	UNIVERSIDAD DON BOSCO FACULTAD DE INGENIERÍA ESCUELA DE COMPUTACIÓN	
CICLO: I/2021	GUIA DE LABORATORIO #05	
	Nombre de la Práctica:	Guía #05- Recursos
	MATERIA:	Desarrollo de Software para Móviles

I. OBJETIVOS

1. Conocer que son los recursos en Android.
2. Conocer los tipos de recursos en Android.
3. Aprender a incorporar recursos alternativos en un proyecto de Android.
4. Brindar soporte a diferentes dispositivos auxiliándonos de recursos.

II. INTRODUCCION TEORICA

Recursos

Los recursos son los archivos adicionales y el contenido estático que usa tu código, como mapas de bits, definiciones de diseño, strings de interfaz de usuario, instrucciones de animación, etc.

Siempre debes externalizar los recursos para aplicaciones, como imágenes y strings de tu código, para que puedas mantenerlos de forma independiente.

También debes proporcionar recursos alternativos para configuraciones de dispositivos específicos, agrupándolos en directorios de recursos con un nombre especial.

En tiempo de ejecución, Android utiliza el recurso adecuado según la configuración actual. Por ejemplo, puedes proporcionar un diseño de interfaz de usuario (IU) diferente según el tamaño de la pantalla o strings diferentes según la configuración de idioma.

Una vez que externalizas los recursos para tu aplicación, puedes acceder a ellos mediante los ID de recursos que se generan en la clase `R` de tu proyecto.

En este documento, se muestra cómo puedes agrupar los recursos en tu proyecto de Android, proporcionar recursos alternativos para configuraciones de dispositivos específicos y acceder a ellos desde el código de tu aplicación u otros archivos XML posteriormente.

Tipos de recursos.

Android posee 2 grandes tipos de recursos:

1. **Recursos del proyecto.**
Son recursos que son exclusivos del proyecto donde residen (recursos alternativos)
2. **Recursos del sistema.**
Son parte del sistema operativo y mantienen compatibilidad con las diferentes versiones del mismo.

Recursos del proyecto.

Debes colocar cada tipo de recurso en un subdirectorio específico del directorio `res/` de tu proyecto. Por ejemplo, esta es la jerarquía de archivos de un proyecto simple:

```
MyProject/
  src/
    MainActivity.java
  res/
    drawable/
      graphic.png
    layout/
      main.xml
      info.xml
    mipmap/
      icon.png
    values/
      strings.xml
```

El directorio `res/` contiene todos los recursos (en subdirectorios): un recurso de imagen, dos recursos de diseño, directorios `mipmap/` para los íconos del selector y un archivo de recursos de strings.

Los nombres del directorio de recursos son importantes y se describen la siguiente tabla.

Carpeta (identificador con clase R)	Descripción breve
res/drawable R.drawable	Archivos de mapas de bits (.png, .9.png, .jpg, .gif) o archivos XML que se han compilado en los siguientes subtipos de recursos de elemento de diseño: Archivos de mapas de bits, Nine patches (mapas de bits reajustables), Listas de estados, Formas, Elementos de diseño de animaciones, Otros elementos de diseño
res/layout R.layout	Archivos XML que contienen layouts.
res/menu R.menu	Archivos XML con descriptores de menús que podemos usar en nuestra aplicación.
res/anim R.anim	Archivos XML que permite definir una animación con interpolación de movimientos.
res/animator R.animator	Archivos XML que permiten modificar las propiedades de objetos a lo largo del tiempo.
res/xml R.xml	Otros archivos XML como archivos de preferencias.

res/raw R.raw	Archivos de Audio, video, texto, etc.
-------------------------	---------------------------------------

Archivos (identificador con clase R)	Descripción breve
res/values	Archivos XML que definen un determinado valor para referencias un color, estilo, cadena.

Tipos de recursos en la carpeta values:

Archivos (identificador)	Descripción breve
strings.xml R.string	Identifica cadenas de caracteres <code><string name="hola">Hola mundo</string></code>
colors.xml R.color	Colores definidos en formato ARGV Los valores se indican en hexadecimal en los formatos #RGB, #ARGB, #RRGGBB ó #AARRGGBB <code><color name="verde_opaco">#0f0</color></code> <code><color name="Rojo_translucido">#80ff0000</color></code>
dimensions.xml R.dimen	Un número seguido de una unidad de medida px – pixeles, mm – milímetros, in – pulgadas, pt – puntos dp – pixeles independientes de la densidad. <code><dimen name="alto">2.2mm</dimen></code> <code><dimen name="tamaño_fuente">16dp</dimen></code>

Archivos (identificador)	Descripción breve
styles.xml R.styles	Serie de atributos que pueden ser aplicados a una vista o actividad. Si se aplican a una actividad se conocen como temas. <code><style name="TextoGrande" parent="@style/Text"></code> <code><item name="android:textSize">20pt</item></code> <code><item name="android:textColor">#000080</item></code> <code></style></code>

R.int	Define un valor entero <code><integer name= "edad">20</integer></code>
R.bool	Define un valor booleano <code><bool name= "continuar">true</bool></code>
R.id	Define un identificador único de recurso. <code><item type="id" name="boton_ok" /></code> <code><item type="id" name="txtNombre" /></code>

Archivos (identificador)	Descripción breve
R.array	Una serie ordenada de elementos. Pueden ser strings, enteros o de recursos (TypedArray) <code><string-array name= "dias_semana"></code> <code> <item>lunes</item></code> <code> <item>martes</item></code> <code></string-array></code> <code><integer-array name= "meses"></code> <code> <item>1</item> <item>2</item><item>3</item></code> <code></integer-array></code> <code><array name= "fotos"></code> <code> <item>@drawable/foto1</item></code> <code> <item>@drawable/foto2</item></code> <code></array></code>

III. PROCEDIMIENTO

Incorporando recursos alternos al proyecto.

A continuación, vamos a incorporar recursos a nuestro proyecto para brindar soporte a una aplicación que permita mostrar 2 idiomas en sus opciones: español e inglés.

Esto dependerá de la configuración que el usuario tenga en su dispositivo móvil a través de las opciones **Administración general->Idioma y entrada->Idioma**.

1. Ejecuta Android Studio y crea un nuevo proyecto. En Project Template selecciona Empty Activity y presiona **Next**.
2. Como nombre **Guia05App**, en Package name escribe **sv.edu.udb.guia05app** en **Save location** escoge la carpeta de tu preferencia, en Language elige Java y el Minimum SDK selecciona API 16: Android 4.1 (Jelly Bean) y Presiona **Finish**.
3. Haz **click derecho** sobre la carpeta **res** y selecciona **new->Android Resource File**
4. En la ventana que aparece a continuación establece los valores como aparece la siguiente imagen:

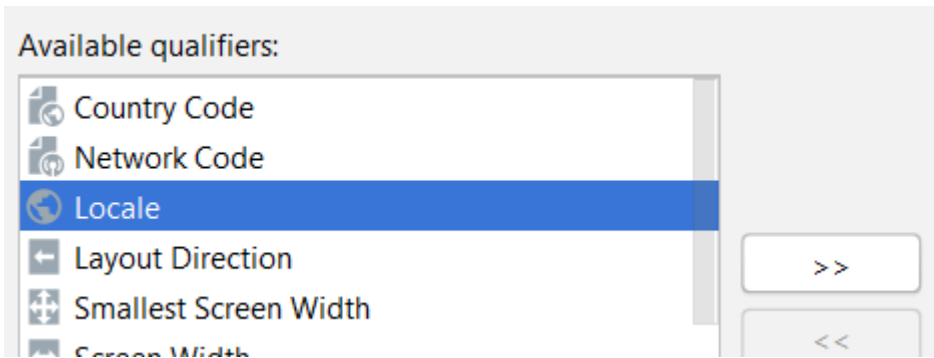


File name:	strings
Resource type:	Values
Root element:	resources
Source set:	main
Directory name:	values

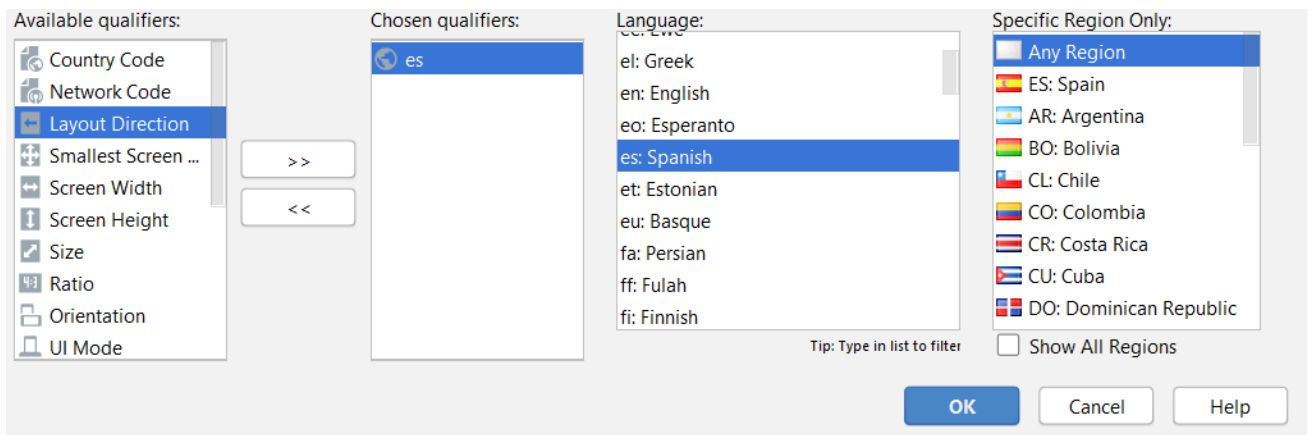
(Debes digitar **strings** en minúscula en File Name)

Asegúrate que quede seleccionado **Values** en la opción **Resource Type**

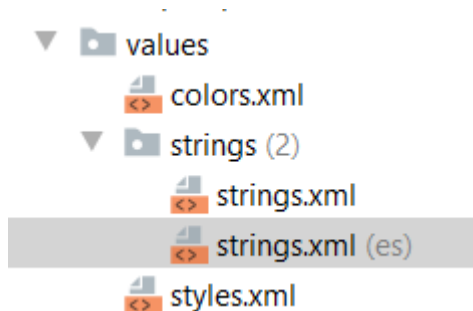
5. En la parte baja selecciona **Locale** haciendo doble clic sobre dicha opción.



6. Ahora en el grupo **Language** selecciona **es:Spanish** y luego haz clic en el botón **OK**.



Observa como ahora tienes 2 archivos **strings.xml** en **res/values/strings** (Pero uno de ellos tiene un sufijo **es**)



7. Edita el archivo **strings.xml** (el que no tiene sufijo) Este archivo representa el contenido en idioma inglés. (archivo por default)
para que tenga el siguiente contenido:

```
1 <resources>
2     <string name="app_name">Guia05App</string>
3     <string name="saludo1">Hello user</string>
4     <string name="saludo2">Welcome</string>
5 </resources>
```

8. Edita el archivo **strings.xml (es)**. Con el siguiente contenido:
(Este archivo representa el contenido en español.)

```
2 <resources>
3     <string name="app_name">Guia05App</string>
4     <string name="saludo1">Hola usuario</string>
5     <string name="saludo2">Bienvenido</string>
6 </resources>
```

Importante.

Observa como las constantes **saludo1** y **saludo2** tienen exactamente el mismo nombre en ambos archivos. Sin embargo, el contenido de dichas constantes es el que cambia de idioma según el archivo donde se configura.

Tu aplicación puede ajustar su contenido en varios idiomas a mostrar dependiendo de cuantos archivos strings.xml hayas configurado (uno por cada idioma)

Para este ejemplo hemos configurado 2 idiomas: inglés (archivo strings.xml por default) y español (archivo strings.xml **con el sufijo es**).

Existen una serie de sufijos que representan a cada idioma, puedes consultarlos aquí:

https://es.wikipedia.org/wiki/ISO_639-1

9. Ahora edita tu archivo de layout (en la carpeta **res/layout/activity_mail.xml**) de la siguiente forma:

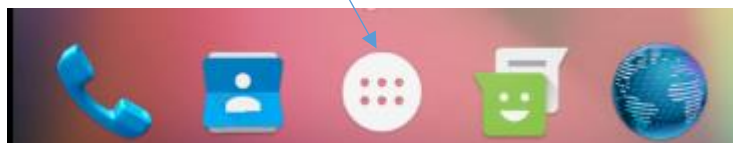
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:text="@string/saludo1"
        android:textSize="32sp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    <TextView
        android:text="@string/saludo2"
        android:textSize="32sp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

</LinearLayout>
```

Observa cómo se ha utilizado los recursos **@string/saludo1** y **@string/saludo2**. De esta forma Android decidirá cual recurso utilizar en tiempo de ejecución dependiendo de la configuración que el usuario haya establecido en su dispositivo móvil.

10. Configura el emulador para el idioma Ingles.
Haz clic en un icono similar al siguiente:



11. Luego en el icono de ajustes:



12. Ahora busca el icono (Idioma e introducción de texto)
(Language & input)
Reconócelo por el icono del mundo.



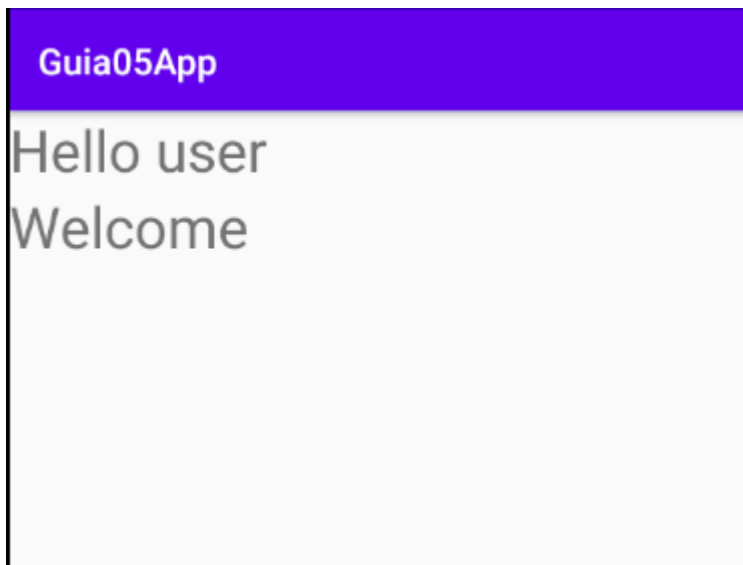
13. Ahora selecciona la opción **Idioma**:

Idioma
Español (España)

14. Busca y selecciona el idioma English
(United States)

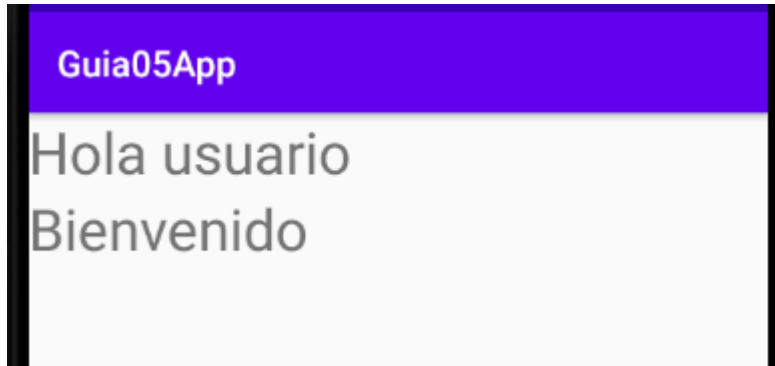
English (United States)

15. Ejecuta la app.
Verás que la aplicación muestra el contenido en inglés.



Vuelve a hacer los cambios en el emulador, pero ahora seleccionando el idioma Español (España).

Regrese a la aplicación o vuélvela a ejecutar y observarás que cambia su contenido a español.



IV. DISCUSION DE RESULTADOS

1. Agrega un archivo **strings** para brindar soporte en el idioma **francés** traduciendo los valores saludo1 y saludo2. Prueba la aplicación haciendo la configuración adecuada para el idioma francés.

V. BIBLIOGRAFIA

- developer.android. (s. f.). Información general sobre los recursos de las aplicaciones, de <https://developer.android.com/guide/topics/resources/providing-resources>