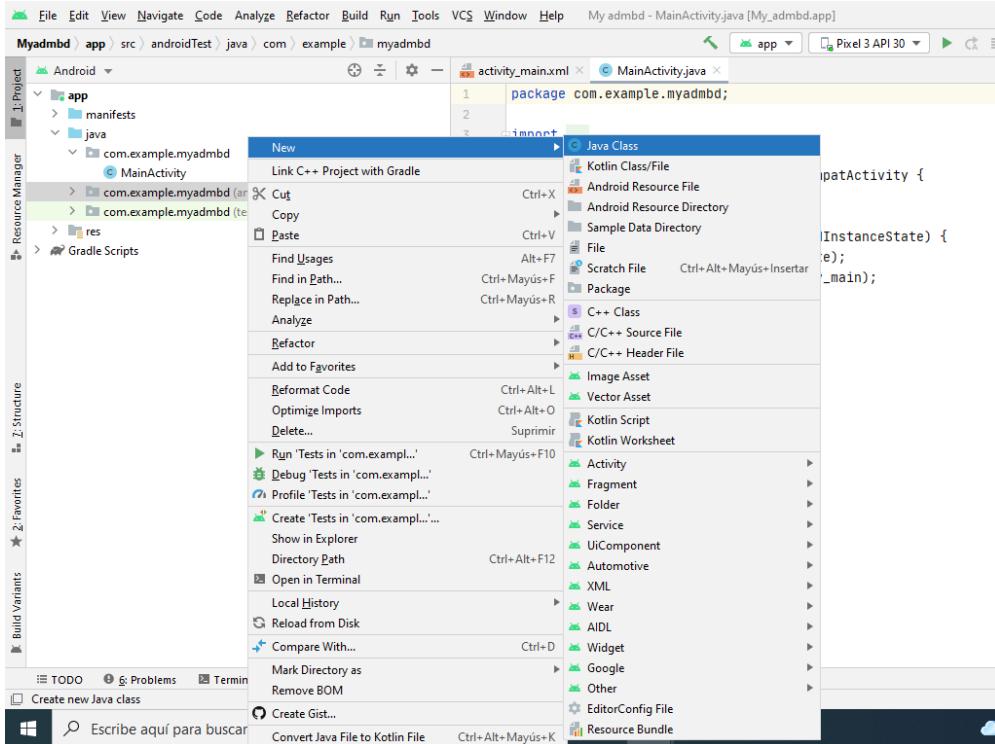


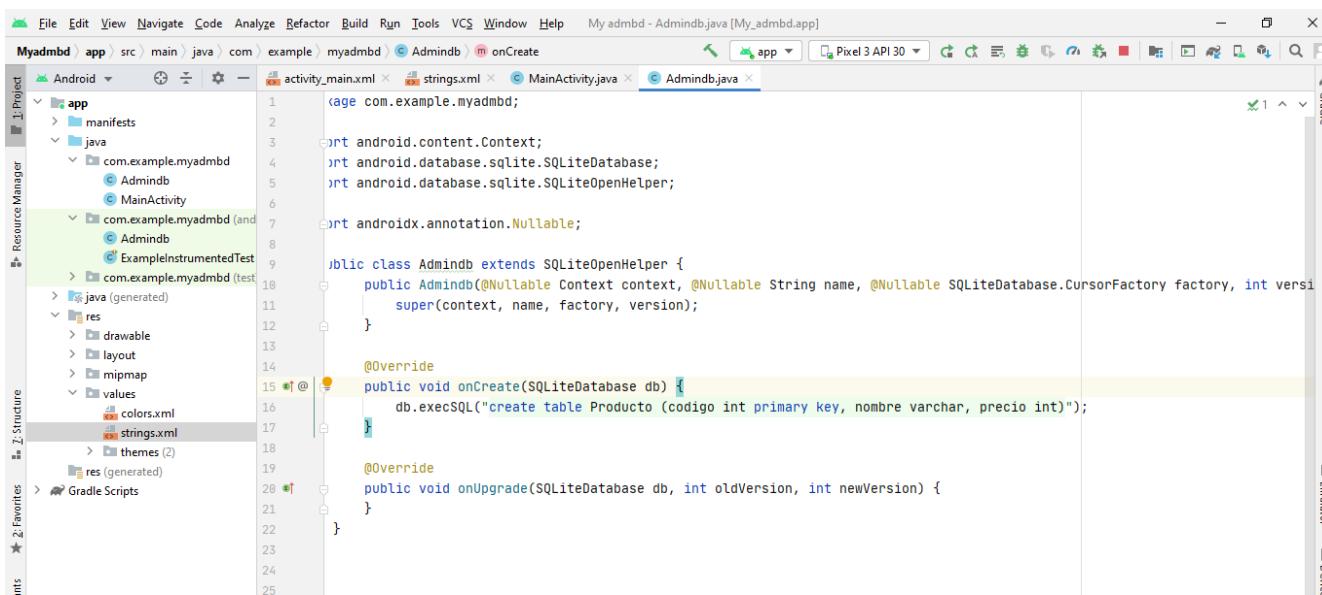
Evidencia día 03 semana 10

En la clase de hoy veremos cómo trabajar con una base de datos en AndroidStudio.

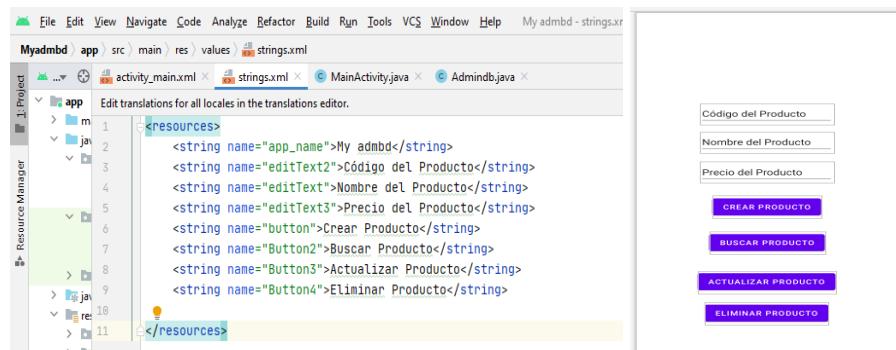
Lo primero hay que hacer es crear una clase para la base de datos con la que vamos a trabajar.



Se crea una clase abstract Admindb, la cual hereda de OpenHelper. Debería dar un error y pinchamos alt+intro y se crearán automáticamente los métodos onCreate y onUpgrade de la clase.



Crear el layout, para este ejercicio se necesitan 3 componentes EditText y 4 componentes Button.
 Crear nombres en strings.xml y luego en atributos asociar a Text y Hint.



Ahora nos vamos al MainActivity (java) para crear los métodos.

Método Crear

```

package com.example.myadmbd;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.EditText;
import android.content.ContentValues;
import android.database.sqlite.SQLiteDatabase;
import android.view.View;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private EditText edcodigo, ednombre, edprecio;

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

        edcodigo = findViewById(R.id.editText2);
        ednombre = findViewById(R.id.editText);
        edprecio = findViewById(R.id.editText3);

    }
    public void CrearProd (View v) {

        Admindb admin = new Admindb (this,"Productos",null,1);
        SQLiteDatabase base=admin.getWritableDatabase();
        String codigo = edcodigo.getText().toString(); //
        String nombre = ednombre.getText().toString();
        String precio = edprecio.getText().toString();

        if (!codigo.isEmpty() && !nombre.isEmpty() && !precio.isEmpty()){
            ContentValues crear = new ContentValues();
            crear.put("codigo", codigo); //crea un código de producto
            crear.put("nombre", nombre); //crea un nombre de producto
            crear.put("precio", precio); //crea un precio de producto
            base.insert("Producto",null,crear);
            base.close(); //cierra la base de datos
            edcodigo.setText("");
        }

    }
}

```

```

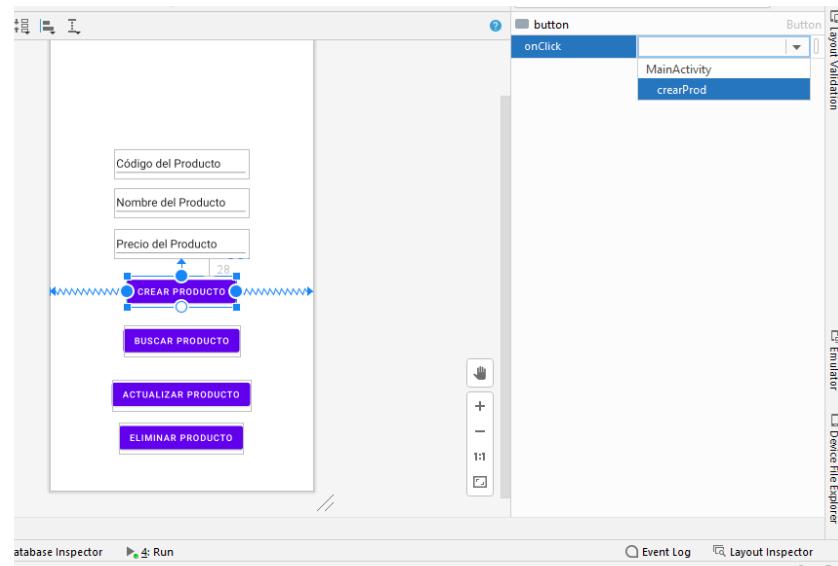
ednombre.setText("");
edprecio.setText("");
Toast.makeText(this, "Producto Creado", Toast.LENGTH_LONG).show();
}else{
    Toast.makeText(this,"Debes llenar todos los campos",Toast.LENGTH_LONG).show(); //Aviso de que se deben llenar
todos los campos al crear un producto
}
}
}

```

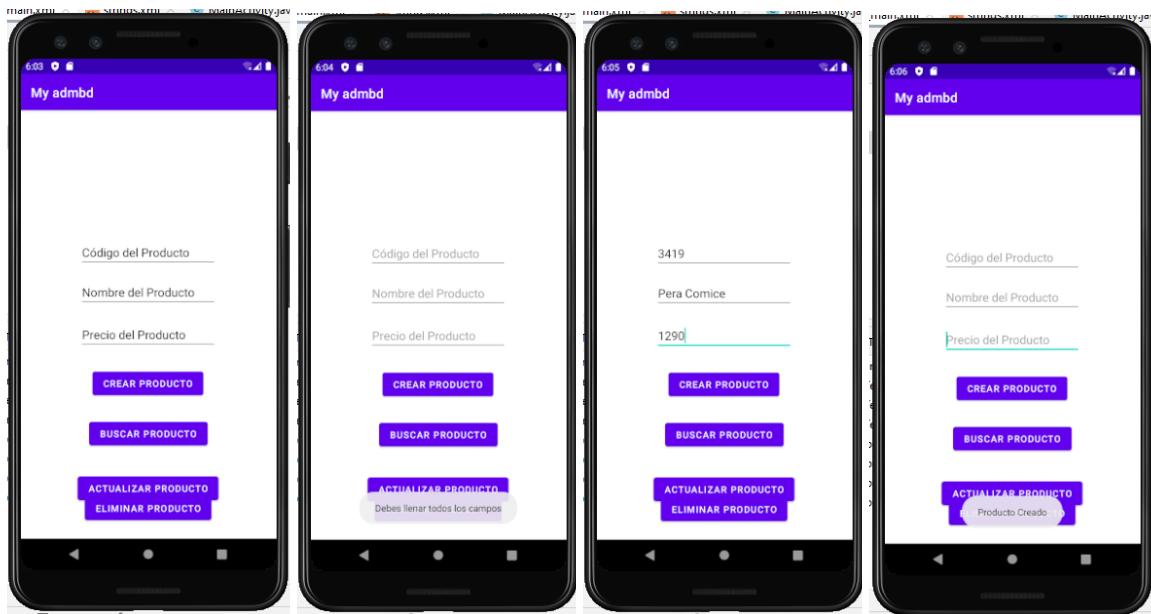
Entonces se crea un objeto de la clase, que administra la base de datos, en este caso de la clase Adminbd y luego se llama al constructor. Se crea un objeto de la clase SQLiteDatabase llamado base, se le pasa el objeto que administra la base de datos llamado admin y luego se abre la base de datos para lectura y escritura, se asigna lo que va devolver edcodigo a una variable string (lo mismo para ednombre, edprecio).

Hay que asegurarse de que los tres parámetros solicitados estén ingresados , de lo contrario debería arrojar un mensaje del else y decir que se debe volver a ingresar los datos. Luego se crea el espacio para ingresar los nuevos valores en la tabla creada anteriormente en la base de datos (en archivo java Admindb). Posteriormente se insertan los valores creados que estaban guardados previamente en el método crear. Y por último hacer un toast con aviso de que el producto fue creado al apretar el botón CREAR PRODUCTO.

Ir al atributo del Button “crear Producto”, buscar la propiedad onclick y asociar el método “Crear Prod”.

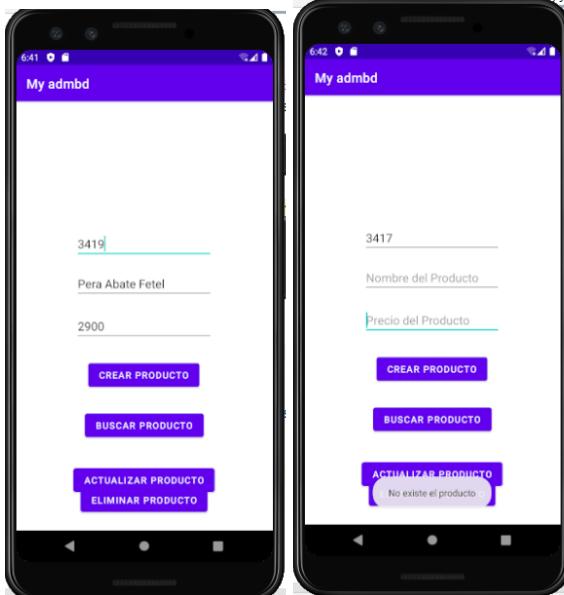


Ejecutamos el emulador



Método Buscar

```
public void Buscar (View v) {  
    Admindb admin = new Admindb(this,"Productos",null,1);  
    SQLiteDatabase base = admin.getWritableDatabase();  
  
    String codigo=edcodigo.getText().toString();  
  
    if (!codigo.isEmpty()){  
        Cursor fila = base.rawQuery("select nombre, precio from Producto where codigo='"+codigo,null);  
  
        if (fila.moveToFirst()){  
            ednombre.setText(fila.getString(0));  
            edprecio.setText(fila.getString(1));  
            base.close();  
        }else {  
            Toast.makeText(this,"No existe el producto", Toast.LENGTH_LONG).show();  
        }  
    }else{  
        Toast.makeText(this,"Debes ingresar un código de producto", Toast.LENGTH_LONG).show();  
    }  
}
```



Reflexión: La clase de hoy estuvo super entretenida y aplicamos todo lo visto en los módulo 1 y 2, cuesta un poco entender los métodos pero es necesario ir paso por paso analizando cada función del código.