



**Guía de Laboratorio N°02(preliminar)**  
**(para desarrollarse en dos semanas)**

**“Introducción a la Interfaz de usuario de Android”**

**Objetivos:**

Que el estudiante:

- Aprenda la declaración e implementación de los controles básicos en la programación Android haciendo uso de **layouts** para realizar el diseño de la interfaz (en XML) y el uso de Fragments (en **java**) para definir el manejo de estos.
- Aprenda a Incorporar nuevos controles de forma intuitiva siguiendo la lógica de controles de similar comportamiento.

**Descripción:**

Esta práctica consistirá en crear un programa que contendrá una serie de **Fragments** que mostrarán el uso de cada uno de los controles básicos de Android (Button, TextView, EditText, CheckBox, etc) con su respectiva interfaz de usuario (usando XML). Cada **Fragment** será llamado por medio de un **Activity** que usaremos para mostrar un “Menú” por medio del componente **drawer\_layout**.



## Índice

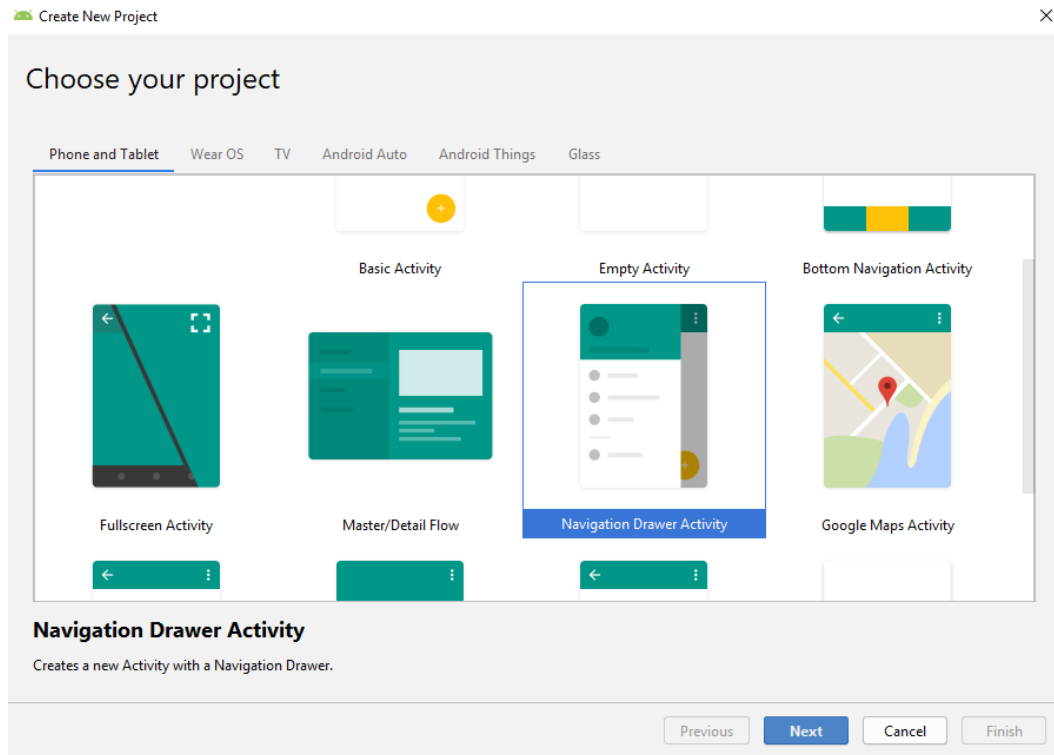
<b>Creación de Proyecto.....</b>	<b>1</b>
<b>Acondicionamiento del Menú (Principal) .....</b>	<b>3</b>
Cambiar la iconografía de la app.....	4
<b>Agregar primer Fragment(Button_Fragment).....</b>	<b>9</b>
Editar primer del Layout (fragment_button) .....	11
Llamar el Fragment de Botón desde el MainActivity .....	13
Configuración de la carga de opciones .....	14
Crear todos los Fragments restantes .....	16
(con Layouts de forma automática) .....	16
<b>Fragments.....</b>	<b>16</b>
Editar el segundo Layout (fragment_text_view).....	17
Editar el segundo fragment (EditView_Fragment) .....	18
Editar el tercer Layout (fragment_edit_text).....	19
Editar el tercer fragment (EditView_Fragment).....	20
Editar el cuarto Layout (fragment_check_box).....	21
Editar el cuarto fragment (Checkbox_Fragment) .....	22
Editar el quinto Layout (fragment_radio_button) .....	25
Editar el quinto fragment (RadioButton_Fragment).....	26
Editar el sexto Layout (fragment_gallery).....	28
Editar el sexto fragment (Gallery_Fragment) .....	28
Editar el séptimo Layout (fragment_spinner).....	31
Editar el séptimo fragment (Spinner_Fragment) .....	32
<b>Tarea opcional .....</b>	<b>34</b>
<b>Anexos .....</b>	<b>35</b>



## Creación de Proyecto

**Proyecto con menú (base un activity llamado navigationDrawer que invoca otros activities)**

Creas un nuevo proyecto, desde el menú de eclipse en **File->New Project** luego **navigationDrawer Activity** ubicado dentro de la subcarpeta **Android** y presione el botón **Next>**.



Presiona **Next**.



UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERIA Y ARQUITECTURA  
ESCUELA DE INGENIERIA DE SISTEMAS INFORMATICOS  
PROGRAMACION PARA DISPOSITIVOS MOVILES  
PDM115

Ciclo I-2019

Para este ejercicio escribe como nombre de la aplicación será **Carnet02**, El proyecto será: **sv.edu.ues.fia.eisi.carnet02** . selecciona **next** y busca las carpeta donde almacenas tus proyectos(carnet) .

Create New Project

### Configure your project

**Name**  
Carnet02

**Package name**  
sv.edu.ues.fia.eisi.carnet02

**Save location**  
C:\Users\Win10\Carnet02

**Language**  
Java

**Minimum API level**  
API 16: Android 4.1 (Jelly Bean)

**Navigation Drawer Activity**  
Creates a new Activity with a Navigation Drawer.

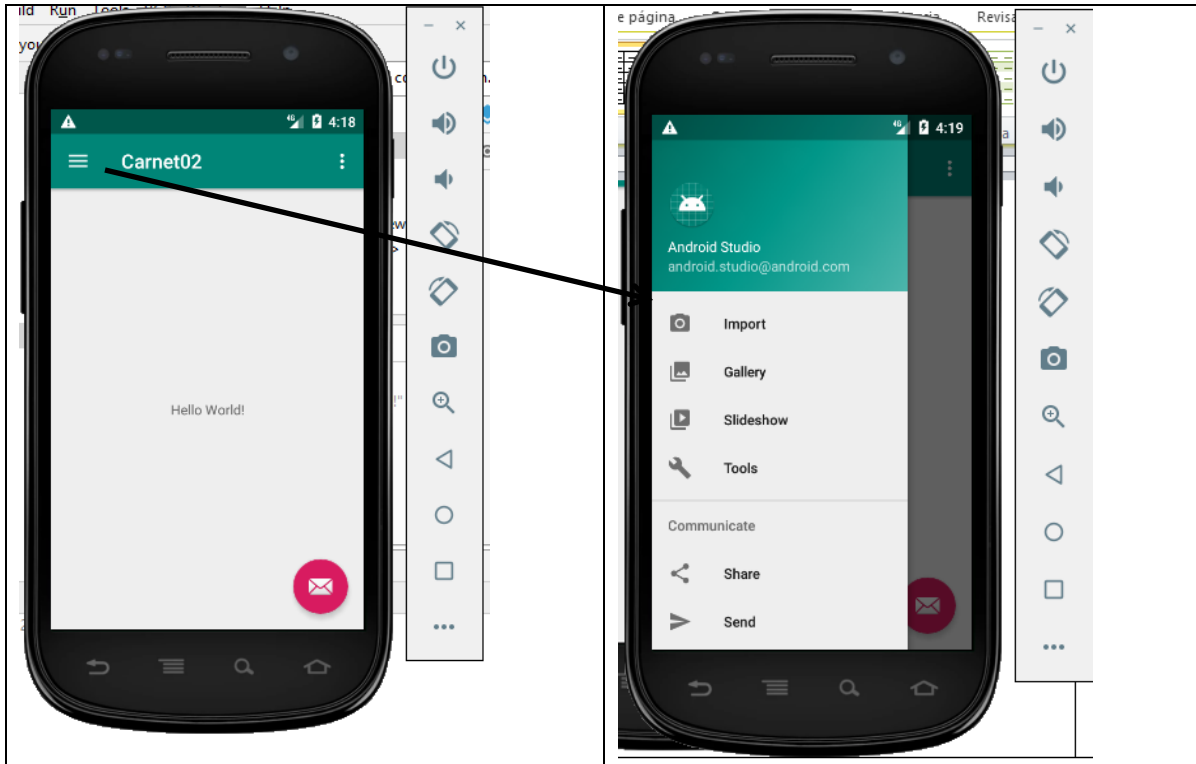
**Information:** Your app will run on approximately 99,6% of devices.  
[Help me choose](#)

This project will support instant apps

Previous Next Cancel Finish

Luego presiona finish

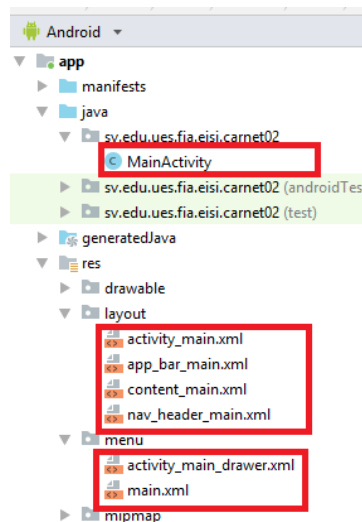
Lo ejecutamos para ver como esta configurado por default.



Con la configuración original se puede activar el menú default con presionar el botón de menú el cual contiene fragments para las opciones :Import, Galery, Slideshow y Tools.

## Acondicionamiento del Menú (Principal)

Inicialmente tendremos el árbol de proyecto así:





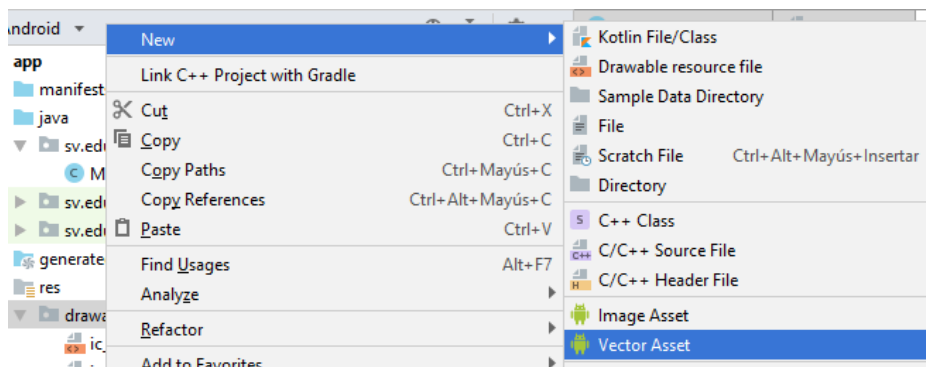
En nuestro menú tendremos las opciones: "Button", "TextView", "EditText", "CheckBox", "RadioButton", "Gallery", "Spinner" y "TabWidget"

Por lo que debemos cambiar en el `activity_main_drawer` los nombres de import por **Button**, Gallery por **TextView**, SlideShow por **EditText** y Tools por **CheckBox** y agregar las opciones: **"RadioButton", "Gallery", "Spinner" y "TabWidget"**

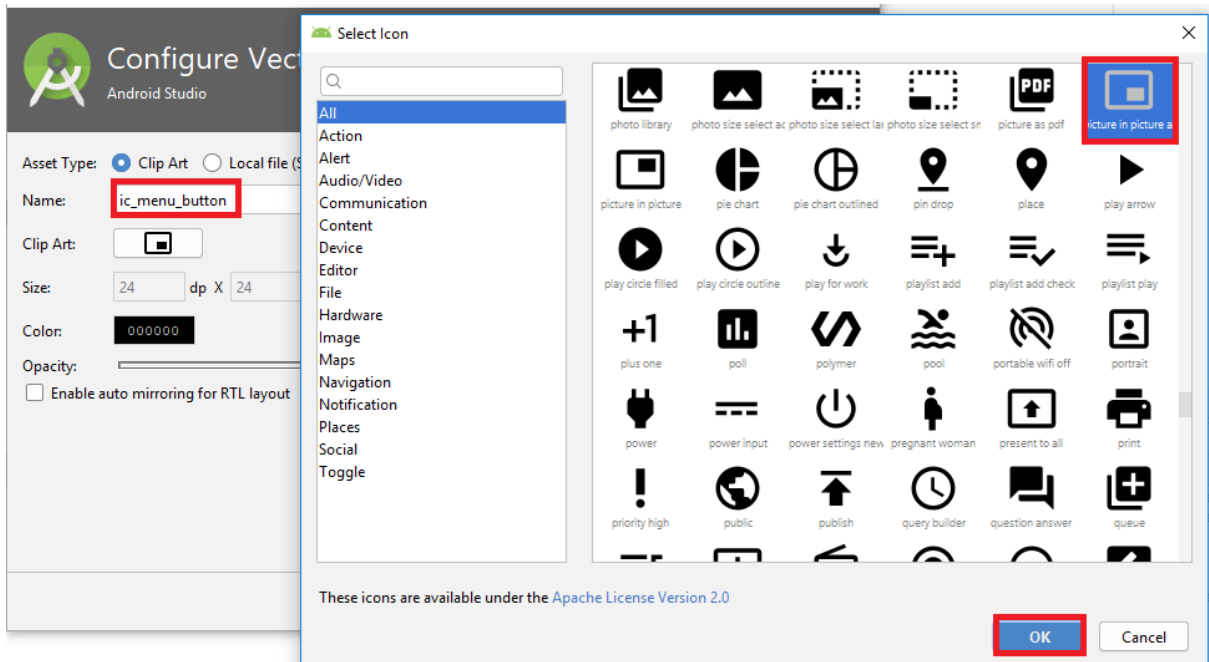
```
activity_main_drawer.xml
tools:showIn="navigation_view">
<group android:checkableBehavior="single">
  <item
    android:id="@+id/nav_button"
    android:icon="@drawable/ic_menu_button"
    android:title="Button" />
  <item
    android:id="@+id/nav_textview"
    android:icon="@drawable/ic_menu_textview"
    android:title="TextView" />
  <item
    android:id="@+id/nav_edittext"
    android:icon="@drawable/ic_menu_edittext"
    android:title="EditText" />
  <item
    android:id="@+id/nav_checkbox"
    android:icon="@drawable/ic_menu_checkbox"
    android:title="EditText" />
</group>
```

## Cambiar la iconografía de la app

Para Agregar iconos vaya a la carpeta de drawable y presione clic derecho, New, Vector Asset

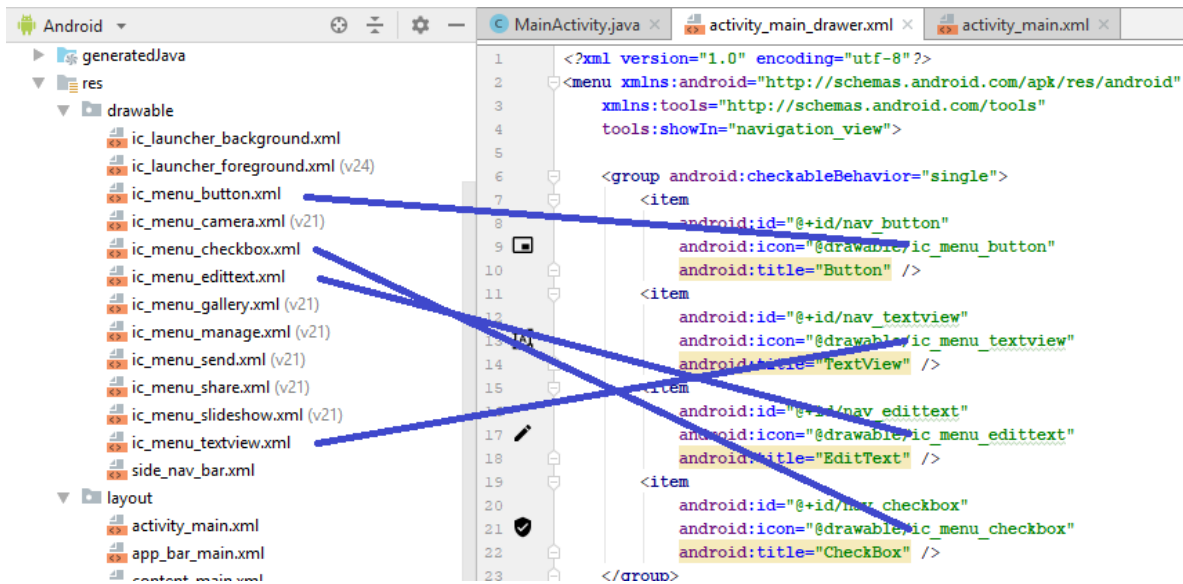


Definale el nombre de acuerdo a un estándar `ic_menu_XXXXX` donde `XXXXX` será el nombre de la opción, en este caso pondremos **button**.



Presionamos ok y realizamos esta tarea hasta completar los 4 iconos.

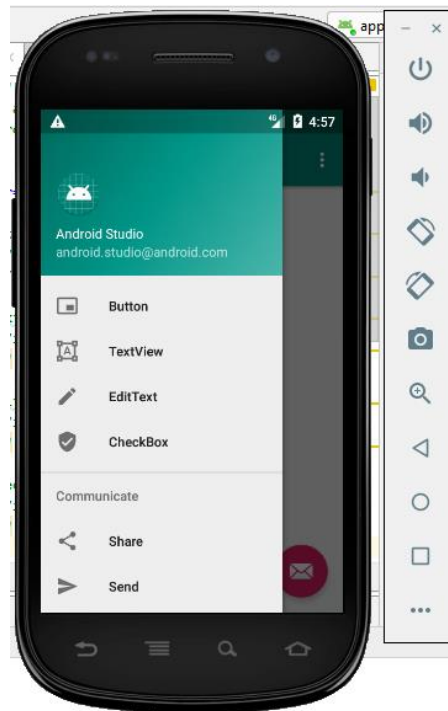
Al finalizar quedaran los recursos como se muestra(a la izquierda y los errores que aparecían en el activity\_main\_drawer desapareceran)



Luego modificamos nuestra MainActivity.java para ajustar los cambios en los identificadores de cada opción como se muestra

<pre> @Override public boolean onNavigationItemSelected(MenuItem item) {     // Handle navigation view item clicks here.     int id = item.getItemId();      if (id == R.id.nav_camera) {         // Handle the camera action     } else if (id == R.id.nav_gallery) {      } else if (id == R.id.nav_slideshow) {      } else if (id == R.id.nav_manage) {      } else if (id == R.id.nav_share) {      } else if (id == R.id.nav_send) {      } } </pre>	<pre> @Override public boolean onNavigationItemSelected(MenuItem item) {     // Handle navigation view item clicks here.     int id = item.getItemId();      if (id == R.id.nav_button) {         // Handle the camera action     } else if (id == R.id.nav_textview) {      } else if (id == R.id.nav_edittext) {      } else if (id == R.id.nav_checkbox) {      } else if (id == R.id.nav_share) {      } else if (id == R.id.nav_send) {      } } </pre>
Antes	Despues

Ejecute la App de nuevo



Y vera los cambios en las opciones...





Realice los pasos anteriores para los otros cuatro controles ("RadioButton", "Gallery", "Spinner" y "TabWidget") con los iconos que usted crea convenientes.

Además, elimine el código del activity\_main\_drawer.xml que se le muestra:

```
MainActivity.java x activity_main_drawer.xml x activity_main.xml x
<item
  android:id="@+id/nav_checkbox"
  android:icon="@drawable/ic_menu_checkbox"
  android:title="CheckBox" />
</group>
<item android:title="Communicate">
  <menu>
    <item
      android:id="@+id/nav_share"
      android:icon="@drawable/ic_menu_share"
      android:title="Share" />
    <item
      android:id="@+id/nav_send"
      android:icon="@drawable/ic_menu_send"
      android:title="Send" />
    </menu>
  </item>
</menu>
```

Elimine los iconos innecesarios de la carpeta drawable(ic\_menu\_camera, ic\_menu\_share, etc)

Con lo anterior el árbol del proyecto y el contenido del archivo activity\_main\_drawer quedara como se muestra

```
Android
├── MainActivity
│   ├── sv.edu.ues.fia.eisi.carnet02 (androidTest)
│   └── sv.edu.ues.fia.eisi.carnet02 (test)
├── generatedJava
└── res
    ├── drawable
    │   ├── ic_launcher_background.xml
    │   ├── ic_launcher_foreground.xml (v24)
    │   ├── ic_menu_button.xml
    │   ├── ic_menu_checkbox.xml
    │   ├── ic_menu_edittext.xml
    │   ├── ic_menu_gallery.xml
    │   ├── ic_menu_manage.xml (v21)
    │   ├── ic_menu_radio_button.xml
    │   ├── ic_menu_spinner.xml
    │   ├── ic_menu_tabwidget.xml
    │   ├── ic_menu_textview.xml
    │   └── side_nav_bar.xml
    └── layout
        ├── activity_main.xml
        ├── app_bar_main.xml
        ├── content_main.xml
        ├── nav_header_main.xml
        └── activity_main_drawer.xml
            ├── activity_main_drawer.xml
            └── main.xml

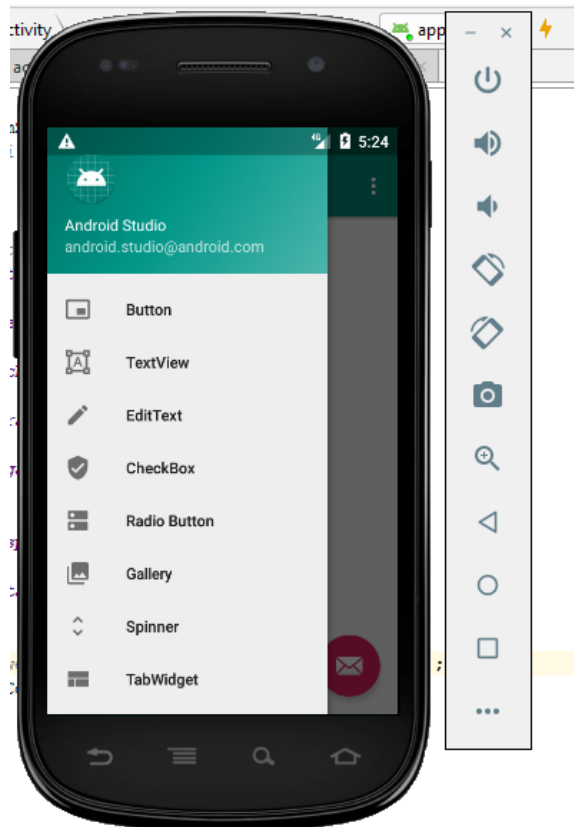
MainActivity.java x content_main.xml x activity_main_drawer.xml x
9
10 android:icon="@drawable/ic_menu_button"
11 android:title="Button" />
12 <item
13 android:id="@+id/nav_textview"
14 android:icon="@drawable/ic_menu_textview"
15 android:title="TextView" />
16 <item
17 android:id="@+id/nav_edittext"
18 android:icon="@drawable/ic_menu_edittext"
19 android:title="EditText" />
20 <item
21 android:id="@+id/nav_checkbox"
22 android:icon="@drawable/ic_menu_checkbox"
23 android:title="CheckBox" />
24 <item
25 android:id="@+id/nav_radiobutton"
26 android:icon="@drawable/ic_menu_radio_button"
27 android:title="Radio Button" />
28 <item
29 android:id="@+id/nav_gallery"
30 android:icon="@drawable/ic_menu_gallery"
31 android:title="Gallery" />
32 <item
33 android:id="@+id/nav_spinner"
34 android:icon="@drawable/ic_menu_spinner"
35 android:title="Spinner" />
36 <item
37 android:id="@+id/nav_tabwidget"
38 android:icon="@drawable/ic_menu_tabwidget"
39 android:title="TabWidget" />
</group>
```



El archivo MainActivity.java quedara así:

```
int id = item.getItemId();  
  
if (id == R.id.nav_button) {  
    // Handle the camera action  
} else if (id == R.id.nav_textview) {  
  
} else if (id == R.id.nav_edittext) {  
  
} else if (id == R.id.nav_checkbox) {  
  
} else if (id == R.id.nav_radiobutton) {  
  
} else if (id == R.id.nav_gallery) {  
  
} else if (id == R.id.nav_spinner) {  
  
} else if (id == R.id.nav_tabwidget) {  
  
}  
DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);  
drawer.closeDrawer(GravityCompat.START);  
return true;  
}
```

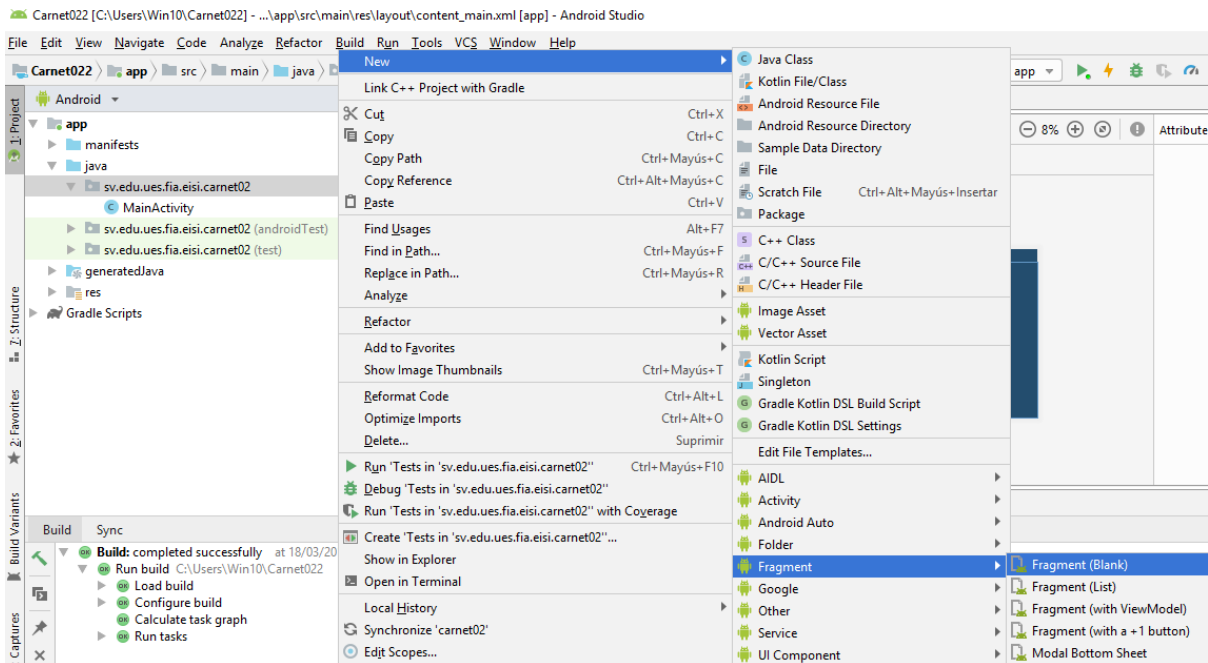
Al verificar lo anterior Vuelva a ejecutarlo.



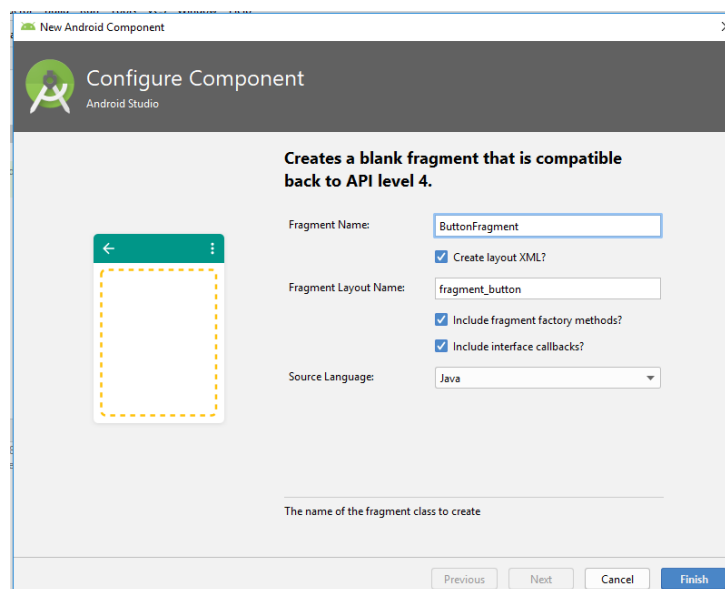
En este momento tiene finalizado su menú sin que se pueda ejecutar ninguna pantalla adicional.

## Agregar primer Fragment(Button\_Fragment)

Para Agregar la opción Button se deberá crear un fragment denominado ButtonFragment, presione clic derecho, New, Fragment.



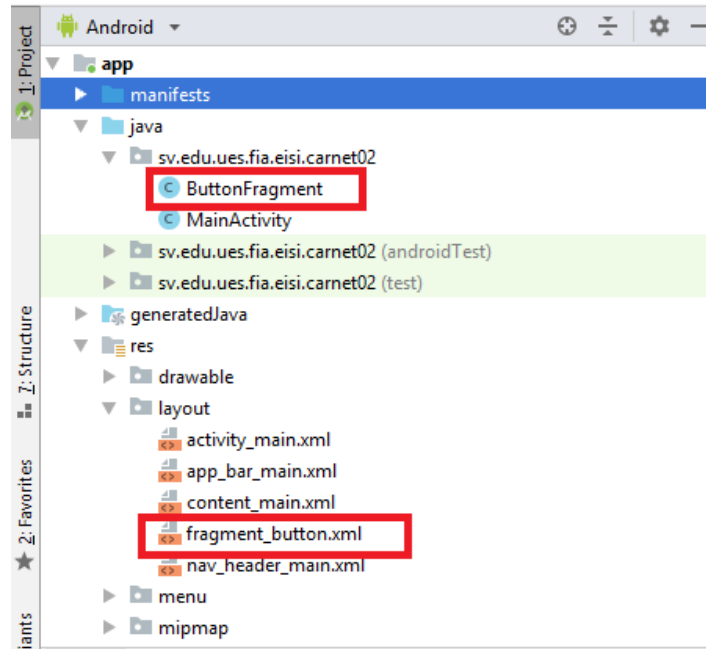
Se selecciona un Blank Fragment



Se le define el nombre (ButtonFragment)



Con lo anterior se crea un archivo de Layout y su programa de aplicacion



En el archivo fuente de java hacemos una eliminacion de todo el codigo fuente a excepcion del metodo onCreateView

```
// TODO: Rename and change types and number of parameters
public static ButtonFragment newInstance(String param1, String param2) {
    ButtonFragment fragment = new ButtonFragment();
    Bundle args = new Bundle();
    args.putString(ARG_PARAM1, param1);
    args.putString(ARG_PARAM2, param2);
    fragment.setArguments(args);
    return fragment;
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_button, container, attachToRoot: false);
}

// TODO: Rename method, update argument and hook method into UI event
public void onPressed(Uri uri) {
    if (mListener != null) {
```

Puede consultar Anexo A si tiene dudas de como queda el cogigo fuente(.java)



## Editar primer del Layout (fragment\_button)

Entre al Layout del Button y sustitúyalo por este código.... Puede utilizar el código de Anexo B

```
<?xml version="1.0" encoding="utf-8" ?>
<?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"
    android:orientation="vertical"
    tools:context=".ButtonFragment">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:layout_width="277dp"
        android:layout_height="69dp"
        android:layout_marginTop="60dp"
        android:text="@string/TextView1_Btn"
        android:textSize="24sp"
        android:textStyle="bold"
        tools:textColor="@tools:sample/avatars" />

    <Button
        android:id="@+id/button"
        android:layout_width="275dp"
        android:layout_height="73dp"
        android:text="@string/Button1"
        android:textColor="@color/colorAccent"
        android:textSize="16sp"
        />
</LinearLayout>
```

Note que aparecen dos errores debido a que no existen las variables string del TextView y el Button...

Para que desaparezcan estos errores cree las variables en el archivo string.XML

```
<resources>
  <string name="app_name">Carnet02</string>
  <string name="navigation_drawer_open">Open navigation drawer</string>
  <string name="navigation_drawer_close">Close navigation drawer</string>
  <string name="nav_header_title">Android Studio</string>
  <string name="nav_header_subtitle">android.studio@android.com</string>
  <string name="nav_header_desc">Navigation header</string>
  <string name="action_settings">Settings</string>

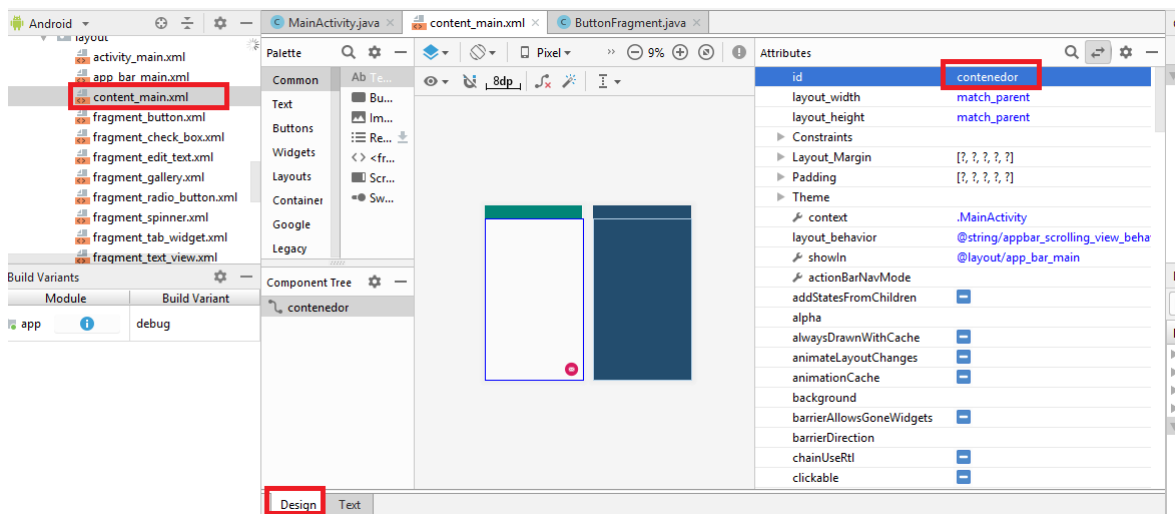
  <!-- TODO: Remove or change this placeholder text -->
  <string name="hello_blank_fragment">Hello blank fragment</string>
  <!-- Variables de boton -->
  <string name="TextView1 Btn">Texto en Mensaje</string>
  <string name="Button1">Presione Boton</string>
</resources>
```

```
<resources>
```

Al crear dichas variables desaparecerán los errores del layout del button

Con lo anterior USTED tiene creado el Fragment y el Layout respectivo, solo falta que se invoque desde el MainActivity

Para ello hay que especificar que el content\_main.xml será el contenedor de los demás fragments, es decir que deberá asignar un id(valor=contenedor) que identifique al layout, para luego indicar al Fragment Manager que levante una instancia con ese layout “contenedor”.





## Llamar el Fragment de Botón desde el MainActivity

Agregue en el método onCreate el código siguiente:

```
DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
    activity: this, drawer, toolbar, "Open navigation drawer", "Close navigation drawer");
drawer.addDrawerListener(toggle);
toggle.syncState();

NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
navigationView.setNavigationItemSelectedListener(this);

FragmentManager fragmentManager=getSupportFragmentManager();
fragmentManager.beginTransaction().replace(R.id.contenedor,new ButtonFragment()).commit();

}

@Override
public void onBackPressed() {
```

Con lo anterior se ha configurado un **fragment default**.

Al finalizar la práctica puede crear un fragment mejor que tenga una imagen o texto que identifique a su aplicación.



## Configuración de la carga de opciones

Actualice el código en el método `onNavigationItemSelected`

```
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();
    FragmentManager fragmentManager=getSupportFragmentManager();
    if (id == R.id.nav_button) {
        fragmentManager.beginTransaction().replace(R.id.contenedor,new ButtonFragment()).commit();
    }
    else if (id == R.id.nav_textview) {
    }
    else if (id == R.id.nav_edittext) {
    }
```

Actualizar el código del Fragment. necesario para la ejecución del método click de botón, implementando la interfaz View. OnClickListener

```
public class ButtonFragment extends Fragment implements View.OnClickListener{
    // TODO: Rename parameter arguments, choose names that match
    Button boton;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        // Inflate the layout for this fragment

        View view = inflater.inflate(R.layout.fragment_button, container, attachToRoot: false);

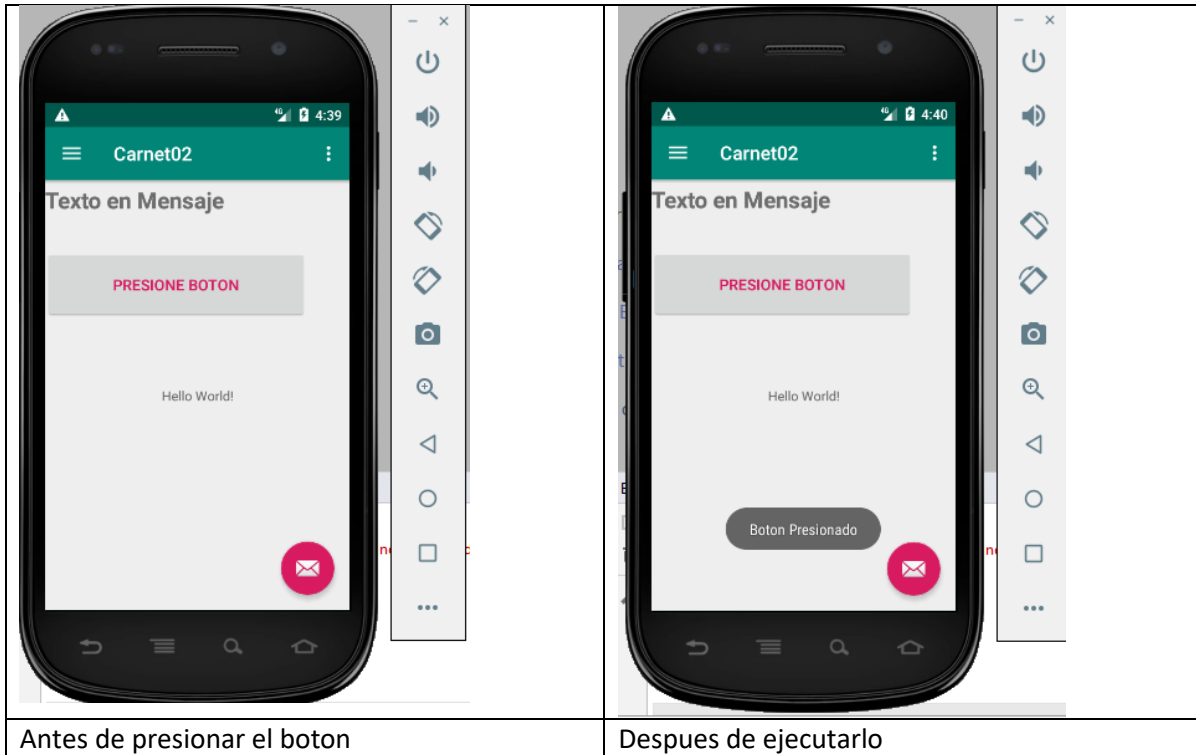
        Button button = (Button) view.findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(getContext(),
                    text: "Boton Presionado", Toast.LENGTH_SHORT).show();
            }
        });
        return view;
    }

    @Override
```

Nota: el otro método `onclick()` no lo utilice pero no debe borrarlo ya que al implementar la interfaz lo requiere es decir que aunque no lo utilice debe estar.



Ejecute la app, a este momento aparecerá...



**Nota:** Suprima el código XML para que no aparezca el mensaje Hello World

En este momento ya termino la programación de la primera opción del menú... continuaremos con la programación de las siguientes 7 opciones (aplicación java e interfaz xml)

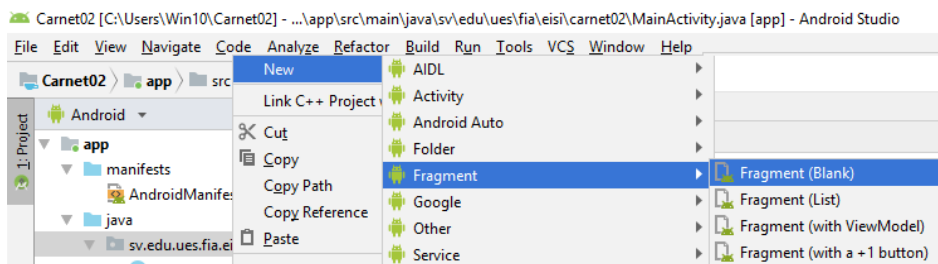
# Crear todos los Fragments restantes (con Layouts de forma automática)

## Fragments

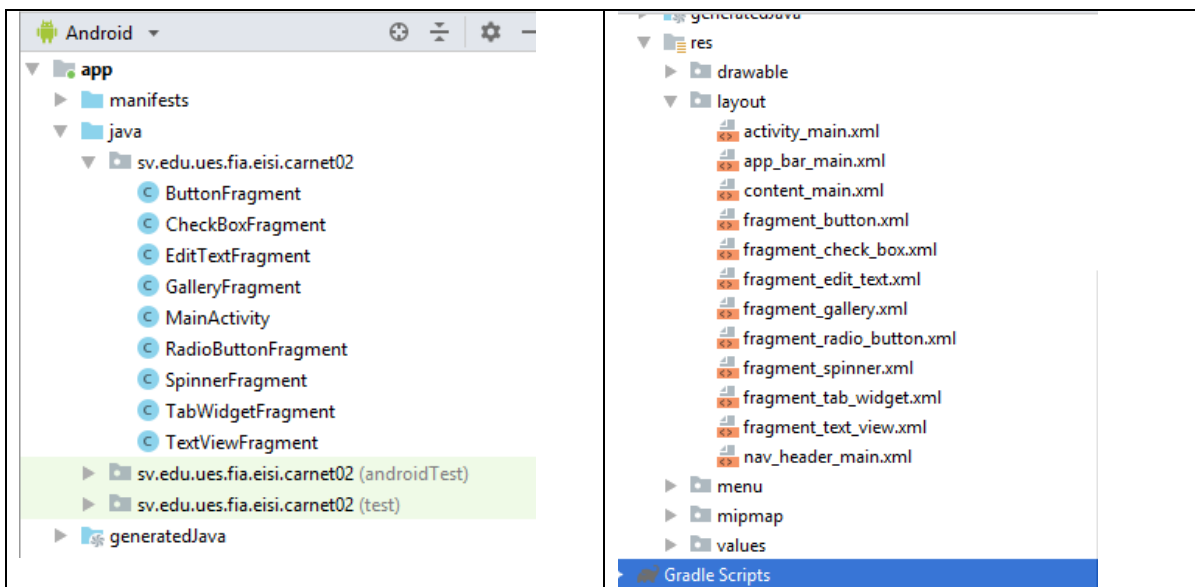
Para cada uno de los fragments restantes haga lo siguiente:

Sobre el paquete `java/sv.edu.eisi.carnet02` presione clic derecho, New, Fragment, Fragment Blank.

Escriba el nombre del fragment: "TextViewFragment", "EditTextFragment", "CheckBoxFragment", "RadioButtonFragment", "GalleryFragment", "SpinnerFragment" y "TabWidgetFragment"



Con lo anterior el árbol de proyecto quedaría así:





Actualice el código en el método `onNavigationItemSelected` para que tenga la invocación de todos los fragments restantes

```
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();

    FragmentManager fragmentManager=getSupportFragmentManager();
    if (id == R.id.nav_button) {
        fragmentManager.beginTransaction().replace(R.id.contenedor,new ButtonFragment()).commit();
    }
    else if (id == R.id.nav_textview) {
        fragmentManager.beginTransaction().replace(R.id.contenedor,new TextViewFragment()).commit();
    }
    else if (id == R.id.nav_edittext) {
        fragmentManager.beginTransaction().replace(R.id.contenedor,new EditTextFragment()).commit();
    }
    else if (id == R.id.nav_checkbox) {
        fragmentManager.beginTransaction().replace(R.id.contenedor,new CheckBoxFragment()).commit();
    }
    else if (id == R.id.nav_radiobutton) {
        fragmentManager.beginTransaction().replace(R.id.contenedor,new RadioButtonFragment()).commit();
    }
    else if (id == R.id.nav_gallery) {
        fragmentManager.beginTransaction().replace(R.id.contenedor,new GalleryFragment()).commit();
    }
    else if (id == R.id.nav_spinner) {
        fragmentManager.beginTransaction().replace(R.id.contenedor,new SpinnerFragment()).commit();
    }
    else if (id == R.id.nav_tabwidget) {
        fragmentManager.beginTransaction().replace(R.id.contenedor,new TabWidgetFragment()).commit();
    }
}
DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
drawer.closeDrawer(GravityCompat.START);
```

## Editar el segundo Layout (fragment\_text\_view)

Entre al layout y sustituya el código por el siguiente



```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout 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=".TextViewFragment">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="@string/text"
        android:textSize="30sp" />
</FrameLayout>
```

Edite el archivo strings para agregar la variable text y póngale el mensaje "2. TextView" como hizo con el button.

## Editar el segundo fragment (EditView\_Fragment)

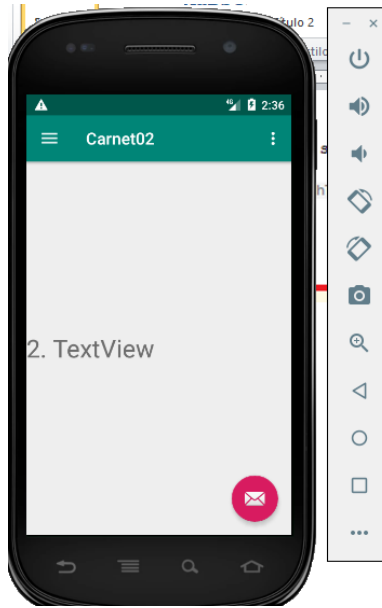
Edite el Fragment EditView\_Fragment.java para que solo quede el método CreateView como hizo con el ButtonFragment. Pero **no implemente la interfaz View.OnClickListener.**

```
package sv.edu.ues.fia.eisi.carnet02;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class TextViewFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_text_view, container, attachToRoot: false);

        return view;
    }
}
```

Vuelva a ejecutar la aplicación y aparecerá como se muestra:



## Editar el tercer Layout (fragment\_edit\_text)

Entre al layout y sustituya el código por el siguiente

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <EditText
        android:id="@+id/edit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/mensajeayuda"/>

    <ToggleButton
        android:id="@+id/toggle"
        android:checked="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_below="@id/edit" />
</RelativeLayout>
```

Al igual que en el layout anterior, defina la variable string **mensajeayuda**



```
<resources>
    <string name="app_name">Carnet02</string>
    <string name="navigation_drawer_open">Open navigation drawer</string>
    <string name="navigation_drawer_close">Close navigation drawer</string>
    <string name="nav_header_title">Android Studio</string>
    <string name="nav_header_subtitle">android.studio@android.com</string>
    <string name="nav_header_desc">Navigation header</string>
    <string name="action_settings">Settings</string>

    <!-- TODO: Remove or change this placeholder text -->
    <string name="hello_blank_fragment">Hello blank fragment</string>
    <!-- Variables de boton -->
    <string name="TextView1_Btn">Texto en Mensaje</string>
    <string name="Button1">Presione Boton</string>
    <string name="text">2. TextView</string>

    <string name="mensajeayuda">3. EditText</string>
</resources>
```

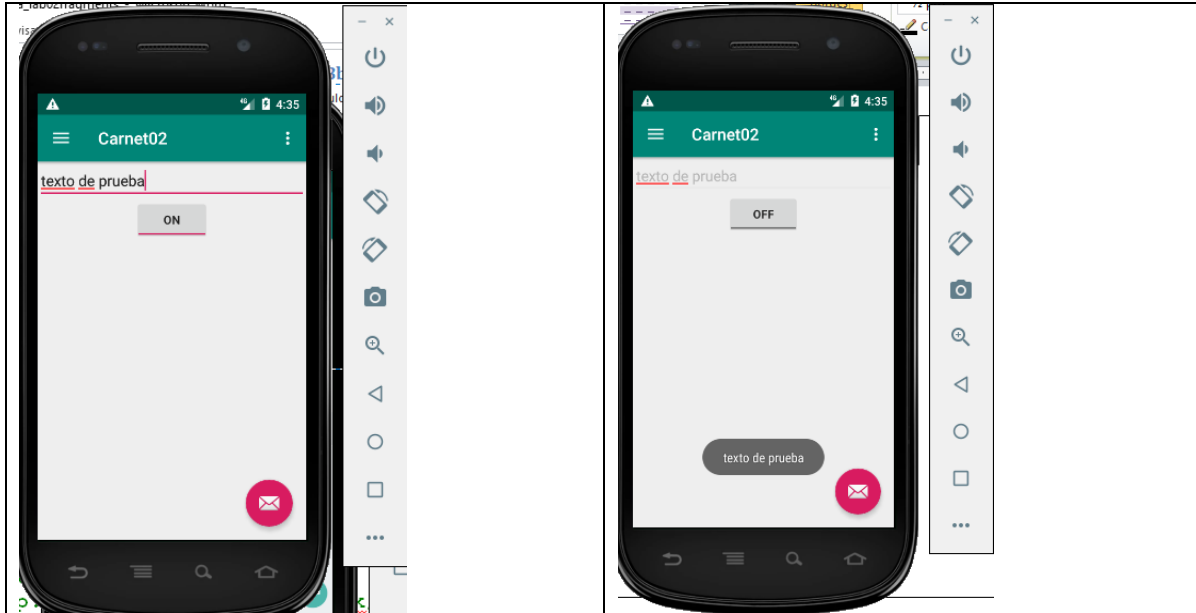
## Editar el tercer fragment (EditView\_Fragment)

Edite el Fragment `EditView_Fragment.java` para que solo quede el método `CreateView` como hizo con el `ButtonFragment`. Este Fragment si implementara **la interfaz `View.OnClickListener`**.

```
public class EditTextFragment extends Fragment implements View.OnClickListener{

    EditText edt;
    ToggleButton bt;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_edit_text, container, attachToRoot: false);
        edt= view.findViewById(R.id.edit);
        bt=view.findViewById(R.id.toggle);
        bt.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                if(bt.isChecked()){
                    edt.setEnabled(true);
                }else{
                    edt.setEnabled(false);
                }
                Toast.makeText(v.getContext(), edt.getText().toString(), Toast.LENGTH_SHORT).show();
            }
        });
        return view;
    }
    public void onClick(View v) {
    }
}
```

Ejecute la aplicación de nuevo



## Editar el cuarto Layout (fragment\_check\_box)

Entre al layout y sustituya el código por el siguiente

```
<?xml version="1.0" encoding="utf-8" ?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:stretchColumns="1">
    <TableRow>
        <TextView android:text="@string/titulocheques" />
    </TableRow>
    <TableRow>
        <TextView android:text="@string/opcion1"/>
        <CheckBox android:id="@+id/checkbox1"
            android:text="@string/text1"
            android:onClick="miMetodo1"
            />
    </TableRow>
    <TableRow>
        <TextView android:text="@string/opcion2" />
        <CheckBox android:id="@+id/checkbox2"
            android:text="@string/text2"
            android:onClick="miMetodo2"
            />
    </TableRow>
    <TableRow>
        <TextView android:text="@string/opcion3" />
        <CheckBox android:id="@+id/checkbox3"
            />
    </TableRow>
</TableLayout>
```



```
        android:text="@string/text3"  
        android:onClick="miMetodo3"  
    />  
</TableRow>  
</TableLayout>
```

Aparecen errores de las variables string... corríjelas con el siguiente código xml

```
<string name="titulochequeos">Listado de Checkbox</string>  
<string name="opcion1">Checkbox 1</string>  
<string name="opcion2">Checkbox 2</string>  
<string name="opcion3">Checkbox 3</string>  
<string name="text1">2</string>  
<string name="text2">4</string>  
<string name="text3">8</string>
```

Con lo anterior la interfaz quedara así(en vista diseño):



## Editar el cuarto fragment (Checkbox\_Fragment)

Edite el Fragment CheckBox\_Fragment.java para que solo quede el método CreateView como hizo con el ButtonFragment. Este Fragment si implementara **la interfaz View.OnClickListener**.

```
package sv.edu.ues.fia.eisi.carnet02;  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.CheckBox;  
import android.widget.Toast;  
  
public class CheckBoxFragment extends Fragment implements  
View.OnClickListener {
```

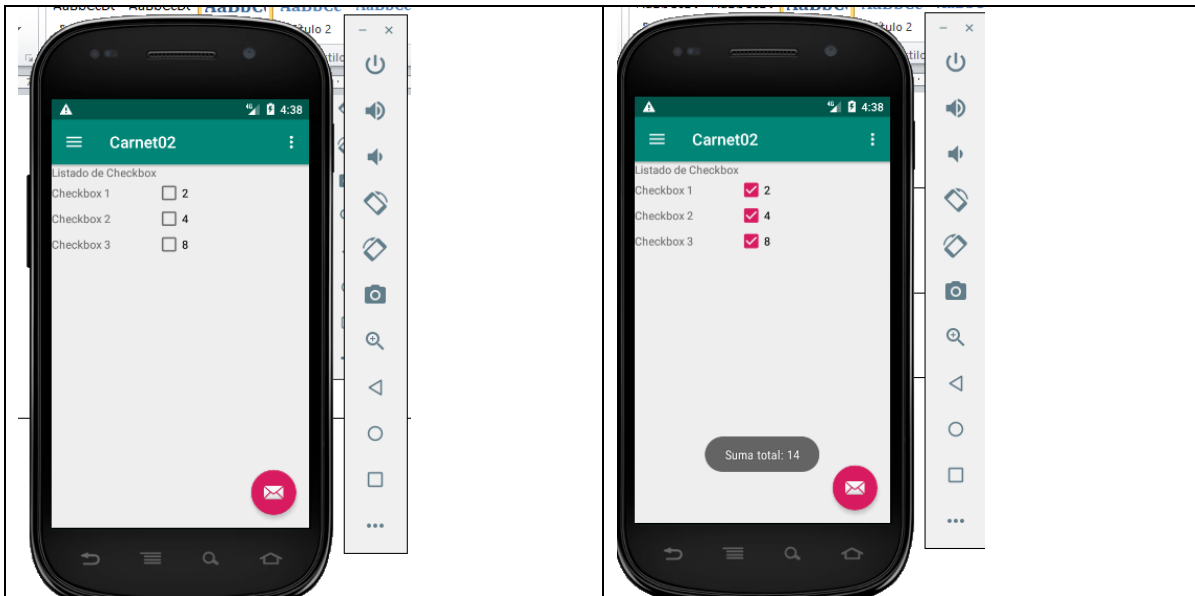




```
int sumador=0;
int acum1=0;
int acum2=0;
int acum3=0;
CheckBox save1, save2, save3;
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
container,
Bundle savedInstanceState) {
View view = inflater.inflate(R.layout.fragment_check_box,
container, false);
save1 = (CheckBox)view.findViewById(R.id.checkbox1);
save2 = (CheckBox)view.findViewById(R.id.checkbox2);
save3 = (CheckBox)view.findViewById(R.id.checkbox3);
save1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
if (save1.isChecked()) {
Toast.makeText(v.getContext(), "Seleccionado
Checkbox1",
Toast.LENGTH_SHORT).show();
acum1=2;
} else {
acum1=0;
}
acumular();
}
});
save2.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
if (save2.isChecked()) {
Toast.makeText(v.getContext(), "Seleccionado
Checkbox2",
Toast.LENGTH_SHORT).show();
acum2=4;
} else {
acum2=0;
}
acumular();
}
});
save3.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
if (save3.isChecked()) {
Toast.makeText(v.getContext(), "Seleccionado
Checkbox3",
Toast.LENGTH_SHORT).show();
acum3=8;
} else {
acum3=0;
}
```

```
    }  
    acumular();  
  }  
});  
return view;  
}  
public void acumular(){  
    sumador=acum1+acum2+acum3;  
    Toast.makeText(this.getContext(), "Suma total: "+ sumador,  
    Toast.LENGTH_LONG).show();  
}  
@Override  
public void onClick(View v) {  
}  
}
```

Ejecute la aplicación de nuevo





## Editar el quinto Layout (fragment\_radio\_button)

Entre al layout y sustituya el código por el siguiente

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

    <RadioGroup
        android:id="@+id/grupo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:layout_centerHorizontal="true">

        <RadioButton
            android:id="@+id/radiobutton1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/RadioButton1"
            />

        <RadioButton
            android:id="@+id/radiobutton2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/RadioButton2"
            />

        <RadioButton
            android:id="@+id/radiobutton3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/RadioButton3"
            />

    </RadioGroup>

    <TextView
        android:id="@+id/texto"
        android:text="@string/mensaje_error_radiobutton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_below="@id/grupo"/>

</RelativeLayout>
```



Aparecen errores de las variables string... corríjalas con el siguiente código xml

```
<string name="RadioButton1">opc 1</string>
<string name="RadioButton2">opc 2</string>
<string name="RadioButton3">opc 3</string>
<string name="mensaje_error_radiobutton">Ningun RadioButton
seleccionado</string>
```

## Editar el quinto fragment (RadioButton\_Fragment)

Edite el Fragment RadioButton\_Fragment.java para que solo quede el método CreateView como hizo con el ButtonFragment. Este Fragment si implementara **la interfaz View.OnClickListener**.

```
package sv.edu.ues.fia.eisi.carnet02;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

public class RadioButtonFragment extends Fragment implements
View.OnClickListener{
    RadioGroup rbg;
    TextView text;
    Button bt1, bt2, bt3;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                            Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_radio_button,
container,
                                false);
        rbg=(RadioGroup) view.findViewById(R.id.grupo);
        bt1 =(RadioButton) view.findViewById(R.id.radiobutton1);
        bt2 =(RadioButton) view.findViewById(R.id.radiobutton2);
        bt3 =(RadioButton) view.findViewById(R.id.radiobutton3);
        text=(TextView) view.findViewById(R.id.texto);
        bt1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                boolean checked = ((RadioButton) v).isChecked();
                // Check which radiobutton was pressed
                if (checked) {
```





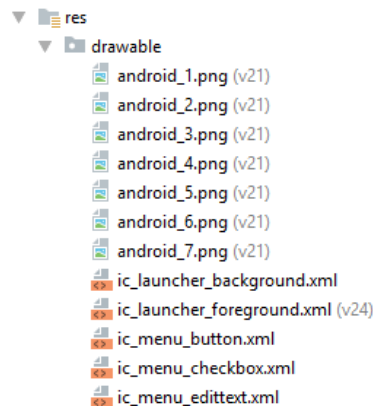
## Editar el sexto Layout (fragment\_gallery)

Entre al layout y sustituya el código por el siguiente

```
<?xml version="1.0" encoding="utf-8"?>
<Gallery xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/gallery"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" >
</Gallery>
```

Descargue la carpeta recursos2 del aula virtual o del ftp de la asignatura y descomprímalo en una carpeta en el escritorio

Luego cópielos archivos (clic derecho, copy) a la carpeta res/drawable de su proyecto (clic derecho, paste)



## Editar el sexto fragment (Gallery\_Fragment)

Edite el Fragment RadioButton\_Fragment.java para que solo quede el método CreateView como hizo con el ButtonFragment. Este Fragment no implementará **la interfaz View.OnClickListener**.

```
package sv.edu.ues.fia.eisi.carnet02;

import android.content.Context;
import android.content.res.TypedArray;
import android.net.Uri;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
```



UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERIA Y ARQUITECTURA  
ESCUELA DE INGENIERIA DE SISTEMAS INFORMATICOS  
PROGRAMACION PARA DISPOSITIVOS MOVILES  
PDM115

Ciclo I-2019

```
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.Gallery;
import android.widget.ImageView;
import android.widget.Toast;

public class GalleryFragment extends Fragment {

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                            Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_gallery,
container,
                                false);

        @SuppressWarnings({ "deprecation", "deprecation" })
        Gallery gallery = (Gallery) view.findViewById(R.id.gallery);
        gallery.setAdapter(new ImageAdapter(view.getContext()));

        gallery.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            public void onItemClick(@SuppressWarnings("rawtypes")
AdapterView parent, View v, int position, long id) {
                //Toast.makeText(view.getContext(), "Usted Visualiza la
imagen numero:" + position, Toast.LENGTH_SHORT).show();
            }
        });
        return view;
    }

    public class ImageAdapter extends BaseAdapter {
        int mGalleryItemBackground;
        private Context mContext;

        private Integer[] mImageIds = {
            R.drawable.android_1,
            R.drawable.android_2,
            R.drawable.android_3,
            R.drawable.android_4,
            R.drawable.android_5,
            R.drawable.android_6,
            R.drawable.android_7
        };

        public ImageAdapter(Context c) {
            mContext = c;
            TypedArray attr =
```



```
mContext.obtainStyledAttributes(R.styleable.GalleryActivity);
    mGalleryItemBackground = attr.getResourceId(
R.styleable.GalleryActivity_android_galleryItemBackground, 0);
    // attr.recycle();
}

public int getCount() {
    return mImageIds.length;
}

public Object getItem(int position) {
    return position;
}

public long getItemId(int position) {
    return position;
}

public View getView(int position, View convertView, ViewGroup
parent) {
    ImageView imageView = new ImageView(mContext);

    imageView.setImageResource(mImageIds[position]);
    imageView.setLayoutParams(new Gallery.LayoutParams(150,
100));
    imageView.setScaleType(ImageView.ScaleType.FIT_XY);
    imageView.setBackgroundResource(mGalleryItemBackground);

    return imageView;
}
}
```

#### Observaciones:

Cabe recalcar que para lograr ejecutar una View de Gallery es necesario que cada componente funcione de acuerdo con el método `setOnItemClickListener(new OnItemClickListener` en cual permite que se mande a la clase `public class ImageAdapter extends BaseAdapter`, la posición de la selección y el ítem que se ha seleccionado. Dentro de la clase `IMAGEADAPTER` es donde se encuentra el diseño de gallery y los métodos que por defecto reconoce como lo es `public int getCount()`. Dentro de esta misma clase se coloca un arreglo en donde se ponen las imágenes que deben agregarse en la carpeta **res-> drawable**

El método `ImageAdapter` es el que permite utilizar lo que es el background por default utilizado en el sistema Android, esto lo hace a través del xml denominado `attrs.xml` el cual lo crearemos posteriormente. Y por último el método `public View getView(int position, View convertView,`



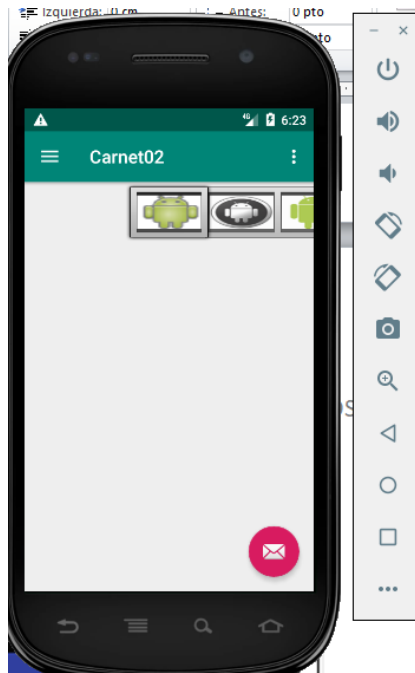


ViewGroup parent) que es en el cual se reúnen todos los elementos esenciales del diseño de Galley para crear lo que es su vista a través del XML GalleryView.

Ahora es necesario irse a la carpeta **res->values** y crear un nuevo resource con el nombre de *attrs.xml*, y colocarle el siguiente código:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <declare-styleable name="GalleryActivity">
    <attr name="android:galleryItemBackground" />
  </declare-styleable>
</resources>
```

Al correr nuestra aplicación y seleccionar la opción "Gallery" de la lista se observará de la siguiente manera:



## Editar el septimo Layout (fragment\_spinner)

Entre al layout y sustituya el código por el siguiente

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



```
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dip"
    android:text="@string/colors_prompt"
/>

<Spinner
    android:id="@+id/spinner"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:prompt="@string/colors_prompt"
/>

</LinearLayout>
```

Agregue las siguientes variables en el archivo strings.xml

```
<string name="colors_prompt">Seleccione un Color</string>
<string-array name="colors_array">
    <item>Blanco</item>
    <item>Negro</item>
    <item>Azul</item>
    <item>Rojo</item>
    <item>Verde</item>
    <item>Amarillo</item>
    <item>Gris</item>
    <item>Rosado</item>
</string-array>
```

## Editar el septimo fragment (Spinner\_Fragment)

Edite el Fragment Spinner\_Fragment.java para que solo quede el método CreateView como hizo con el ButtonFragment. Este Fragment no implementara **la interfaz View.OnClickListener**.

```
package sv.edu.ues.fia.eisi.carnet02;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.Toast;
public class SpinnerFragment extends Fragment implements
```



```
AdapterView.OnItemSelectedListener {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                            Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_spinner,
container,
                                false);

        Spinner spinner = (Spinner) view.findViewById(R.id.spinner);
        ArrayAdapter<CharSequence> adapter =
ArrayAdapter.createFromResource(
            view.getContext(), R.array.colors_array,
android.R.layout.simple_spinner_item);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_
item);

        spinner.setAdapter(adapter);
        spinner.setOnItemSelectedListener(this);

        return view;
    }

    public void onItemClick(AdapterView<?> parent, View view, int pos,
                            long id) {
        Toast.makeText(parent.getContext(), "Color: " +
            parent.getItemAtPosition(pos).toString(),
Toast.LENGTH_LONG).show();

    }

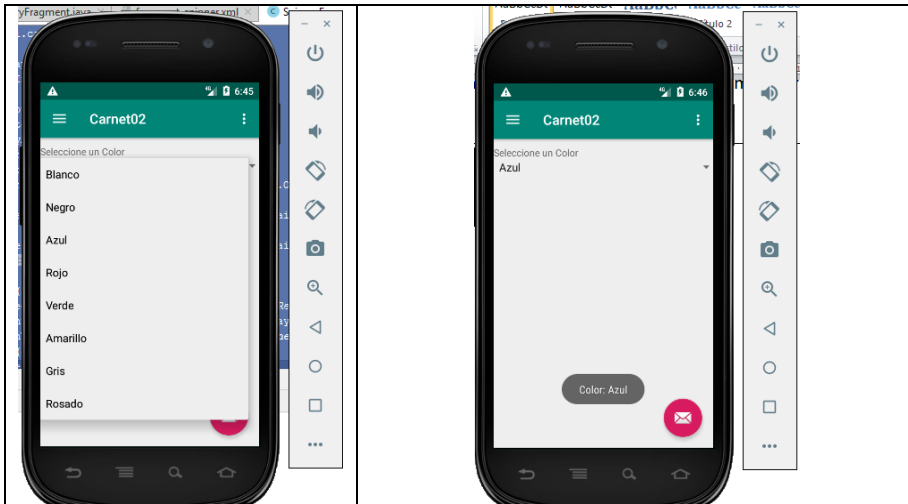
    public void onNothingSelected(AdapterView<?> arg0) {
        // No se Utiliza
    }
}
```

### Observaciones:

Para poder saber que elemento del *Spinner* fue seleccionado debemos auxiliarnos de la Clase *ArrayAdapter* que proporciona un modelo para manejar un arreglo arbitrario de objetos y así poder obtener los atributos o valores del elemento seleccionado.

Así mismo dentro de nuestra Activity es necesario implementar la interfaz *OnItemSelectedListener* que nos permite establecer las acciones a realizar una vez se seleccione un elemento del *Spinner*, en nuestro caso se desplegara un mensaje indicando el color seleccionado.

Y al ejecutar la aplicación y seleccionar la opción "Spinner" del menú, se mostrara de la siguiente manera:



Comprima su carpeta de proyecto y súbalo en el link respectivo (Guía de Laboratorio 2(Android))

## Tarea opcional

Realice a través de una Lista, cada una de las actividades que mandara a llamar a cada uno de los componentes vistos en la presente guía. Para cada uno de los cuales se tendrán las siguientes modificaciones:

- **EditText:** que la información colocada en el edittext aparezca en el Toast y se dé cuando se desactive el edittext.
- **Checkbox:** que al seleccionar cada checkbox se forme una cadena de caracteres.
- **RadioButton:** que haga un contador de cuantas veces se selecciona el radiobutton de un conjunto de tres.
- **TextView:** utilizarlo en la mayoría de las opciones para dar más vista al diseño de la interfaz.
- **Button:** crear tres botones y con la ayuda de los textview realizar diversas operaciones la primera será suma, la segunda resta y la tercera multiplicación.
- **Gallery:** crear su propia galería de fotos y asignar mensajes respecto a la foto seleccionada.



## Anexos

### Anexo A

#### Codigo fuente

```
package sv.edu.ues.fia.eisi.carnet02;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class ButtonFragment extends Fragment {

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                            Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_button, container,
false);
    }
}
```

#### Anexo B Layout de Button

```
<?xml version="1.0" encoding="utf-8" ?>
<?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"
    android:orientation="vertical"
    tools:context=".ButtonFragment">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:layout_width="277dp"
        android:layout_height="69dp"
        android:layout_marginTop="60dp"
        android:text="@string/TextView1_Btn"
        android:textSize="24sp"
        android:textStyle="bold"
        tools:textColor="@tools:sample/avatars" />
```



```
<Button
    android:id="@+id/button"
    android:layout_width="275dp"
    android:layout_height="73dp"
    android:text="@string/Button1"
    android:textColor="@color/colorAccent"
    android:textSize="16sp"
/>
</LinearLayout>
```

### Anexo C String.xml(final)

```
<resources>
    <string name="app_name">Carnet02</string>
    <string name="navigation_drawer_open">Open navigation drawer</string>
    <string name="navigation_drawer_close">Close navigation
drawer</string>
    <string name="nav_header_title">Android Studio</string>
    <string
name="nav_header_subtitle">android.studio@android.com</string>
    <string name="nav_header_desc">Navigation header</string>
    <string name="action_settings">Settings</string>

    <!-- TODO: Remove or change this placeholder text -->
    <string name="hello_blank_fragment">Hello blank fragment</string>
    <!-- Variables de boton -->
    <string name="TextView1_Btn">Texto en Mensaje</string>
    <string name="Button1">Presione Boton</string>
    <string name="text">2. TextView</string>
    <string name="mensajeayuda">3. EditText</string>

    <string name="titulochequeos">Listado de Checkbox</string>
    <string name="opcion1">Checkbox 1</string>
    <string name="opcion2">Checkbox 2</string>
    <string name="opcion3">Checkbox 3</string>
    <string name="text1">2</string>
    <string name="text2">4</string>
    <string name="text3">8</string>

    <string name="RadioButton1">opc 1</string>
    <string name="RadioButton2">opc 2</string>
    <string name="RadioButton3">opc 3</string>
    <string name="mensaje_error_radiobutton">Ningun RadioButton
seleccionado</string>
    <string name="colors_prompt">Seleccione un Color</string>
    <string-array name="colors_array">
        <item>Blanco</item>
        <item>Negro</item>
        <item>Azul</item>
        <item>Rojo</item>
        <item>Verde</item>
```



UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERIA Y ARQUITECTURA  
ESCUELA DE INGENIERIA DE SISTEMAS INFORMATICOS  
PROGRAMACION PARA DISPOSITIVOS MOVILES  
PDM115

Ciclo I-2019

```
<item>Amarillo</item>  
<item>Gris</item>  
<item>Rosado</item>  
</string-array>  
  
</resources>
```