Universidad
LATINA de Panamá
SUMMUM DESIDERIUM SAPIENTIA

Laboratorio #6

Docente: Prof. Oriel Cedeño

Asignatura: Programación VI

Alumno: Mark Trejos – 8-999-582
Kevin Tom- 8-959-1255

Carrera: Lic. Ingeniería en Sistemas

Cuatrimestre: 7mo

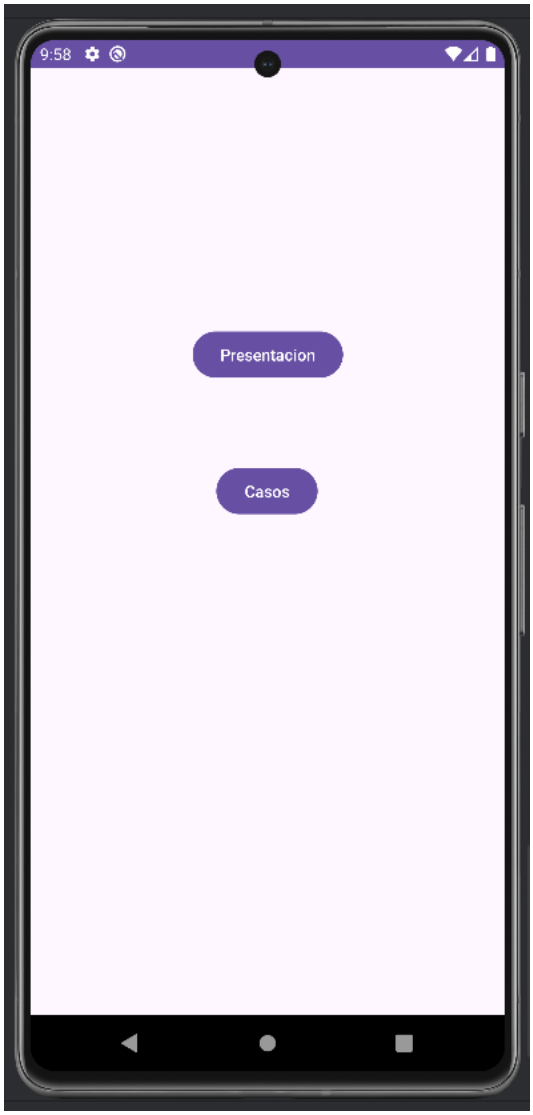
Año: 2024

MainActivity

Este código es una implementación básica de una actividad principal en una aplicación Android utilizando el entorno de desarrollo Android Studio. La clase MainActivity extiende AppCompatActivity y contiene dos métodos, presentacion y caso, que se llaman cuando se hace clic en elementos de la interfaz de usuario. Estos métodos crean instancias de la clase Intent para iniciar otras actividades llamadas presentacion y caso respectivamente.

```java
package com.example.myapplication;

import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class calculadora extends AppCompatActivity {

    private EditText editTextText;
    private TextView txt;
    private double resultado = 0;
    private String operador = "";

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

        editTextText = findViewById(R.id.editTextText);
        txt = findViewById(R.id.txt);

        // Definir los botones
        Button btnSum = findViewById(R.id.btnSum);
        Button btnSubtract = findViewById(R.id.btnSubtract);
        Button btnMultiply = findViewById(R.id.btnMultiply);
        Button btnDivide = findViewById(R.id.btnDivide);
        Button btnOperate = findViewById(R.id.btnOperate);
        Button btnClear = findViewById(R.id.btnClear);
        Button btnReturn = findViewById(R.id.button24);

        // Definir los botones numéricos
        Button btn0 = findViewById(R.id.btn0);
        Button btn1 = findViewById(R.id.btn1);
        Button btn2 = findViewById(R.id.btn2);
        Button btn3 = findViewById(R.id.btn3);
        Button btn4 = findViewById(R.id.btn4);
        Button btn5 = findViewById(R.id.btn5);
        Button btn6 = findViewById(R.id.btn6);
        Button btn7 = findViewById(R.id.btn7);
        Button btn8 = findViewById(R.id.btn8);
        Button btn9 = findViewById(R.id.btn9);

        // Configurar el OnClickListener para los botones de operaciones
        btnSum.setOnClickListener(v -> setOperator("+"));
        btnSubtract.setOnClickListener(v -> setOperator("-"));
        btnMultiply.setOnClickListener(v -> setOperator("*"));
        btnDivide.setOnClickListener(v -> setOperator("/"));
        btnOperate.setOnClickListener(v -> calculateResult());
        btnClear.setOnClickListener(v -> clearInput());
```

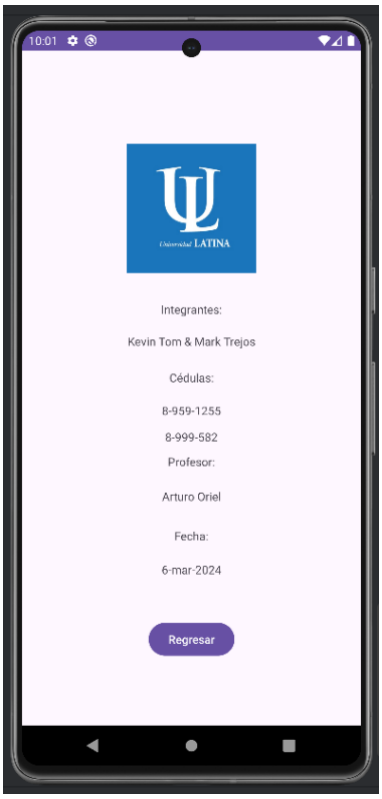
```
54.         btnReturn.setOnClickListener(v -> regresar());
55.
56.         // Configurar el OnClickListener para los botones numéricos
57.         btn0.setOnClickListener(v -> appendNumber("0"));
58.         btn1.setOnClickListener(v -> appendNumber("1"));
59.         btn2.setOnClickListener(v -> appendNumber("2"));
60.         btn3.setOnClickListener(v -> appendNumber("3"));
61.         btn4.setOnClickListener(v -> appendNumber("4"));
62.         btn5.setOnClickListener(v -> appendNumber("5"));
63.         btn6.setOnClickListener(v -> appendNumber("6"));
64.         btn7.setOnClickListener(v -> appendNumber("7"));
65.         btn8.setOnClickListener(v -> appendNumber("8"));
66.         btn9.setOnClickListener(v -> appendNumber("9"));
67.     }
68.
69.     private void setOperator(String operator) {
70.         operador = operator;
71.         resultado = Double.parseDouble(editTextText.getText().toString());
72.         editTextText.setText("");
73.     }
74.
75.     private void calculateResult() {
76.         if (!editTextText.getText().toString().isEmpty()) {
77.             double number = Double.parseDouble(editTextText.getText().toString());
78.             switch (operador) {
79.                 case "+":
80.                     resultado += number;
81.                     break;
82.                 case "-":
83.                     resultado -= number;
84.                     break;
85.                 case "*":
86.                     resultado *= number;
87.                     break;
88.                 case "/":
89.                     if (number != 0) {
90.                         resultado /= number;
91.                     } else {
92.                         editTextText.setText("Error: división entre 0");
93.                     }
94.                     break;
95.             }
96.             editTextText.setText(String.valueOf(resultado));
97.             operador = "";
98.         }
99.     }
100.
101.    private void appendNumber(String number) {
102.        editTextText.append(number);
103.    }
104.
105.    private void clearInput() {
106.        editTextText.setText("");
107.        resultado = 0;
108.        operador = "";
109.    }
110.
111.    private void regresar() {
112.        Intent i = new Intent(this, caso.class);
113.        startActivity(i);
114.    }
115. }
```


9:58
Presentacion
Casos

Presentacion

Este código representa una actividad en una aplicación Android. La clase presentacion extiende AppCompatActivity y se encarga de mostrar la interfaz de usuario definida en el archivo de diseño activity_presentacion.xml. La actividad tiene un único método llamado regresar, que se activa cuando se hace clic en un elemento de la interfaz de usuario. Este método utiliza el método finish() para cerrar la actividad actual y regresar a la actividad anterior en la pila de actividades.

```
 1. package com.example.myapplication;
 2.
 3. import androidx.appcompat.app.AppCompatActivity;
 4.
 5. import android.os.Bundle;
 6. import android.view.View;
 7.
 8. public class presentacion extends AppCompatActivity {
 9.
10.     @Override
11.     protected void onCreate(Bundle savedInstanceState) {
12.         super.onCreate(savedInstanceState);
13.         setContentView(R.layout.activity_presentacion);
14.     }
15.
16.     public void regresar(View view){
17.         finish();
18.     }
19. }
20.
```



Caso

Este código representa otra actividad en una aplicación Android llamada caso. Al igual que las anteriores, extiende AppCompatActivity y se encarga de mostrar la interfaz de usuario definida en el archivo de diseño activity_caso.xml. La actividad tiene dos métodos, calculadora y contactos, que se activan al hacer clic en elementos de la interfaz de usuario. Ambos métodos utilizan la clase Intent para iniciar otras actividades llamadas calculadora y contacto, respectivamente. Además, hay un

tercer método llamado regresar1 que utiliza finish() para cerrar la actividad actual y regresar a la actividad anterior en la pila de actividades.

```java
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;

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

public class caso extends AppCompatActivity {

    private Intent i;

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

    public void calculadora(View view){
        i = new Intent(this, calculadora.class);
        startActivity(i);
    }
    public void contactos(View view){
        i = new Intent(this, contacto.class);
        startActivity(i);
    }

    public void regresar1(View view){
        finish();
    }
}
```

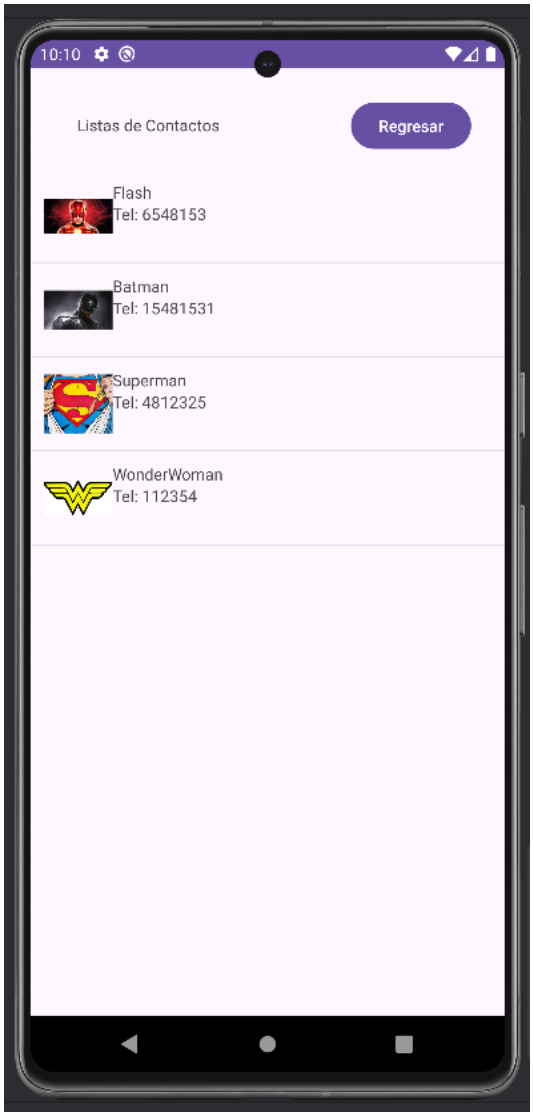
Contacto

Este código representa una actividad de una aplicación Android llamada contacto. La actividad extiende AppCompatActivity y se encarga de mostrar una lista de contactos en una interfaz de usuario definida en el archivo de diseño activity_contacto.xml. En el método onCreate, se inicializa un ListView llamado listView1, y se crea una lista de objetos Person que contienen información sobre algunos contactos ficticios (nombre, imagen y número de teléfono). Estos datos se pasan a un adaptador personalizado llamado PersonAdapter, que a su vez se asigna al ListView para mostrar la información en la interfaz de usuario.

La interfaz de usuario consiste en elementos de lista, donde cada elemento muestra la imagen, nombre y número de teléfono de un contacto. Además, hay un botón de regreso que utiliza el método finish() para cerrar la actividad actual y regresar a la actividad anterior en la pila de actividades cuando se hace clic.

```
1. package com.example.myapplication;
2.
3. import androidx.appcompat.app.AppCompatActivity;
4.
5. import android.os.Bundle;
6. import android.view.View;
```

```
 7. import android.widget.ArrayAdapter;
 8. import android.widget.ListView;
 9.
10. import java.util.ArrayList;
11.
12. public class contacto extends AppCompatActivity {
13.
14.     private ListView listView1;
15.
16.
17.     @Override
18.     protected void onCreate(Bundle savedInstanceState) {
19.         super.onCreate(savedInstanceState);
20.         setContentView(R.layout.activity_contacto);
21.
22.         listView1 = findViewById(R.id.listView1);
23.
24.         ArrayList<Person> arrayList = new ArrayList<>();
25.         arrayList.add(new Person(R.drawable.dc1, "Flash", "Tel: 6548153"));
26.         arrayList.add(new Person(R.drawable.dc2, "Batman", "Tel: 15481531"));
27.         arrayList.add(new Person(R.drawable.dc3, "Superman", "Tel: 4812325"));
28.         arrayList.add(new Person(R.drawable.dc4, "WonderWoman", "Tel: 112354"));
29.
30.         PersonAdapter personAdapter = new PersonAdapter(this, R.layout.item_list, arrayList);
31.
32.         listView1.setAdapter(personAdapter);
33.     }
34.
35.     public void regresar(View view){
36.         finish();
37.     }
38.
39. }
```



Person

Esta clase es utilizada en conjunto con el código anterior para crear objetos **Person** que representan contactos, y estos objetos se utilizan en un adaptador personalizado para mostrar la información en una lista en la interfaz de usuario de la actividad "contacto" de la aplicación Android.
El constructor de la clase **Person** recibe estos tres valores como parámetros y los asigna a los atributos correspondientes. Además, se proporcionan métodos getter y setter para cada atributo, lo que permite acceder y modificar estos valores desde fuera de la clase.

```
 1. package com.example.myapplication;
 2.
 3. public class Person {
 4.     int img;
 5.     String name;
 6.     String desc;
 7.
 8.     public Person(int img, String name, String desc) {
 9.         this.img = img;
10.         this.name = name;
11.         this.desc = desc;
12.     }
```

```
13.
14.     public int getImg() {
15.         return img;
16.     }
17.
18.     public void setImg(int img) {
19.         this.img = img;
20.     }
21.
22.     public String getName() {
23.         return name;
24.     }
25.
26.     public void setName(String name) {
27.         this.name = name;
28.     }
29.
30.     public String getDesc() {
31.         return desc;
32.     }
33.
34.     public void setDesc(String desc) {
35.         this.desc = desc;
36.     }
37. }
38.
```

Pd: No tiene vista.

PersonAdapter

Este código define una clase llamada PersonAdapter que extiende ArrayAdapter en Android. Su propósito es proporcionar una interfaz de conexión entre la lista de objetos Person y la vista de lista (ListView) en la interfaz de usuario. A continuación, se describen las características clave de esta clase:
Constructor: El constructor toma tres parámetros: el contexto de la aplicación, el recurso de diseño para cada elemento de la lista (resource), y la lista de objetos Person a ser adaptada. Estos parámetros son pasados al constructor de la clase base (ArrayAdapter) utilizando la palabra clave super, y también se almacenan en variables miembro de la clase (mContext y mResrouce).
Método getView: Este método se sobrescribe para personalizar la visualización de cada elemento en la lista. Se utiliza un LayoutInflater para inflar la vista a partir del recurso de diseño (mResrouce). Luego, se obtienen referencias a los elementos de la vista (ImageView y dos TextViews) y se establecen con los valores correspondientes del objeto Person en la posición actual de la lista.

```
 1. package com.example.myapplication;
 2.
 3. import android.content.Context;
 4. import android.view.LayoutInflater;
 5. import android.view.View;
 6. import android.view.ViewGroup;
 7. import android.widget.ArrayAdapter;
 8. import android.widget.ImageView;
 9. import android.widget.TextView;
10.
11. import androidx.annotation.NonNull;
```

```java
import androidx.annotation.Nullable;

import java.util.ArrayList;

public class PersonAdapter extends ArrayAdapter<Person> {

    private Context mContext;
    private int mResrouce;
    public PersonAdapter(@NonNull Context context, int resource, @NonNull ArrayList<Person>
objects) {
        super(context, resource, objects);
        this.mContext = context;
        this.mResrouce = resource;
    }

    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {

        LayoutInflater layoutInflater = LayoutInflater.from(mContext);

        convertView = layoutInflater.inflate(mResrouce, parent, false);

        ImageView imageView = convertView.findViewById(R.id.imagen);
        TextView txtName = convertView.findViewById(R.id.texto1);
        TextView txtDesc = convertView.findViewById(R.id.texto2);

        imageView.setImageResource(getItem(position).getImg());
        txtName.setText(getItem(position).getName());
        txtDesc.setText(getItem(position).getDesc());

        return convertView;
    }
}
```

Pd: No tiene vista.

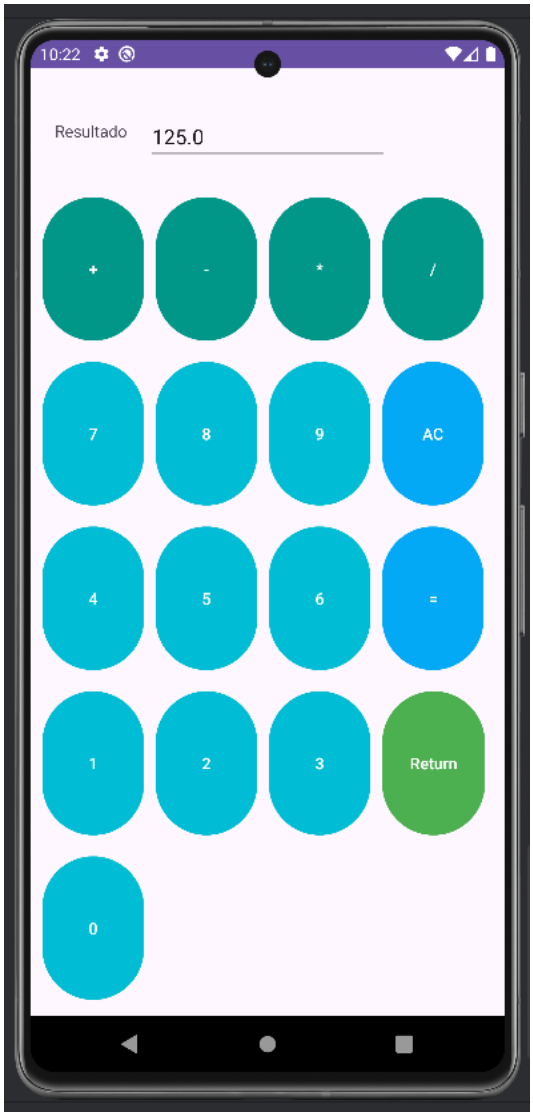
Calculadora

Este código representa una actividad de calculadora en una aplicación Android.
Variables de instancia: Se declaran varias variables de instancia, como editTextText para la entrada de la calculadora, txt para mostrar el resultado, resultado para almacenar el resultado de las operaciones, y operador para mantener el operador seleccionado.
Método onCreate: En este método, se establece la interfaz de usuario definida en activity_calculadora.xml y se asignan los botones y elementos de la interfaz a las variables correspondientes. También se configuran los OnClickListener para los botones numéricos y de operaciones.
Métodos para operaciones: Se definen métodos para realizar operaciones básicas de una calculadora, como setOperator para establecer el operador seleccionado, calculateResult para realizar el cálculo y mostrar el resultado, appendNumber para agregar números a la entrada, y clearInput para borrar la entrada y reiniciar la calculadora.
Método regresar: Este método utiliza un Intent para volver a la actividad "caso" cuando se hace clic en el botón de retorno.

```java
package com.example.myapplication;
```

```java

import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class calculadora extends AppCompatActivity {

    private EditText editTextText;
    private TextView txt;
    private double resultado = 0;
    private String operador = "";

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

        editTextText = findViewById(R.id.editTextText);
        txt = findViewById(R.id.txt);

        // Definir los botones
        Button btnSum = findViewById(R.id.btnSum);
        Button btnSubtract = findViewById(R.id.btnSubtract);
        Button btnMultiply = findViewById(R.id.btnMultiply);
        Button btnDivide = findViewById(R.id.btnDivide);
        Button btnOperate = findViewById(R.id.btnOperate);
        Button btnClear = findViewById(R.id.btnClear);
        Button btnReturn = findViewById(R.id.button24);

        // Definir los botones numéricos
        Button btn0 = findViewById(R.id.btn0);
        Button btn1 = findViewById(R.id.btn1);
        Button btn2 = findViewById(R.id.btn2);
        Button btn3 = findViewById(R.id.btn3);
        Button btn4 = findViewById(R.id.btn4);
        Button btn5 = findViewById(R.id.btn5);
        Button btn6 = findViewById(R.id.btn6);
        Button btn7 = findViewById(R.id.btn7);
        Button btn8 = findViewById(R.id.btn8);
        Button btn9 = findViewById(R.id.btn9);

        // Configurar el OnClickListener para los botones de operaciones
        btnSum.setOnClickListener(v -> setOperator("+"));
        btnSubtract.setOnClickListener(v -> setOperator("-"));
        btnMultiply.setOnClickListener(v -> setOperator("*"));
        btnDivide.setOnClickListener(v -> setOperator("/"));
        btnOperate.setOnClickListener(v -> calculateResult());
        btnClear.setOnClickListener(v -> clearInput());
        btnReturn.setOnClickListener(v -> regresar());

        // Configurar el OnClickListener para los botones numéricos
        btn0.setOnClickListener(v -> appendNumber("0"));
        btn1.setOnClickListener(v -> appendNumber("1"));
        btn2.setOnClickListener(v -> appendNumber("2"));
        btn3.setOnClickListener(v -> appendNumber("3"));
        btn4.setOnClickListener(v -> appendNumber("4"));
        btn5.setOnClickListener(v -> appendNumber("5"));
        btn6.setOnClickListener(v -> appendNumber("6"));
        btn7.setOnClickListener(v -> appendNumber("7"));
        btn8.setOnClickListener(v -> appendNumber("8"));
        btn9.setOnClickListener(v -> appendNumber("9"));
```

```
    }

    private void setOperator(String operator) {
        operador = operator;
        resultado = Double.parseDouble(editTextText.getText().toString());
        editTextText.setText("");
    }

    private void calculateResult() {
        if (!editTextText.getText().toString().isEmpty()) {
            double number = Double.parseDouble(editTextText.getText().toString());
            switch (operador) {
                case "+":
                    resultado += number;
                    break;
                case "-":
                    resultado -= number;
                    break;
                case "*":
                    resultado *= number;
                    break;
                case "/":
                    if (number != 0) {
                        resultado /= number;
                    } else {
                        editTextText.setText("Error: división entre 0");
                    }
                    break;
            }
            editTextText.setText(String.valueOf(resultado));
            operador = "";
        }
    }

    private void appendNumber(String number) {
        editTextText.append(number);
    }

    private void clearInput() {
        editTextText.setText("");
        resultado = 0;
        operador = "";
    }

    private void regresar() {
        Intent i = new Intent(this, caso.class);
        startActivity(i);
    }
}
```


10:22
Resultado
125.0
+
-
*
/
7
8
9
AC
4
5
6
=
1
2
3
Return
0