

Personalizando Listas en Android

Mercedes Wyss
@itrjwyss

<https://github.com/itrjwyss/aKnowledgeJourney>



JavaDay²⁰¹⁶
Tour Totonicapán

Fundamentos de Aplicaciones

- Se programa utilizando Java, sin embargo no son aplicaciones Java, solo utilizan la sintaxis.
- Es un sistema operativo basado en el kernel de Linux.
- Es un sistema Linux multi-usuario, por lo tanto cada aplicación es un usuario diferente.
- Cada aplicación tiene su propia VM.



Maquinas Virtuales

- **Dalvik**: la aplicación se va compilando a medida que se va navegando dentro del app.
- **ART**: compila los procesos y guarda el caché desde el momento de la instalación de la aplicación.



Componentes de la App

- Intents e Intent Filters
- Activities
- Services
- ContentProviders
- Layout
- Broadcast Receivers
- Widget
- Fragments
- Manifest File



Intents e Intent Filters

- Son objetos de mensajería que permiten la comunicación entre los diferentes componentes de la aplicación.
- Los Intents se utilizan desde el código fuente.
- Los Intent Filters se utilizan desde el Manifest File



Intents e Intent Filters

- Iniciar una activity, enviando el Intent ya sea por el método `startActivity()` o `startActivityForResult()`.
- Iniciar un service, enviando el Intent por el método `startService()`;
- Recibir un Broadcast.



Intents e Intent Filters

Existen dos tipos de Intents:

- **Explícitos:** nos sirven para llamar componentes de nuestra aplicación.
- **Implícitos:** nos sirven para llamar componentes ajenos a nuestra aplicación, suelen llamarse por medio de una acción que se desea suceda.



Activities

- Se desarrolla utilizando la arquitectura Modelo-Vista-Controlador (MVC).
- Una activity se puede ver como el controlador en esta arquitectura. El modelo son las demás clases Java.
- Las actividades poseen un Ciclo de vida que es muy importante entender.



Layouts

- Un Layout es la vista en la arquitectura MVC.
- Define la estructura visual de la interfaz de usuario.
- Se puede declarar en archivos xml o programáticamente.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

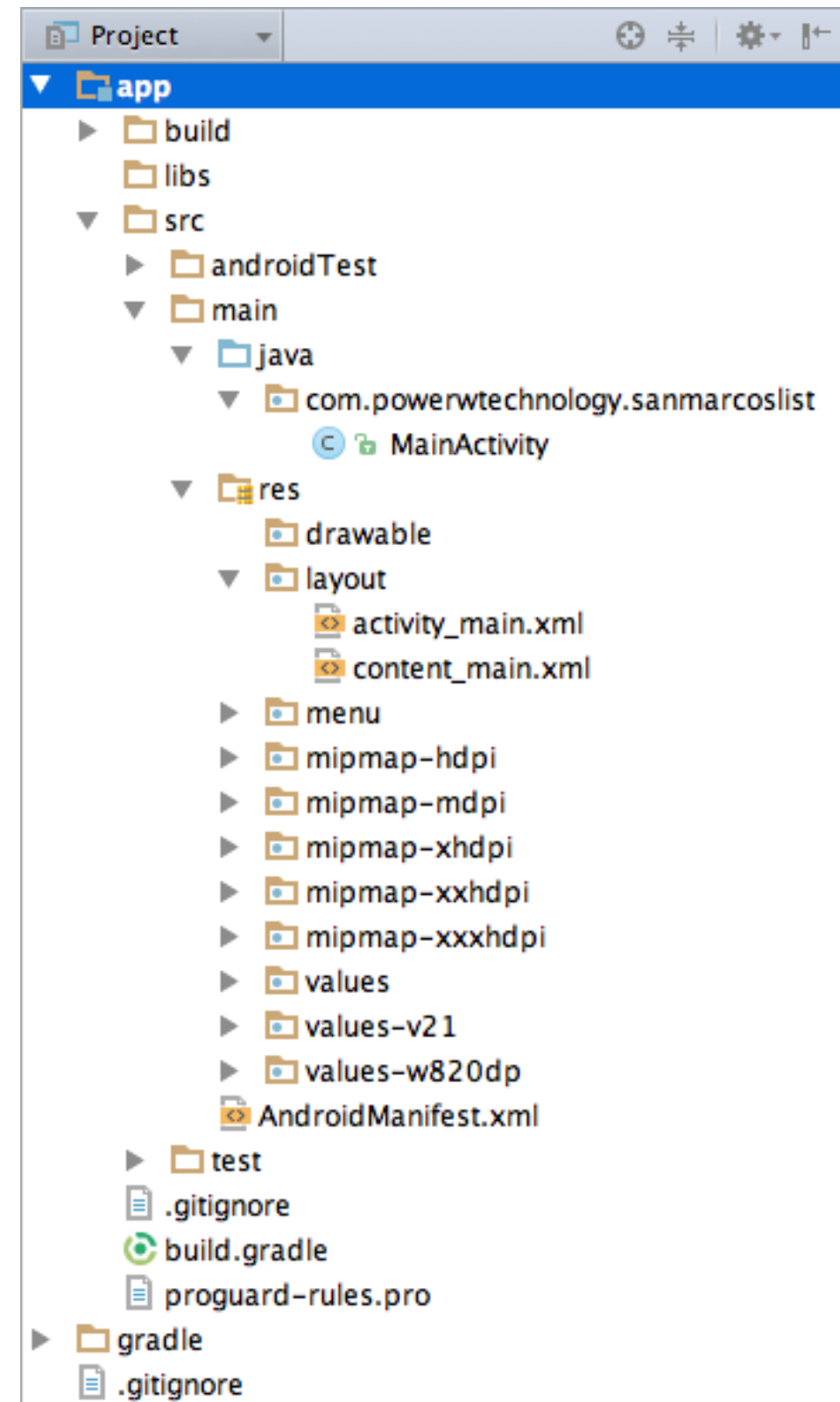


Otros Componentes

- **Services:** son componentes sin interfaz gráfica, estos se ejecutan en segundo plano.
- **Broadcast Receiver:** componente destinado a detectar y reaccionar ante determinadas acciones o eventos globales que se generan en el sistema.
- **Content Provider:** mecanismo definido en android para compartir información entre aplicaciones.



Estructura de un Proyecto



- **/app/src/main/java**
 - Código fuente de nuestra aplicación
 - Aquí se encuentran nuestros modelos y controladores
 - Los componentes de la aplicación (Activities, Services, Broadcast Receivers, entre otros)



- **/app/src/main/res**

- Aquí están todos los recursos que no son código fuente
 - ♦ /drawable/ (imágenes y recursos gráficos en xml)
 - ♦ /mipmap/ (el ícono de la aplicación)
 - ♦ /layout/ (ficheros xml que corresponden a las vistas)



- ♦ /anim/ o /animator/ (animaciones en xml)
- ♦ /menu/ (menús de la aplicación en xml)
- ♦ /xml/ (otros ficheros xml)
- ♦ /raw/ (otros recursos en un formato diferente a xml, como tipografías)
- ♦ /values/ (ficheros xml de recursos como strings, colores, dimensiones)



- `/app/src/main/AndroidManifest.xml`
- `/app/build.gradle`
- `/app/libs/`
- `/app/build/` (Archivo R)



Pasemos al Android Studio



JavaDay²⁰¹⁶
Tour Totonicapán