

UD 05

PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES 24/25 CFGS DAM

PRÁCTICA 1 RADIOBUTTONS EN ANDROID

Autor: Mara Vañó

m.vanoalonso@edu.gva.es

Fecha: 2024/2025

Licencia Creative Commons

versión 4.0

Reconocimiento – NoComercial – CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Índice

Objetivos de la práctica 1

Introducción 1

Enunciado 1

Layout 2

Activity 4

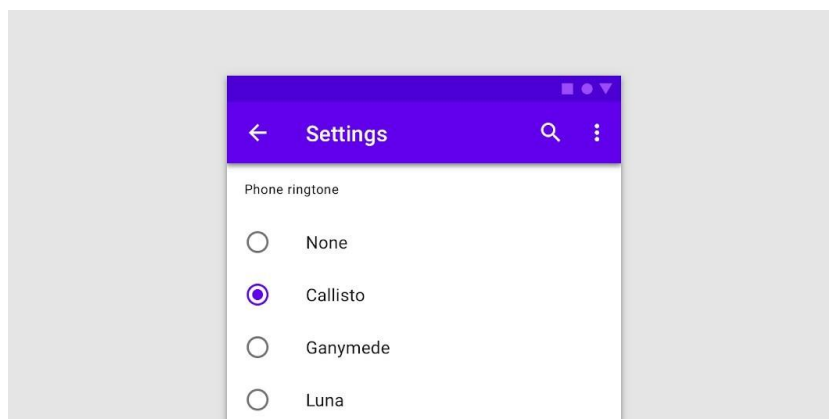
Recursos..... 5

1. Objetivos de la práctica

- Implementar varios RadioButtons y un Radio Group
- Comprender los parámetros del código dentro del activity
- Generar un layout
- Aprender a seleccionar un elemento de todos los elementos disponibles.

2. Introducción

Los RadioButtons son las clases que se utilizan para permitir al usuario hacer selecciones de elementos. La clase RadioGroup es en realidad un contenedor (un conjunto de RadioButton), y su función es agrupar varios RadioButtons y permitir una selección de único elemento. Cuando los agrupas, el sistema garantiza que solo se pueda seleccionar un botón a la vez.



Vamos a realizar un ejemplo donde implementaremos un RadioButton para seleccionar un tono de llamada como aparece en la imagen.

3. Enunciado

Realiza una selección de RadioButtons donde puedas elegir entre varios elementos. Imprime por pantalla el número de la selección y comprueba su funcionamiento en el apartado Run.

Ejemplo:

RadioButton → Ninguno

RadioButton1 → Calisto

RadioButton2 → Ganymede

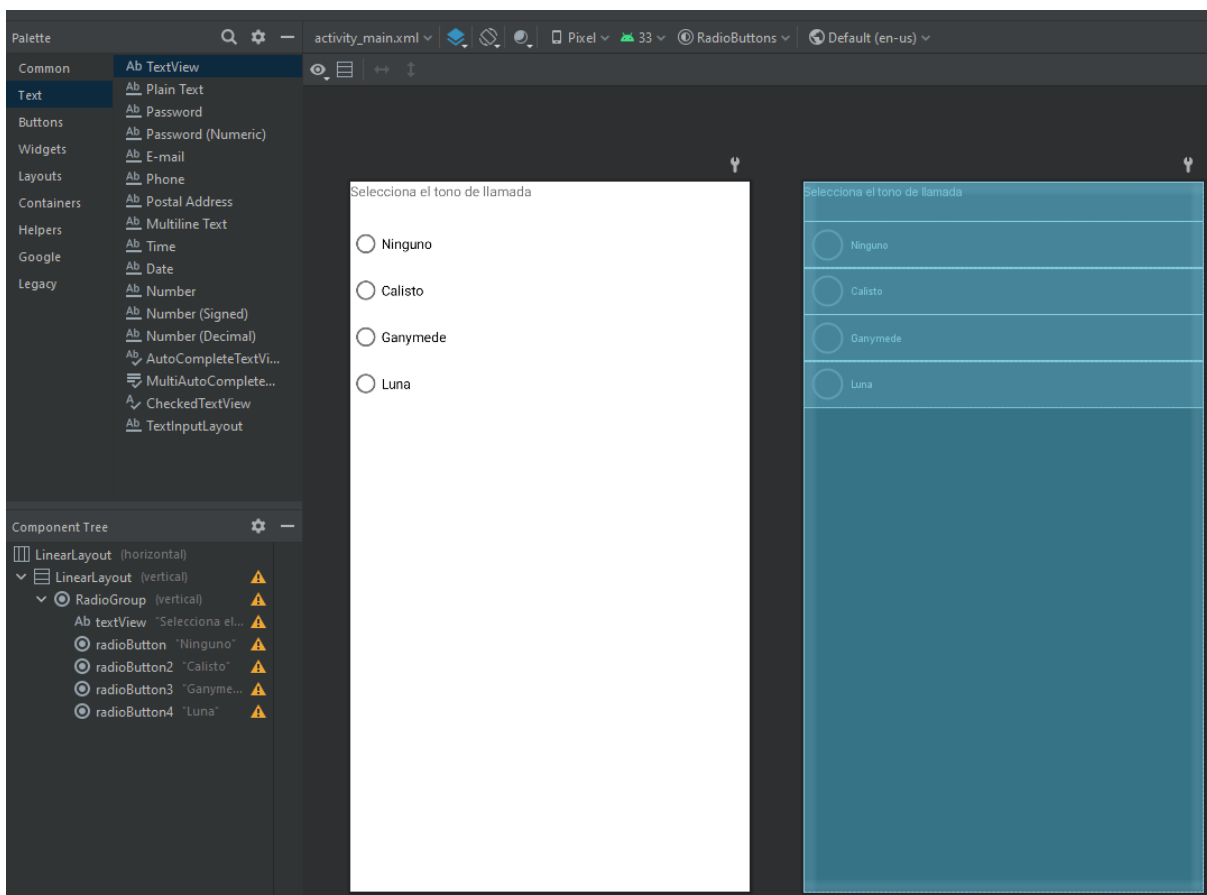
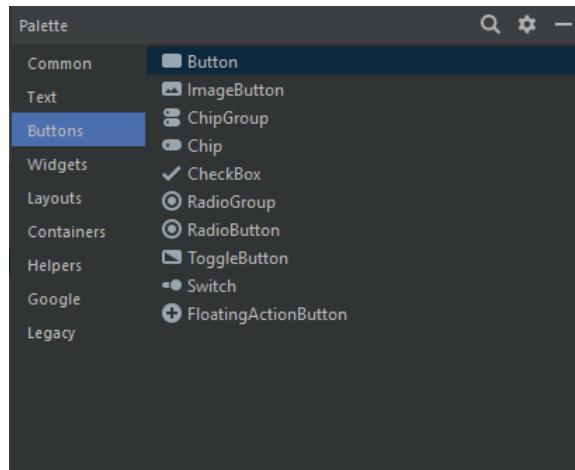
RadioButton3 → Luna

...

4. Layout

Para implementar un `RadioButton` es recomendable utilizar un layout de tipo linear para que la estructura que ordenada pero no es obligatorio. En el caso de seleccionar un `RadioGroup` esta ordenación es automática.

En el apartado de layout -> Buttons debemos seleccionar `RadioGroup` en primer lugar y después añadir las diferentes `RadioButtons` dentro para cada uno de las opciones.



El layout xml quedaría de la siguiente manera:

```
<?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"
    app:layoutDescription="@xml/activity_main_scene"
    tools:context=".MainActivity" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <RadioGroup
            android:layout_width="match_parent"
            android:layout_height="match_parent" >

            <TextView
                android:id="@+id/textView"
                android:layout_width="match_parent"
                android:layout_height="40dp"
                android:text="Selecciona el tono de llamada" />

            <RadioButton
                android:id="@+id/radioButton"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Ninguno" />

            <RadioButton
                android:id="@+id/radioButton2"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Calisto" />

            <RadioButton
                android:id="@+id/radioButton3"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Ganymede" />

            <RadioButton
                android:id="@+id/radioButton4"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Luna" />
        </RadioGroup>
    </LinearLayout>
</LinearLayout>
```

5. Activity

En el activity debemos realizar los siguientes pasos:

1. Importar el widget `RadioGroup` y `RadioButton` al inicio del código.
2. Definir las variables tanto de los `RadioButtons` como el del `RadioGroup` con la instrucción `var radio1:RadioButton? = null;`
3. Utilizar la instrucción `findViewById` para definir el valor de cada variable dentro del estado en el que estemos en este ejemplo es en el `onCreate`.
4. Crear una función `onCheckedChanged` con un `when` para definir el funcionamiento de cada botón.

*En este caso se ha puesto de manera abierta que imprima por pantalla un número para no entrar en detalle en código que no corresponde a la práctica, pero se puede añadir todas las funcionalidades que necesitemos siempre dentro de los corchetes.

```
package com.carlos.radiobutton

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.RadioButton
import android.widget.RadioGroup

var radio1:RadioButton? = null
var radio2:RadioButton? = null
var radio4:RadioButton? = null
var radio3:RadioButton? = null
var radioGroup:RadioGroup? = null

class MainActivity : AppCompatActivity(), RadioGroup.OnCheckedChangeListener {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        radioGroup = findViewById(R.id.radioGroup)
        radio1 = findViewById(R.id.radioButton)
        radio2 = findViewById(R.id.radioButton1)
        radio3 = findViewById(R.id.radioButton2)
        radio4 = findViewById(R.id.radioButton3)
        radioGroup?.setOnCheckedChangeListener(this)
    }

    override fun onCheckedChanged(p0: RadioGroup?, idRadio: Int) {
        when (idRadio) {
            R.id.radioButton -> println("1")
            R.id.radioButton2 -> println("2")
            R.id.radioButton3 -> println("3")
            R.id.radioButton4 -> println("4")
        }
    }
}
```

6. Recursos

Developers Android:

<https://developer.android.com/guide/topics/ui/controls/radiobutton?hl=es-419>

The screenshot shows the Android Developers website in Spanish. The main heading is 'Botones de selección'. Below it, there is a paragraph explaining that selection buttons allow users to choose one option from a group, and that they should be used for mutually exclusive options. A visual example shows a dialog box titled 'ATTENDING?' with three radio buttons: 'Yes' (selected), 'Maybe', and 'No'. Below the example, the text states that each option should be created as a `RadioButton` widget, and they should be grouped within a `RadioGroup`. It also lists the key classes: `RadioButton` and `RadioGroup`. The section 'Cómo responder a eventos de clic' (How to respond to click events) explains that when a user selects a button, the corresponding `RadioButton` object receives a click event, and the controller should define the `onClick` event listener for the button.

Desarrolladores de Android > Documentos > Guías

Botones de selección

Los botones de selección permiten al usuario seleccionar una opción de un conjunto. Debes usar botones de selección para conjuntos opcionales que son mutuamente excluyentes si crees que el usuario necesita ver todas las opciones disponibles una al lado de la otra. Si no es necesario mostrar todas las opciones juntas, usa un [icono giratorio](#) en su lugar.

ATTENDING?

☒ Yes ☐ Maybe ☐ No

Para crear cada opción del botón de selección, crea un `RadioButton` en tu diseño. Sin embargo, debido a que los botones de selección son mutuamente excluyentes, debes agruparlos dentro de un `RadioGroup`. Cuando los agrupas, el sistema garantiza que solo se pueda seleccionar un botón a la vez.

Las clases clave son las siguientes:

- `RadioButton`
- `RadioGroup`

Cómo responder a eventos de clic

Cuando el usuario selecciona uno de los botones de selección, el objeto `RadioButton` correspondiente recibe un evento de clic.

Para definir el controlador de eventos de clic correspondiente a un botón, agrega el atributo `android:onClick` al

En esta página
Cómo responder a eventos de clic