

Tema 1

Conceptos Básicos

1. Materials UI
2. Patrones (Acciones/Navegación)
3. Android Studio
4. Tipo de proyectos
5. Depuración
6. Interfaz de usuario
7. Ejercicios

Material UI

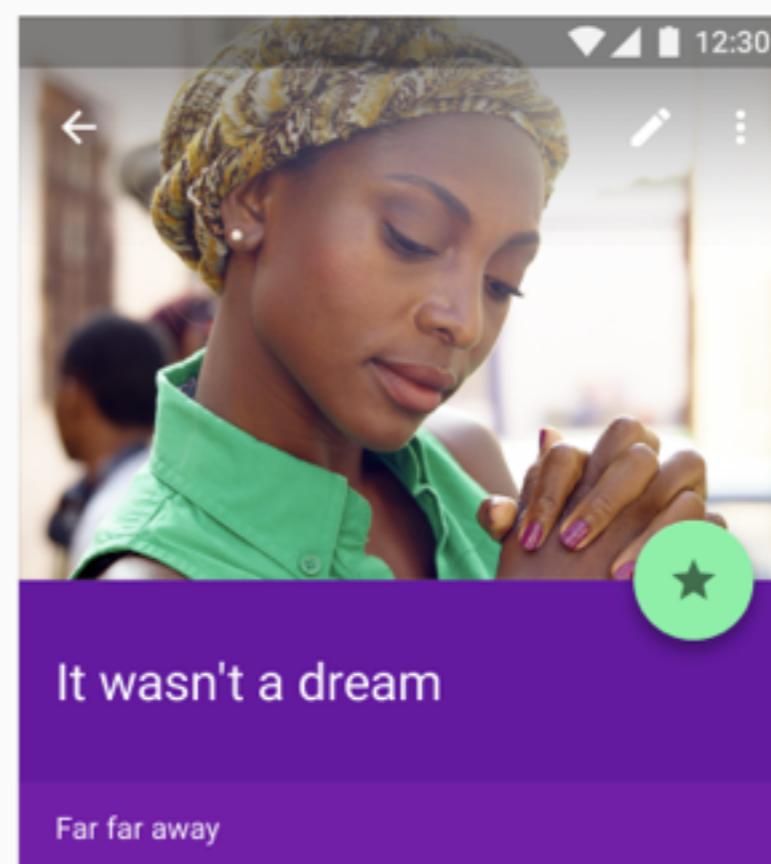
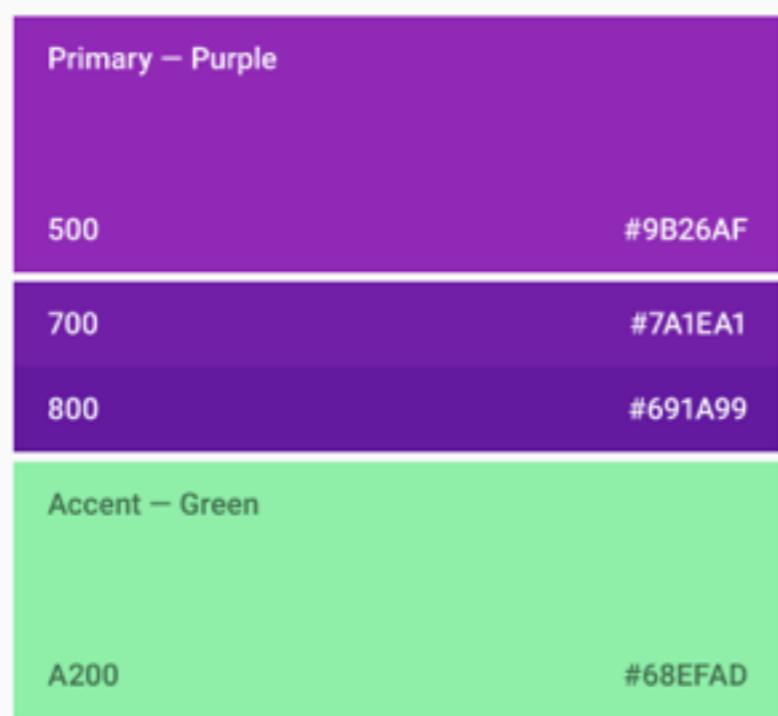
- Desarrollar un sistema que facilite la creación de una experiencia única a través de distintas plataformas y dispositivos.



Material UI: Estilos

- Proporciona una paleta de colores y un esquema de color.

<https://material.google.com/style/color.html>



Material UI: Estilos

Primary color

When using a primary color in your palette, this color should be the most widely used across all screens and components.

Primary – Indigo

500	#3F51B5
100	#C5CAE9
500	#3F51B5
700	#303F9F

Example of a primary color palette with variations for when a darker or lighter version of the color is needed

Material UI: Estilos

Secondary color

Palettes with a secondary color may use this color to indicate a related action or information.

The secondary color may be a darker or lighter variation of the primary color.

Accent – Pink

A200

#FF4081

Fallback

A100

#FF80AB

A400

#F50057

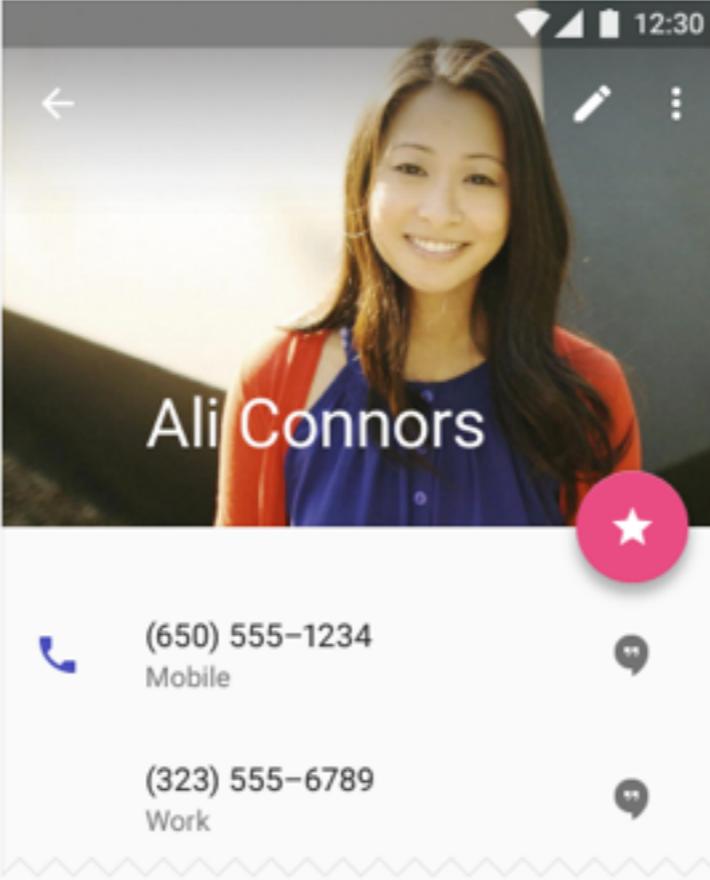
Example of a secondary palette with variations for when a darker or lighter version of the color is needed

Material UI: Estilos

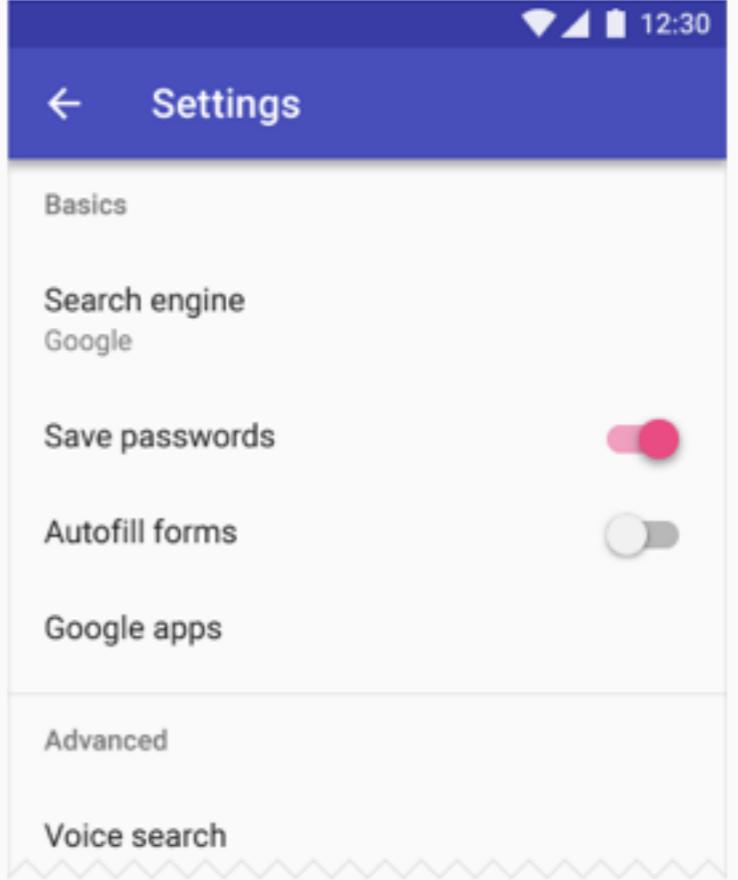
Accent color

The accent should be used for the floating action button and interactive elements, such as:

- Text fields and cursors
- Text selection
- Progress bars
- Selection controls, buttons, and sliders
- Links



Floating action button using the accent color



Switch using the accent color

Material UI: Estilos

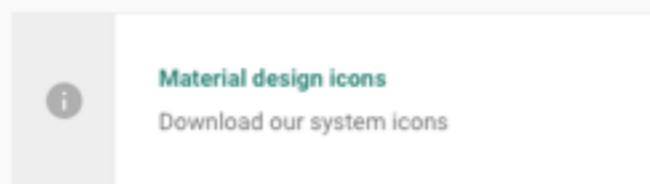
- Proporciona un conjunto de iconos para las acciones mas comunes.

<https://material.google.com/style/icons.html>

System icons

A system icon, or UI icon, symbolizes a command, file, device, or directory. System icons are also used to represent common actions like trash, print, and save.

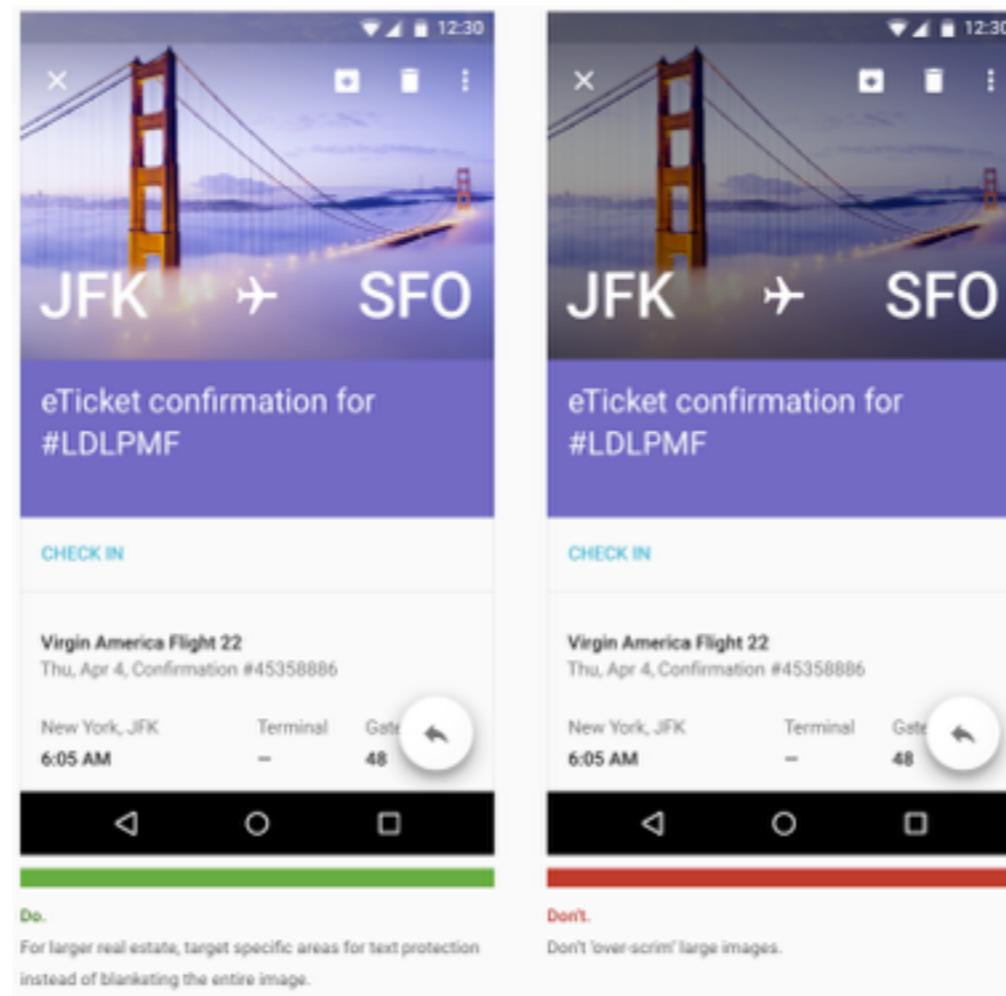
The design of system icons is simple, modern, friendly, and sometimes quirky. Each icon is reduced to its minimal form, with every idea edited to its essence. The designs ensure readability and clarity even at small sizes.



Material UI: Estilos

- Proporciona un conjunto de recomendaciones para el uso de imágenes.

<https://material.google.com/style/imagery.html>

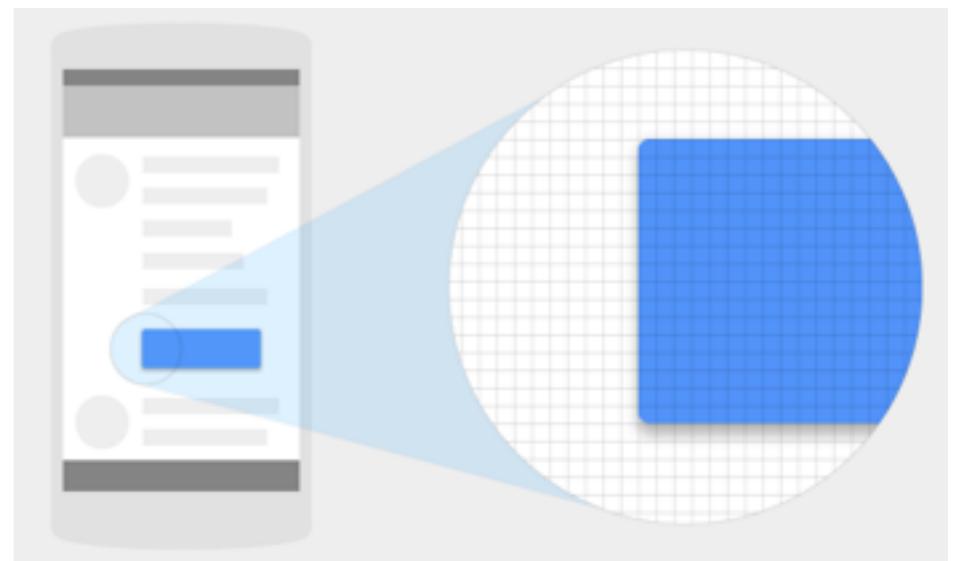


Material UI: Layouts

- Pixel Density:

Es el número de 'pixels' que caben en un pulgada (de pantalla).

screen density = screen width (or height) in pixels / screen width (or height) in inches



Material UI: Layouts

- Density-independent pixel (dp):

Es una unidad flexible que escala de forma uniforme en pantallas de distinto tamaño y densidad.

A dp is equal to one physical pixel on a screen with a density of 160. To calculate dp:

dp = (width in pixels * 160) / screen density

Material UI: Layouts

- Density-independent pixel (dp):

Screen density	Screen width in pixels	Screen width in density-independent pixels
120	180 px	240 dp
160	240 px	240 dp
240	360 px	240 dp

The diagram illustrates the relationship between screen density, screen width in pixels, and screen width in density-independent pixels. It shows three smartphone icons representing different screen densities. The first icon, labeled 'Pixels', has a blue circle in the center of its white content area. The second icon, labeled 'Density Independent Pixels', has a much larger blue circle in the center of its white content area, demonstrating how the same physical pixel size appears smaller at higher screen densities.

Pixels

Density Independent Pixels

Material UI: Layouts

- Scalable Pixel (sp):
 - Es el concepto de 'dp' llevado a las fuentes de texto.
 - La única diferencia es que usa las preferencias de texto del usuario marcadas en la sección de accesibilidad.

Material UI: Layouts

- Density-independent pixel (dp):

En android debemos pensar en 'dp'

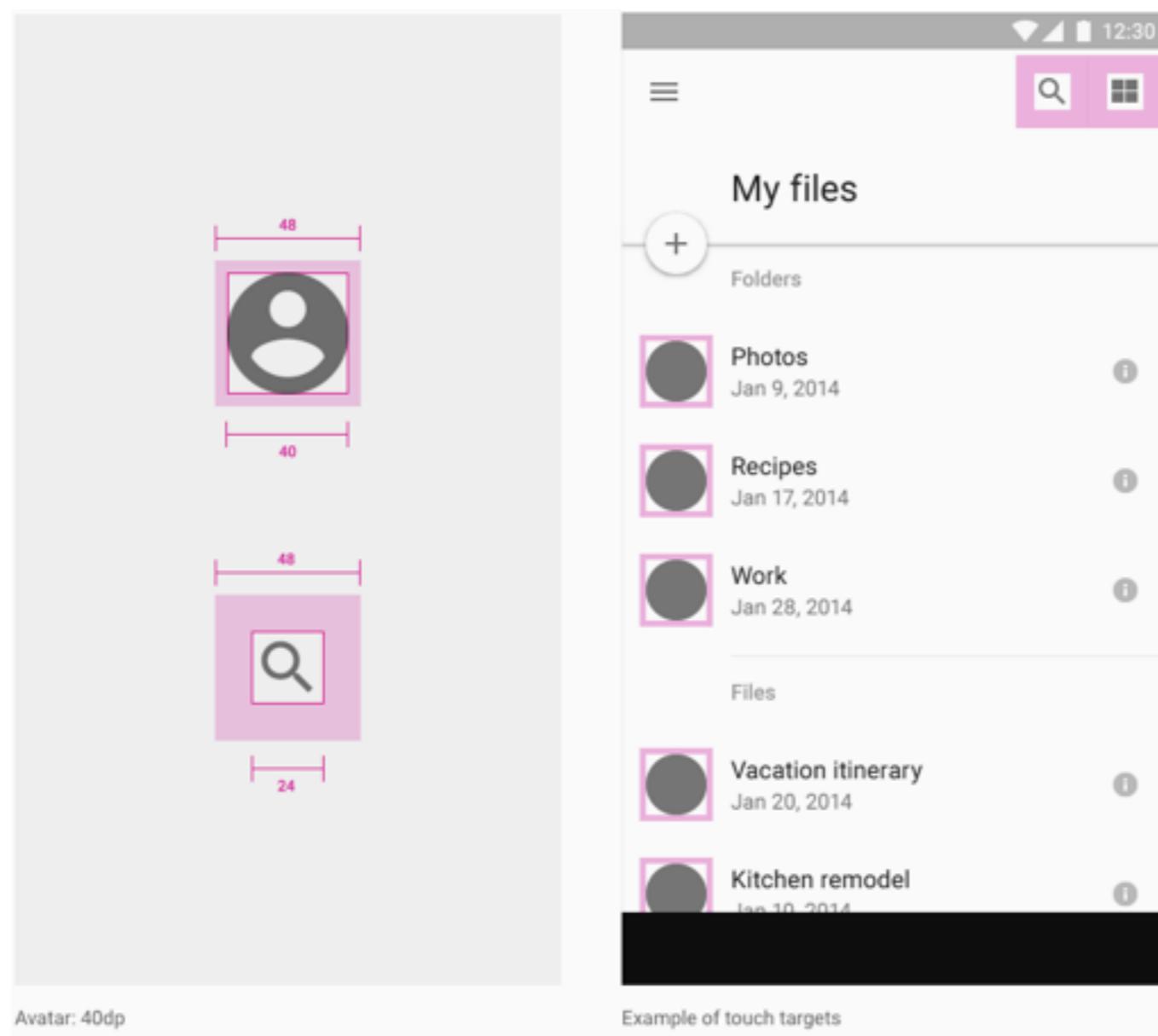
$$dp = (\text{width in pixel} * 160) / \text{density}$$

Ejemplo:

Un icono de 32x32 px en una pantalla
de 320 dpi lo indicaríamos como
16x16dp

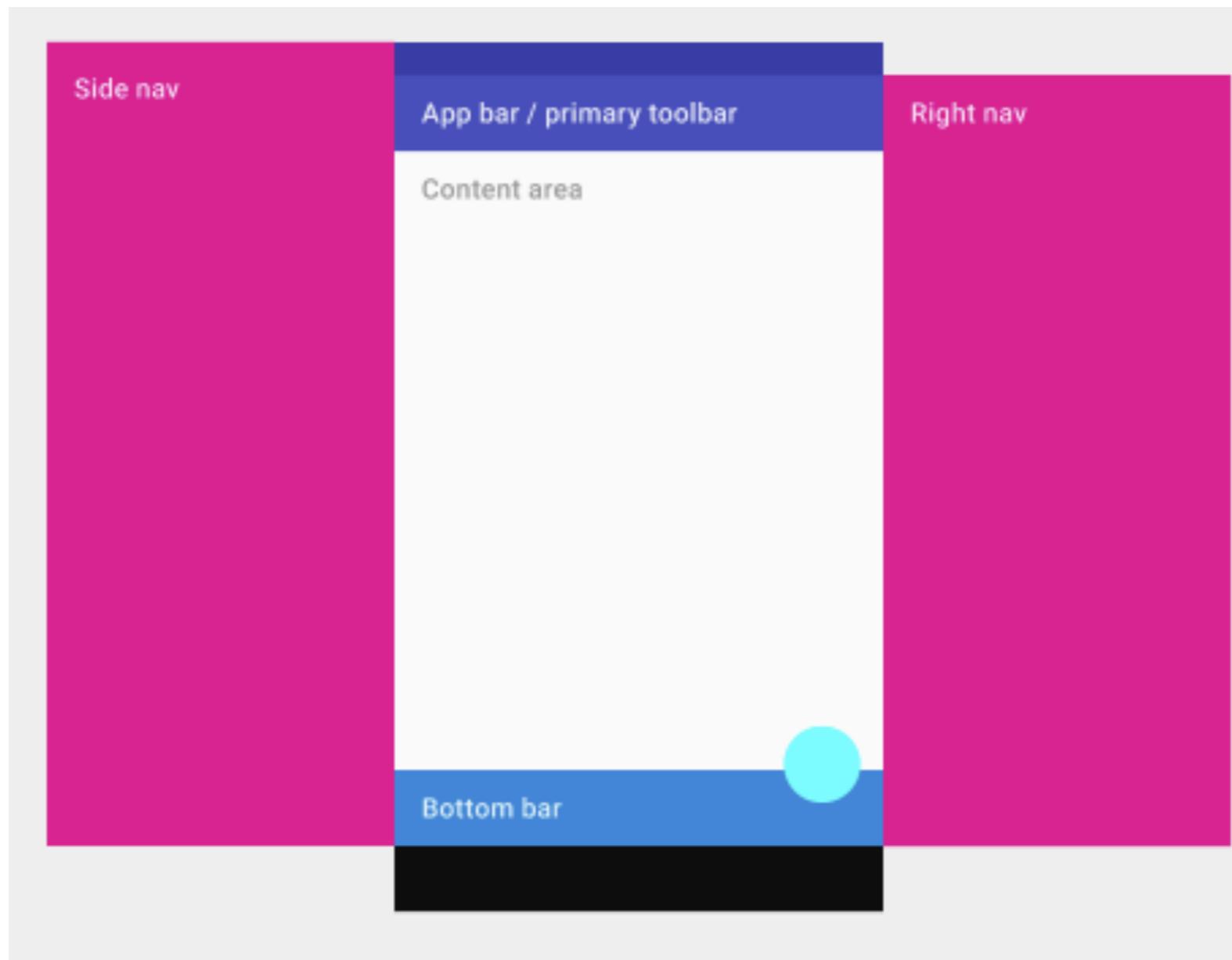
Material UI: Layouts

- Pensado para usarse con los dedos:



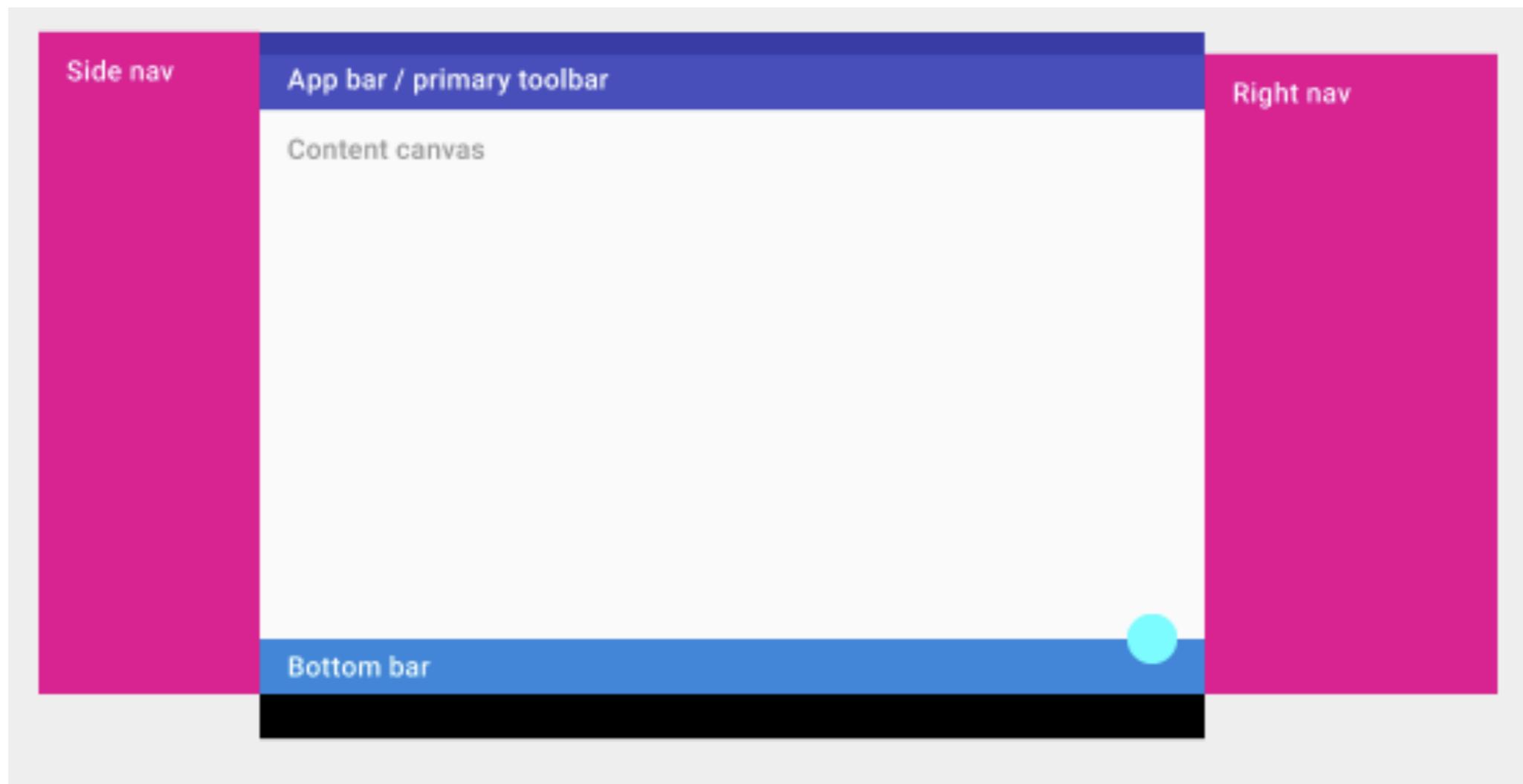
Material UI: Layouts

- Distribución para móviles:



Material UI: Layouts

- Distribución para tablets:



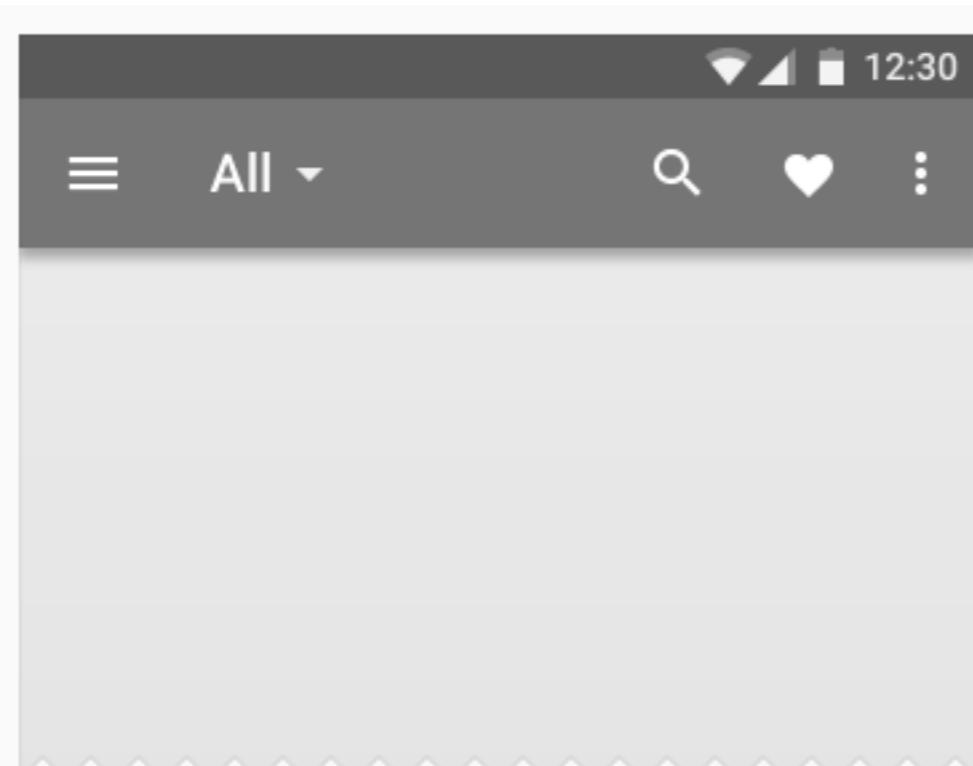
Material UI: Componentes

- Action Bar:

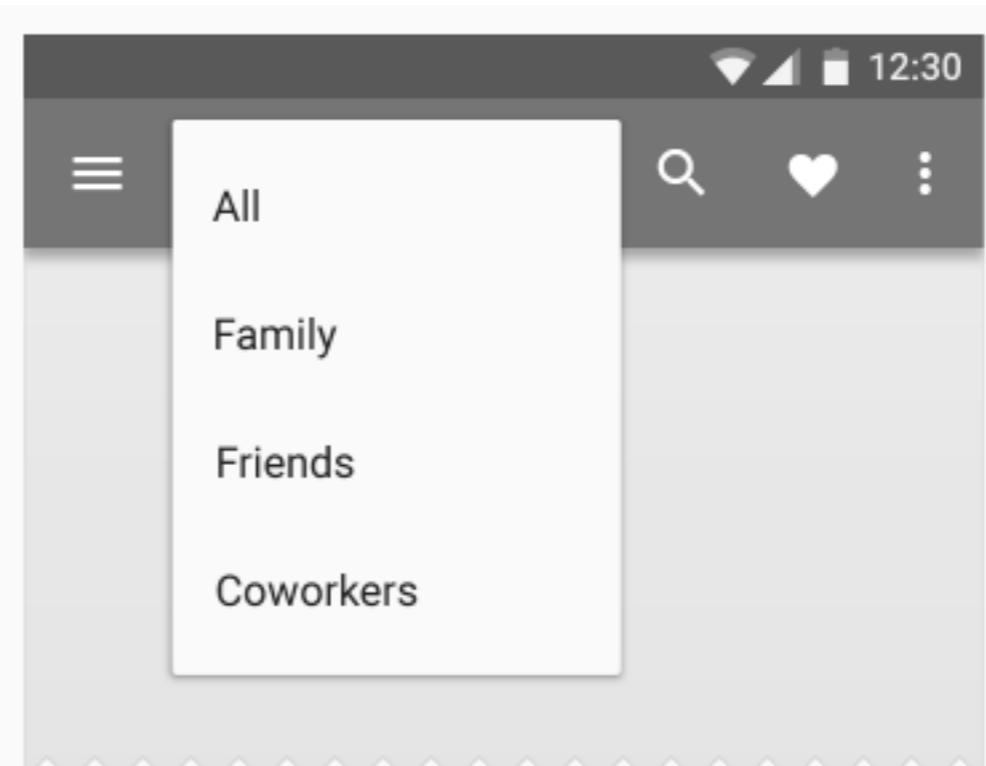


Material UI: Componentes

- Menús:



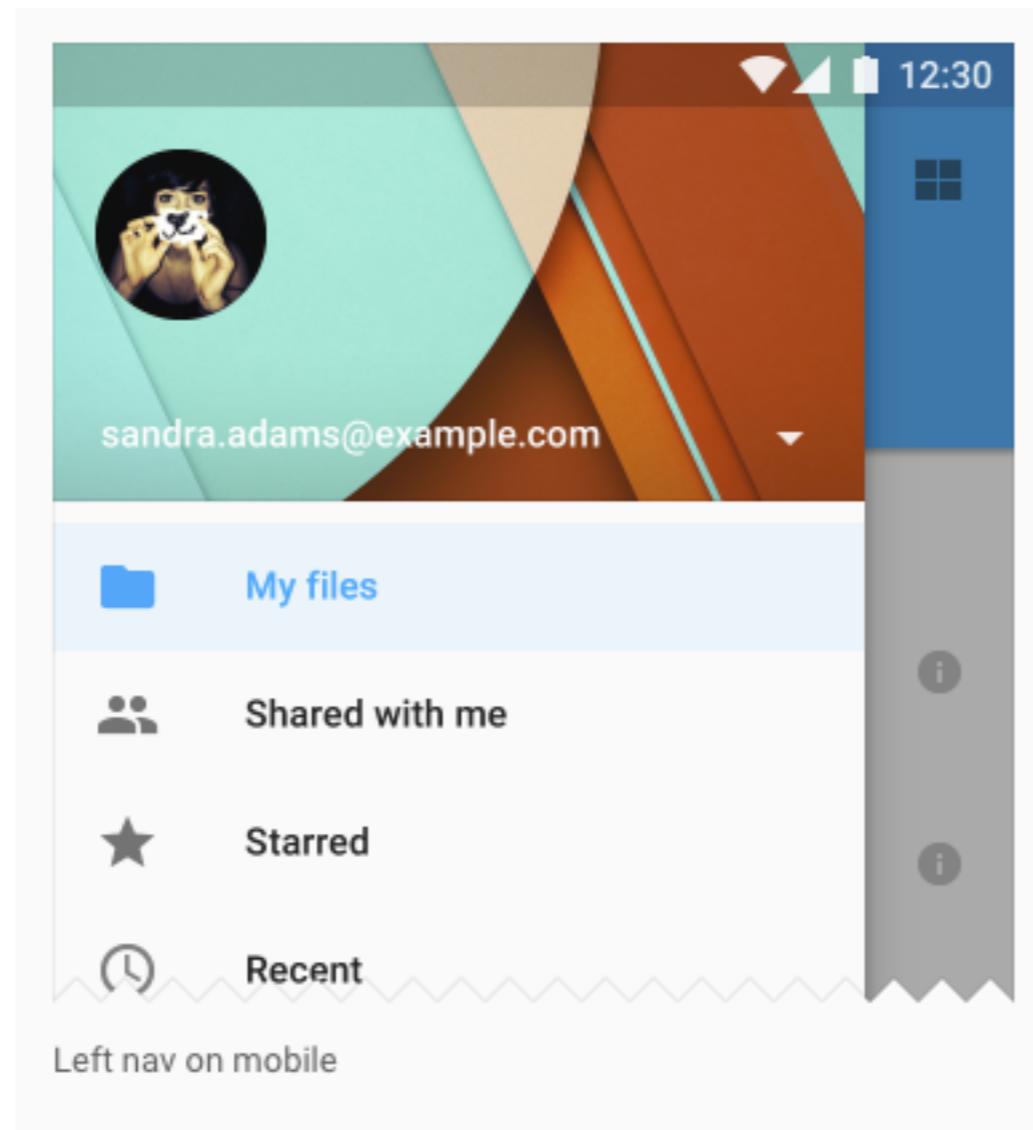
Example of an app bar



Example of menu in an app bar

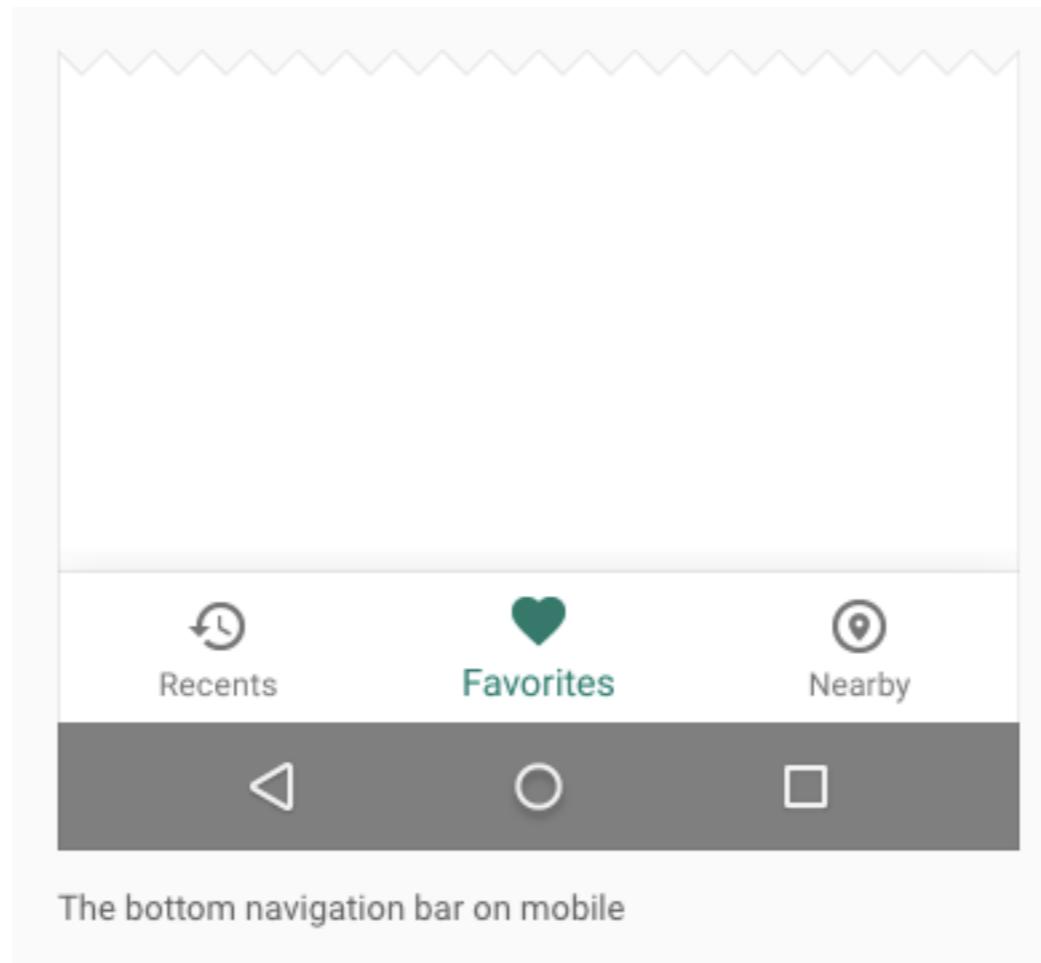
Material UI: Componentes

- Barra lateral:



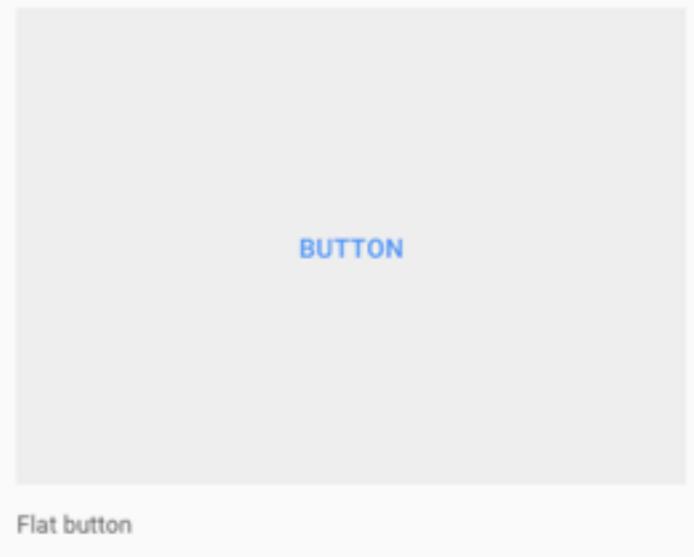
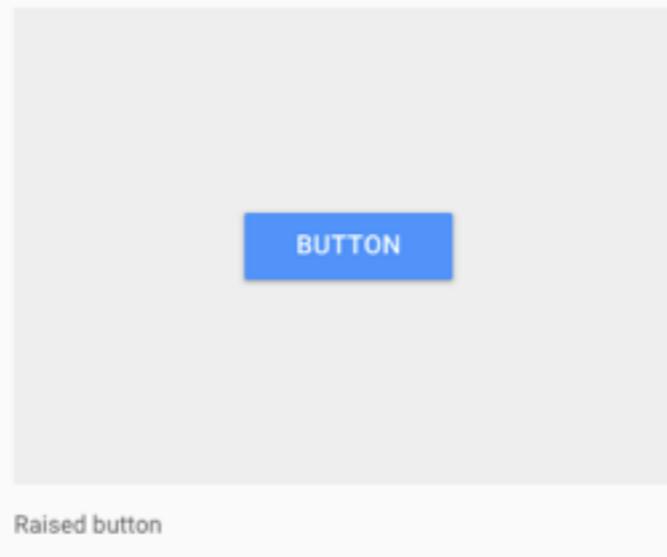
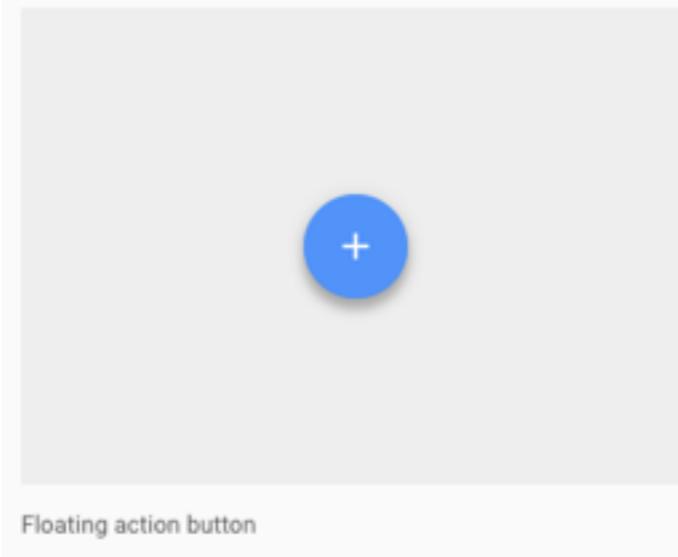
Material UI: Componentes

- Boton de navegación:



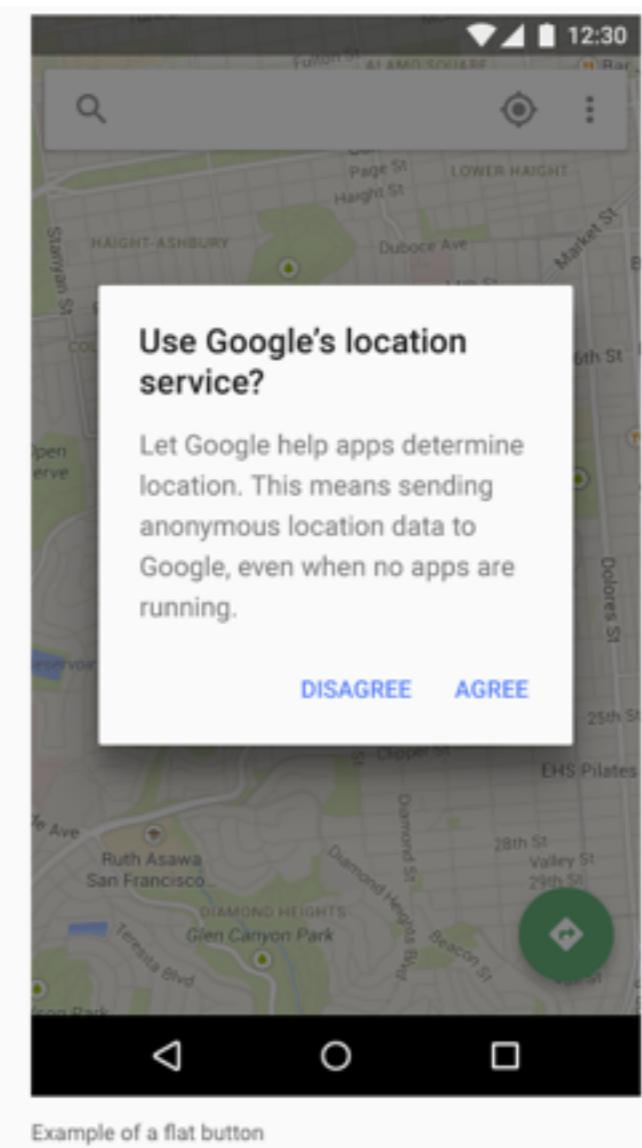
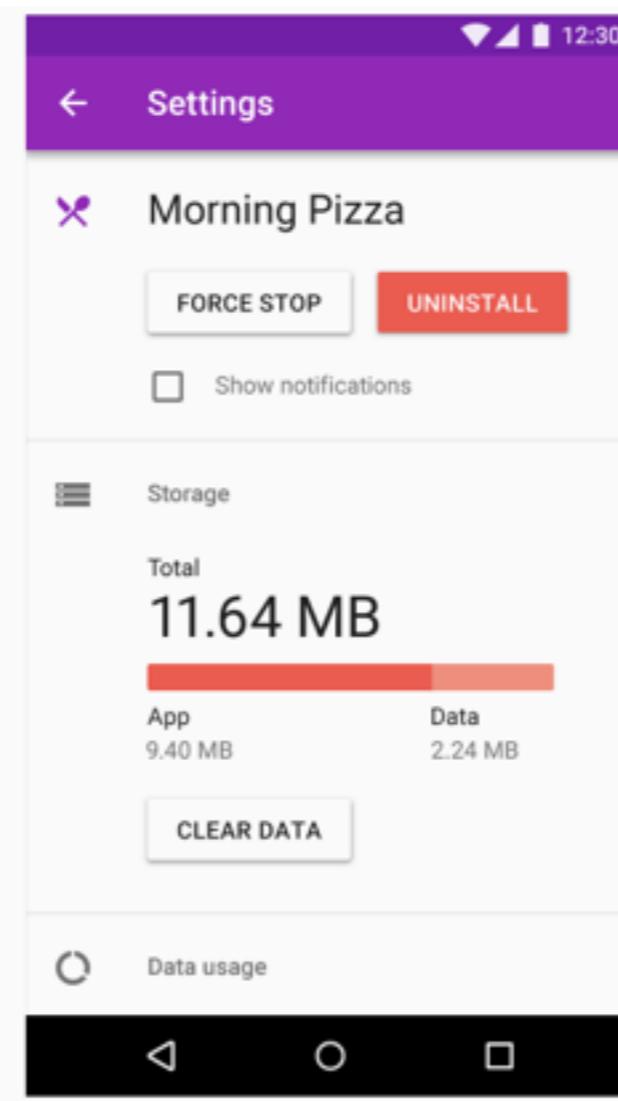
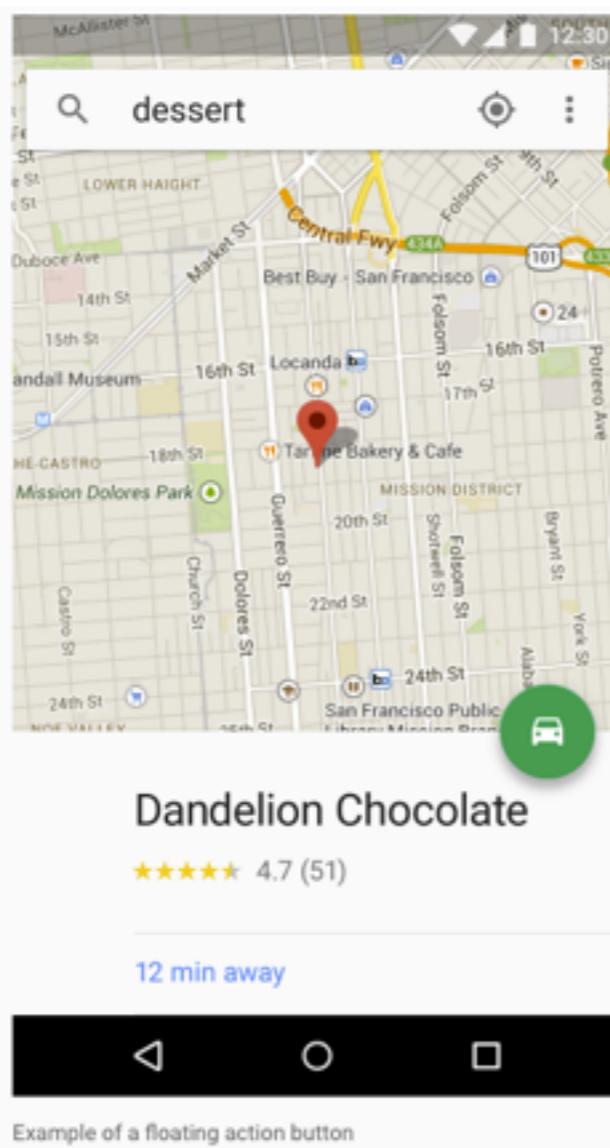
Material UI: Componentes

- Botones:



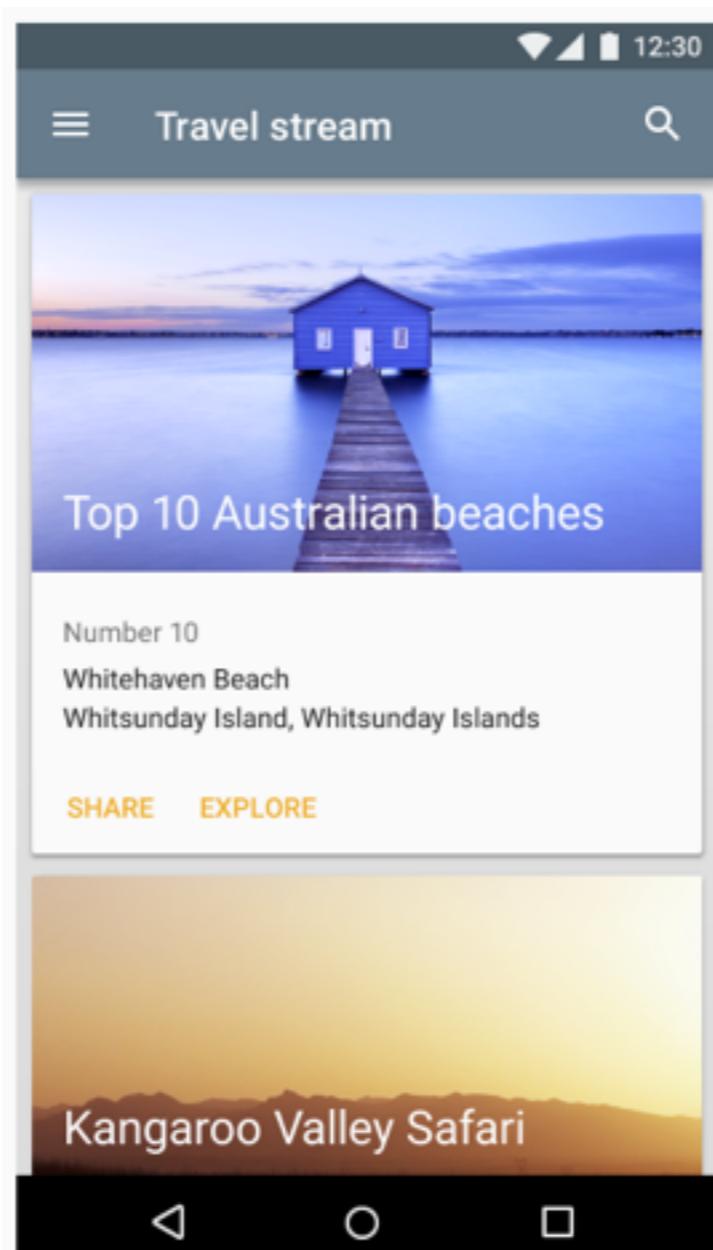
Material UI: Componentes

- Botones:

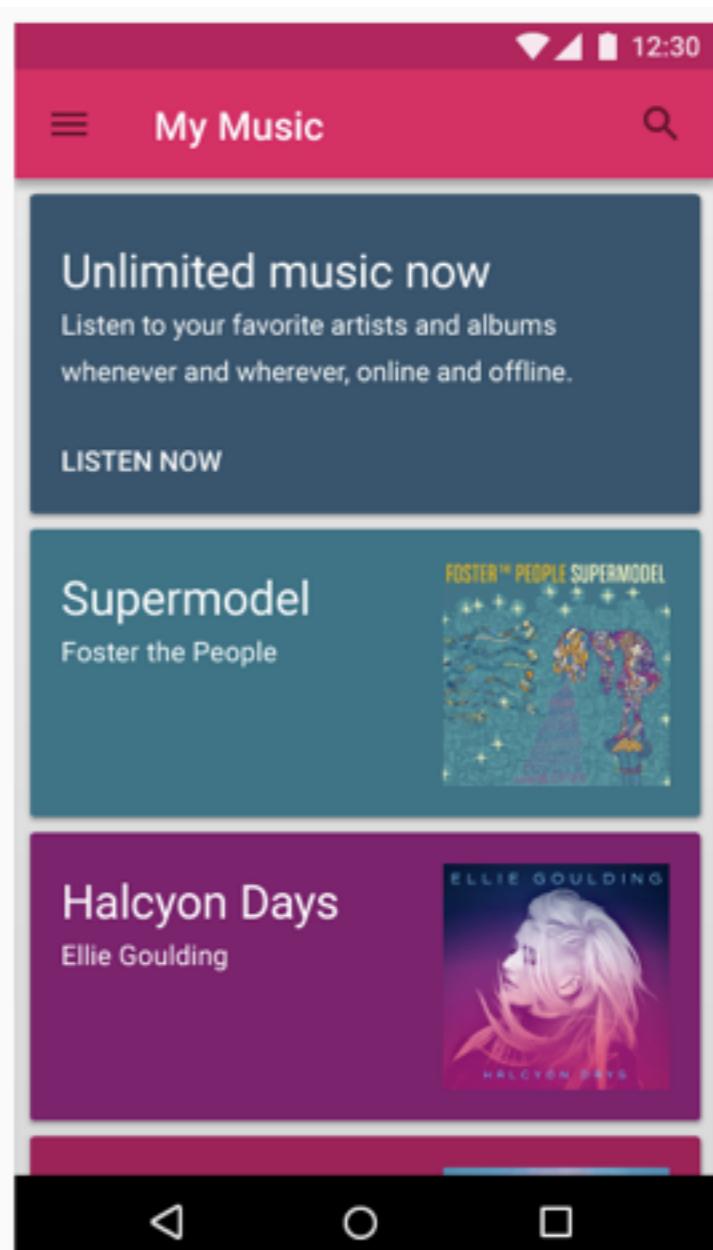


Material UI: Componentes

- Cards:



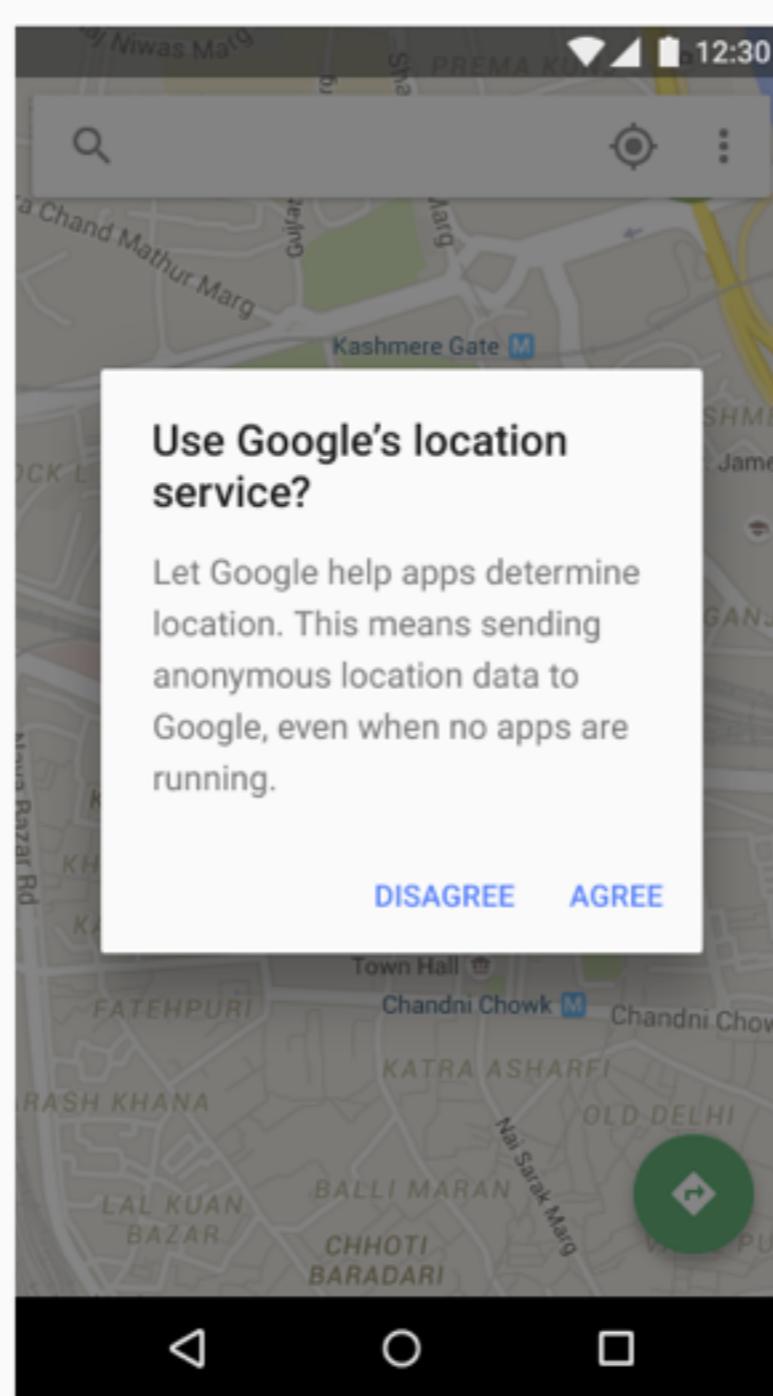
Example of a card collection



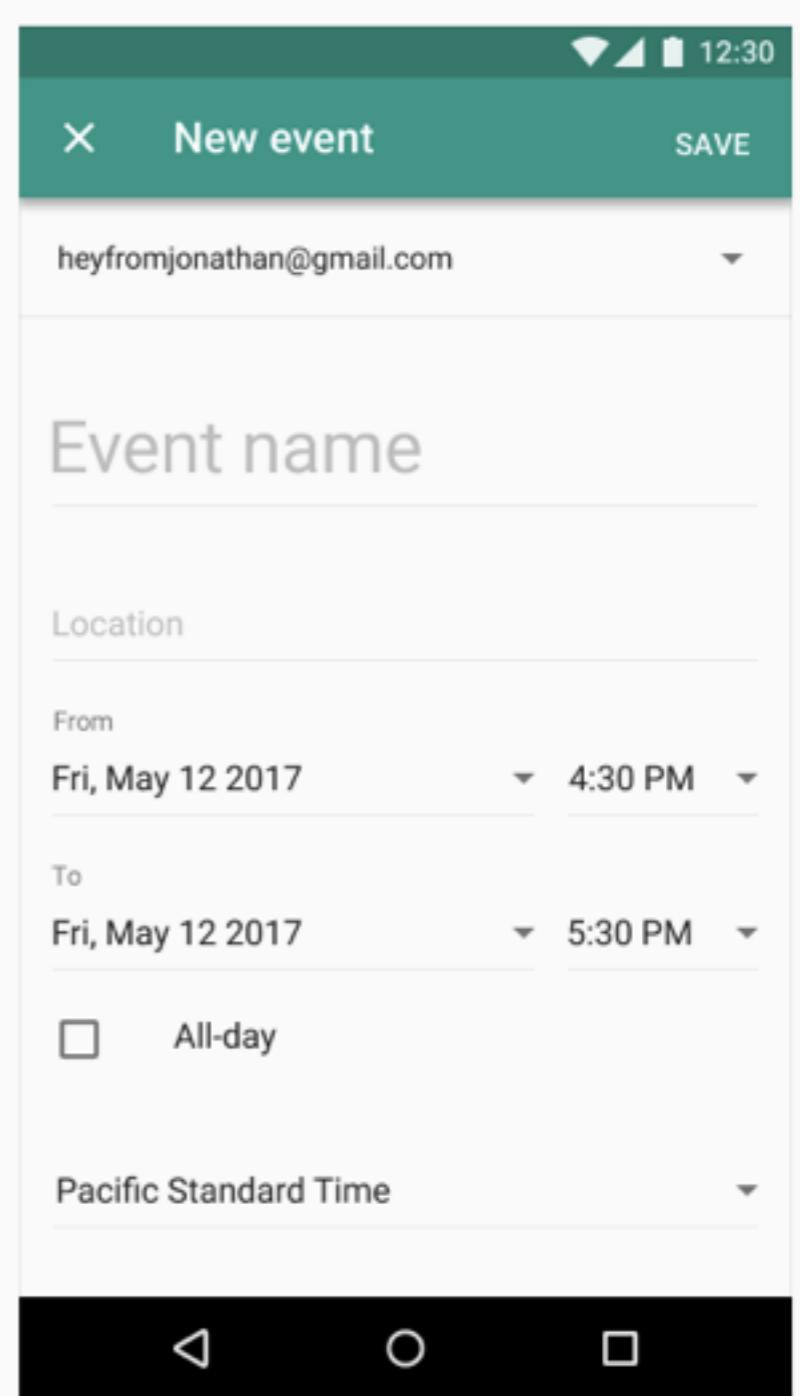
Example of a card collection

Material UI: Componentes

- Dialogos:



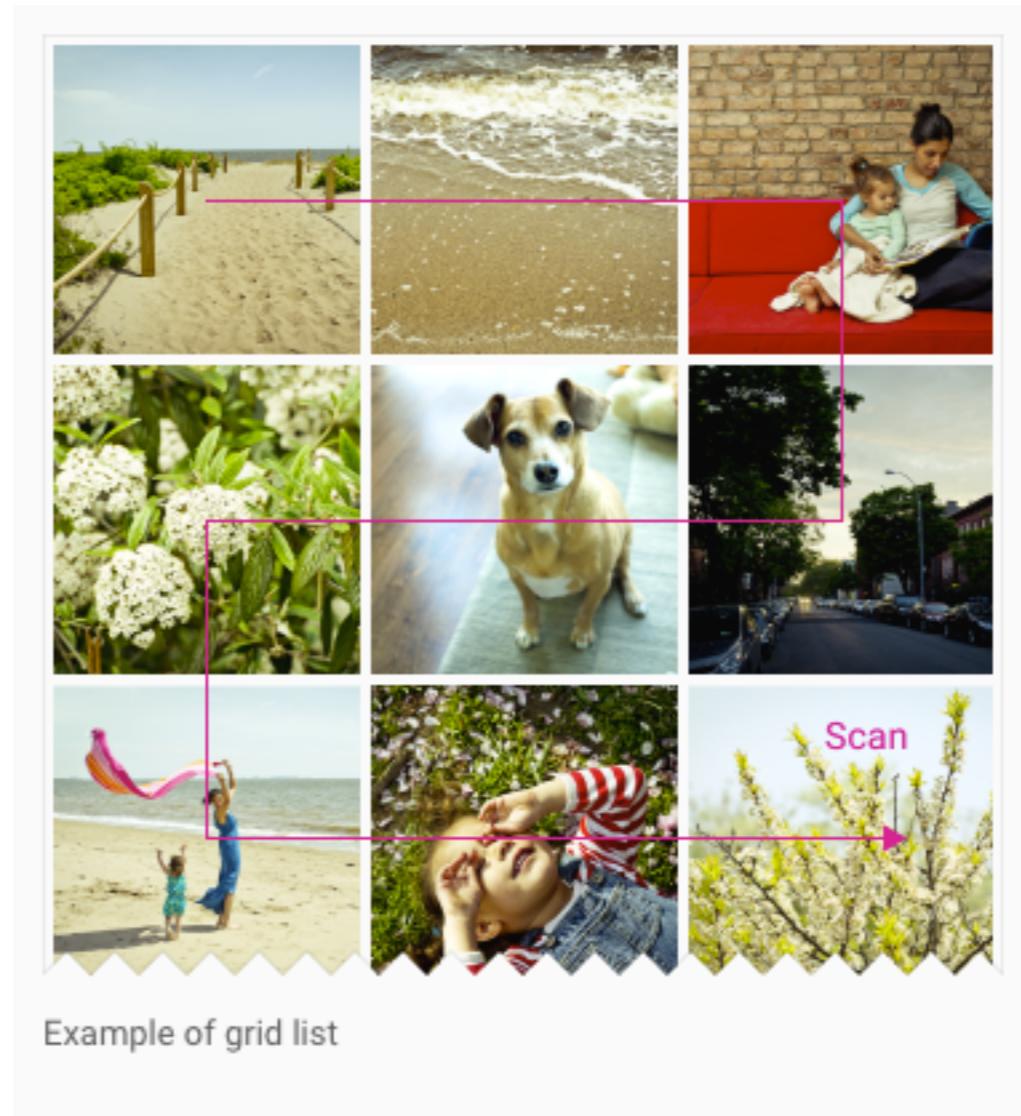
Example of dialog content



Example of a full-screen dialog

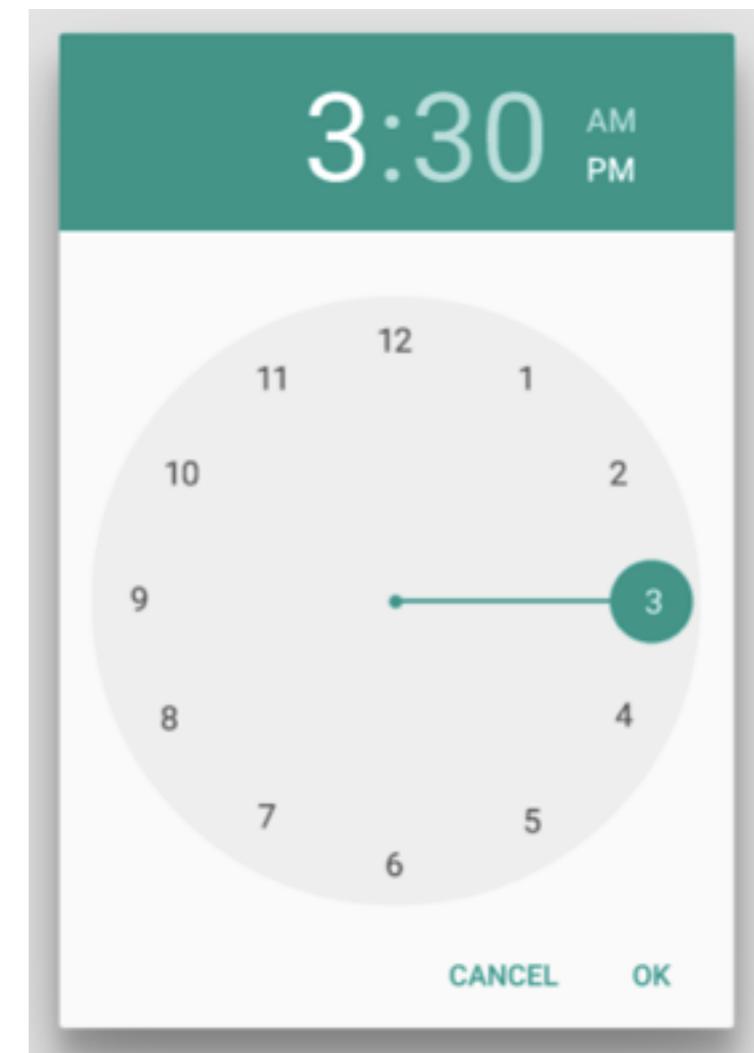
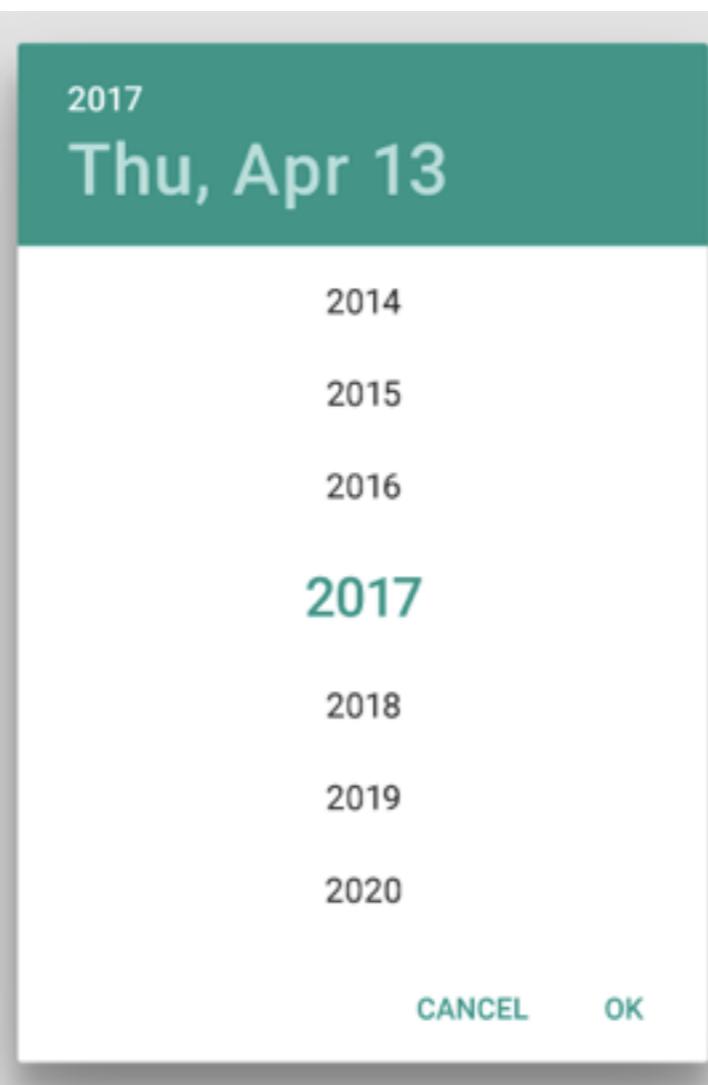
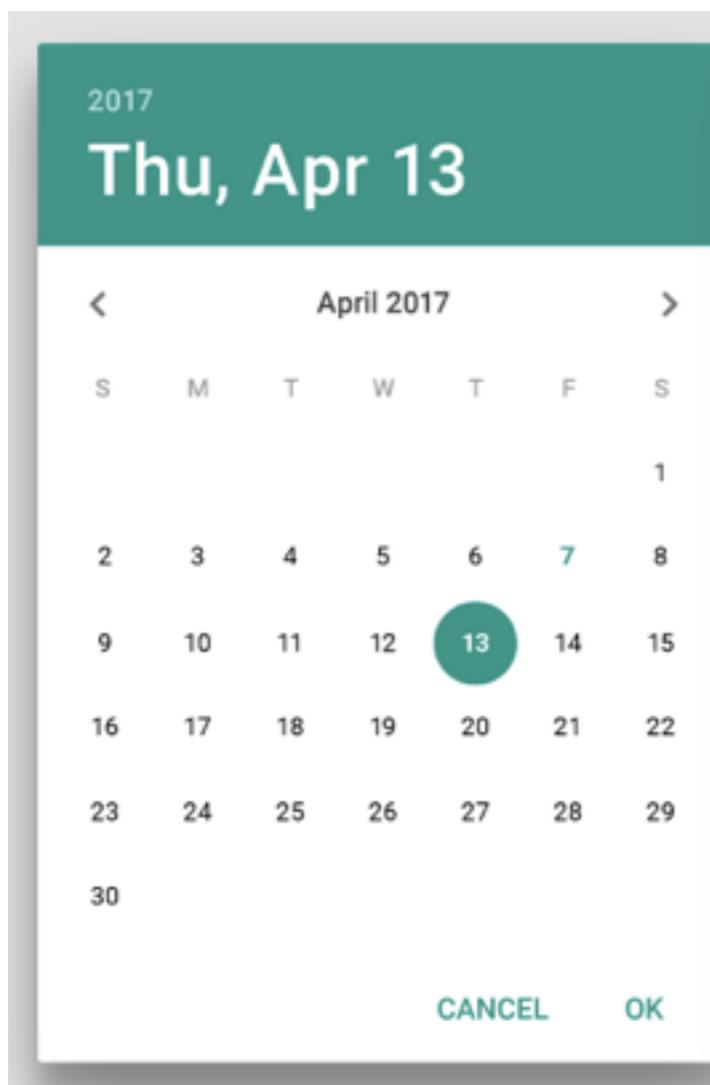
Material UI: Componentes

- Listas (Grid):



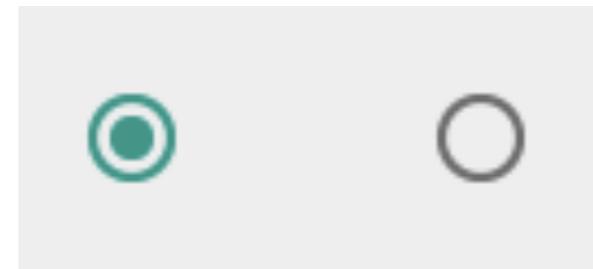
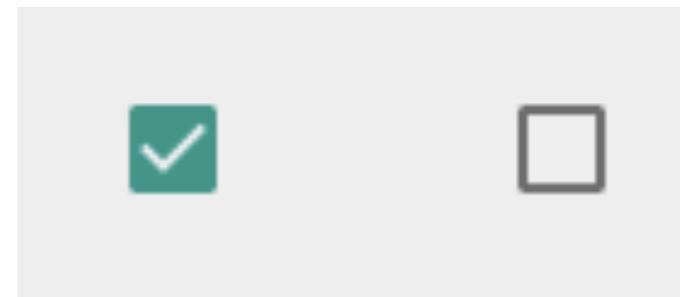
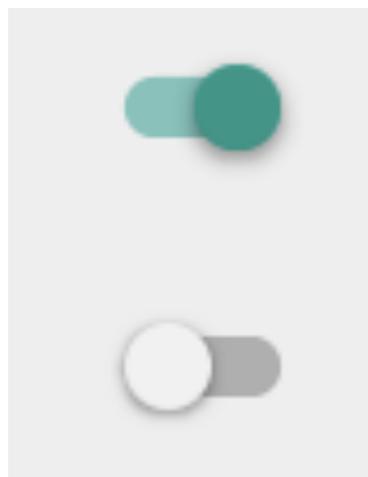
Material UI: Componentes

- Pickers:



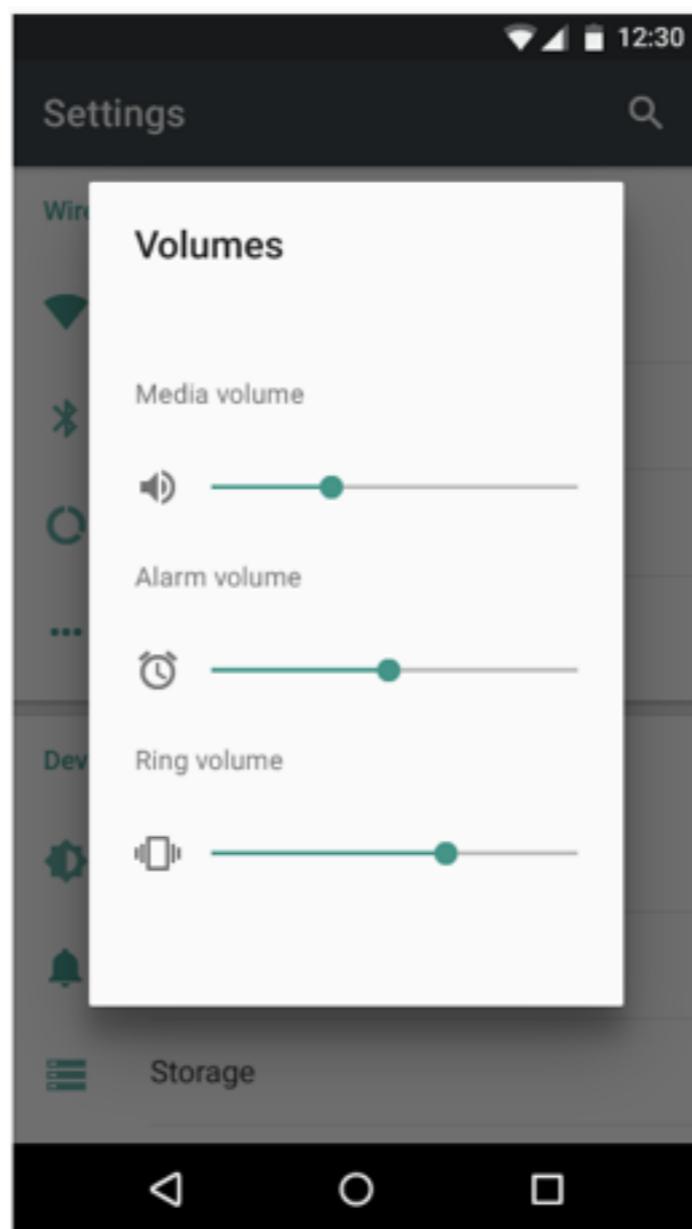
Material UI: Componentes

- Selectores:



Material UI: Componentes

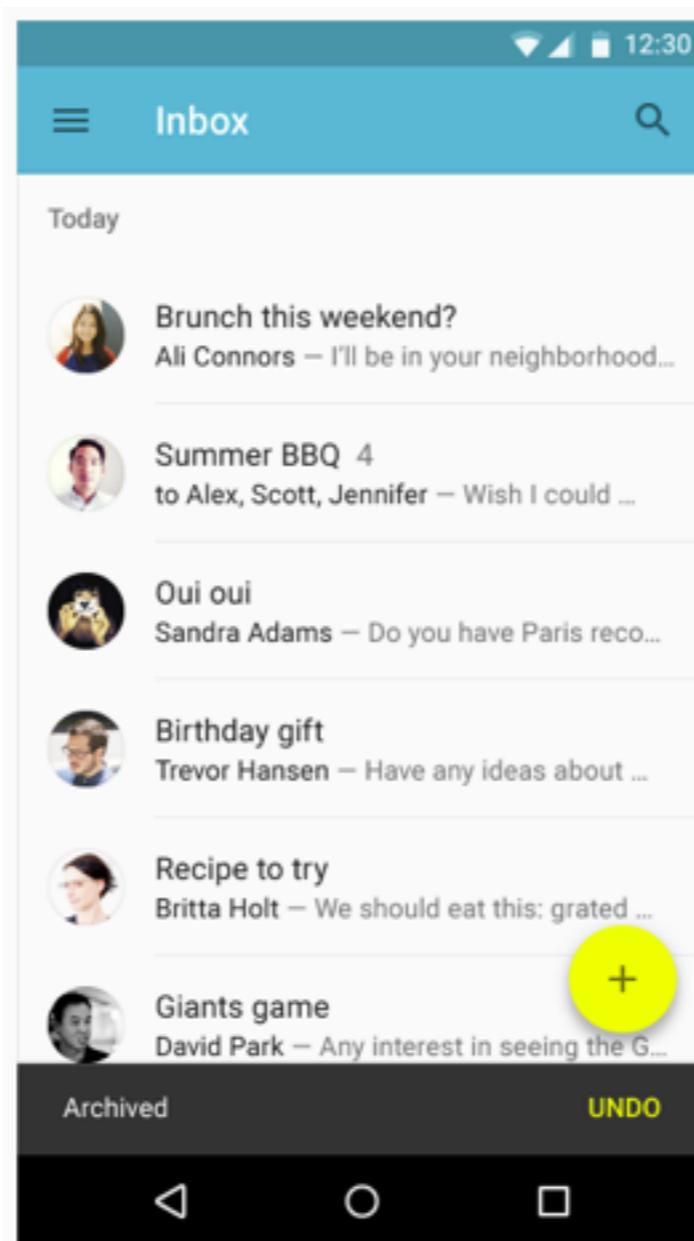
- Sliders:



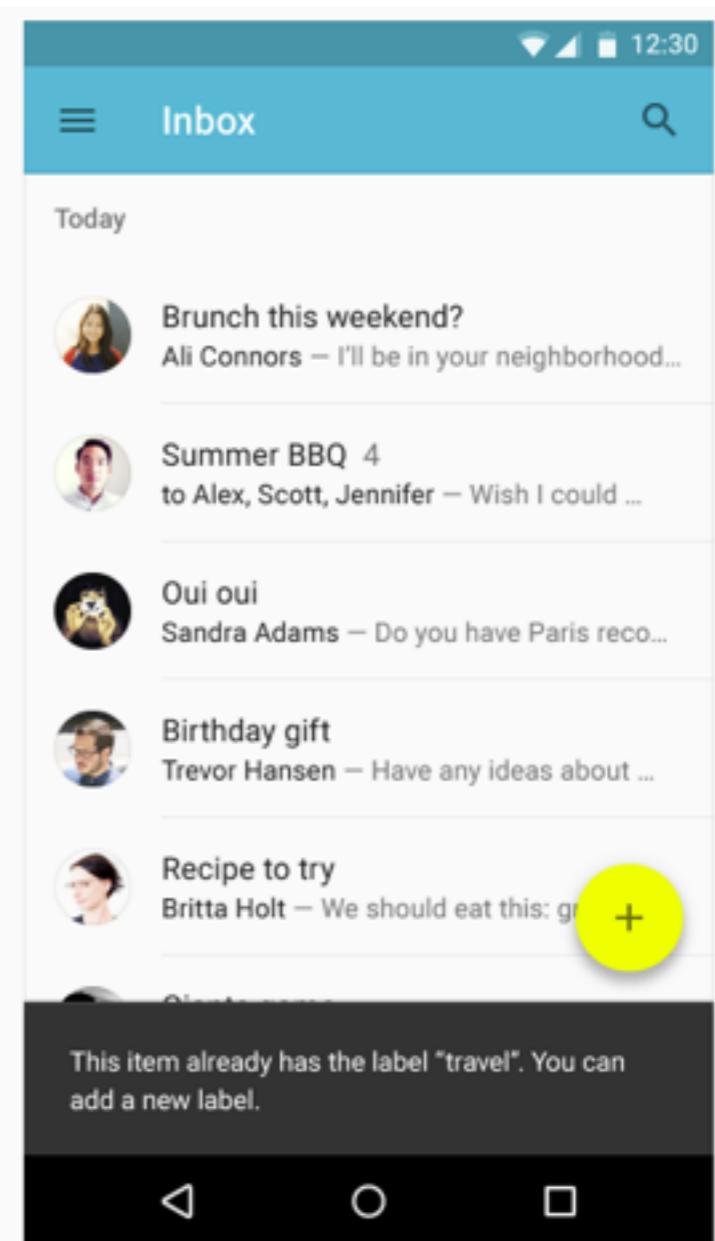
Example of spacing for icons in a slider.

Material UI: Componentes

- Snackbar y Toast:



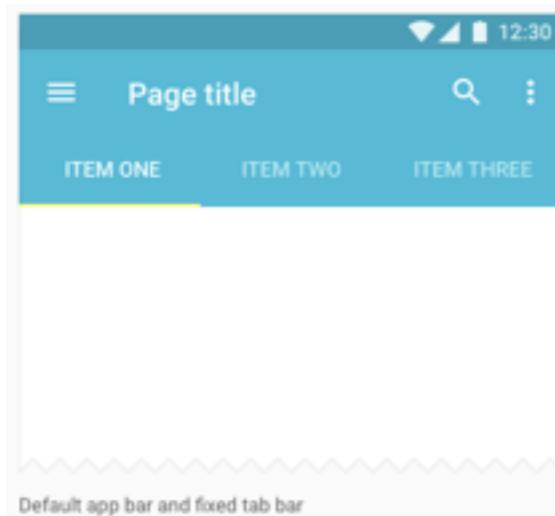
Example of a mobile Snackbar



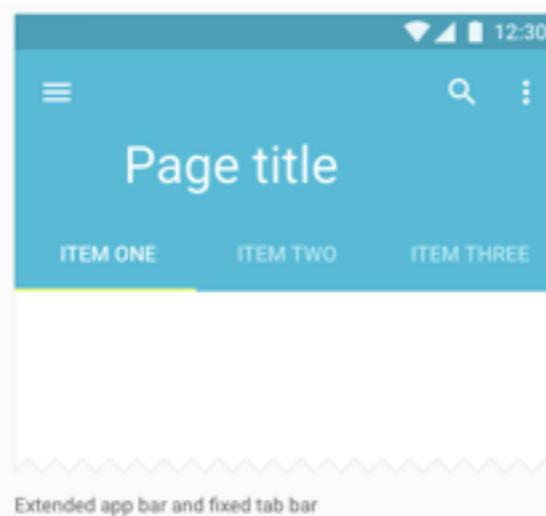
Example of a mobile Snackbar

Material UI: Componentes

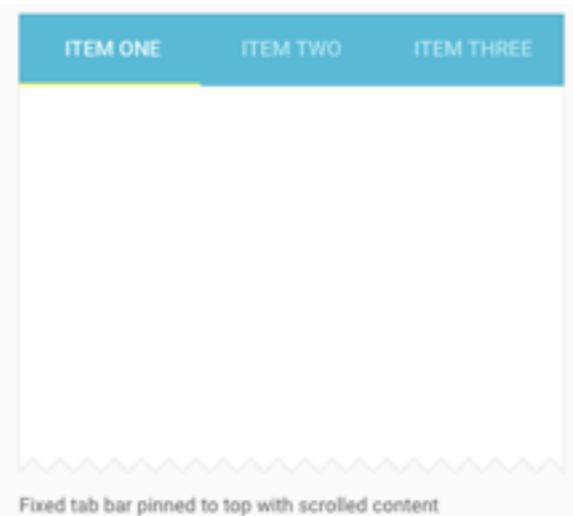
- Tabs



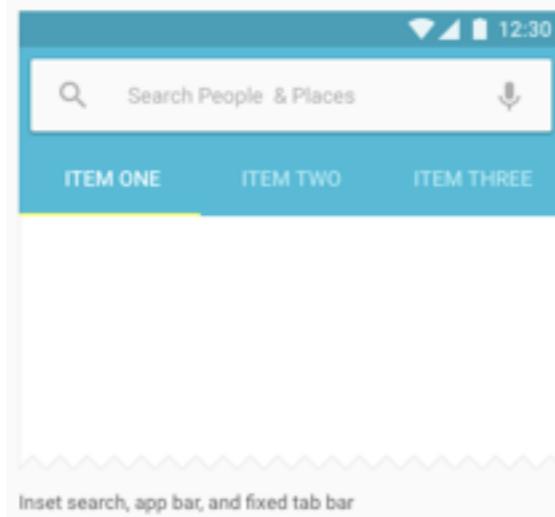
Default app bar and fixed tab bar



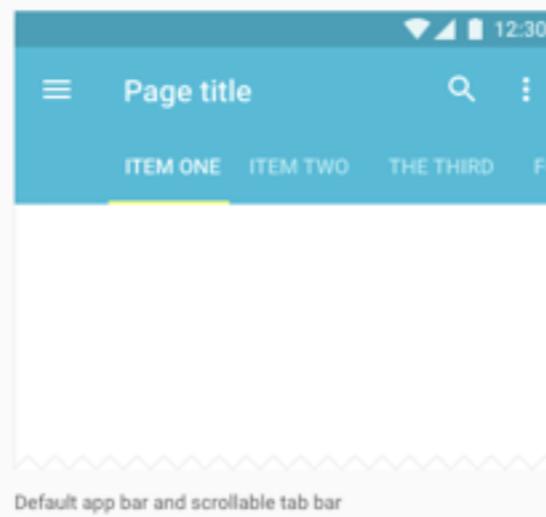
Extended app bar and fixed tab bar



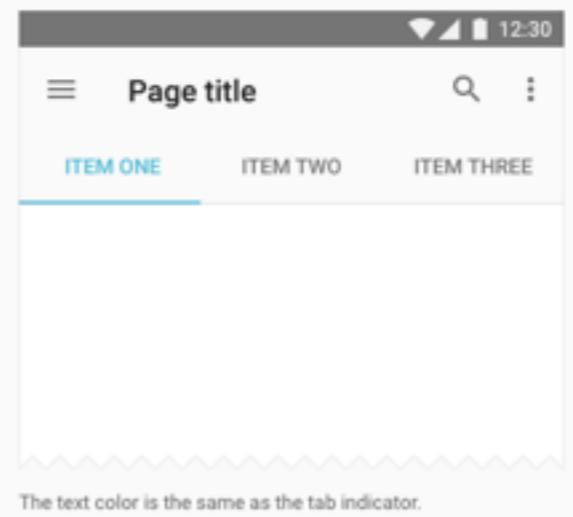
Fixed tab bar pinned to top with scrolled content



Inset search, app bar, and fixed tab bar



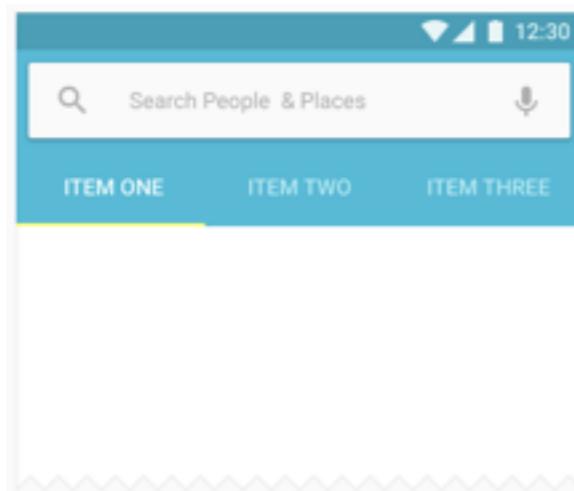
Default app bar and scrollable tab bar



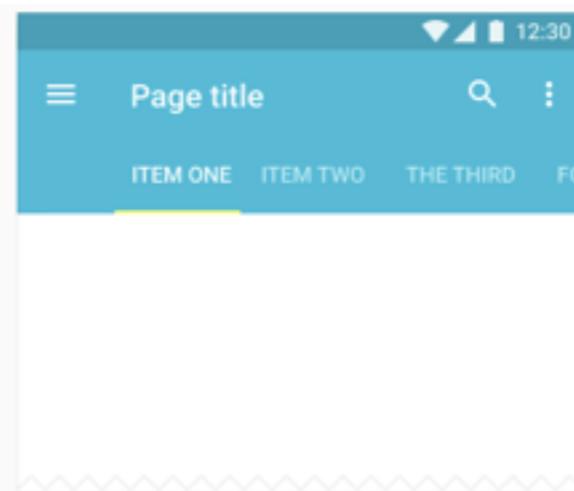
The text color is the same as the tab indicator.

Material UI: Componentes

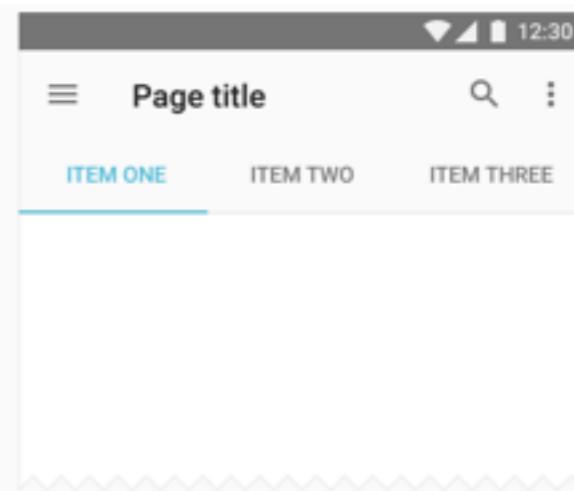
- Tabs



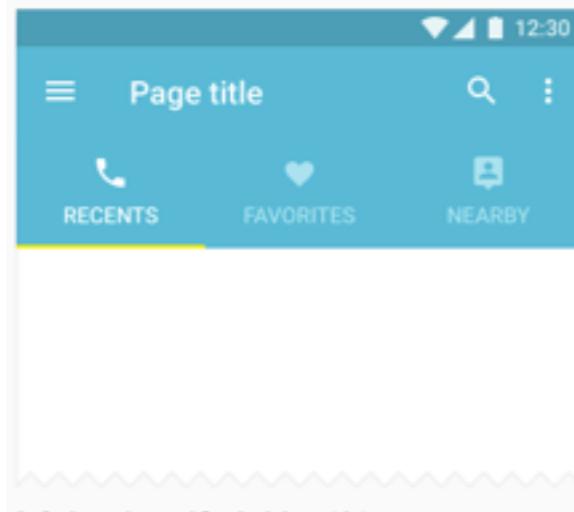
Inset search, app bar, and fixed tab bar



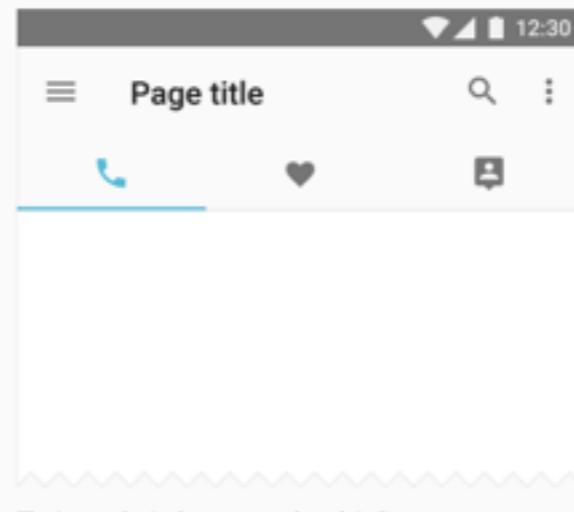
Default app bar and scrollable tab bar



The text color is the same as the tab indicator.



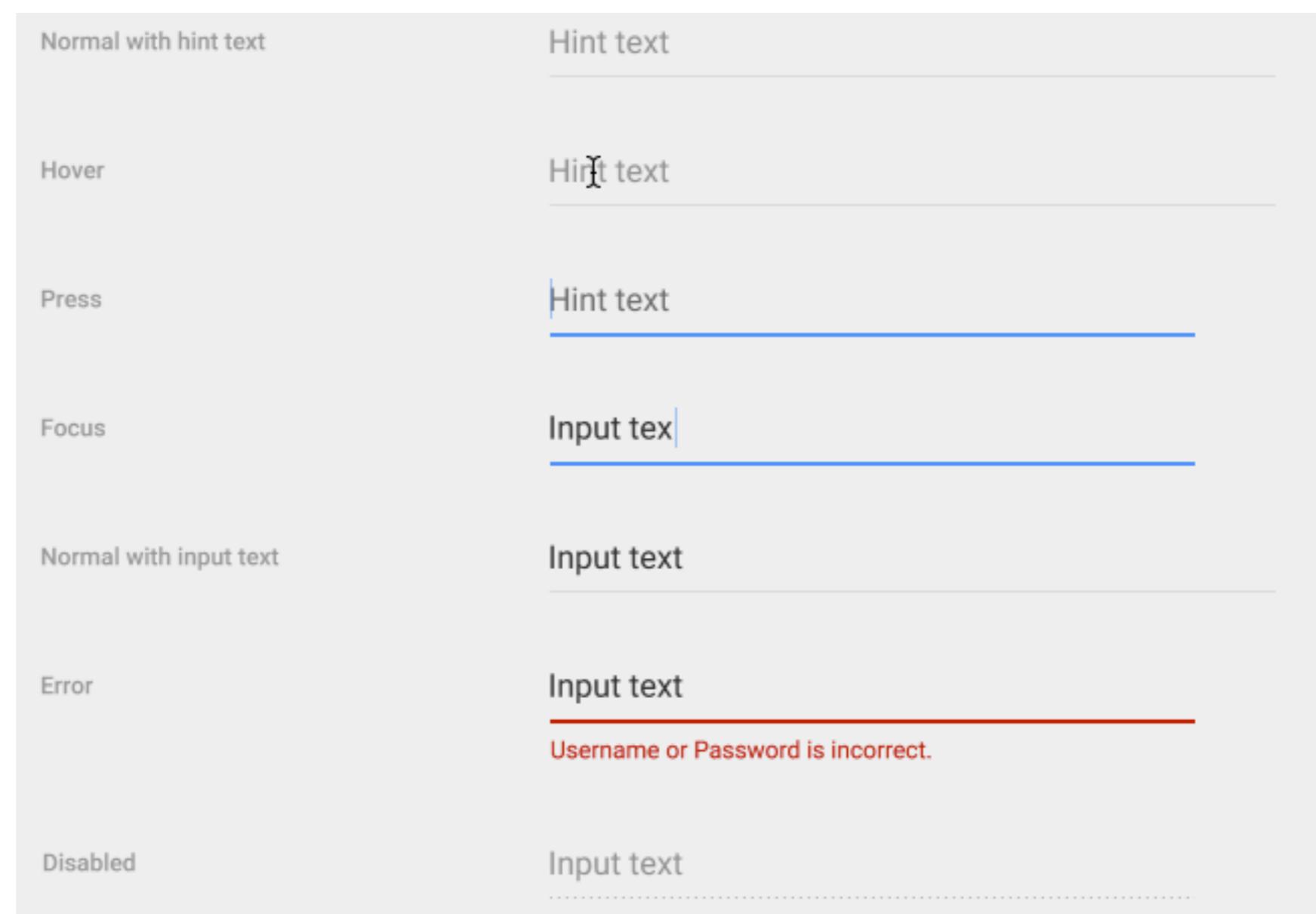
Default app bar and fixed tab bar with icons



The icon color is the same as the tab indicator.

Material UI: Componentes

- Text fields



Material UI: Navegación

- Up button

Devuelve al usuario a la pantalla anterior
de la jerarquía de pantallas.

- Back button

Devuelve al usuario a la pantalla anterior en
sucesión cronológica.

Material UI: Navegación

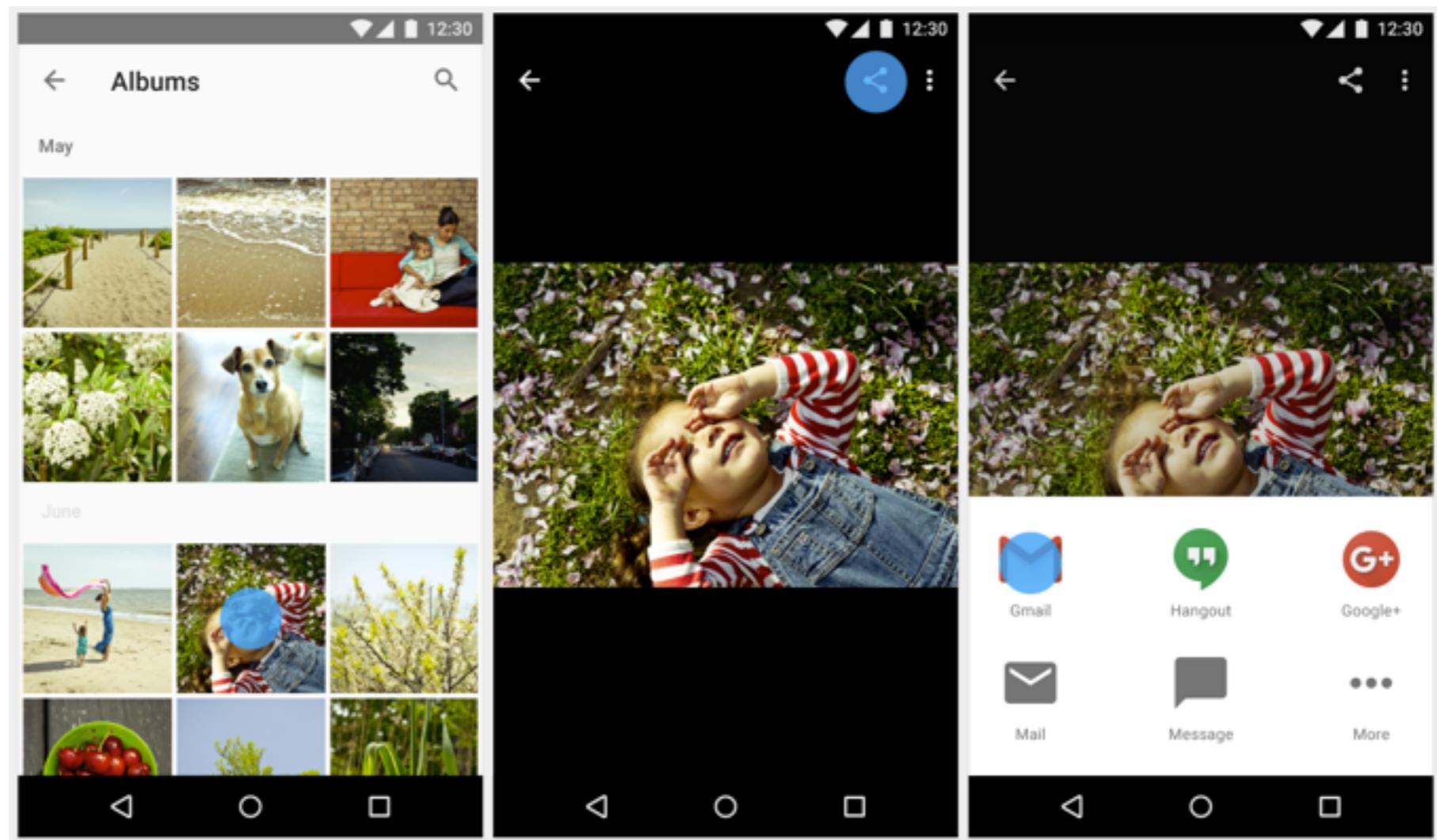
- Embebida



This is a mobile example of an app with embedded navigation.
Common tasks are performed in a strong primary view.

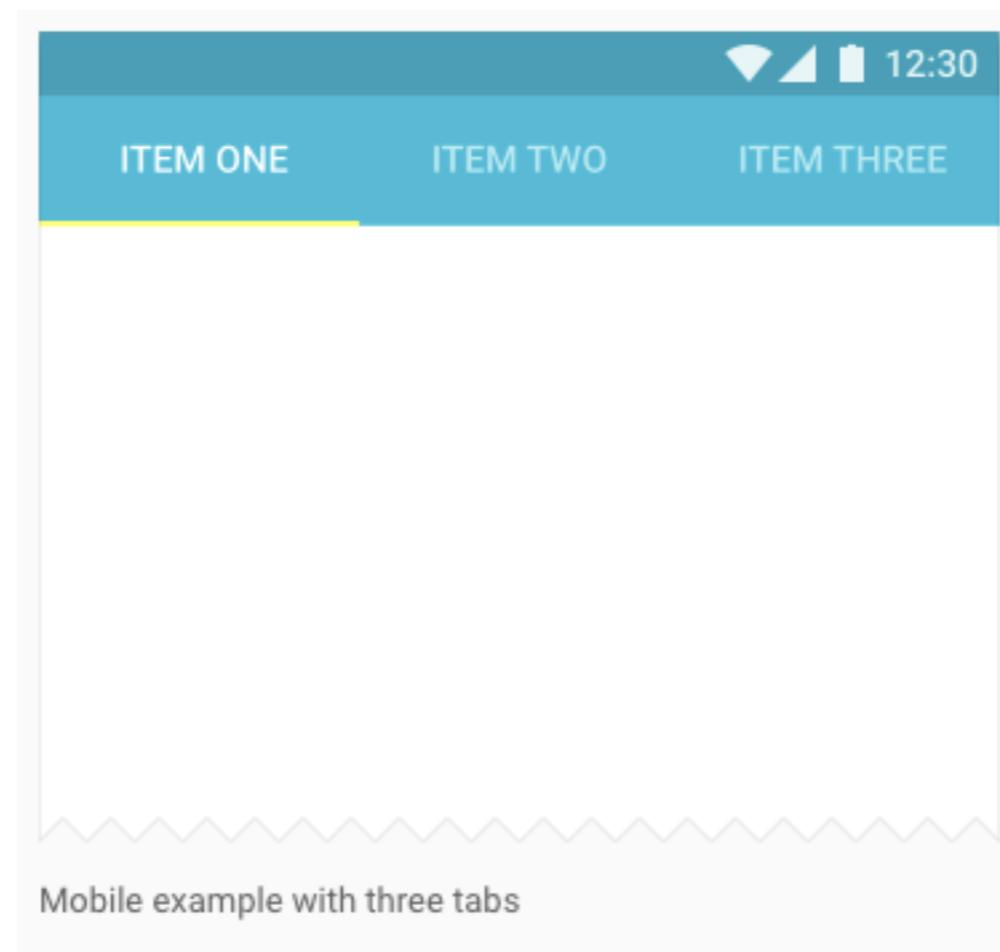
Material UI: Navegación

- Jerárquica



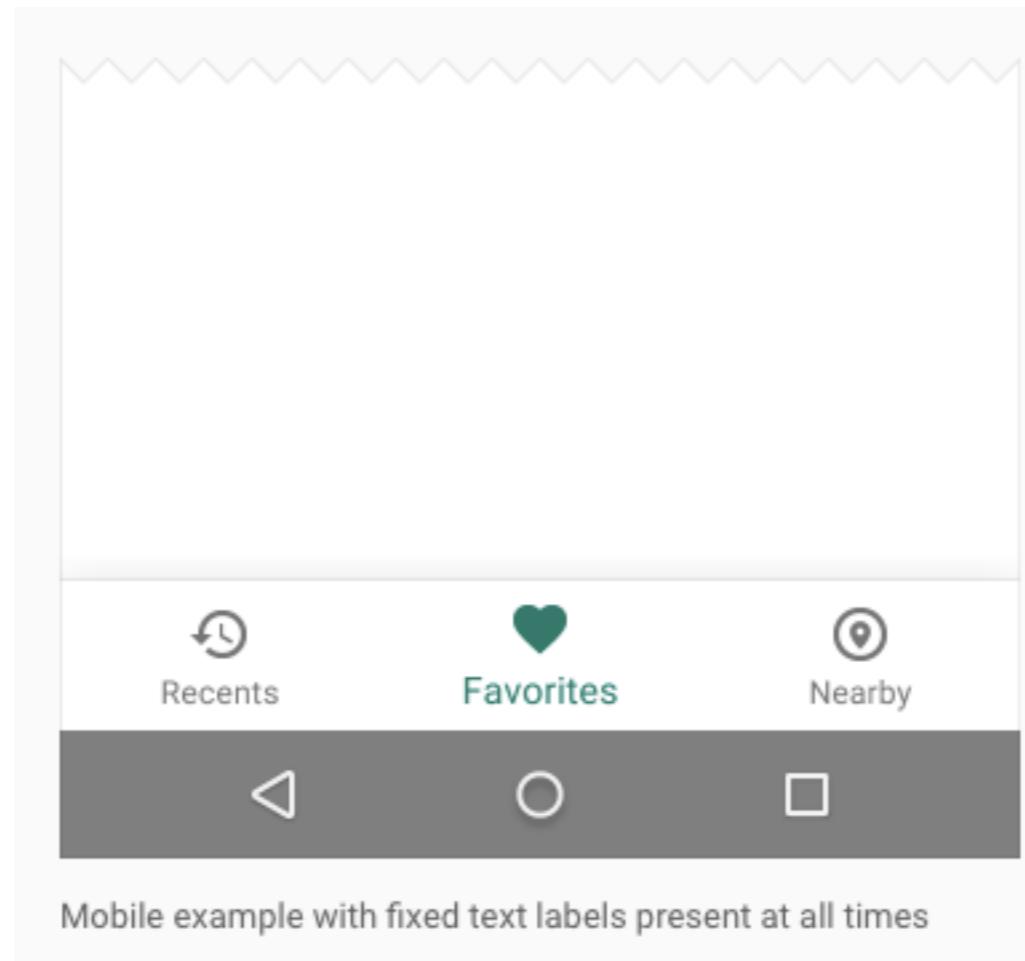
Material UI: Navegación

- Tabs



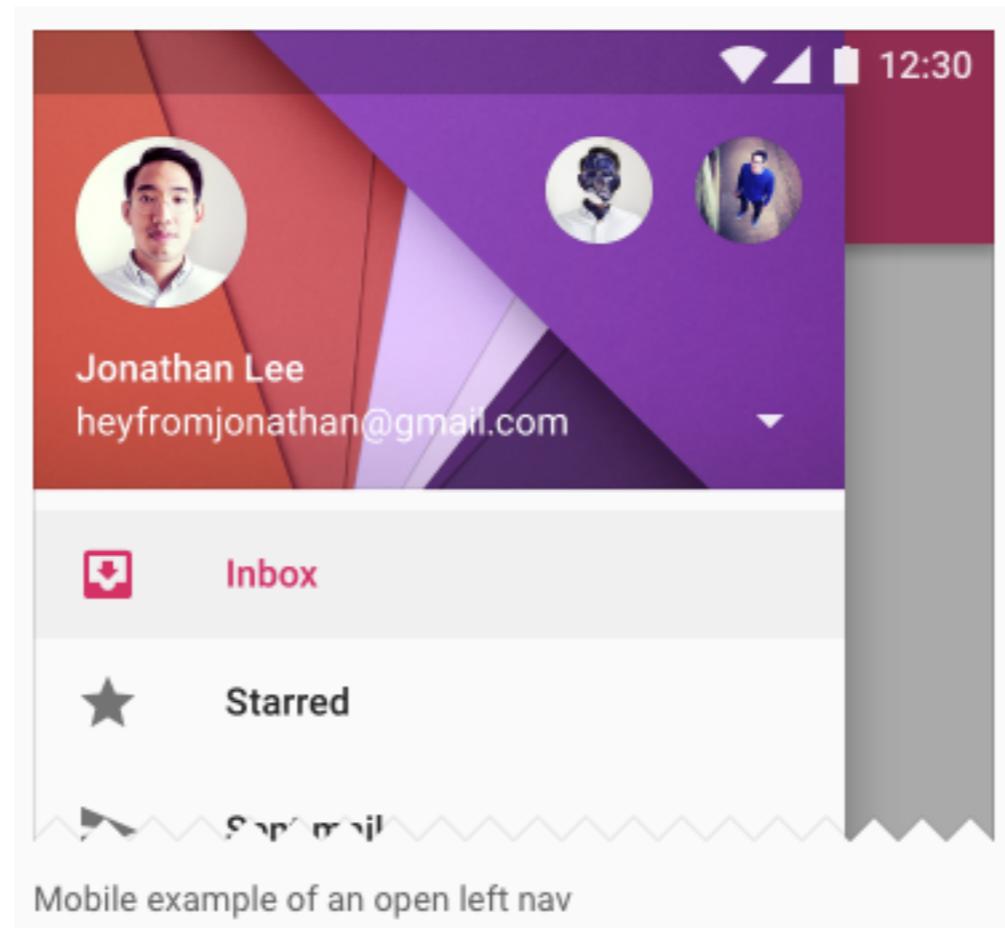
Material UI: Navegación

- Bottom Navigation bar

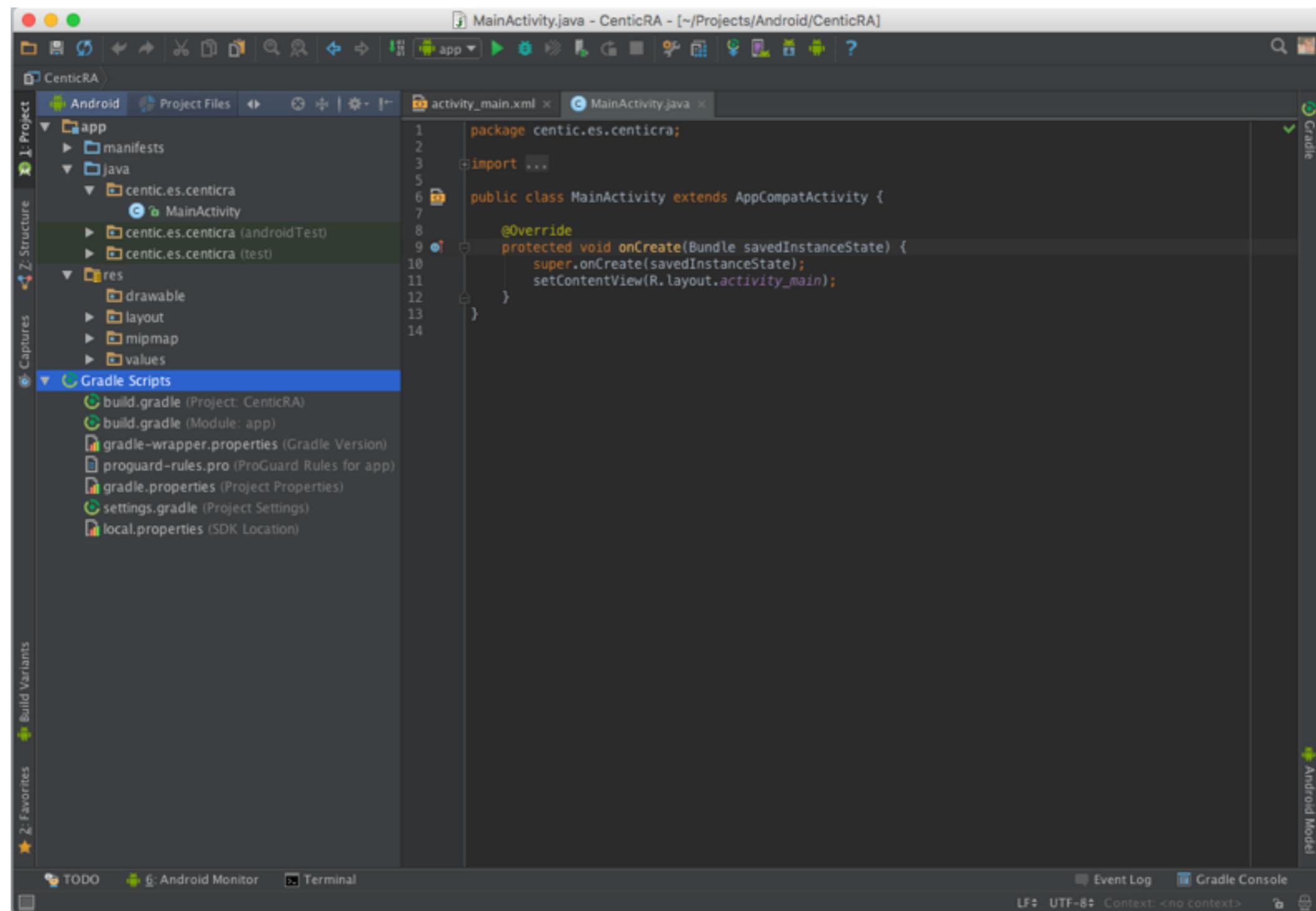


Material UI: Navegación

- Navigation Drawer



Android Studio



Android Studio

- IDE oficial para Android (No más eclipse)
- Integración con Gradle (No más ant)
- Integración con AVD (Android Virtual Device)
- Cambios instantáneos sin recompilar APK
- Plantillas para inicio rápido.
- Integración herramientas de test
- Herramientas de depuración, usabilidad, versionado, etc...
- Integración con NDK
- Soporte para Google Cloud Platform (Push, AppEngine, Firebase)

Android Studio: Proyecto

Tipos de Proyecto:

1. Android App Module
2. Library
3. Google App Engine Module

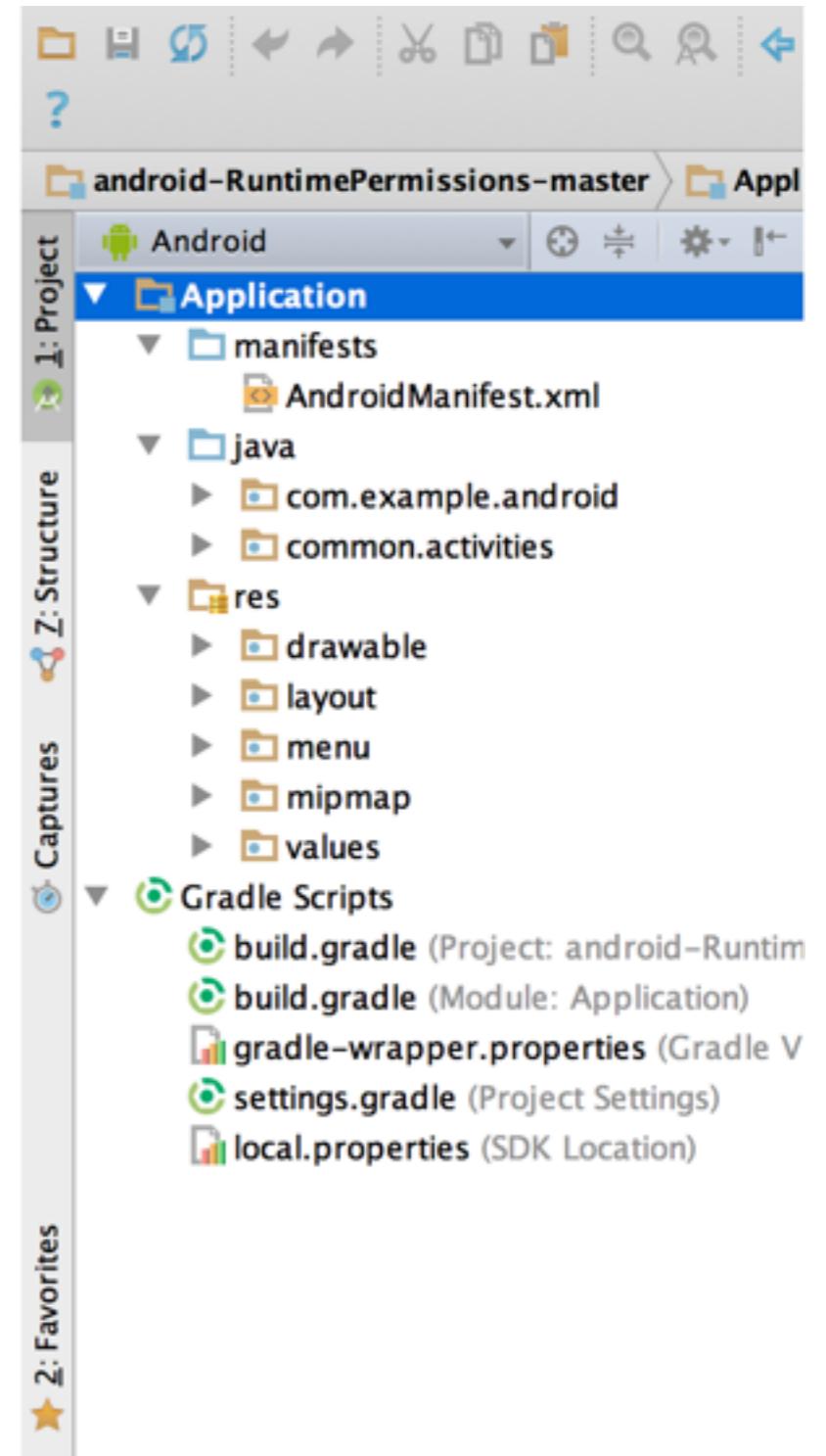
Android Studio: Android App

Application:

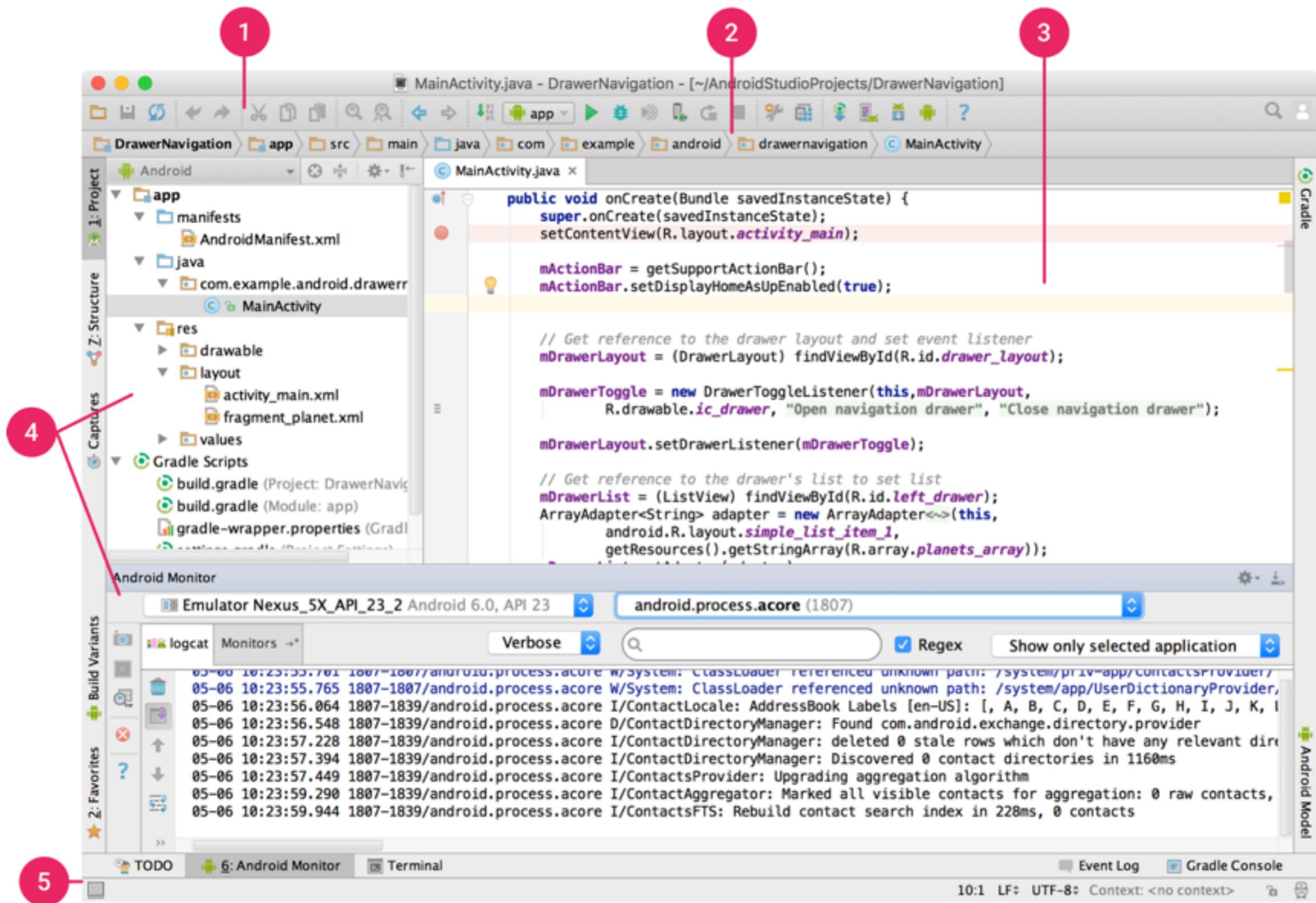
- manifest
- java
- res

Gradle Scripts:

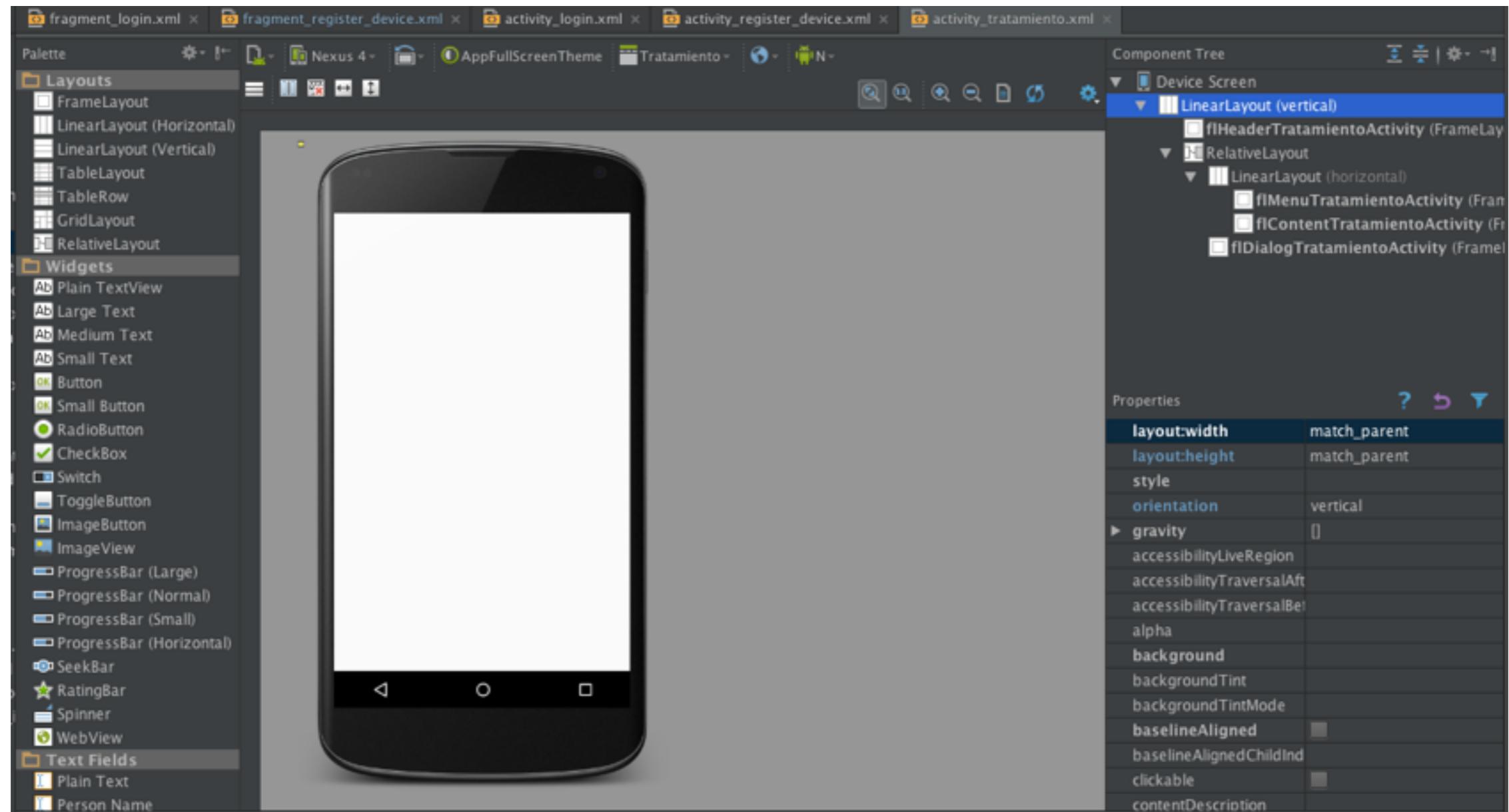
- scripts para construir la aplicación



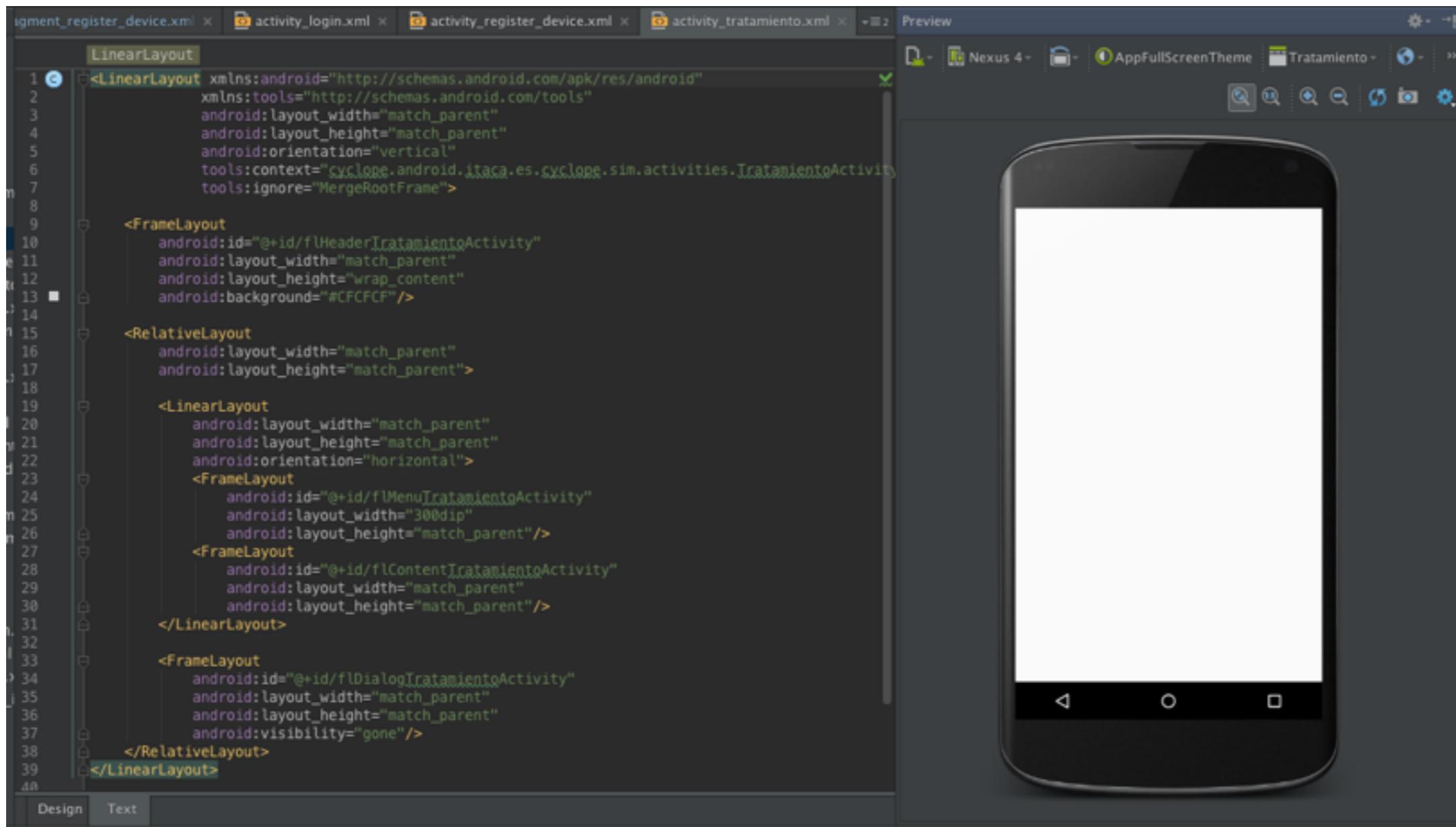
Android Studio: Android App



Android Studio: Android App



Android Studio: Android App



The screenshot shows the Android Studio interface with the XML layout editor open. The title bar displays several tabs: fragment_register_device.xml, activity_login.xml, activity_register_device.xml, activity_tratamiento.xml, Preview, and others. The Preview tab is active, showing a smartphone screen with a white background. The XML code on the left defines a complex layout structure:

```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:tools="http://schemas.android.com/tools"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical"
6     tools:context="cyclope.android.itaca.es.cyclope.sim.activities.TratamientoActivity"
7     tools:ignore="MergeRootFrame">
8
9     <FrameLayout
10        android:id="@+id/flHeaderTratamientoActivity"
11        android:layout_width="match_parent"
12        android:layout_height="wrap_content"
13        android:background="#CFCFCF"/>
14
15     <RelativeLayout
16         android:layout_width="match_parent"
17         android:layout_height="match_parent">
18
19         <LinearLayout
20             android:layout_width="match_parent"
21             android:layout_height="match_parent"
22             android:orientation="horizontal">
23             <FrameLayout
24                 android:id="@+id/flMenuTratamientoActivity"
25                 android:layout_width="300dp"
26                 android:layout_height="match_parent"/>
27             <FrameLayout
28                 android:id="@+id/flContentTratamientoActivity"
29                 android:layout_width="match_parent"
30                 android:layout_height="match_parent"/>
31         </LinearLayout>
32
33         <FrameLayout
34             android:id="@+id/flDialogTratamientoActivity"
35             android:layout_width="match_parent"
36             android:layout_height="match_parent"
37             android:visibility="gone"/>
38     </RelativeLayout>
39 </LinearLayout>
```

The XML code uses various Android XML layout tags like LinearLayout, FrameLayout, and RelativeLayout, along with attributes for layout parameters and context. The preview shows a blank white screen, indicating no content has been added yet.

Android Studio: Android App

Accesos rápidos:

Tool Window	Windows and Linux	Mac
Project	Alt+1	Command+1
Version Control	Alt+9	Command+9
Run	Shift+F10	Control+R
Debug	Shift+F9	Control+D
Android Monitor	Alt+6	Command+6
Return to Editor	Esc	Esc
Hide All Tool Windows	Control+Shift+F12	Command+Shift+F12

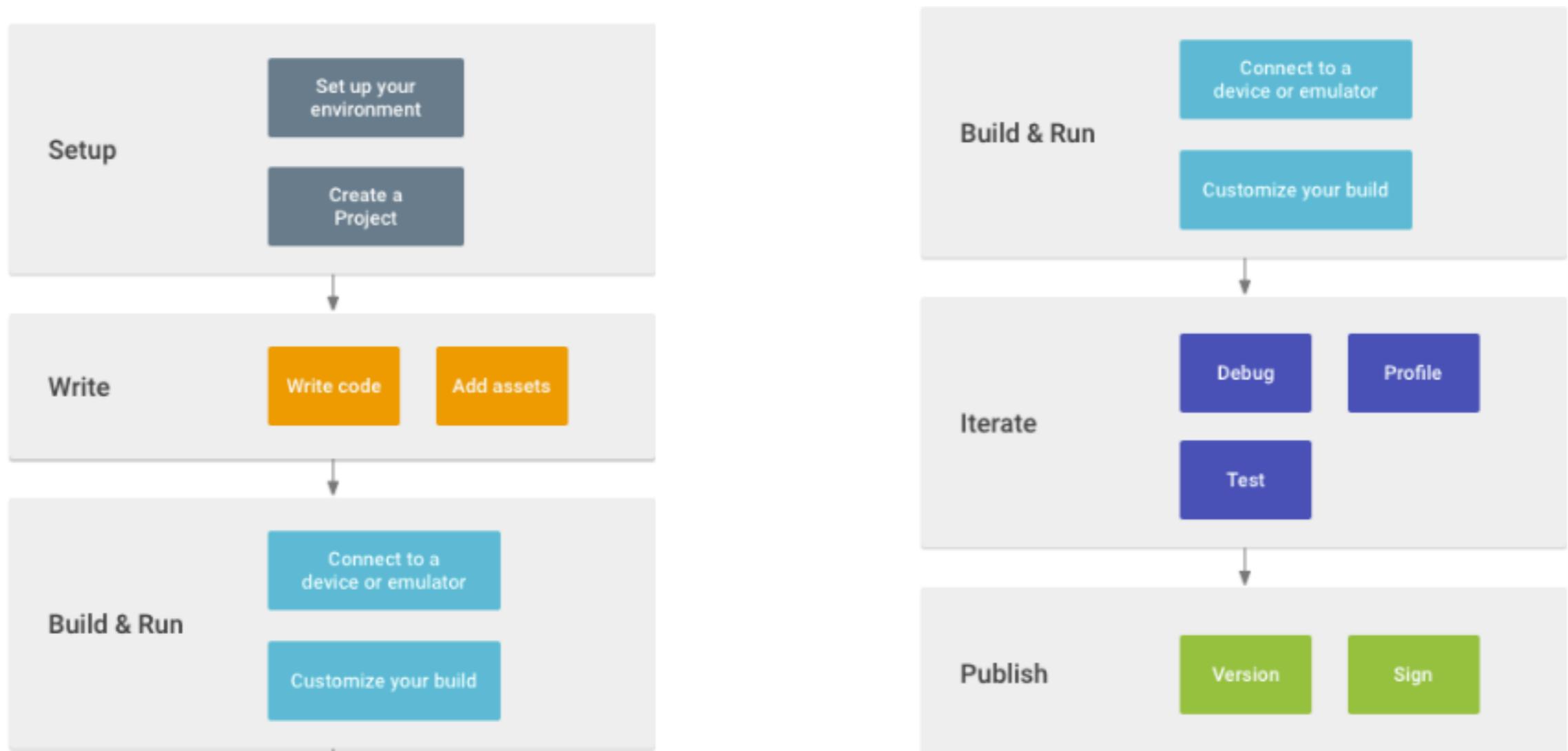
Type	Description	Windows and Linux	Mac
Basic Completion	Displays basic suggestions for variables, types, methods, expressions, and so on. If you call basic completion twice in a row, you see more results, including private members and non-imported static members.	Control+Space	Control+Space
Smart Completion	Displays relevant options based on the context. Smart completion is aware of the expected type and data flows. If you call Smart Completion twice in a row, you see more results, including chains.	Control+Shift+Space	Control+Shift+Space
Statement Completion	Completes the current statement for you, adding missing parentheses, brackets, braces, formatting, etc.	Control+Shift+Enter	Shift+Command+Enter

Android Studio: Android App

Gradle:

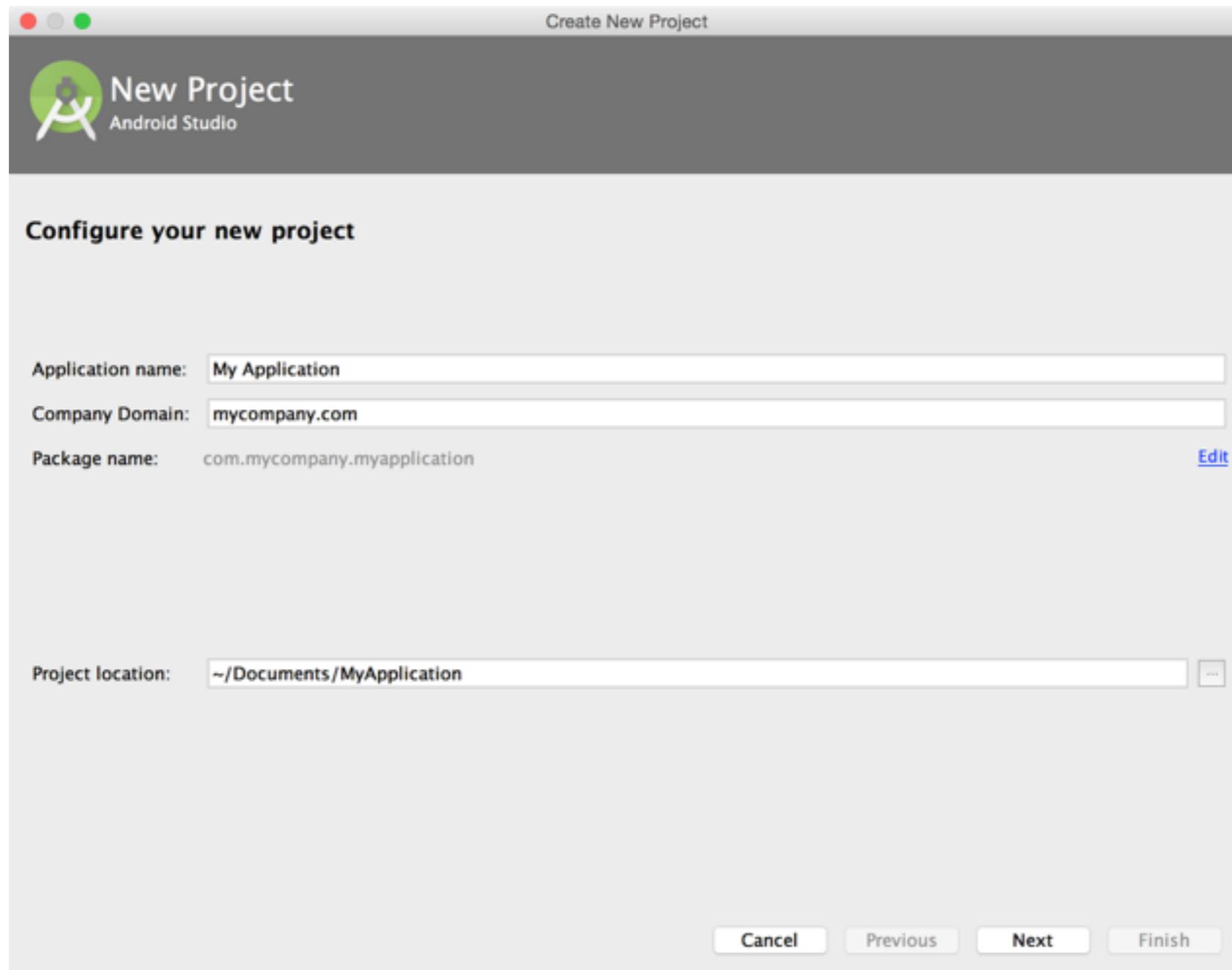
- Personalizar, configurar y ampliar el proceso de '**build**'
- Crear multiples APK's usando el mismo código base.
- Reutilizar código entre proyectos

Android Studio: Workflow



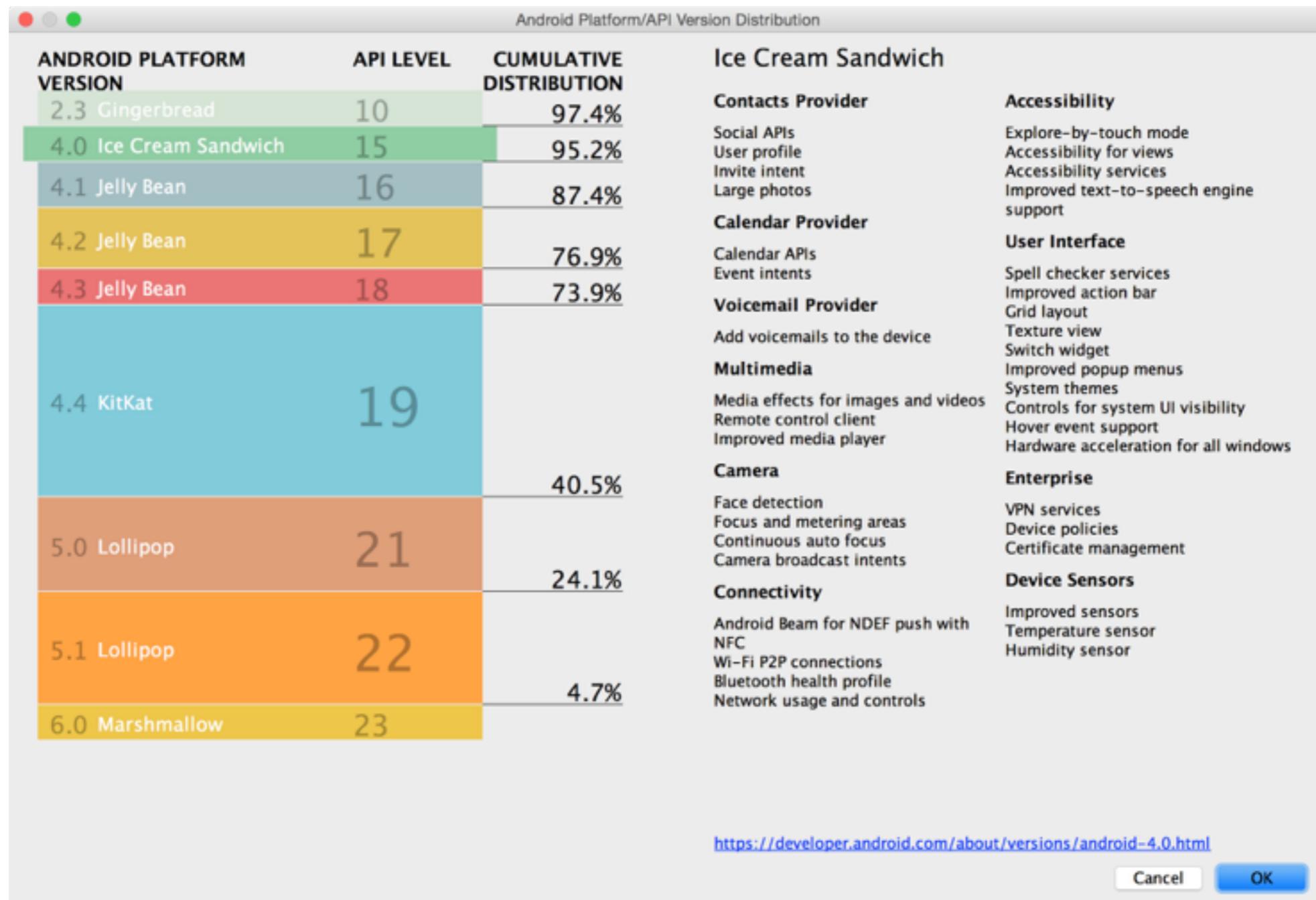
Android Studio: Crear Proyecto

Configurar el proyecto



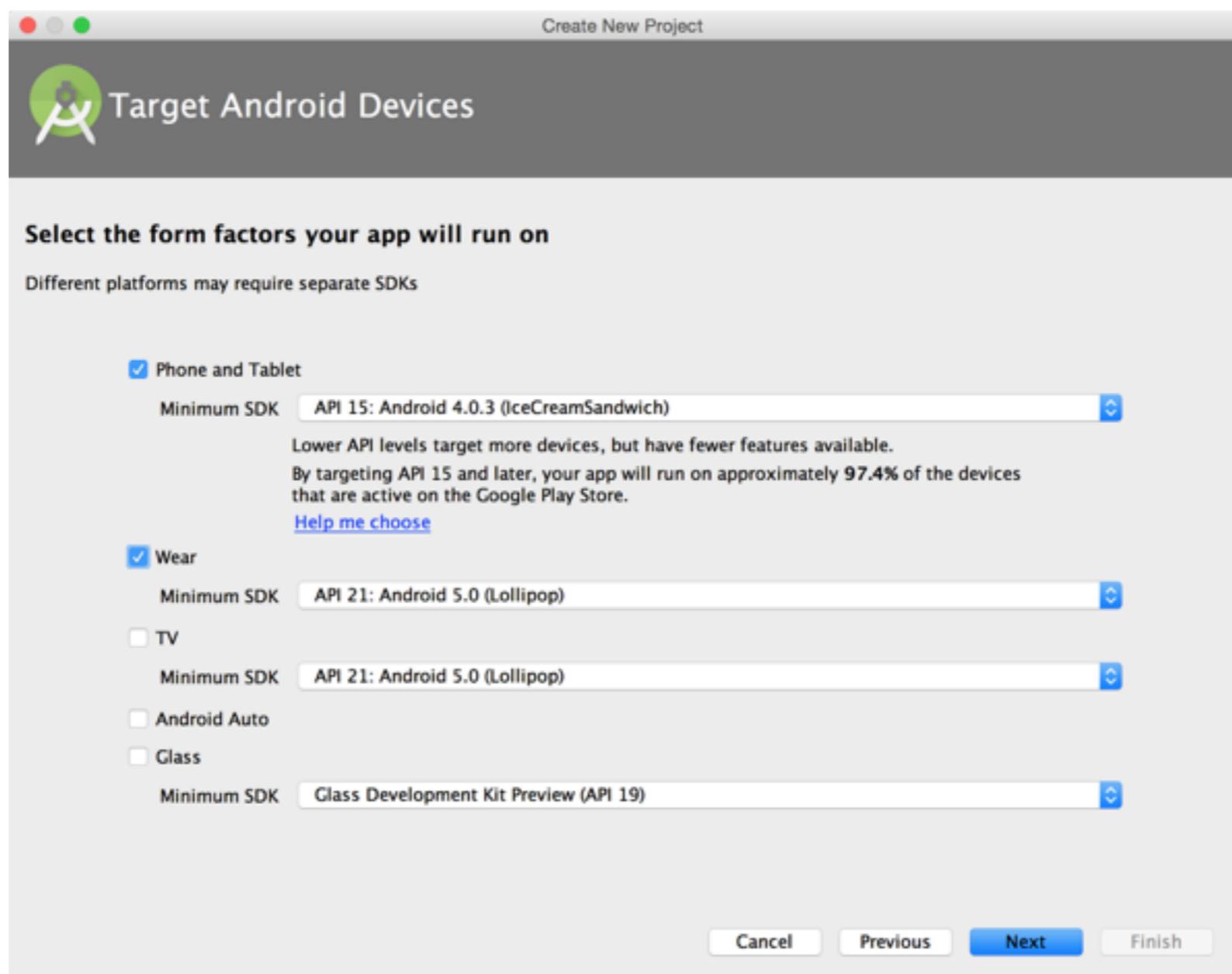
Android Studio: Crear Proyecto

Seleccionar API



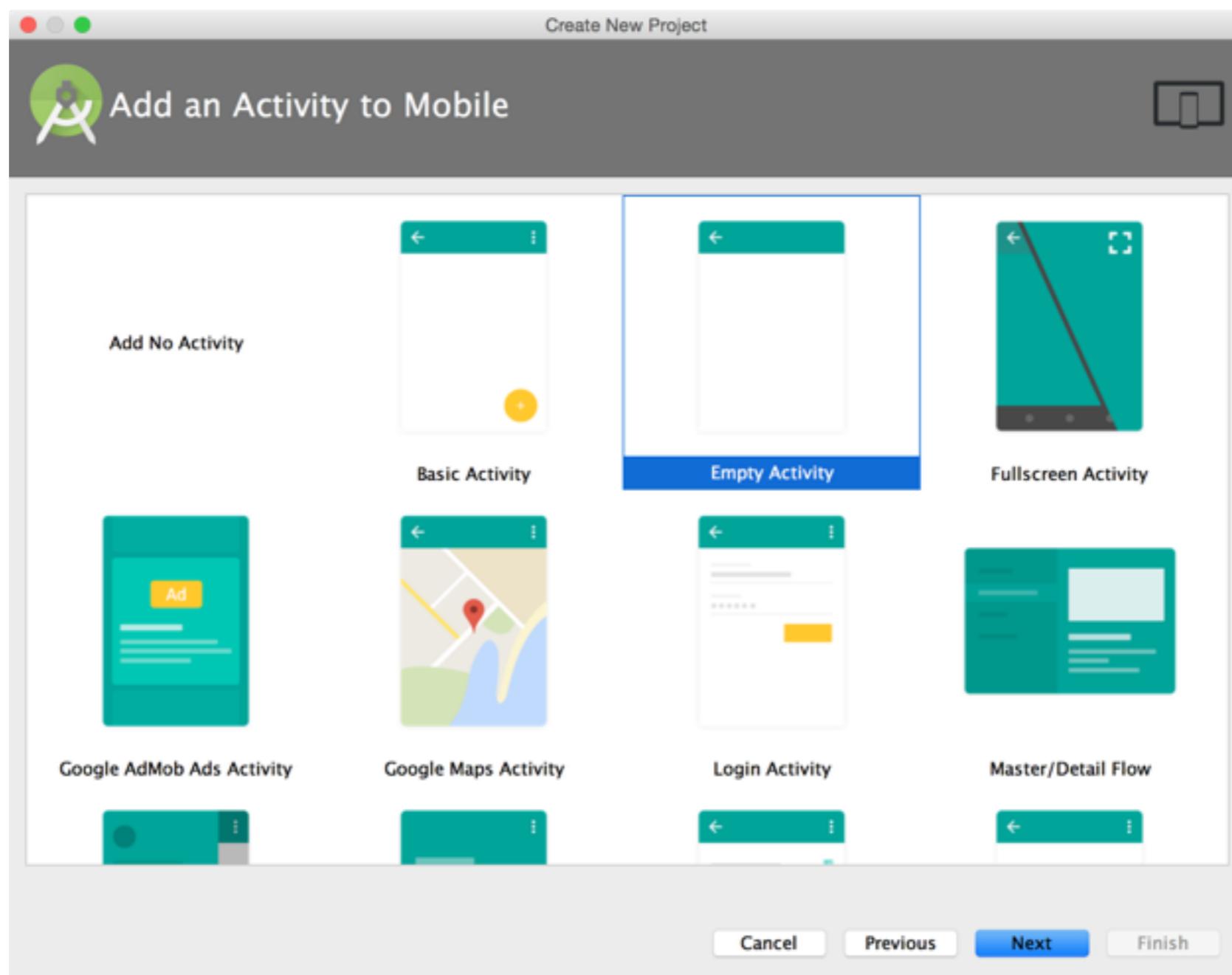
Android Studio: Crear Proyecto

Seleccionar API



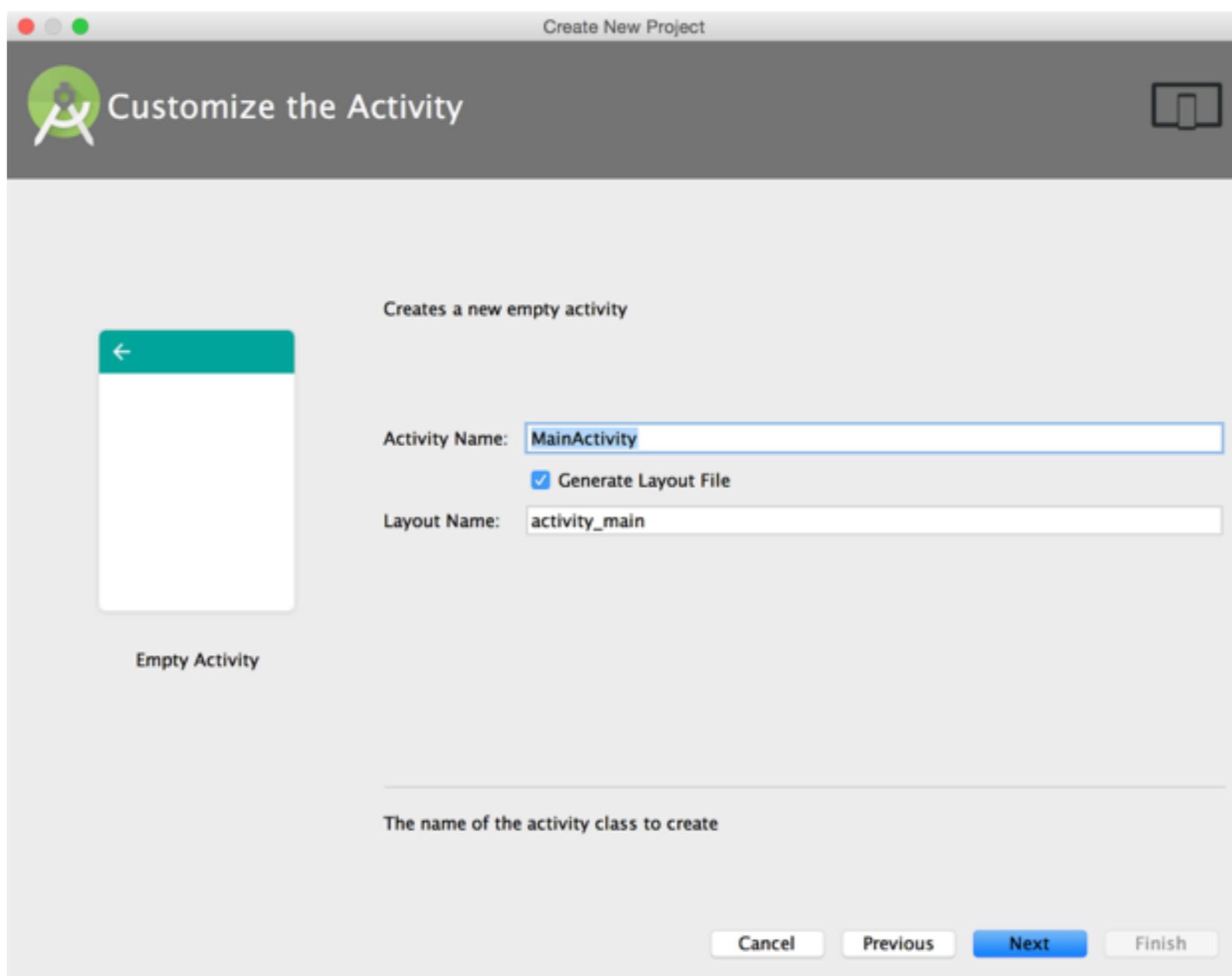
Android Studio: Crear Proyecto

Activity inicial



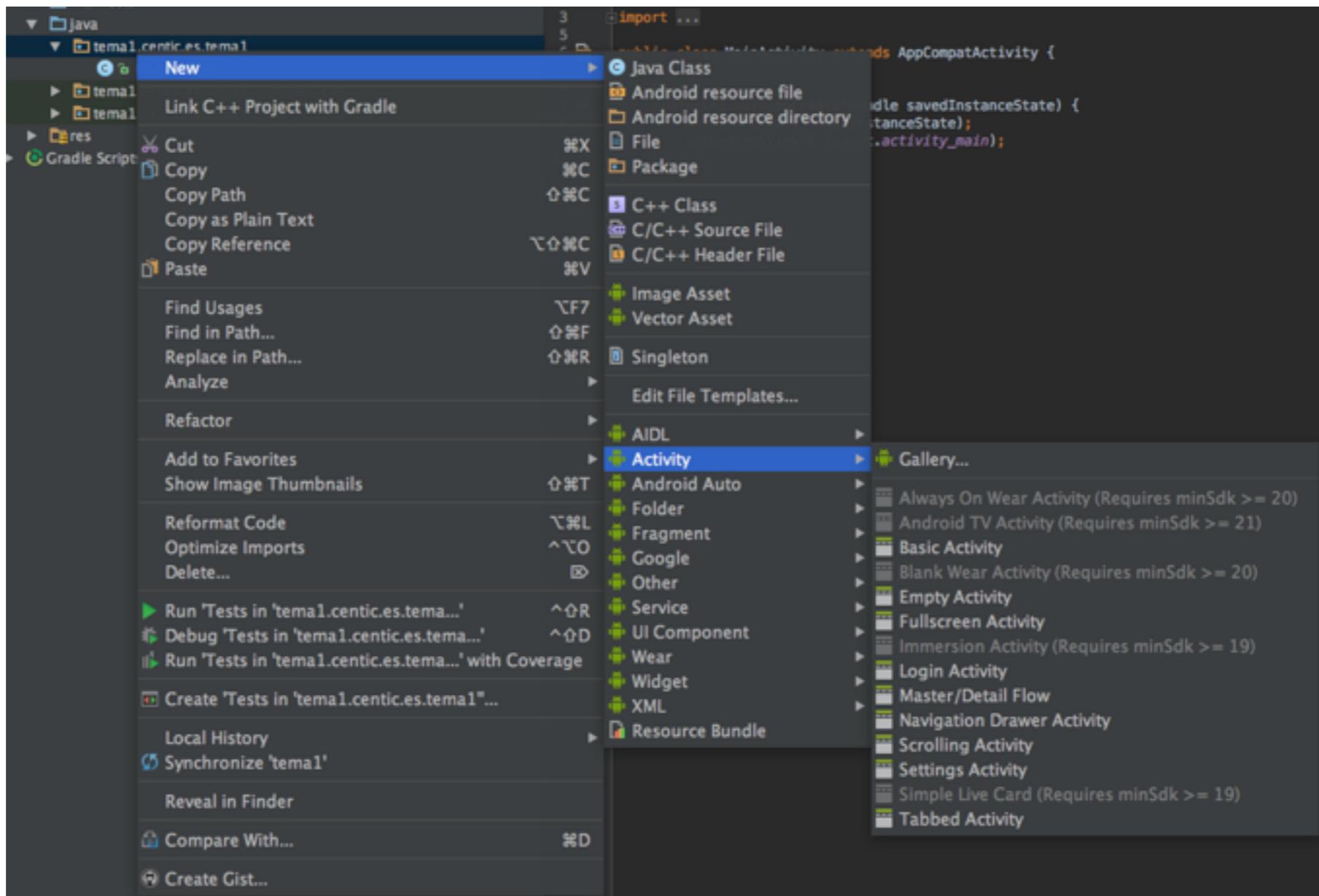
Android Studio: Crear Proyecto

Configurar la Activity



Android Studio: Codificar

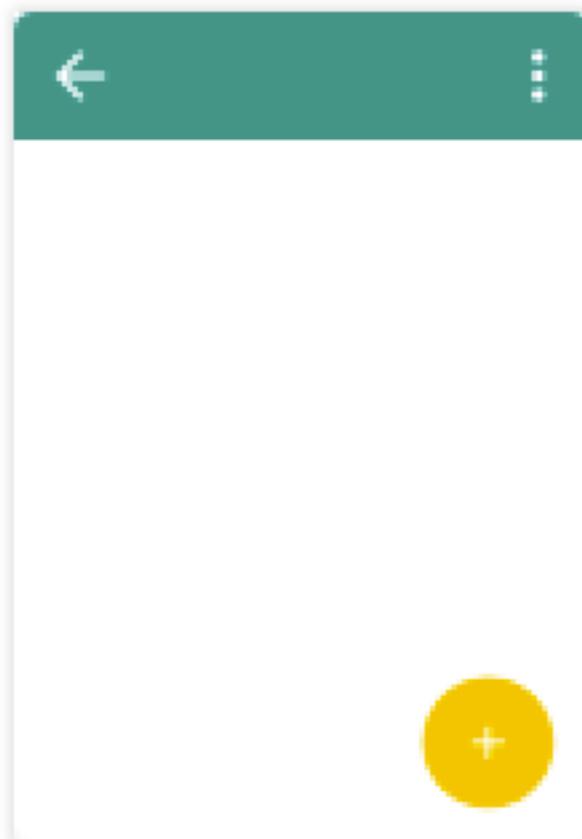
Añadir componentes



Android Studio: Codificar

Añadir componentes

Basic Activity



This template creates a simple app with an app bar and a floating action button. It acts as a starting point for your project by providing commonly used UI components.

This template includes:

- [AppBar](#)
- [FloatingActionButton](#)
- Two layout files: one for the activity and one to separate out text content

Android Studio: Codificar

Añadir componentes

Empty Activity



This template creates an empty activity and a single layout file with sample text content. It allows you to start from scratch when building your app module or activity.

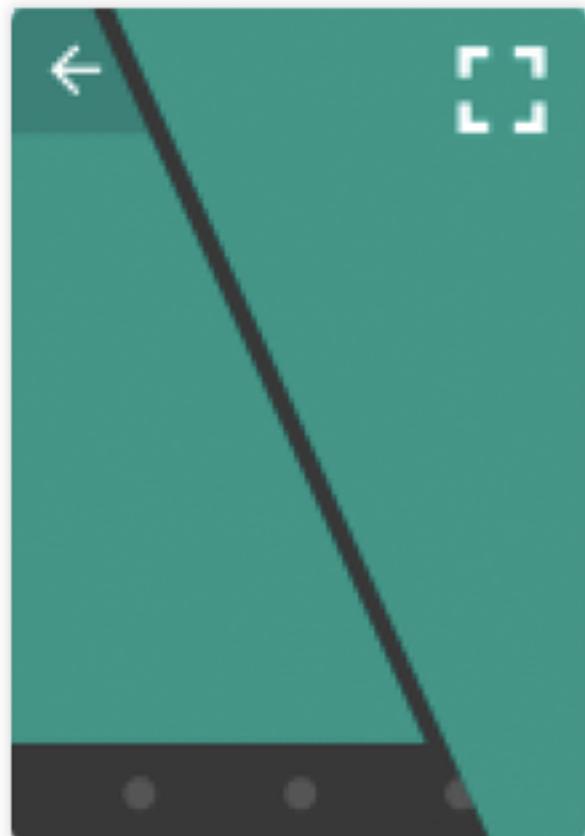
This template includes:

- Single layout file with text content

Android Studio: Codificar

Añadir componentes

Fullscreen Activity



This template creates an app that alternates between a primary fullscreen view and a view with standard user interface (UI) controls. The fullscreen view is the default and a user can activate the standard view by touching the device screen.

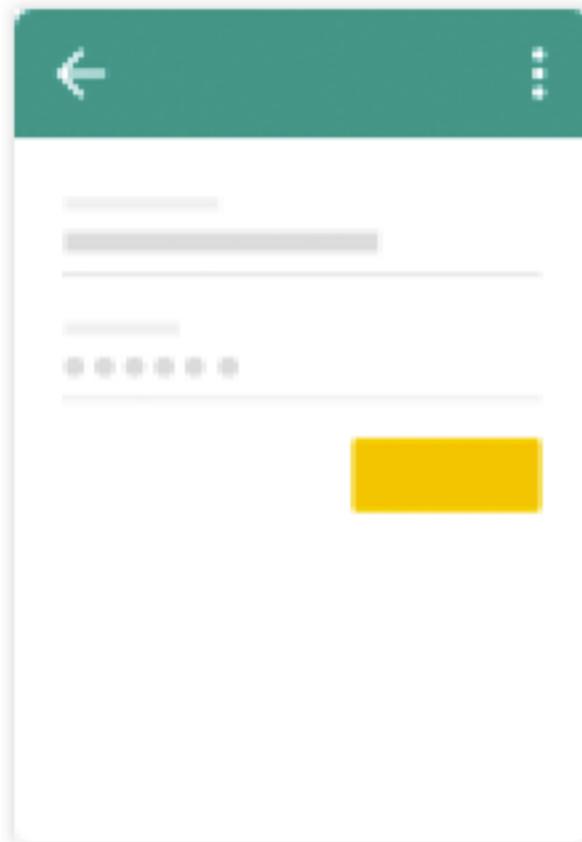
This template includes:

- Touch listener implementation for hiding the standard view elements
- Button that appears in the standard view but does not do anything
- [AppBar](#) for the standard view
- Single layout file with both the fullscreen view and a frame layout for standard view elements

Android Studio: Codificar

Añadir componentes

Login Activity



This template creates a standard login screen. The user interface includes email and password fields and a sign-in button. It is more commonly used as an activity template than as an app module template.

This template includes:

- [AsyncTask](#) implementation for handling network operations separately from the main user interface thread
- Progress indicator during network operations
- Single layout file with the recommended login UI:
 - Email and password input fields
 - Sign-in button

Android Studio: Codificar

Añadir componentes

Master/Detail Flow



This template creates an app that has both an item list display and a display for the details of an individual item. Clicking on an item on the list screen opens a screen with the item's details. The layout of the two displays depends on the device that is running the app.

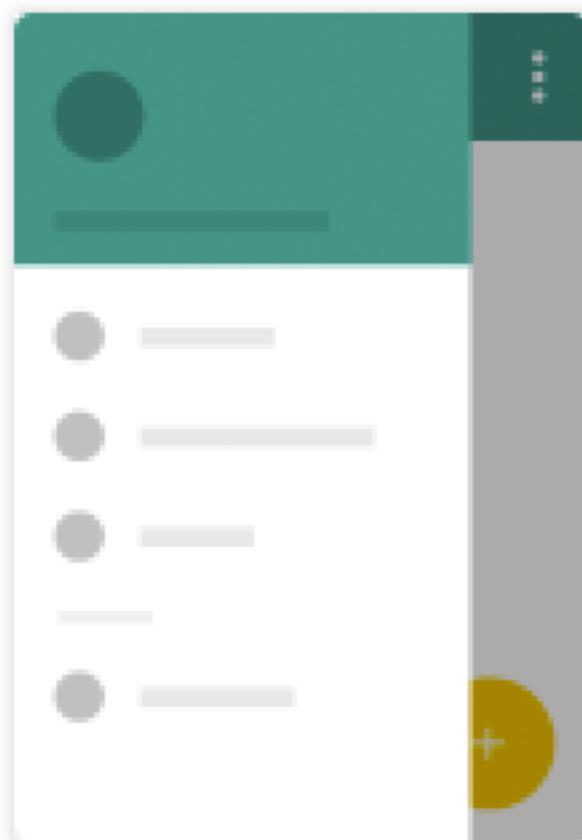
This template includes:

- Activity representing the list of items
- Activity and fragment options for displaying an individual item's details
- [FloatingActionButton](#) on each screen
- [Collapsing toolbar](#) for the item detail screen
- [Alternative resource](#) layout files for different device configurations

Android Studio: Codificar

Añadir componentes

Navigation Drawer Activity



This template creates a **Basic Activity** with a navigation drawer menu. The navigation bar expands from the left or right side of your app and appears in addition to the regular app bar.

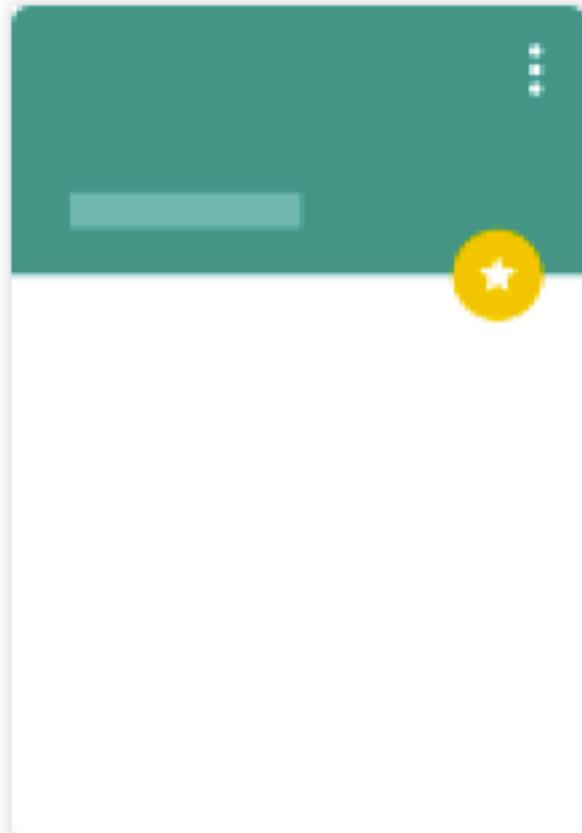
This template includes:

- Navigation drawer implementation with a [DrawerLayout](#), corresponding event handlers, and example menu options
- [AppBar](#)
- [FloatingActionButton](#)
- Layout files for the navigation drawer and the navigation drawer header, in addition to those from the **Basic Activity** template

Android Studio: Codificar

Añadir componentes

Scrolling Activity



This template creates an app with a collapsing toolbar and a scrolling view for long text content. As you scroll down the page, the toolbar, which can serve as a header, automatically condenses, and the floating action button disappears.

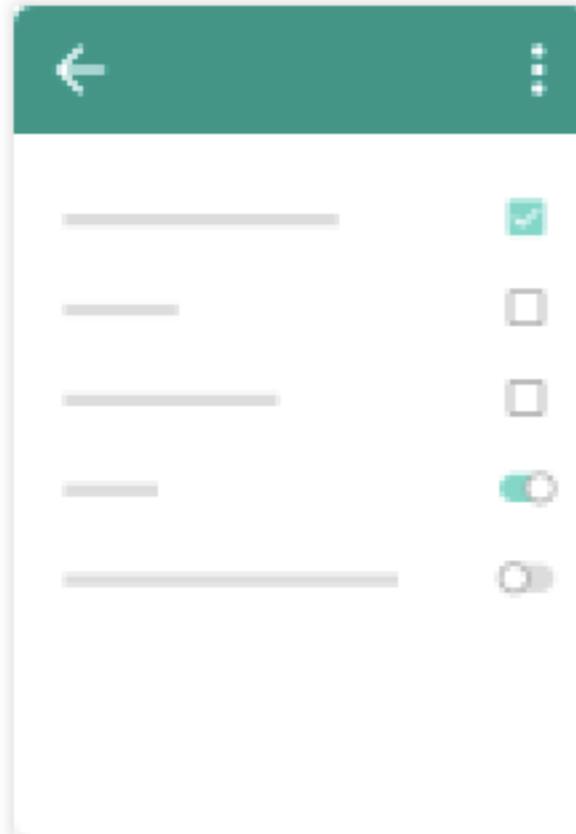
This template includes:

- Collapsing toolbar in place of the regular [AppBar](#)
- [FloatingActionButton](#)
- Two layout files: one for the activity and one to separate out the text content into a [NestedScrollView](#)

Android Studio: Codificar

Añadir componentes

Settings Activity



This template creates an activity that displays [user preferences or settings](#) for an app. It extends the [PreferenceActivity](#) class and is more commonly used as an activity template than as an app module template.

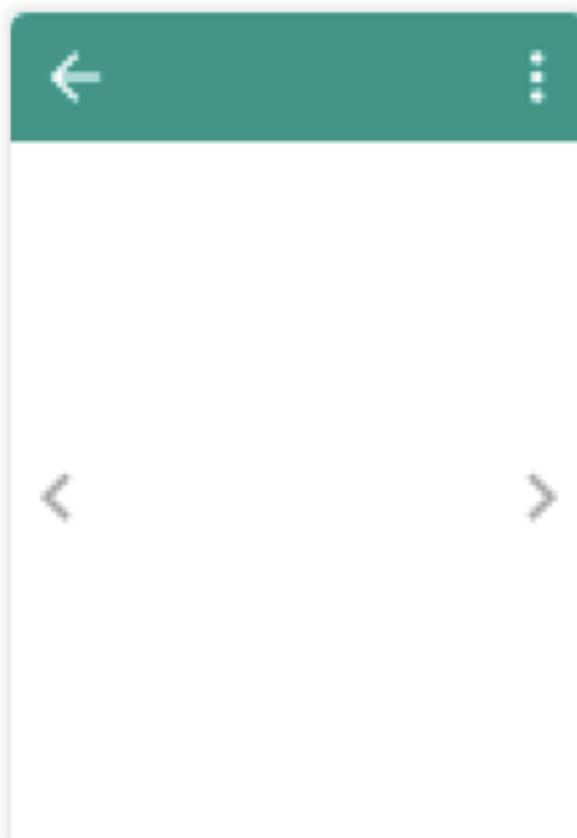
This template includes:

- Activity that extends [PreferenceActivity](#)
- XML files (in the `res/xml/` directory of your project) to define the displayed settings

Android Studio: Codificar

Añadir componentes

Tabbed Activity



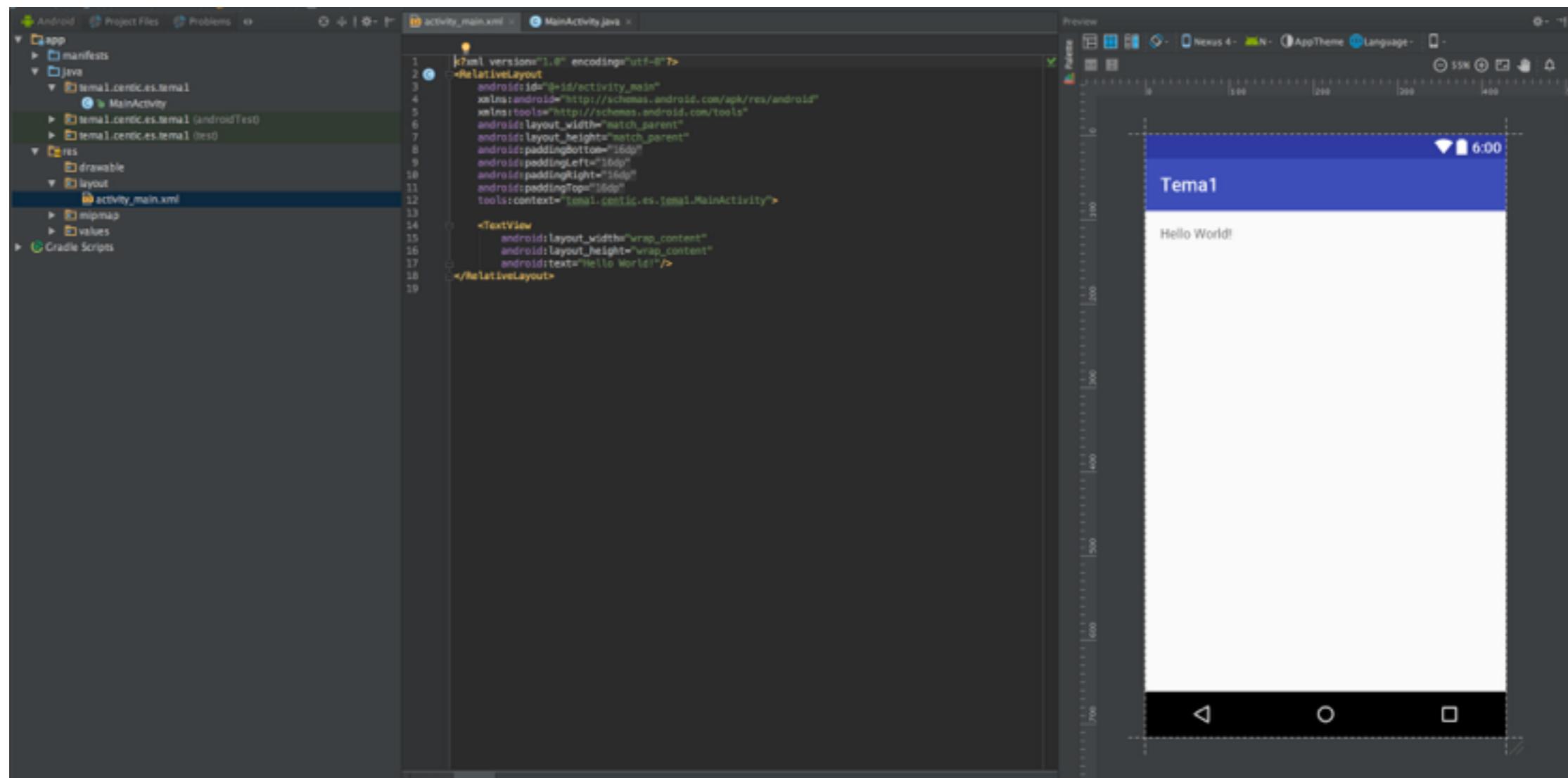
This template creates an app with multiple sections, [swiping navigation](#), and an app bar. The sections are defined as fragments between which you can swipe left and right to navigate.

This template includes:

- [AppBar](#)
- Adapter that extends [FragmentPagerAdapter](#) and creates a fragment for each section
- [ViewPager](#) instance, a layout manager for swiping between sections
- Two layout files: one for the activity and one for individual fragments

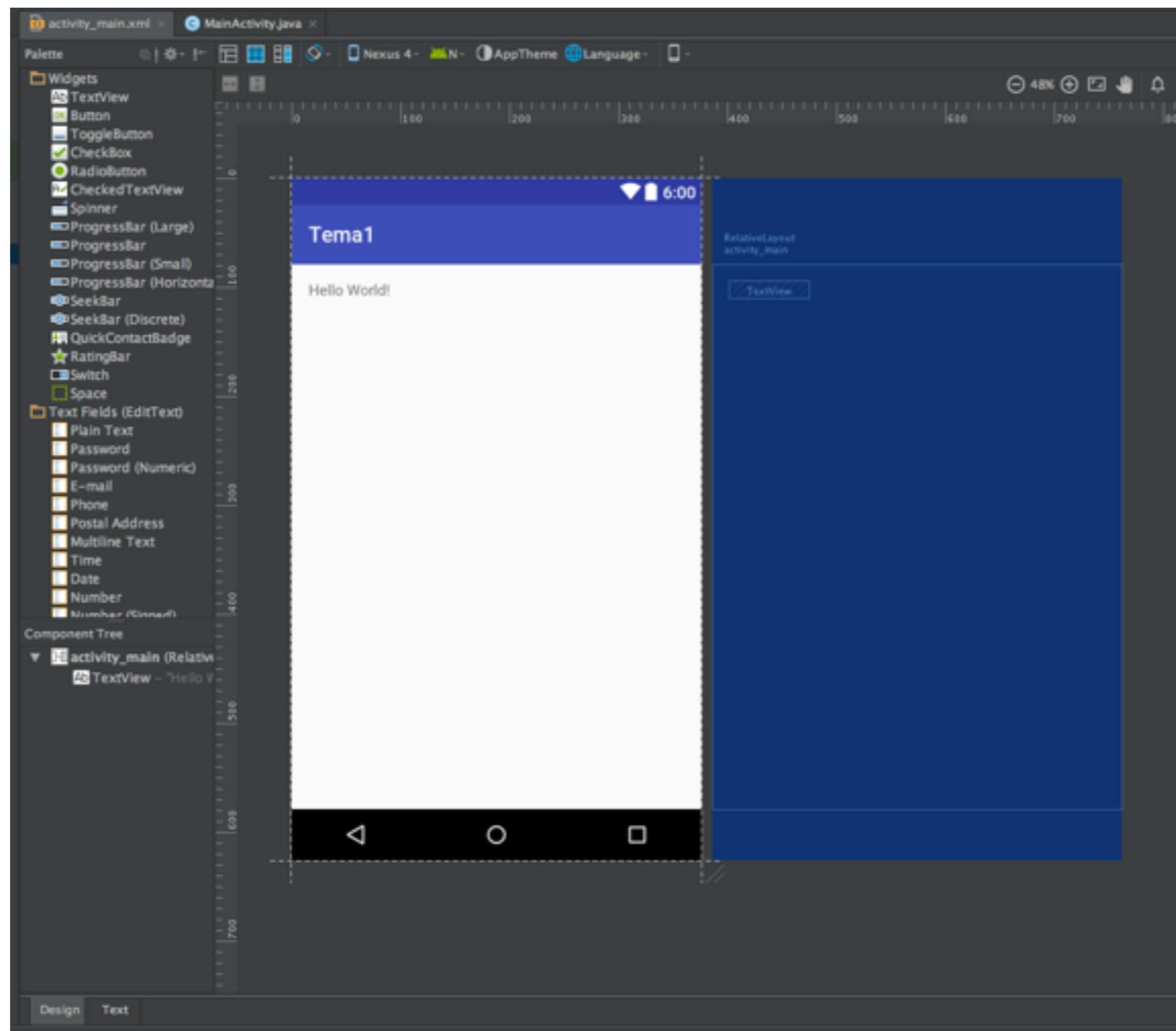
Android Studio: Codificar

Crear la UI

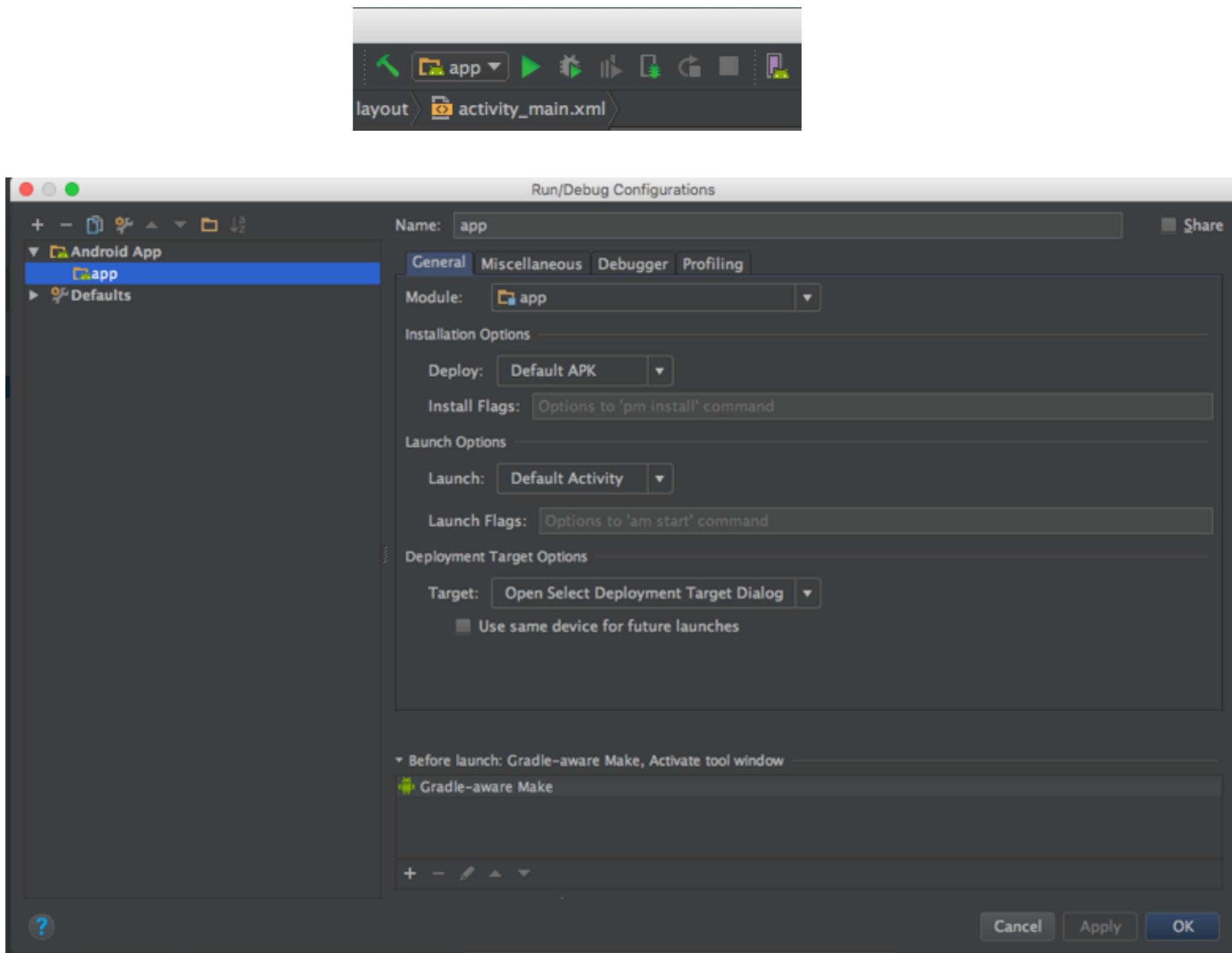


Android Studio: Codificar

Crear la UI



Android Studio: Ejecutar



Android Studio: Ejecutar

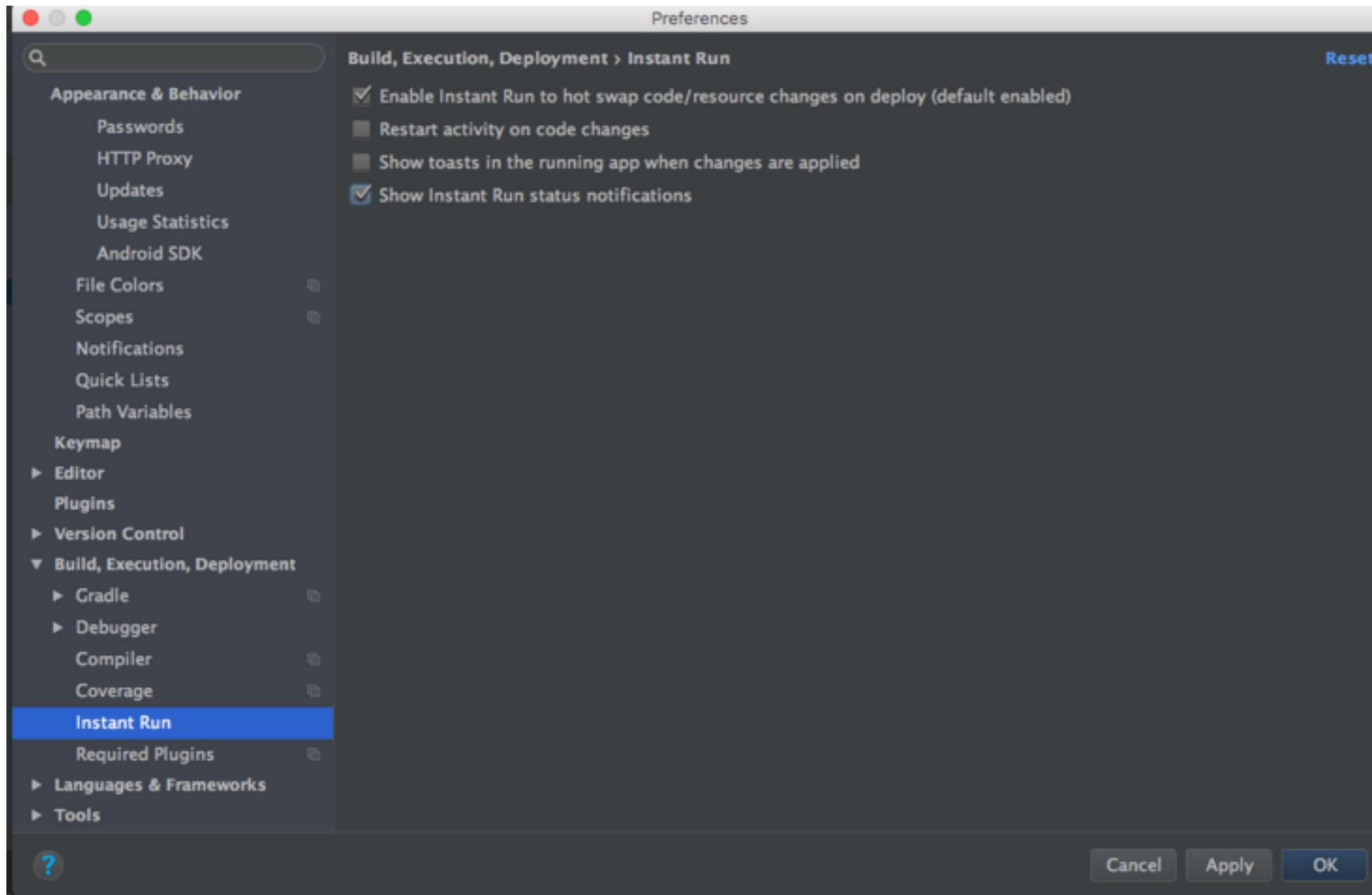
Monitor de procesos

The screenshot shows the Android Studio interface with two main windows visible:

- Gradle Build (Messages tab):** Displays the build log. It starts with "Gradle tasks [:app:generateDebugSources, :app:generateDebugAndroidTestSources, :app:mockableAndroidJar, :app:prepareDebugUnitTestDependencies]". It then shows a "BUILD SUCCESSFUL" message with details: "Total time: 7.489 secs", "0 errors", "0 warnings", and a link to "See complete output in console".
- Gradle Console:** Displays the command-line output of the build process. The tasks listed are:
 - :app:prepareComAndroidSupportTestRunner05Library
 - :app:prepareDebugAndroidTestDependencies
 - :app:compileDebugAndroidTestAidl
 - :app:processDebugAndroidTestManifest
 - :app:compileDebugAndroidTestRenderscript
 - :app:generateDebugAndroidTestBuildConfig
 - :app:generateDebugAndroidTestResValues
 - :app:generateDebugAndroidTestResources
 - :app:mergeDebugAndroidTestResources
 - :app:processDebugAndroidTestResources
 - :app:generateDebugAndroidTestSources
 - :app:mockableAndroidJar
 - :app:preDebugUnitTestBuild UP-TO-DATE
 - :app:prepareDebugUnitTestDependencies

Following these tasks, the message "BUILD SUCCESSFUL" is displayed, followed by "Total time: 7.489 secs".

Android Studio: Ejecutar Instant Run



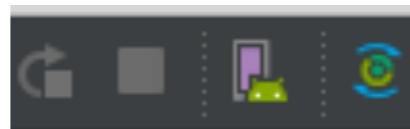
Android Studio: Ejecutar Emulador

Cuando no usarlo:

- WIFI
- Bluetooth
- NFC
- Eventos SD (Eject/insert)
- Eventos auriculares
- USB



Android Studio: Ejecutar Emulador



Android Virtual Device Manager

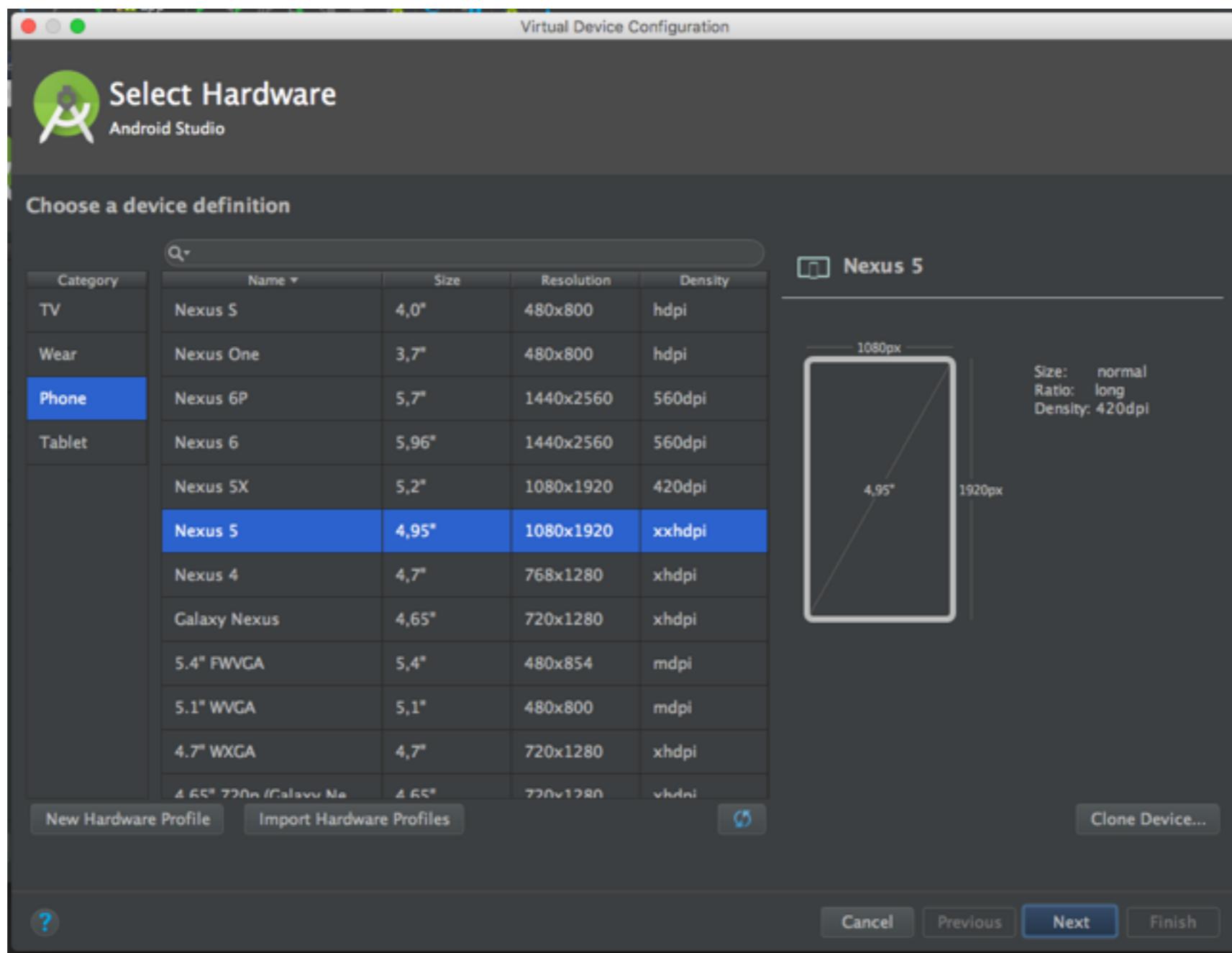
Your Virtual Devices

Android Studio

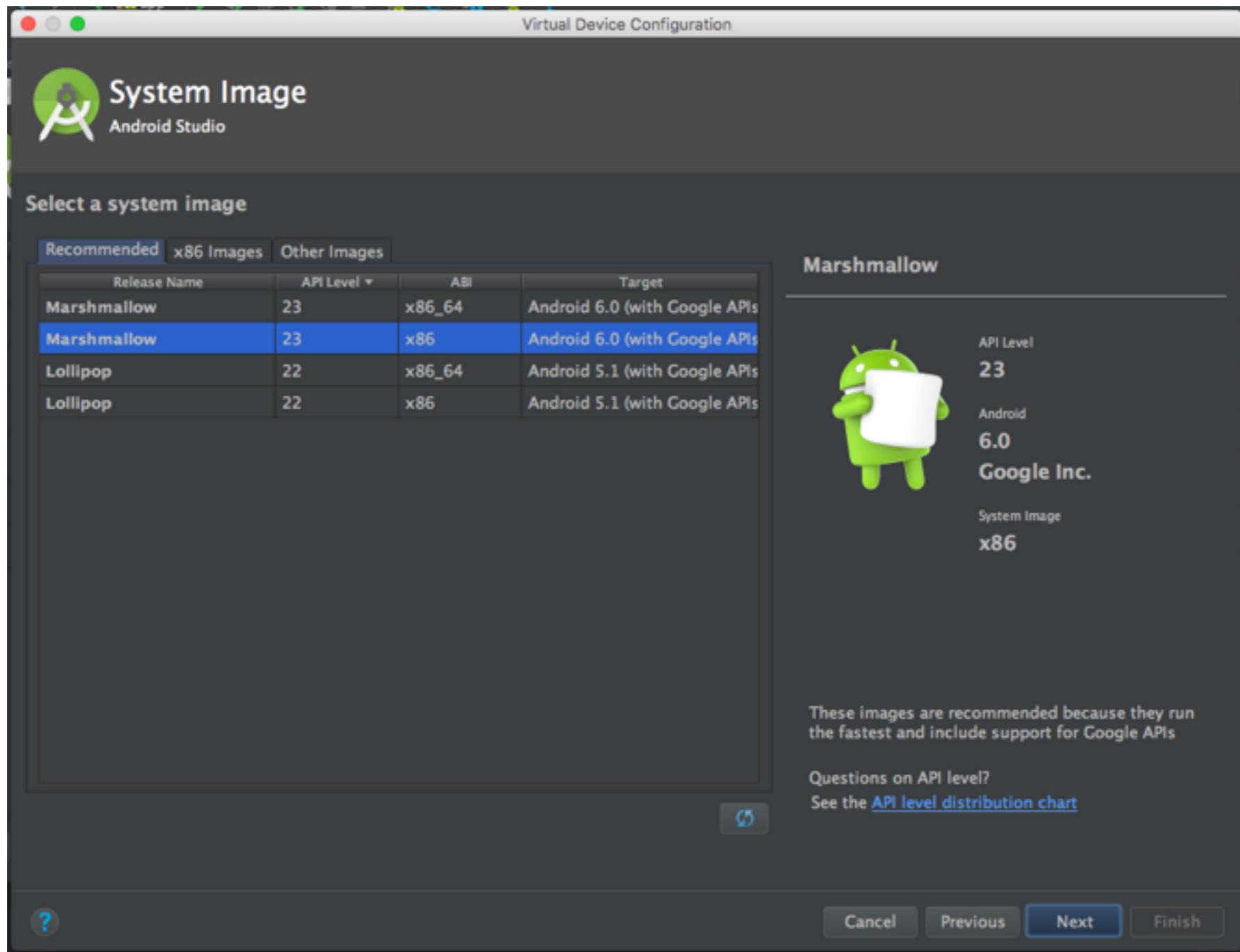
Type	Name	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
Tablet	N923	1536 x 2048: 280dpi	23	Android 6.0 (Google APIs)	x86	1 GB	
Tablet	Nexus 5 API 23 x86	1080 x 1920: xxhdpi	23	Android 6.0 (Google APIs)	x86	1 GB	
Tablet	RN44	720 x 1280: xhdpi	19	Android 4.4 (Google APIs)	x86	1 GB	
Tablet	RNEmu	1080 x 1920: xxhdpi	23	Android 6.0	x86...	1 GB	
Tablet	RNN5	1080 x 1920: xxhdpi	22	Android 5.1 (Google APIs)	x86...	1 GB	
Tablet	Xamarin Android API 15	Unknown Resolution	15	Android 4.0.3	arm	515 MB	Download
Tablet	Xamarin Android API 23	Unknown Resolution	23	Android 6.0	arm	1 GB	Download

? + Create Virtual Device... ⌂

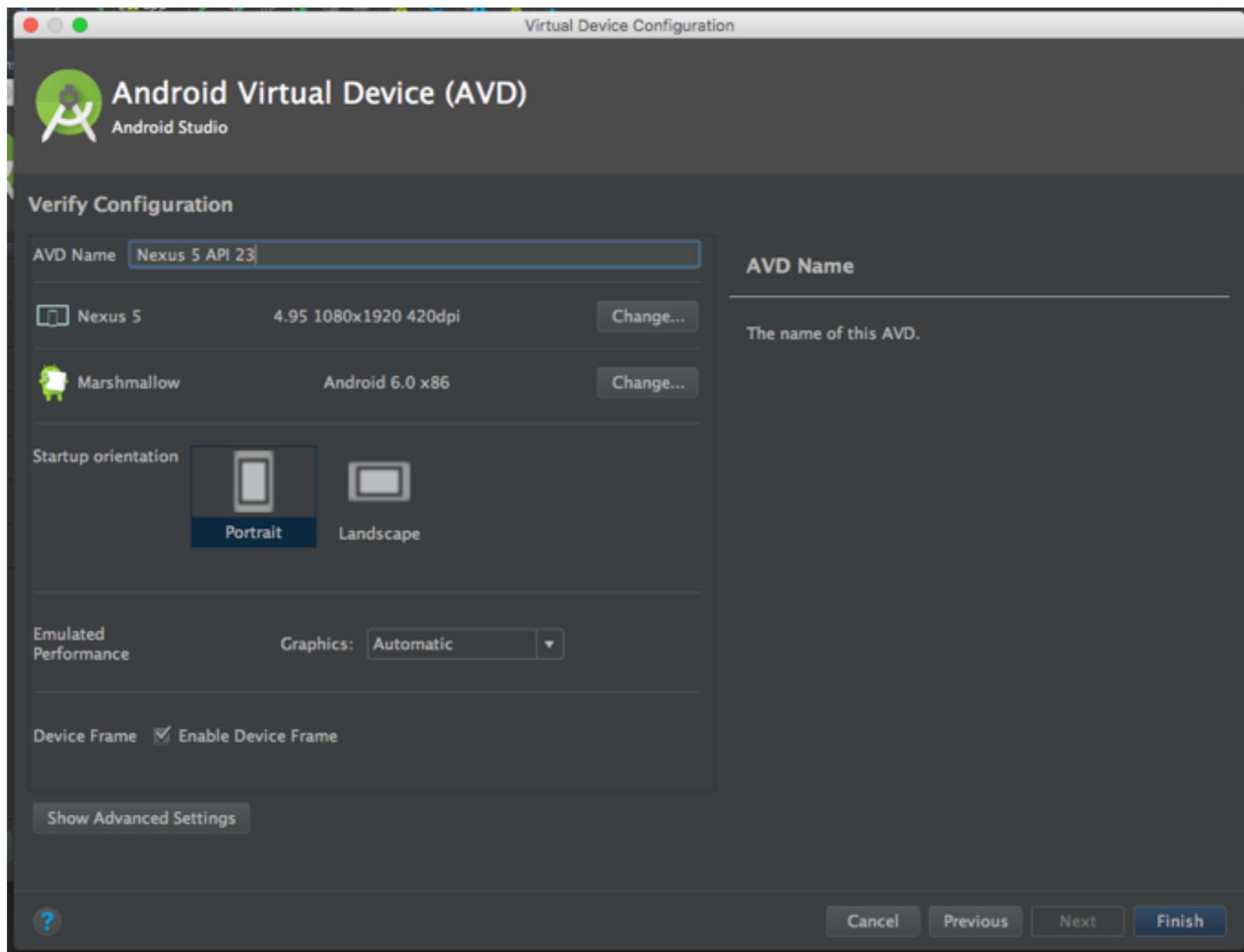
Android Studio: Ejecutar Emulador



Android Studio: Ejecutar Emulador

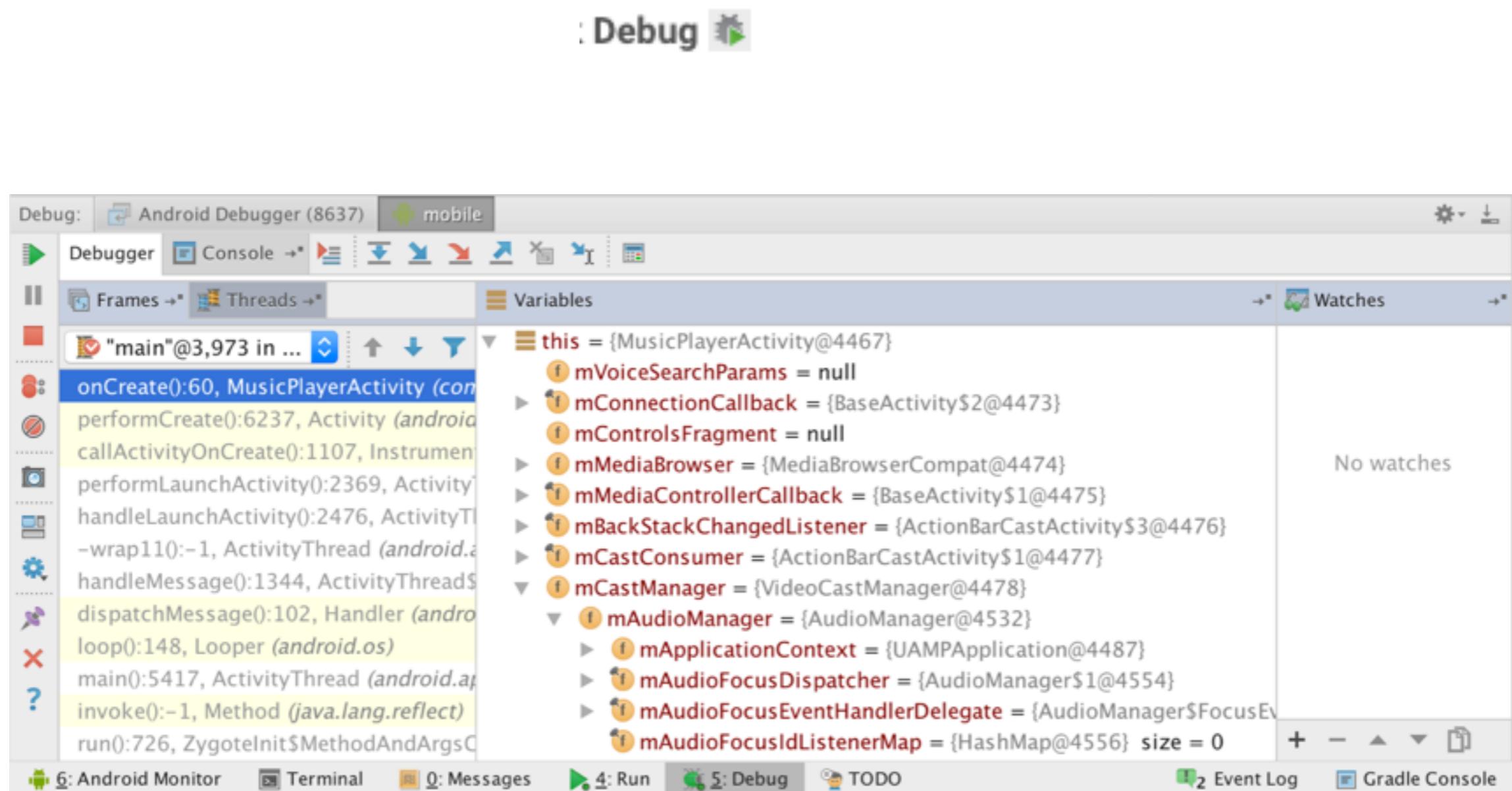


Android Studio: Ejecutar Emulador



Android Studio: Depurar

AndroidStudio Debugger



Android Studio: Depurar

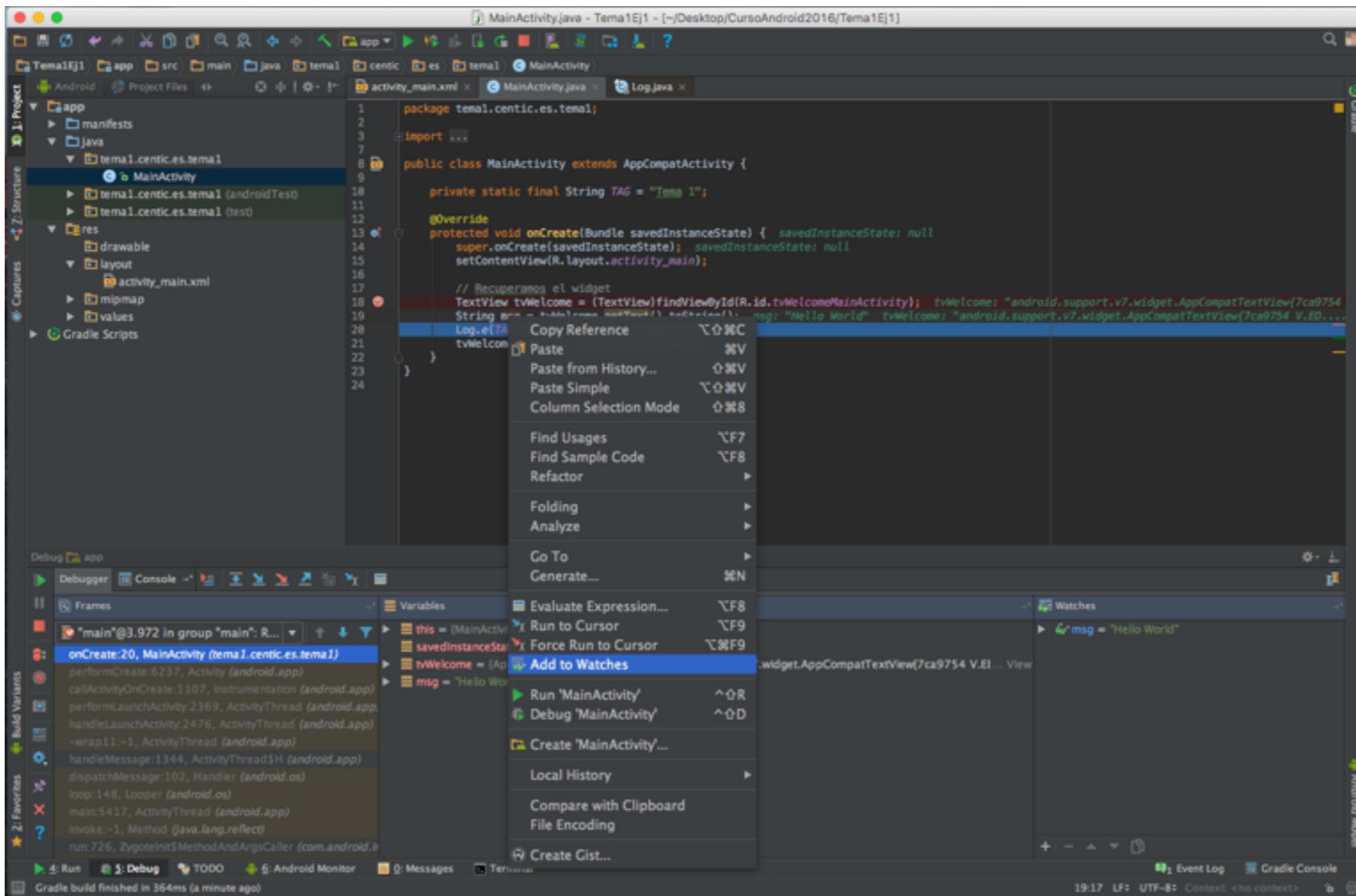
Breakpoint

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "Tema1Ej1". The `MainActivity.java` file is open in the editor.
- Editor:** The code for `MainActivity` is displayed. A red dot at line 18 indicates a breakpoint. The code includes a `Log.e` statement that outputs "Hello World".
- Debugger Tool Window:** The bottom half of the screen shows the "Debug" tool window. It displays the call stack under "Frames" and the local variables under "Variables". The variable `this` is set to `[MainActivity@4253]`. Other variables include `savedInstanceState`, `tvWelcome`, and `msg`.
- Bottom Status Bar:** Shows the message "Gradle build finished in 776ms (a minute ago)".

Android Studio: Depurar

Inspect



Android Studio: Depurar

Logger

- `Log.e(String, String)` (error)
- `Log.w(String, String)` (warning)
- `Log.i(String, String)` (information)
- `Log.d(String, String)` (debug)
- `Log.v(String, String)` (verbose)

```
private static final String TAG = "MyActivity";
...
Log.i(TAG, "MyClass.getView() – get item number " + position);
```

Android Studio: Depurar

Logger

- **Verbose** - Show all log messages (the default).
- **Debug** - Show debug log messages that are useful during development only, as well as the message levels lower in this list.
- **Info** - Show expected log messages for regular usage, as well as the message levels lower in this list.
- **Warn** - Show possible issues that are not yet errors, as well as the message levels lower in this list.
- **Error** - Show issues that have caused errors, as well as the message level lower in this list.
- **Assert** - Show issues that the developer expects should never happen.

The screenshot shows the Android Studio Logcat window. At the top, it displays the emulator configuration "Emulator Nexus_5X_API_23 Android 6.0, API 23" and the package name "com.example.android.displayingbitmaps (2496)". Below the title bar are several controls: "Verbose" dropdown, search bar, "Regex" checkbox, and "Show only selected application" dropdown. The main area is a scrollable list of log entries. Most entries are in black text, indicating they are at the Verbose level or higher. One entry is in red text, indicating it is at the Error level. The red entry is from the StrictMode class, specifically the setClassInstanceLimit method, which is being called on the ImageGridActivity class. The log also shows several GC (Garbage Collection) events, with timestamps and details about the objects being collected and the time taken.

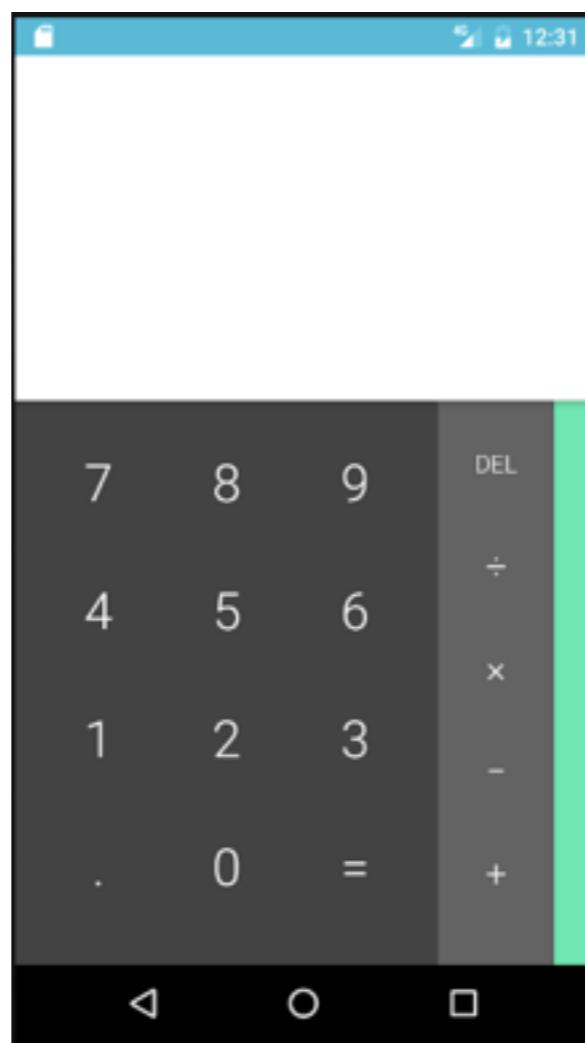
```
0(0B) LOS objects, 12% free, 27MB/31MB, paused 427us total 24.480ms
04-25 06:52:57.605 2496-2496/com.example.android.displayingbitmaps I/art: Starting a blocking GC Explicit
04-25 06:52:57.617 2496-2496/com.example.android.displayingbitmaps I/art: Explicit concurrent mark sweep GC freed 3(96B) AllocSpace objects,
0(0B) LOS objects, 12% free, 27MB/31MB, paused 352us total 11.588ms
04-25 06:52:57.626 2496-2496/com.example.android.displayingbitmaps I/art: Starting a blocking GC Explicit
04-25 06:52:57.635 2496-2496/com.example.android.displayingbitmaps I/art: Explicit concurrent mark sweep GC freed 3(96B) AllocSpace objects,
0(0B) LOS objects, 12% free, 27MB/31MB, paused 382us total 8.195ms
04-25 06:52:57.644 2496-2496/com.example.android.displayingbitmaps E/StrictMode: class com.example.android.displayingbitmaps.ui
    .ImageGridActivity; instances=2; limit=1
                                                android.os.StrictMode$InstanceCountViolation: class com.example
    .android.displayingbitmaps.ui.ImageGridActivity; instances=2; limit=1
                                                at android.os.StrictMode.setClassInstanceLimit(StrictMode
    .java:1)
04-25 06:53:42.232 2496-2496/com.example.android.displayingbitmaps I/art: Starting a blocking GC Explicit
04-25 06:53:42.248 2496-2496/com.example.android.displayingbitmaps I/art: Explicit concurrent mark sweep GC freed 270(16KB) AllocSpace objects,
0(0B) LOS objects, 12% free, 27MB/31MB, paused 384us total 15.932ms
04-25 06:53:42.252 2496-2496/com.example.android.displayingbitmaps I/art: Starting a blocking GC Explicit
```

Android Studio: Ejercicios

1. Crear un emulador para móvil
2. Crear un emulador para tablet
3. Crear un proyecto usando la plantilla pantalla completa
4. Crear un proyecto usando la plantilla Master/Details
5. Crear un proyecto sin usar plantillas
6. En el proyecto sin plantilla, añadir una nueva Activity

Android Studio: Ejercicios

1. En la activity (sin plantilla) crear la UI (usando el editor) para una aplicación calculadora



Android Studio: Ejercicios

1. Mostrar por el monitor los botones pulsados
2. Añadir breakpoint a la pulsación de un botón
3. Recoger el valor de las pulsaciones en una variable e inspeccionar el valor de esta.