Declaramos nuestros elementos como variables y le guardamos el valor obteniéndose a partir de su id.

```

EditText jet1;
Button jbn1;
TextView jtv2;
int n;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //EditText donde se ingresa el numero
    jet1 = (EditText) findViewById(R.id.xet1);
    //Button de aceptar
    jbn1 = (Button) findViewById(R.id.xbn1);
    jbn1.setOnClickListener(this);
    //TextView donde se muestra los resultados
    jtv2 = (TextView) findViewById(R.id.xtv2);
}

```

Creamos una función para cada uno de las demostraciones que se quieren realizar con los números, para saber si es maravilloso, primo y si pertenece a la serie Fibonacci.

```
public static String esMaravilloso(int num){
    String maravilloso;
    if (num<0){
        maravilloso = "No es maravilloso\n";
    }
    else{
        maravilloso = "Es maravilloso\n";
    }
    return maravilloso;
}

public static boolean esPrimo(int num){
    int cont=2;
    boolean primo = true;
    while((primo) && (cont!=num)){
        if (num % cont == 0){
            primo = false;
        }
        cont++;
    }
    return primo;
}

public static boolean esFibo(int num){
    int cont = 1, cont1, cont2 = 0;
    boolean fibo = false;
    if (num > 0){
        do {
            cont1 = cont;
            cont = cont1 + cont2;
            cont2 = cont1;
            if (cont == num){
                fibo = true;
            }
        }while(cont <= num);
    }
    return fibo;
}
```

En cada caso nos regresa un valor, en la función “esMaravilloso” nos regresa directo el string, porque los números maravillosos son los naturales, no los negativos o fraccionarios, es por eso que con un if basta.

En la función “esPrimo” nos regresa un boolean que nos sirve para ir comprobando en el ciclo while.

Y en la última función “esFibo” Vamos comprobando si nuestro número es igual al contador (el cual contiene los numeros de la serie Fibonacci) y nos regresa un boolean que lo podemos interpretar con un if para mostrar el mensaje.

Eso lo declaramos en la función onClick, la cual se creó al momento de definir el setOnClickListener en el botón.

```
@Override
public void onClick(View v) {

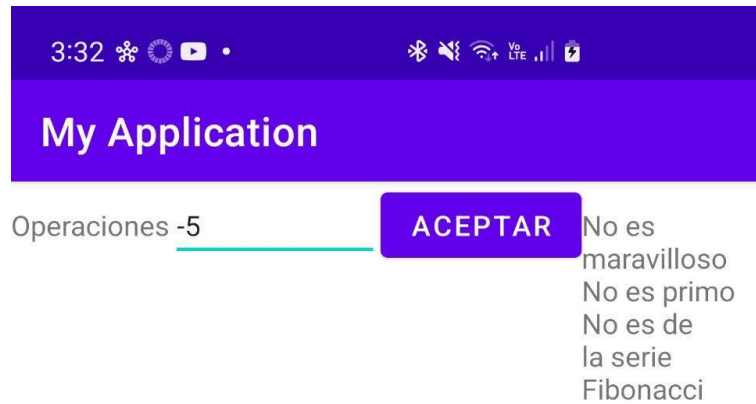
    n = Integer.parseInt(jet1.getText().toString());
    //Numero maravilloso
    jtv2.append(esMaravilloso(n));

    //El numero es primo?
    if (esPrimo(n)){
        jtv2.append("Es primo\n");
    }
    else{
        jtv2.append("No es primo\n");
    }

    //Es de la serie Fibonacci?
    if (esFibo(n)){
        jtv2.append("Es fibonacci\n");
    }
    else{
        jtv2.append("No es de la serie Fibonacci\n");
    }
    jtv2.append("\n");
}
```

Para comprobar el funcionamiento, ejecutamos en nuestro dispositivo android. Para dar un ejemplo, ingresaré un número negativo, el cual no es maravilloso, ni primo, ni pertenece a la serie Fibonacci, otro número natural primo y que no pertenezca a la serie Fibonacci y por último un número primo que si corresponda a la serie Fibonacci.

Número negativo.



Primo que no pertenece a la serie Fibonacci.

3:32

My Application

Operaciones 79

ACEPTAR

No es maravilloso

No es primo

No es de la serie Fibonacci

Es maravilloso

Es primo

No es de la serie Fibonacci

< - . ^ ? : , ...

1

2

3

4

5

6

7

8

9

0

@

#

\$

%

^

&

\*

¿

¡

1/2

-

'

"

:

;

,

?

⌫

ABC

,

< Español (US) >

.

↩

Primo que si pertenece a la serie Fibonacci.

3:33

My Application

Operaciones 89

ACEPTAR

No es  
maravilloso  
No es primo  
No es de  
la serie  
Fibonacci  
  
Es maravilloso  
Es primo  
No es de  
la serie  
Fibonacci  
  
Es maravilloso  
Es primo  
Es fibonacci

< - . ^ ? : , ...

1 2 3 4 5 6 7 8 9 0

@ # \$ % ^ & \* ¿ ¡

1/2 - ' " : ; , ?

ABC , < Español (US) > . ↵