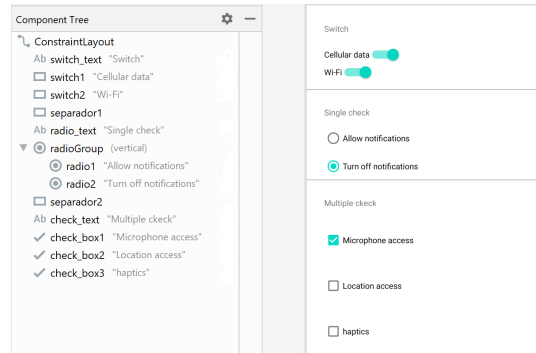


# Ejercicio resuelto controles de selección

[Descargar estos apuntes](#)

Vamos a desarrollar una interfaz donde integraremos los diferentes controles de selección que hemos visto en los apuntes:



El archivo xml correspondiente a esta interfaz sería:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.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"
    tools:context=".MainActivity">
```

9

```
<TextView
```

```
    android:id="@+id/switch_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Switch"
    android:layout_margin="30dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent"/>
```

18

```
<com.google.android.material.switchmaterial.SwitchMaterial
```

```
    android:id="@+id/switch1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:checked="true"
    android:text="Cellular data"
    android:layout_margin="30dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="@id/switch_text" />
```

26

```
<com.google.android.material.switchmaterial.SwitchMaterial
```

```
    android:id="@+id/switch2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="30dp"
    android:layout_marginLeft="48dp"
    android:layout_marginTop="88dp"
    android:checked="true"
    android:text="Wi-Fi"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="@id/switch1" />
```

40

```
<View
```

```
    android:id="@+id/separador1"
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:layout_marginTop="5dp"
    android:background="@android:color/darker_gray"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toBottomOf="@id/switch2" />
```

50

```
<TextView
```

```
    android:id="@+id/radio_text"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Single check"
        android:layout_margin="30dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="@id/separador1"/>

```

59

<RadioGroup

```

        android:id="@+id/radioGroup"
        android:checkedButton="@+id/radio_button_1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="30dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="@id/radio_text">

```

<RadioButton

```

        android:id="@+id/radio1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        android:text="Allow notifications"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="@id/radio_text" />

```

<RadioButton

```

        android:id="@+id/radio2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        android:text="Turn off notifications"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="@id/radio1" />

```

85

</RadioGroup>

87

<View

```

        android:id="@+id/separador2"
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:layout_marginTop="5dp"
        android:background="@android:color/darker_gray"
        app:layout_constraintBottom_toBottomOf="@id/radioGroup"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintLeft_toLeftOf="parent" />

```

97

<TextView

```

        android:id="@+id/check_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Multiple ckeck"
        android:layout_margin="30dp"
        app:layout_constraintLeft_toLeftOf="parent"

```

```

        app:layout_constraintTop_toTopOf="@id/separador2"/>

<CheckBox
    android:id="@+id/check_box1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="30dp"
    android:checked="true"
    android:text="Microphone access"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/check_text" />

<CheckBox
    android:id="@+id/check_box2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="30dp"
    android:text="Location access"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/check_box1" />

<CheckBox
    android:id="@+id/check_box3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="30dp"
    android:text="haptics"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/check_box2" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

#### Aclaraciones:

- **Líneas 9,50 y 97:** `TextView` para identificar los tipos de control de entrada. Recordar que las vistas incluidas en el mismo deberán llevar al menos dos anclajes que determinen su posición.
- **Líneas 18-26:** para definir el primer `Switch`.
- **Líneas 40 y 87:** líneas separadoras entre tipos de entrada, se definen con `View`.
- **Líneas 59-85:** `RadioGroup` que agrupa los tres `RadioButton`.
- **Líneas 59-85:** primer `CheckBox`.

Solamente nos falta analizar el código de nuestra aplicación, donde iremos viendo como controlar las acciones sobre los elementos de la vista.

```

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        val view = binding.root
        setContentView(view)

        binding.checkBox1.setOnCheckedChangeListener { buttonView, isChecked ->
            if (isChecked){
                Toast.makeText(applicationContext,"Microphone access CHECKED",Toast.LENGTH_SHORT).show()
            }else{
                Toast.makeText(applicationContext,"Microphone access UNCHECKED",Toast.LENGTH_SHORT).show()
            }
        }

        binding.radioGroup.setOnCheckedChangeListener { _, checkedId ->
            when(checkedId){
                binding.radio1.id-> Toast.makeText(applicationContext,"Allow notification",Toast.LENGTH_SHORT).show()
                binding.radio2.id-> Toast.makeText(applicationContext,"Turn off notification",Toast.LENGTH_SHORT).show()
            }
        }

        binding.switch1.setOnCheckedChangeListener { buttonView, isChecked->
            if (isChecked){
                Toast.makeText(applicationContext,"Celular data ON",Toast.LENGTH_LONG).show()
            }else{
                Toast.makeText(applicationContext,"Celular data OFF",Toast.LENGTH_LONG).show()
            }
        }
    }
}

```

#### Aclaraciones:

- **Líneas 9-15:** implementamos el `setOnCheckedChangeListener` para cada `CheckBox` comprobamos su estado y visualizamos un `Toast`
- **Líneas 17-22:** el escuchador del `RadioGroup` nos permite saber que `id` de `RadioButton` se ha pulsado.
- **Líneas 24-30:** escuchador para `Switch` similar al primero.
- Falta implementar los escuchadores del resto de vistas...