Check Box y Radio Buttons

Controles de Interfaz de Usuario

Introducción

Hasta ahora vimos distintos objetos de la interfaz de usuario como Etiquetas (Small Text, Large Text, etc), TextView y Botones.

Ahora vamos a aprender el uso de
CheckBox y ***RadioButtons.***

CheckBox

- Se suele utilizar para marcar o desmarcar opciones en una aplicación

- En Android está representado por la clase CheckBox

Propiedades CheckBox

Para implementar los CheckBox en nuestro layout podemos usar:

```
<CheckBox android:id="@+id/ChkMarcame"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/marcame"
    android:checked="false" />
```

Extiende todas las propiedades (hereda) del TextView, por lo tanto tiene los mismos atributos y se agregan los inherentes al cuadro del check

`android:checked` =”true” para marcado
`android:checked` =”false” para desmarcado

Propiedades CheckBox

En la aplicación podremos utilizar los métodos ***isChecked()*** para conocer el estado del control, y ***setChecked(estado)*** para establecer un estado concreto para el control.

```
if (checkBox.isChecked()) {
    checkBox.setChecked(false);
}
```

Métodos CheckBox

Los eventos que dispara este control pueden ser: ***OnClick*** nos indicará cuándo se ha pulsado sobre el checkbox.
Dentro de este evento consultaremos el estado del control con ***isChecked()*** como acabamos de ver.

```
cbMarcame = (CheckBox)findViewById(R.id.ChkMarcame);

cbMarcame.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        boolean isChecked = ((CheckBox)view).isChecked();

        if (isChecked) {
            cbMarcame.setText("Checkbox marcado!");
        }
        else {
            cbMarcame.setText("Checkbox desmarcado!");
        }
    }
});
```

Métodos CheckBox

Otro evento que podríamos utilizar es ***onCheckedChanged***, nos informa que ha cambiado el estado del control.

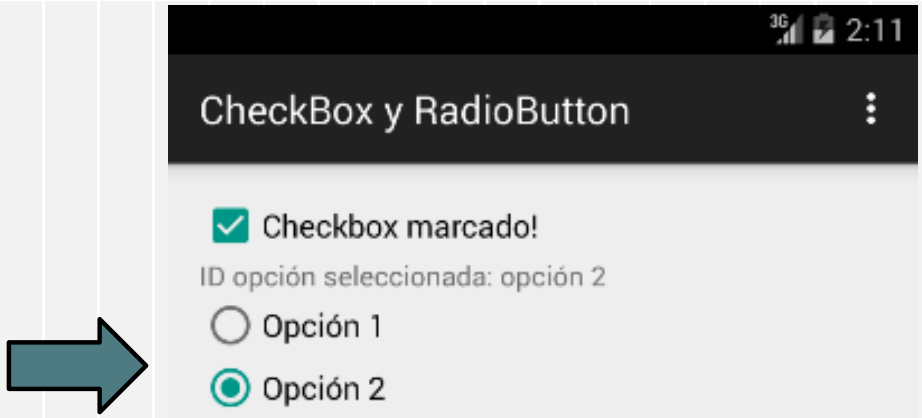
Lógica para implementar este evento: tras capturar el evento, y dependiendo del nuevo estado del control (variable isChecked recibida como parámetro), haremos una acción u otra:

```
cbMarcame = (CheckBox)findViewById(R.id.ChkMarcame);

cbMarcame.setOnCheckedChangeListener(new CheckBox.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (isChecked) {
            cbMarcame.setText("Checkbox marcado!");
        }
        else {
            cbMarcame.setText("Checkbox desmarcado!");
        }
    }
});
```

RadioButtons

- Puede estar marcado o desmarcado al igual que el CheckBox.
- En Android está representado por la ***clase CheckBox***
- Funcionan en grupo y sólo una opción del grupo puede estar seleccionada. (desmarca auromáticamente el resto)



Propiedades RadioButtons

- Se puede definir este objeto en nuestra interfaz con:

```
<RadioGroup android:id="@+id/GrbGrupo1"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <RadioButton android:id="@+id/RbOpcion1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/opcion_1" />

    <RadioButton android:id="@+id/RbOpcion2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/opcion_2" />
</RadioGroup>
```

- El grupo de botones se puede definir como ***RadioGroup*** que será el que contenga a los ***RadioButton*** donde sólo 1 = true.

Propiedades RadioButtons

- Hay que definir

```
<RadioGroup android:id="@+id/GrbGrupo1"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <RadioButton android:id="@+id/RbOpcion1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/opcion_1" />

    <RadioButton android:id="@+id/RbOpcion2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/opcion_2" />
</RadioGroup>
```

Métodos RadioButton

Los eventos que permiten manipular el control desde nuestro código javason:

- ***check(id)*** para marcar una opción determinada mediante su ID,
- ***clearCheck()*** para desmarcar todas las opciones, y
- ***getCheckedRadioButtonId()*** que como su nombre indica devolverá el ID de la opción marcada (o el valor -1 si no hay ninguna marcada).

Veamos un ejemplo:

```
RadioGroup rg = (RadioGroup)findViewById(R.id.GrbGrupo1);
rg.clearCheck();
rg.check(R.id.RbOpcion1);
int idSeleccionado = rg.getCheckedRadioButtonId();
```

Métodos RadioButton

En el evento ***onClick*** debemos averiguar cuándo se pulsa cada uno de los botones del grupo.

Generalmente ***utilizaremos un mismo listener*** para todos los radiobutton del grupo, por lo que lo definiremos de forma independiente y después lo asignaremos a todos los botones.

```
private TextView lblMensaje;
private RadioButton rbOpcion1;
private RadioButton rbOpcion2;
//...
lblMensaje = (TextView)findViewById(R.id.LblSeleccion);
rbOpcion1 = (RadioButton)findViewById(R.id.RbOpcion1);
rbOpcion2 = (RadioButton)findViewById(R.id.RbOpcion2);

View.OnClickListener listener = new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String opcion = "";
        switch(view.getId()) {
            case R.id.RbOpcion1:
                opcion = "opción 1";
                break;
            case R.id.RbOpcion2:
                opcion = "opción 2";
                break;
        }
        lblMensaje.setText("ID opción seleccionada: " +
                                                opcion);
    }
};

rbOpcion1.setOnClickListener(listener);
rbOpcion2.setOnClickListener(listener);
```

Métodos RadioButton

En el evento ***onCheckedChange*** nos informará de los cambios en el elemento seleccionado dentro de un grupo. La diferencia con el control ***CheckBox*** es que este evento está asociado al ***RadioGroup***, y no a los diferentes ***RadioButton*** del grupo.

Veamos cómo tratar este evento con un ejemplo donde una etiqueta de texto cambie de valor al seleccionar cada opción:

```
rgOpciones = (RadioGroup)findViewById(R.id.GrbGrupo1);
rgOpciones.setOnCheckedChangeListener(new RadioGroup.
OnCheckedChangeListener() {
    public void onCheckedChanged(RadioGroup group, int checkedId) {

        String opcion = "";
        switch(checkedId) {
            case R.id.RbOpcion1:
                opcion = "opción 1";
                break;
            case R.id.RbOpcion2:
                opcion = "opción 2";
                break;
        }
        lblMensaje.setText("ID opción seleccionada: " + opcion);
    }
});
```

Proyecto de Ejemplo

Descargar el proyecto de ejemplo android-check-radio.rar:
https://drive.google.com/a/gylgroup.com/folderview?id=0B4MhWvqxkfYlfmNWQUN3N09XUGtTUDhxc3pNX1EwRF93NzFLQzlQZEFqbktldm1SSG1uTjQ&usp=sharing

Para probar otras características de los botones explorar el proyecto android-botones.rar