

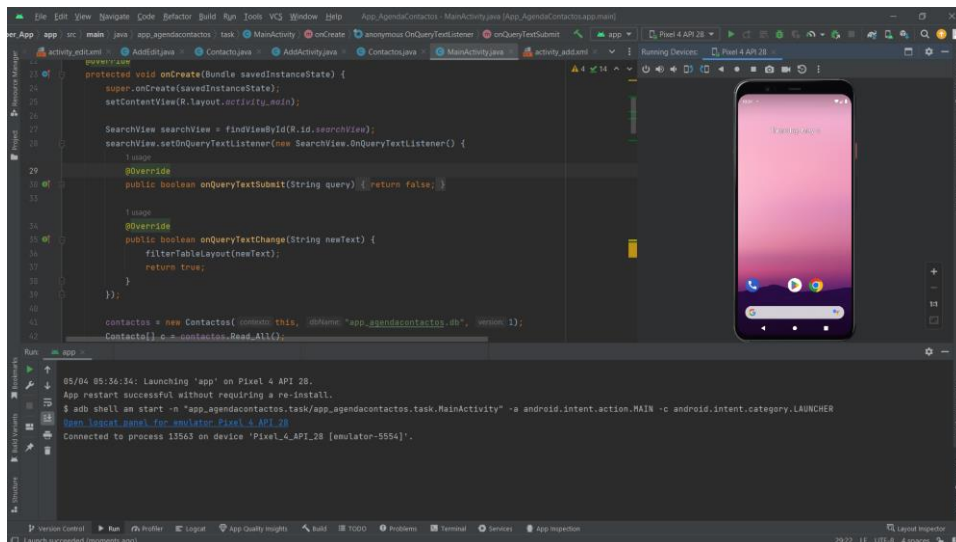


UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE SOFTWARE
ASIGNATURA: APLICACIONES MOVILES NIVEL: 07

Nombres completos:	Roberth Gabriel Lima Carvajal
Fecha:	04/05/2023
Tema:	Agenda de Contactos
Objetivo de esta actividad:	Desarrollar una aplicación móvil con la ayuda de Android Studio para la creación, la lectura, la actualización y eliminación de contactos (CRUD) tipo agenda.

INDICACIONES:

Para la realización de esta actividad se debe de tener el IDE de Android Studio, e instalado una maquina virtual móvil, para la ejecución del código desarrollado.



Una vez tenido todo esto, deberemos de programar una aplicación móvil tipo agenda de contactos, de forma que podamos realizar todas las acciones de un CRUD, y mostrar los datos en una tabla en una vista principal, mientras que el agregar nuevos contactos se deberá de realizar en otra vista:

AGENDA DE CONTACTOS

Buscar

ADD

Nombre	Teléfono	Dirección	Email
Juan	0965231253	El Olivo	@
Juan	0965231253	El Olivo	@
Juan	0965231253	El Olivo	@
Juan	0965231253	El Olivo	@

AGENDA DE CONTACTOS

Nombre

Teléfono

Dirección

Email

DELETE

SAVE

CANCELAR

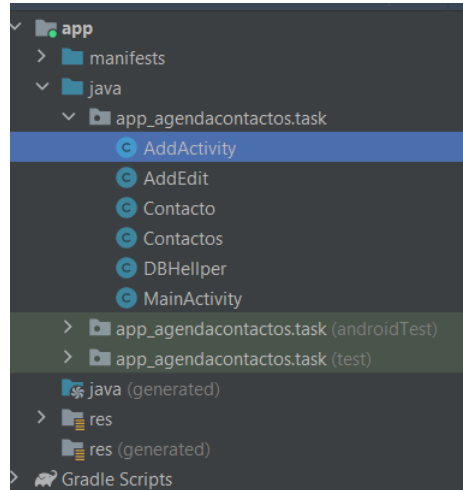
Seguro desea borrar?

Si

NO

DESARROLLO:

1. Archivos creados (Clase de Objeto, DBHellper, Clase con los métodos para el Objeto y los mainActivity para las vistas creadas):



2. Código dentro de cada archivo:

- Contacto

```
1 package app_agendacontactos.task;
2
3 public class Contacto {
4     6 usages
5     public String Id;
6     6 usages
7     public String Nombre;
8     6 usages
9     public String Telefono;
10    6 usages
11    public String Direccion;
12    6 usages
13    public String Mail;
14 }
15
```

- Contactos

```
• package app_agendacontactos.task;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.widget.TableLayout;

import java.util.ArrayList;
import java.util.List;

public class Contactos {
    private DBHellper dbHelper;
    private SQLiteDatabase db;

    public Contactos(Context contexto, String dbName, int
version) {
        dbHelper = new DBHellper(contexto, dbName, null,
```

```

version);
    }

    public Contacto Create(String id, String nombre, String
telefono, String direccion, String mail) {
        db = dbHelper.getWritableDatabase();

        ContentValues row = new ContentValues();
        row.put("id", id);
        row.put("nombre", nombre);
        row.put("telefono", telefono);
        row.put("direccion", direccion);
        row.put("mail", mail);

        long qty = db.insert("contactos", null, row);
        if (qty > 0) {
            Contacto data = new Contacto();
            data.Id = id;
            data.Nombre = nombre;
            data.Telefono = telefono;
            data.Direccion = direccion;
            data.Mail = mail;
            return data;
        } else {
            return null;
        }
    }

    public Contacto Read_One(String id) {
        db = dbHelper.getReadableDatabase();
        Cursor cr = db.rawQuery("SELECT id, nombre, tele-
fono, direccion, mail FROM contactos WHERE id = '" + id +
"'", null);
        if (cr.getCount() > 0) {
            Contacto conta = new Contacto();
            cr.moveToNext();
            conta.Id = cr.getString(0);
            conta.Nombre = cr.getString(1);
            conta.Telefono = cr.getString(2);
            conta.Direccion = cr.getString(3);
            conta.Mail = cr.getString(4);

            return conta;
        } else {
            return null;
        }
    }

    public Contacto[] Read_All() {
        db = dbHelper.getReadableDatabase();
        Cursor cr = db.rawQuery("SELECT id, nombre, tele-
fono, direccion, mail FROM contactos ORDER BY id", null);
        if (cr.getCount() > 0) {
            Contacto[] datos = new Contacto[cr.getCount()];
            Contacto conta;
            int i = 0;

            while (cr.moveToNext()) {
                conta = new Contacto();
                conta.Id = cr.getString(0);
                conta.Nombre = cr.getString(1);
                conta.Telefono = cr.getString(2);
                conta.Direccion = cr.getString(3);
                conta.Mail = cr.getString(4);

                datos[i++] = conta;
            }
        }
    }

```

```

    }

    return datos;
} else {
    return null;
}
}

public Contacto[] Read_ByNombre(String find) {
    db = dbHelper.getReadableDatabase();
    Cursor cr = db.rawQuery("SELECT id, nombre, telefono, direccion, mail FROM contactos WHERE nombre LIKE '%" +
find + "%' ORDER BY nombre", null);
    if (cr.getCount() > 0) {
        Contacto[] datos = new Contacto[cr.getCount()];
        Contacto conta;

        int i = 0;

        while (cr.moveToNext()) {
            conta = new Contacto();
            conta.Id = cr.getString(0);
            conta.Nombre = cr.getString(1);
            conta.Telefono = cr.getString(2);
            conta.Direccion = cr.getString(3);
            conta.Mail = cr.getString(4);
        }
        return datos;
    } else {
        return null;
    }
}

public boolean Update(String id, String nombre, String telefono, String direccion, String mail) {
    db = dbHelper.getWritableDatabase();
    ContentValues row = new ContentValues();
    row.put("id", id);
    row.put("nombre", nombre);
    row.put("telefono", telefono);
    row.put("direccion", direccion);
    row.put("mail", mail);

    int qty = db.update("contactos", row, "id='" + id + "'", null);
    return qty > 0;
}

public boolean Delete(String id) {
    db = dbHelper.getWritableDatabase();
    int qty = db.delete("contactos", "id='" + id + "'", null);
    return qty > 0;
}

public List<Contacto> obtenerDatos() {
    db = dbHelper.getReadableDatabase();
    Cursor cursor = db.rawQuery("SELECT id, nombre, telefono, direccion, mail FROM contactos ORDER BY nombre", null);

    List<Contacto> datos = new ArrayList<>();
    Contacto conta;
    if (cursor.moveToFirst()) {
        do {
            conta = new Contacto();
            conta.Id = cursor.getString(0);

```

```

        conta.Nombre = cursor.getString(1);
        conta.Telefono = cursor.getString(2);
        conta.Direccion = cursor.getString(3);
        conta.Mail = cursor.getString(4);
    } while (cursor.moveToNext());
}
cursor.close();
db.close();
return datos;
}
}

```

- DBHelper

```

• package app_agendacontactos.task;

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

import androidx.annotation.Nullable;

public class DBHelper extends SQLiteOpenHelper {

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

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        sqLiteDatabase.execSQL("CREATE TABLE IF NOT EXISTS
contactos( " +
            "id text(10) not null primary key," +
            "nombre text(50)," +
            "telefono text(50)," +
            "direccion text(50)," +
            "mail text(20))");
    }

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

```

- MainActivity

```

• package app_agendacontactos.task;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.view.GestureDetectorCompat;

import android.content.Intent;
import android.graphics.Rect;
import android.os.Bundle;
import android.view.GestureDetector;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewGroup;
import android.widget.SearchView;

```

```

import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    Contactos contactos;
    private TableRow touchedRow;

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

        SearchView searchView = findViewById(R.id.search-
View);
        searchView.setOnQueryTextListener(new SearchView.On-
QueryTextListener() {
            @Override
            public boolean onQueryTextSubmit(String query) {
                return false;
            }

            @Override
            public boolean onQueryTextChange(String newText)
{
                filterTableLayout(newText);
                return true;
            }
        });

        contactos = new Contactos(this, "app_agendacontac-
tos.db", 1);
        Contacto[] c = contactos.Read_All();
        TableLayout tableLayout = findViewById(R.id.table-
Layout);

        final GestureDetectorCompat gestureDetectorCompat =
new GestureDetectorCompat(this, new GestureDetector.Simple-
OnGestureListener() {
            @Override
            public boolean onDoubleTap(MotionEvent e) {
                //View view = getCurrentFocus();
                if (touchedRow != null) {
                    String id = ((TextView)
touchedRow.getChildAt(0)).getText().toString();
                    String nombre = ((TextView)
touchedRow.getChildAt(1)).getText().toString();
                    String telefono = ((TextView)
touchedRow.getChildAt(2)).getText().toString();
                    String direccion = ((TextView)
touchedRow.getChildAt(3)).getText().toString();
                    String mail = ((TextView)
touchedRow.getChildAt(4)).getText().toString();

                    Intent intent = new Intent(MainActiv-
ity.this, AddEdit.class);
                    intent.putExtra("id", id);
                    intent.putExtra("nombre", nombre);
                    intent.putExtra("telefono", telefono);
                    intent.putExtra("direccion", direccion);
                    intent.putExtra("mail", mail);
                    MainActivity.this.startActivity(intent);
                    return true;
                }
                return false;
            }
        });
    }
}

```

```

    });

    if (c != null) {
        for (Contacto conta : c) {
            TableRow row = new TableRow(this);
            row.setLayoutParams(new TableRow.LayoutParams(
                TableRow.LayoutParams.MATCH_PARENT,
                ViewGroup.LayoutParams.WRAP_CONTENT
            ));

            TextView txtId = new TextView(this);
            txtId.setText(conta.Id);
            row.addView(txtId);

            TextView txtNombre = new TextView(this);
            txtNombre.setText(conta.Nombre);
            row.addView(txtNombre);

            TextView txtTelefono = new TextView(this);
            txtTelefono.setText(conta.Telefono);
            row.addView(txtTelefono);

            TextView txtDireccion = new TextView(this);
            txtDireccion.setText(conta.Direccion);
            row.addView(txtDireccion);

            TextView txtEmail = new TextView(this);
            txtEmail.setText(conta.Mail);
            row.addView(txtEmail);

            row.setOnTouchListener(new View.OnTouchListener() {
                @Override
                public boolean onTouch(View v, MotionEvent event) {
                    if(event.getAction() == MotionEvent.ACTION_DOWN) {
                        touchedRow = (TableRow) v;
                    }
                    // Pasar el evento al detector de gestos compatibles
                    GestureDetectorCompat.onTouchEvent(event);
                    return true;
                }
            });

            tableLayout.addView(row);
        }
    }

    public void addView(View view) {
        Intent btnAdd = new Intent(this, AddActivity.class);
        startActivity(btnAdd);
    }

    private void filterTableLayout(String query) {
        TableLayout tableLayout = findViewById(R.id.tableLayout);
    }

```

```

        for (int i = 1; i < tableLayout.getChildCount();
i++) {
            TableRow row = (TableRow) tableLayout.
getChildAt(i);
            boolean shouldShow = false;

            for (int j = 0; j < row.getChildCount(); j++) {
                View view = row.getChildAt(j);
                if (view instanceof TextView) {
                    TextView textView = (TextView) view;
                    String text =
textView.getText().toString().toLowerCase();
                    if (text.contains(query.toLowerCase()))
{
                        shouldShow = true;
                        break;
                    }
                }
            }

            if (shouldShow) {
                row.setVisibility(View.VISIBLE);
            } else {
                row.setVisibility(View.GONE);
            }
        }
    }
}

```

- AddActivity

- ```

package app_agendacontactos.task;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.view.GestureDetectorCompat;

import android.content.Intent;
import android.graphics.Rect;
import android.os.Bundle;
import android.view.GestureDetector;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewGroup;
import android.widget.SearchView;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
 Contactos contactos;
 private TableRow touchedRow;

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

 SearchView searchView = findViewById(R.id.search-
View);
 searchView.setOnQueryTextListener(new SearchView.On-
QueryTextListener() {
 @Override
 public boolean onQueryTextSubmit(String query) {
 return false;
 }
 });
 }
}

```



```

 }

 @Override
 public boolean onQueryTextChange(String newText)
 {
 filterTableLayout(newText);
 return true;
 }
});

contactos = new Contactos(this, "app_agendacontactos.db", 1);
Contacto[] c = contactos.Read_All();
TableLayout tableLayout = findViewById(R.id.tableLayout);

final GestureDetectorCompat gestureDetectorCompat =
new GestureDetectorCompat(this, new GestureDetector.SimpleOnGestureListener() {
 @Override
 public boolean onDoubleTap(MotionEvent e) {
 //View view = getCurrentFocus();
 if (touchedRow != null) {
 String id = ((TextView)
touchedRow.getChildAt(0)).getText().toString();
 String nombre = ((TextView)
touchedRow.getChildAt(1)).getText().toString();
 String telefono = ((TextView)
touchedRow.getChildAt(2)).getText().toString();
 String direccion = ((TextView)
touchedRow.getChildAt(3)).getText().toString();
 String mail = ((TextView)
touchedRow.getChildAt(4)).getText().toString();

 Intent intent = new Intent(MainActivity.this, AddEdit.class);
 intent.putExtra("id", id);
 intent.putExtra("nombre", nombre);
 intent.putExtra("telefono", telefono);
 intent.putExtra("direccion", direccion);
 intent.putExtra("mail", mail);
 MainActivity.this.startActivity(intent);
 return true;
 }
 return false;
 }
});

if (c != null) {
 for (Contacto conta : c) {
 TableRow row = new TableRow(this);
 row.setLayoutParams(new TableRow.LayoutParams(
Params(
 TableRow.LayoutParams.MATCH_PARENT,
 ViewGroup.LayoutParams.WRAP_CONTENT
));

 TextView txtId = new TextView(this);
 txtId.setText(conta.Id);
 row.addView(txtId);

 TextView txtNombre = new TextView(this);
 txtNombre.setText(conta.Nombre);
 row.addView(txtNombre);
 }
}

```

```

 TextView txtTelefono = new TextView(this);
 txtTelefono.setText(conta.Telefono);
 row.addView(txtTelefono);

 TextView txtDireccion = new TextView(this);
 txtDireccion.setText(conta.Direccion);
 row.addView(txtDireccion);

 TextView txtEmail = new TextView(this);
 txtEmail.setText(conta.Mail);
 row.addView(txtEmail);

 row.setOnTouchListener(new View.OnTouchListener() {
 @Override
 public boolean onTouch(View v, MotionEvent event) {
 if(event.getAction() == MotionEvent.ACTION_DOWN) {
 touchedRow = (TableRow) v;
 }
 // Pasar el evento al detector de gestos compatibles
 GestureDetectorCompat.onTouchEvent(event);
 return true;
 }
 });

 tableLayout.addView(row);
 }
}

public void addView(View view) {
 Intent btnAdd = new Intent(this, AddActivity.class);
 startActivity(btnAdd);
}

private void filterTableLayout(String query) {
 TableLayout tableLayout = findViewById(R.id.tableLayout);

 for (int i = 1; i < tableLayout.getChildCount(); i++) {
 TableRow row = (TableRow) tableLayout.getChildAt(i);
 boolean shouldShow = false;

 for (int j = 0; j < row.getChildCount(); j++) {
 View view = row.getChildAt(j);
 if (view instanceof TextView) {
 TextView textView = (TextView) view;
 String text = textView.getText().toString().toLowerCase();
 if (text.contains(query.toLowerCase())) {
 shouldShow = true;
 break;
 }
 }
 }
 }
}

```

```

 if (shouldShow) {
 row.setVisibility(View.VISIBLE);
 } else {
 row.setVisibility(View.GONE);
 }
 }
}

```

- AddEdit

```

• package app_agendacontactos.task;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
import android.app.AlertDialog;
import android.content.DialogInterface;

import androidx.appcompat.app.AppCompatActivity;

public class AddEdit extends AppCompatActivity {

 Contactos contactos;
 EditText txtId, txtNombre, txtTelefono, txtDireccion,
 txtMail;

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

 contactos = new Contactos(this, "app_agendacontactos.db", 1);

 Bundle extras = getIntent().getExtras();
 String id = extras.getString("id");
 String nombre = extras.getString("nombre");
 String telefono = extras.getString("telefono");
 String direccion = extras.getString("direccion");
 String mail = extras.getString("mail");

 txtId = findViewById(R.id.txtIdEdit);
 txtNombre = findViewById(R.id.txtNombreEdit);
 txtTelefono = findViewById(R.id.txtTelefonoEdit);
 txtDireccion = findViewById(R.id.txtDireccionEdit);
 txtMail = findViewById(R.id.txtEmailEdit);

 txtId.setText(id);
 txtNombre.setText(nombre);
 txtTelefono.setText(telefono);
 txtDireccion.setText(direccion);
 txtMail.setText(mail);

 }

 public void btnCancelar(View view) {
 Intent btnCancelar = new Intent(this, MainActivity.class);
 startActivity(btnCancelar);
 }
}

```

```

 public void cmdUpdate_onClick(View v) {
 boolean resultado = contactos.Update(
 txtId.getText().toString(),
 txtNombre.getText().toString(),
 txtTelefono.getText().toString(),
 txtDireccion.getText().toString(),
 txtMail.getText().toString()
);
 if (resultado == true)
 Toast.makeText(this, "REGISTRO ACTUALIZADO OK",
 Toast.LENGTH_SHORT).show();
 else
 Toast.makeText(this, "ERROR!! REGISTRO NO ACTU-
 ALIZADO", Toast.LENGTH_SHORT).show();
 }

 public void cmdDelete_onClick(View v) {
 AlertDialog.Builder builder = new AlertDia-
 log.Builder(this);
 builder.setMessage("¿Está seguro de que desea elimi-
 nar este registro?")
 .setCancelable(false)
 .setPositiveButton("Sí", new DialogInter-
 face.OnClickListener() {
 public void onClick(DialogInterface dia-
 log, int id) {
 boolean resultado = contactos.De-
 delete(txtId.getText().toString());
 if (resultado == true) {
 Toast.makeText(getApplication-
 Context(), "REGISTRO BORRADO OK",
 Toast.LENGTH_SHORT).show();
 txtId.setText("");
 txtNombre.setText("");
 txtTelefono.setText("");
 txtDireccion.setText("");
 txtMail.setText("");
 } else {
 Toast.makeText(getApplication-
 Context(), "ERROR: REGISTRO NO ENCONTRADO !!!",
 Toast.LENGTH_SHORT).show();
 }
 }
 })
 .setNegativeButton("No", new DialogInter-
 face.OnClickListener() {
 public void onClick(DialogInterface dia-
 log, int id) {
 // No hacer nada si se selecciona
 "No"
 }
 });
 AlertDialog alert = builder.create();
 alert.show();
 }
 }
}

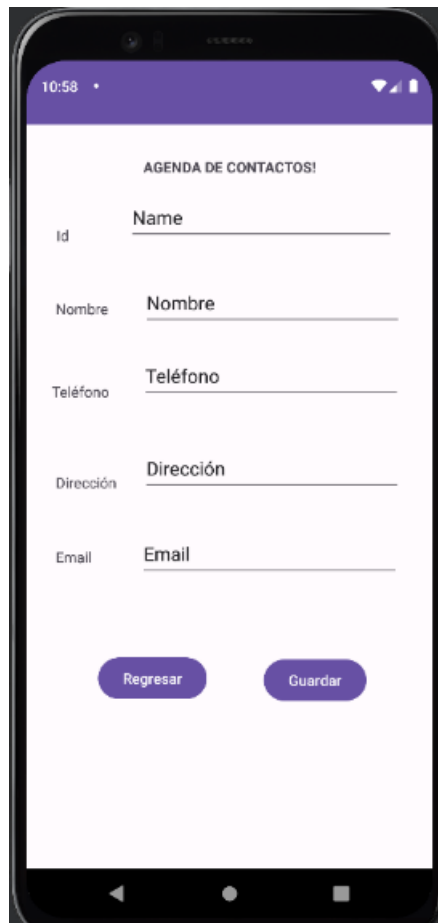
```

### 3. Pantallas creadas:

- Pantalla principal



- Pantalla para agregar un nuevo Contacto



- Pantalla para editar un registro de contacto

The screenshot shows a mobile application interface titled "AGENDA DE CONTACTOS!". It displays the details of a contact with ID "01". The fields are as follows:

| Field     | Value              |
|-----------|--------------------|
| Id        | 01                 |
| Nombre    | Roberth Lima       |
| Teléfono  | 0967732818         |
| Dirección | EL Olivo           |
| Email     | rglimac@utn.edu.ec |

At the bottom of the form, there are three buttons: "Eliminar" (left), "Editar" (right), and "Regresar" (centered below the others).

- Pantalla de confirmación para la eliminación de un registro de contacto

This screenshot shows the same contact details as the previous image, but with a confirmation dialog box overlaid in the center. The dialog box contains the text "¿Está seguro de que desea eliminar este registro?" and two buttons: "NO" and "SÍ".

A close-up of the confirmation dialog box, showing the text "¿Está seguro de que desea eliminar este registro?" and the "NO" and "SÍ" buttons.

4. Probando funcionalidades:

- Agregar

AGENDA DE CONTACTOS!

Id 06

Nombre Diego Trejo

Teléfono 0692315235

Dirección Ibarra

Email djtrejo@utn.edu.ec

Regresar Guardar

REGISTRO INSERTADO OK

AGENDA DE CONTACTOS!

Search icon Add

| ID | Nombre          | Teléfono   | Dirección          | Email    |
|----|-----------------|------------|--------------------|----------|
| 01 | Robert Lima     | 0967732818 | EL Olivo           | rglimac  |
| 02 | Dilan Ramirez   | 0956231807 | Otavalo            | drivald  |
| 03 | Jorge Jaramillo | 0963251201 | Residencia Sevilla | jajaram  |
| 05 | Daniel Añazco   | 0965231852 | El Olivo           | danx@    |
| 06 | Diego Trejo     | 0692315235 | Ibarra             | djtrejo@ |

| ID | Nombre          | Teléfono   | Dirección          | Email    |
|----|-----------------|------------|--------------------|----------|
| 01 | Robert Lima     | 0967732818 | EL Olivo           | rglimac  |
| 02 | Dilan Ramirez   | 0956231807 | Otavalo            | drivald  |
| 03 | Jorge Jaramillo | 0963251201 | Residencia Sevilla | jajaram  |
| 05 | Daniel Añazco   | 0965231852 | El Olivo           | danx@    |
| 06 | Diego Trejo     | 0692315235 | Ibarra             | djtrejo@ |

- Editar

11:06 •

AGENDA DE CONTACTOS!

Id 06

Nombre Diego Trejo

Teléfono 0692315235

Dirección Ibarra

Email djtrejo@utn.edu.ec

Eliminar Editar

Regresar

11:07 •

AGENDA DE CONTACTOS!

Id 06

Nombre Diego Trejo

Teléfono 0692315235

Dirección Tulcan

Email djtrejo@utn.edu.ec

Eliminar Editar

Regresar

Regresar

REGISTRO ACTUALIZADO OK

11:08 •

AGENDA DE CONTACTOS!

🔍 Add

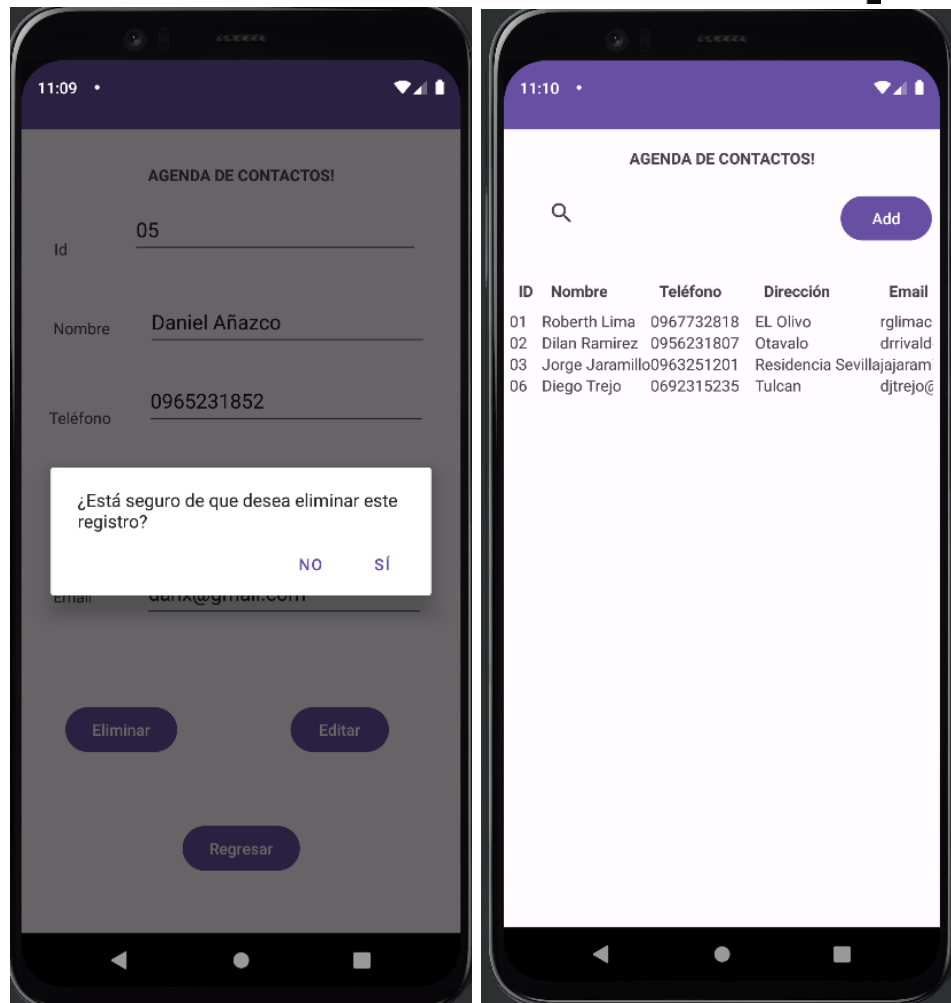
| ID | Nombre          | Teléfono   | Dirección          | Email    |
|----|-----------------|------------|--------------------|----------|
| 01 | Roberth Lima    | 0967732818 | EL Olivo           | rglimac  |
| 02 | Dilan Ramirez   | 0956231807 | Otavaló            | drivald  |
| 03 | Jorge Jaramillo | 0963251201 | Residencia Sevilla | jarami   |
| 05 | Daniel Añazco   | 0965231852 | El Olivo           | danx@    |
| 06 | Diego Trejo     | 0692315235 | Tulcan             | djtrejo@ |

| ID | Nombre          | Teléfono   | Dirección          | Email    |
|----|-----------------|------------|--------------------|----------|
| 01 | Roberth Lima    | 0967732818 | EL Olivo           | rglimac  |
| 02 | Dilan Ramirez   | 0956231807 | Otavaló            | drivald  |
| 03 | Jorge Jaramillo | 0963251201 | Residencia Sevilla | jarami   |
| 05 | Daniel Añazco   | 0965231852 | El Olivo           | danx@    |
| 06 | Diego Trejo     | 0692315235 | Tulcan             | djtrejo@ |



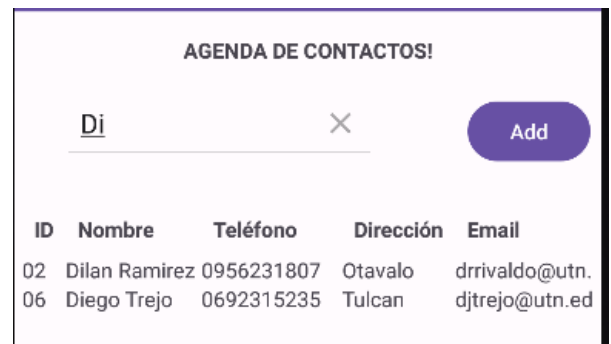
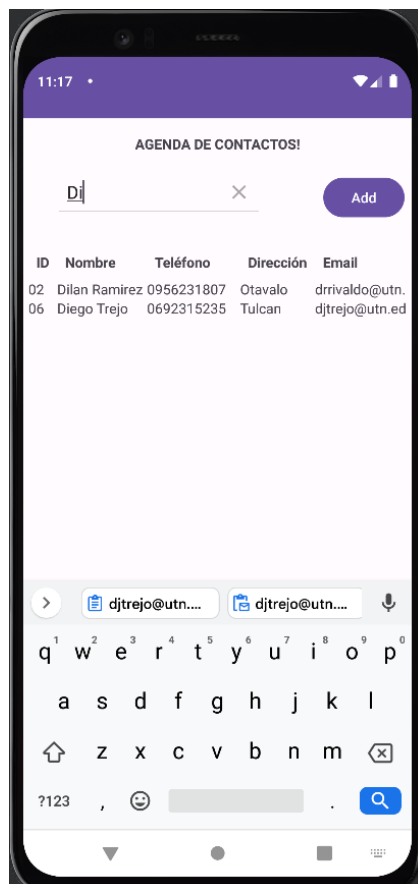
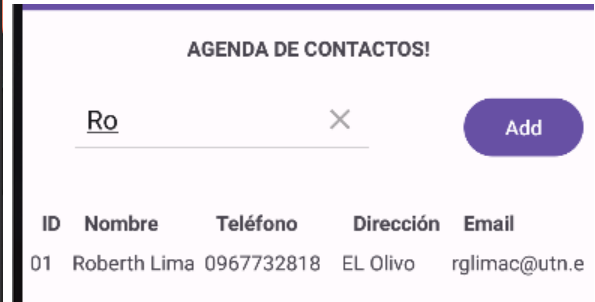
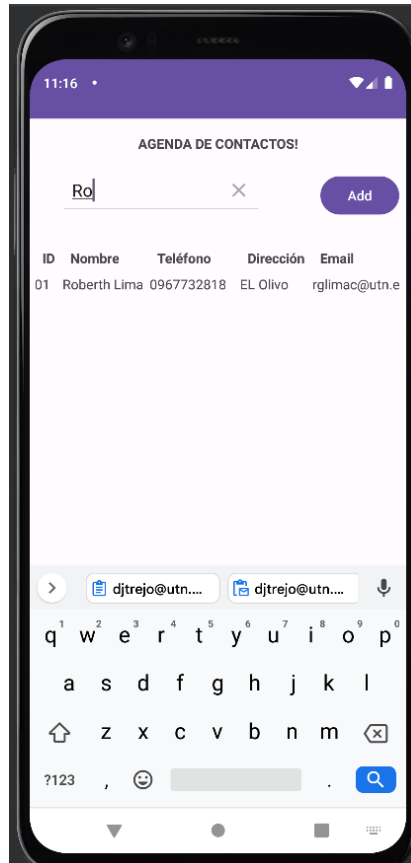
- Eliminar registro de contacto creado previamente con el nombre de Daniel Añazco:

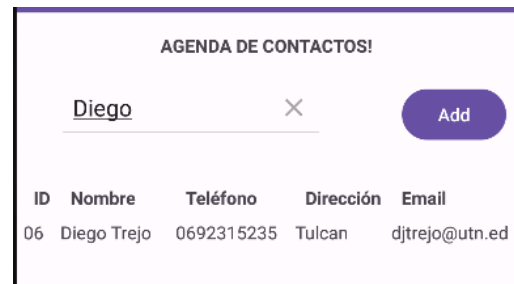
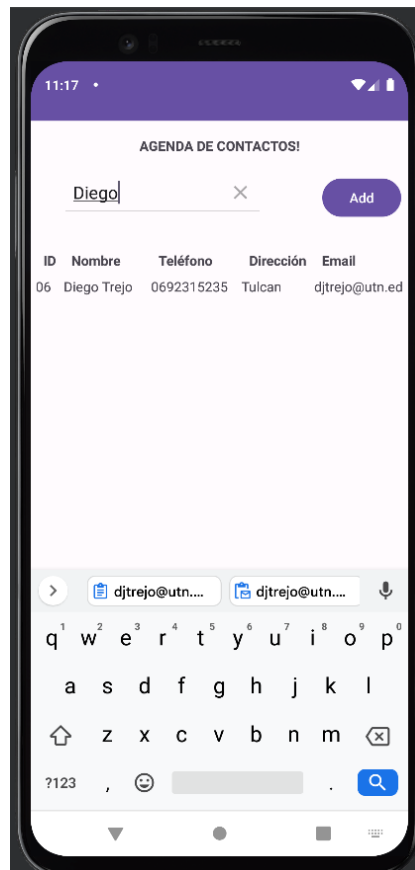
| ID | Nombre          | Teléfono   | Dirección          | Email    |
|----|-----------------|------------|--------------------|----------|
| 01 | Roberth Lima    | 0967732818 | EL Olivo           | rglimac  |
| 02 | Dilan Ramirez   | 0956231807 | Otavalo            | drivald  |
| 03 | Jorge Jaramillo | 0963251201 | Residencia Sevilla | jarami   |
| 05 | Daniel Añazco   | 0965231852 | El Olivo           | danx@    |
| 06 | Diego Trejo     | 0692315235 | Tulcan             | djtrejo@ |



| ID | Nombre          | Teléfono   | Dirección          | Email    |
|----|-----------------|------------|--------------------|----------|
| 01 | Roberth Lima    | 0967732818 | EL Olivo           | rglimac  |
| 02 | Dilan Ramirez   | 0956231807 | Otavalo            | drivald  |
| 03 | Jorge Jaramillo | 0963251201 | Residencia Sevilla | jarami   |
| 06 | Diego Trejo     | 0692315235 | Tulcan             | djtrejo@ |

- Buscar registro dentro de la tabla:





## CONCLUSIONES:

La creación de una app en Android Studio utilizando Java y SQLite para una agenda de contactos es una tarea factible y pese a parecer sencilla en realidad si representa un gran desafío debido a la complejidad que esto posee debido al poco conocimiento del tema de programación móvil y de base de datos externas a las ya utilizadas, dado que Android Studio proporciona herramientas y plantillas para el desarrollo de aplicaciones con este propósito nos resulta mucho más dinámico la realización de dicha app.

La utilización de una base de datos SQLite para almacenar y administrar los registros de contactos resulta eficiente y práctica, ya que SQLite es un sistema de gestión de bases de datos muy ligero y de fácil implementación en dispositivos móviles.

La implementación de las funciones CRUD (Crear, Leer, Actualizar y Eliminar) es fundamental para el correcto funcionamiento de la app, ya que permiten realizar operaciones básicas de manejo de datos sobre los registros de contactos.

La utilización de patrones de diseño de software, como el patrón Modelo-Vista-Controlador (MVC), puede resultar útil para organizar el código de la aplicación y facilitar su mantenimiento y escalabilidad.

El diseño de la interfaz de usuario (UI) de la app es un aspecto clave para brindar una buena experiencia de usuario (UX), y debe ser intuitiva y fácil de usar para facilitar la navegación y el uso de las funciones de la aplicación.