

Programación Multimedia y dispositivo móviles

Teresa Carrero

Desarrollo Aplicaciones Multiplataforma

Table of contents

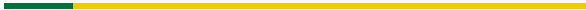
1. Objetivos
2. Button
3. Checkboxes
4. Radio Buttons
5. Conclusión

Objetivos



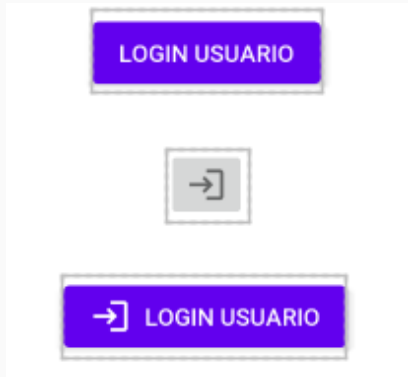
- Conocer los controles básicos en Android
- Utilizar el control Button
- Utilizar el control Checkboxes
- Utilizar el control Radio Buttons

Button



Button

Un botón consiste en un texto o un ícono (o ambos) que comunica la acción que ocurrirá cuando el usuario lo toque.



Button con texto

```
1 <Button
2     android:id="@+id/button"
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     android:layout_marginTop="32dp"
6     android:text="@string/btn_login"
7     app:layout_constraintEnd_toEndOf="parent"
8     app:layout_constraintStart_toStartOf="parent"
9     app:layout_constraintTop_toTopOf="parent" />
```

Button con Icono

```
1 <ImageButton
2     android:id="@+id/imageButton"
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     android:layout_marginTop="32dp"
6     app:layout_constraintEnd_toEndOf="parent"
7     app:layout_constraintStart_toStartOf="parent"
8     app:layout_constraintTop_toBottomOf="@+id/button"
9     app:srcCompat="@drawable/ic_login" />
```


Button con Con texto y un ícono

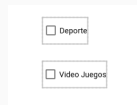
```
1  <Button
2      android:id="@+id/button2"
3      android:layout_width="wrap_content"
4      android:layout_height="wrap_content"
5      android:layout_marginTop="32dp"
6      android:drawableLeft="@drawable/ic_login_blanco"
7      android:text="@string/btn_login"
8      app:layout_constraintEnd_toEndOf="parent"
9      app:layout_constraintHorizontal_bias="0.498"
10     app:layout_constraintStart_toStartOf="parent"
11     app:layout_constraintTop_toBottomOf="@+id/
    imageButton" />
```

Checkboxes



Checkboxes

Las casillas de verificación permiten que el usuario seleccione una o más opciones de un conjunto. Por lo general, debes presentar cada opción de casilla de verificación.



Para crear cada opción, crea un objeto `CheckBox` en tu diseño. Dado que un conjunto de opciones de casilla de verificación le permite al usuario seleccionar varios elementos, se administra cada casilla por separado y debes registrar un objeto de escucha de clics para cada una.

Crear Checkboxes

```
1  <CheckBox
2      android:id="@+id/check_deporte"
3      android:layout_width="wrap_content"
4      android:layout_height="wrap_content"
5      android:layout_marginTop="36dp"
6      android:text="@string/checkBox_deporte"
7      app:layout_constraintEnd_toEndOf="parent"
8      app:layout_constraintHorizontal_bias="0.448"
9      app:layout_constraintStart_toStartOf="parent"
10     app:layout_constraintTop_toTopOf="parent" />
```

Deberemos de tener tantos elementos CheckBox como necesite la actividad.

Cuando el usuario selecciona una casilla de verificación, el objeto `CheckBox` recibe un evento de clic.

A fin de definir el controlador de eventos de clic para una casilla de verificación, se debe agregar el atributo **`android:onClick`** al elemento **`CheckBox`** en el diseño XML. El valor de este atributo debe ser el nombre del método al que deseas llamar en respuesta a un evento de clic. La Activity que aloja el diseño debe implementar el método correspondiente.

```
1 <CheckBox
2     android:id="@+id/check_deporte"
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     android:layout_marginTop="36dp"
6     android:onClick="evaluar"
7     android:text="@string/checkBox_deporte"
8     app:layout_constraintEnd_toEndOf="parent"
9     app:layout_constraintHorizontal_bias="0.448"
10    app:layout_constraintStart_toStartOf="parent"
11    app:layout_constraintTop_toTopOf="parent" />
```

En la Activity que aloja este diseño, el siguiente método procesa el evento de clic para ambas casillas de verificación:

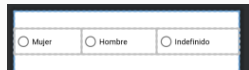
```
1  public void evaluar(View view) {
2
3      CheckBox checkBox = (CheckBox) view;  //creamos
         variable y inicializamos
4      boolean checkchek = checkBox.isChecked();
5
6      //evaluar
7      switch (view.getId()) {
8          case R.id.check_deporte:
9              if (checkchek){
10                 Toast.makeText(this, "Deporte", Toast.
LENGTH_SHORT).show();
11             }else{
12                 Toast.makeText(this, " DESESCHEAR
Deporte", Toast.LENGTH_SHORT).show();
13             }
14             break;
15     .....
}
```

Radio Buttons



Radio Buttons

Los botones de selección permiten al usuario seleccionar una opción de un conjunto.

A screenshot of a web form with three radio buttons. The first button is labeled 'Mujer', the second 'Hombre', and the third 'Indefinido'. The 'Mujer' button is selected, indicated by a small dot in the center of the circle.

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.

Para crear cada opción del botón de selección, crea un `RadioButton`. 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.

```
1 <RadioGroup
2     android:layout_width="0dp"
3     android:layout_height="wrap_content"
4     android:layout_marginTop="32dp"
5     android:orientation="horizontal"
6     app:layout_constraintEnd_toEndOf="parent"
7     app:layout_constraintStart_toStartOf="parent"
8     app:layout_constraintTop_toTopOf="parent" >
9
10    <RadioButton
11        android:id="@+id/radioButton_mujer"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:layout_weight="1"
15        android:onClick="opcionSeleccionada"
16        android:text="Mujer" />
17    ....
```

Deberemos de tener tantos elementos Radio Button como necesite la actividad.

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 elemento **`RadioButton`** en tu diseño XML. El valor de este atributo debe ser el nombre del método al que deseas llamar en respuesta a un evento de clic. La Activity que aloja el diseño debe implementar el método correspondiente.

Evento onClick

Dentro de la Activity que aloja este diseño, el siguiente método procesa el evento de clic para ambos botones de selección:

```
1 public void opcionSeleccionada(View view) {  
2     RadioButton radioButton = (RadioButton) view;  
3  
4     switch (view.getId()){  
5         case R.id.radioButton_mujer:  
6             Toast.makeText(this, "Eres Mujer", Toast.  
LENGTH_SHORT).show();  
7         case R.id.radioButton_hombre:  
8             Toast.makeText(this, "Eres Hombre", Toast.  
LENGTH_SHORT).show();  
9         case R.id.radioButton_indefinido:  
10            Toast.makeText(this, "Eres Indefinido",  
Toast.LENGTH_SHORT).show();  
11     }  
12 }
```

Conclusión

Dudas?

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

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

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