

# PRACTICA 08.

## OBJETIVO GENERAL.

En esta práctica, mediante una función matemática, vamos a generar una cantidad de números aleatorios dependiendo de la selección del usuario.



## OBJETIVOS ESPECÍFICOS:

- Utilizar LinearLayout como contenedor principal de la aplicación.
- Crear objetos Actividad Origen:
  - dOS TextView : Utilizado para desplegar mensaje
  - uN EditText : Para introducir un Numero que indica el total de números a calcular.
  - Un Button : Utilizado para calcular números aleatorios en función del dato proporcionado por el usuario.
- Crear un evento que calcule la totalidad de los números indicados por el usuario.

## CREACION DE LA INTERFACE GRAFICA.

Usaremos el código siguiente para la interface gráfica.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.delio.app08.MainActivity">

    <TextView
        android:id="@+id/txtMensaje"
        android:text="Indica cuantos numeros aleatorios deseas "
        android:textSize="30sp"
        android:textAlignment="center"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <EditText
        android:id="@+id/edtCantidad"
        android:inputType="number"
        android:hint="0"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/btnProcesar"
        android:textSize="24sp"
        android:text="Calcular Numeros"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

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

</LinearLayout>
```

## DESARROLLO DE LA LOGICA DE APLICACIÓN.

El código inicial de nuestra actividad principal se verá:

```
public class MainActivity extends AppCompatActivity {

    private TextView txtNumerosCalculados;
    private EditText edtCantidad;
    private Button btnProcesar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        txtNumerosCalculados = (TextView)
findViewById(R.id.txtNumeroCalculados);
        edtCantidad = (EditText) findViewById(R.id.edtCantidad);
        btnProcesar = (Button) findViewById(R.id.btnProcesar);

        btnProcesar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                calcularNumeros();
            }
        });
    }

    public void calcularNumeros(){

        String miNumero = "";
        txtNumerosCalculados.setText("");

        Random random = new Random();
        random.setSeed(System.currentTimeMillis());

        Integer tope =
Integer.parseInt(edtCantidad.getText().toString());
        for(int k=1; k<=tope; k++){
            miNumero = "Mi numero es : " +
String.valueOf(random.nextInt()) + " \n";
            txtNumerosCalculados.append(miNumero);
        }
    }
}
```

Solo explicaremos la lógica del método del evento que calcula los números aleatorios. Esto se logra mediante la clase Random() y configurando el algoritmo en función de la hora de ejecución de la aplicación (línea 33).

32	Random random = new Random();
33	random.setSeed(System.currentTimeMillis());

Hacemos un ciclo haciendo como limite el valor numérico introducido por el usuario ( `inputType number` ), al mismo tiempo usamos una concatenación de textos para finalmente utilizar un `TextView` (`txtNumerosCalculados`) donde se irán agregando uno a uno mediante el método `append` (línea 47).

```
43 Integer tope = Integer.parseInt(editCantidad.getText().toString());
44 for(int k=1; k<=tope; k++){
45     miNumero = "Mi numero es : " +
46     String.valueOf(random.nextInt()) + " \n";
47     txtNumerosCalculados.append(miNumero);
48 }
```

NOTA: Las funciones aquí utilizadas son las mismas que se usan en el java normal. Usted puede generar números limitados en tamaño, por ejemplo, si solo quisiera calcular números aleatorios hasta un numero en específico, deberá de indicarlo como parámetro.