

JAVA EN ANDROID

Día 2

Oscar Chávez-Bosquez

<http://chavezbosquez.github.io/diplomado/>

Contenido

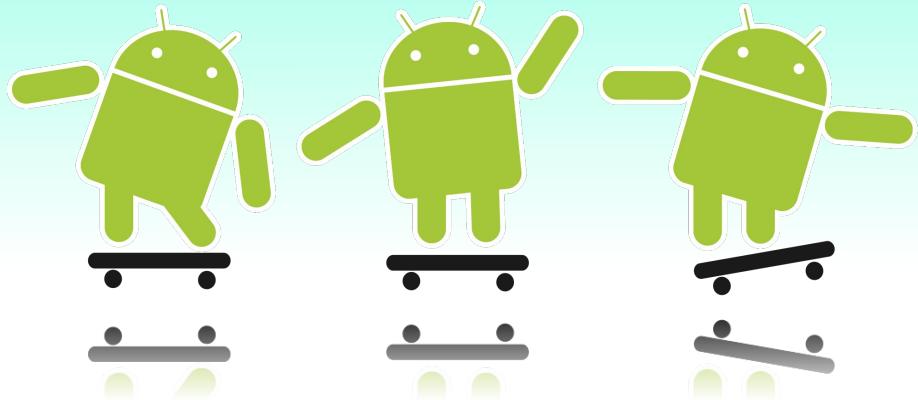


1. Introducción a Android
2. Instalación de Android Studio
3. Desarrollo de Apps
4. ¡Hola, mundo!
5. Calculadoraapp

Contenido



- 1. Introducción a Android**
2. Instalación de Android Studio
3. Desarrollo de Apps
4. ¡Hola, mundo!
5. Calculadoraapp



- Android es un **ecosistema**
- Android es una plataforma
- Android es una arquitectura

Ecosistema

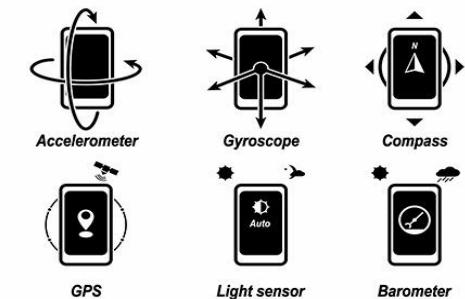
- Sistema operativo móvil basado en Linux 
- UI adaptada para pantallas táctiles:
 - Gestos: *swiping, tapping, pinching*
 - Teclado virtual
 - Bluetooth, USB, periféricos
- Altamente personalizable (dispositivos / fabricantes)

Ecosistema

- Utilizado en el 72% de todos los smartphones
- Diversos dispositivos: relojes, TVs, automóviles, ...
- *Open source*

Sensores

- Rotación del dispositivo
- Traslado ajusta la posición en el mapa
- Inclinación para controlar juegos o juguetes físicos
- Movimientos rápidos desactivan interacciones de videojuegos





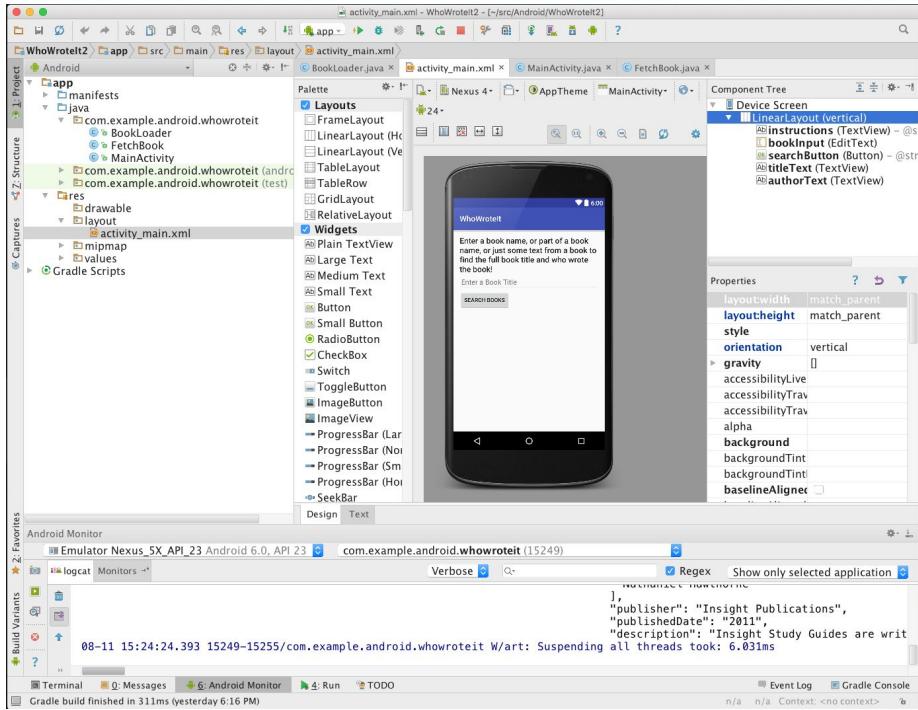
- Android es un ecosistema
- Android es una **plataforma**
- Android es una arquitectura

Android Software Developer Kit (SDK)

- Herramientas de desarrollo (*debugger, monitor, IDE*)
- Bibliotecas (mapas, *wearables*)
- Dispositivos virtuales (emuladores)
- Documentación:
<https://developer.android.com/?hl=es>



Android Studio



- IDE oficial Android

Desarrollo, ejecución,
depuración y
empaquetamiento de
apps

Google Play store

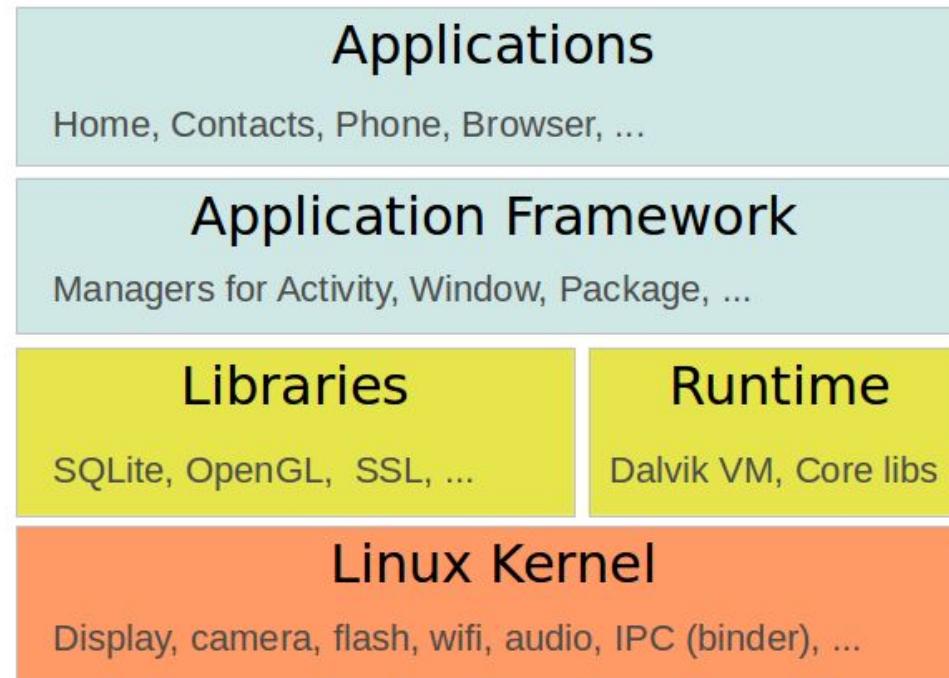
- Publicar apps
- Tienda oficial de Android
- Servicio de distribución digital operado por Google
- <https://play.google.com/store>





- Android es un ecosistema
- Android es una plataforma
- Android es una **arquitectura**

Stack de Android



Kernel de Linux

- Gestión de memoria de bajo nivel y subprocessos
- Seguridad
- *Drivers*

Bibliotecas C/C++

- Las bibliotecas base en C / C ++ brindan acceso a los componentes y servicios del sistema nativo nativo de Android.

Hardware Abstraction Layer (HAL)

- Interfaces estándar que exponen las capacidades de hardware del dispositivo como bibliotecas
- Ejemplos: cámara, módulo bluetooth

Aplicaciones

- Apps de sistema
- Apps desarrolladas



PROTECTED APPS	SYSTEM APPS	THIRD PARTY
These apps are installed on your device. Be cautious while denying or faking as it could affect the functionality of your device.		
	Android System	       
	Android System WebView	
	Audio Effects	
	Basic Daydreams	
	Bluetooth MIDI Service	
	Bluetooth Share	  

Java API Framework

- Todo el conjunto de funciones de Android está disponible a través de APIs escritas Java.
 - Jerarquía de clase **View** para crear UIs
 - *Notification manager*
 - Clase **Activity** para el ciclo de vida y la navegación

Android runtime

- Cada aplicación se ejecuta en su propio proceso con su propia instancia de Android Runtime.

Contenido



1. Introducción a Android
2. **Instalación de Android Studio**
3. Desarrollo de Apps
4. ¡Hola, mundo!
5. Calculadoraapp

Android Studio

android
studio



- Descarga:

<https://developer.android.com/studio>

- Guía de usuario:

<https://developer.android.com/studio/intro>

- Instalación:

<https://developer.android.com/studio/install>

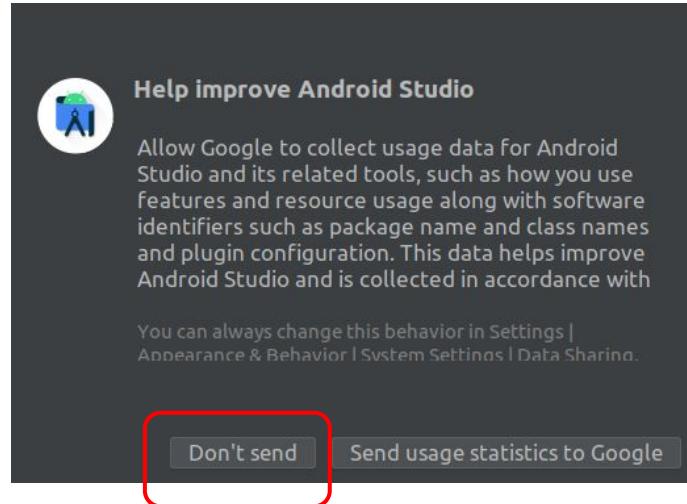
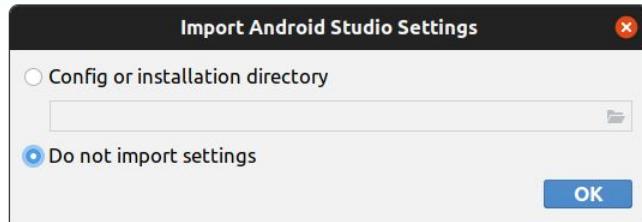
- Cómo crear una IU responsive con ConstraintLayout:
<https://developer.android.com/training/constraint-layout>
- Firma tu app:
<https://developer.android.com/studio/publish/app-signing?hl=es>

¿Problemas?

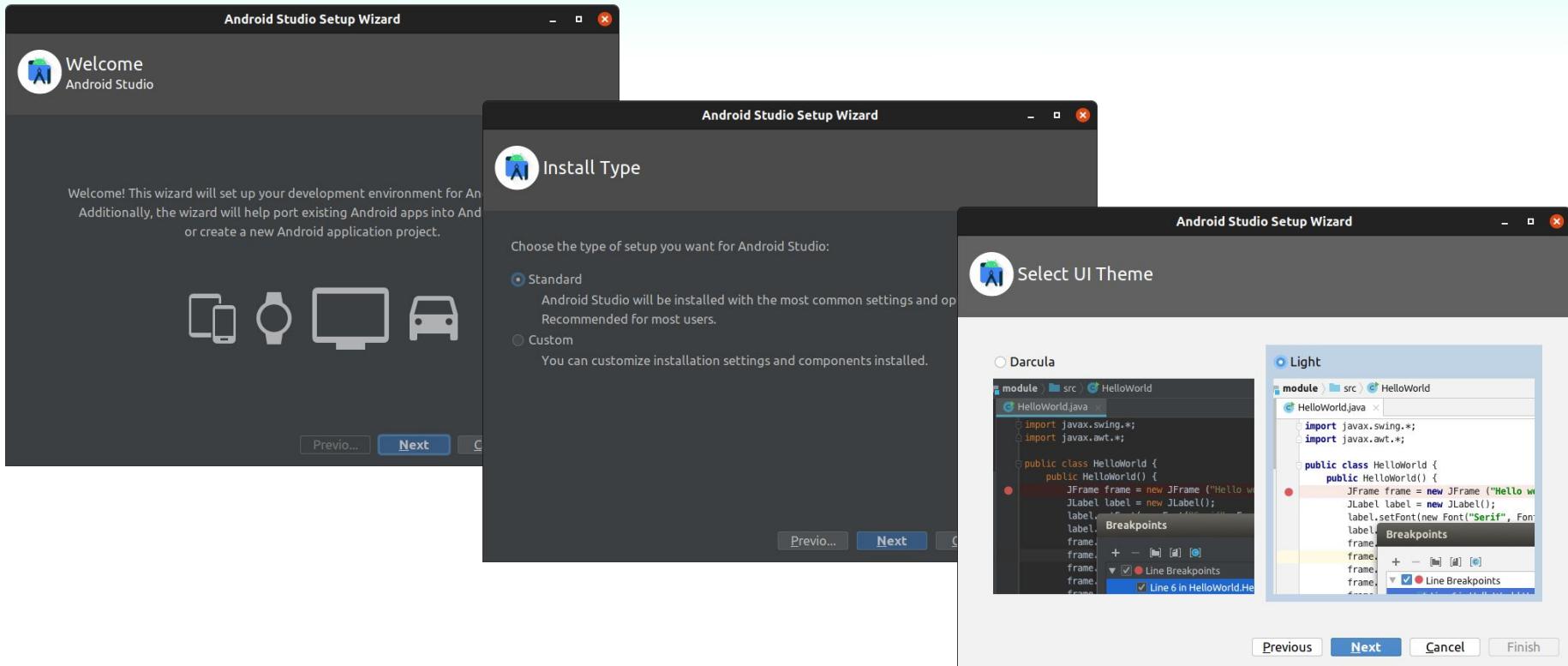


- :ERROR Android emulator gets killed
- :Error while waiting for device: Time out after 300seconds waiting for emulator to come online
- :IntelliJ shows method parameter hints on usage - How to disable it

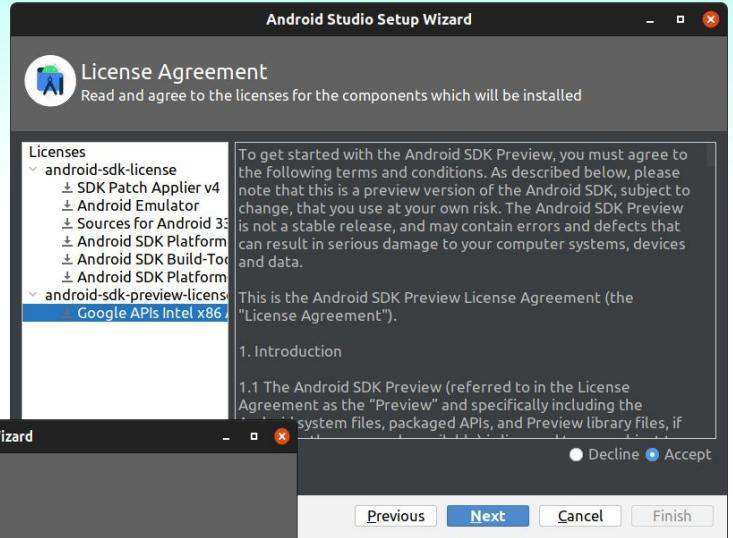
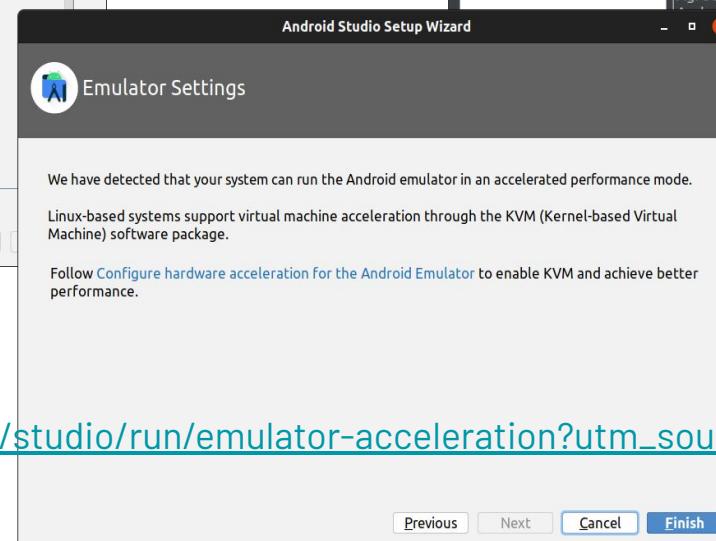
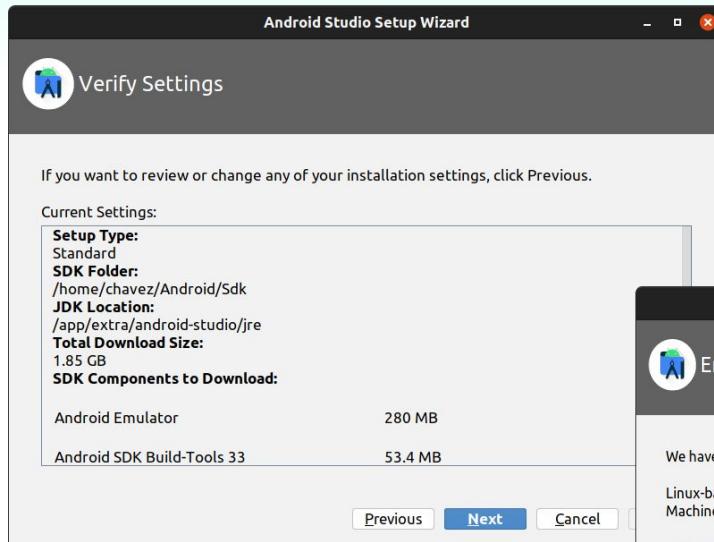
Inicio



Configuración

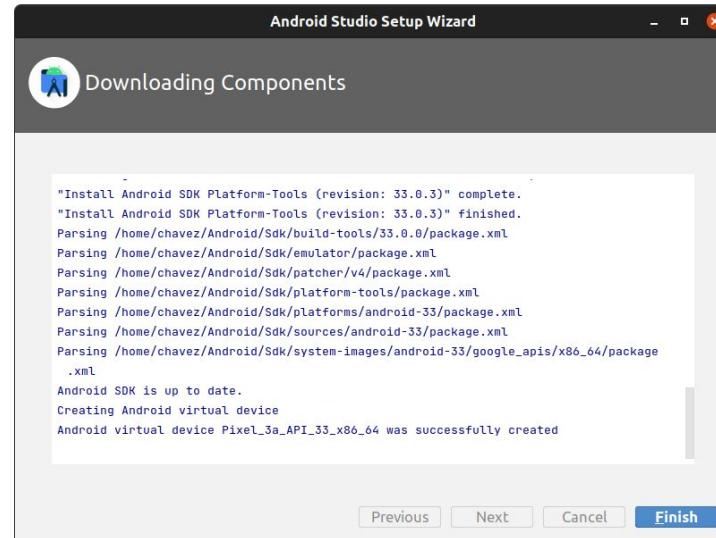
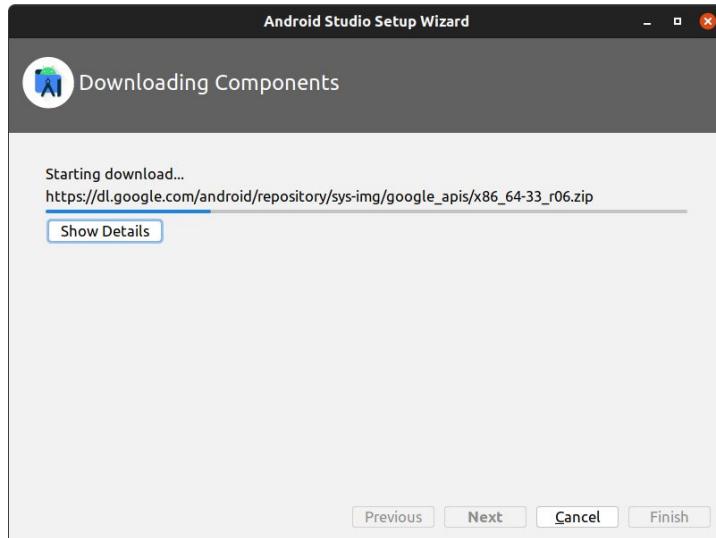


Configuración

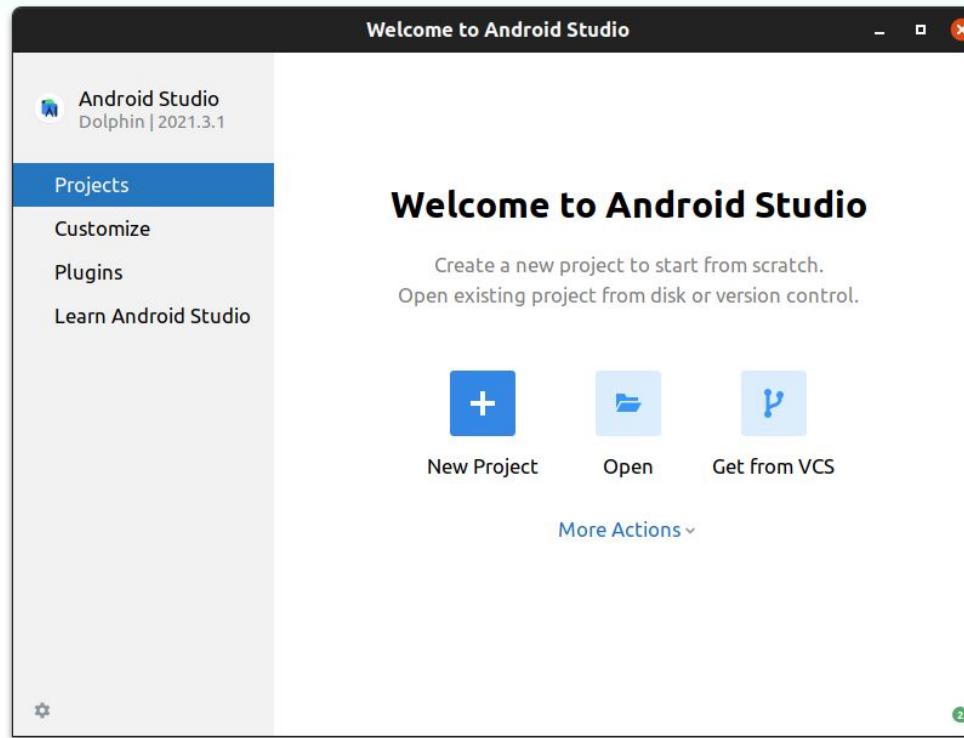


https://developer.android.com/studio/run/emulator-acceleration?utm_source=android-studio#vm-linux

Descarga final



Inicio



Contenido



1. Introducción a Android
2. Instalación de Android Studio
- 3. Desarrollo de Apps**
4. ¡Hola, mundo!
5. Calculadoraapp

¿Qué es una app?

- Una o más pantallas interactivas
- Escritas en **Java** y **XML**
- Usa el *Android Software Development Kit (SDK)*
- Usa bibliotecas Android y el *Android Application Framework*
- Se ejecuta en la *Android Runtime Virtual machine (ART)*

Retos a considerar

- Múltiples tamaños y resoluciones de pantalla
- **Rendimiento:** apps *responsivas* y fluidas
- **Seguridad:** código fuente y datos del usuario
- **Compatibilidad:** funciona en plataformas anteriores
- **Marketing:** considerar el mercado y a los usuarios

Bloques de construcción de apps

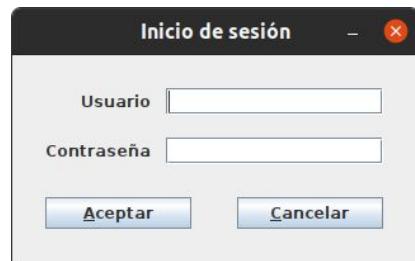
- **Recursos:** *layouts*, imágenes, texto y colores en archivos XML y archivos de medios
- **Componentes:** *activities*, servicios y clases en código Java
- **Manifest:** información acerca de la app para el runtime
- **Build configuration:** versiones de APK en archivos de configuración Gradle

Documentos para desarrolladores



- Introducción:
<https://developer.android.com/guide/index.html>
- Introducción a las Activities:
<https://developer.android.com/guide/components/activities/intro-activities>
- Diseño de UIs:
<https://developer.android.com/guide/topics/ui/declaring-layout>

Conceptos preliminares

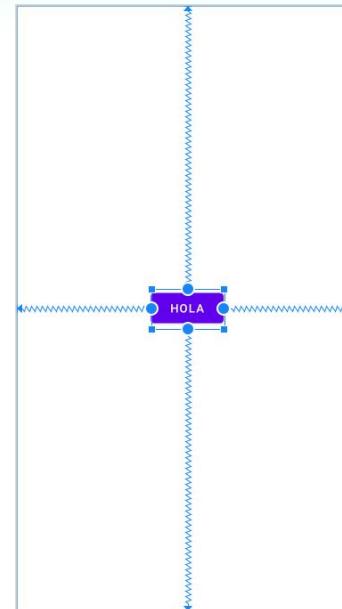


JFrame



Activity

Conceptos preliminares



Component

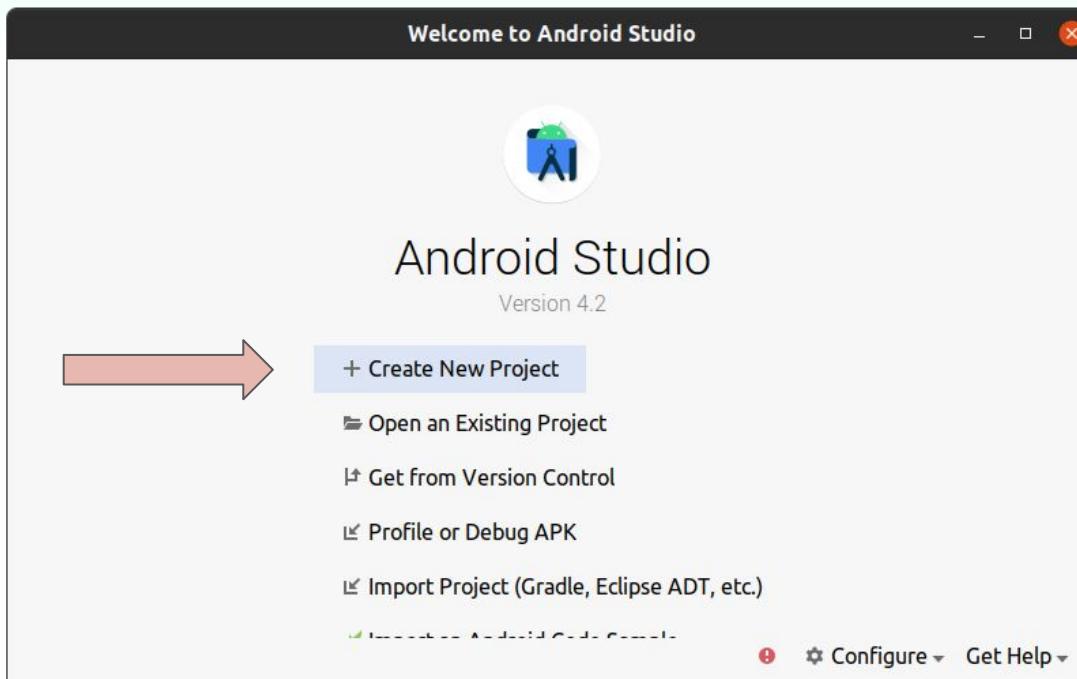
View

Contenido

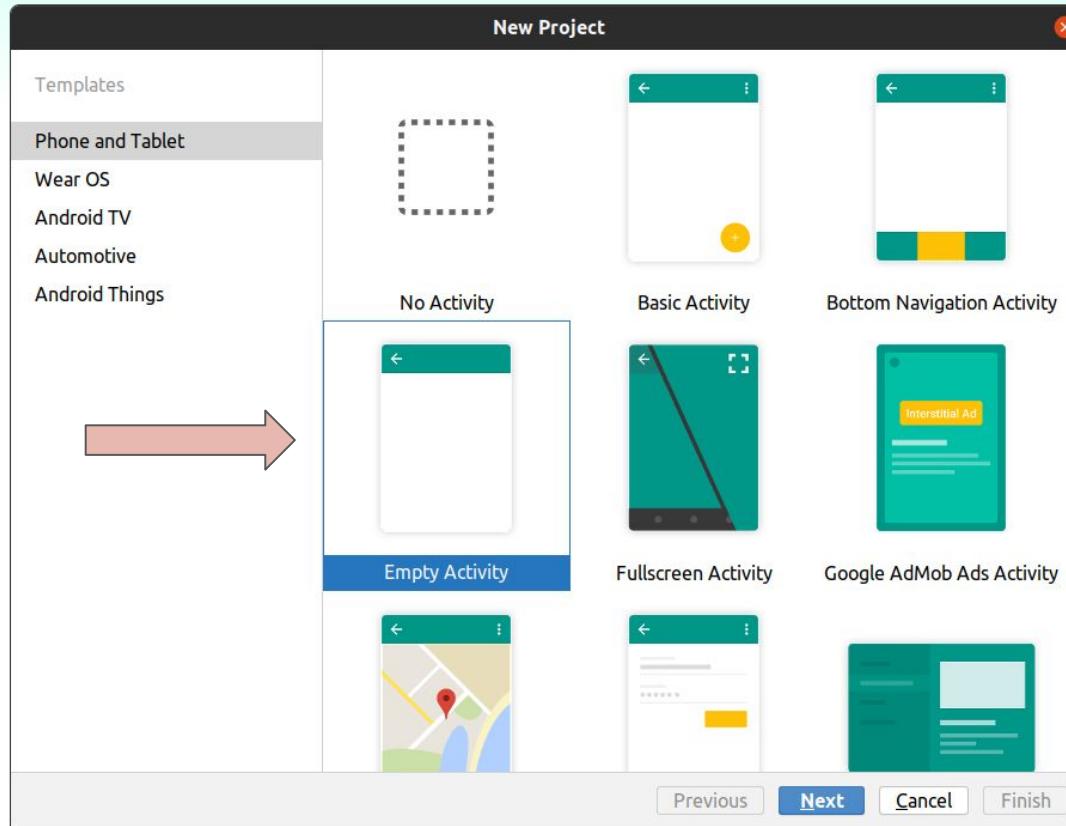


1. Introducción a Android
2. Instalación de Android Studio
3. Desarrollo de Apps
- 4. ¡Hola, mundo!**
5. Calculadoraapp

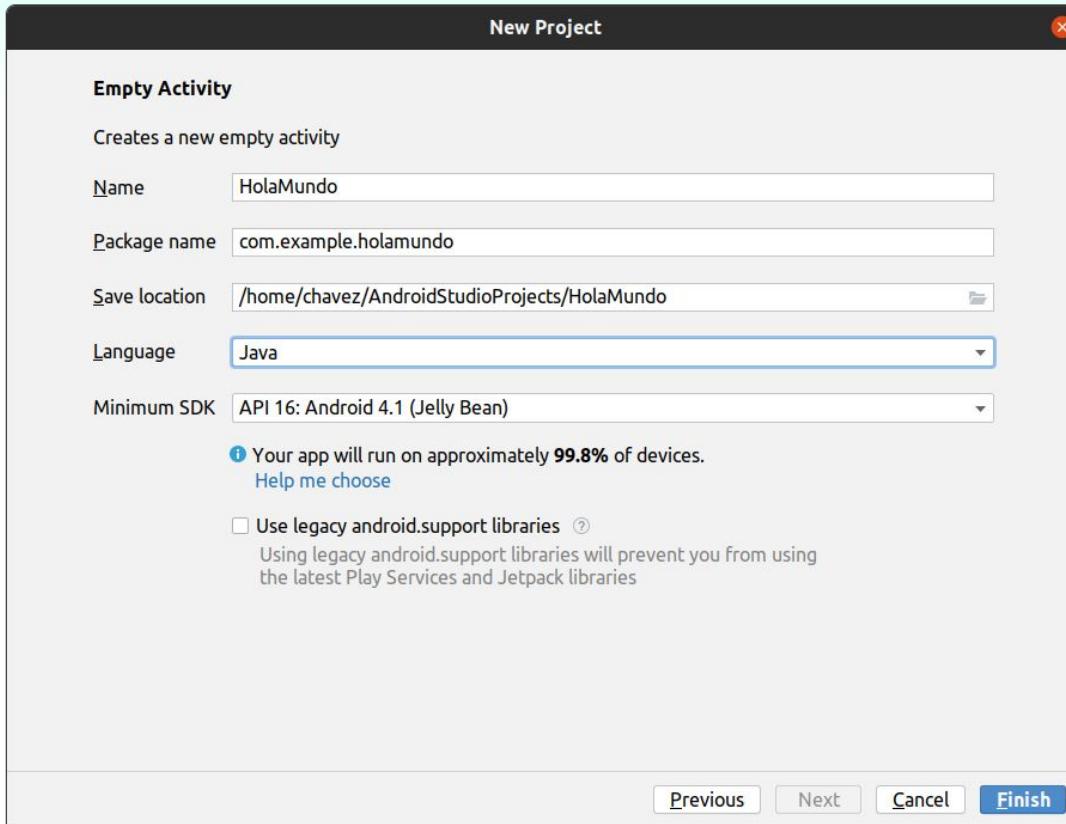
Nuevo proyecto



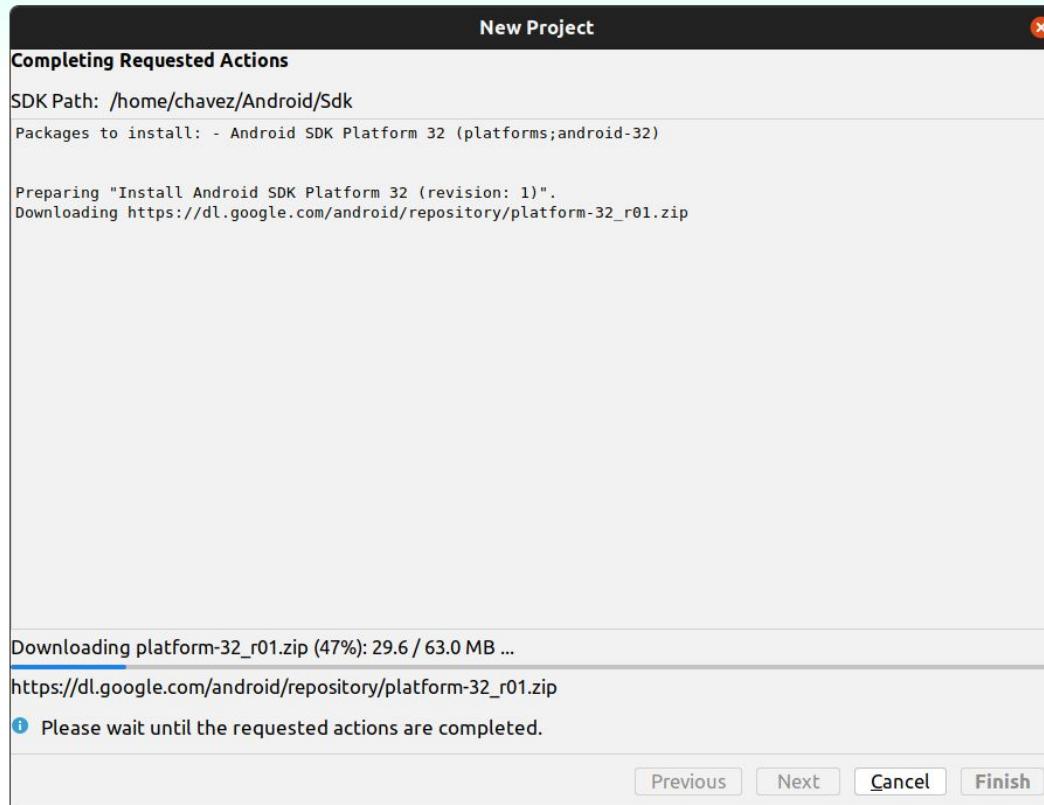
Activity vacía



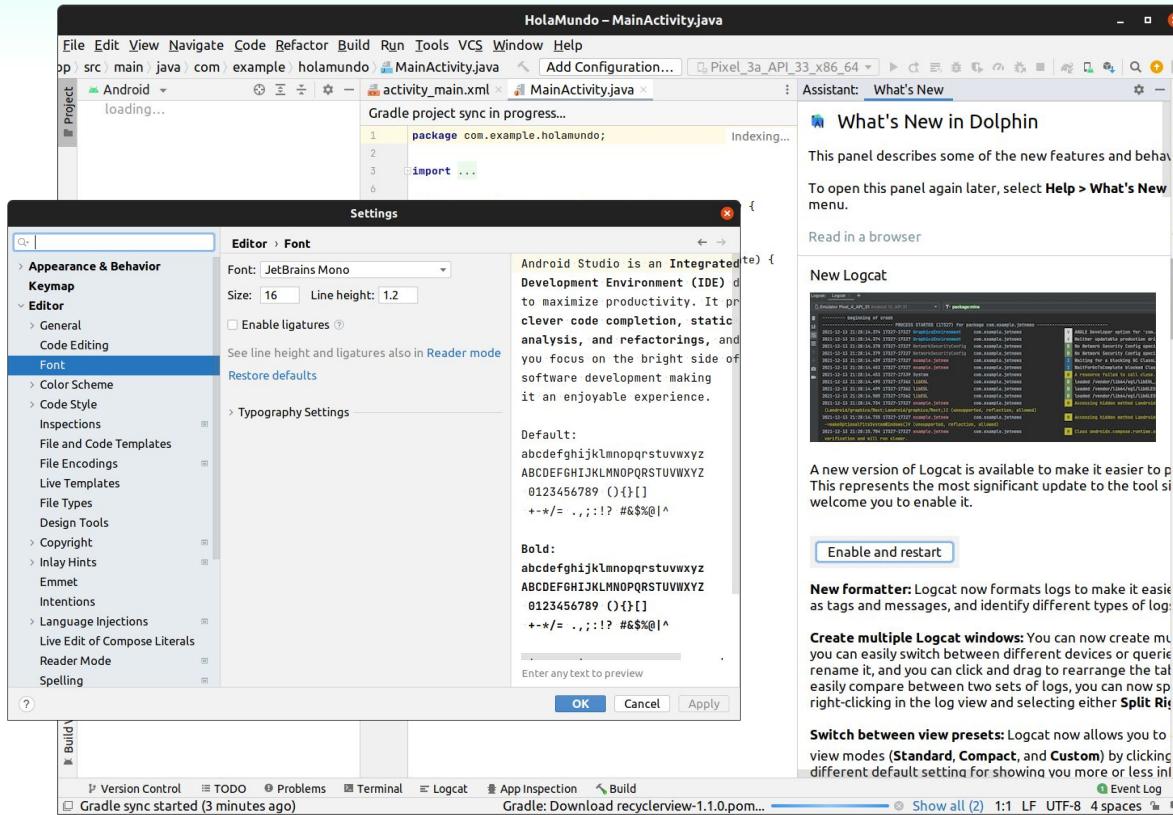
HolaMundo



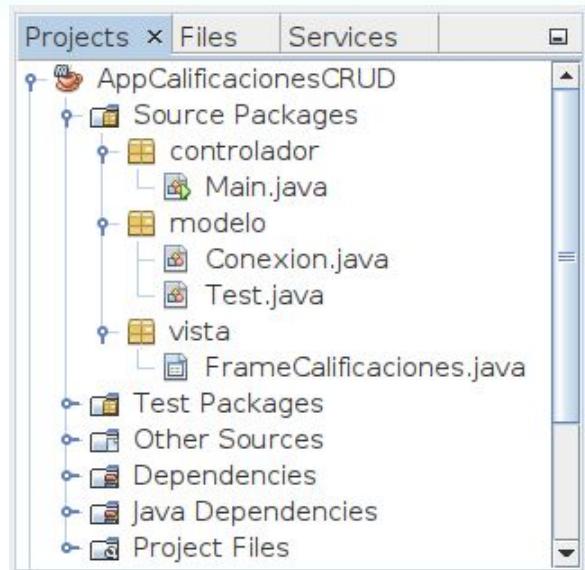
Una descarga más



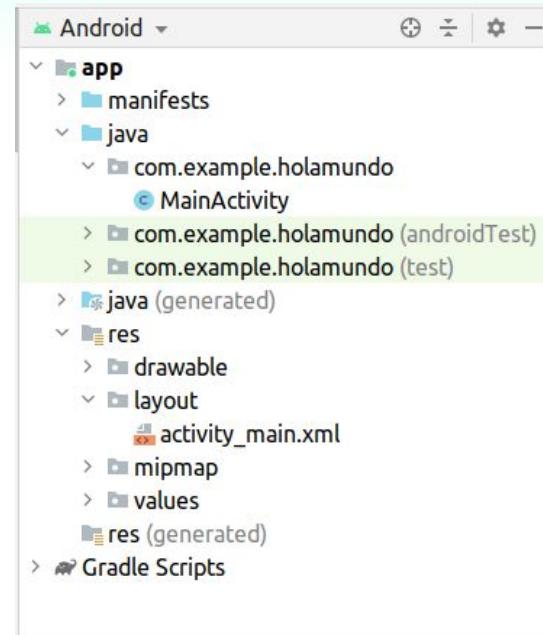
Configurando el IDE



Analogía con NetBeans

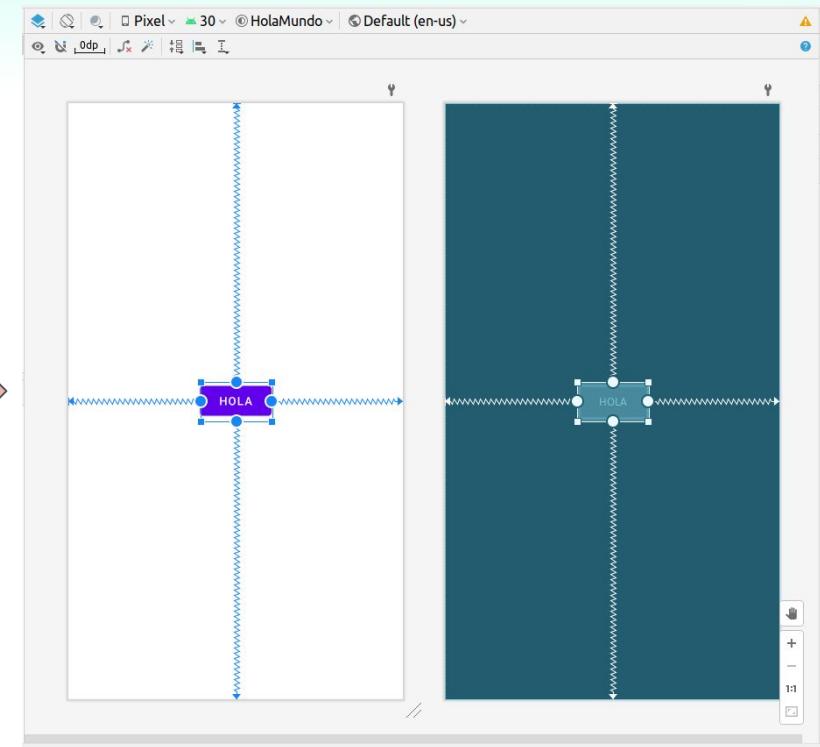
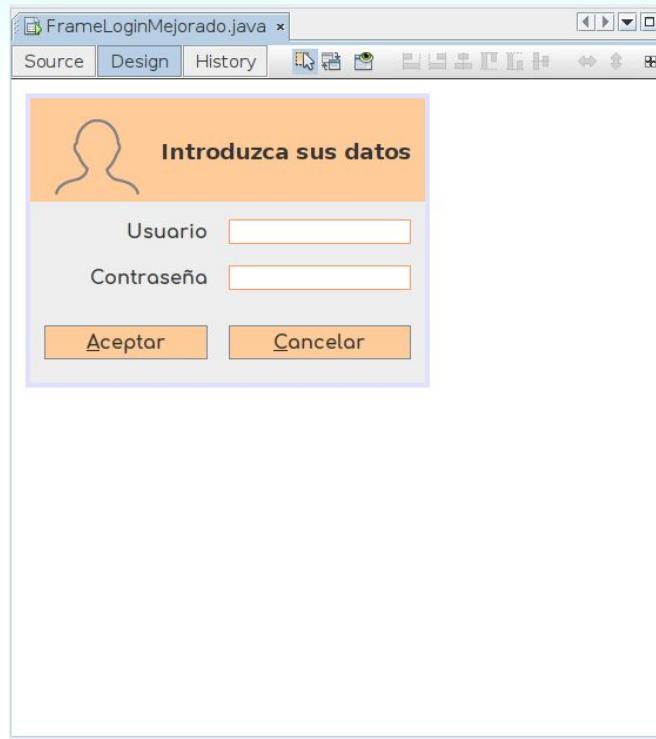


Panel de proyecto



Panel de proyecto

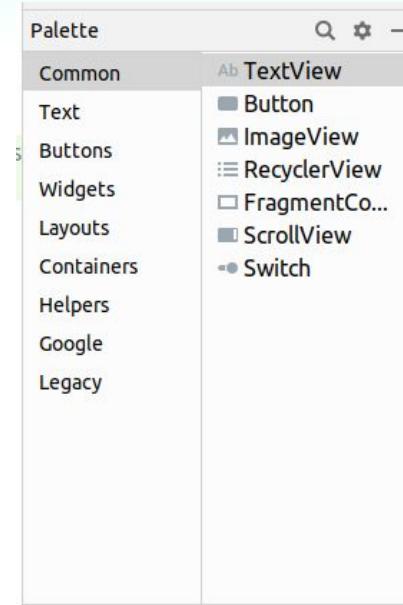
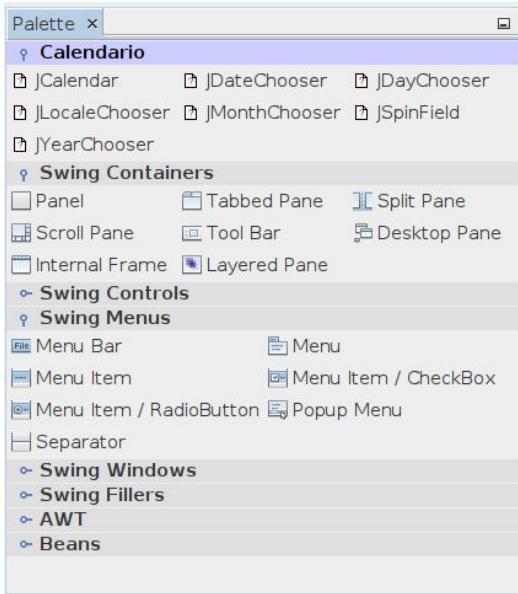
Analogía con NetBeans



Diseñador de UIs

Diseñador de UIs

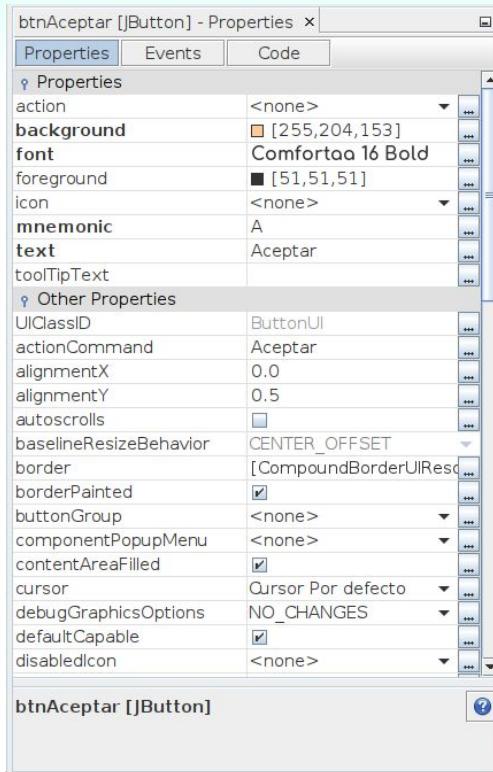
Analogía con NetBeans



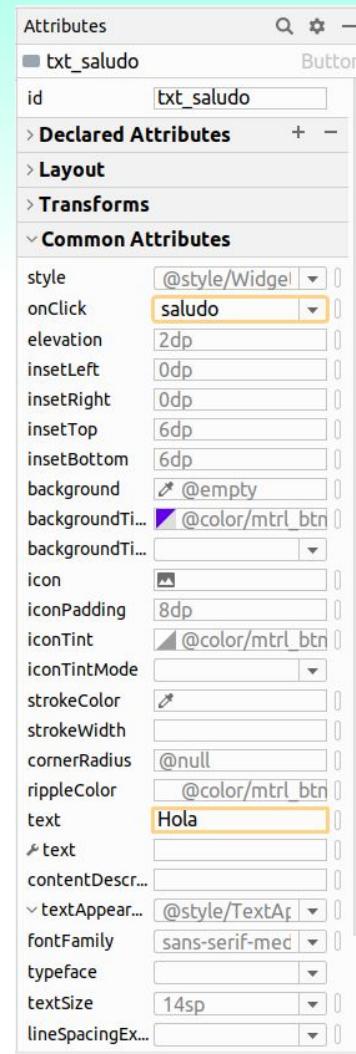
Paleta de
componentes

Paleta de
Vistas

Analogía con NetBeans

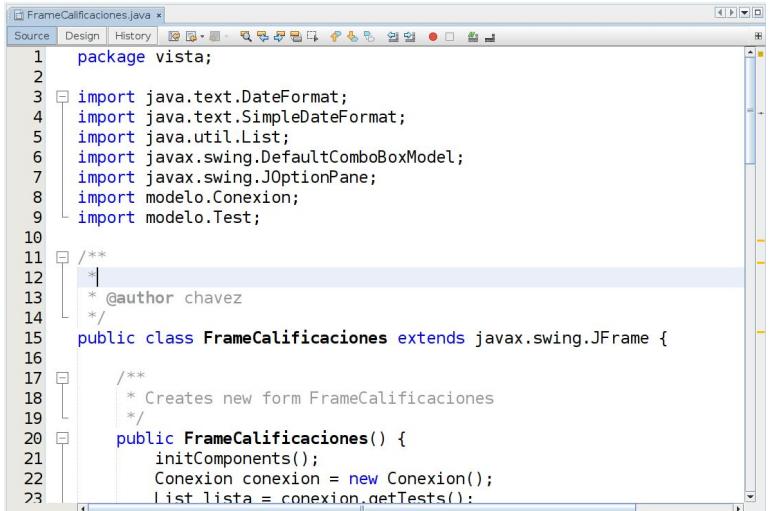


Panel de propiedades



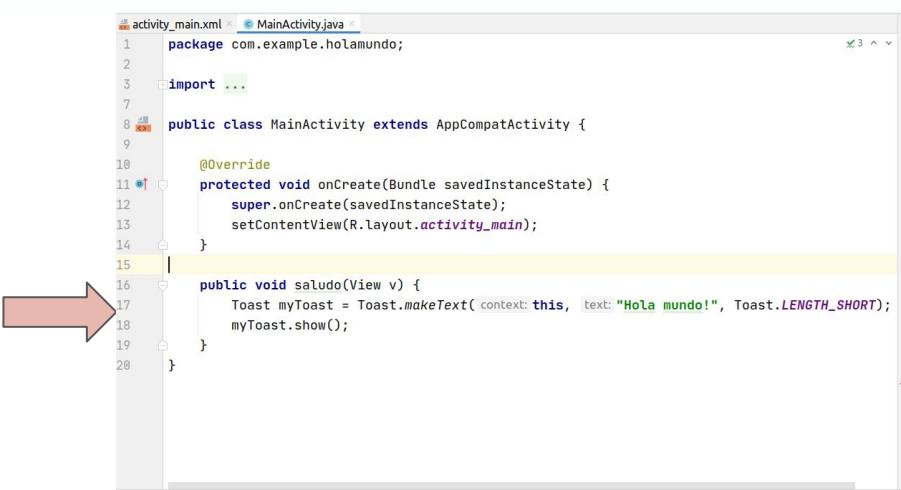
Panel de atributos

Analogía con NetBeans



```
FrameCalificaciones.java
Source Design History ...
1 package vista;
2
3 import java.text.DateFormat;
4 import java.text.SimpleDateFormat;
5 import java.util.List;
6 import javax.swing.DefaultComboBoxModel;
7 import javax.swing.JOptionPane;
8 import modelo.Conexion;
9 import modelo.Test;
10
11 /**
12 * @author Chavez
13 */
14 public class FrameCalificaciones extends javax.swing.JFrame {
15
16     /**
17      * Creates new form FrameCalificaciones
18      */
19     public FrameCalificaciones() {
20         initComponents();
21         Conexion conexion = new Conexion();
22         list lista = conexion.getTests();
23     }
24 }
```

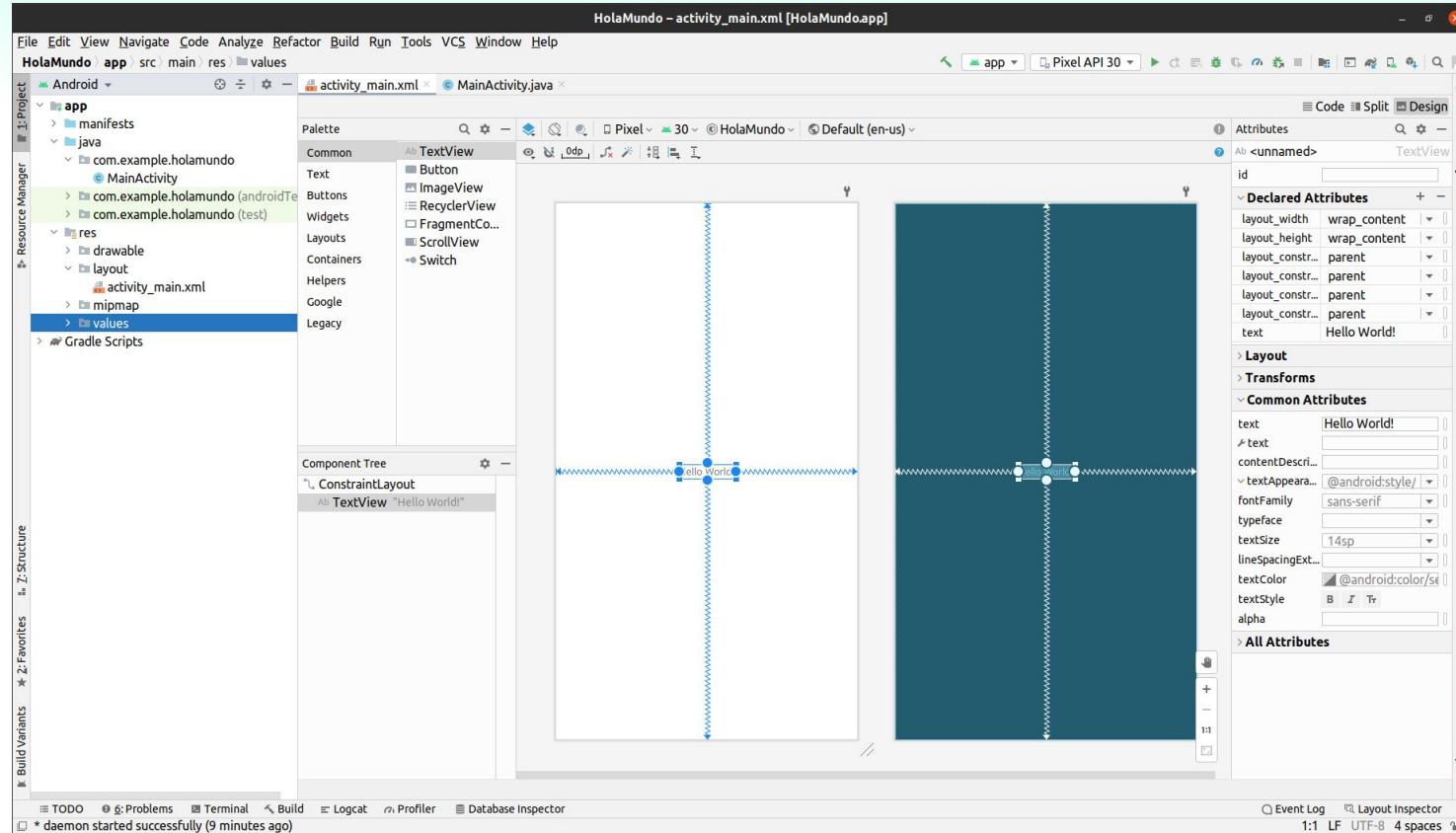
Editor de código



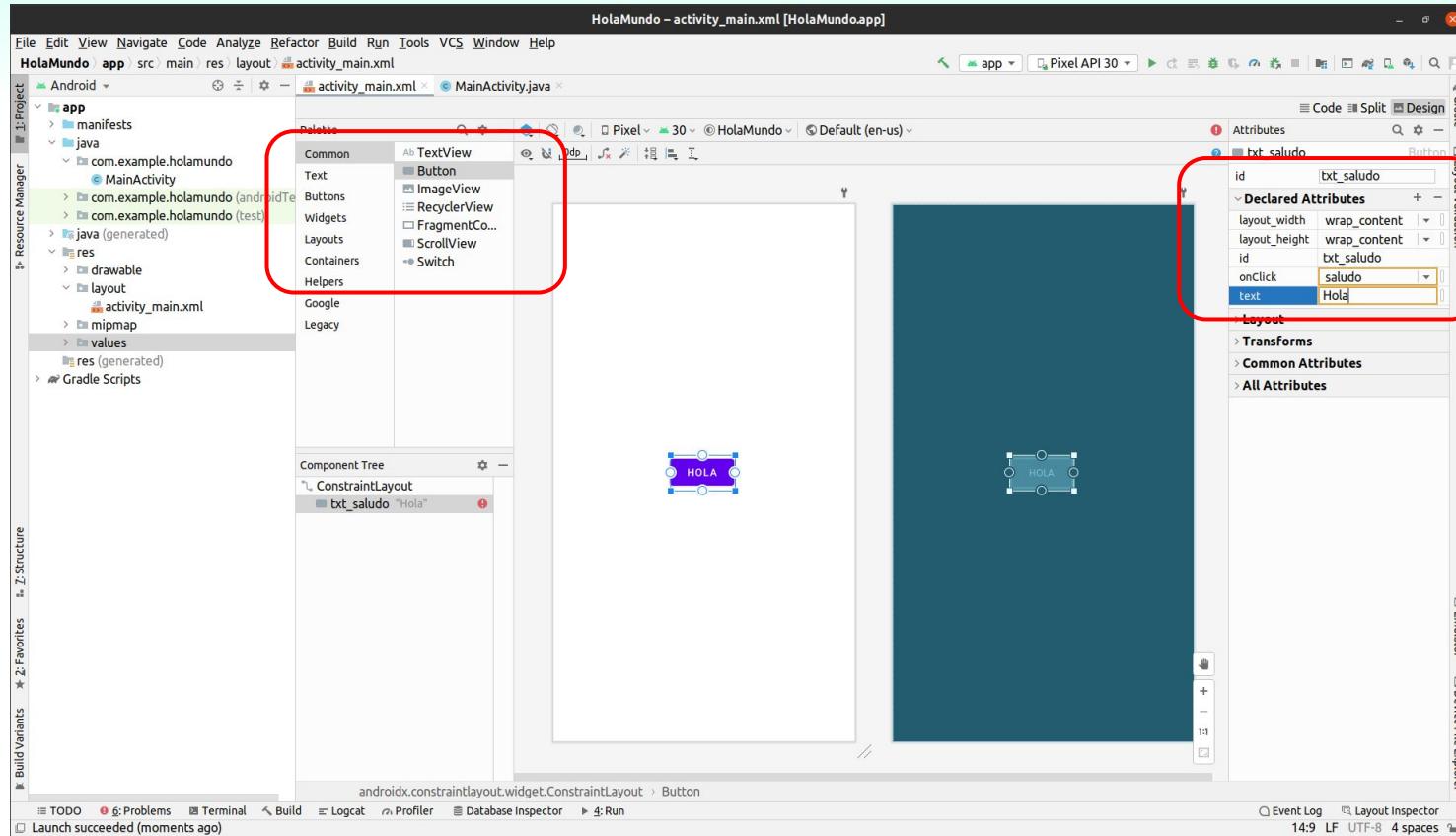
```
activity_main.xml  MainActivity.java
activity_main.xml  MainActivity.java ...
1 package com.example.holamundo;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main);
11    }
12
13    public void saludo(View v) {
14        Toast myToast = Toast.makeText(context: this, text: "Hola mundo!", Toast.LENGTH_SHORT);
15        myToast.show();
16    }
17
18 }
19
20 }
```

Editor de código

Diseño vs. Código



Diseño



Código

The screenshot shows the Android Studio interface with the project 'HolaMundo' open. The code editor displays the `MainActivity.java` file, which contains the following Java code:

```
package com.example.holamundo;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

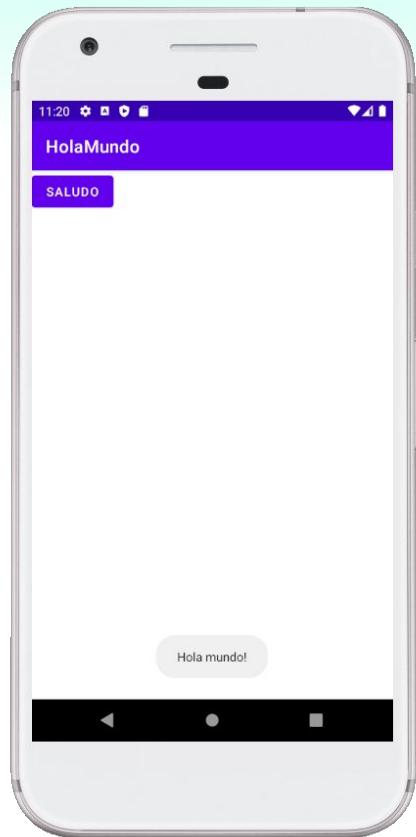
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void saludo(View v) {
        Toast myToast = Toast.makeText(context: this, text: "Hola mundo!", Toast.LENGTH_SHORT);
        myToast.show();
    }
}
```

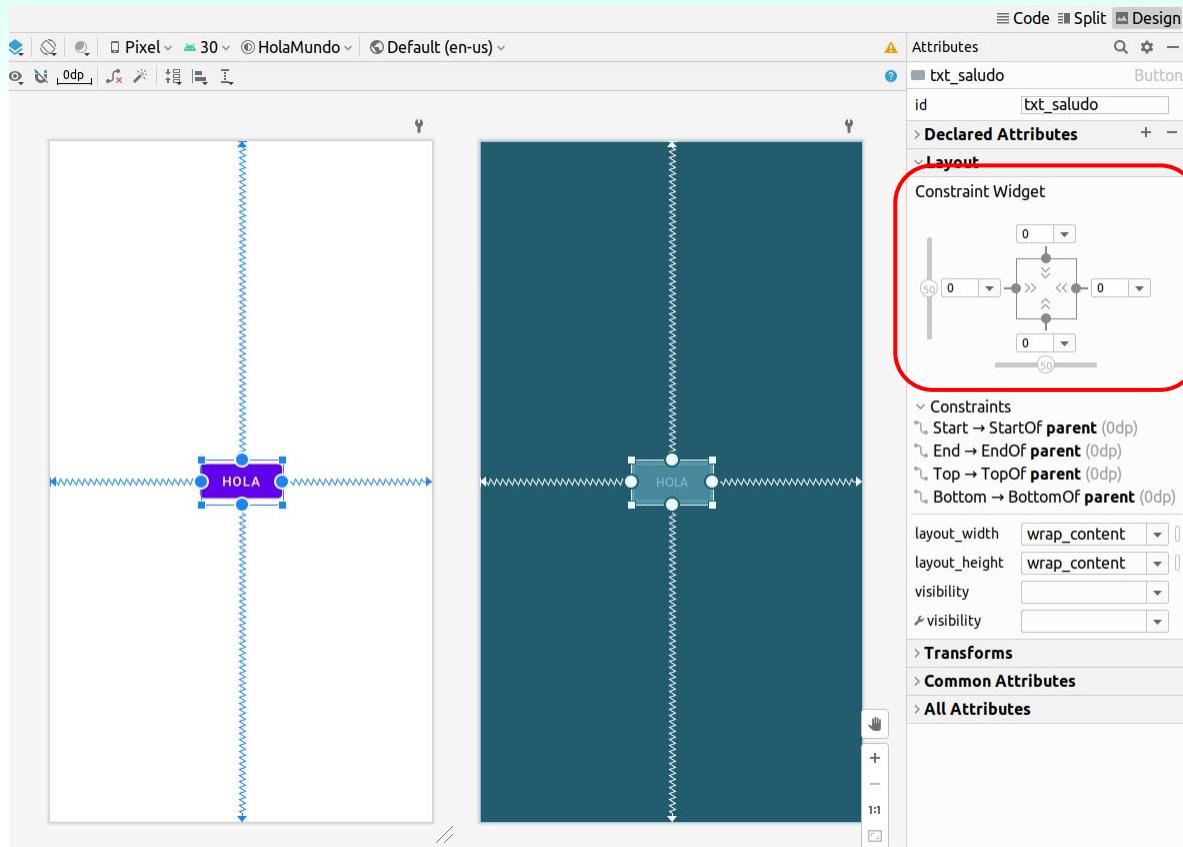
A red box highlights the `saludo` method and its implementation. The code uses `context: this` instead of the standard `this`. This is a common mistake when working with Java, as the context is not automatically passed to the `makeText` method.

The Android Studio interface includes the Project, Resource Manager, and Structure panes on the left; the Editor pane in the center; and various toolbars and status bars at the bottom.

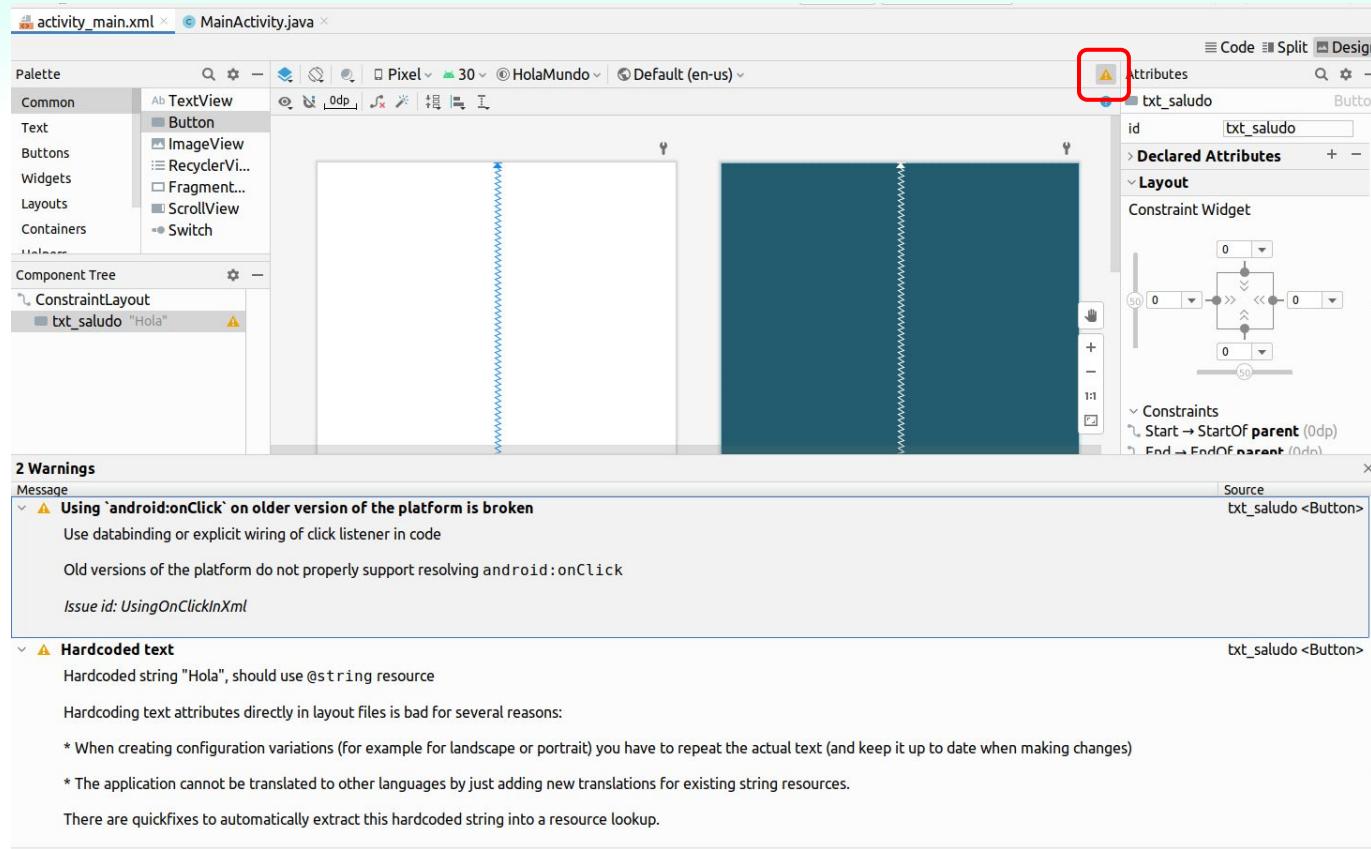
Ejecución



Layout



¿Advertencias?



Advertencia #1

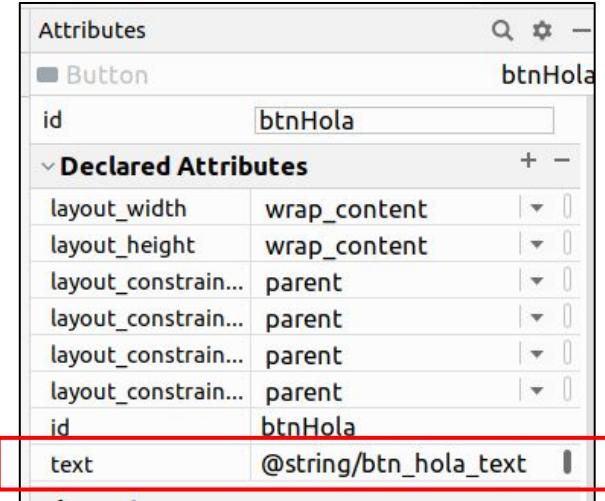
1. activity_main.xml: Eliminar método **onClick** (modo diseño)
2. MainActivity.java:

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
    private Button btnSaludo;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        btnSaludo = findViewById(R.id.btn_saludo);  
        btnSaludo.setOnClickListener(this);  
    }  
  
    @Override  
    public void onClick(View v) {  
        if (v == btnSaludo) {  
            Toast myToast = Toast.makeText(this, "Hola mundo!", Toast.LENGTH_SHORT);  
            myToast.show();  
        }  
    }  
}
```

Advertencia #2

1. activity_main.xml: Eliminar atributo **text** (modo Diseño)

2. activity_main.xml:



3. strings.xml:

```
<resources>
    <string name="app_name">HolaMundo</string>
    <string name="btn_hola_text">Hola</string>
</resources>
```

Eliminar una app del emulador

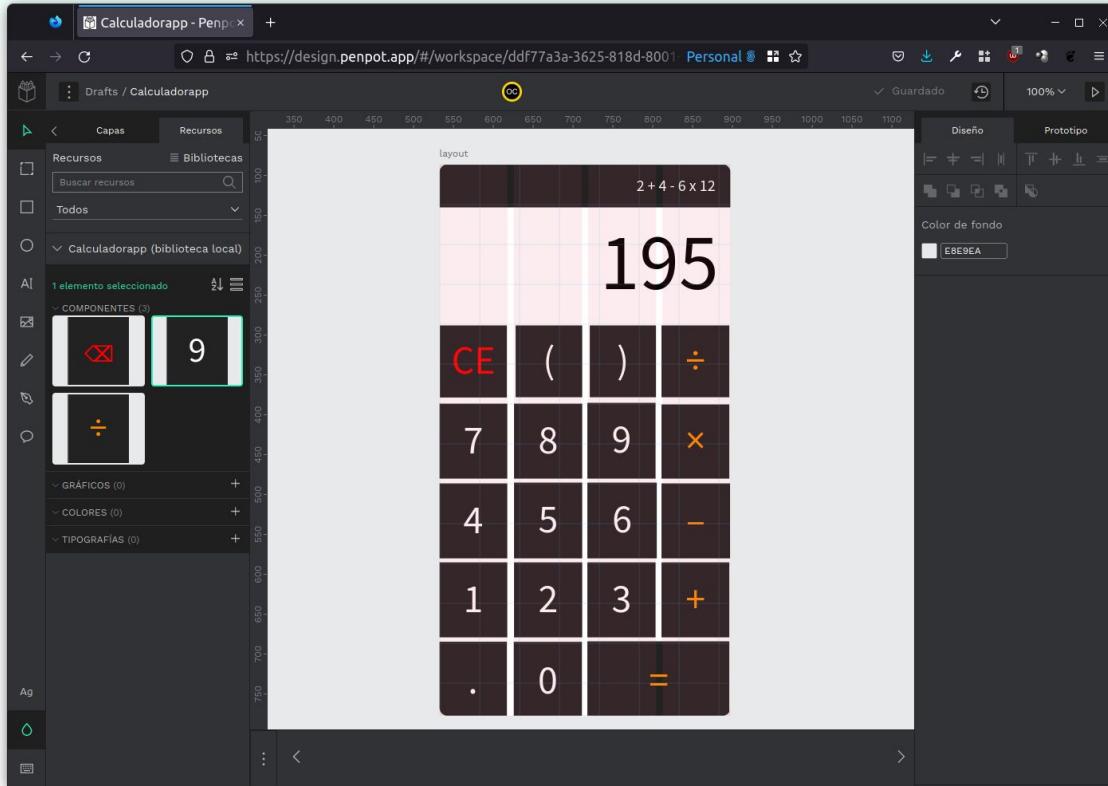
- *Settings → Apps & notifications*
- Seleccionar la app
- *Uninstall*

Contenido



1. Introducción a Android
2. Instalación de Android Studio
3. Desarrollo de Apps
4. ¡Hola, mundo!
5. Calculadoraapp □

Calculadorapp

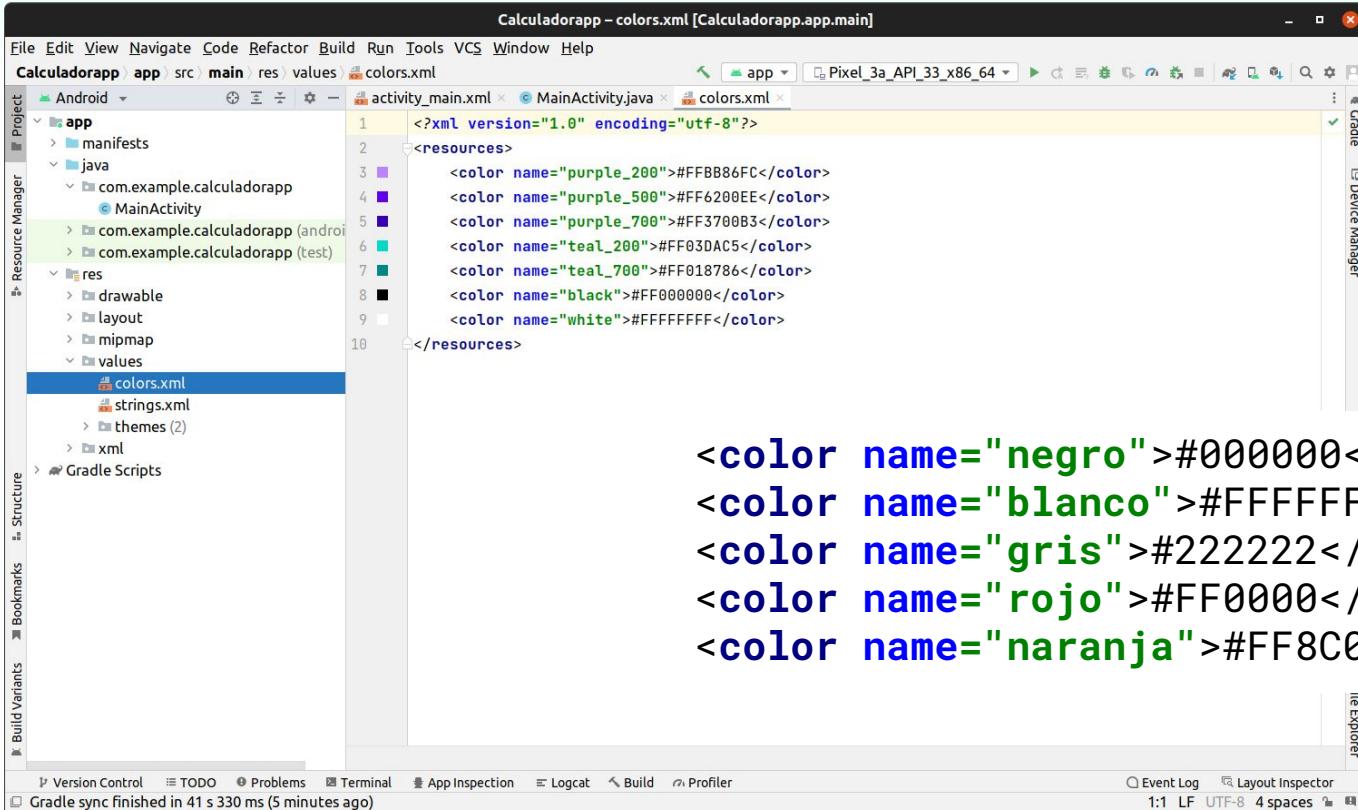


Diseño

The screenshot shows the homepage of the Penpot website. At the top, there's a navigation bar with links for "The Project", "About us", "Community", "Help center", "Log in", and a prominent green "Sign up for free" button. A banner at the top announces "We're announcing our Series A funding!" with a "Learn more" link. The main headline reads "Design Freedom for Teams". Below the headline, a paragraph explains that Penpot is an open-source design and prototyping platform for cross-domain teams, based on web standards (SVG) and powered by the community. A "Get started, it's free!" button is visible, along with a "Product Hunt" badge featuring a red arrow pointing upwards. At the bottom, a preview of the Penpot interface shows a timeline with various design and prototype elements.

<https://penpot.app>

Configuración de colores



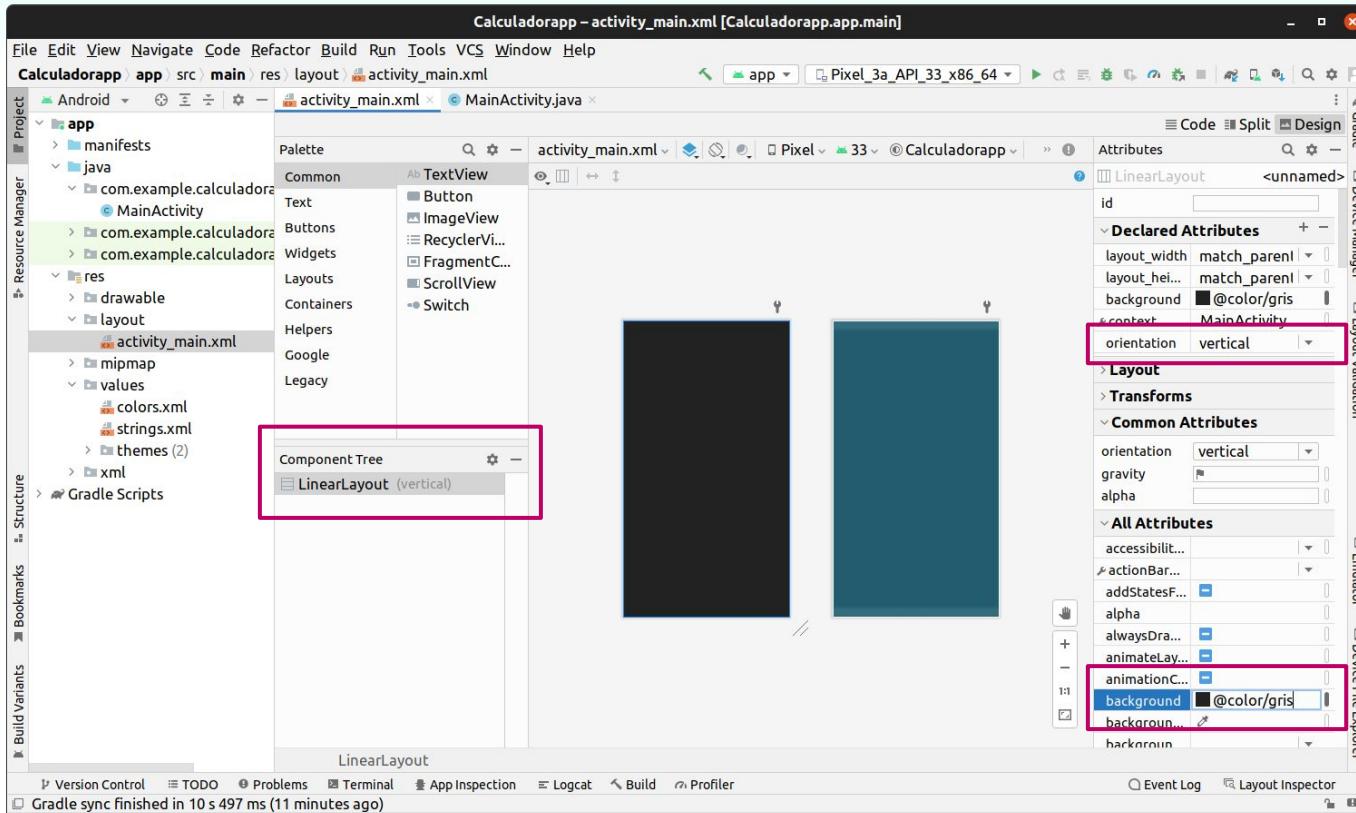
The screenshot shows the Android Studio interface with the project 'Calculadorapp' open. The 'colors.xml' file is selected in the Project structure. The code editor displays the XML configuration for colors:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFF</color>
</resources>
```

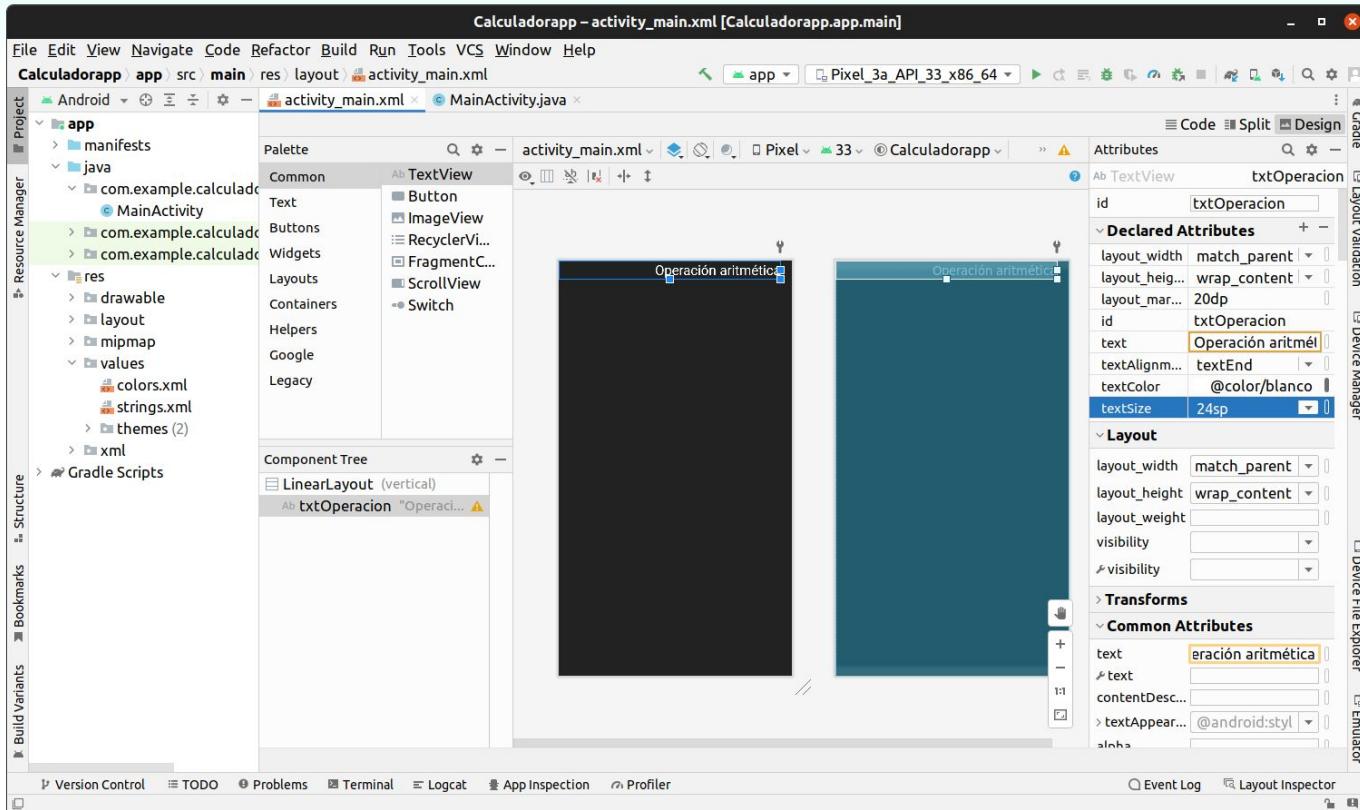
Below the code editor, a list of color names and their corresponding hex codes is displayed:

- <color name="negro">#000000</color>
- <color name="blanco">#FFFFFF</color>
- <color name="gris">#222222</color>
- <color name="rojo">#FF0000</color>
- <color name="naranja">#FF8C00</color>

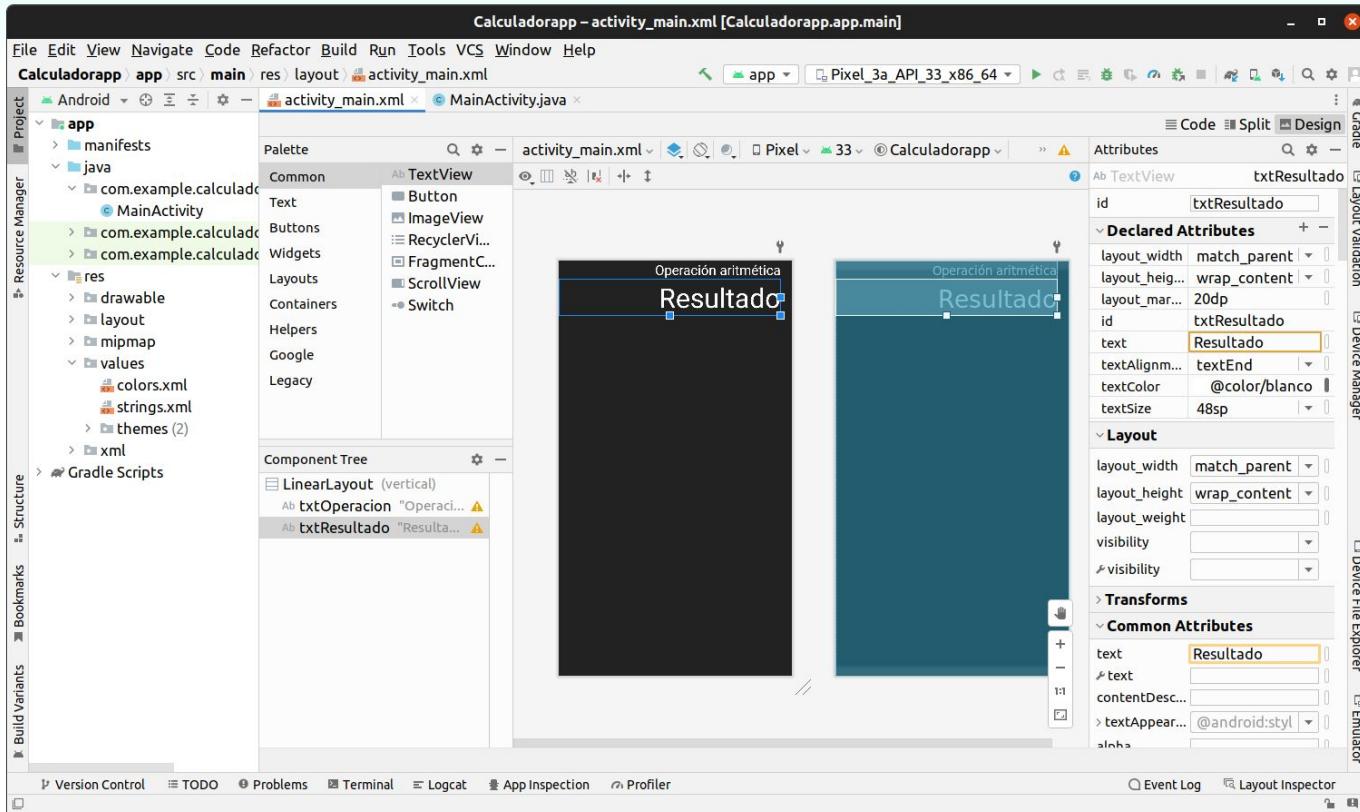
activity_main.xml: LinearLayout vertical



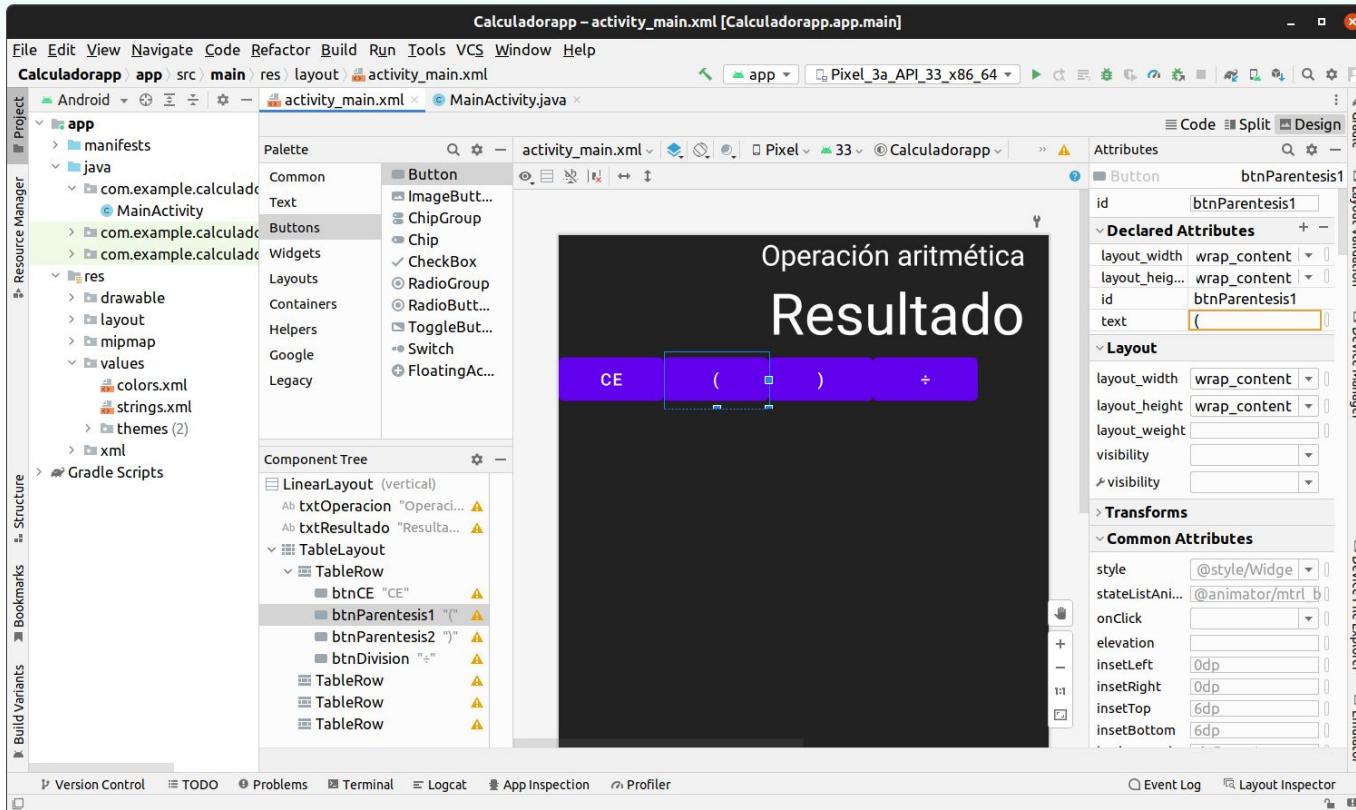
TextView 1



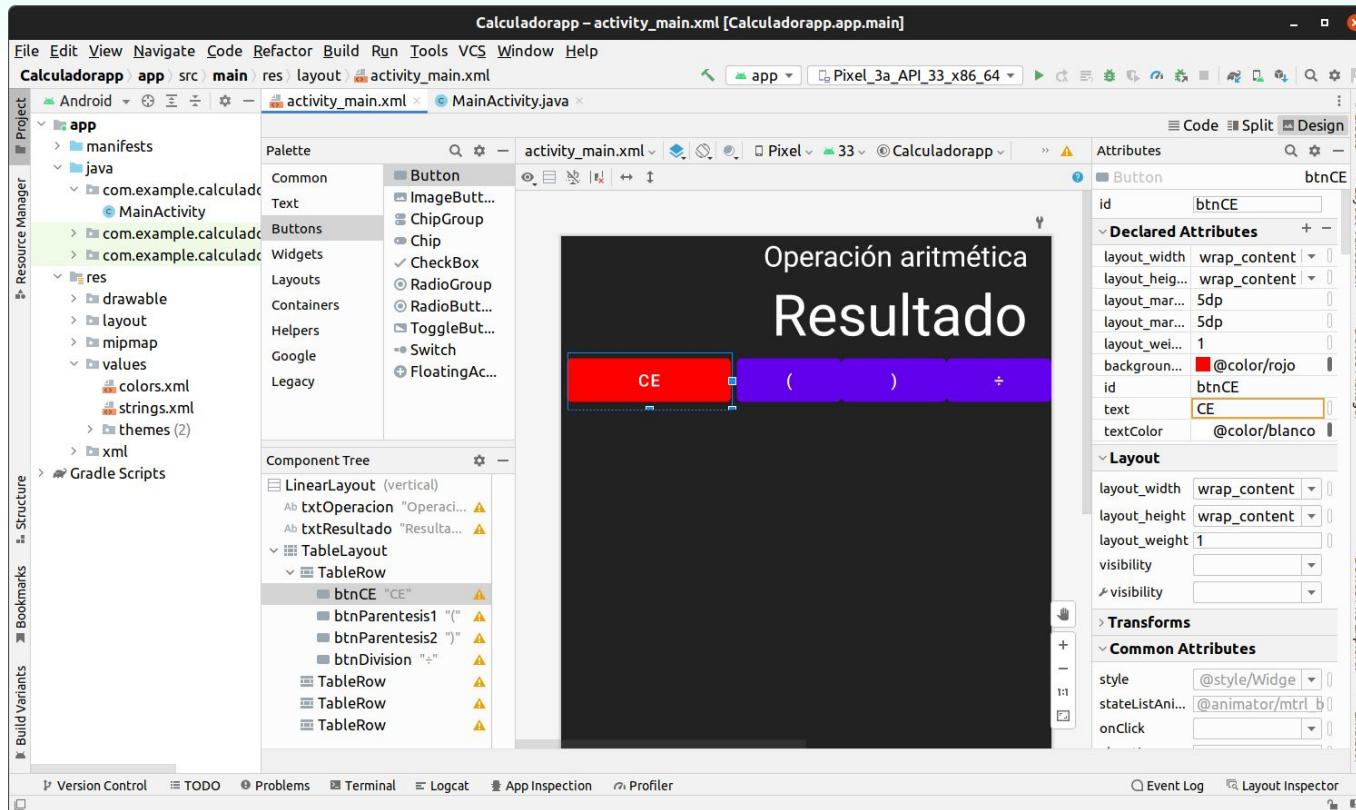
TextView 2



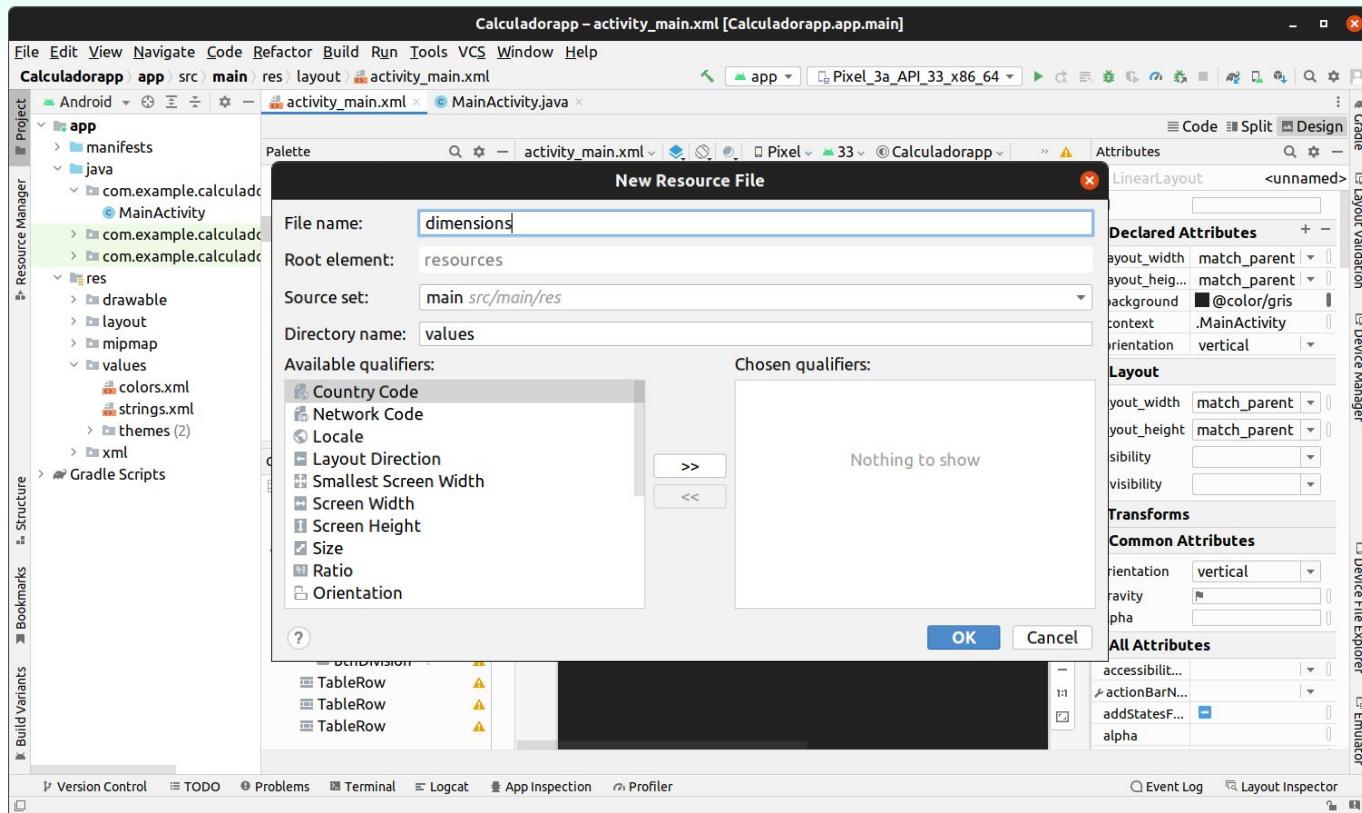
Botones en un TableLayout / TableRow



Personalizando los botones



Nuevo archivo de recursos



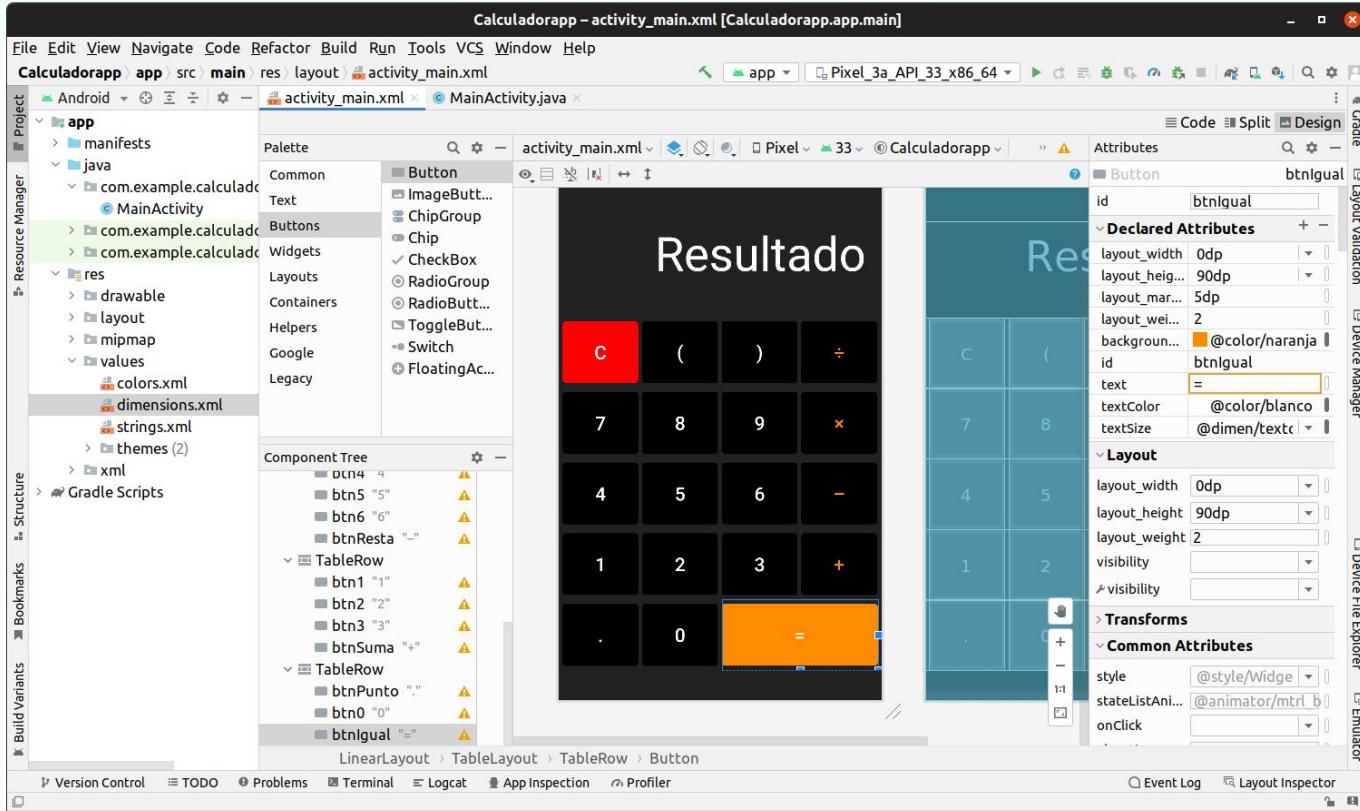
Tamaño de texto de los botones

The screenshot shows the Android Studio interface with the project 'Calculadorapp' open. The 'dimensions.xml' file is selected in the 'values' directory of the 'res' folder. The code editor displays the following XML:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <dimen name="texto_boton">24sp</dimen>
</resources>
```

The line containing the dimension declaration is highlighted with a yellow background. The 'Resource Manager' tool window on the left shows other files like 'colors.xml', 'strings.xml', and 'themes'. The bottom navigation bar includes tabs for Version Control, TODO, Problems, Terminal, Logcat, App Inspection, Profiler, Event Log, Layout Inspector, and Emulator.

Botones



Diseño final

Calculadorapp – activity_main.xml [Calculadorapp.app.main]

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help

Calculadorapp app src main res layout activity_main.xml

Project

Resource Manager

Build Variants Bookmarks

Common Text View

Text Buttons

Widgets RecyclerVi...

Layouts FragmentC...

Containers ScrollView

Helpers

Google

Legacy

Component Tree

LinearLayout (vertical)

Ab txtOperacion "Operación aritmética"

Ab txtResultado "Resultado"

TableLayout

TableRow

btnCE "(c)"

btnParentesis1 "("

btnParentesis2 ")"

btnDivision "÷"

TableRow

btn7 "7"

btn8 "8"

btn9 "9"

0 .

=

Attributes

id: txtOperacion

Declared Attributes

layout_width: match_parent

layout_height: 0dp

layout_margin: 20dp

layout_weight: 1

id: txtOperacion

text: Operación aritmética

textAlignment: textEnd

textColor: @color/blanco

textSize: 24sp

Layout

layout_width: match_parent

layout_height: 0dp

layout_weight: 1

visibility: visible

visibility: visible

Transforms

Common Attributes

text: Operación aritmética

text:

contentDesc:

textAppears: @android:style/TextAppearance

Layout Validation Device Manager

Device File Explorer Emulator

33_x86_64

Code Split Design

Attributes

id: txtResultado

Declared Attributes

layout_width: match_parent

layout_height: 0dp

layout_margin: 20dp

layout_weight: 1

id: txtResultado

text: Resultado

textAlignment: textEnd

textColor: @color/blanco

textSize: 60sp

Layout

layout_width: match_parent

layout_height: 0dp

layout_weight: 1

visibility: visible

visibility: visible

Transforms

Common Attributes

text: Resultado

text:

contentDesc:

textAppears: @android:style/TextAppearance

Layout Validation Device Manager

Device File Explorer Emulator

Event Log Layout Inspector

Version Control TODO Problems Terminal Logcat App Inspection Profiler

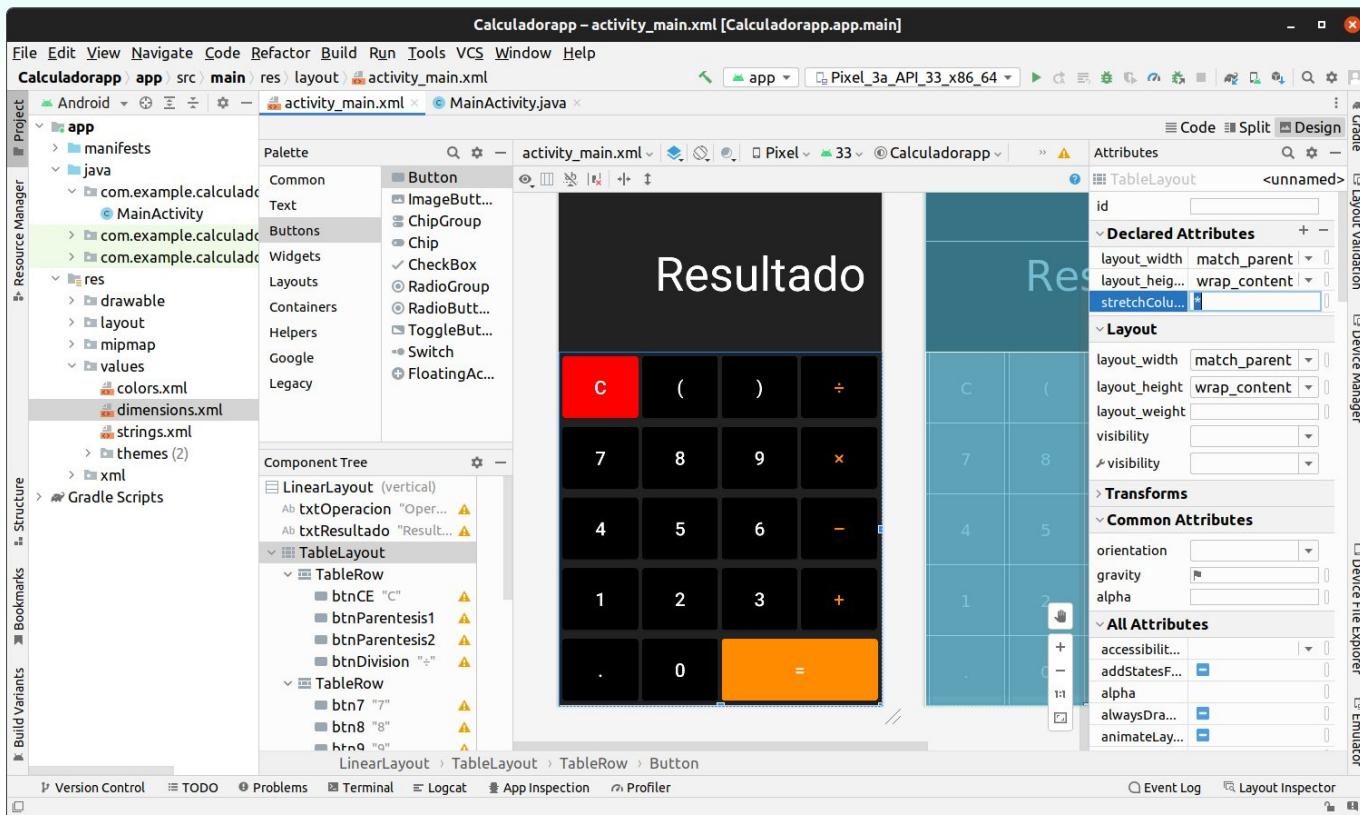
* daemon started successfully (2 minutes ago)

Event Log Layout Inspector

Version Control TODO Problems Terminal Logcat App Inspection Profiler

* daemon started successfully (a minute ago)

Diseño final



1. Acceso a los TextView

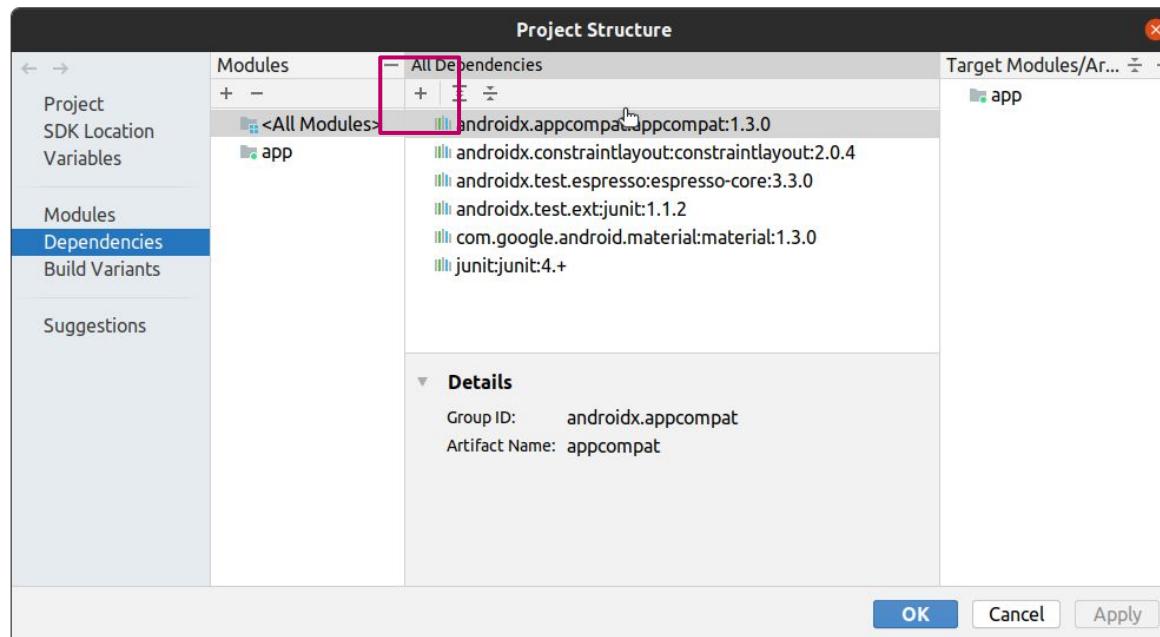
```
public class MainActivity extends AppCompatActivity {  
    TextView txtOperacion;  
    TextView txtResultado;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        txtOperacion = findViewById(R.id.txt_operacion);  
        txtResultado = findViewById(R.id.txt_resultado);  
        txtOperacion.setText("");  
        txtResultado.setText("");  
    }  
}
```

2. Funcionalidad de los Button

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
    ...  
    Button btnCE;  
    Button btn1;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        ...  
        btnCE = findViewById(R.id.btn_CE);  
        btn1 = findViewById(R.id.btn_1);  
        btnCE.setOnClickListener(this);  
        btn1.setOnClickListener(this);  
    }  
    private void setOperando(String numero) {  
        String s = txtOperacion.getText().toString();  
        txtOperacion.setText(s + numero);  
    }  
    @Override  
    public void onClick(View view) {  
        if (view == btn1) {  
            this.setOperando("1");  
        }  
    }  
}
```

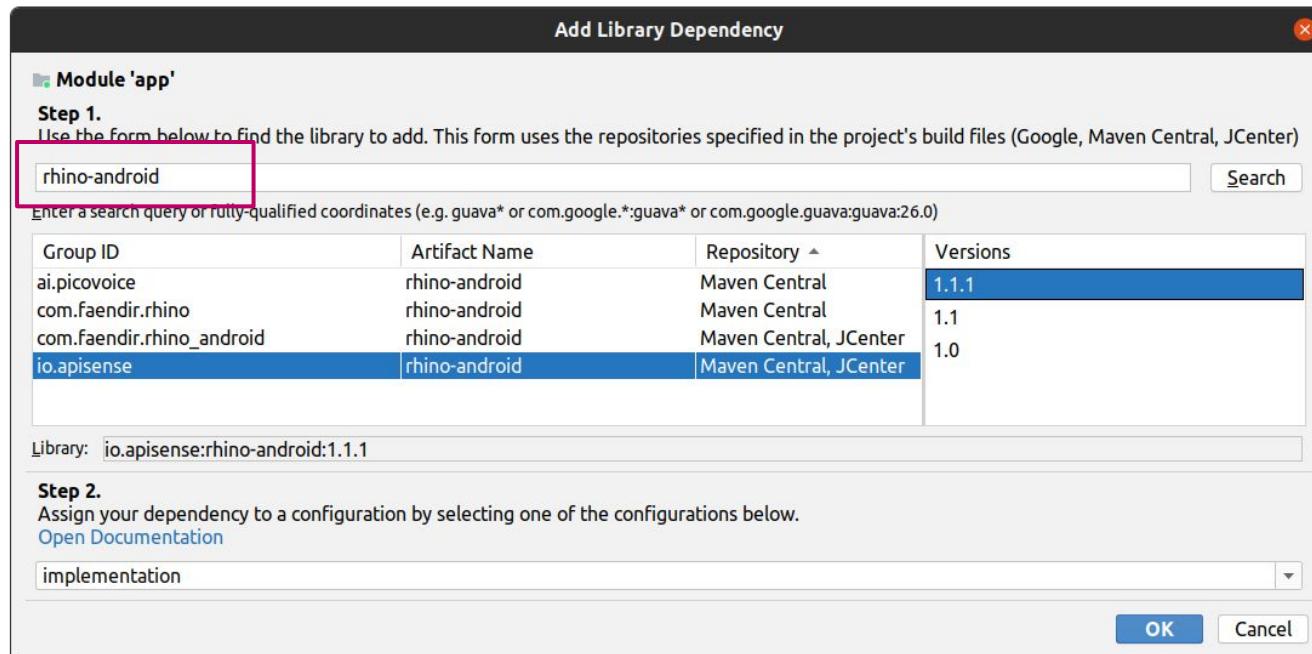
3. Importar biblioteca externa

- *File → Project structure*



3. Importar biblioteca externa

- *Library Dependency*



4. Calcular resultado

```
private void calcularResultado() {
    String operacion = txtOperacion.getText().toString();
    Double resultado = null;
    ScriptEngine motor = new ScriptEngineManager().getEngineByName("rhino");
    try {
        resultado = (Double)motor.eval(operacion);
    } catch (ScriptException e) {
        Toast.makeText(this, "Operación no válida", Toast.LENGTH_SHORT).show();
    }
    if (resultado != null) {
        txtResultado.setText(resultado.toString());
    }
}
```

¡Listo!

1. Instalamos Android Studio
2. ¡Hola, Mundo!
3. Calculadoraapp 