

Tema 3

Recursos

1. Recursos
2. L10n
3. Color
4. Drawable
5. Layouts
6. Estilos
7. Vistas personalizadas
8. Ejercicios

Tema 3

Recursos



Tema 3

Recursos

```
MyProject/  
  src/  
    MainActivity.java  
  res/  
    drawable/  
      graphic.png  
    layout/  
      main.xml  
      info.xml  
    mipmap/  
      icon.png  
    values/  
      strings.xml
```

<code>animator/</code>	XML files that define property animations .
<code>anim/</code>	XML files that define tween animations . (Property animations can also be saved in this directory, but the <code>animator/</code> directory is preferred for property animations to distinguish between the two types.)
<code>color/</code>	XML files that define a state list of colors. See Color State List Resource
<code>drawable/</code>	<p>Bitmap files (<code>.png</code>, <code>.9.png</code>, <code>.jpg</code>, <code>.gif</code>) or XML files that are compiled into the following drawable resource subtypes:</p> <ul style="list-style-type: none">• Bitmap files• Nine-Patches (re-sizable bitmaps)• State lists• Shapes• Animation drawables• Other drawables <p>See Drawable Resources.</p>

Tema 3

Recursos

```
MyProject/  
  src/  
    MainActivity.java  
  res/  
    drawable/  
      graphic.png  
    layout/  
      main.xml  
      info.xml  
    mipmap/  
      icon.png  
    values/  
      strings.xml
```

<code>mipmap/</code>	Drawable files for different launcher icon densities. For more information on managing launcher icons with <code>mipmap/</code> folders, see Managing Projects Overview .
<code>layout/</code>	XML files that define a user interface layout. See Layout Resource .
<code>menu/</code>	XML files that define application menus, such as an Options Menu, Context Menu, or Sub Menu. See Menu Resource .
<code>raw/</code>	<p>Arbitrary files to save in their raw form. To open these resources with a raw InputStream, call Resources.openRawResource() with the resource ID, which is <code>R.raw.filename</code>.</p> <p>However, if you need access to original file names and file hierarchy, you might consider saving some resources in the <code>assets/</code> directory (instead of <code>res/raw/</code>). Files in <code>assets/</code> are not given a resource ID, so you can read them only using AssetManager.</p>

Tema 3

Recursos

```
MyProject/  
  src/  
    MainActivity.java  
  res/  
    drawable/  
      graphic.png  
    layout/  
      main.xml  
      info.xml  
    mipmap/  
      icon.png
```

<code>values/</code>	<p>XML files that contain simple values, such as strings, integers, and colors.</p> <p>Whereas XML resource files in other <code>res/</code> subdirectories define a single resource based on the XML filename, files in the <code>values/</code> directory describe multiple resources. For a file in this directory, each child of the <code><resources></code> element defines a single resource. For example, a <code><string></code> element creates an <code>R.string</code> resource and a <code><color></code> element creates an <code>R.color</code> resource.</p> <p>Because each resource is defined with its own XML element, you can name the file whatever you want and place different resource types in one file. However, for clarity, you might want to place unique resource types in different files. For example, here are some filename conventions for resources you can create in this directory:</p> <ul style="list-style-type: none">• <code>arrays.xml</code> for resource arrays (typed arrays).• <code>colors.xml</code> for color values• <code>dimens.xml</code> for dimension values.• <code>strings.xml</code> for string values.• <code>styles.xml</code> for styles. <p>See String Resources, Style Resource, and More Resource Types.</p>
<code>xml/</code>	<p>Arbitrary XML files that can be read at runtime by calling <code>Resources.getXML()</code>. Various XML configuration files must be saved here, such as a searchable configuration.</p>

Tema 3

Recursos Conceptos

Screen size

Actual physical size, measured as the screen's diagonal.

For simplicity, Android groups all actual screen sizes into four generalized sizes: small, normal, large, and extra-large.

Screen density

The quantity of pixels within a physical area of the screen; usually referred to as dpi (dots per inch).

For example, a "low" density screen has fewer pixels within a given physical area, compared to a "normal" or "high" density screen.

For simplicity, Android groups all actual screen densities into six generalized densities: low, medium, high, extra-high, extra-extra-high, and extra-extra-extra-high.

Tema 3

Recursos Conceptos

Orientation

The orientation of the screen from the user's point of view. This is either landscape or portrait, meaning that the screen's aspect ratio is either wide or tall, respectively. Be aware that not only do different devices operate in different orientations by default, but the orientation can change at runtime when the user rotates the device.

Resolution

The total number of physical pixels on a screen. When adding support for multiple screens, applications do not work directly with resolution; applications should be concerned only with screen size and density, as specified by the generalized size and density groups.

Density-independent pixel (dp)

A virtual pixel unit that you should use when defining UI layout, to express layout dimensions or position in a density-independent way.

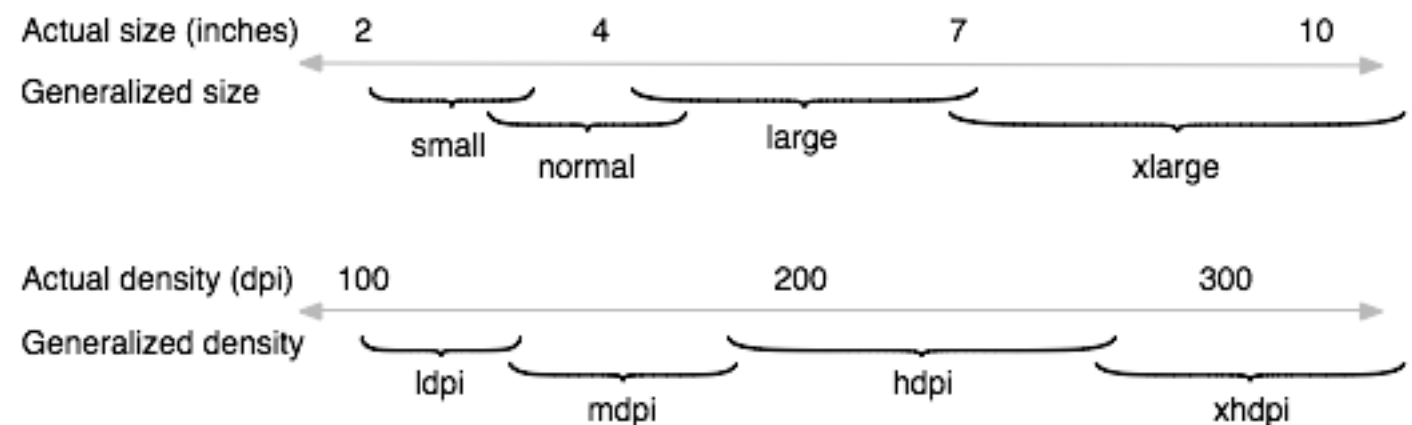
The density-independent pixel is equivalent to one physical pixel on a 160 dpi screen, which is the baseline density assumed by the system for a "medium" density screen. At runtime, the system transparently handles any scaling of the dp units, as necessary, based on the actual density of the screen in use. The conversion of dp units to screen pixels is simple: $px = dp * (dpi / 160)$. For example, on a 240 dpi screen, 1 dp equals 1.5 physical pixels. You should always use dp units when defining your application's UI, to ensure proper display of your UI on screens with different densities.

Tema 3

Recursos

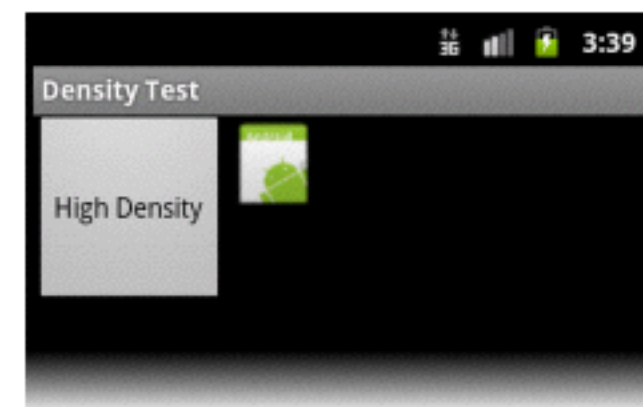
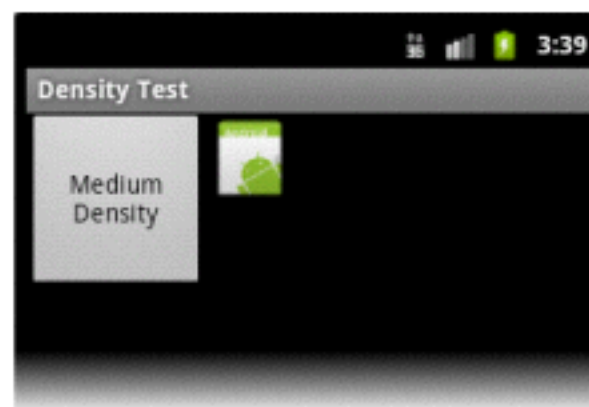
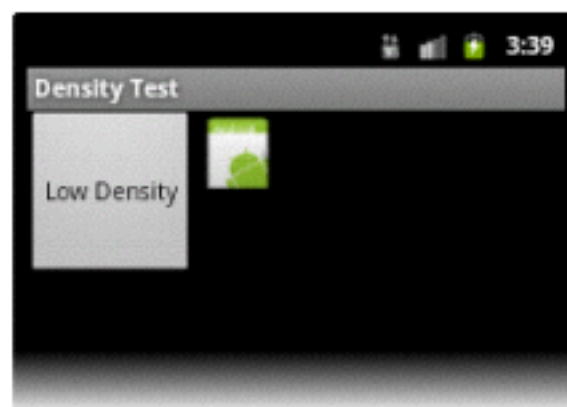
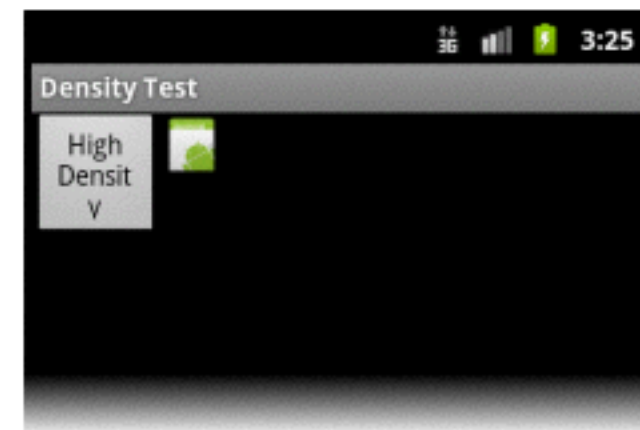
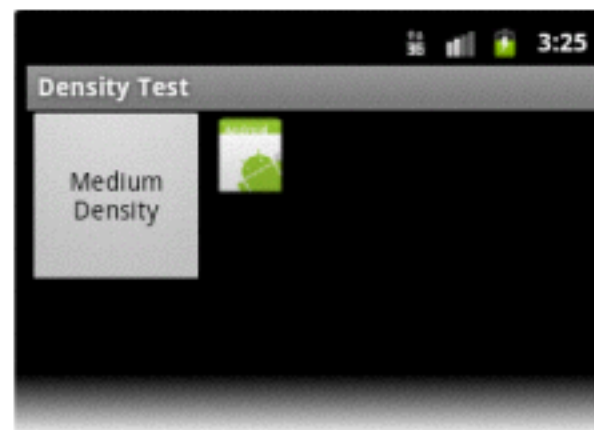
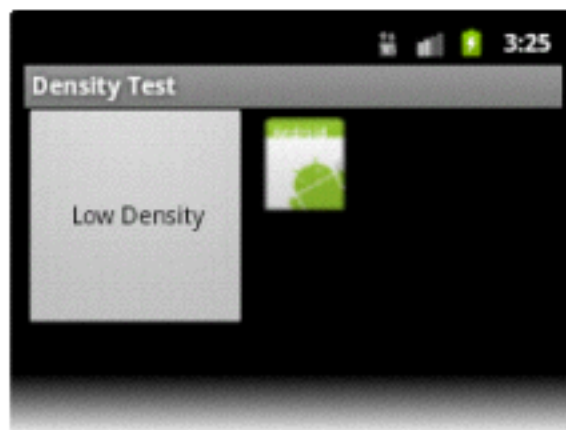
- *ldpi* (low) ~120dpi
- *mdpi* (medium) ~160dpi
- *hdpi* (high) ~240dpi
- *xhdpi* (extra-high) ~320dpi
- *xxhdpi* (extra-extra-high) ~480dpi
- *xxxhdpi* (extra-extra-extra-high) ~640dpi

- *xlarge* screens are at least 960dp x 720dp
- *large* screens are at least 640dp x 480dp
- *normal* screens are at least 470dp x 320dp
- *small* screens are at least 426dp x 320dp



Tema 3

Recursos



Tema 3

Qualifiers

Size	<code>small</code>	Resources for <i>small</i> size screens.
	<code>normal</code>	Resources for <i>normal</i> size screens. (This is the baseline size.)
	<code>large</code>	Resources for <i>large</i> size screens.
	<code>xlarge</code>	Resources for <i>extra-large</i> size screens.

Orientation	<code>land</code>	Resources for screens in the landscape orientation (wide aspect ratio).
	<code>port</code>	Resources for screens in the portrait orientation (tall aspect ratio).

Tema 3

Qualifiers

Density	ldpi	Resources for low-density (<i>ldpi</i>) screens (~120dpi).
	mdpi	Resources for medium-density (<i>mdpi</i>) screens (~160dpi). (This is the baseline density.)
	hdpi	Resources for high-density (<i>hdpi</i>) screens (~240dpi).
	xhdpi	Resources for extra-high-density (<i>xhdpi</i>) screens (~320dpi).
	xxhdpi	Resources for extra-extra-high-density (<i>xxhdpi</i>) screens (~480dpi).
	xxxhdpi	Resources for extra-extra-extra-high-density (<i>xxxhdpi</i>) uses (~640dpi). Use this for the launcher icon only, see note above.
	nodpi	Resources for all densities. These are density-independent resources. The system does not scale resources tagged with this qualifier, regardless of the current screen's density.
	tvdpi	Resources for screens somewhere between mdpi and hdpi; approximately 213dpi. This is not considered a "primary" density group. It is mostly intended for televisions and most apps shouldn't need it—providing mdpi and hdpi resources is sufficient for most apps and the system will scale them as appropriate. If you find it necessary to provide tvdpi resources, you should size them at a factor of 1.33*mdpi. For example, a 100px x 100px image for mdpi screens should be 133px x 133px for tvdpi.

Tema 3

Qualifiers

```
res/layout/my_layout.xml           // layout for normal screen size ("default")
res/layout-large/my_layout.xml     // layout for large screen size
res/layout-xlarge/my_layout.xml    // layout for extra-large screen size
res/layout-xlarge-land/my_layout.xml // layout for extra-large in landscape orientation

res/drawable-mdpi/graphic.png      // bitmap for medium-density
res/drawable-hdpi/graphic.png      // bitmap for high-density
res/drawable-xhdpi/graphic.png     // bitmap for extra-high-density
res/drawable-xxhdpi/graphic.png    // bitmap for extra-extra-high-density

res/mipmap-mdpi/my_icon.png        // launcher icon for medium-density
res/mipmap-hdpi/my_icon.png        // launcher icon for high-density
res/mipmap-xhdpi/my_icon.png       // launcher icon for extra-high-density
res/mipmap-xxhdpi/my_icon.png      // launcher icon for extra-extra-high-density
res/mipmap-xxxhdpi/my_icon.png     // launcher icon for extra-extra-extra-high-density
```

<https://developer.android.com/guide/topics/resources/providing-resources.html#AlternativeResources>

Tema 3

Qualifiers

smallestWidth	<p><code>sw<N>dp</code></p> <p>Examples:</p> <p><code>sw600dp</code></p> <p><code>sw720dp</code></p>	<p>The fundamental size of a screen, as indicated by the shortest dimension of the available screen area. Specifically, the device's smallestWidth is the shortest of the screen's available height and width (you may also think of it as the "smallest possible width" for the screen). You can use this qualifier to ensure that, regardless of the screen's current orientation, your application's has at least <code><N></code> dps of width available for its UI.</p> <p>For example, if your layout requires that its smallest dimension of screen area be at least 600 dp at all times, then you can use this qualifier to create the layout resources, <code>res/layout-sw600dp/</code>. The system will use these resources only when the smallest dimension of available screen is at least 600dp, regardless of whether the 600dp side is the user-perceived height or width. The smallestWidth is a fixed screen size characteristic of the device; the device's smallestWidth does not change when the screen's orientation changes.</p> <p>The smallestWidth of a device takes into account screen decorations and system UI. For example, if the device has some persistent UI elements on the screen that account for space along the axis of the smallestWidth, the system declares the smallestWidth to be smaller than the actual screen size, because those are screen pixels not available for your UI.</p> <p>This is an alternative to the generalized screen size qualifiers (small, normal, large, xlarge) that allows you to define a discrete number for the effective size available for your UI. Using smallestWidth to determine the general screen size is useful because width is often the driving factor in designing a layout. A UI will often scroll vertically, but have fairly hard constraints on the minimum space it needs horizontally. The available width is also the key factor in determining whether to use a one-pane layout for handsets or multi-pane layout for tablets. Thus, you likely care most about what the smallest possible width will be on each device.</p>
---------------	--	--

Tema 3

Qualifiers

Available screen width	<p>w<N>dp</p> <p>Examples:</p> <p>w720dp</p> <p>w1024dp</p>	<p>Specifies a minimum available width in dp units at which the resources should be used—defined by the <N> value. The system's corresponding value for the width changes when the screen's orientation switches between landscape and portrait to reflect the current actual width that's available for your UI.</p> <p>This is often useful to determine whether to use a multi-pane layout, because even on a tablet device, you often won't want the same multi-pane layout for portrait orientation as you do for landscape. Thus, you can use this to specify the minimum width required for the layout, instead of using both the screen size and orientation qualifiers together.</p>
Available screen height	<p>h<N>dp</p> <p>Examples:</p> <p>h720dp</p> <p>h1024dp</p> <p>etc.</p>	<p>Specifies a minimum screen height in dp units at which the resources should be used—defined by the <N> value. The system's corresponding value for the height changes when the screen's orientation switches between landscape and portrait to reflect the current actual height that's available for your UI.</p> <p>Using this to define the height required by your layout is useful in the same way as w<N>dp is for defining the required width, instead of using both the screen size and orientation qualifiers. However, most apps won't need this qualifier, considering that UIs often scroll vertically and are thus more flexible with how much height is available, whereas the width is more rigid.</p>

Tema 3

Qualifiers Ejemplos

```
res/layout/main_activity.xml          # For handsets  
res/layout-sw600dp/main_activity.xml  # For tablets
```

```
res/layout/main_activity.xml          # For handsets (smaller than 600dp available width)  
res/layout-sw600dp/main_activity.xml  # For 7" tablets (600dp wide and bigger)  
res/layout-sw720dp/main_activity.xml  # For 10" tablets (720dp wide and bigger)
```

```
res/layout/main_activity.xml          # For handsets (smaller than 600dp available width)  
res/layout-w600dp/main_activity.xml  # Multi-pane (any screen with 600dp available width or more)
```

Tema 3

Qualifiers

Consejos

1. Usar `wrap_content` y `match_parent`
2. Usar DP nunca PX
3. No usar `AbsoluteLayout`
4. Crear imágenes para cada densidad

Tema 3

Acceder a los recursos

Syntax

Here's the syntax to reference a resource in code:

```
[<package_name>.]R.<resource_type>.<resource_name>
```

- `<package_name>` is the name of the package in which the resource is located (not required when referencing resources from your own package).
- `<resource_type>` is the `R` subclass for the resource type.
- `<resource_name>` is either the resource filename without the extension or the `android:name` attribute value in the XML element (for simple values).

Tema 3

Acceder a los recursos

```
// Load a background for the current screen from a drawable resource
getWindow().setBackgroundDrawableResource(R.drawable.my_background_image) ;

// Set the Activity title by getting a string from the Resources object, because
// this method requires a CharSequence rather than a resource ID
getWindow().setTitle(getResources().getText(R.string.main_title));

// Load a custom layout for the current screen
setContentView(R.layout.main_screen);

// Set a slide in animation by getting an Animation from the Resources object
mFlipper.setInAnimation(AnimationUtils.loadAnimation(this,
    R.anim.hyperspace_in));

// Set the text on a TextView object using a resource ID
TextView msgTextView = (TextView) findViewById(R.id.msg);
msgTextView.setText(R.string.hello_message);
```

```
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/submit" />
```

Tema 3

Acceder a los recursos del sistema

```
<EditText id="text"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:textColor="?android:textColorSecondary"  
    android:text="@string/hello_world" />
```

```
setListAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, myarray));
```

Tema 3

L10N

1. `res/values/strings.xml`

Contains English text for all the strings that the application uses, including text for a string named `title`.

2. `res/values-fr/strings.xml`

Contain French text for all the strings, including `title`.

3. `res/values-ja/strings.xml`

Contain Japanese text for all the strings except `title`.

Tema 3

L10N < Android 7

User Settings	App Resources	Resource Resolution
fr_CH	default (en) de_DE es_ES fr_FR it_IT	Try fr_CH => Fail Try fr => Fail Use default (en)

Tema 3

L10N \geq Android 7

User Settings	App Resources	Resource Resolution
1. fr_CH	default (en) de_DE es_ES fr_FR it_IT	Try fr_CH => Fail Try fr => Fail Try children of fr => fr_FR Use fr_FR

Tema 3

Color

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <color name="colorPrimary">#3F51B5</color>
4   <color name="colorPrimaryDark">#303F9F</color>
5   <color name="colorAccent">#FF4081</color>
6   <color name="even_item">#FFFFFF</color>
7   <color name="odd_item">#C3C9EF</color>
8 </resources>
9
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android" >
  <item
    android:color="hex_color"
    android:state_pressed=["true" | "false"]
    android:state_focused=["true" | "false"]
    android:state_selected=["true" | "false"]
    android:state_checkable=["true" | "false"]
    android:state_checked=["true" | "false"]
    android:state_enabled=["true" | "false"]
    android:state_window_focused=["true" | "false"] />
</selector>
```

Tema 3

Drawable
Bitmaps

JPEG, PNG, GIF

```
<ImageView  
    android:layout_height="wrap_content"  
    android:layout_width="wrap_content"  
    android:src="@drawable/myimage" />
```


Tema 3

Drawable State List

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android"
    android:constantSize=["true" | "false"]
    android:dither=["true" | "false"]
    android:variablePadding=["true" | "false"] >
    <item
        android:drawable="@[package:]drawable/drawable_resource"
        android:state_pressed=["true" | "false"]
        android:state_focused=["true" | "false"]
        android:state_hovered=["true" | "false"]
        android:state_selected=["true" | "false"]
        android:state_checkable=["true" | "false"]
        android:state_checked=["true" | "false"]
        android:state_enabled=["true" | "false"]
        android:state_activated=["true" | "false"]
        android:state_window_focused=["true" | "false"] />
    </selector>
```

Tema 3

Layout <include/>

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/app_bg"
    android:gravity="center_horizontal">

    <include layout="@layout/titlebar"/>

    <TextView android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello"
        android:padding="10dp" />

    ...

</LinearLayout>
```

Tema 3

Layout
<include/>

```
<merge xmlns:android="http://schemas.android.com/apk/res/android">

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/add"/>

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/delete"/>

</merge>
```

Tema 3

Estilos

```
<TextView  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:textColor="#00FF00"  
    android:typeface="monospace"  
    android:text="@string/hello" />
```

```
<TextView  
    style="@style/CodeFont"  
    android:text="@string/hello" />
```

Tema 3

Definir estilos

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="CodeFont" parent="@android:style/TextAppearance.Medium">
        <item name="android:layout_width">fill_parent</item>
        <item name="android:layout_height">wrap_content</item>
        <item name="android:textColor">#00FF00</item>
        <item name="android:typeface">monospace</item>
    </style>
</resources>
```

Tema 3

Heredar estilos

```
<style name="GreenText" parent="@android:style/TextAppearance">  
    <item name="android:textColor">#00FF00</item>  
</style>
```

```
<style name="CodeFont.Red">  
    <item name="android:textColor">#FF0000</item>  
</style>
```

```
<style name="CodeFont.Red.Big">  
    <item name="android:textSize">30sp</item>  
</style>
```

Tema 3

Propiedades

<code>android:breakStrategy</code>	Break strategy (control over paragraph layout).
<code>android:bufferType</code>	Determines the minimum type that <code>getText()</code> will return.
<code>android:capitalize</code>	If set, specifies that this <code>TextView</code> has a textual input method and should automatically capitalize what the user types.
<code>android:cursorVisible</code>	Makes the cursor visible (the default) or invisible.
<code>android:digits</code>	If set, specifies that this <code>TextView</code> has a numeric input method and that these specific characters are the ones that it will accept.
<code>android:drawableBottom</code>	The drawable to be drawn below the text.
<code>android:drawableEnd</code>	The drawable to be drawn to the end of the text.
<code>android:drawableLeft</code>	The drawable to be drawn to the left of the text.
<code>android:drawablePadding</code>	The padding between the drawables and the text.
<code>android:drawableRight</code>	The drawable to be drawn to the right of the text.
<code>android:drawableStart</code>	The drawable to be drawn to the start of the text.
<code>android:drawableTint</code>	Tint to apply to the compound (left, top, etc.) drawables.
<code>android:drawableTintMode</code>	Blending mode used to apply the compound (left, top, etc.) drawables tint.
<code>android:drawableTop</code>	The drawable to be drawn above the text.
<code>android:editable</code>	If set, specifies that this <code>TextView</code> has an input method.
<code>android:editorExtras</code>	Reference to an <code><input-extras></code> XML resource containing additional data to supply to an input method, which is private to the implementation of the input method.

Tema 3

Aplicar estilos

```
<TextView  
    style="@style/CodeFont"  
    android:text="@string/hello" />
```

```
<application android:theme="@style/CustomTheme">
```

```
<activity android:theme="@android:style/Theme.Dialog">
```

```
<activity android:theme="@android:style/Theme.Translucent">
```


Tema 3

Aplicar estilos

```
<color name="custom_theme_color">#b0b0ff</color>
<style name="CustomTheme" parent="android:Theme.Light">
    <item name="android:windowBackground">@color/custom_theme_color</item>
    <item name="android:colorBackground">@color/custom_theme_color</item>
</style>
```

```
<activity android:theme="@style/CustomTheme">
```

Tema 3

Heredar de estilos predefinidos

```
<style name="LightThemeSelector" parent="android:Theme.Light">  
    ...  
</style>
```

```
<style name="LightThemeSelector" parent="android:Theme.Holo.Light">  
    ...  
</style>
```

<https://android.googlesource.com/platform/frameworks/base/+refs/heads/master/core/res/res/values/themes.xml>

Tema 3

Ejercicios

Sobre la aplicación biblioteca de la unidad 2:

1. Crear dos vistas para la pantalla de listado de libros, una para teléfonos y otra para tabletas.
2. Traducir la aplicación al Inglés y al castellano.
3. Crear un estilo general y aplicarlo a toda la aplicación
4. Crear un estilo para teléfonos y otro para tabletas.