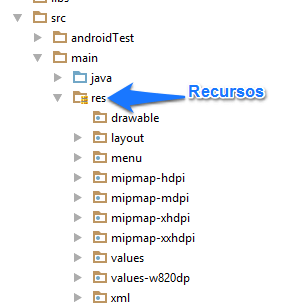
|  |  |
| --- | --- |
|  | **UNIVERSIDAD DON BOSCO**  **FACULTAD DE INGENIERÍA – ESCUELA DE COMPUTACIÓN** |
|  | TALLER ANDROID  **TEMA II - Recursos** |

* **I Objetivos**
* Que el alumno comprenda el uso de los recursos en android.
* A partir del uso de recursos, pueda crear estilos, internacionalización, imágenes, layouts para orientación de pantalla, entre otros.
* **II Introducción**

**¿Qué Son Los Recursos En Android?**

Los recursos son archivos o datos externos que soportan el comportamiento de nuestra aplicación Android. Estos se componen de **imágenes**, **strings**, **colores**, **estilos**, etc.

Formalmente en un proyecto, estos elementos se encuentran en la carpeta **src/main/res**. Allí encontrarás subdirectorios que agrupan los diferentes tipos de recursos.



La idea del uso de recursos es dividir el código de la app para mantener independencia. Todo con el fin de agregar variaciones de los archivos para adaptar la aplicación a diferentes tipos de pantallas, idiomas, versiones,  dimensiones, configuraciones de orientación, etc.

*Por ejemplo…*

No es lo mismo el espacio de un teléfono móvil a una tableta. Sin embargo se pueden crear variaciones de los recursos para que la aplicación se adapte a la densidad de pantalla de cada uno.

**Tipos De Recursos**

Los grupos de recursos se dividen en subdirectorios. Cada uno de ellos contiene archivos que cumplen una función específica dentro de la aplicación. Se debe respetar esta estructura de documentos para no tener problemas en la ejecución.

La siguiente es una [tabla de la mayoría de los tipos de recursos](http://developer.android.com/intl/es/guide/topics/resources/providing-resources.html) que podemos crear:

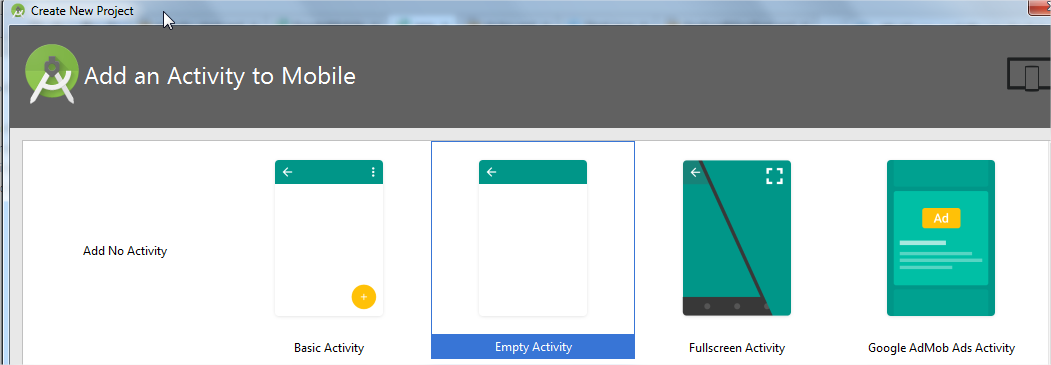
|  |  |
| --- | --- |
| **Nombre del subdirectorio** | **Contenido** |
| animator/ | Alberga archivos XML con animaciones de propiedades para objetos. |
| anim/ | Contiene archivos XML que representan animaciones especiales para views. |
| color/ | Aquí encontrarás archivos XML con definiciones de listas de estados de color (**Color State List**). Estas listas determinan el color de un componente dependiendo del estado en que se encuentre. |
| drawable/ | Recursos gráficos que puedan ser proyectados en pantalla. Generalmente encontrarás archivos de imagen como **.png**, **.jpg** o **.gif**, sin embargo es posible usar otros como: archivos nine-patch, listas de estado, drawables con múltiples niveles, drawables con figuras 2D definidas en XML, y muchas más. |
| mipmap/ | Contiene el icono de la aplicación para evitar distorsión entre varias densidades de pantalla. |
| layout/ | Archivos XML que contienen definiciones de la interfaz de usuario. |
| menu/ | Archivos XML que establecen las características para los menús usados en la interfaz. Normalmente contienen definiciones sobre los ítems albergados en un menú y las agrupaciones entre ellos. |
| raw/ | Almacena aquí todos aquellos archivos que deseas leer directamente como un flujo de caracteres estándar (InputStream). |
| values/ | Archivos XML que contienen datos simples como enteros, strings, booleanos, colores. Puedes agrupar estos elementos en las siguientes convenciones de archivos:   * **strings.xml** para cadenas * **colors.xml** para definiciones de colores en hexadecimal. * **dimens.xml** para dimensiones de márgenes, padding, tamaños, etc. * **styles.xml** para los estilos de interfaz que tendrá la aplicación. * **arrays.xml** para arreglos de elementos. |
| xml/ | Contiene archivos XML con definiciones especiales requeridas en acciones con el framework de Android. O para usos que requieras en tus requerimientos, como [parsear una estructura XML](http://www.hermosaprogramacion.com/2015/07/tutorial-parsing-xml-con-xmlpullparser-android/) directamente en la aplicación. |

Posteriormente veremos el uso de recursos para diferentes dimensiones de pantalla.

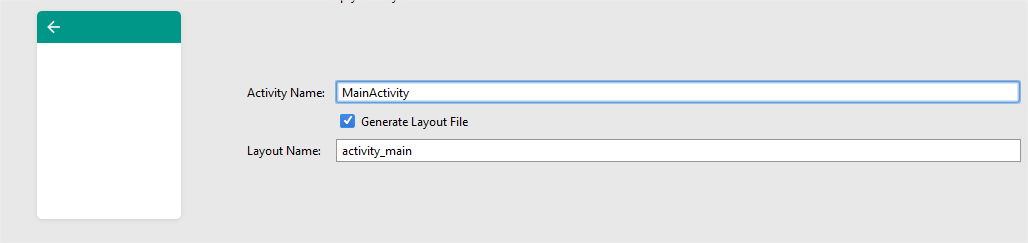
* **III Desarrollo**

**Crear un nuevo proyecto.**

En ésta ocasión agregaremos una actividad en blanco:



Dejaremos el nombre de la actividad como MainActivity. Nótese que se crea de manera conjunta un Layout xml denominado activity\_main



**Internacionalización**

Procederemos a modificar un fichero de strings (strings.xml). Éste contendrá los textos de la aplicación. En la práctica es más eficiente mantener el fichero de texto, esto nos permitirá crear ficheros que se utilizan dependiendo del idioma del dispositivo. Por defecto, si un idioma no se encuentra, se utiliza el fichero strings.xml y también es recomendable que éste se encuentre en inglés.

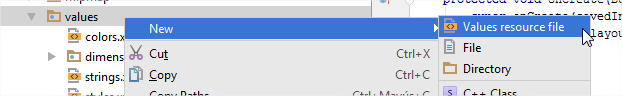
strings.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**resources**>  <**string name="app\_name"**>SOMGUIA02</**string**>  <**string name="tittle"**>Vegetarian?</**string**>  <**string name="vegetarian\_food"**>  A vegetarian is someone who doesn\'t eat meat,  and mostly eats foods that come from plants,  like grains, fruits, vegetables, and nuts.  Some stricter vegetarians avoid more than just meat.  They also avoid animal products,  which are nonmeat foods that come from animals.  Some examples would be milk (from cows) and eggs (from chickens).  </**string**>   <**string name="ordernow"**>  Local favorites,\n delivered faster\n . Order Now!!!  </**string**> </**resources**> |

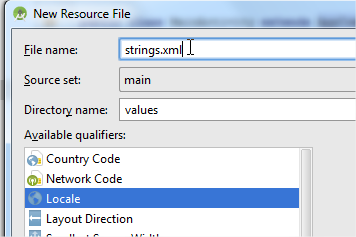
Ahora agregaremos un nuevo recurso strings.xml para soportar el idioma español.

Haremos lo siguiente.

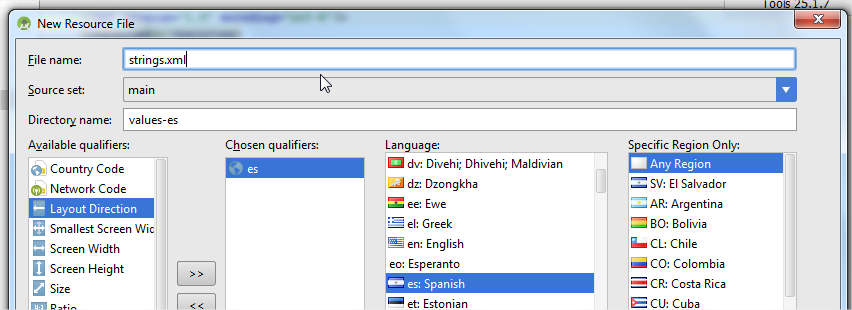
Clic derecho en la carpeta values y seleccionar:



Ahora elegiremos un Nuevo locale:



En la siguiente figura se muestra las opciones que debe ir seleccionando:



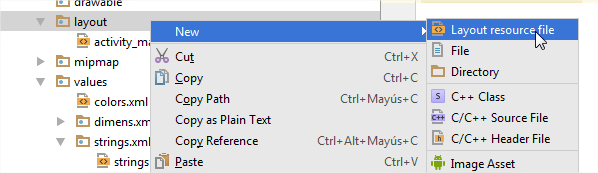
Proceda a editar el nuevo fichero (notará que se están juntos, con la diferencia que uno tiene la palabra (es))

strings.xml (es)

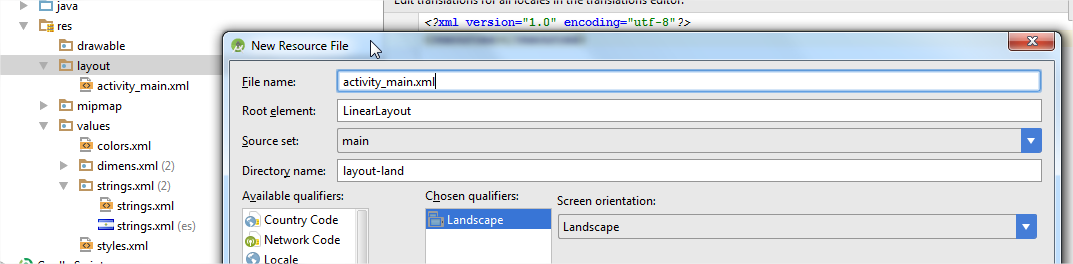
|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**resources**>  <**string name="app\_name"**>SOMGUIA02</**string**>  <**string name="tittle"**>¿Vegetariano?</**string**>  <**string name="vegetarian\_food"**>  Un vegetariano es alguien que no come carne,  y en su mayoría come alimentos que provienen de las plantas,  como granos, frutas, verduras y frutos secos.  Algunos vegetarianos más estrictos evitan algo más que carne:  También evitan productos de origen animal,  los cuales son los alimentos no cárnicos que provienen de animales.  Algunos ejemplos serían la leche (de vaca) y los huevos (de gallina).  </**string**>   <**string name="ordernow"**>  Tus tiendas de conveniencia, \n entrega rápida\n ¡Ordena ahora!  </**string**>  </**resources**> |

**Definiendo Layout Land para orientación de pantalla.**

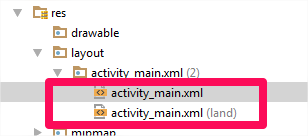
A continuación crearemos un Layout que será el que se muestre cuando cambiemos la orientación de la pantalla (horizontal). Crearemos un nuevo Layout Resource. Las siguientes imágenes muestran el procedimiento:



Valores que tendrá el nuevo recurso:



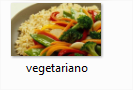
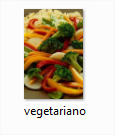
El resultado será el siguiente.



Tome en cuenta que posteriormente en esta guía modificaremos los layouts para que tengan diseños diferentes.

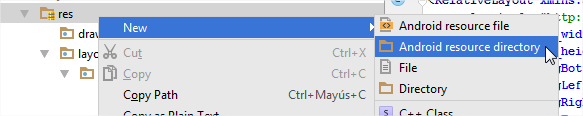
**Creando una carpeta drawable para los recursos que utilizarán otra orientación de pantalla.**

A continuación agregaremos dos imágenes para que sean utilizados según la orientación de la pantalla. Las imágenes serán los fondos de pantalla de la aplicación

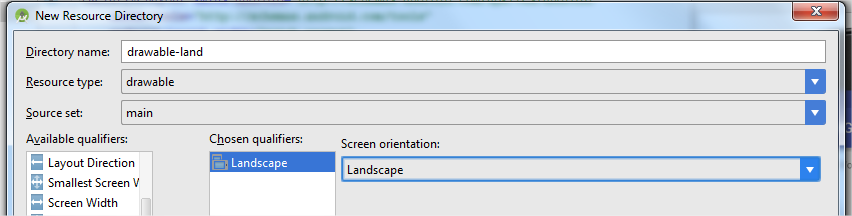


Crearemos una carpeta para los recursos drawable-land con los siguientes pasos:

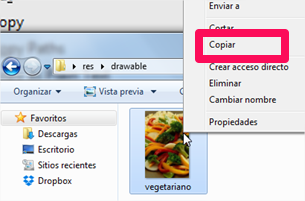
Clic derecho en la carpeta res:



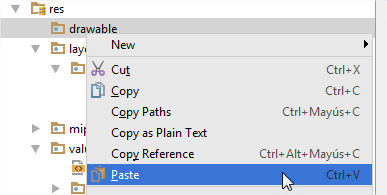
Seleccionar folder y elegir Landscape



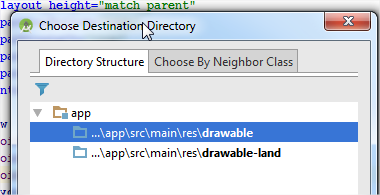
Posteriormente copiaremos la imagen y la agregaremos a la carpeta drawable



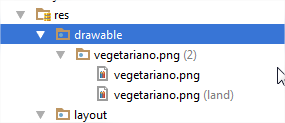
Ahora la pegaremos en nuestra carpeta drawable



Y seleccionaremos la carpeta drawable:



Ahora usted debe agregar la imagen land para drawable-land. El resultado será el siguiente:



Desde la vista previa invierta la posición del dispositivo para observar los resultados:



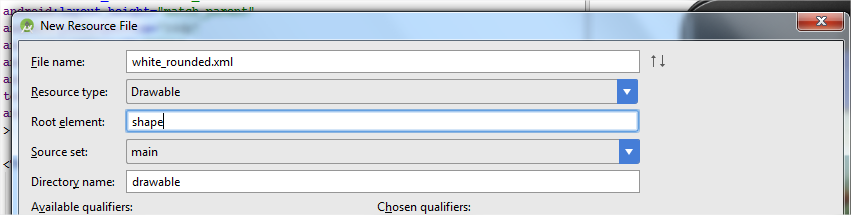
**Agregando un recurso “shape”**

Un shape es un archivo XML que define una forma geométrica, incluyendo colores y degradados.

Dando clic derecho en res seleccionaremos:



Y agregaremos los siguientes valores:



|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**shape xmlns:android="http://schemas.android.com/apk/res/android"**>  <**stroke  android:width="1dp"  android:color="#000000"** />  <**solid android:color="#d1ffffff"** />  <**padding  android:left="5dp"  android:right="5dp"  android:top="5dp"** />  <**corners android:radius="10dp"** /> </**shape**> |

Le corresponde a usted agregar un recurso xml denominado green\_rounded.xml (siga el procedimiento anterior)

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**shape xmlns:android="http://schemas.android.com/apk/res/android"**>  <**stroke  android:width="1dp"  android:color="#000000"** />  <**solid android:color="#5cbb5f"** />  <**padding  android:left="5dp"  android:right="5dp"  android:top="5dp"** />  <**corners android:radius="5dp"** /> </**shape**> |

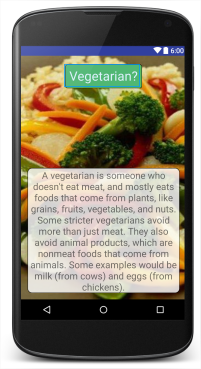
**Modificando los estilos**

Un recurso de estilo define el formato y apariencia de una interfaz de usuario. Un estilo se puede aplicar a un View individual (como un control) o a la totalidad de una actividad o aplicación (se puede definir en el AndroidManifest).

|  |
| --- |
| <**resources**>   *<!-- Base application theme. -->* <**style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar"**>  *<!--QUITAMOS EL ACTIONBAR-->  <!-- Customize your theme here. -->* <**item name="colorPrimary"**>@color/colorPrimary</**item**>  <**item name="colorPrimaryDark"**>@color/colorPrimaryDark</**item**>  <**item name="colorAccent"**>@color/colorAccent</**item**>  </**style**>   *<!-- Padding de layouts -->* <**style name="layoutandpadding"**>  <**item name="android:padding"**>15dp</**item**>  </**style**>   *<!-- Titulos centrados -->* <**style name="centerTitle"**>  <**item name="android:padding"**>10dp</**item**>*<!--Padding-->* <**item name="android:textSize"**>30sp</**item**>*<!--Texto-->* <**item name="android:layout\_centerHorizontal"**>true</**item**>*<!--Centrar en RelativeLayout-->* <**item name="android:layout\_gravity"**>center</**item**>*<!--Centrar en LinearLayout-->* <**item name="android:textColor"**>#ffffff</**item**>  <**item name="android:layout\_margin"**>5dp</**item**>  </**style**>   *<!-- Descripciones -->* <**style name="textDescription"**>  <**item name="android:textSize"**>22sp</**item**>  <**item name="android:gravity"**>center</**item**>  </**style**>   *<!-- Texto de ordenar ahora -->* <**style name="textOrderNow"**>  <**item name="android:gravity"**>right|bottom</**item**>  <**item name="android:textColor"**>#ffffff</**item**>  <**item name="android:textSize"**>25sp</**item**>  <**item name="android:textStyle"**>bold|italic</**item**>  </**style**>  </**resources**> |

**Modificando el fichero de Layout activity\_main.xml**

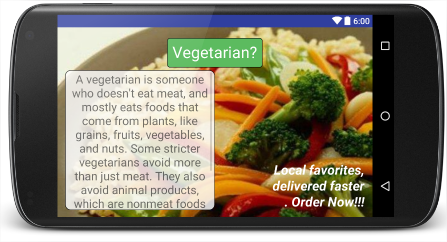
|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context="udb.edu.sv.somguia02.MainActivity"  android:background="@drawable/vegetariano"  style="@style/layoutandpadding"** >   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="@string/tittle"  android:background="@drawable/green\_rounded"  style="@style/centerTitle"** />   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="vertical"  android:layout\_alignParentBottom="true"** >   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:text="@string/vegetarian\_food"  style="@style/textDescription"  android:background="@drawable/white\_rounded"** />  </**LinearLayout**>  </**RelativeLayout**> |



**Modificando el activity\_main.xml (land)**

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical" android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="@drawable/vegetariano"  style="@style/layoutandpadding"** >  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="@string/tittle"  android:background="@drawable/green\_rounded"  style="@style/centerTitle"** />  <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"** >  <**ScrollView  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:background="@drawable/white\_rounded"** >   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"** >   <**TextView  android:layout\_width="0dp"  android:layout\_weight="1"  android:layout\_height="wrap\_content"  style="@style/textDescription"  android:text="@string/vegetarian\_food"** />   </**LinearLayout**>  </**ScrollView**>  <**TextView  android:layout\_width="0dp"  android:layout\_weight="1"  android:layout\_height="match\_parent"  android:text="@string/ordernow"  style="@style/textOrderNow"** /> </**LinearLayout**> </**LinearLayout**> |

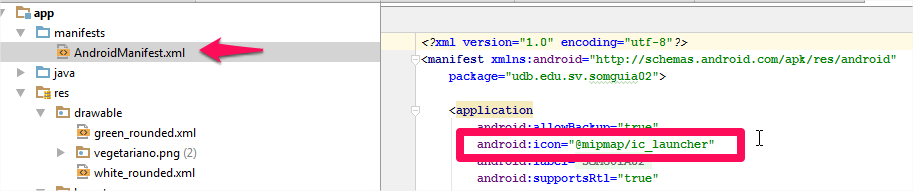
Resultado esperado:



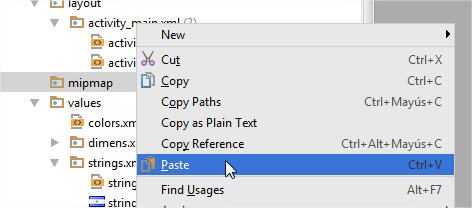
**Cambiar ícono de la aplicación:**

Busque un ícono para su aplicación con el nombre ic\_launcher.png. La aplicación por defecto tiene una serie de íconos ic\_launcher, por lo que tendrá que eliminarlos.

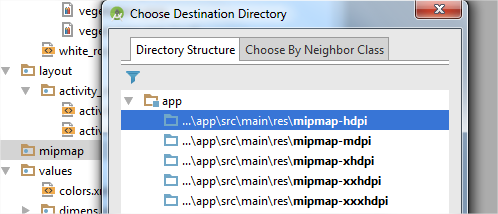
Por otra parte, si no desea eliminar los archivos, puede definir el ícono modificándolo desde el AndroidManifest



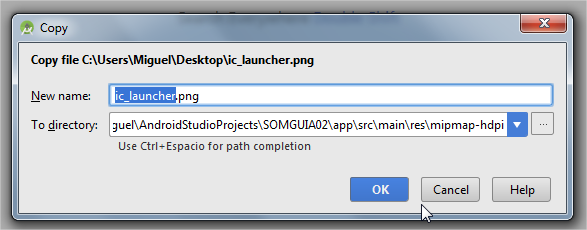
Para agregar el ícono al proyecto seleccione:

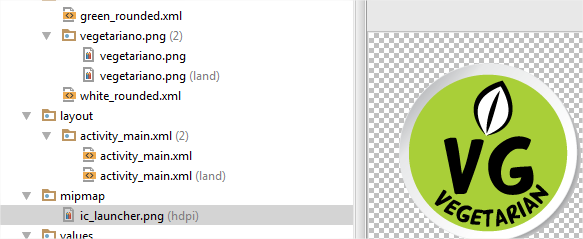


Seleccionar un mipmap

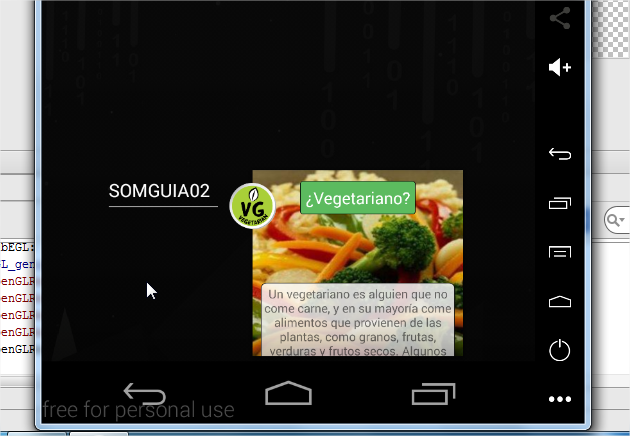


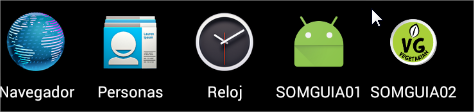
Y el nombre del recurso.



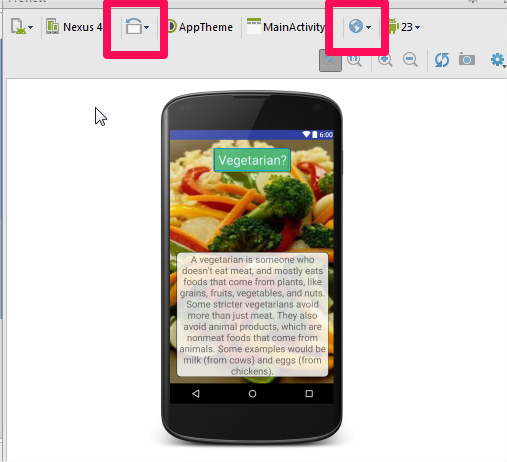


Resultado esperado:





Desde la vista previa, cambie la orientación y el idioma del dispositivo:

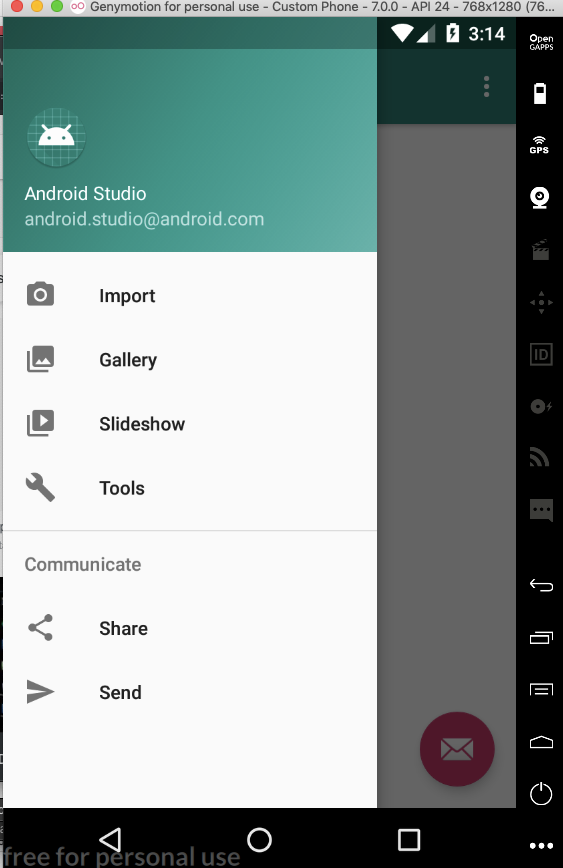


* **IV Ejercicios**

|  |
| --- |
| 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 2 para proyecto:**

1. Cree un menú de navegación de aplicación. Puede utilizar la galería de actividades y utilizar Navigation Drawer Activity como base. Adecue el contenido y los íconos a los elementos que usted necesite.



1. Agregue una actividad llamada ListaClientesActivity. Únicamente debe contener estilo y diseño. Posteriormente se llenará la información.
2. Agregue una nueva actividad llamada ListaProductosActivity. Únicamente debe contener estilo y diseño. Posteriormente se llenará la información.
3. Agregue una actividad llamada CotizacionesActivity. Únicamente debe contener estilo y diseño. Posteriormente se llenará la información.
4. Internacionalice su formulario de clientes. Agregue los textos de información en español e inglés, evitando utilizar strings HardCoded en el xml.
5. Internacionalice el menú que vaya a generar.