

1 Xerar Views nun Layout en tempo de execución. Layout ScrollView

1.1 Sumario

- 1 Introducción
- 2 Caso práctico
 - ◆ 2.1 O XML do layout
 - ◆ 2.2 O código Java

1.2 Introducción

- A continuación vanse crear os elementos dun `TableLayout` en tempo de execución.
- Os elementos dun `TableLayout` son as celas e cada unha vai ter un compoñente **ImageView**. Podería ser un botón, un texto, etc.
- Ademais a través do Layout **ScrollView** vaise permitir facer scroll.

- Referencias
 - ◆ O control `ScrollView`: <http://developer.android.com/reference/android/widget/ScrollView.html>
 - ◆ O control `TableLayout`: <http://developer.android.com/reference/android/widget/TableLayout.html>

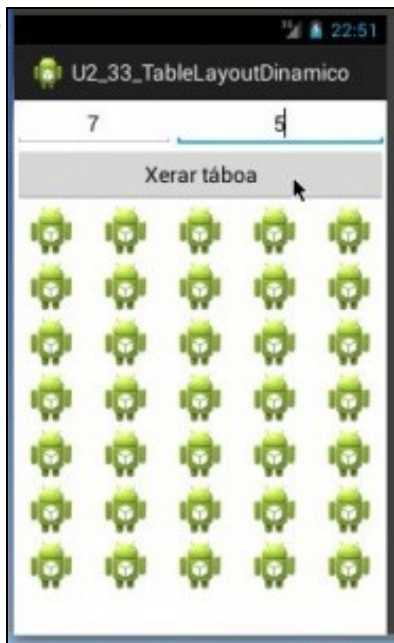
1.3 Caso práctico

- Creamos unha novo proxecto: **U2_33_TableLayoutDinamico**

- `TableLayout` Dinámico



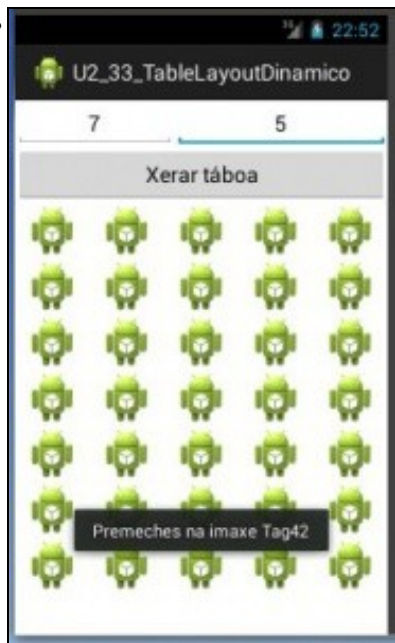
Ao iniciar a aplicación introducimos o número de filas e columnas que desexamos que teña o `TableLayout` que está disposto debaixo do botón.



Xeramos a táboa, e esta cárgase con ImageViews cuxa imaxe é a icona da aplicación.



Se prememos sobre unha imaxe.



Aparece un Toast amosando o seu Tag que indica a fila e columna na que está esa imaxe.

1.3.1 O XML do layout

- Imos centrarnos no novidoso:

```
<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"
    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="onXerarTaboaClick"
        android:text="Xerar táboa" />

    <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**: é un layout que permite activar o scroll vertical. Co cal, os elementos que haxa dentro poden crecer en vertical ao seu antollo, pois sempre se poderá realizar scroll.
 - ◆ Dentro dese Layout definiuse un `TableLayout` que non ten ningunha fila. Estas crearanse en tempo de execucion.
- **Liña 44**: observar que a táboa ten un ID para poder acceder a ela dende Java.

1.3.2 O código Java

```

package com.example.u2_33_tablelayoutdinamico;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.WindowManager;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.Toast;

public class U2_33_TableLayoutDinamico extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_u2_33__table_layout_dinamico);
    }

    public void onXerarTaboaClick(View v) {
        TableLayout taboa = (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());

        taboa.removeAllViews();

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

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

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

```

```
}  
}  
  
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    // Inflate the menu; this adds items to the action bar if it is present.  
    getMenuInflater().inflate(R.menu.u2_33__table_layout_dinamico, menu);  
    return true;  
}  
  
}
```

- **Liña 25:** no obxecto **taboa** apuntamos á táboa definida no layout.
- **Liña 29,30:** f e c indican cantas filas e columnas debe ter o `TableLayout`. Lembrar que `parseInt()` é un método estático da clase `Integer` en Java.
- **Liña 32:** eliminamos todas as vistas que poida ter a táboa.
- **Liña 35,36:** engadimos tantos `TableRows` como filas nos indicou o usuario.
- **Liña 39,40,41:** creamos un obxecto `ImageView` co seu `Tag` e imaxe asociada.
- **Liña 42:** engadimos en cada fila tantas columnas como nos indicou o usuario a través de engadir, nese caso vistas do tipo `ImageView`.
- **Liña 43 a 51** para cada unha das imaxes chamamos ao seu `Listener` que estará a espera de que se faga click nesa mesma imaxe que se acaba de crear.
- **Liña 46 a 50** cando se preme nunha imaxe podemos coñecer, neste caso, cal é o seu `Tag`.