



Trabajo Práctico N° 1

A continuación, desarrollaremos nuestro primer proyecto de Android basado en una sencilla aplicación. Construiremos nuestra aplicación asumiendo que ya hemos instalado y configurado tanto nuestro entorno de desarrollo, el SDK y el emulador. Además, nosotros utilizaremos el entorno de desarrollo Android Studio. Esta actividad la dividiremos en varios prácticos para no recargar los contenidos pedagógicos.

Comenzando – Haciendo Uso del Asistente de Nuevos Proyectos

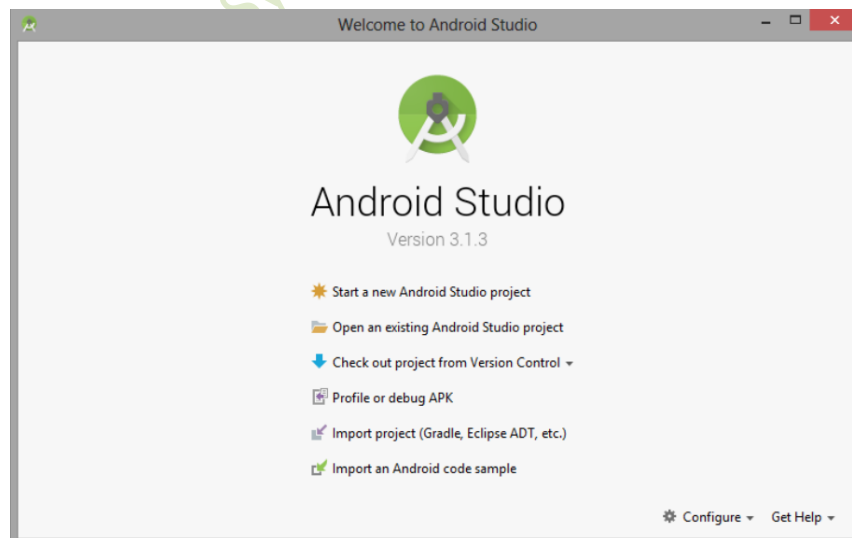
1. Lo primero que debemos hacer es saber dónde alojaremos nuestra aplicación. Según el entorno de desarrollo que estemos utilizando, el alojamiento del proyecto puede presentar varias formas de crearlo y ubicarlo en algún espacio en tu computadora. Dado que utilizaremos el entorno de desarrollo llamado **Android Studio**, podremos alojarlo donde queramos.

Reglas para un Almacenamiento Óptimo

Para almacenar tus proyectos, es aconsejable que no utilices caracteres que hagan uso de acentos, diéresis, caracteres como las eñes “ñ”, numeral “#”, etc. Utiliza caracteres más genéricos y si sabes inglés, utiliza los caracteres propios de esta lengua. Esto evitará que puedas tener conflictos dentro del proyecto. También, evita almacenar los proyectos dentro de carpetas y subcarpetas anidadas muy profundamente. Es aconsejable que uses una carpeta para almacenar cada uno de tus proyectos.


Como consejo importante es que todas tus prácticas, trates de almacenarlas previamente en una carpeta principal la que le darás del nombre de **CursoAndroid**. Por favor, crea esta carpeta tal cual, sin espacios entre ambas palabras. Procura colocar la carpeta en el directorio raíz **C:** de Windows para obtener los mejores resultados.

2. Abre tu entorno de **Android Studio**. Una vez hecho esto, en pantalla tendrías la siguiente figura.

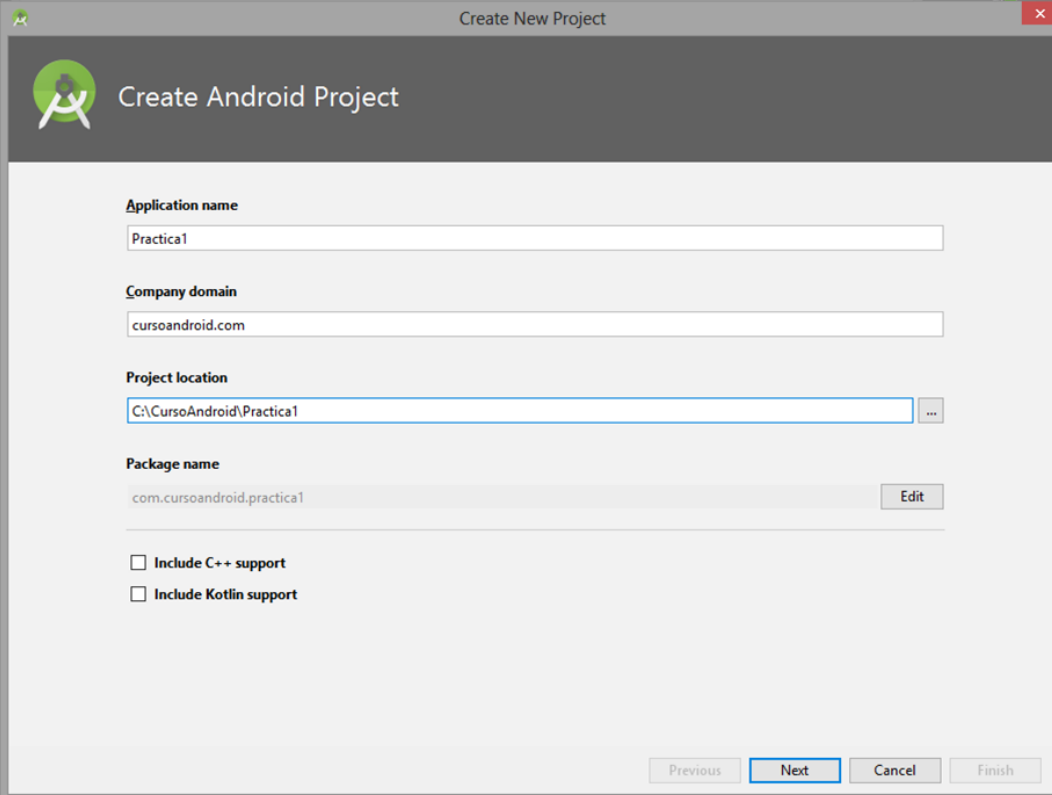


3. Observa que existen varias opciones. Dado que vamos a crear nuestra primera aplicación, haremos clic entonces en la opción que dice **Start a new Android Studio project**.



Importante: Las aplicaciones que se hayan hecho con anterioridad, suelen cargarse adjunto a esta ventana y por tanto, nos permite acceder a esos proyectos más rápidamente. Se pueden desvincular de esta ficha, haciendo clic en el icono  que se asoma en el sector del encabezado esquina derecha, cuando nos posamos con el puntero del ratón sobre esta sección.

- Una vez hecho esto, se abrirá la siguiente ventana en la que colocaremos varios datos. Por ejemplo, datos para el nombre de nuestra futura aplicación, el paquete e incluso, la ubicación del proyecto en el disco. Recuerda que lo almacenarás en la carpeta principal **CursoAndroid**.



Los datos para esta ventana son: En el campo **Application Name** coloca **Practica1**. Luego en **Company Domain** coloca **cursoandroid.com**. Finalmente en **Project location** y haciendo clic en el botón en el extremo derecho, para pasar a buscar la carpeta principal para almacenar el proyecto. Bien, una vez hecho esto, la ruta quedaría como **c:\CursoAndroid**. Finalmente, has clic en el botón **Next**.

- Una vez hecho esto, al pasar a la siguiente pantalla, podrás observar la siguiente imagen. Aquí podrás configurar el versionado de la aplicación y el destino para el tipo de dispositivo.



Create New Project

Target Android Devices

Select the form factors and minimum SDK
Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

☒ **Phone and Tablet**
API 16: Android 4.1 (Jelly Bean) ▾
By targeting **API 16 and later**, your app will run on approximately **99,2%** of devices. [Help me choose](#)
☐ Include Android Instant App support

☐ **Wear**
API 21: Android 5.0 (Lollipop) ▾

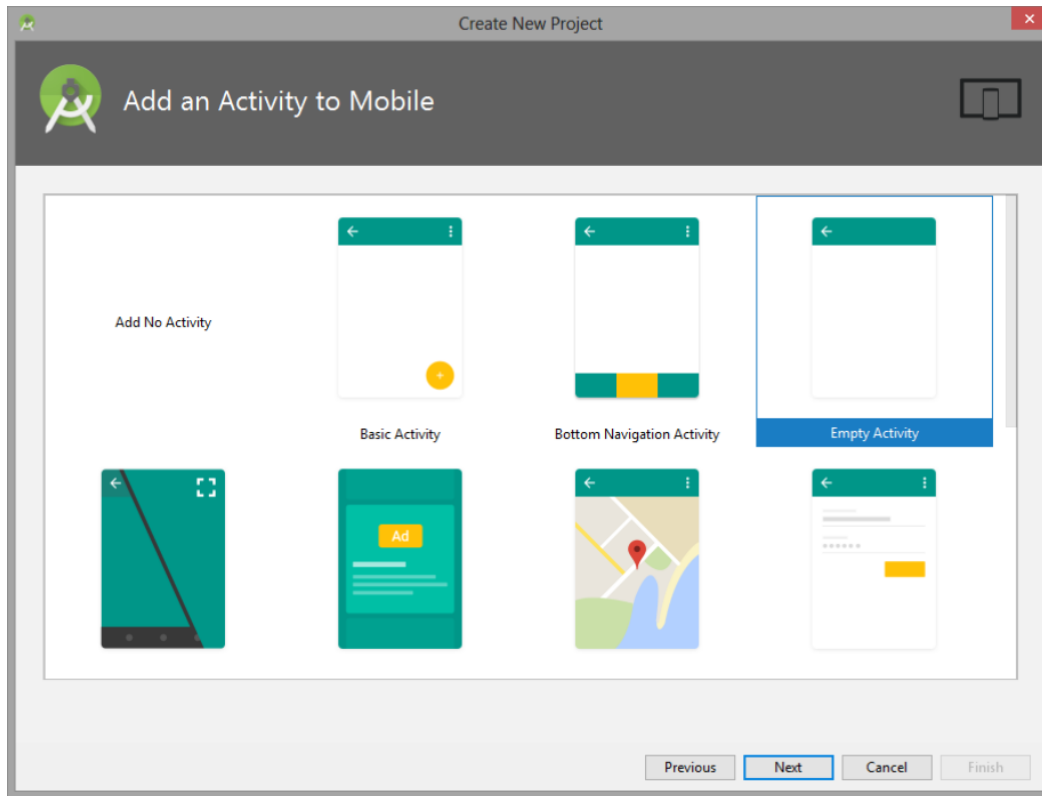
☐ **TV**
API 21: Android 5.0 (Lollipop) ▾

☐ **Android Auto**

☐ **Android Things**
API 24: Android 7.0 (Nougat) ▾

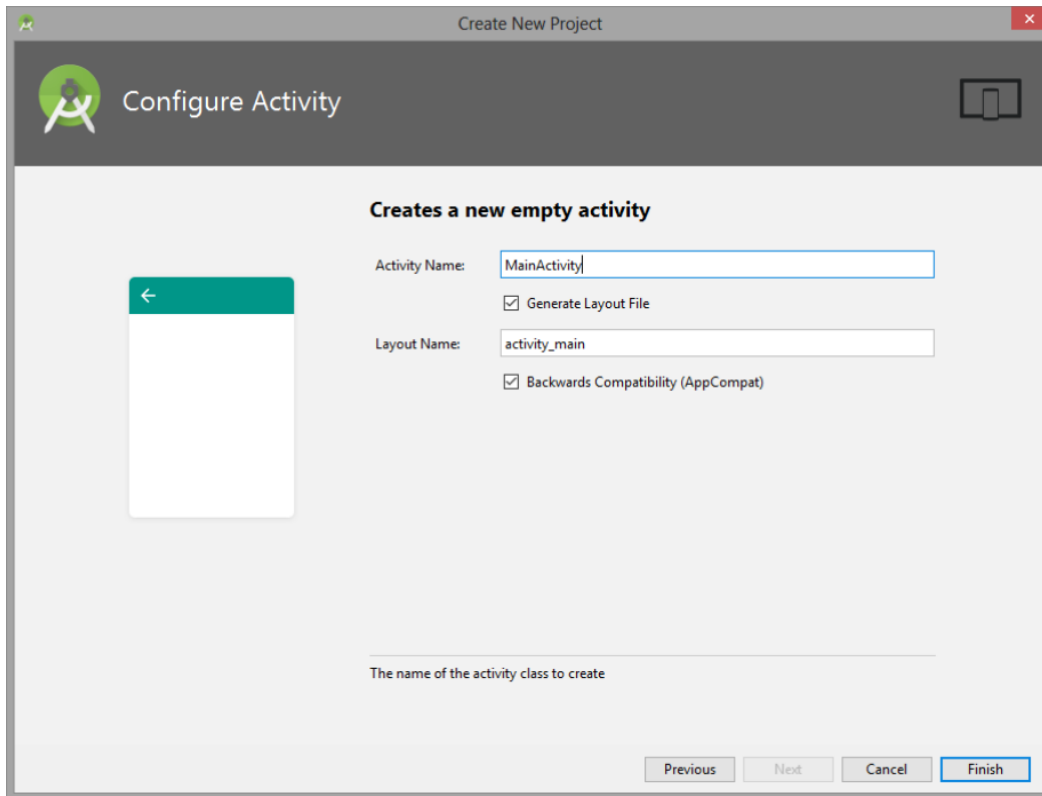
Previous Next Cancel Finish

6. En este proceso, tan solo dejaremos marcada la primera opción y además, debemos asegurarnos tener seleccionado en el cuadro combinado la versión de **Minim SDK: API 16: Android 4.1 (Jelly Bean)**. A continuación, haz clic en el botón **Next**.



7. A continuación, en la ventana siguiente, pasaremos a seleccionar la plantilla que contiene un **Activity** vacía. Por tanto, seleccionaremos la opción que dice **Empty Activity**. A continuación, has clic en el botón **Next**.
8. En la siguiente ventana nos pide y muestra si deseamos cambiar el nombre para nuestra **Activity** para la capa de diseño de nuestra aplicación. Por defecto, dejaremos los nombres que nos sugiere el asistente. Finalmente, has clic en el botón **Finish**.

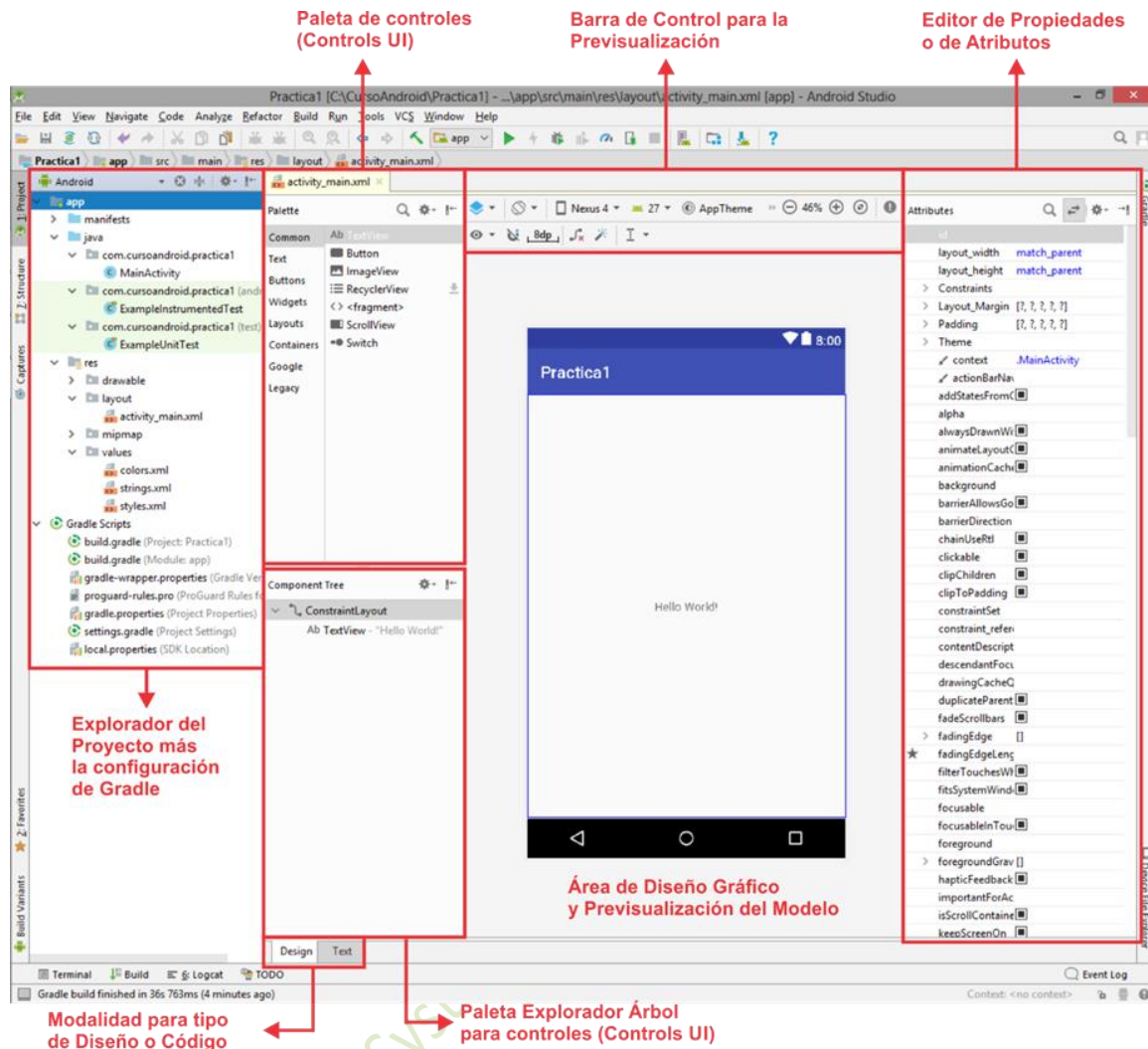
Importante: Los nombres que se le brinden a las capas para el diseño de la **Activity**, es recomendable que se haga utilizando caracteres ANSI, es decir, no usar letras como la eñe "ñ", los acentos o diéresis, etc. En la pantalla, nos encontramos con dos campos. El campo **Activity Name**, como convención, siempre se escribe con la nomenclatura de la primera palabra se escribe en mayúscula y que se debe incluir la palabra *activity*. Es por ello que cuando escribes en ese campo, en el campo de abajo llamado **Layout Name** se escribe automáticamente separando las palabras con un guion bajo. Es recomendable que no escribas en este campo porque podrías romper la estructura.



9. Antes de continuar, verifica que las casillas **Generate Layout File** y **Backward Compatibility (AppCompat)** esté marcadas. Una vez que hallas hecho clic en el botón **Finish**, el asistente junto a la plantilla seleccionada, construirá nuestro proyecto junto con todos los recursos para la aplicación. Luego, podrás empezar a verificar la carga de los módulos en pantalla y podrás acceder a cada una de las partes del proyecto. En sucesivas prácticas, iremos profundizando cada una de estas áreas e incluso, hablaremos de los Emuladores o Máquinas Virtuales donde haremos todas las pruebas necesarias para nuestras aplicaciones.

Preparado y Listo para Comenzar a Diseñar y Desarrollar

Una vez que el asistente construya el proyecto desde la plantilla y la prepare dentro del entorno de desarrollo, estarás listo para comenzar a diseñar o desarrollar tu futura aplicación. Resulta importante pasar revista por el área en donde trabajarás sobre tu proyecto. Observa la siguiente imagen.



El **Explorador de Proyectos** te permite navegar a través de cada una de las carpetas del proyecto y donde hallarás todos los recursos. Las carpetas se encuentran conceptualmente separadas y son representadas distintos colores. La carpeta principal del proyecto **app**, agrupa a todas las carpetas del proyecto es la principal. En la sección de carpetas color cian-oscuro, se encuentran la carpeta **manifests** donde se aloja el archivo del manifiesto del proyecto. Luego una carpeta llamada **java** donde normalmente construirás tu software. En esta sección existen tres carpetas. Una de ellas es donde se contruye el software y las otras dos restantes, son utilizadas para realizar todo tipo de pruebas y depuraciones.

La carpetas de color gris-oscuro, permiten trabajar sobre el área del diseño, basado en los archivos de recursos, descriptivos, etc. Su carpeta principal es **res**. En esta sección podrás incluir y crear recursos tales como capas de actividades, fragmentos, menús, cadenas literales, estilos, recursos de imágenes, audio, vídeo, etc. Esta sección se encuentra dividida en varias carpetas anidadas a la carpeta principal **res**. Por tanto, dentro de esta gran carpeta, te encontrarás con carpetas tales como **drawable**, **layout**, **menu**, **values**, **string**, **styles**, **colors**, **mipmap**, **assets**, etc.

Algunas de las carpetas mencionadas en estos ejemplos, normalmente no se crean durante la preparación del proyecto de parte del asistente. Deberán entonces ser creadas después, como por ejemplo la carpeta **assets** que es utilizada para almacenar archivos especiales, tales como los de ayuda de una aplicación aunque podrían tratarse de otros tipos archivos como de audio, vídeo,



etc. Otra es la carpeta **menu** que es utilizada para almacenar los archivos que permiten la construcción de los menús para las capas de actividades.

Luego tenemos la sección de Gradle para compilar el proyecto. Esta herramienta ofrece un proceso amplio de soluciones que mejoran los procesos de compilación y además, permite gestionar el proceso de vistas en el área de diseño gráfico del entorno de desarrollo Android Studio. Gracias a esta herramienta, podemos realizar el diseño y la visualización de los trabajos. Esto lo puedes observar en la sección **Área de Previsualización**.

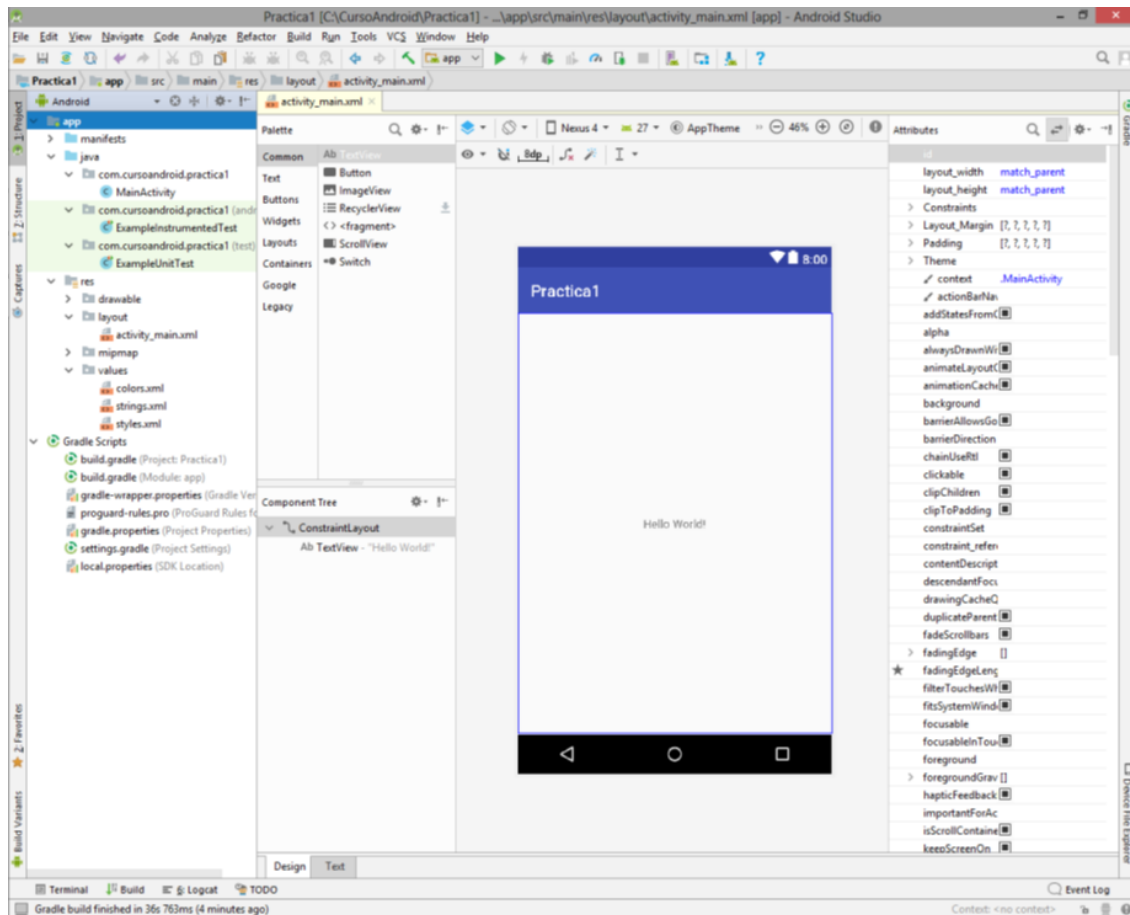
Otra sección muy importante es el área donde podemos visualizar los controles que utilizamos al incluirlos en la capa gráfica junto a la sección donde podemos observar en detalles las propiedades de los controles dibujados en el área de desarrollo de diseño. Esto lo vemos en los **Explorador de Controles UI y Editor de Propiedades o Atributos**.

Sección del Código

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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" />

</android.support.constraint.ConstraintLayout>
```



Código de Lado de Diseño y Área de Dibujado para el Diseño

```
package com.cursoandroid.practical;  
  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
  
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

Código de Lado de Java

En la primera sección en esta parte del instructivo, podemos observar que existen dos áreas de operaciones para el diseño y el desarrollo. En estas áreas se puede diseñar las interfaces escribiendo los controles bajo el lenguaje de XML o bien, haciendo uso de la interfaz gráfica mediante el mecanismo de arrastrar, soltar, estirar, etc. Por otro lado tenemos el área de Java donde escribiremos gran parte de la codificación del software. En consecuencia, ambas áreas serán cruciales para el desarrollo del software.