

MIT

Global Startup Labs

México 2013

<http://gsl.mit.edu>
Coming Soon!

Lección 02 – Android Layouts, Views,
Menus, Usability

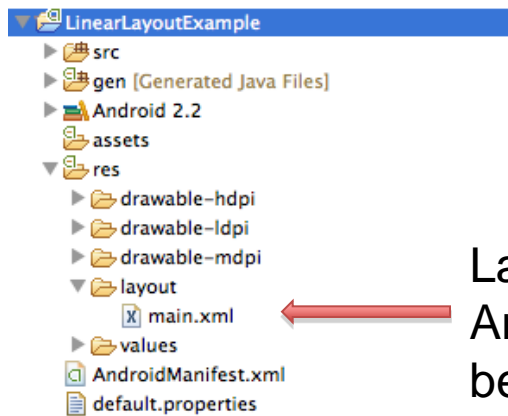


Agenda

- Layouts
- Views and Widgets
- Menus
- Usability
- Lab – Settling In

Layouts

- Definido en 2 maneras
 - XML layout files (Declarative)

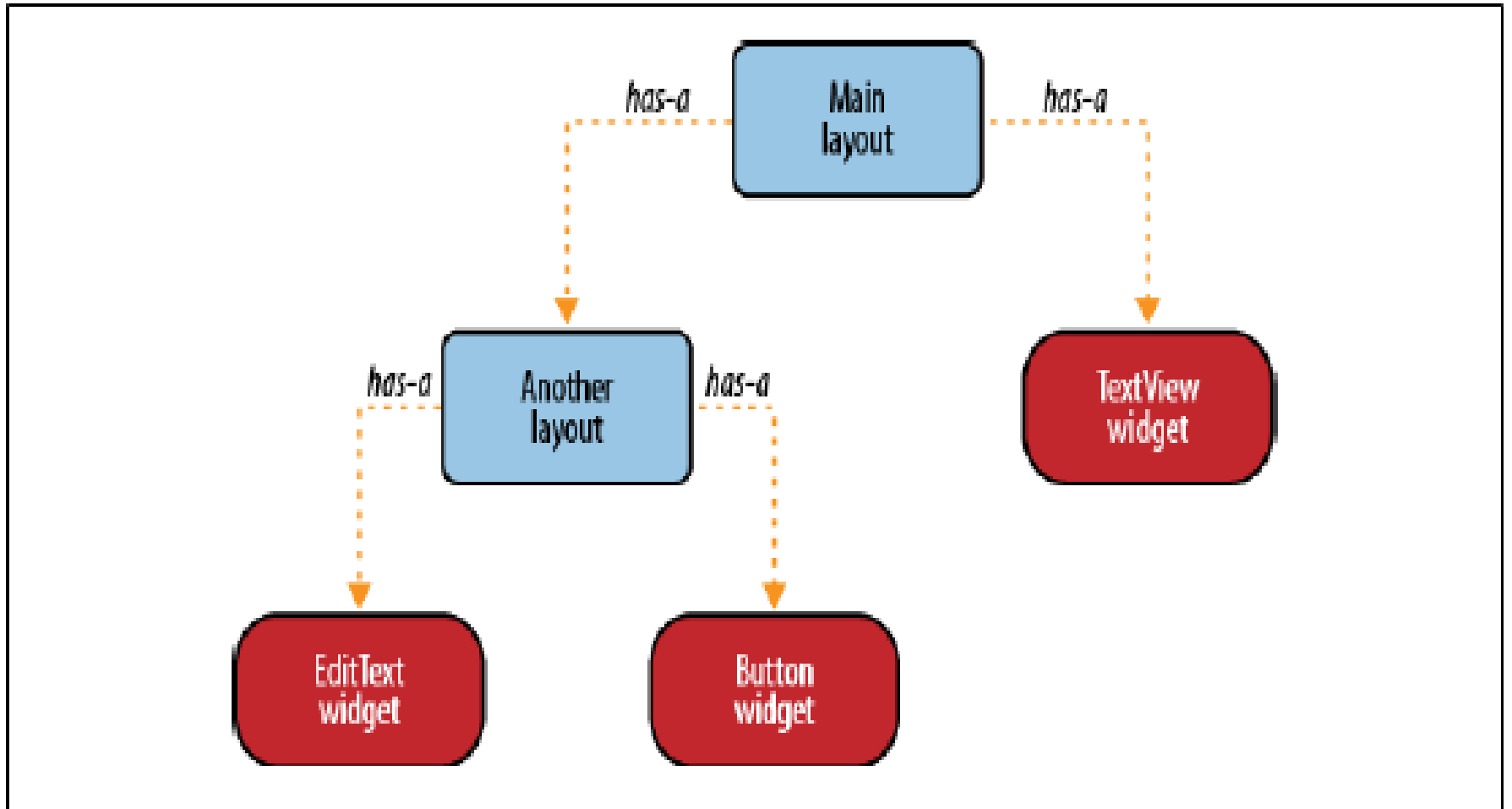


Layout file main.xml is auto-generated when an Android project is created in Eclipse. App layout can be defined in this file in XML.

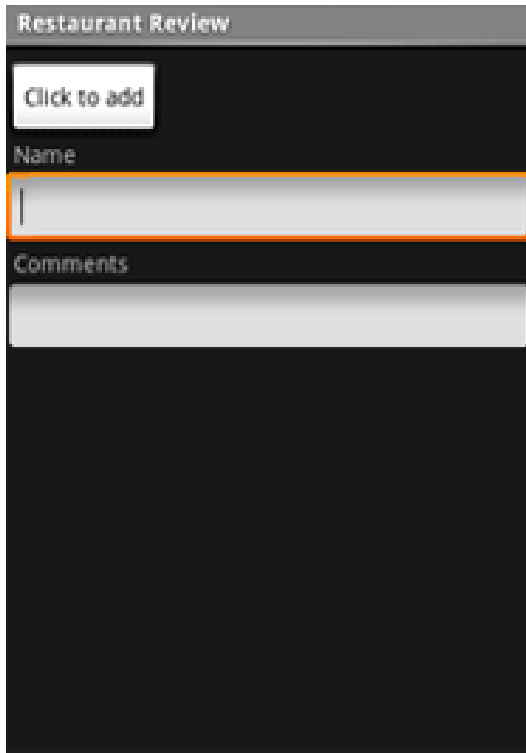
- Using code (e.g. in the onCreate() method)

Think HTML, Java

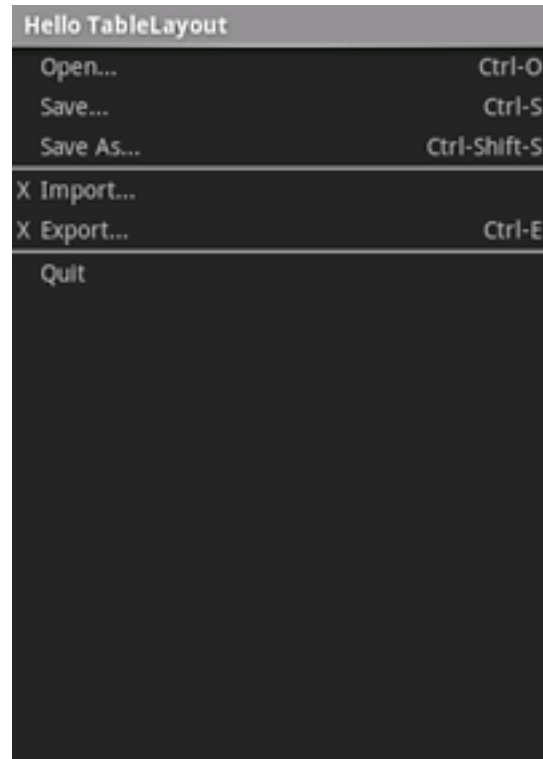
- XML como HTML
- Layout como Java container
- Use XML for static content
- Use Layout for dynamic content, to organize,group



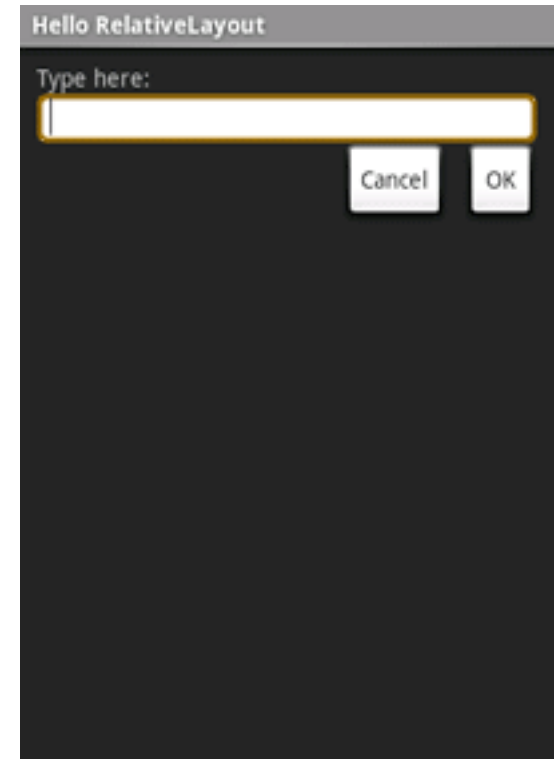
Some Layouts



LinearLayout



TableLayout



RelativeLayout

LinearLayout

- Arrange components one after another, left-to-right, top-to-bottom:

```
<?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="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>
```

Hello, I am a TextView

Hello, I am a Button

TableLayout

- Position components in rows and columns:

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:stretchColumns="1">
    <TableRow>
        <TextView
            android:text="@string/table_layout_4_open"
            android:padding="3dip" />
        <TextView
            android:text="@string/table_layout_4_open_shortcut"
            android:gravity="right"
            android:padding="3dip" />
    </TableRow>

    <TableRow>
        <TextView
            android:text="@string/table_layout_4_save"
            android:padding="3dip" />
        <TextView
            android:text="@string/table_layout_4_save_shortcut"
            android:gravity="right"
            android:padding="3dip" />
    </TableRow>
</TableLayout>
```



RelativeLayout

- Position and align components relative to other components:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/blue"
    android:padding="10px" >

    <TextView android:id="@+id/label"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Type here:" />

    <EditText android:id="@+id/entry"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="@android:drawable/editbox_background"
        android:layout_below="@id/label" />

</RelativeLayout>
```



`android:layout_below` is an attribute that can be used only with `RelativeLayout`. Other such attributes include `layout_alignParentRight`, and `layout_toLeftOf`.

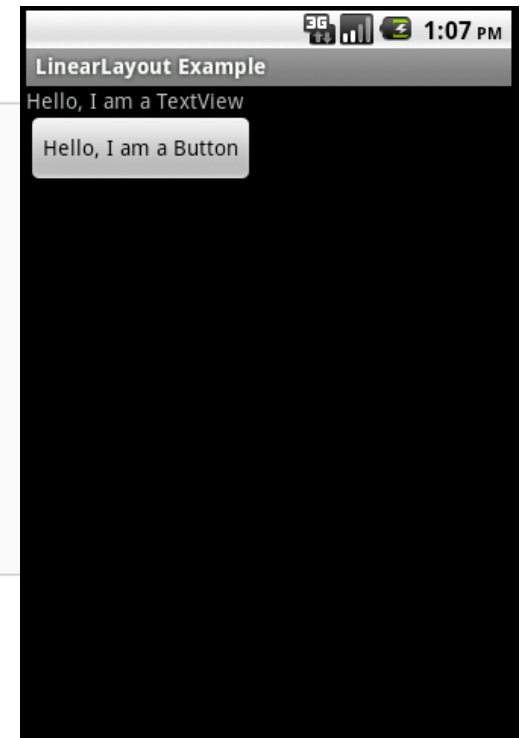
Views

- What they are: UI components
- Some common views and widgets:
 - Button
 - EditText (a text box)
 - TextView (a text label)
 - ListView
 - GridView
 - TabView
 - Spinner (a drop-down menu)
 - CheckBox
 - RadioButton
 - ToggleButton
 - RatingBar
 - MapView (for embedding Google Maps objects in applications)
 - WebView (for embedding web browsers in applications)

Adding Views to Layouts

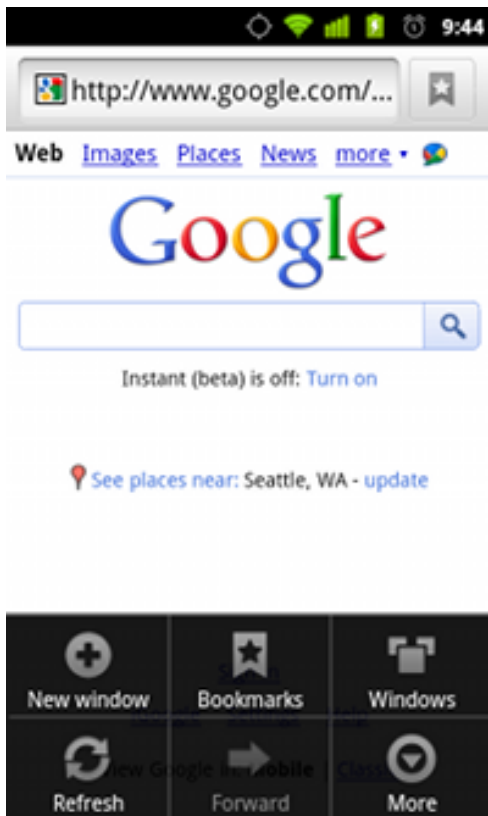
- Example: adding a button and text label to a LinearLayout:

```
<?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="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>
```

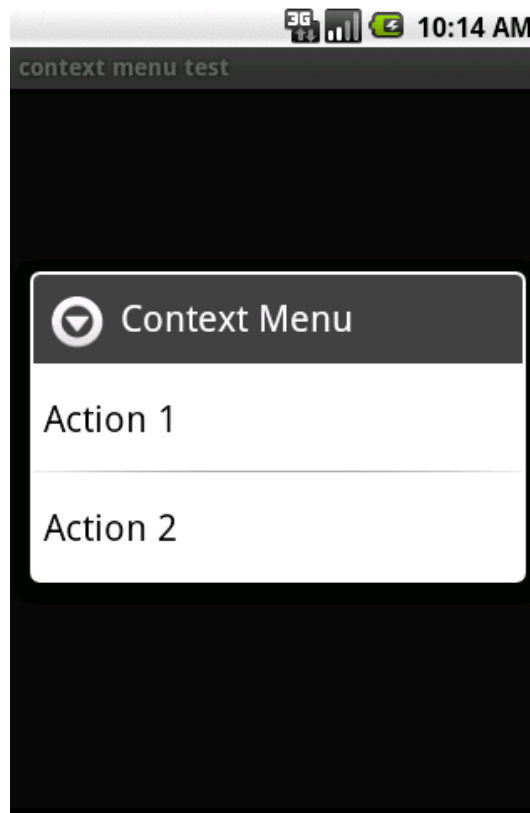


Menus

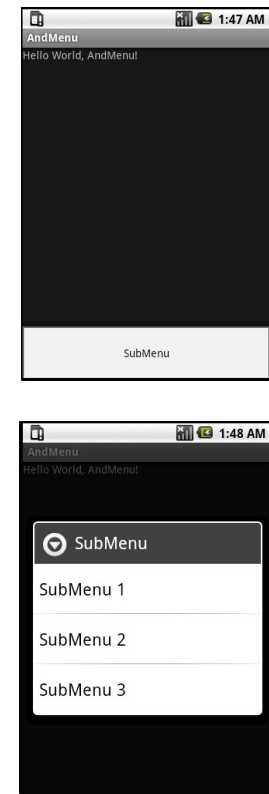
Options Menu



Context Menu



SubMenu



OptionsMenu Example

- Step 1: Implement `onCreateOptionsMenu()` method

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    boolean result = super.onCreateOptionsMenu(menu);

    menu.add(Menu.NONE, 0, 0, "Activity One");
    menu.add(Menu.NONE, 1, 1, "Activity Two");

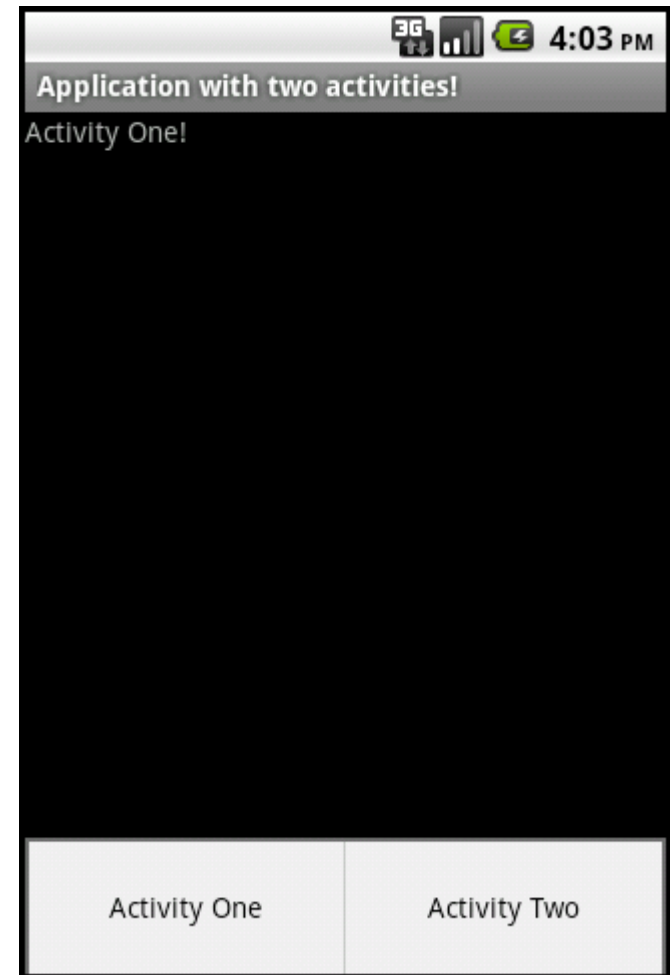
    return result;
}
```

- Step 2: Implement `onOptionsItemSelected()` method

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int itemIndex = item.getItemId();

    if (itemIndex == 0){
        //first menu button pressed. do something here
    }
    else if (itemIndex == 1){
        // second menu button pressed. do something here
    }

    return super.onOptionsItemSelected(item);
}
```



UI - Usability

- Por qué?

UI - Usability

- Simple
- Consistente
- Responsive
- Feedback Inmediato

You are not the User

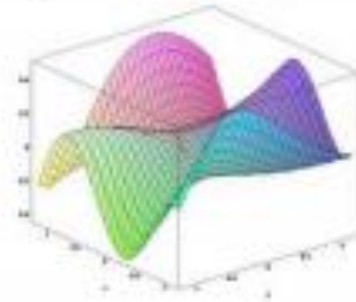
- UI se trata de comunicarte con los usuarios
 - Usuarios NO SON COMO TU
- Usuario siempre tiene la razón

Usability Definido

- Usability: Qué tan bien los usuarios pueden usar la funcionalidad del sistema
- Dimensiones de Usability
 - Learnability: es facil de aprender?
 - Efficiency: después de aprender, es facil de usar?
 - Safety: errores pocos y recuperables
- Otras dimensiones: ergonomics, aesthetics: satisfaction, happiness, pleasure

Usability Is Only One Attribute of a System

- Software designers have a lot to worry about:
 - Functionality
 - Performance
 - Cost
 - Security
 - Dependability
 - Usability
 - Size
 - Reliability
 - Standards
 - Marketability
- Many design decisions involve tradeoffs among different attributes
- We'll take an extreme position in this class



DEVELOPERS

DESIGNERS

PROJECT MANAGERS

QA

SEEN BY
DEVELOPERS



SEEN BY
DESIGNERS



SEEN BY
PROJECT
MANAGERS



SEEN BY
QA



UI Lab

- Cheap prototyping
- User Feedback

Settling In

