

# Tema 2

## Interfaz de Usuario

1. Manifest
2. Activities
3. Fragments
4. Layouts
5. Views

# Tema 2

## Manifest.xml

Declaración de los componentes usados por nuestra app

1. Permisos
2. API Level
3. Requisitos Hardware y Software
4. Bibliotecas adicionales

# Tema 2

## Componentes

```
<?xml version="1.0" encoding="utf-8"?>
<manifest ... >
    <application android:icon="@drawable/app_icon.png" ... >
        <activity android:name="com.example.project.ExampleActivity"
                  android:label="@string/example_label" ... >
            </activity>
            ...
        </application>
    </manifest>
```

# Tema 2

## Capacidades de los componentes

```
<manifest ... >
    ...
    <application ... >
        <activity android:name="com.example.project.ComposeEmailActivity">
            <intent-filter>
                <action android:name="android.intent.action.SEND" />
                <data android:type="*/*" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

# Tema 2

## Requisitos

```
<manifest ... >
    <uses-feature android:name="android.hardware.camera.any"
                  android:required="true" />
    <uses-sdk android:minSdkVersion="7" android:targetSdkVersion="19" />
    ...
</manifest>
```

# Tema 2

Requisitos (hasta Lollipop 5.1, 22)

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android.app.myapp" >
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    ...
</manifest>
```

<https://developer.android.com/reference/android/Manifest.permission.html>

# Tema 2

## Requisitos (Marshallow 6.0, 23)

```
// Assume thisActivity is the current activity
int permissionCheck = ContextCompat.checkSelfPermission(thisActivity,
    Manifest.permission.WRITE_CALENDAR);
```

[https://developer.android.com/training/permissions/  
requesting.html](https://developer.android.com/training/permissions/requesting.html)

# Tema 2

## Requisitos (Lollipop 5.0)

Permission Group	Permissions
CALENDAR	<ul style="list-style-type: none"><li>• READ_CALENDAR</li><li>• WRITE_CALENDAR</li></ul>
CAMERA	<ul style="list-style-type: none"><li>• CAMERA</li></ul>
CONTACTS	<ul style="list-style-type: none"><li>• READ_CONTACTS</li><li>• WRITE_CONTACTS</li><li>• GET_ACCOUNTS</li></ul>
LOCATION	<ul style="list-style-type: none"><li>• ACCESS_FINE_LOCATION</li><li>• ACCESS_COARSE_LOCATION</li></ul>
MICROPHONE	<ul style="list-style-type: none"><li>• RECORD_AUDIO</li></ul>
PHONE	<ul style="list-style-type: none"><li>• READ_PHONE_STATE</li><li>• CALL_PHONE</li><li>• READ_CALL_LOG</li><li>• WRITE_CALL_LOG</li><li>• ADD_VOICEMAIL</li><li>• USE_SIP</li><li>• PROCESS_OUTGOING_CALLS</li></ul>
SENSORS	
SMS	<ul style="list-style-type: none"><li>• BODY_SENSORS</li><li>• SEND_SMS</li><li>• RECEIVE_SMS</li><li>• READ_SMS</li><li>• RECEIVE_WAP_PUSH</li><li>• RECEIVE_MMS</li></ul>
STORAGE	
	<ul style="list-style-type: none"><li>• READ_EXTERNAL_STORAGE</li><li>• WRITE_EXTERNAL_STORAGE</li></ul>

# Tema 2

## Ejemplo

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="cyclope.android.itaca.es.cyclope.sim" >

    <uses-permission android:name="android.permission.INTERNET" />

    <!-- Keeps the processor from sleeping when a message is received. -->
    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />

    <!-- External storage for caching. -->
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_GPS" />
    <uses-permission android:name="android.permission.ACCESS_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />

    <application
        android:name=".CiclopeApp"
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="GeosysTrack"
        android:theme="@style/AppTheme" >
        <meta-data
            android:name="AA_DB_NAME"
            android:value="geosystec.db" />
        <meta-data
            android:name="AA_DB_VERSION"
            android:value="5" />

        <activity
            android:name=".activities.RecomendacionesListActivity"
            android:configChanges="keyboard|keyboardHidden|orientation|screenSize"
            android:label="GeosysTrack"
            android:screenOrientation="landscape" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".activities.RegisterDeviceActivity"
            android:configChanges="keyboard|keyboardHidden|orientation|screenSize"
            android:label="GeosysTrack :: Registrar dispositivo"
            android:parentActivityName=".activities.RecomendacionesListActivity"
            android:screenOrientation="landscape" >
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value=".activities.RecomendacionesListActivity" />
        </activity>
    </application>
</manifest>
```

```
<service
    android:name=".services.UlisesService"
    android:enabled="true"
    android:exported="true" >
</service>
<service
    android:name=".services.AlarmsService"
    android:exported="false" >
</service>

<receiver
    android:name=".services.AlarmsReceiver"
    android:enabled="true"
    android:exported="true" >
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />
        <action android:name="cyclope.android.itaca.es.cyclope.services.ACTION_ALARM" />
        <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
    </intent-filter>
</receiver>

<activity
    android:name=".activities.CloseTratamientoActivity"
    android:theme="@style/AppFullScreenTheme"
    android:label="GeosysTrack :: Fin del TRATAMIENTO"
    android:configChanges="keyboard|keyboardHidden|orientation|screenSize"
    android:parentActivityName=".activities.RecomendacionesListActivity"
    android:windowSoftInputMode="adjustPan|stateHidden"
    android:screenOrientation="landscape" >
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value=".activities.RecomendacionesListActivity" />
</activity>
</application>

</manifest>
```

# Tema 2

## Activities

### Crear la Activity.

#### `onCreate()`

You must implement this method. The system calls this when creating your activity. Within your implementation, you should initialize the essential components of your activity. Most importantly, this is where you must call `setContentView()` to define the layout for the activity's user interface.

#### `onPause()`

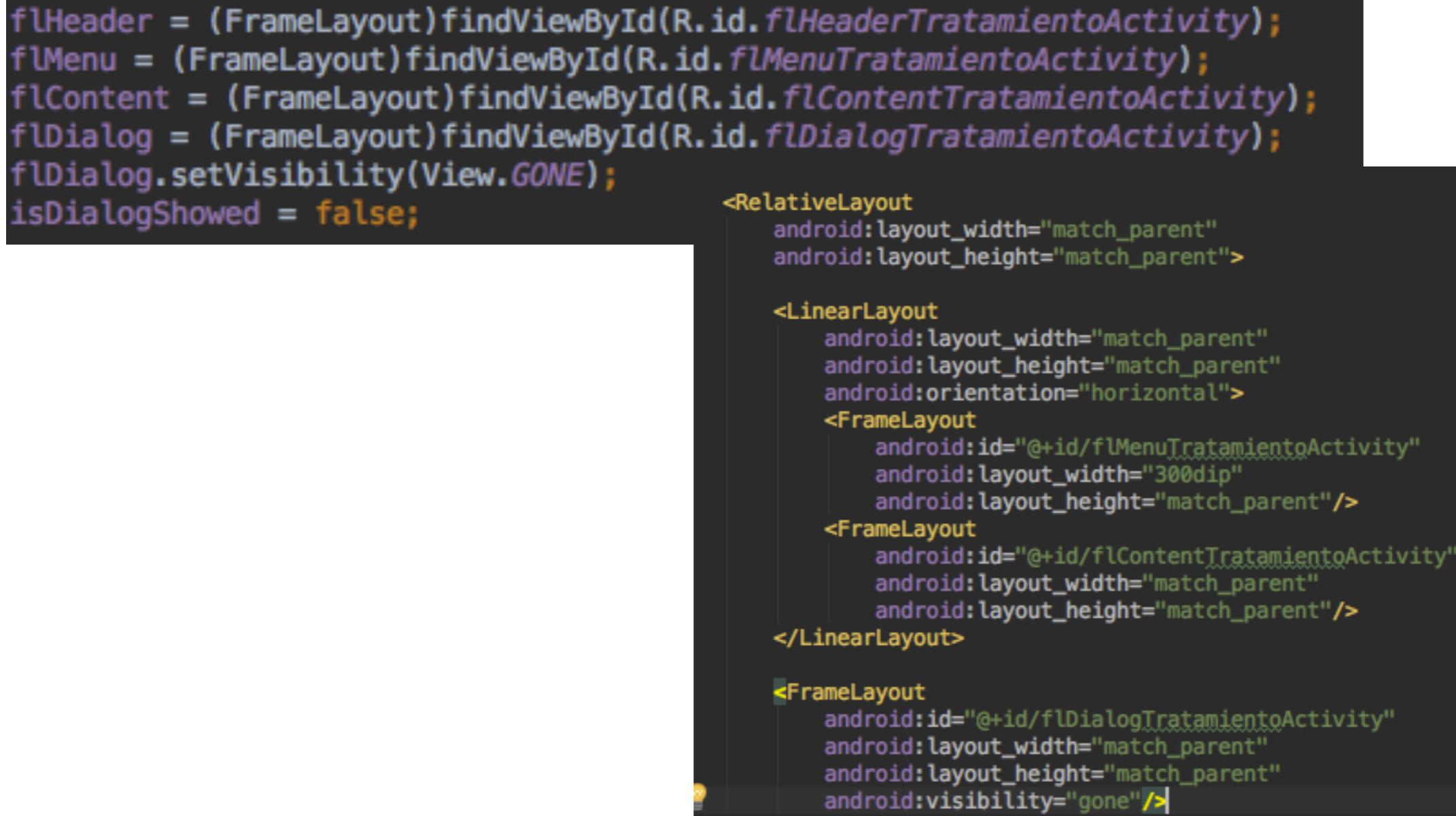
The system calls this method as the first indication that the user is leaving your activity (though it does not always mean the activity is being destroyed). This is usually where you should commit any changes that should be persisted beyond the current user session (because the user might not come back).

# Tema 2

## Activities

findViewById(R.id.idcomponente)

```
flHeader = (FrameLayout) findViewById(R.id.flHeaderTratamientoActivity);
flMenu = (FrameLayout) findViewById(R.id.flMenuTratamientoActivity);
flContent = (FrameLayout) findViewById(R.id.flContentTratamientoActivity);
flDialog = (FrameLayout) findViewById(R.id.flDialogTratamientoActivity);
flDialog.setVisibility(View.GONE);
isDialogShowed = false;
```



```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">
        <FrameLayout
            android:id="@+id/flMenuTratamientoActivity"
            android:layout_width="300dp"
            android:layout_height="match_parent"/>
        <FrameLayout
            android:id="@+id/flContentTratamientoActivity"
            android:layout_width="match_parent"
            android:layout_height="match_parent"/>
    </LinearLayout>

    <FrameLayout
        android:id="@+id/flDialogTratamientoActivity"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:visibility="gone"/>
```

# Tema 2

## Activities

### Declarar Activity en el Manifest

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

# Tema 2

## Activities

### Declarar Activity Principal

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

# Tema 2

## Activities

### Iniciar Activity

```
Intent intent = new Intent(this, SignInActivity.class);
startActivity(intent);
```

```
Intent intent = new Intent(Intent.ACTION_SEND);
intent.putExtra(Intent.EXTRA_EMAIL, recipientArray);
startActivity(intent);
```

# Tema 2

## Activities

### Activity Implicita

```
<activity class=".NotesList" android:label="@string/title_notes_list">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <action android:name="android.intent.action.EDIT" />
        <action android:name="android.intent.action.PICK" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="vnd.android.cursor.dir/vnd.google.note" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.GET_CONTENT" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="vnd.android.cursor.item/vnd.google.note" />
    </intent-filter>
</activity>
```

# Tema 2

## Activities

### Activity Implicita

- **ACTION\_VIEW** `content://contacts/people/1` – Display information about the person whose identifier is "1".
- **ACTION\_DIAL** `content://contacts/people/1` – Display the phone dialer with the person filled in.
- **ACTION\_VIEW** `tel:123` – Display the phone dialer with the given number filled in. Note how the VIEW action does what is considered the most reasonable thing for a particular URI.
- **ACTION\_DIAL** `tel:123` – Display the phone dialer with the given number filled in.
- **ACTION\_EDIT** `content://contacts/people/1` – Edit information about the person whose identifier is "1".

# Tema 2

## Activities

### Iniciar Activity

```
private void pickContact() {  
    // Create an intent to "pick" a contact, as defined by the content provider URI  
    Intent intent = new Intent(Intent.ACTION_PICK, Contacts.CONTENT_URI);  
    startActivityForResult(intent, PICK_CONTACT_REQUEST);  
}  
  
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    // If the request went well (OK) and the request was PICK_CONTACT_REQUEST  
    if (resultCode == Activity.RESULT_OK && requestCode == PICK_CONTACT_REQUEST) {  
        // Perform a query to the contact's content provider for the contact's name  
        Cursor cursor = getContentResolver().query(data.getData(),  
            new String[] {Contacts.DISPLAY_NAME}, null, null, null);  
        if (cursor.moveToFirst()) { // True if the cursor is not empty  
            int columnIndex = cursor.getColumnIndex(Contacts.DISPLAY_NAME);  
            String name = cursor.getString(columnIndex);  
            // Do something with the selected contact's name...  
        }  
    }  
}
```

# Tema 2

## Activities

### Detener Activity

#### finish

Added in API level 1

```
void finish ()
```

Call this when your activity is done and should be closed. The `ActivityResult` is propagated back to whoever launched you via `onActivityResult()`.

#### finishActivity

Added in API level 1

```
void finishActivity (int requestCode)
```

Force finish another activity that you had previously started with `startActivityForResult(Intent, int)`.

Parameters	
<code>requestCode</code>	<code>int</code> : The request code of the activity that you had given to <code>startActivityForResult()</code> . If there are multiple activities started with this request code, they will all be finished.

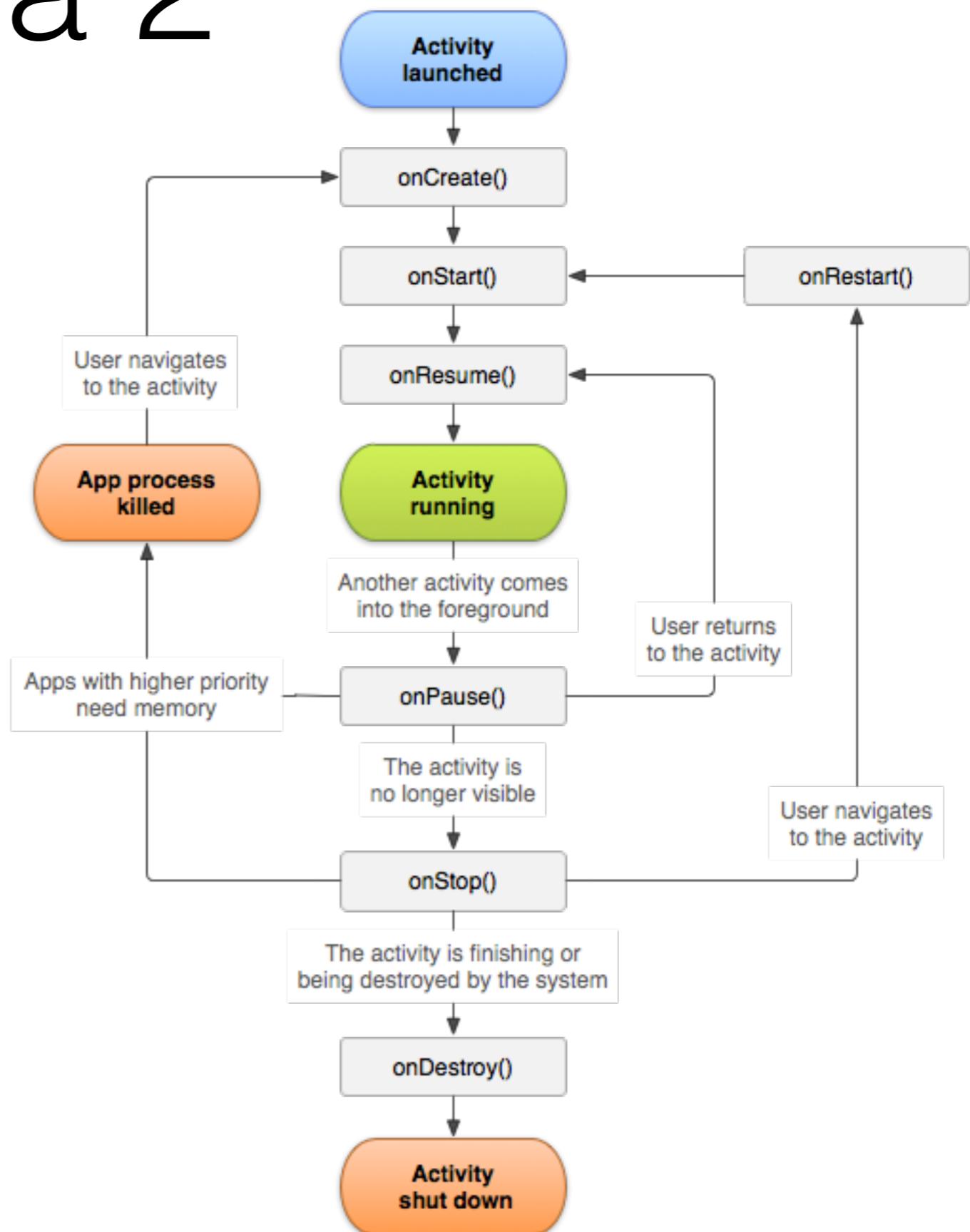
# Tema 2

```
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.  
    }  
}
```

# Tema 2

## Activities

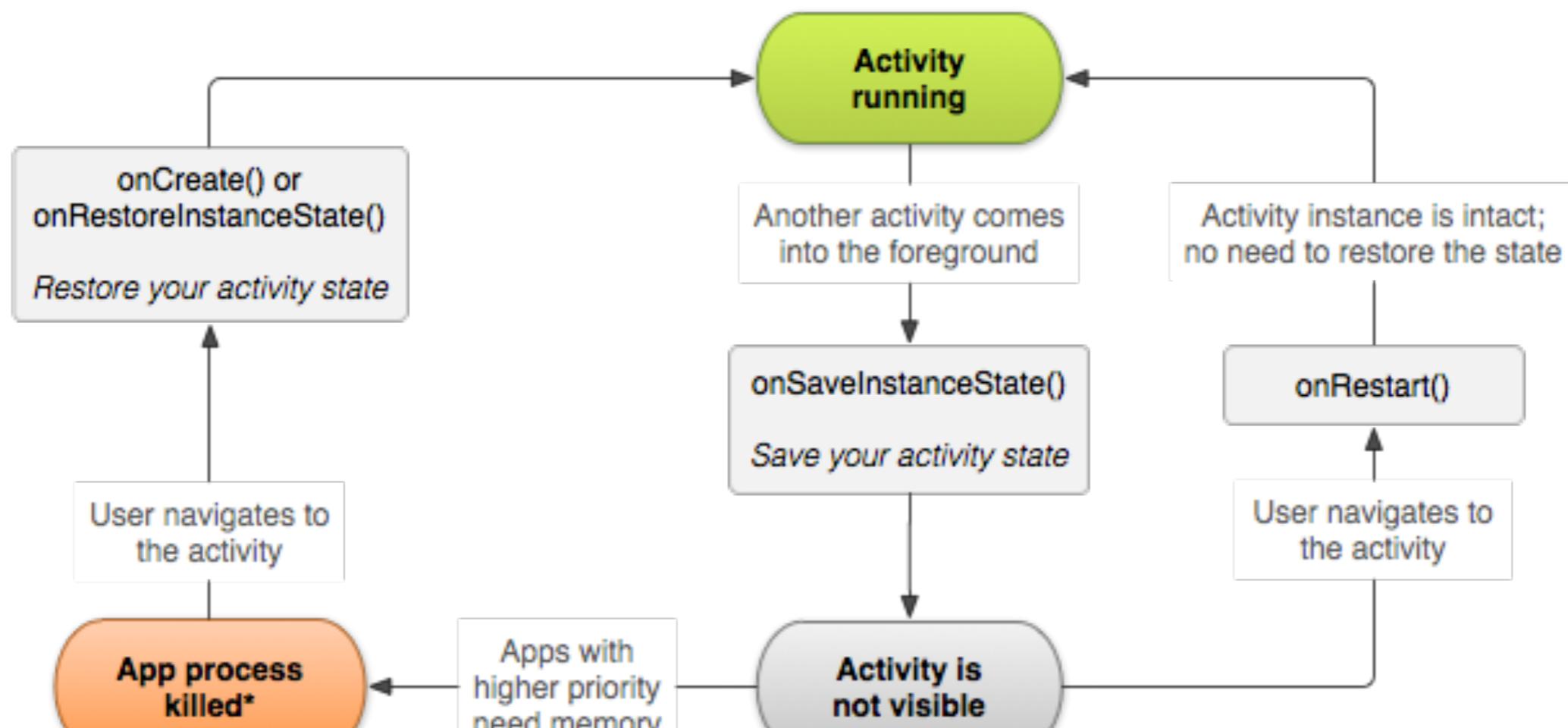
## Ciclo de vida



# Tema 2

## Activities

### Guardar el estado



\*Activity instance is destroyed, but the state from onSaveInstanceState() is saved

# Tema 2

## Activities

### Guardar el estado

```
static final String STATE_SCORE = "playerScore";
static final String STATE_LEVEL = "playerLevel";
...

@Override
public void onSaveInstanceState(Bundle savedInstanceState) {
    // Save the user's current game state
    savedInstanceState.putInt(STATE_SCORE, mCurrentScore);
    savedInstanceState.putInt(STATE_LEVEL, mCurrentLevel);

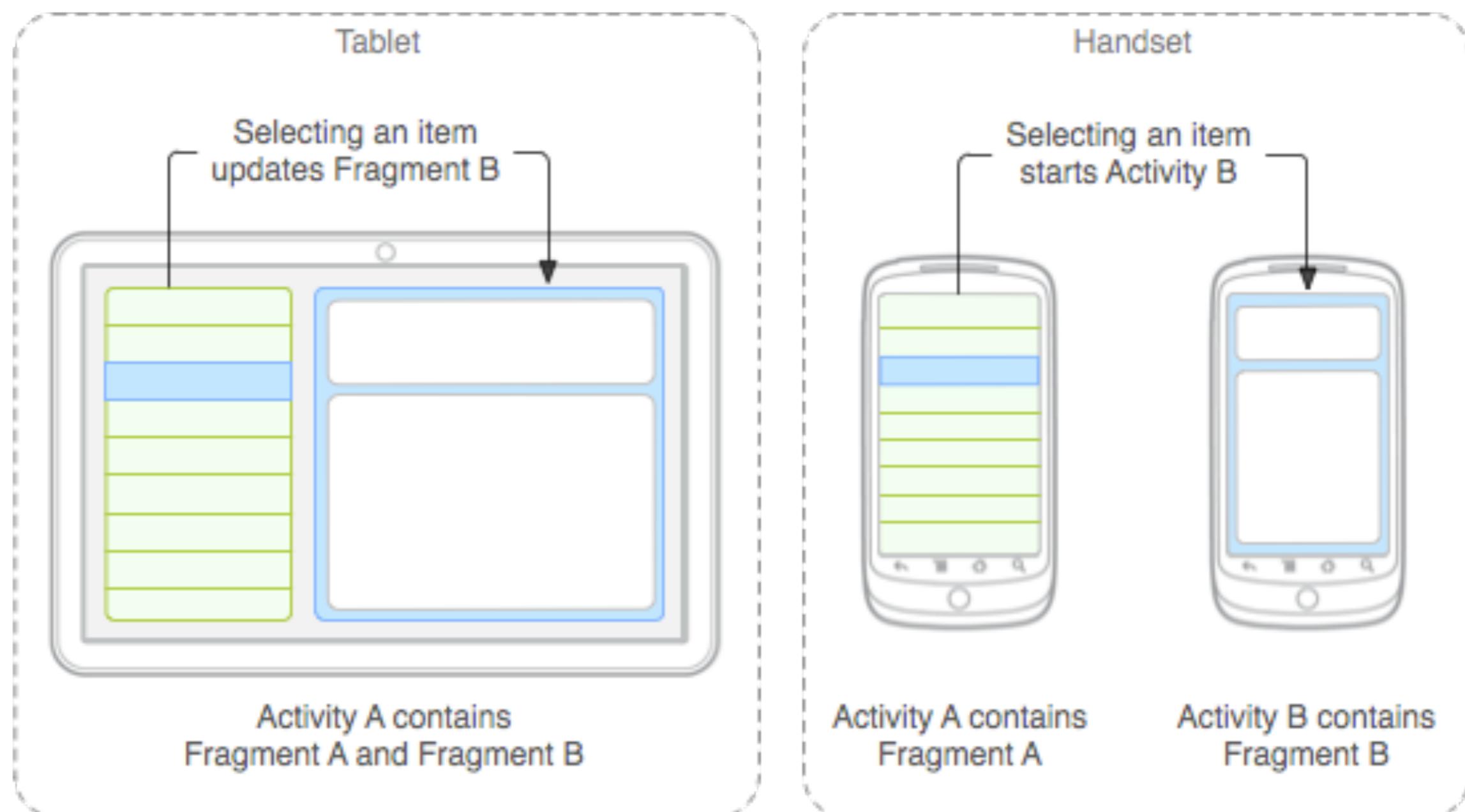
    // Always call the superclass so it can save the view hierarchy state
    super.onSaveInstanceState(savedInstanceState);
}
```

```
public void onRestoreInstanceState(Bundle savedInstanceState) {
    // Always call the superclass so it can restore the view hierarchy
    super.onRestoreInstanceState(savedInstanceState);

    // Restore state members from saved instance
    mCurrentScore = savedInstanceState.getInt(STATE_SCORE);
    mCurrentLevel = savedInstanceState.getInt(STATE_LEVEL);
}
```

# Tema 2

## Fragments



# Tema 2

## Fragments

### onCreate()

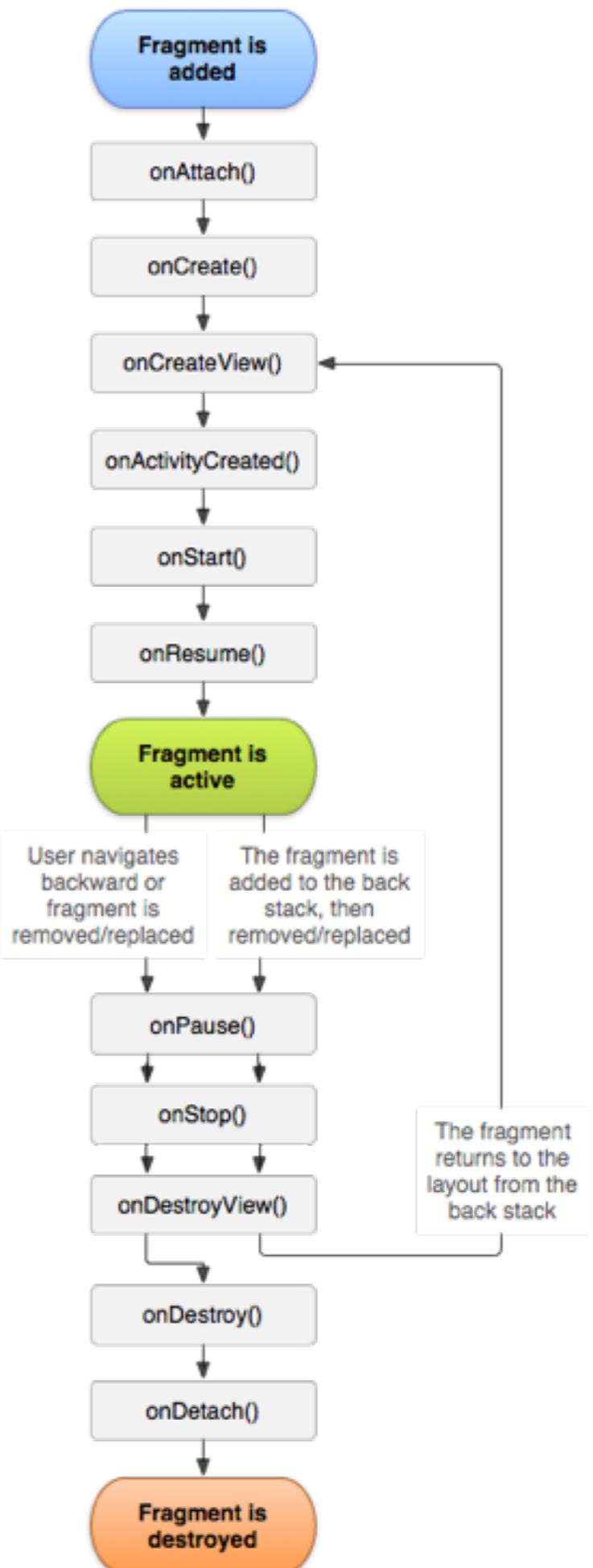
The system calls this when creating the fragment. Within your implementation, you should initialize essential components of the fragment that you want to retain when the fragment is paused or stopped, then resumed.

### onCreateView()

The system calls this when it's time for the fragment to draw its user interface for the first time. To draw a UI for your fragment, you must return a [View](#) from this method that is the root of your fragment's layout. You can return null if the fragment does not provide a UI.

### onPause()

The system calls this method as the first indication that the user is leaving the fragment (though it does not always mean the fragment is being destroyed). This is usually where you should commit any changes that should be persisted beyond the current user session (because the user might not come back).



# Tema 2

## Fragments

### Crear vista

```
public static class ExampleFragment extends Fragment {  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
                             Bundle savedInstanceState) {  
        // Inflate the layout for this fragment  
        return inflater.inflate(R.layout.example_fragment, container, false);  
    }  
}
```

# Tema 2

## Fragments

### Unir a la actividad

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <fragment android:name="com.example.news.ArticleListFragment"
        android:id="@+id/list"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="match_parent" />
    <fragment android:name="com.example.news.ArticleReaderFragment"
        android:id="@+id/viewer"
        android:layout_weight="2"
        android:layout_width="0dp"
        android:layout_height="match_parent" />
</LinearLayout>
```

# Tema 2

## Fragments

### Unir a la actividad

```
FragmentManager fragmentManager = getFragmentManager();
FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
```

```
ExampleFragment fragment = new ExampleFragment();
fragmentTransaction.add(R.id.fragment_container, fragment);
fragmentTransaction.commit();
```

# Tema 2

## Fragments

### Unir a la actividad

```
// Create new fragment and transaction
Fragment newFragment = new ExampleFragment();
FragmentTransaction transaction = getFragmentManager().beginTransaction();

// Replace whatever is in the fragment_container view with this fragment,
// and add the transaction to the back stack
transaction.replace(R.id.fragment_container, newFragment);
transaction.addToBackStack(null);

// Commit the transaction
transaction.commit();
```

# Tema 2

## Fragments

### Hablar con la activity

```
public static class FragmentA extends ListFragment {  
    ...  
    // Container Activity must implement this interface  
    public interface OnArticleSelectedListener {  
        public void onArticleSelected(Uri articleUri);  
    }  
    ...  
}
```

```
public static class FragmentA extends ListFragment {  
    OnArticleSelectedListener mListener;  
    ...  
    @Override  
    public void onAttach(Activity activity) {  
        super.onAttach(activity);  
        try {  
            mListener = (OnArticleSelectedListener) activity;  
        } catch (ClassCastException e) {  
            throw new ClassCastException(activity.toString() + " must implement OnArticleSelectedListener");  
        }  
    }  
    ...  
}
```

# Tema 2

## Fragments

### Resumed

The fragment is visible in the running activity.

### Paused

Another activity is in the foreground and has focus, but the activity in which this fragment lives is still visible (the foreground activity is partially transparent or doesn't cover the entire screen).

### Stopped

The fragment is not visible. Either the host activity has been stopped or the fragment has been removed from the activity but added to the back stack. A stopped fragment is still alive (all state and member information is retained by the system). However, it is no longer visible to the user and will be killed if the activity is killed.

### Activity State

Created

Started

Resumed

Paused

Stopped

Destroyed

### Fragment Callbacks

onAttach()

onCreate()

onCreateView()

onActivityCreated()

onStart()

onResume()

onPause()

onStop()

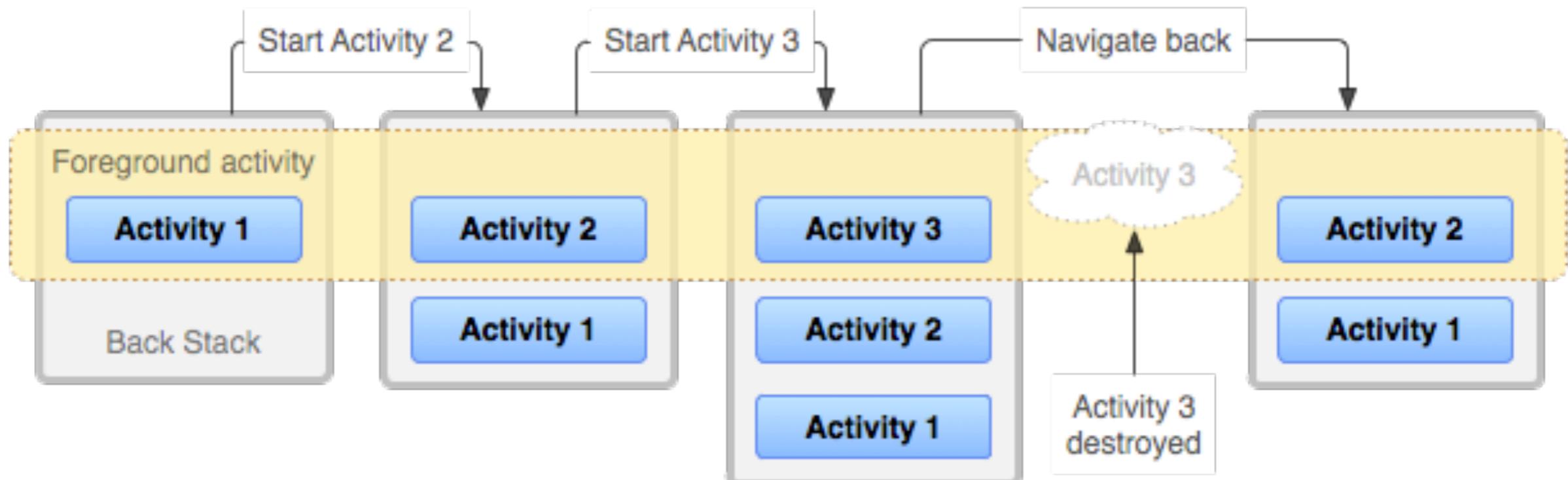
onDestroyView()

onDestroy()

onDetach()

# Tema 2

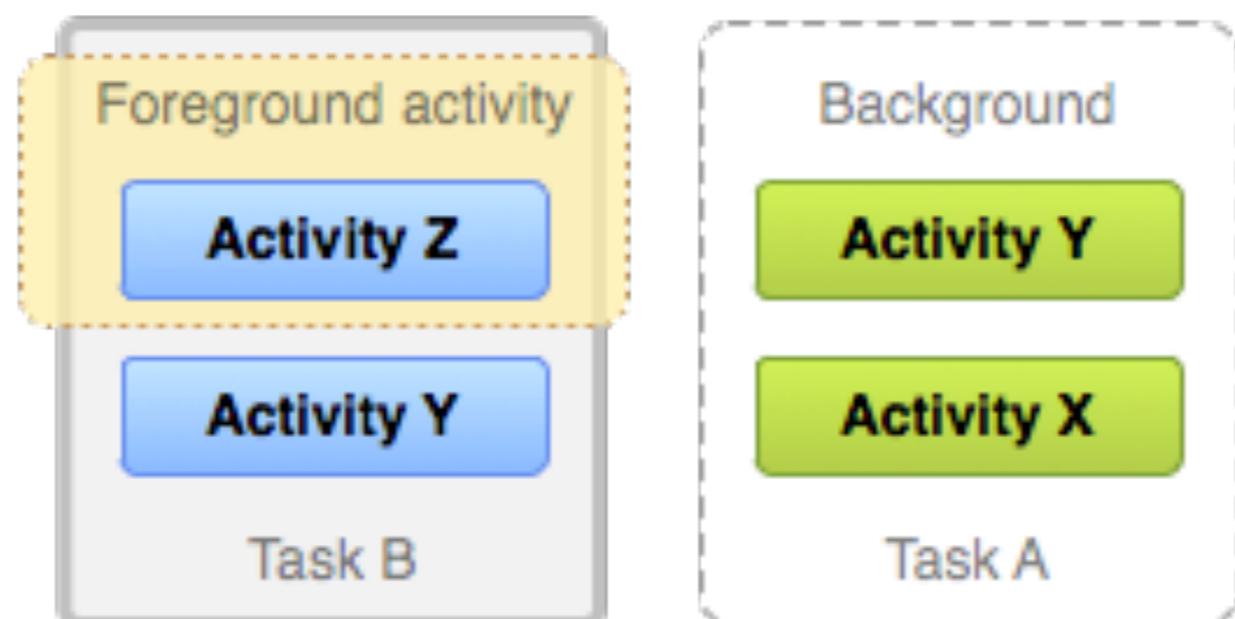
## Task & Back Stack



# Tema 2

## Task & Back Stack

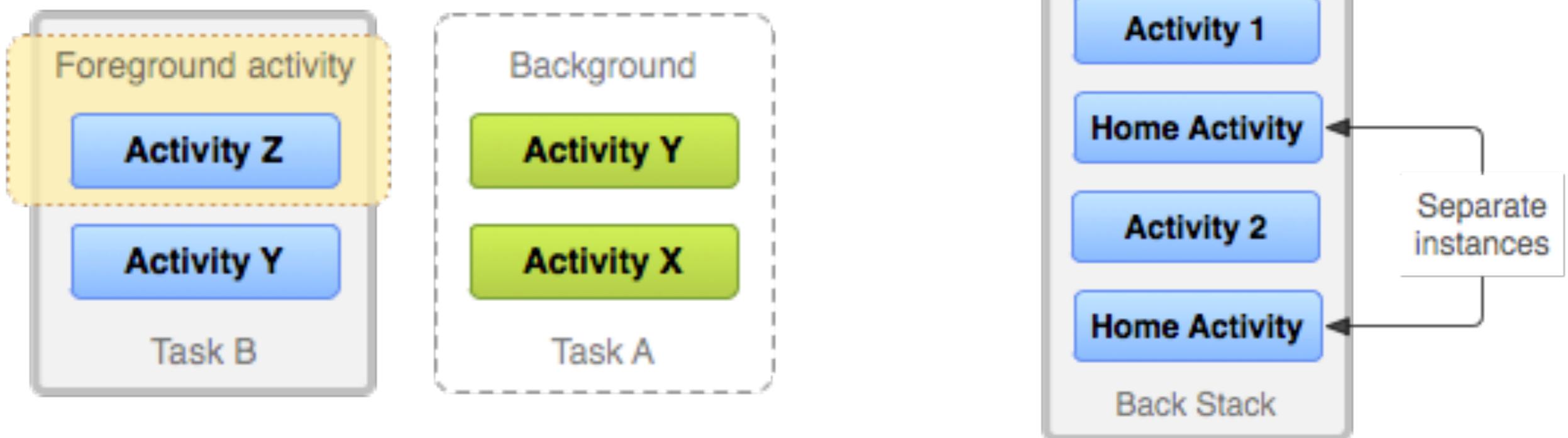
### HomeButton



# Tema 2

## Task & Back Stack

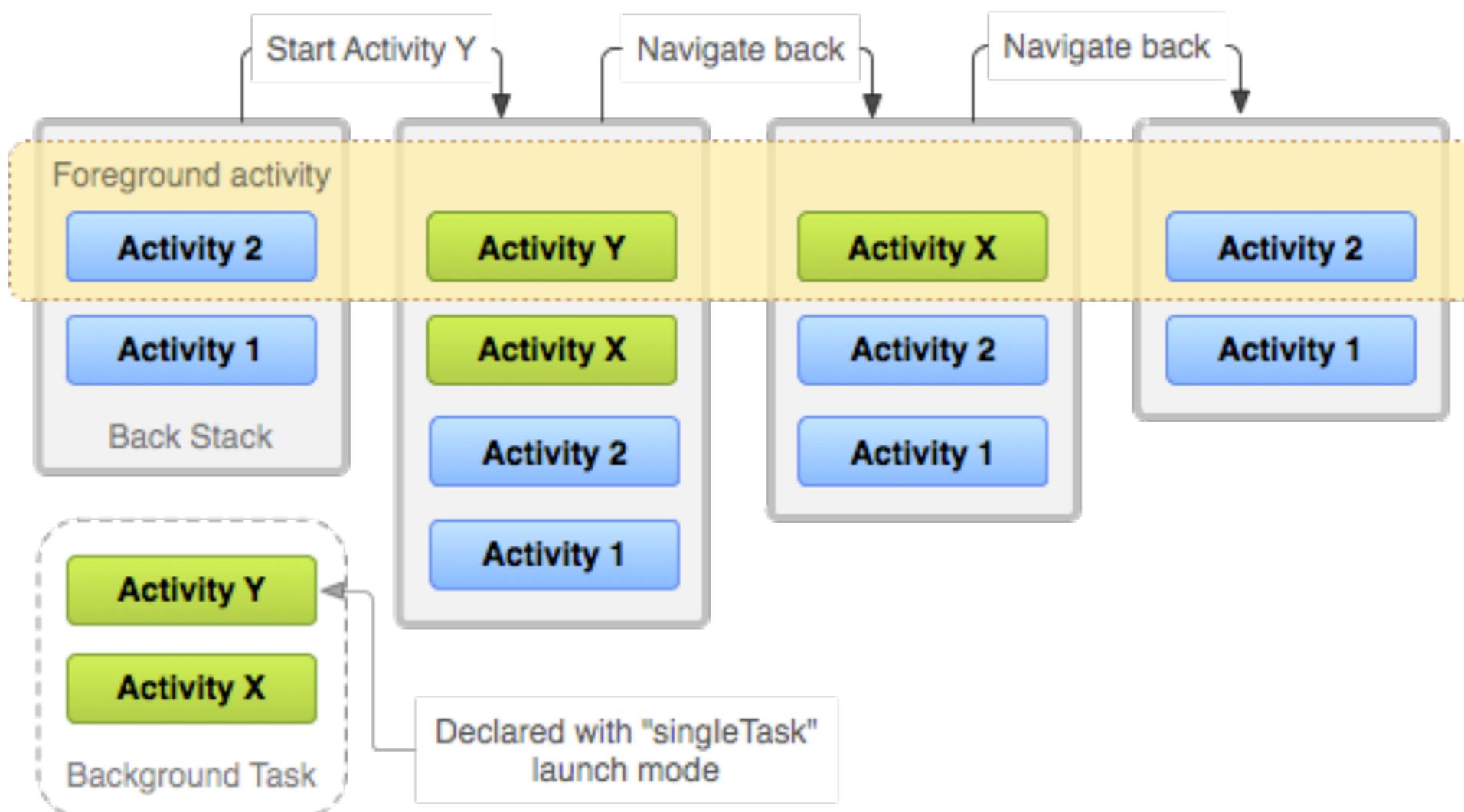
### MultiTask



# Tema 2

## Task & Back Stack

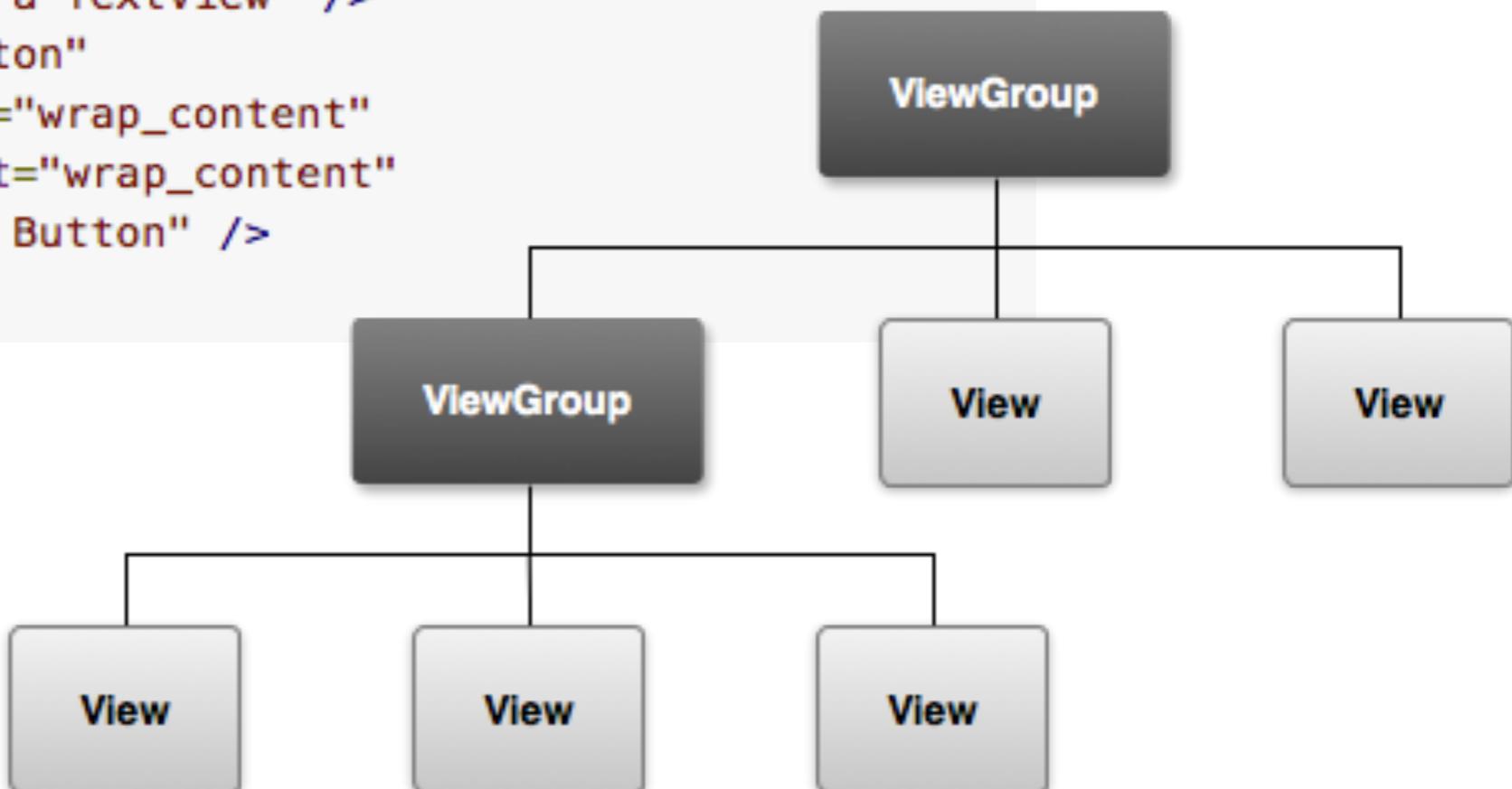
Excepción “launchMode=“SingleTask””



# Tema 2

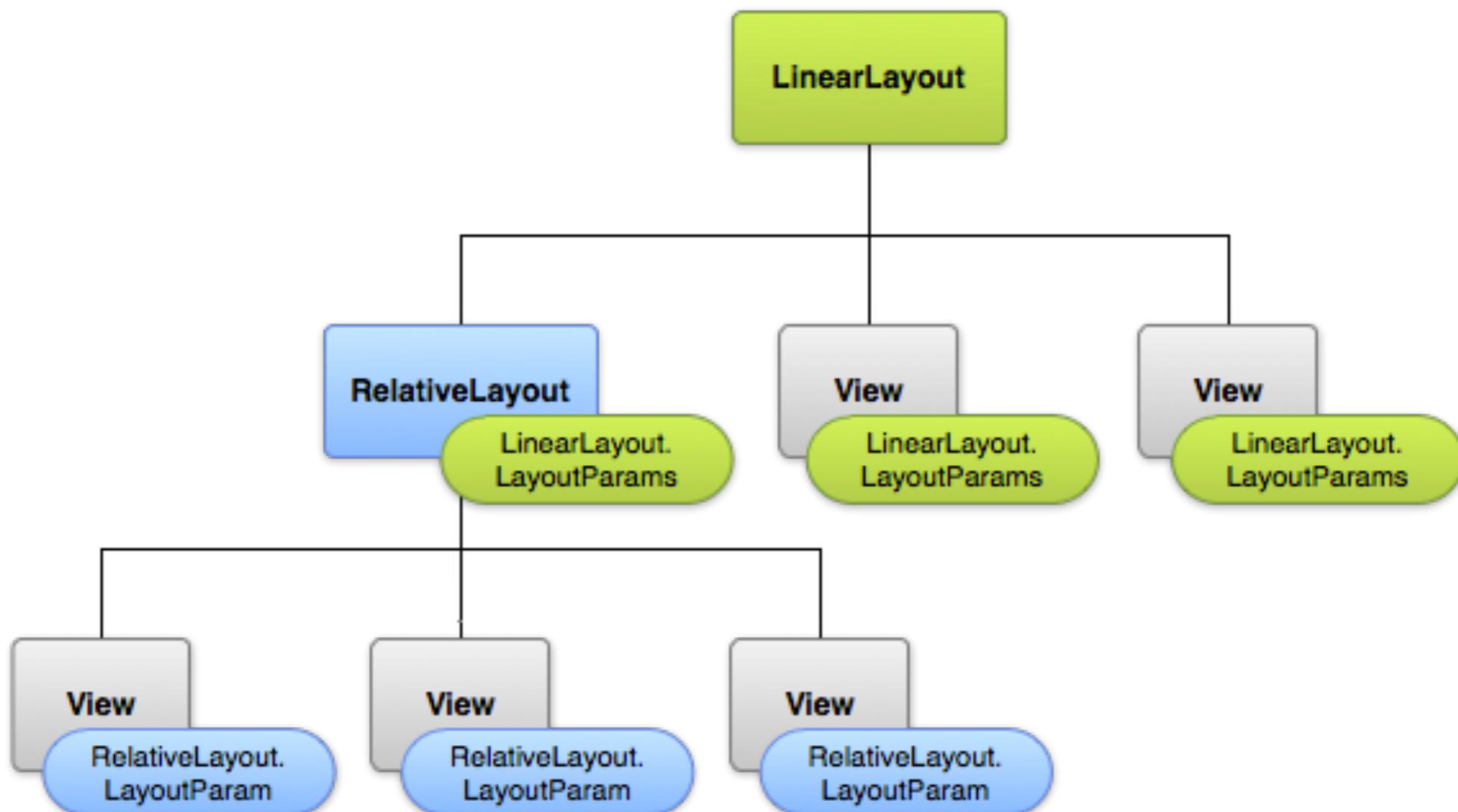
## View & ViewGroup

```
<?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>
```



# Tema 2

## Layouts



# Tema 2

## Layouts

### Linear Layout



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.

### Relative Layout



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).

### Web View

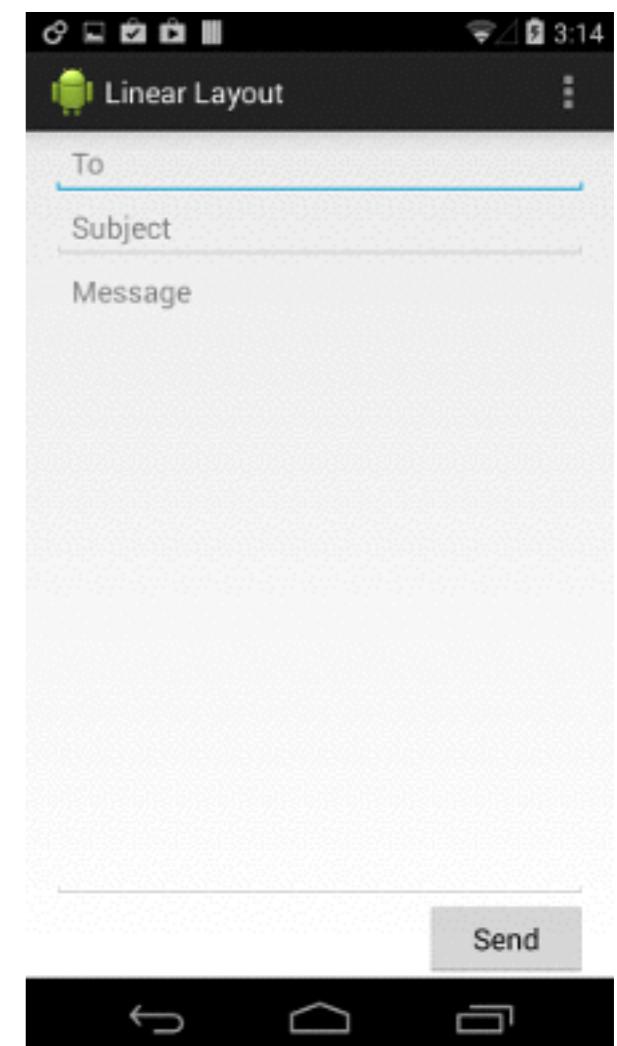


Displays web pages.

# Tema 2

## Linear Layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:orientation="vertical" >
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/to" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/subject" />
    <EditText
        android:layout_width="match_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>
```



# Tema 2

## Linear Layout

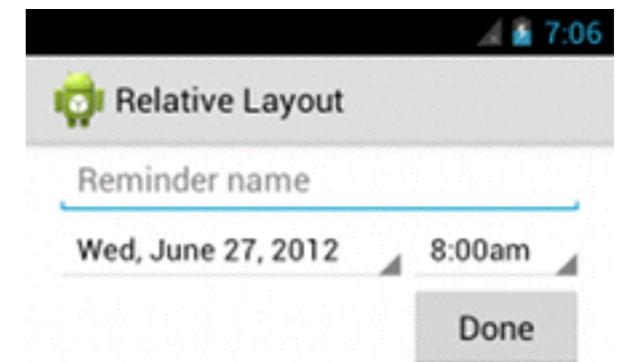
<code>android:baselineAligned</code>	When set to false, prevents the layout from aligning its children's baselines.
<code>android:baselineAlignedChildIndex</code>	When a linear layout is part of another layout that is baseline aligned, it can specify which of its children to baseline align to (that is, which child TextView).
<code>android:divider</code>	Drawable to use as a vertical divider between buttons.
<code>android:gravity</code>	Specifies how an object should position its content, on both the X and Y axes, within its own bounds.
<code>android:measureWithLargestChild</code>	When set to true, all children with a weight will be considered having the minimum size of the largest child.
<code>android:orientation</code>	Should the layout be a column or a row? Use "horizontal" for a row, "vertical" for a column.
<code>android:weightSum</code>	Defines the maximum weight sum.

<code>android:layout_gravity</code>	Standard gravity constant that a child supplies to its parent.
<code>android:layout_weight</code>	

# Tema 2

## Relative Layout

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp" >
    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/reminder" />
    <Spinner
        android:id="@+id/dates"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/name"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/times" />
    <Spinner
        android:id="@+id/times"
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/name"
        android:layout_alignParentRight="true" />
    <Button
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/times"
        android:layout_alignParentRight="true"
        android:text="@string/done" />
</RelativeLayout>
```



# Tema 2

## Relative Layout

<code>android:layout_above</code>	Positions the bottom edge of this view above the given anchor view ID.
<code>android:layout_alignBaseline</code>	Positions the baseline of this view on the baseline of the given anchor view ID.
<code>android:layout_alignBottom</code>	Makes the bottom edge of this view match the bottom edge of the given anchor view ID.
<code>android:layout_alignEnd</code>	Makes the end edge of this view match the end edge of the given anchor view ID.
<code>android:layout_alignLeft</code>	Makes the left edge of this view match the left edge of the given anchor view ID.
<code>android:layout_alignParentBottom</code>	If true, makes the bottom edge of this view match the bottom edge of the parent.
<code>android:layout_alignParentEnd</code>	If true, makes the end edge of this view match the end edge of the parent.
<code>android:layout_alignParentLeft</code>	If true, makes the left edge of this view match the left edge of the parent.
<code>android:layout_alignParentRight</code>	If true, makes the right edge of this view match the right edge of the parent.
<code>android:layout_alignParentStart</code>	If true, makes the start edge of this view match the start edge of the parent.
<code>android:layout_alignParentTop</code>	If true, makes the top edge of this view match the top edge of the parent.
<code>android:layout_alignRight</code>	Makes the right edge of this view match the right edge of the given anchor view ID.
<code>android:layout_alignStart</code>	Makes the start edge of this view match the start edge of the given anchor view ID.
<code>android:layout_alignTop</code>	Makes the top edge of this view match the top edge of the given anchor view ID.
<code>android:layout_alignWithParentIfMissing</code>	If set to true, the parent will be used as the anchor when the anchor cannot be found for layout_toLeftOf, layout_toRightOf, etc.

# Tema 2

## Relative Layout

<code>android:layout_below</code>	Positions the top edge of this view below the given anchor view ID.
<code>android:layout_centerHorizontal</code>	If true, centers this child horizontally within its parent.
<code>android:layout_centerInParent</code>	If true, centers this child horizontally and vertically within its parent.
<code>android:layout_centerVertical</code>	If true, centers this child vertically within its parent.
<code>android:layout_toEndOf</code>	Positions the start edge of this view to the end of the given anchor view ID.
<code>android:layout_toLeftOf</code>	Positions the right edge of this view to the left of the given anchor view ID.
<code>android:layout_toRightOf</code>	Positions the left edge of this view to the right of the given anchor view ID.
<code>android:layout_toStartOf</code>	Positions the end edge of this view to the start of the given anchor view ID.

# Tema 2

## Eventos

```
// Create an anonymous implementation of OnClickListener
private OnClickListener mCorkyListener = new OnClickListener() {
    public void onClick(View v) {
        // do something when the button is clicked
    }
};

protected void onCreate(Bundle savedInstanceState) {
    ...
    // Capture our button from layout
    Button button = (Button)findViewById(R.id.corky);
    // Register the onClick listener with the implementation above
    button.setOnClickListener(mCorkyListener);
    ...
}

// Create an anonymous implementation of OnClickListener
private OnClickListener mCorkyListener = new OnClickListener() {
    public void onClick(View v) {
        // do something when the button is clicked
    }
};

protected void onCreate(Bundle savedInstanceState) {
    ...
    // Capture our button from layout
    Button button = (Button)findViewById(R.id.corky);
    // Register the onClick listener with the implementation above
    button.setOnClickListener(mCorkyListener);
    ...
}
```

# Tema 2

## `onClick()`

From [View.OnClickListener](#). This is called when the user either touches the item (when in touch mode), or focuses upon the item with the navigation-keys or trackball and presses the suitable "enter" key or presses down on the trackball.

## `onLongClick()`

From [View.OnLongClickListener](#). This is called when the user either touches and holds the item (when in touch mode), or focuses upon the item with the navigation-keys or trackball and presses and holds the suitable "enter" key or presses and holds down on the trackball (for one second).

## `onFocusChange()`

From [View.OnFocusChangeListener](#). This is called when the user navigates onto or away from the item, using the navigation-keys or trackball.

## `onKey()`

From [View.OnKeyListener](#). This is called when the user is focused on the item and presses or releases a hardware key on the device.

## `onTouch()`

From [View.OnTouchListener](#). This is called when the user performs an action qualified as a touch event, including a press, a release, or any movement gesture on the screen (within the bounds of the item).

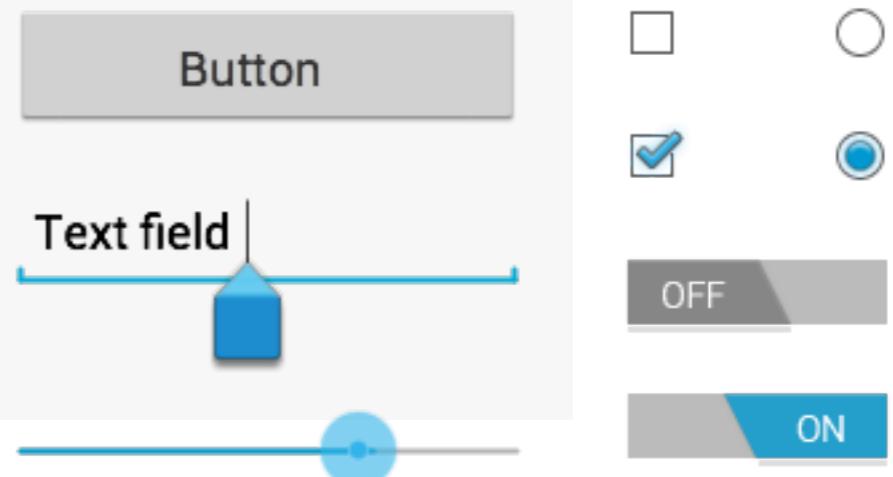
## `onCreateContextMenu()`

From [View.OnCreateContextMenuListener](#). This is called when a Context Menu is being built (as the result of a sustained "long click"). See the discussion on context menus in the [Menus developer guide](#).

# Tema 2

## Views

```
<?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="horizontal">
    <EditText android:id="@+id/edit_message"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="@string/edit_message" />
    <Button android:id="@+id/button_send"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_send"
        android:onClick="sendMessage" />
</LinearLayout>
```



# Tema 2

## Views

Control Type	Description	Related Classes
Button	A push-button that can be pressed, or clicked, by the user to perform an action.	<a href="#">Button</a>
Text field	An editable text field. You can use the <a href="#">AutoCompleteTextView</a> widget to create a text entry widget that provides auto-complete suggestions	<a href="#">EditText</a> , <a href="#">AutoCompleteTextView</a>
Checkbox	An on/off switch that can be toggled by the user. You should use checkboxes when presenting users with a group of selectable options that are not mutually exclusive.	<a href="#">CheckBox</a>
Radio button	Similar to checkboxes, except that only one option can be selected in the group.	<a href="#">RadioGroup</a> <a href="#">RadioButton</a>
Toggle button	An on/off button with a light indicator.	<a href="#">ToggleButton</a>
Spinner	A drop-down list that allows users to select one value from a set.	<a href="#">Spinner</a>
Pickers	A dialog for users to select a single value for a set by using up/down buttons or via a swipe gesture. Use a <a href="#">DatePicker</a> widget to enter the values for the date (month, day, year) or a <a href="#">TimePicker</a> widget to enter the values for a time (hour, minute, AM/PM), which will be formatted automatically for the user's locale.	<a href="#">DatePicker</a> , <a href="#">TimePicker</a>

# Tema 2

## Text View

```
<TextView  
    android:id="@+id/tvAuthorBookDetailFragment"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_below="@+id/ivCoverBookDetailFragment"  
    android:layout_margin="16dp"  
    android:gravity="center_horizontal"  
    android:text="Resumen del libro" />
```

### Fragment Basics

Article One

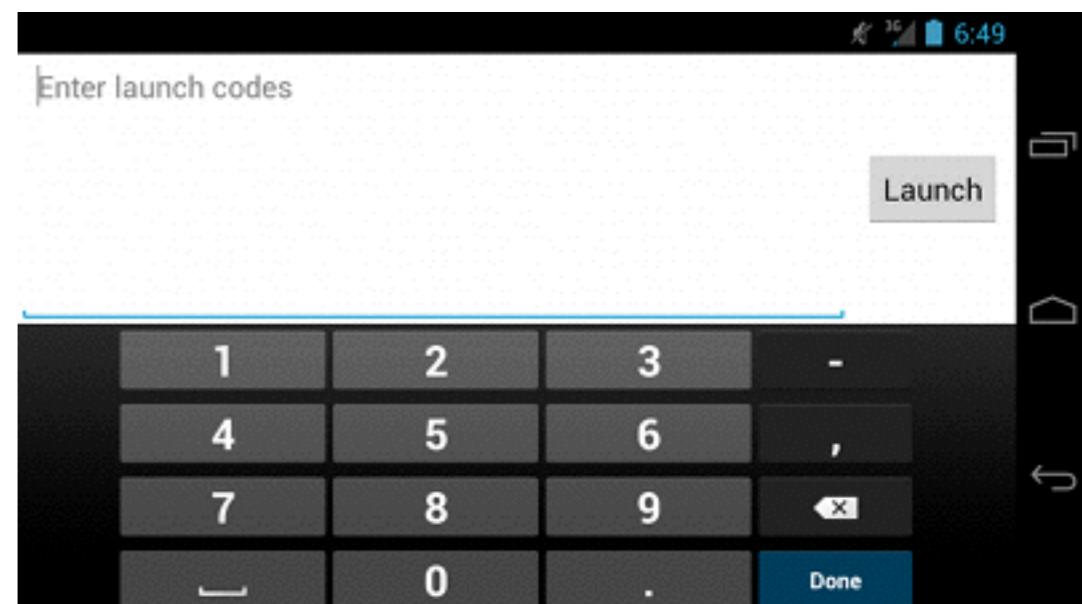
```
@Override  
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
    // Inflate the layout for this fragment  
    View v = inflater.inflate(R.layout.fragment_book_detail, container, false);  
  
    tvAuthor = (TextView)v.findViewById(R.id.tvAuthorBookDetailFragment);  
    tvDetails = (TextView)v.findViewById(R.id.tvDetailsBookDetailFragment);  
    updateBook();  
  
    return v;  
}
```

```
private void updateBook() {  
    if(mBook == null) {  
        return;  
    }  
  
    if(tvAuthor != null) {  
        tvAuthor.setText(mBook.getAuthor());  
    }  
  
    if(tvDetails != null) {  
        tvDetails.setText(mBook.getResume());  
    }  
}
```

# Tema 2

## Text Fields

```
<EditText  
    android:id="@+id/postal_address"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:hint="@string/postal_address_hint"  
    android:inputType="textPostalAddress |  
        textCapWords |  
        textNoSuggestions" />
```

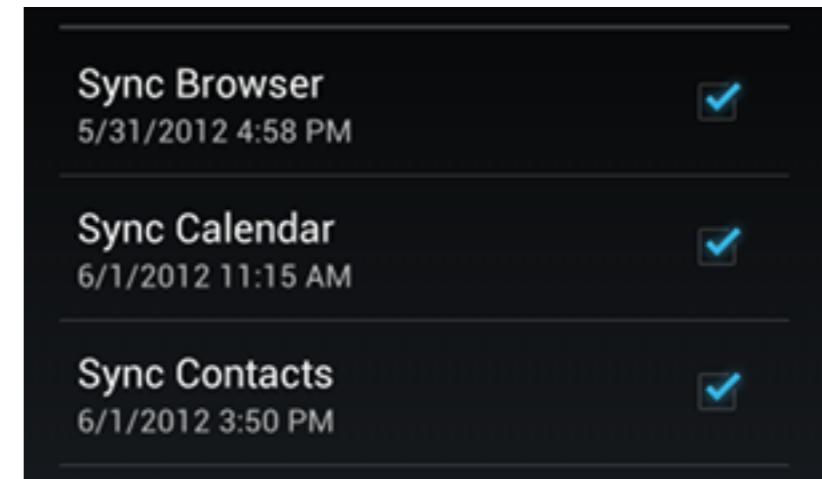


```
EditText editText = (EditText) findViewById(R.id.search);  
editText.setOnEditorActionListener(new OnEditorActionListener() {  
    @Override  
    public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {  
        boolean handled = false;  
        if (actionId == EditorInfo.IME_ACTION_SEND) {  
            sendMessage();  
            handled = true;  
        }  
        return handled;  
    }  
});
```

# Tema 2

## Check Box

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <CheckBox android:id="@+id/checkbox_meat"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/meat"
        android:onClick="onCheckboxClicked"/>
    <CheckBox android:id="@+id/checkbox_cheese"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/cheese"
        android:onClick="onCheckboxClicked"/>
</LinearLayout>
```



```
public void onCheckboxClicked(View view) {
    // Is the view now checked?
    boolean checked = ((CheckBox) view).isChecked();

    // Check which checkbox was clicked
    switch(view.getId()) {
        case R.id.checkbox_meat:
            if (checked)
                // Put some meat on the sandwich
            else
                // Remove the meat
            break;
        case R.id.checkbox_cheese:
            if (checked)
                // Cheese me
            else
                // I'm lactose intolerant
            break;
        // TODO: Veggie sandwich
    }
}
```

# Tema 2

## Radio Buttons

```
<?xml version="1.0" encoding="utf-8"?>
<RadioGroup xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <RadioButton android:id="@+id/radio_pirates"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/pirates"
        android:onClick="onRadioButtonClicked"/>
    <RadioButton android:id="@+id/radio_ninjas"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/ninjas"
        android:onClick="onRadioButtonClicked"/>
</RadioGroup>
```

ATTENDING?

Yes

Maybe

No

```
public void onRadioButtonClicked(View view) {
    // Is the button now checked?
    boolean checked = ((RadioButton) view).isChecked();

    // Check which radio button was clicked
    switch(view.getId()) {
        case R.id.radio_pirates:
            if (checked)
                // Pirates are the best
                break;
        case R.id.radio_ninjas:
            if (checked)
                // Ninjas rule
            break;
    }
}
```

# Tema 2

## Spinner

```
<Spinner  
    android:id="@+id/planets_spinner"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content" />
```

jay@gmail.com

Home

Home

Work

Other

Custom

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
    <string-array name="planets_array">  
        <item>Mercury</item>  
        <item>Venus</item>  
        <item>Earth</item>  
        <item>Mars</item>  
        <item>Jupiter</item>  
        <item>Saturn</item>  
        <item>Uranus</item>  
        <item>Neptune</item>  
    </string-array>  
</resources>
```

# Tema 2

## Spinner

jay@gmail.com

Home

Home

Work

Other

Custom

```
Spinner spinner = (Spinner) findViewById(R.id.spinner);
// Create an ArrayAdapter using the string array and a default spinner layout
ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
    R.array.planets_array, android.R.layout.simple_spinner_item);
// Specify the layout to use when the list of choices appears
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
// Apply the adapter to the spinner
spinner.setAdapter(adapter);
```

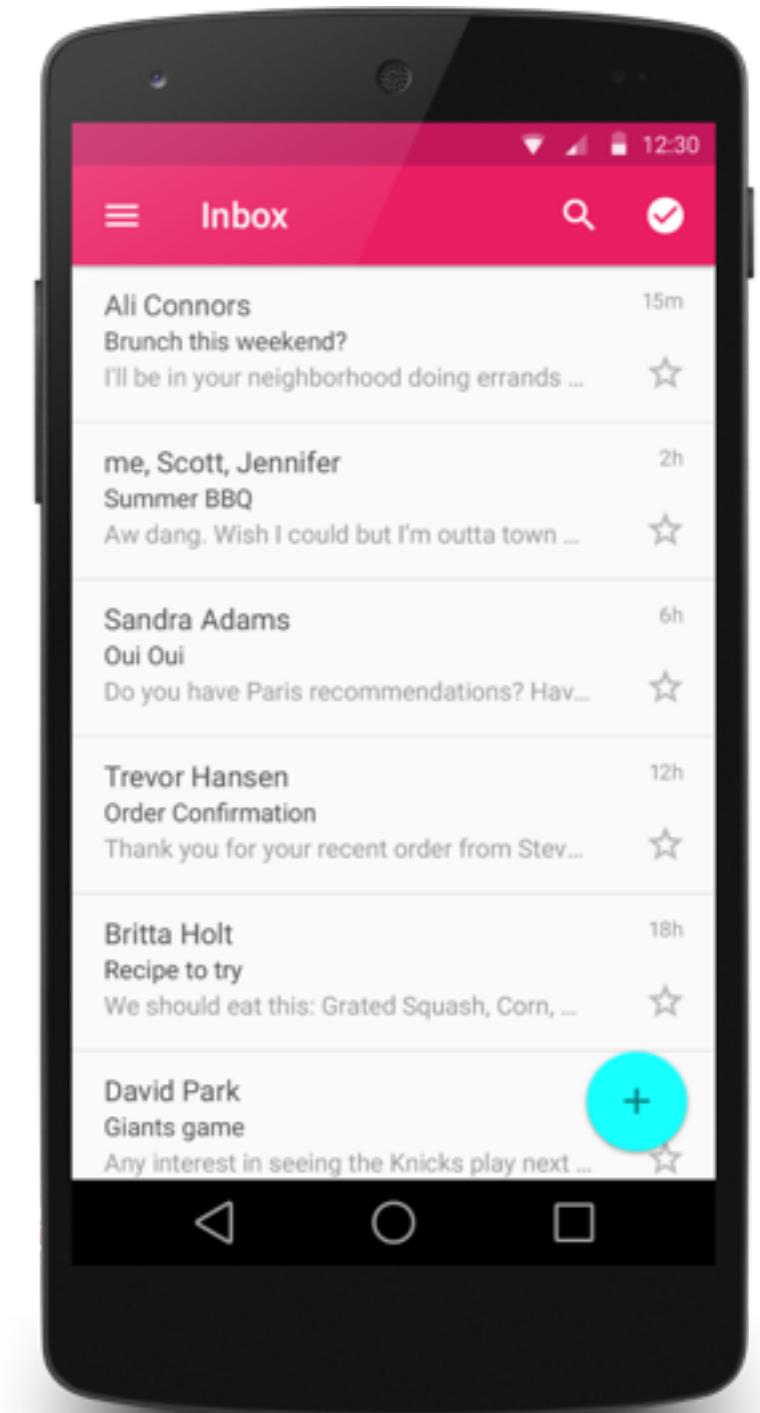
```
public class SpinnerActivity extends Activity implements OnItemSelectedListener {
    ...
    public void onItemSelected(AdapterView<?> parent, View view,
        int pos, long id) {
        // An item was selected. You can retrieve the selected item using
        // parent.getItemAtPosition(pos)
    }
    public void onNothingSelected(AdapterView<?> parent) {
        // Another interface callback
    }
}
```

```
Spinner spinner = (Spinner) findViewById(R.id.spinner);
spinner.setOnItemSelectedListener(this);
```

# Tema 2

## Listas (Recycler view)

```
<!-- A RecyclerView with some commonly used attributes -->
<android.support.v7.widget.RecyclerView
    android:id="@+id/my_recycler_view"
    android:scrollbars="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```



# Tema 2

## Listas (Recycler view)

```
public class MyActivity extends Activity {
    private RecyclerView mRecyclerView;
    private RecyclerView.Adapter mAdapter;
    private RecyclerView.LayoutManager mLayoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.my_activity);
        mRecyclerView = (RecyclerView) findViewById(R.id.my_recycler_view);

        // use this setting to improve performance if you know that changes
        // in content do not change the layout size of the RecyclerView
        mRecyclerView.setHasFixedSize(true);

        // use a linear layout manager
        mLayoutManager = new LinearLayoutManager(this);
        mRecyclerView.setLayoutManager(mLayoutManager);

        // specify an adapter (see also next example)
        mAdapter = new MyAdapter(myDataset);
        mRecyclerView.setAdapter(mAdapter);
    }
    ...
}
```

# Tema 2

## Listas (Recycler view)

```
public class MyAdapter extends RecyclerView.Adapter<MyAdapter.ViewHolder> {  
    private String[] mDataset;  
  
    // Provide a reference to the views for each data item  
    // Complex data items may need more than one view per item, and  
    // you provide access to all the views for a data item in a view holder  
    public static class ViewHolder extends RecyclerView.ViewHolder {  
        // each data item is just a string in this case  
        public TextView mTextView;  
        public ViewHolder(TextView v) {  
            super(v);  
            mTextView = v;  
        }  
    }  
  
    // Provide a suitable constructor (depends on the kind of dataset)  
    public MyAdapter(String[] myDataset) {  
        mDataset = myDataset;  
    }  
}
```

# Tema 2

## Listas (Recycler view)

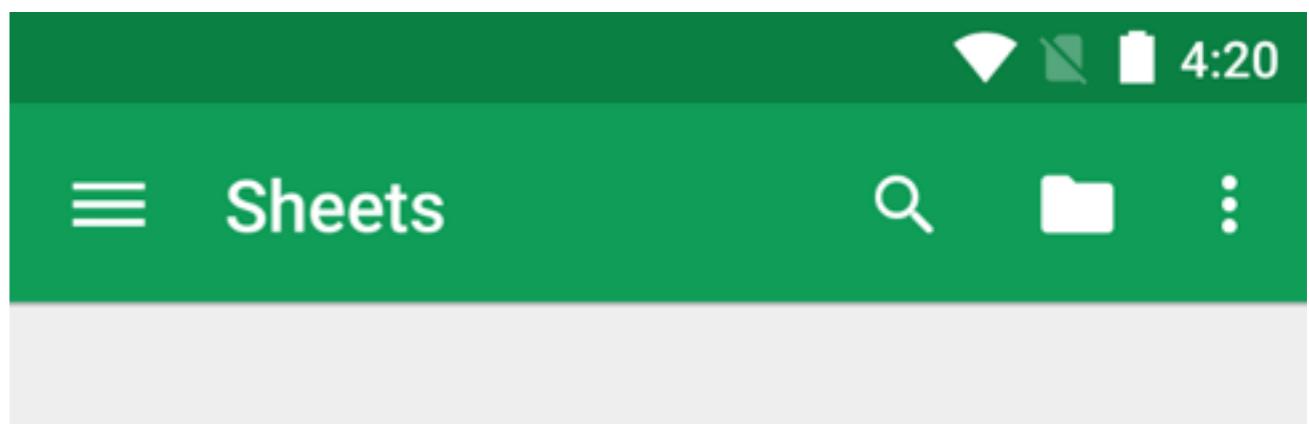
```
// Create new views (invoked by the layout manager)
@Override
public MyAdapter.ViewHolder onCreateViewHolder(ViewGroup parent,
                                              int viewType) {
    // create a new view
    View v = LayoutInflater.from(parent.getContext())
        .inflate(R.layout.my_text_view, parent, false);
    // set the view's size, margins, paddings and layout parameters
    ...
    ViewHolder vh = new ViewHolder(v);
    return vh;
}

// Replace the contents of a view (invoked by the layout manager)
@Override
public void onBindViewHolder(ViewHolder holder, int position) {
    // - get element from your dataset at this position
    // - replace the contents of the view with that element
    holder.mTextView.setText(mDataset[position]);
}

// Return the size of your dataset (invoked by the layout manager)
@Override
public int getItemCount() {
    return mDataset.length;
}
```

# Tema 2

## Application Bar



# Tema 2

Application Bar

Añadir Toolbar

```
public class MyActivity extends AppCompatActivity {  
    // ...  
}
```

```
<application  
    android:theme="@style/Theme.AppCompat.Light.NoActionBar"  
/>
```

```
<android.support.v7.widget.Toolbar  
    android:id="@+id/my_toolbar"  
    android:layout_width="match_parent"  
    android:layout_height="?attr/actionBarSize"  
    android:background="?attr/colorPrimary"  
    android:elevation="4dp"  
    android:theme="@style/ThemeOverlay.AppCompat.ActionBar"  
    app:popupTheme="@style/ThemeOverlay.AppCompat.Light"/>
```

# Tema 2

## Application Bar

### Añadir Toolbar

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_my);  
    Toolbar myToolbar = (Toolbar) findViewById(R.id.my_toolbar);  
    setSupportActionBar(myToolbar);  
}
```

#### getSupportActionBar

`ActionBar getSupportActionBar ()`

Support library version of `getActionBar()`.

Retrieve a reference to this activity's ActionBar.

##### Returns

`ActionBar`

The Activity's ActionBar, or null if it does not have one.

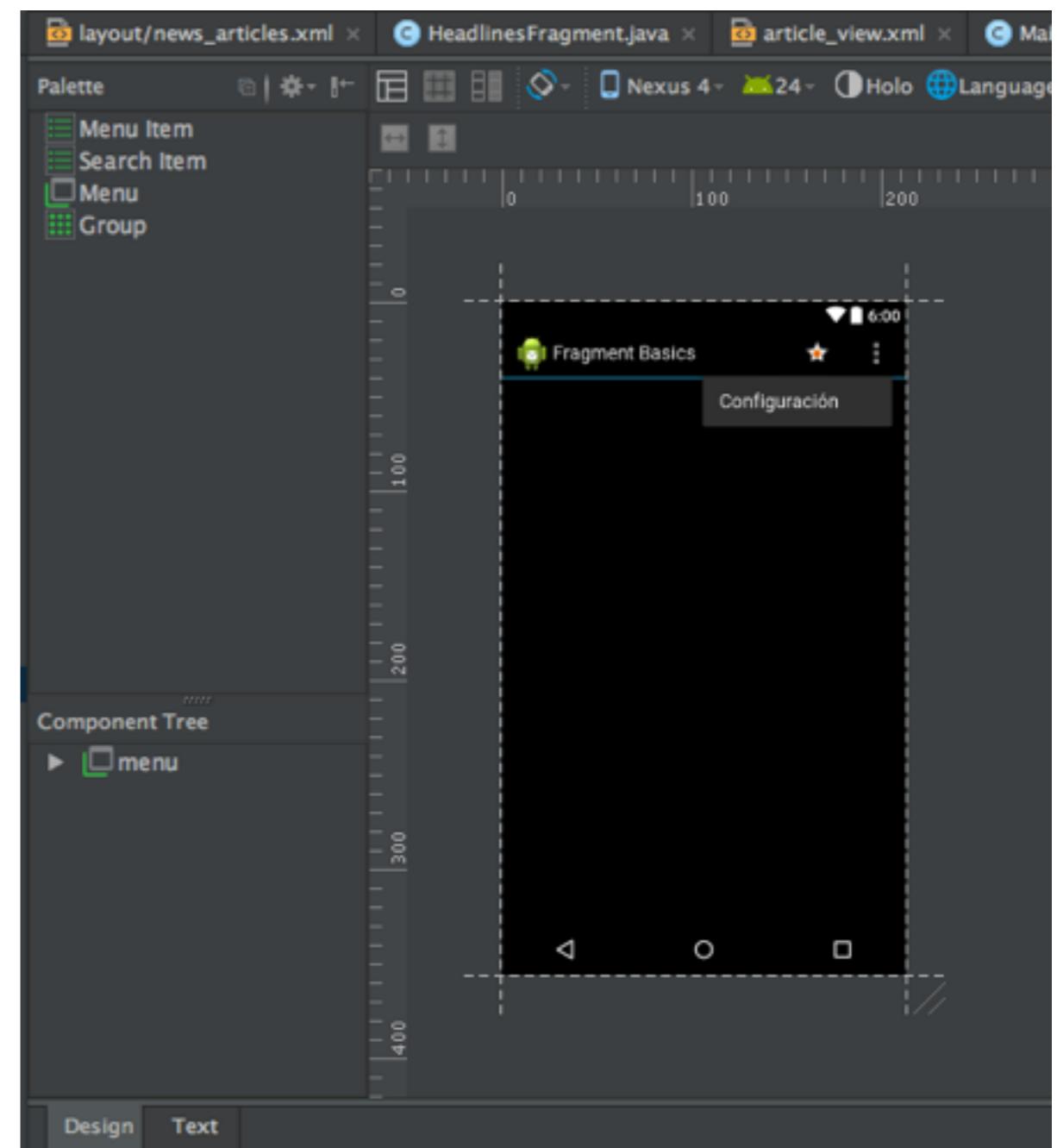
# Tema 2

## Application Bar

### Añadir acciones

The screenshot shows the Android Studio interface with the XML code for a menu resource. The code defines a menu with two items: one for marking an item as favorite and another for settings.

```
<?xml version="1.0" encoding="utf-8"?>
<menu>
    <item
        android:id="@+id/action_favorite"
        android:icon="@android:drawable/star_on"
        android:title="@string/action_favorite"
        app:showAsAction="ifRoom"/>
    <!-- Settings, should always be in the overflow -->
    <item
        android:id="@+id/action_settings"
        android:title="@string/action_settings"
        app:showAsAction="never"/>
</menu>
```



# Tema 2

## Application Bar

### Responder a Acciones

```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    super.onCreateOptionsMenu(menu);  
  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.main_activity, menu);  
    return true;  
}  
  
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.action_favorite:  
            Toast.makeText(this, getString(R.string.action_favorite), Toast.LENGTH_SHORT).show();  
            return true;  
        case R.id.action_settings:  
            Toast.makeText(this, getString(R.string.action_settings), Toast.LENGTH_SHORT).show();  
            return true;  
        default:  
            return onOptionsItemSelected(item);  
    }  
}
```

# Tema 2

## Application Bar

### Acción Up

```
<application ... >
    ...
    <!-- The main/home activity (it has no parent activity) -->
    <activity
        android:name="com.example.myfirstapp.MainActivity" ...>
        ...
    </activity>

    <!-- A child of the main activity -->
    <activity
        android:name="com.example.myfirstapp.MyChildActivity"
        android:label="@string/title_activity_child"
        android:parentActivityName="com.example.myfirstapp.MainActivity" >

        <!-- Parent activity meta-data to support 4.0 and lower -->
        <meta-data
            android:name="android.support.PARENT_ACTIVITY"
            android:value="com.example.myfirstapp.MainActivity" />
    </activity>
</application>
```

# Tema 2

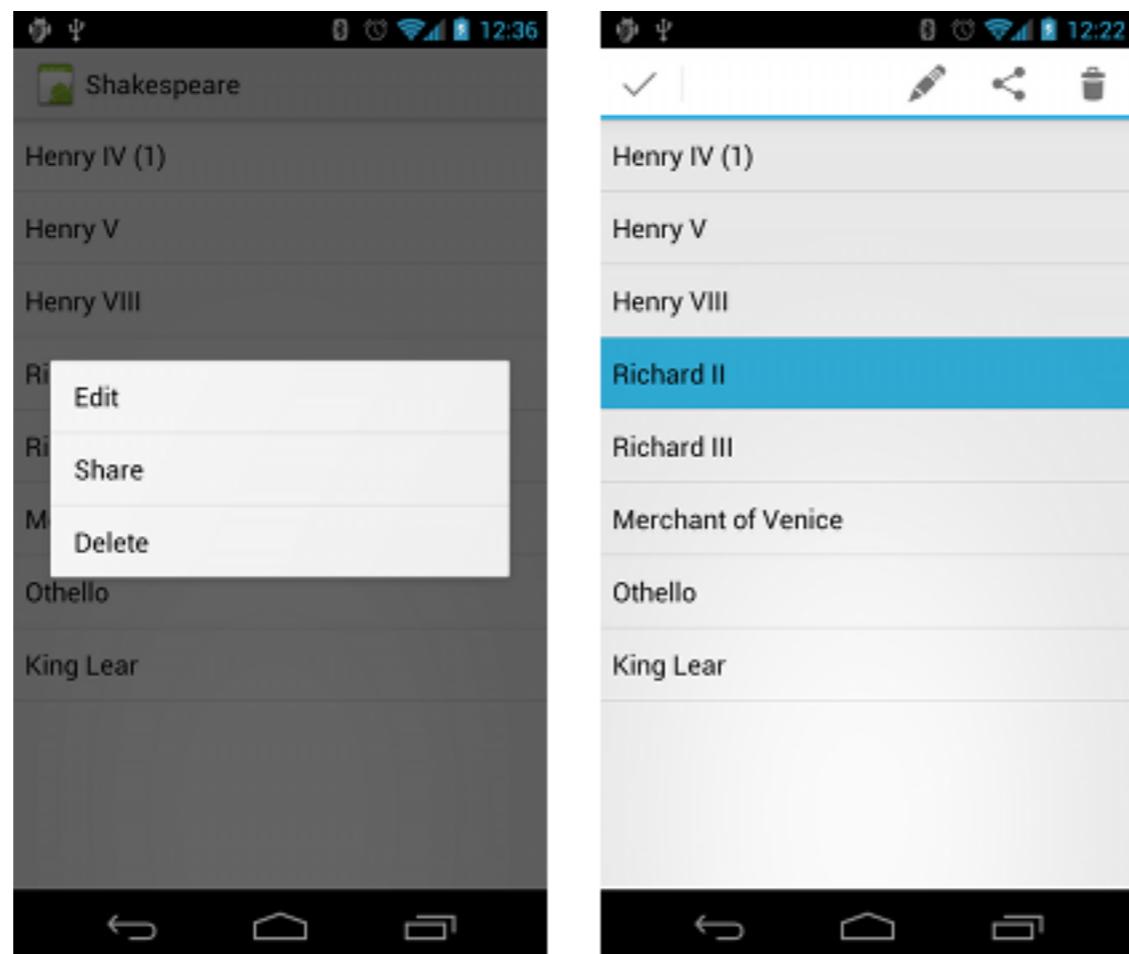
## Application Bar

### Acción Up

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_my_child);  
  
    // my_child_toolbar is defined in the layout file  
    Toolbar myChildToolbar =  
        (Toolbar) findViewById(R.id.my_child_toolbar);  
    setSupportActionBar(myChildToolbar);  
  
    // Get a support ActionBar corresponding to this toolbar  
    ActionBar ab = getSupportActionBar();  
  
    // Enable the Up button  
    ab.setDisplayHomeAsUpEnabled(true);  
}
```

# Tema 2

## Menu Contextual (long-click)



# Tema 2

## Menu Contextual Flotante

### registerForContextMenu

Added in API level 1

```
void registerForContextMenu (View view)
```

Registers a context menu to be shown for the given view (multiple views can show the context menu). This method will set the [View.OnCreateContextMenuListener](#) on the view to this activity, so [onCreateContextMenu\(ContextMenu, View, ContextMenuInfo\)](#) will be called when it is time to show the context menu.

#### Parameters

<b>view</b>	<a href="#">View</a> : The view that should show a context menu.
-------------	------------------------------------------------------------------

# Tema 2

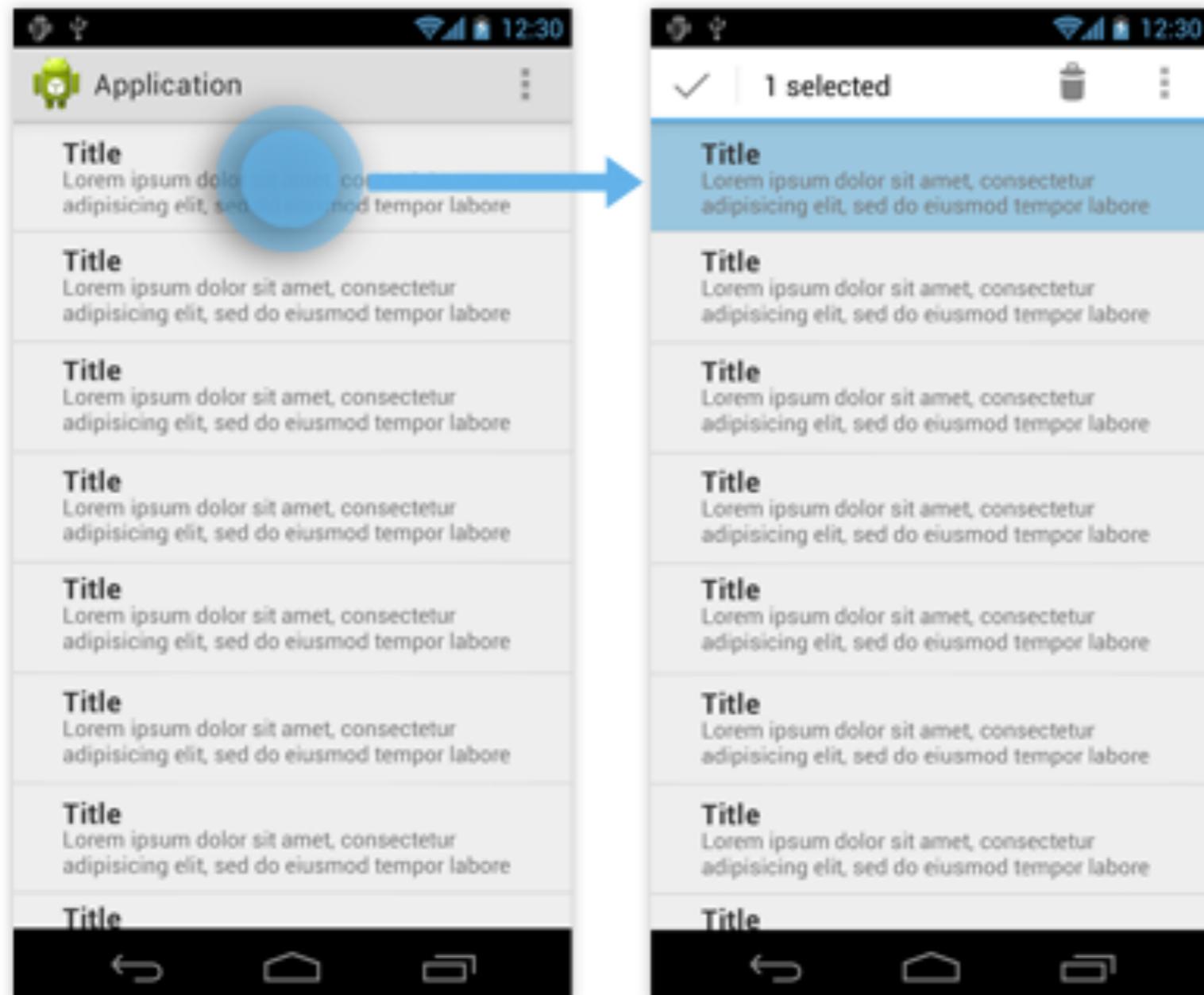
## Menu Contextual Flotante

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
                               ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.context_menu, menu);
}
```

```
@Override
public boolean onContextItemSelected(MenuItem item) {
    AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();
    switch (item.getItemId()) {
        case R.id.edit:
            editNote(info.id);
            return true;
        case R.id.delete:
            deleteNote(info.id);
            return true;
        default:
            return super.onContextItemSelected(item);
    }
}
```

# Tema 2

# Menu Contextual Action Mode



# Tema 2

## Menu Contextual Action Mode

```
private ActionMode.Callback mActionModeCallback = new ActionMode.Callback() {

    // Called when the action mode is created; startActionMode() was called
    @Override
    public boolean onCreateActionMode(ActionMode mode, Menu menu) {
        // Inflate a menu resource providing context menu items
        MenuInflater inflater = mode.getMenuInflater();
        inflater.inflate(R.menu.context_menu, menu);
        return true;
    }

    // Called each time the action mode is shown. Always called after onCreateActionMode, but
    // may be called multiple times if the mode is invalidated.
    @Override
    public boolean onPrepareActionMode(ActionMode mode, Menu menu) {
        return false; // Return false if nothing is done
    }

    // Called when the user selects a contextual menu item
    @Override
    public boolean onActionItemClicked(ActionMode mode, MenuItem item) {
        switch (item.getItemId()) {
            case R.id.menu_share:
                shareCurrentItem();
                mode.finish(); // Action picked, so close the CAB
                return true;
            default:
                return false;
        }
    }

    // Called when the user exits the action mode
    @Override
    public void onDestroyActionMode(ActionMode mode) {
        mActionMode = null;
    }
};
```

# Tema 2

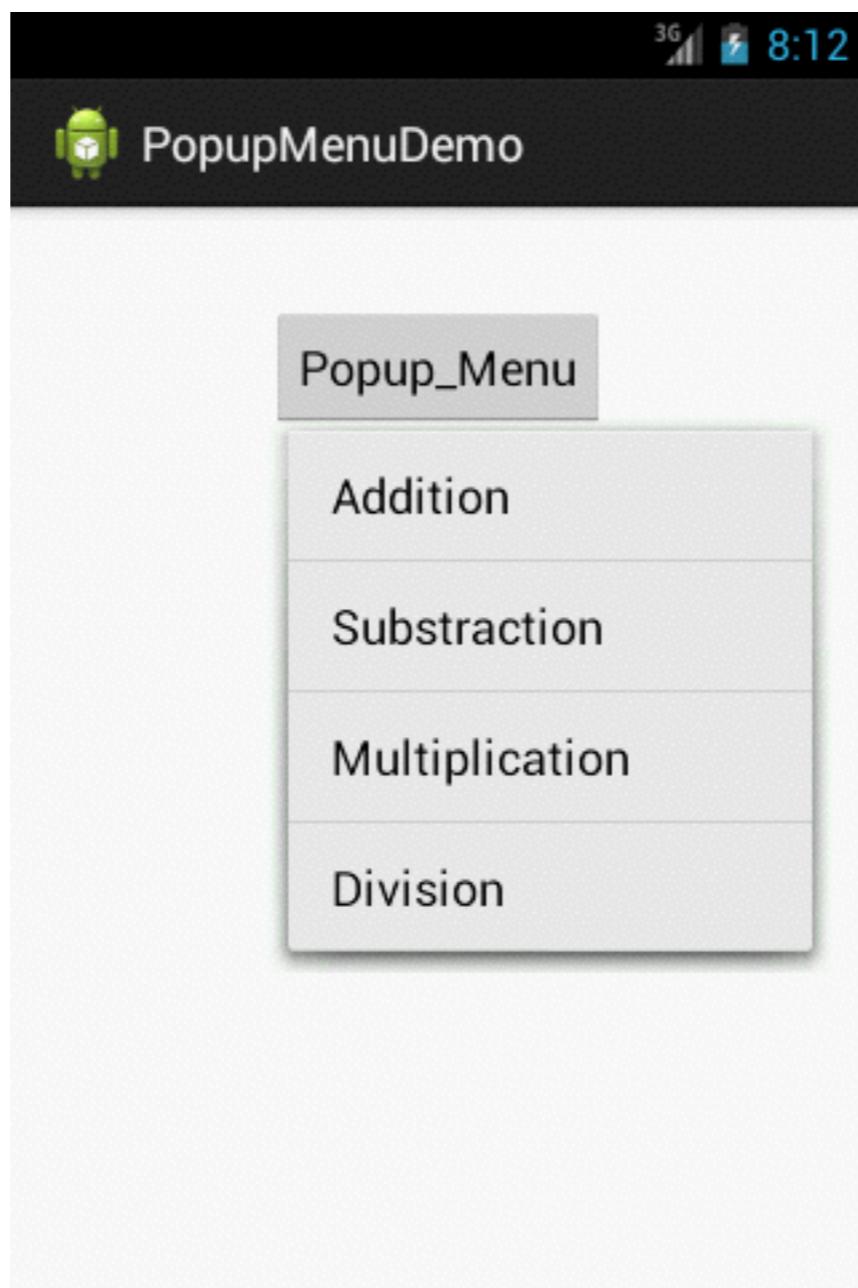
## Menu Contextual Action Mode

```
someView.setOnLongClickListener(new View.OnLongClickListener() {
    // Called when the user long-clicks on someView
    public boolean onLongClick(View view) {
        if (mActionMode != null) {
            return false;
        }

        // Start the CAB using the ActionMode.Callback defined above
        mActionMode = getActivity().startActionMode(mActionModeCallback);
        view.setSelected(true);
        return true;
    }
});
```

# Tema 2

## Popup Menu



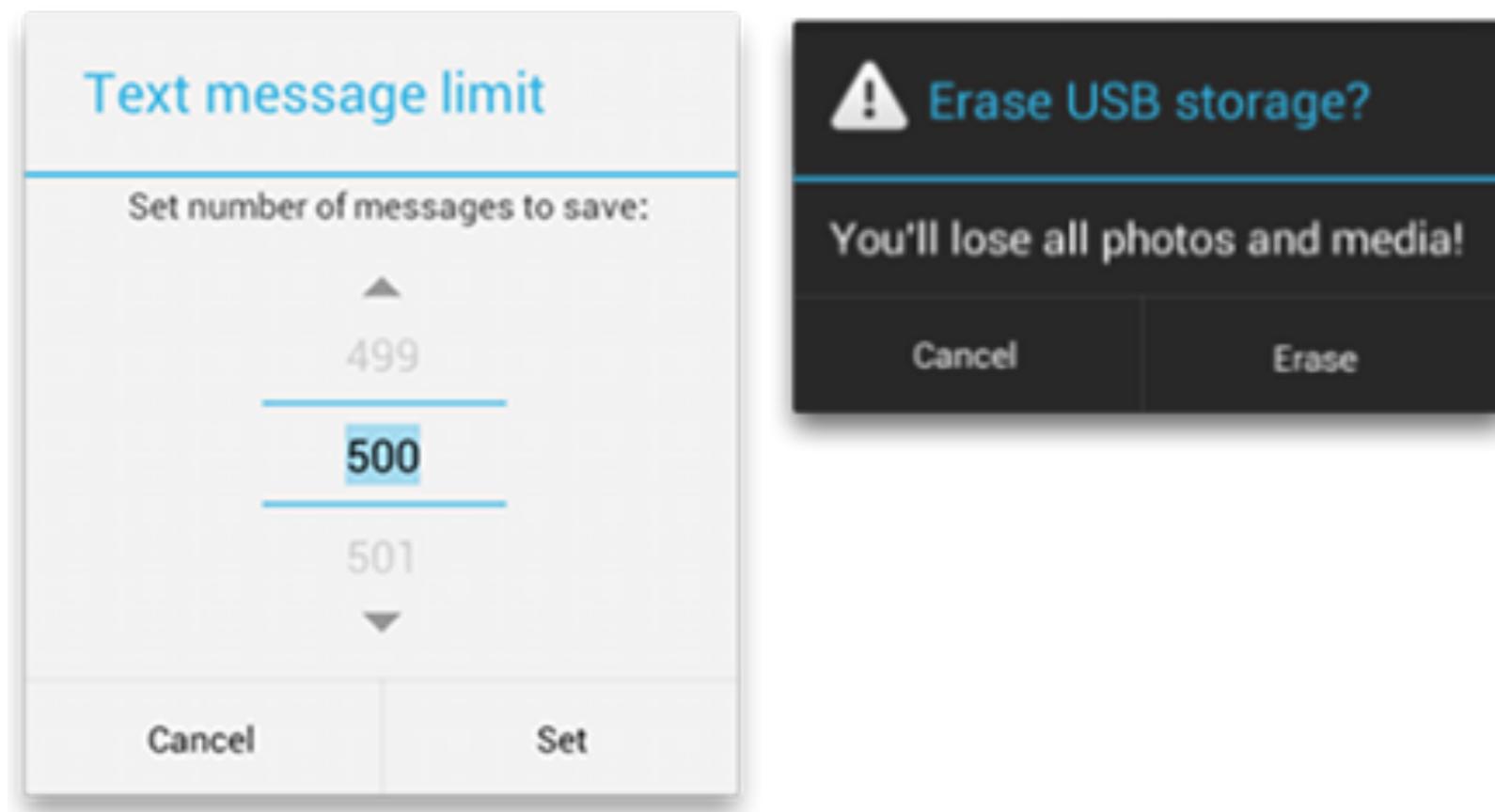
# Tema 2

## Popup Menu

```
public void showMenu(View v) {  
    PopupMenu popup = new PopupMenu(this, v);  
  
    // This activity implements OnMenuItemClickListener  
    popup.setOnMenuItemClickListener(this);  
    popup.inflate(R.menu.actions);  
    popup.show();  
}  
  
@Override  
public boolean onMenuItemClick(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.archive:  
            archive(item);  
            return true;  
        case R.id.delete:  
            delete(item);  
            return true;  
        default:  
            return false;  
    }  
}
```

# Tema 2

## Dialogos



# Tema 2

## Dialogos

Fire missles?

Cancel

Fire

```
public class FireMissilesDialogFragment extends DialogFragment {  
    @Override  
    public Dialog onCreateDialog(Bundle savedInstanceState) {  
        // Use the Builder class for convenient dialog construction  
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());  
        builder.setMessage(R.string.dialog_fire_missiles)  
            .setPositiveButton(R.string.fire, new DialogInterface.OnClickListener() {  
                public void onClick(DialogInterface dialog, int id) {  
                    // FIRE ZE MISSILES!  
                }  
            })  
            .setNegativeButton(R.string.cancel, new DialogInterface.OnClickListener() {  
                public void onClick(DialogInterface dialog, int id) {  
                    // User cancelled the dialog  
                }  
            });  
        // Create the AlertDialog object and return it  
        return builder.create();  
    }  
}
```

```
public void confirmFireMissiles() {  
    DialogFragment newFragment = new FireMissilesDialogFragment();  
    newFragment.show(getSupportFragmentManager(), "missiles");  
}
```

# Tema 2

## Dialogos

Fire missles?

Cancel

Fire

### show

```
void show (FragmentManager manager,  
          String tag)
```

Display the dialog, adding the fragment to the given FragmentManager. This is a convenience for explicitly creating a transaction, adding the fragment to it with the given tag, and committing it. This does *not* add the transaction to the back stack. When the fragment is dismissed, a new transaction will be executed to remove it from the activity.

Parameters	
manager	FragmentManager: The FragmentManager this fragment will be added to.
tag	String: The tag for this fragment, as per FragmentTransaction.add.

### dismiss

```
void dismiss ()
```

Dismiss the fragment and its dialog. If the fragment was added to the back stack, all back stack state up to and including this entry will be popped. Otherwise, a new transaction will be committed to remove the fragment.

### dismissAllowingStateLoss

```
void dismissAllowingStateLoss ()
```

Version of `dismiss()` that uses `FragmentTransaction.commitAllowingStateLoss()`. See linked documentation for further details.

# Tema 2

## Alert Dialog

### 1. Title

This is optional and should be used only when the content area is occupied by a detailed message, a list, or custom layout. If you need to state a simple message or question (such as the dialog in figure 1), you don't need a title.

### 2. Content area

This can display a message, a list, or other custom layout.

### 3. Action buttons

There should be no more than three action buttons in a dialog.

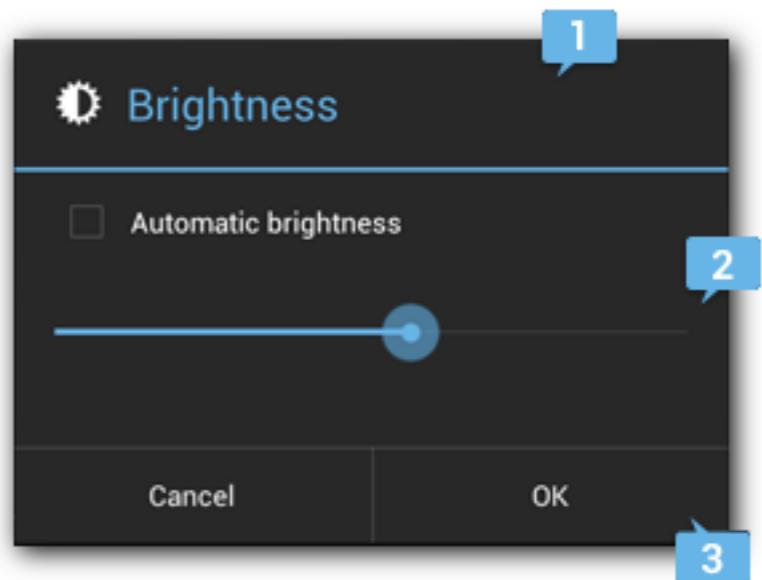


Figure 2. The layout of a dialog.

```
// 1. Instantiate an AlertDialog.Builder with its constructor
AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());

// 2. Chain together various setter methods to set the dialog characteristics
builder.setMessage(R.string.dialog_message)
    .setTitle(R.string.dialog_title);

// 3. Get the AlertDialog from create()
AlertDialog dialog = builder.create();
```

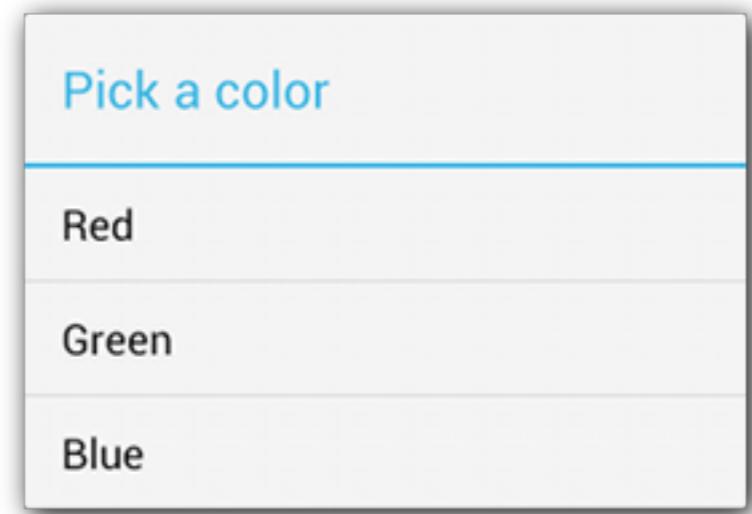
# Tema 2

## Alert Dialog

```
AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
// Add the buttons
builder.setPositiveButton(R.string.ok, new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int id) {
        // User clicked OK button
    }
});
builder.setNegativeButton(R.string.cancel, new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int id) {
        // User cancelled the dialog
    }
});
// Set other dialog properties
...
// Create the AlertDialog
AlertDialog dialog = builder.create();
```

# Tema 2

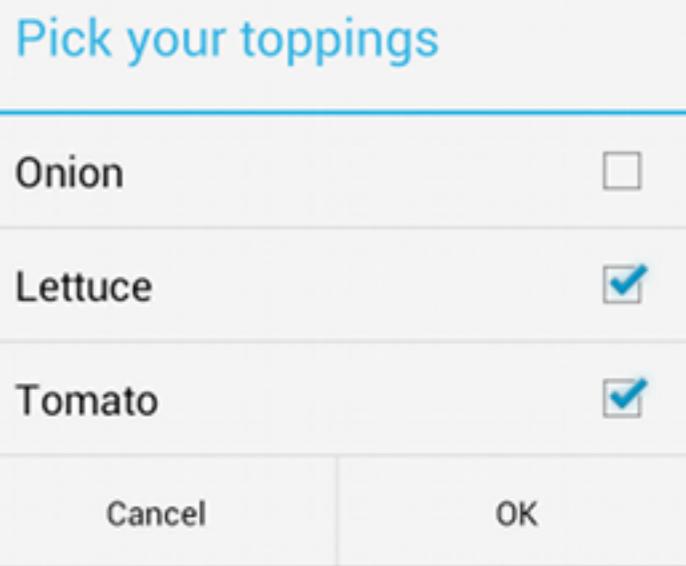
## Alert Dialog



```
@Override  
public Dialog onCreateDialog(Bundle savedInstanceState) {  
    AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());  
    builder.setTitle(R.string.pick_color)  
        .setItems(R.array.colors_array, new DialogInterface.OnClickListener() {  
            public void onClick(DialogInterface dialog, int which) {  
                // The 'which' argument contains the index position  
                // of the selected item  
            }  
        });  
    return builder.create();  
}
```

```
@Override
public Dialog onCreateDialog(Bundle savedInstanceState) {
    mSelectedItems = new ArrayList(); // Where we track the selected items
    AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
    // Set the dialog title
    builder.setTitle(R.string.pick_toppings)
    // Specify the list array, the items to be selected by default (null for none),
    // and the listener through which to receive callbacks when items are selected
    .setMultiChoiceItems(R.array.toppings, null,
        new DialogInterface.OnMultiChoiceClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which,
                boolean isChecked) {
                if (isChecked) {
                    // If the user checked the item, add it to the selected items
                    mSelectedItems.add(which);
                } else if (mSelectedItems.contains(which)) {
                    // Else, if the item is already in the array, remove it
                    mSelectedItems.remove(Integer.valueOf(which));
                }
            }
        })
    // Set the action buttons
    .setPositiveButton(R.string.ok, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int id) {
            // User clicked OK, so save the mSelectedItems results somewhere
            // or return them to the component that opened the dialog
            ...
        }
    })
    .setNegativeButton(R.string.cancel, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int id) {
            ...
        }
    });
}

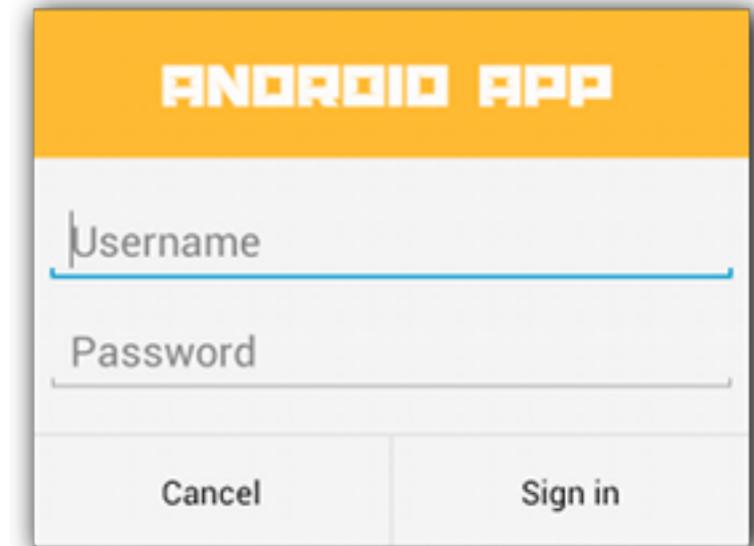
return builder.create();
}
```



# Tema 2

## Custom Dialog

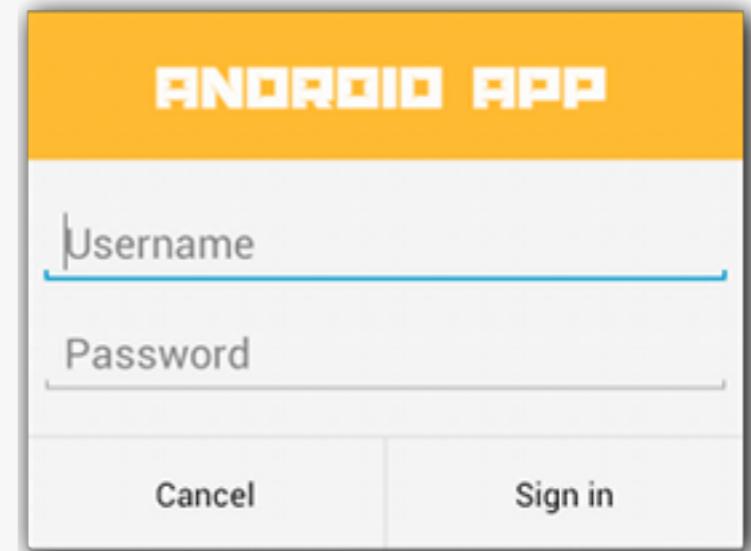
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <ImageView
        android:src="@drawable/header_logo"
        android:layout_width="match_parent"
        android:layout_height="64dp"
        android:scaleType="center"
        android:background="#FFFFBB33"
        android:contentDescription="@string/app_name" />
    <EditText
        android:id="@+id/username"
        android:inputType="textEmailAddress"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:layout_marginLeft="4dp"
        android:layout_marginRight="4dp"
        android:layout_marginBottom="4dp"
        android:hint="@string/username" />
    <EditText
        android:id="@+id/password"
        android:inputType="textPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="4dp"
        android:layout_marginLeft="4dp"
        android:layout_marginRight="4dp"
        android:layout_marginBottom="16dp"
        android:fontFamily="sans-serif"
        android:hint="@string/password"/>
</LinearLayout>
```



# Tema 2

## Custom Dialog

```
@Override  
public Dialog onCreateDialog(Bundle savedInstanceState) {  
    AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());  
    // Get the layout inflater  
    LayoutInflator inflater = getActivity().getLayoutInflater();  
  
    // Inflate and set the layout for the dialog  
    // Pass null as the parent view because its going in the dialog layout  
    builder.setView(inflater.inflate(R.layout.dialog_signin, null))  
    // Add action buttons  
    .setPositiveButton(R.string.signin, new DialogInterface.OnClickListener() {  
        @Override  
        public void onClick(DialogInterface dialog, int id) {  
            // sign in the user ...  
        }  
    })  
    .setNegativeButton(R.string.cancel, new DialogInterface.OnClickListener() {  
        public void onClick(DialogInterface dialog, int id) {  
            LoginDialogFragment.this.getDialog().cancel();  
        }  
    });  
    return builder.create();  
}
```



# Tema 2

## Ejercicios

Crear una biblioteca personal

- A. Pantalla de Login/Register.(No funcional)
- B. Pantalla de listado de libros
- C. Pantalla de detalle de cada libro
- D. Poder marcar un libro como favorito
- E. Añadir y eliminar libros (Sin persistencia)