

## Desarrollo de Aplicaciones Multiplataforma

Programación multimedia y  
dispositivos móviles



## Índice

Que quiero que me entreguéis.....	3
El fichero Word.....	3
Fichero comprimido.....	3
Arquitectura .....	5
Flujo de trabajo .....	5
Añadir otros emuladores .....	5
Gestión de errores y logs.....	10
Donde se encuentran los logs.....	10
ERROR: Android Emulator Screen Size not the same as device frame.....	10
Errores con el emulador .....	11
Pregunta de la sesión de hoy .....	11
Veamos las partes de Android Studio.....	12
Estructura de ficheros.....	12
Componentes android studio.....	13
Activity_main.xml .....	13
Layout .....	14
Biblioteca de vinculación de datos (Data binding) .....	14
Empecemos a practicar .....	15
Practica HolaCaracola.....	15
Refuerzo de la detección de errores .....	16
Pregunta de la sesión.....	18
Practicar con el layout.....	18
Práctica conversor de bytes-Diseño .....	31
Programar el botón y gestionar valores de ítems .....	37
Gestion de spinners / lista de valores .....	40
Rellenar spinner a través de código .....	40
Rellenar spinner a través de un recurso .....	42
Cómo usar los valores del spinner .....	44
Cálculo de conversión.....	45



## Que quiero que me entreguéis

Básicamente necesitaré dos ficheros: un Word y un fichero comprimido

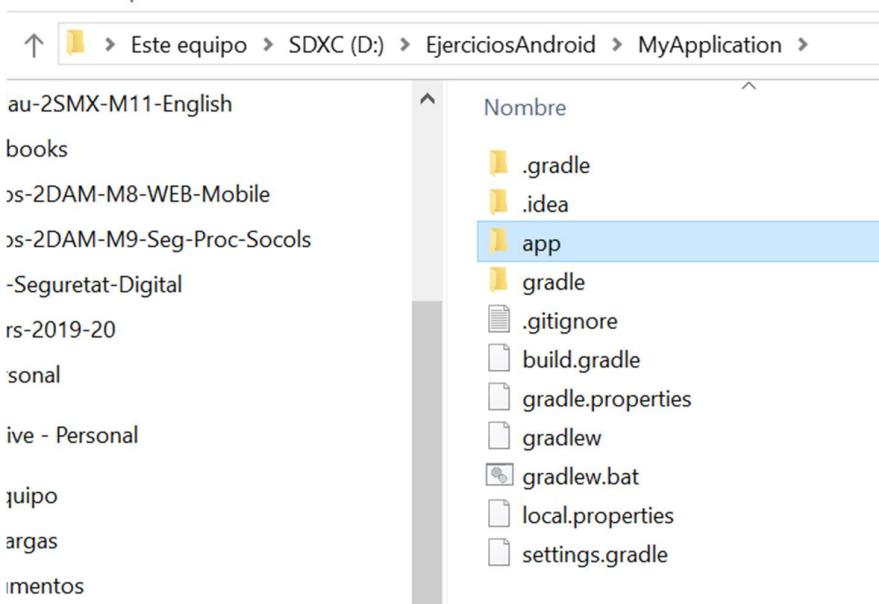
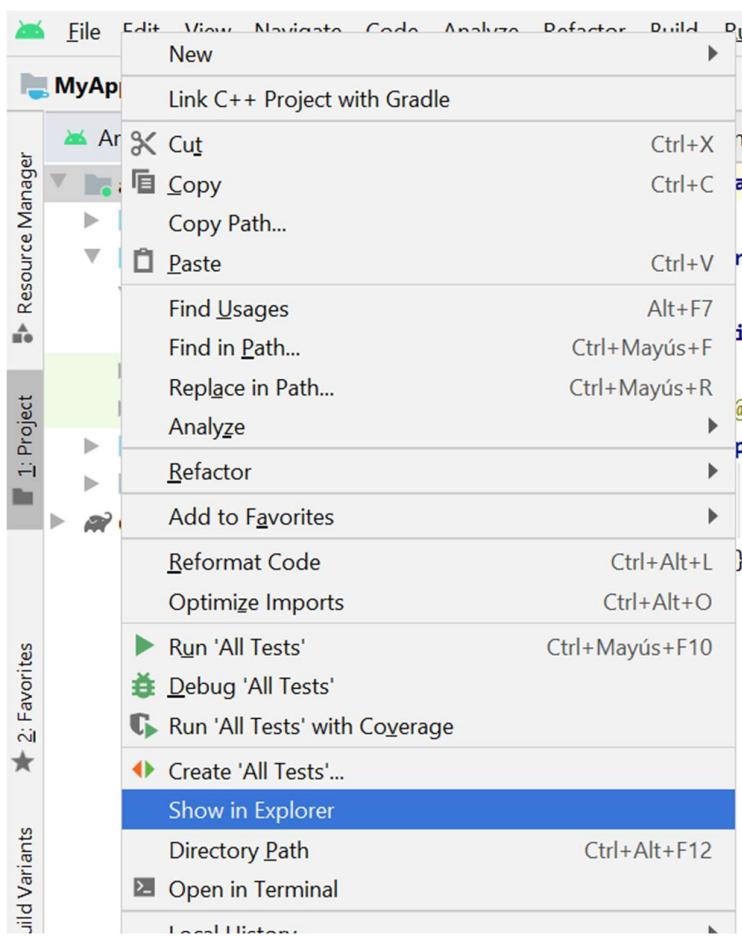
### El fichero Word

Tenéis que usar la plantilla que os pide la actividad y en ella me tendréis que copiar el contenido de los siguientes ficheros:

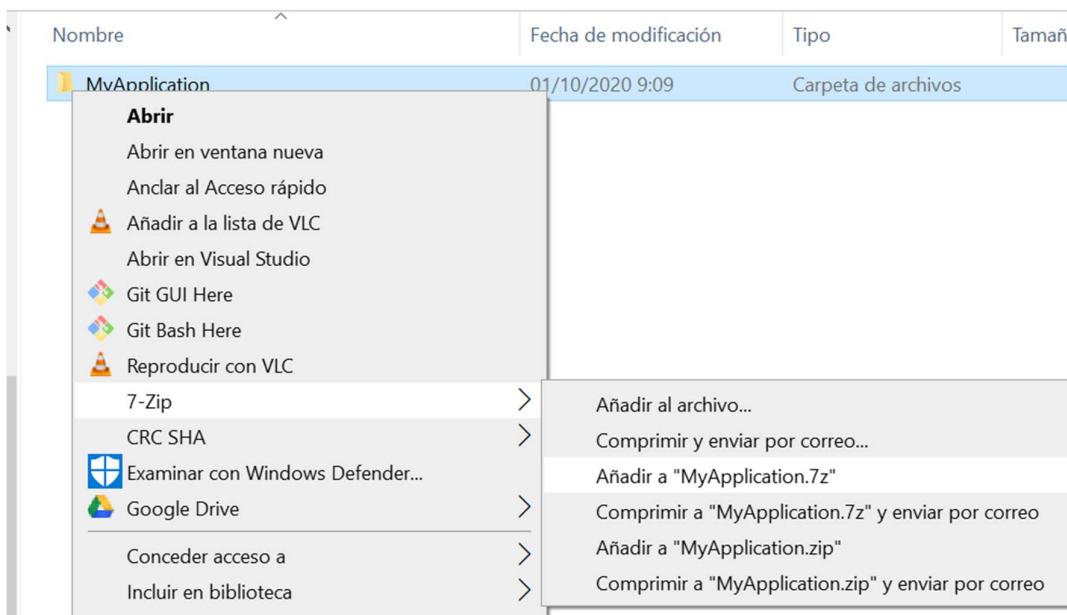
- AndroidManifest.xml
- Activity\_main.xml. Id a la sección de “Code” y copiad el XML
- Strings.xml
- Cualquier XML que hayáis añadido
- MainActivity.java
- Capturas de pantallas en donde se pueda ver la ejecución correcta del emulador

### Fichero comprimido

Una vez finalizado el trabajo, me tenéis que comprimir la carpeta en donde está el proyecto. Para ello si apretáis botón derecho sobre “App” si usáis la visión de “Android” podréis ir al directorio en donde se ha guardado



Os posicionáis sobre el nivel superior y comprimid el directorio entero



Los dos ficheros se han de subir en el Moodle.

## Arquitectura

<https://developer.android.com/guide/platform?hl=es-419>

## Flujo de trabajo

<https://developer.android.com/studio/workflow?hl=es-419>

## Añadir otros emuladores

Instalar el emulador escogido en tu ordenador

GenyMotion es gratuito para uso personal



## Programación multimedia y dispositivos móviles

The screenshot shows the Genymotion website's header with various navigation links. Below the header, there are three main service offerings represented by icons and descriptions:

- PaaS (Platform as a Service)**: Represented by a cloud icon with a shopping cart and a hexagon. Description: \$0.5 per hour per device (+cloud providers fees).
- SaaS (Software as a Service)**: Represented by a cloud icon with two circles. Description: \$0.05 per minute per device.
- Desktop**: Represented by a laptop icon with an Android logo. Description: \$412 / year (BUSINESS), \$136 / year (INDIE), Free for personal use.

Three options



### PaaS (Platform as a Service)

\$0.5 per hour per device  
(+cloud providers fees)

### SaaS (Software as a Service)

\$0.05 per minute per device

### Desktop

\$412 / year (BUSINESS)  
\$136 / year (INDIE)  
Free for personal use

## Crear una cuenta de GenyMotion

<https://www.genymotion.com/download/>

The screenshot shows the Genymotion download page. It features a large pink button in the center with the text "Download Genymotion Desktop 3.1.1". To the left of this button, there are two options for downloading the software:

- with VirtualBox:** A pink button labeled "Download for Windows - 254MB".
- without VirtualBox:** A pink button labeled "Download for Windows - 36MB".

To the right of the download buttons, there is a section titled "System Requirements" which lists the following requirements:

- Microsoft Windows 8, 8.1, 10 (32/64 bit)
- 64 bit CPU, with VT-x or AMD-V/SVM capability enabled in BIOS settings
- Hardware accelerated GPU
- 400 MB disk space
- 4GB RAM
- VirtualBox 6.0.4

Below the system requirements, there are two links: "Checksum (with VirtualBox)" and "Checksum (without VirtualBox)".

## Download Genymotion Desktop 3.1.1

This image provides a detailed view of the download options for Genymotion Desktop 3.1.1. On the left, there is a pink box containing the following information:

- A cloud icon with a downward arrow.
- The text "with VirtualBox:" followed by a pink button labeled "Download for Windows - 254MB".
- The text "without VirtualBox:" followed by a pink button labeled "Download for Windows - 36MB".
- A link labeled "How to register my license".

On the right, there is a white box titled "System Requirements" which lists the following system requirements:

- Microsoft Windows 8, 8.1, 10 (32/64 bit)
- 64 bit CPU, with VT-x or AMD-V/SVM capability enabled in BIOS settings
- Hardware accelerated GPU
- 400 MB disk space
- 4GB RAM
- VirtualBox 6.0.4

Below the system requirements, there are two links: "Checksum (with VirtualBox)" and "Checksum (without VirtualBox)".

## Instalación siguiente-siguiente

Al abrir te has de conectar con tu cuenta creada previamente



## Genymotion requires a license



**Use of Genymotion requires a license**

Genymotion is a professional tool for which all kinds of profit-making businesses need a valid license. A very light version of Genymotion is available without a license, but strictly restricted to a personal use.

[Buy a license \(if you don't already have one\)](#)

I have a license

Personal Use

[BACK](#) [NEXT](#)

?

Añadir dispositivos:

### Virtual device installation

[Filters](#) [X](#)

[Search](#)

Type	Device	Android API	Size	Density	Source	⋮
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	Google Nexus 4	4.4 - API 19	768 x 1280	320 - XHDPI	Genymotion	ⓘ
...	Google Nexus 5	4.4 - API 19	1080 x 1920	480 - XXHDPI	Genymotion	ⓘ
...	Google Nexus 7	4.4 - API 19	800 x 1280	213 - TVDPI	Genymotion	ⓘ

[CANCEL](#) [NEXT](#)



### Name

Google Nexus 5

### Display

Predefined

1080 x 1920 ▾

480 - XXHDPI ▾

Custom

Start in full-screen mode

### System

Android version

4.4

Memory size

2048

### Android system options

Show Android navigation bar

Use virtual keyboard for text input

### Network mode

NAT (default)

Bridge [?](#)



**Device installed**

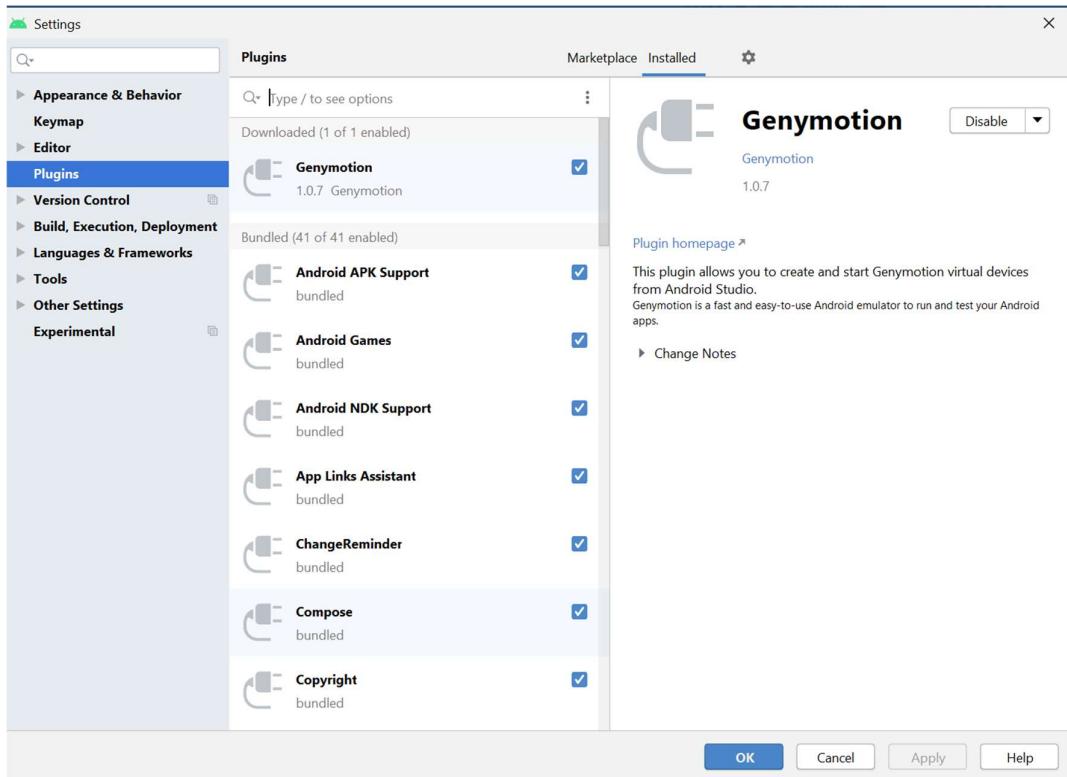


Device "Google Nexus 5" has been installed.

**START**



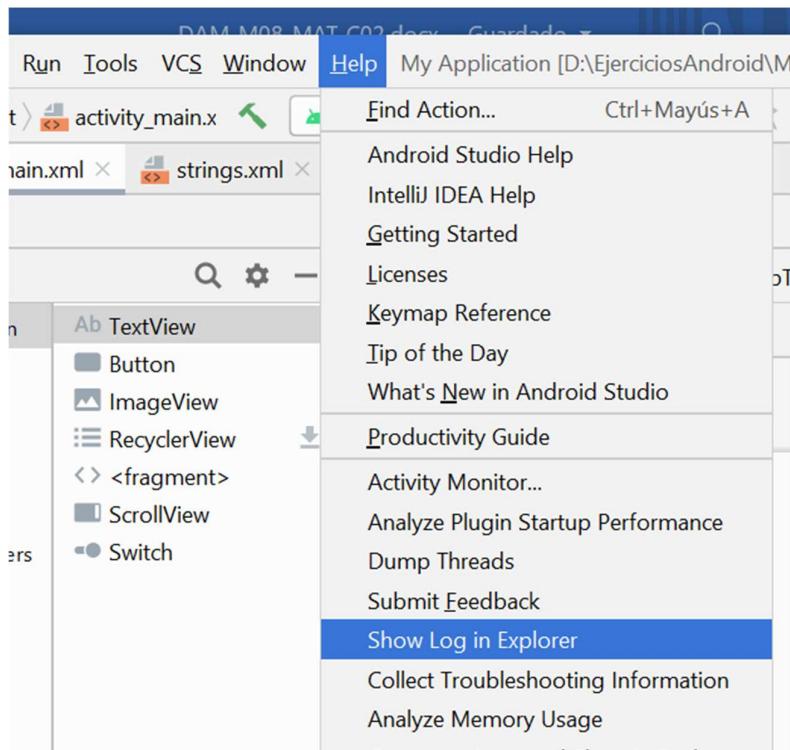
## Añadir el emulador escogido como plugin





## Gestión de errores y logs

Donde se encuentran los logs



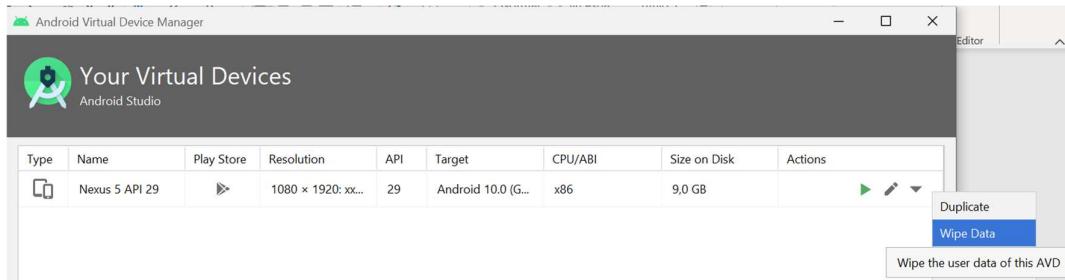
ERROR: Android Emulator Screen Size not the same as device frame

<https://stackoverflow.com/questions/44013216/android-emulator-screen-size-not-the-same-as-device-frame>

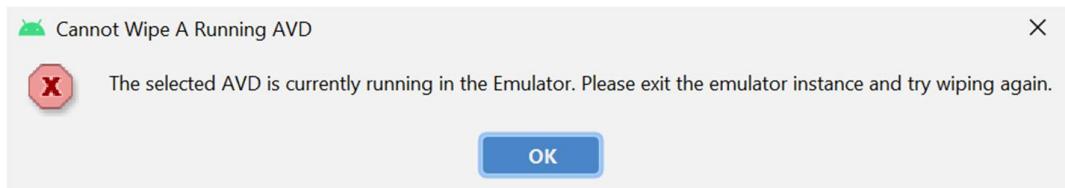


## Errores con el emulador

Si tenéis errores con el emulador lo mejor es limpiar la cache. Para ello nos vamos al AVG manager y limpiamos cache “Wipe Data”



Si nos aparece este error



Tendremos que matar la sesión desde el SO y/o reiniciar el Android Studio y/o reiniciar nuestro PC.

## Pregunta de la sesión de hoy

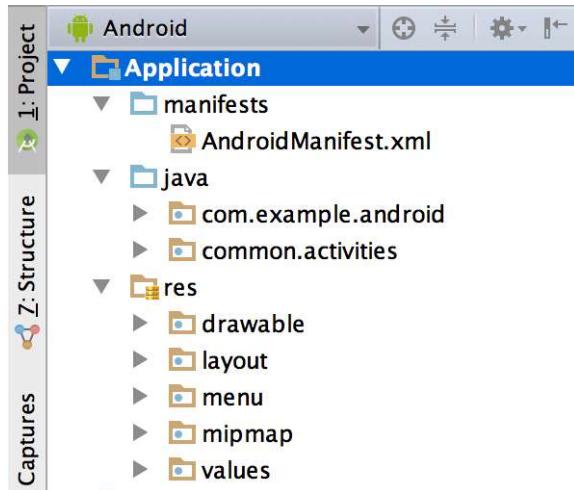
Explicar cómo borrar proyecto

<https://www.youtube.com/watch?v=daF2mNC3y3s>



## Veamos las partes de Android Studio

### Estructura de ficheros



- Carpeta manifests: Contiene el archivo AndroidManifest.xml. Con configuración de quien es nuestra actividad principal. En “label” definimos el nombre de la aplicación
- Carpeta java: Contiene los archivos de código fuente Java, incluido el código de prueba de Junit, para test unitarios. Es la sección en donde crearemos nuevas clases.
- Carpeta res: Contiene todos los recursos sin código, como diseños XML, strings de IU e imágenes de mapa de bits.
  - Drawable: contiene los gif, jpg, ...
  - Layout: donde se localizan las vistas
  - Mipmap: se encuentra el ícono de nuestra aplicación. Aquí vemos que tenemos un ic\_launcher y después un ic\_launcher\_round porque en Android 7 es donde necesitamos iconos redondos. En las versiones anteriores con nuestro ícono normal basta
  - Values:
    - Color
    - String
    - Styles
- Gradle
  - Build.Gradle (Module):
    - compileSdkVersion: versión de SDK que usa para compilar
    - minSdkVersion: SDK mínimo compatible
    - targetSdkVersion: SDK máximo compatible
    - buildTypes: información de la reléase o firmar una aplicación u optimizarla para que ocupe menos espacio
    - dependencies

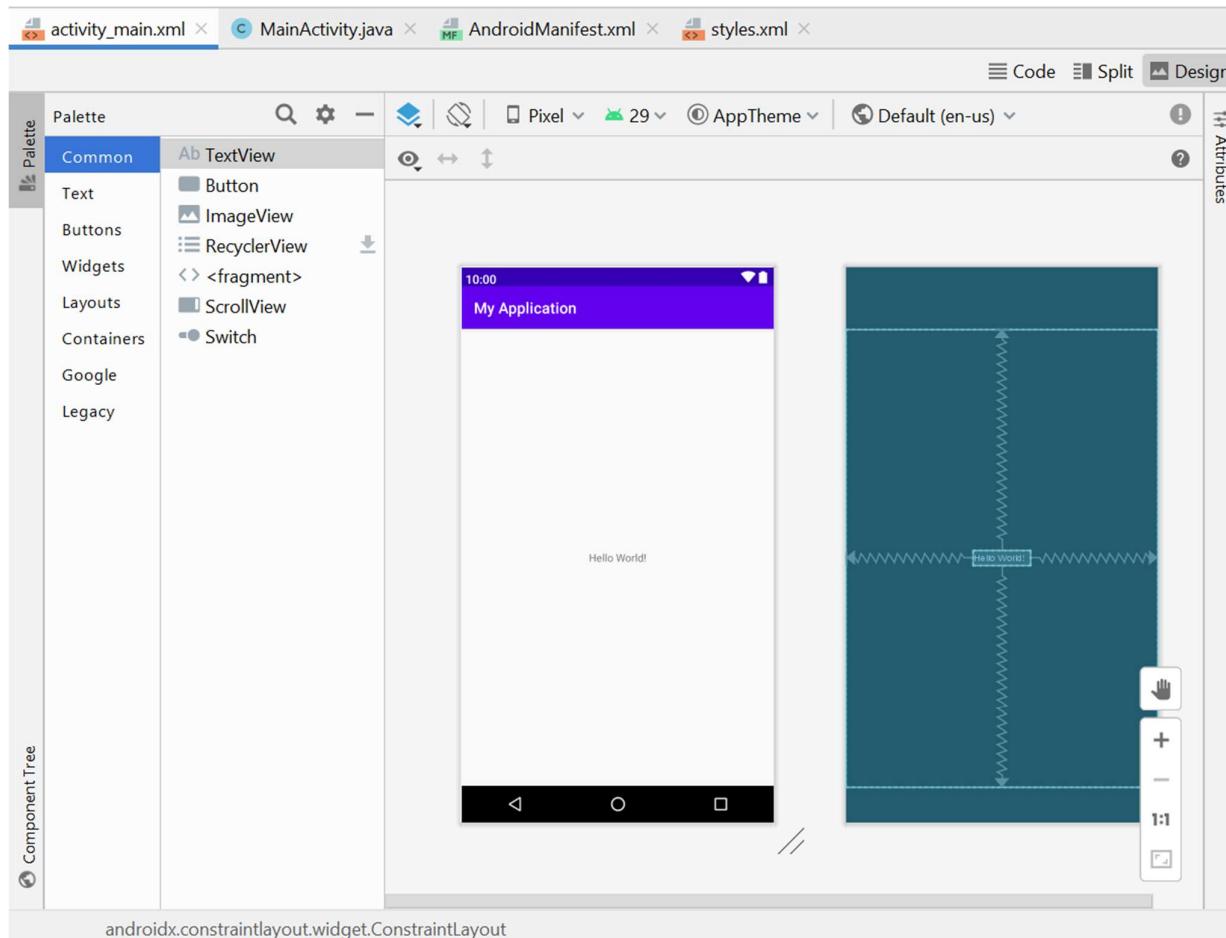
**Truco:** uso del control para localizar elementos dependientes entre diferentes ficheros



## Componentes android studio

- Activity: <https://developer.android.com/guide/components/activities/intro-activities>
- Fragment: <https://developer.android.com/guide/components/fragments?hl=es-419>
- Services: <https://developer.android.com/guide/components/services?hl=es-419>
- Content Provider: <https://developer.android.com/guide/topics/providers/content-providers?hl=es-419>
- Broadcast/Transcripciones: <https://developer.android.com/guide/components/broadcasts>
- Intent: <https://developer.android.com/guide/components/intents-filters?hl=es-419>

## Activity\_main.xml





## Layout

<https://developer.android.com/guide/topics/ui/declaring-layout?hl=es-419>

Usan formato XML

- View vs ViewGroup
  - View se suelen llamar “widgets” y pueden ser texto, botones, ...
  - Los ViewGroup son los Layout/diseños y los principales son
    - Linear Layout: horizontal y vertical
    - Relative Layout
    - Constraint Layout
    - Frame Layout
- Formato genérico para los tags de XML:  
<https://developer.android.com/guide/topics/resources/layout-resource?hl=es-419>

Biblioteca de vinculación de datos (Data binding)

<https://developer.android.com/topic/libraries/data-binding?hl=es>



## Empecemos a practicar

Practica HolaCaracola

Ctrl + May + F

En res/values/strings.xml se cambia el “app\_name”

The screenshot shows the 'Find in Path' search interface in Android Studio. The search term 'holaca' is entered. The search scope is set to 'Directory'. The results show the string definition in strings.xml:

```
<string name="app_name">HolaCaracola</string>
android:text="HolaCaracola!"
```

The screenshot shows the contents of strings.xml in code editor mode. The string definition is highlighted:

```
1 <resources>
2   <string name="app_name">HolaCaracola</string>
3 </resources>
```



En /res/layout/activity\_mail.xml se modifica el TextView correspondiente la propiedad Text

Find in Path  Match case  Words  Regex ?  File mask: \*.java

holaca 2 matches

In Project Module Directory Scope

src\Android\MyApplication\app\src\main\res\values\strings.xml

```
<string name="app_name">HolaCaracola</string>
android:text="HolaCaracola!"
```

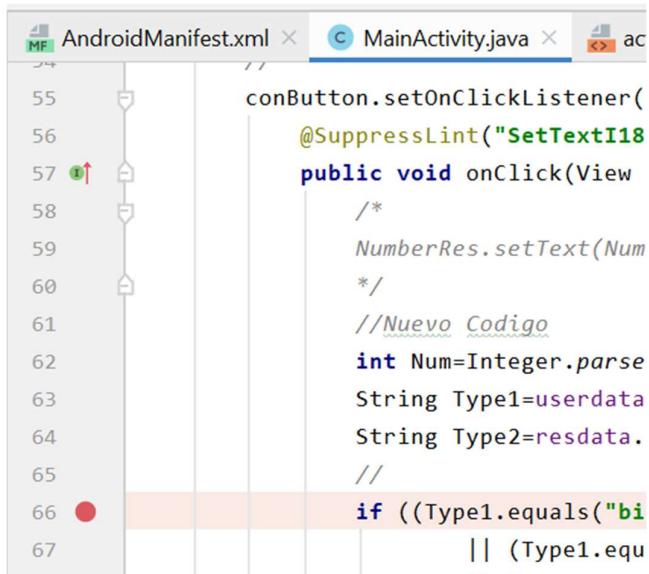
activity\_main.xml app/src/main/res/layout

```
8
9     <TextView
10        android:layout_width="wrap_content"
11        android:layout_height="wrap_content"
12        android:text="HolaCaracola!"
13        app:layout_constraintBottom_toBottomOf="parent"
14        app:layout_constraintLeft_toLeftOf="parent"
15        app:layout_constraintRight_toRightOf="parent"
16        app:layout_constraintTop_toTopOf="parent" />
```

## Refuerzo de la detección de errores

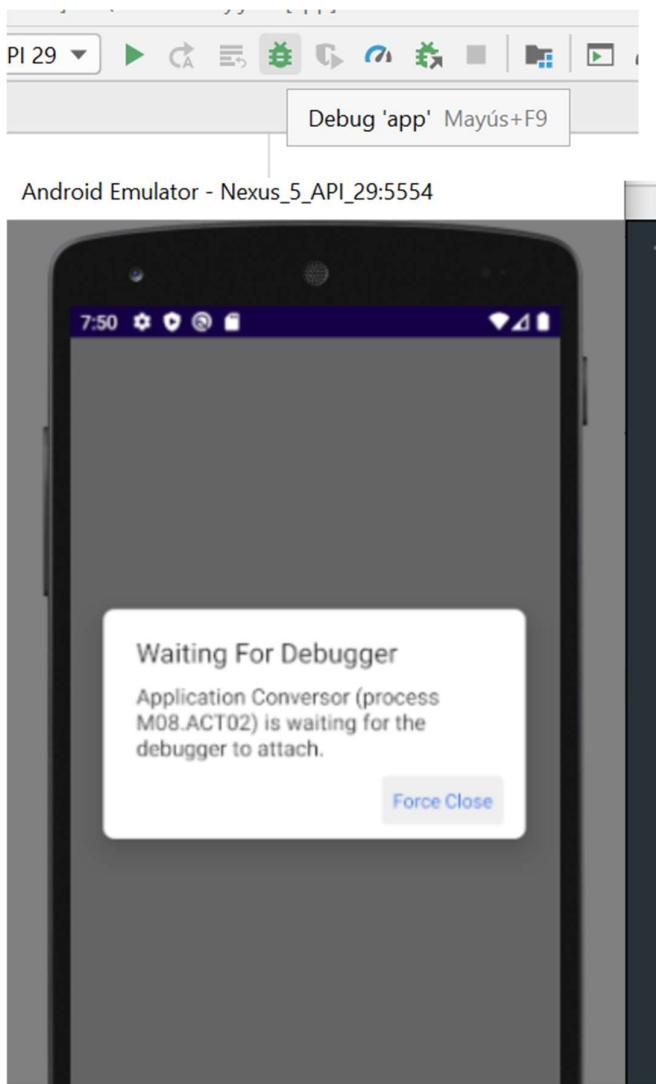
A la hora de detectar mejor los errores va muy bien borrar todos los mensajes previos y ejecutar de nuevo el emulador.

También va muy bien el poder usar el modo debug para saber por donde va el programa



```
AndroidManifest.xml x MainActivity.java x ac
55     conButton.setOnClickListener(
56         @SuppressLint("SetTextI18n")
57         public void onClick(View
58             /*
59                 NumberRes.setText(Num
60             */
61             //Nuevo Código
62             int Num=Integer.parse
63             String Type1=userdata
64             String Type2=resdata.
65             /*
66             if ((Type1.equals("bi
67                 || (Type1.equals("

The code handles a button click event, sets text to a resource, and performs integer parsing from user and resource data. A comment indicates a new code addition.
```



Os paso un enlace de youtube que os puede ir bien:



[https://www.youtube.com/watch?v=k9rOy-qH\\_rY](https://www.youtube.com/watch?v=k9rOy-qH_rY)

## Pregunta de la sesión

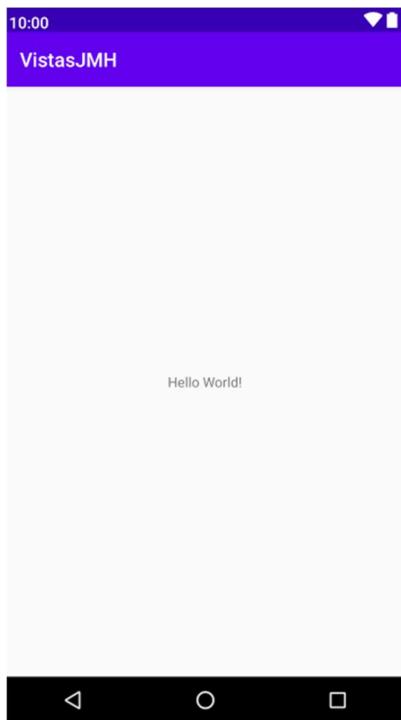
Explicar para que sirve el modo debug

## Practicar con el layout

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



```
    android:layout_width="match_parent" // se ajusta al parent
    android:layout_height="wrap_content" // se ajusta al contenido
```

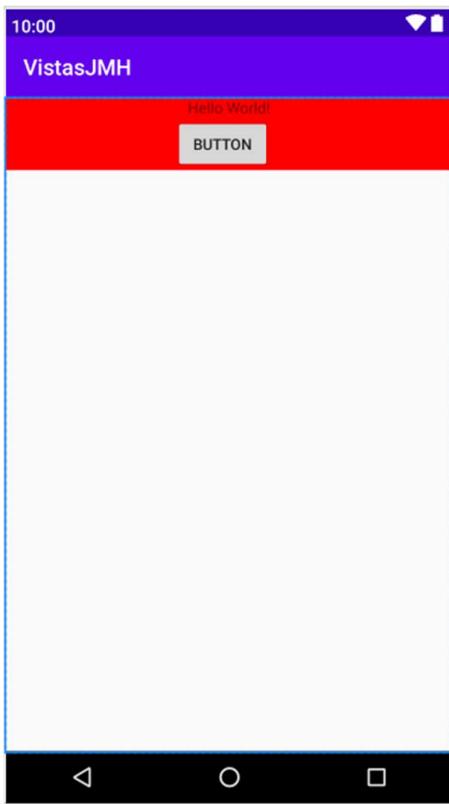
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#f00">

        <TextView
            android:id="@+id/tvSaludo"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello World!"
            android:layout_centerHorizontal="true"/>

        <Button
            android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button"
            android:layout_alignRight="@+id/tvSaludo"
            android:layout_below="@+id/tvSaludo"/>

    </RelativeLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#f00">

        <TextView
            android:id="@+id/tvSaludo"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello World!"
            android:layout_centerHorizontal="true"/>

        <Button
            android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button"
            android:layout_alignRight="@+id/tvSaludo"
            android:layout_below="@+id/tvSaludo"/>

        <LinearLayout
            android:id="@+id/llMain"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="#0f0">
    
```



```
    android:layout_below="@+id/button"
    android:orientation="horizontal">

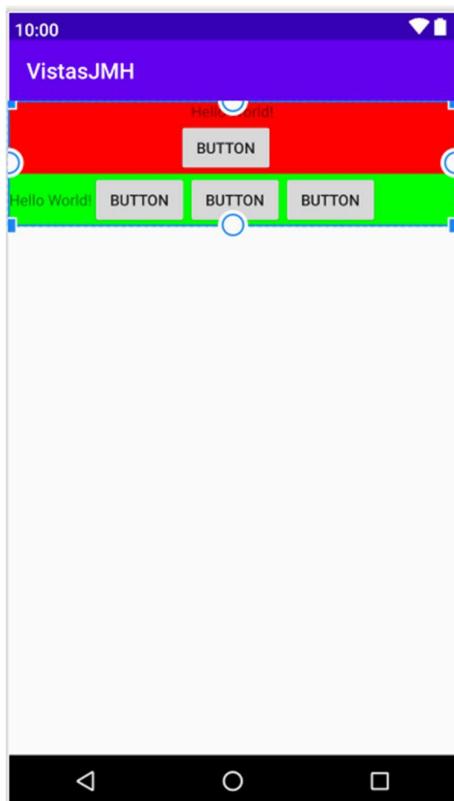
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"/>

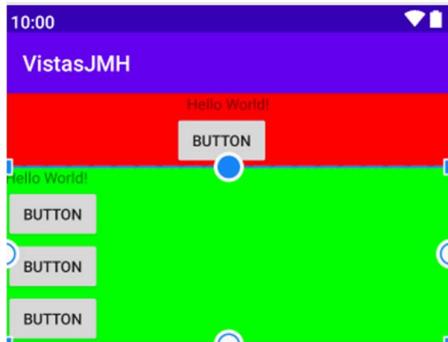
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"/>
</LinearLayout>

</RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```



Si cambio el linear layout la orientación a vertical me aparece:



En la sección verde, modifíco el segundo botón para ponerlo al centro y añado un “Framelayout”

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#f00">

        <TextView
            android:id="@+id/tvSaludo"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello World!"
            android:layout_centerHorizontal="true"/>

        <Button
            android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button"
            android:layout_alignRight="@+id/tvSaludo"
            android:layout_below="@+id/tvSaludo"/>

        <LinearLayout
            android:id="@+id/llMain"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="#0f0"
            android:layout_below="@+id/button"
            android:orientation="vertical">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
```



```
        android:text="Hello World! "/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"
        android:layout_gravity="center"/>

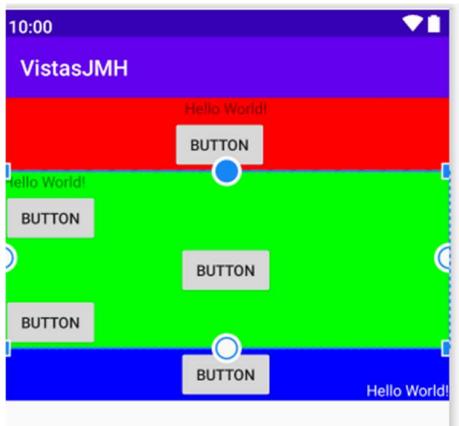
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"/>
</LinearLayout>

<FrameLayout
    android:id="@+id/flMain"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#00f"
    android:layout_below="@id/llMain">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:textColor="#fff"
        android:layout_gravity="bottom|right"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"
        android:layout_gravity="center_horizontal"/>
</FrameLayout>

</RelativeLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```



Añadimos más elementos

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#f00">

        <TextView
            android:id="@+id/tvSaludo"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello World!"
            android:layout_centerHorizontal="true"/>

        <Button
            android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button"
            android:layout_alignRight="@+id/tvSaludo"
            android:layout_below="@+id/tvSaludo"/>

        <LinearLayout
            android:id="@+id/llMain"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="#0f0"
            android:layout_below="@+id/button"
            android:orientation="vertical">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Hello World!"/>

            <Button>
```



```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"
        android:layout_gravity="center"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"/>
</LinearLayout>

<FrameLayout
    android:id="@+id/flMain"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#00f"
    android:layout_below="@+id/l1Main">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:textColor="#fff"
        android:layout_gravity="bottom|right"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"
        android:layout_gravity="center_horizontal"/>
</FrameLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:background="#fff"
    android:gravity="center"
    android:layout_below="@+id/flMain">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Texto de prueba"
        android:textColor="@android:color/holo_blue_light"
        android:textAllCaps="true"
        android:gravity="center_horizontal"
        android:textAppearance="?android:attr/textAppearanceLarge"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="introduce tu nombre"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Guardar"
```



```
        android:textAllCaps="false"/> >

    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@mipmap/ic_launcher"
        android:background="?attr/selectableItemBackgroundBorderless"/>

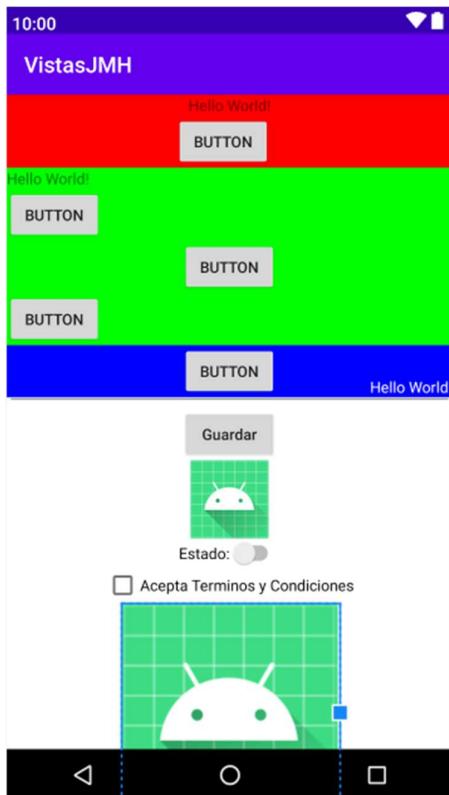
    <Switch
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Estado:"
        tools:ignore="UseSwitchCompatOrMaterialXml" />

    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Acepta Terminos y Condiciones"/>

    <ImageView
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:src="@mipmap/ic_launcher"/>
</LinearLayout>

</RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```





Ahora vamos a añadir dos linear layout dentro de otro y además vamos dividir la pantalla por la mitad con “layout\_weight”

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#f00"
        tools:ignore="MissingConstraints">

        <TextView
            android:id="@+id/tvSaludo"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello World!"
            android:layout_centerHorizontal="true"/>

        <Button
            android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button"
            android:layout_alignRight="@+id/tvSaludo"
            android:layout_below="@+id/tvSaludo"/>

        <LinearLayout
            android:id="@+id/llMain"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="#0f0"
            android:layout_below="@+id/button"
            android:orientation="horizontal">

            <LinearLayout
                android:id="@+id/llMain1"
                android:layout_weight="0.5"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:background="#0f0"
                android:orientation="vertical">

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Hello World!"/>

                <Button
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Button"/>
            
```



```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button"
    android:layout_gravity="center"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button"/>
</LinearLayout>
<LinearLayout
    android:id="@+id/l1Main2"
    android:layout_weight="0.5"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:background="#c6ce00"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"
        android:layout_gravity="center"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"/>
</LinearLayout>

</LinearLayout>

<FrameLayout
    android:id="@+id/flMain"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#00f"
    android:layout_below="@+id/l1Main">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:textColor="#fff"
        android:layout_gravity="bottom|right"/>

    <Button
        android:layout_width="wrap_content"
```



```
        android:layout_height="wrap_content"
        android:text="Button"
        android:layout_gravity="center_horizontal"/>
    
```

```
</FrameLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:background="#fff"
    android:gravity="center"
    android:layout_below="@+id/flMain">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Texto de prueba"
        android:textColor="@android:color/holo_blue_light"
        android:textAllCaps="true"
        android:gravity="center_horizontal"
        android:textAppearance="?android:attr/textAppearanceLarge"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="introduce tu nombre"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Guardar"
        android:textAllCaps="false"/>

    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@mipmap/ic_launcher"
        android:background="?attr/selectableItemBackgroundBorderless"/>

    <Switch
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Estado:"
        tools:ignore="UseSwitchCompatOrMaterialXml" />

    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Acepta Términos y Condiciones"/>

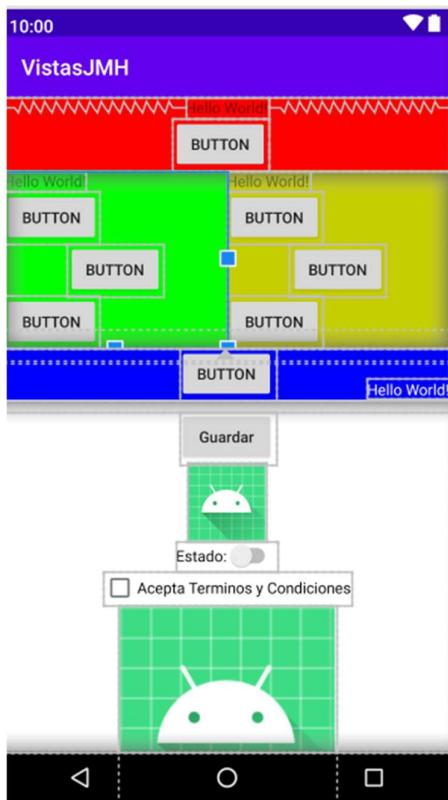
    <ImageView
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:src="@mipmap/ic_launcher"/>

```

```
</LinearLayout>

</RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

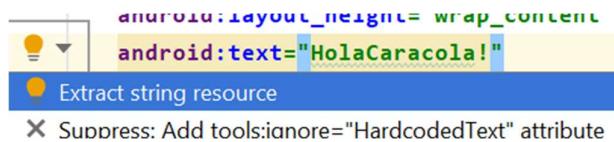


Los colores en hexadecimal se pueden encontrar en: <https://encycolorpedia.es/a4c639>

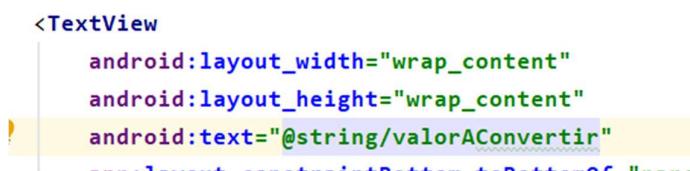
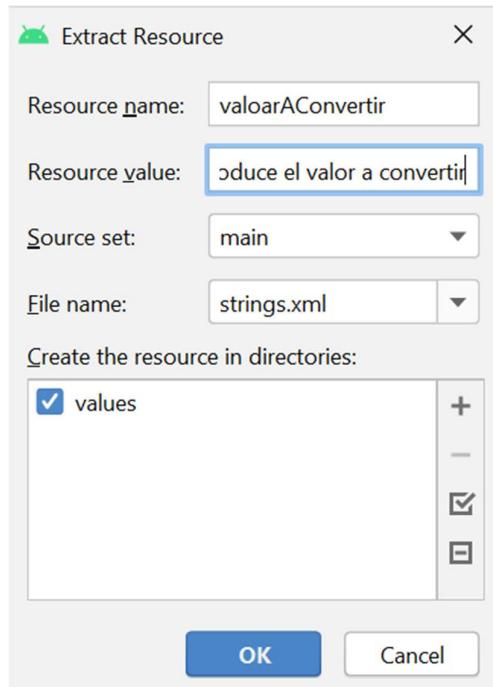
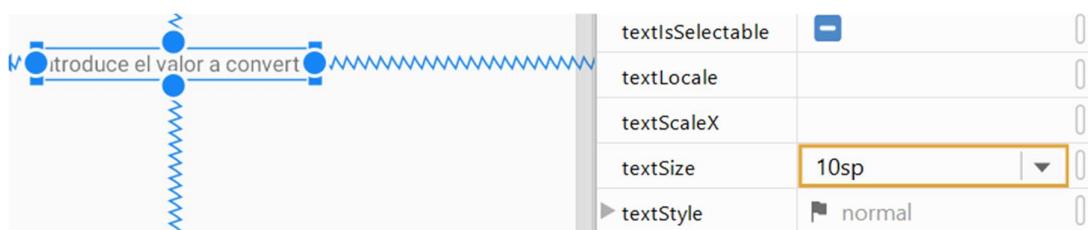


## Práctica conversor de bytes-Diseño

Primero empezar a usar los recursos de los “parámetros” o string para los textos.



Cambiamos el tamaño en la sección de propiedades



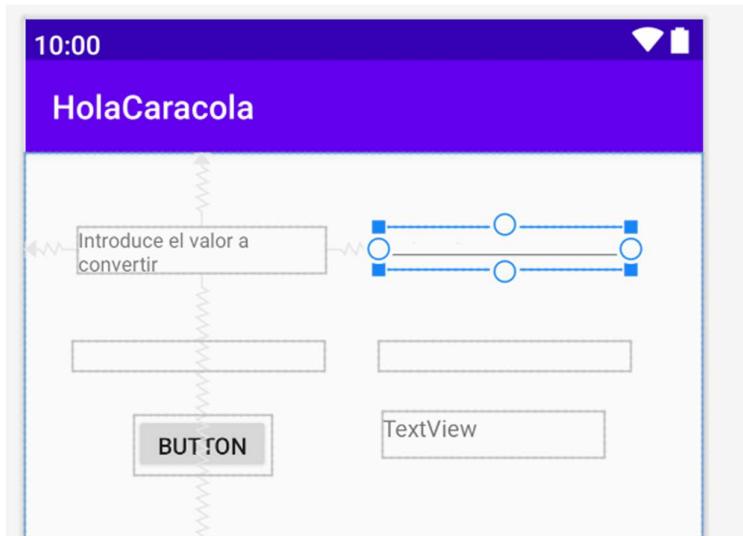


The screenshot shows the Android Studio interface with the strings.xml file open. The tab bar at the top includes activity\_main.xml, strings.xml (which is currently selected), MainActivity.java, AndroidManifest.xml, and styles.xml. A status bar message below the tabs says "Translations for all locales in the translations editor." The code in strings.xml is as follows:

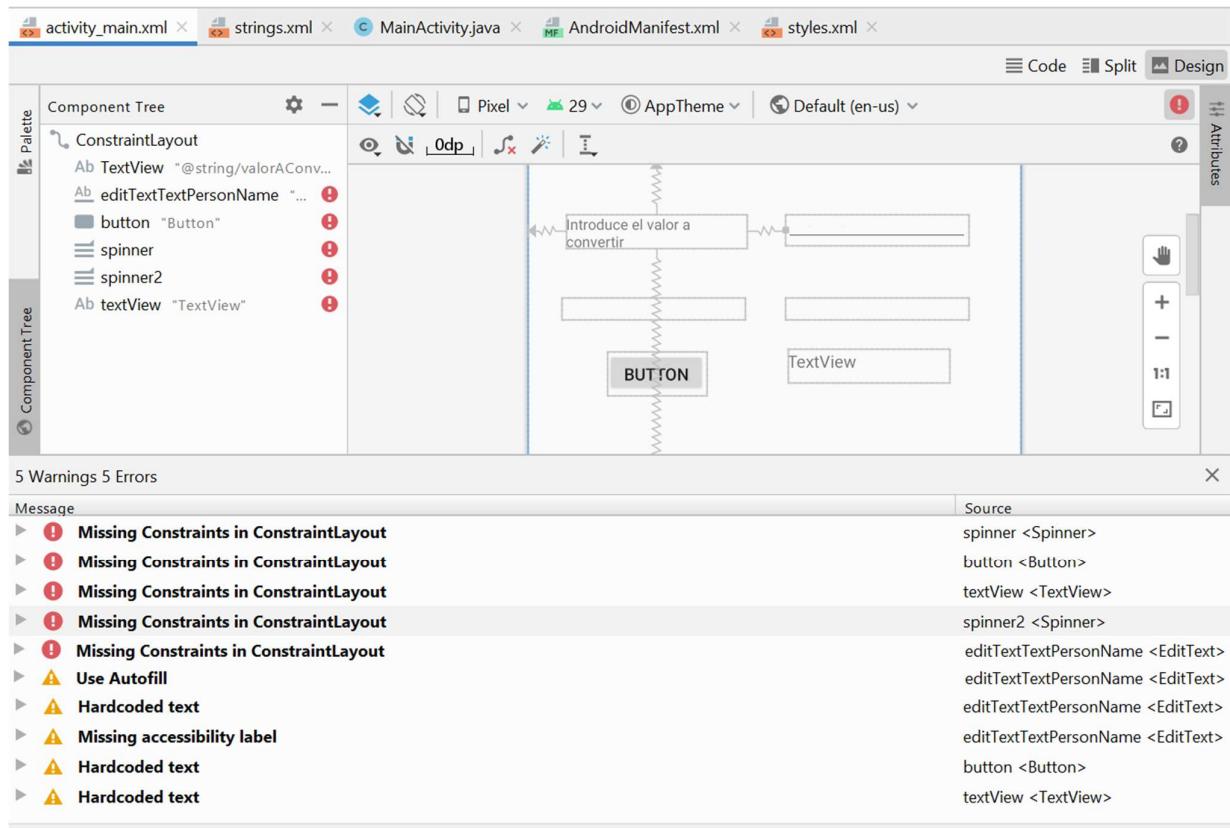
```
<resources>
    <string name="app_name">HolaCaracola</string>
    <string name="valorAConvertir">Introduce el valor a convertir</string>
</resources>
```



Añado elementos en la pantalla:

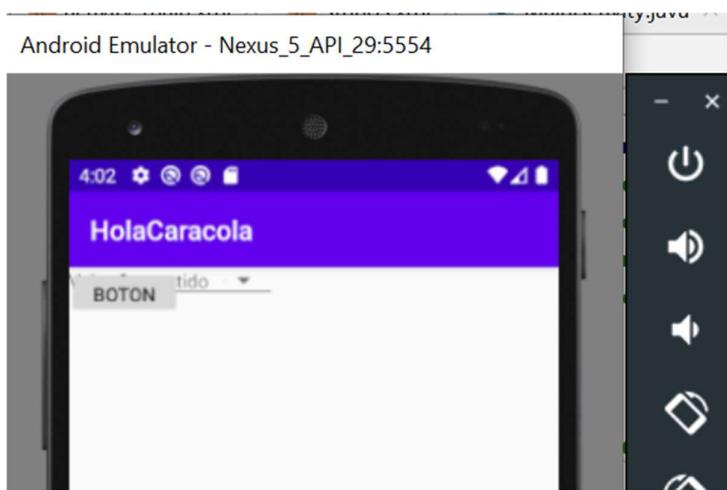


Y vemos que me aparecen varios avisos y errores



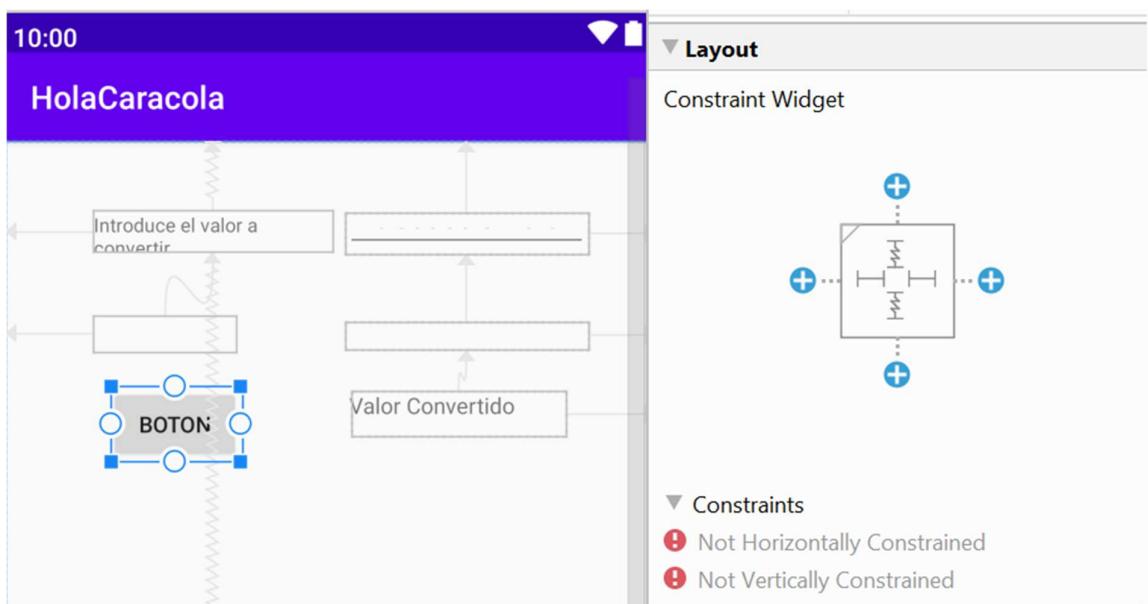
Los que son “Hardcode text” son los que he explicado anterioridad y son de tipo Warning (en amarillo), pero los errores (en rojo) se han de corregir.

En este caso si no lo corrojo y lo ejecuto me aparece lo siguiente:



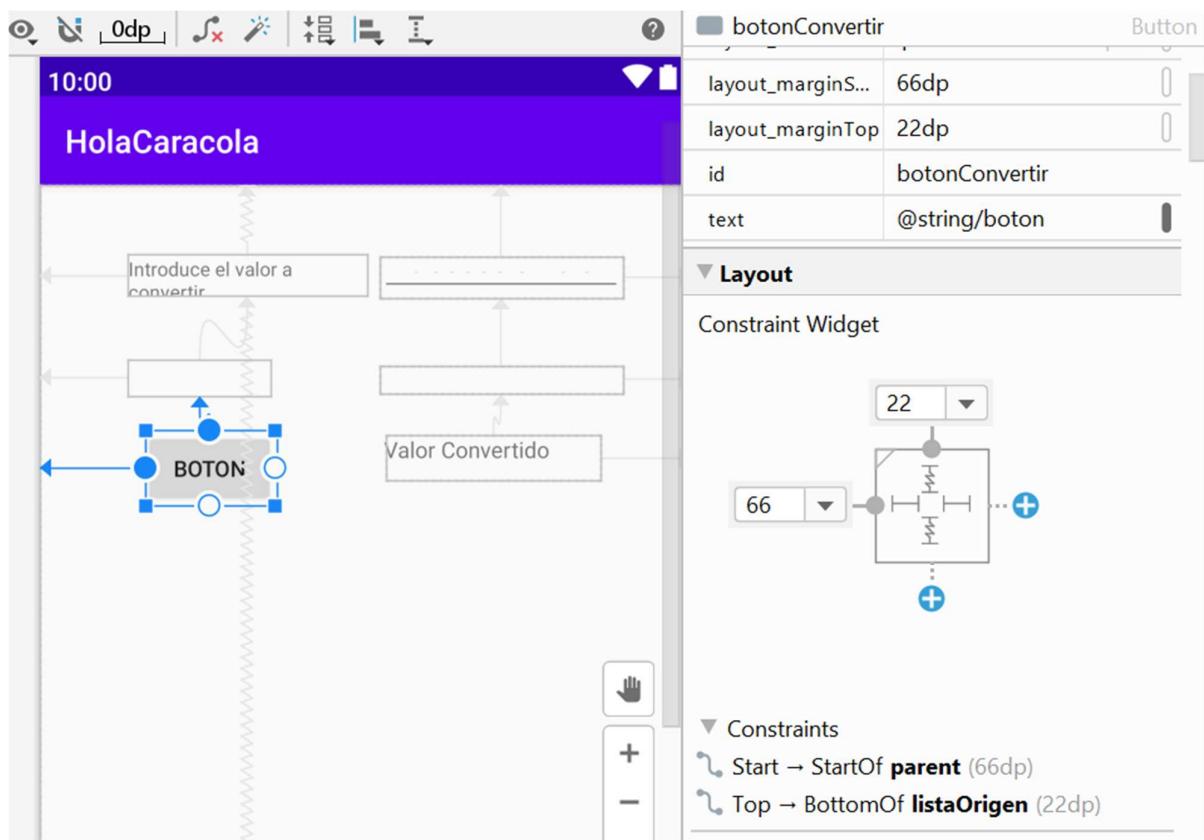
Para solucionarlo, tengo que ir elemento por elemento para definir los constraint

Antes:

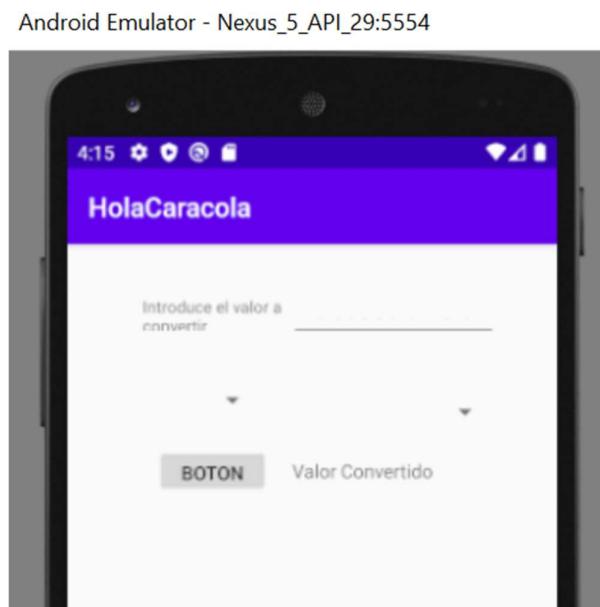




Después:

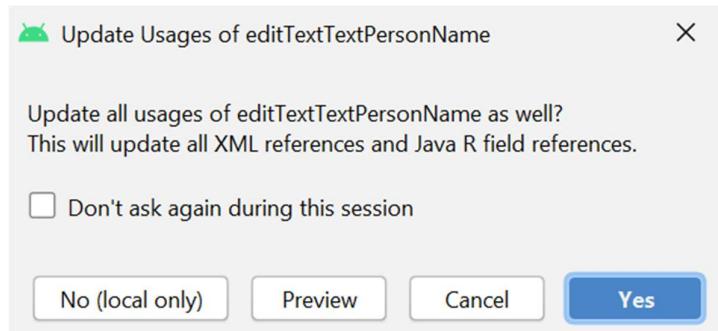


El resultado final podría ser este:



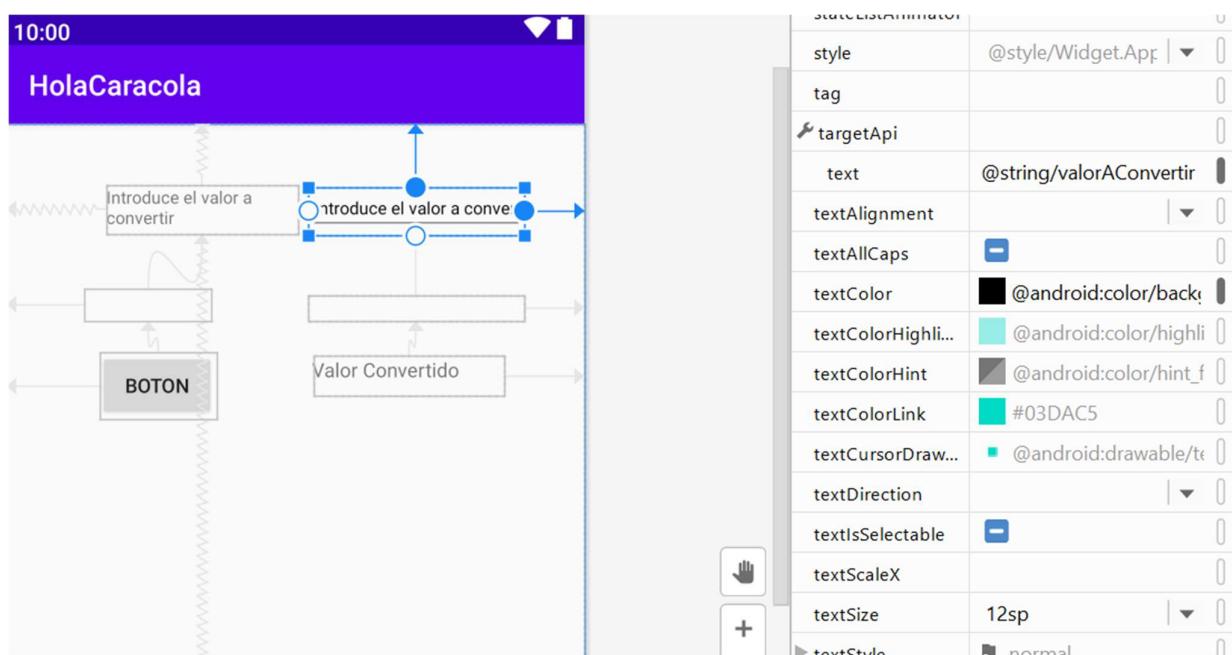


Si modiflico el nombre de un id que posee constraints con otros elementos:



En el elemento editable se ha de poner color y tamaño para que se vea mejor lo escrito:

```
android:textColor="@android:color/background_dark"
android:textSize="12sp"
```



En mi caso sería interesante eliminar el valor text ya que no deseo que el campo editable tenga por defecto ningún valor.

---



## Programar el botón y gestionar valores de ítems

Para ir realizando el código poco a poco y familiarizarnos con el uso de los valores de los elementos, vamos a realizar la parte de código que al apretar el botón visualice el valor del campo que queremos convertir en el campo en donde ha de aparecer el resultado de la conversión.

Para ello usaremos este código

```
activity_main.xml × strings.xml × MainActivity.java × AndroidManifest.xml × styles.xml ×
1 package com.example.myapplication;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.os.Bundle;
6 import android.view.View;
7 import android.widget.Button;
8 import android.widget.EditText;
9 import android.widget.TextView;
10
11 public class MainActivity extends AppCompatActivity {
12     private EditText varValorAConvertir;
13     private TextView varValorConvertido;
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18         Button miBoton = findViewById(R.id.botonConvertir);
19         varValorAConvertir = findViewById(R.id.valorAConvertir);
20         varValorConvertido = findViewById(R.id.valorConvertido);
21         miBoton.setOnClickListener (new View.OnClickListener() {
22             public void onClick(View v) {
23                 // Llamada a un método
24                 varValorConvertido.setText(varValorAConvertir.getText().toString());
25             }
26         });
27     }
28 }
```

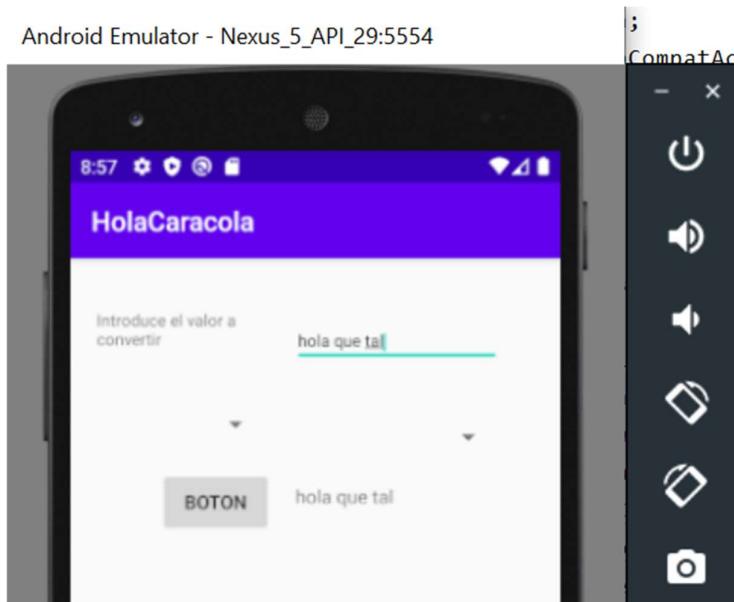
También podemos evitar declarar las variables al principio y declararlas al inicializarlas



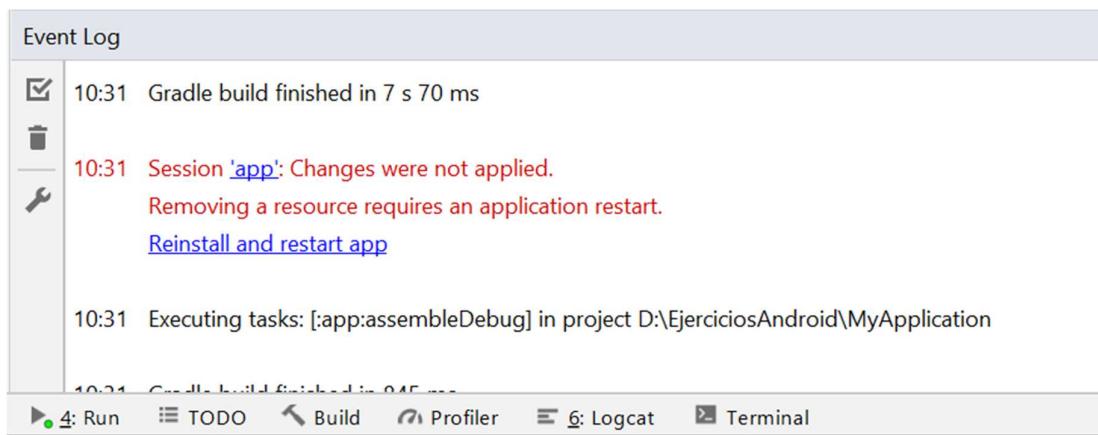
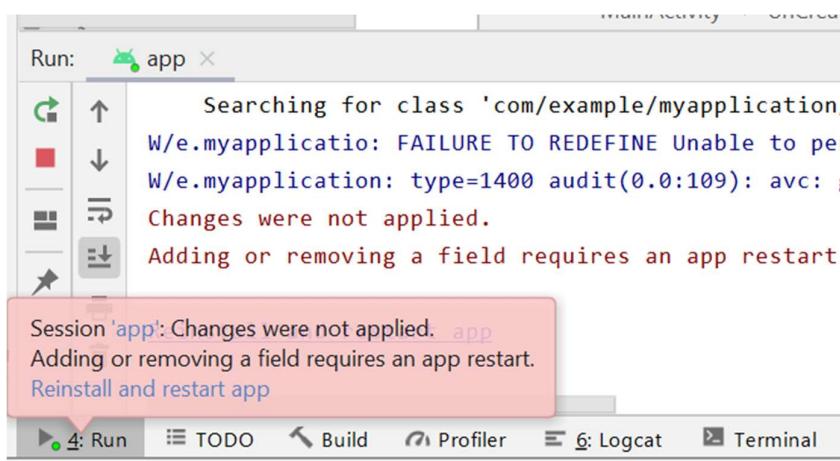
```
activity_main.xml strings.xml MainActivity.java AndroidManifest.xml styles.xml

1 package com.example.myapplication;
2 import androidx.appcompat.app.AppCompatActivity;
3 import android.os.Bundle;
4 import android.view.View;
5 import android.widget.Button;
6 import android.widget.EditText;
7 import android.widget.TextView;
8 public class MainActivity extends AppCompatActivity {
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13        Button miBoton = findViewById(R.id.botonConvertir);
14        final EditText varValorAConvertir = findViewById(R.id.valorAConvertir);
15        final TextView varValorConvertido = findViewById(R.id.valorConvertido);
16        miBoton.setOnClickListener (new View.OnClickListener() {
17            public void onClick(View v) {
18                // Llamada a un método
19                varValorConvertido.setText(varValorAConvertir.getText().toString());
20            }
21        });
22    }
23}
```

El resultado sería:



Si a la hora de ejecutar el emulador tenemos algún campo editable en uso, lo más seguro es que no podamos ejecutarlo con el “ctrl + F10” y tendremos que “Reinstall and restart app”. El propio Android Studio ya te ayuda a realizarlo.





## Gestion de spinners / lista de valores

<https://developer.android.com/guide/topics/ui/controls/spinner>

Esta es la vista del “activity\_main” de mis dos spinners

The screenshot shows the Android Studio interface with the activity\_main.xml file open. The code defines two spinners:

```
47
48     <Spinner
49         android:id="@+id/listaOrigen"
50         android:layout_width="90dp"
51         android:layout_height="22dp"
52         android:layout_marginStart="56dp"
53         android:layout_marginTop="40dp"
54         app:layout_constraintStart_toStartOf="parent"
55         app:layout_constraintTop_toBottomOf="@+id/labelValorAConvertir" />
56
57     <Spinner
58         android:id="@+id/listaDestino"
59         android:layout_width="153dp"
60         android:layout_height="18dp"
61         android:layout_marginTop="43dp"
62         android:layout_marginEnd="43dp"
63         app:layout_constraintEnd_toEndOf="parent"
64         app:layout_constraintTop_toBottomOf="@+id/valorAConvertir" />
65
66     <TextView
```

Para llenar la información del spinner tenemos que usar un SpinnerAdapter (como un ArrayAdapter) si la lista se encuentra en el código (hardCode) o bien un CursorAdapter si se encuentra en una base de datos.

Rellenar spinner a través de código

En nuestro caso, tenemos que realizar dos acciones. La primera definir la lista en el strings.xml

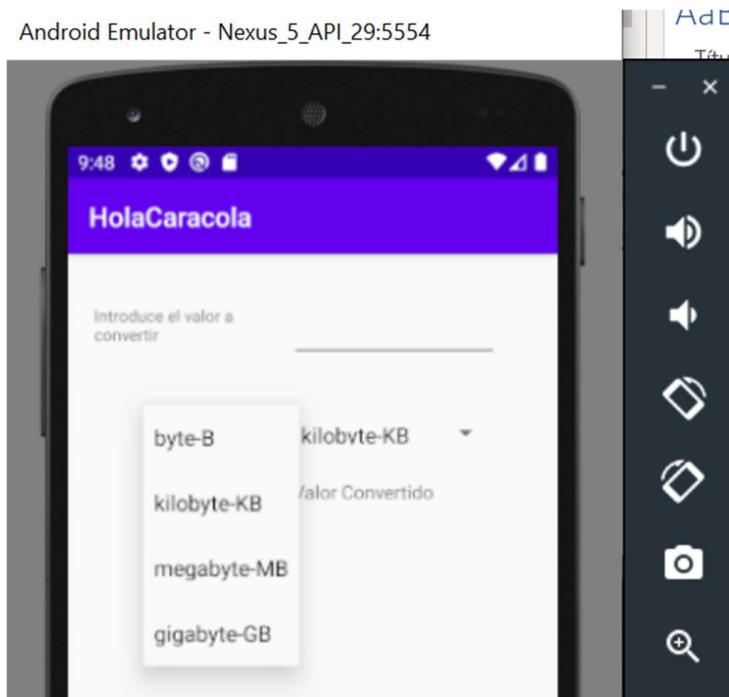


```
activity_main.xml x strings.xml x MainActivity.java  
Edit translations for all locales in the translations editor.  
1 <resources>  
2     <string name="app_name">HolaCaracola</string>  
3     <string name="valorAConvertir">Introduzca el valor a convertir</string>  
4     <string name="valorConvertido">Valor Convertido</string>  
5     <string name="boton">Botón</string>  
6     <string-array name="arrayBytes">  
7         <item>byte-B</item>  
8         <item>kilobyte-KB</item>  
9         <item>megabyte-MB</item>  
10        <item>gigabyte-GB</item>  
11    </string-array>  
12 </resources>
```

Y a continuación usarla para llenar la información nada más arrancar la app. Para ello añadiremos en el MainActivity.java la siguientes líneas de código

```
Spinner listaOrigen = (Spinner) findViewById(R.id.listaOrigen);  
Spinner listaDestino = (Spinner) findViewById(R.id.listaDestino);  
ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource( context: this, R.array.arrayBytes, android.R.layout.simple_spinner_item);  
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);  
listaOrigen.setAdapter(adapter);  
listaDestino.setAdapter(adapter);
```

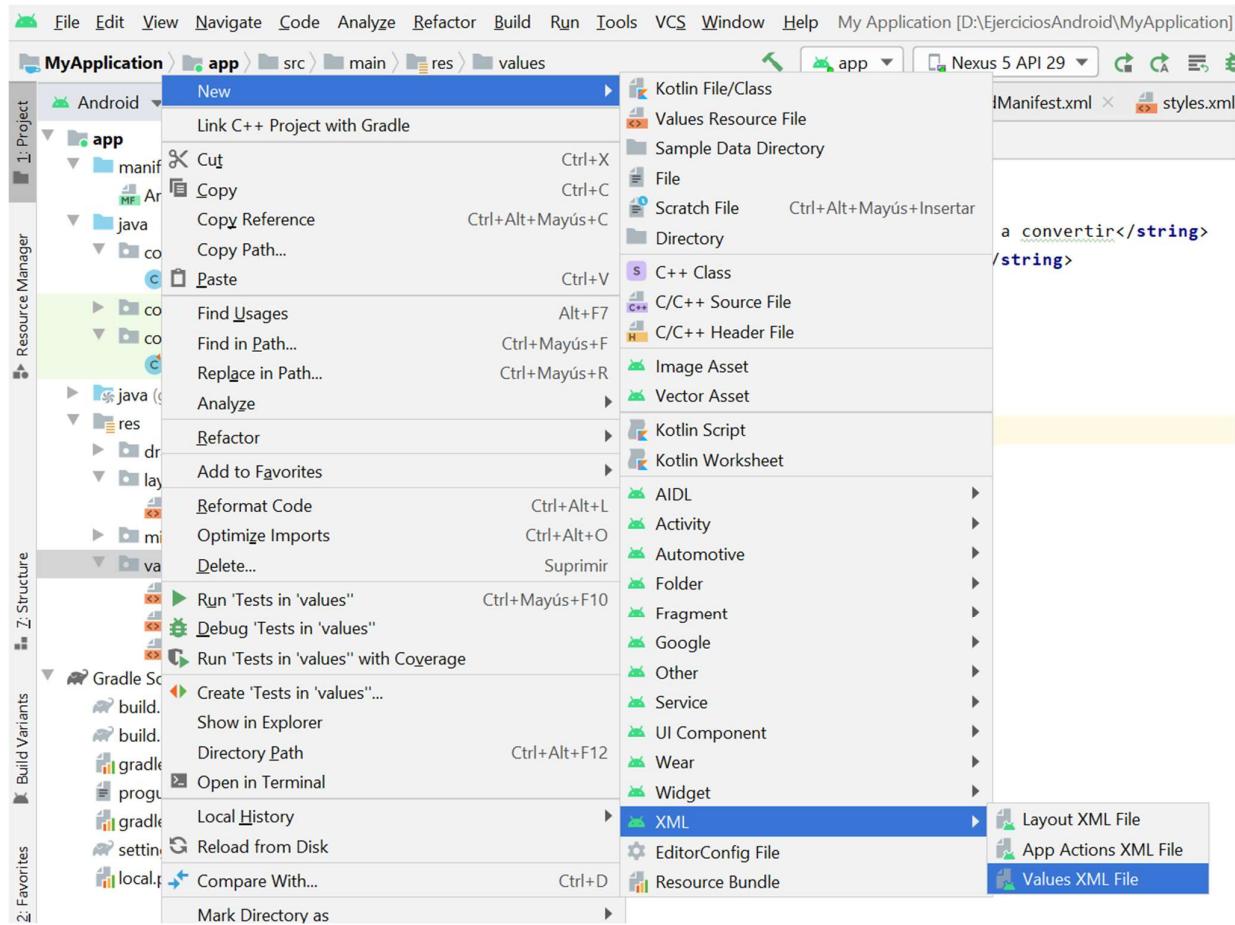
El resultado sería el siguiente:



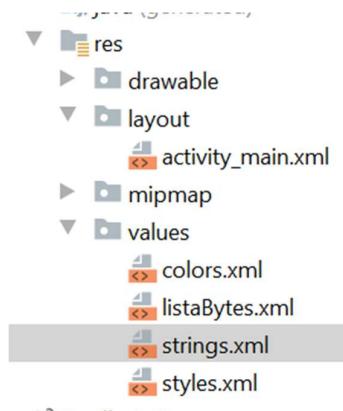


Rellenar spinner a través de un recurso

Primero creamos un nuevo fichero XML en la sección de recursos



En la sección de “res” nos genera un nuevo fichero xml





Informamos los posibles valores. En el nombre del array le pongo uno diferente para que no existan errores

Nota: añado un “s” en “bytes” para diferenciar las dos listas

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="arrayBytes2">
        <item>bytes-B</item>
        <item>kilobytes-KB</item>
        <item>megabytes-MB</item>
        <item>gigabytes-GB</item>
    </string-array>
</resources>
```

En la sección de entries añado la lista arrayBytes2:

The screenshot shows the Android Studio Resource Manager. At the top, there's a header with 'Attributes' and icons for search, settings, and close. Below it, a spinner labeled 'listaOrigen' is selected. In the 'Common Attributes' section, the 'spinnerMode' is set to 'dropdown'. The 'entries' attribute is highlighted with a red border, and its value is set to 'arrayBytes2'. The 'background' attribute is also visible. Below this, a 'Pick a Resource' dialog is open, showing a list of arrays under the 'Array' tab. Two arrays are listed: 'arrayBytes' and 'arrayBytes2'. Both arrays have two items: 'byte-B', 'kilobyte-KB', 'megabyte-MB', 'gigabyte-GB' | 1 version. The 'arrayBytes2' item is currently selected.



En XML sería:

```
<Spinner  
    android:id="@+id/listaOrigen"  
    android:layout_width="136dp"  
    android:layout_height="26dp"  
    android:layout_marginStart="56dp"  
    android:layout_marginTop="40dp"  
    android:entries="@array/arrayBytes2"  
    android:spinnerMode="dropdown"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/labelValorAConvertir" />
```

Como he modificado la listaOrigen, tengo que acordarme de comentar la línea de código que la rellenaba al arrancar la aplicación

```
// Inicializamos valores del Spinner  
final Spinner listaOrigen = (Spinner) findViewById(R.id.listaOrigen);  
Spinner listaDestino = (Spinner) findViewById(R.id.listaDestino);  
ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(context: this, R.ar  
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);  
//listaOrigen.setAdapter(adapter);  
listaDestino.setAdapter(adapter);
```

Ahora sí, us desplegable acaba con “s” y el otro no



Cómo usar los valores del spinner

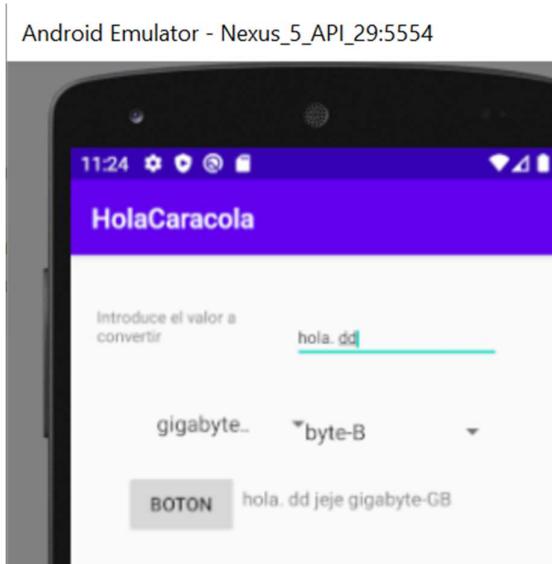
Ahora voy a modificar el código para que cuando apriete el botón me visualice el valor introducido y el valor de un spinner

En la programación del botón usaría el “getSelectedItem().toString()” del spinner



```
// Inicializamos valores del Spinner
final Spinner listaOrigen = (Spinner) findViewById(R.id.listaOrigen);
Spinner listaDestino = (Spinner) findViewById(R.id.listaDestino);
ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource( context: this, R.array.arrayBytes, android.R.layout.simple_spinner_dropdown_item);
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
listaOrigen.setAdapter(adapter);
listaDestino.setAdapter(adapter);
// Programamos las acciones que realizará el botón
miBoton.setOnClickListener (new View.OnClickListener() {
    public void onClick(View v) {
        varValorConvertido.setText(varValorAConvertir.getText().toString() + " jeje "
+ listaOrigen.getSelectedItem().toString());
    }
});
```

El resultado sería



Cálculo de conversión

Por último, sólo queda realizar el cálculo de la conversión. Para ello a mi se me ocurre crear una función que transforme los valores escogidos del spinner en bytes y aplicar luego una fórmula de conversión.