

TextView: RadioButton #1

November 27, 2020

Introducción

El RadioButton es un control de selección para elegir una opción de entre un conjunto. Se trata de un control de exclusión mutua, es decir, la selección de uno implica el cambio automático del que previamente estaba seleccionado. Pueden agruparse mediante RadioGroup, de tal manera que, de los que están incluidos en este grupo, solo uno de ellos puede y debe estar marcado.

1 Diseño del elemento

Se crea el radiobutton desde el diseño o a través del código. Se crea primero el RadioGroup para ir luego añadiendo los RadioButton (los elementos).

```
1 <RadioGroup
2     android:id="@+id/radioGrupo "
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     android:layout_marginTop="24dp"
6     android:orientation="vertical"
7     app:layout_constraintEnd_toEndOf="parent"
8     app:layout_constraintStart_toStartOf="parent"
9     app:layout_constraintTop_toBottomOf="@+id/textView">
10
11     <RadioButton
12         android:id="@+id/opcion1 "
13         android:layout_width="match_parent "
14         android:layout_height="wrap_content"
15         android:text="@string/mensaje_op1 "
16         android:onClick="ComprobarOpcion "
17         android:checked="true"/>
18
19     <RadioButton
20         android:id="@+id/opcion2 "
21         android:layout_width="match_parent "
22         android:layout_height="wrap_content"
```

```

23         android:onClick="ComprobarOpcion"
24         android:text="@string/mensaje_op2" />
25
26     <RadioButton
27         android:id="@+id/opcion3"
28         android:layout_width="match_parent"
29         android:layout_height="wrap_content"
30         android:onClick="ComprobarOpcion"
31         android:text="@string/mensaje_op3" />
32 </RadioGroup>

```

Algorithm 1: RadioButton

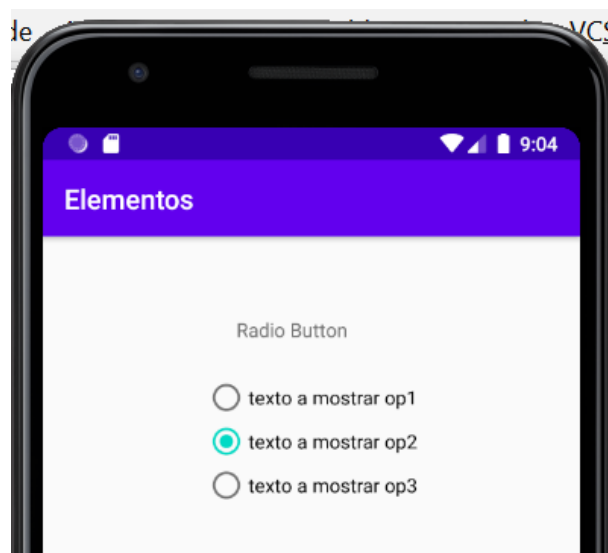
Se ha añadido los valores para los string "mensaje_op1",...

```

strings.xml
1
2 <resources>
3     <string name="app_name">Elementos</string>
4
5     <string name="mensaje_op1">La opcion elegida es la 1 </string>
6     <string name="mensaje_op2">La opcion elegida es la 2 </string>
7     <string name="mensaje_op3">La opcion elegida es la 3 </string>
8 </resources>

```

El resultado al ejecutar el proyecto es:



2 Atributos interesantes

El atributo a destacar es **android:orientation** que afecta a todos los elementos del RadioButton, va a servir para poner los elementos de RadioGroup en vertical o horizontal.

El atributo **android:checked** para determinar el estado de cada radiobutton, con el valor *true* o *false* hace que esa opción este activada o no.

Otra de los atributos que tenemos que utilizar es **android:onClick**, a este atributo le vamos a asociar un método java para programar lo que tiene que realizar cuando se selecciona este item.

3 Lógica del RadioButton

En este apartado vamos a explicar el funcionamiento del elemento, para ello vamos a trabajar con el fichero *ActivityMain*.

Esto puedes implementarlo de muchas maneras distintas pero la base es la siguiente. Primero podemos comprobar si el botón está checkeado/marcado, y luego, con un case ver qué ocurre dependiendo de que marque una cosa u otra.

3.1 Obtener el valor del RadioButton

La obtención del estado de un radio button se realiza con el método **isChecked()** de la superclase **CompoundButton**. El retorno es booleano, donde *true* representa el estado *On* y *false* el estado *Off*.

A continuación se pone el código del MainActivity para mostrar el mensaje que tiene el string de la opción seleccionada.

```
1
2 public class MainActivity extends AppCompatActivity {
3
4     public Toast toast1;
5
6     @Override
7     protected void onCreate(Bundle savedInstanceState) {
8         super.onCreate(savedInstanceState);
9         setContentView(R.layout.activity_main);
10    }
11
12    public void ComprobarOpcion(View view) {
13        //
14        boolean estado = ((RadioButton) view).isChecked();
15
16        // Comprobamos si la opcion esta marcada
17        switch(view.getId()) {
18            case R.id.opcion1:
19                if (estado)
```

```

20         displayToast(getString(R.string.mensaje_op1));
21         //mensaje de la opcion1
22         break;
23     case R.id.opcion2:
24         if (estado)
25             displayToast(getString(R.string.mensaje_op2));
26             break;
27     case R.id.opcion3:
28         if (estado)
29             displayToast(getString(R.string.mensaje_op3));
30             break;
31     default:
32         // Si hay seleccionado
33         break;
34 }
35 }
36
37
38 //metodo para mostrar el mensaje
39
40 public void displayToast(String mensaje) {
41     Toast toast1= Toast.makeText(getApplicationContext() , mensaje , Toast.LENGTH_SHORT);
42     toast1.show();
43 }

```

Algorithm 2: RadioButton

4 Métodos y clases utilizadas

La clase que estamos utilizando es RadioButton que hereda de CompoundButton.

Api: <https://developer.android.com/reference/android/widget/RadioButton?hl=en>

Lo primero que realizamos es convertir la clase view a clase RadioButton para poder utilizar el método **isChecked()**. El resultado de este método es true o false si la opción esta marcada.

```

1
2 // se puede poner asi
3 RadioButton radio=(RadioButton) view;
4 boolean estado=radio.isChecked();
5 // o así
6 //boolean estado = ((RadioButton) view).isChecked();

```

El método **getId()** de la clase View permite saber el identificador asignado.

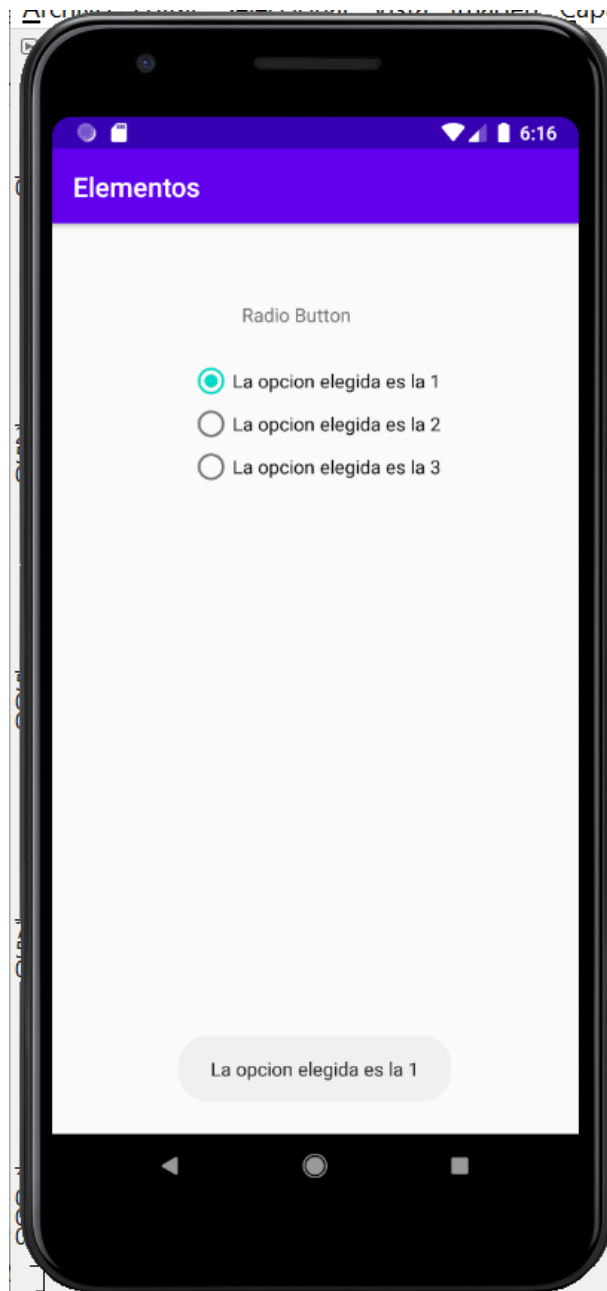
```

1
2 switch (view.getId())

```

5 Resultado

El resultado final es el siguiente:

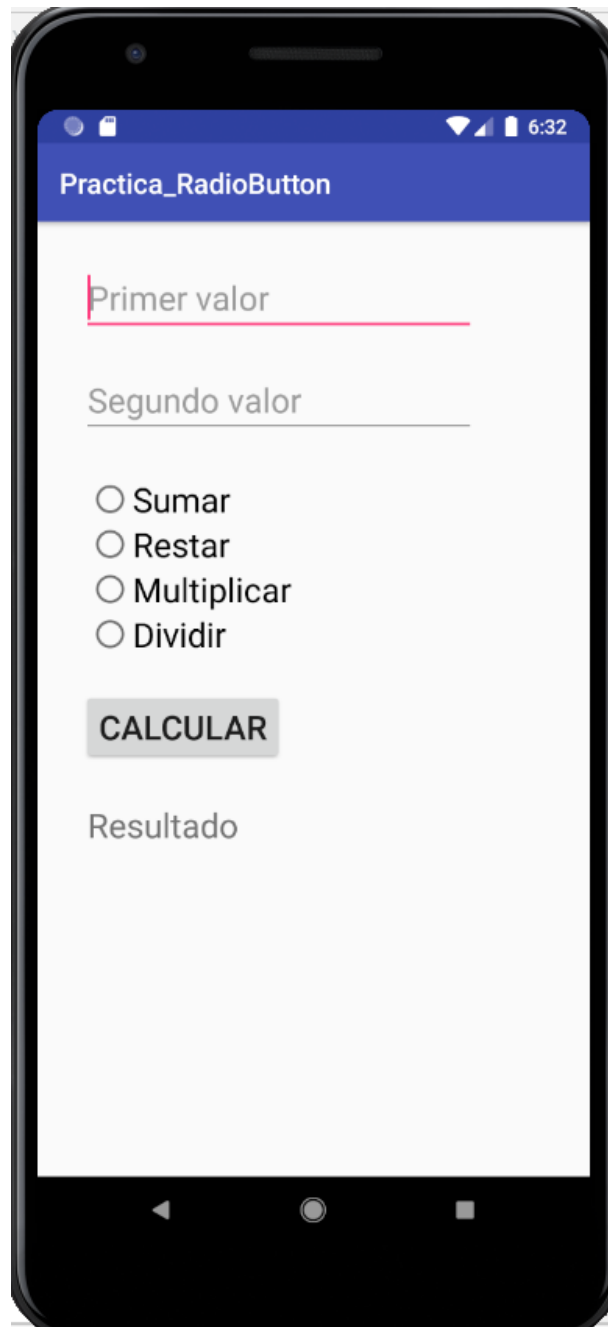


6 Propuestas de proyectos

6.1 Calculadora

Realizar un proyecto con dos EditText numerico, un RadioGroup con cuatro opciones (sumar, restar, multiplicar y dividir) y TextView donde se mostrar el resultado una vez que se ha introducido dos números y se ha seleccionado una opción del RadioGroup

El resultado final debe ser el siguiente.



6.2 Tipo de cliente

Realizar un proyecto donde se debe mostrar los campos **Razón social**, **Representante** y **Número de empleados** si se selecciona *Cliente corporativo*. De lo contrario, mostrar los campos para **nombre completo** y **profesión** para la opción *Particular*.

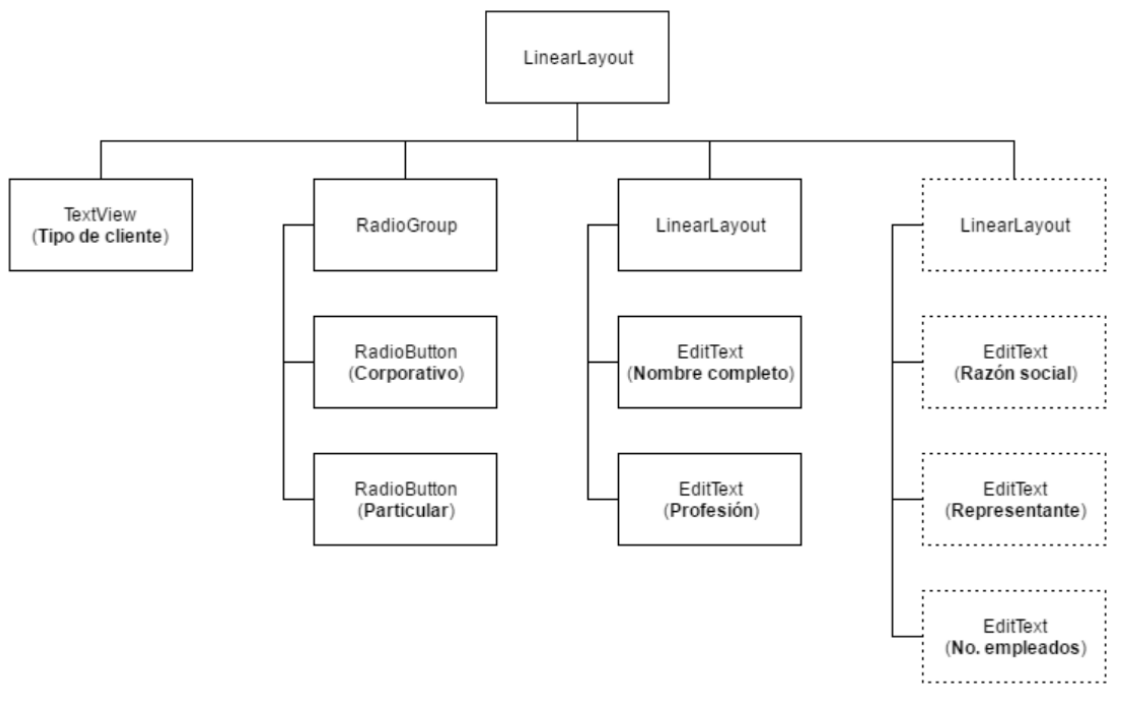
Se debe de tener:

- Texto informativo sobre el tipo de cliente

- Conjunto de dos radios.
- Contenedor de formulario con los campos de texto

El RadioButton marcado por defecto será *Particular* por lo que tendremos los campos de texto asociados visibles al inicio. Sin embargo al seleccionar *Corporativo* debes intercambiar la visibilidad de estos elementos con los asociados a este tipo de cliente.

El siguiente diagrama resume la estructura del layout:



Lo primero a tener en cuenta es que el LinearLayout corporativo y el particular deben de tener los atributos:

- **android:id**: para identificar el layout
- **android:visibility**: para indicar si el layout es visible o no. Los valores son: *VISIBLE* si esta visible y *GONE* si no esta visible.

Se debe utilizar el método **setVisibility()** para hacer visible o ocultar un layout.

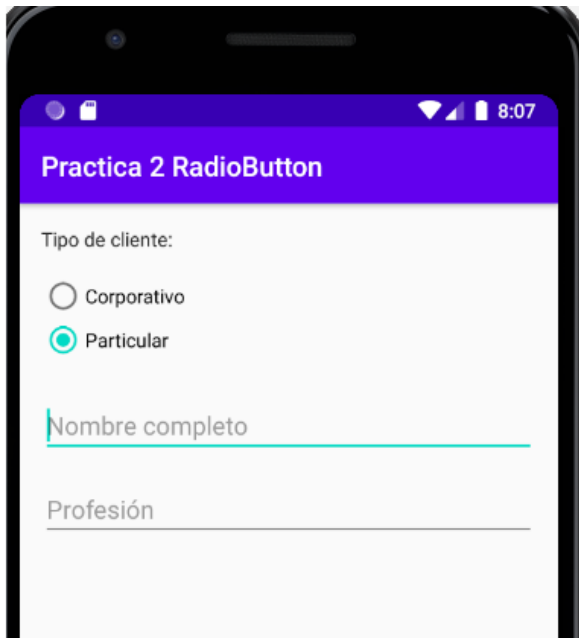
Debemos crear un método para hacer visible o ocultar el layout particular o corporativo.

```

1 private void mostrarParticular(boolean b) {
2     // contenedorParticular.setVisibility(b ? View.VISIBLE : View.GONE);
3     // contenedorCorporativo.setVisibility(b ? View.GONE : View.VISIBLE);
4
5     if (b){
6         contenedorParticular.setVisibility( View.VISIBLE);
7         contenedorCorporativo.setVisibility(View.GONE );
8     }
9     else
  
```

```
10     {  
11         contenedorParticular.setVisibility( View.GONE);  
12         contenedorCorporativo.setVisibility(View.VISIBLE );  
13     }  
14 }
```

El resultado final debe ser:



The screenshot shows a mobile app interface titled "Practica 2 RadioButton". Under the heading "Tipo de cliente:", there are two radio buttons: "Corporativo" (unselected) and "Particular" (selected, indicated by a blue dot). Below the radio buttons, there are two text input fields: "Nombre completo" and "Profesión". The "Nombre completo" field has a blue underline, suggesting it is the active input field.



The screenshot shows the same mobile app interface. In this state, the "Corporativo" radio button is selected (indicated by a blue dot), and the "Particular" radio button is unselected. The text input fields "Razón social", "Representante", and "Número de empleados" are visible below the radio buttons.