|  |  |
| --- | --- |
|  | **UNIVERSIDAD DON BOSCO**  **FACULTAD DE INGENIERÍA – ESCUELA DE COMPUTACIÓN** |
|  | TALLER ANDROID. **TEMA I**  LinearLayout, RelativeLayout, ConstraintLayout. Tamaños de pantalla. |

* **I Objetivos**
* Conocer el IDE de desarrollo Android Studio.
* Utilizar los distintos Layouts para crear interfaces.
* **II Introducción**

¿Qué es Android?

Android es un sistema operativo basado en el kernel de Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tablets; y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró. Android fue presentado en 2007 junto la fundación del Open Handset Alliance (un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008. Los dispositivos de Android venden más que las ventas combinadas de Windows Phone e iOS.

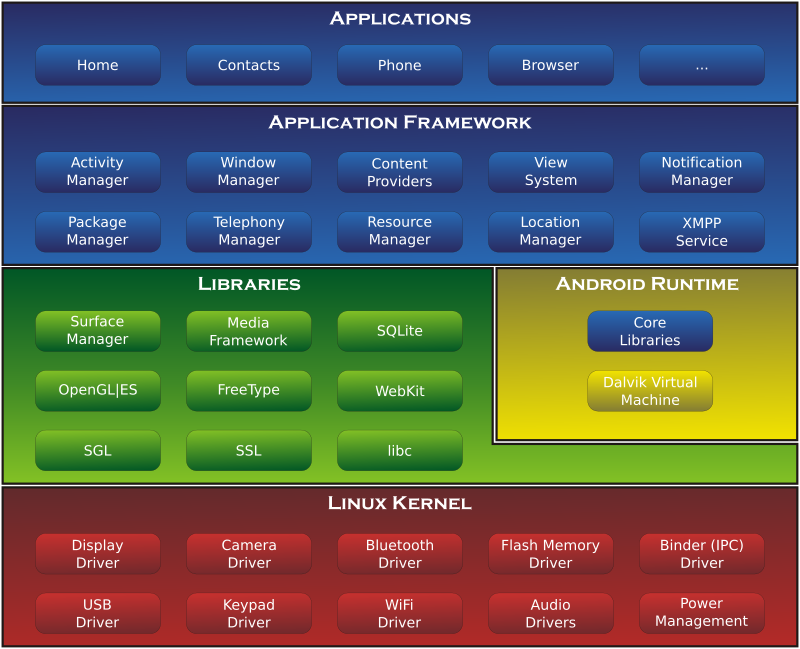
Características del sistema Android

1. El código de Android es abierto: Google liberó Android bajo licencia Apache. Cualquier persona puede realizar una aplicación para Android.

2. Hoy día hay más de 2 millones de aplicaciones para android (<http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/> ), aproximadamente 2/3 son gratis. Además la libertad de código permite adaptar Android a muchos otros dispositivos además de teléfonos celulares. Está implantado en Tablets, GPS, relojes, microondas… incluso hay por internet una versión de Android para PC.

3. El sistema Android es capaz de hacer funcionar a la vez varias aplicaciones y además se encarga de gestionarlas, dejarlas en modo suspensión si no se utilizan e incluso cerrarlas si llevan un periodo determinado de inactividad. De esta manera se evita un consumo excesivo de batería. Ésta es una de sus mayores ventajas por la rapidez con la que carga una aplicación abierta previamente.

Mapa de la estructura del Sistema Operativo Android:



¿Cómo se crean las aplicaciones para Android?

Para crear aplicaciones en Android, es necesario contar con un IDE de desarrollo. El IDE oficial para desarrollo de aplicaciones es Android Studio <http://developer.android.com/sdk/index.html>.

Las características de Android Studio son:

Sistema de construcción: Utiliza un robusto sistema de construcción llamado Gradle.

* Utiliza un DSL (Lenguaje Específico de Dominio) basado en Groovy (nada de XMLs complejos)
* Nos permite añadirlo a nuestros servidores de integración continua
* Construcción por convención (al estilo Maven)
* Compatible con dependencias Maven

Ayudas para la codificación:

La edición es más fluida, la refactorización más potente y un análisis de código que nos ayudará a mejorar considerablemente nuestro código.

Previsualización de recursos: elementos como imágenes, drawables, vistas, etc, son inmediatamente visibles en el editor.

Generación de recursos:

Android Studio incluye diálogos para facilitar la generación de recursos para distintas configuraciones. Uno de los mejores ejemplos es a la hora de crear un string. Podemos seleccionar el cualificador de región y se nos mostrará una lista de países que podemos ir seleccionando. Esto resulta tremendamente útil porque incorpora todos los cualifiadores con sus variantes así que no tendremos que recordarlos de memoria.

Detección de errores:

Este IDE incorpora de forma transparente una herramienta de análisis de código que nos ayuda a mejorar la calidad del mismo. Además se realiza de forma automática, de forma que conforme vas escribiendo te va indicando posibles fallos o mejoras que puedes corregir en el código. Métodos sin utilizar, lógica demasiado compleja o condiciones que nunca van a darse serán detectadas por el IDE y nos permitirá con un “par de clicks” corregir dichos problemas.

Refactorización:

La refactorización es mucho más potente en Android Studio. Ahora podemos cambiar el nombre de una imagen y automáticamente se cambiará el nombre del identificador. A la inversa funcionaría exactamente igual.

Ayudas para el diseño:

La previsualización de los layouts está bastante cuidada, marcando los elementos sobre los que estamos trabajando y permitiendo elegir entre distintos dispositivos, configuraciones, temas, etc.

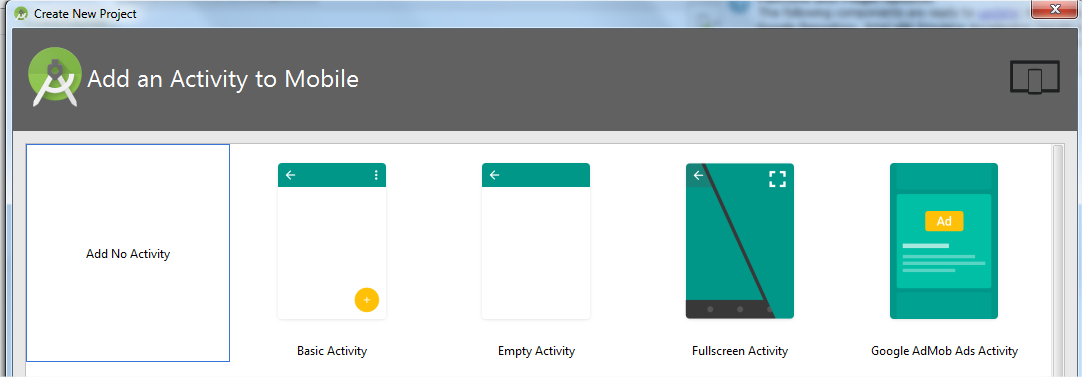
Acceso a los servicios de Google:

Google ha hecho especial hincapié en facilitar el uso de sus servicios en nuestras aplicaciones. Con Android Studio se desea facilitar el acceso a todas estas herramientas desde nuestro entorno de desarrollo. Han empezado integrando los servicios del GCM y han introducido un nuevo plugin (ADT Translation Manager Plugin) para la traducción de nuestras aplicaciones e integrado con el servicio de traducción de la Google Play Developer Console.

El principal sitio de soporte e información al programador es: <http://developer.android.com/index.html>

* **III Desarrollo**

Crearemos un nuevo proyecto. En la pantalla “Add and activity to Mobile” no agregaremos ninguna Actividad (Las crearemos desde 0).

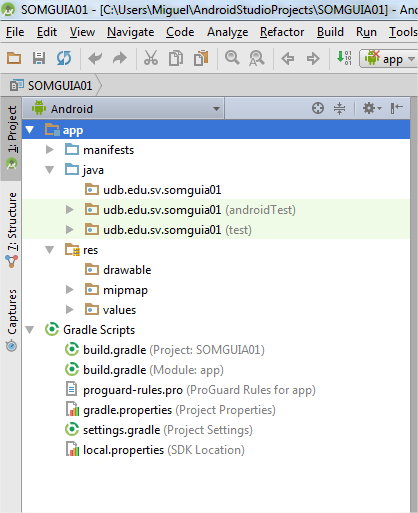


En nuestro panel izquierdo nos muestra la pestaña “Project” donde se incluye la estructura del proyecto y las carpetas que contienen:

Carpeta manifests con archivo de configuración o AndroidManifest.xml

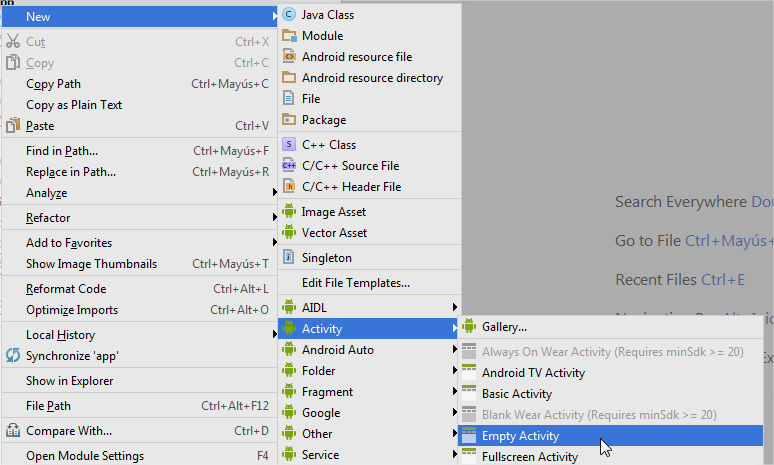
Carpeta de source con archivos de código fuente del proyecto (java)

Carpeta res con los recursos de aplicación

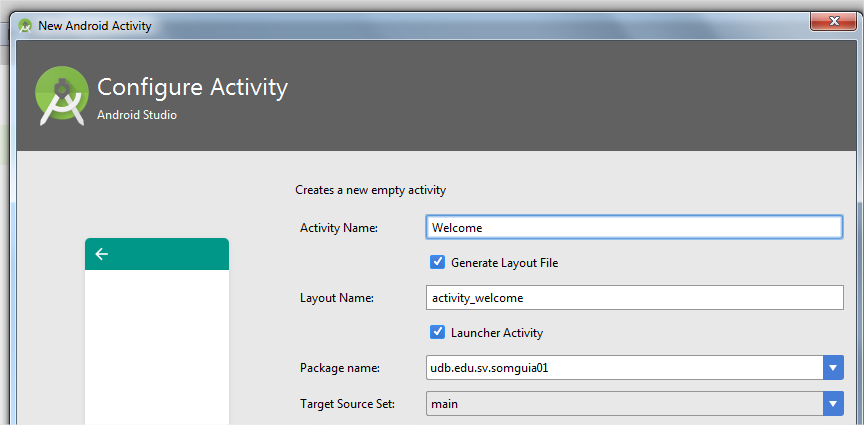


**LinearLayout**

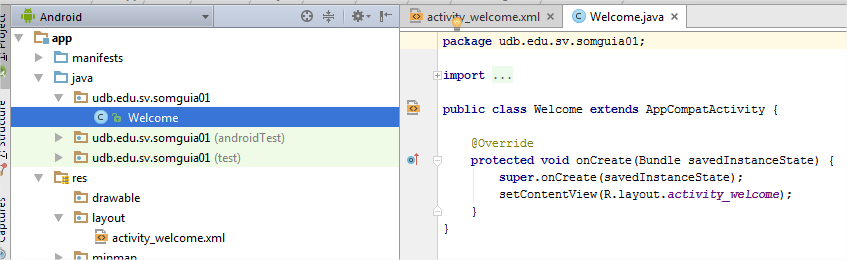
Crearemos una actividad nueva en blanco.



Le asignaremos el nombre de “Welcome”:

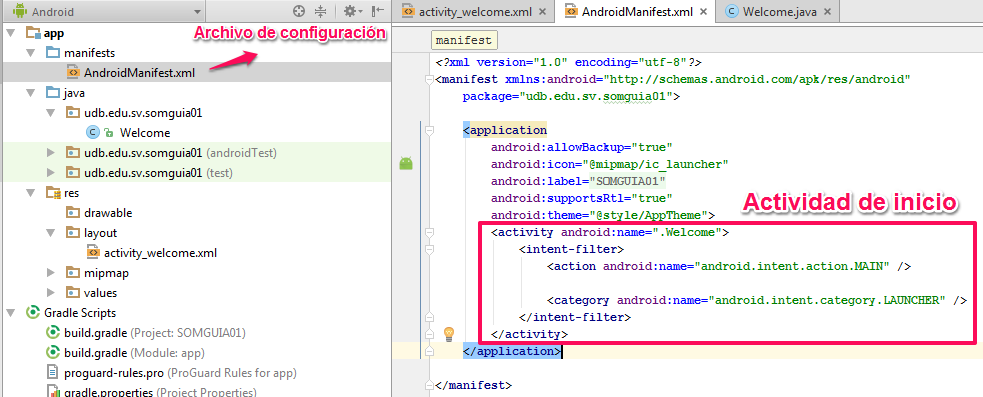


Se crearan dos archivos. Uno denominado Welcome.java y otro denominado activity\_welcome.java

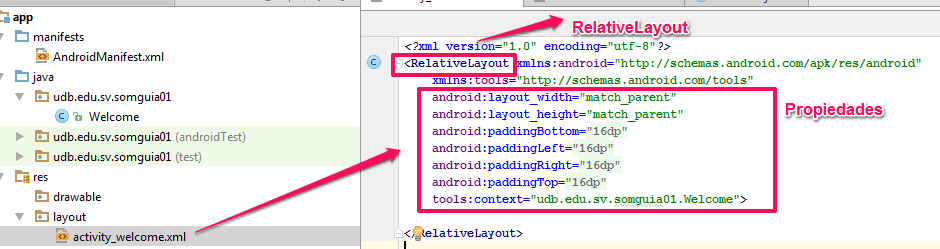


El archivo de configuración del proyecto AndroidManifest.xml contará ahora con una sección denominada “Activity”. Ésta hace referencia a las actividades o “Escenas” de la aplicación (Nótese la actividad creada llamada: Welcome).

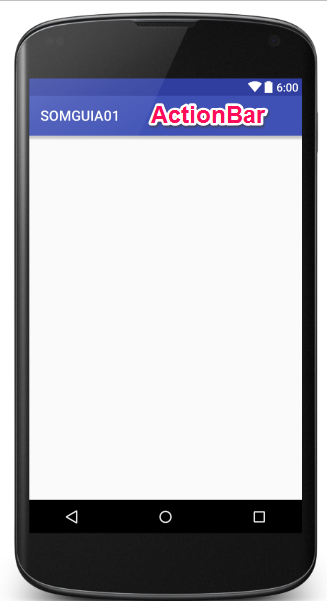
Por otra parte, cuenta con una propiedad denominada “intent-filter” que permite definir la “Actividad de inicio o arranque” para nuestra aplicación.



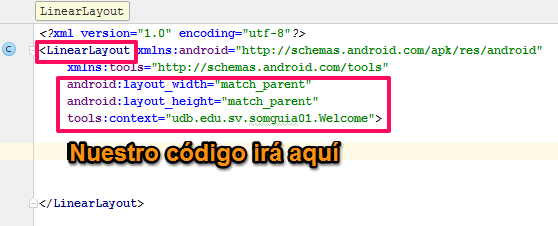
Por defecto la Actividad será “RelativeLayout” o “ConstraintLayout”. Posteriormente cambiaremos este por LinearLayout.



El diseño inicial contiene un “ActionBar” que se agrega por defecto. Sin embargo posteriormente será retirado.



Modificar el Layout para que sea “LinearLayout” y dejaremos en él las propiedades básicas:



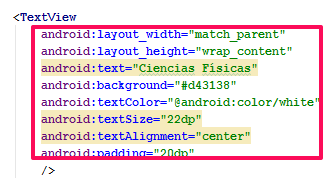
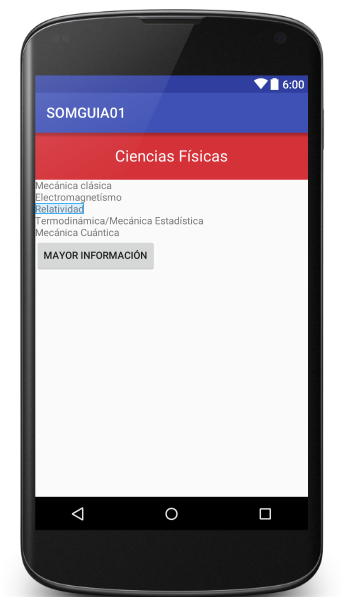
Agregaremos código para generar los TextViews básicos y un botón:

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context="udb.edu.sv.somguia01.Welcome"  android:orientation="vertical"** >   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Ciencias Físicas"** />   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Mecánica clásica"** />   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Electromagnetísmo"** />   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Relatividad"** />   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Termodinámica/Mecánica Estadística"** />   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Mecánica Cuántica"** />   <**Button  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="MAYOR INFORMACIÓN"** />   </**LinearLayout**> |

Resultado Esperado:



Agregaremos propiedades al primer TextView.

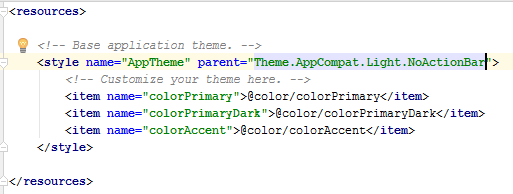


Como podemos ver, asignar una serie de propiedades al control contempla la asignación de varias líneas y atributos.

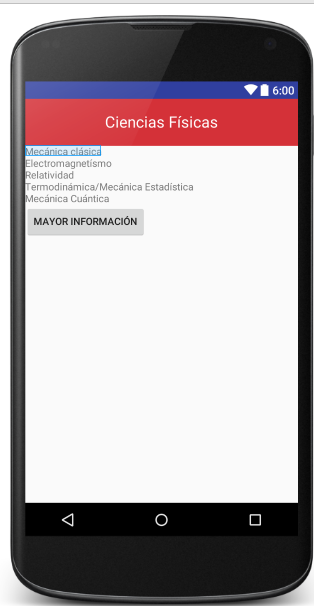
Ahora vamos a suprimir el ActionBar. Para ello modificamos values->styles.xml y modificaremos el AppTheme:



Asignaremos uno que no contenga el ActionBar:



Vista de la modificación:



A continuación crearemos un estilo para cada uno los demás elementos del LinearLayout:



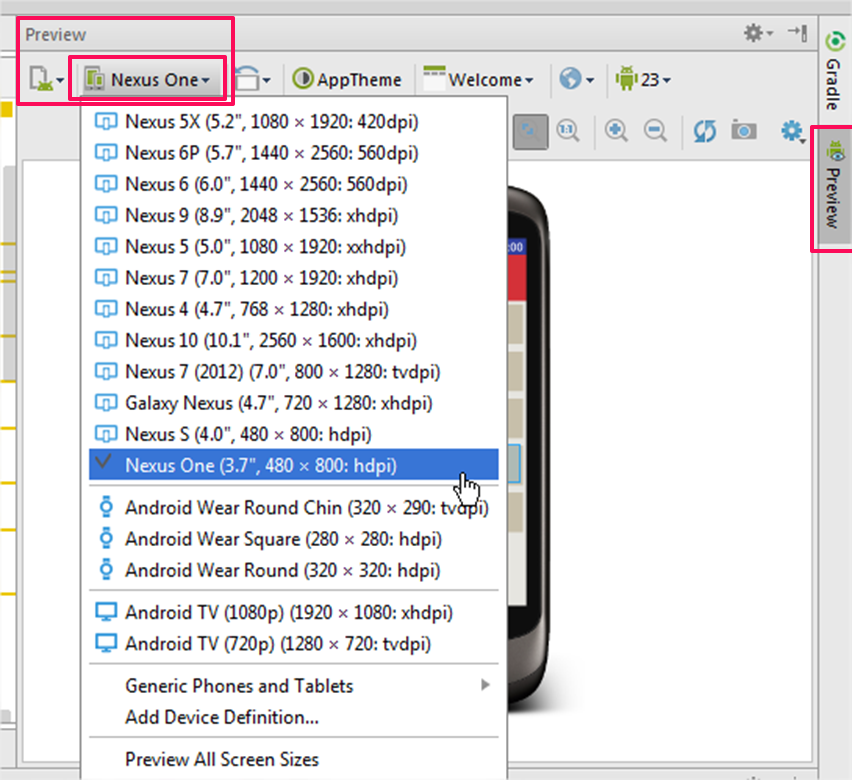
Las modificaciones del Layout son las siguientes:

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context="udb.edu.sv.somguia01.Welcome"  android:orientation="vertical"** >   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Ciencias Físicas"  android:background="#d43138"  android:textColor="@android:color/white"  android:textSize="22dp"  android:textAlignment="center"  android:padding="20dp"** />   <**TextView  android:text="Mecánica clásica"  style="@style/txtbrown"** />   <**TextView  android:text="Electromagnetísmo"  style="@style/txtbrown"** />   <**TextView  android:text="Relatividad"  style="@style/txtbrown"** />   <**TextView   android:text="Termodinámica/Mecánica Estadística"  style="@style/txtbrown"** />   <**TextView  android:text="Mecánica Cuántica"  style="@style/txtbrown"** />   <**Button  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="MAYOR INFORMACIÓN"  android:background="#636363"  android:textColor="@android:color/white"  android:layout\_gravity="center"  android:padding="5dp"  android:layout\_margin="1dp"** /> </**LinearLayout**> |

Resultado esperado:



Puede ver en los preview de diferentes dispositivos:



A continuación será necesario ejecutar la aplicación en el emulador AVD.

Consulte a su docente cómo puede crear un emulador y cómo ejecutar su aplicación en él.

Imagen ilustrativa:

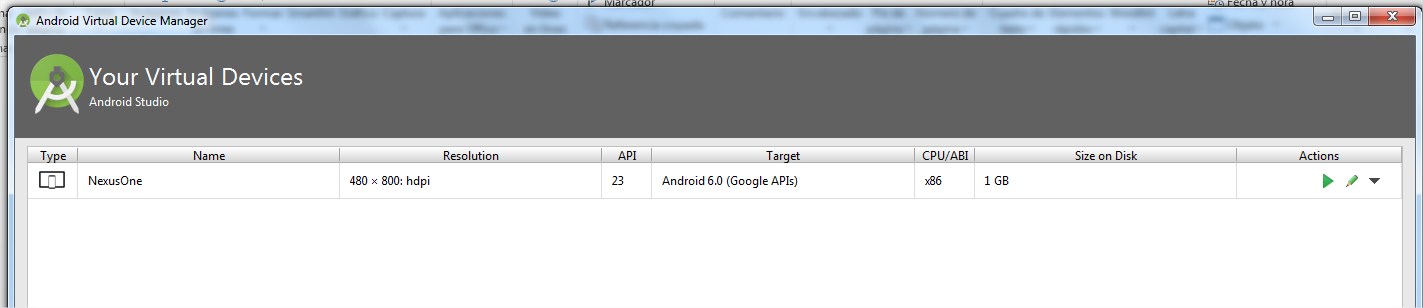
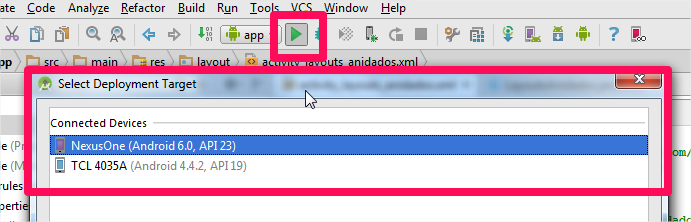
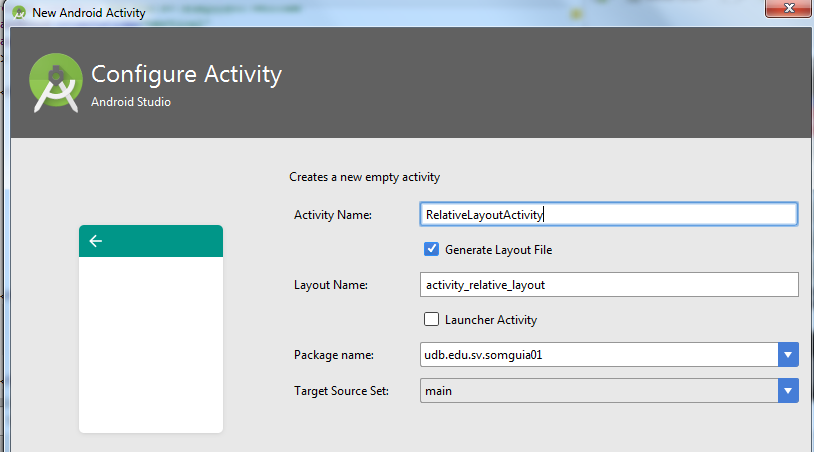


Imagen ilustrativa:

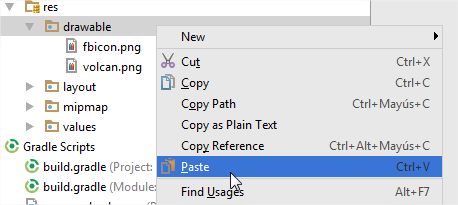
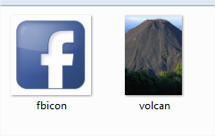


**RelativeLayout**

Crear una nueva Actividad denominada RelativeLayoutActivity



Es necesario incluir dos imágenes que se encontrarán disponibles como “Drawables” en la carpeta res/drawables. Seleccionamos desde nuestro navegador de archivos y los incluimos en dicha carpeta:



Agregaremos el siguiente código xml en el LayoutCreado:

|  |
| --- |
| *<?***xml version=”1.0” encoding=”utf-8”***?>* <**RelativeLayout xmlns:android=”http://schemas.android.com/apk/res/android”  xmlns:tools=”http://schemas.android.com/tools”  android:layout\_width=”match\_parent”  android:layout\_height=”match\_parent”  android:paddingBottom=”@dimen/activity\_vertical\_margin”  android:paddingLeft=”@dimen/activity\_horizontal\_margin”  android:paddingRight=”@dimen/activity\_horizontal\_margin”  android:paddingTop=”@dimen/activity\_vertical\_margin”  tools:context=”udb.edu.sv.somguia01.RelativeLayoutActivity”**> <**TextView  android:layout\_width=”wrap\_content”  android:layout\_height=”wrap\_content”  android:text=”GUIA TURISMO”  android:layout\_alignParentTop=”true”  android:layout\_alignParentLeft=”true”  android:id=”@+id/texto1”** />   <**TextView  android:layout\_width=”wrap\_content”  android:layout\_height=”wrap\_content”  android:text=”EL SALVADOR”  android:layout\_alignParentTop=”true”  android:layout\_alignParentRight=”true”  android:gravity=”right”  android:id=”@+id/texto2”** />   <**EditText  android:layout\_width=”match\_parent”  android:layout\_height=”wrap\_content”  android:id=”@+id/etlugar”  android:layout\_below=”@+id/texto1”  android:maxLines=”20”  android:minLines=”8”  android:lines=”10”  android:scrollbars=”vertical”  android:inputType=”textMultiLine”  android:gravity=”top|left”  android:hint=”Menciona tu lugar preferido”** />   <**Button  android:layout\_width=”wrap\_content”  android:layout\_height=”wrap\_content”  android:id=”@+id/btnadd”  android:layout\_below=”@+id/etlugar”  android:text=”Agregar”** />   <**Button  android:layout\_width=”wrap\_content”  android:layout\_height=”wrap\_content”  android:paddingLeft=”60dp”  android:paddingRight=”60dp”  android:id=”@+id/btnfind”  android:layout\_toRightOf=”@+id/btnadd”  android:layout\_below=”@+id/etlugar”  android:text=”Buscar”** />   <**Button  android:layout\_width=”100dp”  android:layout\_height=”100dp”  android:layout\_alignParentBottom=”true”  android:layout\_centerHorizontal=”true”  android:background=”@drawable/fbicon”  style=”?borderlessButtonStyle”** />  </**RelativeLayout**> |

El resultado esperado:



Sin embargo la apariencia gráfica es poco intuitiva. Modificaremos el diseño para mejorar la estética (Note que únicamente se agregan propiedades a los controles agregados.):

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:paddingBottom="@dimen/activity\_vertical\_margin"  android:paddingLeft="@dimen/activity\_horizontal\_margin"  android:paddingRight="@dimen/activity\_horizontal\_margin"  android:paddingTop="@dimen/activity\_vertical\_margin"    android:background="@drawable/volcan"** >  <**RelativeLayout android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:id="@+id/vista1"  android:background="#aa148b2f"  android:orientation="horizontal"** >  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="GUIA TURISMO"  android:layout\_alignParentTop="true"  android:layout\_alignParentLeft="true"  android:id="@+id/texto1"  android:textColor="@android:color/white"  android:textStyle="bold"  android:padding="10dp"** />   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="EL SALVADOR"  android:layout\_alignParentTop="true"  android:layout\_alignParentRight="true"  android:gravity="right"  android:id="@+id/texto2"  android:textColor="@android:color/white"  android:textStyle="bold"  android:padding="10dp"** />   </**RelativeLayout**>   <**EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/etlugar"  android:layout\_below="@+id/vista1"  android:background="@android:color/white"  android:maxLines="20"  android:minLines="8"  android:lines="10"  android:scrollbars="vertical"  android:inputType="textMultiLine"  android:gravity="top|left"  android:hint="Menciona tu lugar preferido"  android:layout\_margin="20dp"** />   <**Button  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:id="@+id/btnadd"  android:layout\_below="@+id/etlugar"  android:text="+"  android:textSize="30dp"  android:background="#aaffffff"  android:textColor="@android:color/white"** />   <**Button  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:id="@+id/btnfind"  android:layout\_toRightOf="@+id/btnadd"  android:layout\_below="@+id/etlugar"  android:text="Buscar"  android:textColor="@android:color/white"  android:background="#9bd065"  android:paddingLeft="60dp"  android:paddingRight="60dp"** />   <**Button  android:layout\_width="100dp"  android:layout\_height="100dp"  android:layout\_below="@+id/btnfind"  android:layout\_marginTop="50dp"  android:layout\_centerHorizontal="true"  android:background="@drawable/fbicon"  style="?borderlessButtonStyle"** />  </**RelativeLayout**> |

Resultado esperado:



Necesitaremos cambiar la actividad principal para que sea “RelativeLayoutActivity” la escena de inicio. Realizaremos definiendo la propiedad el intent-filter del archivo AndroidManifest.xml:



|  |
| --- |
| <**application  android:allowBackup="true"  android:icon="@mipmap/ic\_launcher"  android:label="@string/app\_name"  android:supportsRtl="true"  android:theme="@style/AppTheme"**>  <**activity android:name=".Welcome"**>   </**activity**>  <**activity android:name=".RelativeLayoutActivity"**>  <**intent-filter**>  <**action android:name="android.intent.action.MAIN"** />   <**category android:name="android.intent.category.LAUNCHER"** />  </**intent-filter**>  </**activity**> </**application**> |

* **IV Ejercicios**

|  |
| --- |
| Proyecto: **Sistema de cotizaciones móvil.** |
| Localmente vía GreenDAO, almacenará su base de datos de clientes. Debe guardar nombre, dirección, teléfono, email.  Listará categorías de productos, al seleccionarlo desplegará los productos con sus precios. Esta información será accedida vía webservice.  El usuario vía GreenDAO podrá guardar la información de la cotización. Deberá guardar el nombre de usuario de aplicación, el correo(destino), los IDs de los item en la cotización individualmente, cantidad, precio sugerido (webservice) precio ofertado (manual) y el total.  Finalmente, la información será enviada en formato json vía petición POST hacia el webservice, el cual enviará un correo detallado hacia el cliente.  La aplicación pedirá usuario y password (Esto estará habilitado al final del taller). |

**Ejercicio 1 para proyecto:**

* Investigue los Android UI Guidelines.
* Crear una SplashScreen para su aplicación.
* Crear formulario de Clientes. (Aún no se almacenarán los datos).
* Investigue qué es CardView y utilícelo en su formulario.