



Universidad Latina de Panamá

Facultad de Ingeniería

Nombre

Alexander Zhuo 8-965-959

Brian Rodríguez 8-949-684

Materia: Programación VI

Profesor: Oriel Cedeño

Laboratorio 6

2024

```
package com.example.lab_6;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

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

    public void Presentacion(View view){
        Intent i = new Intent(this, Presentation.class);
        startActivity(i);
    }

    public void Casos(View view){
        Intent i = new Intent(this,Cases.class);
        startActivity(i);
    }

}
```

Sentencias import: Importan las clases y paquetes necesarios para el código.

Clase MainActivity: Esta es la clase principal de la actividad que extiende AppCompatActivity.

Método onCreate: Este método se llama cuando se crea la actividad. Establece la vista de contenido al diseño especificado en activity_main.xml.

Método Presentacion: Este método está asociado con un clic de botón (probablemente en el diseño). Crea un intent para iniciar la actividad Presentation y luego inicia esa actividad.

Método Casos: Similar a Presentacion, este método está asociado con otro clic de botón. Crea un intent para iniciar la actividad Cases y luego inicia esa actividad.



```
package com.example.lab_6;

import android.os.Bundle;
import android.view.View;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class Presentation extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_presentation);

    }
    public void Salir(View view){
        finish();
    }
}
```

Como se puede observar en la aplicación se crearon textview para los textos, se agregó una imagen del logo de la universidad

`public void Salir(View view)`: Este método es llamado cuando se hace clic en el botón "Salir". Su función es finalizar la actividad actual, cerrando la ventana de presentación y volviendo a la actividad anterior o finalizando la aplicación.

7:46



Universidad
LATINA *de Panamá*
SUMMUM DESIDERIUM SAPIENTIA

Universidad Latina

Facultad de Ingeniería en Sistemas

Lic. De ingeniería en Sistemas Informáticos

Materia: Programación VI

Nombre del Profesor: Oriel Cedeño

Estudiantes: Brian Rodríguez 8-949-684

Alexander Zhuo 8-965-959

6 de marzo de 2024

Salir

```

package com.example.lab_6;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class Cases extends AppCompatActivity {

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

    }
    public void Calculadora(View view){
        Intent i = new Intent(this, Cal.class);
        startActivity(i);
    }
    public void Lista(View view){
        Intent i = new Intent(this, List.class);
        startActivity(i);
    }
    public void Salir(View view){
        finish();
    }
}

```

public void Calculadora(View view): Este método es llamado cuando se hace clic en el botón "Calculadora". Crea un intent para iniciar la actividad Cal y luego inicia esa actividad.

public void Lista(View view): Este método es llamado cuando se hace clic en el botón "Lista". Crea un intent para iniciar la actividad List y luego inicia esa actividad.

public void Salir(View view): Este método es llamado cuando se hace clic en el botón "Salir". Su función es finalizar la actividad actual, cerrando la ventana de casos y volviendo a la actividad anterior o finalizando la aplicación.



```

package com.example.lab_6;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class Cal extends AppCompatActivity {
    // 0->nada; 1->suma; 2->resta; 3->mult; 4->div
    private int oper = 0;
    private double numero1 = 0;
    private TextView tv_num1;
    private TextView tv_num2;
    private Button btnIgual2, btnBorrar2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cal);
        btnIgual2 = findViewById(R.id.btnIgual);
        btnBorrar2 = findViewById(R.id.btnBorrar);
        tv_num1 = findViewById(R.id.tv_num1);
        tv_num2 = findViewById(R.id.tv_num2);

        btnIgual2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                double num2 = Double.parseDouble(tv_num2.getText().toString());
                double res = 0.0;
                switch (oper) {
                    case 1:
                        res = numero1 + num2;
                        break;
                    case 2:
                        res = numero1 - num2;
                        break;
                    case 3:
                        res = numero1 * num2;
                        break;
                }
            }
        });
    }
}

```



```

        case 4:
            res = numero1 / num2;
            break;
        }
        tv_num2.setText(String.valueOf(res));
        tv_num1.setText("");
    }
});

btnBorrar2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        tv_num1.setText("");
        tv_num2.setText("");
        oper = 0;
    }
});
}

public void clicNumero(View v) {
    String num2 = tv_num2.getText().toString();

    int id = v.getId();

    if (id == R.id.btn0) {
        tv_num2.setText(num2 + "0");
    } else if (id == R.id.btn1) {
        tv_num2.setText(num2 + "1");
    } else if (id == R.id.btn2) {
        tv_num2.setText(num2 + "2");
    } else if (id == R.id.btn3) {
        tv_num2.setText(num2 + "3");
    } else if (id == R.id.btn4) {
        tv_num2.setText(num2 + "4");
    } else if (id == R.id.btn5) {
        tv_num2.setText(num2 + "5");
    } else if (id == R.id.btn6) {
        tv_num2.setText(num2 + "6");
    } else if (id == R.id.btn7) {
        tv_num2.setText(num2 + "7");
    } else if (id == R.id.btn8) {
        tv_num2.setText(num2 + "8");
    } else if (id == R.id.btn9) {
        tv_num2.setText(num2 + "9");
    }
}

```

```

        } else if (id == R.id.btnPunto) {
            tv_num2.setText(num2 + ".");
        }
    }

    public void clicOperacion(View view) {
        String num2 = tv_num2.getText().toString();
        numero1 = Double.parseDouble(num2);
        tv_num2.setText("");

        int id = view.getId();

        if (id == R.id.btnSumar) {
            tv_num1.setText(num2 + "+");
            oper = 1;
        } else if (id == R.id.btnRestar) {
            tv_num1.setText(num2 + "-");
            oper = 2;
        } else if (id == R.id.btnMult) {
            tv_num1.setText(num2 + "*");
            oper = 3;
        } else if (id == R.id.btnDividir) {
            tv_num1.setText(num2 + "/");
            oper = 4;
        }
    }

    public void Salir(View view){
        finish();
    }
}

```

La clase Cal es una actividad que maneja una calculadora simple.

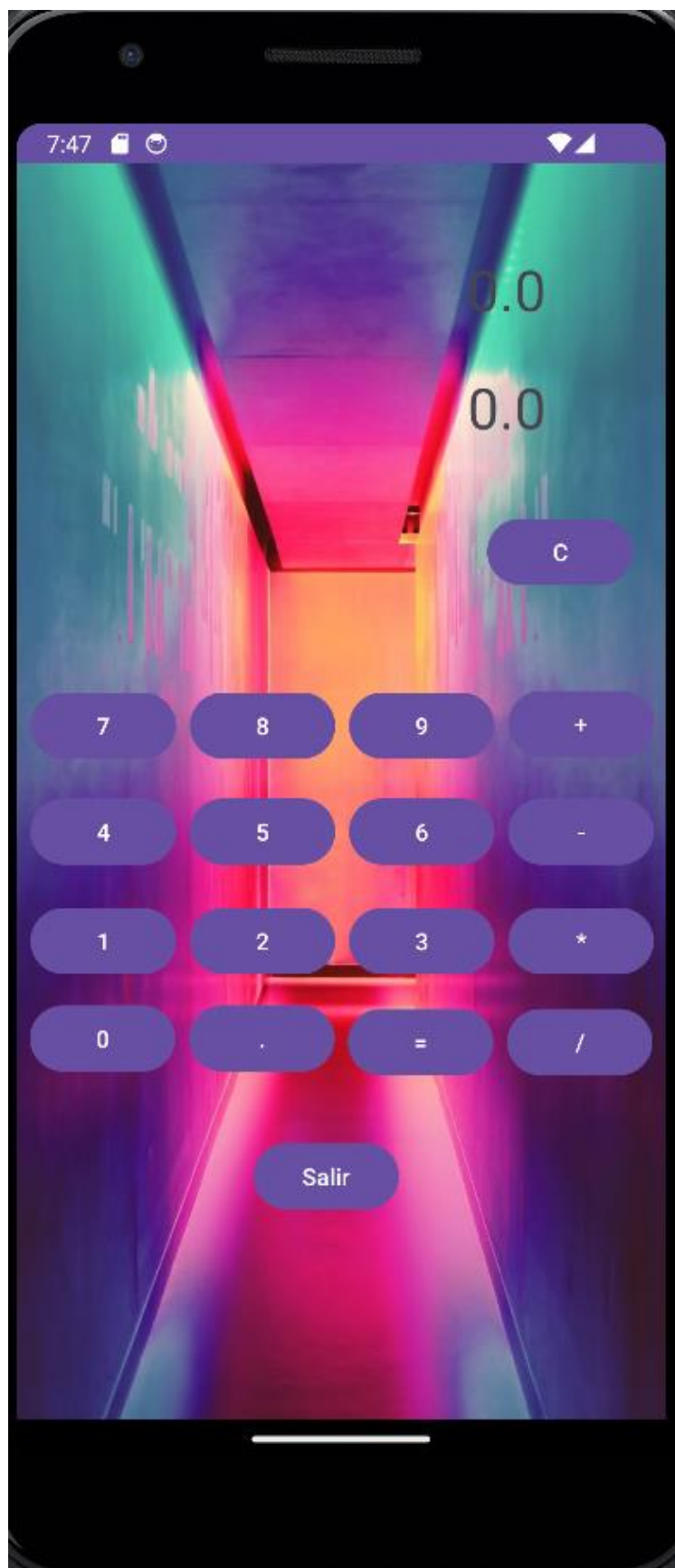
La actividad tiene elementos de interfaz de usuario como TextView para mostrar los números y el resultado, y botones para los dígitos, operaciones y funciones de la calculadora.

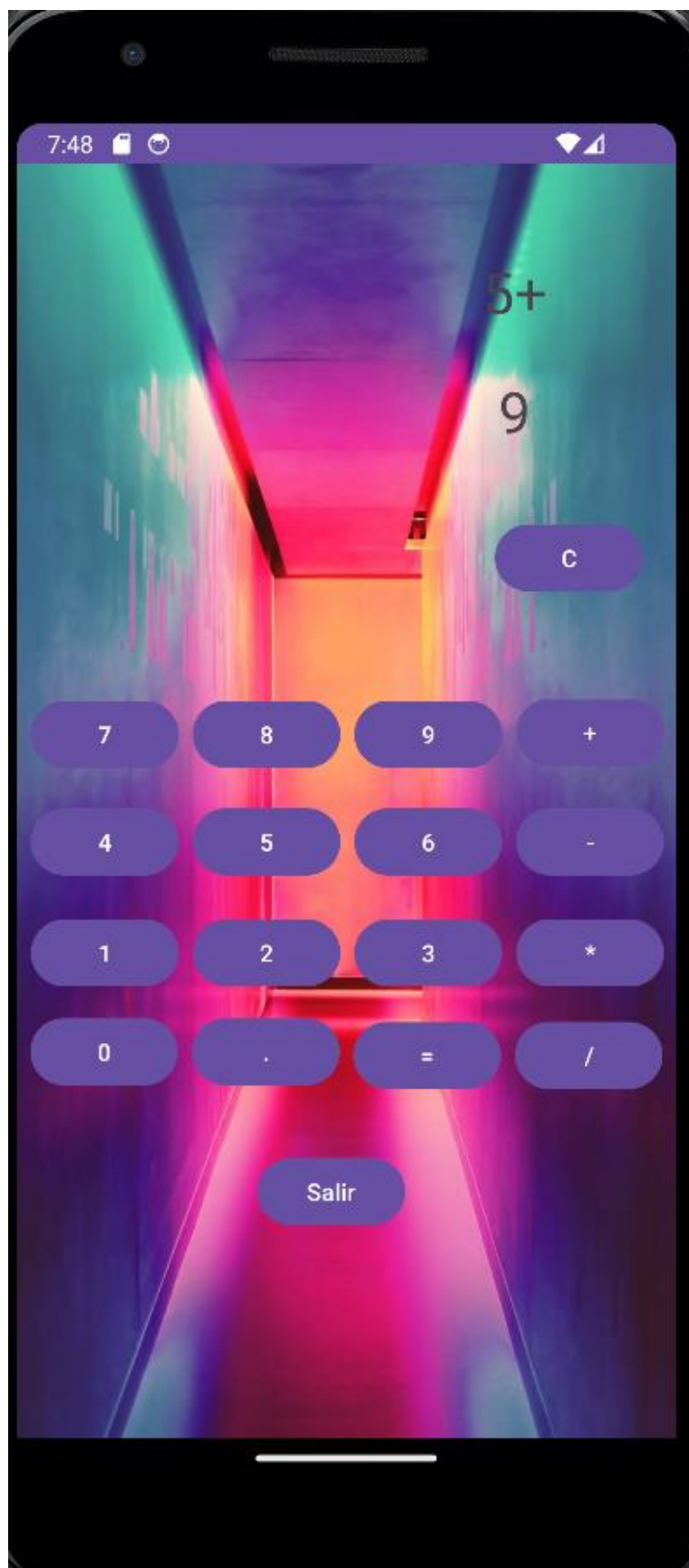
Se utiliza un conjunto de variables para realizar un seguimiento del estado de la calculadora, incluidos el primer número (numero1), el operador actual (oper), y referencias a las vistas (tv_num1 y tv_num2).

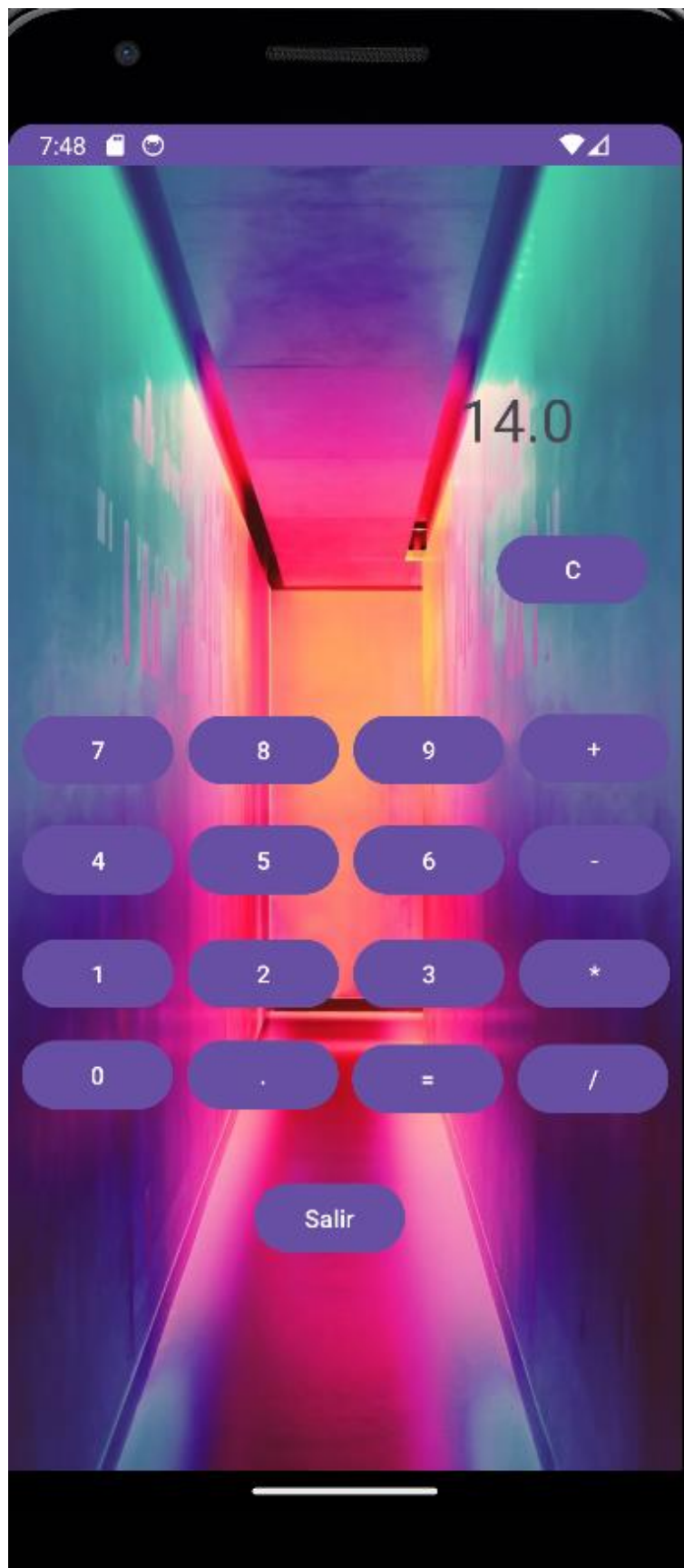
En el método onCreate, se inicializan los elementos de la interfaz de usuario y se configuran los eventos de clic para los botones de igual y borrar.

Los métodos clicNumero y clicOperacion son llamados cuando se hace clic en los botones numéricos y de operación, respectivamente. Estos métodos actualizan las vistas y las variables de estado.

El método Salir se utiliza para finalizar la actividad cuando se hace clic en el botón "Salir".







```

package com.example.lab_6;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ListView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.util.ArrayList;

public class List extends AppCompatActivity {
    private ListView LV1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list);
        LV1 = findViewById(R.id.LV);
        Button addButton = findViewById(R.id.Button);

        // Crear una lista de contactos
        java.util.List<Contact> contacts = new ArrayList<>();
        contacts.add(new Contact("John Doe", "123456789", "john@example.com"));
        // Añadir más contactos según sea necesario

        // Crear y configurar el adaptador personalizado
        CustomAdapter adapter = new CustomAdapter(this, R.layout.lists_items_contacts, contacts);
        LV1.setAdapter(adapter);
        addButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Agregar un nuevo contacto a la lista
                contacts.add(new Contact("Nuevo Contacto", "987654321", "nuevo@example.com"));
                // Notificar al adaptador que los datos han cambiado
                adapter.notifyDataSetChanged();
            }
        });
    }
    public void Salir(View view){
        finish();
    }
}

```

```
}  
}
```

```
package com.example.lab_6;
```

```
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.AdapterView;  
import android.widget.AdapterView;  
import android.widget.AdapterView;  
import android.widget.AdapterView;  
import android.widget.AdapterView;
```

```
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;
```

```
import java.util.List;
```

```
public class CustomAdapter extends ArrayAdapter<Contact> {  
    private Context mContext;  
    private int mResource;  
  
    public CustomAdapter(@NonNull Context context, int resource, @NonNull List<Contact> objects) {  
        super(context, resource, objects);  
        mContext = context;  
        mResource = resource;  
    }  
}
```

```
@NonNull
```

```
@Override
```

```
public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
    LayoutInflater inflater = LayoutInflater.from(mContext);  
    convertView = inflater.inflate(mResource, parent, false);
```

```
    ImageView imageView = convertView.findViewById(R.id.imageView);  
    TextView textViewName = convertView.findViewById(R.id.textViewName);  
    TextView textViewPhone = convertView.findViewById(R.id.textViewPhone);  
    TextView textViewEmail = convertView.findViewById(R.id.textViewEmail);
```

```
    Contact contact = getItem(position);
```

```
    if (contact != null) {  
        // Aquí deberías cargar la imagen del contacto en imageView
```

```

        textViewName.setText(contact.getName());
        textViewPhone.setText(contact.getPhoneNumber());
        textViewEmail.setText(contact.getEmail());
    }

    return convertView;
}
}

package com.example.lab_6;

public class Contact {
    private String name;
    private String phoneNumber;
    private String email;
    // Puedes agregar más campos según sea necesario

    // Constructor
    public Contact(String name, String phoneNumber, String email) {
        this.name = name;
        this.phoneNumber = phoneNumber;
        this.email = email;
    }

    // Getters y Setters
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    public String getEmail() {
        return email;
    }
}

```



```
    }  
  
    public void setEmail(String email) {  
        this.email = email;  
    }  
}
```

List es una actividad que muestra una lista de contactos en un ListView.

Se utiliza un botón (addButton) para agregar un nuevo contacto a la lista.

La lista de contactos (contacts) se crea y se llena con al menos un contacto inicial.

Se crea un adaptador personalizado (CustomAdapter) y se le asigna a la ListView.

Cuando se hace clic en el botón de agregar, se agrega un nuevo contacto a la lista y se notifica al adaptador que los datos han cambiado.

CustomAdapter extiende ArrayAdapter y trabaja con objetos de tipo Contact.

En el método getView, se infla la vista del elemento de la lista (convertView) a partir del diseño definido en lists_items_contacts.xml.

Se obtienen referencias a los elementos de la vista (ImageView y TextViews).

Se recupera el objeto Contact correspondiente a la posición actual en la lista.

Se actualizan los elementos de la vista con los datos del contacto.

Contact es una clase de modelo que representa un contacto.

Tiene campos para el nombre, número de teléfono y correo electrónico del contacto.

Se proporciona un constructor para inicializar los campos.

Se proporcionan métodos de acceso (getters y setters) para cada campo.

