

PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES

U.T. 3. INTERFAZ DE USUARIO

PRÁCTICA 3.5 y 3.6:

ANDROID STUDIO

AUTOR: Vincenzo Rojas Carrera

CICLO FORMATIVO DE GRADO SUPERIOR DE DESARROLLO DE APLICACIONES MULTIPLATAFORMA

[Práctica 3.5: Botones de radio 3](#_jga0uahsso6)

[Crear el proyecto 3](#_cdifpjko1g06)

[Diseñar el layout (activity\_main.xml) 3](#_geaaa568n0bo)

[Añadir cadenas en strings.xml 4](#_ef3eorw1bee2)

[Código en MainActivity.java 4](#_i2s9ndhbi1kj)

[Sin boton: 9](#_3v38pls600oh)

[Declaraciones 9](#_fl65jnrn515b)

[onCreate() 9](#_aeydyfat1klg)

[Método actualizarTexto() 10](#_o20o6ou4qlcp)

[Práctica 3.6: TableLayout y Switch 11](#_k907b4h8k2ba)

[Crear el proyecto 11](#_3gmufj291c7q)

[Diseñar la interfaz (layout) 11](#_w1ey6zgeidek)

[Configurar filas y botones 11](#_qrv2j1kg4soc)

[Añadir los Switch y TextView 11](#_1n4mv563jvxn)

[Añadir cadenas en res/values/strings.xml 13](#_y2d9nqdyeko8)

[Programar la lógica en MainActivity.java 13](#_i4czj4shihfc)

[Añadir botón de Reset 16](#_5mz6acz8lrim)

# Práctica 3.5: Botones de radio

### **Crear el proyecto**

* **Nombre de la app:** AppRadioGroup
* **Dominio:** VincenzoRojas.appradiogroup
* **Min SDK:** la recomendada por Android Studio
* **Empty View Activity**

### **Diseñar el layout (activity\_main.xml)**

En la **paleta de componentes**, busca **Buttons → RadioGroup**. Arrástralo al diseño.

En sus propiedades, cambia el id a: android:id="@+id/radioGroup1" Dentro del RadioGroup, arrastra **tres RadioButton**. Asigna los id:  
  
android:id="@+id/radio1"

android:id="@+id/radio2"

android:id="@+id/radio3"

En el campo **text**, usa cadenas desde strings.xml (las crearemos en el siguiente paso):  
  
android:text="@string/radioAndroid"

android:text="@string/radioIOS"

android:text="@string/radioWindows"

Añade:  
Un **TextView** arriba para el texto de la pregunta (ej. “¿Qué sistema operativo móvil prefieres?”).

Un **Button** (texto **Aceptar)** (android:onClick="botonAceptar").

Otro **TextView** abajo (oculto al inicio):  
  
 android:id="@+id/tOculto"

android:visibility="gone"

### 

### **Añadir cadenas en strings.xml**

<resources>

<string name="app\_name">AppRadioGroup</string>

<string name="tPregunta">¿Qué sistema operativo móvil prefieres?</string>

<string name="aceptar">Aceptar</string>

<string name="radioAndroid">Android</string>

<string name="radioIOS">iOS</string>

<string name="radioWindows">Windows Phone</string>

<string name="RespAndroid">Es una plataforma de código abierto.</string>

<string name="RespIOS">Es un sistema exclusivo de Apple.</string>

<string name="RespWindows">Fue descontinuado por Microsoft.</string>

</resources>

### **Código en MainActivity.java**

package VincenzoRojas.appradiogroup;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.RadioGroup;

import android.widget.RadioButton;

import android.widget.CheckBox;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

RadioGroup botones;

CheckBox ch1, ch2, ch3;

TextView tx;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

botones = findViewById(R.id.radioGroup1);

tx = findViewById(R.id.tOculto);

}

public void botonAceptar(View v) {

tx.setVisibility(View.VISIBLE);

tx.setText("");

// RadioButtons

if (botones.getCheckedRadioButtonId() == R.id.radio1) {

tx.setText(getString(R.string.RespAndroid));

} else if (botones.getCheckedRadioButtonId() == R.id.radio2) {

tx.setText(getString(R.string.RespIOS));

} else if (botones.getCheckedRadioButtonId() == R.id.radio3) {

tx.setText(getString(R.string.RespWindows));

}

}

**(Alternativa) Sin botón Aceptar**

Si prefieres que el mensaje aparezca al cambiar la selección del RadioGroup en onCreate:

RadioGroup rg = findViewById([R.id](http://r.id/).radioGroup1);

rg.setOnCheckedChangeListener((group, checkedId) -> {

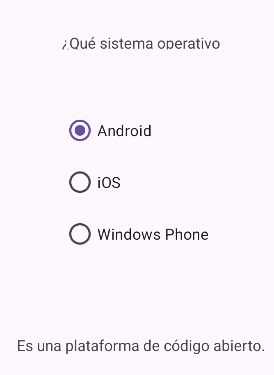
tx.setVisibility(View.VISIBLE);

if (checkedId == R.id.radio1) tx.setText(R.string.RespAndroid);

else if (checkedId == R.id.radio2) tx.setText(R.string.RespIOS);

else if (checkedId == R.id.radio3) tx.setText(R.string.RespWindows);

});



Borrar el botón de activity\_main.xml, ya que no se usara

Vamos a añadir **los CheckBox a tu app de RadioGroup.** Dentro del mismo LinearLayout donde tienes tu RadioGroup, añade tres CheckBox:

<CheckBox

android:id="@+id/ch1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/camara"/>

<CheckBox

android:id="@+id/ch2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/bluetooth"/>

<CheckBox

android:id="@+id/ch3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/sonidoDolby"/>

Añade los textos que se mostrarán si se marca cada checkbox:

<string name="camara">Cámara</string>

<string name="bluetooth">Bluetooth</string>

<string name="sonidoDolby">Sonido Dolby</string>

<string name="RespCamara">Cámara seleccionada</string>

<string name="RespBluetooth">Bluetooth seleccionado</string>

<string name="RespSonidoDolby">Sonido Dolby seleccionado</string>

* Los Resp\* son los mensajes que se añadirán al TextView cuando se pulse “Aceptar”, cuando hay boton.

Declarar los CheckBox en MainActivity.java

Dentro de la clase:

CheckBox ch1, ch2, ch3;

Y en onCreate() vincúlalos:

ch1 = findViewById(R.id.ch1);

ch2 = findViewById(R.id.ch2);

ch3 = findViewById(R.id.ch3);

## 

Añade el código para que **se muestre el texto correspondiente si están marcados**:

public void botonAceptar(View v) {

tx.setVisibility(View.VISIBLE);

tx.setText(""); // limpiamos antes de añadir

// RadioButtons

if (botones.getCheckedRadioButtonId() == R.id.radio1) {

tx.setText(getString(R.string.RespAndroid));

} else if (botones.getCheckedRadioButtonId() == R.id.radio2) {

tx.setText(getString(R.string.RespIOS));

} else if (botones.getCheckedRadioButtonId() == R.id.radio3) {

tx.setText(getString(R.string.RespWindows));

}

// CheckBoxes

if (ch1.isChecked()) tx.append("\n" + getString(R.string.RespCamara));

if (ch2.isChecked()) tx.append("\n" + getString(R.string.RespBluetooth));

if (ch3.isChecked()) tx.append("\n" + getString(R.string.RespDolby));

}

Esto permite que el usuario pueda marcar **uno, dos o los tres** checkboxes, y los textos correspondientes se añadirán al TextView.

## 

## Sin boton:

MainActivity.java

### **Declaraciones**

RadioGroup botones;

CheckBox ch1, ch2, ch3;

TextView tx;

### **onCreate()**

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

botones = findViewById(R.id.radioGroup1);

ch1 = findViewById(R.id.ch1);

ch2 = findViewById(R.id.ch2);

ch3 = findViewById(R.id.ch3);

tx = findViewById(R.id.tOculto);

// Listener para RadioGroup

botones.setOnCheckedChangeListener((group, checkedId) -> actualizarTexto());

// Listeners para CheckBoxes

ch1.setOnCheckedChangeListener((buttonView, isChecked) -> actualizarTexto());

ch2.setOnCheckedChangeListener((buttonView, isChecked) -> actualizarTexto());

ch3.setOnCheckedChangeListener((buttonView, isChecked) -> actualizarTexto());

}

### 

### **Método actualizarTexto()**

private void actualizarTexto() {

tx.setVisibility(View.VISIBLE);

tx.setText(""); // Limpiamos antes de añadir

// RadioButtons

int checkedId = botones.getCheckedRadioButtonId();

if (checkedId == R.id.radio1) tx.setText(getString(R.string.RespAndroid));

else if (checkedId == R.id.radio2) tx.setText(getString(R.string.RespIOS));

else if (checkedId == R.id.radio3) tx.setText(getString(R.string.RespWindows));

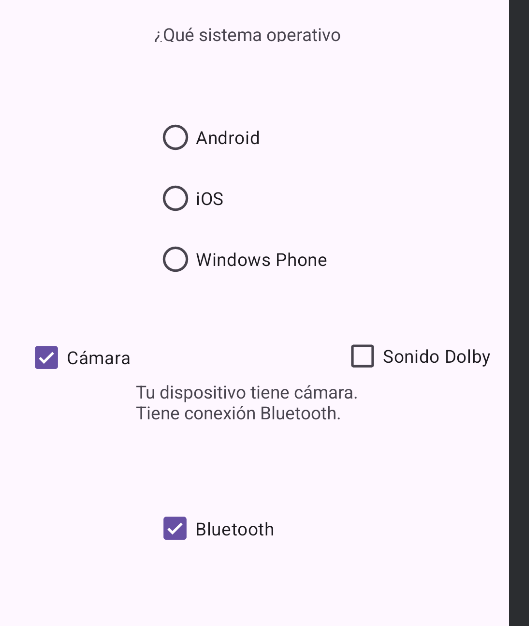
// CheckBoxes

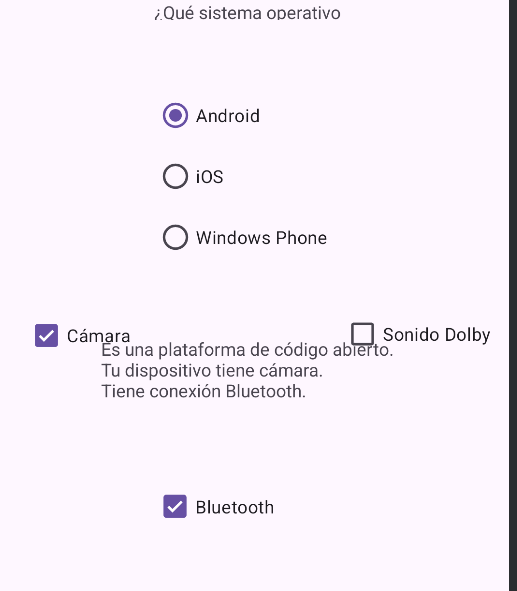
if (ch1.isChecked()) tx.append("\n" + getString(R.string.RespCamara));

if (ch2.isChecked()) tx.append("\n" + getString(R.string.RespBluetooth));

if (ch3.isChecked()) tx.append("\n" + getString(R.string.RespDolby));

}





# Práctica 3.6: TableLayout y Switch

## **Crear el proyecto**

1. Abre **Android Studio** → **New Project**
2. Plantilla: **Empty Views Activity**
3. Nombre: **AppTableLayout**
4. Package name: VincenzoRojas.nombre
5. Finaliza y espera que se genere el proyecto.

## **Diseñar la interfaz (layout)**

1. Abre el archivo **activity\_main.xml**.
2. En la **paleta → Layouts**, arrastra un **TableLayout** al lienzo.

## **Configurar filas y botones**

1. En el **Component Tree**, deja solo **3 TableRow**.
2. En la primera fila, arrastra 3 botones. Ids: b1, b2, b3
3. En la segunda fila, agrega 3 botones. Ids: b4, b5, b6
4. En la tercera fila, agrega. Ids: b7, b8, b9
5. En **cada TableRow**, cambia:
   * android:layout\_weight = "1"
6. Selecciona **todos los botones (Ctrl + clic)** y cambia:
   * android:layout\_height = "match\_parent"
   * android:layout\_weight = "1"
   * android:layout\_margin = "10dp"
   * android:textSize = "14sp"
   * android:text="" → déjalo vacío

## **Añadir los Switch y TextView**

Encima del TableLayout, agrega:

<TextView

android:id="@+id/textoTitulo"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/titulo\_juego"

android:textSize="24sp"

android:layout\_marginBottom="16dp"

android:layout\_gravity="center\_horizontal" />

<Switch

android:id="@+id/switchX"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/juega\_x"

android:checked="true" />

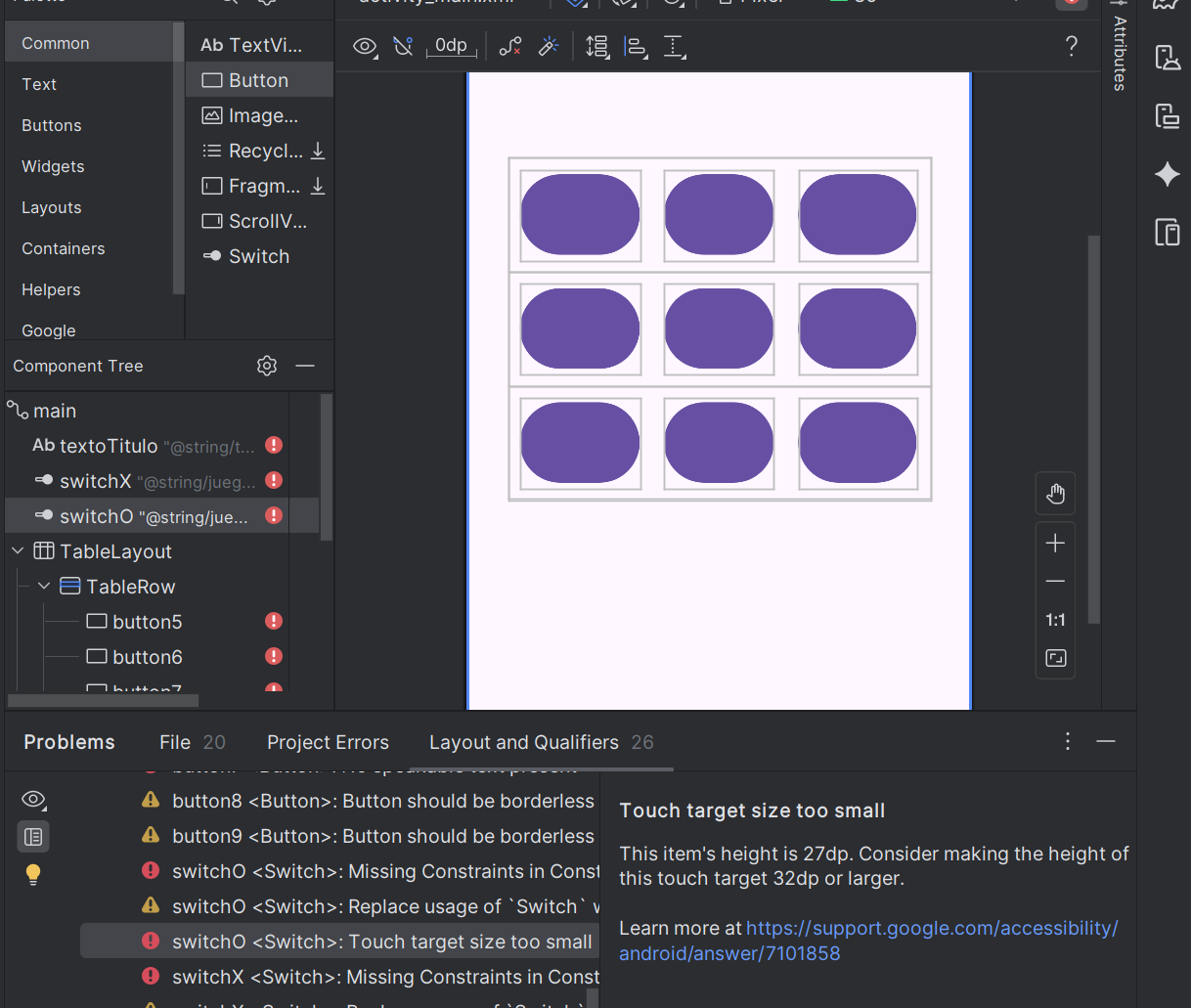
<Switch

android:id="@+id/switchO"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/juega\_o" />



## 

## **Añadir cadenas en res/values/strings.xml**

Abre **strings.xml** y añade:

<string name="titulo\_juego">Juego de las Tres en Raya</string>

<string name="juega\_x">Juega X</string>

<string name="juega\_o">Juega O</string>

<string name="gana\_x">Ha ganado X</string>

<string name="gana\_o">Ha ganado O</string>

<string name="empate">Empate</string>

<string name="reset">Reiniciar</string>

## **Programar la lógica en MainActivity.java**

Abre **MainActivity.java** y escribe:

package VincenzoRojas.apptablelayout;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.Switch;

import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

Button[] botones = new Button[9];

Switch switchX, switchO;

TextView textoTitulo;

char jugador = 'X';

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

botones[0] = findViewById(R.id.b1);

botones[1] = findViewById(R.id.b2);

botones[2] = findViewById(R.id.b3);

botones[3] = findViewById(R.id.b4);

botones[4] = findViewById(R.id.b5);

botones[5] = findViewById(R.id.b6);

botones[6] = findViewById(R.id.b7);

botones[7] = findViewById(R.id.b8);

botones[8] = findViewById(R.id.b9);

switchX = findViewById(R.id.switchX);

switchO = findViewById(R.id.switchO);

textoTitulo = findViewById(R.id.textoTitulo);

View.OnClickListener listener = v -> {

Button b = (Button) v;

if (b.getText().toString().equals("")) {

b.setText(String.valueOf(jugador));

comprobarJugada();

cambioJugador();

}

};

for (Button b : botones) {

b.setOnClickListener(listener);

}

}

private void cambioJugador() {

jugador = (jugador == 'X') ? 'O' : 'X';

switchX.setChecked(jugador == 'X');

switchO.setChecked(jugador == 'O');

}

private void comprobarJugada() {

String[][] casillas = new String[3][3];

for (int i = 0; i < 9; i++) {

casillas[i / 3][i % 3] = botones[i].getText().toString();

}

for (int i = 0; i < 3; i++) {

if (!casillas[i][0].equals("") && casillas[i][0].equals(casillas[i][1]) && casillas[i][0].equals(casillas[i][2])) {

gana(casillas[i][0]);

return;

}

if (!casillas[0][i].equals("") && casillas[0][i].equals(casillas[1][i]) && casillas[0][i].equals(casillas[2][i])) {

gana(casillas[0][i]);

return;

}

}

if (!casillas[0][0].equals("") && casillas[0][0].equals(casillas[1][1]) && casillas[0][0].equals(casillas[2][2])) {

gana(casillas[0][0]);

return;

}

if (!casillas[0][2].equals("") && casillas[0][2].equals(casillas[1][1]) && casillas[0][2].equals(casillas[2][0])) {

gana(casillas[0][2]);

}

}

private void gana(String ganador) {

textoTitulo.setText("Ha ganado " + ganador);

for (Button b : botones) {

b.setEnabled(false);

}

}

}

## 

## **Añadir botón de Reset**

En tu layout:

<Button

android:id="@+id/reset"

android:text="@string/reset"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

Y en MainActivity.java:

Button reset = findViewById(R.id.reset);

reset.setOnClickListener(v -> {

for (Button b : botones) {

b.setText("");

b.setEnabled(true);

}

textoTitulo.setText(R.string.titulo\_juego);

jugador = 'X';

switchX.setChecked(true);

switchO.setChecked(false);

});

