



# Elementos de la interfaz, navegación e interacción

Componentes de un proyecto Android

***Utilizar elementos de interfaz de usuario básicos del entorno Android para la implementación de un prototipo de acuerdo a las especificaciones entregadas.***

- Unidad 1: Ambiente de desarrollo y sus elementos de configuración.
- Unidad 2: Elementos de la interfaz, navegación e interacción.
- Unidad 3: Fundamentos de GIT y GitHub.



Te encuentras aquí



## ¿Qué aprenderás en esta sesión?

- *Reconocer los distintos componentes, pantalla y características en un proyecto Android Studio.*

**/\* Distinguir los tipos de lenguajes  
usados en el proyecto, y su  
funcionalidad \*/**

# Gradle

- Es un sistema de automatización de construcción, diseñado para construcciones multi-proyecto.
- Utiliza un lenguaje especializado para su propósito (Groovy).
  - En el caso de Android también se puede programar en Kotlin, lo que entrega gran versatilidad.

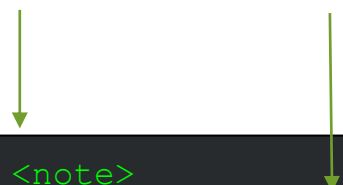


# XML

- Es un **lenguaje de etiquetas** desarrollado por el World Wide Web Consortium ([W3C](#)).
- Se utiliza para almacenar datos en forma legible para humanos.
- Se enfoca en el sentido semántico de los datos, dejando de lado la presentación.
- Las etiquetas representan una jerarquía de conceptos anidados.
- Cada etiqueta tiene un inicio <TAG> y un fin </TAG>.

Inicio de  
etiqueta

Fin de  
etiqueta



```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>
    <text>Don't forget me!</text>
    <alarm>Sunday 23, 06:00</alarm>
  </body>
</note>
```

# XML

*¿Dónde se utiliza?*



- AndroidManifest.xml
- Layouts
- Recursos de la app:
  - dimensiones (dimens.xml)
  - textos (strings.xml)
  - colores (colors.xml)
  - estilos (styles.xml)
  - temas (themes.xml)
- Archivos vectoriales para imágenes.

# Java/Kotlin

El código de la aplicación puede ser escrito tanto en Kotlin como en Java, incluso pueden convivir en la misma app llamando una clase en Java desde Kotlin y viceversa (interoperables).

Algunas diferencias entre ambos son:

Kotlin	Java
Orientado a objetos y funcional	Orientado a objetos
Todo es objeto	Tiene tipos primitivos de datos (int, long, boolean)
Funciones de extensión	Comprobación de excepciones
Smart casts	Operador ternario



# Tipos de lenguaje en un proyecto

## Actividad

1. ¿Qué lenguaje se utiliza en los elementos visuales de la app?
1. ¿Qué lenguaje está involucrado en la creación del APK o AAB?
1. ¿Se puede utilizar Java y Kotlin en el mismo proyecto?



# Resumen de tipos de lenguajes en un proyecto

Componente	Lenguaje	Función
Gradle	Groovy / Kotlin	Construir la app
app	Kotlin / Java	Código fuente de la app
Vistas / Layouts	XML	Diseño de pantallas, vectores, recursos de dimensiones
Manifiesto (AndroidManifest.xml)	XML	Describe la información esencial de la app para el SO y Google Play

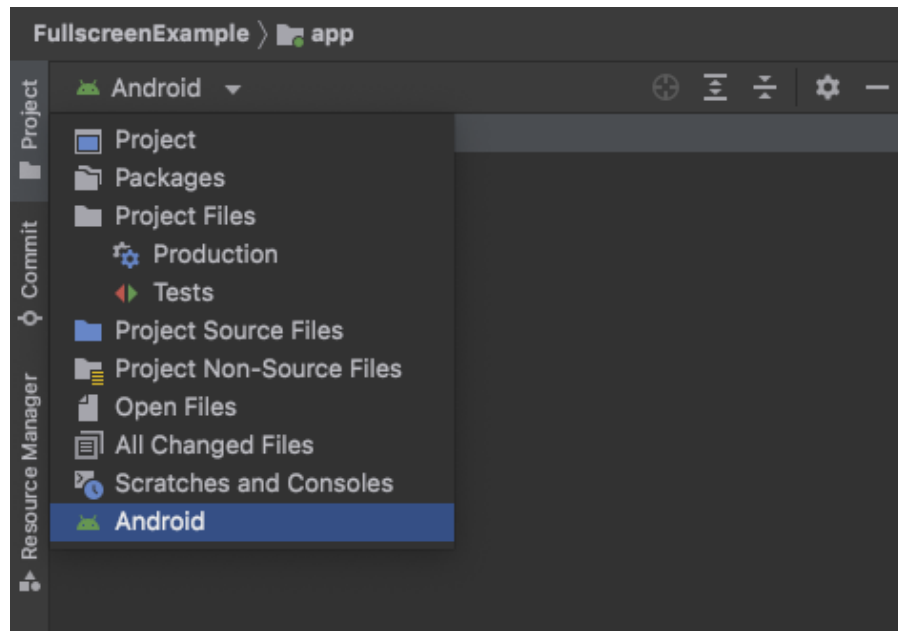
**/\* Navegar entre los distintos modos de  
visualización (panel) del árbol de  
directorios en Android Studio \*/**

# Vista de proyecto

## *Tipos de vistas de proyectos*

Android Studio permite mostrar el contenido de un proyecto utilizando distintas vistas.

Las vistas más utilizadas son de proyecto (Project) y de Android (por defecto)



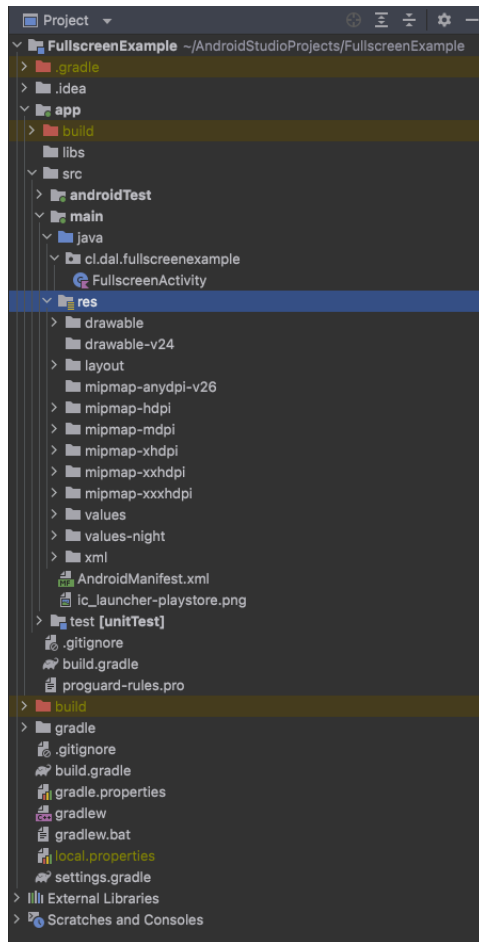
# Vista de proyecto

## Project

La vista Project muestra la estructura del directorio, mostrando cada archivo en su ubicación y muestra los archivos y carpetas ocultas.

Por ejemplo, app/src/main contiene el código de la app.

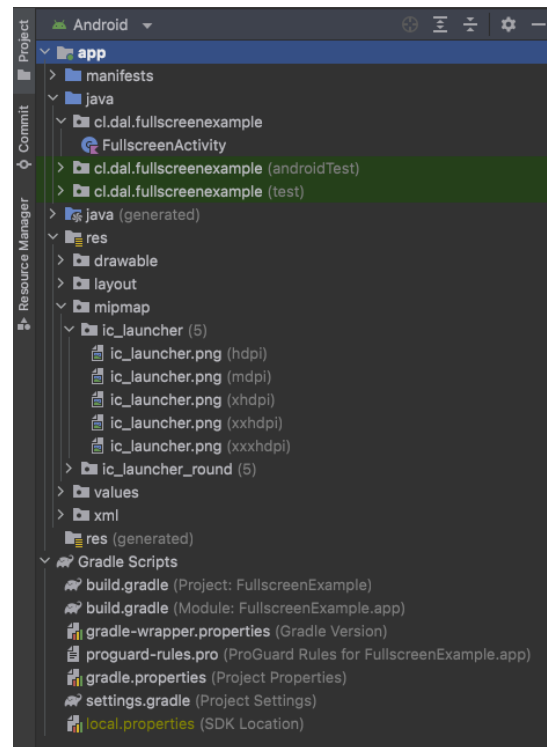
- app/main/res contiene los recursos gráficos que vamos a utilizar, dividiendo por carpetas específicas para cada resolución.



# Vista de proyecto

## Android

- La vista de proyecto Android agrupa archivos y carpetas para facilitar el trabajo con aplicaciones Android.
- En el caso de los app/res, agrupa el ícono de la app en la misma carpeta y muestra cada densidad por separado.
- Los scripts de Gradle también se agrupan y muestran todos independientemente de su ubicación.



## Comparar la visualización del proyecto Android con la del explorador de archivos:

- ¿Pará qué utilizarías la vista de Project por sobre la de Android?
- ¿Pará qué utilizarías la vista de Android por sobre la Project?



**/\* Conocer la definición y funcionalidad  
de cada componente en un proyecto  
Android \*/**



# Resumen de componentes de un proyecto Android

app	carpeta principal	Su estructura de archivos y carpetas tienen el código.
Manifiesto	AndroidManifest.xml	Describe información esencial de la app para el SO y Google Play.
Assets	<ul style="list-style-type: none"><li>● Resources</li><li>● Colors</li><li>● Dimens</li><li>● Strings</li><li>● Styles</li><li>● Themes</li></ul>	Archivos adicionales con contenido estático que se almacenan en el directorio res/. En tiempo de ejecución, es Android el que hace uso de los recursos apropiados basados en la configuración actual del dispositivo.

# Resumen de componentes de un proyecto Android

Drawable	<ul style="list-style-type: none"><li>● Mapas de bit (jpg, png, gif)</li><li>● Lista de estados</li><li>● Listas de capas</li></ul>	Concepto general para un gráfico que se puede dibujar en la pantalla o administrar la cómo se ven las vistas según su estado.
Raw	<ul style="list-style-type: none"><li>● Archivos de video (.mp4)</li><li>● Archivos de audio</li><li>● Archivos de texto (.json)</li></ul>	Son archivos arbitrarios para guardar SIN procesar. (por ejemplo, un archivo de audio).

***/\* Activity \*/***

# Activity

## Concepto



- Una actividad es la ventana donde la app dibuja su interfaz de usuario.
- Todas las aplicaciones tienen al menos 1 actividad: La actividad principal, que es la primera actividad que se muestra al ejecutar la app.
- La mayoría de las apps tienen varias pantallas, lo que implica tener varias actividades.

# Activity

1

El sistema operativo Android inicia el código de una app usando una instancia de Activity.

2

La clase Activity es un componente clave de una app Android.

3

La forma en que se inician y se crean las actividades es una parte fundamental de toda app.

4

En las aplicaciones móviles, las interacciones del usuario con la app no comienzan siempre en el mismo lugar.

5


**Importante:** Una vez que publiques tu app no deberías cambiar los nombres de las actividades

# Agregar una Activity

## Configuración del manifiesto

Cada actividad que será utilizada en la app debe estar declarada en el archivo **AndroidManifest.xml**

La actividad debe ser declarada agregando el elemento **<activity>** como objeto secundario del elemento **<application>**



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="cl.dal.fullscreenexample">
    <application
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:theme="@style/Theme.FullscreenExample">
        <activity
            android:name=".FullscreenActivity"
            android:configChanges="orientation|keyboardHidden|screenSize"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.FullscreenExample.Fullscreen">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

# Agregar una Activity

## Configuración del manifiesto

El único atributo obligatorio es **android:name** que especifica el nombre de la actividad.

Hay otros atributos opcionales que definen características como el ícono, una etiqueta o un tema.

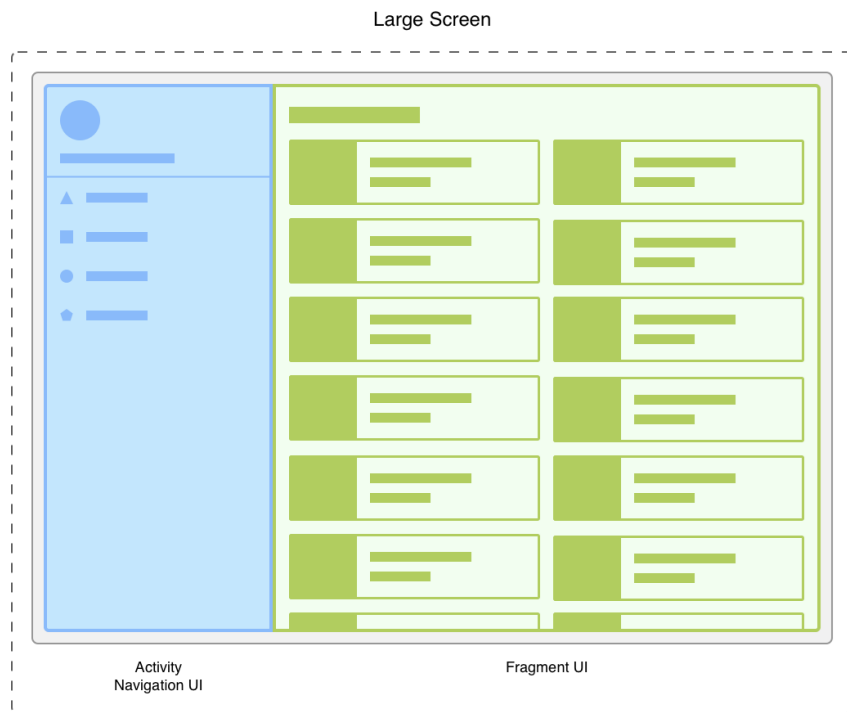
La actividad principal se indica agregando **<intent-filter>** para definir su categoría como **LAUNCHER**.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="cl.dal.fullscreenexample">
    <application
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:theme="@style/Theme.FullscreenExample">
        <activity
            android:name=".FullscreenActivity"
            android:configChanges="orientation|keyboardHidden|screenSize"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.FullscreenExample.Fullscreen">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

***/\* Fragment \*/***



# Fragment (Fragmento)



- Un Fragment corresponde a una porción reutilizable de la interfaz de usuario, que define y administra su propio diseño y sus eventos de entrada.
- Los fragmentos siempre deben ser alojados por una actividad y otro fragmento y NO pueden existir por sí solos.
- La jerarquía de vistas del fragmento forma parte de la jerarquía de vistas del *host*, que aloja al fragmento.

La pantalla grande tiene un panel de navegación controlado por la actividad y una lista cuadrada controlada por el fragmento.

# Fragment

## Modularidad

- Los fragmentos, al tener la capacidad de ser reutilizables, entregan además modularidad, permitiendo dividir la interfaz de usuario (IU) en distintas partes.
- Puedes usar varias instancias de la misma clase de fragmento dentro de la misma actividad, en varias actividades o incluso como elemento secundario de otro fragmento.
- Solo debes proporcionar a un fragmento con la lógica necesaria para administrar su propia IU.



La pantalla contiene un barra de navegación inferior controlada por la actividad y una lista lineal controlada por el fragmento.

# Activity & Fragment

Las actividades son el lugar ideal para colocar elementos globales en la interfaz de usuario de tu app, como por ejemplo, un panel lateral de navegación.

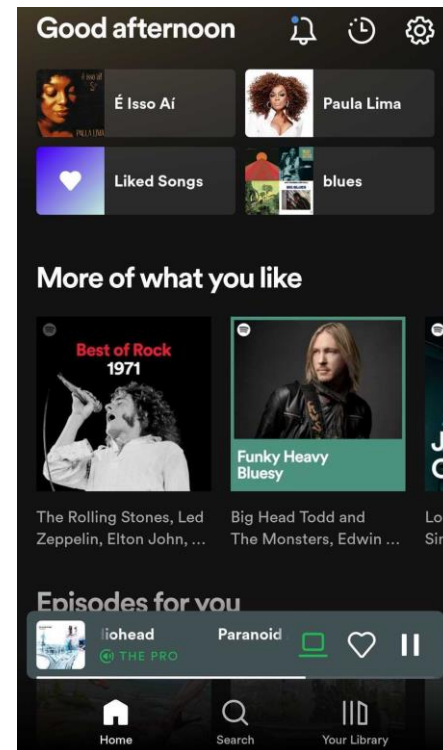
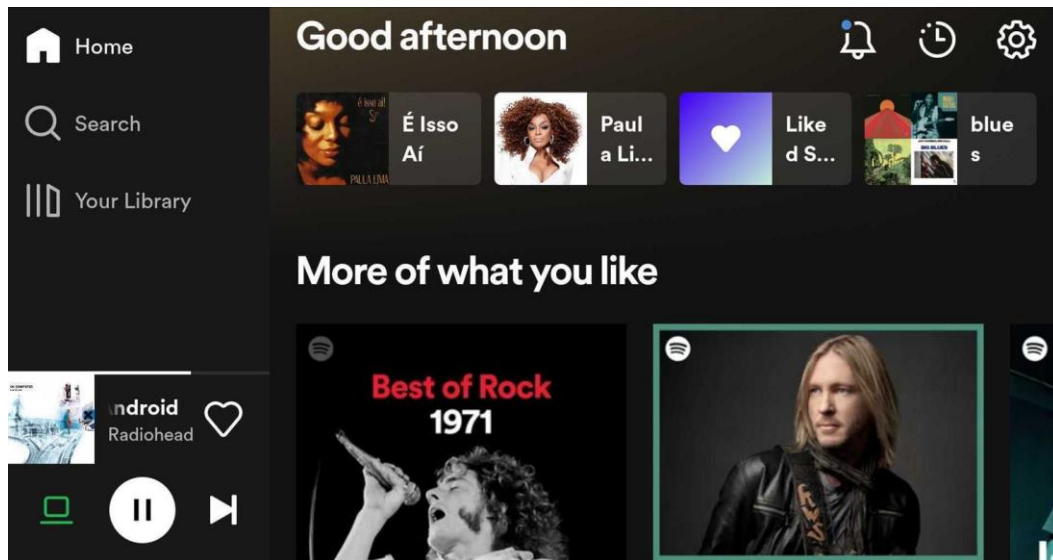
Los fragmentos son más adecuados para definir y administrar la IU de una sola pantalla o de una parte de ella.

# Ejercicio de visualización - “Activity & Fragment ”



# Activity & Fragment - Ejercicio de visualización

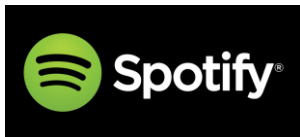
## Pantallas de Spotify



# Activity & Fragment

## *Pantalla de Spotify*

1. ¿Cuántas actividades están visibles al usar la app?
2. ¿Cuántos fragmentos están visibles al usar la app?





## Próxima sesión...

- *Utilizar elementos ViewGroups y Views requeridos para satisfacer la especificación entregada.*

**{desafío}**  
**latam\_**

*Academia de  
talentos digitales*

