**GUIA DE DESARROLLO ANDROID – CLASE COMPONENTES BASICOS - ARQUITECTURA**

OBJETIVOS

* Realizar la creación de un proyecto con Android Studio
* Conocer los fundamentos básicos de un proyecto en Android

INDICACIONES

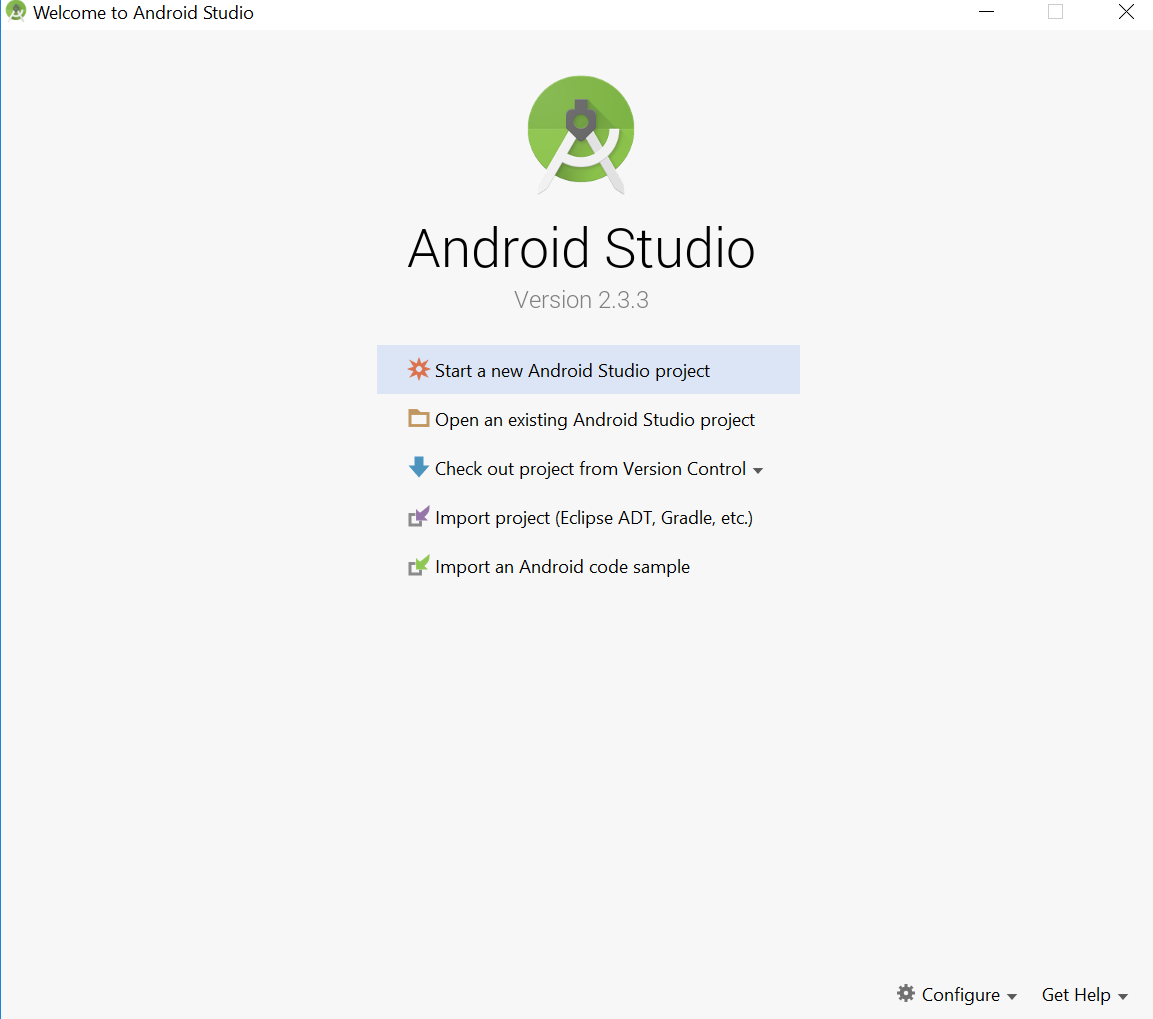
* Seguir la siguiente guía de desarrollo durante la clase.

Al finalizar la guía realice lo siguiente:

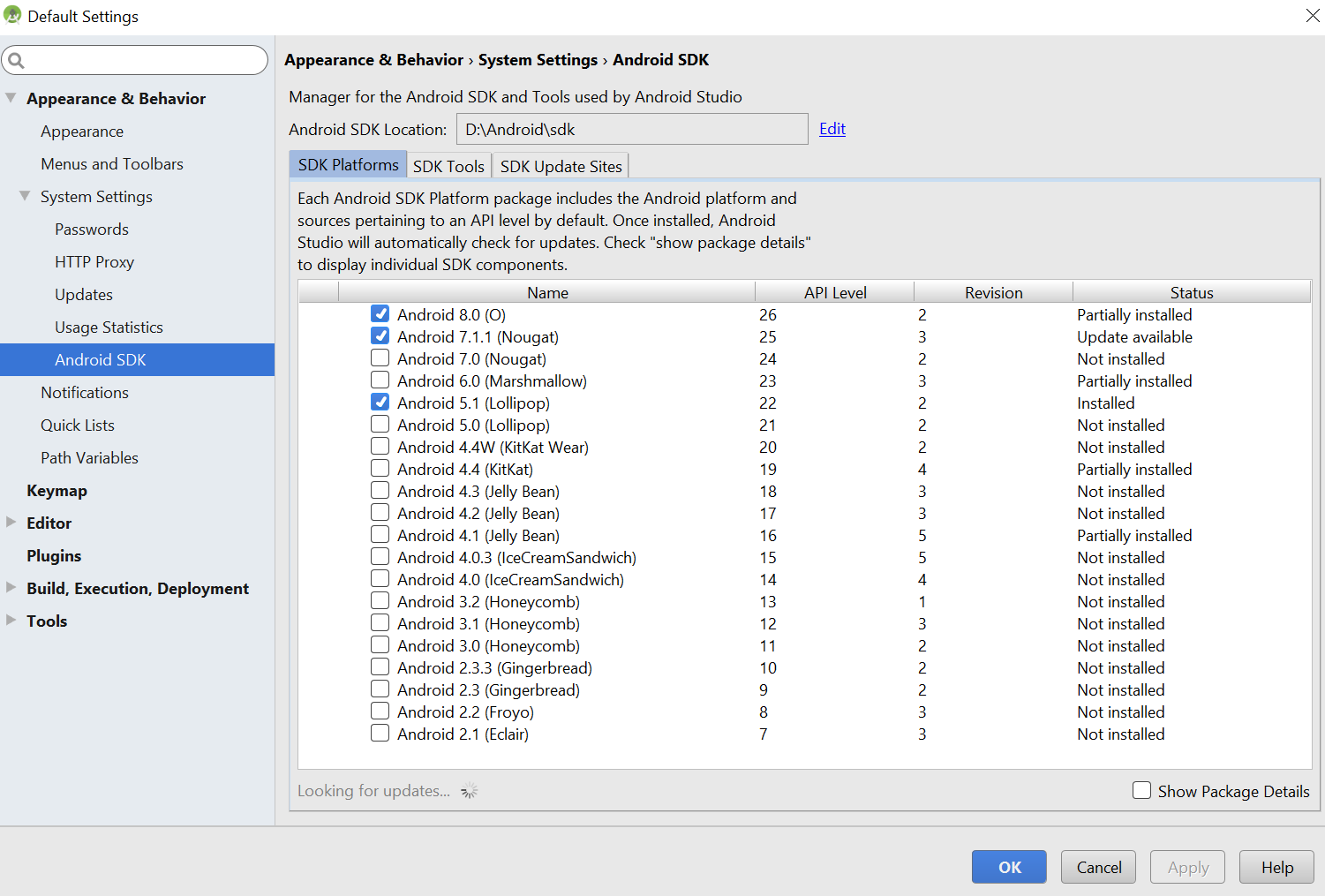
1. Subir en el aula virtual el proyecto desarrollado
2. Subir un documento con las capturas de pantalla del proyecto desarrollado.

**Configuración Android Studio:**

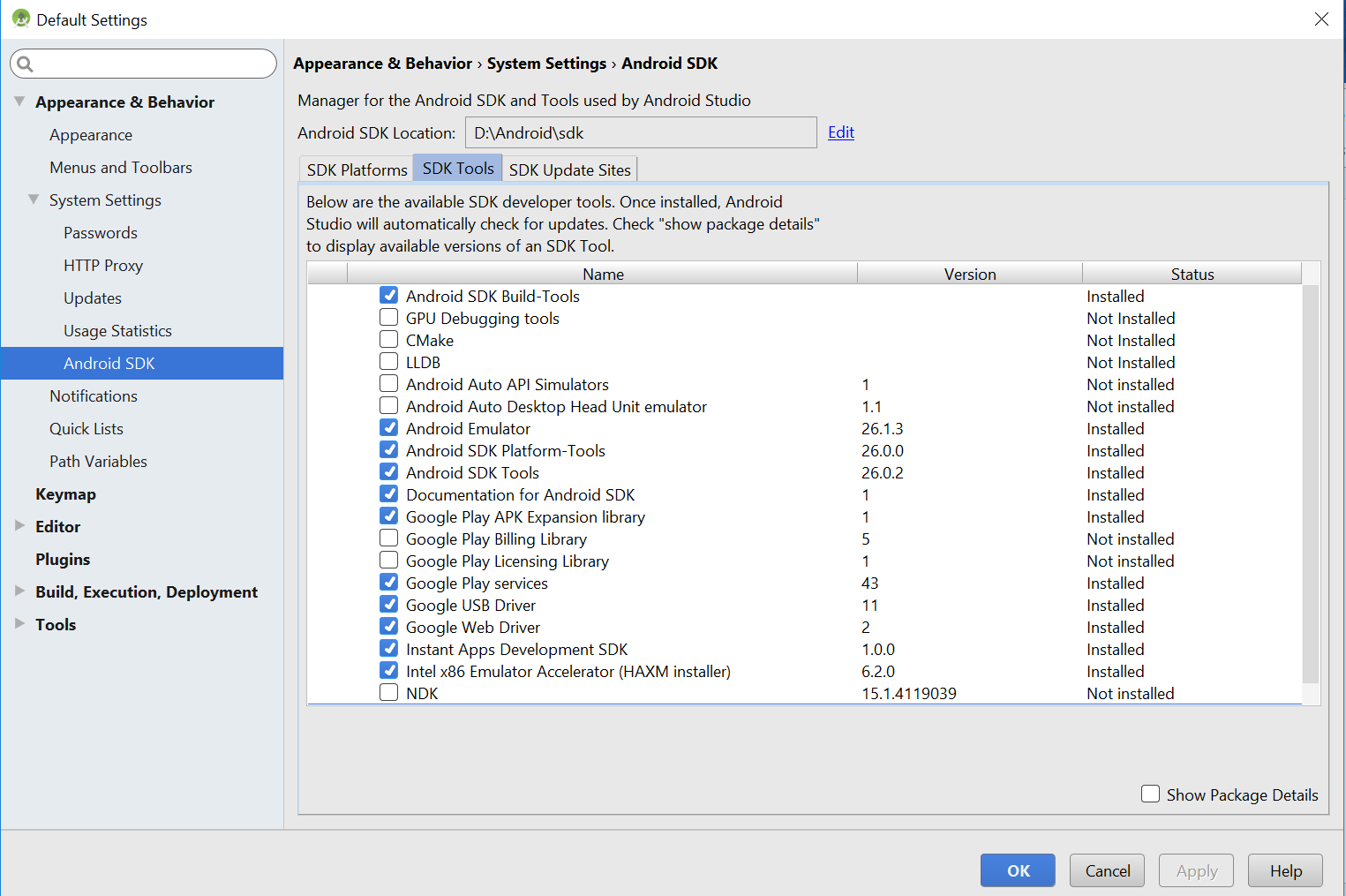
1. Es necesario tener instalado el Android Studio en el computador.
2. Al ejecutar el Android Studio le aparecer la siguiente ventana:



1. Para validar que tenemos el kit de desarrollo instalado, seleccionar la opción **Configurar**.
2. Seleccionaremos las versiones de Android con las que vayamos a trabajar. Para esto en el tab SDK Platform seleccionaremos las versión que vayamos a requerir. En caso no este instalado el SDK requerido, es necesario seleccionar la versión y desacargarla desde la interface que nos ofrece el AS.

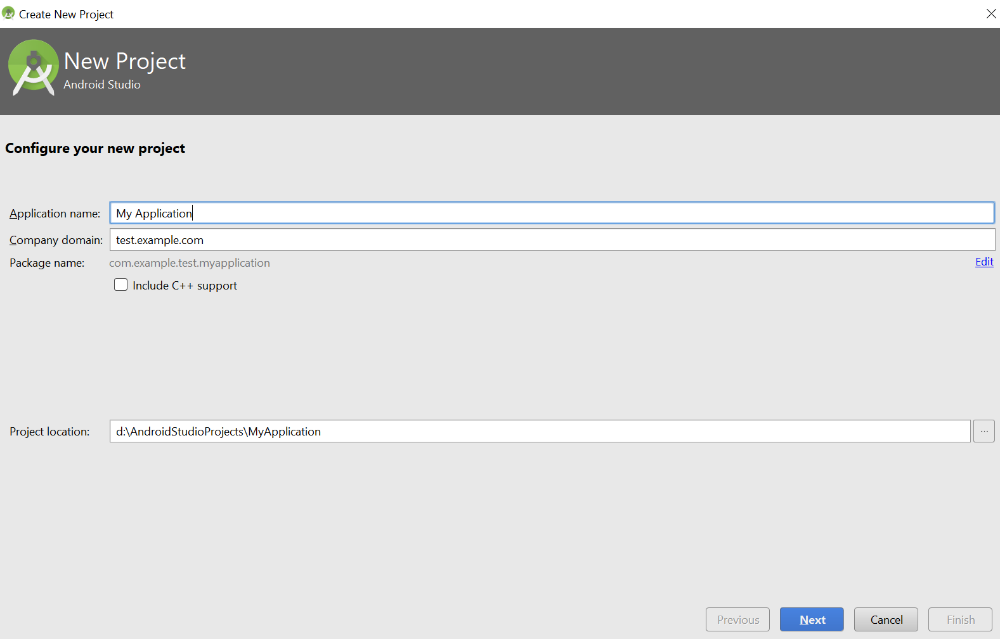


1. Adicional al SDK, hay algunos utilitarios que nos servirán durante el desarrollo. Para ello debemos seleccionar el tab **SDK Tools**. Asegurarse que se tengan instalados los siguientes paquetes.



**Creando un nuevo Proyecto:**

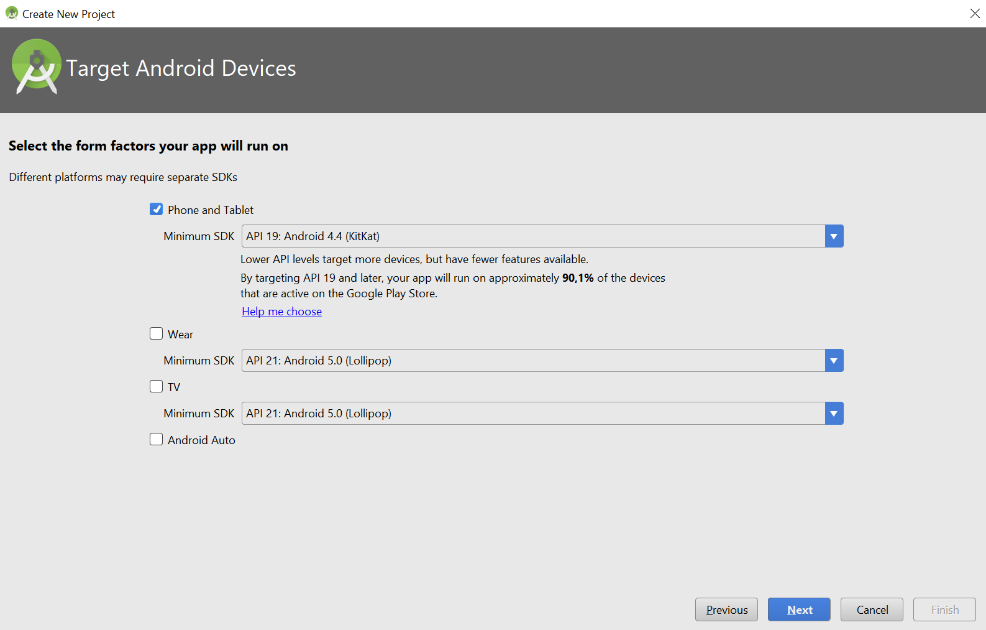
1. En la pantalla de Inicio seleccionamos la opción Start a new Android Studio Project. En seguida nos carga una pantalla como la que se muestra a continuación:



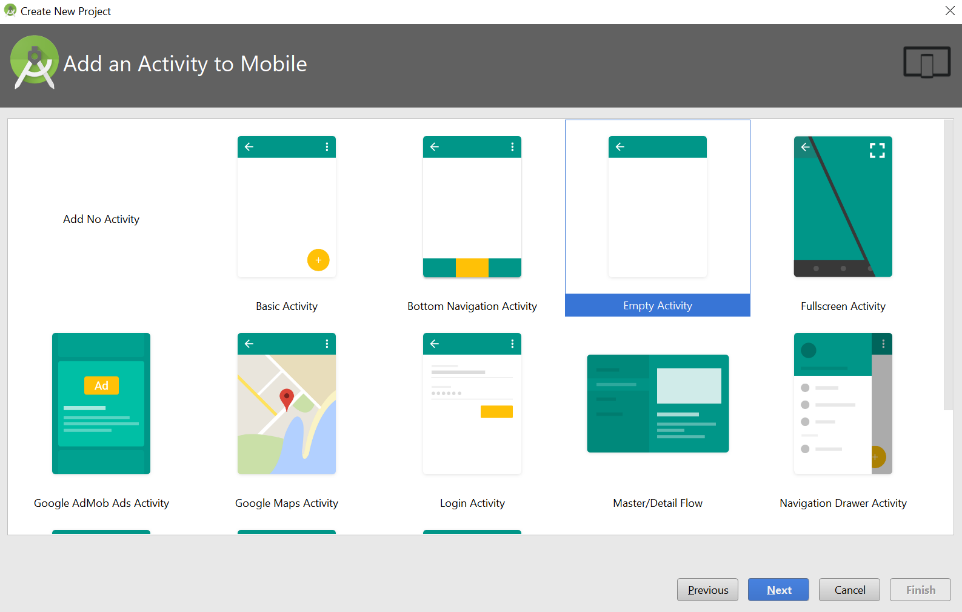
1. Complete los siguientes datos:

* Application Name: AndroidApp1
* Company domain: androidapp1.upc.edu.pe
* Seleccione la ruta donde se almacenará su Proyecto. No es necesario incluir el soporte a C++

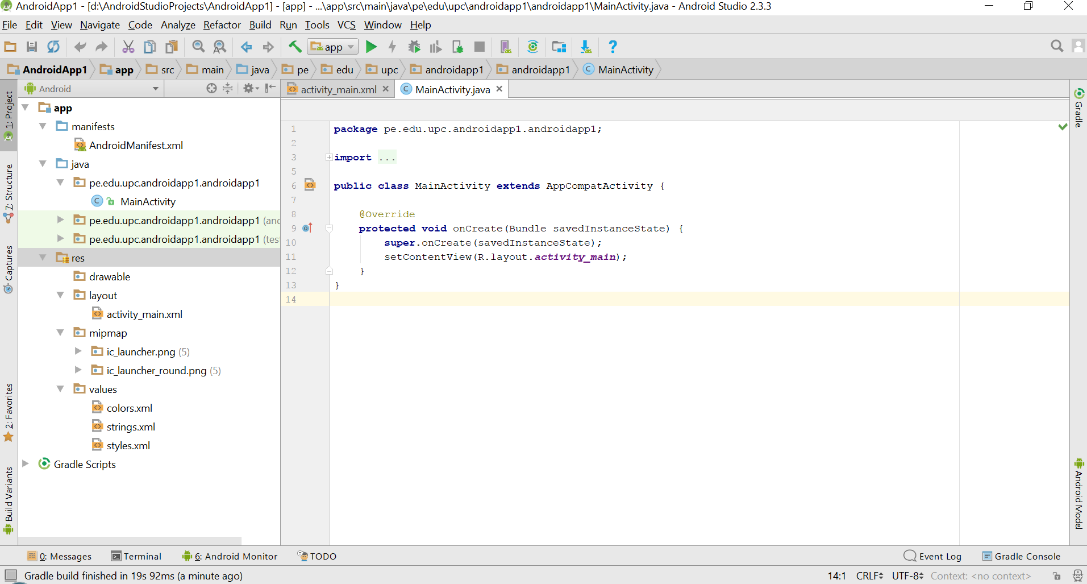
1. Seleccione la versión del SDK sobre el cual desarrollará su proyecto.



1. Seleccione la opción Empty Activity.

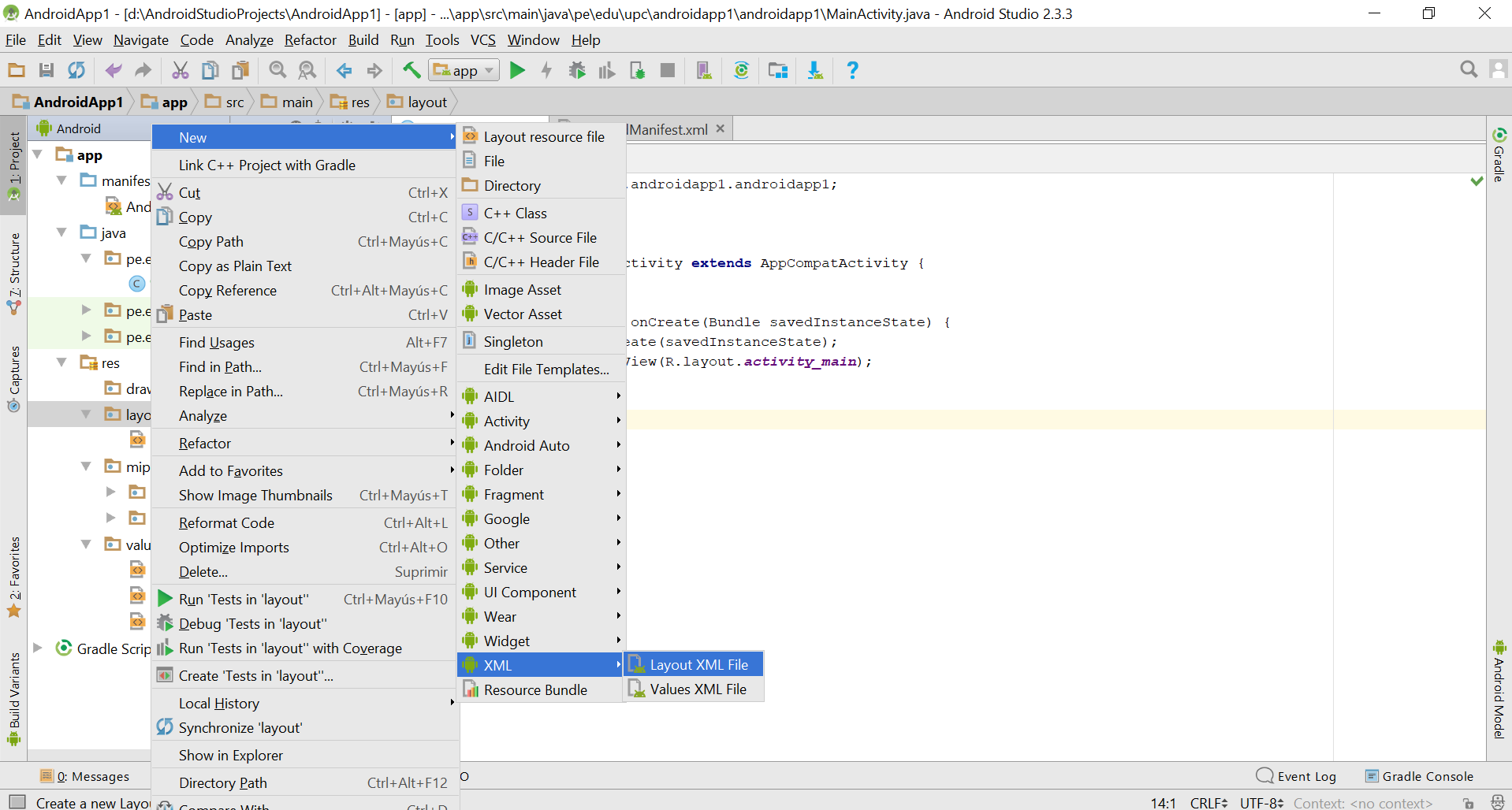


1. Finalice la creación del proyecto.
2. Una vez que se ha creado el proyecto se mostrará una estructura como la siguiente:

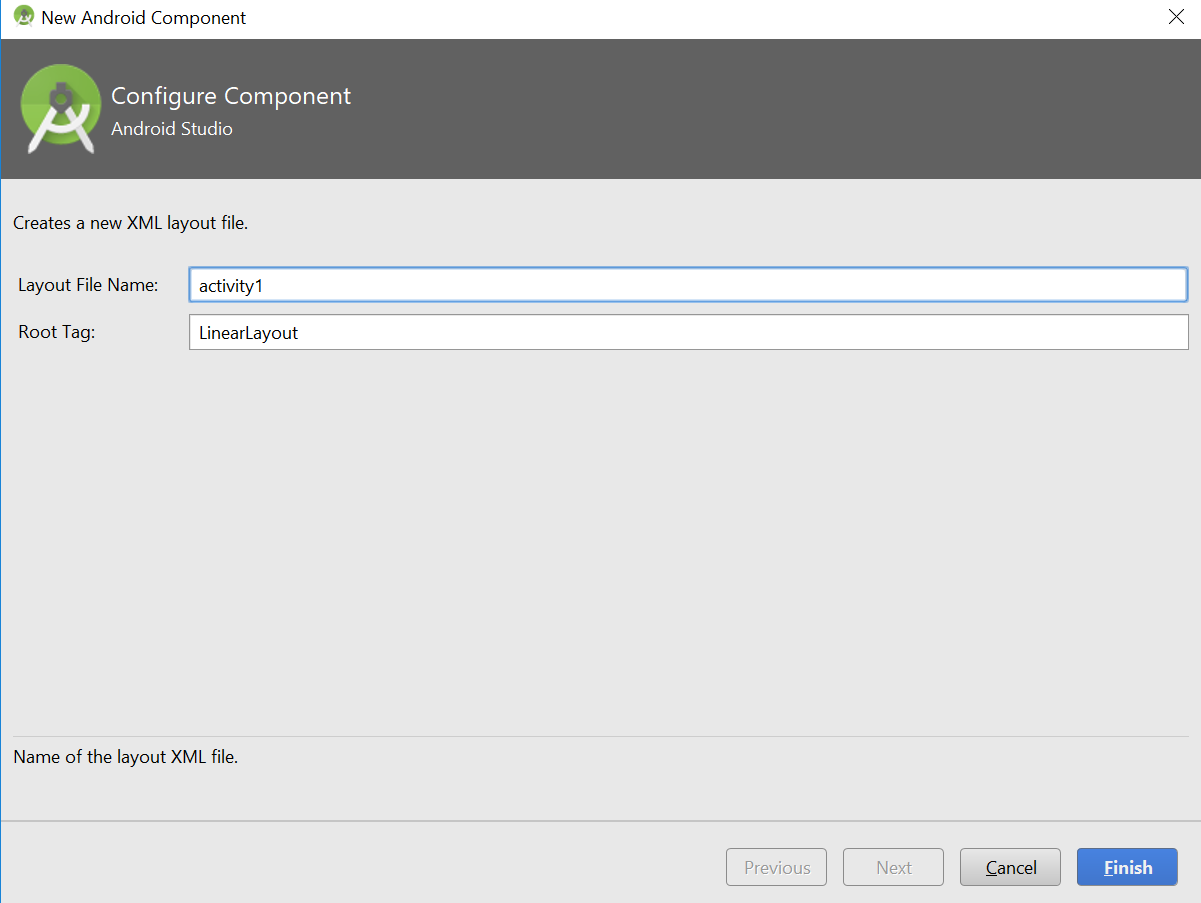


**CREAR XML DE PRESENTACION**

1. Primero tenemos que crear los archivos XML que van a servirnos como interfaces de usuario.
2. Para eso, dentro de la carpeta res/layout cree un primer archivo de la siguiente manera:



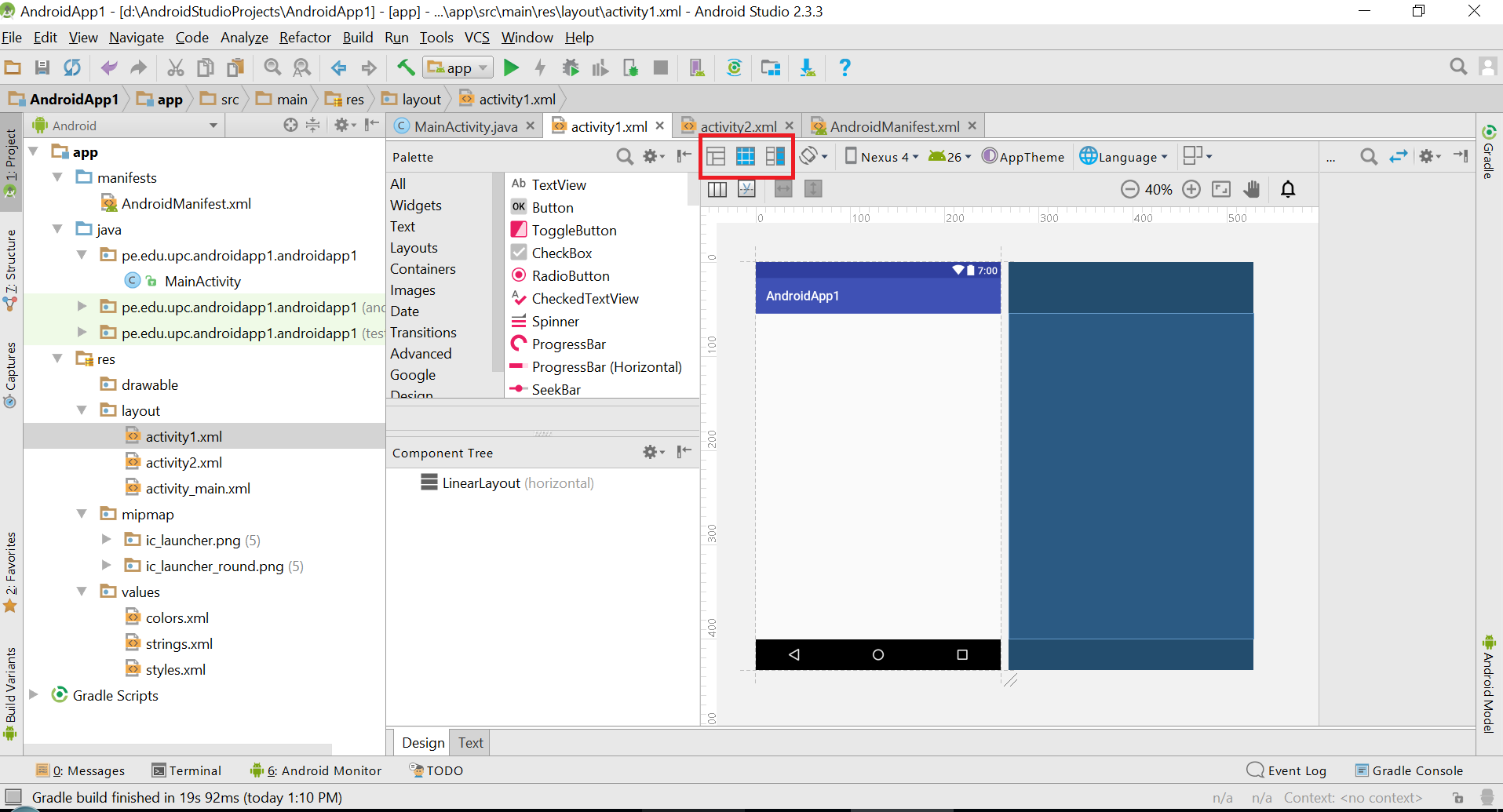
Ingrese el nombre del archivo: **activity1** y root tag dejarla tal c



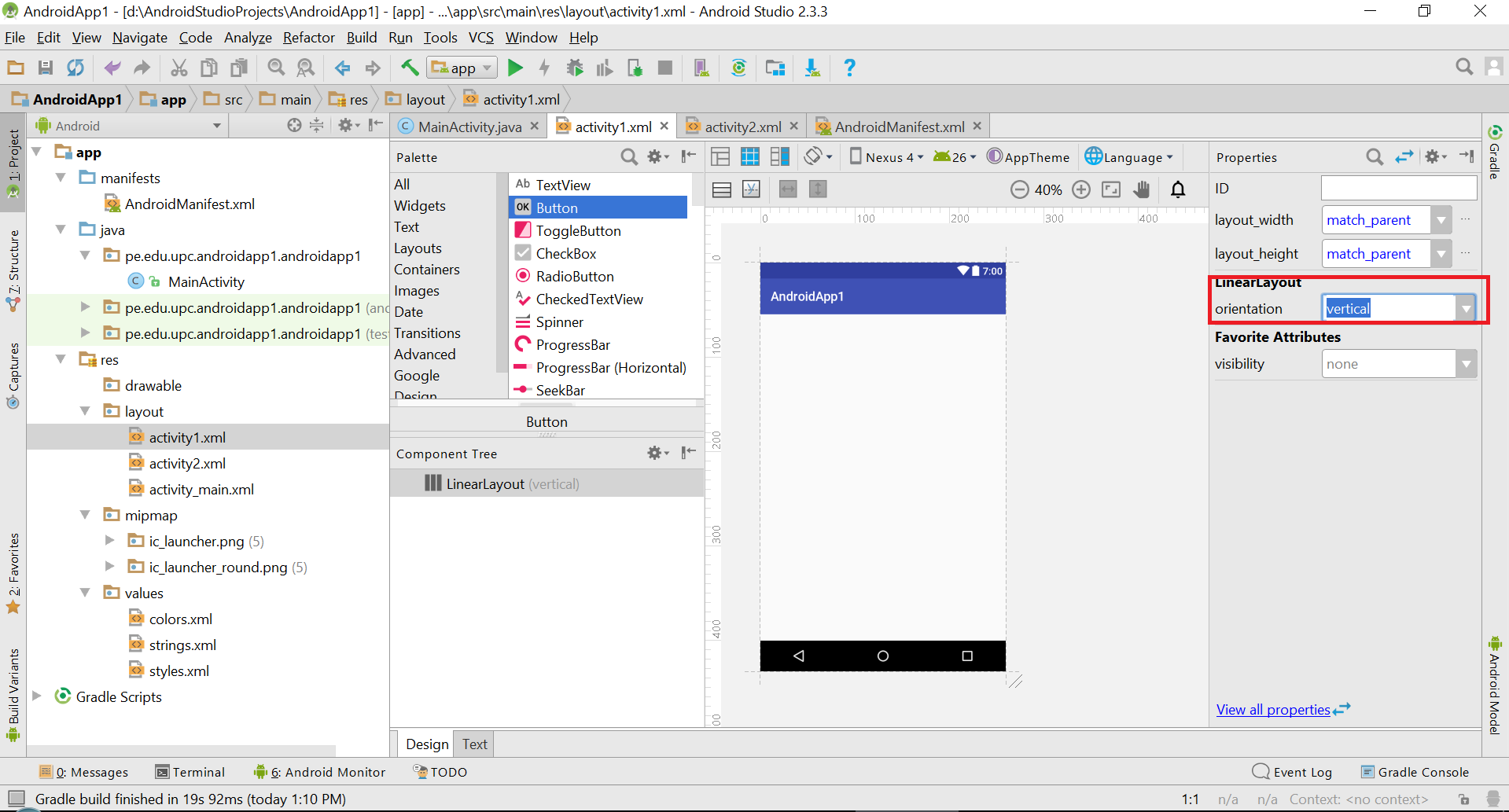
Repita los mismos pasos y cree el archivo **activity2**

**PARTE 1 - EDITAR INTERFASES DE USUARIO**

1. Abra el archivo **activity1**. Tendrá una vista como la siguiente. Si gusta puede trabajar en modo blueprint, mixta o normal.

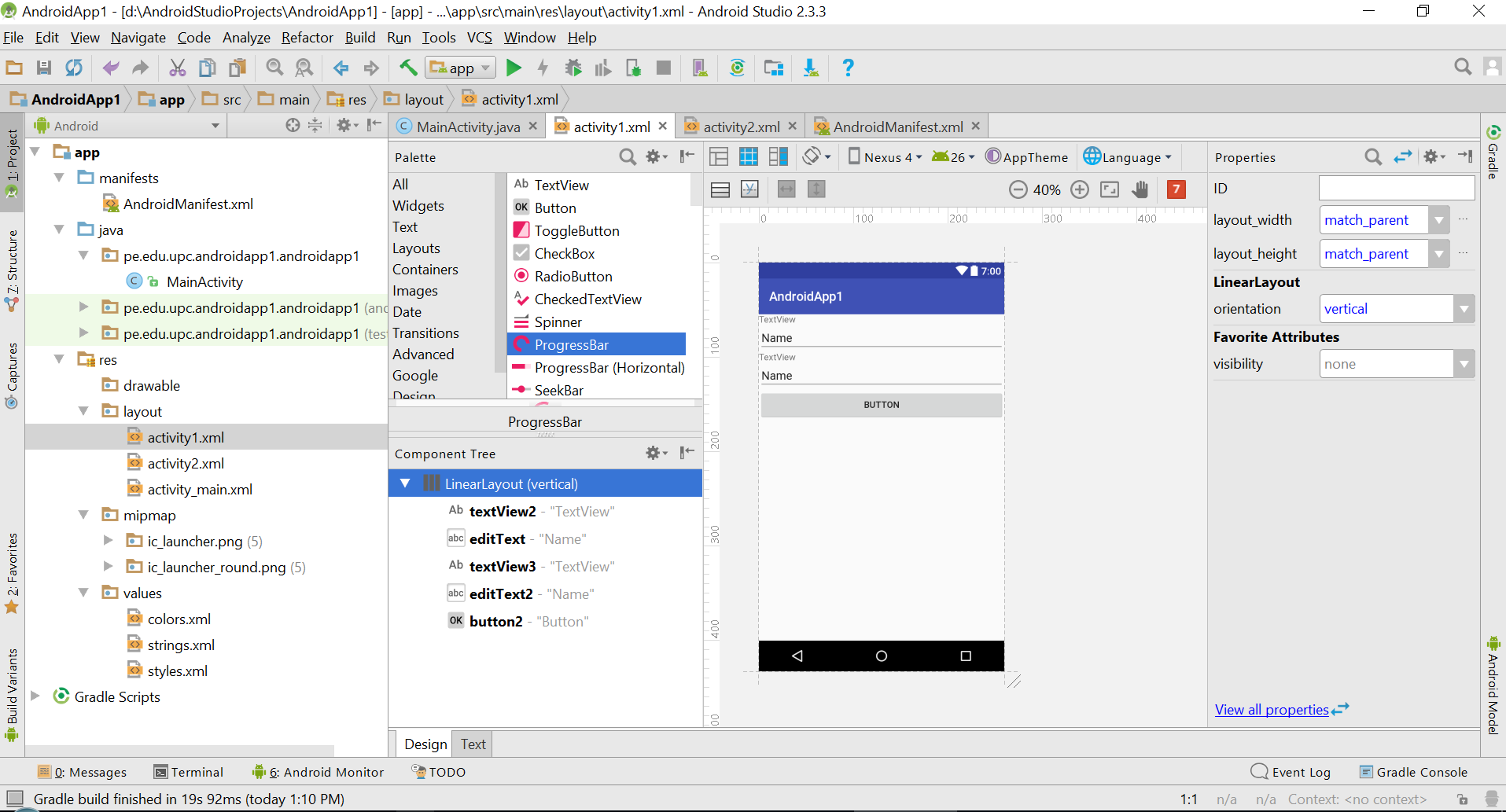


1. El Layout es la forma en la cual se van a organizar los elementos dentro de la pantalla. Para esta primera actividad elegiremos el LinearLayout, orientación vertical.

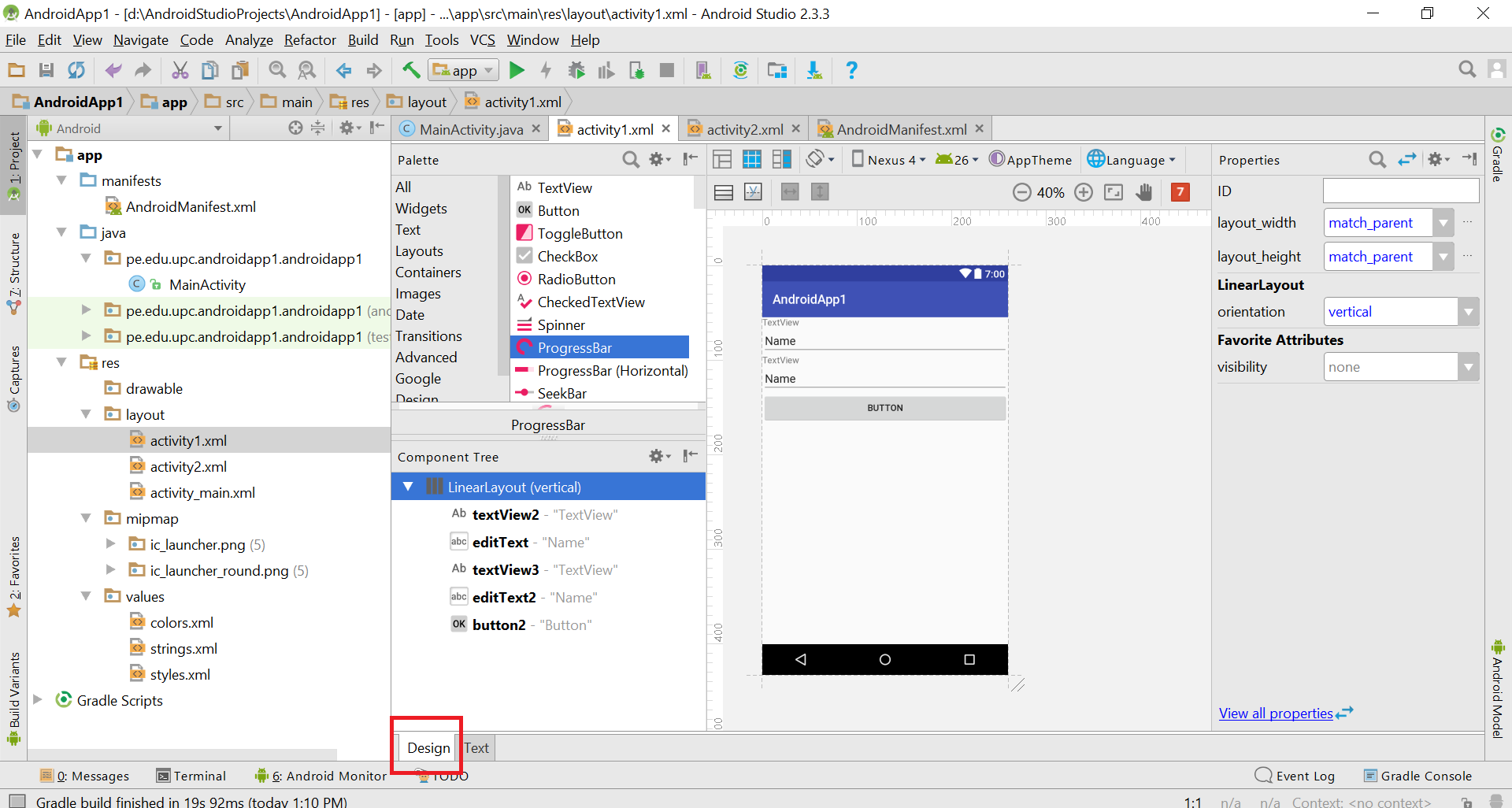


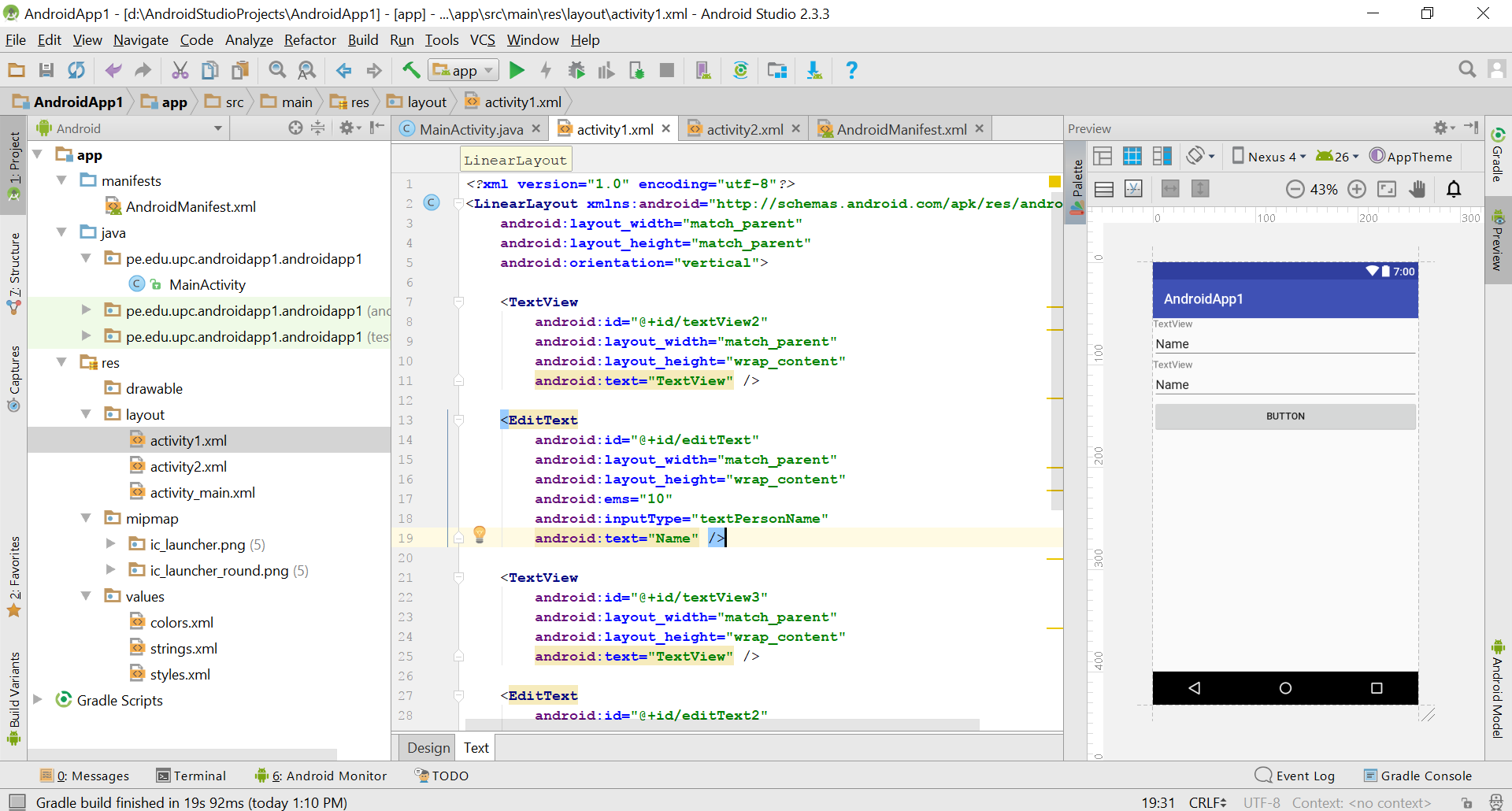
1. Agregaremos los siguientes elementos a la pantalla:

* Una etiqueta(TEXTVIEW)
* Una Caja de Texto(PLAIN TEXT)
* Una segunda etiqueta(TEXTVIEW)
* Una segunda caja de texto(PLAIN TEXT)
* Un botón(BUTTON).

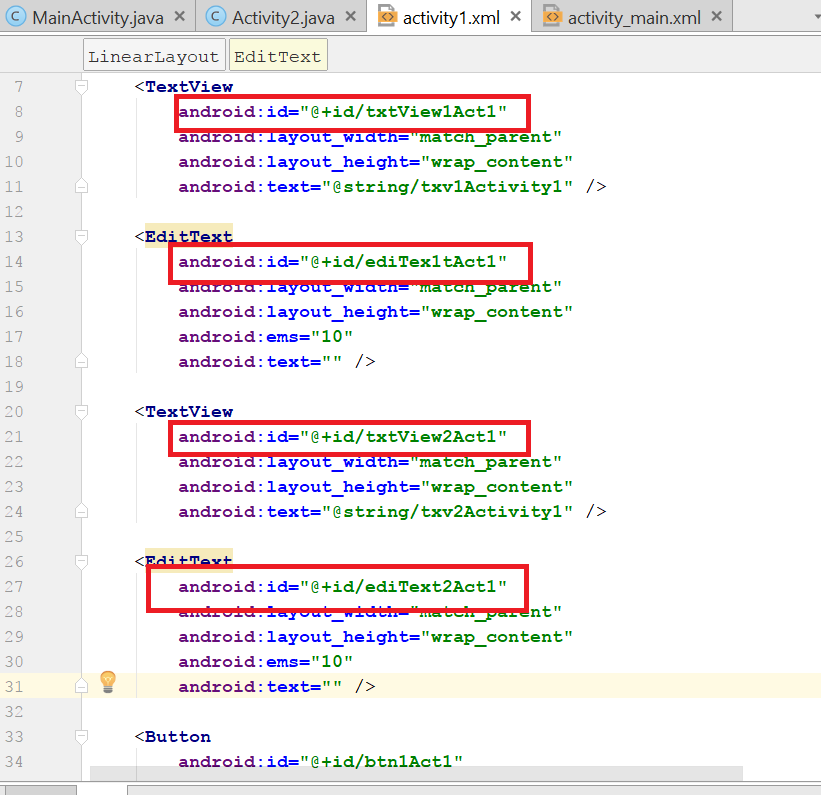


La interface creada podrá verse en versión XML. Para eso cambie la vista de Diseño a Texto.

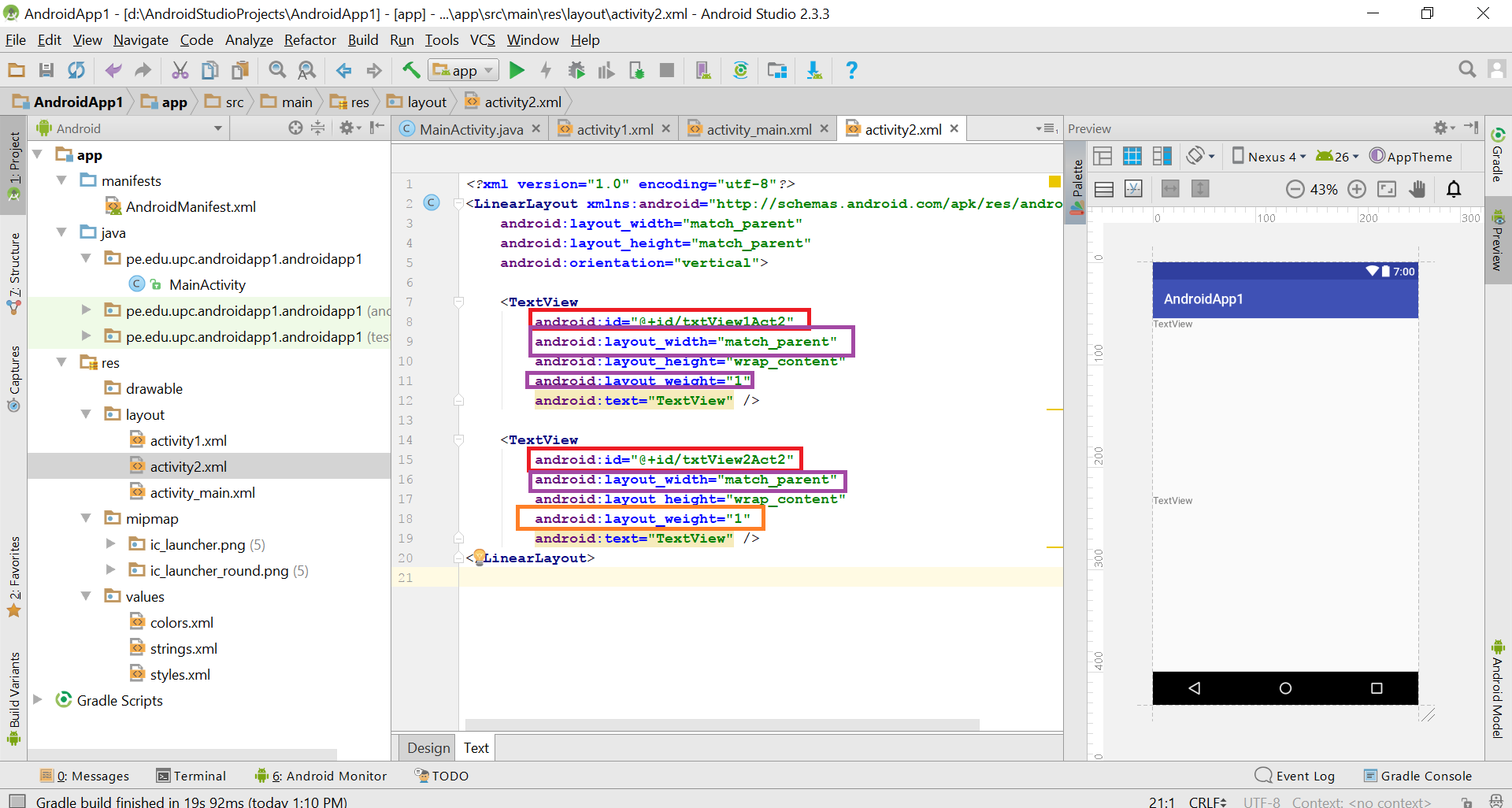




1. Es necesario asignar un nombre a cada identificador de cada componente. Esto para que pueda ser invocado desde las clases Java y poder trabajar con la información a mostrar-leer. Cambiar los valores de los identificadores de la siguiente manera:

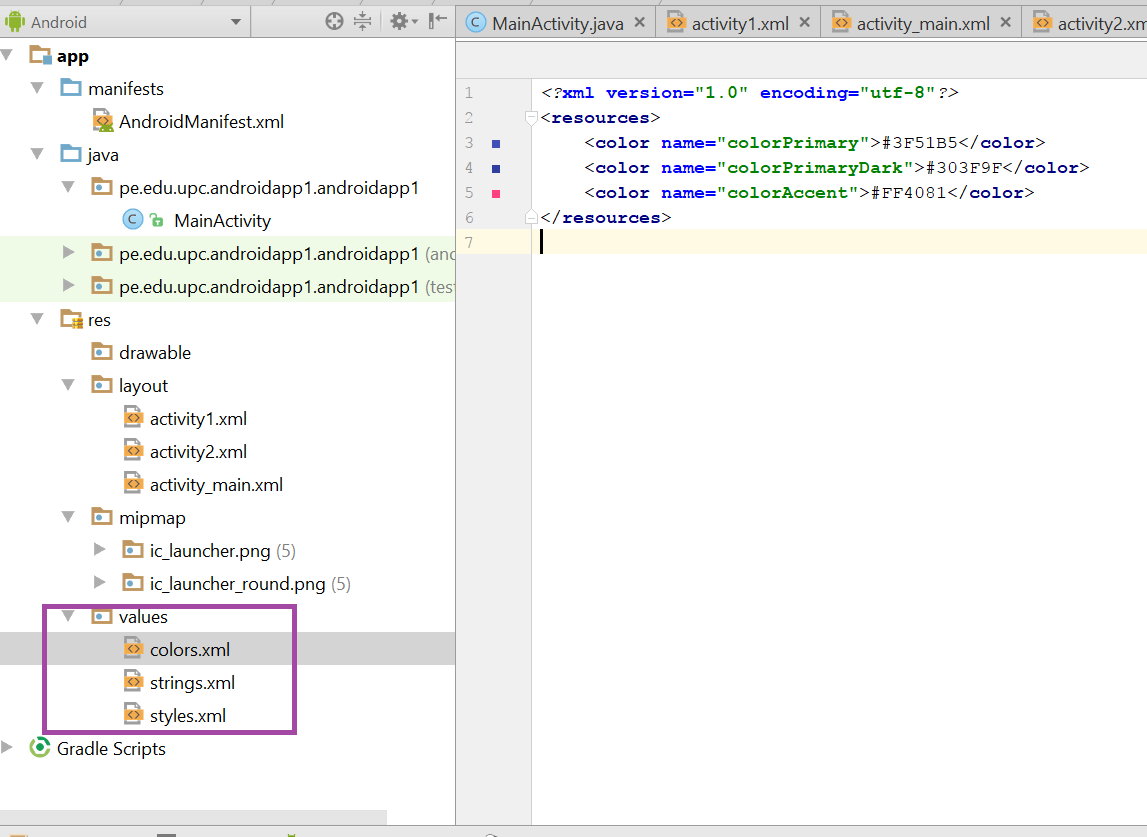


1. Editar la segunda pantalla **activity2** con 2 TextView, el primero que tenga el id txtView1Act2 y txtView2Act2. Adicionalmente el primer texto tiene que ocupar el 50% de el tamaño de pantalla y el segundo el 50% restante. Para esto vamos a agregar la propiedad **weight** y un valor de **1** a cada uno. Además si queremos que el elemento ocupe el 100% del espacio de la pantalla a lo ancho, la propiedad **layout\_width** debe de tener el valor de **match\_parent**

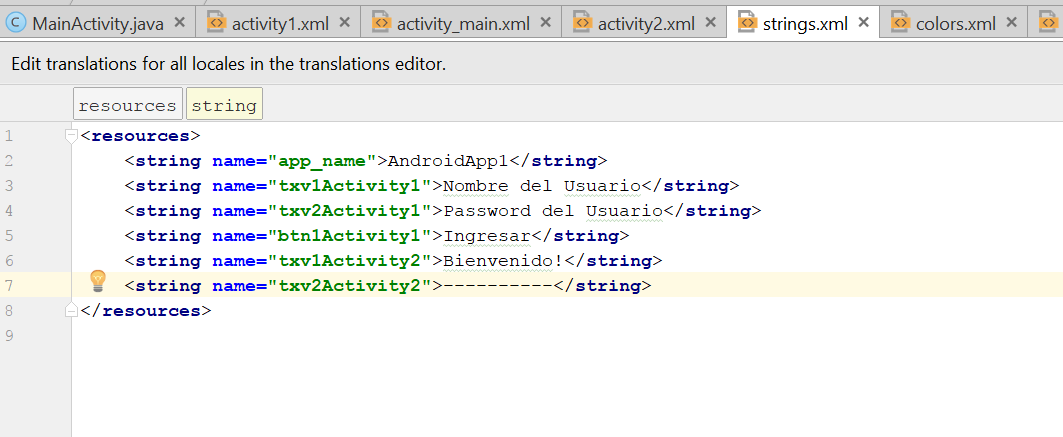


**UTILIZAR ARCHIVO RECURSOS – ARCHIVO STRING-COLORS-STYLES**

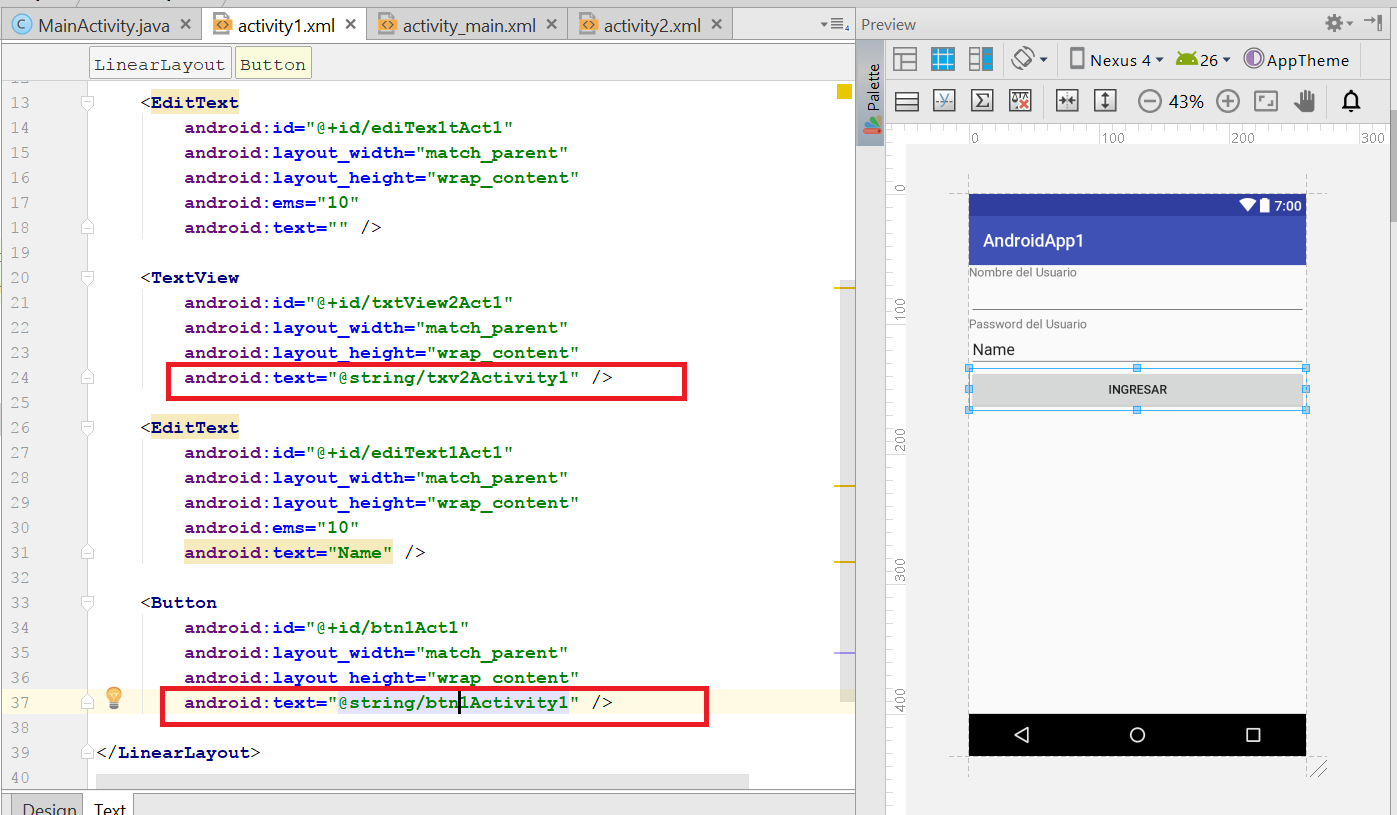
Como parte de los recursos que nos ofrece Android, tenemos 3 archivos que nos permiten configurar colores propios, textos así como un estilo completo a ser utilizado en la capa de presentación. En esta guía trabajaremos con colores y textos.

****

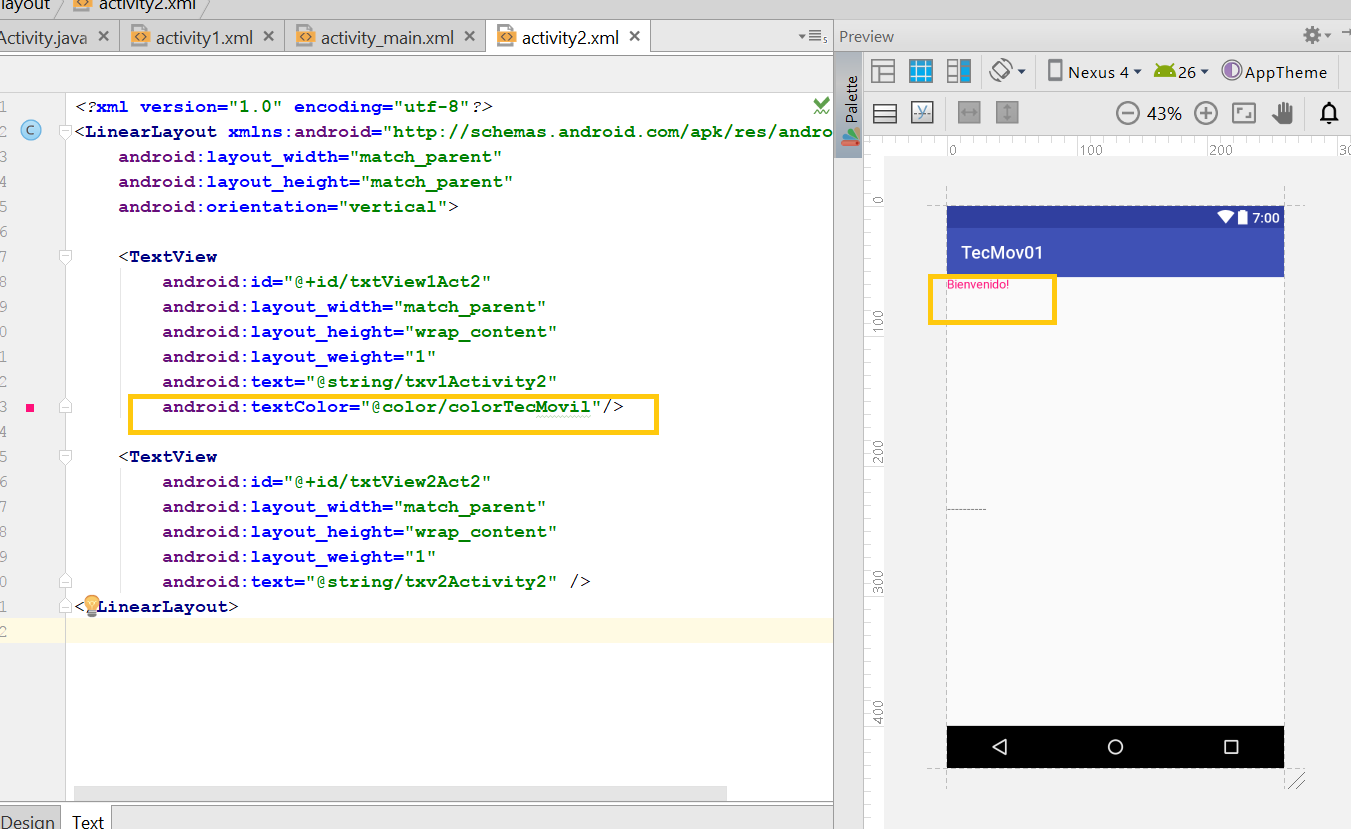
1. Archivo String: Este archivo nos da por defecto un tag con su respectivo valor(nombre del aplicativo). En este caso vamos a agregar 5 etiquetas, que corresponerán 3 a la primera pantalla desarrollada y 2 a la segunda pantalla.



1. Reemplazar los valores en las 2 pantallas desarrolladas. Para poder hacer referencia solo es necesario hacer referencia al valor registrado en el archivo strings. Automaticamente se actualizará el valor en la pantalla de diseño

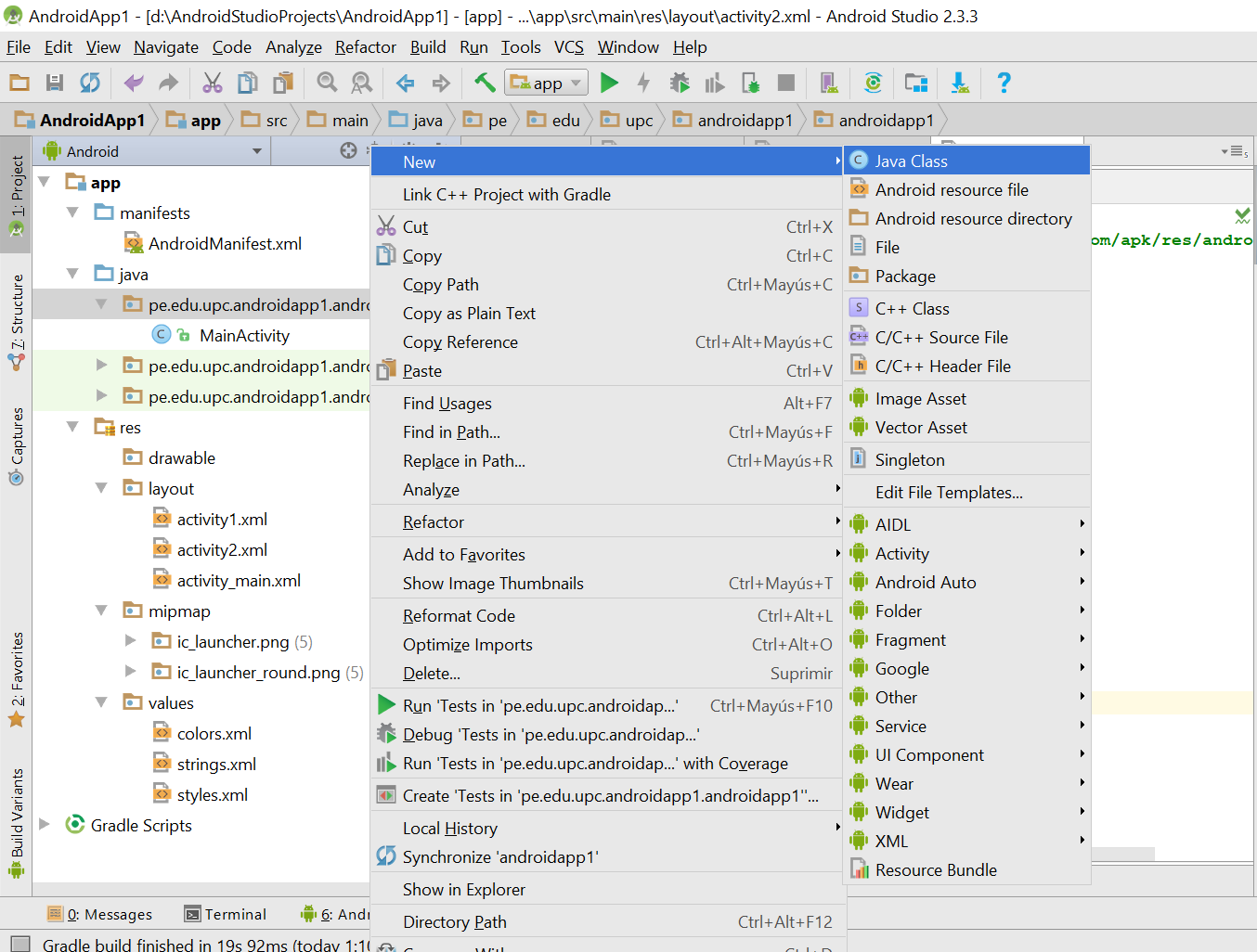


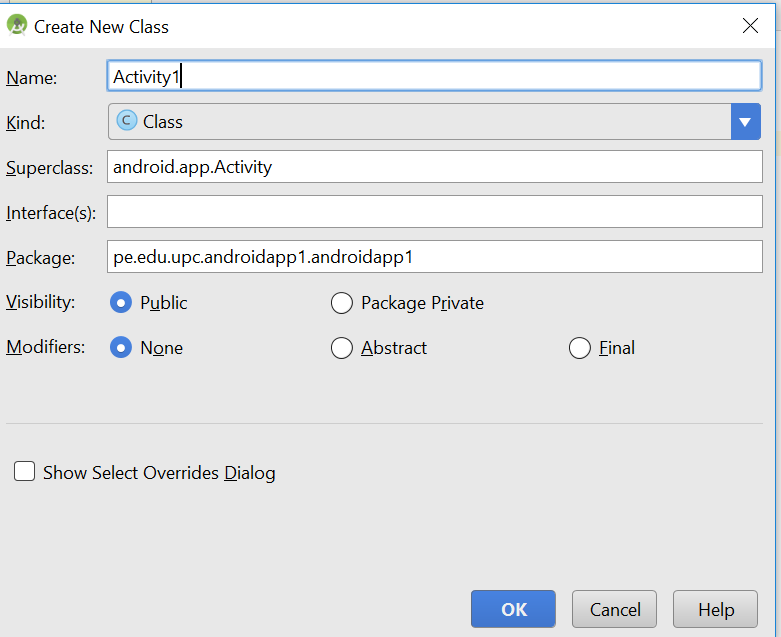
Para el caso del color, en el archivo colors se debe de crear un TAG con su valor. Con el tag creado, agregar en la segunda pantalla para el primer textview el atributo color.



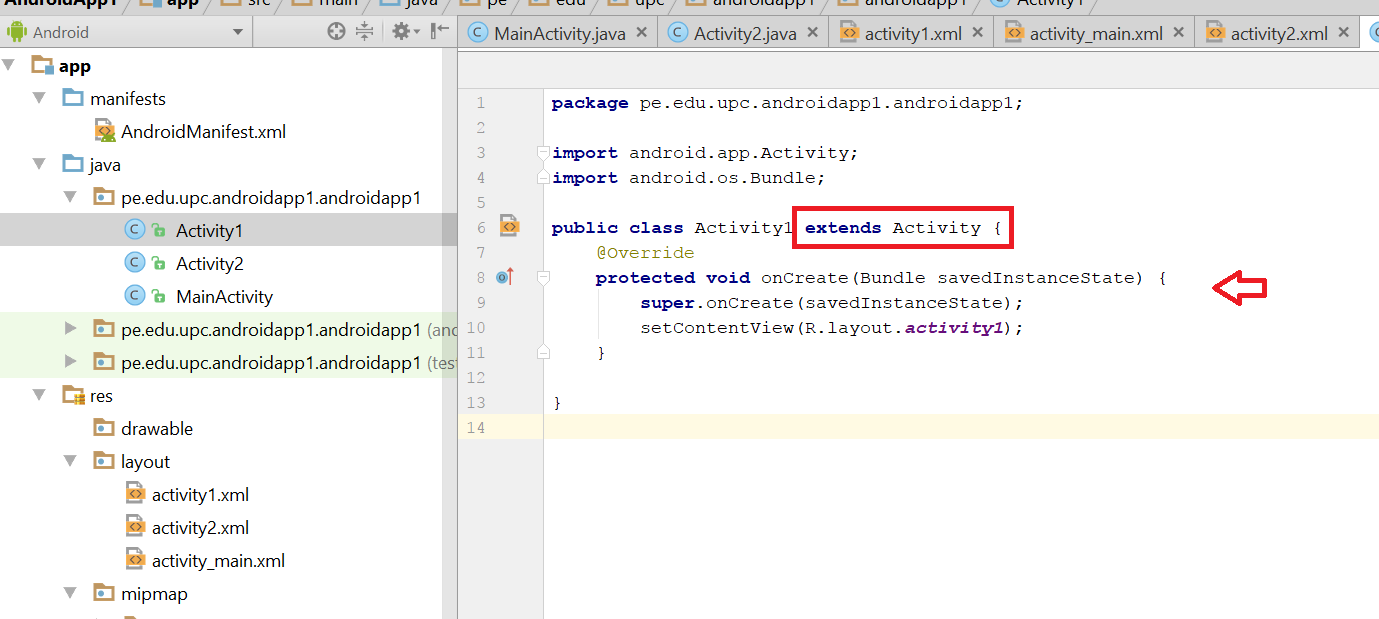
PARTE 2 – CREACION DE CLASES

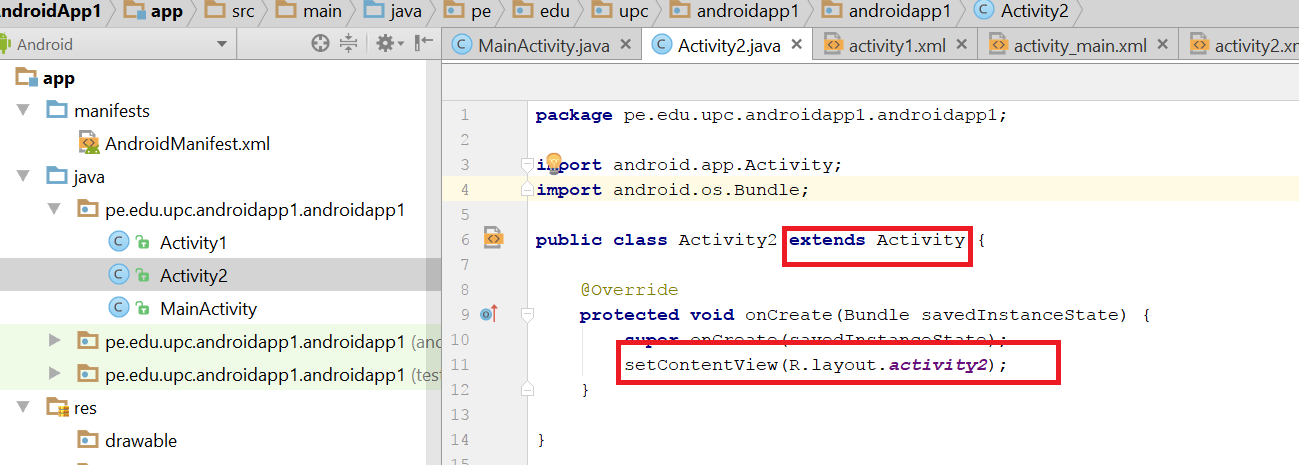
1. Toda la lógica que se quiera colocar en el aplicativo, tiene que ser mediante archivos Java. Para esto debemos de crear 2 clases. La primera manejará la data y eventos de nuestra primera pantalla y la segunda clase manejará la segunda pantalla. Para esto debe seguir los siguientes pasos para crear las clases Java:



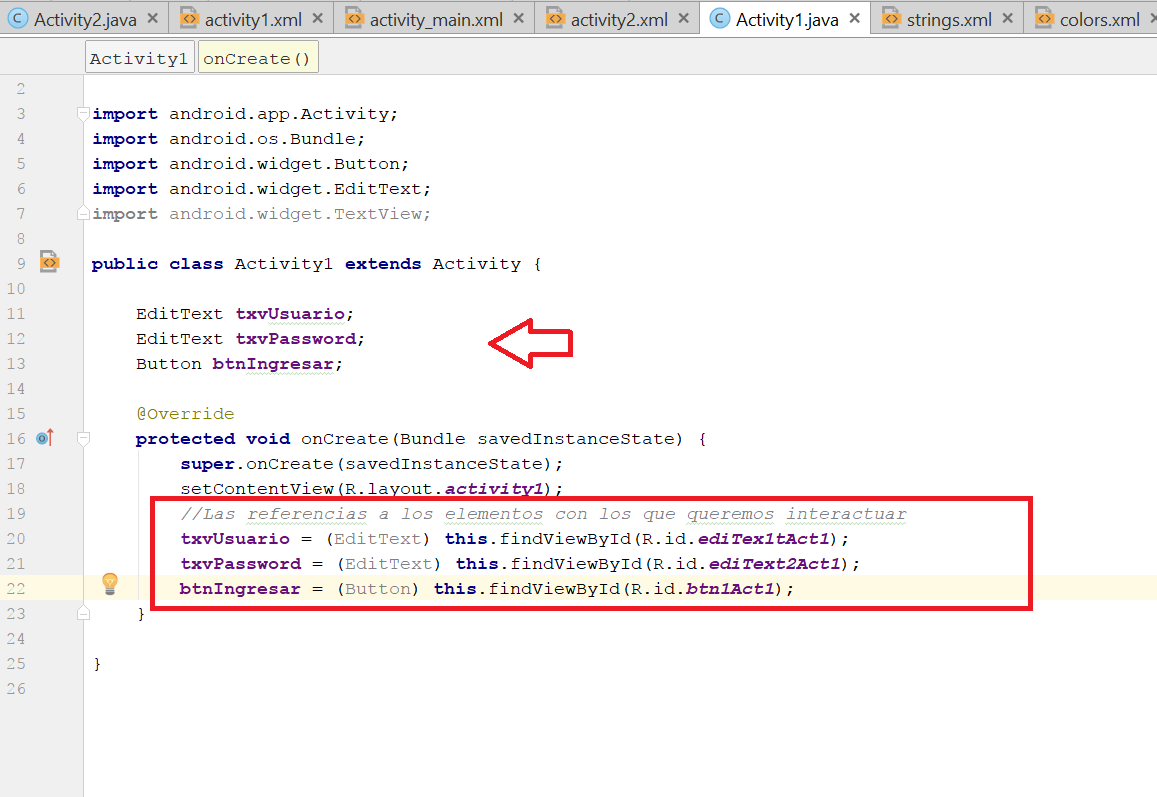


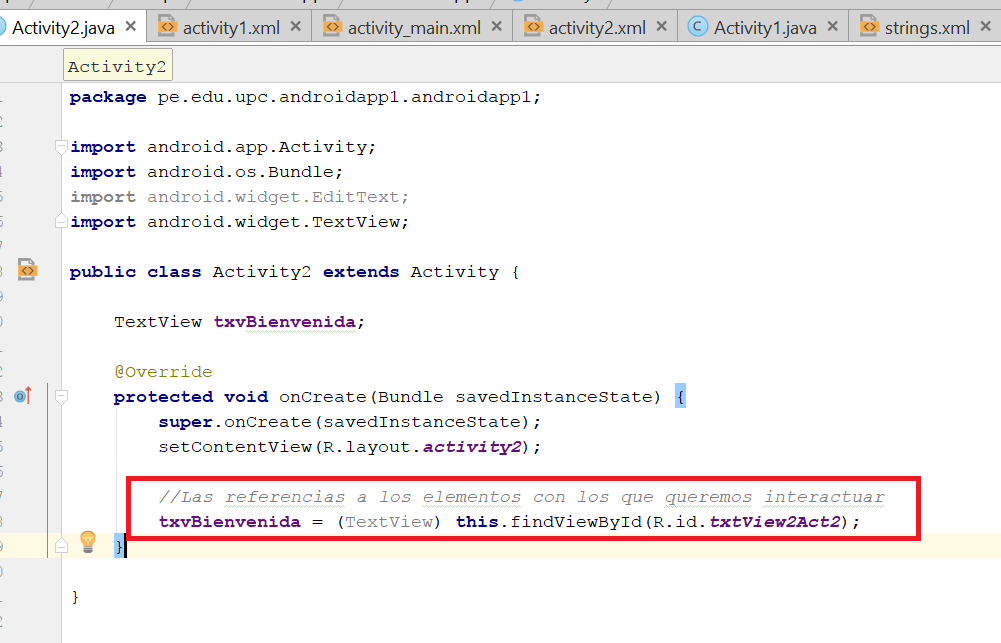
1. Cuando se creen tanto la clase Activity 1 y 2, validar que ambas clases extiendan de la clase Activity. Adicionalmente sobreescribir el método oncreate. Para la pantalla 1, como va a hacer referencia al xml **activity1**, se debe de colocar en el método setContentView la referencia a activity1. Lo mismo realizar para la clase Activity2. De esta manera controlaremos 1 a 1 la información y eventos de cada pantalla diseñada.



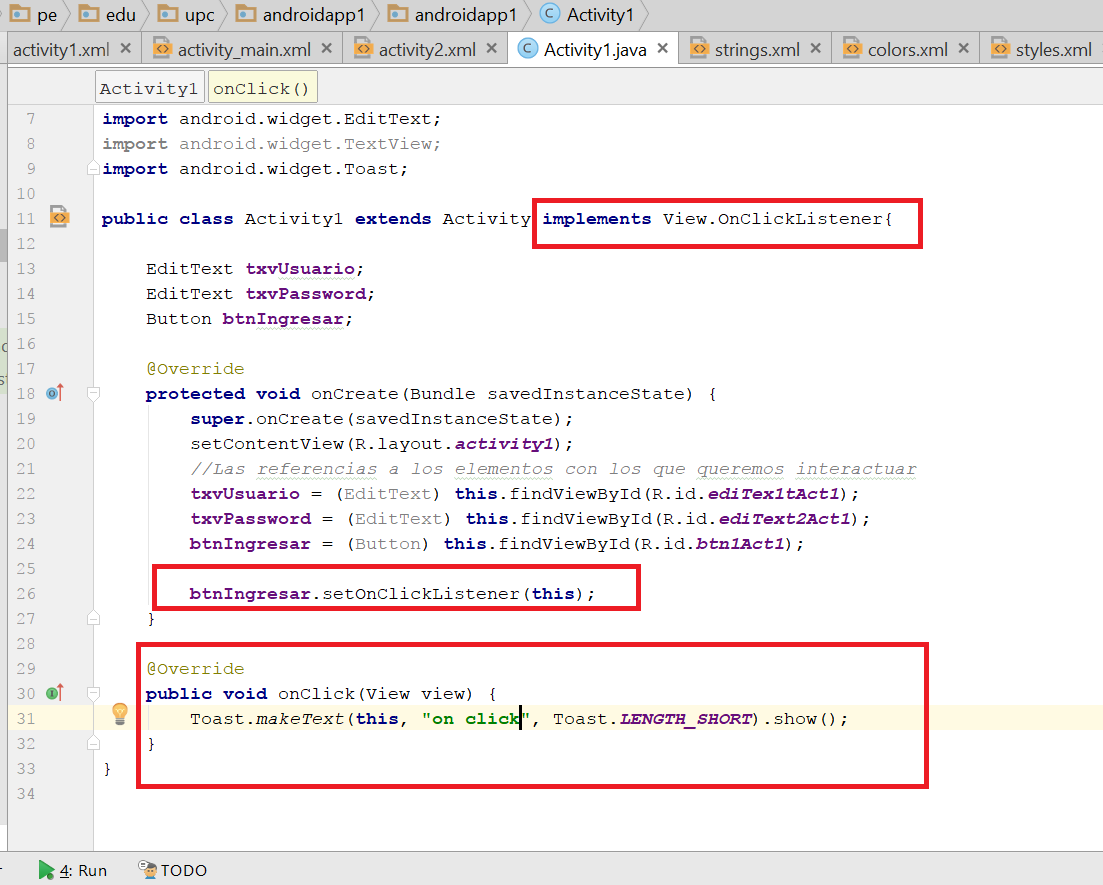


1. Para obtener la referencia a los elementos del formulario, es necesario realizar una referencia manual. Para esto declaramos en el formulario 1 los elementos a referenciar. Lo mismo realizamos en el formulario 2.





Adicionalmente, para detectar el evento de OnClick en el botón de logeo es necesario implementar el Listener respectivo. En este caso haremos que la clase Activity1 implemente la interfase Listener.



Finalmente para invocar desde el Activity1 al Activity2 cuando se presione el onClick en el botón, es necesario ejecutar la siguiente instrucción:

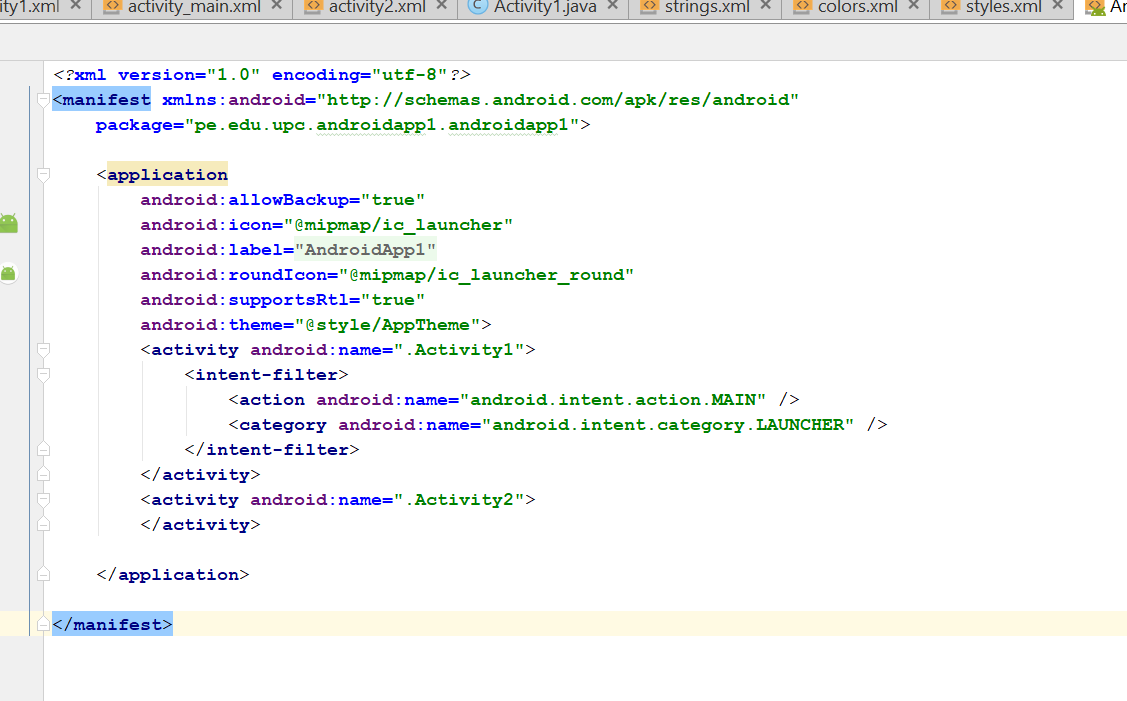


PARTE 3 – DECLARACION DE ARCHIVO MANIFEST

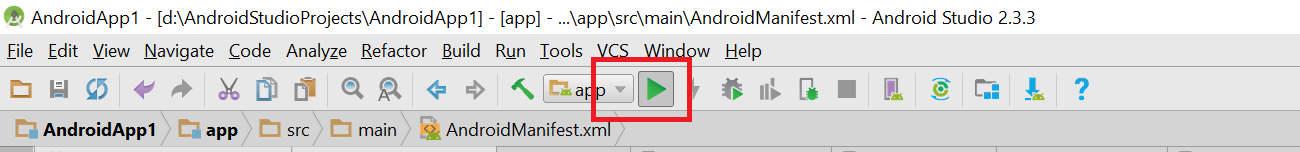
En el archivo Manifest debemos de declarar todas las actividades que vayamos a utilizar dentro del aplicativo. Si alguna no se declara, entonces nos saldrá un error en tiempo de ejecución.

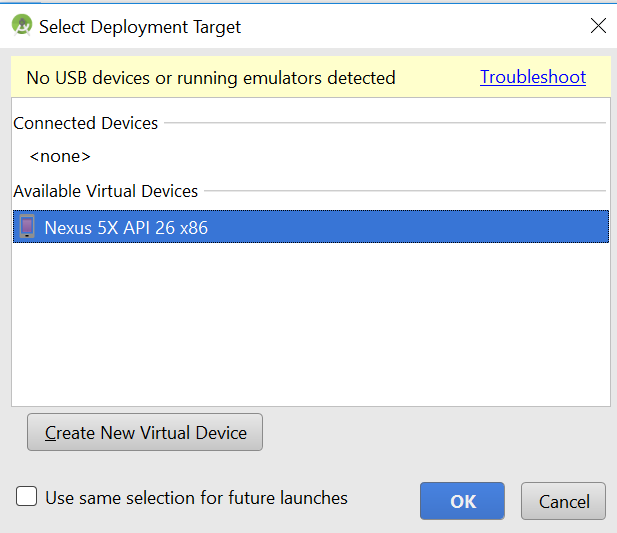
Adicionalmente, se debe de declarar dentro de **una sola actividad** que contenga el **intent-filter**, esta notación le indicará a Android que esa será la primera pantalla que se ejecute al iniciar el aplicativo.

Su archivo Manifest debe de quedar de la siguiente manera



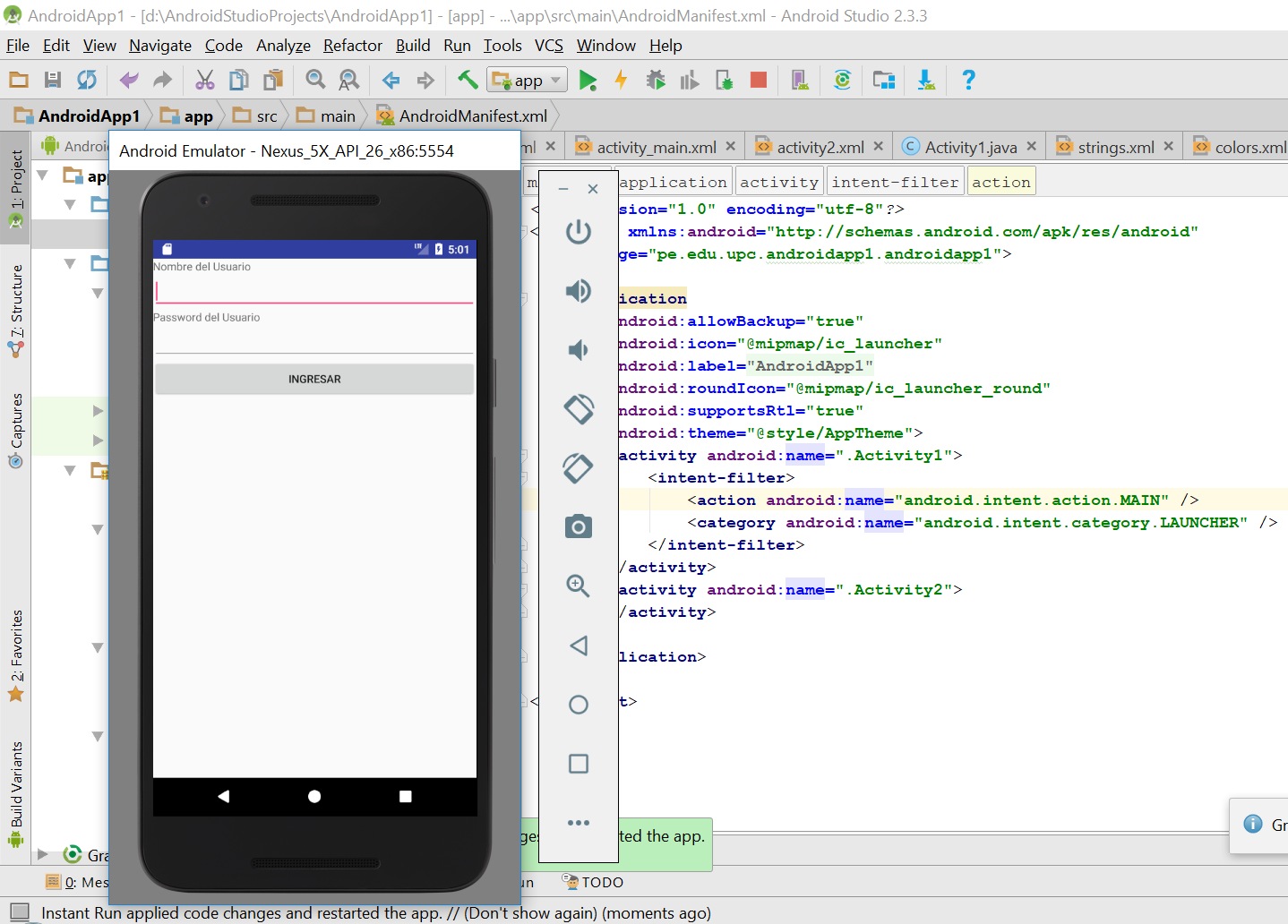
PASO 4 – EJECUTAR EL EMULADOR.





En caso de que no tenga un dispositivo virtual disponible, puede crear uno seleccionando la opción Create New Virtual Device. Se recomienda que el dispositivo maneje el mismo SDK del proyecto.

Finalmente se ejecutará el emulador y se mostrará el aplicativo creado.



**TAREA**

1. Cree una clase que permita ser un controlador. Dicha clase permitirá que el usuario ingresado en la pantalla 1 pueda ser vista dentro del saludo de bienvenida de la pantalla 2.
2. Agregar un botón en la primera pantalla que permita mostrar una tercera pantalla que sea de recuperación de usuario. Crear la pantalla de recuperación de usuario.
3. Validar que al presionar el botón de Ingreso en la pantalla 1, se valide que los datos estén completos o no. En caso de que este incompleto, mostrar un texto en rojo debajo del botón de ingreso que indique que los datos están incompletos.