

INTRODUCCIÓN AL ANDROID



¿QUÉ VAMOS A TRATAR?

- Programación utilizando XML vs código JAVA para UI
- Explicación de los diferentes Layouts
- Explorar widgets básicos (TextEdit, EditText y Button)
- DDMS
- Diseñador de pantallas
- Patrones de diseño de pantallas Android

VIEWS

- Cada Activity normalmente referencia varias Views
- Son lo que el usuario ve e interactúa
- Gestiona el layout (xml), provee elementos de texto, botones, pinta gráficos en la pantalla.
- Recepciona los eventos originados por el usuario -> event listeners (ejm controles de touch screen)
- Un activity controla las diferentes Views

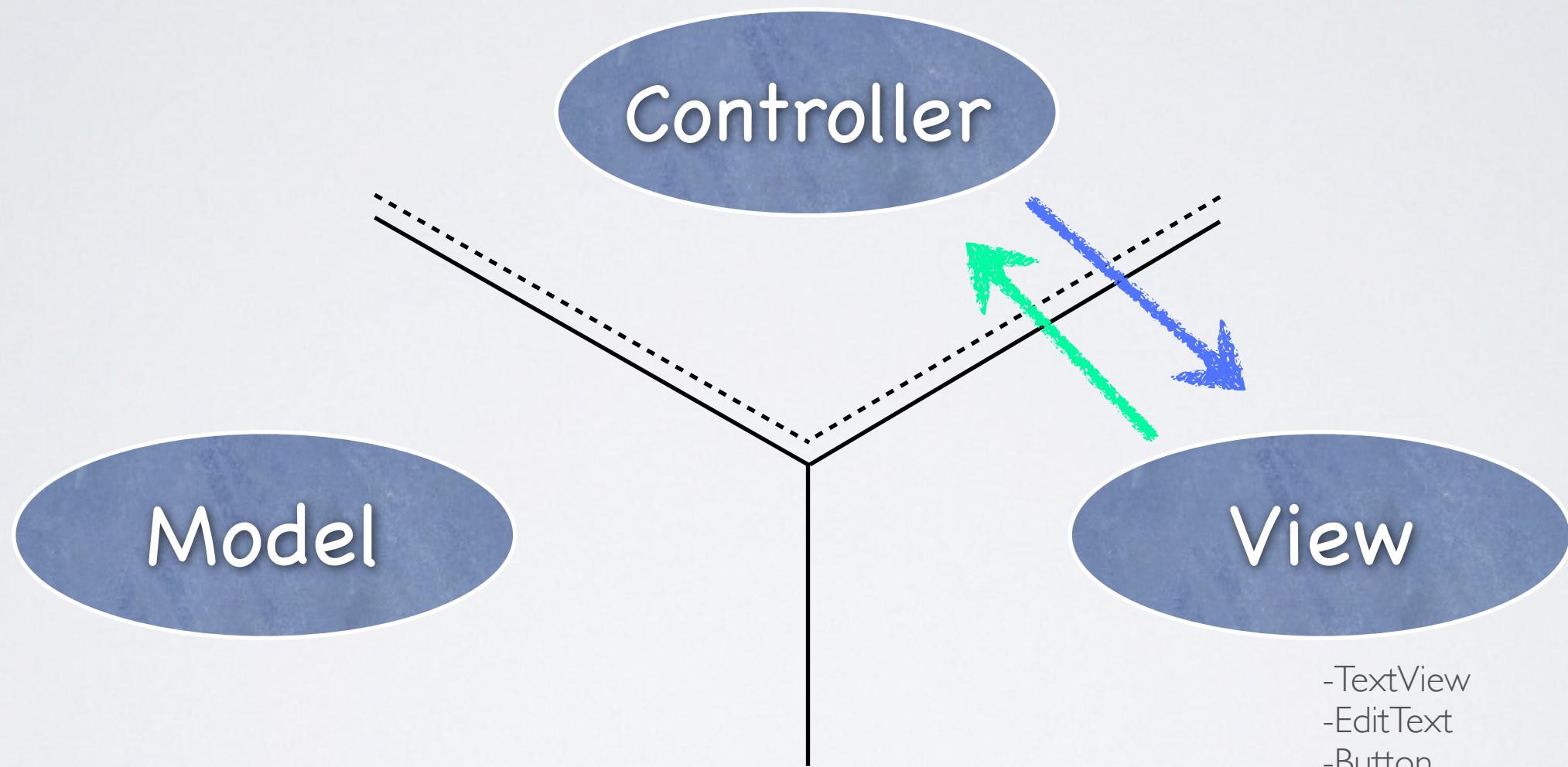
-Activity

Controller

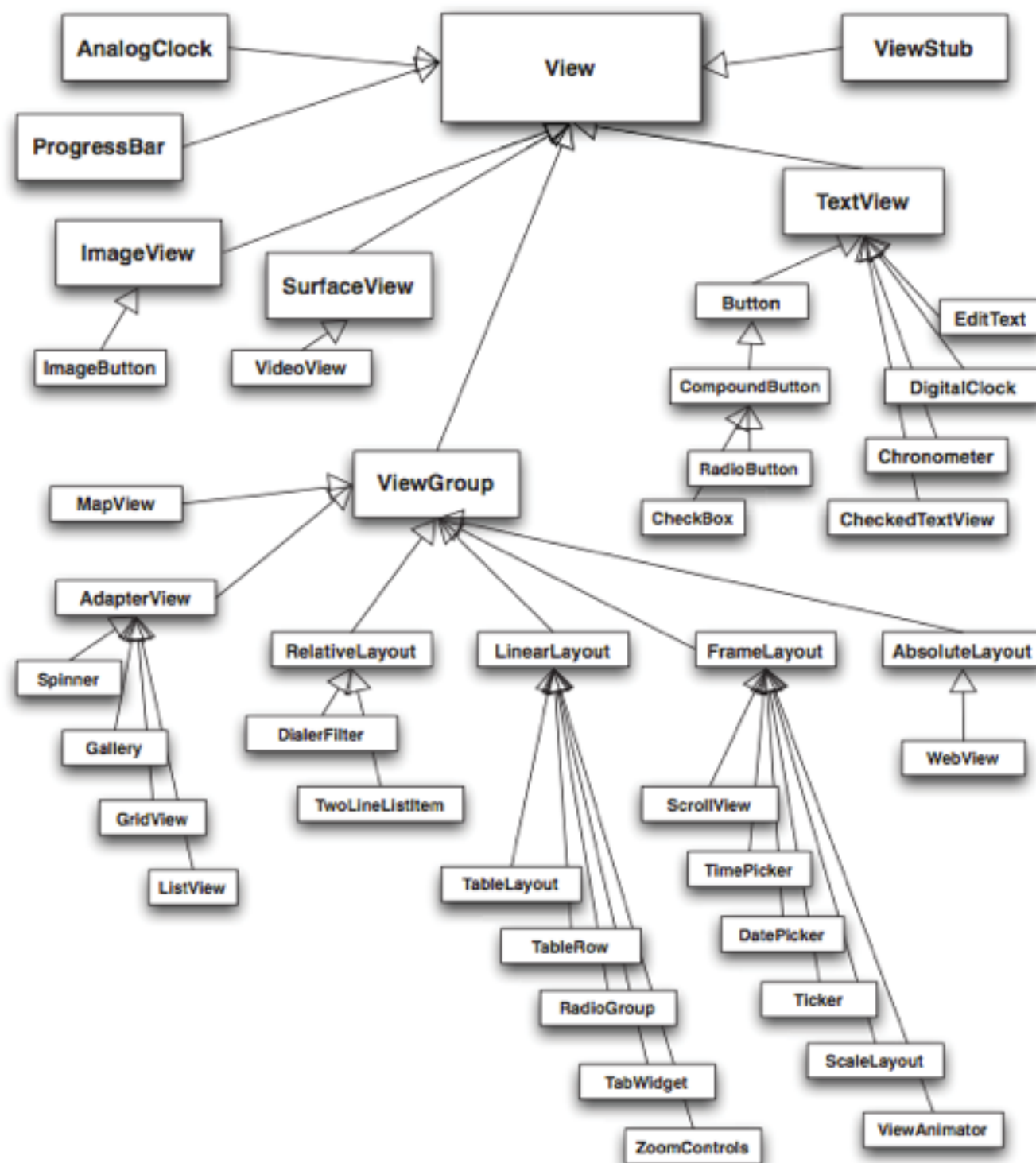
Model

View

-TextView
-EditText
-Button



TIPOS DE VIEWS



CONTROLES MÁS USADOS

- Botones (`Button` / `ImageButton`)
- Cuadros de texto (`EditText`)
 - Tipos de teclado (`inputType`)
- Checkboxes (`CheckBox`)
- Radio buttons (`RadioButton` / `RadioGroup`)
- Botones on/off (`ToggleButton`)

LAYOUTS

- Ordenan la UI
- Organizadores (agrupadores) de “widgets” gráficos (Views).
- Ordenan sus objetos hijos según una lógica.
- Se crearon para solucionar problemas de distintos tamaños de pantallas.

LAYOUTS

LinearLayout

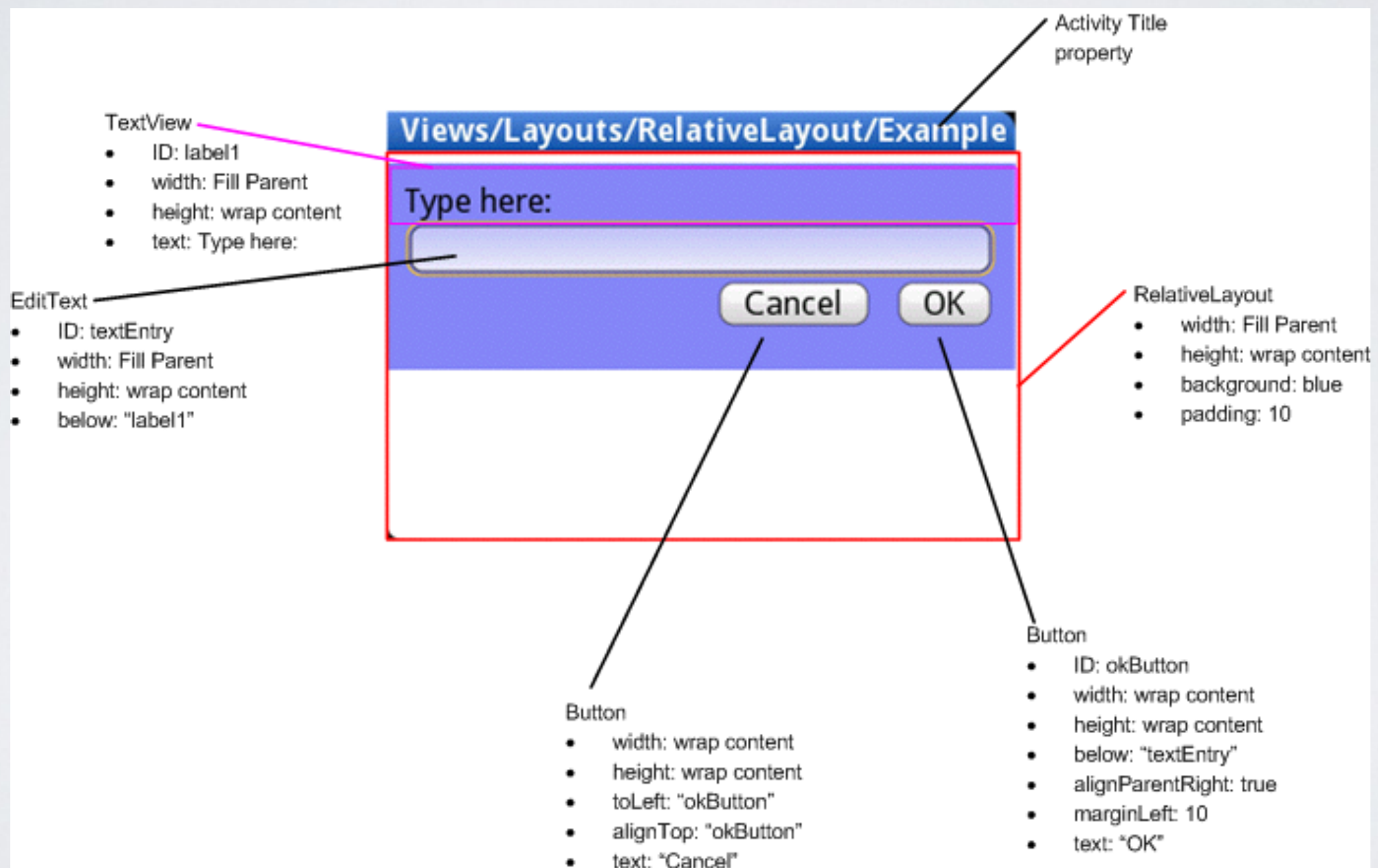
The diagram illustrates a LinearLayout containing two identical forms side-by-side. Each form is a light blue rounded rectangle with a grey header bar labeled "Restaurant Review". Below the header is a yellow button labeled "Click to add". Underneath the button are two white input fields: one labeled "Name" and one labeled "Comments". The "Comments" field is a larger text area. The entire layout is labeled "Linear Layout" at the bottom.

The screenshot shows an Android application titled "LinearLayout Demo". The status bar at the top displays "3G", signal strength, battery, and the time "10:57 PM". The app's interface has a black background. At the top, there is a grey bar with the text "LinearLayout Demo". Below this bar are three white input fields arranged horizontally. The first field is labeled "Back" and contains a back arrow icon. The second field is labeled "First Name" and is currently active, with a cursor visible. The third field is labeled "Last Name".

LINEARLAYOUT

- Atributos principales
- **android:orientation** : Orden de los Views internos. Puede ser ordenamiento horizontal o vertical.
- **android:gravity** : Especifica cómo se ubican los View hijos, en referencia a su contenedor.

RELATIVELAYOUT



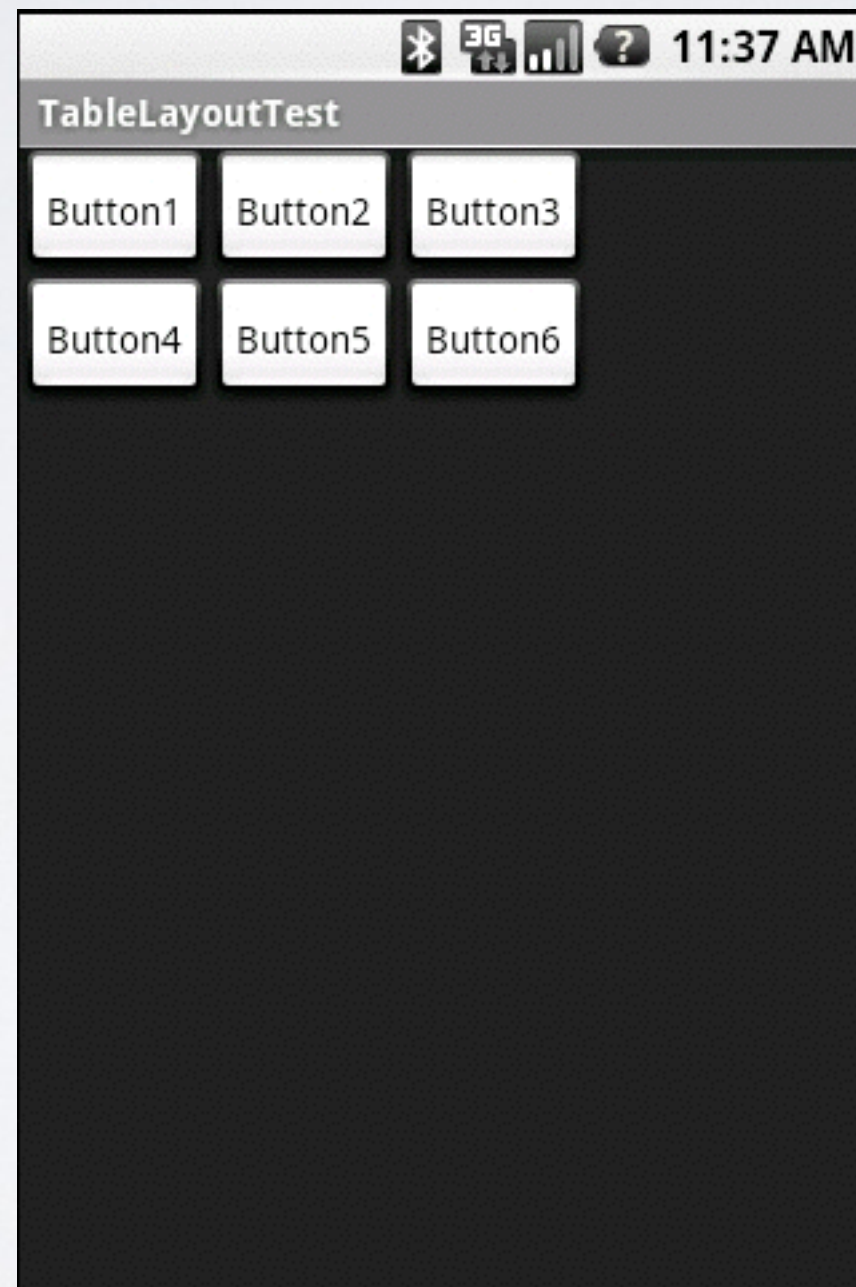
RELATIVELAYOUT

- Posición de un hijos está en relación a otro hijo (hermano) o a su padre (RelativeLayout).
- Nos permite eliminar los Layouts dentro de otros Layouts

ATRIBUTOS PRINCIPALES

<u>android:layout_above</u>	Posiciona el eje de abajo encima del de la vista dada
<u>android:layout_alignBottom</u>	Hace que el eje inferior de la vista este alineado con el de la vista dada
<u>android:layout_alignLeft</u>	Hace que el eje izquierdo de la vista este alineado con el de la vista dada
<u>android:layout_alignParentBottom</u>	Si es true, hace que el eje de abajo de esta vista este alineado con el eje inferior del padre.
<u>android:layout_alignParentLeft</u>	Si es true, hace que el eje de izquierda de esta vista este alineado con el eje izquierda del padre.
<u>android:layout_alignParentRight</u>	Si es true, hace que el eje de derecha de esta vista este alineado con el eje derecha del padre.
<u>android:layout_alignParentTop</u>	Si es true, hace que el eje superior de esta vista este alineado con el eje superior del padre.
<u>android:layout_alignRight</u>	Eje derecho de la vista alineado con el eje derecho de la vista dada.
<u>android:layout_alignTop</u>	Eje superior de la vista alineado con el eje superior de la vista dada.
<u>android:layout_below</u>	Posiciona el eje superior de la visa con el eje inferior de la vista dada.
<u>android:layout_centerHorizontal</u>	Si es true, centra el hijo horizontalmente en el padre
<u>android:layout_centerInParent</u>	Si es true, centra el hijo horizontalmente y verticalmente en el padre
<u>android:layout_centerVertical</u>	Si es true, centra el hijo verticalmente en el padre
<u>android:layout_toLeftOf</u>	Posiciona el eje derecho de la vista con el eje izquierdo de la vista dada.
<u>android:layout_toRightOf</u>	Posiciona el eje izquierdo de la vista con el eje derecho de la vista dada.

TABLELAYOUT

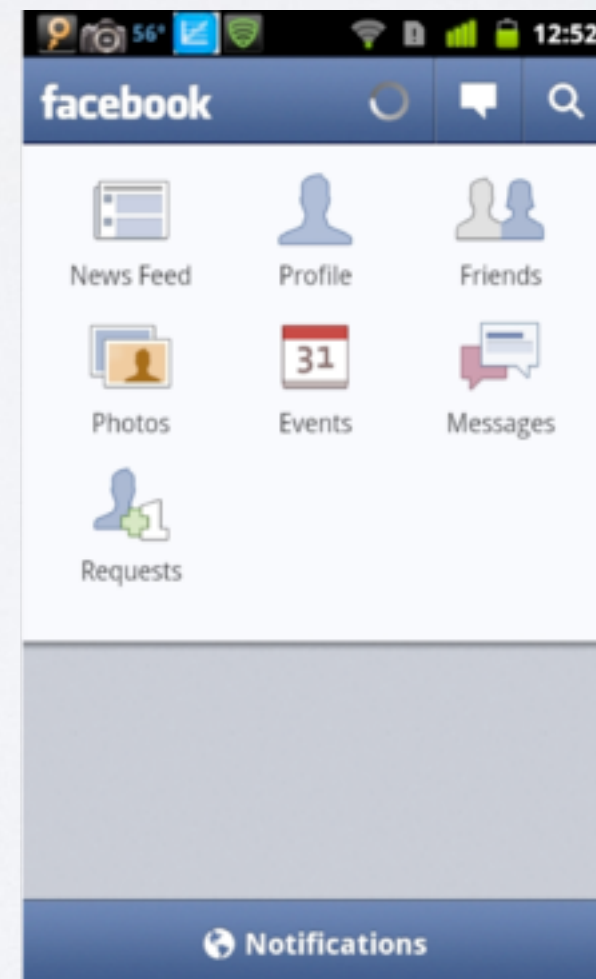
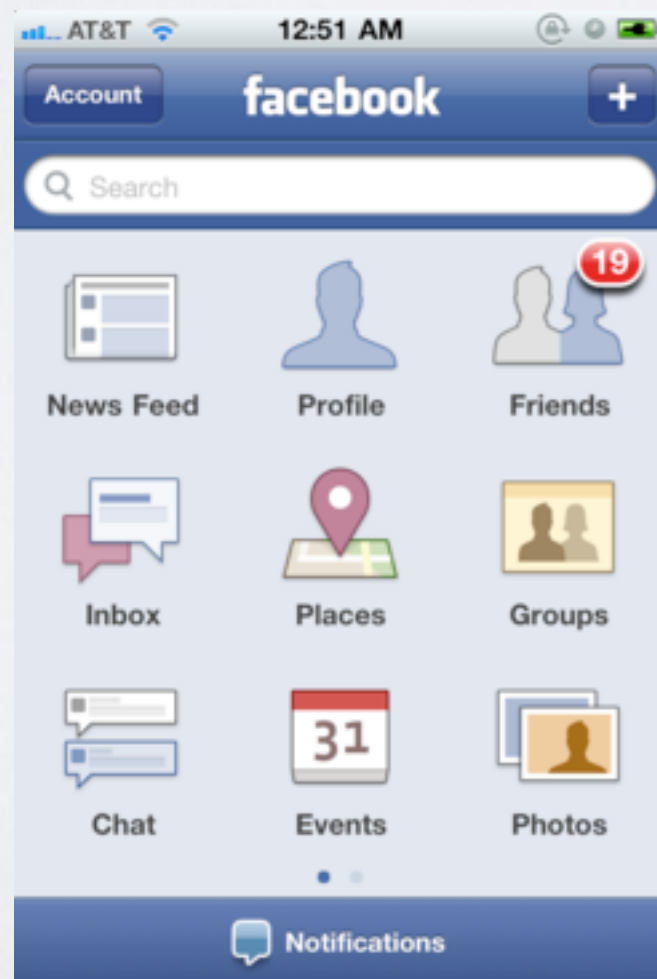


TABLELAYOUT

- Organiza sus hijos en filas y columnas.
- No muestra bordes para sus celdas.
- Cada celda solamente puede contener un objeto View.

PATRONES DE DISEÑO ANDROID

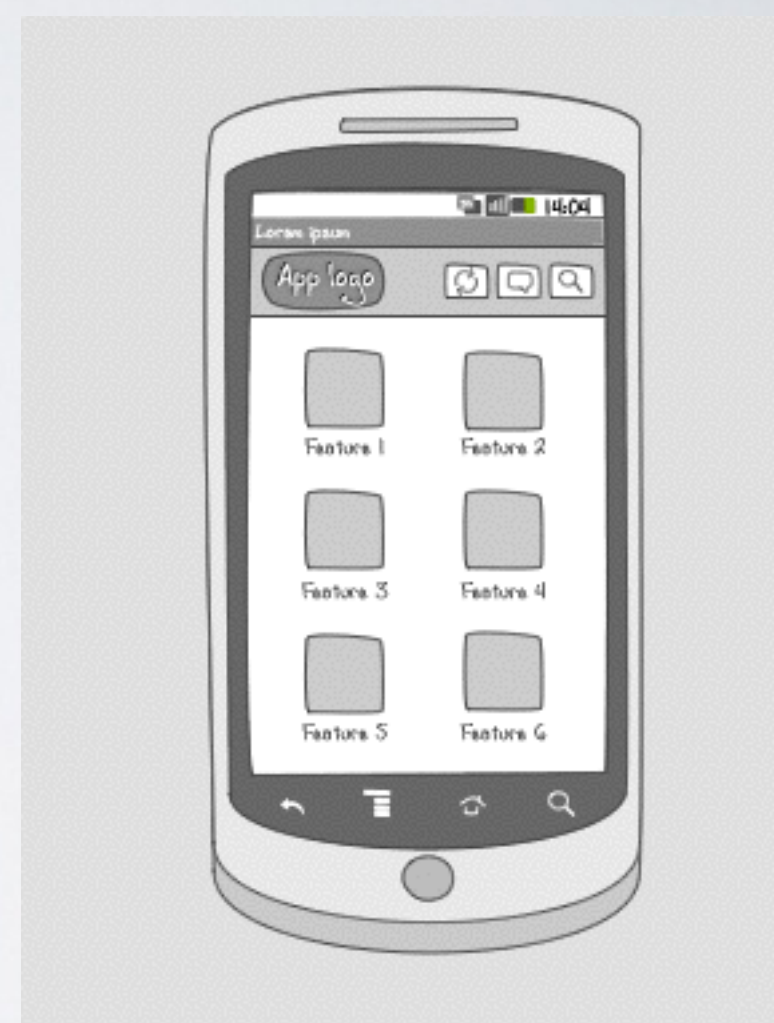
- Patrones que definen un aplicativo Android
- Es para darle una experiencia *conocida* al usuario



ESTRUCTURA COMÚN DE UNA APP

- Aplicaciones de una vista (la calculadora).
- Aplicaciones que cambian entre distintas vista sin profundizar mucho en los datos (el teléfono).
- Aplicaciones que contienen gran cantidad de información lo cual nos permite una navegabilidad compleja.

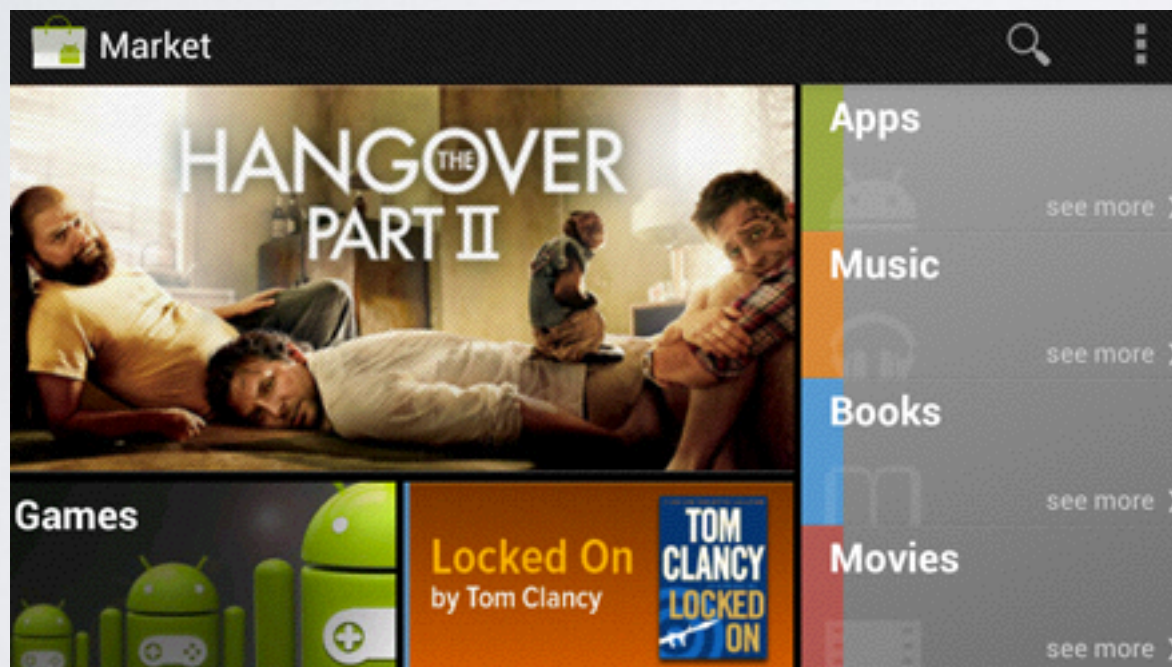
ANTES



TIPOS DE PANTALLAS

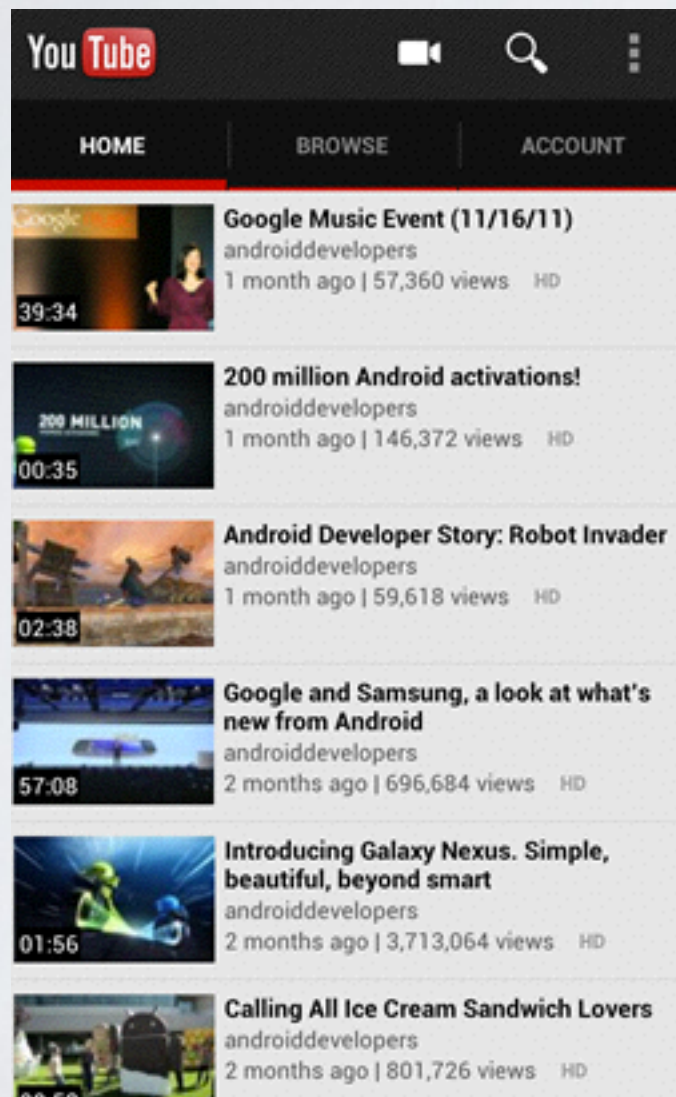
Top Level

- Pantalla inicial
- Es lo que ve los usuarios luego de iniciar la app
- Poner contenido
- No usarla solamente como navegación.
- Impresionar al usuario.
- “Qué es lo que mis usuarios más probablemente harán en mi aplicación.



TIPOS DE PANTALLA

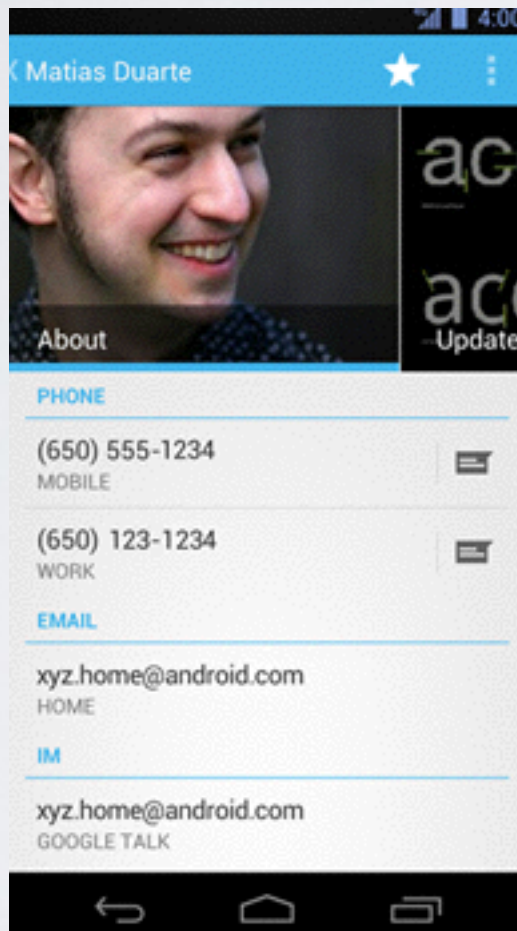
Categorías



- Nos permite navegar entre diferentes categorías.
- Se debe ver directamente el nivel de detalle.
- Dirigido a la organización de los datos.
- Si las categorías están relacionadas, la barra de tabs pueden ser scrolleables. Caso contrario es necesario dejarlas fijas.

TIPOS DE PANTALLA

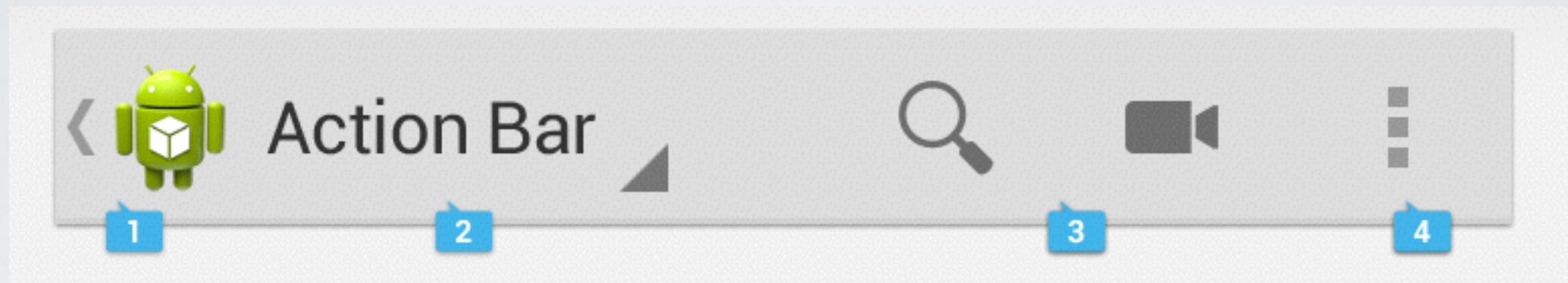
Pantalla de Detalle



- Nos permite ver e interactuar con nuestra data
- El layout utilizado depende del tipo de data que se está mostrando.
- Hacer sencilla la navegación entre las diferentes pantallas de detalle.

NAVEGABILIDAD

Action Bar



1. Logo de la aplicación
2. Control de vistas. Si nuestra aplicación muestra los datos en diferentes vistas.
3. Botones de acción. Los botones más importantes dentro de TODA nuestra aplicación.
4. Acciones menos utilizadas.