

# Introducción al desarrollo Android

# Contexto Histórico Android

# Contexto Histórico

---

Android es el sistema operativo utilizado en más de mil millones de teléfonos inteligentes y tablets.

# Contexto Histórico

---

A cada versión de Android se le ha puesto el nombre de un dulce, pues se dice por ahí que “nos endulzan la vida”

# Contexto Historico

---

Si hay una nueva versión de Android eso significa  
que ahora es posible realizar algo nuevo en  
nuestro dispositivo

## Contexto historico

---

2003 Android Comenzó siendo una  
compañía creada por Andy Rubin,  
Rich Miner y otros

**Andy Rubin**  
@arubin



---

**Rich Miner**  
@richminer



---

**Android Inc.**

# Contexto histórico

---

Desarrollar un sistema operativo inteligente, que tuviese en cuenta la localización y los gustos de su propietario y actuase en consecuencia.

# Contexto Histórico

---

2005 Google compra la compañía y  
el proyecto y deciden poner todo su  
esfuerzo para lanzar un buen  
producto al mercado



# Android 1.0

Alpha  
Apple Pie

Primera versión de Android

2008 salió al mercado el  
HTC Dream, el primer  
smartphone con el sistema  
operativo Android 1.0

Apple Pie

---



# Android 1.1

Beta  
Banana Bread

Febrero 2009 una versión  
dedicada a arreglar fallos  
y bugs de Android 1.0  
Apple Pie.

---



# Android 1.5

Cupcake

Abril 2009

Teclado táctil desplegable  
QWERTY con predicción de  
escritura en la pantalla

Bluetooth

**SDK, el cual permitió el  
desarrollo de aplicaciones  
por parte de terceros.**

— — —



# Android 1.6

Donut

Septiembre 2009

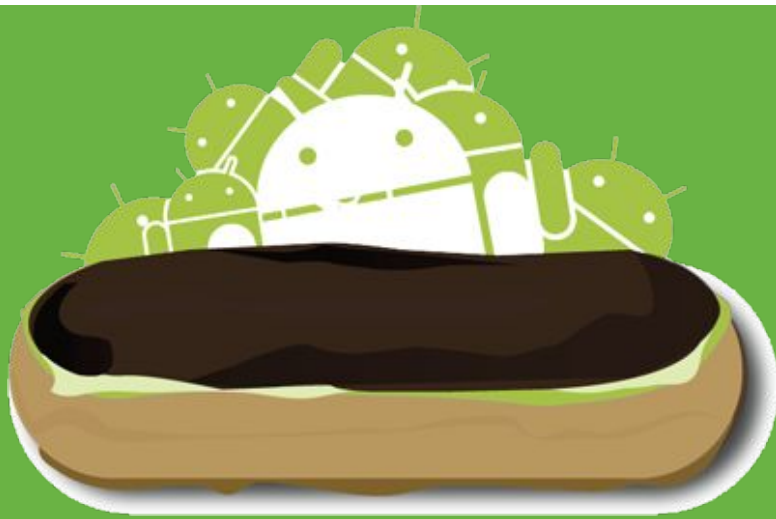
Cuadro de búsqueda rápida



Diversidad de tamaños de pantalla

Android Market



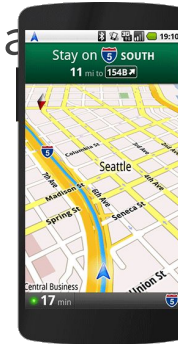


# Android 2.0

Eclair

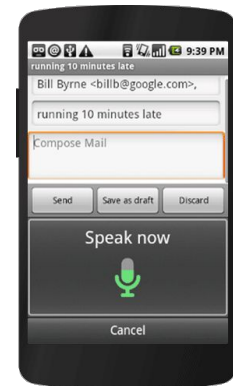
Noviembre 2009

Google Maps Navigation



Pantalla de inicio personalizada

Dictado por voz





# Android 2.2

Eclair

Mayo 2010

Mejora en las funciones  
por voz



Zona Wifi portatil

Mejoras en rendimiento

Dalvik JIT

Motor JavaScript V8





# Android 2.3

Gingerbread

Diciembre 2010

Creación de videojuegos,  
pues se tiene acceso  
completo a gráficos,  
controles del dispositivo  
y audio

Implementó sensor NFC

Gestión de Batería

---



# Android 3.0

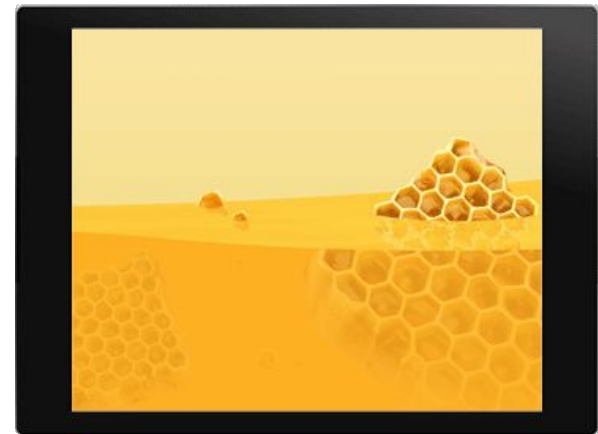
Honeycomb

Febrero 2011

Diseño para Tablet

Botones touch

Botón multitask





# Android 4.0

Ice Cream Sandwich

Octubre 2011

Pantalla inicio personalizada, carpetas para aplicaciones

Control de uso de datos





# Android 4.1

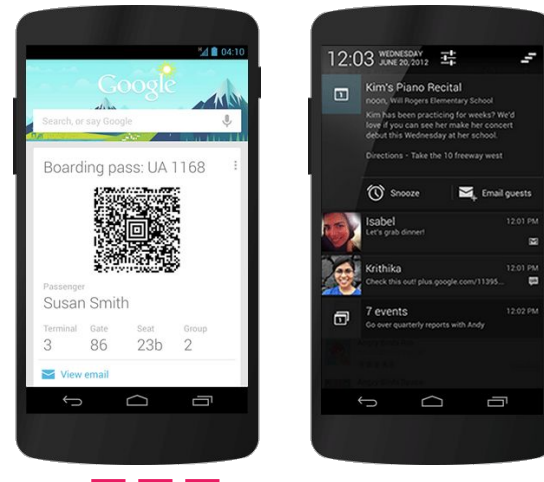
Jelly Bean

Julio 2012

Google Now

Notificaciones  
Desplegables

Multiusuario





# Android 4.4

Kit Kat

Octubre 2013

Ok Google

Teléfono Inteligente  
aprender de tus  
preferencias

Contactos

Llamadas

Google Maps

— — —



# Android 5.0

Lollipop

Junio 2014

Material Design

Multipantalla

Notificaciones en la  
pantalla de bloqueo





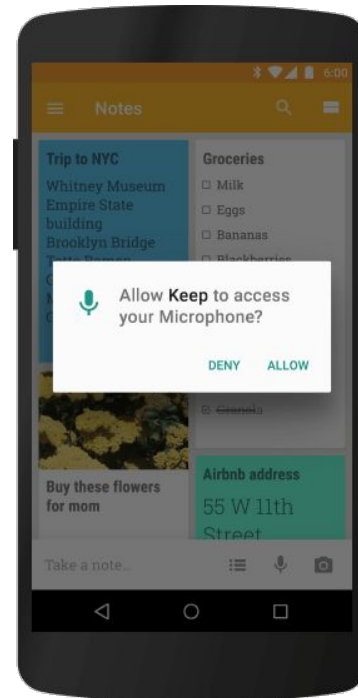
# Android 6.0

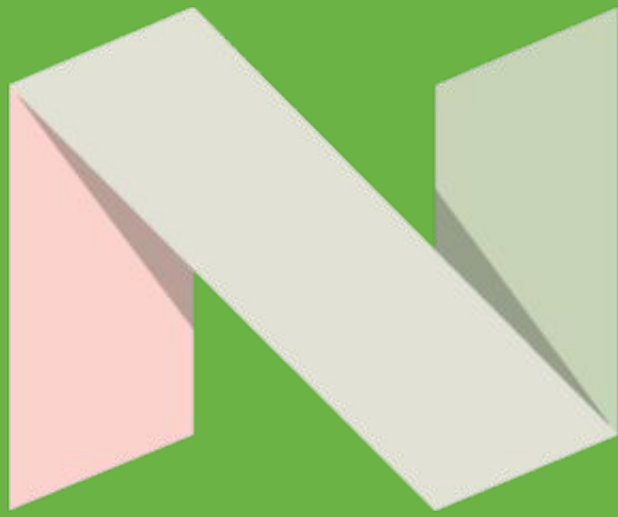
Marshmallow

Agosto 2015

Permisos

Batería





# Android 7.0

Nougat

Mayo 2016

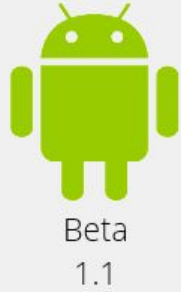
Incremento:

Lenguaje

Notificaciones

Seguridad

---



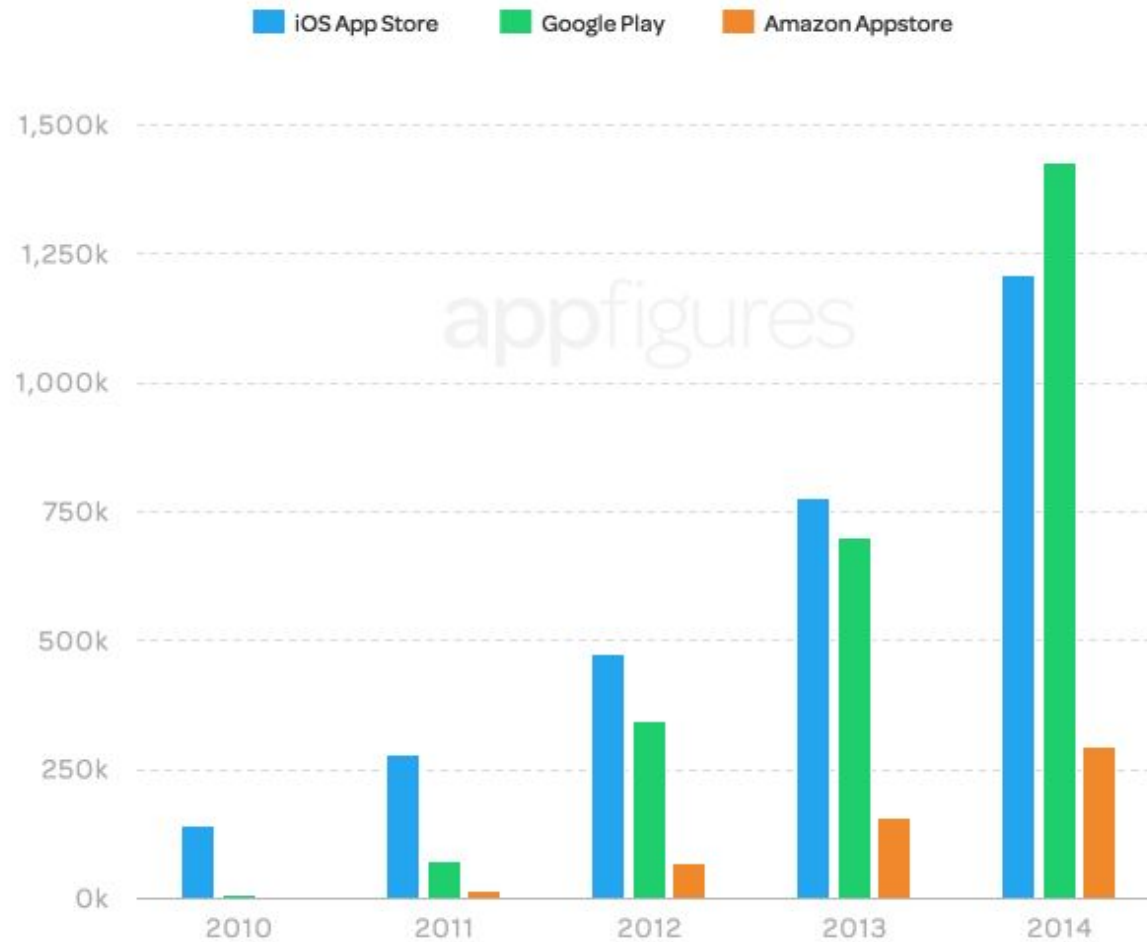
?

**Estado actual**

**1000 millones**

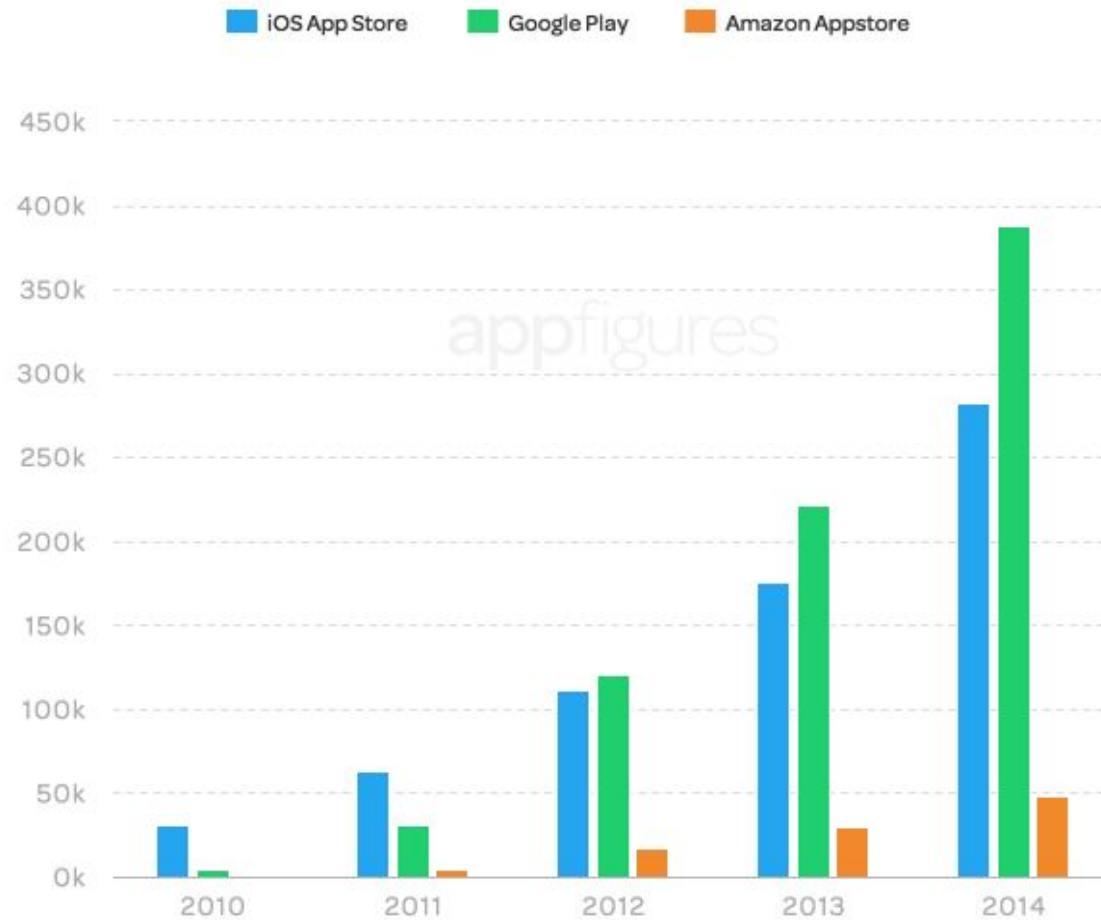
**--- usuarios**

## Total Number of Apps by App Store



**Más aplicaciones**

## Total Number of Developers by App Store



**Mayor número de desarrolladores**

# Requerimientos para desarrollar

# Requerimientos para desarrollar SISTEMA

<https://developer.android.com/studio/index.html#Requirements>

# Requerimientos para desarrollar SISTEMA

— — —

## Windows

Microsoft® Windows® 7/8/10 (32 o 64 bits)

## Mac

Mac® OS X® 10.8.5 o versiones posteriores hasta 10.11.4 (El Capitan).

## Linux

GNOME o KDE de escritorio.

Distribución de 64 bits capaz de ejecutar aplicaciones de 32 bits.

# Requerimientos para desarrollar SISTEMA

— — —

- 2 GB de memoria RAM como mínimo; se recomiendan 8.
- 2 GB de espacio mínimo disponible en el disco;
- Resolución de pantalla mínima de 1280 x 800.
- Para el emulador acelerado: Sistema operativo de 64 bits y procesador Intel® compatible con Intel® VT-x, Intel® EM64T (Intel® 64) y la funcionalidad Execute Disable (XD) Bit.

# Requerimientos para desarrollar SOFTWARE

- Java
- Android Studio
- SDK Android

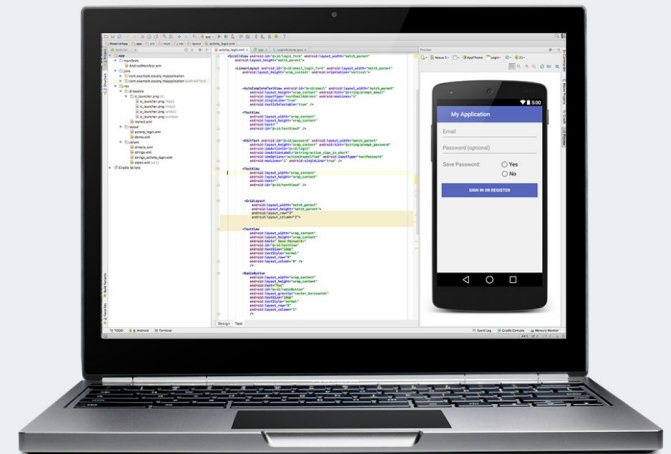
## Android Studio

El IDE oficial para Android

Android Studio proporciona las herramientas más rápidas para la creación de aplicaciones en todos los tipos de dispositivos Android.

La edición de códigos de primer nivel, la depuración, las herramientas de rendimiento, un sistema de compilación flexible y un sistema instantáneo de compilación e implementación te permiten concentrarte en la creación de aplicaciones únicas y de alta calidad.

**CÓMO DESCARGAR ANDROID STUDIO**  
2.1.2.0 FOR MAC (291 MB)



# SDK Android

---

- Versiones de API de Android
- Emuladores para dispositivos virtuales
- SDK Tools

# Android Studio

# Android Studio

---

## Basado en IntelliJ IDEA

- Instant run
- Editor de código inteligente
- Diseño para equipos
- Desarrollo para todos los dispositivos Android

# Android Studio

---

- Instant Run
- Cuando hagas clic en Run o Debug, la función Instant Run de Android Studio aplicará los cambios en el código y los recursos en tu aplicación en ejecución. Esta interpreta de manera inteligente los cambios y a menudo los entrega sin reiniciar tu app ni volver a compilar tu APK, para que puedas ver los efectos de inmediato
- Editor de código inteligente
- Al ofrecer compleción avanzada de código, refactorización y análisis de código, el editor de código inteligente te permite escribir un código más eficaz, trabajar más rápido y ser más productivo. A medida que escribes, Android Studio proporciona sugerencias en una lista desplegable. Simplemente presiona Tab para insertar el código.
- 
- **Diseño para equipos**
- **Android Studio se integra con herramientas de control de versión, como GitHub y Subversion, para que puedas mantener a tu equipo actualizado respecto de los cambios en proyectos y compilaciones. El sistema de compilación de código abierto de Gradle te permite adaptar la compilación**

# Android Studio

— — —

- Desarrollo para todos los dispositivos Android
- Android Studio proporciona un entorno unificado en el cual puedes compilar apps para teléfonos y tablets Android, y para dispositivos Android Wear, Android TV y Android Auto. Los módulos de código estructurado te permiten dividir tu proyecto en unidades de funcionalidad que puedes crear, probar y depurar de manera independiente.
-

# Arquitectura de Android

# Proyecto de Android

— — —

- **Librerías de código**
- **Archivos de recursos y vistas**
- **Código fuente**
- **Android Manifest.**

# Ejecutable APK

— — —

- apk es tipo de empaquetado, comprimido zip
- Contiene archivos .dex
- Resources
  - imagenes
  - layouts
  - etc.

# Ejecutable APK

---

## *App code*

Bytecode  
`classes.dex`

Native code  
`libs/<arch>/*.so`

## *Resources*

`res/`

`resources.arsc`

## *Misc*

`assets/`

`META-INF/`

`AndroidManifest.xml`

# Configuración SDK Manager

# Configuración Emulador

**Creando un “Hola Mundo”**

# Recordando Importancia de POJO

# Plain Old Java Object

---

**“Un Objeto plano y a la  
antigua”**

---

# Java POJO

---

El nombre existe en el marco de la programación orientada a objetos

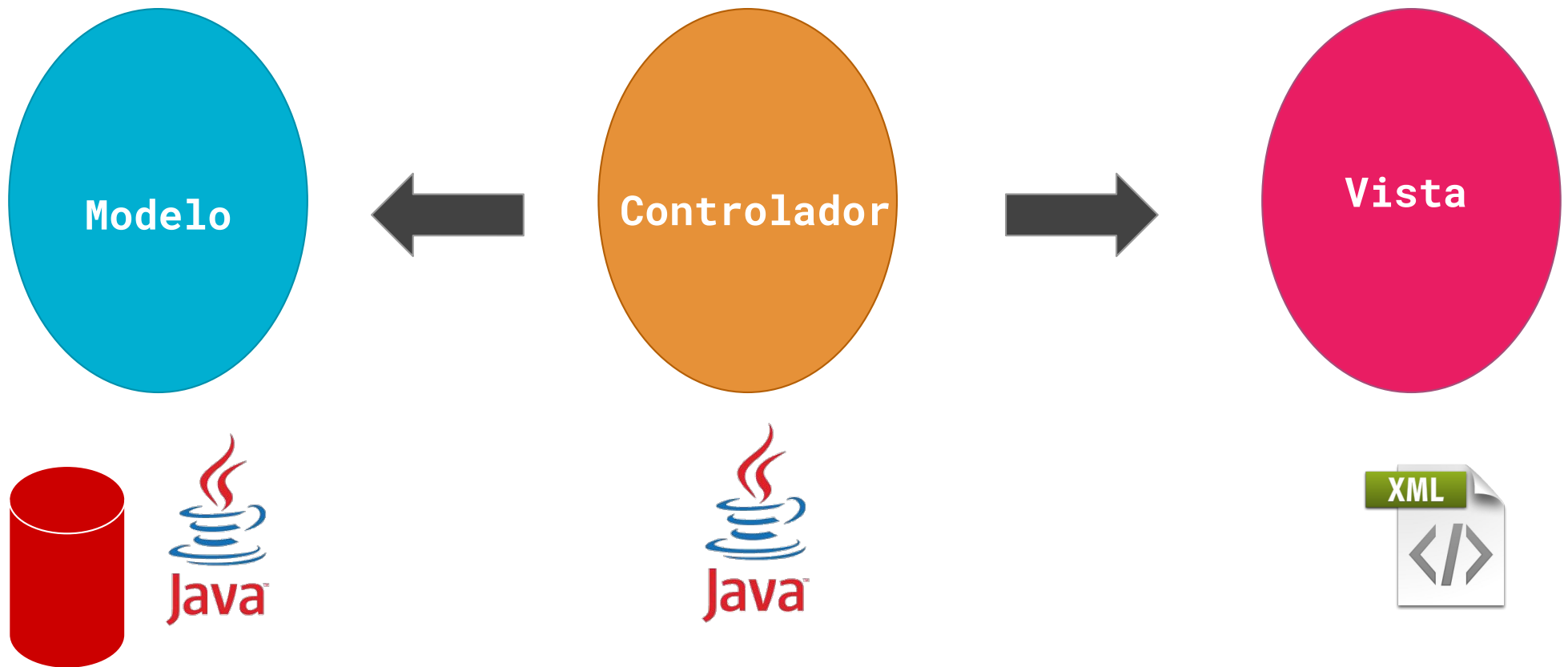
Este es utilizado para describir estructuras de objetos

**Java sirve para enfatizar el uso de clases simples**

**Por qué Android es MVC**

# Android MVC

---

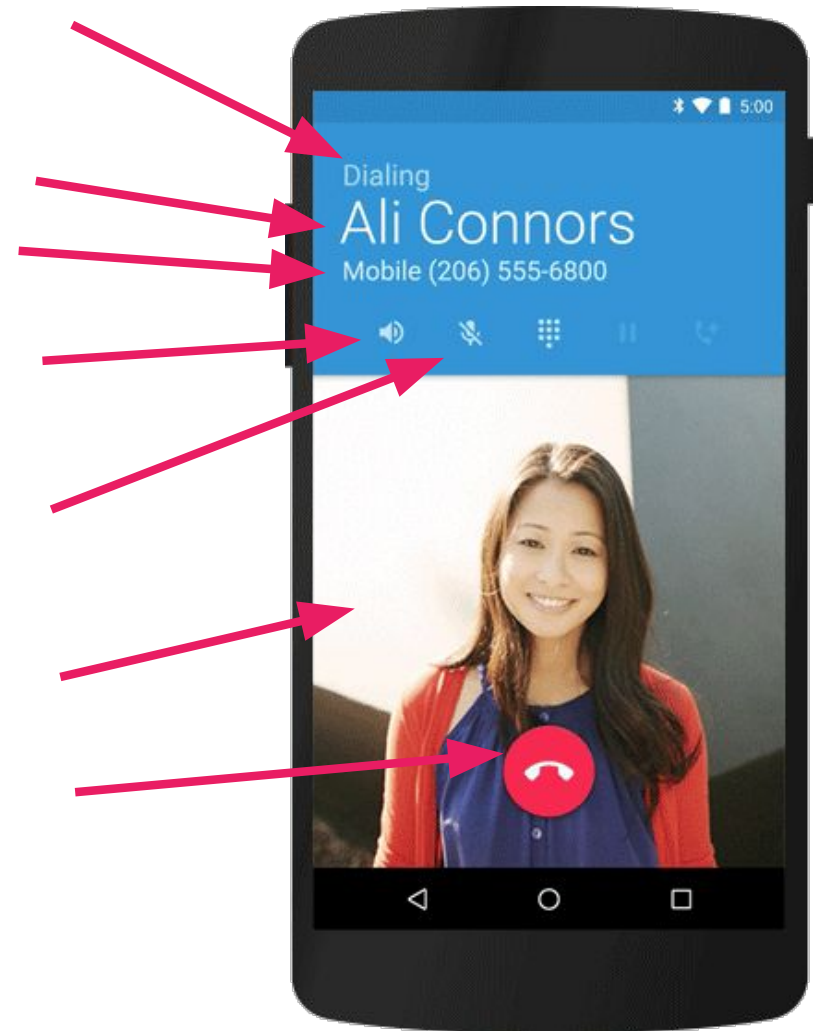


# XML vs Editor Gráfico

**LinearLayout y  
RelativeLayout**

# Views

Un view es un objeto que sirve para dibujar algo en la pantalla con el cual el usuario pueda interactuar



# View

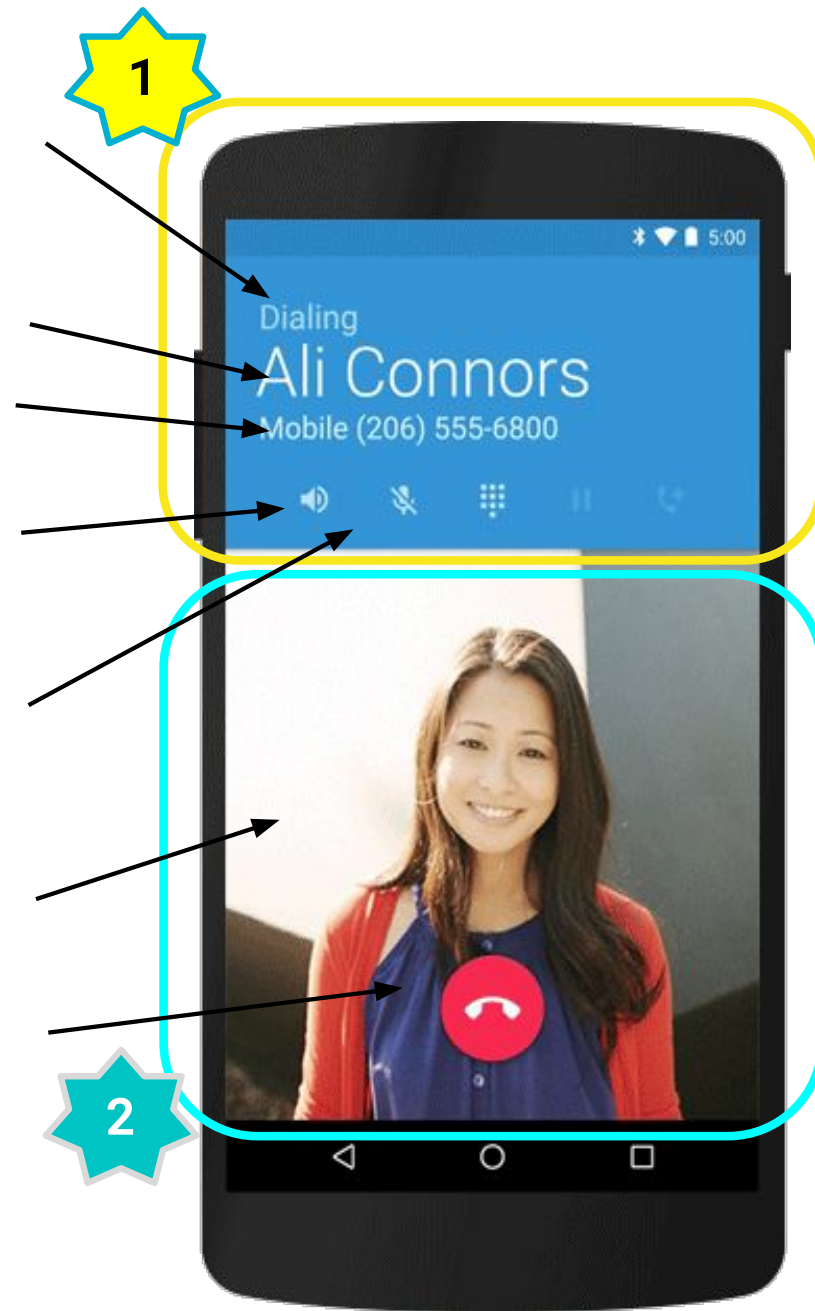
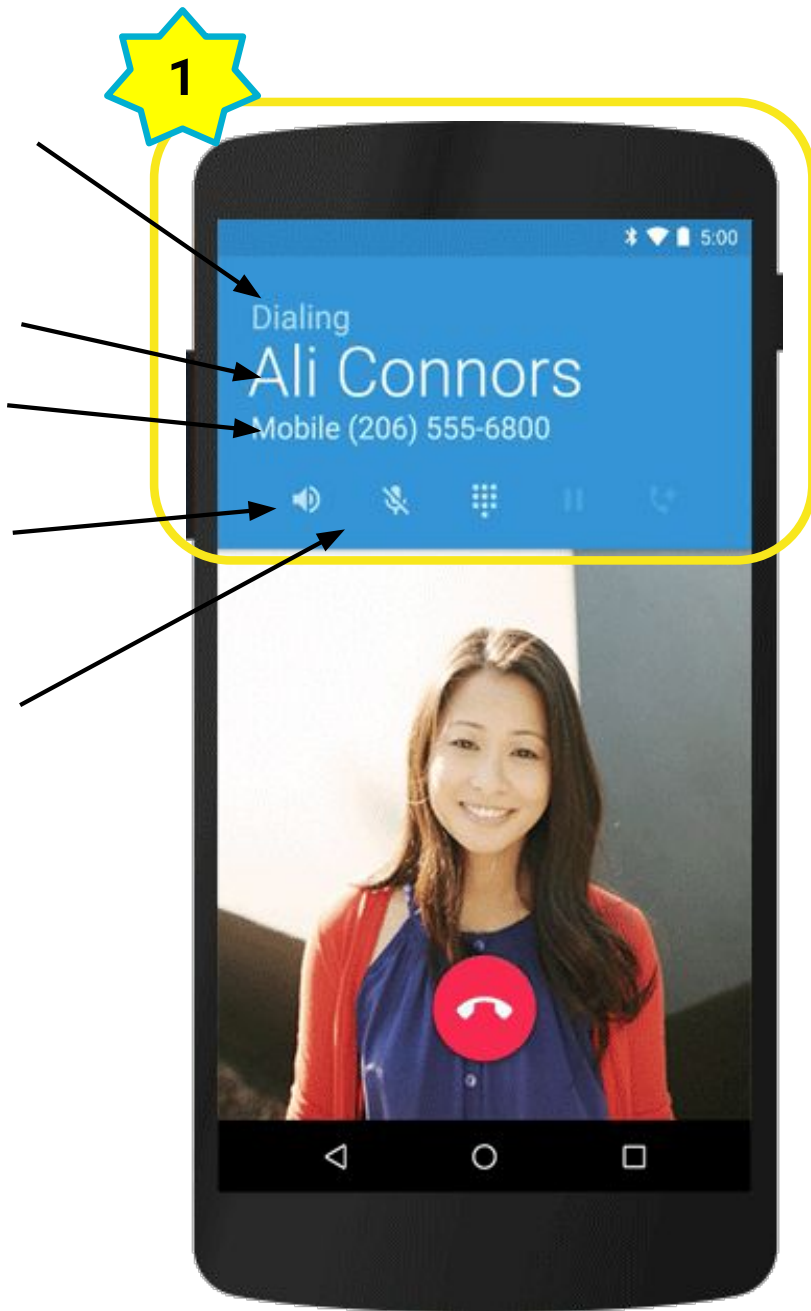
```
public class View
```

```
extends Object implements Drawable.Callback, KeyEvent.Callback,
```

```
java.lang.Object
```

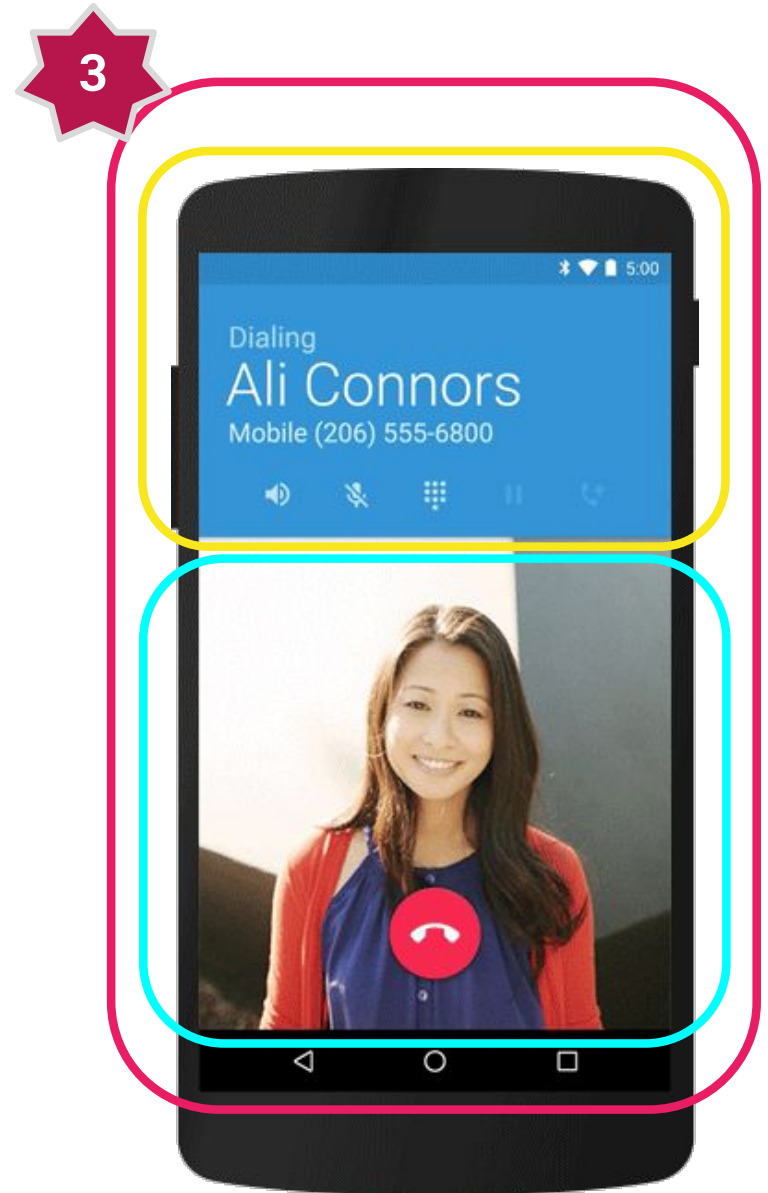
```
↳ android.view.View
```

<https://developer.android.com/reference/android/view/View.html>



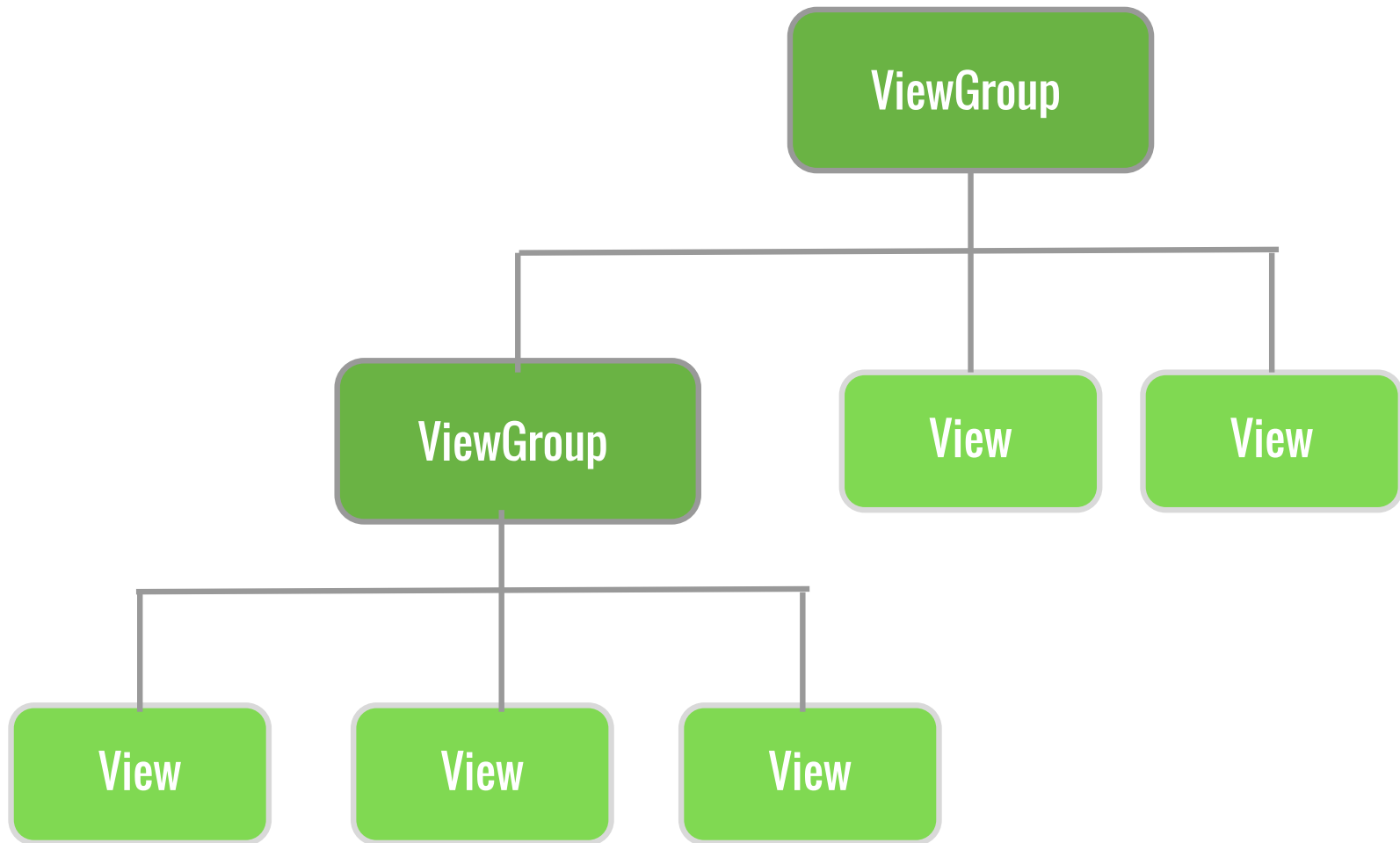
# ViewGroup

Es un objeto que se compone  
de otros Views ordenados,  
dando como resultado un  
**Layout**

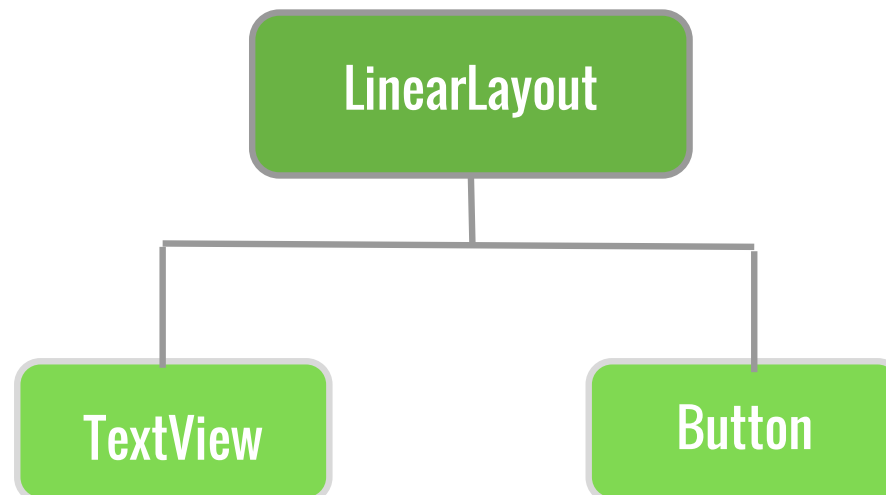


# Layouts

---



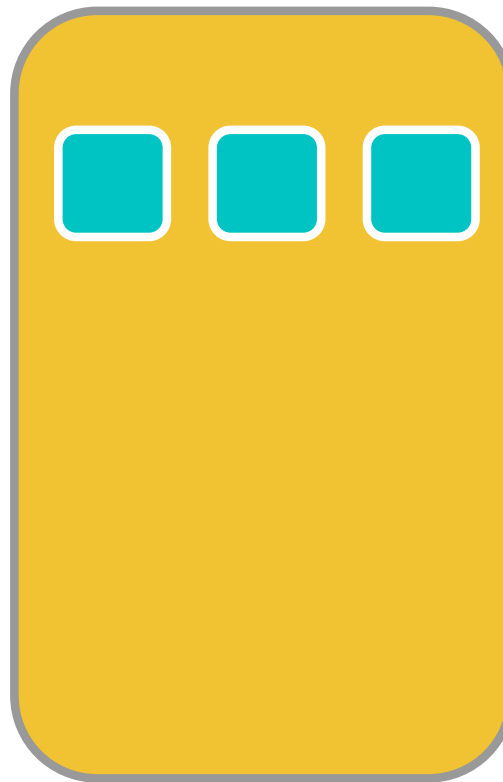
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"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



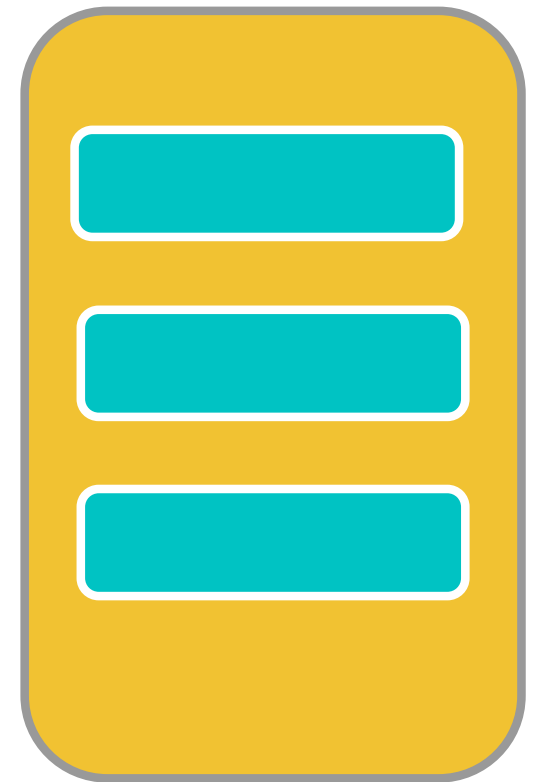
# LinearLayout

---

Es un ViewGroup que  
alinea a todos los  
hijos en una sola  
dirección **Vertical** u  
**Horizontal**



**Horizontal**



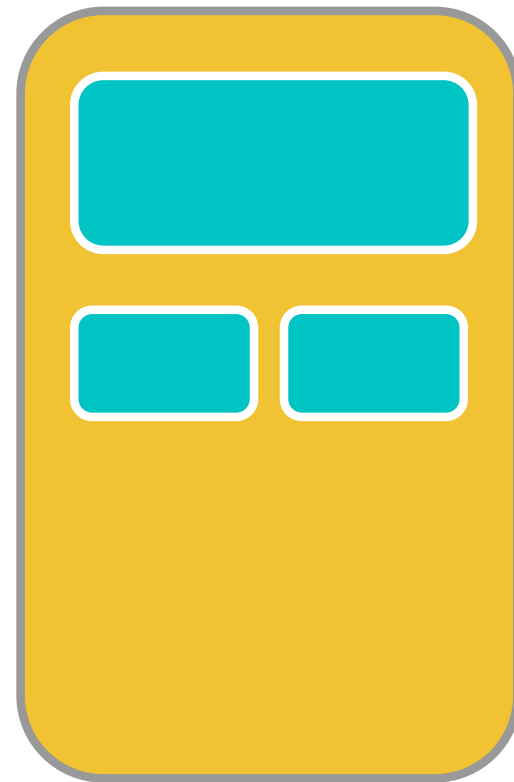
**Vertical**

# RelativeLayout

---

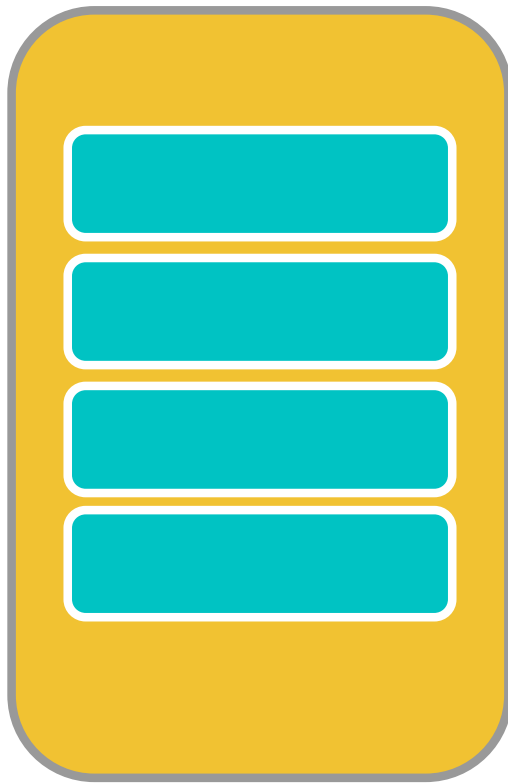
Es un ViewGroup que alinea a todos los hijos en posiciones relativas

- View
- Padre

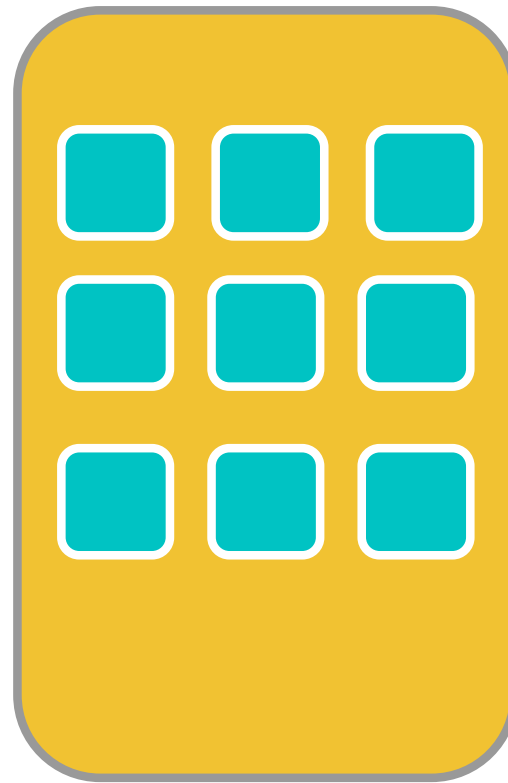


# Más Layouts

---



**ListView**

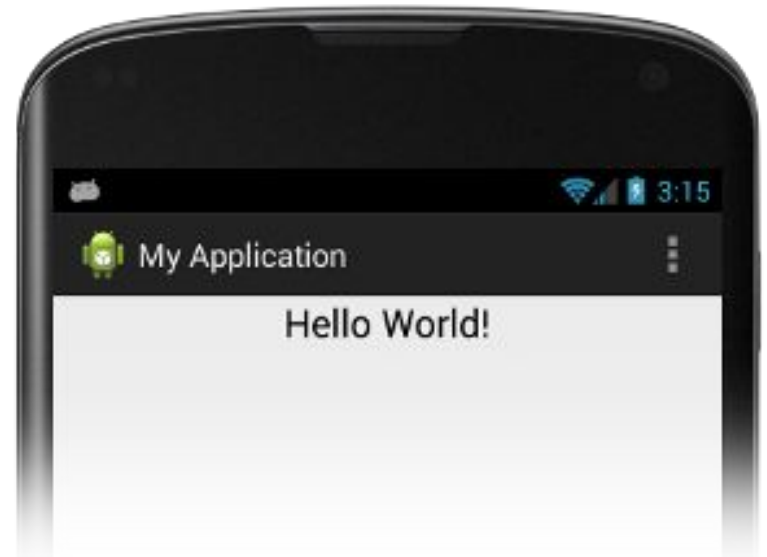


**GridView**

# Widgets básicos

# TextView

Una etiqueta de texto



```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Hello World!" />
```

---

# TextView

public class TextView

extends [View](#) implements [ViewTreeObserver.OnPreDrawListener](#)

[java.lang.Object](#)

↳ [android.view.View](#)

↳ android.widget.TextView

<https://developer.android.com/reference/android/widget/TextView.html>

# EditText

Elemento que nos permite  
ingresar diferentes tipos  
de textos



```
<EditText  
    android:id="@+id/email_address"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:hint="@string/email_hint"  
    android:inputType="textEmailAddress" />
```

— — —

# EditText

```
public class EditText  
extends TextView
```

```
java.lang.Object
```

```
↳ android.view.View
```

```
↳ android.widget.TextView
```

```
↳ android.widget.EditText
```

<https://developer.android.com/reference/android/widget/EditText.html>

# Button

Es una porción de texto, imagen o ambos que nos ayudan a establecer comunicación con acciones cuando el usuario lo toca



```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_text"
    ... />

<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/button_icon"
    ... />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_text"
    android:drawableLeft="@drawable/button_icon"
    ... />
```

# Button

public class Button  
extends **TextView**

**java.lang.Object**

↳ **android.view.View**

↳ **android.widget.TextView**

↳ android.widget.Button

<https://developer.android.com/reference/android/widget/Button.html>

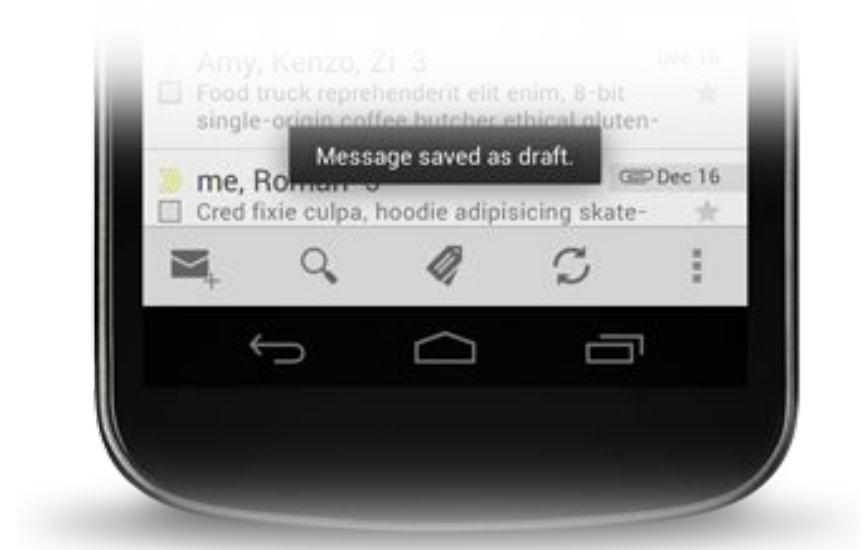
# Button. Actions

---

- `onClick`
- `setOnClickListener`

# Toast

Es un mensaje que nos  
notifica retroalimentación  
sobre alguna acción  
realizada



```
Toast.makeText(context, text, duration).show();
```

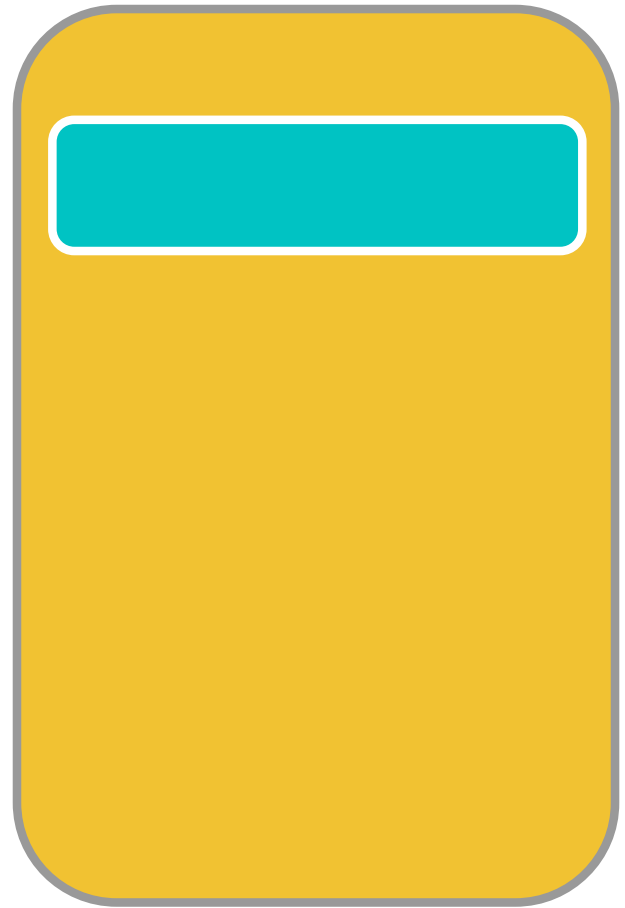
```
Toast toast = new Toast(getApplicationContext());  
toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0);  
toast.setDuration(Toast.LENGTH_LONG);  
toast.setView(layout);  
toast.show();
```

— — —

# Diseño Responsive

# match\_parent

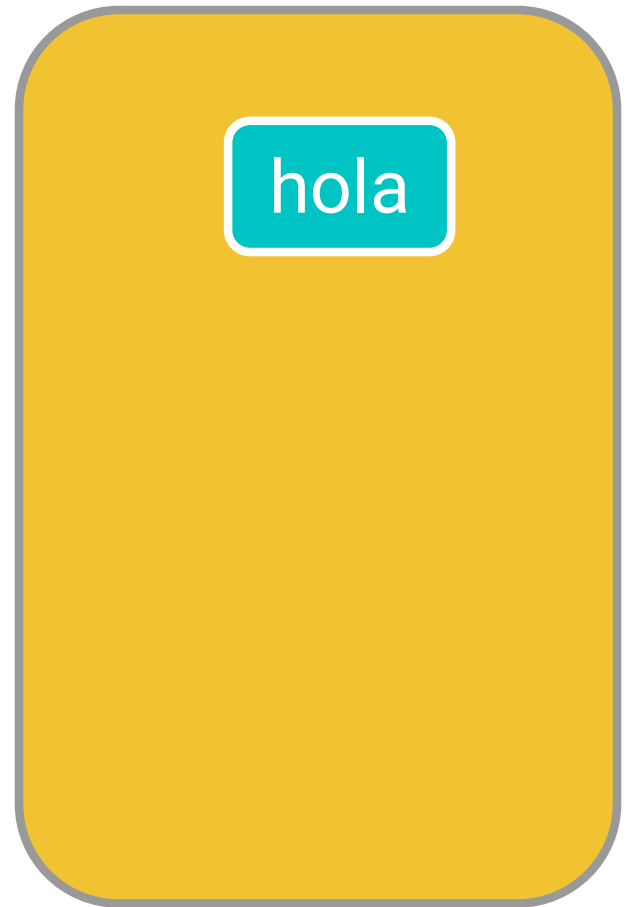
ajusta al padre



---

# wrap\_content

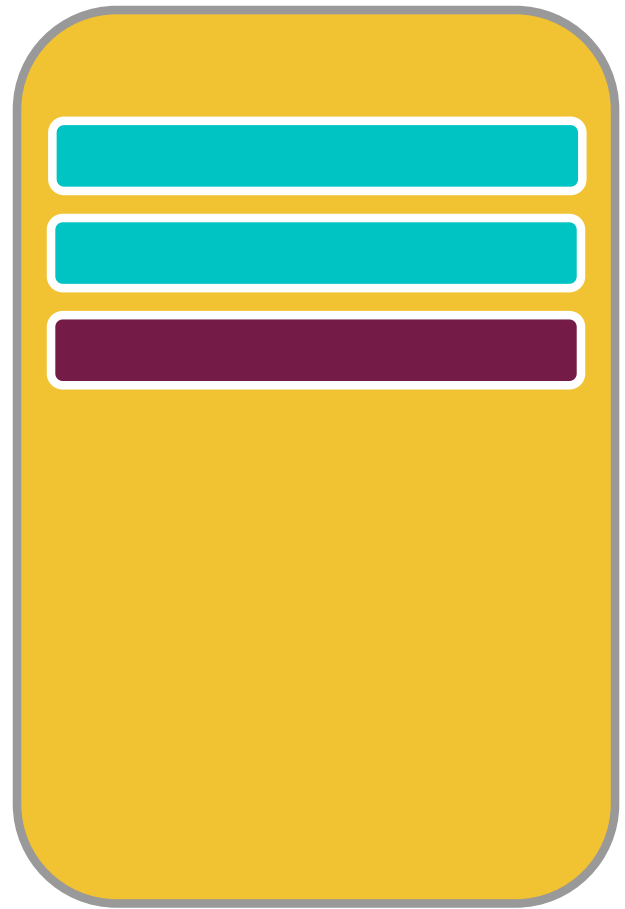
ajusta al contenido



---

# portrait

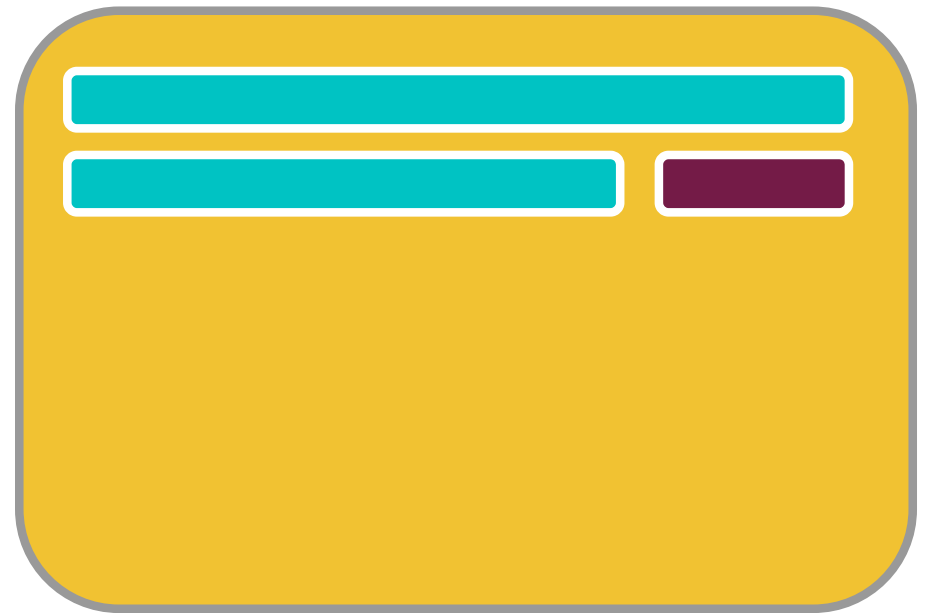
Dispositivo en posición  
vertical



---

# landscape

Dispositivo en posición  
horizontal



---

# large

Dispositivos más grandes



---

# Material Design

# Material Design

Es una guía integral para el diseño visual, de movimientos y de interacción en distintas plataformas y dispositivos



---

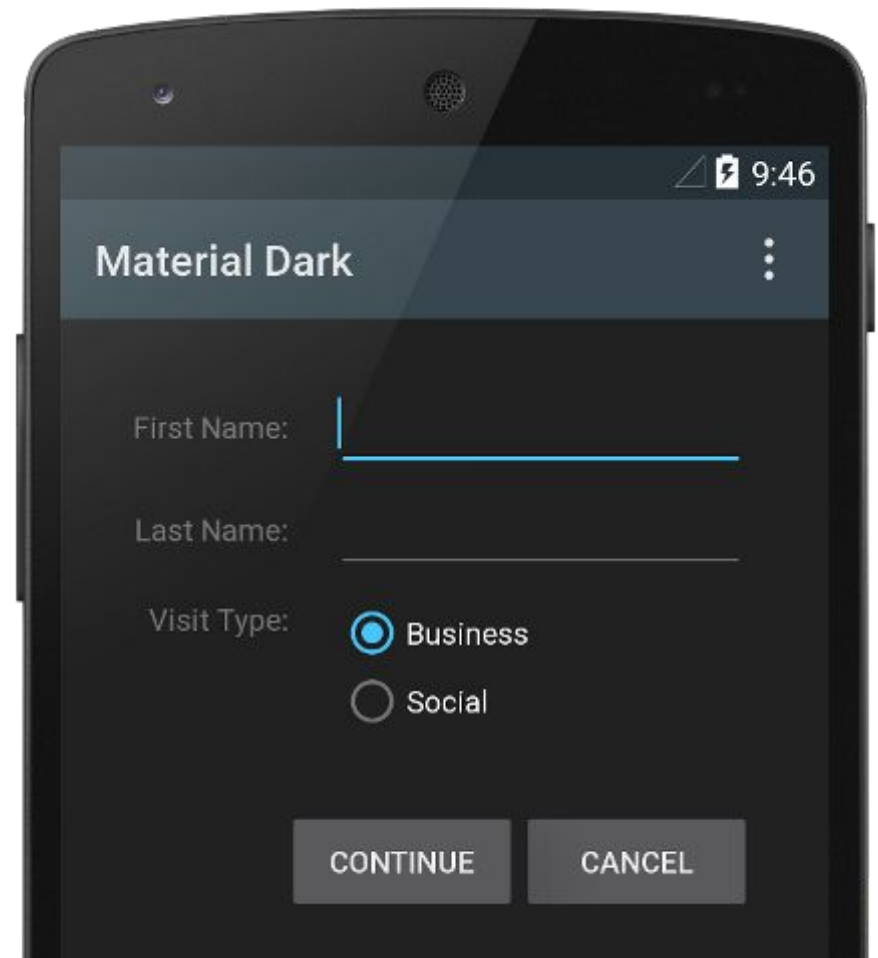
# Material Design

- Tema nuevo
- Nuevos Widgets
- Nuevas librerías para animaciones

---

# Material Design Theme

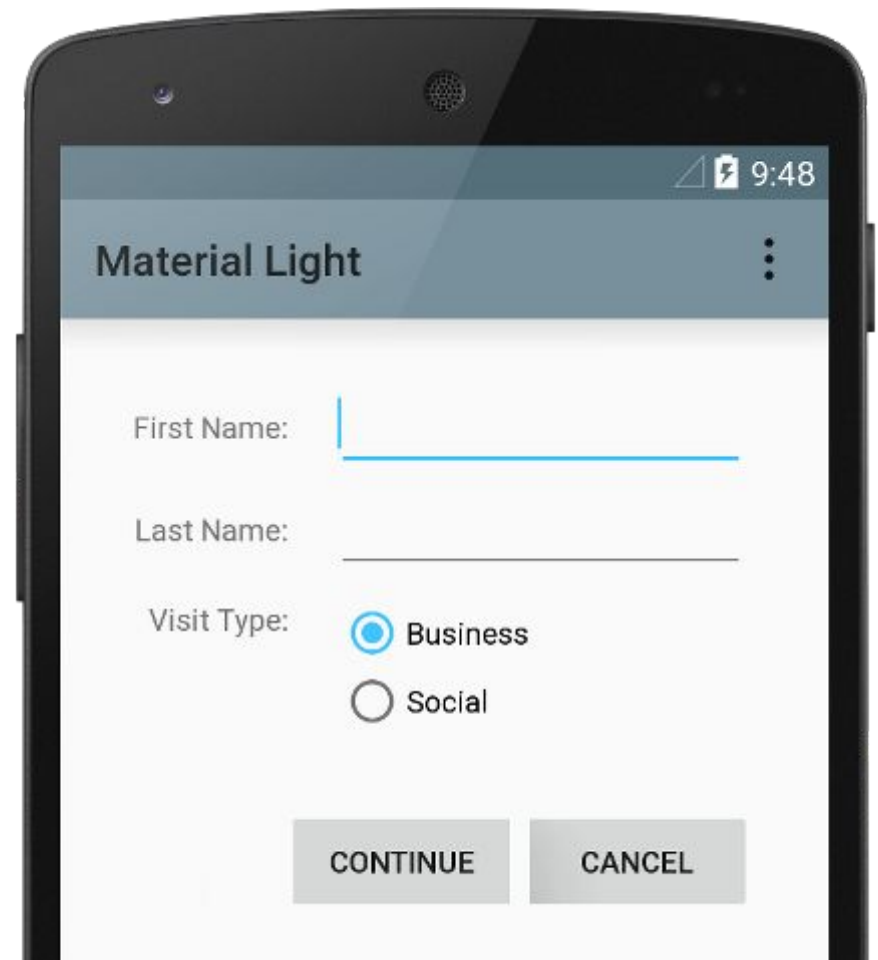
Obscuro



---

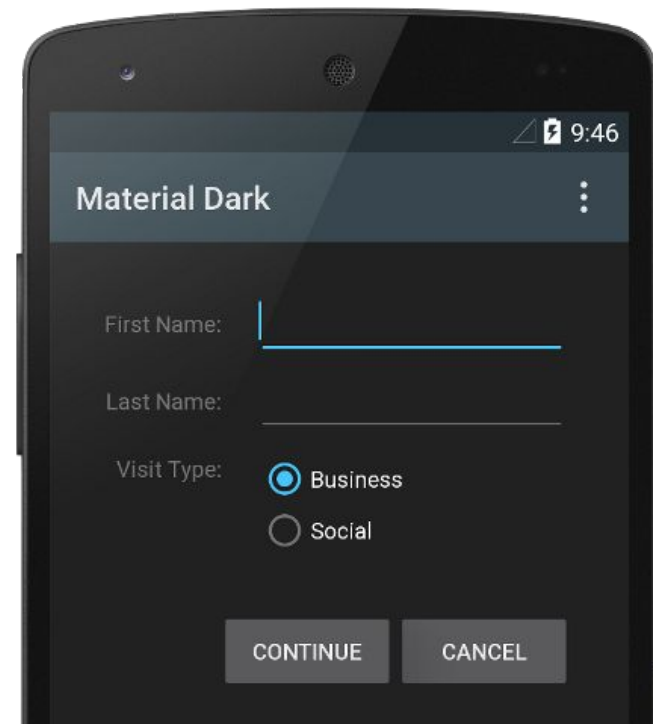
# Material Design Theme

Claro



# Material Design Theme

Obscuro

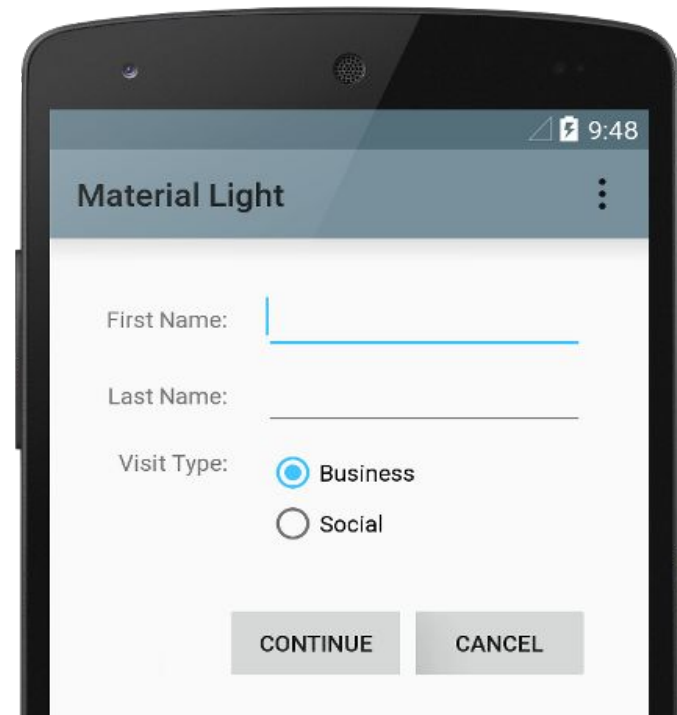


`@android:style/Theme.Material`

---

# Material Design Theme

Claro

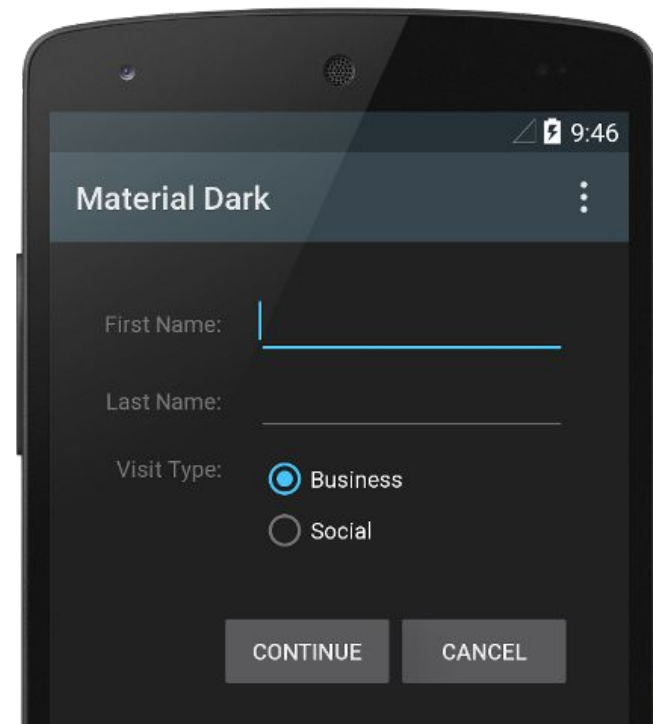


```
@android:style/Theme.Material.Light
```

```
@android:style/Theme.Material.Light.  
DarkActionBar
```

# Material Design Theme

Obscuro

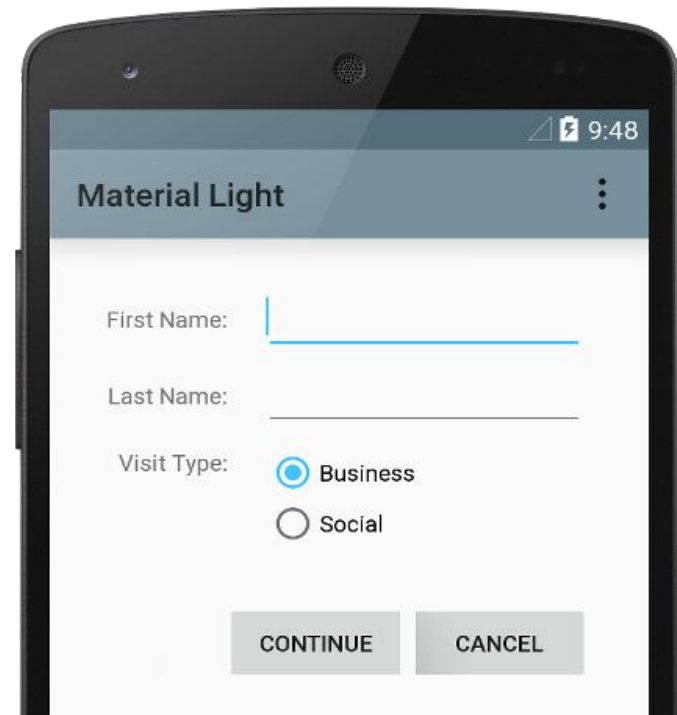


`@android:style/Theme.Material`

---

# Material Design Theme

Claro



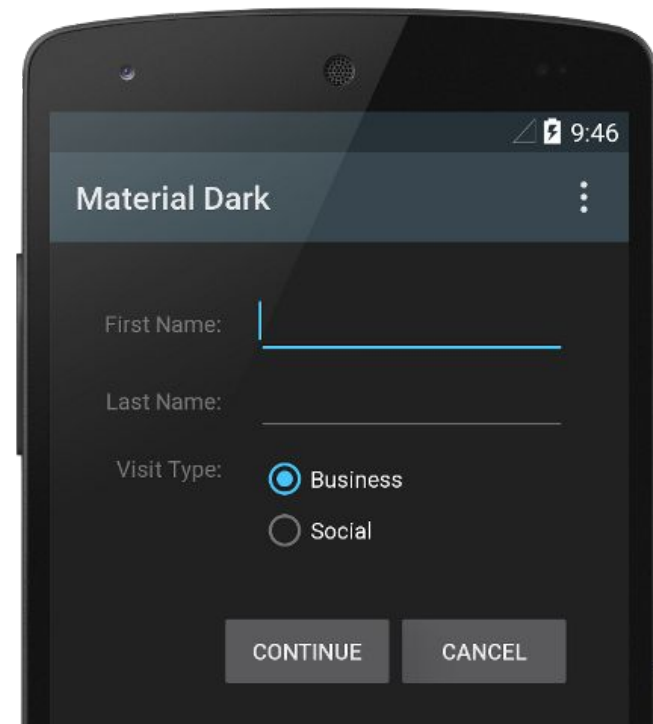
```
@android:style/Theme.Material.Light
```

```
@android:style/Theme.Material.Light.  
DarkActionBar
```

Theme.AppCompat

# Material Design Theme

Obscuro

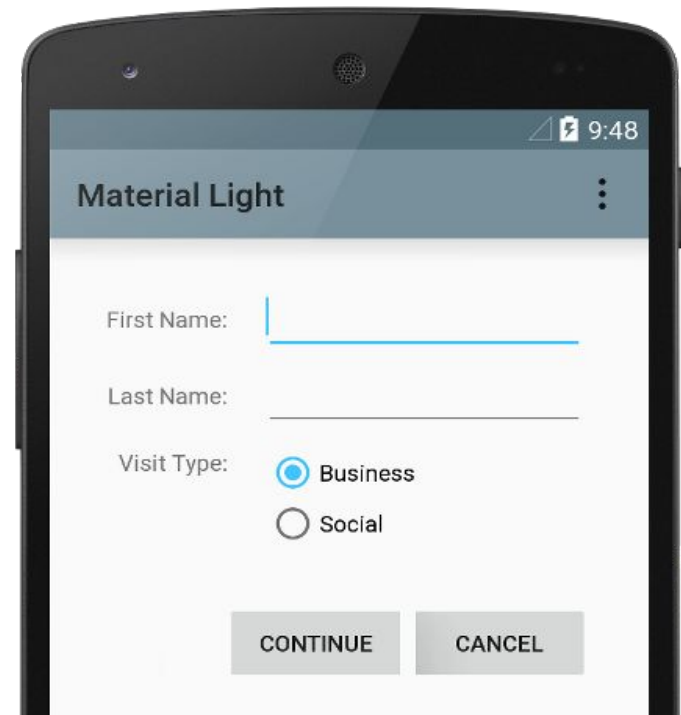


Theme.Material

---

# Material Design Theme

Claro



`Theme.AppCompat.Light`

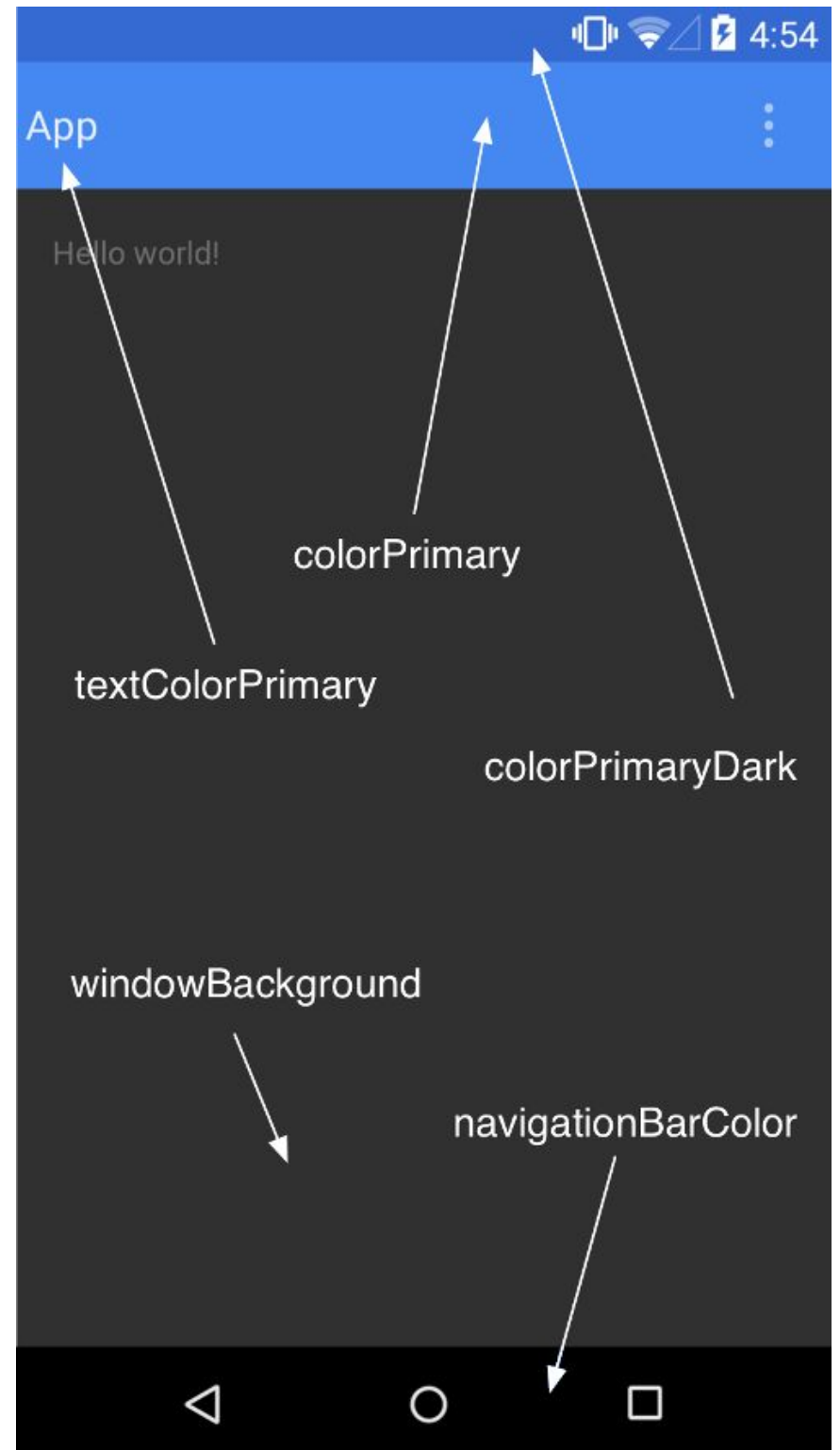
`Theme.AppCompat.Light.DarkActionBar`

**styles.xml**

---

# Theme configuration

colorPrimary  
colorPrimaryDark  
colorAccent



Red	
500	#F44336
50	#FFEBEE
100	#FFCDD2
200	#EF9A9A
300	#E57373
400	#EF5350
500	#F44336
600	#E53935
700	#D32F2F
800	#C62828
900	#B71C1C

Pink	
500	#E91E63
50	#FCE4EC
100	#F8BBD0
200	#F48FB1
300	#F06292
400	#EC407A
500	#E91E63
600	#D81B60
700	#C2185B
800	#AD1457
900	#880E4F

Purple	
500	#9C27B0
50	#F3E5F5
100	#E1BEE7
200	#CE93D8
300	#BA68C8
400	#AB47BC
500	#9C27B0
600	#8E24AA
700	#7B1FA2
800	#6A1B9A
900	#4A148C

<https://material.google.com/style/color.html>

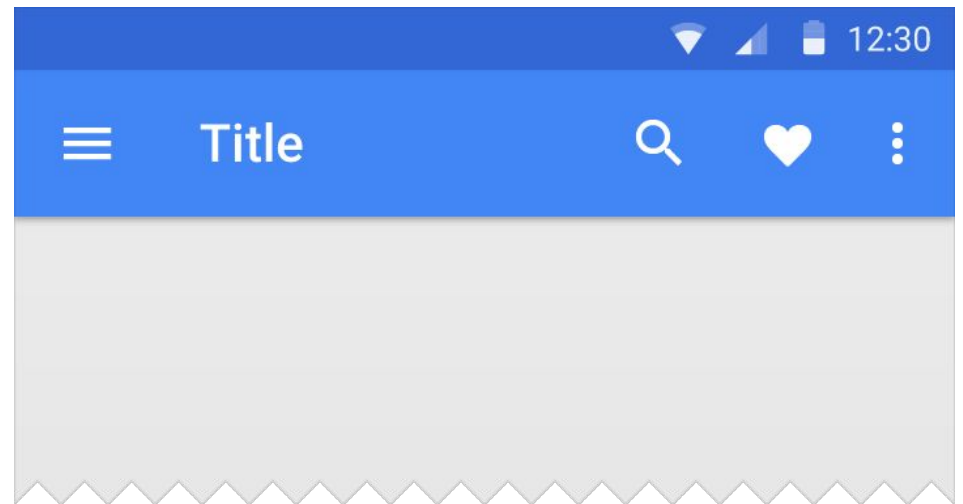
[materialpalette.com](http://materialpalette.com)

# colors.xml

---

# Widgets Material Design

# Toolbar



# Toolbar

public class Toolbar

extends **ViewGroup**

**java.lang.Object**

↳ **android.view.View**

↳ **android.view.ViewGroup**

↳ android.support.v7.widget.Toolbar

<https://developer.android.com/reference/android/support/v7/widget/Toolbar.html>

# TextInputLayout TextInputEditText

Account data

Serverlogin

Password

ABBRECHEN OK

---

# TextInputEditText

```
public class TextInputEditText  
extends AppCompatActivity
```

```
java.lang.Object
```

```
↳ android.view.View
```

```
↳ android.widget.TextView
```

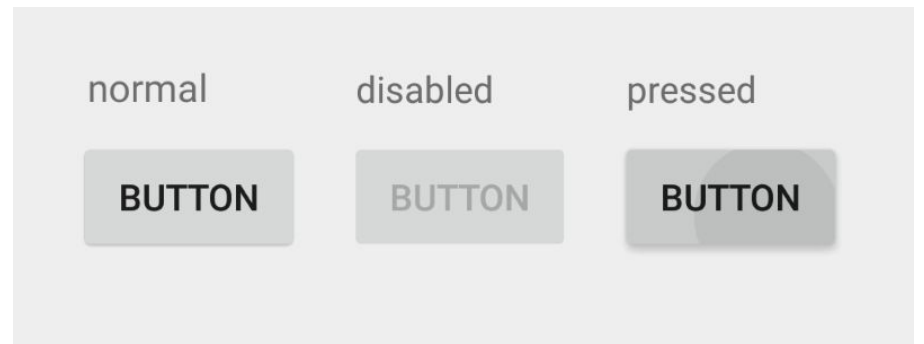
```
↳ android.widget.EditText
```

```
↳ android.support.v7.widget.AppCompatActivity
```

```
↳ android.support.design.widget.TextInputEditText
```

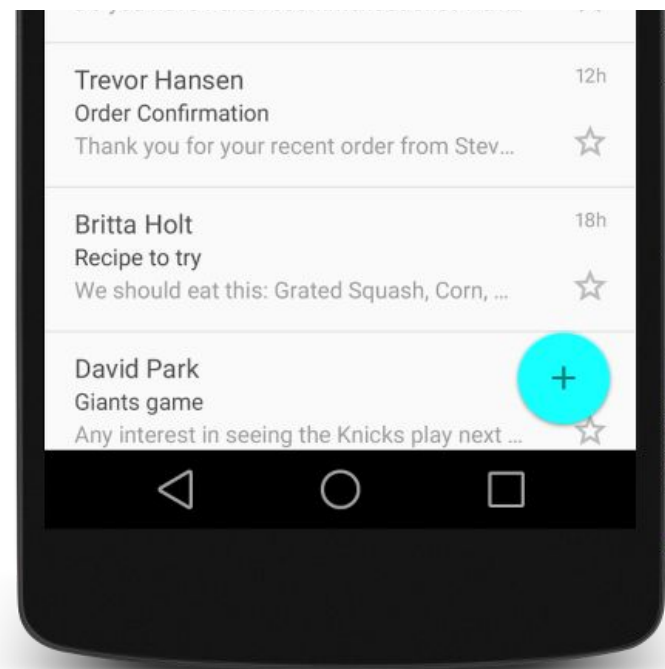
<https://developer.android.com/reference/android/support/design/widget/TextInputEditText.html>

# Raised Button



---

# Floating Action Button



---

# FloatingActionButton

```
public class FloatingActionButton
```

```
extends ImageButton
```

```
java.lang.Object
```

```
↳ android.view.View
```

```
↳ android.widget.ImageView
```

```
↳ android.widget.ImageButton
```

```
↳ android.support.design.widget.FloatingActionButton
```

<https://developer.android.com/reference/android/support/design/widget/FloatingActionButton.html>

# CardView



---

# CardView

public class CardView

extends [FrameLayout](#)

[java.lang.Object](#)

↳ [android.view.View](#)

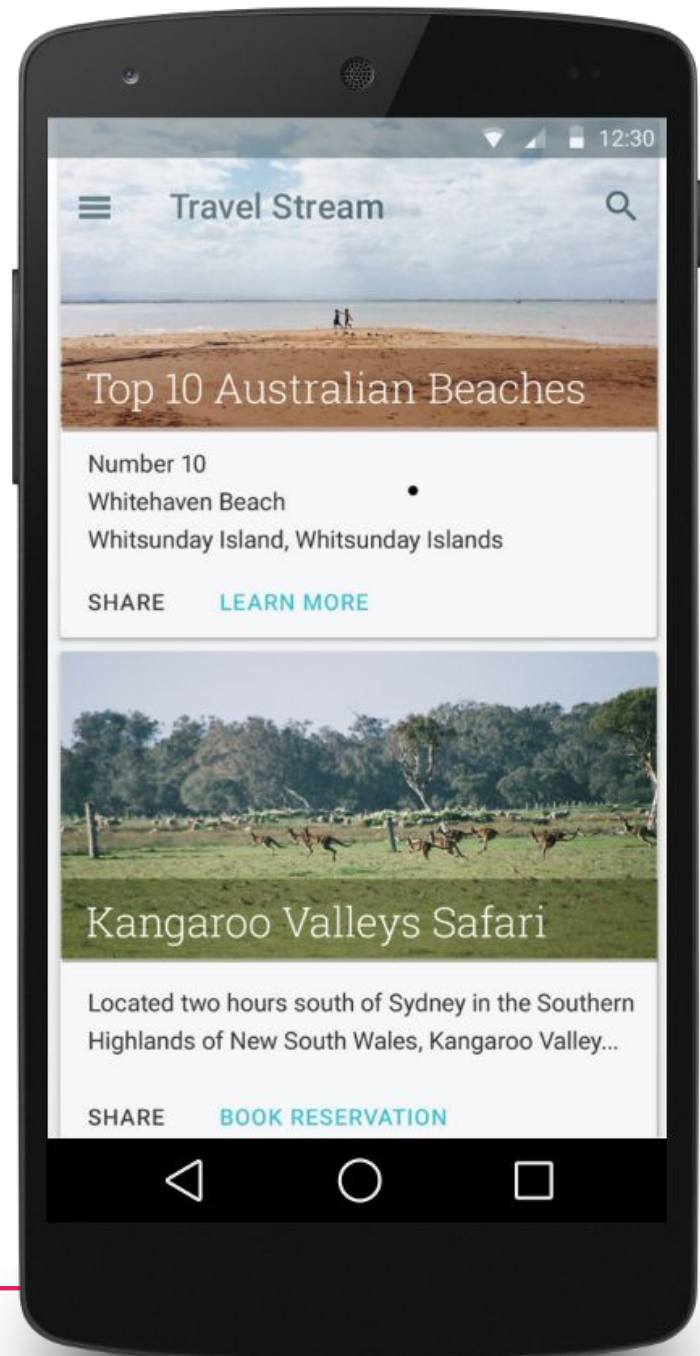
↳ [android.view.ViewGroup](#)

↳ [android.widget.FrameLayout](#)

↳ [android.support.v7.widget.CardView](#)

<https://developer.android.com/reference/android/support/v7/widget/CardView.html>

# RecyclerView



# RecyclerView

public class RecyclerView

extends **ViewGroup** implements **ScrollingView**, **NestedScrollingChild**

**java.lang.Object**

↳ **android.view.View**

↳ **android.view.ViewGroup**

↳ android.support.v7.widget.RecyclerView

▼ Known Direct Subclasses

**HorizontalGridView**, **VerticalGridView**

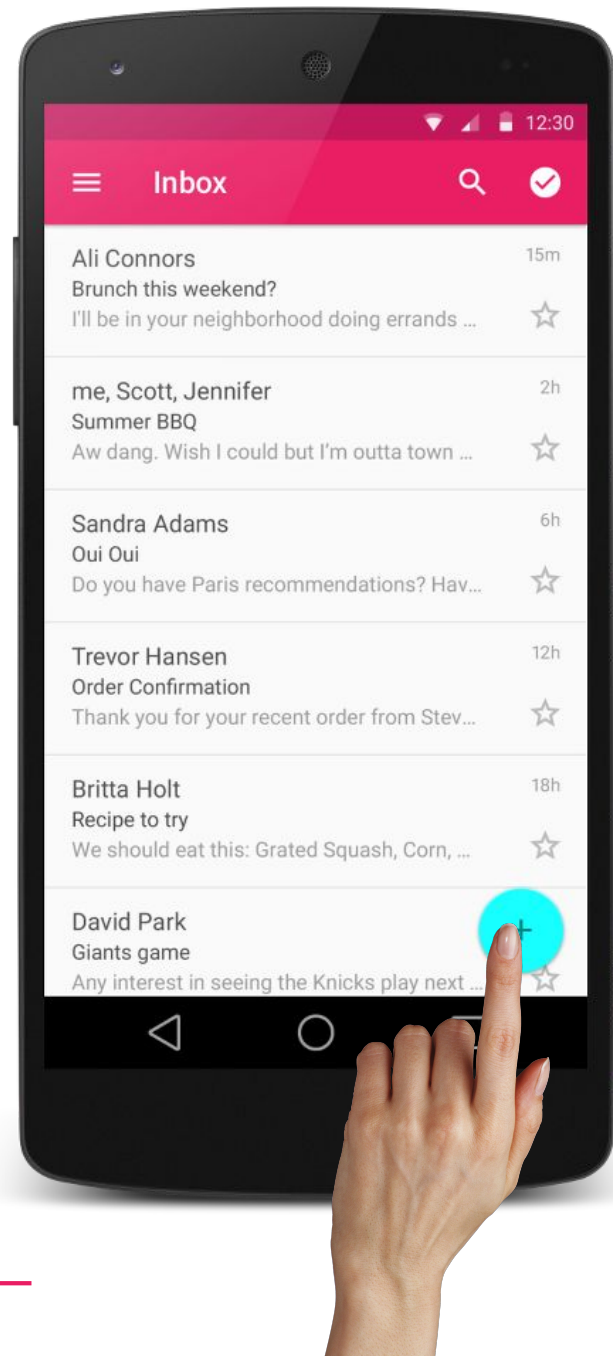
<https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html>

[https://developer.android.com/topic/libraries/support-library/  
---features.html#v7](https://developer.android.com/topic/libraries/support-library/---features.html#v7)

**Activity**

# Activity

Es un componente de una aplicación que nos provee de una pantalla para que el usuario interactúe con ella



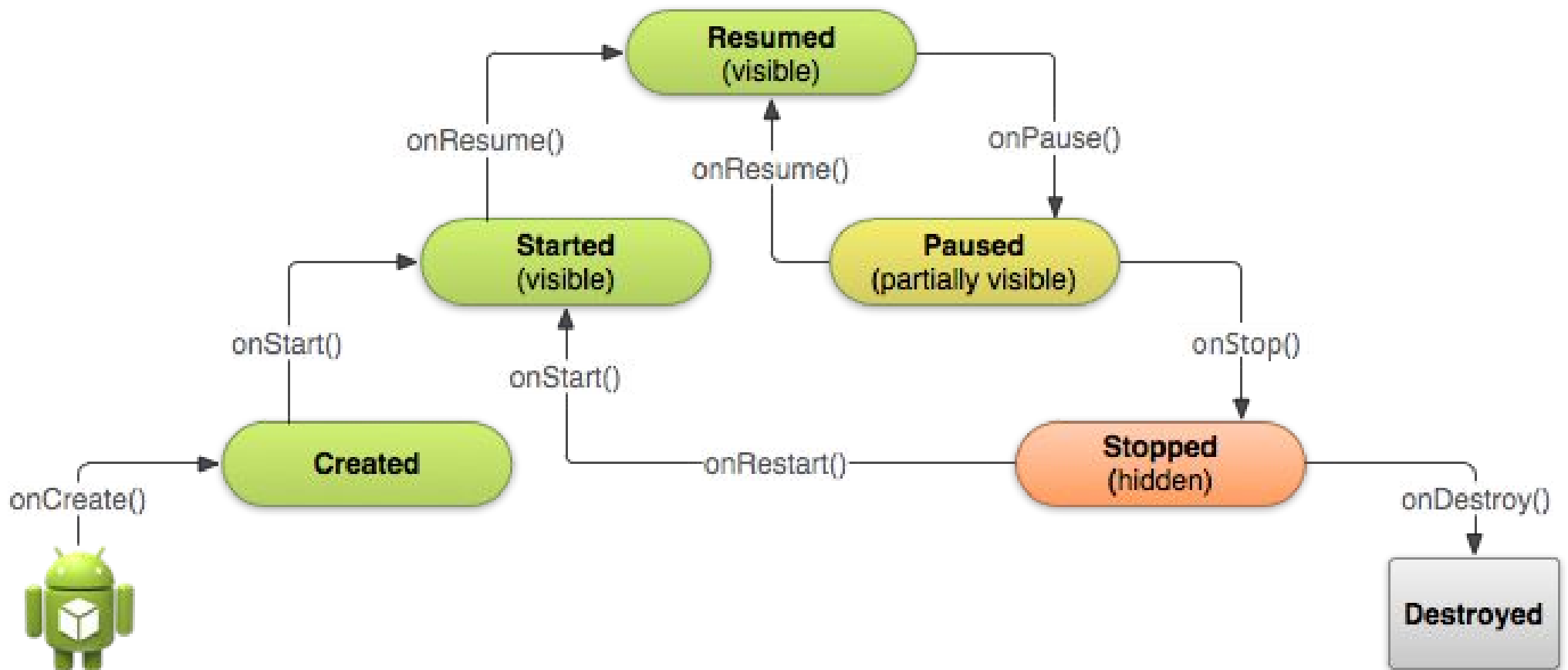
# Activity

---



**AndroidManifest.xml**

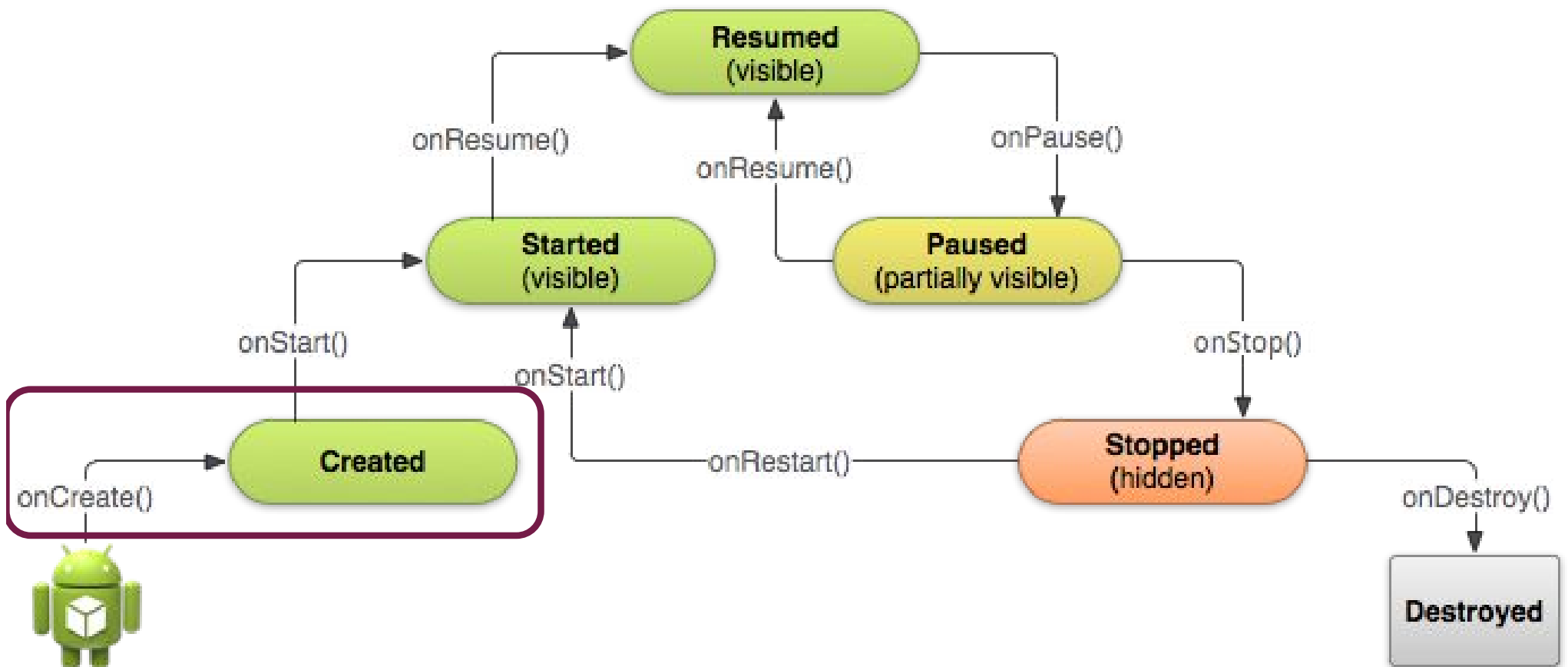
# Ciclo de Vida



```
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about to be "paused").
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}
```

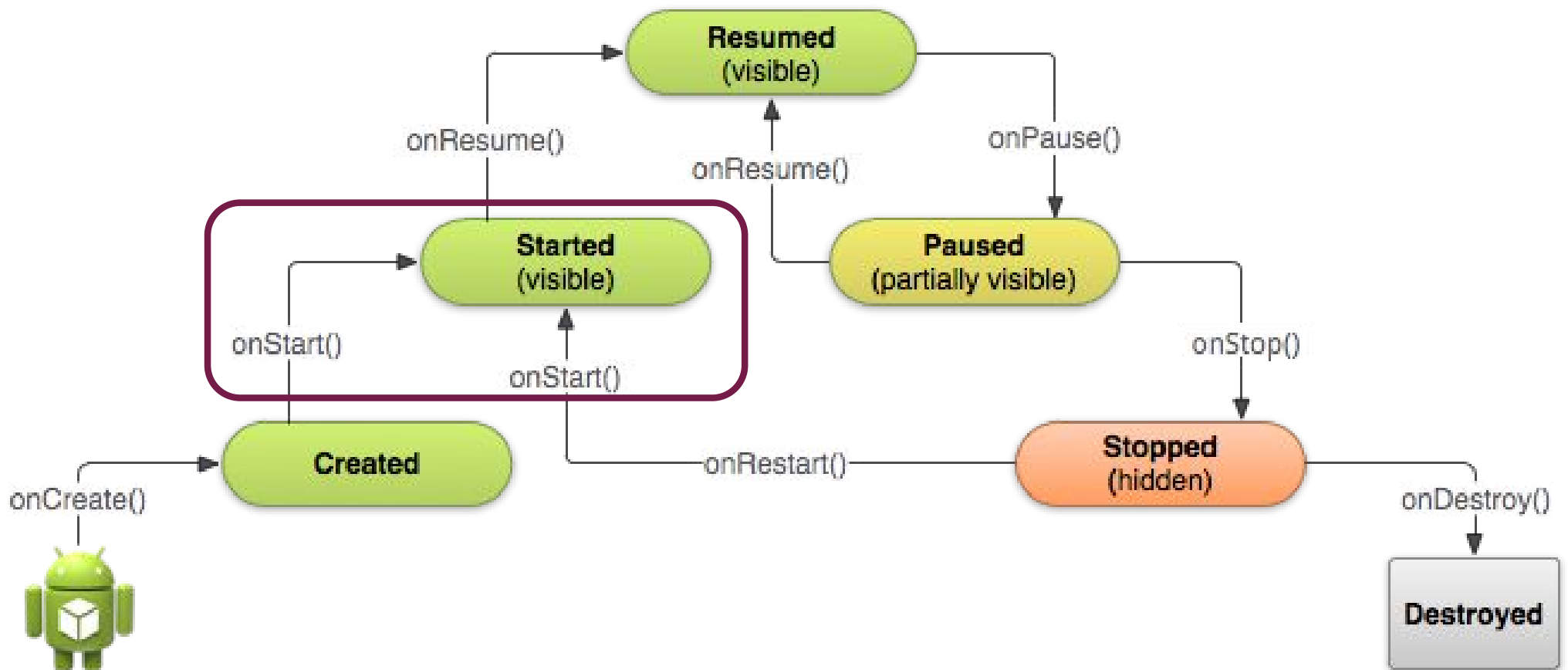
# Ciclo de Vida

---



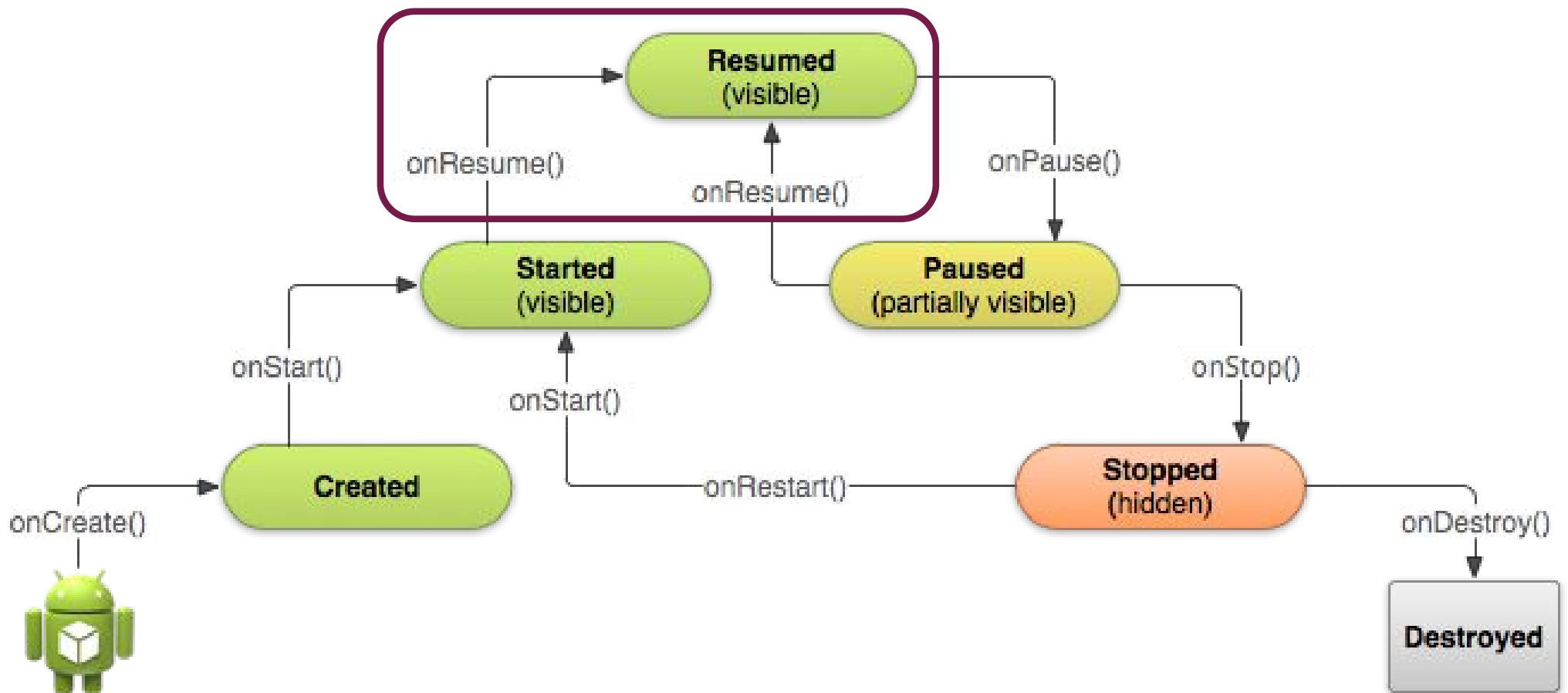
```
public class ExampleActivity extends Activity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        // The activity is being created.  
    }  
    @Override  
    protected void onStart() {  
        super.onStart();  
        // The activity is about to become visible.  
    }  
    @Override  
    protected void onResume() {  
        super.onResume();  
        // The activity has become visible (it is now "resumed").  
    }  
    @Override  
    protected void onPause() {  
        super.onPause();  
        // Another activity is taking focus (this activity is about to be "paused").  
    }  
    @Override  
    protected void onStop() {  
        super.onStop();  
        // The activity is no longer visible (it is now "stopped")  
    }  
    @Override  
    protected void onDestroy() {  
        super.onDestroy();  
        // The activity is about to be destroyed.  
    }  
}
```

# Ciclo de Vida



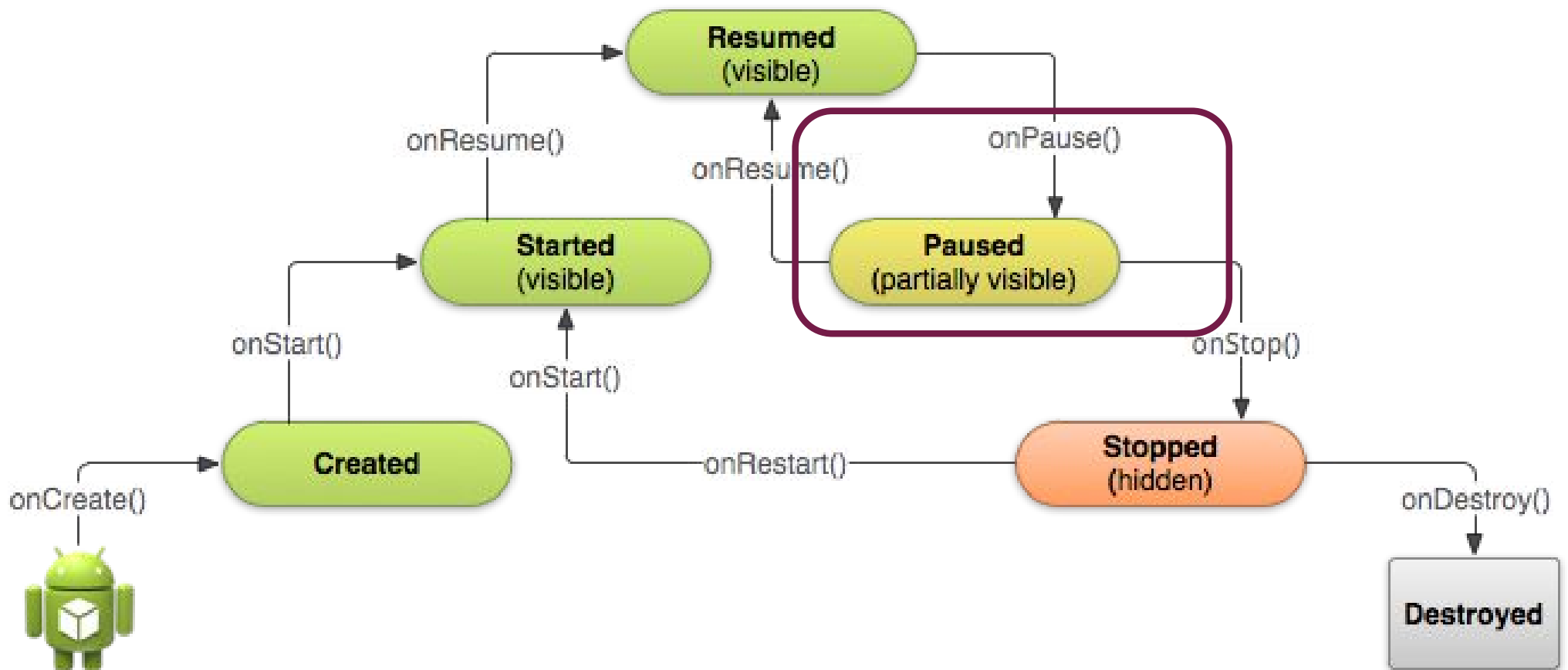
```
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about to be "paused").
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}
```

# Ciclo de Vida



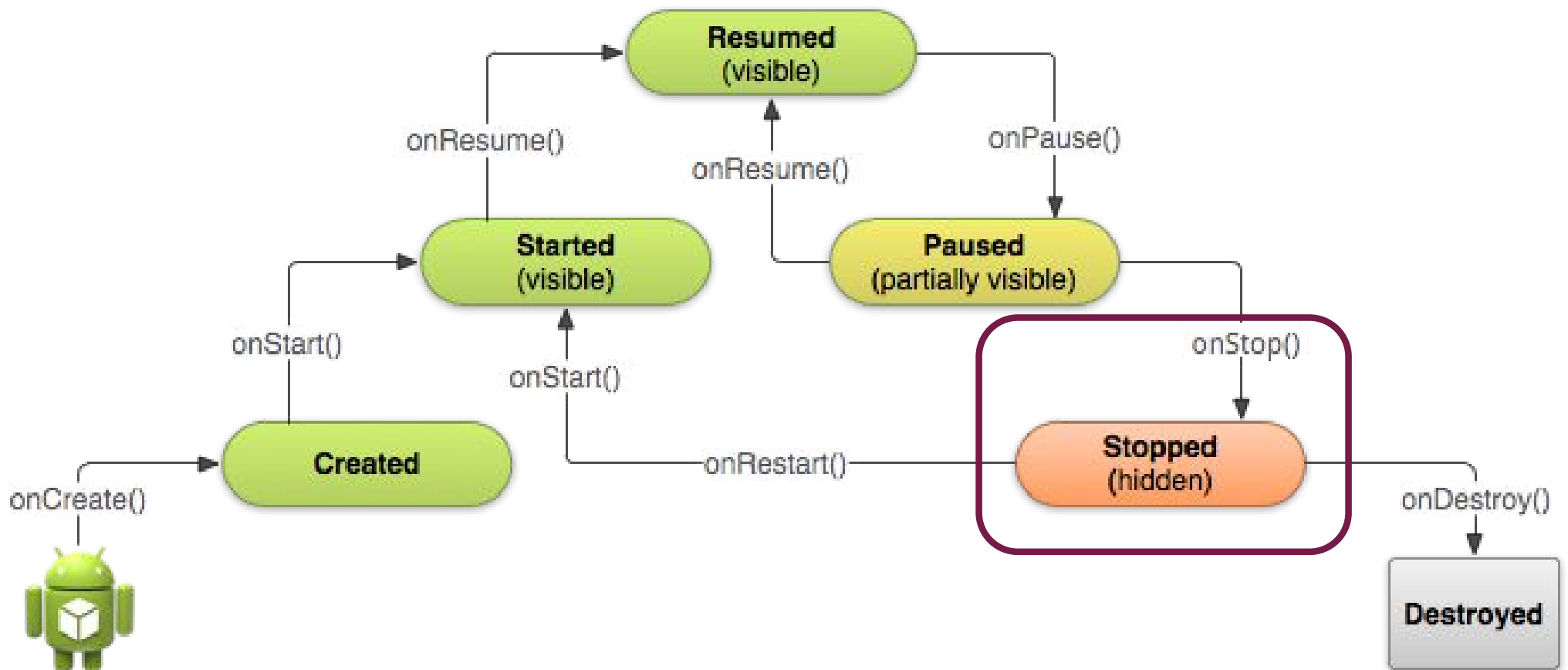
```
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about to be "paused").
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}
```

# Ciclo de Vida



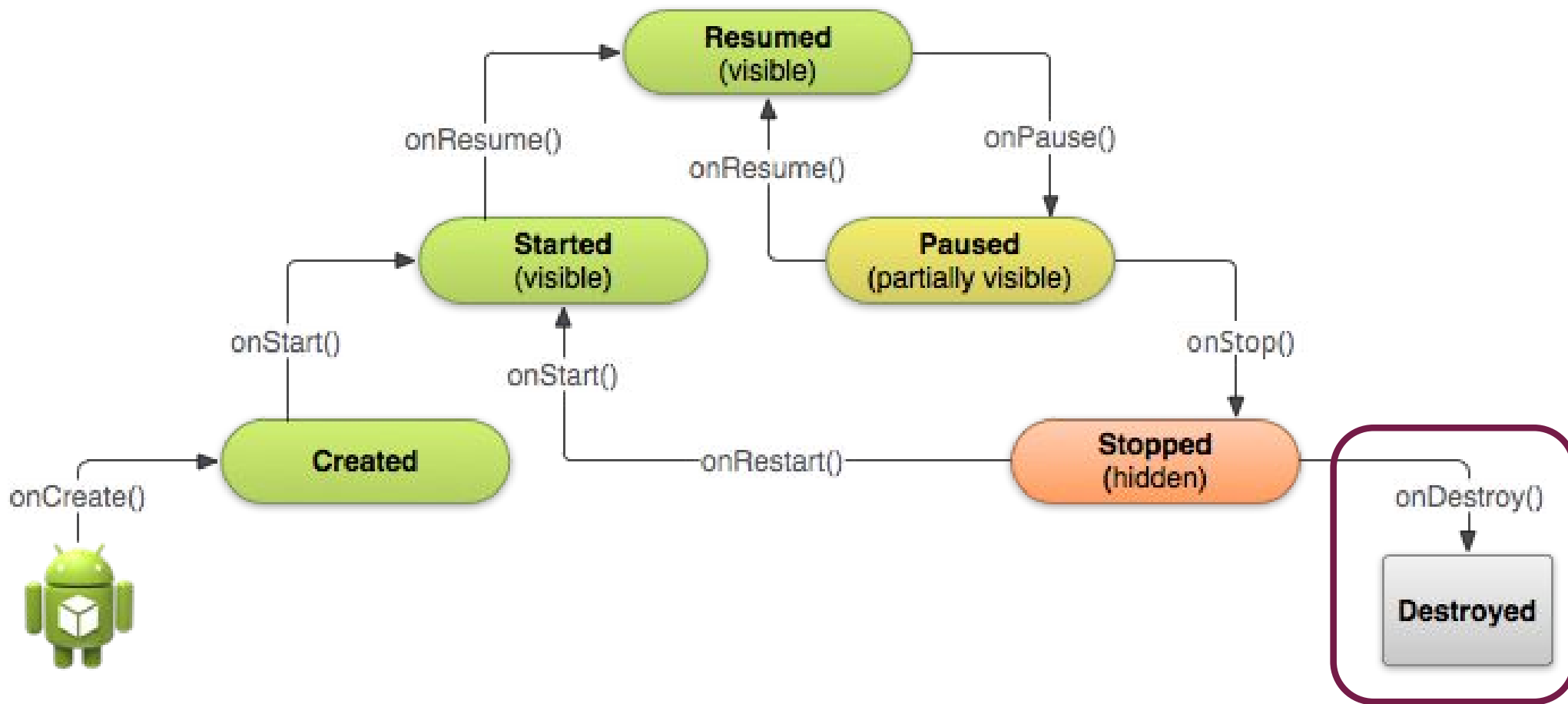
```
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about to be "paused").
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}
```

# Ciclo de Vida



```
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about to be "paused").
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}
```

# Ciclo de Vida



```
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about to be "paused").
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}
```

# Internacionalización de textos

# Internacionalización Android strings.xml

---

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="string_tittle">Bienvenidos</string>
    <string name="string_message">Hola mundo!</string>
</resources>
```

# Internacionalización Android strings.xml

---

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="string_tittle">Welcome</string>
    <string name="string_message">Hello World!</string>
</resources>
```

# Intents

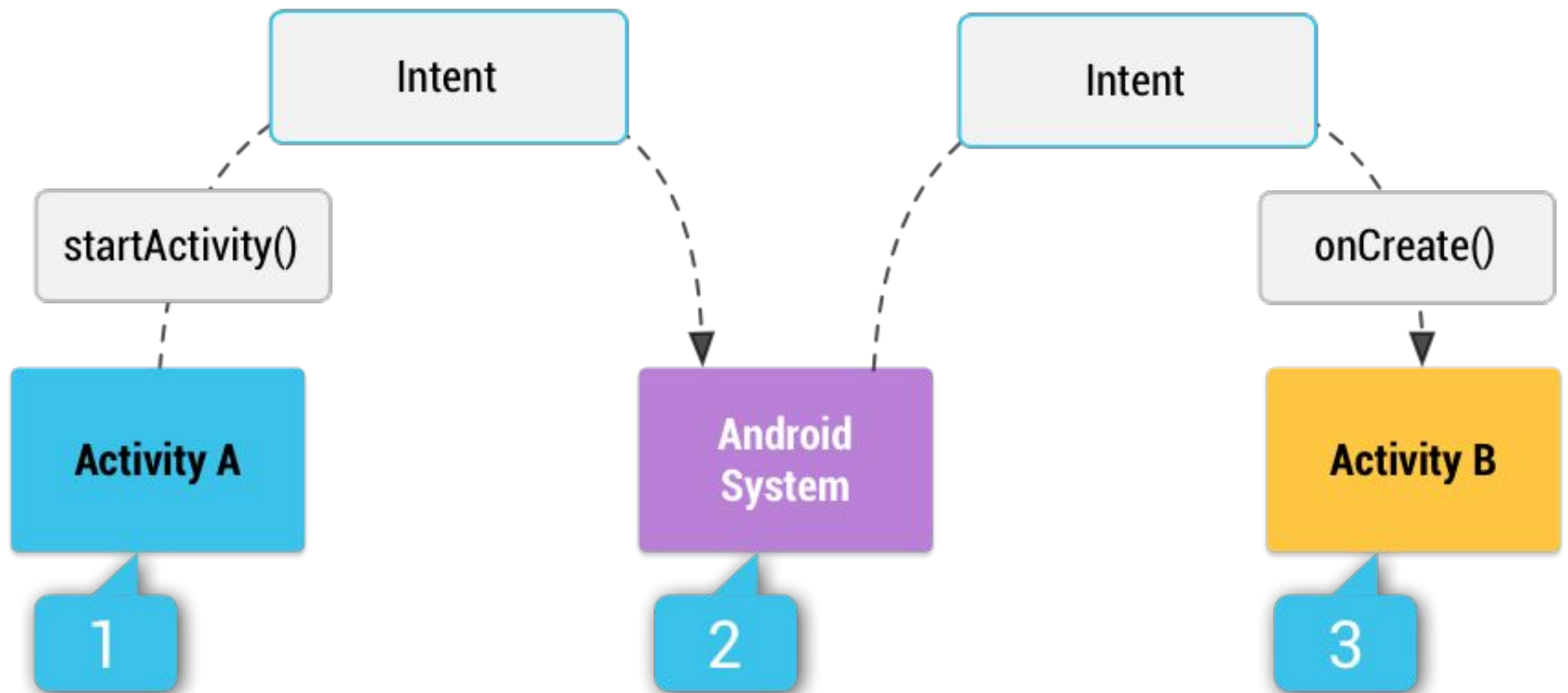
# Intents

---

Cuando queremos unir componentes  
en una aplicación usamos intents

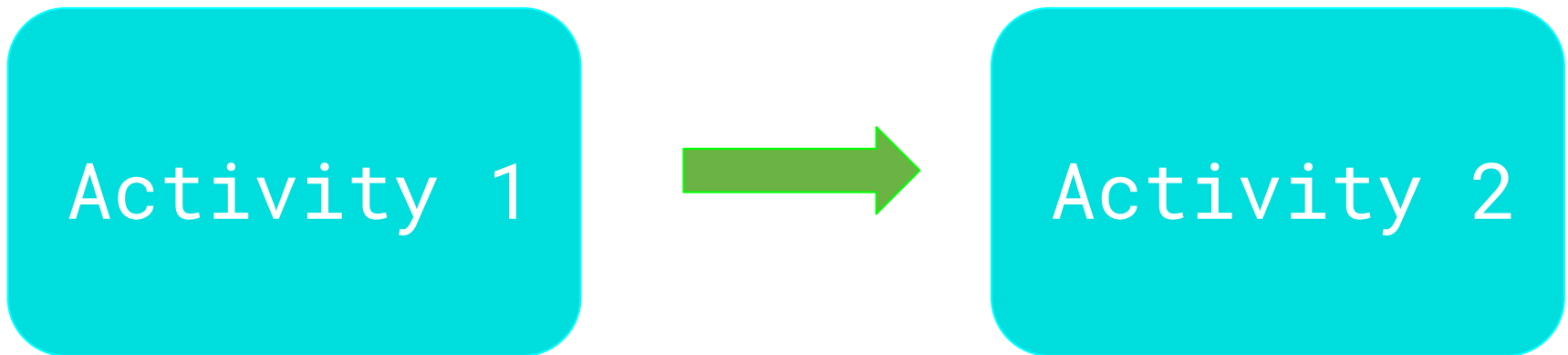
# Intents

---



# Intents

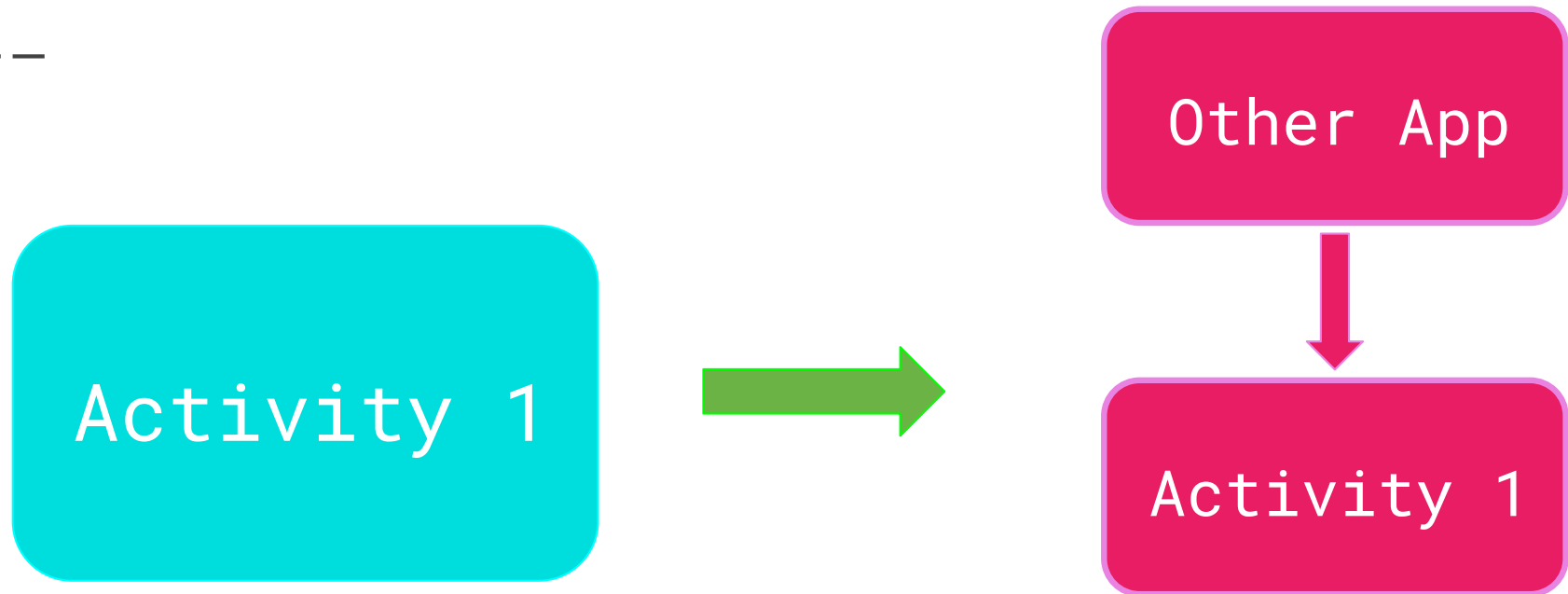
---



**Explicitos**

# Intents

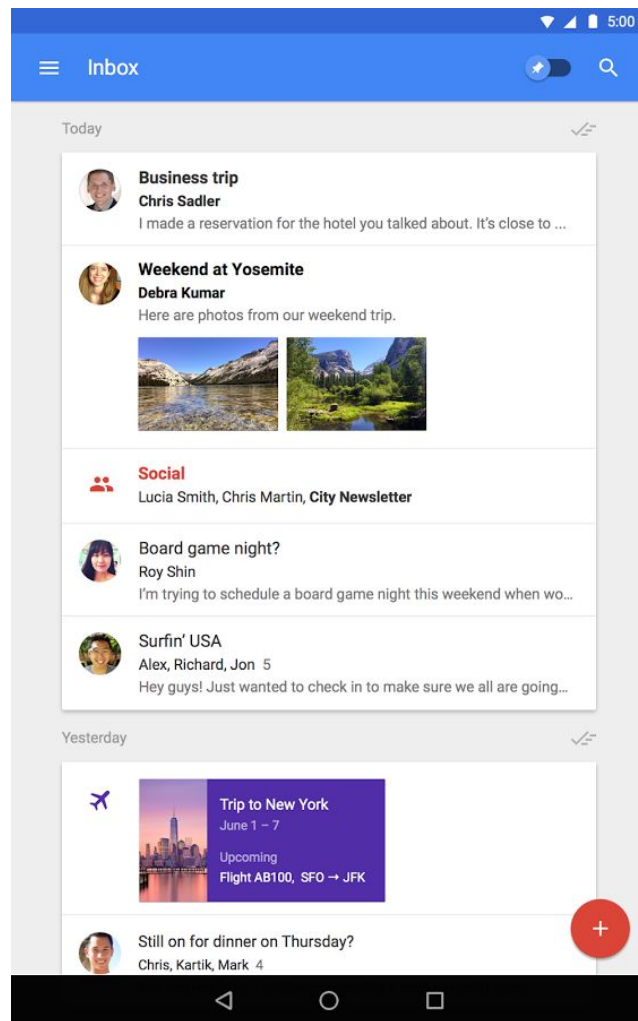
---



## Implicitos

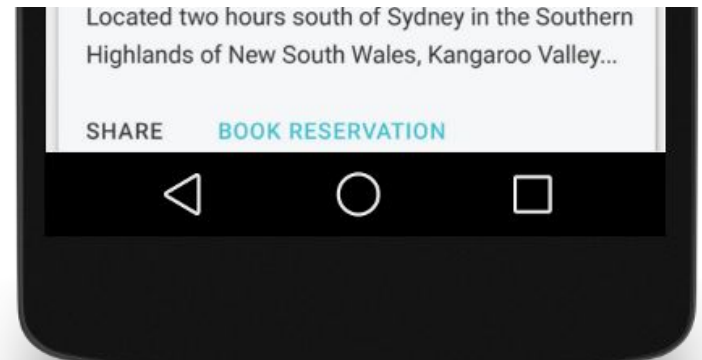
# Intents

— — —



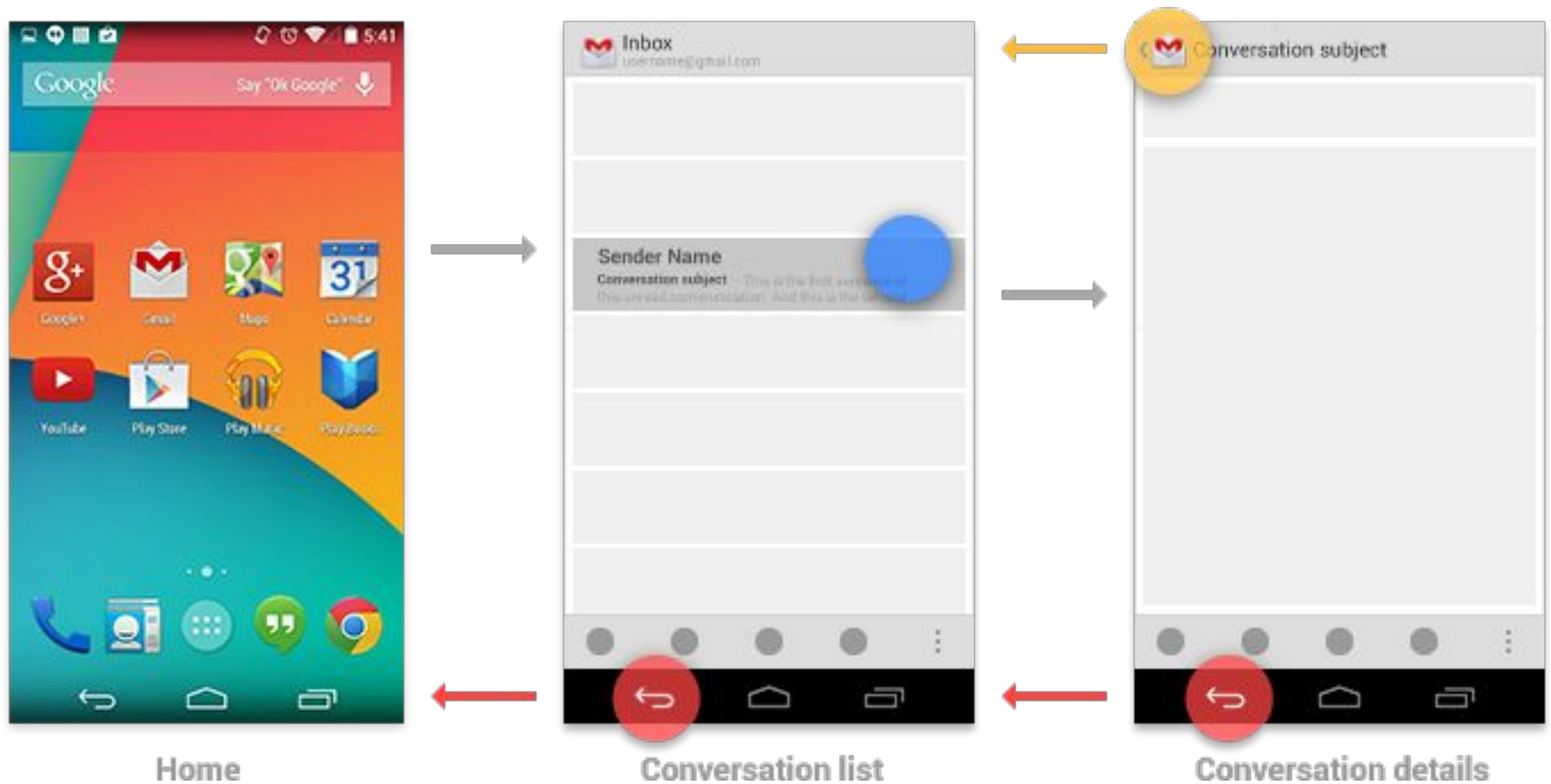
**Botón Up  
vs.  
Botón Back**

# Botón Back

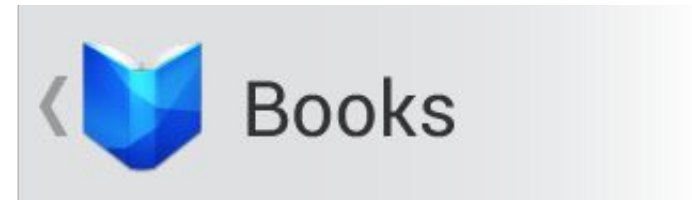
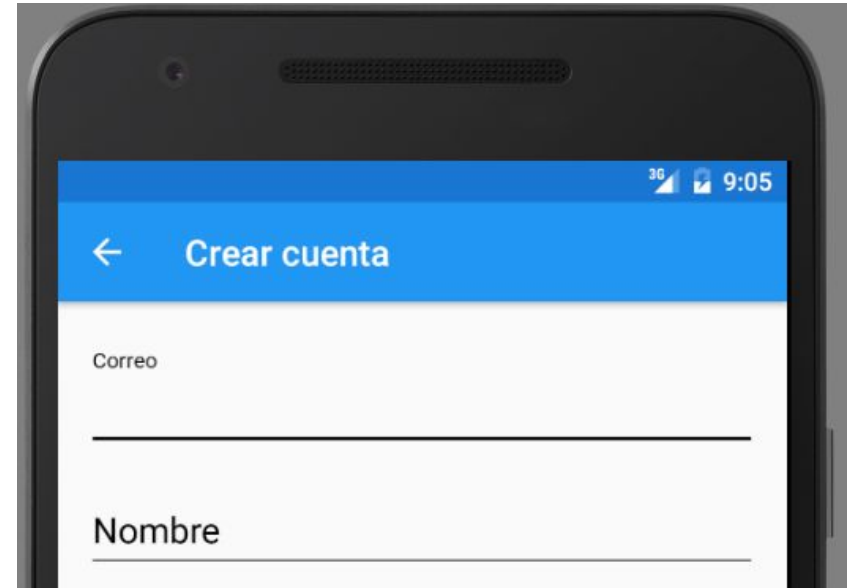


---

# Botón Back

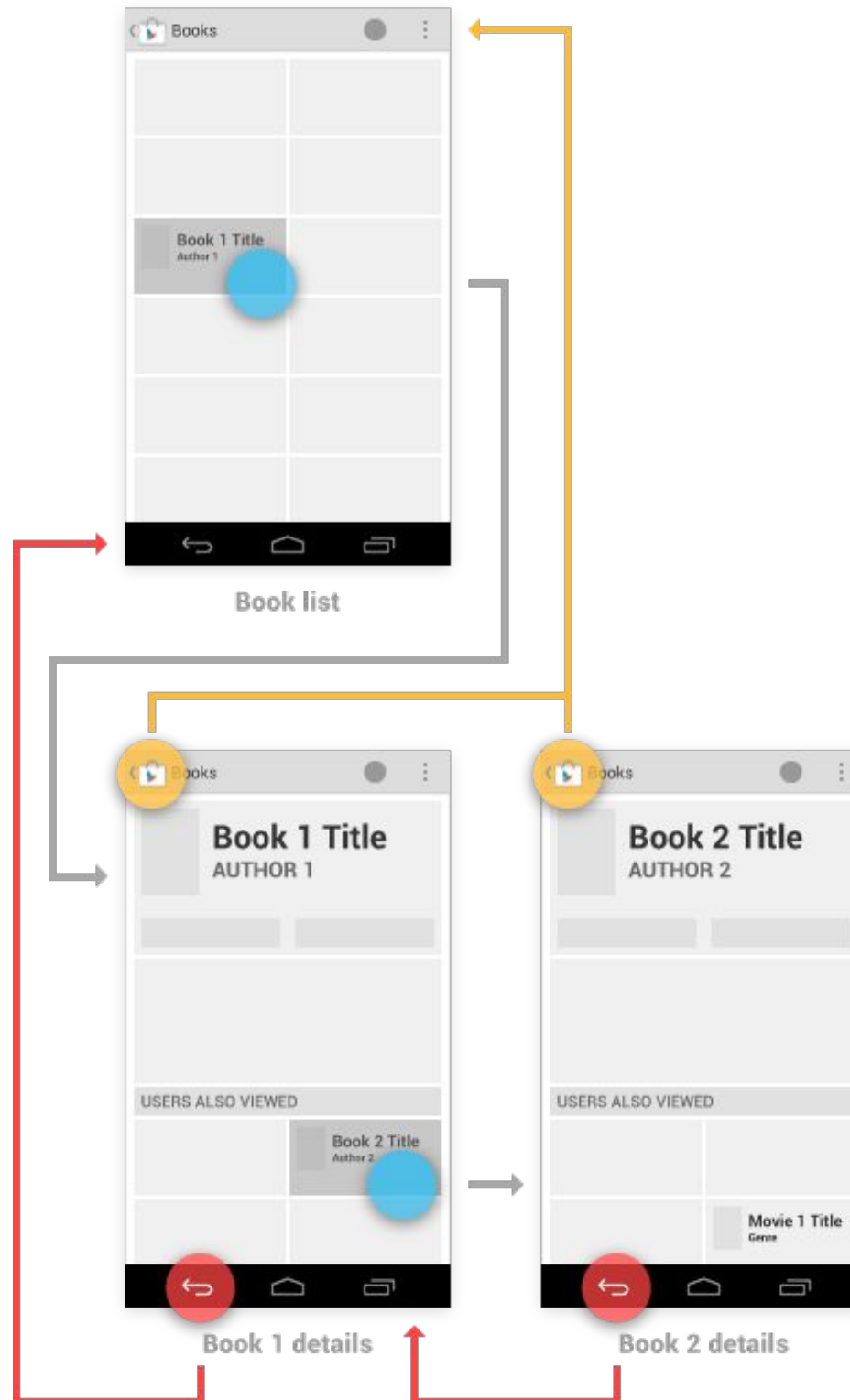


# Botón Up



---

# Botón Up



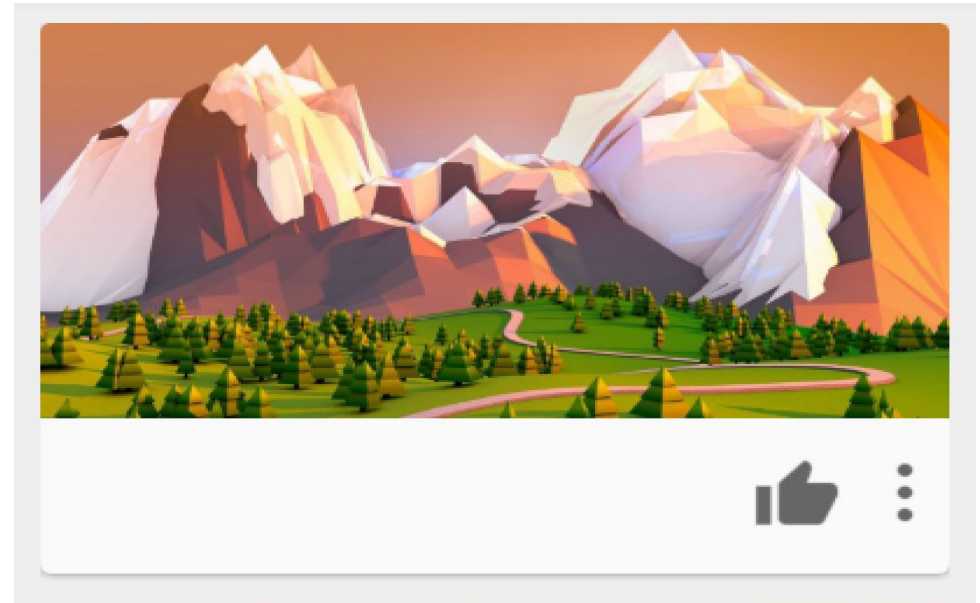
# CardView

# CardView

Es un ViewGroup

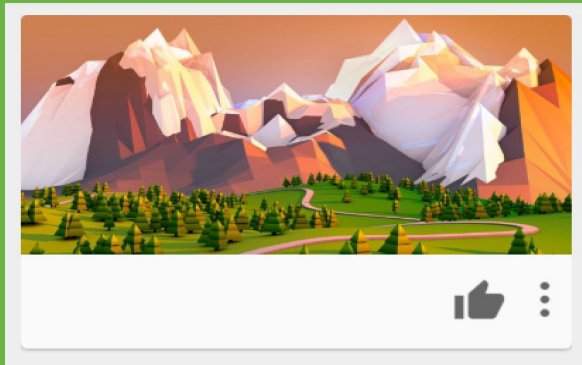
Nos permite mostrar  
información dentro de  
tarjetas

Podemos modificar los  
bordes y añadir sombras si  
así lo queremos



---

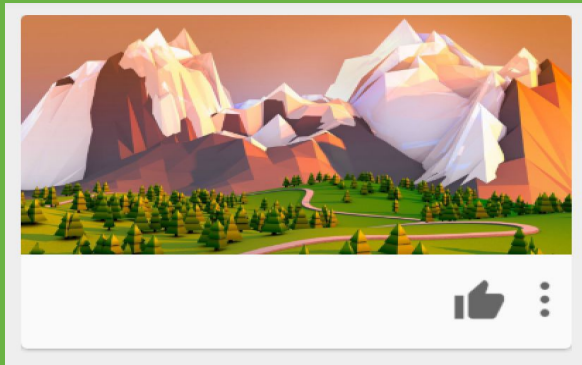
# CardView



- Etiqueta CardView que proviene del soporte
- Layout interno para definir la estructura de los elementos
- Elemento ImageView y dos ImageButtons

---

# CardView



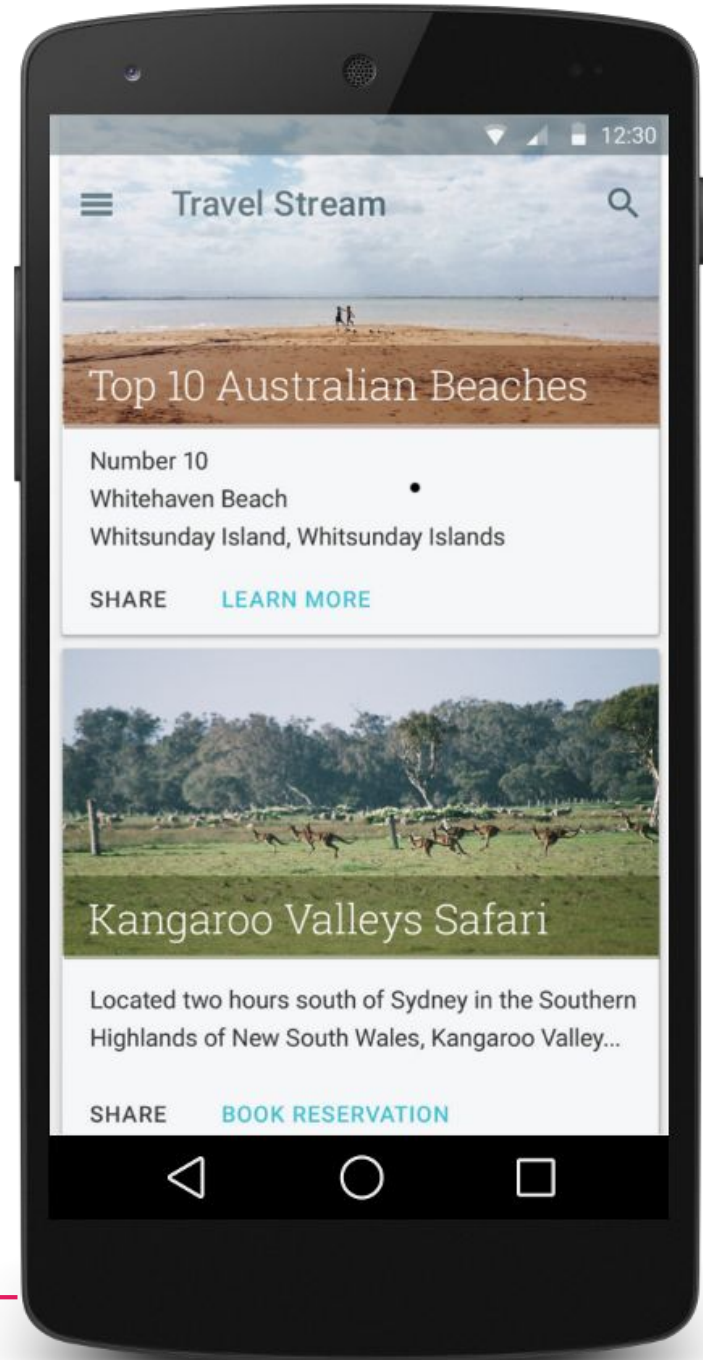
Es necesario  
añadir la  
librería de  
soporte en  
gradle

---

# RecyclerView

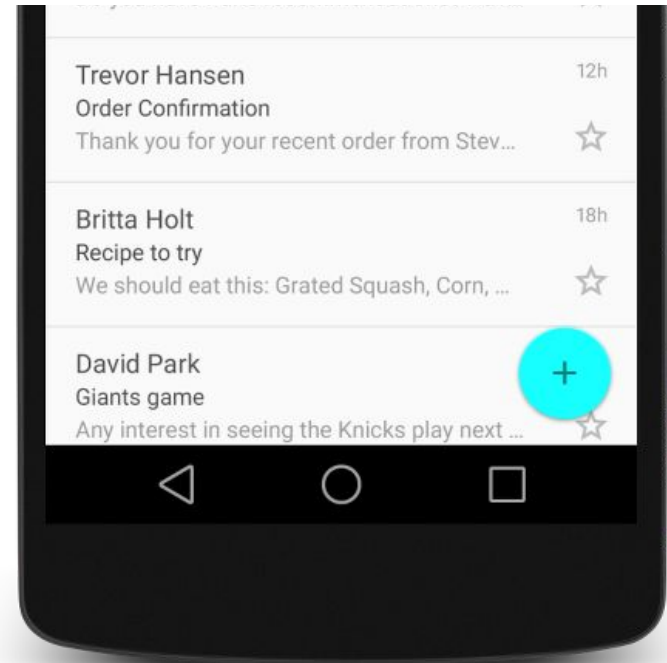
# RecyclerView

Más eficiente que ListView



# Floating Action Button

# Floating Action Button

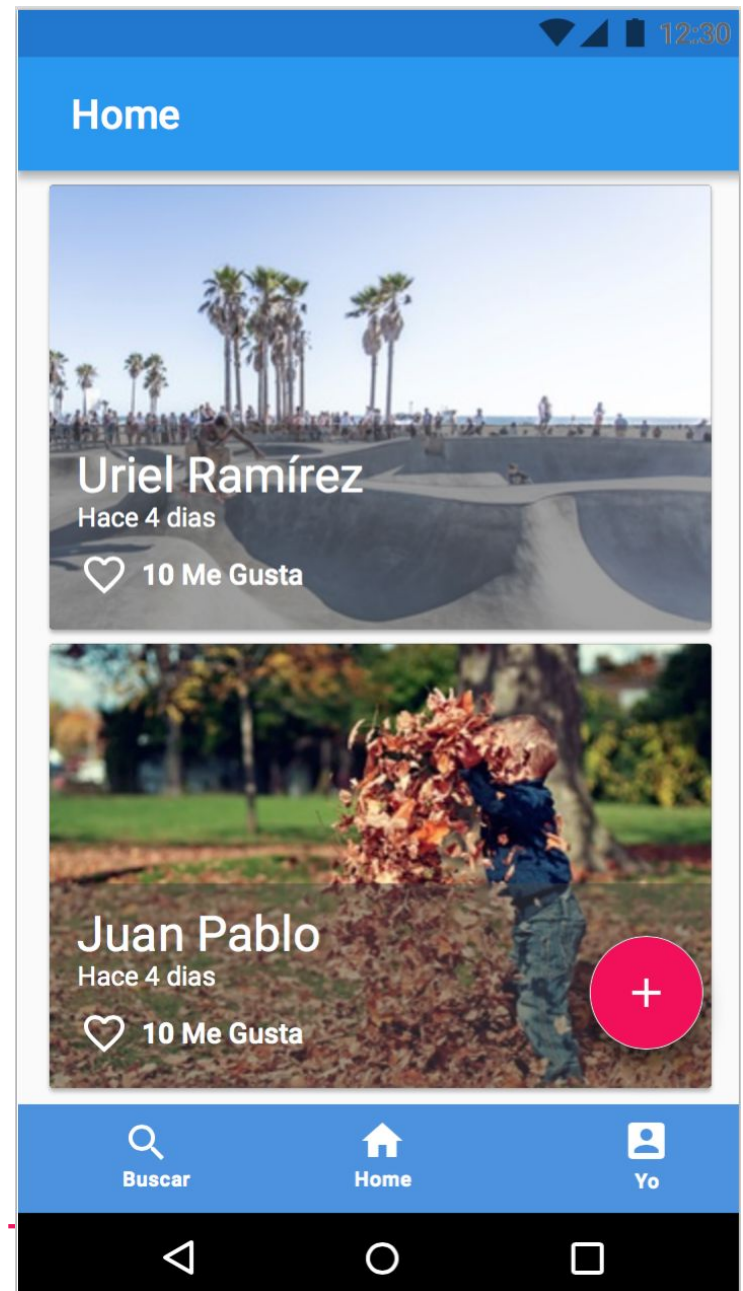


```
<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_marginLeft="@dimen/fab_margin"
    android:layout_marginRight="@dimen/fab_margin"
    android:layout_marginBottom="16dp"
    android:src="@drawable/ic_plus" />
```

# RecyclerView

# RecyclerView

Más eficiente que ListView

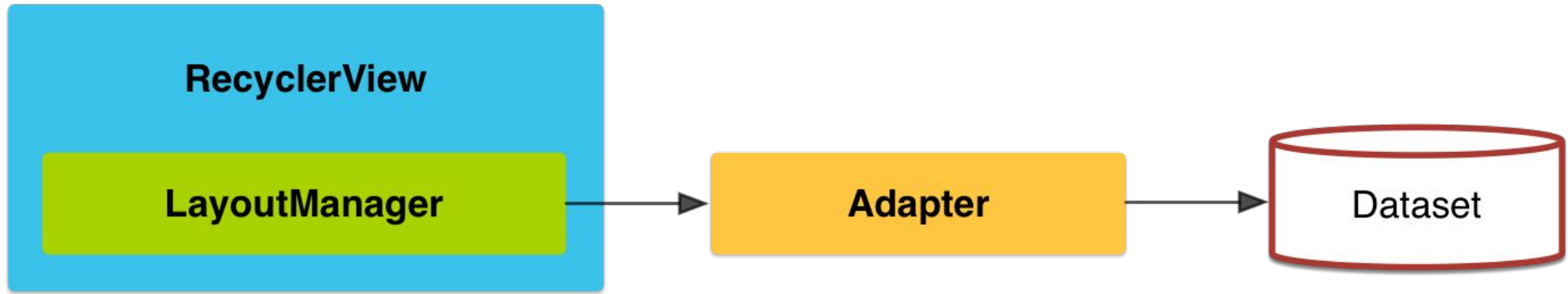


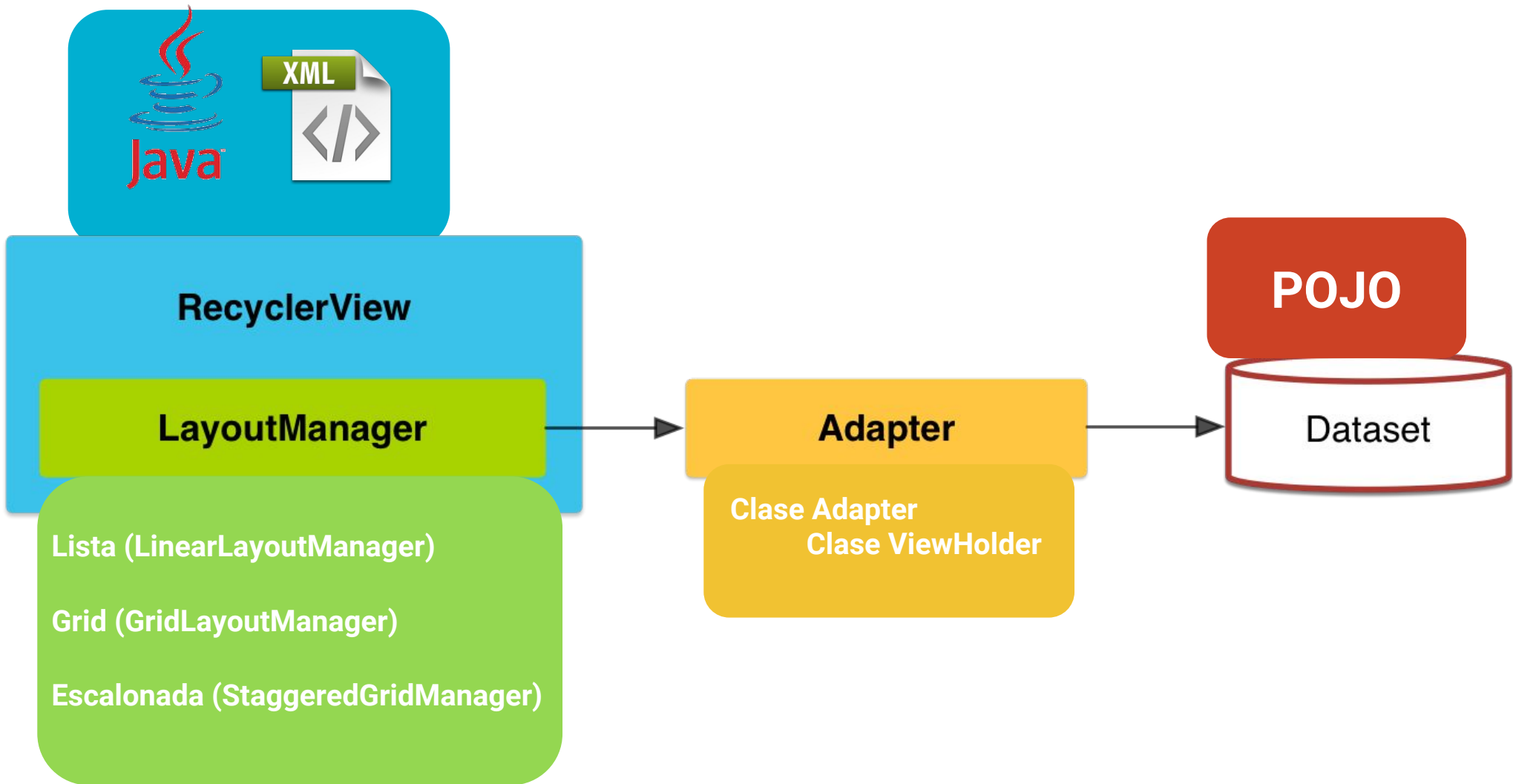
**RecyclerView**

**LayoutManager**

**Adapter**

**Dataset**





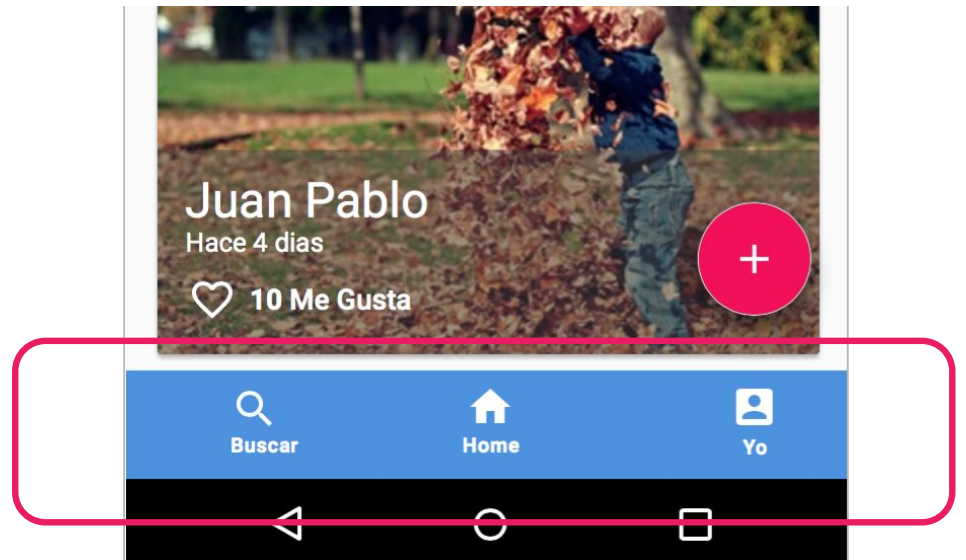
# RecyclerView ItemDecoration

---

**BottomBar**

# BottomBar

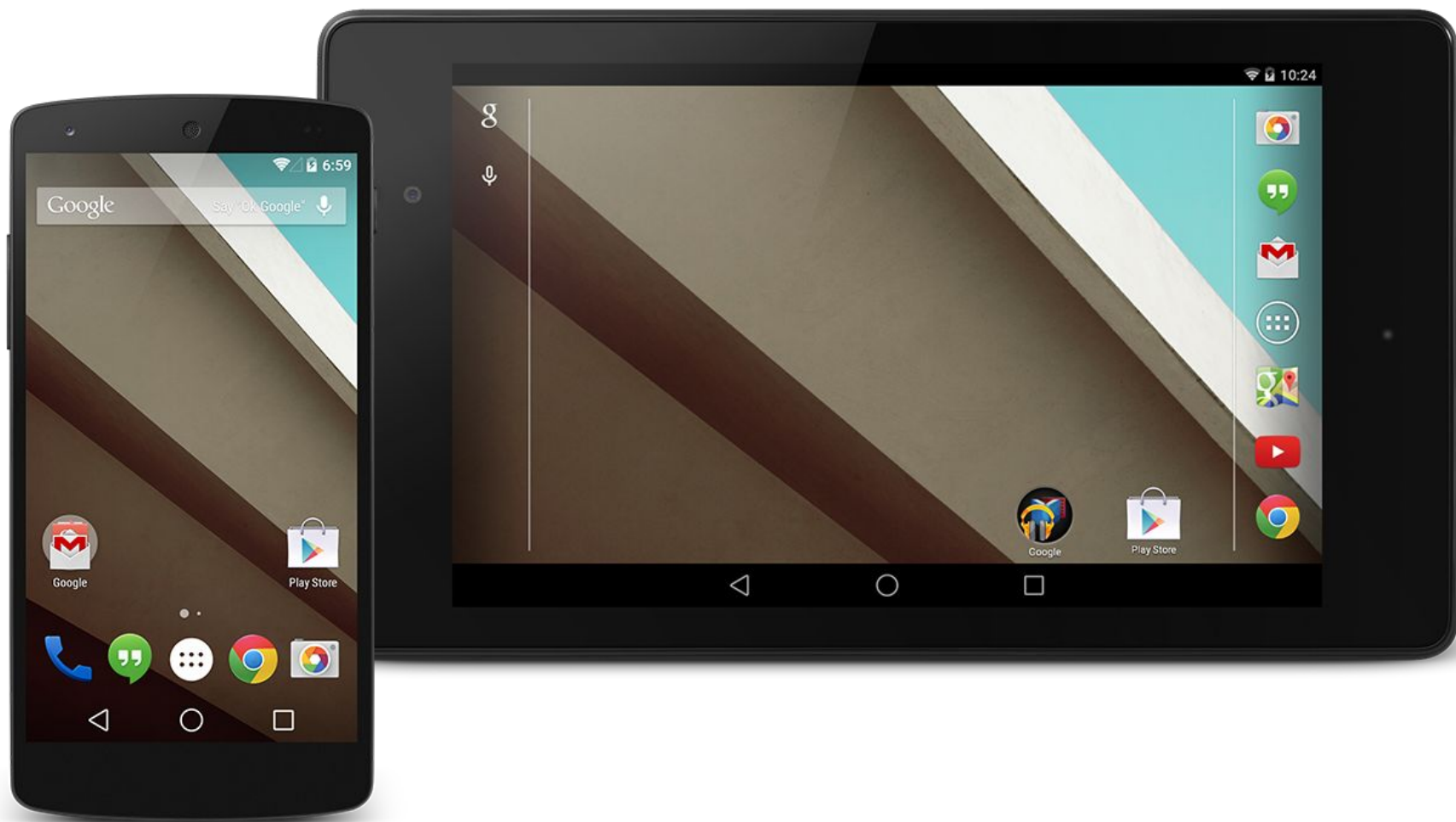
No existe librería nativa  
POR EL MOMENTO

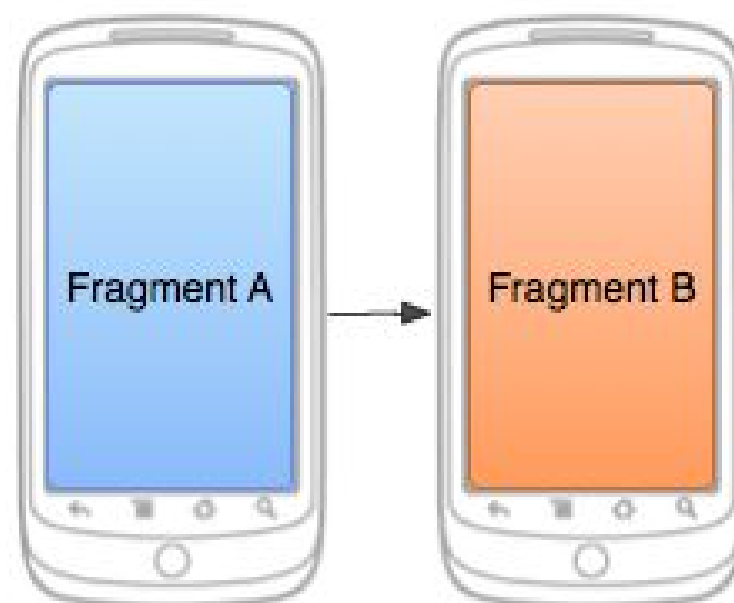
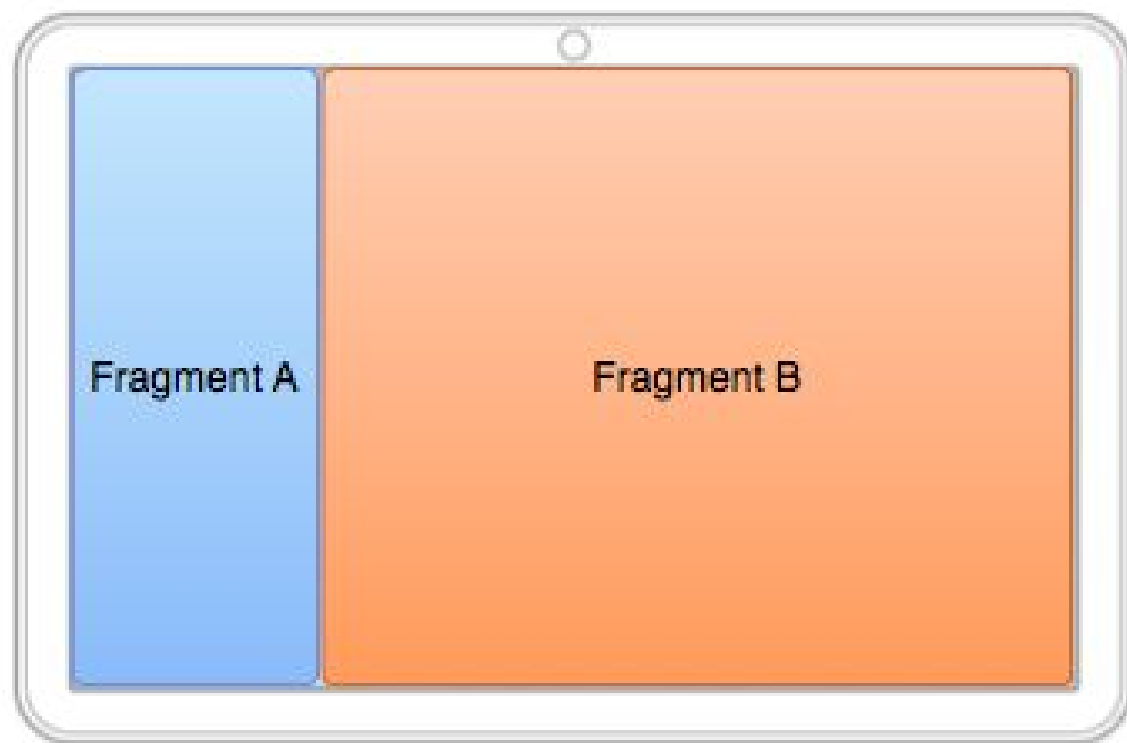


<https://github.com/roughike/BottomBar>

# Fragments

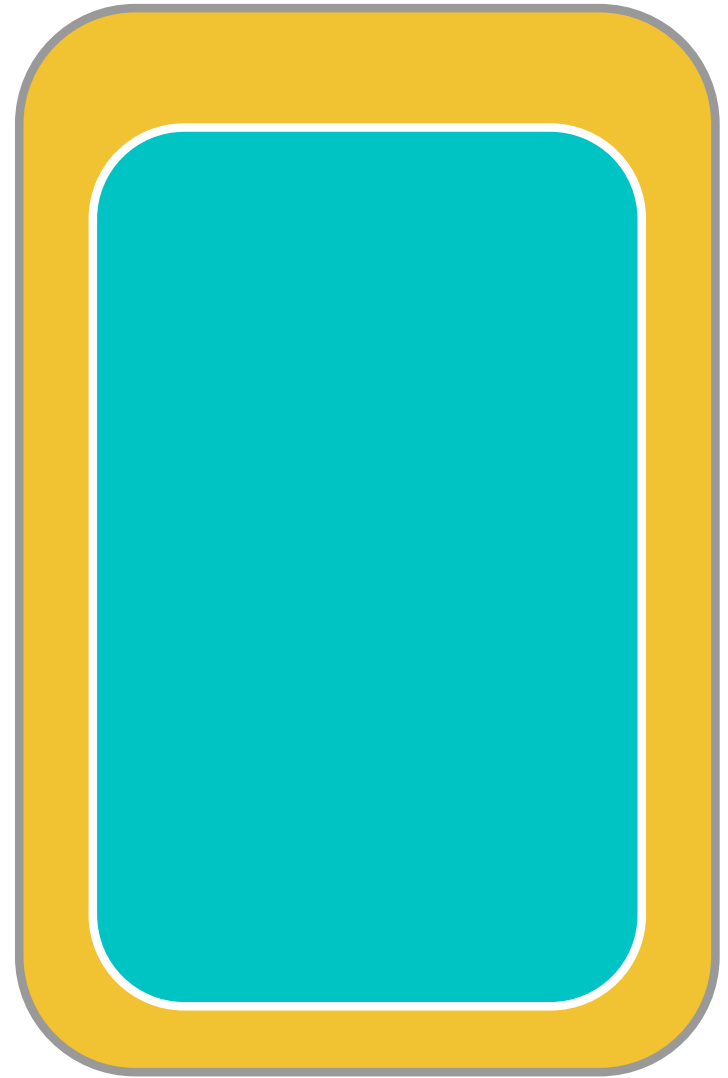




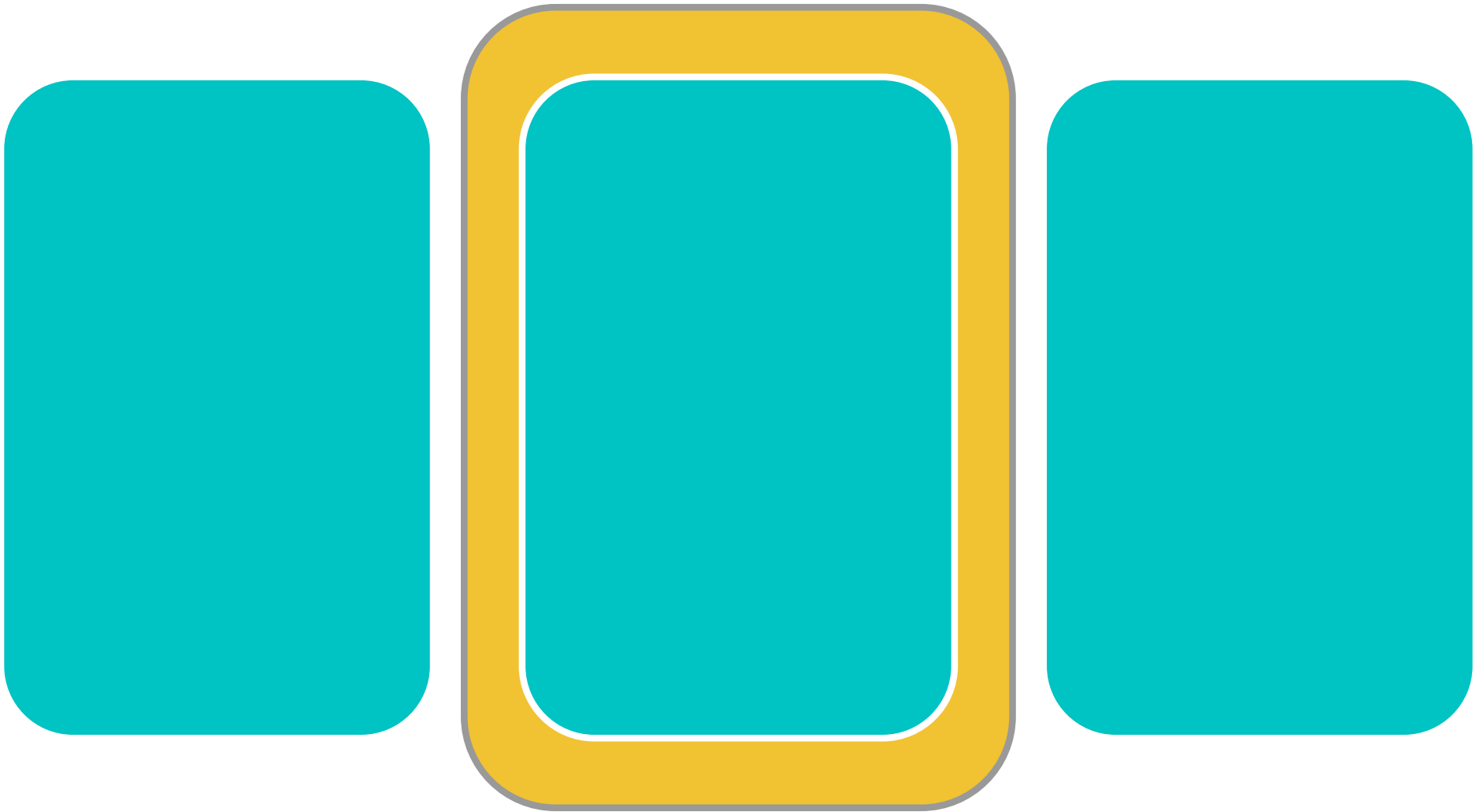


# Container

# Fragments



---



# Fragments

- FragmentManager
- Transacciones
- Commits

---

**CoordinatorLayout,  
AppBarLayout y  
CollapsingToolbarLayout**

# CoordinatorLayout

- Un super poderoso  
FrameLayout
- Ayuda a especificar  
comportamientos  
interacciones en los  
views hijos
- Vistas flotantes

---

# AppBarLayout

- Es esencialmente un `LinearLayout` vertical
- Nos permite manejar características de Material Design en `AppBar`

---

# CollapsingToolbarLayout

- Es un Wrapper para Toolbar el cual implementa un efecto collapsable en toolbar y los elementos hijos
- Hijo directo de AppBarLayout

---

**Imagen Redonda**

# Imagen Redonda



Byron Martinez

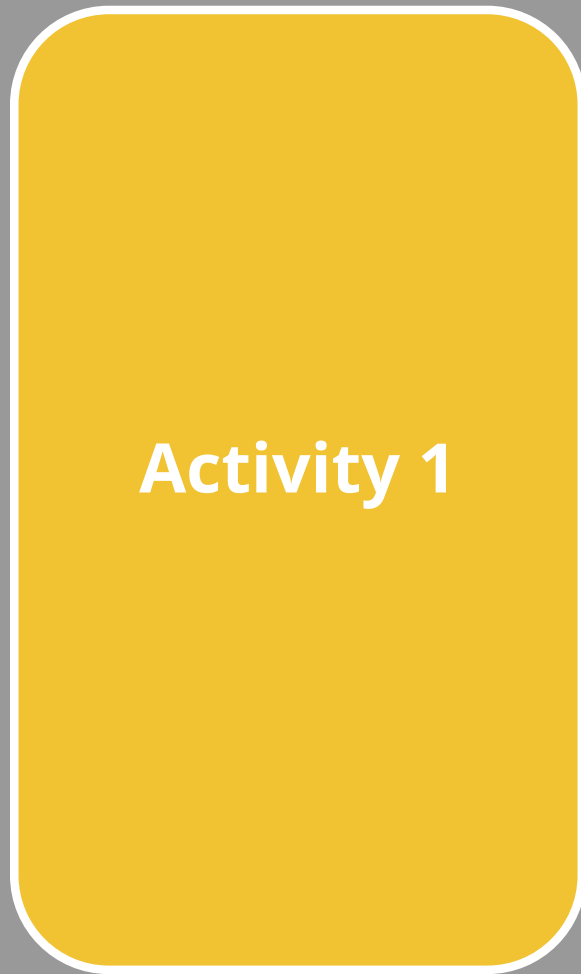
---

<https://github.com/hdodenhof/CircleImageView>

# Transiciones Material Design

**Fade, Slide Explode**

---



**Activity 1**

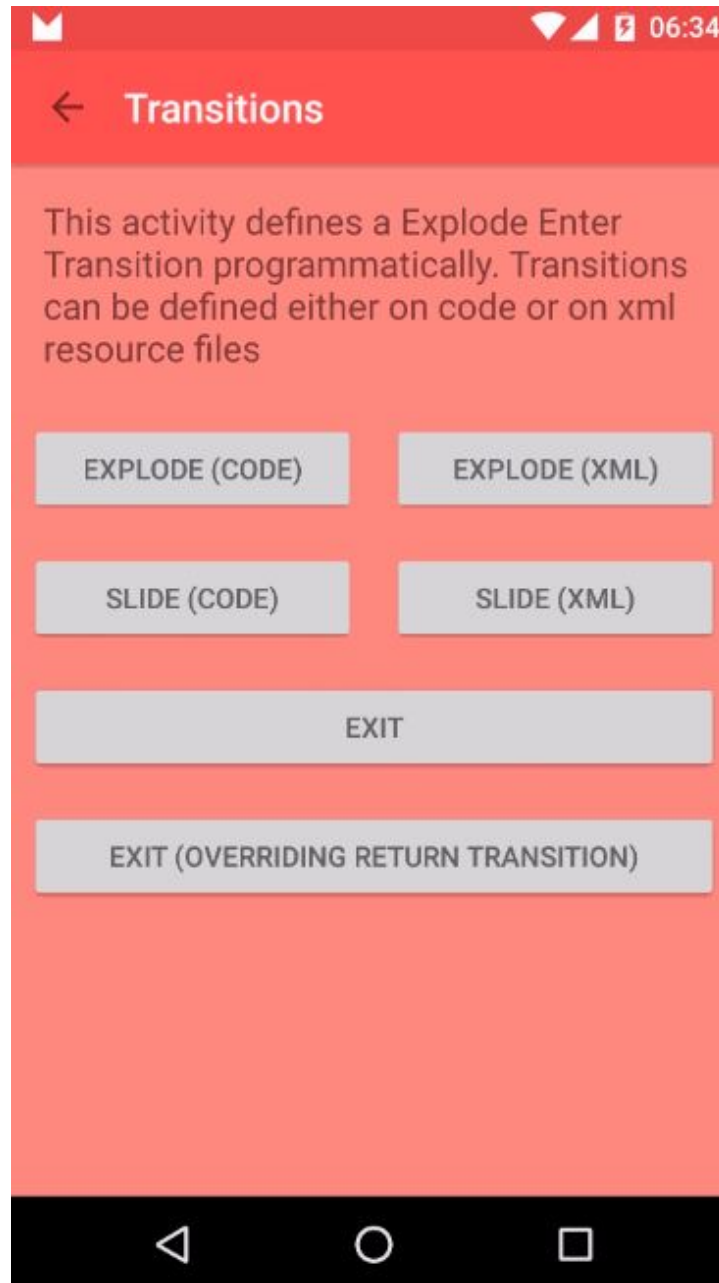
**exitTransition()**

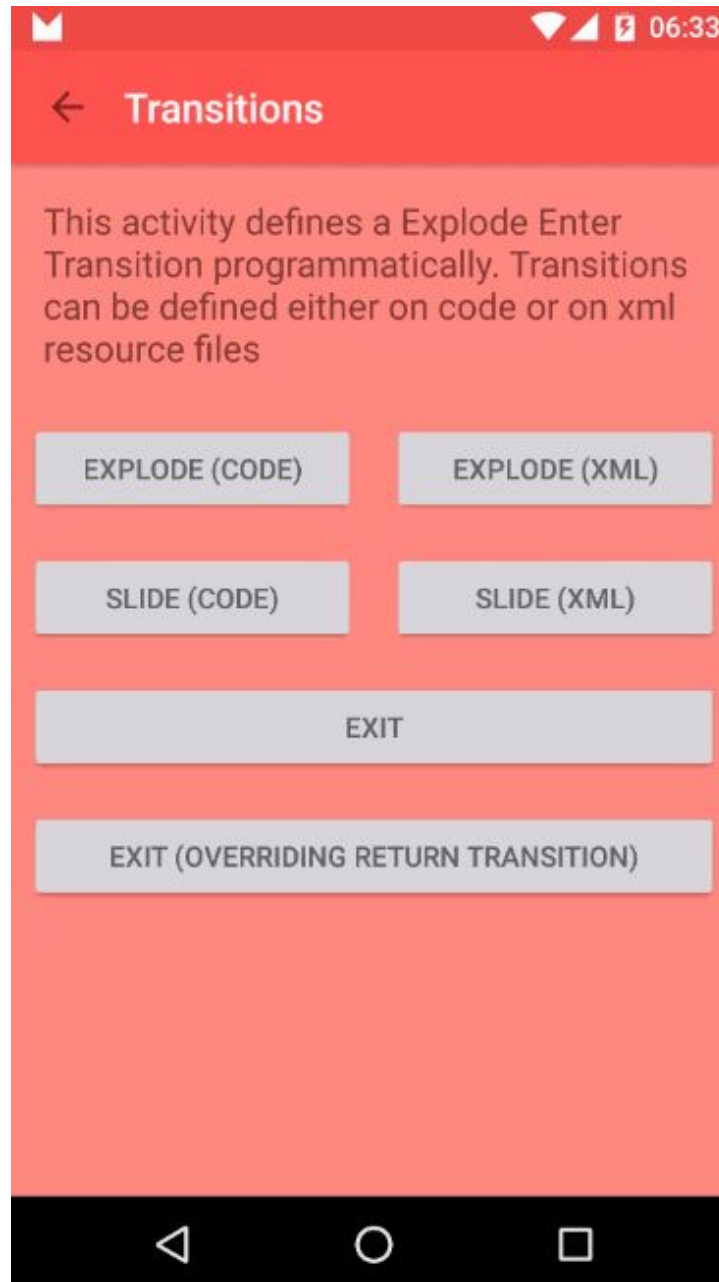
**startActivity(A2)**

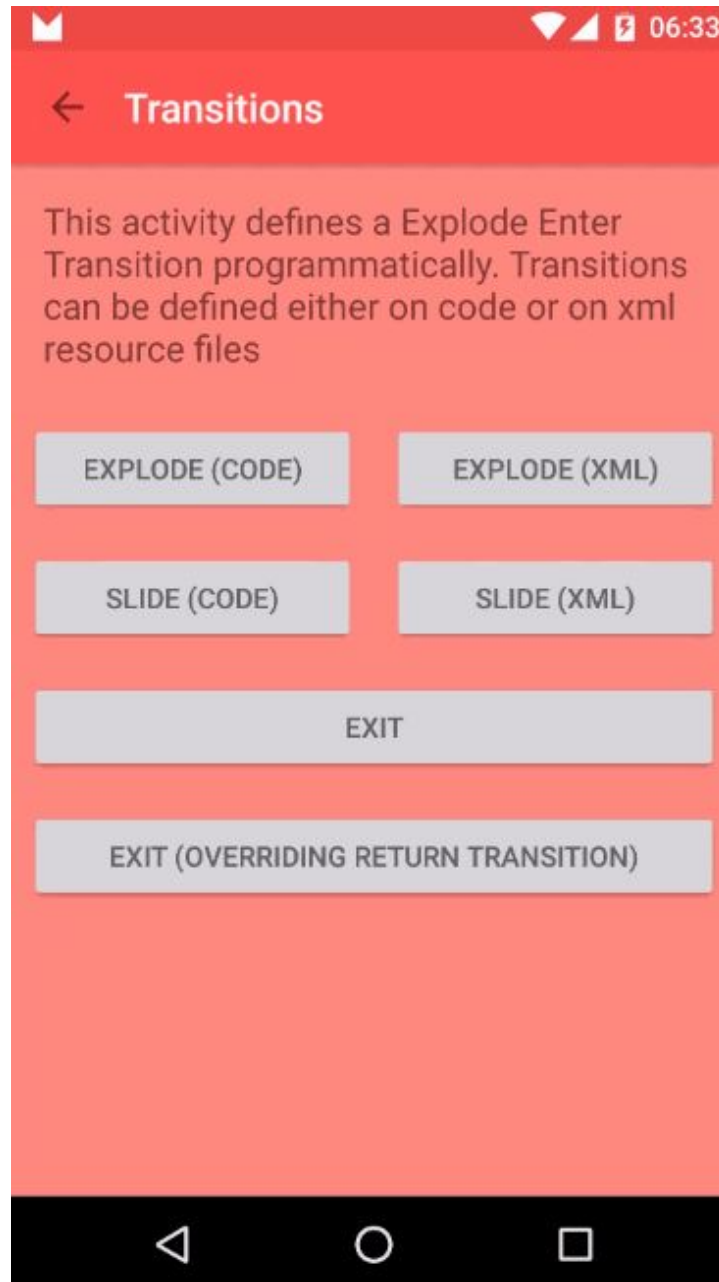


**Activity 2**

**enterTransition()**





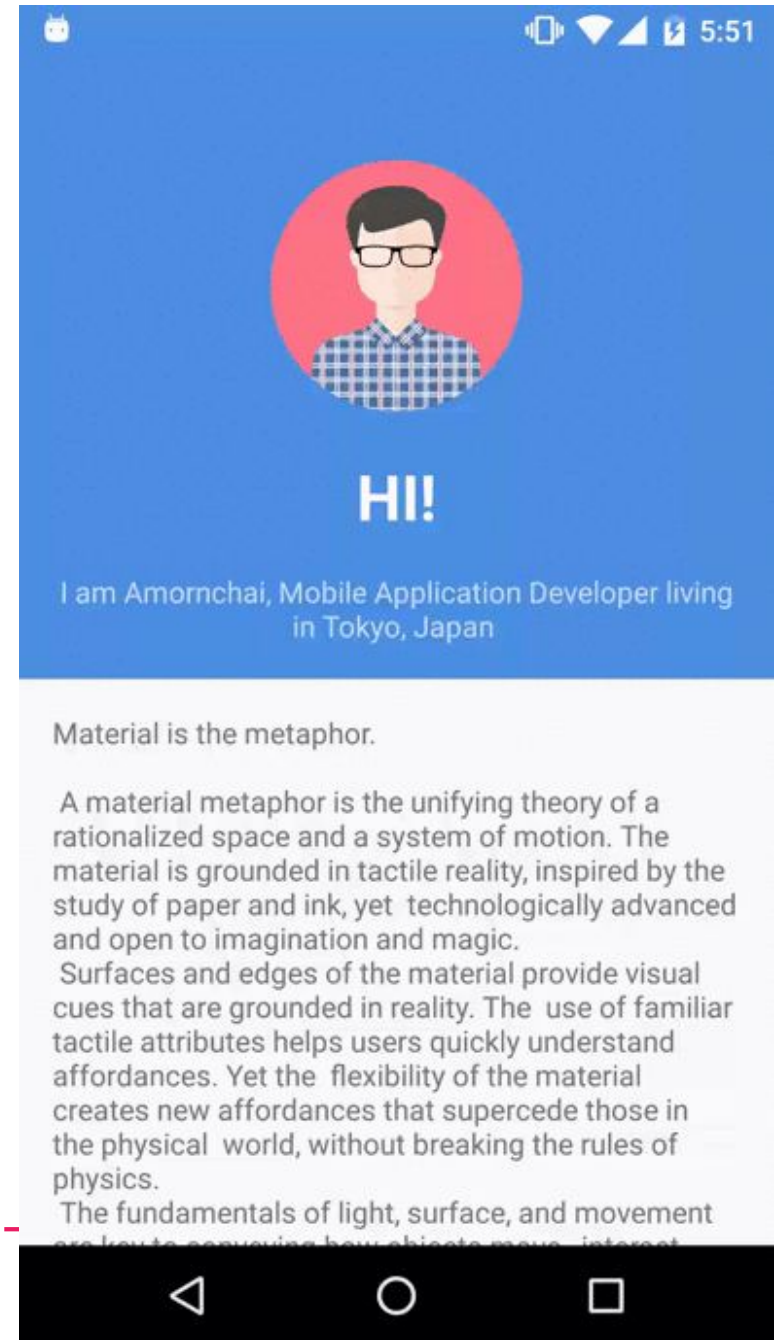


# Fade

- Es esencialmente un `LinearLayout` vertical
- Nos permite manejar características de Material Design en `AppBar`

---

# Comportamientos Dependientes



<https://github.com/zoonooz/simple-view-behavior>

**Anahí Salgado**  
**@anncode**