

INDICE

<u>1.1.</u>	<u>INTRODUCCIÓN</u>	<u>1</u>
<u>1.2.</u>	<u>CASO PRÁCTICO</u>	<u>1</u>
<u>1.2.1.</u>	<u>XML DO LAYOUT</u>	<u>4</u>
<u>1.2.2.</u>	<u>CÓDIGO JAVA</u>	<u>5</u>

1.1. Introducción

- A continuación, se crearán los elementos de un TableLayout en tiempo de ejecución.
- Los elementos de un TableLayout son las celdas y cada una tendrá un componente ImageView. Podría ser un botón, un texto, etc.
- Además a través del Layout ScrollView se va a permitir hacer scroll.

- **Referencias:**
- O control ScrollView: <https://developer.android.com/reference/android/widget/ScrollView.html>
- O control TableLayout: <https://developer.android.com/reference/android/widget/TableLayout.html>

1.2. Caso práctico

- Creamos un nuevo proyecto: **U2_33_TableLayoutDinamico**

TableLayout Dinámico



Al iniciar la aplicación introducimos el número de filas y columnas que deseamos que tenga el TableLayout que está dispuesto debajo del botón.



Generamos la tabla y se carga con ImageViews cuya imagen es el icono de la aplicación.



Si hacemos clic en una imagen.



Aparece un Toast mostrando su etiqueta indicando la fila y columna en la que se encuentra esa imagen.

1.2.1. XML do Layout

➤ Nos centraremos en lo novedoso.

```
<?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">

    <TableLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:stretchColumns="*" >

        <TableRow>

            <EditText
                android:id="@+id/et_filas"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:hint="Filas"
                android:gravity="center"
                android:imeOptions="actionDone"
                android:inputType="number" />

            <EditText
                android:id="@+id/et_columnas"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="center"
                android:hint="Columnas"
                android:inputType="number" />
        </TableRow>
    </TableLayout>

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="onGenerarTablaClick"
        android:text="Generar tabla" />

    <ScrollView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >

        <TableLayout
            android:id="@+id/taboa_dinamica"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:stretchColumns="*" >
        </TableLayout>
    </ScrollView>
</LinearLayout>
```

- **ScrollView**: es un layout que permite activar el scroll vertical. Con lo cual, los elementos que haya dentro pueden crecer en vertical a su antojo, pues siempre se podrá realizar scroll.
 - ✓ Dentro de este Layout se definió una TableLayout que no tenga ninguna fila. Estas se crearán en tiempo de ejecución.
- **Línea resaltada**: observar que la tabla tiene un ID para poder acceder a ella desde Java.

1.2.2. Código Java

```
package olga100.proyectosandroid.pmdm_2020_2021.U2_33_TableLayoutDinamico;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.Toast;

import olga100.proyectosandroid.pmdm_2020_2021.R;

public class U2_33_TableLayoutDinamico extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_u2_33_table_layout_dinamico);
    }

    public void onGenerarTablaClick(View v) {
        TableLayout tabla = (TableLayout) findViewById(R.id.taboa_dinamica);
        EditText filas = (EditText) findViewById(R.id.et_filas);
        EditText columnas = (EditText) findViewById(R.id.et_columnas);

        int f = Integer.parseInt(filas.getText().toString());
        int c = Integer.parseInt(columnas.getText().toString());

        tabla.removeAllViews();

        for (int i = 1; i <= f; i++) {
            TableRow fila = new TableRow(this);
            tabla.addView(fila);

            for (int j = 1; j <= c; j++) {
                ImageView imagen = new ImageView(this);
                imagen.setTag("Tag" + i + j);
                imagen.setImageResource(R.drawable.icon_ok);
                fila.addView(imagen);
                imagen.setOnClickListener(new View.OnClickListener() {

                    @Override
                    public void onClick(View v) {
                        // TODO Auto-generated method stub
                        ImageView img = (ImageView) v;
                        Toast.makeText(getApplicationContext(), "Pulsaste en la imagen " +
img.getTag(), Toast.LENGTH_SHORT).show();
                    }
                });
            }
        }
    }
}
```

```
        });  
    }  
}  
}
```

- **Línea 25:** en el objeto tabla apuntamos a la tabla definida en el layout.
- **Línea 29,30:** f e c indican cuantas filas y columnas debe tener el TableLayout. Recordar que parseInt() es un método estático de la clase Integer en Java.
- **Línea 32:** eliminamos todas las vistas que pueda tener la tabla.
- **Línea 35,36:** añadimos tantos TableRows como filas nos indicó el usuario.
- **Línea 39,40,41:** creamos un objeto ImageView con su Tag e imagen asociada.
- **Línea 42:** añadimos en cada fila tantas columnas como nos indicó el usuario a través de añadir, en ese caso vistas del tipo ImageView.
- **Línea 43 a 51** para cada una das imágenes chamamos a su Listener que estará a espera de que se haga click en esa misma imagen que se acaba de crear.
- **Línea 46 a 50** cuando se pulsa una imagen podemos conocer, en ese caso, cuál es su Tag.