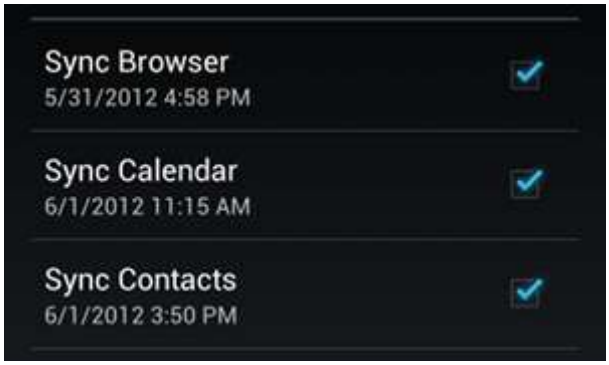
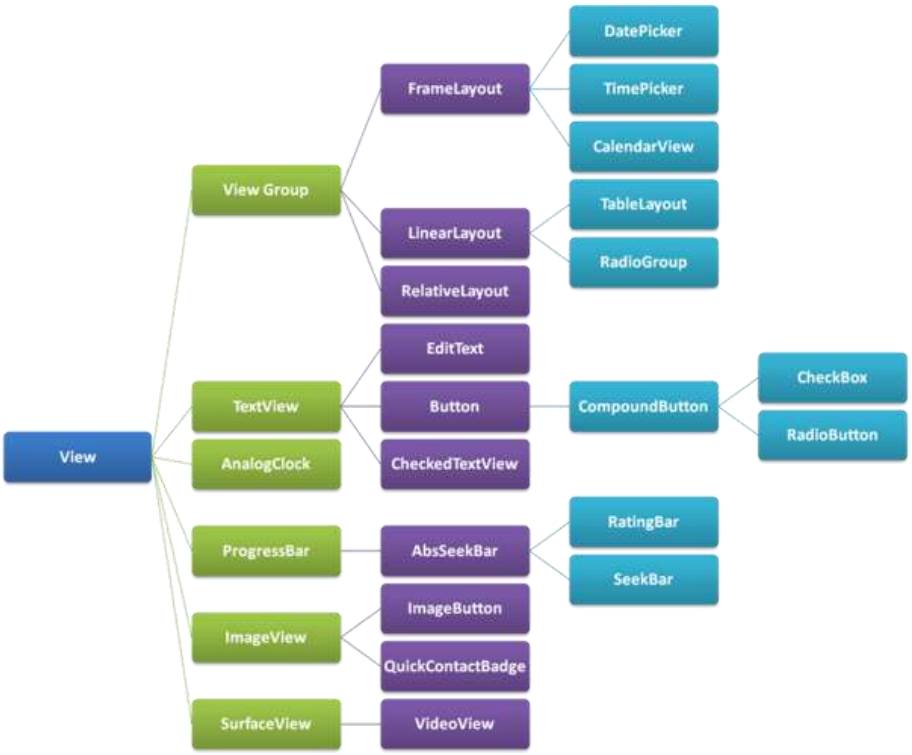
INDICE

1.1. Introducción

- Un **CheckBox** permite al usuario elegir una o más opciones dentro de un conjunto.



- Este control como ToggleButton y Switch hereda de la clase **CompoundButton**, que a su vez hereda de la clase **Button**.



- Por tanto, funcionan de la misma forma:

- Tiene 2 estados (True/False), que podemos comprobar con el método isChecked().
- Podemos cambiar su estado por programación llamando al método setChecked (boolean).
- Podemos gestionar el cambio de estado de un CheckBox con la llamada al método setOnCheckedChangeListener (OnCheckedChangeListener)

1.2. Caso práctico 1

- Seguimos trabajando a partir del proyecto base..

  Si no lo hemos creado antes, crear un paquete llamado **UI** como un subpaquete de su paquete principal.

  Dentro del paquete de IU, crearemos un nuevo paquete llamado: **Checkboxes**.

- Dentro del paquete **Checkboxes**, crear una nueva 'Empty Activity' de nombre: **UD02_01_CkeckBox** tipo Launcher.

  Modificar el archivo **AndroidManifiest.xml** y agregar una label a la actividad como hemos hecho anteriormente.

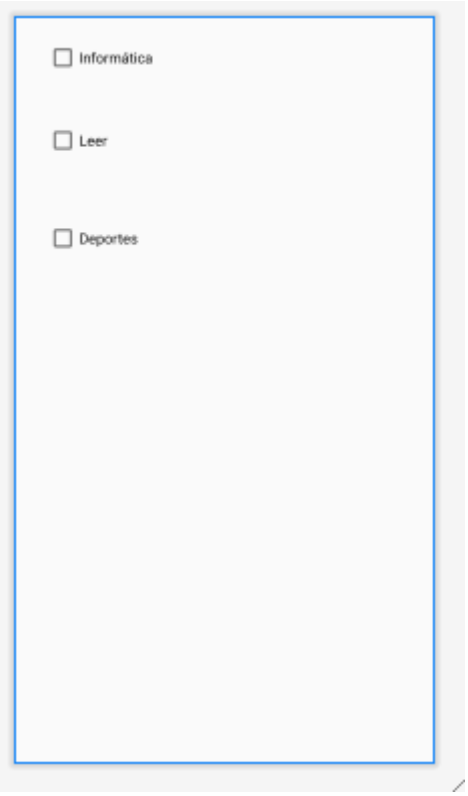
- Crearemos una aplicación en la que el usuario indique sus aficiones ("Aficiones").

Como aún no sabemos cómo gestionar eventos, en el método onCreate nos referiremos a los tres checkbox y modificaremos varias propiedades.

CheckBox

Arrastramos un checkbox

Cambiamos el texto. Podemos ver otras propiedades del View. El símbolo con guión indica que tiene asignado el valor por defecto. Se clicamos o cambiamos a true/false.



Colocamos los tres CheckBox´s a nuestro gusto.

Código del Layout

activity_u_d02_01__check_box.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/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=".UI.Checkboxes.UD02_01_CheckBox">

    <CheckBox
        android:id="@+id/chkInformatica"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="Informática"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <CheckBox
        android:id="@+id/chkLeer"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="Leer"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/chkInformatica" />

    <CheckBox
        android:id="@+id/chkDeporte"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="Deportes"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/chkLeer" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Código de la Activity

UD02_01_CheckBox.java

```
package olga100.proyectosandroid.pmdm_2020_2021.UI.Checkboxes;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Color;
import android.os.Bundle;
import android.widget.CheckBox;
import android.widget.Toast;

import olga100.proyectosandroid.pmdm_2020_2021.R;

public class UD02_01_CheckBox extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_u_d02_01__check_box);

        CheckBox checkLeer = findViewById(R.id.chkLeer);
        Toast.makeText(this,checkLeer.getText().toString() + " " + String.valueOf(checkLeer.isChecked()), Toast.LENGTH_SHORT).show();
        //Mostramos un mensaje con su texto y estado

        ((CheckBox)findViewById(R.id.chkInformatica)).setChecked(true); //Podemos no crear una variable para guardar la referencia al checkbox.
        //En este caso cambiamos el estado del checkbox

        CheckBox chkDeportes = findViewById(R.id.chkDeporte);
        chkDeportes.setBackgroundColor(Color.BLUE); //Cambiamos otras propiedades. Podemos mediante programación cualquier propiedad del diseñador.
        chkDeportes.setTextColor(Color.RED);
        chkDeportes.setText("NUEVA AFICION");
    }
}
```

1.3. Métodos mínimos a conocer en el manejo de los CheckBox

- Referenciar a un CheckBox con el método findViewById.
- Recuperar el contenido del texto asociado.
- Cambiar el contenido del texto, pudiendo emplear recursos guardados en **`/res/values`**.
- Añadir nuevo texto a un existente.
- Modificar propiedades básicas, como color, tamaño, visibilidad,...

```
CheckBox chk = findViewById(R.id.chk_Uno);
// Referenciamos a un CheckBox que este en el Layout de la Activity

        Toast.makeText(getApplicationContext(),chk.getText() + " " +
String.valueOf(chk.isChecked()),Toast.LENGTH_LONG).show();   // Obtenemos el  texto
y estado del check con isChecked()

        chk.setText("Nuevo texto Check");
// Cambiamos el texto del check

        chk.setChecked(true);
// Cambiamos el estado del check (on/off) => Lanza el evento anterior por tener el
setOnXXX antes
```