Crear una nueva activity, click derecho en res y seguir según la imagen.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Al crearla, se debe asignar un nombre, intentado poner el tag “Activity” después del nombre, por ejemplo “DatosActivity”

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Al crear la actividad, se creará junto con su respectivo layout.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Para el ejemplo, se agregará un editText del tipo plain text y un botón para enviar el dato escrito a una segunda actividad, la cual tendrá un TextView y un botón que me permitirá cerrar la actividad y volver a la principal.

 Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

El código para la activity principal es el siguiente:

public class MainActivity extends AppCompatActivity {  
  
 EditText etDato1;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 etDato1 = findViewById(R.id.*etDato1*);  
 }

Código que me permitirá tener obtener el dato escrito y luego tendremos el método que ejecutará el botón (no olvidar asignar el método al botón en la vista.)

public void enviar(View v){  
  
 Intent i = new Intent(this, DatosActivity.class);  
  
 i.putExtra("miDato", etDato1.getText().toString());  
  
 startActivity(i);  
  
  
}

Cabe mencionar que si se desea solamente abrir un activity, no es necesario agregar un putExtra y se puede abrir directamente con el startActivity.

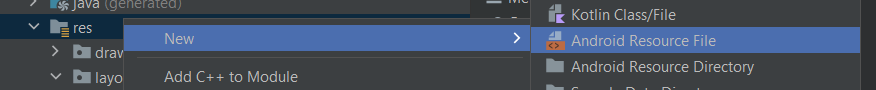
Y luego en la siguiente Actiivity se debe cargar de la siguiente forma, para poder obtener el bundle con todos los datos enviados por el intent anterior (si es que los hay)

public class DatosActivity extends AppCompatActivity {  
  
 private TextView tvMostrar;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_datos*);  
  
 tvMostrar = (TextView) findViewById(R.id.*tvMostrar*);  
  
 Bundle bundle = getIntent().getExtras();  
  
 String dato = bundle.getString("miDato");  
  
 tvMostrar.setText(dato);  
  
 }

Y para programar el botón salir, solo se debe crear el siguiente método y asignarlo al botón.

public void salir(View v){  
   
 finish();  
}

Para crear una ActionBar se debe dar click derecho a la carpeta res y seleccionar un nuevo resource file de acuerdo con la siguiente imagen:



Se le debe asignar un nombre y luego cambiar el resource type a “Menú”, recordando que el nombre del archivo no acepta caracteres especiales ni mayúsculas.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Al crearlo, se generará una nueva carpeta llamada menú con un layout totalmente independiente a las activities, este nuevo layout corresponde al menú y puede ser invocado desde cualquier actividad.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Recordar buscar iconos y pegarlos en la carpeta mipmap.

Ahora en el layout del menú, aparecerán las opciones de menú o “menú item” que se pueden ir agregando una a una.

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Cada menú item tiene su cuadro de propiedades y se le puede asignar un icono y forma de visualización.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Cuando se selecciona la opción “ShowAsAction”, se pueden seleccionar las opciones:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

If room mostrará el icono activo mientras haya espacio en la actionbar, mientras que always mostrará siempre el icono, por otra parte, la opción “never” mantendrá siempre oculta la opción.

Una vez agregadas las alternativas