ANDROID - INTRODUCCIÓN A ANDROID

INTRODUCCIÓN

CONTENIDO

Part I.
Primera Aplicación para Android

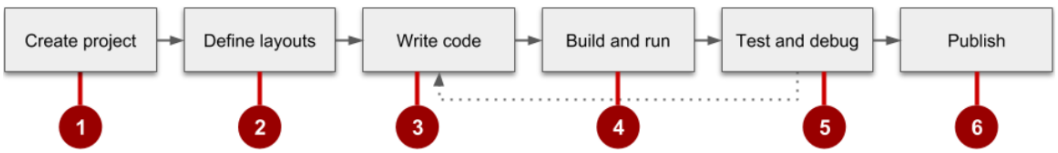
OBJETIVOS

Vamos a describir como se desarrolla aplicaciones usando Android Studio, que es un entorno de desarrollo integrado (IDE) para Android.

I.1 PROCESO DE DESARROLLO

Un proyecto de aplicación para Android comienza con una idea y una definición de los requisitos necesarios para realizar esa idea. Es necesario esbozar interfaces de usuario (UI) para las diversas funciones de la aplicación. Para mostrar cómo se vería una interfaz de usuario y cómo funcionaría, vamos a realizar dibujos, maquetas y prototipos.

Cuando esté listo para comenzar a codificar, use Android Studio para seguir los siguientes pasos:



1. Cree el proyecto en Android Studio y elija una plantilla adecuada.

2. Defina un diseño para cada pantalla que tenga elementos de IU. Puede colocar elementos de la interfaz de usuario en la pantalla utilizando el editor de diseño o puede escribir código directamente en el lenguaje de marcado extensible (XML).

3. Escriba código utilizando el lenguaje de programación Java. Cree código fuente para todos los componentes de la aplicación.

4. Cree y ejecute la aplicación en dispositivos reales y virtuales. Utilice la configuración de compilación predeterminada o cree compilaciones personalizadas para diferentes versiones de su aplicación.

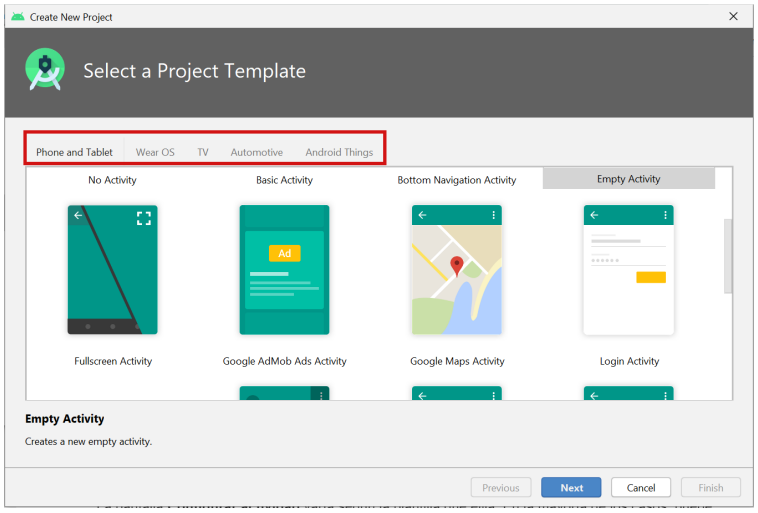
5. Pruebe y depure la lógica y la interfaz de usuario de la aplicación.

6. Publique la aplicación ensamblando el APK final (archivo de paquete) y distribuyéndolo a través de canales como Google Play.

1.2 INICIAR UN PROYECTO DE ANDROID STUDIO

Una vez que haya instalado con éxito el IDE de Android Studio, haga doble clic en el icono de la aplicación de Android Studio para iniciarlo. Haga clic en Iniciar un nuevo proyecto de Android Studio. Lo primero a realizar es elegir los dispositivos de destino donde se van a ejecutar la app.

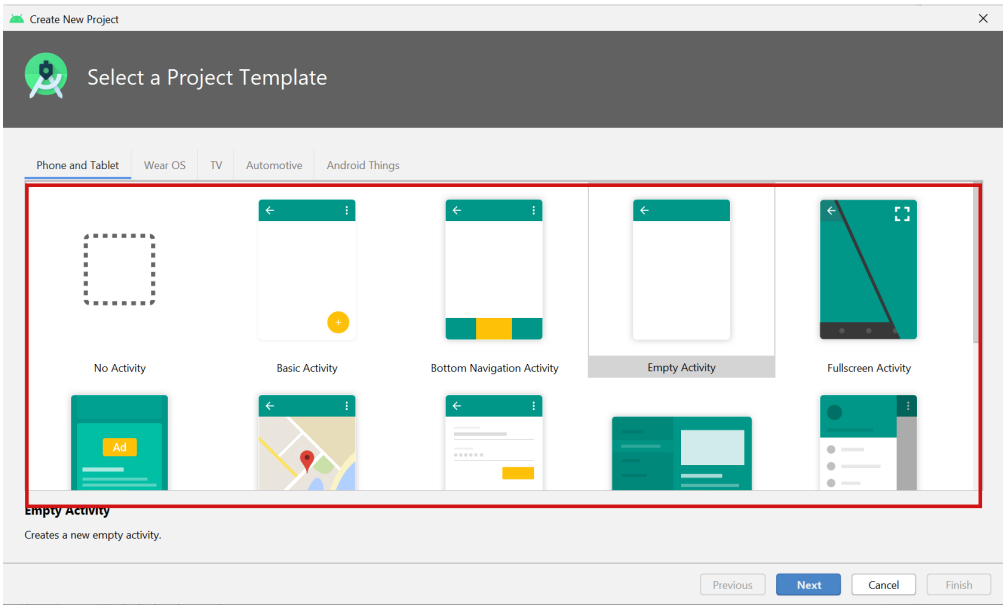
Al elegir Dispositivos Android de destino, el teléfono y la tableta se seleccionan de forma predeterminada las distintas plantillas de las actividades que se puede elegir. Solo tenemos que desplazarnos por las distintas pestañas.



1.3 ELEGIR UNA PLANTILLA DE ACTIVIDAD

Una **Activity** es un componente crucial de cualquier aplicación de Android. Una Activity normalmente tiene un diseño asociado a él que define cómo aparecen los elementos de la interfaz en una pantalla.

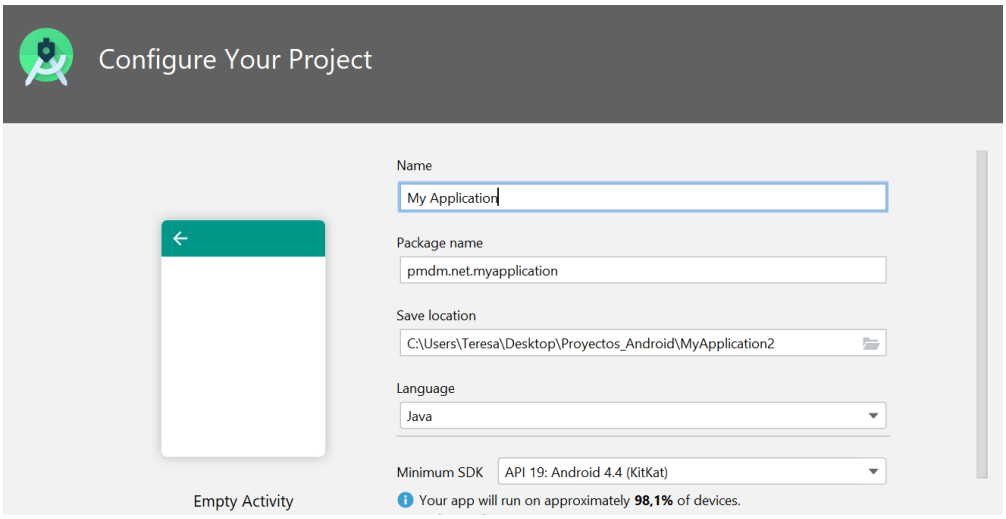
Android Studio completa previamente su proyecto con un código mínimo para un Activity diseño. Las Activity plantillas disponibles van desde una plantilla prácticamente en blanco (Agregar sin actividad) hasta una Activity que incluye navegación y un menú de opciones.



Puede personalizar el Activity después de seleccionar su plantilla. Por ejemplo, la opción Actividad vacía proporciona un único Activity recurso de diseño para la pantalla. La pantalla Configurar actividad aparece después de hacer clic en Siguiente . En la pantalla Configurar actividad , puede aceptar el nombre de uso común para Activity(como MainActivity), o puede cambiar el nombre.

Tenemos que indicar el nombre del proyecto.

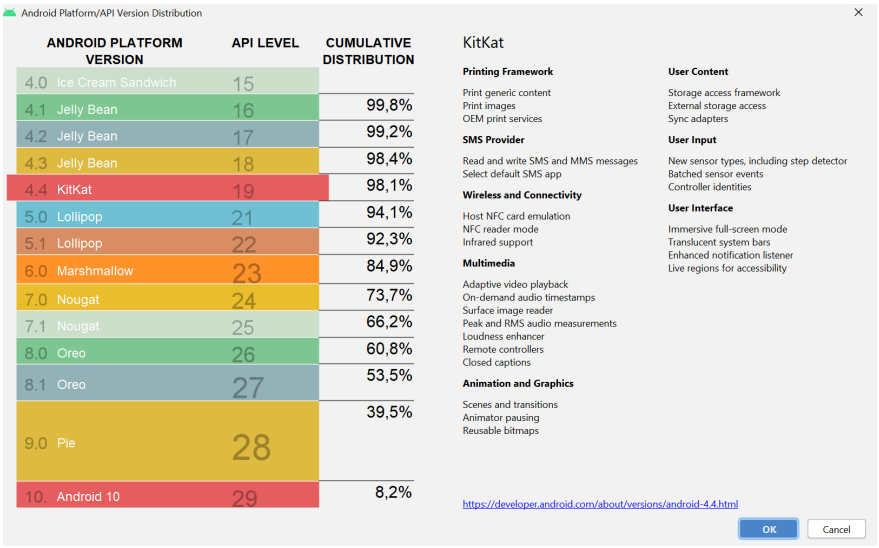
Elegir un dominio de empresa que debe ser único , tenga en cuenta que las aplicaciones publicadas en Google Play deben tener un nombre de paquete único. Debido a que los dominios son únicos, anteponer el nombre de la aplicación con su nombre, o el nombre de dominio de su empresa, debe proporcionar un nombre de paquete adecuada-



mente único. Si no planea publicar la aplicación, puede aceptar el dominio de ejemplo predeterminado.

Tambien debemos de indicar donde vamos a guardar las app y por ultimo elegimos la version donde se van a ejecutar la app.

El SDK mínimo declara la versión mínima de Android para su aplicación. Cada versión sucesiva de Android proporciona compatibilidad para aplicaciones que se crearon con las API de versiones anteriores. Eso significa que su aplicación siempre debe ser compatible con futuras versiones de Android, si usa las API de Android documentadas.

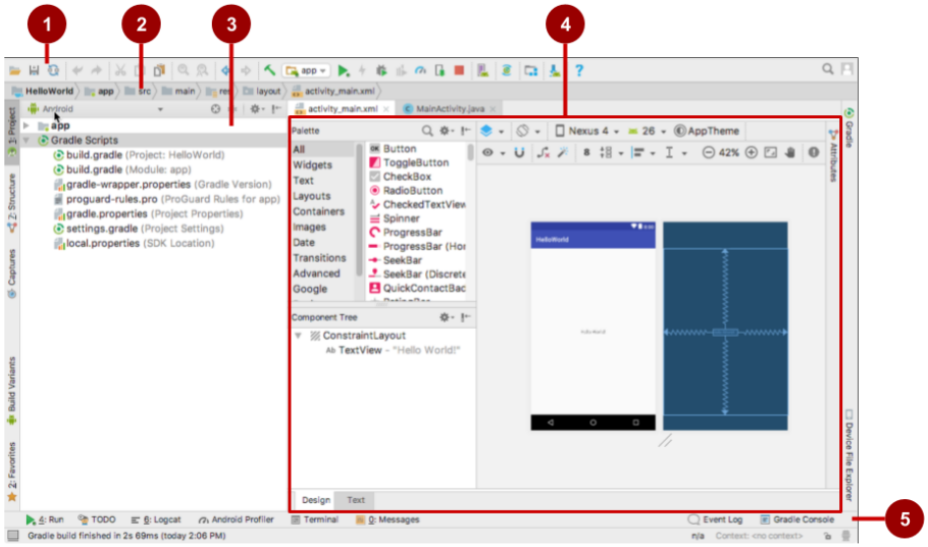


Android Studio crea una carpeta para tus proyectos y compila el proyecto con Gradle

.

1.4 EXPLORANDO UN PROYECTO

Un proyecto de Android Studio contiene todo el código fuente y todos los recursos de una aplicación. Los recursos incluyen diseños, cadenas, colores, dimensiones e imágenes. La ventana principal de Android Studio está formada por varias áreas lógicas o paneles , como se muestra en la figura siguiente.



En la figura de arriba:

1. Barra de herramientas: proporciona una amplia gama de acciones, incluida la ejecución de la aplicación de Android y el lanzamiento de herramientas de Android.

2. Barra de navegación: navega por el proyecto y abre archivos para editarlos.

3. Panel del proyecto : muestra los archivos del proyecto en una jerarquía. La jerarquía seleccionada en la figura anterior es Android .

4. Editor: el contenido de un archivo seleccionado en el proyecto. Por ejemplo, después de seleccionar un diseño (como se muestra en la figura anterior), el panel del editor muestra el editor de diseño con herramientas para editar el diseño. Después de seleccionar un archivo de código Java, el panel del editor muestra el código Java con herramientas para editar el código.

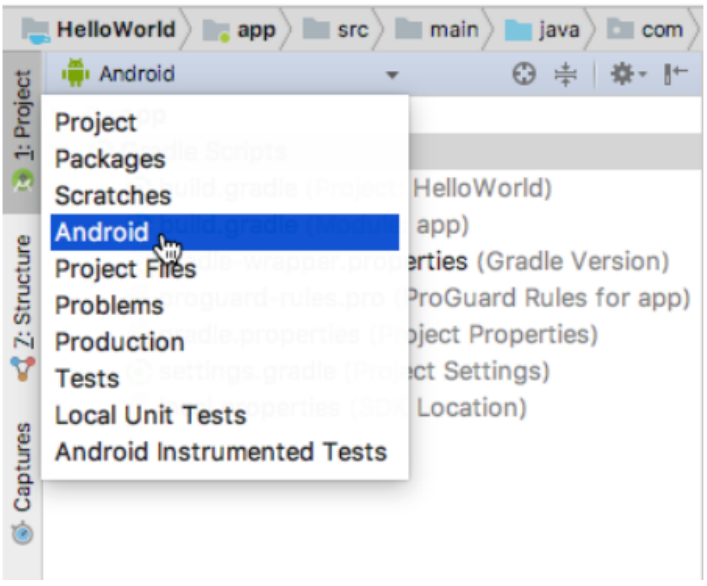
5. Pestañas a lo largo de la izquierda, la derecha y la parte inferior de la ventana: puede hacer clic en las pestañas para abrir otros paneles, como Logcat para abrir el panel Logcat con mensajes de registro, o TODO para administrar tareas.

La barra de estado en la parte inferior de la ventana de Android Studio muestra el estado del proyecto y del propio Android Studio, así como las advertencias o mensajes. Puede ver el progreso de la construcción en la barra de estado.

1.5 USANDO EL PANEL PROYECTO

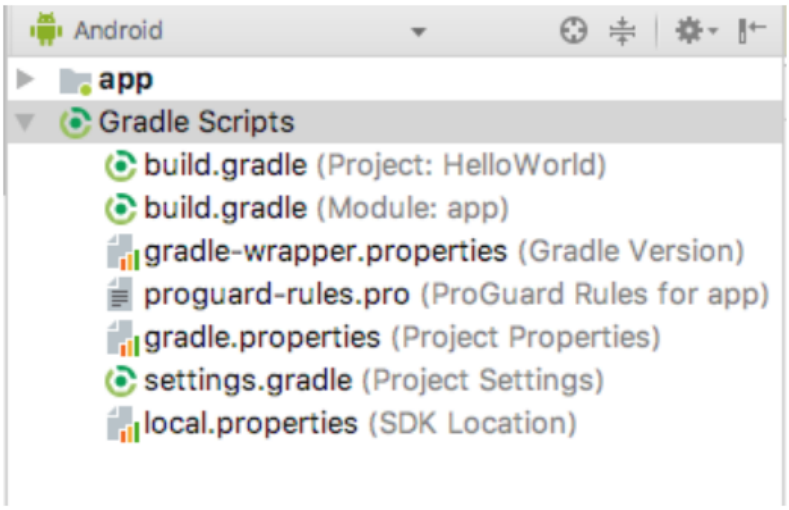
Puede ver la organización del proyecto de varias formas en el panel Proyecto. Si aún no está seleccionado, haga clic en la pestaña Proyecto . (La pestaña Proyecto está en la columna de pestañas vertical en el lado izquierdo de la ventana de Android Studio).

Aparece el panel Proyecto. Para ver el proyecto en la jerarquía de proyectos estándar de Android, seleccione Android en la flecha hacia abajo en la parte superior del panel Proyecto .



1.6 ARCHIVOS GRADLE

Cuando crea por primera vez un proyecto de aplicación, el panel **Proyecto> Android** aparece con la Gradle Scripts carpeta expandida como se muestra a continuación. Esta carpeta contiene todos los archivos necesarios para el sistema de compilación.



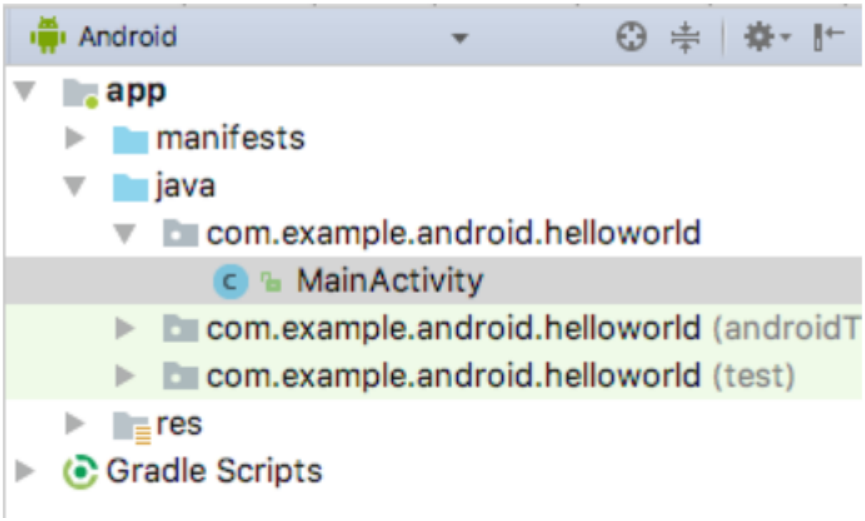
El build.gradle(Module:app)archivo especifica bibliotecas adicionales y la configuración de compilación del módulo. Este archivo también incluye una lista de dependencias , que son bibliotecas requeridas por el código,

1.7 CÓDIGO DE LA APLICACIÓN

Para ver y editar el código Java, expanda la appcarpeta, la javacarpeta y la com.example.android.hellow Haga doble clic en el MainActivity javaarchivo para abrirlo en el editor de código.

La javacarpeta contiene tres subcarpetas:

- com.example.hello.helloworld(o el nombre de dominio que haya especificado) : todos los archivos de un paquete están en una carpeta que lleva el nombre del paquete. Para su aplicación Hello World, hay un paquete y solo contiene MainActivity.java. La primera Activity(pantalla) que ve el usuario, que también inicializa los



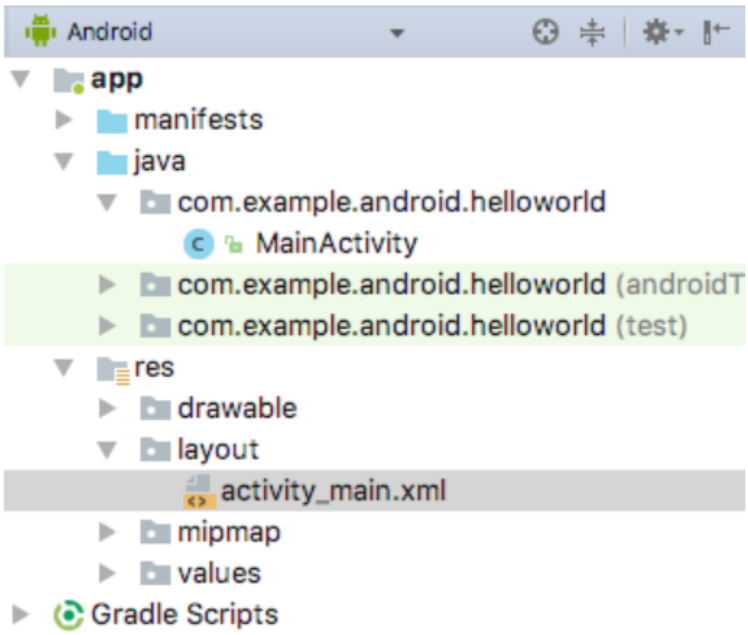
recursos de toda la aplicación, se llama habitualmente MainActivity. (La extensión del archivo se omite en el panel Proyecto> Android).

- com.example.hello.helloworld(androidTest): Esta carpeta es para sus pruebas instrumentadas y comienza con un archivo de prueba esqueleto.
- com.example.hello.helloworld(test): Esta carpeta es para sus pruebas unitarias y comienza con un archivo de prueba unitaria esqueleto creado automáticamente.

I.8 ARCHIVOS DE DISEÑO

Para ver y editar un archivo de diseño, expanda la rescarpeta y la layoutcarpeta para ver el archivo de diseño. En la figura siguiente, se llama al archivo de diseño activity_main.xml.

Haga doble clic en el archivo para abrirlo en el editor de diseño. Los archivos de diseño están escritos en XML.



1.9 ARCHIVOS DE RECURSOS

La res carpeta contiene recursos, como diseños, cadenas e imágenes. Una Activityse asocia generalmente con un diseño de interfaz de usuario de puntos de vista que se define como un archivo XML. Este archivo XML generalmente recibe su nombre Activity. La rescarpeta incluye estas subcarpetas:

- drawable: Almacena todas las imágenes de tu aplicación en esta carpeta.
- layout: Cada Activitytiene al menos un archivo de diseño XML que describe la interfaz de usuario. Para Hello World, esta carpeta contiene activity_main.xml.
- mipmap: Los iconos del lanzador se almacenan en esta carpeta. Hay una subcarpeta para cada densidad de pantalla admitida. Android usa la densidad de pantalla (la cantidad de píxeles por pulgada) para determinar la resolución de imagen requerida. Android agrupa todas las densidades de pantalla reales en densidades generalizadas, como media (mdpi), alta (hdpi) o extra extra extra alta (xxxhdpi). La ic_launcher.pngcarpeta contiene los iconos de iniciador predeterminados para todas las densidades admitidas por su aplicación.

- values: En lugar de codificar valores como cadenas, dimensiones y colores en sus archivos XML y Java, se recomienda definirlos en sus respectivos values archivos. Esta práctica hace que sea más fácil cambiar los valores y mantener los valores consistentes en su aplicación.

La values subcarpeta incluye estas subcarpetas:

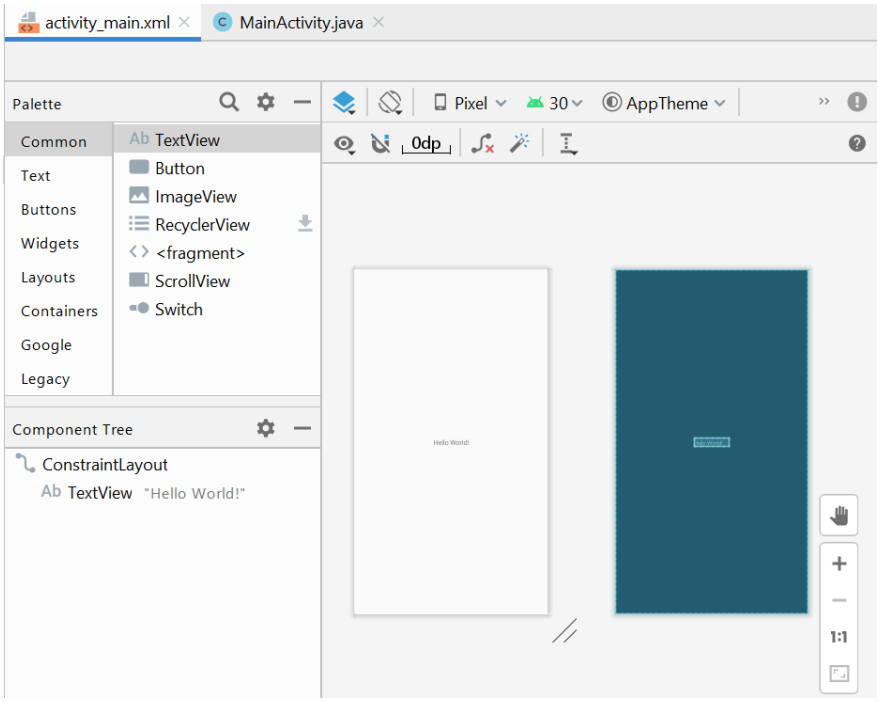
- colors.xml: Muestra los colores predeterminados para el tema elegido. Puede agregar sus propios colores o cambiar los colores según los requisitos de su aplicación.
- dimens.xml: Almacena los tamaños de vistas y objetos para diferentes resoluciones.
- strings.xml: Crea recursos para todas tus cadenas. Hacer esto facilita la traducción de las cadenas a otros idiomas.
- styles.xml: Todos los estilos de tu aplicación y tema van aquí. Los estilos ayudan a que su aplicación tenga un aspecto coherente para todos los elementos de la interfaz de usuario.

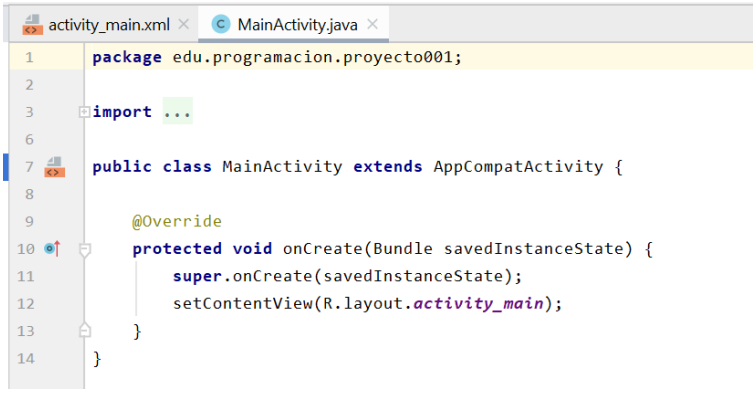
I.10 USANDO EL PANEL DEL EDITOR

Si selecciona un archivo, aparece el panel del editor. Aparece una pestaña para el archivo para que pueda abrir varios archivos y alternar entre ellos. Por ejemplo, si hace doble clic en el archivo de diseño activity_main.xml en el panel Proyecto> Android , el editor de diseño aparece como se muestra a continuación.

Si hace doble clic en el archivo MainActivity en el panel Proyecto> Android , el editor cambia al editor de código como se muestra a continuación, con una pestaña para MainActivity.java :

En la parte superior del MainActivity.java archivo hay una packagedeclaración que define el paquete de la aplicación. Esta declaración de paquete va seguida de un importbloque condensado con ..., como se muestra en la figura anterior. Haga clic en los


activity_main.xml
MainActivity.java
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Google
Legacy
TextView
Button
ImageView
RecyclerView
<fragment>
ScrollView
Switch
Component Tree
ConstraintLayout
TextView "Hello World!"
Pixel
30
AppTheme
0dp
Hello World!
1:1


activity_main.xml
MainActivity.java
1 package edu.programacion.proyecto001;
2
3 import ...
6
7 public class MainActivity extends AppCompatActivity {
8
9 @Override
10 protected void onCreate(Bundle savedInstanceState) {
11 super.onCreate(savedInstanceState);
12 setContentView(R.layout.activity_main);
13 }
14 }

puntos para expandir el bloque y verlo. Las import declaraciones importan las bibliotecas necesarias para la aplicación.

I.11 COMPRENDER EL MANIFIESTO DE ANDROID

Antes de que el sistema Android pueda iniciar un componente de la aplicación como un Activity, el sistema debe saber que Activityexiste. Lo hace leyendo el AndroidManifest.xmlarchivo de la aplicación , que describe todos los componentes de su aplicación de Android. Cada uno Activitydebe estar incluido en este archivo XML, junto con todos los componentes de la aplicación.

Para ver y editar el AndroidManifest.xmlarchivo, expanda la manifestscarpeta en el panel Proyecto> Android y haga doble clic AndroidManifest.xml. Su contenido aparece en el panel de edición:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.programacion.proyecto001" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Proyecto001"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme" >
        <activity android:name=".MainActivity" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

I.11.0.1 *Etiqueta de aplicación y espacio de nombres de Android*

El manifiesto de Android está codificado en XML y siempre usa el espacio de nombres de Android:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.programacion.proyecto001" >
```

La packageexpresión muestra el nombre de paquete exclusivo de la nueva aplicación. No cambie la expresión del paquete después de que se publique la aplicación.

La <application>etiqueta, con su </application>etiqueta de cierre , define la configuración del manifiesto para toda la aplicación.

1.11.1 Copia de seguridad automática

El android:allowBackupatributo habilita la copia de seguridad automática de los datos de la aplicación:

```
android:allowBackup="true"
```

Establecer el android:allowBackup atributo en true permite realizar una copia de seguridad de la aplicación automáticamente y restaurarla según sea necesario. Los usuarios invierten tiempo y esfuerzo para configurar aplicaciones. Cambiar a un nuevo dispositivo puede cancelar toda esa cuidadosa configuración. El sistema realiza esta copia de seguridad automática de casi todos los datos de la aplicación de forma predeterminada, y lo hace sin que el desarrollador tenga que escribir ningún código de aplicación adicional.

1.11.2 El icono de la aplicación

El android:iconatributo establece el icono de la aplicación:

```
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
```

El android:icon atributo asigna a la aplicación un icono en la mipmapcarpeta (dentro de la res carpeta en el panel Proyecto> Android). El icono aparece en la pantalla de

inicio o en la pantalla Buscar aplicaciones para iniciar la aplicación. El icono también se utiliza como icono predeterminado para los componentes de la aplicación.

1.11.3 Recursos de cadenas y etiquetas de la aplicación

El android:labelatributo muestra la cadena "Hello World"resaltada. Si hace clic en la cadena, cambia para mostrar el recurso de cadena @string/app_name:

```
android:label="Proyecto001"
```

Después de abrir el strings.xml archivo, puede ver que el nombre de la cadena app_name está configurado en Hello World. Puede cambiar el nombre de la aplicación cambiando la Hello World cadena por otra.

1.11.4 Tema de la aplicación

El android:theme atributo establece el tema de la aplicación, que define la apariencia de los elementos de la interfaz de usuario, como el texto:

```
android:theme="@style/AppTheme" >
```

El themea tributo se establece en el tema estándar AppTheme.

1.12 EJECUTAR LA APLICACIÓN EN UN EMULADOR O DISPOSITIVO

Con los emuladores de dispositivos virtuales, puede probar una aplicación en diferentes dispositivos, como tabletas o teléfonos inteligentes, con diferentes niveles de API para diferentes versiones de Android, para asegurarse de que se vea bien y funcione para la

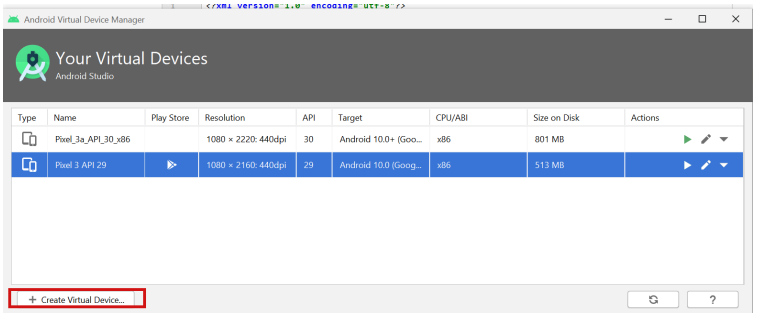
mayoría de los usuarios. No tiene que depender de tener un dispositivo físico disponible para el desarrollo de aplicaciones.

El administrador de dispositivos virtuales Android (AVD) crea un dispositivo virtual o un emulador que simula la configuración de un tipo particular de dispositivo con Android. Utilice AVD Manager para definir las características de hardware de un dispositivo y su nivel de API, y guardarlo como una configuración de dispositivo virtual. Cuando inicia el emulador de Android, lee una configuración específica y crea un dispositivo emulado en su computadora que se comporta exactamente como una versión física de ese dispositivo.

1.12.1 Creando un dispositivo virtual

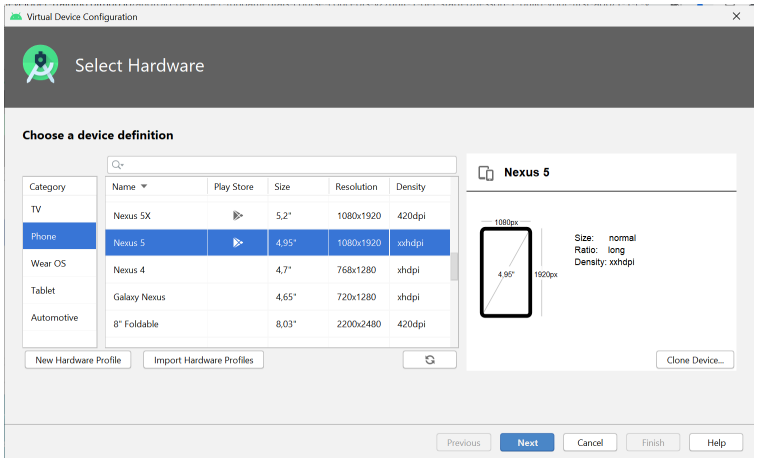
Para ejecutar un emulador en su computadora, use AVD Manager para crear una configuración que describa el dispositivo virtual. Seleccione Herramientas> Android> AVD Manager , o haga clic en el icono AVD Manager Icono de AVD Manager en la barra de herramientas.

Los Sus dispositivos virtuales de pantalla aparece mostrando todos los dispositivos virtuales creados por el usuario. Haga clic en el botón + Crear dispositivo virtual para crear un nuevo dispositivo virtual.



Puede seleccionar un dispositivo de una lista de dispositivos de hardware predefinidos. Para cada dispositivo, la tabla proporciona una columna para su tamaño de pantalla diagonal (Tamaño), resolución de pantalla en píxeles (Resolución) y densidad de píxeles (Densidad). Por ejemplo, la densidad de píxeles del dispositivo Nexus 5 es xxhdpi, lo que significa que la aplicación usa los íconos en la xxhdpicarpeta de la mipmapcarpeta.

Asimismo, la aplicación utiliza diseños y elementos de diseño de carpetas definidas para esa densidad.



Después de hacer clic en Siguiente , aparece la pantalla Imagen del sistema para elegir la versión del sistema Android para el dispositivo. La pestaña Recomendado muestra los sistemas recomendados para el dispositivo. Hay más versiones disponibles en las pestañas Imágenes x86 y Otras imágenes . Si aparece un enlace de descarga junto a una versión de imagen del sistema, aún no está instalado. Haga clic en el enlace para iniciar la descarga y haga clic en Finalizar cuando haya terminado.

1.12.1.1 *Ejecutando la aplicación en el dispositivo virtual*

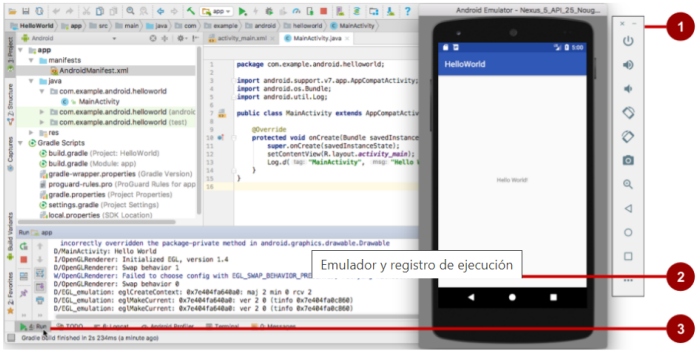
Para ejecutar la aplicación en el dispositivo virtual que creó en la sección anterior, siga estos pasos:

1. En Android Studio, seleccione Ejecutar> Ejecutar aplicación o haga clic en el Icono de ejecución de Android Studio icono Ejecutar en la barra de herramientas.

2. En la ventana Seleccionar destino de implementación, en Emuladores disponibles, seleccione el dispositivo virtual que creó y haga clic en Aceptar .

El emulador se inicia y arranca como un dispositivo físico. Dependiendo de la velocidad de su computadora, el proceso de inicio puede demorar un poco. La aplicación se compila y, una vez que el emulador está listo, Android Studio carga la aplicación en el emulador y la ejecuta.

Debería ver la aplicación creada a partir de la plantilla Actividad vacía ("Hola mundo") como se muestra en la siguiente figura, que también muestra el panel Ejecutar de Android Studio que muestra las acciones realizadas para ejecutar la aplicación en el emulador.



La figura anterior muestra el emulador y el registro de ejecución:

1. El emulador que ejecuta la aplicación.
2. El panel Ejecutar , que muestra las acciones realizadas para instalar y ejecutar la aplicación.
3. La pestaña Ejecutar , en la que hace clic para abrir o cerrar el panel Ejecutar .