



PROGRAMA DE FORMACIÓN  
DOCENTE DE LA UNIVERSIDAD  
MIGUEL HERNÁNDEZ.

**Desarrollo de aplicaciones móviles  
para la docencia universitaria**

abril 2013

## A.5. Estructura de una app

# Estructura de una app

## **Bloques fundamentales de una app:**

- Activity
- Fragment
- Layouts
- UI Widgets
- Input events
- Gestures

# Estructura de una app

## Activity

- Es el componente central de una app.
- Una actividad suele estar representada por una pantalla. Aunque es posible tener asociadas varias pantallas a una misma actividad, imaginarse las actividades como pantallas es la forma más sencilla de entender una app.

<http://developer.android.com/guide/components/activities.html>

# Estructura de una app

## Activity

Las actividades deben declararse en el fichero Manifest.xml, dentro del elemento <application>:

```
<manifest ... >
  <application ... >
    <activity android:name=".ExampleActivity" />
    ...
  </application ... >
  ...
</manifest >
```

# Estructura de una app

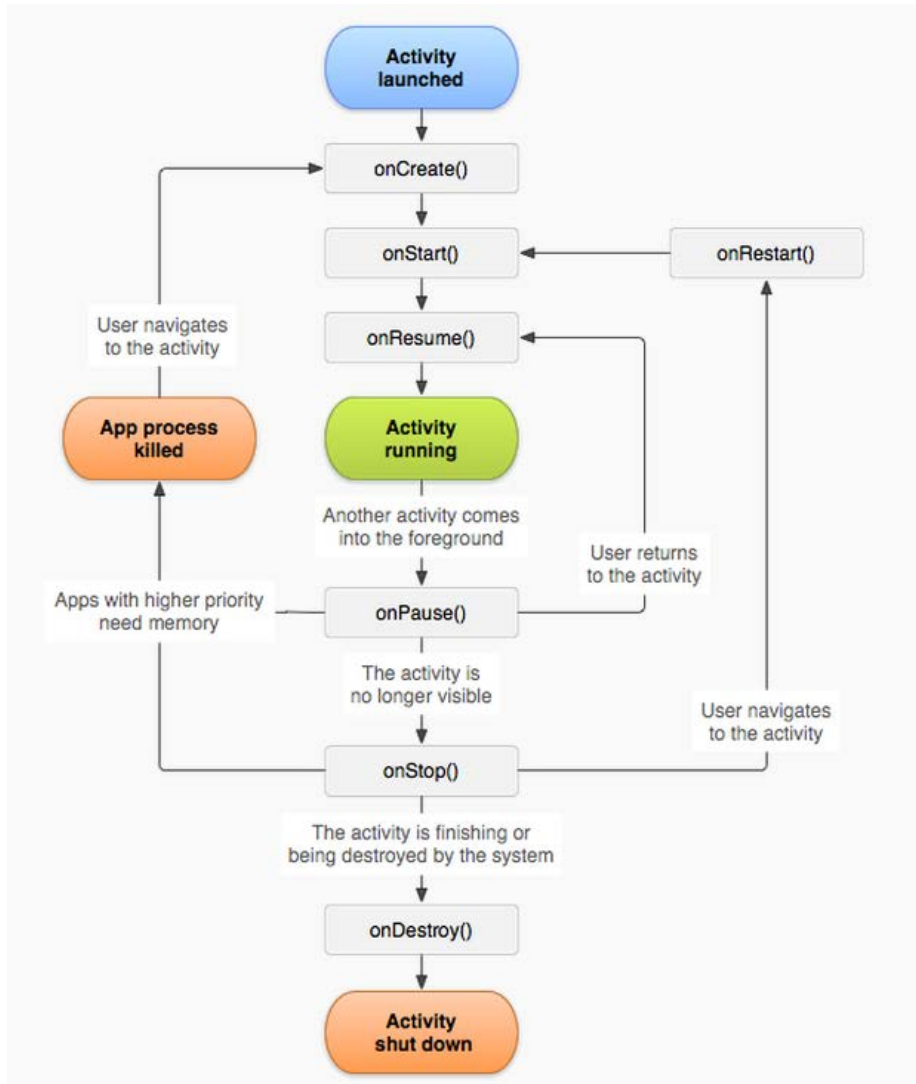
## Activity

Siempre debemos declarar una actividad “lanzadera” o principal, que será la primera que muestra la app al iniciarse:

```
<activity android:name=".ExampleActivity" android:icon="@drawable/app_icon">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

# Estructura de una app

## Activity Lifecycle

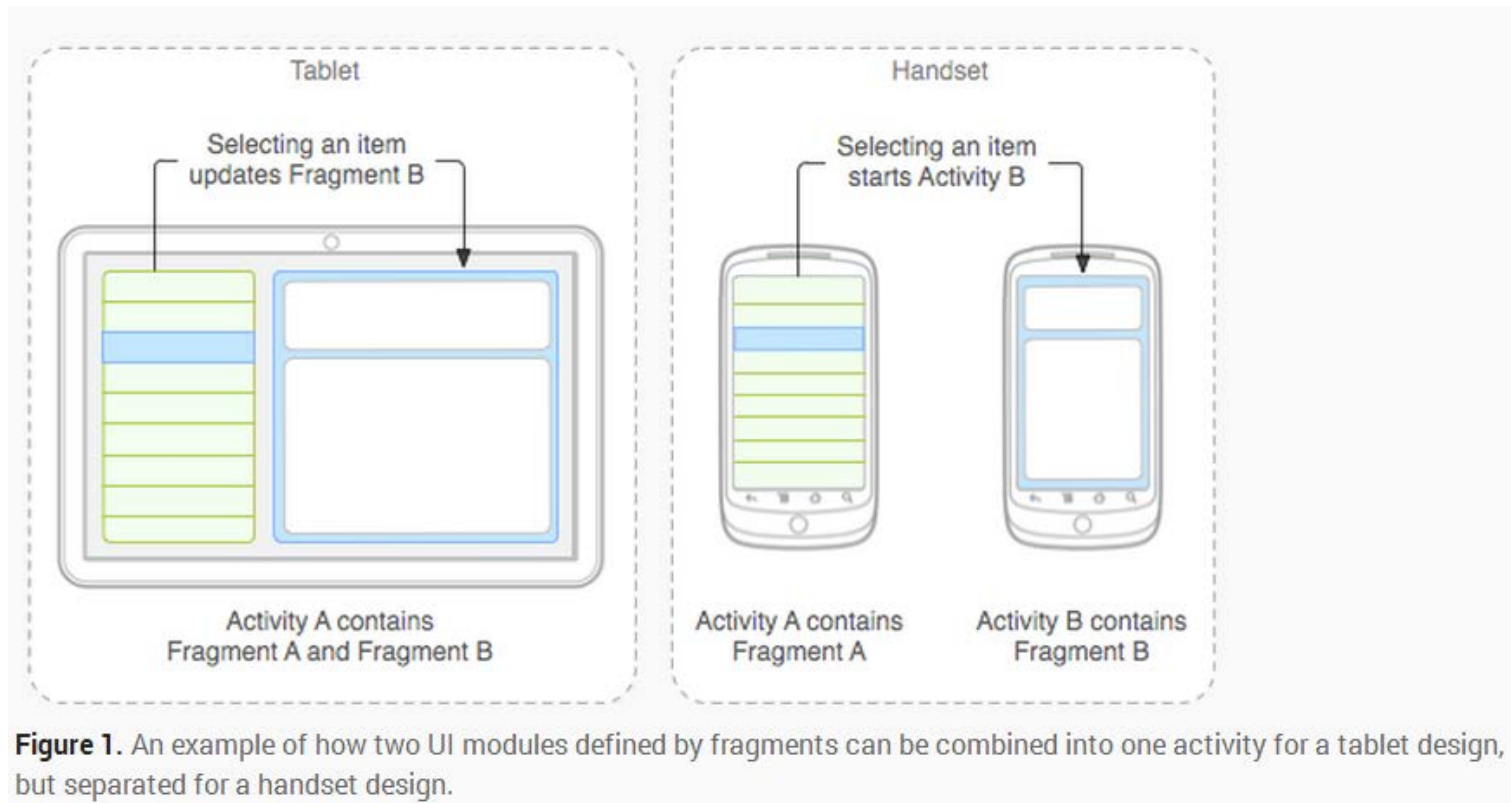


```
public class ExampleActivity extends Activity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        // The activity is being created.  
    }  
    @Override  
    protected void onStart() {  
        super.onStart();  
        // 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.  
    }  
}
```

# Estructura de una app

## Fragment

- Es una subactividad. Se introdujo en Honeycom (3.0) para hacer apps más escalables entre tabletas y teléfonos.






**Figure 1.** An example of how two UI modules defined by fragments can be combined into one activity for a tablet design, but separated for a handset design.

# Estructura de una app

## Layouts

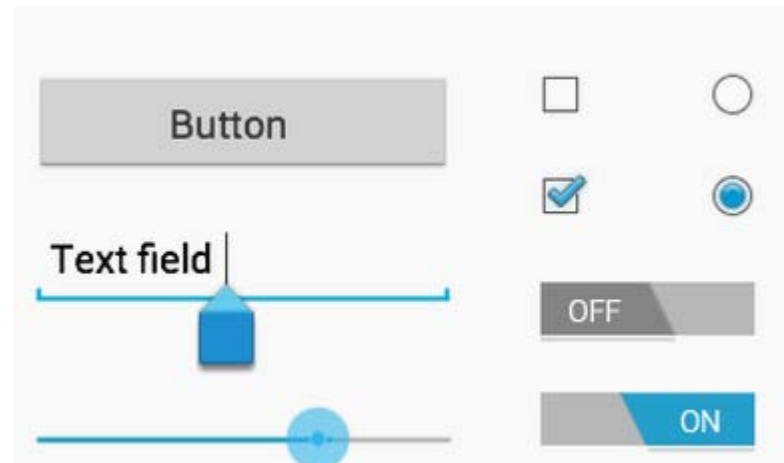
- Son agrupamientos de **User Interface (UI) widgets**. Una actividad o un fragmento tienen un layout como contenedor de los objetos que se muestran.
- Los layouts pueden contener a su vez otros layouts formando estructuras jerárquicas.

Linear Layout	Relative Layout	Web View
		
<p>A layout that organizes its children into a single horizontal or vertical row. It creates a scrollbar if the length of the window exceeds the length of the screen.</p>	<p>Enables you to specify the location of child objects relative to each other (child A to the left of child B) or to the parent (aligned to the top of the parent).</p>	<p>Displays web pages.</p>

# Estructura de una app

## UI widgets

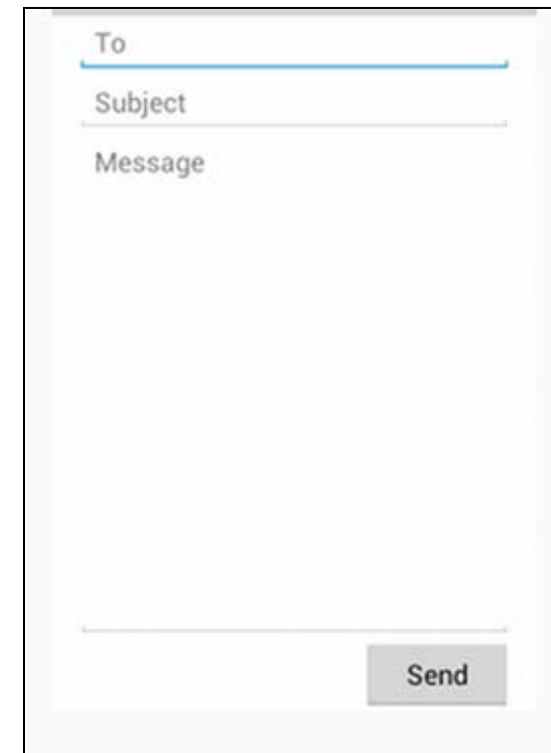
- Son componentes individuales como botones, textos, imágenes, ...
- Hay muchos tipos diferentes de UI widgets, desde muy sencillos como las etiquetas de texto hasta widgets más complejos para realizar galerías o listados de objetos. Son flexibles y pueden customizarse simplemente cambiando el valor de sus parámetros



# Estructura de una app

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:orientation="vertical" >
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/to" />
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/subject" />
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="top"
        android:hint="@string/message" />
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="@string/send" />
</LinearLayout>
```

Ejemplo de layout (*LinearLayout*)  
con 4 UI widgets: 3 *EditText* y 1 *Button*



# Estructura de una app

## Input Events

Hay más de una manera de interceptar los eventos de la interacción del usuario con la aplicación. Se deben capturar los eventos del objeto específico con el cuál el usuario interactúa.


***Event listeners:*** capturan la interacción del usuario con un objeto (por ejemplo, hacer click en un botón)

***Event handlers:*** además de capturar la interacción del usuario con un objeto, realizan una acción más compleja

<http://developer.android.com/guide/topics/ui/ui-events.html>


# Estructura de una app

## Gestures




The image shows three smartphone screens illustrating different gestures. The first screen shows a list of names with a blue circle highlighting a name. The second screen shows the same list with a blue circle highlighting a name and a blue bar at the bottom. The third screen shows the same list with a blue arrow pointing left, indicating a swipe gesture.


**Touch**  
Triggers the default functionality for a given item.

 **Action**  
Press, lift

**Long press**  
Enters data selection mode. Allows you to select one or more items in a view and act upon the data using a contextual action bar. Avoid using long press for showing contextual menus.

 **Action**  
Press, wait, lift

**Swipe**  
Scrolls overflowing content, or navigates between views in the same hierarchy.

 **Action**  
Press, move, lift

<http://developer.android.com/design/patterns/gestures.html>

# Estructura de una app

## Gestures



### Drag

Rearranges data within a view, or moves data into a container (e.g. folders on Home Screen).



#### Action

Long press, move, lift



### Double touch

Zooms into content. Also used as a secondary gesture for text selection.



#### Action

Two touches in quick succession



### Pinch open

Zooms into content.



#### Action

2-finger press, move outwards, lift



### Pinch close

Zooms out of content.



#### Action

2-finger press, move inwards, lift

<http://developer.android.com/design/patterns/gestures.html>