

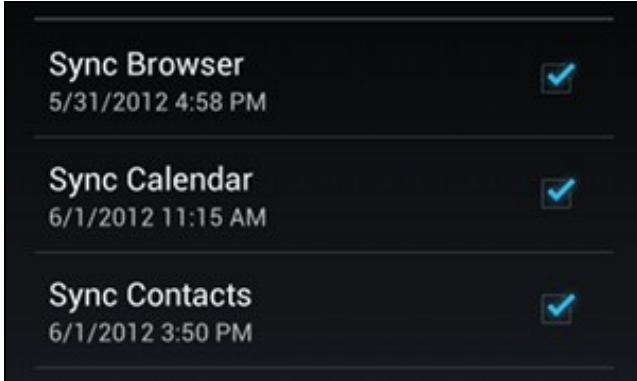
1 CheckBox

1.1 Sumario

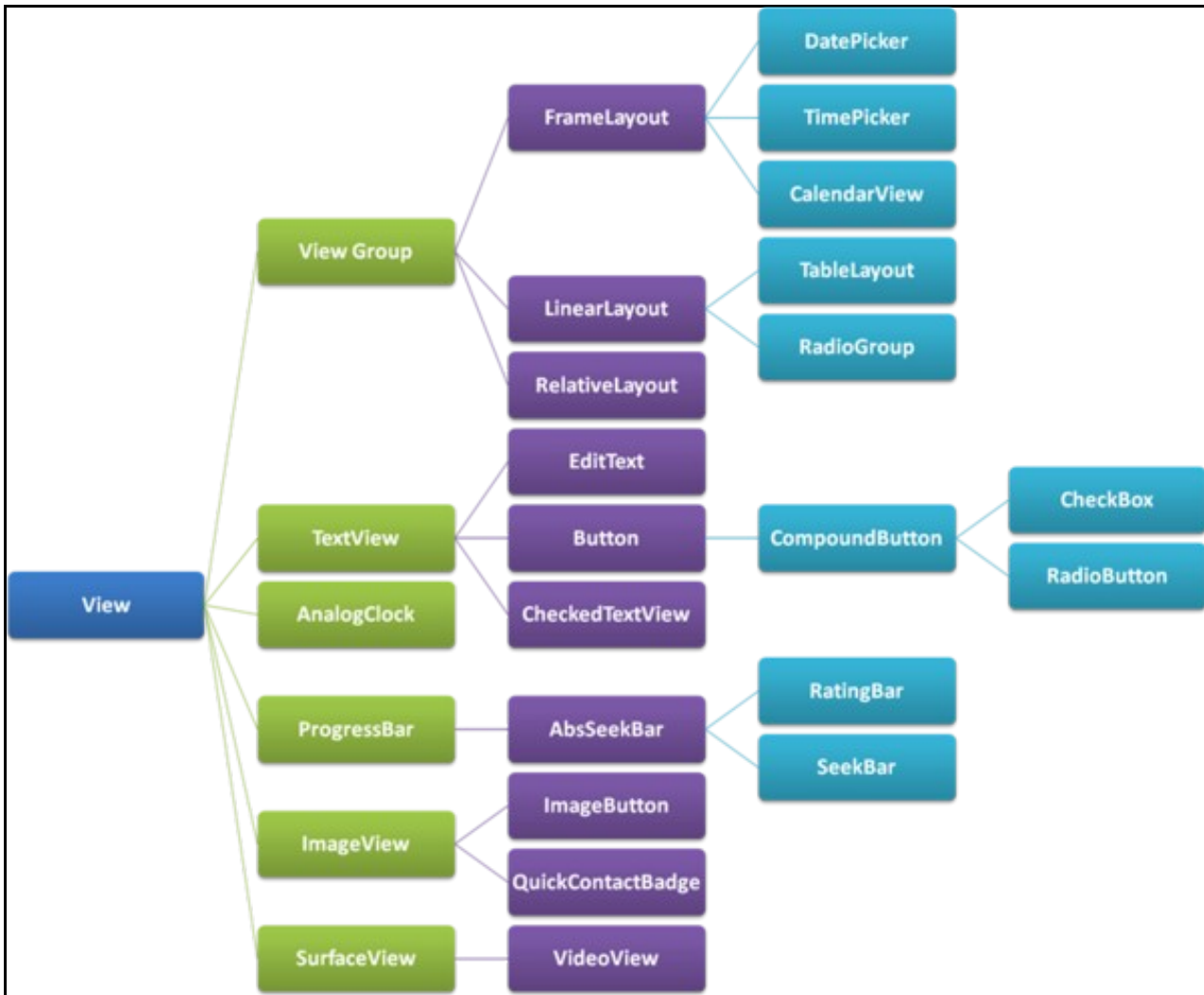
- 1 Introducción
- 2 Caso práctico
 - ◆ 2.1 Layout xml
 - ◆ 2.2 Código Java

1.2 Introducción

- Un **CheckBox** permite ao usuario elixir unha ou máis opcións dentro dun conxunto.



- Este control ao igual que ToggleButton e Switch herda da clase **CompoundButton**, quen, á súa vez, herda da clase **Button**.



• Imaxe obtida de: <http://www.itcsolutions.eu/2011/08/27/android-tutorial-4-procedural-vs-declarative-design-of-user-interfaces>

- Por tanto, funcionan da mesma maneira, pero ademais este control:
 - ◆ Ten 2 estados (True/False), que podemos comprobar co método **isChecked ()**.

• Para xestionar os eventos podemos facelo da mesma forma que nos botóns.

• Para usalo nun XML layout usar a etiqueta `<CheckBox/>`

• **Referencias:**

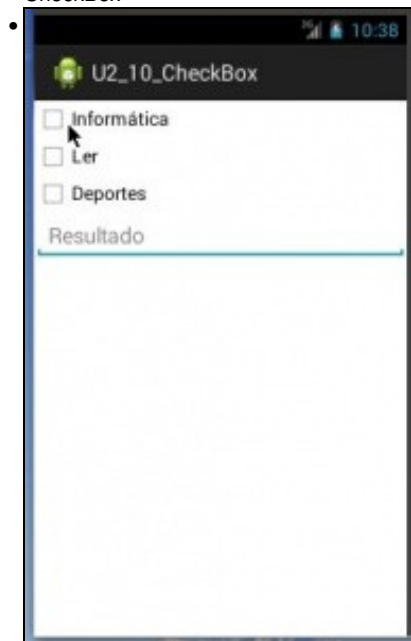
- ◆ O control CheckBox: <http://developer.android.com/reference/android/widget/CheckBox.html>
- ◆ Introducción ao CheckBox: <http://developer.android.com/guide/topics/ui/controls/checkbox.html>

1.3 Caso práctico

- Creamos un novo proxecto: **U2_10_CheckBox**

- Iremos crear unha aplicación na que o usuario indique as súas afeccións ("Aficiones").

- CheckBox



Ao entrar na aplicación non hai ningunha marcada.



Cando se selecciona unha afección aparece unha mensaxe nunha caixa de texto indicando o que se seleccionou.



Cando se deselecciona unha afección aparece unha mensaxe nunha caixa de texto indicando o que se deseleccionou.

1.3.1 Layout xml

- Cada **CheckBox** chama a un método cando se marca ou desmarca ese CheckBox.
- O método é o mesmo para os tres.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <CheckBox
        android:id="@+id/chk_informatica"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="controlarAfeccions"
        android:text="Informática" />

    <CheckBox
        android:id="@+id/chk_ler"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="controlarAfeccions"
        android:text="Ler" />

    <CheckBox
        android:id="@+id/chk_deporte"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="controlarAfeccions"
        android:text="Deportes" />

    <EditText
        android:id="@+id/txt_resultado"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Resultado" />

</LinearLayout>
```

1.3.2 Código Java

```
package com.example.u2_10_checkbox;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.CheckBox;
import android.widget.EditText;

public class U2_10_CheckBox extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_u2_10__checkbox_box);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.u2_10__checkbox_box, menu);
        return true;
    }

    public void controlarAfecciones(View view) {
        CheckBox check = (CheckBox) view;

        EditText caixa = (EditText) findViewById(R.id.txt_resultado);
        if (check.isChecked())
            caixa.setText("Marcaches: " + check.getText());
        else
            caixa.setText("Desmarcaches: " + check.getText());
    }
}
```

- **Liña 25**, recibimos a Vista do elemento pulsado hai que facer un casting para convertelo ao tipo CheckBox.
- **Liña 29**, comprobamos se se marcou ou desmarcou o control.