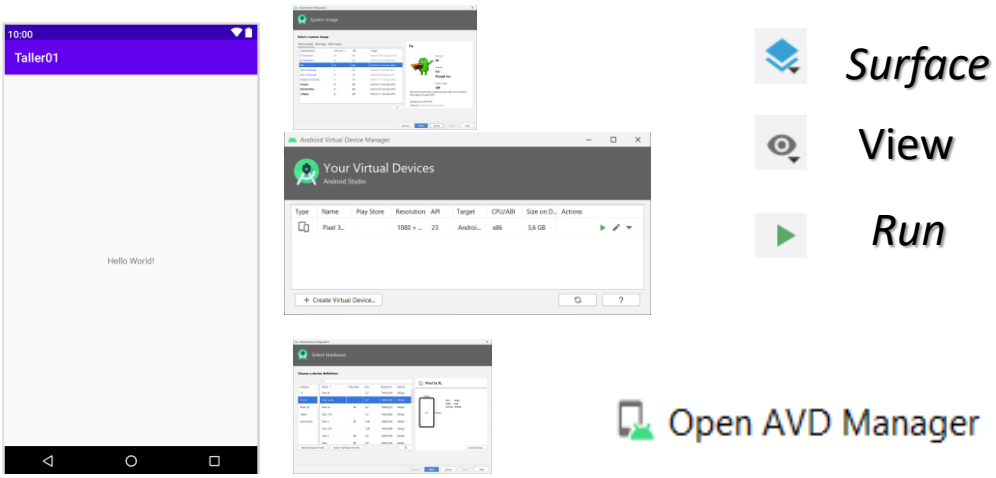
Taller 1: HOLA MUNDO

REQUERIMIENTOS

- ✓ Abrir Android Studio
- ✓ Crear un Proyecto "Empty Activity" (Lollipod 5.0)
- ✓ Explicar Propiedades/Atributos del Layout (Screen). (nombre, color, etc.).
- ✓ Conocer los componentes del Pallet
- ✓ Explicar los atributos generales de los componentes
- ✓ Manejo de Componentes sobre el Layout (RelativeLayout)
- ✓ Implementar un programa que muestre el mensaje en el TextView de: HOLA MUNDO
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
- ✓ Compilar y Revisar Errores.

Pantallas



Procedimiento

Metodología:
(Verificar nombre de Componentes utilizados sobre el Layout)

Codificación en MainActivity (app►java►com.example.____)

Declarar variables para cada componente del Layout (bajo public class MainActivity extends AppCompatActivity {)

```
public class MainActivity extends AppCompatActivity {

 //label
   Button btn1;
   TextView txtview1;
```

Inicializar los componentes visuales en la variables antes declaradas (bajo protected void onCreate(Bundle savedInstanceState) {)

```
@Override
 protected void onCreate(Bundle savedInstanceState) {
```

```
   super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_main);

     btn1 = (Button) findViewById(R.id.button1);
     txtview1 = (TextView) findViewById(R.id.textView);
```

Programación en el evento click del botón (bajo las línea del paso anterior)

```
     btn1.setOnClickListener(new View.OnClickListener() {
       @Override
        public void onClick(View v) {

       txtview1.setText("Hola Mundo");
          }
       });
```

IMPORTANTE:

Poder cambiar a minúsculas el texto en los botones
Ingresar activty_main.xml
Y en el componente dentro poner el siguiente código

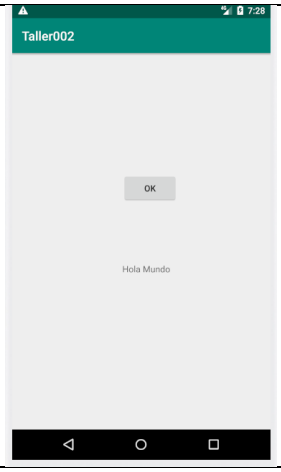
```
android:textAllCaps="false";
```

Taller 2: Saludo con Botón

REQUERIMIENTOS

- ✓ Abrir Android Studio
- ✓ Crear un Proyecto "Empty Activity" (Lollipod 5.0)
- ✓ Explicar Propiedades del Layout (Screen). (nombre, color, etc.).
- ✓ Conocer los componentes del Pallet
- ✓ Insertar los siguientes componentes:
- ✓ 1 TextView (Label)
- ✓ 1Button
- ✓ Explicar los atributos generales de los componentes
- ✓ Implementar un programa que muestre el mensaje en el TextView ,
- ✓ al presionar el botón Buttonn1.
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
- ✓ Compilar y Revisar Errores.

PANATLLAS



PROCEDIMIENTOS

PROPIEDADES DEL BUTTON MAS USADAS:

textColor = Color del texto dentro del Botón
backgroundTint = Color de relleno del Botón
rippleColor = Color del botón al presionar

Metodología:

- (Verificar nombre de Componentes utilizados sobre el Layout)
  Codificación en MainActivity (app►java►com.example.____)
- Declarar variables para cada componente del Layout (bajo **public class MainActivity extends AppCompatActivity {)**

```
public class MainActivity extends AppCompatActivity {
    //label
    Button btn1;
    TextView txtview1;
```

- Inicializar los componentes visuales en la variables antes declaradas (bajo **protected void onCreate(Bundle savedInstanceState) {**)

```
@Override
 protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
```

✓ **btn1** = (Button) findViewById(R.id.***button1***);
txtview1 = (TextView) findViewById(R.id.***textView***);

Metodo 1: Programar el evento click

- Programación en el evento click del botón (bajo las línea del paso anterior)

```
btn1.setOnClickListener(new View.OnClickListener() {
  @Override
   public void onClick(View v) {

  txtview1.setText("Hola Mundo");
    }
  });
```

Metodo 2: Crear una función para llamar desde el evento click

Antes de la ultima llave } va el codigo:

```
public void mensaje(View view)
{
   txtv1.setText("Hola Mundo, Bienvenido");
}
}//////esta es la llave final
```

****

En el evento **Onclick** del botón …llamamos a la función creada “mensaje”

IMPORTANTE:

Cambiar titulo del toolbar; ingrar al **AndroidManifest.xml y poner el siguiente codigo**

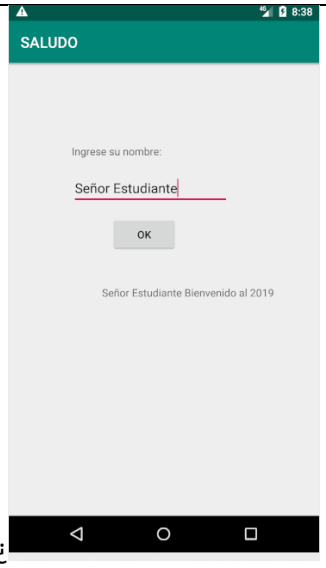
app:title="titulo toolbar"/>

Taller 3: Saludo Personalizado

REQUERIMIENTOS

- ✓ Crear un Proyecto "Empty Activity" (Lollipod 5.0)
- ✓ Insertar los siguientes componentes:
- ✓ 2 TextView(label)
- ✓ 1 PlainText (EditText = TextBox)
- ✓ 1Button
- ✓ ***Implementar un programa que muestre el mensaje de***
- ✓ ***Bienvenida en el TextView con el nombre ingresado***
- ✓ ***, al presionar el botón Buttonn1.***
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
- ✓ Compilar y Revisar Errores.

PANTALLAS



¿

PROCEDIMIENTOS

Metodología:

- (Verificar nombre de Componentes utilizados sobre el Layout - RelativeLayout)

Codificación en MainActivity (app►java►com.example.____)

- Declarar variables para cada componente del Layout (bajo **public class MainActivity extends AppCompatActivity {**)

public class MainActivity **extends** AppCompatActivity {

✓ *//variables*

TextView **lbl1**;
TextView **lbl2**;
Button **btn1**;
EditText **txt1**;

- Inicializar los componentes visuales en la variables antes declaradas (bajo **protected void onCreate(Bundle savedInstanceState) {**)

```
@Override
 protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
```

- ✓ *//inicializamos variables(componentes)*

```
lbl1 = (TextView)findViewById(R.id.textView1);
lbl2 = (TextView)findViewById(R.id.textView2);
btn1 = (Button)findViewById(R.id.button1);
txt1 = (EditText)findViewById(R.id.editText1);
```

- Programación en el evento click del botón (bajo las línea del paso anterior)

```
btn1.setOnClickListener(new View.OnClickListener() {
 @Override
  public void onClick(View v) {

      lbl2.setText(txt1.getText()+" Bienvenido al 2019 ");
   }
  });
```

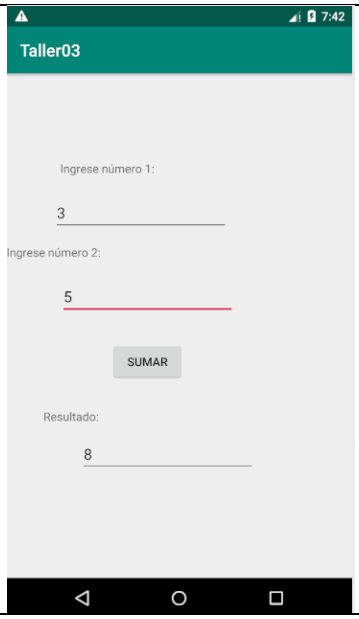
IMPORTANTE:
Cambiar titulo del toolbar; ingrar al **AndroidManifest.xml y poner el siguiente codigo**
app:title="titulo toolbar"/>

Taller 4: Manejo de Números/Operaciones

REQUERIMIENTOS

- ✓ Crear un Proyecto “Empty Activity” (Lollipod 5.0)
- ✓ Insertar los siguientes componentes:
  - 3 TextView(label)
  - 3 EditText (TextBox)
  - 1Button
- ✓ Implementar un programa que me permita ingresar 2 números y al presionar el botón se muestre el resultado en un EditText
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
- ✓ Compilar y Revisar Errores.

PANTALLAS



REQUERIMEINTOS

Metodología:
(Verificar nombre de Componentes utilizados sobre el Layout - RelativeLayout)

Codificación en MainActivity (app►java►com.example.____)

Declarar variables para cada componente del Layout (bajo public class MainActivity extends AppCompatActivity {)

```
public class MainActivity extends AppCompatActivity {

 //variables
//variables
Button btn1;
EditText txt1,txt2,txt3;
```

Inicializar los componentes visuales en la variables antes declaradas (bajo protected void onCreate(Bundle savedInstanceState) {)

```
   @Override
    protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_main);

//inicializamos variables(componentes)


btn1 = (Button)findViewById(R.id.button1);
txt1 = (EditText)findViewById(R.id.editText1);
txt2 = (EditText)findViewById(R.id.editText2);
txt3 = (EditText)findViewById(R.id.editText3);
```

Método 1: Crear el evento Onclik

Programación en el evento click del botón (bajo las línea del paso anterior)

```
     btn1.setOnClickListener(new View.OnClickListener() {
       @Override
        public void onClick(View v) {


        int n1 = Integer.valueOf(txt1.getText().toString());
        int n2 = Integer.valueOf(txt2.getText().toString());
        int resultado = n1 + n2;
        txt3.setText(""+resultado);

     otro metododo
      float num1 = Float.parseFloat(txt1.getText().toString());
         float num2 =  Float.parseFloat(txt2.getText().toString());




          }
       });
```

Método 2: Crear una función y llamar desde el evento Onclik
(antes de la ultima llave)

```
public void suma(View view)
{
  int a, b, c;
  a=  Integer.valueOf(txt1.getText().toString());
  b=  Integer.valueOf(txt2.getText().toString());

  c=a+b;
  Lblrespuesta.setText("Resultado: "+c);
```

```
    //deshabilitar plain text
      txt1.setEnabled(false);
      txt2.setEnabled(false);


}
```

IMPORTANTE:

IMPORTANTE:

Cambiar titulo del toolbar; ingrar al AndroidManifest.xml y poner el siguiente código:
app:title="titulo toolbar"/>

Deshabilitar plaintex (texbox)

```
   txt1.setEnabled(false);
   txt2.setEnabled(false);
```

Propiedad: digits
Permite controlar el ingreso de datos 0123456789.

Para activar el focus de un campo:

```
txt1. requestFocus();   //el campo que quiera
```

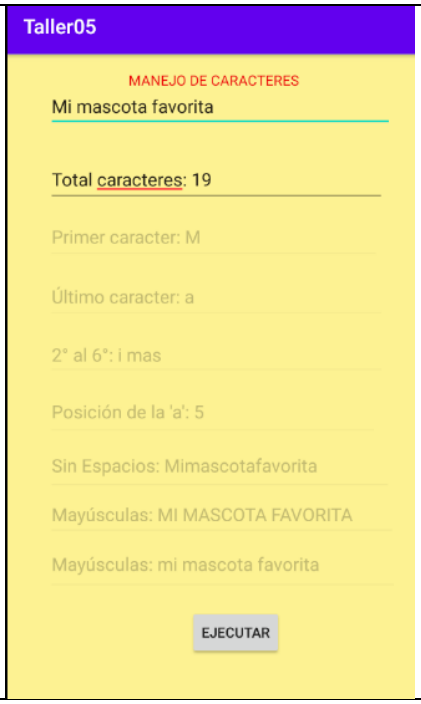
http://www.hermosaprogramacion.com/2016/03/edittext-android/
DarkActionBar

Taller 5: Manejo de Caracteres/Operaciones

REQUERIMIENTOS

- ✓ Crear un Proyecto "Empty Activity" (Lollipod 5.0)
- ✓ Insertar los siguientes componentes:
  - 9 (TextView(label)
  - 1 EditText (TextBox)
  - 1 Button
- ✓ ***Implementar un programa que muestre el uso de las***
- ✓ ***funciones para texto usadas para el manejo de texto***
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
- ✓ Compilar y Revisar Errores.
- ✓

PANTALLAS



PROCEDIMIENTO

Metodología:

- (Verificar nombre de Componentes utilizados sobre el Layout - RelativeLayout)

Codificación en MainActivity (app►java►com.example.____)

- Declarar variables para cada componente del Layout (bajo **public class MainActivity extends AppCompatActivity {**)

public class MainActivity **extends** AppCompatActivity {

✓ *//variables*

//variables

Button **btn1**;
EditText **txt1**,**txt2**,**txt3**,**txt4**,**txt5**,**txt6**,**txt7**,**txt8**,**txt9**;

- ✓ Inicializar los componentes visuales en la variables antes declaradas (bajo **protected void onCreate(Bundle savedInstanceState) {**)

```
@Override
 protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
```

- ✓ *//inicializamos variables(componentes)*

```
txt1 = (EditText) findViewById(R.id.editText);
txt2 = (EditText) findViewById(R.id.editText2);
txt3 = (EditText) findViewById(R.id.editText3);
txt4 = (EditText) findViewById(R.id.editText4);
txt5 = (EditText) findViewById(R.id.editText5);
txt6 = (EditText) findViewById(R.id.editText6);
txt7 = (EditText) findViewById(R.id.editText7);
txt8 = (EditText) findViewById(R.id.editText8);
txt9 = (EditText) findViewById(R.id.editText9);


btn1 = (Button) findViewById(R.id.button);
```

- ✓ Programación en el evento click del botón (bajo las línea del paso anterior)

```
btn1.setOnClickListener(new View.OnClickListener()
{
  public void onClick(View v)
  {

    //captura la frase en 1 variable
String cadena;
cadena = txt1.getText().toString();

//contar caracteres
int total = cadena.length();
txt2.setText(""+total);

//Primero
txt3.setText(""+cadena.substring(0,1));


//ultimo
txt4.setText(""+cadena.substring((total-1),total));

//2 al 6
txt5.setText(""+cadena.substring(1,6));

//busca "a"
int pos = cadena.indexOf("a", 0);
txt6.setText(""+(pos+1));
```

```
//quita espacio
String nuevacadena = cadena.replace(" ","");
txt7.setText(nuevacadena.toString());

//mayu
txt8.setText(cadena.toUpperCase());

//minus
txt9.setText(cadena.toLowerCase());

        }
      });
```

IMPORTANTE:
Para deshabilitar los plain tex

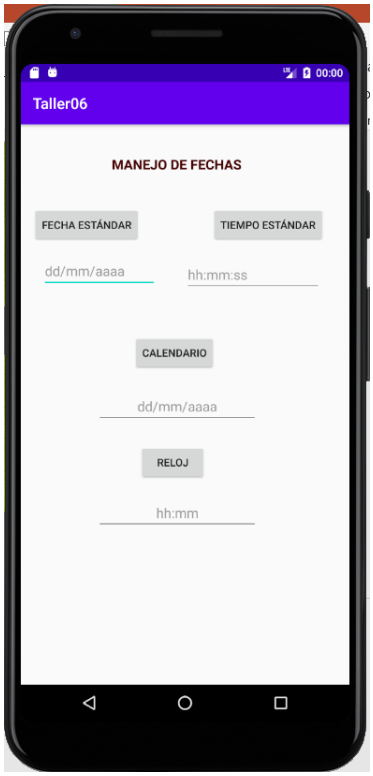
```
txt2.setEnabled(false);
txt3.setEnabled(false);
txt4.setEnabled(false);
txt5.setEnabled(false);
txt6.setEnabled(false);
txt7.setEnabled(false);
txt8.setEnabled(false);
txt9.setEnabled(false);
```

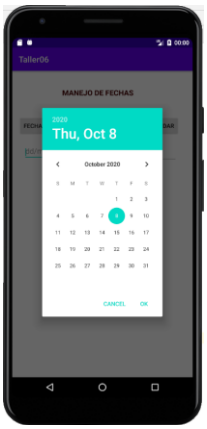
Taller 6: App Manejo de Fechas

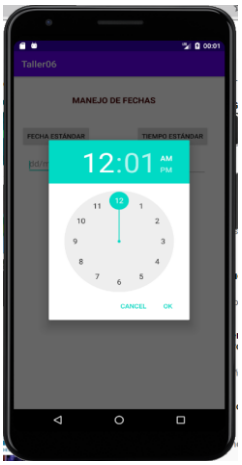
REQUERIMIENTOS

- ✓ Crear un Proyecto “Empty Activity” (Lollipod 5.0)
- ✓ Insertar los siguientes componentes:
  - 1 texview (Label)
  - 4 (buttons)
  - 4 EditText (plain text)
- ✓ Implementar un programa que permita conocer el uso y captura de fechas/horas
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
- ✓ Compilar y Revisar Errores.

PANTALLAS







PROCEDIMIENTO

Implementacion con los eventos **setOnClickListener(new View setOnClickListener)……**

Boton carga fecha estandar

```
btn1.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {

      txt1.setText( String.valueOf(android.text.format.DateFormat.format("dd/MM/yyyy",
new java.util.Date())));
```

```
    //Toast.makeText(getApplicationContext(),"Hola", Toast.LENGTH_SHORT).show();
  }
});
```

Boton carga tiempo estándar

```
btn2.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
    txt2.setText( String.valueOf(android.text.format.DateFormat.format("hh:mm:ss", new
java.util.Date())));
  }
});
```

Boton llama a Calendario

```
btn3.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {

    Calendar cal = Calendar.getInstance();
    int anio = cal.get(Calendar.YEAR);
    int mes = cal.get(Calendar.MONTH);
    int dia = cal.get(Calendar.DAY_OF_MONTH);

    DatePickerDialog dpd = new DatePickerDialog(MainActivity.this, new
DatePickerDialog.OnDateSetListener() {
      @Override
      public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {

        month=month+1;
        String fecha= dayOfMonth + "/"+ month +"/"+ year;
        txt3.setText(fecha);

      }
    },anio,mes,dia);

    dpd.show();
  }
});
```

Botón llama reloj

```
btn4.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {

    Calendar calc = Calendar.getInstance();
    int hora = calc.get(Calendar.HOUR_OF_DAY);
    int minuto = calc.get(Calendar.MINUTE);
```

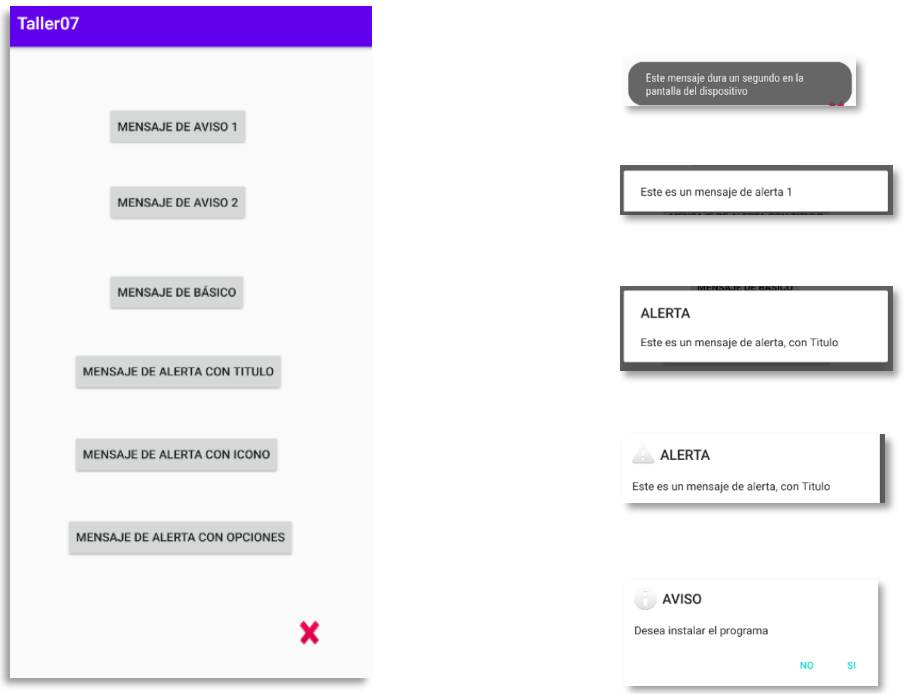
```
      TimePickerDialog tpd = new TimePickerDialog(MainActivity.this, new
TimePickerDialog.OnTimeSetListener() {
          @Override
          public void onTimeSet(TimePicker view, int hourOfDay, int minute) {

              txt4.setText(hourOfDay + ":" + minute);
          }
      },hora, minuto,false);

      tpd.show();
   }
});
```

Taller 7: Mensajes y Notificaciones

REQUERIMIENTOS

- ✓ Crear un Proyecto “Empty Activity” (Lollipod 5.0)
- ✓ Insertar los siguientes componentes:
- ✓ 6 Button
- ✓ ***Implementar un programa que permita conocer***
- ✓ ***el uso de avisos y mensajes de alertas.***
- ✓ ***ReaLIzar un ejemplo del usos de mensajes de alerta X***
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
- ✓ Compilar y Revisar Errores.
- ✓

PANTALLAS



PROCEDIMIENTO

Implementacion con los eventos **setOnClickListener(new View setOnClickListener)……**

Declarar variables propias

Button btn1,btn2,btn3,btn4,btn5,btn6,btn7;

Asociar variables a componentes gráficos

```
@Override
protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_main);

   btn1 = (Button) findViewById(R.id.button);
   btn2 = (Button) findViewById(R.id.button2);
   btn3 = (Button) findViewById(R.id.button3);
   btn4 = (Button) findViewById(R.id.button4);
   btn5 = (Button) findViewById(R.id.button5);
   btn6 = (Button) findViewById(R.id.button6);
   btn7 = (Button) findViewById(R.id.button7);
```

Poner el código para cada botón antes de las 2 llaves }}
Usar eventyos propios para cada Boton setOnListener

BOTON 1

```
btn1.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      Toast.makeText(getApplicationContext(),"Este mensaje dura medio segundo en la
pantalla del dispositivo", Toast.LENGTH_SHORT).show();
   }
});
```

BOTON 2

```
btn2.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      Toast.makeText(getApplicationContext(),"Este mensaje dura un segundo en la
pantalla del dispositivo", Toast.LENGTH_SHORT).show();
   }
});
```

BOTON 3

```
btn3.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {

      AlertDialog.Builder alerta = new AlertDialog.Builder(MainActivity.this);

      alerta.setMessage("Este es un mensaje de alerta 1");
      alerta.show();
```

```
    }
});
```

BOTON 4

```
btn4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        AlertDialog.Builder alerta = new AlertDialog.Builder(MainActivity.this);

        alerta.setTitle("ALERTA");
        alerta.setMessage("Este es un mensaje de alerta, con Titulo");
        alerta.show();

    }
});
```

BOTON 5

```
 btn5.setOnClickListener(new View.OnClickListener() {
     @Override
     public void onClick(View v) {

         AlertDialog.Builder alerta = new AlertDialog.Builder(MainActivity.this);

         alerta.setTitle("ALERTA");
         alerta.setMessage("Este es un mensaje de alerta, con Titulo")
                 .setIcon(android.R.drawable.ic_dialog_alert);
         alerta.show();

     }
 });
```

BOTON 6

```
 btn6.setOnClickListener(new View.OnClickListener() {
     @Override
     public void onClick(View v) {


         AlertDialog.Builder alerta = new AlertDialog.Builder(MainActivity.this);

         alerta.setTitle("AVISO");
         alerta.setMessage("Desea instalar el programa")
                 .setIcon(android.R.drawable.ic_dialog_info)

                 .setCancelable(false)
```

```
            .setPositiveButton("SI", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {

                }
            })
            .setNegativeButton("NO", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {

                }
            });
        alerta.show();

    }
});
```

BOTON 7 CERRAR X

```
btn7.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        AlertDialog.Builder alerta = new AlertDialog.Builder(MainActivity.this);

        alerta.setTitle("AVISO");
        alerta.setMessage("Desea salir de la aplicación")
                .setIcon(android.R.drawable.ic_dialog_alert)

                .setCancelable(false)
                .setPositiveButton("SI", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {


                        System.exit(0);


                    }
                })
                .setNegativeButton("NO", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {

                    }
                });
        alerta.show();
```

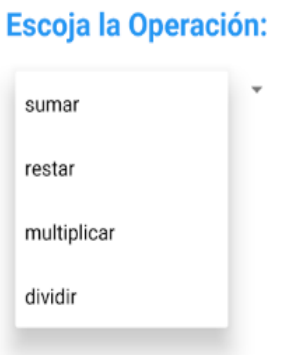
```
    }
});


////notificación snack
Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
        .setAction("Action", null).show();
```

Taller 8: Estructuras de Control Condicionales "IF" ***(MANEJO DE SPINNER)***

REQUERIMIENTOS

- ✓ Crear un Proyecto "Empty Activity" (Lollipod 5.0)
- ✓ Insertar los siguientes componentes:
  - 5 TextView (labels)
  - 2 plainText (edit tex)
  - 1 Spinner (grupo Container)
  - 2 Buttons
- ✓ ***Implementar un programa que permita realizar las operaciones indicadas, seleccionada desde un desplegable (spinner).***
- ✓ ***Explicar y conocer el uso del componente spinner***
- ✓ ***Hacer énfasis en las estructuras de control-condiconal***
- ✓ ***Usar avisos o mensajes de alerta para controlar errores.***
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
- ✓ Compilar y Revisar Errores.

PANTALLAS



PROCEDIMIENTO

Declarar variables propias

```
Button btn1,btn2, btnsalir;
TextView lbl1, lbl2 ,lbl3 ,lbl4 , lblresultado;
EditText txt1, txt2;
Spinner spn1; @Override
```

Asociar variables a componentes gráficos

```
//titulo
  getSupportActionBar().setTitle("ESTRUCTURA DE CONTROL - IF");

  //LOGICA VS GRAFICA

  btn1=(Button)findViewById(R.id.button);
  btn2=(Button)findViewById(R.id.button2);
  btnsalir =(Button)findViewById(R.id.button3);

  lbl1 =(TextView)findViewById(R.id.textView);
  lbl2 =(TextView)findViewById(R.id.textView2);
  lbl3 =(TextView)findViewById(R.id.textView3);
  lbl4 =(TextView)findViewById(R.id.textView4);
  lblresultado =(TextView)findViewById(R.id.textView5);

  txt1=(EditText)findViewById(R.id.editTextNumber);
  txt2=(EditText)findViewById(R.id.editTextNumber2);

  spn1 = (Spinner)findViewById(R.id.spinner);


  //cargar datos en el spinner
  String []  operaciones = {"sumar", "restar", "multiplicar", "dividir"};
 ArrayAdapter <String> adapter = new
ArrayAdapter<String>(MainActivity.this,R.layout.support_simple_spinner_dropdown_item,operaciones);
  spn1.setAdapter(adapter);
```

Implementacion con los eventos **setOnClickListener(new View setOnClickListener)……**

BOTON 1

```
btn1.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
```

```
float x1,x2,resul;

if( (txt1.getText().toString().equals(""))  || (txt2.getText().toString().equals("")))
{
   Toast.makeText(MainActivity.this,"Existen campos vacios!!",Toast.LENGTH_SHORT).show();
   lblresultado.setText("Resultado:");
}
else
{
   x1 = Float.valueOf(txt1.getText().toString());
   x2 = Float.valueOf(txt2.getText().toString());


   //operacion suma
   if(spn1.getSelectedItem().toString().equals("sumar"))
   {
      resul = x1+x2;
      lblresultado.setText("Resultado: "+resul);

      //deshabilito
      txt1.setEnabled(false);
      txt2.setEnabled(false);
      btn1.setEnabled(false);
      btn2.setEnabled(true);

   }


   //operacion resta
   if(spn1.getSelectedItem().toString().equals("restar"))
   {
      resul = x1-x2;
      lblresultado.setText("Resultado: "+resul);

      //deshabilito
      txt1.setEnabled(false);
      txt2.setEnabled(false);
      btn1.setEnabled(false);
      btn2.setEnabled(true);

   }

   //operacion multiplicacion
   if(spn1.getSelectedItem().toString().equals("multiplicar"))
   {
      resul = x1*x2;
      lblresultado.setText("Resultado: "+resul);

      //deshabilito
      txt1.setEnabled(false);
      txt2.setEnabled(false);
      btn1.setEnabled(false);
      btn2.setEnabled(true);

   }

   //operacion division
   if(spn1.getSelectedItem().toString().equals("dividir"))
   {

     if(txt2.getText().toString().equals("0"))
     {
        Toast.makeText(MainActivity.this,"Imposible operar con divisor 0",Toast.LENGTH_SHORT).show();
```

```
                lblresultado.setText("");
            }
            else
            {
                resul = x1/x2;
                lblresultado.setText("Resultado: "+resul);

                //deshabilito
                txt1.setEnabled(false);
                txt2.setEnabled(false);
                btn1.setEnabled(false);
                btn2.setEnabled(true);


            }

          }

        }

      }
    });
```

BOTON 2 – limpiar

```
btn2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            txt1.setText("");
            txt2.setText("");
            lblresultado.setText("Resultado:");

            txt1.setEnabled(true);
            txt1.requestFocus();
            txt1.setEnabled(true);
            btn1.setEnabled(true);

            btn2.setEnabled(false);


        }
    });
```

BOTON 2 – SALIR

```
//boton salir
    btnsalir.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
```

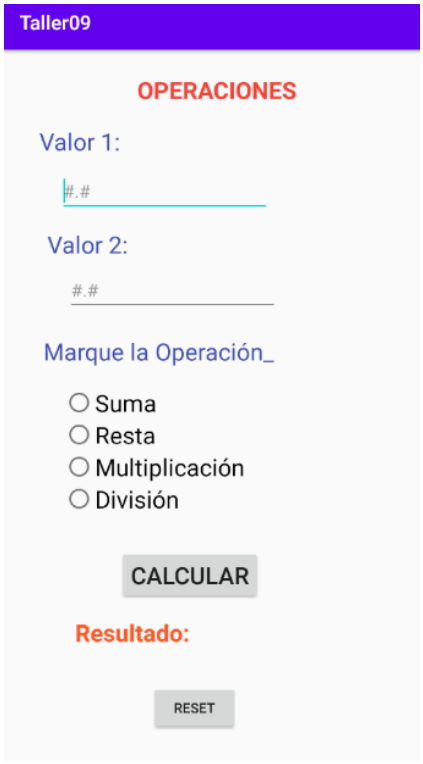
```
        AlertDialog.Builder alerta  = new AlertDialog.Builder(MainActivity.this);
        alerta.setTitle("Aviso");
        alerta.setMessage("Desea Cerrar la Aplicación");
        alerta.setIcon(android.R.drawable.ic_dialog_alert);
        alerta.setCancelable(false);
        alerta.setPositiveButton("Si", new DialogInterface.OnClickListener() {
           @Override
           public void onClick(DialogInterface dialog, int which) {
              System.exit(0);
           }
        });

        alerta.setNegativeButton("No", new DialogInterface.OnClickListener() {
           @Override
           public void onClick(DialogInterface dialog, int which) {

           }
        });

       alerta.show();



    }
});
```

Taller 9: Estructructiras de Control Alternativas "swich"***(manejo de radiobuttons)***

REQUERIMIENTOS

- ✓ Crear un Proyecto "Empty Activity" (Lollipod 5.0)
- ✓ Insertar los siguientes componentes:
  - 5 TextView (labels)
  - 2 plainText (edit tex)
  - 1 GroupRadio
  - 4 Radio Buttons
  - 2 Buttons
- ✓ ***Implementar un programa que permita realizar las operaciones indicadas, seleccionada desde los RadioButtons.***
- ✓ ***Explicar y conocer el uso del componente RadioButtons***
- ✓ ***Hacer énfasis en las estructuras de control-alternativas***
- ✓ ***Usar avisos o mensajes de alerta para controlar errores.***
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
- ✓ Compilar y Revisar Errores.

PANTALLAS



PROCEDIMIENTO

IMPORTANTE

//poner los componentes indicados
Agregar primero el RadioGroup y dentro los RadioButton
Al RadioGroup hay q ponerle un ID manual

Declarar variables propias

```
Button btn1,btn2;
EditText txt1,txt2;
TextView lbl1,lbl2,lbl3,lbl4,lblrespuesta;
RadioButton rbtn1, rbtn2, rbtn3, rbtn4;
RadioGroup rbng1;
int op;

@Override
```

Asociar variables a componentes gráficos

```
    //titulo
      getSupportActionBar().setTitle("ESTRUCTURA DE CONTROL - IF");

     //LOGICA VS GRAFICA

  btn1 = (Button) findViewById(R.id.button);
  btn2 = (Button) findViewById(R.id.button2);

  txt1 = (EditText) findViewById(R.id.editTextTextPersonName);
  txt2 = (EditText) findViewById(R.id.editTextTextPersonName2);

  lbl1= (TextView) findViewById(R.id. textView);
  lbl2= (TextView) findViewById(R.id. textView2);
  lbl3= (TextView) findViewById(R.id. textView5);
  lbl4= (TextView) findViewById(R.id. textView3);
  lblrespuesta = (TextView) findViewById(R.id. textView4);

  rbtn1 = (RadioButton) findViewById(R.id. radioButton);
  rbtn2 = (RadioButton) findViewById(R.id. radioButton2);
  rbtn3 = (RadioButton) findViewById(R.id. radioButton3);
  rbtn4 = (RadioButton) findViewById(R.id. radioButton4);

  rbng1 = (RadioGroup) findViewById(R.id.rg);
```

```
   txt1.requestFocus();
   btn2.setEnabled(false);
  rbng1 = (RadioGroup) findViewById(R.id.rg);

//no olvidar  los inicios por defecto de algunos componentes
```

Implementacion con los eventos **setOnClickListener(new View setOnClickListener)……**

CLICK RadioButton *********Capturar el click en los radioButons

```
rbtn1.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      op=1;

     // Toast.makeText(getApplicationContext(),"Op ="+ op, Toast.LENGTH_SHORT).show();
   }
});


rbtn2.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      op=2;

      //Toast.makeText(getApplicationContext(),"Op ="+ op, Toast.LENGTH_SHORT).show();
   }
});



rbtn3.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      op=3;

      //Toast.makeText(getApplicationContext(),"Op ="+ op, Toast.LENGTH_SHORT).show();
   }
});




rbtn4.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      op=4;

      //Toast.makeText(getApplicationContext(),"Op ="+ op, Toast.LENGTH_SHORT).show();
   }
```

```
});
```

BOTON 1

```
btn1.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {



     if(txt1.getText().toString().isEmpty()   ||  txt2.getText().toString().isEmpty())
      {
         Toast.makeText(getApplicationContext(), "Existen Campos Vacios", Toast.LENGTH_SHORT).show();
      }
      else {

        float num1 = Float.parseFloat(txt1.getText().toString());
        float num2 =  Float.parseFloat(txt2.getText().toString());
         switch (op) {
            case 1: {

               float r = num1+num2;
               lblrespuesta.setText("Resultado: "+ r);

               txt1.setEnabled(false);
               txt2.setEnabled(false);
               btn1.setEnabled(false);
               btn2.setEnabled(true);
               rbtn1.setEnabled(false);
               rbtn2.setEnabled(false);
               rbtn3.setEnabled(false);
               rbtn4.setEnabled(false);

               break;

            }

            case 2: {

               float respueta= num1-num2;
               lblrespuesta.setText("Resultado: "+ respueta);

               txt1.setEnabled(false);
               txt2.setEnabled(false);
               btn1.setEnabled(false);
               btn2.setEnabled(true);
               rbtn1.setEnabled(false);
               rbtn2.setEnabled(false);
               rbtn3.setEnabled(false);
               rbtn4.setEnabled(false);

               break;
            }

            case 3: {

               float respueta=num1*num2;
               lblrespuesta.setText("Resultado: "+ respueta);
```

```
            txt1.setEnabled(false);
            txt2.setEnabled(false);
            btn1.setEnabled(false);
            btn2.setEnabled(true);
            rbtn1.setEnabled(false);
            rbtn2.setEnabled(false);
            rbtn3.setEnabled(false);
            rbtn4.setEnabled(false);

            break;
          }

          case 4: {

            if(  Float.parseFloat(txt2.getText().toString()) == 0.0 )
            {
              Toast.makeText(getApplicationContext(), "División para cero", Toast.LENGTH_SHORT).show();
              lblrespuesta.setText("Resultado: ");
            }
            else {

              float respueta = num1 / num2;
              lblrespuesta.setText("Resultado: " + respueta);
              txt1.setEnabled(false);
              txt2.setEnabled(false);
              btn1.setEnabled(false);
              btn2.setEnabled(true);
              rbtn1.setEnabled(false);
              rbtn2.setEnabled(false);
              rbtn3.setEnabled(false);
              rbtn4.setEnabled(false);


            }
            break;
          }

          default: {
            Toast.makeText(getApplicationContext(), "Escoja la Operación", Toast.LENGTH_SHORT).show();
            break;
          }
        }
      }

  }
});
```

BOTON 2 – limpiar

```
btn2.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {

    txt1.setText("");
    txt2.setText("");
    lblrespuesta.setText("Resultado:");
```

```
        txt1.setEnabled(true);
        txt1.requestFocus();
        txt2.setEnabled(true);


        btn1.setEnabled(true);
        btn2.setEnabled(false);
        rbtn1.setEnabled(true);
        rbtn2.setEnabled(true);
        rbtn3.setEnabled(true);
        rbtn4.setEnabled(true);

        rbtn1.setChecked(false);
        rbtn2.setChecked(false);
        rbtn3.setChecked(false);
        rbtn4.setChecked(false);

        op=0;


    }
});
```

BOTON 2 – SALIR

```
//boton salir
        btnsalir.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                AlertDialog.Builder alerta  = new AlertDialog.Builder(MainActivity.this);
                alerta.setTitle("Aviso");
                alerta.setMessage("Desea Cerrar la Aplicación");
                alerta.setIcon(android.R.drawable.ic_dialog_alert);
                alerta.setCancelable(false);
                alerta.setPositiveButton("Si", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        System.exit(0);
                    }
                });

                alerta.setNegativeButton("No", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {

                    }
                });
```

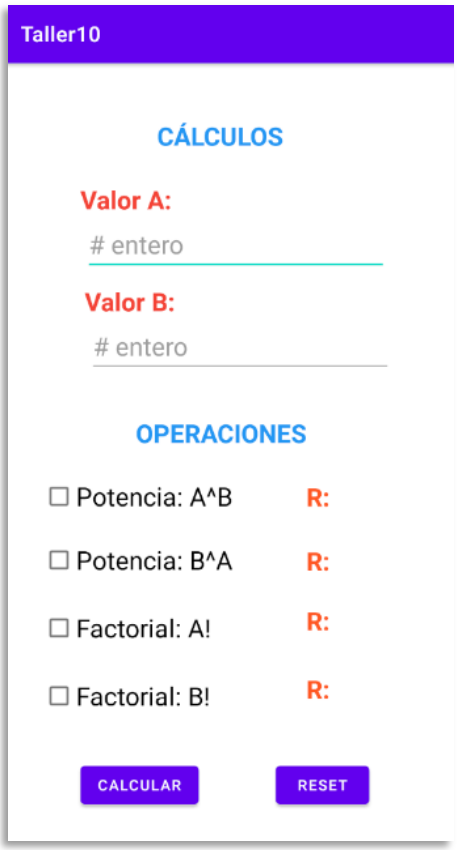
```
        alerta.show();



    }
});
```

Taller 10: Estructutras de Control Repetitivas “for, while, do while” ***(manejo de checkbox)***

REQUERIMIENTOS

- ✓ Crear un Proyecto “Empty Activity” (Lollipod 5.0)
- ✓ Insertar los siguientes componentes:
  - 8 TextView (labels)
  - 2 plainText (edit tex)
  - 4 CheckBox
  - 2 Buttons
- ✓ ***Implementar un programa que permita realizar las operaciones indicadas, seleccionada desde los CheckBox.***
- ✓ ***Explicar y conocer el uso del componente CheckBox***
- ✓ ***Hacer énfasis en las estructuras de control-repetitivas***
- ✓ ***Usar avisos o mensajes de alerta para controlar errores.***
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
- ✓ Compilar y Revisar Errores.

PANTALLAS



PROCEDIMIENTO

IMPORTANTE

Declarar variables propias

******//Crear variables propias

```
TextView lbl1,lbl2,lbl3,lbl4, lblresp1, lblresp2, lblresp3, lblresp4;
Button btn1, btn2;
EditText txt1, txt2;
CheckBox chb1, chb2, chb3, chb4;
@Override
```

Asociar variables a componentes gráficos

```
    //titulo
      getSupportActionBar().setTitle("ESTRUCTURA DE CONTROL - IF");

      //LOGICA VS GRAFICA

   lbl1 = (TextView) findViewById(R.id.textView);
   lbl2 = (TextView) findViewById(R.id.textView2);
   lbl3 = (TextView) findViewById(R.id.textView3);
   lbl4 = (TextView) findViewById(R.id.textView4);
   lblresp1 = (TextView) findViewById(R.id.textView5);
   lblresp2 = (TextView) findViewById(R.id.textView6);
   lblresp3 = (TextView) findViewById(R.id.textView7);
   lblresp4 = (TextView) findViewById(R.id.textView8);

   btn1 = (Button) findViewById(R.id.button);
   btn2 = (Button) findViewById(R.id.button2);

   txt1 = (EditText) findViewById(R.id.editTextTextPersonName);
   txt2 = (EditText) findViewById(R.id.editTextTextPersonName2);

   chb1 = (CheckBox) findViewById(R.id.checkBox);
   chb2 = (CheckBox) findViewById(R.id.checkBox2);
   chb3 = (CheckBox) findViewById(R.id.checkBox3);
   chb4 = (CheckBox) findViewById(R.id.checkBox4);

//no olvidar  los inicios por defecto de algunos componentes
```

Implementacion con los eventos **setOnClickListener(new View setOnClickListener)……**

BOTON 1

```
btn1.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {

      if (txt1.getText().toString().isEmpty() || txt2.getText().toString().isEmpty()) {
         Toast.makeText(getApplicationContext(), "Existen Campos Vacios",
Toast.LENGTH_SHORT).show();
      }
      else
      {


      if( chb1.isChecked())
      {
         ///// Exponente A:B
        int base,exponente;
        base = Integer.parseInt(txt1.getText().toString());
        exponente = Integer.parseInt(txt2.getText().toString());

        int acum1=1;
        for(int i=1; i<=exponente; i++)
        {
           acum1=acum1*base;
        }
        lblresp1.setText("R: "+acum1);


      }


      if( chb2.isChecked())
      {
         ///// Exponente B:A
         int base2,exponente2;
         base2 = Integer.parseInt(txt1.getText().toString());
         exponente2 = Integer.parseInt(txt2.getText().toString());

         int acum2=1;
         for(int i=1; i<=base2; i++)
         {
            acum2=acum2*exponente2;
         }
         lblresp2.setText("R: "+acum2);


      }
```

```
    if( chb3.isChecked())
    {
       ///// Factorial A
       int numero;
       numero = Integer.parseInt(txt1.getText().toString());

       int cont=1;
       int acum3=1;
      while(cont<= numero)
      {
         acum3 = acum3*cont;
         cont++;
      }
       lblresp3.setText("R: "+acum3);


    }


    if( chb4.isChecked())
    {
        //// Factorial b

       int numero2;
       numero2 = Integer.parseInt(txt2.getText().toString());

       int cont=1;
       int acum4=1;
       do
       {
          acum4 = acum4*cont;
          cont++;
       }
       while(cont<= numero2);

       lblresp4.setText("R: "+acum4);


    }


    if (chb1.isChecked()== false && chb2.isChecked()== false && chb3.isChecked()== false
&& chb4.isChecked()== false )
    {
       Toast.makeText(getApplicationContext(), "Marque Operación a realizar",
Toast.LENGTH_SHORT).show();
    }
```

```
      }//fin primer if

   }
   });
```

BOTON 2 – limpiar

```
btn2.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {

      txt1.setText("");
      txt2.setText("");
      txt1.requestFocus();

      lblresp1.setText("R:");
      lblresp2.setText("R:");
      lblresp3.setText("R:");
      lblresp4.setText("R:");


      chb1.setChecked(false);
      chb2.setChecked(false);
      chb3.setChecked(false);
      chb4.setChecked(false);
   }
});
```

BOTON 2 – SALIR

```
//boton salir
      btnsalir.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {

            AlertDialog.Builder alerta  = new AlertDialog.Builder(MainActivity.this);
            alerta.setTitle("Aviso");
            alerta.setMessage("Desea Cerrar la Aplicación");
            alerta.setIcon(android.R.drawable.ic_dialog_alert);
            alerta.setCancelable(false);
            alerta.setPositiveButton("Si", new DialogInterface.OnClickListener() {
               @Override
               public void onClick(DialogInterface dialog, int which) {
                  System.exit(0);
```

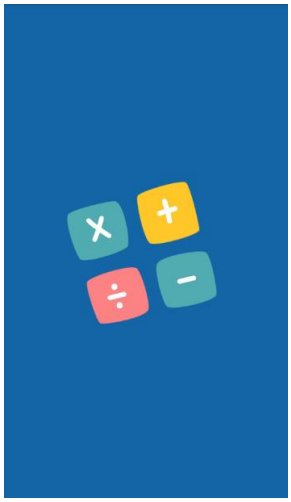
```
            }
        });

        alerta.setNegativeButton("No", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {

            }
        });

        alerta.show();



    }
});
```

Taller 11: Splash (Pantalla de Inicio)

REQUERIMIENTOS

- ✓ Crear un Proyecto “Empty Activity” (Lollipod 5.0)
- ✓ Crear +2 Activity(Layout)
- ✓ Insertar los siguientes componentes MainActivity1:
  - 1 ImageView
  - 1 TextView (labels)
- ✓ MainActivity2 :
  - 1 TextView (labels)
  - 1 Button Salir
- ✓ **Implementar un aplicación que permite mostrar un splash de bienvenida aproximadamente por 2 segundos y luego cargue la pantalla principal de la APP**
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
- ✓ Compilar y Revisar Errores.

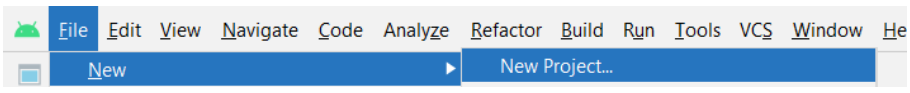
PANTALLAS

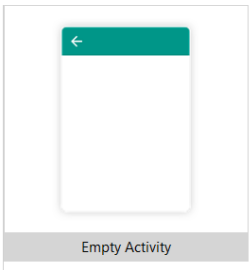


PROCEDIMIENTO

Crear el MainActivity2

File > New > Project





En MainAcitivty1

Declarar variables propias

Asociar variables a componentes gráficos

```
@Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main1);
----------
----------

 //funcion para quitar Action Bar
       getSupportActionBar().hide;

//Evento para el Timer

      new Handler().postDelayed(new Runnable() {
         @Override
         public void run() {

            Intent menu = new Intent(MainActivity4.this, MainActivity.class);
            startActivity(menu);
            finish();
         }
      },2000);
```

//////IMPORTENTE////////

En el caso de que el Layout splash no sea el primer Activity se debe proseguir de la siguiente manera:

1. Entrar al archivo ***AndroidManifest.xml***; buscar la activity con la que quiere arrancar la app (splash)
2. Cortar el siguiente código que le corresponde a la ***MainActivity1***

```
<intent-filter>
   <action android:name="android.intent.action.MAIN" />

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

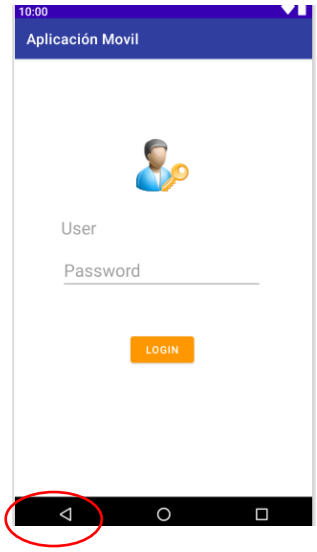
1. Y colocarlo dentro de la ***activity*** con la que desea inciar
2. <activity android:name=".MainActivity***4***"> ****aqui*** </activity>

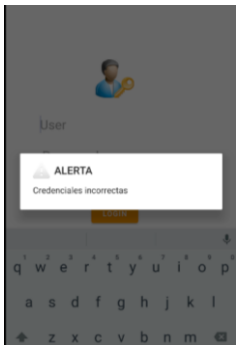
Taller 12: Autenticación (Login y Password)

REQUERIMIENTOS

- ✓ Crear un Proyecto "Empty Activity" (Lollipod 5.0)
- ✓ Insertar los siguientes componentes ***MainActivity***:
  - 1 ImageView (Widgets)
  - 2 plainText (edit tex)
  - 1 Buttons
- ✓ Crear ***MainActivity2 :***
  - 1 TextView (labels)
- ✓ ***Implementar una aplicación que permita validar el ingreso con sus credenciales antes de ir a la pantalla principal.***
- ✓ ***Explicar y conocer el propiedades de Barra de Acción***
- ✓ ***Usar avisos o mensajes de alerta para controlar errores.***
- ✓ ***Progamar el evento atrás de la barra de Navegación ◄***
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
- ✓ Compilar y Revisar Errores.
- ✓

PANTALLAS





PROCEDIMIENTO

MainActivity1 (Login)

Declarar variables propias

```
Button btn1;
EditText txt1;
EditText txt2;
@Override
```

Asociar variables a componentes gráficos

```
protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main2);

//ACCIONES PARA EL ACTION BAR
getSupportActionBar().setTitle(" ACCESO");   //titulo en la barra de acción
getSupportActionBar().setDisplayShowHomeEnabled(true);   // opcional muestra icono
getSupportActionBar().setIcon(R.mipmap.ic_launcher);         //opcional muestra icono
//ver informacion CAMBIAR ICONO DE LA APLICACIÓN


//LOGICA VS GRAFICA
btn1 = (Button) findViewById(R.id.button);
txt1 = (EditText) findViewById(R.id.editTextTextPersonName);
txt2 = (EditText) findViewById(R.id.editTextTextPassword);


//programamos el boton de LOGIN, controlamos credenciales y mostramos Mensajes informativos  (antes de las 2 llaves }})

btn1.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {

      if((txt1.getText().toString().equals("user"))  && (txt2.getText().toString().equals("123")))
      {
         Intent pantallamenu = new Intent(MainActivity.this, MainActivity2.class);
         startActivity(pantallamenu);
         finish();
      }
      else
      {
         AlertDialog.Builder alerta = new AlertDialog.Builder(MainActivity.this);
         alerta.setTitle("ALERTA");
         alerta.setMessage("Credenciales incorrectas")
               .setIcon(android.R.drawable.ic_dialog_alert);
         alerta.show();

         txt1.setText("");
         txt2.setText("");
         txt1.requestFocus();

      }

   }
});
```

//programamos el boton de ATRÁS ◄ (barra de navegación).

1. Click derecho en la zona de Código, **Generate-Override Methods** el metodo de nombre **onBackPressed()**
2. Una vez creado el método para el evento programamos

```
@Override
public void onBackPressed() {
  //super.onBackPressed();   esta línea se dehabilita
  //ponemos el código para cerrar

 //codigo1
MainActivity.super.onDestroy();
android.os.Process.killProcess(android.os.Process.myPid());

  //alternativa
   System.Exit(0)


//SUPER CIERRE---CUANDO HAY MUCHOS LAYOUTS

Intent salir = new Intent(Intent.ACTION_MAIN);
salir.addCategory(Intent.CATEGORY_HOME);
salir.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
startActivity(salir);
System.exit(0);



//o usar el Alert.Builder para confirmar el cierre
AlertDialog.Builder alerta  = new AlertDialog.Builder(MainActivity.this);
        alerta.setTitle("Aviso");
        alerta.setMessage("Desea Cerrar la Aplicación");
        alerta.setIcon(android.R.drawable.ic_dialog_alert);
        alerta.setCancelable(false);
        alerta.setPositiveButton("Si", new DialogInterface.OnClickListener() {
           @Override
           public void onClick(DialogInterface dialog, int which) {
              System.exit(0);
           }
        });

        alerta.setNegativeButton("No", new DialogInterface.OnClickListener() {
           @Override
           public void onClick(DialogInterface dialog, int which) {

           }
        });

       alerta.show();
}
```

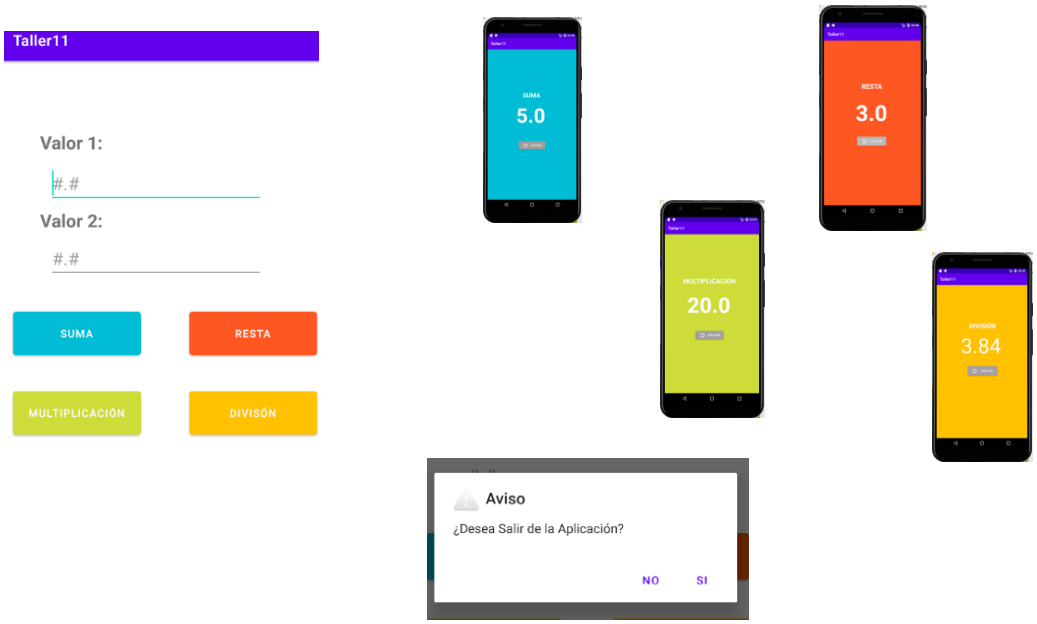
Taller 13: Uso de Varios Screen (Activity, Layout, pantallas)

REQUERIMIENTOS

- ✓ Crear un Proyecto "Empty Activity" (Lollipod 5.0)
- ✓ Insertar los siguientes componentes ***MainActivity***:
  - 2 TextView (labels)
  - 2 plainText (edit tex)
  - 4 Buttons
- ✓ Crear *MainActivity2*, *MainActivity3*, *MainActivity4*, *MainActivity5*:
  - 2 TextView (labels)
  - 1 Buttons

- ✓ **Implementar un programa que permita realizar las operaciones ; en pantallas diferentes (Activity)**
- ✓ **Explicar y conocer el manejo de Activity (ida/regreso)**
- ✓ **Hacer énfasis en el manejo de eventos (botón ATRAS).**
- ✓ **Pasos de valores entre Activitys.**
- ✓ **Usar avisos o mensajes de alerta para controlar errores.**
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
  Compilar y Revisar Errores.

PANTALLAS



PROCEDIMIENTO

*****// Pantalla Principal MainActivity
Poner los componentes requeridos

****//Crear los demás Activity
File – New – Activity – Empty Activity

Poner los componentes Necesario (2 Textview, 1 Button, Color de Background)

********// Crear variables propias en la **MainActivity**

```
Button btn1,btn2,btn3,btn4;
TextView lbl1,lbl2;
EditText txt1,txt2;
```

**********// Asociar variables a los componentes **MainActivity**

```
@Override
protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_main);

    getSupportActionBar.setTitle("MENU PRINCIPAL")

   btn1 = (Button) findViewById(R.id.button);
   btn2 = (Button) findViewById(R.id.button2);
   btn3 = (Button) findViewById(R.id.button3);
   btn4 = (Button) findViewById(R.id.button4);

   lbl1 = (TextView) findViewById(R.id.textView5);
   lbl2 = (TextView) findViewById(R.id.textView6);


   txt1 = (EditText) findViewById(R.id.editTextTextPersonName);
   txt2 = (EditText) findViewById(R.id.editTextTextPersonName2);


   txt1.setText("");
   txt2.setText("");


//Programar los botones según la operación para que llame al Layout correspondiente
Y pasarle los valores del MainActivity

//Boton1 SUMA al Layout MainActivity2

btn1.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {

      if( txt1.getText().toString().isEmpty() ||  txt2.getText().toString().isEmpty())
      {
         Toast.makeText(  getApplicationContext() ,"Existen Campos Vacio", Toast.LENGTH_SHORT).show();
      }
      else
      {

         //Llamado a MainActivity2
         Intent pantallasuma = new Intent(MainActivity.this, MainActivity2.class);

         //Paso de valores al MainActivity2
          pantallasuma.putExtra("prm1", txt1.getText().toString());
          pantallasuma.putExtra("prm2", txt2.getText().toString());
```

```
        //Inciar MainActivity2
          startActivity(pantallasuma);

        //Cerrar MainActivty Origen
          finish();

    }
  }
});



//Boton2 RESTA al Layout MainActivity3
btn2.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {

    if( txt1.getText().toString().isEmpty() ||  txt2.getText().toString().isEmpty())
    {
      Toast.makeText(  getApplicationContext() ,"Existen Campos Vacio",
Toast.LENGTH_SHORT).show();
    }
    else
    {

    Intent pantallaresta = new Intent(MainActivity.this, MainActivity3.class);
      pantallaresta.putExtra("prm1", txt1.getText().toString());
      pantallaresta.putExtra("prm2", txt2.getText().toString());
    startActivity(pantallaresta);
    finish();
    }
  }
});


//Boton3 MULTIPLICAR al Layout MainActivity4
btn3.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {

    if( txt1.getText().toString().isEmpty() ||  txt2.getText().toString().isEmpty())
    {
      Toast.makeText(  getApplicationContext() ,"Existen Campos Vacio",
Toast.LENGTH_SHORT).show();
    }
    else
    {

    Intent pantallamulti = new Intent(MainActivity.this, MainActivity4.class);
      pantallamulti.putExtra("prm1", txt1.getText().toString());
      pantallamulti.putExtra("prm2", txt2.getText().toString());
    startActivity(pantallamulti);
    finish();
    }

  }
```

```
});

//Boton4 DIVIDIR al Layout MainActivity5
btn4.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {


      if( txt1.getText().toString().isEmpty() ||  txt2.getText().toString().isEmpty() ||
Float.parseFloat(txt2.getText().toString()) == 0.0)
      {
         Toast.makeText(  getApplicationContext() ,"Comprobar: Campos Vacios/Division para cero",
Toast.LENGTH_SHORT).show();
      }
      else
      {
      Intent pantalladv = new Intent(MainActivity.this, MainActivity5.class);
         pantalladv.putExtra("prm1", txt1.getText().toString());
         pantalladv.putExtra("prm2", txt2.getText().toString());

      startActivity(pantalladv);
      finish();
      }
   }
});




**************************************************
MainActivity2  - “Suma”
MainActivity3  - “Resta”
MainActivity4  - “Multiplicación”
MainActivity5  - “División”

**************************************************
*****Crear variables propias

Button btn1;
TextView respsuma;

****Asociar Variables a Componentes
btn1 = (Button) findViewById(R.id.button5);
respsuma = (TextView) findViewById(R.id.textView7);



*****Captura valores enviados  desd la Activity Principal y realizar la operación

String num1 = getIntent().getStringExtra("prm1");
String num2 = getIntent().getStringExtra("prm2");
@Override

Evento OnCreate
float res =   Float.parseFloat(num1) +  Float.parseFloat(num2);
respsuma.setText(res + "");
```

REGRESAR AL LAYOUT ANTERIOR – METODO 1
******Programar el botón VOLVER de cada Layout

```
btn1.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {

      Intent pantallamenu = new Intent(MainActivity2.this, MainActivity.class);
      startActivity(pantallamenu);
      finish();


   }
});
```

REGRESAR AL LAYOUT ANTERIOR – METODO 2
******Programar el botón atrás ◄ barra de navegación

1.Click derecho en la zona de Código(parte final antes de las }}), Generate-Override Methods el metodo de nombre onBackPressed()
2.Una vez creado el método para el evento programamos

```
@Override
public void onBackPressed() {
   Intent pantallamenu = new Intent(MainActivityX.this, MainActivityY.class);
      startActivity(pantallamenu);
      finish();
}
```

REGRESAR AL LAYOUT ANTERIOR – METODO 3
****** icono atrás en el <- ActionBar

OJO: QUITAR finish(); del código Intent.

En el archivo AndroidManifest.xml, dirigirse al activity en la que quieres programar el regreso(<-)
Poner el parámetro android:parentActivityName=".actividad a donde ir"

Ejemplo 2 al menu

```
<activity android:name=".MainActivity2"
android:parentActivityName=".MainActivity"></activity>
```

Ejemplo 3 al menu

```
<activity android:name=".MainActivity3"
android:parentActivityName=".MainActivity"></activity>
```

//SUPER CIERRE---CUANDO HAY MUCHOS LAYOUTS

```
        Intent salir = new Intent(Intent.ACTION_MAIN);
        salir.addCategory(Intent.CATEGORY_HOME);
        salir.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(salir);
        System.exit(0);
```

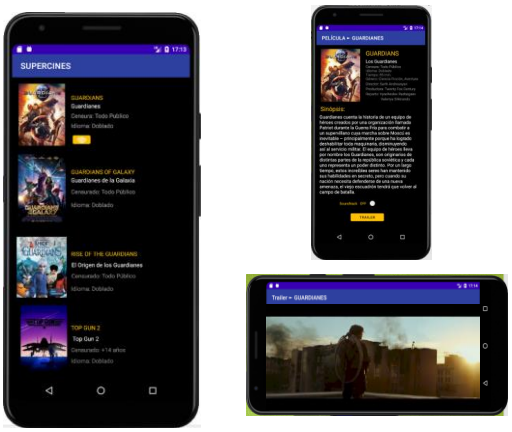
Taller 14: App Multimedia

REQUERIMIENTOS

- ✓ Crear un Proyecto "Empty Activity" (Lollipod 5.0)
- ✓ Crear 3 Activity(Layout)
- ✓ Insertar los siguientes componentes ***MainActivity***:
  - 1 ScrollView (Containers)
  - ImageView
  - 4 TextView (labels)
  - 1Button

  (por cada película)
- ✓ ***MainActivity2 :***
  - ImageView
  - 4 TextView (labels)
  - 1 Switvh(commons)
  - 1 Button
- ✓ ***MainActivity3 : (generar el layout lanscape)***
  - VideoView
- ✓ ***Implementar un aplicación que permite interactuar con contenido multimedia (imagen, texto, audio, video)***
- ✓ ***Hacer énfasis en el manejo de eventos (botón ATRAS).***
- ✓ ***Usar avisos o mensajes de alerta para controlar errores.***
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
- ✓ Compilar y Revisar Errores.

Pantallas



Procedimiento

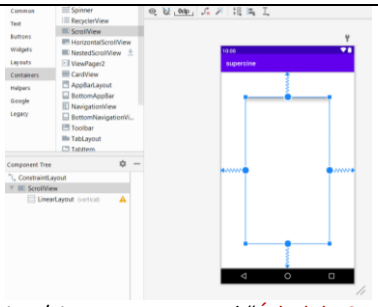
```
//cambiar del titulo del action bar
getSupportActionBar().setTitle("Acceso");
```

Crear MainActivity 2, MainActivity 3

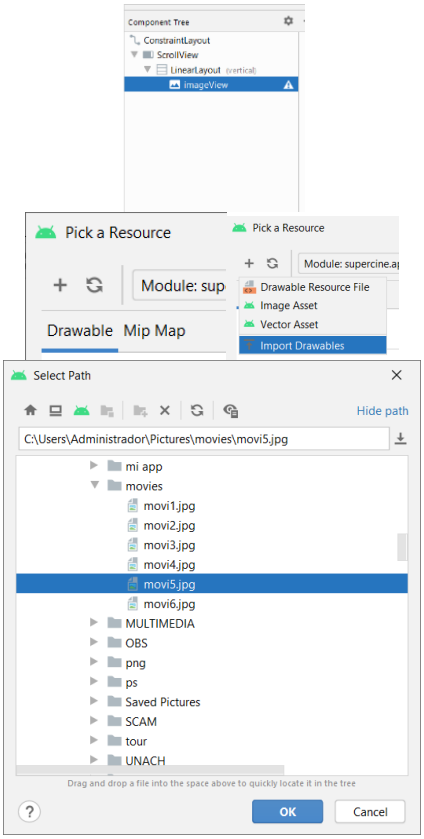
Files-New-Activity-Emty Activity

Sobre el MainActivity1

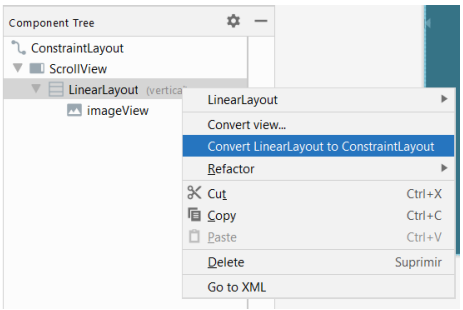
Agregar el componente ScrollView (ocupar toda la pantalla)

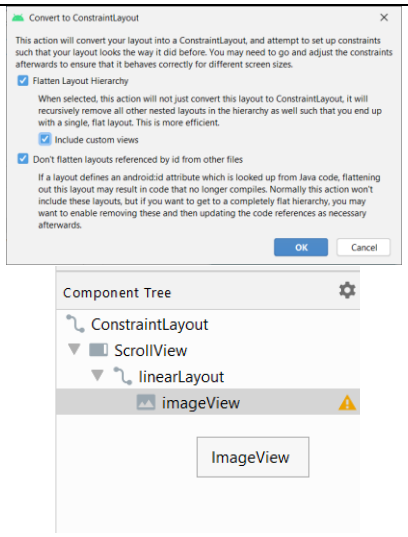


Agregar ImageView dentro del ScrollView/LinearLayaout en el “Árbol de Componetes” y agregar las imágenes personalizadas de 3 portadas de películas:

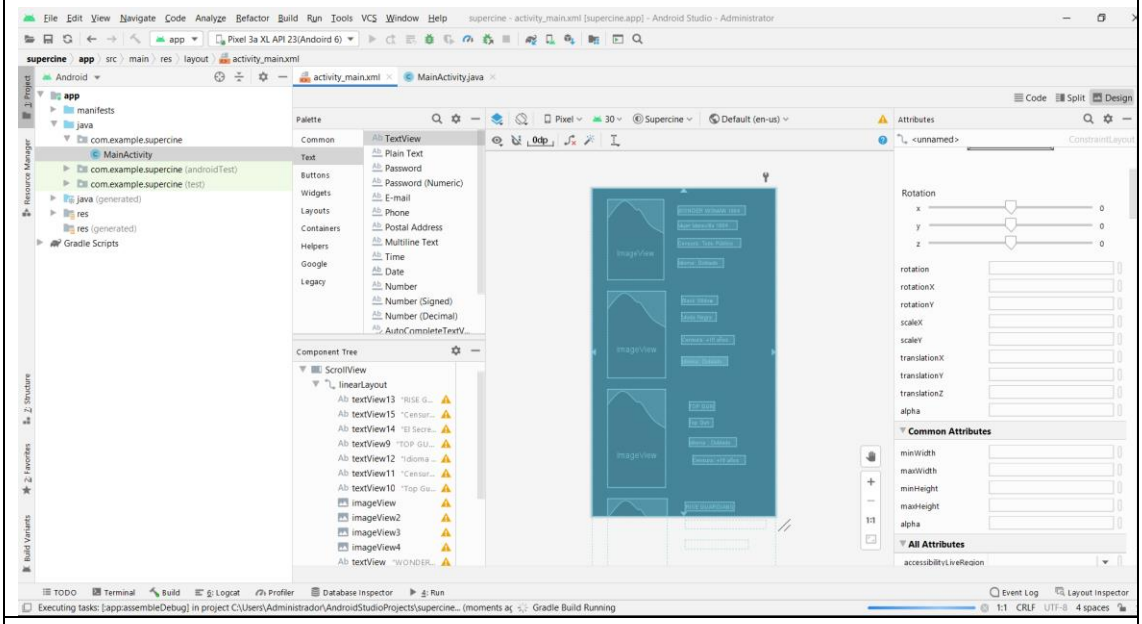


Para poder Ordenar lo componentes dentro del ScrollView/LinearLayout convertirlo a ConstrainLayout; click derecho en linearLayout:





Agregar el resto de Componentes necesarios hasta obtener el diseño propuesto (se recomienda trabajar en BLUEPRINT para acomodar los componentes)



Programamos el Boton (Button1) correspondiente a la primera película (Recuerde Crear el Activity2)
//Declaro las variables propias ausar:

```
Button btn1;
```

//Asigno variables propias a los componentes

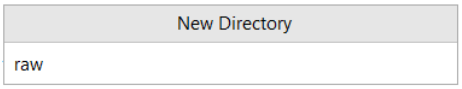
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);



   btn1= (Button) findViewById(R.id.button);

}
}
```

//Creamos el evento OnClickLister y enlazamos al MainActivity2

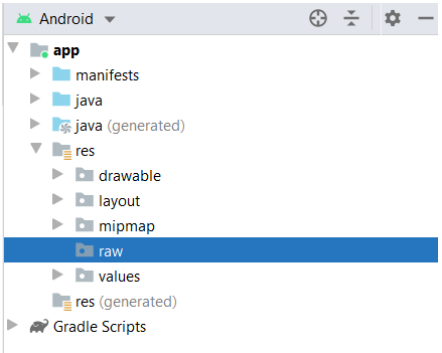
```
btn1= (Button) findViewById(R.id.button);


btn1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        Intent peli1 = new Intent(MainActivity.this, MainActivity2.class);
        startActivity(peli1);
        finish();

    }
});
```

//Agregamos Sonido al click del botón usando la librería **SoundPool**

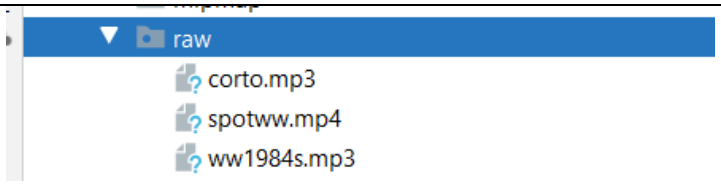
Creamos la carpeta raw (manualmente), en el arbol de proyecto, click derecho sobre el directorio res ; new-Directory y ponemos de nombre raw



El árbol se verá así:



Sobre esta nueva carpeta(directorio) copiamos (ctr+c, ctr+v) todos los elementos multimedios (sonidos, audios, videos- mp3,mp4); los nombres de archivo deben estar en minúsculas sin espacios.



//Sonido en el clik del botón, librería SoundPool (para sonidos cortos)

//Creamos las variables propias

```
SoundPool sp1;
int id_sonido;
```

//en el evento créate MainActivity creamos el objeto

```
sp1 = new SoundPool(1, AudioManager.STREAM_MUSIC,1);
id_sonido=sp1.load(this, R.raw.corto, 1);
```

//en el evento OnClikListener del button1 reproducimos el sonido

```
sp1.play(id_sonido,1,1, 1,0,1);
```

Sobre el MainActivity2

Agregar los componentes necesarios

- ✓ ***MainActivity2 :***
  - ImageView
  - 4 TextView (labels)
  - 1 Switvh(commons)
  - 1 Button

//Poner soundtrack en el botón tipo switch ()

//declaro variable propia para el switch y para el componente MediaPlayer

```
Switch sw1;
MediaPlayer mp;
```

//asigno variables propias a los comp0jnetes del sistema

```
sw1= (Switch)findViewById(R.id.switch1);

mp = MediaPlayer.create(this,R.raw.guardians);
getSupportActionBar().setTitle(" PELÍCULA ► WONDER WOMAN 1984");
```

//en el evento click del switch valido y activo el sonido (no olvidar copiar el sonido en la carpeta raw)

```
sw1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        if (sw1.isChecked())
        {

            sw1.setText("Soundtrack ON");
            mp.start();
```

```
            }
            else
            {
                sw1.setText("Soundtrack OFF");
                mp.pause();

            }


        }
    });
```

//creo la función del onBackPress para el botón de retroceder al layout anterior y pongo detener sonido(stop)

Click derecho, en el código, GENERATE-METHODS OVERRIDE – onBackPressed

```
    @Override
    public void onBackPressed() {

        //super.onBackPressed();
        mp.stop();

        Intent pantallamenu = new Intent(MainActivity2.this, MainActivity.class);
        startActivity(pantallamenu);
        finish();

    }
```

//programar el boton Trailer (llama al Activity3, no olvidar crearla) para mostrar video Trailer

// declaro variable propia

```
    Button btn2;
```

//Asigno variable al componente

```
    Btn2 = (Button) findViewById(R.id.button2);
```

//Programo el evento onClickListener

```
    btn2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Intent trailer1 = new Intent(MainActivity2.this, MainActivity3.class);
            startActivity(trailer1);
            finish();

        }
    });
```

Sobre el MainActivity3 (no olvidar copiar el video mp4 al directorio raw)

//Agrego el componente VideoView , que ocupe toda el Layout

//Crear Varible propia del Componente VideoView

```
    VideoView videotrailer;
```

//En el Evento OnCreate() asigno la varible al componte, asigno path de video y agrego el controller

```
    getSupportActionBar().setTitle(" Trailer ► WONDER WOMAN 1984");
```

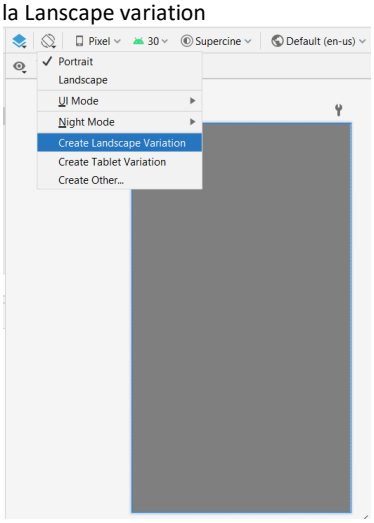
```
        videotrailer = (VideoView) findViewById(R.id.videoView);

        videotrailer.setVideoURI(Uri.parse("android.resource://"+ getPackageName() + "/" + R.raw.spotww));
        MediaController mc = new MediaController(this);
        videotrailer.setMediaController(mc);
        videotrailer.start();
```

//programo el evento OnBackPressed

```
@Override
public void onBackPressed() {
    //super.onBackPressed();
    Intent peli1 = new Intent(MainActivity3.this, MainActivity2.class);
    startActivity(peli1);
    finish();


}
```

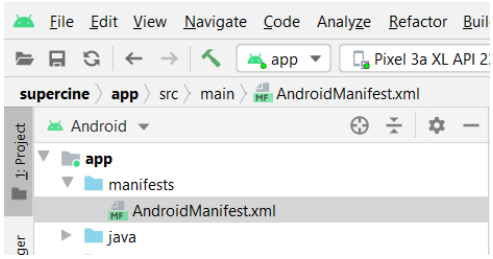
HABILITAR MODO LANSCAPE ACTIVITY3

Estando en la Activity escojida , crear la Lanscape variation



Notaremos que ahora tenemos 2 layouts idénticos pero de diferente orientación

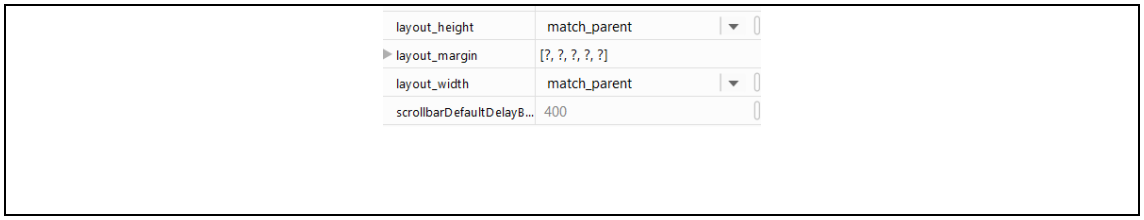
Nos dirigimos al archivo xml llamado AndroidManifest.xml



Buscamos el Activity que queremos mostrar en modo Lanscape (horizontal) y agregamos un parámetro a la etiqueta correspondiente.

```
<activity android:name=".MainActivity3"  android:screenOrientation="landscape" ></activity>
```

```
Nota: si quiere ocupar todo el ancho y alto del VideoView poner el las propiedades del componente:

Layout_Height match_parent
Laayout_Width match_parente
```

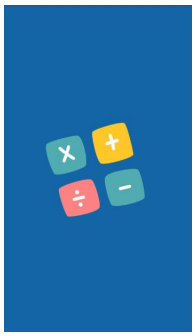

layout_height
match_parent
layout_margin
[?, ?, ?, ?, ?]
layout_width
match_parent
scrollbarDefaultDelayB...
400

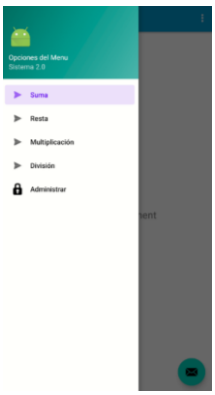
Taller 15: Menu Drawer

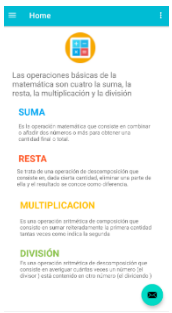
REQUERIMIENTOS

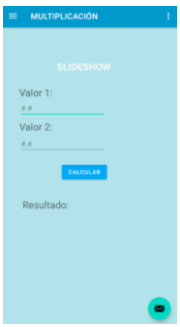
- ✓ Crear un Proyecto con 1"Empty Activity" (Lollipod 5.0)
- ✓ Agregar 1 Navigation Drawer Activity
- ✓ Insertar los siguientes componentes ***MainActivity1***:
  1 ImageView
  1 TextView (labels)
- ✓ Insertar los siguientes componentes ***MainDrawerActivity : (Menu)***
  4 Items (4 Fragmentos)
- ✓ Insertar los siguientes componentes ***MainActivity2 : (Login)***
  1 ImageView
  2 TextView (labels)
  2 PlainText(txtbox)
  1 Button
- ✓ ***Implementar un aplicación que permite mostrar la información basada en un menú SLIDE***
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
- ✓ Compilar y Revisar Errores.

PANTALLAS







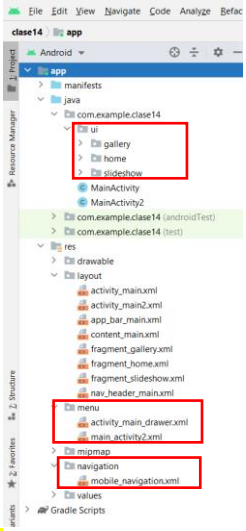


PROCEDIMIENTO

Creamos nuestra Proyecto

1. Agregamos una actividad **Empty Activity – "MainActivity" (Splash)**
2. Agregamos una actividad **Navigation Drawer Activity** - "**MainActivity2**"(**Menu Slide**)

Arbol una vez creado NAvigation Drawer Ativity



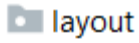
3. Programamos el Splash (MainActivity) - Colomamos una imagen/texto

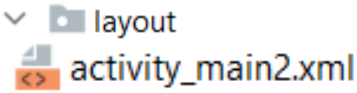
Declarar variables propias

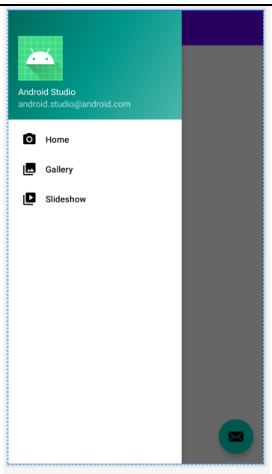
Asociar variables a componentes gráficos

```
@Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
----------
----------

 //funcion para quitar Action Bar
       getSupportActionBar().hide;

//Evento para el Timer

       new Handler().postDelayed(new Runnable() {
          @Override
          public void run() {

             Intent menu = new Intent(MainActivity.this, MainActivity2.class);
             startActivity(menu);
             finish();
          }
       },2000);
```

4. Adecuamos Nuestro MainActivy2 (Menu)

Explicaión de manejo: 



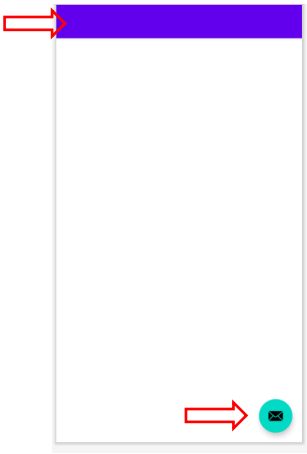



```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://s
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

    <include
        android:id="@+id/app_bar_main"
        layout="@layout/app_bar_main"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <com.google.android.material.navigation.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header_main"
        app:menu="@menu/activity_main_drawer" />
</androidx.drawerlayout.widget.DrawerLayout>
```

No editable

layout

app_bar_main.xml



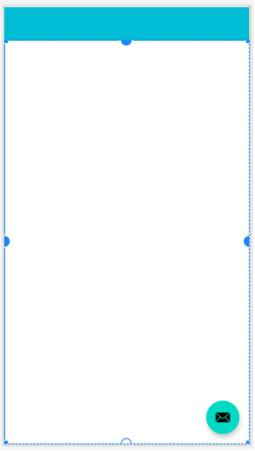
```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout xmlns:android="http://schema
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity2">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/Theme.Clase14.AppBarOverlay">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/Theme.Clase14.PopupOverlay" />

    </com.google.android.material.appbar.AppBarLayout>

    <include layout="@layout/content_main" />

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="16dp"
        app:srcCompat="@android:drawable/ic_dialog_email" />

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

Editable

- Color (Barra) background
- Icon(floatinButton) srcCompac

layout

content_main.xml



```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.and
    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"
    app:layout_behavior="com.google.android.material.appbar.AppBarLayout$Scrolli..."
    tools:showIn="@layout/app_bar_main">

    <fragment
        android:id="@+id/nav_host_fragment_content_main"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

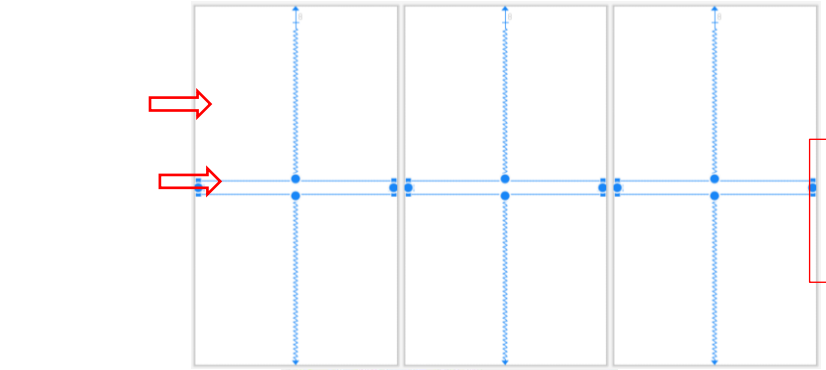
NO Editable

layout

fragment_gallery.xml

fragment_home.xml

fragment_slideshow.xml



Editable

- Color (Background)
- TextView (Visibilty) false

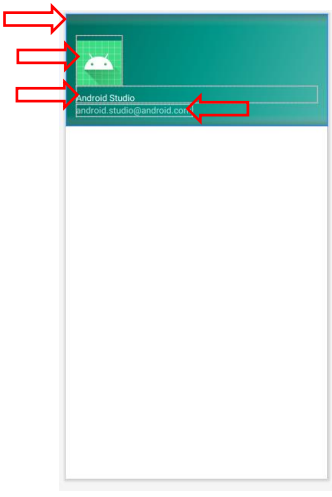
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android=
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ui.gallery.GalleryFragment">

    <TextView
        android:id="@+id/text_gallery"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:textAlignment="center"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns: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"
    tools:context=".ui.home.HomeFragment">

    <TextView
        android:id="@+id/text_home"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:textAlignment="center"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:andro
    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"
    tools:context=".ui.slideshow.SlideshowFragment">

    <TextView
        android:id="@+id/text_slideshow"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:textAlignment="center"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

layout

nav_header_main.xml




```
<?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"
    android:layout_width="match_parent"
    android:layout_height="176dp"
    android:background="@drawable/side_nav_bar"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingLeft="16dp"
    android:paddingTop="16dp"
    android:paddingRight="16dp"
    android:paddingBottom="16dp"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:contentDescription="Navigation header"
        android:paddingTop="8dp"
        app:srcCompat="@mipmap/ic_launcher_round" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="8dp"
        android:text="Android Studio"
        android:textAppearance="@style/TextAppearance.AppCompat.Body1" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="android.studio@android.com" />
</LinearLayout>
```

Editable

- TextView (Texto, Color)
- Image (srcCompact 42px)
- LinearLayout(Background)

Color de Fondo puede editarse el archivo que están la carpeta res>drawable **side_nav_bar.xml**

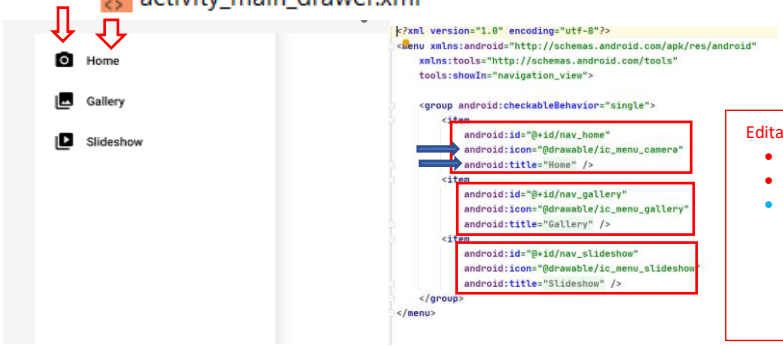
O poner un color mate

Explicaión de manejo:

menu

menu

activity_main_drawer.xml

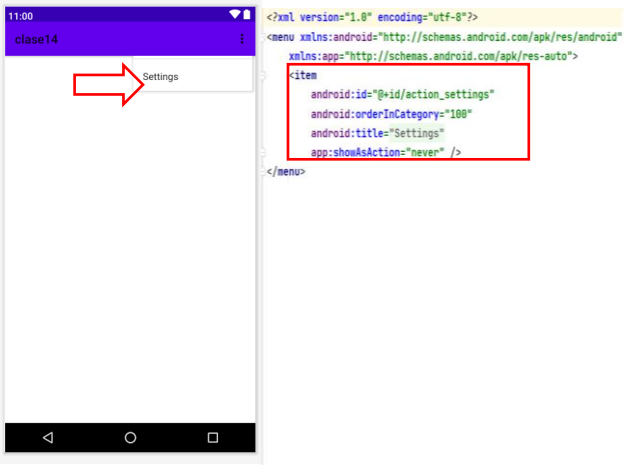


Editable

- Item:title (xml - Title)
- Item:icon(xml-icon)
- Se puede agregar mas Items al menu (copiar/pegar), pero primero se debe crear otro fragment/activity y poner el nombre de este

menu

main_activity2.xml



Editable

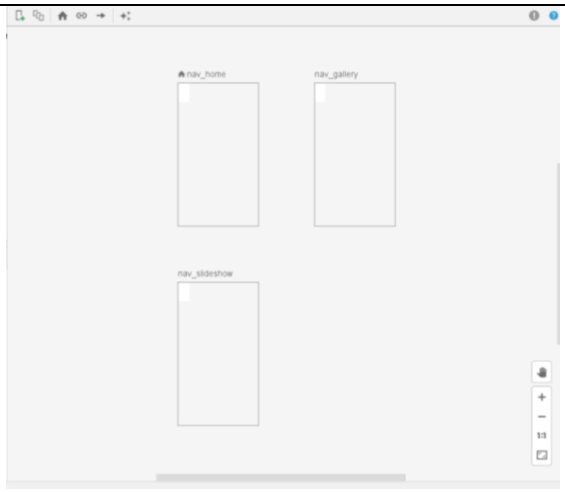
- Item:title (xml - Title)
- Title
- Visible = false(ocultar)

navigation

mobile_navigation.xml

Aquí se pueden crear/agreagr mas fragmentos.

Todos los que están en este pantalla se cargaran a través del MenuDrawer, pero hay que crear el Item (activity_main_drawer.xml) y ahí colocar el nombre del fragmento nuevo para que tenga conectividad



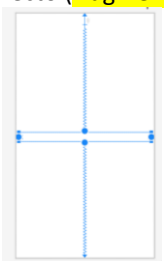
Parte Logica del proyecto

Si borramos el Texview que viene por defecto en los fragments, es ncesario ***borrar/comentar*** esta porción de código, C/fragment

****

Agregamos información (componentes) al Fragment Home.

1. Borrar el Texview que viene por defecto (fragment_home.xml) DESIGN

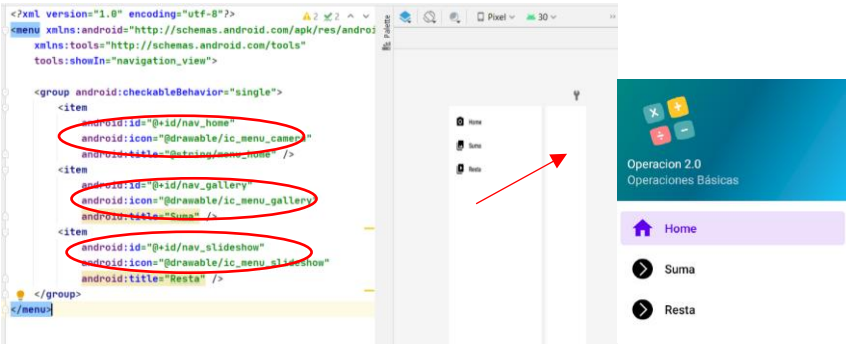


2. Borrar/Comentar el código referente al TexView borrado (HomeFragment.java)

```
/*final TextView textView = binding.textGallery;
    galleryViewModel.getText().observe(getViewLifecycleOwner(), new Observer<String>() {
      @Override
      public void onChanged(@Nullable String s) {
        textView.setText(s);
      }
    });*/
```

NOTA: si desea manejar imagnes en un fragment usar el componente **View** (no ImageView), cragar imagene en la propiedad background...

3. Cambiar los **títulos e iconos de la opciones del menú** en el archivo **activity_main_drawer.xml (code)** en base a las opciones a mostrar

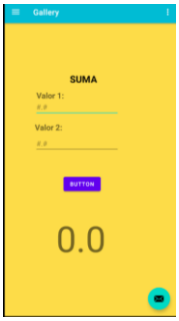


4. Configuración del Diseño de los FRAGMENT que corresponde a la operación:

SUMA ***fragment_gallery.xml***

Borrar o comentar

```
/*final TextView textView = binding.textGallery;
    galleryViewModel.getText().observe(getViewLifecycleOwner(), new Observer<String>() {
        @Override
        public void onChanged(@Nullable String s) {
            textView.setText(s);
        }
    });*/
```



Programamos los eventos

Declaramos Variables propias

```
public class GalleryFragment extends Fragment {

    Button btnsuma;
    TextView resultadosuma;
    EditText txt1, txt2;
```

Asociamos Variables (bajo binding)

```
binding = FragmentGalleryBinding.inflate(inflater, container, false);
View root = binding.getRoot();

    btnsuma = (Button)binding.button;
    resultadosuma = (TextView)binding.textView14;
    txt1 = (EditText)binding.editTextTextPersonName;
    txt2 = (EditText)binding.editTextTextPersonName2;
```

Programamos el boton (Bajo la Asociaon de Variables)

```
    btnsuma = (Button)binding.button;
    resultadosuma = (TextView)binding.textView14;
    txt1 = (EditText)binding.editTextTextPersonName;
    txt2 = (EditText)binding.editTextTextPersonName2;


  btnsuma.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {


        if (txt1.getText().toString().isEmpty() || txt2.getText().toString().isEmpty())
        {
          Toast.makeText(getContext(),"Existen Campos VAcios", Toast.LENGTH_SHORT).show();

        }
        else
        {
          float resul =  Float.parseFloat(txt1.getText().toString()) +Float.parseFloat(txt2.getText().toString());
          resultadosuma.setText(""+resul);
        }
```

RESTA *fragment_slideshow.xml*

Borrar o comentar

```
/*final TextView textView = binding.textGallery;
    galleryViewModel.getText().observe(getViewLifecycleOwner(), new Observer<String>() {
      @Override
      public void onChanged(@Nullable String s) {
        textView.setText(s);
      }
    });*/
```

Programamos los eventos

Variables propias

```
public class GalleryFragment extends Fragment {

  Button btnsuma;
  TextView resultadosuma;
  EditText txt1, txt2;
```

Asociamos Variables

```
binding = FragmentGalleryBinding.inflate(inflater, container, false);
View root = binding.getRoot();

    btnsuma = (Button)binding.button;
    resultadosuma = (TextView)binding.textView14;
    txt1 = (EditText)binding.editTextTextPersonName;
    txt2 = (EditText)binding.editTextTextPersonName2;
```

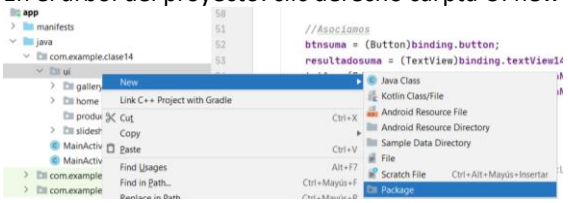
Programamos el boton (bajo la declaracion)

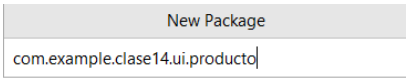
```
    btnsuma.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
```

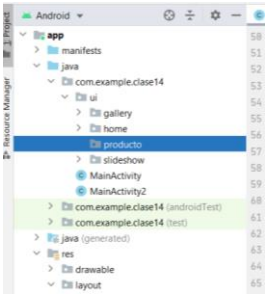
```
if (txt1.getText().toString().isEmpty() || txt2.getText().toString().isEmpty())
{
   Toast.makeText(getContext(),"Existen Campos VAcios", Toast.LENGTH_SHORT).show();

}
else
{
   float resul =  Float.parseFloat(txt1.getText().toString()) +Float.parseFloat(txt2.getText().toString());
   resultadosuma.setText(""+resul);

}
```

5. Agregamos los FRAGMENTS que faltan para las operaciones MULTIPLICACION Y DIVISION

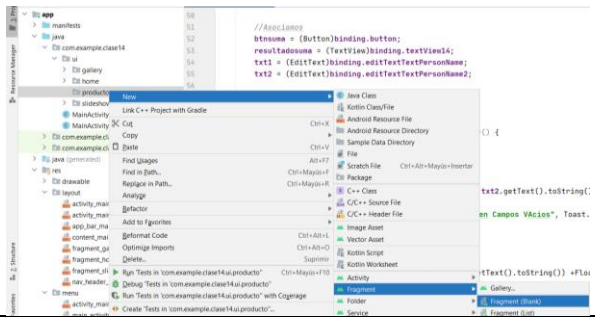
- En el árbol del proyecto: clic derecho carpta UI new package ,"producto"

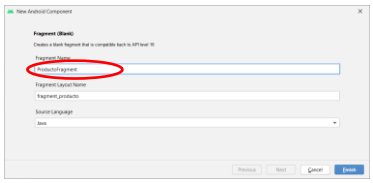


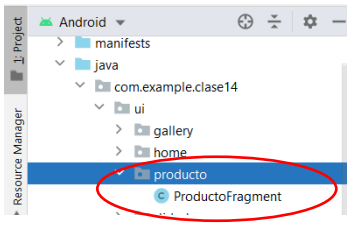




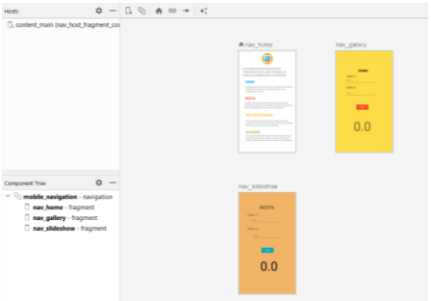
- Click derecho en la nueva carpeta producto y agregar un Fragment (BLank)



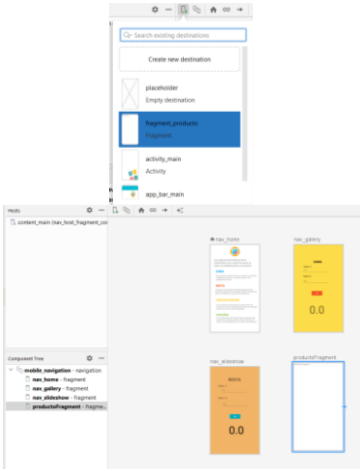




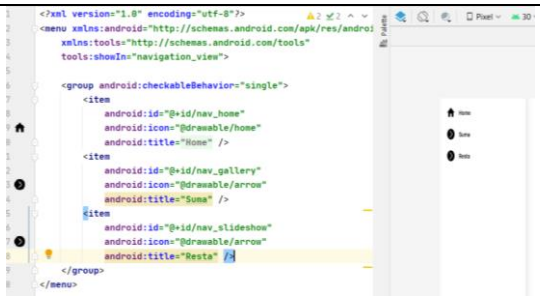
6. Agregar el nuevo **FRAGMENT** **ProductoFragment**, al mapa de navegación **mobile_navigation.xml**

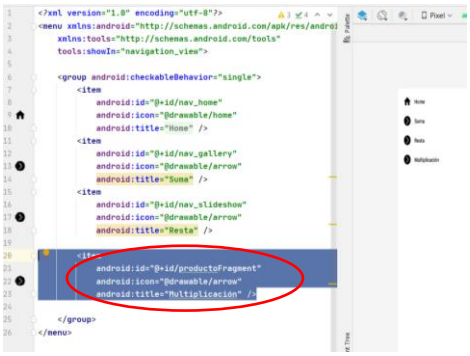


- Clik en el icono de new fragment y escoger el fragment_producto



7. Crear el ITEM en el menú para vincular al fragment nuevo (producto_fragment), archivo **activity_main_drawer.xml (code copiar/pegar/modificar el nombre del fragment)**



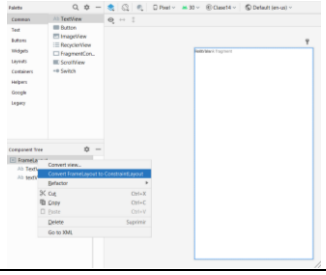


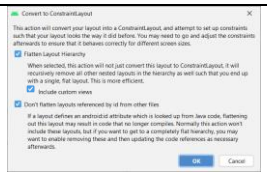
8. Registrar en fragment_producto (y otros) en el mainactivy2 (Drawer para que se muestre el ≡ en vez de la <-) para el regreso al menu

    - En el Archivo MainActivity2.java (MenuDrawer) editar la linea de código y poner los frgamets (Blank) aumentasdos

9. Cambiar de LinearLayout a ConstrainLayout en Diseño de **fragment_producto.xml**

    - Click derecho LinearLayout





- Poner los componenestes necesrios

- Programamos en el FRAGMENT(BLANK) fragment_producto.xml

Declaramos variables propias

```
public class ProductoFragment extends Fragment {

  Button btnmulti;
  EditText txt1, txt2;
  TextView lblresultado;
  View vista;      //esta variable nos permite manejar los eventos del fragment(blank)

 @Override
 public View onCreateView(LayoutInflater inflater, ViewGroup container,
                          Bundle savedInstanceState) {
```

Cambiamos la estructutura el método Create para reconocer los eventos VISTA

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

   vista=inflater.inflate(R.layout.fragment_producto, container, false);


**********************aqui progaramar***************


    return vista;
}
```

Asociamos las variables

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

   vista=inflater.inflate(R.layout.fragment_producto, container, false);


  //asociamos las variables
    btnmulti= vista.findViewById(R.id.button3);
    txt1 = vista.findViewById(R.id.editTextNumber3);
    txt2 = vista.findViewById(R.id.editTextNumber4);
    lblresultado = vista.findViewById(R.id.textView22);


   return vista;
```

```
}
```

Programamos el boton

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

   vista=inflater.inflate(R.layout.fragment_producto, container, false);


  //asociamos las variables
    btnmulti= vista.findViewById(R.id.button3);
    txt1 = vista.findViewById(R.id.editTextNumber3);
    txt2 = vista.findViewById(R.id.editTextNumber4);
    lblresultado = vista.findViewById(R.id.textView22);


    //boton
    btnmulti.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {


            if (txt1.getText().toString().isEmpty() || txt2.getText().toString().isEmpty())
            {
                Toast.makeText(getContext(),"Existen Campos VAcios", Toast.LENGTH_SHORT).show();

            }
            else
            {
                float resul =  Float.parseFloat(txt1.getText().toString()) *
Float.parseFloat(txt2.getText().toString());
                lblresultado.setText(""+resul);


            }


        }
    });


    return vista;

}
```

NOTA...dentro de esta estructura de puede programar todo normal.

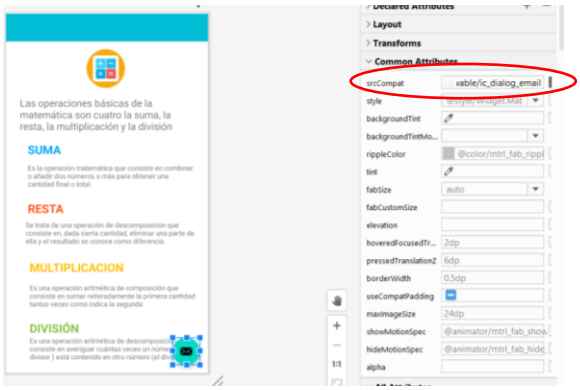
10. Cambiar el texto para mostrar en el ActionBar de cada Fragment

- Editar el archivo mobile_navigation.xml **(code)** y cambiar en nombre en la propiedad label de cada fragment.

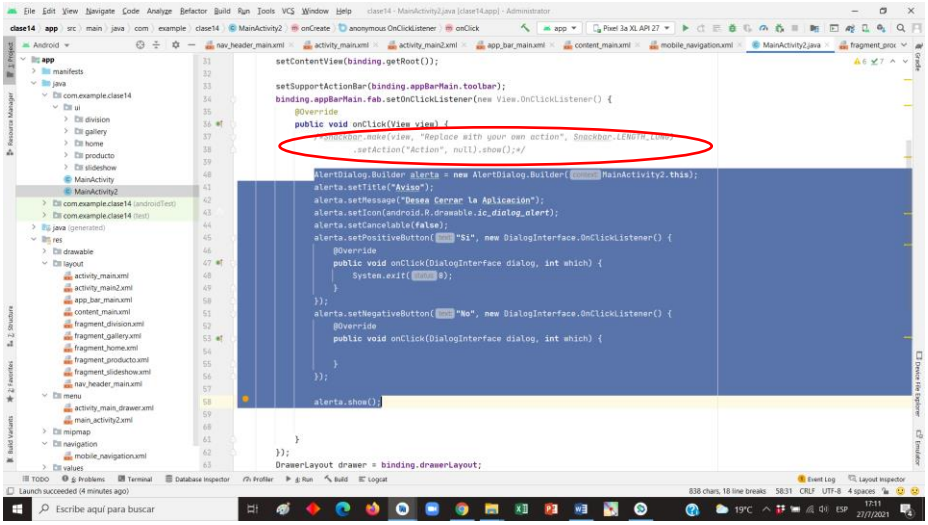
```
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/mobile_navigation"
    app:startDestination="@+id/nav_home">

    <fragment
        android:id="@+id/nav_home"
        android:name="com.example.clase14.ui.home.HomeFragment"
        android:label="@string/menu_home"
        tools:layout="@layout/fragment_home" />

    <fragment
        android:id="@+id/nav_gallery"
        android:name="com.example.clase14.ui.gallery.GalleryFragment"
        android:label="SUMA"
        tools:layout="@layout/fragment_gallery" />

    <fragment
        android:id="@+id/nav_slideshow"
        android:name="com.example.clase14.ui.slideshow.SlideshowFragment"
        android:label="RESTA"
        tools:layout="@layout/fragment_slideshow" />
    <fragment
        android:id="@+id/productoFragment"
        android:name="com.example.clase14.ui.producto.ProductoFragment"
        android:label="MUTIPLICACIÓN"
        tools:layout="@layout/fragment_producto" />
    <fragment
        android:id="@+id/fragment_division"
        android:name="com.example.clase14.ui.division.Fragment_division"
        android:label="DIVISIÓN"
        tools:layout="@layout/fragment_division" />
</navigation>
```

11. Cambiar el Fab button  por el evento salir/cerrar la aplicación

- Cambiar el icono para mostrar, dirigirse al archivo **app_bar_main.xml** propiedades/atributos **srCompact** poner un icono del sistema de la lista de inconos y el color backgroundTint



- Programar en el evento click del Fab button, sobre el archivo **MainActivity2.java** (Menu Darwer) y poner el código de "Cerrar con confirmacion" (AlertDialog.Builder). no olvide comentar el código que ahí se muestra.



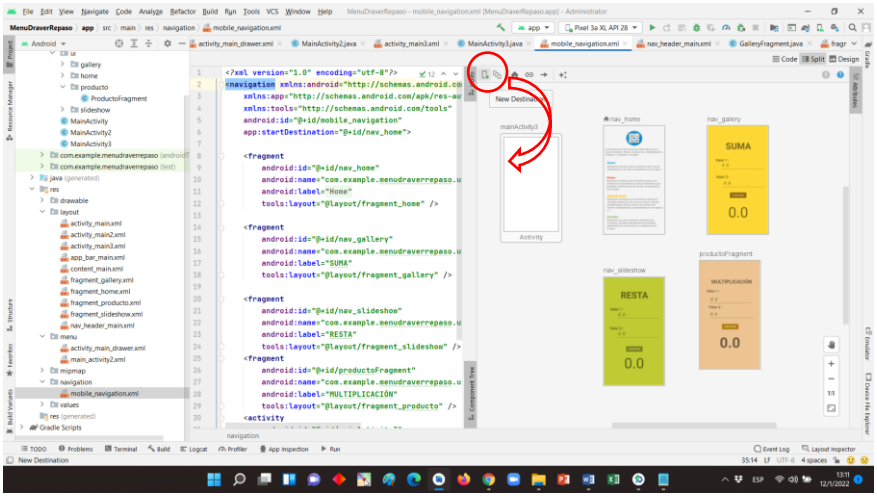
REPETIR LOS PASOS DEL 6 AL 10 PARA AGRRGAR LOS FRAGMENTS NECESARIOAS

Del mismo modo si desea acoplar un **Layout** nuevo (no fragment) ejemplo para un login o un About también deberá rperitr los paso del 6 Al 10 (excepto el paso 8)….

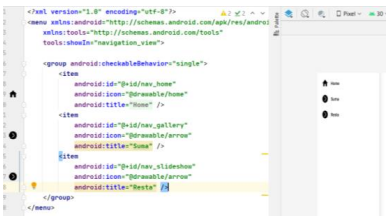
no olvidar desactivar o desprogramar el botón de navegación atrás ◄ para no volver del Layout al MenuDrawer (), si asi lo deseara.

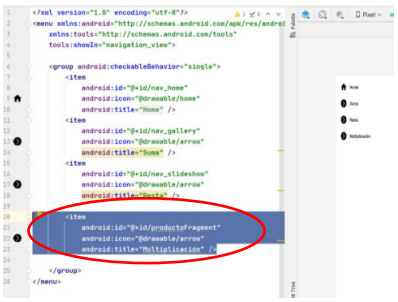
1. New – Activity – Empty Activity

2. Agregar en el activity al archivo: mobile_navigation.xml



3. Crear el ITEM en el menú para vincular al Activity nuevo (ActivitMain3), archivo **activity_main_drawer.xml (code copiar/pegar/modificar el nombre del fragment**





8 (no)

9. Realizar el diseño (grafica y logica) en el activity

10. Editar el archivo mobile_navigation.xml **(code)** y cambiar en nombre en la propiedad label de cada fragment.

NOTA: ACTUALIZAR EN EL EMULARO LA VISTA PARA QUE SE MUESTRE EL MAINACTIVITY3

Taller 16: Uso de Camara

https://developer.android.com/training/camera/photobasics?hl=es-419

REQUERIMIENTOS

- ✓ Crear un Proyecto “Empty Activity” (Lollipod 5.0)
- ✓ Crear +1 Activity(Layout)
- ✓ Insertar los siguientes componentes MainActivity1:
  - ✓ 1 TextView
  - ✓ 1 ImageView
  - ✓ 1 Button (operciones)
- ✓ Implementar un aplicación que permite manejar la cámara de fotos, tomar fotografías, guardar la fotografía en el dispositivo y mostrar un preview la misma
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo), Configurar las propiedades y uso de cámara del AVD
- ✓ Compilar y Revisar Errores.

PANTALLAS

PROCEDIMIENTO

Creamos nuestra Proyecto

1. MainActivity.java poner el código (variables propias)

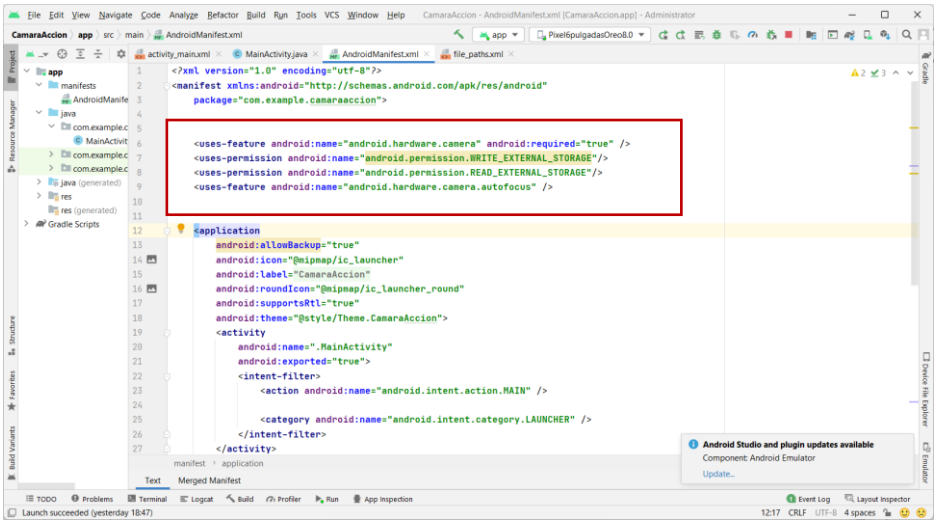
```
TextView lbltitulo;
Button btncamara;
ImageView fotopreviw;

//varibles necesarias para manejo de fotos
File xphotofile;
String name;
Bitmap bitmap;
```

2. MainActivity.java poner el código (asociar parte lógica vs parte grafica)

```
txttitulo = (TextView) findViewById(R.id.textview);
btncamara = (Button) findViewById(R.id.button);
fotopreviw =(ImageView) findViewById(R.id.imageView);
```

3. En AndroidManifest habilitar el permiso de uso de camara y memoria interna:

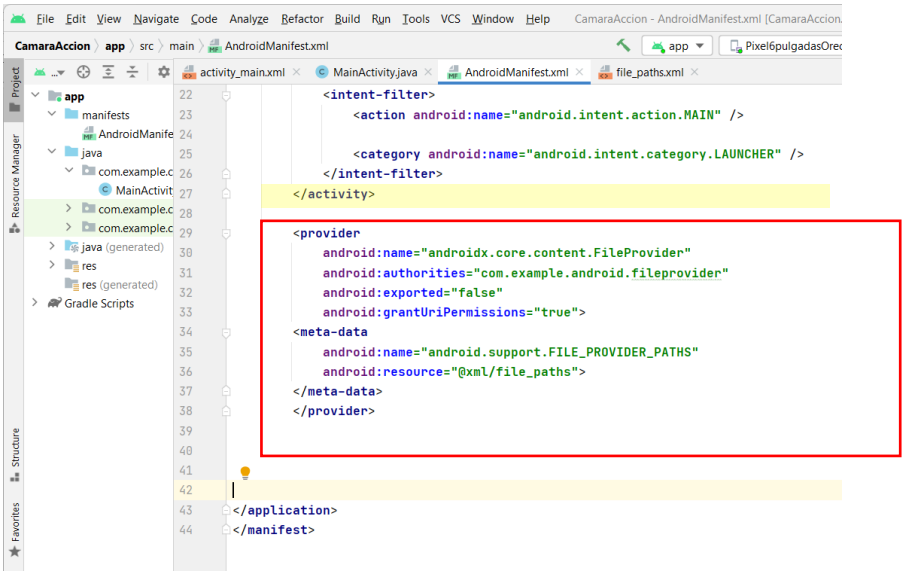
```
<uses-feature android:name="android.hardware.camera" android:required="true" />
<uses-feature android:name="android.hardware.camera.autofocus" />

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```



4. Incorporar en el AndroidManifest el código para el manejo del almacenamiento interno; Despues de `</activity>` y antes del `</application>`

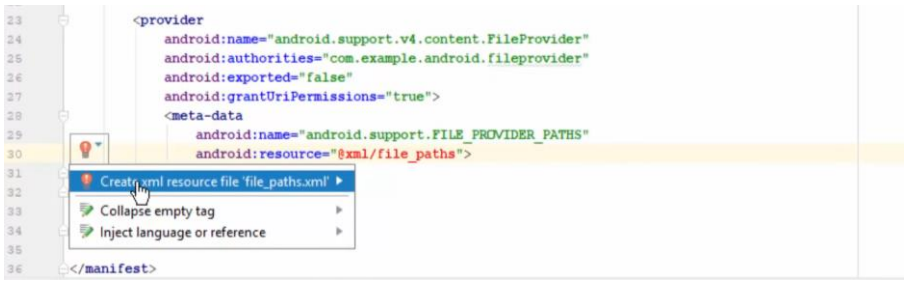
```
<provider
    android:name="androidx.core.content.FileProvider"
    android:authorities="com.example.PROYECTONAME.provider" (digitamos provider)
    android:exported="false"
    android:grantUriPermissions="true">
<meta-data
    android:name="android.support.FILE_PROVIDER_PATHS" (digitamos FILE_PROVIDER_PATHS)
    android:resource="@xml/file_paths">
```

```
</meta-data>
</provider>
```

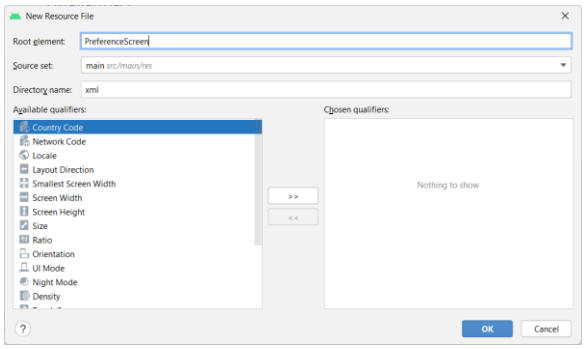


5. Crear el archive file_paths.xml que almacena la dirección donde se almacenaran las fotografías.

- ✓ Click derecho en el foco rojo o Alt+enter y crear xml…..



- ✓ Presionar ok en la siguiente pantalla sin modificar nada



- ✓ Se va a crear un activity file_paths.xml, en cual nos cambairemos a vista de código y lo editamos de la siguiente manera: (ver nombre del proyecto)

```xml
<?xml version="1.0" encoding="utf-8"?>

<paths xmlns:android="http://schemas.android.com/apk/res/android" >

<external-path
    name="my_images"
    path="Android/data/com.example.camaraaccion/files/Pictures" />
</paths>
```

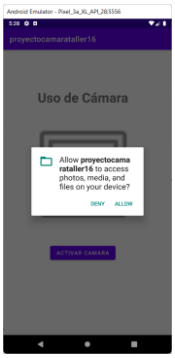
6. Volvemos al MainActiviy.java y ponemos la línea de código que da permiso al uso de memoria para dispositivos

Bajo las variables relacionas

```
txttitulo = (TextView) findViewById(R.id.textview);
btncamara = (Button) findViewById(R.id.button);
fotopreviw =(ImageView) findViewById(R.id.imageView);
```

```
//permiso para uso de memoria para dispositivo
if (ContextCompat.checkSelfPermission(MainActivity.this, Manifest.permission.WRITE_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED)

{

      ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE, Manifest.permission.CAMERA}, 1000);

}
```

asi



7. Crear los siguientes métodos antes de la ultima }:

tomarFoto = Levanta la camara y permite capturar con confirmación

createImageFile = Metodo que construye el nombre de archivo y ubica el path de almacenamiento

onActivityResult = Muestra la vista previa de la foto capturada por la camara

//////////////////////////////// tomarFoto ////////////////////////

```
//metodo tomar foto
static final int REQUEST_TAKE_PHOTO = 1;
public void tomarFoto(View view)
{
    //Levanta la camara
    Toast.makeText(MainActivity.this,"Levantando la camara", Toast.LENGTH_SHORT).show();
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);



    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        // Create the File where the photo should go
        File photoFile = null;
        try {
           photoFile = createImageFile();
            Toast.makeText(MainActivity.this,"Creando Archivo", Toast.LENGTH_SHORT).show();
        } catch (IOException ex) {
            // Error occurred while creating the File
        }
        // Continue only if the File was successfully created
        if (photoFile != null) {
            Uri photoURI = FileProvider.getUriForFile(this, "com.example.PROYECTONAME.provider", photoFile);
            xphotofile= photoFile;
            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);
            startActivityForResult(takePictureIntent, REQUEST_TAKE_PHOTO);
        }

    }
}
```

//////////////////////////// createImageFile ////////////////////////////////

```
//metodo crear archivo
    String currentPhotoPath;
    private File createImageFile() throws IOException
    {
        // Create an image file name
        String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
        String imageFileName = "Backup_" + timeStamp + "_";
        File storageDir1 = getExternalFilesDir(Environment.DIRECTORY_PICTURES);
        File image = File.createTempFile(imageFileName,  ".jpg",storageDir1);

        // Save a file: path for use with ACTION_VIEW intents
        currentPhotoPath = image.getAbsolutePath();
        name=currentPhotoPath;

        return image;

    }
```

//////////////////////////// onActivityResult ////////////////////////////////

```
//vista previa en el Imview
static final int REQUEST_IMAGE_CAPTURE =1;

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {

        //Trasnforma la foto xphotofile a bitmap y muestra en el view
        Bitmap bitmap = BitmapFactory.decodeFile(xphotofile.getAbsolutePath());
        fotopreview.setImageBitmap(bitmap);

        // guardarFotoGallery();
        btncamara.setEnabled(false);

    }
    else {
        Toast.makeText(MainActivity.this,"Cancelo el guardado de la foto", Toast.LENGTH_SHORT).show();
    }
}
```

8. Programar el evento butto onlisterclick o en el evento Onclick (GUI) poner el evento principal tomarFoto()

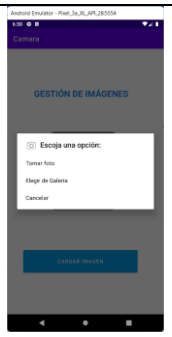
NOTA: DeviceFileExplores▶ ver archivo: App File(Show Internal): Android/data/files/picture

CARGAR IMAGEN DE GALERIA Y GUARDAR EN LA MEMORIA

Mostrar un menú de opciones : Tomar Foto, Mostrar Galleria , Cancelar

1. Programar dentro del envento onclickListener el menú con DialogBuilder

```
btnfoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {


            CharSequence [] opciones ={"Tomar foto", "Elegir de Galeria" , "Cancelar"};

            AlertDialog.Builder alerta = new AlertDialog.Builder(MainActivity.this);

            alerta.setIcon(android.R.drawable.ic_menu_camera);
            alerta.setTitle("Escoja una opción:");
            alerta.setCancelable(false);
            alerta.setItems(opciones, new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int i) {

                    if(opciones[i].equals("Tomar foto"))
                    {

                        //aqui llamar el meto tomar foto, retirar View view del método para poder llamarlo
                          directo
                        tomarFoto();

                    }
                    else
                    {
                        if(opciones[i].equals("Elegir de Galeria"))
                        {

                            Intent galeria = new Intent(Intent.ACTION_GET_CONTENT,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);

                            galeria.setType("image/");
                            startActivityForResult(galeria.createChooser(galeria,"Seleccione"),10);


                        }
                        else
                        {
                            if(opciones[i].equals("Cancelar"))
                            {
                                Toast.makeText(MainActivity.this, "Cancelo", Toast.LENGTH_SHORT).show();
                            }

                        }


                    }


                }
            });
            alerta.show();



        }
    });



}
```


Android Emulator - Pixel_3a_XL_API_28:5554
Camara
GESTIÓN DE IMÁGENES
Escoja una opción:
Tomar foto
Elegir de Galeria
Cancelar
CARGAR IMAGEN

TALLER 17: APP CON BASES DE DATOS LIGERAS (SQLite)

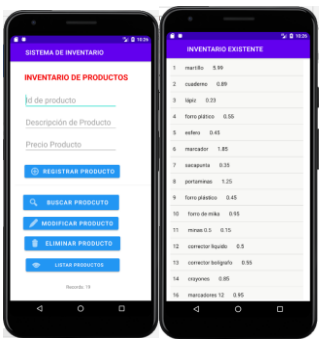
Requerimientos

- ✓ Crear un Proyecto "Empty Activity" (Lollipod 5.0)
- ✓ Crear +2 Activity(Layout)
- ✓ Insertar los siguientes componentes ***MainActivity1***:
  - 1 EditTex
  - 5 Button (operciones)
  - 2 TextView (titulo, contador)
- ✓ ***MainActivity2 :***
  ListView (Legacy)
- ✓ ***Implementar un aplicación que permite administrar una base de datos ligera con todos sus operaciones básicas.***
- ✓ ***Usar el Software BD Software SQLite para comprobar los datos.***
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
- ✓ Compilar y Revisar Errores.

Descargar en instalar DB browser for SQlite
https://sqlitebrowser.org/dl/

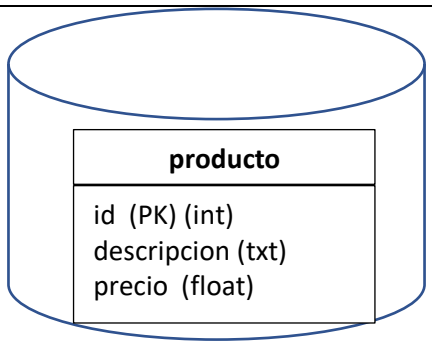
https://www.youtube.com/watch?v=TxkdWX3UaNk

Pantallas

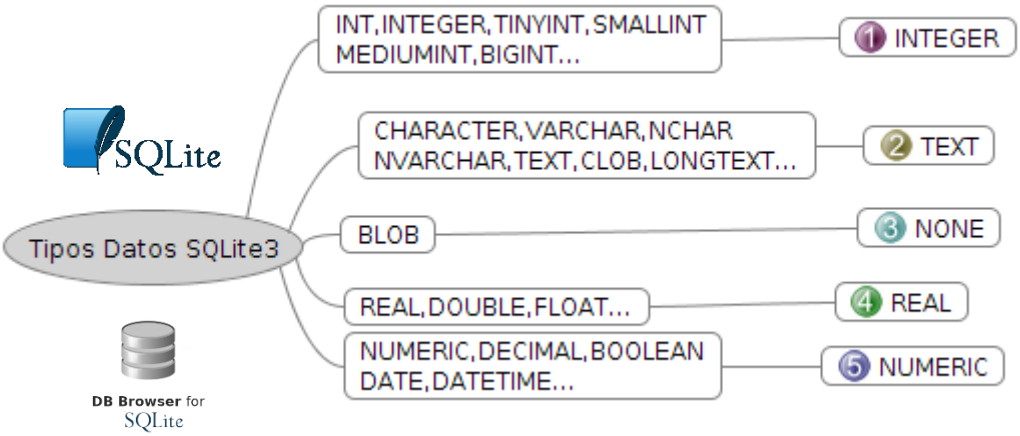


Diseño BD

MiBase




```
create table producto
(
id int primary key
nombre text
precio real
)
```

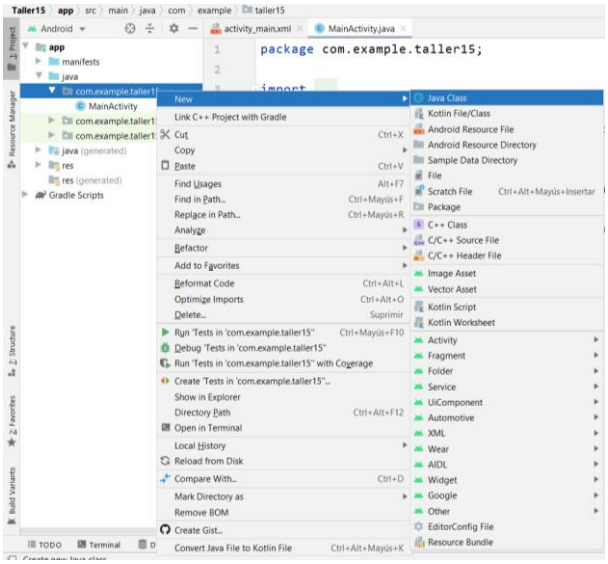


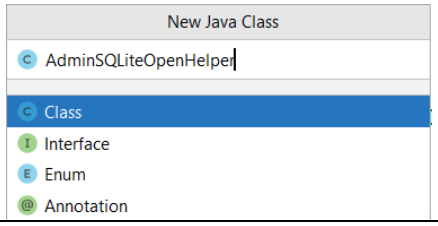
Procedimiento

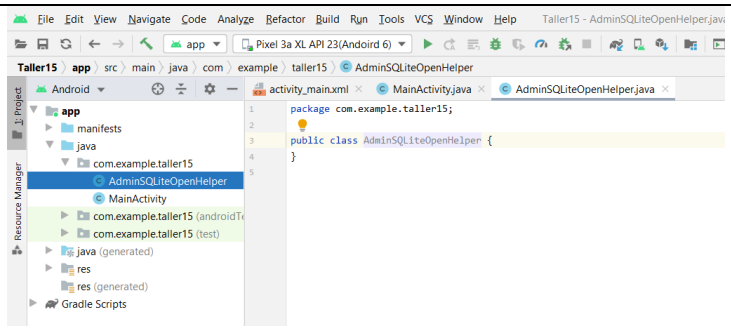
1. Implementar una clase de JAVA para habilitar la librería de conexión a bases de datos SQLite: **SQLiteOpenHelper**

//Crear clase para conexión

Click derecho (árbol de proyecto) en la carpeta que mantiene la parte lógica del app (java►com…..)
New-JavaClass, y asignamos un nombre "AdminSQLiteOpenHelper"







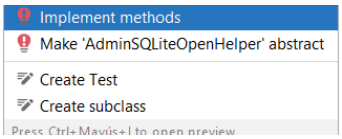
Importamos la librería "import android.database.sqlite.SQLiteOpenHelper"

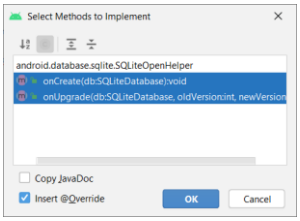
```
package com.example.taller15;

import android.database.sqlite.SQLiteOpenHelper;

public class AdminSQLiteOpenHelper {
}
```

Relacionamos mi clase con la librería importada ***extends***

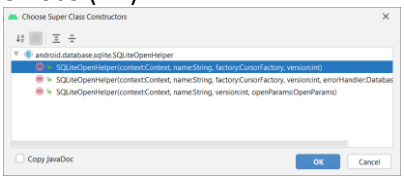
```
public class AdminSQLiteOpenHelper extends SQLiteOpenHelper{
}
```

Generamos automáticamente las clases OnCreate y OnUpgrate
Click en el foco ( ) implementar métodos OnCreate y OnUpgrate …OK





Crear el Constructor Click en el foco ( ) Create constructor matching super (4 parámetros) y ok



Quedará asi:

```
package com.example.taller15;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;
```

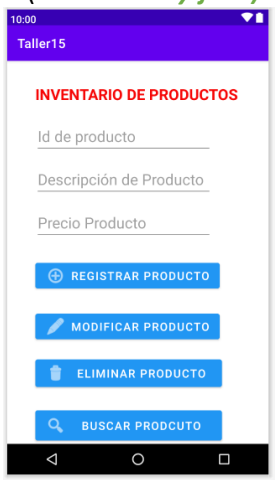
```
public class AdminSQLiteOpenHelper extends SQLiteOpenHelper{
    public AdminSQLiteOpenHelper(@Nullable Context context, @Nullable String name, @Nullable
SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {

    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    }
}
```

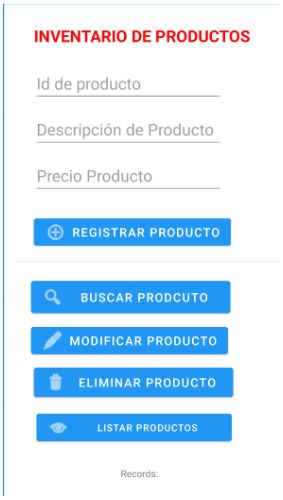
Le cambianos el nombre de la variable de `sqLiteDatabase` a `db`

Insertamos la sentencia para crear la tabla dentro de la base de datos en el método ***OnCreate***

```
@Override
public void onCreate(SQLiteDatabase db) {
   db.execSQL("create table producto(id int primary key, descripcion text, precio real)");

}
```

2. Diseñar el Interfaz de la app (componentes) , crear variables propias y asignar las componentes a las variables propias.(***MainActivity.java)***






```
public class MainActivity extends AppCompatActivity {

    EditText txtid, txtdescripcion, txtprecio;
    Button btnregistrar, btnmodificar, btneliminar, btnbuscar;

        FloatingActionButton btncerrar;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        getSupportActionBar().setTitle("      SISTEMA DE INVENTARIO");


        txtid=(EditText)findViewById(R.id.editTextTextPersonName);
        txtdescripcion=(EditText)findViewById(R.id.editTextTextPersonName2);
        txtprecio=(EditText)findViewById(R.id.editTextTextPersonName3);

        btnregistrar = (Button)findViewById(R.id.button);
        btnmodificar = (Button)findViewById(R.id.button2);
        btneliminar = (Button)findViewById(R.id.button3);
        btnbuscar = (Button)findViewById(R.id.button4);

         FloatingActionButton btncerrar;

    }
}
```

3. Crearemos los métodos para las operaciones : Registrar, Modificar, Eliminar, Buscar, Contar, Mostrar:

Se pueden crear métodos propios, o usar los métodos OnClickListener
Para trabajar con BD SQLite es importante mantener este orden.

- Crear/Conectarse BD
- Abrir BD
- Ejecutar SQL
- Cerrar BD

//INSERCIÓN

Método 1: ContentValues
Método 2: sentencias sql

**********METODO 1 : ContentValues **********

```
//metodo registrar producto
public void registrarproducto(View view)
{
     //Crear/Conectarse BD
    AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper( this, "MiBase", null,1);

    //abrir BD
    SQLiteDatabase BaseDeDatos = cnx.getWritableDatabase();



       //campturo lo que esta en los text
       String id = txtid.getText().toString();
       String descripcion = txtdescripcion.getText().toString();
       String precio = txtprecio.getText().toString();


     if(!id.isEmpty() && !descripcion.isEmpty() && !precio.isEmpty())
     {


    //Ejecutar SQL
    ContentValues values = new ContentValues();
    values.put("id", id);
    values.put("descripcion", descripcion);
    values.put("precio", precio);

    BaseDeDatos.insert("producto", null, values);


    //Cerrar BD
     BaseDeDatos.close();
     Toast.makeText(this,"Dato Almacenado",Toast.LENGTH_SHORT).show();

   //limpiar campos
          txtid.setText("");
          txtdescripcion.setText("");
          txtprecio.setText("");
          txtid.requestFocus();



     }
     else {
         Toast.makeText(this,"Existen Campos vacios",Toast.LENGTH_SHORT).show();

     }

}
```

********** METODO 2 : Sentencias Sql **********

```
//metodo registrar producto
public void registrarproducto(View view)
{
     //Crear/Conectarse BD
    AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper( this, "MiBase", null,1);

    //abrir BD
    SQLiteDatabase BaseDeDatos = cnx.getWritableDatabase();
```

```
       //capturo lo que esta en los text
       String id = txtid.getText().toString();
       String descripcion = txtdescripcion.getText().toString();
       String precio = txtprecio.getText().toString();

     if(!id.isEmpty() && !descripcion.isEmpty() && !precio.isEmpty())
     {



    //Ejecutar SQL

   //insert into producto (id, descripcion, precio) values(1,'martillo', 12.99)

    String sql="insert into producto (id, descripcion, precio) values("+ id +",'"+ descripcion +"',"+ precio +")";
    BaseDeDatos.execSQL(sql);

    //Cerrar BD
     BaseDeDatos.close();
     Toast.makeText(this,"Dato Almacenado",Toast.LENGTH_SHORT).show();

//Limpiar campos
           txtid.setText("");
           txtdescripcion.setText("");
           txtprecio.setText("");
           txtid.requestFocus();



     }
     else {
         Toast.makeText(this,"Existen Campos vacios",Toast.LENGTH_SHORT).show();

     }

}
```

Nota: Se recomienda trabajar con Try Cath, para evitar que se cuelgue la app.

**********Método 1**********

```
//metodo registrar producto
public void registrarproducto(View view)
{
     //Crear/Conectarse BD
    AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper( this, "MiBase", null,1);

    //abrir BD
    SQLiteDatabase BaseDeDatos = cnx.getWritableDatabase();



       //capturo lo que esta en los text
       String id = txtid.getText().toString();
       String descripcion = txtdescripcion.getText().toString();
       String precio = txtprecio.getText().toString();

     if(!id.isEmpty() && !descripcion.isEmpty() && !precio.isEmpty())
     {


    ContentValues values = new ContentValues();
    values.put("id", id);
    values.put("descripcion", descripcion);
    values.put("precio", precio);


    try {

  //Ejecutar SQL
    BaseDeDatos.insert("producto", null, values);

    //Cerrar BD
     BaseDeDatos.close();
     Toast.makeText(this,"Dato Almacenado",Toast.LENGTH_SHORT).show();

   //aquí limpiar los campos
           txtid.setText("");
           txtdescripcion.setText("");
           txtprecio.setText("");
           txtid.requestFocus();


    }
    catch (Exception e)
    {
        Toast.makeText(this,"error "+e,Toast.LENGTH_SHORT).show();
        BaseDeDatos.close();
    }

     }
```

```
      else {
          Toast.makeText(this,"Existen Campos vacios",Toast.LENGTH_SHORT).show();

          Snackbar.make(view, "Existen Campos Vacios", Snackbar.LENGTH_LONG).show();

            // Snackbar.make(view, "Existen Campos Vacios",Snackbar.LENGTH_LONG).setAction("BOTON", new
            View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Toast.makeText(MainActivity.this, "Existen campos vacios", Toast.LENGTH_SHORT).show();
                }
            }).show();


      }

}
```

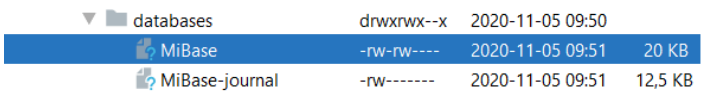
*********método 2**********

```
public void registrarproducto(View view)
{
     //Crear/Conectarse BD
    AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper( this, "MiBase", null,1);

    //abrir BD
    SQLiteDatabase BaseDeDatos = cnx.getWritableDatabase();



       //capturo lo que esta en los text
       String id = txtid.getText().toString();
       String descripcion = txtdescripcion.getText().toString();
       String precio = txtprecio.getText().toString();

     if(!id.isEmpty() && !descripcion.isEmpty() && !precio.isEmpty())
     {


    //Ejecutar SQL
   //insert into producto (id, descripcion, precio) values(1,'martillo', 12.99)

     try {

    String sql="insert into producto (id, descripcion, precio) values("+ id +",'"+ descripcion +"',"+ precio +")";
    BaseDeDatos.execSQL(sql);

    //Cerrar BD
     BaseDeDatos.close();
     Toast.makeText(this,"Dato Almacenado",Toast.LENGTH_SHORT).show();

    //aquí limpiar los campos
           txtid.setText("");
           txtdescripcion.setText("");
           txtprecio.setText("");
           txtid.requestFocus();


      }
      catch (Exception e)
      {
          Toast.makeText(this,"error "+e,Toast.LENGTH_SHORT).show();
          BaseDeDatos.close();
      }


      }
      else {
          Toast.makeText(this,"Existen Campos vacios",Toast.LENGTH_SHORT).show();

      }

}
```

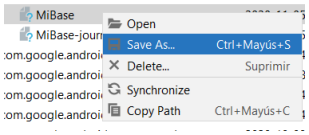
Verificar la Base de Datos usando ***BD Browser for SQLite***

Ir al panel de Explorador de Archivos del Dispositivo ubicado al lado derecho del IDE

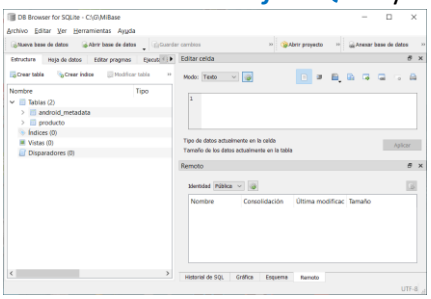
Paht: data▶data▶com.example.taller15▶databases

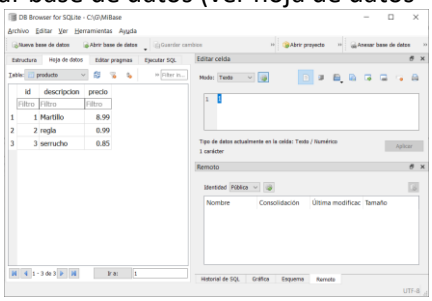


Click derecho SAVE AS



Abrir el software ***BD Browser for SQLite*** y cargar base de datos (ver hoja de datos - bd)





//Modificación

- Crear/Conectarse BD
- Abrir BD
- Ejecutar SQL
- Cerrar BD

Método 1: ContentValues
Método 2: sentencias sql

**********METODO 1 : ContentValues **********

```
public void modificarproducto(View view)
{
    //Crear/Conectarse BD
    AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper( this, "MiBase", null,1);

    //abrir BD
    SQLiteDatabase BaseDeDatos = cnx.getWritableDatabase();

    //capturo lo que esta en los text
    String id = txtid.getText().toString();
    String descripcion = txtdescripcion.getText().toString();
    String precio = txtprecio.getText().toString();


    if(!id.isEmpty() && !descripcion.isEmpty() && !precio.isEmpty())
    {
        ContentValues values = new ContentValues();
        values.put("id", id);
        values.put("descripcion", descripcion);
```

```
        values.put("precio", precio);



        //ejecutamos la TranSQL
         int  band = BaseDeDatos.update("producto",values, "id = " + id,null);


        //cerramos BD
        BaseDeDatos.close();

        if(band==1)
        {
            Toast.makeText(this,"Datos Modificados",Toast.LENGTH_SHORT).show();

            //limpiar campos
            txtid.setText("");
            txtdescripcion.setText("");
            txtprecio.setText("");
            txtid.requestFocus();
        }
        else
        {
            Toast.makeText(this,"Producto no existe, imposible modificar",Toast.LENGTH_SHORT).show();

            //limpiar campos
            txtid.setText("");
            txtdescripcion.setText("");
            txtprecio.setText("");
            txtid.requestFocus();
        }


    }
    else
    {
        Toast.makeText(this,"Existen Campos vacios",Toast.LENGTH_SHORT).show();
    }

}
```

*********método 2**********

```
public void modificarproducto(View view)
{
    //Crear/Conectarse BD
    AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper( this, "MiBase", null,1);

    //abrir BD
    SQLiteDatabase BaseDeDatos = cnx.getWritableDatabase();

    //capturo lo que esta en los text
    String id = txtid.getText().toString();
    String descripcion = txtdescripcion.getText().toString();
    String precio = txtprecio.getText().toString();


    if(!id.isEmpty() && !descripcion.isEmpty() && !precio.isEmpty())
    {

      try {
      //ejecutar update producto set descripción='algo', precio=25.55 where id = 1

     String sql = "update producto set descripcion = '" + descripcion + "', precio = " + precio + " where id = " + id + " ";
                BaseDeDatos.execSQL(sql);
                Toast.makeText(this,"Datos Modificados",Toast.LENGTH_SHORT).show();

               //cierra la BD
                BaseDeDatos.close();

                //limpiar campos
                txtid.setText("");
                txtdescripcion.setText("");
                txtprecio.setText("");
                txtid.requestFocus();
        }
        catch (Exception e)
        {

                Toast.makeText(this,"Problemas: Producto no existe, imposible modificar",Toast.LENGTH_SHORT).show();

                //limpiar campos
                txtid.setText("");
                txtdescripcion.setText("");
                txtprecio.setText("");
                txtid.requestFocus();
```

```
                //cierra la BD
                BaseDeDatos.close();
            }


    }
    else
    {
        Toast.makeText(this,"Existen Campos vacios",Toast.LENGTH_SHORT).show();

           //limpiar campos
                txtid.setText("");
                txtdescripcion.setText("");
                txtprecio.setText("");
                txtid.requestFocus();
    }

}
```

//Eliminación

- Crear/Conectarse BD
- Abrir BD
- Ejecutar SQL
- Cerrar BD

**********Método 1: ContentValues

```
public void eliminarproducto(View view) {

        //Crear/Conectarse BD
        AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper(this, "MiBase", null, 1);

        //abrir BD
        SQLiteDatabase BaseDeDatos = cnx.getWritableDatabase();

        //capturo el codigo del producto
        String id = txtid.getText().toString();


       if(!id.isEmpty())
        {
           //ejecutamos la transql
            int cantidad = BaseDeDatos.delete("producto","id ="+ id, null);

            //cerramos BD
            BaseDeDatos.close();


            if(cantidad==1)
            {
                Toast.makeText(this,"Producto Eliminado",Toast.LENGTH_SHORT).show();
                //limpiar campos
                txtid.setText("");
                txtdescripcion.setText("");
                txtprecio.setText("");
                txtid.requestFocus();

            }
            else
            {
                Toast.makeText(this,"Producto no existe",Toast.LENGTH_SHORT).show();
                //limpiar campos
                txtid.setText("");
                txtdescripcion.setText("");
                txtprecio.setText("");
                txtid.requestFocus();
            }


        }
        else
        {
            Toast.makeText(this,"No se ha especificado el ID del producto",Toast.LENGTH_SHORT).show();
        }


    }
```

**********Método 2: sentencias sql

```
public void eliminarproducto(View view) {

        //Crear/Conectarse BD
        AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper(this, "MiBase", null, 1);

        //abrir BD
        SQLiteDatabase BaseDeDatos = cnx.getWritableDatabase();

        //capturo el codigo del producto
        String id = txtid.getText().toString();

if (!id.isEmpty()) {


  try {

      //ejecutamos la transql : delete from producto where id = 1;
      String sql = "delete from producto where id = " + id;
      BaseDeDatos.execSQL(sql);

      //cerramos BD
      BaseDeDatos.close();

      Toast.makeText(this,"Producto Eliminado",Toast.LENGTH_SHORT).show();
      //limpiar campos
      txtid.setText("");
      txtdescripcion.setText("");
      txtprecio.setText("");
      txtid.requestFocus();

  }
  catch (Exception e)
  {
      Toast.makeText(this,"Producto No existe",Toast.LENGTH_SHORT).show();

      //cerramos BD
      BaseDeDatos.close();
  }

}
else
{
    Toast.makeText(this,"No se ha especificado el ID del producto",Toast.LENGTH_SHORT).show();
}

}
```

//Buscar

- Crear/Conectarse BD
- Abrir BD
- Ejecutar SQL
- Cerrar BD

```
public void buscarproducto(View view)
{
    //Crear/Conectarse BD
    AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper(this, "MiBase", null, 1);

    //abrir BD
    SQLiteDatabase BaseDeDatos = cnx.getWritableDatabase();

    //capturo el codigo del producto
    String id = txtid.getText().toString();

    if(!id.isEmpty())
    {
        //ejecutar transql
         Cursor fila = BaseDeDatos.rawQuery("select * from producto where id = "+ id , null);

         if(fila.moveToFirst())
         {
             //mostrar datos en los campos
             txtdescripcion.setText(fila.getString(1));
             txtprecio.setText(fila.getString(2));

             //cerramos BD
             BaseDeDatos.close();
         }
         else
         {
             Toast.makeText(this,"Producto no existe",Toast.LENGTH_SHORT).show();
             txtid.setText("");
             txtdescripcion.setText("");
```

```
            txtprecio.setText("");
            txtid.requestFocus();

            //cerramos BD
            BaseDeDatos.close();

        }

    }
    else
    {
        Toast.makeText(this,"Debe ingresar Id de producto para buscar",Toast.LENGTH_SHORT).show();

    }

}
```

//Contar Registros

- Crear/Conectarse BD
- Abrir BD
- Ejecutar SQL
- Cerrar BD
-

```
public int contarregitros ()
{


    //Crear/Conectarse BD
    AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper(this, "MiBase", null, 1);

    //abrir BD
    SQLiteDatabase BaseDeDatos = cnx.getWritableDatabase();

    //ejecuta sql
    String sql= "select id from producto";
    int num = BaseDeDatos.rawQuery(sql,null).getCount();
    //cerramos BD
    BaseDeDatos.close();


    return num;
}
```

//mostramos el número de registros en un TextView(label) en el evento Create

```
txttotal.setText("Records: "+ contarregitros());
```

//Mostrar Registros – List View(SIMPLE)

- Crear/Conectarse BD
- Abrir BD para lectura
- Ejecutar SQL
- Recorrer lista y llenar ArrayList
- Cerrar BD

//en la Activity2

//Agregamos al Layout el componete ListView (Legacy)(ocupando todo el screen)

//creamos una función que muestre y devuelva una Lista de Datos “ArrayList”

```
public ArrayList mostrarprodutos()
{
```

```
 //conectar/Conectarse BD
    AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper( this, "MiBase", null,1);

    //abrir BD
    SQLiteDatabase BaseDeDatos = cnx.getReadableDatabase();

    //ejecutar sql
    String sql = "Select * from producto";
    Cursor registros = BaseDeDatos.rawQuery(sql,null);



//Recorrer la lista
    ArrayList datos = new ArrayList();


    while(registros.moveToNext())
    {
        //vista lineal
        datos.add(registros.getString(0) + "   \n   "+ registros.getString(1) + "   \n   " + registros.getString(2) );

         //sino se muestra ponemos esto TOAST para destrabar y depsues lo comentamos
          //Toast.makeText(MainActivity3.this, ""+registros.getString(0),Toast.LENGTH_LONG ).show();

    }

     //cerramos BD
    BaseDeDatos.close();


      return  datos;
}
```

//mostramos los datos en el ListView

//creamos las variables globales necesarias

```
ListView listview;


//variables para mostrar
ArrayList lista ;
ArrayAdapter adaptador;
```

//en el evento necesario cargamos los datos al ListView; en este caso en el créate del Activity

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2);

    getSupportActionBar().setTitle("         INVENTARIO EXISTENTE");
    ListaProductos = (ListView) findViewById(R.id.ListView);



    //mostrar datos lisview
    lista = mostrarprodutos();
    adaptador = new ArrayAdapter(this, R.layout.support_simple_spinner_dropdown_item,lista);
    listview.setAdapter(adaptador);


}
```

NOTA: se puede poner unos textvie por encima del listview para mostrar como etiquetas

Seleccionar un Item del Listview

```
//Evento seleccionar un item del lisview
listview.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
        Toast.makeText(MainActivity3.this , ""+mostrarprodutos().get(i), Toast.LENGTH_SHORT).show();


    }
});
```

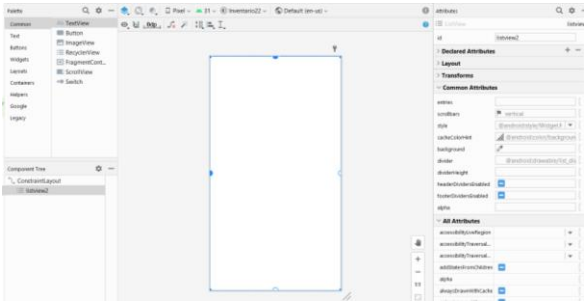
//Mostrar Registros – List View(CLASE)

https://www.youtube.com/watch?v=j1SFFVUt64I

- Crear/Conectarse BD
- Abrir BD para lectura
- Ejecutar SQL
- Recorrer lista y llenar ArrayList
- Cerrar BD

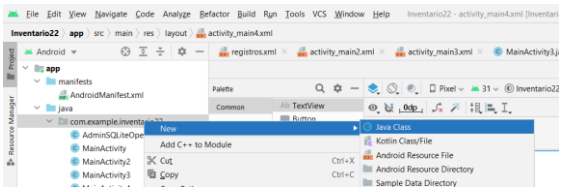
//Craer la Activity#

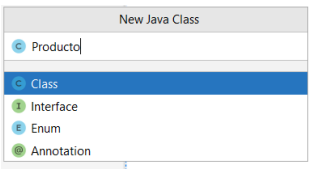
1. Agregamos al Layout el componete ListView (Legacy)(ocupando todo el screen), poner nombre(id) al componente ListView



2. creamos la CLASE correspondiente a la Tabla de la BD a mostrar.
Ejemplo: Producto

Carpeta Java -> com.example.nombreproyecto -> New -> Java Class



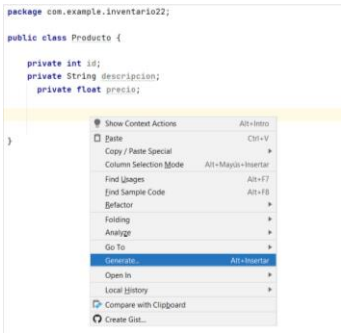


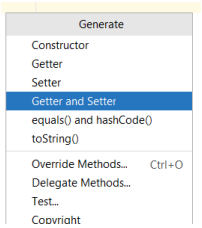
3. Agregamos las Variables locales privadas (campos de la tabla) a la Clase.

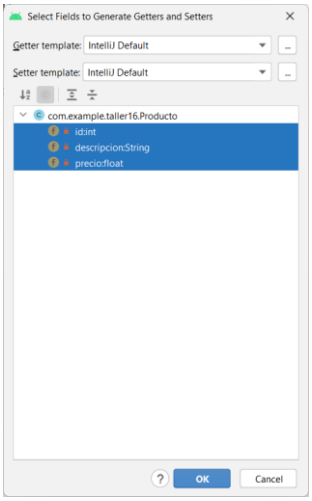
```java
package com.example.inventario22;

public class Producto {

    private int id;
    private String descripcion;
      private float precio;

}
```

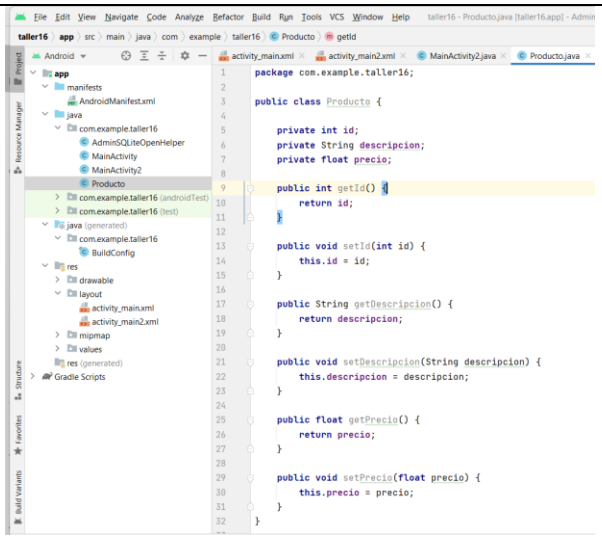
4. Creamos los métodos Get y Set para cada variable (automaticamente)

Cick derecho – Generate









```
package com.example.taller16;

public class Producto {

    private int id;
    private String descripcion;
    private float precio;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

    public float getPrecio() {
        return precio;
    }

    public void setPrecio(float precio) {
        this.precio = precio;
    }
}
```

5. en la MainActivity.java (correspondiente al listview) crear la variables globales

```
ListView listview;

ArrayList<String> ListaInformacion;
ArrayList<Producto> ListaProductos;


//asociar la logica vs grafica en la parte correspondiente
listview = (ListView)findViewById(R.id.listview2);
```

6. Creamos el Metodo mostrarproductos() arriba de la ultima } “métodos propios”

Metodo que permite abstraer los datos de la BD

```
public void mostrarprodutos() {

    //conectar/Conectarse BD
    AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper(this, "MiBase", null, 1);

    //abrir BD
    SQLiteDatabase BaseDeDatos = cnx.getReadableDatabase();

    //ejecutar sql
    String sql = "Select * from producto";
    Cursor cursor = BaseDeDatos.rawQuery(sql, null);

    Producto producto = null; //clase

    //Recorrer la lista
    listaProductos = new ArrayList<Producto>();

    while (cursor.moveToNext()) {
        producto = new Producto();
        producto.setId(cursor.getInt(0));
        producto.setDescripcion(cursor.getString(1));
        producto.setPrecio(cursor.getFloat(2));

        listaProductos.add(producto);
    }

   ListaInformacion = new ArrayList<>();  //suele apaercer error<> pero desaperece al asigar al add al final

   for (int i=0 ; i<ListaProductos.size(); i++ )
   {
       ListaInformacion.add(ListaProductos.get(i).getId()+" - "+ListaProductos.get(i).getDescripcion());
   }

}
```

7. Cargamos los Datos en el ListView (GUI), en evento principal onCreate

```
mostrarprodutos();  //se llena la lista de información
ArrayAdapter adapter = new ArrayAdapter(this, R.layout.support_simple_spinner_dropdown_item,listaInformacion);
listview.setAdapter(adapter);
```

Seleccionar un Item del Listview

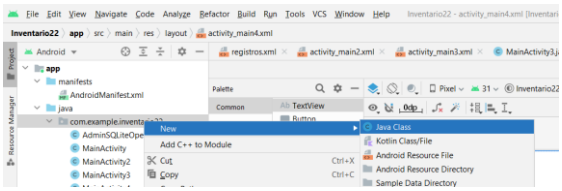
Layout donde se muestra la información (ListView)

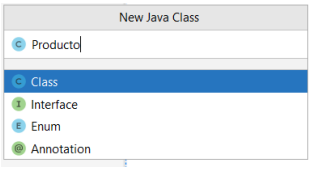
```
//Evento seleccionar un item del lisview
//evento OnCLick Item del ListView
listview.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

        //Variables de Captura
        int ide = listaProductos.get(position).getId();
        String descripcion = listaProductos.get(position).getDescripcion();
        Float precio = listaProductos.get(position).getPrecio();

        //mostrar o pasar como parametros a otro item (uno o todos los valores)
        Toast.makeText(MainActivity3.this,""+ ide , Toast.LENGTH_SHORT).show();

    }
});
```

//Mostrar Registros – (PERSONALIZADO)
Cardview - RecycleView

//// Crear una Clase respecto a la tabla a mostrar. Ejemplo Producto

Carpeta Java -> com.example.nombreproyecto -> New -> Java Class



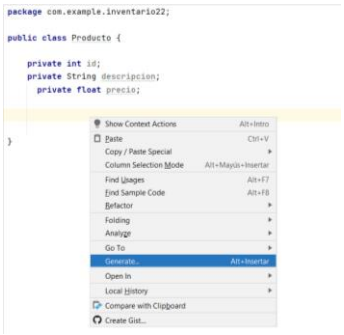


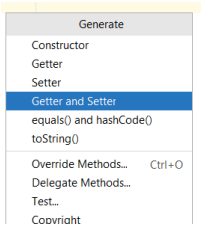
3. Agregamos las Variables locales privadas (campos de la tabla) a la Clase.

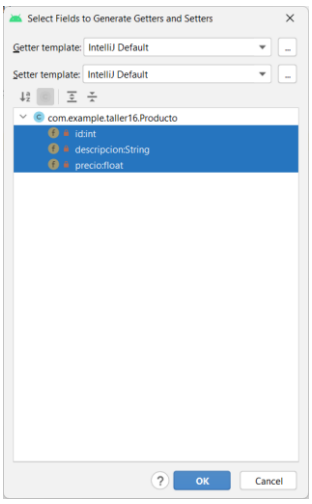
```
package com.example.inventario22;

public class Producto {

    private int id;
    private String descripcion;
      private float precio;

}
```

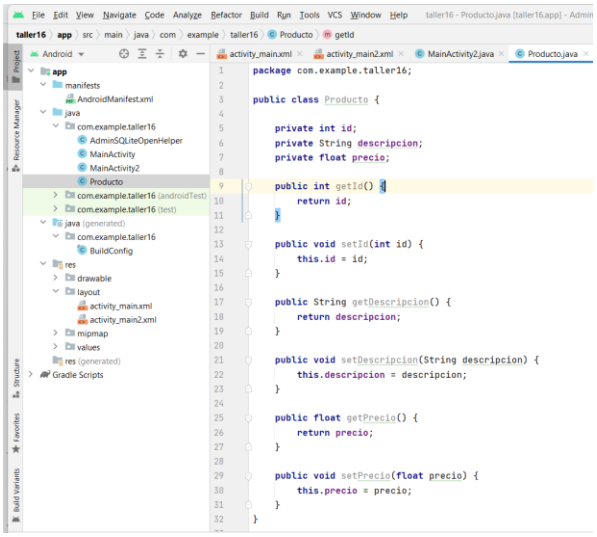
4. Creamos los métodos Get y Set para cada variable (automaticamente)

Cick derecho – Generate





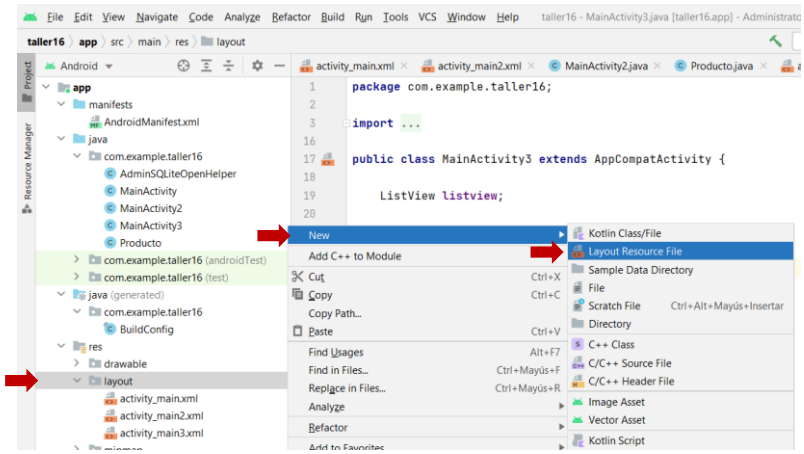


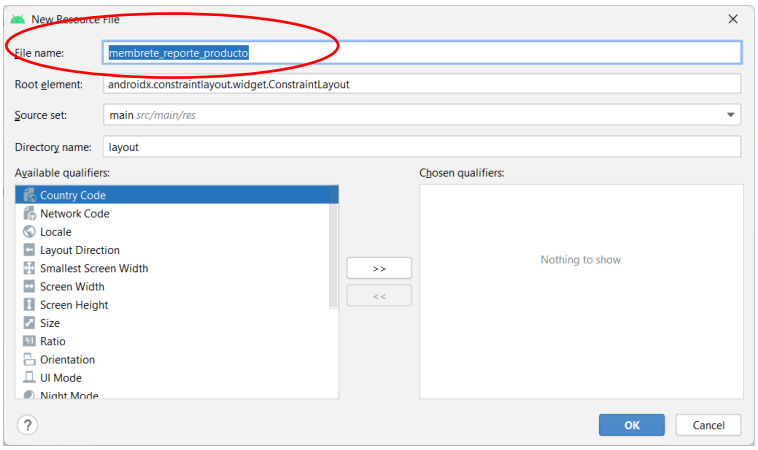


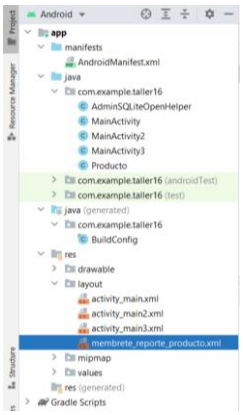
//Crear el Diseño del Reporte en un archivo xml

(Proyecto-Layout-Layout Resource File)
membrete_reporte_producto.xlm

1. Crear el archivo *membrete_reporte_producto.xlm* en la carpeta Layout



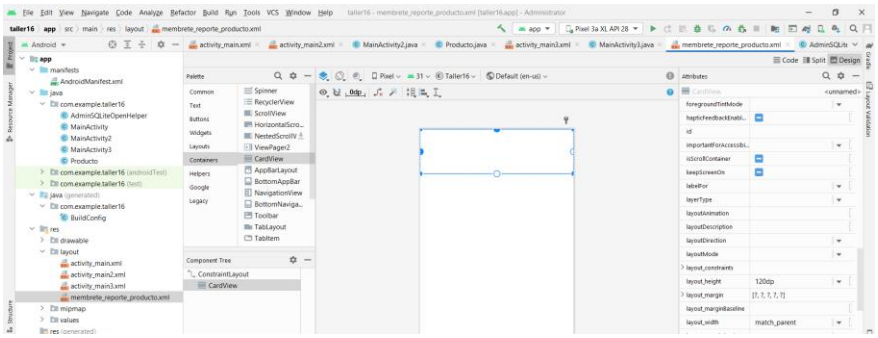




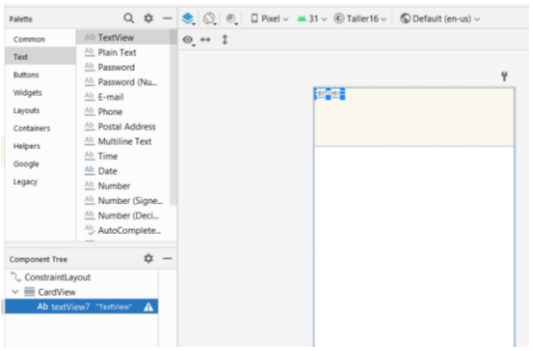
2. En (Design) del archivo *membrete_reporte_producto.xlm* crear el diseño usando un CardView, TextView, ImageView

- Poner un CardView sobre el Layout de *membrete_reporte_producto.xlm fijar coordenaddas de posición y considerar los atributos;*
  layout_width: match_parent

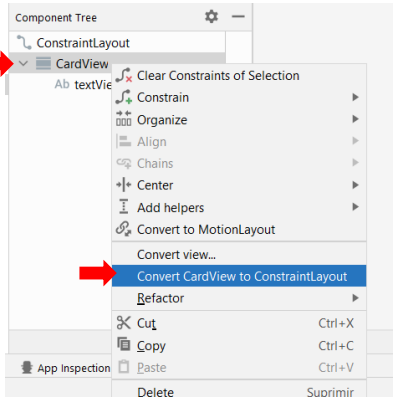
layout_heigth: 120dp
bachgroundTint: ####

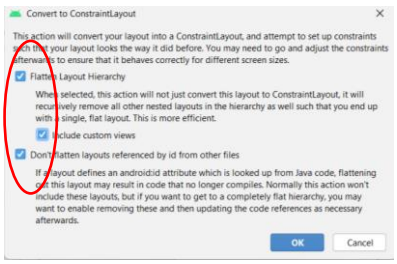


- Poner 1 componente dentro del CardView (TextView) Usar el Arbol de Componentes para poner el componente (Arrastrar)

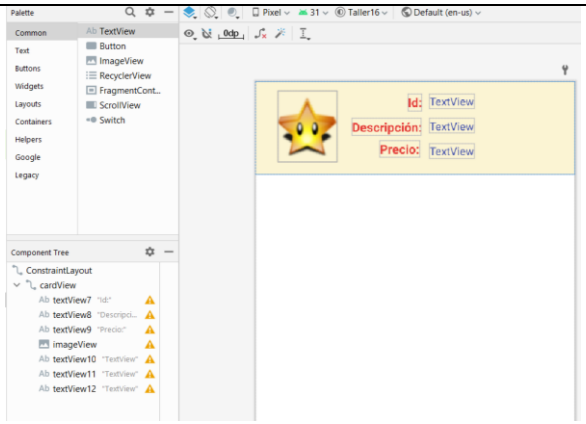


- Convertir CardView en ConstraintLayout

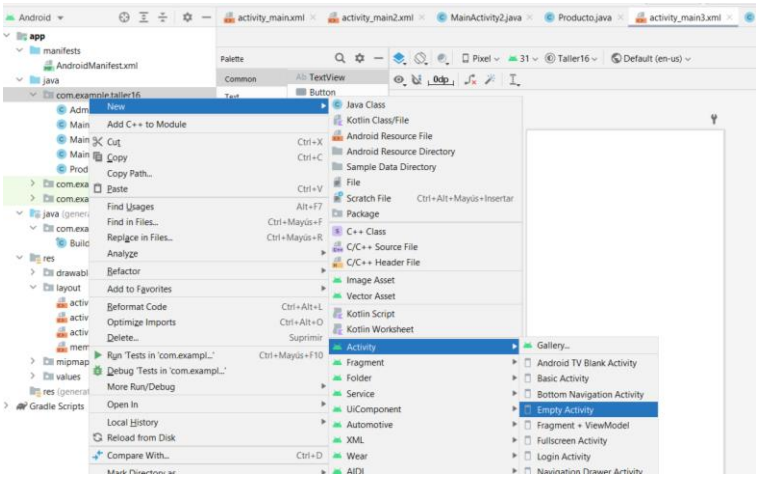




- Agregar el resto de componente al Layout (Arrastrar al Layout directamente) y fijar las coordenas sobre le CardView.

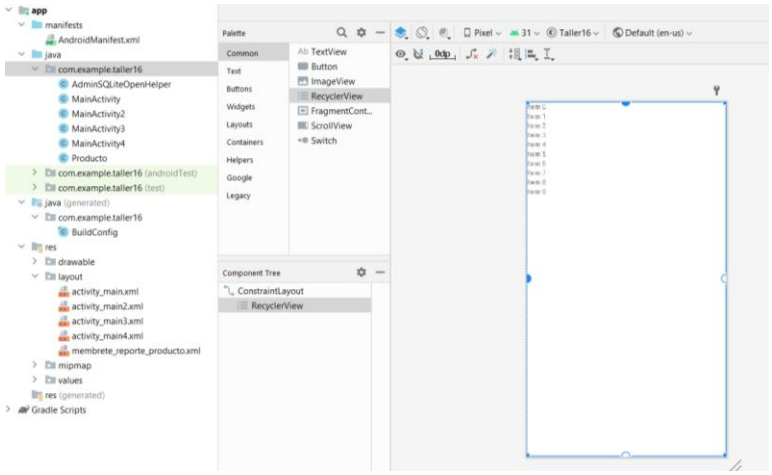


//Crear un Layout (Activity Empty) para mostrar la información (reporte)



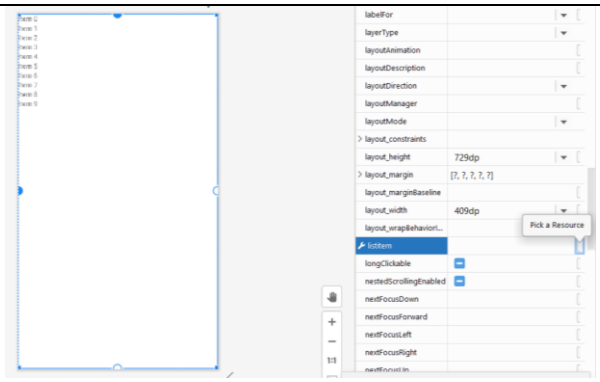
//Colocar una RecyclerView en el Layout que mostrará la información **(coordenadas)**

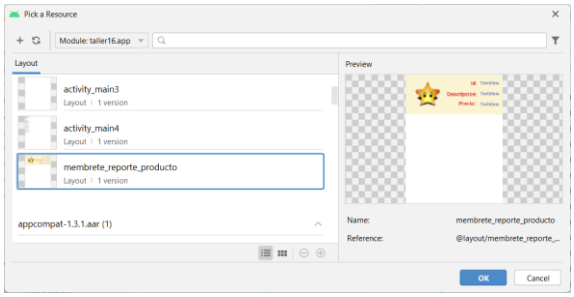
- Poner id al RecyclerView: recyclerview1

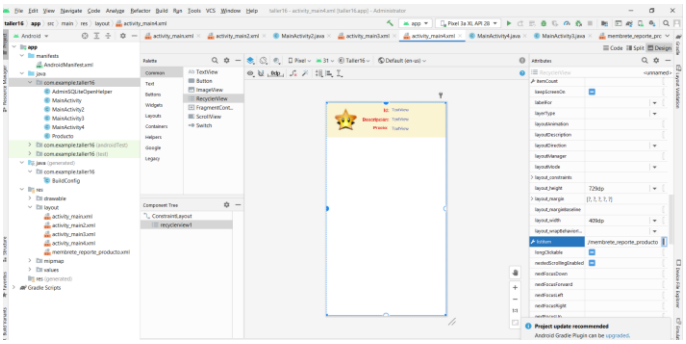


- Cargamos el diseño (*membrete_reporte_producto.xlm*) al RecyclerView

  Atributo de ReciclerVIew: Listitem: *membrete_reporte_producto.xlm*

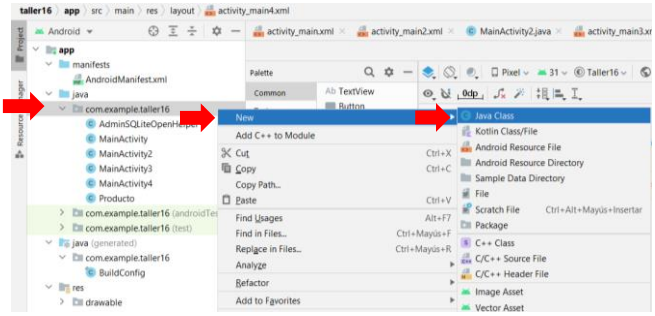


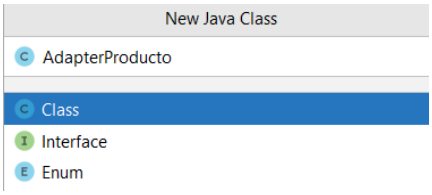


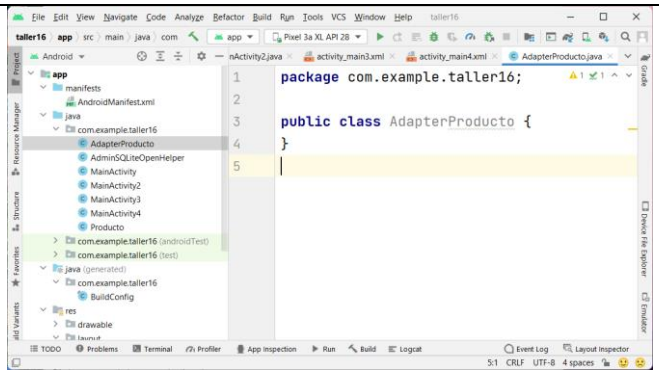


//Crear el **Adaptador** que permitirá configurar la vista de los datos (trayendo de la Clase Producto y poniendolos sobre los componentes del CardView

1. Agregar un Clase al proyecto (carpeta proyecto) : ***AdapterProducto.java***

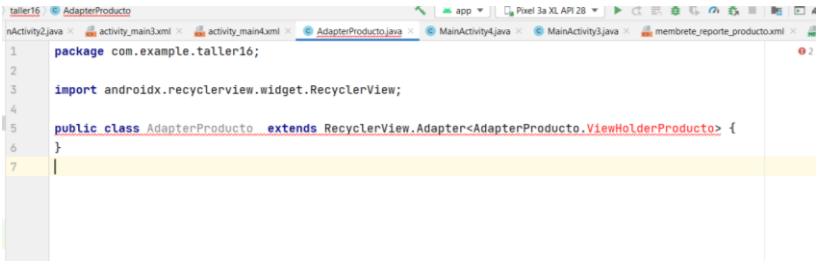




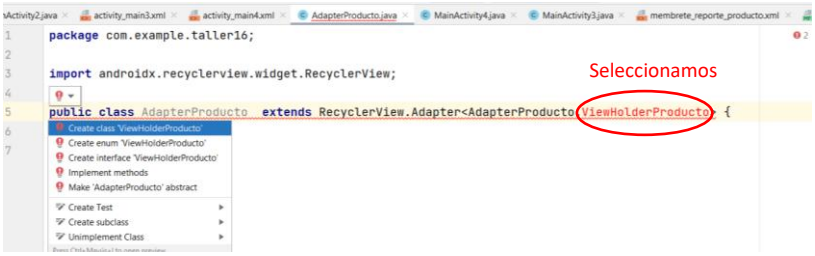


2. Extendemos la clase para que reconozca el RecyclerView y asigamos un nombre al método ViewHolder : **`ViewHolderProducto`**

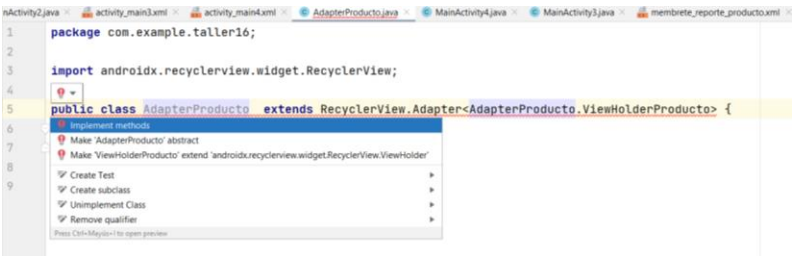
```
extends RecyclerView.Adapter<AdapterProducto.ViewHolderProducto>
```

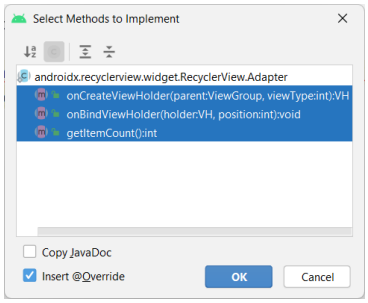


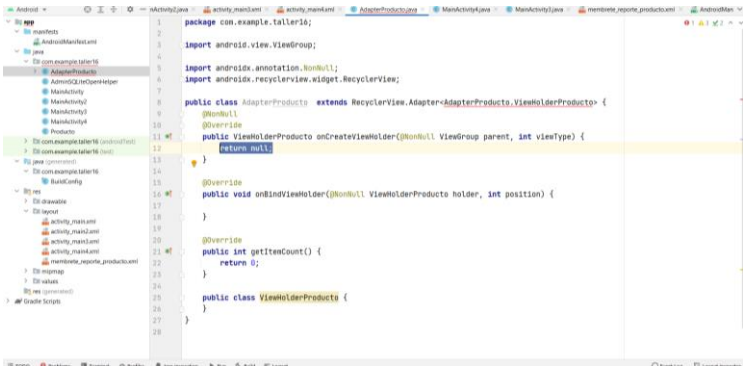
3. Creamos la clase **`ViewHolderProducto`** **usando** 



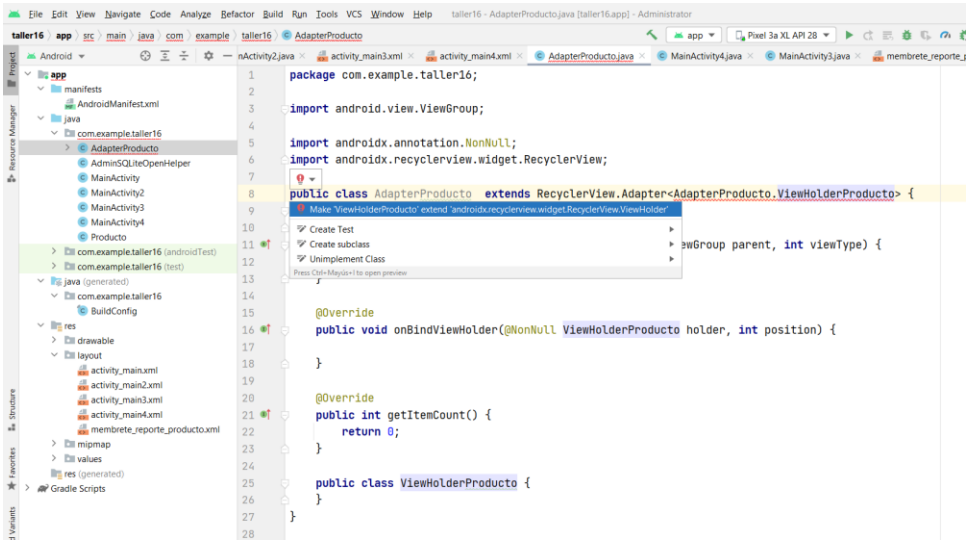
4. Implementamos los metodos correspondinetes a la clase AdapterProducto usando 

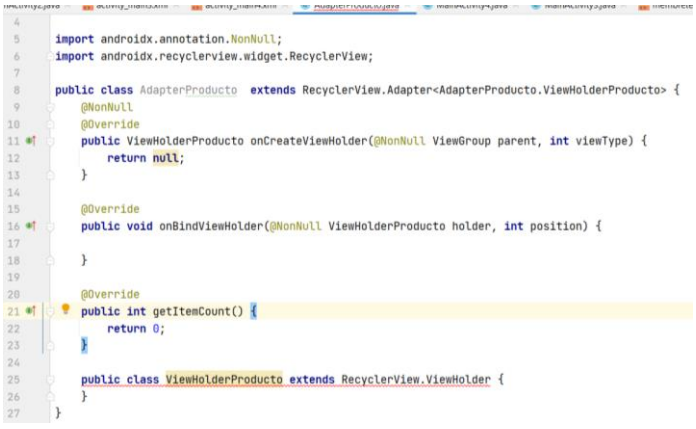




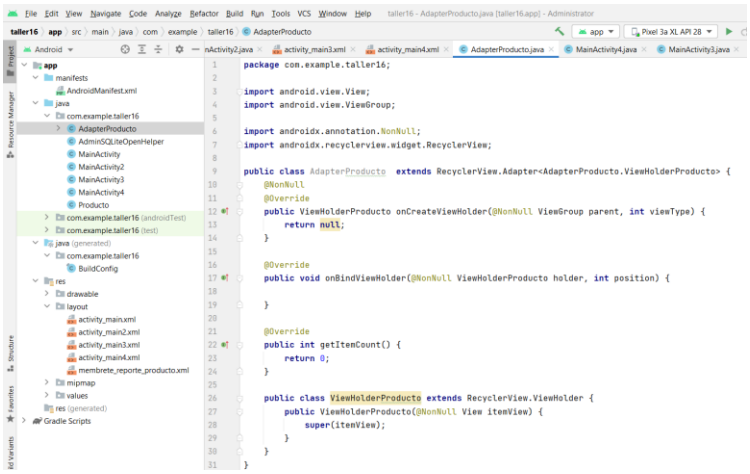


5. Extender la subclase ViewHOlderProducto usando 





6. Crear el Constructor de la clase ViewHolderProducto usando 



7. Crear una variabla global en la Clase Pirncipal AdapterProducto que hace referencia a la clase Producto

```
ArrayList<Producto> ListaProductos;
```

8. Crear el constructor de la clase principal AdapterProducto bajo nuestra variable global (manualmente)

```
public  AdapterProducto(ArrayList<Producto> listaProductos)
{
 this.ListaProductos = listaProductos;

}
```

```java
package com.example.taller16;

import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

public class AdapterProducto  extends RecyclerView.Adapter<AdapterProducto.ViewHolderProducto> {

    //Variable Global
    ArrayList<Producto> ListaProductos;

    public  AdapterProducto(ArrayList<Producto> listaProductos)
    {
     this.ListaProductos = listaProductos;

    }


    @NonNull
    @Override
    public ViewHolderProducto onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        return null;
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolderProducto holder, int position) {
```

9. Llenar el adaptador con campos del diseño
"método `ViewHolderProducto onCreateViewHolder`"

```java
public ViewHolderProducto onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.membrete_reporte_producto,null,false);

    return new ViewHolderProducto(view);
}
```

```java
@NonNull
@Override
public ViewHolderProducto onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.membrete_reporte_producto, root: null, attachToRoot: false);
    return new ViewHolderProducto(view);

}

@Override
public void onBindViewHolder(@NonNull ViewHolderProducto holder, int position) {

}
```

10. Enlazar los campos de la vista de diseño con la Clase Producto/Objetos que se mostraran.(crear variables locales para capturar lo componentes de la vista)
Método "`public class ViewHolderProducto extendsRecyclerView.ViewHolder`"

```java
public class ViewHolderProducto extends RecyclerView.ViewHolder {

    //variables locales
    TextView id,descripcion, precio;


    public ViewHolderProducto(@NonNull View itemView) {
        super(itemView);

        id = (TextView)itemView.findViewById(R.id.textView10);
        descripcion=(TextView) itemView.findViewById(R.id.textView11);
        precio = (TextView) itemView.findViewById(R.id.textView12);



    }
}
```

```
public class ViewHolderProducto extends RecyclerView.ViewHolder {

    //variables locales
    TextView id,descripcion, precio;


    public ViewHolderProducto(@NonNull View itemView) {
        super(itemView);

        id = (TextView)itemView.findViewById(R.id.textView10);
        descripcion=(TextView) itemView.findViewById(R.id.textView11);
        precio = (TextView) itemView.findViewById(R.id.textView12);



    }
}
```

12. Cargamos el total de productos ingresados para que se muestren
Método "**public int** getItemCount()"

```
public int getItemCount() {
    return ListaProductos.size();
}
```

13. Llenamos los Campos de la Vista con los datos del Objeto/Clase que consumen de la base de datos.
Método "**public void** onBindViewHolder"

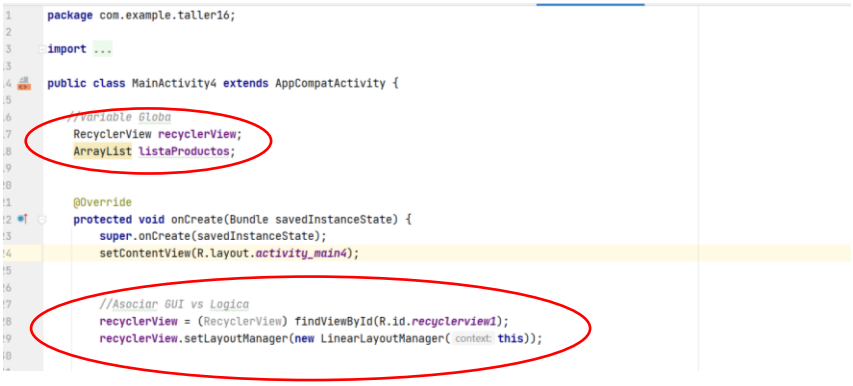
```
public void onBindViewHolder(@NonNull ViewHolderProducto holder, int position) {

  holder.id.setText(String.valueOf(ListaProductos.get(position).getId()));
  holder.descripcion.setText(ListaProductos.get(position).getDescripcion());
  holder.precio.setText(String.valueOf(ListaProductos.get(position).getPrecio()));

}
```

```
@Override
public void onBindViewHolder(@NonNull ViewHolderProducto holder, int position) {

  holder.id.setText(String.valueOf(ListaProductos.get(position).getId()));
  holder.descripcion.setText(ListaProductos.get(position).getDescripcion());
  holder.precio.setText(String.valueOf(ListaProductos.get(position).getPrecio()));

}
```

//Crear el metodo MostrarDatos en el Layout / Acitivity donde esta el RecyclerView

1. Declarar la variable global para el RecyclerView, y la variable para guardar los datos del producto y Asociar a su componte grafico(recycler) e indicar el MangerLayout en donde mostrar

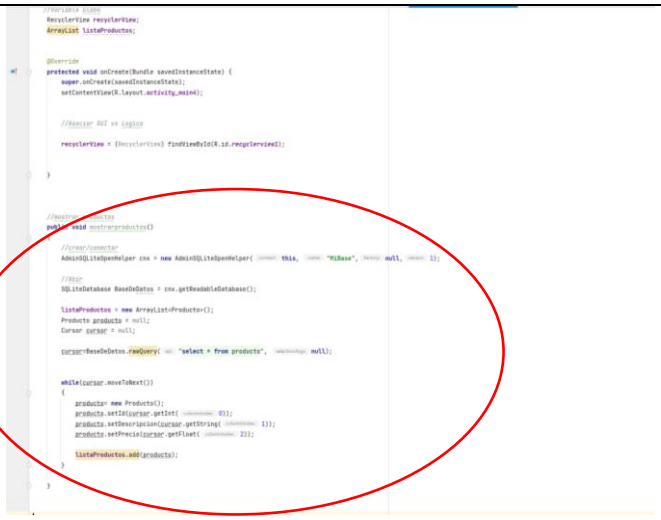
```
//Variable Globa
 RecyclerView recyclerView;
 ArrayList listaProductos;


 //Asociar GUI vs Logica
recyclerView = (RecyclerView) findViewById(R.id.recyclerview1);
recyclerView.setLayoutManager(new LinearLayoutManager(this));
```



```
package com.example.taller16;

import ...

public class MainActivity4 extends AppCompatActivity {

    //Variable Globa
    RecyclerView recyclerView;
    ArrayList listaProductos;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main4);


        //Asociar GUI vs Logica
        recyclerView = (RecyclerView) findViewById(R.id.recyclerview1);
        recyclerView.setLayoutManager(new LinearLayoutManager( context: this));
```

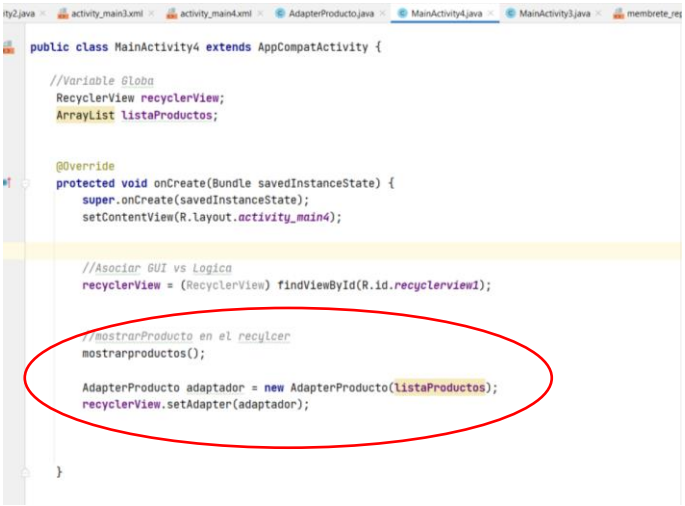
2. Metodo MostrarProductos “método propio”

```
//mostrar productos
public void mostrarproductos()
{
    //crear/conectar
    AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper(this, "MiBase",null,1);

    //Abir
    SQLiteDatabase BaseDeDatos = cnx.getReadableDatabase();

    listaProductos = new ArrayList<Producto>();
    Producto producto = null;
    Cursor cursor = null;

    cursor=BaseDeDatos.rawQuery("select * from producto", null);


    while(cursor.moveToNext())
    {
        producto= new Producto();
        producto.setId(cursor.getInt(0));
        producto.setDescripcion(cursor.getString(1));
        producto.setPrecio(cursor.getFloat(2));

        listaProductos.add(producto);
    }

}
```



3. Sacar los datos al Recycler del LAyout
Clase principal bajo las Asocion de Variables graf vs logica

```
//mostrarProducto en el recylcer
mostrarproductos();

AdapterProducto adaptador = new AdapterProducto(listaProductos);
recyclerView.setAdapter(adaptador);
```



4. Progrmar el evento del botón para llamar al Layout que tiene el Recyclerview

Poner un boto para ver detalles
Xml diseño

Programar el botón en el AdapterProducto

```
public ViewHolderProducto(@NonNull View itemView) {
    super(itemView);

   id = (TextView)itemView.findViewById(R.id.textView10);
   descripcion=(TextView) itemView.findViewById(R.id.textView11);
   precio = (TextView) itemView.findViewById(R.id.textView12);

   btndetalle = (Button) itemView.findViewById(R.id.button6);

   btndetalle.setOnClickListener(new View.OnClickListener() {
       @Override
       public void onClick(View v) {

       //Toast.makeText(id.getContext(), "id."+id.getText().toString(), Toast.LENGTH_SHORT).show();
           //Toast.makeText([redacted], "Aki", Toast.LENGTH_SHORT).show();
           //intent

       }
   });



}
```

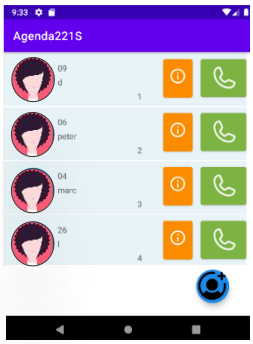
TALLER 18: Lista de Contactos (BD y uso de Camara)

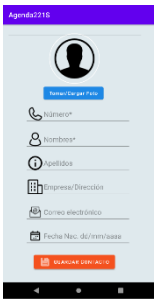
https://www.youtube.com/watch?v=ZmOOMCSVe20

REQUERIMIENTOS

- ✓ Crear un Proyecto "Empty Activity" (Lollipod 5.0)
- ✓ Crear 3 Activity(Layout)
- ✓ Insertar los siguientes componentes
- ✓ ***MainActivity***: Menu de Contactos
  - RecycleView
  - FlotingActionButton
- ✓ ***MainActivity2 : Ficha de Contactos***
  - 6 TextView (Número, Nombres, Apellidos, Empresa, Mail, Fecha Nac,)
  - 1 ImageView (foto)
  - 1 Boton (cargar foto)
  - 1 Boton Guardar datos
  - 1 Button (picker calendario)
- ✓ ***MainActivity3 : Detalle de Contactos***
  - ListView/Recycler(CardView)/WebView
- ✓ ***Implementar un aplicación que permite crear una Agenda de Contactos Telefonicos***
- ✓ ***Usar el SoftwareXXAMP para crear el servidor de BD local***
- ✓ ***Usar avisos o mensajes de alerta para controlar errores.***
- ✓ Agregar un AVD (Virtual Device) para correr la app móvil (optimo)
- ✓ Compilar y Revisar Errores.

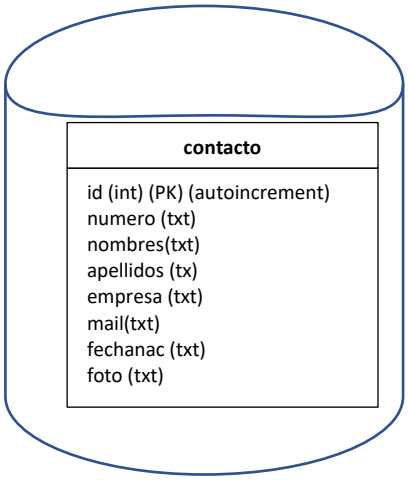
PANTALLAS





DISEÑO DE LA BD "AGENDA"

MiAgenda



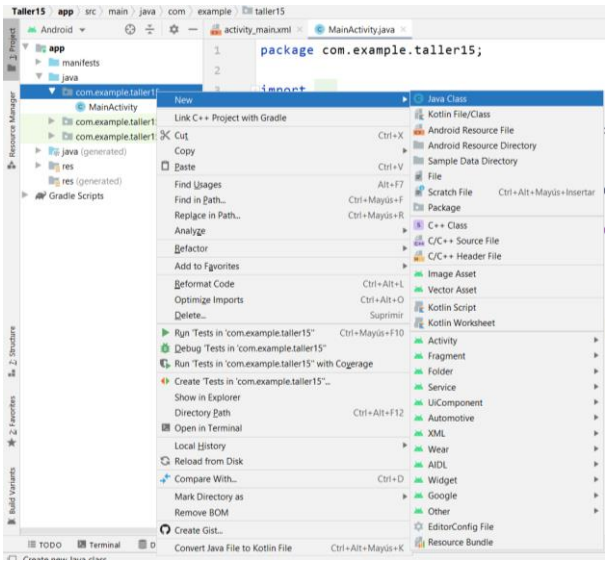
create table contacto (id integer primary key autoincrement, numero text, nombres text, apellidos text, empresa text, mail text, fechanac text, foto text)

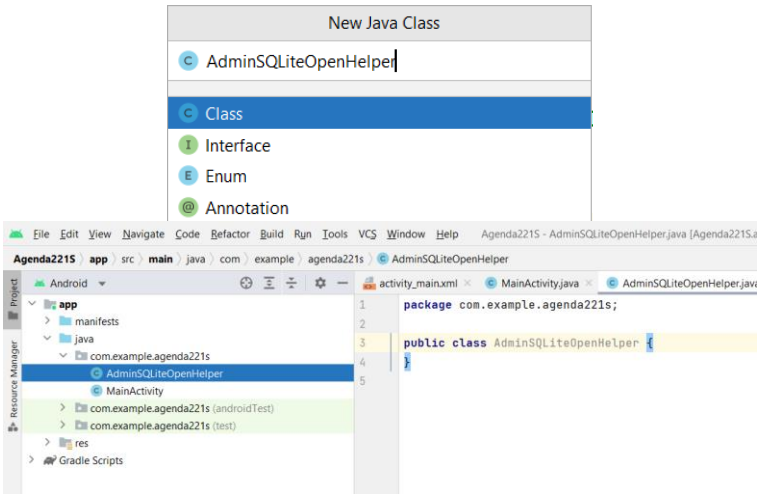
PROCEDIMIENTO

1. Implementar una clase de JAVA para habilitar la librería de conexión a bases de datos SQLite: SQLiteOpenHelper

//Crear clase para conexión

Click derecho (árbol de proyecto)en la carpeta que mantiene la parte lógica del app (java►com…..) New-JavaClass, y asignamos un nombre "AdminSQLiteOpenHelper"





Importamos la librería "import android.database.sqlite.SQLiteOpenHelper"

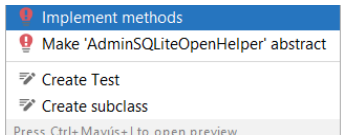
```
package com.example.agenda221s;

import android.database.sqlite.SQLiteOpenHelper;

public class AdminSQLiteOpenHelper {
}
```

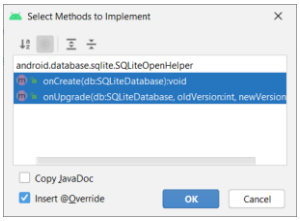
Relacionamos mi clase con la librería importada ***extends***

```
public class AdminSQLiteOpenHelper extends SQLiteOpenHelper{
}
```

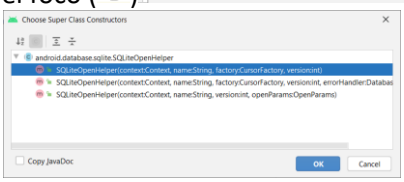
Generamos automáticamente las clases OnCreate y OnUpgrate

Click en el foco ( ) implementar métodos OnCreate y OnUpgrate …OK





Crear el Constructor Click en el foco ( ) Create constructor matching super (4 parámetros) y ok



Quedará asi:

```
package com.example.agenda221s;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

public class AdminSQLiteOpenHelper extends SQLiteOpenHelper {
    public AdminSQLiteOpenHelper(@Nullable Context context, @Nullable String name, @Nullable
SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {

    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

    }
}
```

Le cambianos el nombre de la variable de sqLiteDatabase a db

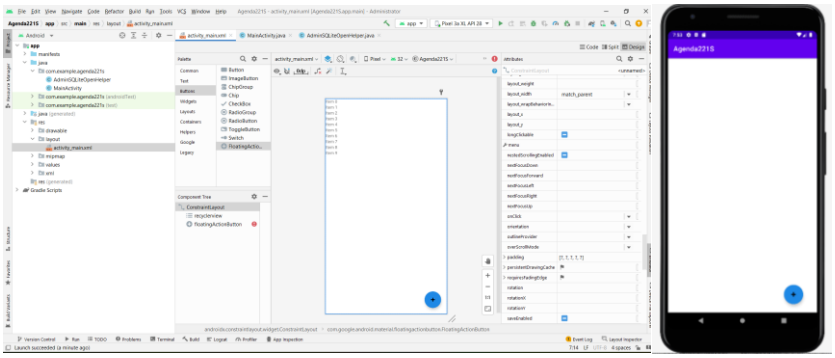
Insertamos la sentencia para crear la tabla “contacto”dentro de la base de datos en el método ***OnCreate***

```
@Override
public void onCreate(SQLiteDatabase db) {

   db.execSQL("create table contacto (id integer primary key autoincrement, numero
text, nombres text, apellidos text, empresa text, mail text, fechanac text, foto
text)");

}
```

2. Diseñar el Interfaz de la app (componentes) , crear variables propias y asignar las componentes a las variables propias.(*MainActivity.java) , (MainActivity2)*

//MainActivity.java

- Agregar un RecycleView (id: recyclerview , coordenadas, maxImagesize, fabcustomesize)
- Agegar un FloatingActionButton (coordenadas)



- **Creamos y Asociamos variables**

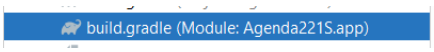
```
public class MainActivity extends AppCompatActivity {

    //varaibales propias
    FloatingActionButton btn;
    RecyclerView recyclerView;


    @Override

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //asociar variables
    btn=(FloatingActionButton) findViewById(R.id.floatingActionButton);
    recyclerView = (RecyclerView) findViewById(R.id.recyclerview);

  }
}
```

//MainActivity2.java

- 7 TextView (Número, Nombres, Apellidos, Empresa, Mail, Fecha Nac, foto)
  propiedades: drawableLef para los iconos
- 1 Boton (cargar foto)
- 1 Boton Guardar datos
- 1 ImageViewCicrular (foto)

Nota: Agregar el CircleImageView

a). Agregar dependencia en Gradle(Module)




```
implementation 'de.hdodenhof:circleimageview:3.1.0'
```

Sync Now

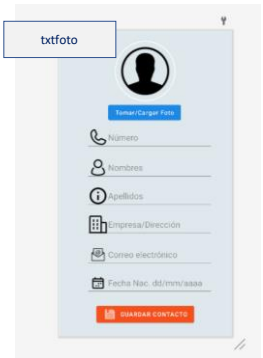
b). Copiar el siguiente código XML para crear el componente activity_main2.xml (CODE)

```
<de.hdodenhof.circleimageview.CircleImageView
xmlns:app="http://schemas.android.com/apk/res-auto"
android:id="@+id/profile_image"
android:layout_width="96dp"
android:layout_height="96dp"
android:src="@drawable/profile"
app:civ_border_width="2dp"
app:civ_border_color="#FF000000"/>
```

retornar a Desing

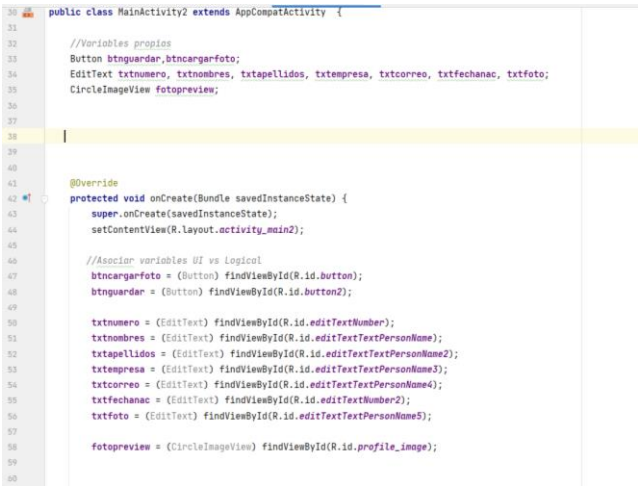
c). Poner coordenas del componente y asiganar una de imagen (src) Color borde (civ_border_color), acho de borde (civ_border_width)

Diseño final



Creamos y Asociamos variables

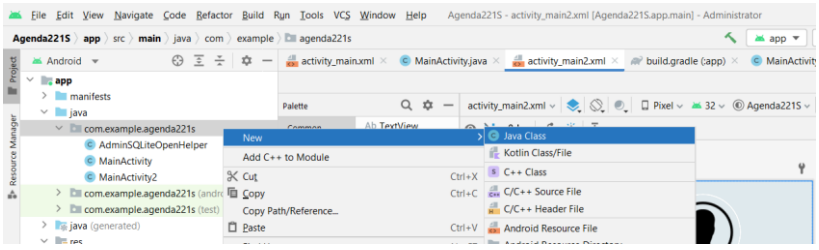
```java
public class MainActivity2 extends AppCompatActivity {

    //Variables propias
    Button btnguardar,btncargarfoto;
    EditText txtnumero, txtnombres, txtapellidos, txtempresa, txtcorreo, txtfechanac, txtfoto;
    CircleImageView fotopreview;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        //Asociar variables UI vs Logical
        btncargarfoto = (Button) findViewById(R.id.button);
        btnguardar = (Button) findViewById(R.id.button2);

        txtnumero = (EditText) findViewById(R.id.editTextNumber);
        txtnombres = (EditText) findViewById(R.id.editTextTextPersonName);
        txtapellidos = (EditText) findViewById(R.id.editTextTextPersonName2);
        txtempresa = (EditText) findViewById(R.id.editTextTextPersonName3);
        txtcorreo = (EditText) findViewById(R.id.editTextTextPersonName4);
        txtfechanac = (EditText) findViewById(R.id.editTextNumber2);
        txtfoto = (EditText) findViewById(R.id.editTextNumber5);

          fotopreview = (CircleImageView) findViewById(R.id.profile_image);


    }
}
```

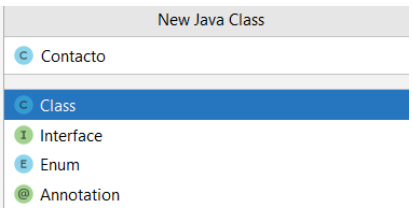


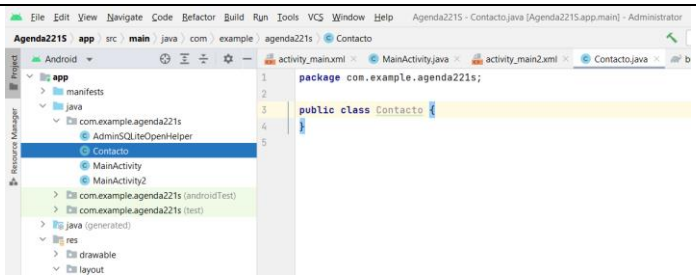
3. Crear la Vista en el RecyclerView (MainActivity1)

a.) Crear una Clase respecto a la tabla a mostrar. Ejemplo Contacto

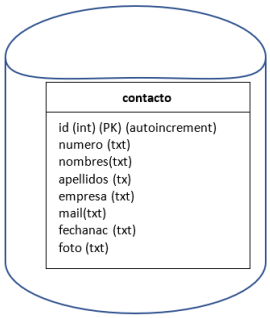
Carpeta Java -> com.example.nombreproyecto -> New -> Java Class







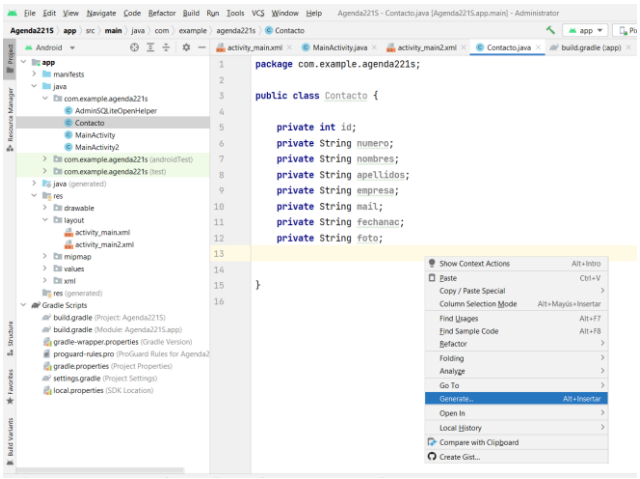
b). Agregamos las Variables locales privadas (campos de la tabla) a la Clase Contacto.

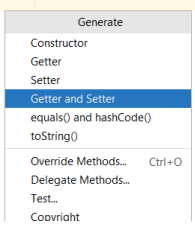


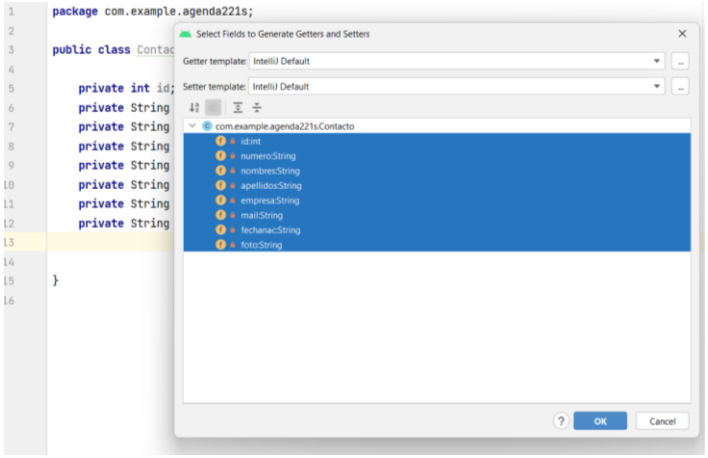

```
package com.example.agenda221s;

public class Contacto {

    private int id;
    private String numero;
    private String nombres;
    private String apellidos;
    private String empresa;
    private String mail;
    private String fechanac;
    private String foto;

}
```

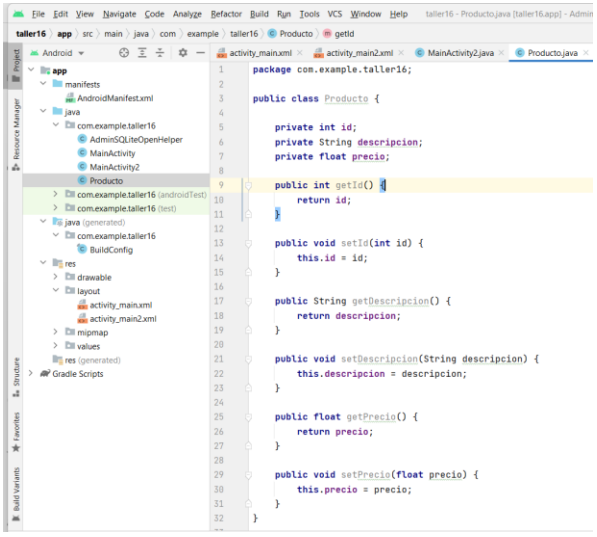
c. Creamos los métodos Get y Set para cada variable (automaticamente)

Cick derecho – Generate



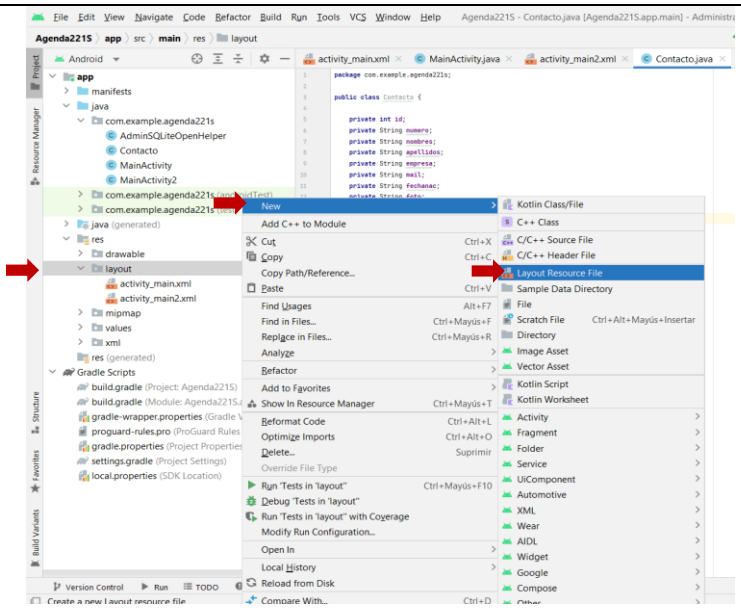


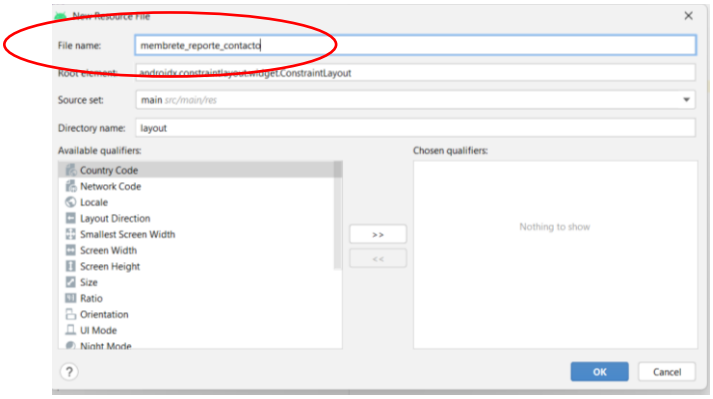


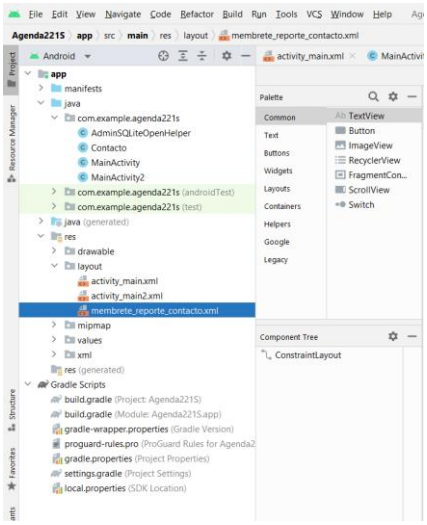


d. Crear el Diseño del Reporte en un archivo xml
(Proyecto-Layout-Layout Resource File)
membrete_reporte_contacto.xml

- Crear el archivo *membrete_reporte_contacto.xml* en la carpeta Layout



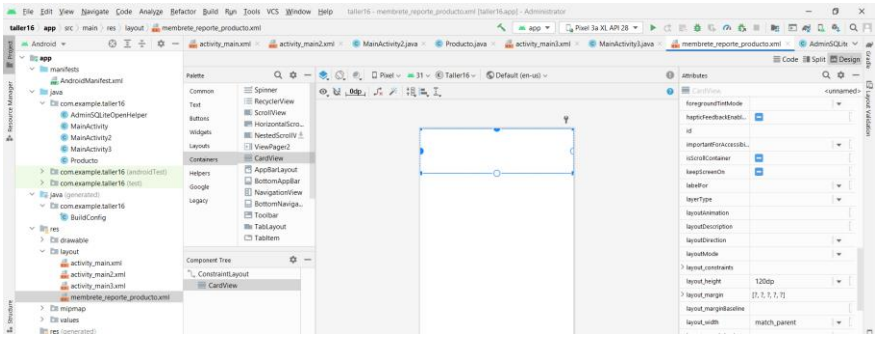




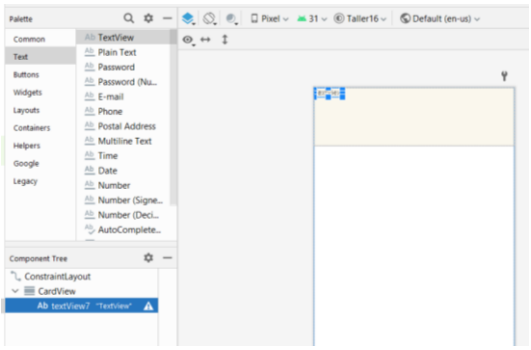
e. En (Design) del archivo *membrete_reporte_contacto.xml* crear el diseño usando un CardView, TextView, ImageView

- Poner un CardView sobre el Layout de *membrete_reporte_contacto.xml fijar coordenaddas de posición y considerar los atributos;*
  layout_width: match_parent
  layout_heigth: 120dp

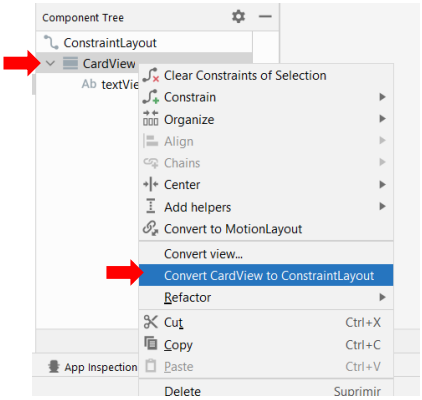
bachgroundTint: ####

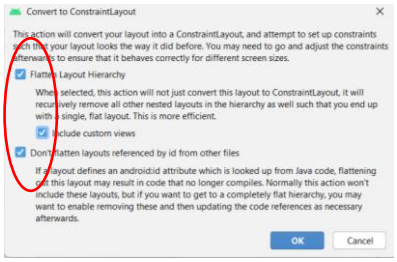


- Poner 1 componente dentro del CardView (TextView) Usar el Arbol de Componentes para poner el componente (Arrastrar)



- Convertir CardView en ConstraintLayout

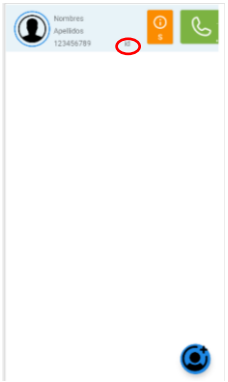




- Agregar el resto de componente al Layout (Arrastrar al Layout directamente) y fijar las coordenas sobre le CardView. (puede usar el ImageCirclerView compiando el código xml correspondiente al componente CODE)

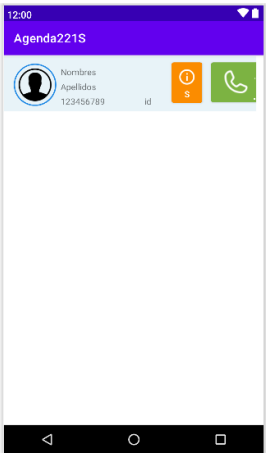
```
<de.hdodenhof.circleimageview.CircleImageView
xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
android:id="@+id/profile_image"
android:layout_width="96dp"
android:layout_height="96dp"
android:src="@drawable/profile"
app:civ_border_width="2dp"
app:civ_border_color="#FF000000"/>
```



- Cargamos el diseño (*membrete_reporte_contacto.xml*) al RecyclerView

  En el MainActivity1►RecyclerView
  Atributo de RecyclerVIew: Listitem: *membrete_reporte_contacto.xml*

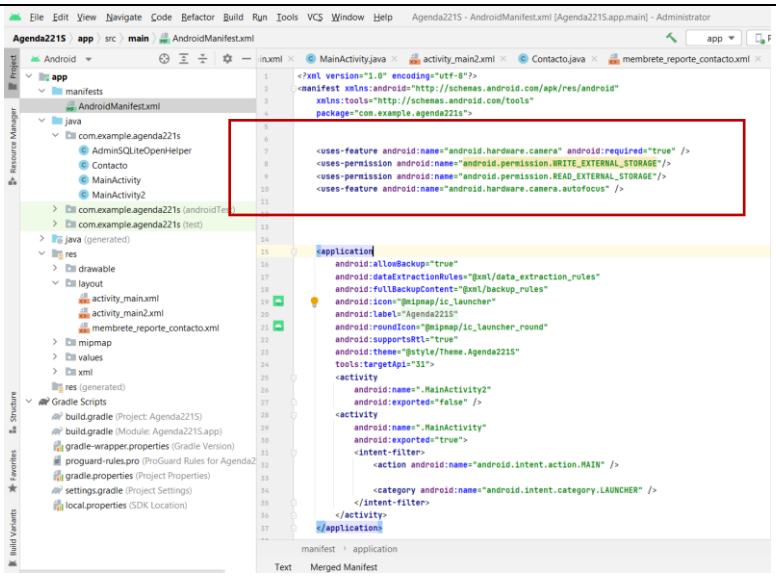


4. Guardar datos en la base de datos SQLite (MainActivity2)

a. Cargar Fotografia (Camara o Galeria) (taller 16)

1. En AndroidManifest habilitar el permiso de uso de camara y memoria interna:

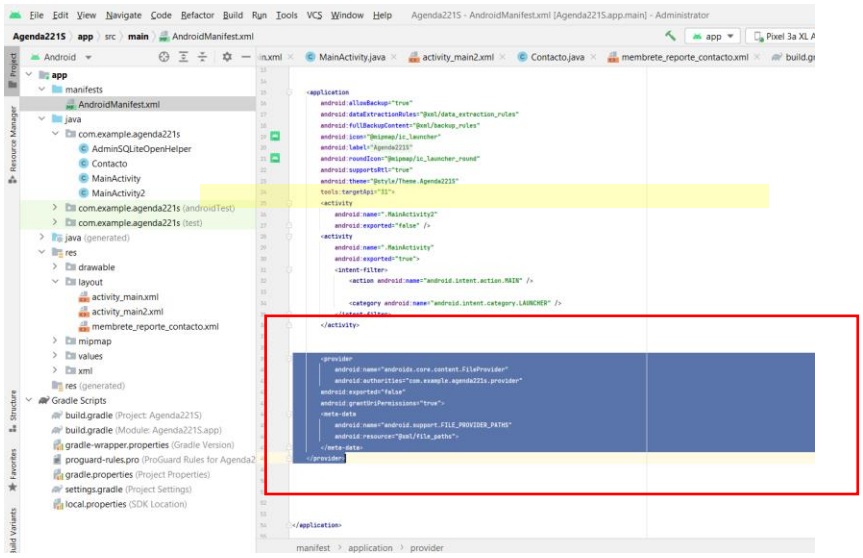
```
<uses-feature android:name="android.hardware.camera" android:required="true" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-feature android:name="android.hardware.camera.autofocus" />
```



2. Incorporar en el AndroidManifest el código para el manejo del almacenamiento interno;
   Despues de `</activity>` y antes del `</application>`

```
<provider
    android:name="androidx.core.content.FileProvider"
    android:authorities="com.example.PROYECTONAME.provider"
    android:exported="false"
    android:grantUriPermissions="true">
<meta-data
    android:name="android.support.FILE_PROVIDER_PATHS"
    android:resource="@xml/file_paths">
</meta-data>
</provider>
```

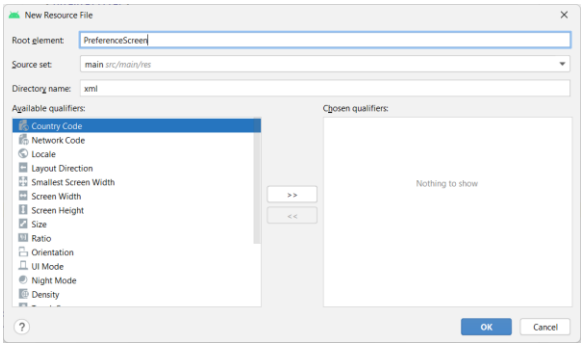
{digitamos provider}



3. Crear el archive file_paths.xml que almacena la dirección donde se almacenaran las fotografías.

- ✓ **Click derecho en el foco rojo o Alt+enter y crear xml…..**

- ✓ **Presionar ok en la siguiente pantalla sin modificar nada**



- ✓ **Se va a crear un activity file_paths.xml, en cual nos cambairemos a vista de código y lo editamos de la siguiente manera: (ver nombre del proyecto)**

```
<?xml version="1.0" encoding="utf-8"?>

<paths xmlns:android="http://schemas.android.com/apk/res/android" >

<external-path
    name="my_images"
    path="Android/data/com.example.camaraaccion/files/Pictures" />
</paths>
```

4. Volvemos al **MainActiviy2.java** (donde esta la foto)y ponemos la línea de código que da permiso al uso de memoria para dispositivos

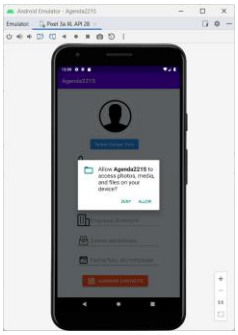
Bajo las variables relacionas

```
btncargarfoto = (Button) findViewById(R.id.button);
fotopreviw =(ImageView) findViewById(R.id.imageView);
```

```
//permiso para uso de memoria para dispositivo
if (ContextCompat.checkSelfPermission(MainActivity2.this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) != PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(MainActivity2.this, Manifest.permission.CAMERA) !=
PackageManager.PERMISSION_GRANTED)
{

    ActivityCompat.requestPermissions(MainActivity2.this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE, Manifest.permission.CAMERA}, 1000);

}
```

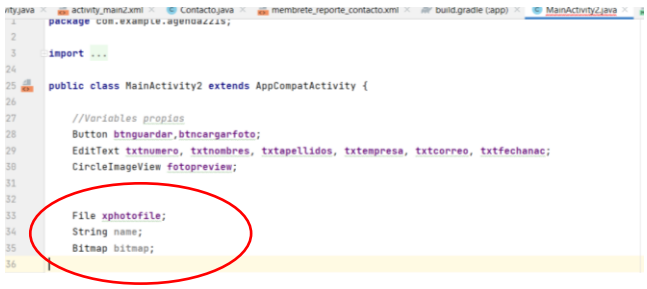
```
alt+intro = para agregar libredias nuevas
```

asi



5. Crear los siguientes métodos propios antes de la **ultima }**:

```
//crear estas variables propias
File xphotofile;
String name;
Bitmap bitmap;
```



```
package com.example.agenda2215;

import ...

public class MainActivity2 extends AppCompatActivity {

    //Variables propias
    Button btnguardar,btncargarfoto;
    EditText txtnumero, txtnombres, txtapellidos, txtempresa, txtcorreo, txtfechanac;
    CircleImageView fotopreview;


    File xphotofile;
    String name;
    Bitmap bitmap;
```

alt+intro = para agregar libredias nuevas

EXPLICACION DE METODOS A CREAER

tomarFoto = Levanta la camara y permite capturar con confirmación
createImageFile = Metodo que construye el nombre de archivo y ubica el path de almacenamiento
onActivityResult = Muestra la vista previa de la foto capturada por la camara

//////////////////////////////// tomarFoto ////////////////////////

```
//metodo tomar foto
static final int REQUEST_TAKE_PHOTO = 1;
public void tomarFoto(View view)
{
    //Levanta la camara
    Toast.makeText(MainActivity2.this,"Levantando la camara", Toast.LENGTH_SHORT).show();
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);


    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        // Create the File where the photo should go
        File photoFile = null;
        try {
           photoFile = createImageFile();
            Toast.makeText(MainActivity2.this,"Creando Archivo", Toast.LENGTH_SHORT).show();
        } catch (IOException ex) {
            // Error occurred while creating the File
        }
        // Continue only if the File was successfully created
        if (photoFile != null) {
            Uri photoURI = FileProvider.getUriForFile(this, "com.example.PROYECTONAME.provider",
photoFile);
            xphotofile= photoFile;
            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);
            startActivityForResult(takePictureIntent, REQUEST_TAKE_PHOTO);
        }

    }
}
```

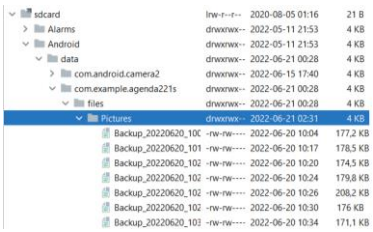
//////////////////////////// createImageFile ////////////////////////////////

```
//metodo crear archivo
    String currentPhotoPath;
    private File createImageFile() throws IOException
    {
        // Create an image file name
        String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
        String imageFileName = "Backup_" + timeStamp + "_";
        File storageDir1 = getExternalFilesDir(Environment.DIRECTORY_PICTURES);
        File image = File.createTempFile(imageFileName,  ".jpg",storageDir1);

        // Save a file: path for use with ACTION_VIEW intents
        currentPhotoPath = image.getAbsolutePath();
        name=currentPhotoPath;

        return image;

    }
```

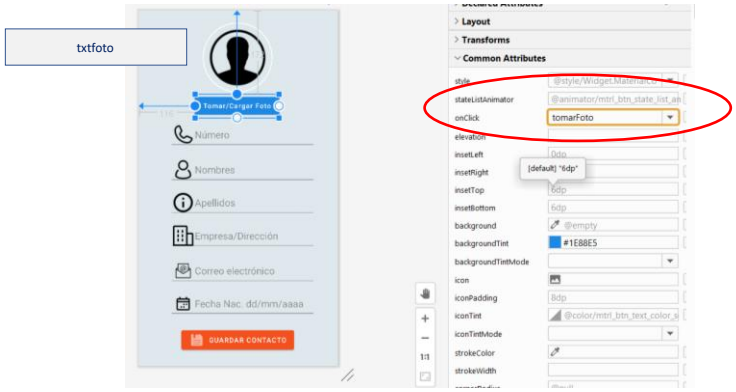
//////////////////////////// onActivityResult ////////////////////////////////

```
//vista previa en el Imview
static final int REQUEST_IMAGE_CAPTURE =1;

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {

        //Trasnforma la foto xphotofile a bitmap y muestra en el view
        Bitmap bitmap = BitmapFactory.decodeFile(xphotofile.getAbsolutePath());
        fotopreview.setImageBitmap(bitmap);

        // guardarFotoGallery();
        btncargarfoto.setEnabled(false);

         // cerrar y llamar a mainactivity 1 (Intent)



    }
    else {
        Toast.makeText(MainActivity2.this,"Cancelo el guardado de la foto", Toast.LENGTH_SHORT).show();
    }


}
```

Nota: las fotografías en el device se guardan en sdcard/android/data/nombreproyecto/files/picture



6. Programar el evento butto onlisterclick btncargarfoto o en el evento Onclick (GUI) poner el evento principal tomarFoto()



b. Alamcenar datos en la base (MainActivity2)

1. Crearemos los métodos Registrar antes de la ultima}

Se pueden crear métodos propios, o usar los métodos OnClickListener
Para trabajar con BD SQLite es importante mantener este orden.

- Crear/Conectarse BD
- Abrir BD
- Ejecutar SQL
- Cerrar BD

//INSERCIÓN

Método 2: sentencias sql

********** METODO 2 : Sentencias Sql **********

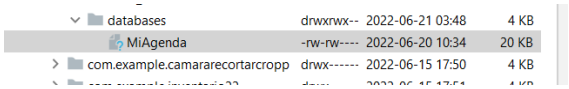
```
//metodo insertar contacto
public void insertacontacto()
{

    //capturo lo que esta en los text
    //id es autonumerico
    String numero = txtnumero.getText().toString();
    String nombres = txtnombres.getText().toString();
    String apellidos = txtapellidos.getText().toString();
    String empresa = txtempresa.getText().toString();
    String correo = txtcorreo.getText().toString();
    String fechanac = txtfechanac.getText().toString();
    String foto = txtfoto.getText().toString();

    if(numero.isEmpty() || nombres.isEmpty())
    {
       txtnumero.setError("Campo Obligatorio");
       txtnombres.setError("Campo Obligatorio");
    }
    else
    {

        //Crear/Conectarse BD
       AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper( this, "MiAgenda", null,1);
```

```
        //abrir BD
        SQLiteDatabase BaseDeDatos = cnx.getWritableDatabase();

        try {
                                                //no se pone ID por que es autonumeric
            String sql="insert into contacto (numero, nombres, apellidos, empresa, mail, fechanac, foto) values('"+ numero
+"','"+ nombres +"','"+ apellidos +"' ,'"+ empresa +"', '"+ correo +"','"+ fechanac +"' ,'"+ foto +"')";

           BaseDeDatos.execSQL(sql);

            //Cerrar BD
            BaseDeDatos.close();
            Toast.makeText(this,"Contacto Creado",Toast.LENGTH_SHORT).show();

               ///*///

            //aqui limpiar los campos
            txtnumero.setText("");
            txtnombres.setText("");
            txtapellidos.setText("");
            txtempresa.setText("");
            txtcorreo.setText("");
            txtfechanac.setText("");
            txtfoto.setText("");
            fotopreview.setImageResource(R.drawable.contact2);


            txtnumero.requestFocus();


        }
        catch (Exception e)
        {
            Toast.makeText(this,"error "+e,Toast.LENGTH_SHORT).show();
            BaseDeDatos.close();
        }

    }

}
```

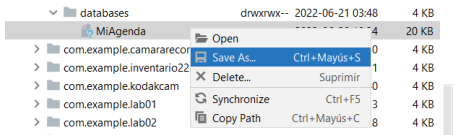
Verificar la Base de Datos usando ***BD Browser for SQLite***

Ir al panel de Explorador de Archivos del Dispositivo ubicado al lado derecho del IDE

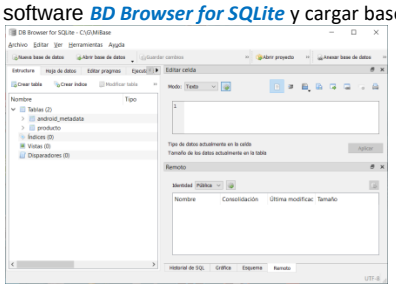
Paht: data▶data▶com.example.**nombreproyecto**▶databases

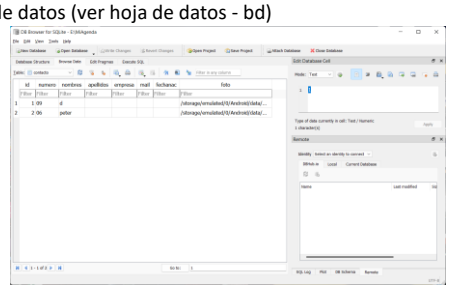


Click derecho SAVE AS



Abrir el software ***BD Browser for SQLite*** y cargar base de datos (ver hoja de datos - bd)



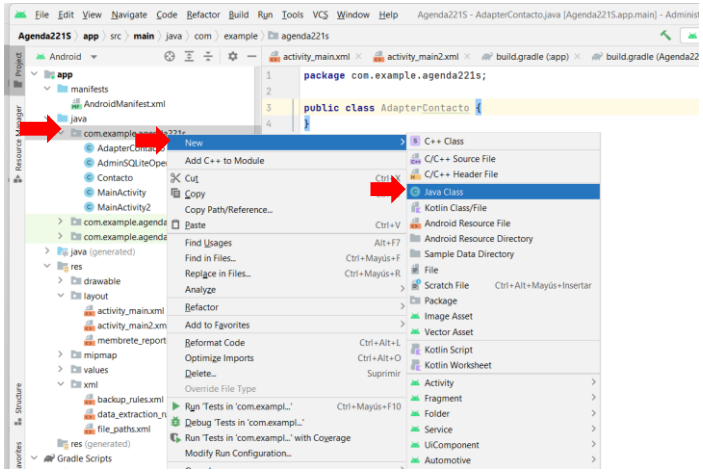


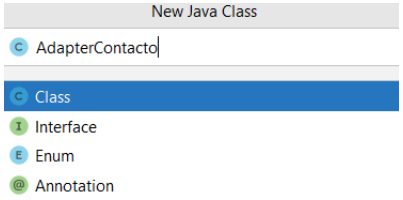
5. Mostrar Datos almacenados DB, en la Pantalla Principal (Recycler View) (MainActivity1)

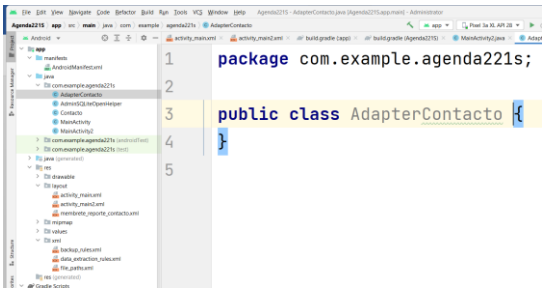
a. Crear el Adaptador (Clase) para consumir datos de la BD SQlite

//Crear el **Adaptador** que permitirá configurar la vista de los datos (trayendo de la Clase Contacto y poniendolos sobre los componentes del CardView

1. Agregar un Clase al proyecto (carpeta proyecto) : ***AdapterContacto.java***

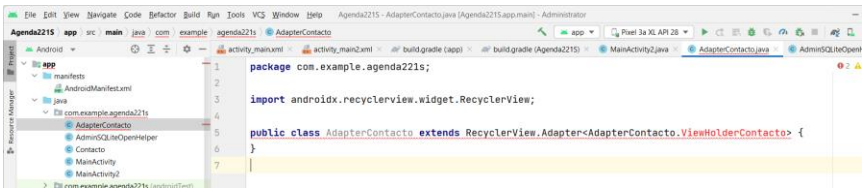




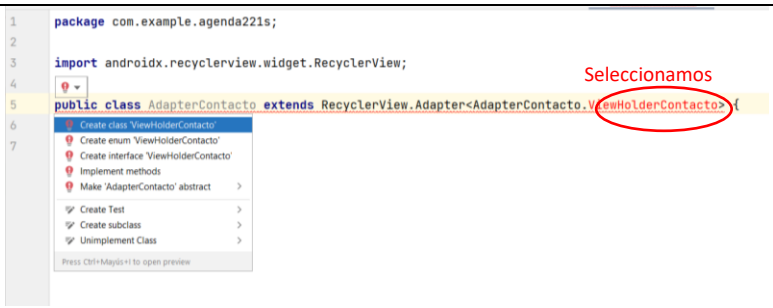


2. Extendemos la clase para que reconozca el RecyclerView y asigamos un nombre al método ViewHolder : **`ViewHolderContacto`**

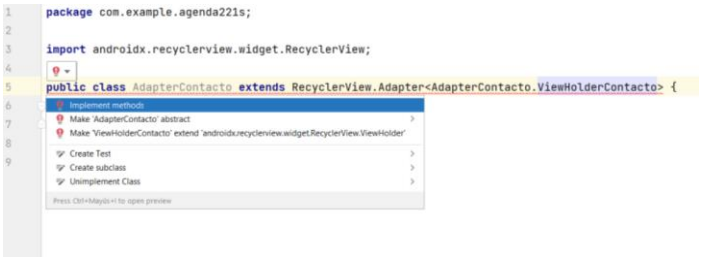
```
extends RecyclerView.Adapter<AdapterContacto.ViewHolderContacto>
```

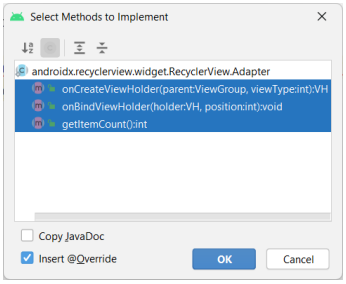


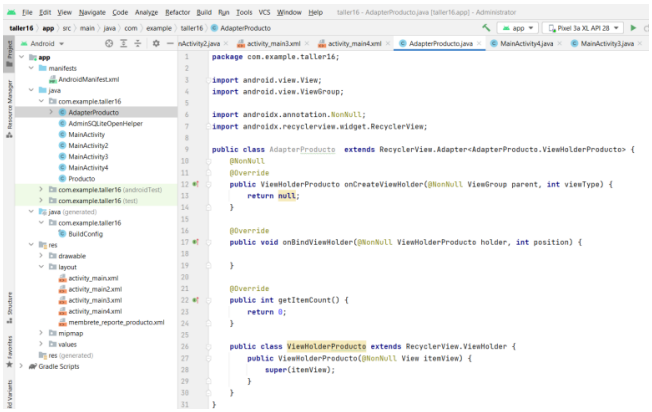
3. Creamos la clase **`ViewHolderCOntacto`** **usando** 



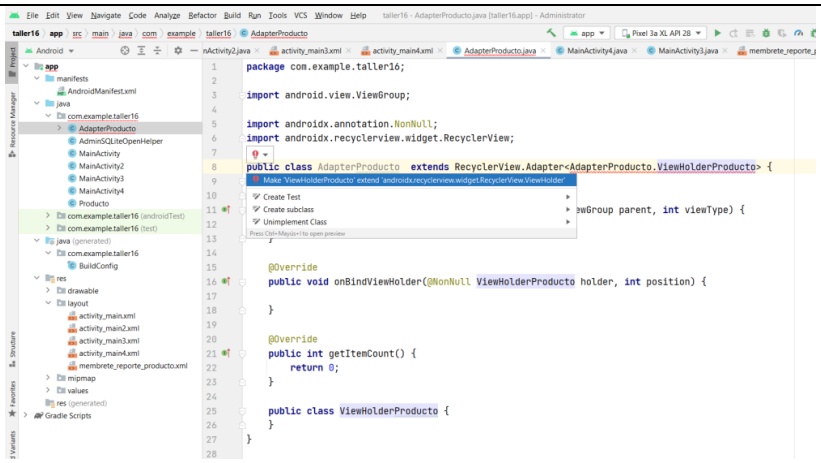
4. Implementamos los metodos correspondinetes a la clase AdapterContacto usando 







5. Extender la subclase ViewHolderContacto usando 

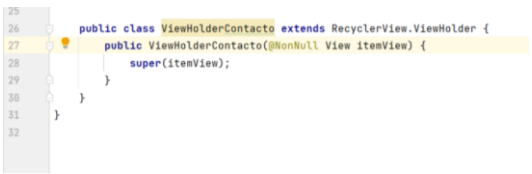


```
package com.example.taller16;

import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

public class AdapterProducto  extends RecyclerView.Adapter<AdapterProducto.ViewHolderProducto> {
    Make 'ViewHolderProducto' extend 'androidx.recyclerview.widget.RecyclerView.ViewHolder'
    Create Test
    Create subclass
    Unimplement Class
    Press Ctrl+Mayús+I to open preview

    @Override
    public void onBindViewHolder(@NonNull ViewHolderProducto holder, int position) {

    }

    @Override
    public int getItemCount() {
        return 0;
    }

    public class ViewHolderProducto {
    }
}
```

```
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

public class AdapterContacto extends RecyclerView.Adapter<AdapterContacto.ViewHolderContacto> {
    @NonNull
    @Override
    public ViewHolderContacto onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        return null;
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolderContacto holder, int position) {

    }

    @Override
    public int getItemCount() {
        return 0;
    }

    public class ViewHolderContacto extends RecyclerView.ViewHolder {
    }
}
```

6. Crear el Constructor de la clase ViewHolderContacto usando 

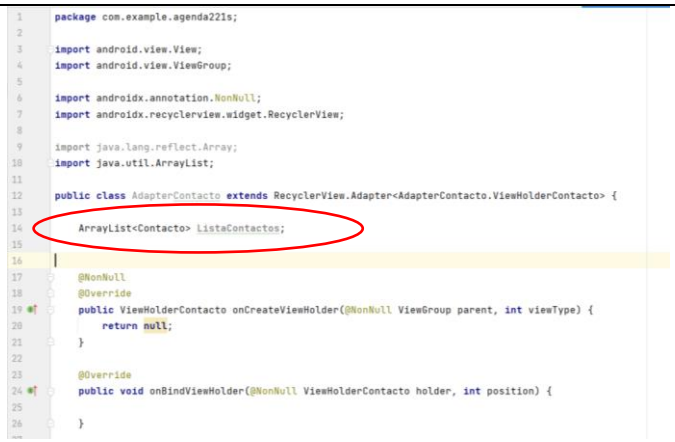
```
    @Override
    public int getItemCount() {
        return 0;
    }

public class ViewHolderContacto extends RecyclerView.ViewHolder {
    Create constructor matching super
    Press Ctrl+Mayús+I to open preview
```



```
    public class ViewHolderContacto extends RecyclerView.ViewHolder {
        public ViewHolderContacto(@NonNull View itemView) {
            super(itemView);
        }
    }
}
```

7. Crear una variabla global en la Clase Pirncipal AdapterContacto que hace referencia a la clase Contacto de tipo ArrayList

```
ArrayList<Contacto> ListaContactos;
```



```
package com.example.agenda221s;

import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.lang.reflect.Array;
import java.util.ArrayList;

public class AdapterContacto extends RecyclerView.Adapter<AdapterContacto.ViewHolderContacto> {

    ArrayList<Contacto> ListaContactos;

    @NonNull
    @Override
    public ViewHolderContacto onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        return null;
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolderContacto holder, int position) {

    }
```

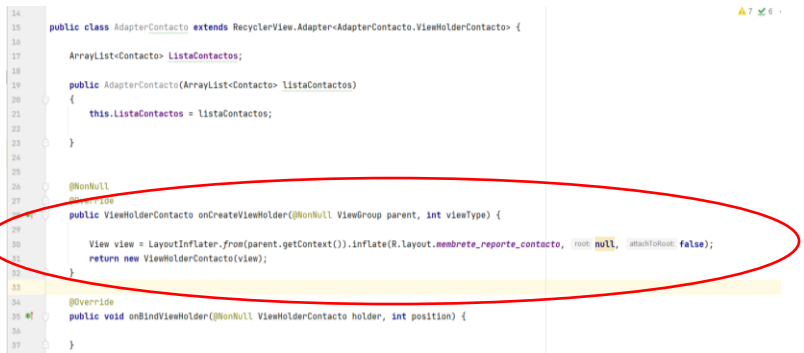
8. Crear el constructor de la clase principal AdapterContacto bajo nuestra variable global (manualmente)

```
Public AdapterContacto(ArrayList<Contacto> listaContactos)
{
 this.ListaContactos = listaContactos;

}
```

```
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.lang.reflect.Array;
import java.util.ArrayList;

public class AdapterContacto extends RecyclerView.Adapter<AdapterContacto.ViewHolderContacto> {

    ArrayList<Contacto> ListaContactos;

    public AdapterContacto(ArrayList<Contacto> listaContactos)
    {
        this.ListaContactos = listaContactos;

    }

    @NonNull
    @Override
```

9. Llenar el adaptador con campos del diseño
 “método ViewHolderContacto onCreateViewHolder”

```
public ViewHolderContacto onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.membrete_reporte_contacto, null, false);
    return new ViewHolderContacto(view);
}
```



```
public class AdapterContacto extends RecyclerView.Adapter<AdapterContacto.ViewHolderContacto> {

    ArrayList<Contacto> ListaContactos;

    public AdapterContacto(ArrayList<Contacto> listaContactos)
    {
        this.ListaContactos = listaContactos;

    }

    @NonNull
    @Override
    public ViewHolderContacto onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.membrete_reporte_contacto, root: null, attachToRoot: false);
        return new ViewHolderContacto(view);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolderContacto holder, int position) {

    }
```

10. Enlazar los campos de la vista de diseño con la Clase Contacto/Objetos que se mostraran.(crear variables locales para capturar los componentes de la vista)

Método "`public class ViewHolderContacto extendsRecyclerView.ViewHolder`"

```
public class ViewHolderContacto extends RecyclerView.ViewHolder {

    //variables locales respecto a los componentes
    TextView id, numero, nombres, apellidos, empresa, mail, fechanac;
    CircleImageView foto;

    public ViewHolderContacto(@NonNull View itemView) {
        super(itemView);

        // ascoair variables logicas vs.variables del cardview
        id = (TextView) itemView.findViewById(R.id.textView4);
        numero = (TextView) itemView.findViewById(R.id.textView);
        nombres = (TextView) itemView.findViewById(R.id.textView2);
        apellidos = (TextView) itemView.findViewById(R.id.textView3);
        foto = (CircleImageView) itemView.findViewById(R.id.profile_image);


    }
}
```

```
public class ViewHolderContacto extends RecyclerView.ViewHolder {

    //variables locales respecto a los componentes del cardview
    TextView id, numero, nombres, apellidos, empresa, mail, fechanac;
    CircleImageView foto;

    public ViewHolderContacto(@NonNull View itemView) {
        super(itemView);

        //asociar variables logigas vs Graficas del Cardview
        id = (TextView)itemView.findViewById(R.id.textView4);
        numero = (TextView) itemView.findViewById(R.id.textView);
        nombres = (TextView) itemView.findViewById(R.id.textView2);
        apellidos = (TextView) itemView.findViewById(R.id.textView3);
        foto = (CircleImageView) itemView.findViewById(R.id.profile_image);

    }
}
}
```

11. Cargamos el total de Contactos ingresados para que se muestren
Método "`public int getItemCount()`"

```
public int getItemCount() {
    return ListaContactos.size();
}
```

```
@Override
public void onBindViewHolder(@NonNull ViewHolderContacto holder, int position) {

}

@Override
public int getItemCount() {
    return ListaContactos.size();
}

public class ViewHolderContacto extends RecyclerView.ViewHolder {

    //variables locales respecto a los componentes del cardview
    TextView id, numero, nombres, apellidos, empresa, mail, fechanac;
    CircleImageView foto;
```

12. Llenamos los Campos de la Vista con los datos del Objeto/Clase que consumen de la base de datos.
Método "`public void onBindViewHolder`"

```
public void onBindViewHolder(@NonNull ViewHolderContacto holder, int position) {

    holder.id.setText(String.valueOf(ListaContactos.get(position).getId()));
    holder.numero.setText(ListaContactos.get(position).getNumero());
    holder.nombres.setText(ListaContactos.get(position).getNombres());
    holder.apellidos.setText(ListaContactos.get(position).getApellidos());
    holder.foto.setImageURI(Uri.parse(ListaContactos.get(position).getFoto()));

}
```

```java
@Override
public void onBindViewHolder(@NonNull ViewHolderContacto holder, int position) {

    holder.id.setText(String.valueOf(ListaContactos.get(position).getId()));
    holder.numero.setText(ListaContactos.get(position).getNumero());
    holder.nombres.setText(ListaContactos.get(position).getNombres());
    holder.apellidos.setText(ListaContactos.get(position).getApellidos());
    holder.foto.setImageURI(Uri.parse(ListaContactos.get(position).getFoto()));

}
```

b. Crear el Metodo Mostrar Datos (MainActivity1)

//Crear el metodo MostrarDatos en el Layout / Acitivity donde esta el RecyclerView

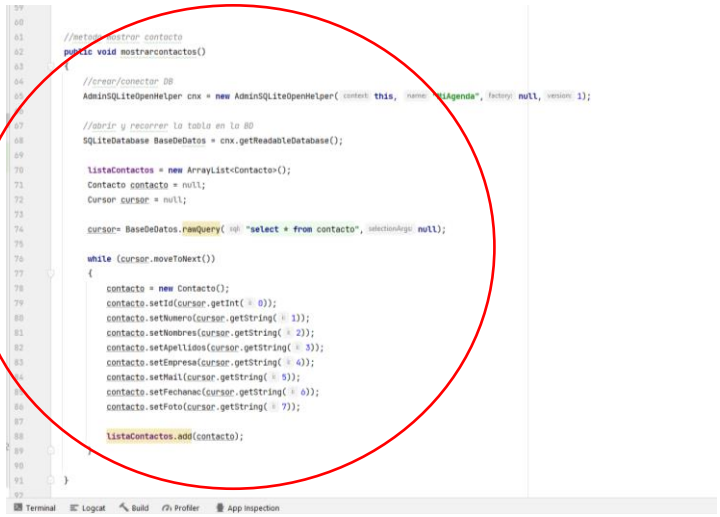
1. Declarar la variable global para el RecyclerView, y la variable para guardar los datos del contacto y Asociar a su componte grafico(recycler) e indicar el MangerLayout en donde mostrar

```java
//Variable Global para recyclerview
 RecyclerView recyclerView;
 ArrayList listaProductos;


 //Asociar GUI vs Logica
recyclerView = (RecyclerView) findViewById(R.id.recyclerview1);
recyclerView.setLayoutManager(new LinearLayoutManager(this));
```

```java
package com.example.agenda221s;

import ...


public class MainActivity extends AppCompatActivity {

    //varaibales propias
    FloatingActionButton btn;
    RecyclerView recyclerView;

    //variable arraylist para el recycler
    ArrayList listaContactos;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //asociar variables
        btn=(FloatingActionButton) findViewById(R.id.floatingActionButton);
        recyclerView = (RecyclerView) findViewById(R.id.recyclerview);

        recyclerView.setLayoutManager(new LinearLayoutManager( context: this));
```

2. Metodo MostrarContacto() “método propio” antes de la última }

```java
//metodo mostrar contacto
public void mostrarcontactos()
{
    //crear/conectar DB
    AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper(this, "MiAgenda",null,1);

    //abrir y recorrer la tabla en la BD
    SQLiteDatabase BaseDeDatos = cnx.getReadableDatabase();

     listaContactos = new ArrayList<Contacto>();
     Contacto contacto = null;
     Cursor cursor = null;

     cursor= BaseDeDatos.rawQuery("select * from contacto",null);

     while (cursor.moveToNext())
     {
         contacto = new Contacto();
         contacto.setId(cursor.getInt(0));
         contacto.setNumero(cursor.getString(1));
         contacto.setNombres(cursor.getString(2));
         contacto.setApellidos(cursor.getString(3));
         contacto.setEmpresa(cursor.getString(4));
         contacto.setMail(cursor.getString(5));
         contacto.setFechanac(cursor.getString(6));
         contacto.setFoto(cursor.getString(7));

         listaContactos.add(contacto);
     }

}
```



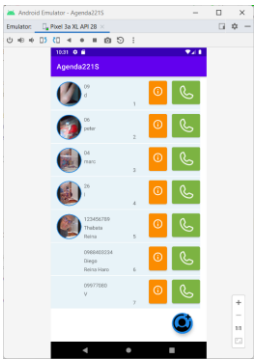
```
//metodo mostrar contacto
public void mostrarcontactos()
{
    //crear/conectar DB
    AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper( context: this,  name: "BDAgenda",  factory: null,  version: 1);

    //abrir y recorrer la tabla en la BD
    SQLiteDatabase BaseDeDatos = cnx.getReadableDatabase();

     listaContactos = new ArrayList<Contacto>();
     Contacto contacto = null;
     Cursor cursor = null;

     cursor= BaseDeDatos.rawQuery( sql: "select * from contacto",  selectionArgs: null);

     while (cursor.moveToNext())
     {
         contacto = new Contacto();
         contacto.setId(cursor.getInt( i: 0));
         contacto.setNumero(cursor.getString( i: 1));
         contacto.setNombres(cursor.getString( i: 2));
         contacto.setApellidos(cursor.getString( i: 3));
         contacto.setEmpresa(cursor.getString( i: 4));
         contacto.setMail(cursor.getString( i: 5));
         contacto.setFechanac(cursor.getString( i: 6));
         contacto.setFoto(cursor.getString( i: 7));

         listaContactos.add(contacto);
     }
}
```

Terminal Logcat Build Profiler App Inspection

3. Sacar los datos al Recycler del LAyout
Clase principal bajo la Asocision de Variables graf vs logica

```
//mostrar COntacto
mostrarcontactos();
AdapterContacto adapatador = new AdapterContacto(listaContactos);
recyclerView.setAdapter(adapatador);
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

//asociar variables
btn=(FloatingActionButton) findViewById(R.id.floatingActionButton);
recyclerView = (RecyclerView) findViewById(R.id.recyclerview);

recyclerView.setLayoutManager(new LinearLayoutManager( context: this));


//mostrar COntacto
mostrarcontactos();
AdapterContacto adapatador = new AdapterContacto(listaContactos);
recyclerView.setAdapter(adapatador);


//boton flotante
btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent formulario = new Intent( packageContext: MainActivity.this, MainActivity2.class);
        startActivity(formulario);

    }
});


}//penultimo
```



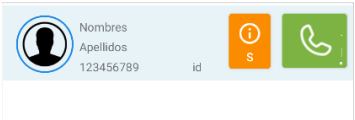
Nota: Al regresar del MainActivity2(Formulario) al MainActivity1(Agenda), para actualizar la lsita de contactos se puede agregar el código de “Recuperar Instance” ponerlo antes del evento OnCreate

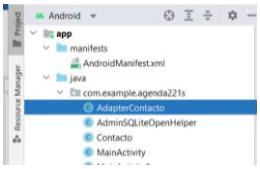
```
if (savedInstanceState != null) {
    //mostrar COntacto
    mostrarcontactos();
    AdapterContacto adapatador = new AdapterContacto(listaContactos);
    recyclerView.setAdapter(adapatador);
}
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    if (savedInstanceState != null) {
        //mostrar COntacto
        mostrarcontactos();
        AdapterContacto adapatador = new AdapterContacto(listaContactos);
        recyclerView.setAdapter(adapatador);
    }

    setContentView(R.layout.activity_main);

    //asociar variables
    btn=(FloatingActionButton) findViewById(R.id.floatingActionButton);
    recyclerView = (RecyclerView) findViewById(R.id.recyclerview);
```

c. Programar el Boton “Llamada Telefonica”

Clase AdapterContacto Metodo `ViewHolderProducto`





Declarar varibales respecto a los botonos a usar y asociar componentes

```
Button btnllamar, btninfocontacto;
```

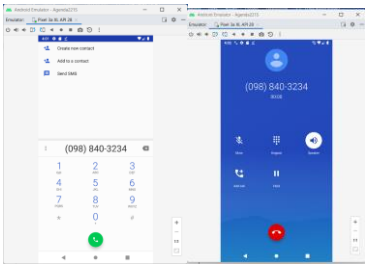
```
@Override
public int getItemCount() { return ListaContactos.size(); }

public class ViewHolderContacto extends RecyclerView.ViewHolder {

    //variables locales respecto a los componentes del cardview
    TextView id, numero, nombres, apellidos, empresa, mail, fechanac;
    CircleImageView foto;


    Button btnllamar, btninfocontacto;

    public ViewHolderContacto(@NonNull View itemView) {
        super(itemView);

        //asociar varibales logigas vs Graficas del Cardview
        id = (TextView)itemView.findViewById(R.id.textView4);
        numero = (TextView) itemView.findViewById(R.id.textView);
        nombres = (TextView) itemView.findViewById(R.id.textView2);
        apellidos = (TextView) itemView.findViewById(R.id.textView3);
        foto = (CircleImageView) itemView.findViewById(R.id.profile_image);

        btnllamar = (Button) itemView.findViewById(R.id.button3);
        btninfocontacto = (Button)itemView.findViewById(R.id.button4);
```

Crear el evento “setOnClickLister” para los botones bajo la asociación de variables

```
//Asociar variables logigas vs Graficas del Cardview
id = (TextView)itemView.findViewById(R.id.textView4);
numero = (TextView) itemView.findViewById(R.id.textView);
nombres = (TextView) itemView.findViewById(R.id.textView2);
apellidos = (TextView) itemView.findViewById(R.id.textView3);
foto = (CircleImageView) itemView.findViewById(R.id.profile_image);

btnllamar = (Button) itemView.findViewById(R.id.button3);
btninfocontacto = (Button)itemView.findViewById(R.id.button4);


//boton llamada telefonica
btnllamar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
      Toast.makeText(id.getContext(), "Aqui llama al numero", Toast.LENGTH_SHORT).show();

    }
});


//boton infocontacto
btninfocontacto.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Toast.makeText(id.getContext(), "Mas info", Toast.LENGTH_SHORT).show();

    }
});
```

Codigo para activar el dial de llamada

```
//boton llamada telefonica
btnllamar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        //hacer llamada
        String dial = "tel:" + numero.getText().toString();
        view.getContext().startActivity(new Intent(Intent.ACTION_DIAL, Uri.parse(dial)));
        Systems.exit(0);


    }
});
```



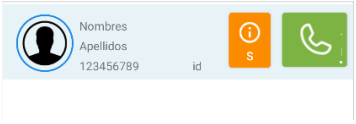
d. Programar el Boton “INFO”

Nota: para el botón info, si dese ver el id selecionado en un toast usar.

```
Toast.makeText(view.getContext(), "Id="+ id.getText().toString(), Toast.LENGTH_SHORT).show();
```

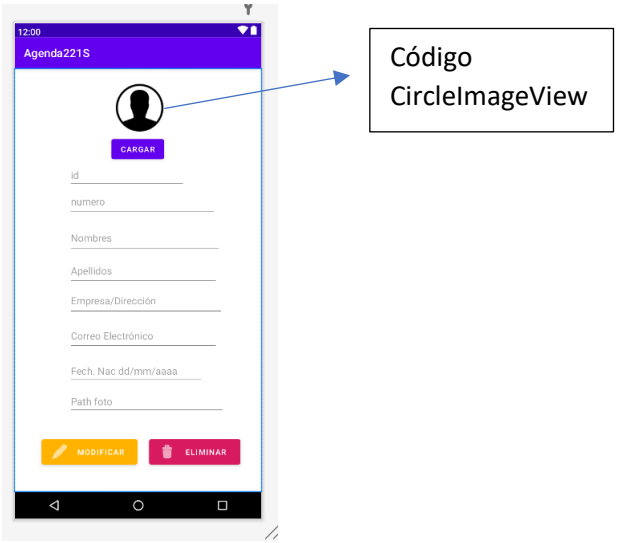
Poner un boto para ver detalles

Xml diseño



Programar el botón INFO en el AdapterContacto que lleve al Layot 3

►Diseño de Layout 3, Variables Propias y Asociar




```
//Varianles propias
EditText txtid,txtnumero, txtnombres, txtapellidos, txtempresa, txtcorreo, txtfechanac, txtfoto;
Button btnmodificar, btneliminar, btncargarfoto;
CircleImageView foto;
@Override
```

```
//Asociamos
txtid = (EditText) findViewById(R.id.editTextNumber3);
txtnumero = (EditText) findViewById(R.id.editTextNumber4);
txtnombres = (EditText) findViewById(R.id.editTextTextPersonName6);
txtapellidos = (EditText) findViewById(R.id.editTextTextPersonName7);
txtempresa = (EditText) findViewById(R.id.editTextTextPersonName8);
txtcorreo = (EditText) findViewById(R.id.editTextTextPersonName9);
txtfechanac = (EditText)findViewById(R.id.editTextNumber5);
txtfoto = (EditText) findViewById(R.id.editTextTextPersonName10);

btnmodificar = (Button)findViewById(R.id.button5);
btneliminar = (Button)findViewById(R.id.button6);
btncargarfoto = (Button)findViewById(R.id.button7);

foto = (CircleImageView)findViewById(R.id.profile_image);
```

►Boton info (adapter contacto)llama al layout 3 y pasa el parámetro id (PK)

En el ViewHolderContacto

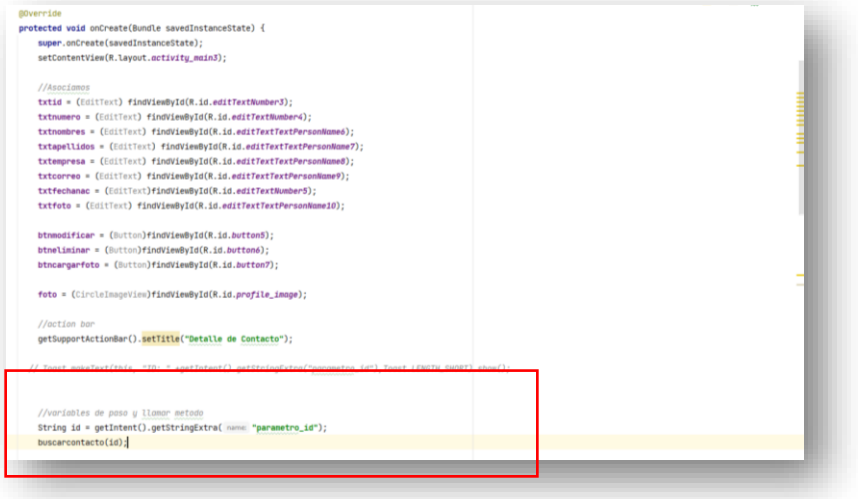
```
//boton llamar contacto
 ..........................................

//boton infocontacto
btninfocontacto.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Toast.makeText(view.getContext(), "Id="+ id.getText().toString(),
Toast.LENGTH_SHORT).show();

        Intent formulario = new Intent(view.getContext(),MainActivity3.class);
        formulario.putExtra("parametro_id",id.getText().toString());
        view.getContext().startActivity(formulario);

        //System.exit(0);


    }
});
```

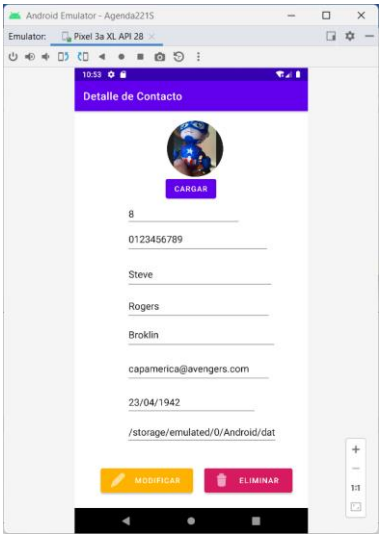
► en el layout 3 crear el método bucarcontacto

```
public void buscarcontacto(String id)
{
    //Crear/Conectarse BD
    AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper(this, "MiAgenda", null, 1);

    //abrir BD
    SQLiteDatabase BaseDeDatos = cnx.getReadableDatabase();


    if(!id.isEmpty())
    {
        //ejecutar transql
        Cursor fila = BaseDeDatos.rawQuery("select * from contacto where id = "+ id , null);

        if(fila.moveToFirst())
        {
            //mostrar datos en los campos
            txtid.setText(fila.getString(0));
            txtnumero.setText(fila.getString(1));
            txtnombres.setText(fila.getString(2));
            txtapellidos.setText(fila.getString(3));
            txtempresa.setText(fila.getString(4));
            txtcorreo.setText(fila.getString(5));
            txtfechanac.setText(fila.getString(6));
            txtfoto.setText(fila.getString(7));

            foto.setImageURI(Uri.parse(txtfoto.getText().toString()));


            //cerramos BD
            BaseDeDatos.close();
        }
        else
        {
            Toast.makeText(this,"Contacto no existe",Toast.LENGTH_SHORT).show();

            //cerramos BD
            BaseDeDatos.close();

        }

    }
    else
    {
        Toast.makeText(this,"Debe ingresar Id ",Toast.LENGTH_SHORT).show();

    }

}
```

►en el Layout 3, recibir el parámetro del layout 2 y pasarlo al método buscarcontacto

```
//variables de paso y llamar metodo
String id = getIntent().getStringExtra("parametro_id");
buscarcontacto(id);
```





*Ocultar campo ID y Path foto

Programar Cargar foto, Modificar y Eliminar

Cargar foto

a. Cargar Fotografia (Camara o Galeria) Uso_de_Camara

Eliminar

1. Implemetar el medoto eliminarcontacto() antes de la ulitma llaves en el layout3

```
//metodo elminar contacto
public void eliminarcontacto() {

    //Crear/Conectarse BD
    AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper(this, "MiAgenda", null, 1);

    //abrir BD
    SQLiteDatabase BaseDeDatos = cnx.getWritableDatabase();

    //capturo el codigo del producto
    String id = txtid.getText().toString();

    if (!id.isEmpty()) {

        try {

            //ejecutamos la transql : delete from producto where id = 1;
            String sql = "delete from contacto where id = " + id;
            BaseDeDatos.execSQL(sql);

            //cerramos BD
            BaseDeDatos.close();

            Toast.makeText(this,"Contacto Eliminado, actualizar lista",Toast.LENGTH_SHORT).show();

            //limpiar campos
          /* txtid.setText("");
            txtnombres.setText("");
            txtapellidos.setText("");
            txtempresa.setText("");
            txtid.requestFocus();*/
            finish();

        }
        catch (Exception e)
        {
            Toast.makeText(this,"Producto No existe",Toast.LENGTH_SHORT).show();

            //cerramos BD
            BaseDeDatos.close();
        }

    }
    else
    {
        Toast.makeText(this,"No se ha especificado el ID del producto",Toast.LENGTH_SHORT).show();
```

```
            }

        }
```

2. Llamar al evento eliminarcontacto() pero confirmar la eliminación AlertBuilder

```
        //boton eliminar
        btneliminar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                AlertDialog.Builder alerta = new AlertDialog.Builder(MainActivity3.this);
                alerta.setIcon(android.R.drawable.ic_delete);
                alerta.setTitle("Aviso");
                alerta.setMessage("¿Esta seguro de eliminar el contacto permannetememte?");
                alerta.setCancelable(false);

                alerta.setPositiveButton("Si", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialogInterface, int i) {

                        //lamar metodo eliminar producto
                        eliminarcontacto();
                    }
                });

                alerta.setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialogInterface, int i) {

                    }
                });
                alerta.show();
            }
        });
```

Modificar

1. Implemetar el medoto modificarcontacto() antes de la ulitma llaves en el layout3

```
//metodo modificarcontacto
public void modificarcontacto()
{
    //Crear/Conectarse BD
    AdminSQLiteOpenHelper cnx = new AdminSQLiteOpenHelper( this, "MiAgenda", null,1);

    //abrir BD
    SQLiteDatabase BaseDeDatos = cnx.getWritableDatabase();

    //capturo lo que esta en los text
    String id = txtid.getText().toString();
    String numero = txtnumero.getText().toString();
    String nombres = txtnombres.getText().toString();
    String apellidos = txtapellidos.getText().toString();
    String empresa = txtempresa.getText().toString();
    String mail = txtcorreo.getText().toString();
    String fechanac = txtfechanac.getText().toString();
    String foto = txtfoto.getText().toString();


    if(!id.isEmpty() && !numero.isEmpty()  && !nombres.isEmpty() )
    {

        try {
            //ejecutar update contacto set numero='algo', nombres='algo', apellidos='algo', empresa='algo',
mail='algo', fechanac='algo', foto='algo' where id = 1

            String sql = "update contacto set numero = '"+ numero +"' , nombres = '"+ nombres +"', apellidos ='"+
apellidos +"', empresa='"+ empresa +"' , mail='"+ mail +"', fechanac='"+ fechanac +"', foto = '"+foto+"' where id
= " + id + "";
            BaseDeDatos.execSQL(sql);
            Toast.makeText(this,"Datos de Contacto Actualizado, actualizar lista",Toast.LENGTH_SHORT).show();

            //cierra la BD
            BaseDeDatos.close();

            //limpiar campos
             finish();
        }
        catch (Exception e)
        {

            Toast.makeText(this,"Problemas: COntacto no existe, imposible modificar",Toast.LENGTH_SHORT).show();

            //limpiar campos
             finish();
```

```
            //cierra la BD
            BaseDeDatos.close();
        }


    }
    else
    {
        Toast.makeText(this,"Existen Campos vacios",Toast.LENGTH_SHORT).show();

        txtnumero.setError("Campo Obligatorio");
        txtnombres.setError("Campo Obligatorio");


    }

}
```

//Boton Modificar

```
            //boton actualizar contacto
            btnmodificar.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {

                    AlertDialog.Builder alerta = new AlertDialog.Builder(MainActivity3.this);
                    alerta.setIcon(android.R.drawable.ic_menu_edit);
                    alerta.setTitle("Aviso");
                    alerta.setMessage("¿Desea actualizar los datos del Contacto?");
                    alerta.setCancelable(false);

                    alerta.setPositiveButton("Si", new DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialogInterface, int i) {

                            //lamar metodo eliminar producto
                            modificarcontacto();

                        }
                    });

                    alerta.setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialogInterface, int i) {

                        }
                    });


                    alerta.show();

                }
            });
```

INVESTIGACION

VALIDA ACCESO SEGUN NIVEL

En el Layout LOGIN poner un TextView Lblnivel

Crear un Layout para el otro Ususario

Crear el archivo php que reciba Usuario y Password y devuelva campo NIVEL

►Copiar en la carpeta de publicación

validar_acceso.php

```
<?php
include 'conexion.php';
$usu_usuario=$_GET['usuariophp'];
$usu_password=$_GET['passwordphp'];
$sql="select nivel from usuario where login='".$usu_usuario."' and  pass='".$usu_password."'";
$resultado= $conexion -> query($sql);
while($fila = $resultado -> fetch_array())
{
        $usuario[] = array_map('utf8_encode', $fila);
}
echo json_encode($usuario);
$resultado -> close();
?>
```

►Crear el método validaracceso en el layou LOGIn

```
public void validaracceso (String URL)
{

    JsonArrayRequest jsonArrayRequest = new JsonArrayRequest(URL, new
Response.Listener<JSONArray>() {
```

```
        @Override
        public void onResponse(JSONArray response) {


            JSONObject jsonObject = null;
            for(int i=0 ; i<response.length() ; i++)
            {
                try{
                    jsonObject = response.getJSONObject(i);
                    lblnivel.setText(jsonObject.getString("nivel"));

                }
                catch (Exception err)
                {
                    Toast.makeText(MainActivity.this, err.toString(),Toast.LENGTH_SHORT);
                }
                Toast.makeText(MainActivity.this, "CARGANDO"+lblnivel.getText().toString(),
Toast.LENGTH_SHORT );
            }


        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(MainActivity.this, "Dato No existe", Toast.LENGTH_SHORT).show();
            txtUsuario.setText("");
            txtPassword.setText("");

            txtUsuario.requestFocus();

        }
    });
    RequestQueue requestQueue = Volley.newRequestQueue(this);
    requestQueue.add(jsonArrayRequest);


}
```

►Adecuar el método validar_usuario para enviar a un layout u otro

```
private void  validarusuario (String URL)
{
    StringRequest stringRequest = new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
              if(!response.isEmpty())
              {


//METODO UNO REDIRIGIR A UN SCREEN EN BASE AL NIVEL
         if(lblnivel.getText().toString().equals("1")) {

                    Toast.makeText(MainActivity.this, "Credenciales Correctas", Toast.LENGTH_SHORT).show();
                    Intent pantallaprincipal = new Intent(MainActivity.this, MainActivity2.class);
                    startActivity(pantallaprincipal);
                    finish();
                }

                if(lblnivel.getText().toString().equals("2")) {

                    Toast.makeText(MainActivity.this, "Credenciales Correctas", Toast.LENGTH_SHORT).show();
                    Intent pantallausuario = new Intent(MainActivity.this, MainActivity4.class);
                    startActivity(pantallausuario);
                    finish();
                }


//METODO DOS – ENVIAR PARAMETROS PARA QUE EL SCREEN DECIDA SI HABILITAR/BLOQUEAR ALGO

Intent menu = new Intent(MainActivity.this, MainActivity2.class);

//envio variables
menu.putExtra("nivel", lblnivel.getText().toString());

startActivity(menu);
finish();
```

```
            }
            else
            {
                Toast.makeText(MainActivity.this, "Credenciales Incorrectas", Toast.LENGTH_SHORT).show();
                txtUsuario.setText("");
                txtPassword.setText("");
                txtUsuario.requestFocus();

            }

        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {

            Toast.makeText(MainActivity.this, error.toString(), Toast.LENGTH_SHORT).show();
            txtUsuario.setText("");
            txtPassword.setText("");
            txtUsuario.requestFocus();

        }
    })
    {
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {

            Map<String,String> paramatros = new HashMap<String,String>();

             paramatros.put("usuariophp", txtUsuario.getText().toString());
             paramatros.put("passwordphp", txtPassword.getText().toString());


            return paramatros;
        }
    };
    RequestQueue requestQueue = Volley.newRequestQueue(this);
    requestQueue.add(stringRequest);
```

►llamar a los métodos validar_usuario y validar acceso en el Boton OK

```
btnIngresar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        if (!txtUsuario.getText().toString().isEmpty() && !txtPassword.getText().toString().isEmpty() ) {


validaracceso("http://192.168.100.18:8080/MiTienda/validar_acceso.php?usuariophp="+txtUsuario.getText().toString()+"&&passwordph
p="+txtPassword.getText().toString()+"");

            validarusuario("http://192.168.100.18:8080/MiTienda/validar_usuario.php");
        }
        else
        {
            Toast.makeText(MainActivity.this, "Campos vacios", Toast.LENGTH_SHORT).show();
        }


    }
```

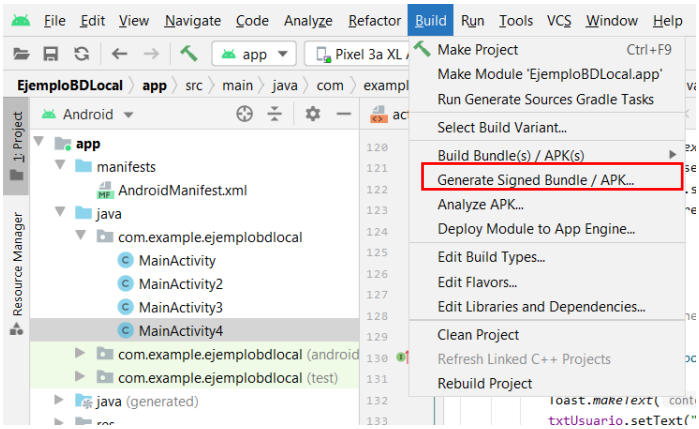
METODO 2 – RECIBIE VALORES Y DECIE

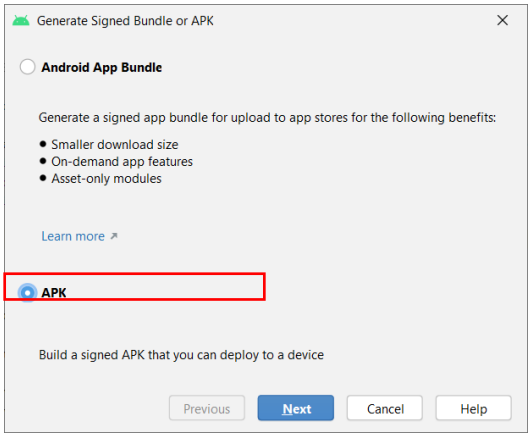
(agregar un componente visual para gauardar el valor enviado desded el screen anterior)

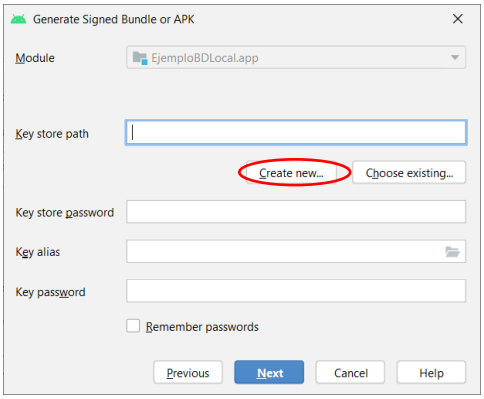
```
lblnivel = (TextView)findViewById(R.id.textView4);


//guardo parametro
```

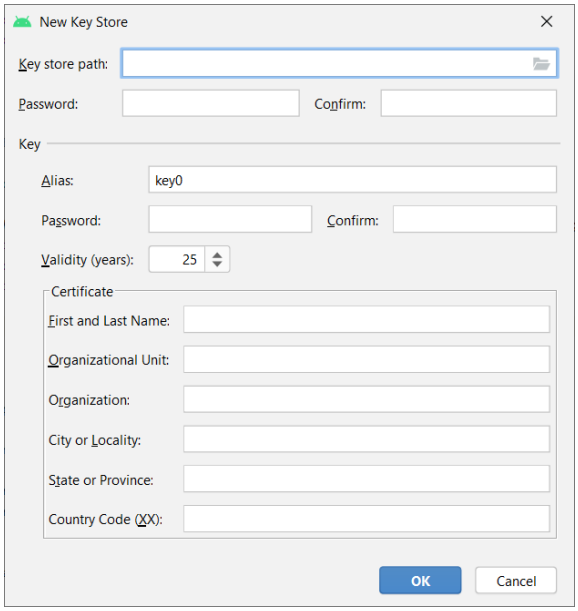
```
   lblnivel.setText(""+getIntent().getStringExtra("nivel"));


   if (lblnivel.getText().toString().equals("2"))
   {
       //btneliminar.setVisibility(View.INVISIBLE);
       btneliminar.setEnabled(false);
       //Toast.makeText(MainActivity2.this,"Aki desdhabiito",
Toast.LENGTH_SHORT).show();
   }
   else {

    //aki todo el código a ejecutar normalmente

}
```

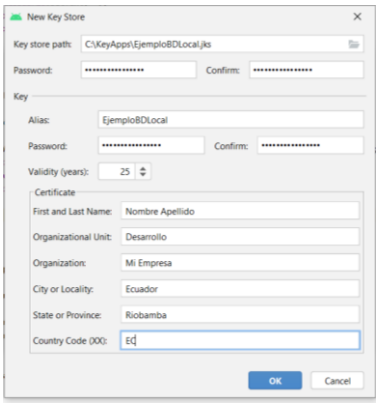
TALLER: CREACION APK

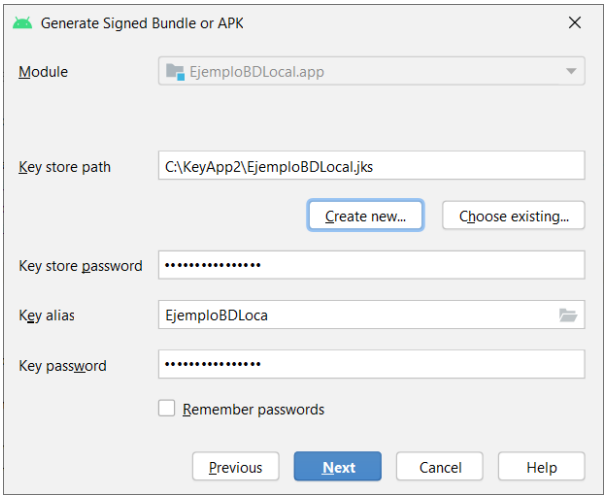


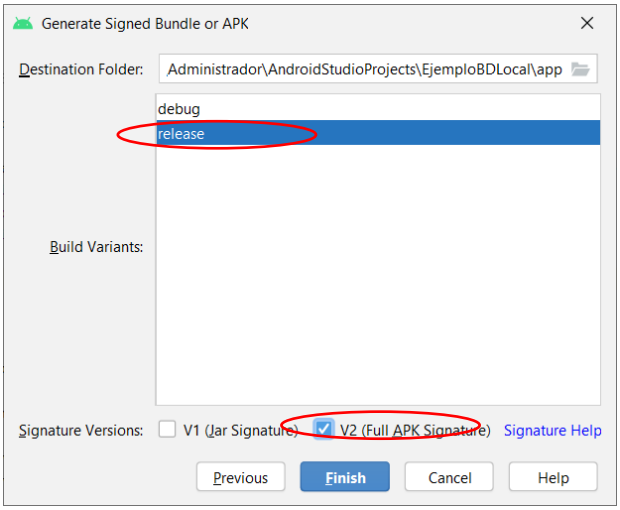


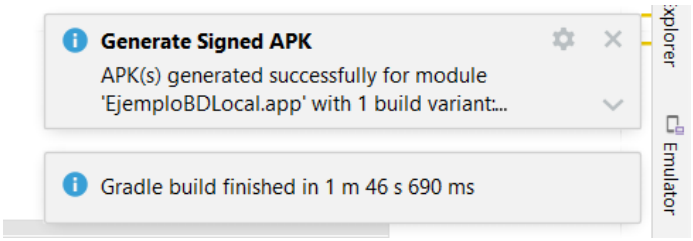



New Key Store
Key store path:
Password:
Confirm:
Key
Alias:
key0
Password:
Confirm:
Validity (years):
25
Certificate
First and Last Name:
Organizational Unit:
Organization:
City or Locality:
State or Province:
Country Code (XX):
OK
Cancel

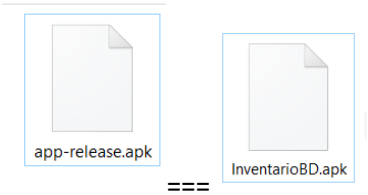

New Key Store
Key store path: C:\KeyApps\EjemploBDLocal.jks
Password:
Confirm:
Key
Alias: EjemploBDLocal
Password:
Confirm:
Validity (years): 25
Certificate
First and Last Name: Nombre Apellido
Organizational Unit: Desarrollo
Organization: Mi Empresa
City or Locality: Ecuador
State or Province: Riobamba
Country Code (XX): EC
OK
Cancel


Generate Signed Bundle or APK
Module
EjemploBDLocal.app
Key store path
C:\KeyApp2\EjemploBDLocal.jks
Create new...
Choose existing...
Key store password
Key alias
EjemploBDLoca
Key password
Remember passwords
Previous
Next
Cancel
Help





C:\Users\Administrador\AndroidStudioProjects\EjemploBDLocal\app\release



===

POR SI QUIERES SOLICIONAR ESTE ERROR

https://www.youtube.com/watch?v=ltzJ3DmX8gE