

Documentación Actividad 09

ListView

Desarrollo móvil integral

Jose Manuel Bautista Morales

Docente: ING. Abel Jerónimo Vargas

10 de noviembre de 2023

9.º GRADO

ÍNDICE

ÍNDICE

DESARROLLO

código java

1. MainActivity.java

2. GeometricFugure.java

3. GeometricFugureAdapter.java

4. FigureDetailsActivity.java

Código XML

1. Activity Main.xml

EJECUCIÓN

DESARROLLO

código java

1. MainActivity.java

```
1 package com.example.act_09_listview_personalizado;
2 import androidx.appcompat.app.AppCompatActivity;
3 import android.os.Bundle;
4 import android.view.View;
5 import android.widget.AdapterView;
6 import android.widget.AdapterView.OnItemClickListener;
7 import android.widget.ArrayAdapter;
8 import android.widget.ListView;
9 import java.util.ArrayList;
10
11 public class MainActivity extends AppCompatActivity {
12
13     private ListView listViewFigures;
14     private GeometricFigureAdapter adapter;
15     private ArrayList<GeometricFigure> figuresList;
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21
22         figuresList = new ArrayList<>();
23         setupGeometricFigures();
24
25         adapter = new GeometricFigureAdapter( context: this, R.layout.list_item_figure, figuresList);
26         listViewFigures = findViewById(R.id.listViewFigures);
27         listViewFigures.setAdapter(adapter);
28
29         listViewFigures.setOnItemClickListener(new AdapterView.OnItemClickListener() {
30             @Override
31             public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
32                 handleFigureClick(position);
33             }
34         });
35     }
36 }
```

- **Creación de la Actividad (MainActivity):**
 - La clase **MainActivity** extiende **AppCompatActivity**, que es una clase base para las actividades de Android.
 - Se sobrescribe el método **onCreate** que se llama cuando se crea la actividad.
- **Inicialización de la Interfaz de Usuario:**
 - **setContentView(R.layout.activity_main);**: Establece la interfaz de usuario de esta actividad utilizando el archivo de diseño XML llamado **activity_main**.

- `listViewFigures = findViewById(R.id.listViewFigures);`
Obtiene una referencia al `ListView` definido en el diseño.
- **Configuración de la Lista de Figuras:**
 - `figuresList`: Una lista de objetos `GeometricFigure` que representan diferentes figuras geométricas.
 - `setupGeometricFigures()`: Método que inicializa la lista de figuras geométricas con algunos valores iniciales.
- **Adaptador (`GeometricFigureAdapter`):**
 - `adapter`: Un adaptador personalizado (`GeometricFigureAdapter`) que conecta la lista de figuras (`figuresList`) al `ListView`.
 - `adapter = new GeometricFigureAdapter(this, R.layout.list_item_figure, figuresList);` Crea un nuevo adaptador personalizado.
- **Manejo de Clics en Elementos de la Lista:**
 - `listViewFigures.setOnItemClickListener(...)`: Establece un listener para manejar clics en elementos de la lista. Cuando un elemento se hace clic, se llama a `handleFigureClick(position)`.
- **Manejo de Clics en Figuras (`handleFigureClick`):**
 - Obtiene la figura geométrica correspondiente al elemento de la lista que se hizo clic.
 - Crea un intent para iniciar la `FigureDetailsActivity` y pasa el nombre de la figura seleccionada como un extra llamado "selectedFigureName".
 - Si la figura es un "Triángulo Escaleno", podría agregar lógica adicional para pasar información específica a esa figura.

```

        listViewFigures.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                handleFigureClick(position);
            }
        });
    }

    1 usage  @ Joseju
    private void setupGeometricFigures() {
        figuresList.add(new GeometricFigure( name:"Hexágono", R.drawable.hexagono, perimeterFormula:"6 × L", areaFormula:" (3√3/2) × L^2", volumeFormula:""));
        figuresList.add(new GeometricFigure( name:"Triángulo Equilátero", R.drawable.triangulo, perimeterFormula:" 3 × L", areaFormula:" (√3/4) × L^2", volumeFormula:""));
        figuresList.add(new GeometricFigure( name:"Cuadrado", R.drawable.cuadrado, perimeterFormula:" L × 4 ", areaFormula:"L^2 ", volumeFormula:""));
        figuresList.add(new GeometricFigure( name:"Cubo", R.drawable.cubo, perimeterFormula:" 12 × L", areaFormula:" 6 × L^2", volumeFormula:" L^3"));
    }

    1 usage  @ Joseju
    private void handleFigureClick(int position) {
        GeometricFigure clickedFigure = figuresList.get(position);

        Intent intent = new Intent( packageContext.this, FigureDetailsActivity.class);
        intent.putExtra( name:"selectedFigureName", clickedFigure.getName());

        if ("Triángulo Escaleno".equals(clickedFigure.getName())) {
            // Agregar lógica para pasar información adicional específica para el triángulo escaleno
        }

        startActivity(intent);
    }
}

```

2. GeometricFigure.java

```

package com.example.act_09_listview_personalizado;

9 usages  @ Joseju
public class GeometricFigure {
    2 usages
    private String name;
    2 usages
    private int imageResource;
    2 usages
    private String perimeterFormula;
    2 usages
    private String areaFormula;
    2 usages
    private String volumeFormula;

    4 usages  @ Joseju
    public GeometricFigure(String name, int imageResource, String perimeterFormula, String areaFormula, String volumeFormula) {
        this.name = name;
        this.imageResource = imageResource;
        this.perimeterFormula = perimeterFormula;
        this.areaFormula = areaFormula;
        this.volumeFormula = volumeFormula;
    }

    4 usages  @ Joseju
    public String getName() {
        return name;
    }

    1 usage  @ Joseju
    public int getImageResource() {
        return imageResource;
    }
}

```

Este código define una clase llamada `GeometricFigure` en Java. A continuación, se proporciona una explicación de la clase:

- **Atributos de la Clase:**
 - `name`: Representa el nombre de la figura geométrica.
 - `imageResource`: Es un identificador de recurso de imagen que podría usarse para representar visualmente la figura geométrica.
 - `perimeterFormula`: Fórmula para el perímetro de la figura geométrica.
 - `areaFormula`: Fórmula para el área de la figura geométrica.
 - `volumeFormula`: Fórmula para el volumen de la figura geométrica.
- **Constructor:**
 - `public GeometricFigure(String name, int imageResource, String perimeterFormula, String areaFormula, String volumeFormula)`: Es el constructor de la clase que se utiliza para crear instancias de `GeometricFigure`. Recibe valores para todos los atributos y los asigna a las variables de instancia correspondientes.
- **Métodos Get:**
 - `getName()`: Devuelve el nombre de la figura.
 - `getImageResource()`: Devuelve el identificador de recurso de imagen.
 - `getPerimeterFormula()`: Devuelve la fórmula del perímetro.
 - `getAreaFormula()`: Devuelve la fórmula del área.
 - `getVolumeFormula()`: Devuelve la fórmula del volumen.

```

        this.volumeFormula = volumeFormula;
    }

    4 usages  👤 joseju
    public String getName() {
        return name;
    }

    1 usage  👤 joseju
    public int getImageResource() {
        return imageResource;
    }

    1 usage  👤 joseju
    public String getPerimeterFormula() {
        return perimeterFormula;
    }

    1 usage  👤 joseju
    public String getAreaFormula() {
        return areaFormula;
    }

    1 usage  👤 joseju
    public String getVolumeFormula() {
        return volumeFormula;
    }
}

```

3. GeometricFugureAdapter.java

```

public class GeometricFigureAdapter extends ArrayAdapter<GeometricFigure> {
    1 usage  ▲ joseju
    public GeometricFigureAdapter(Context context, int resource, ArrayList<GeometricFigure> objects) {
        super(context, resource, objects);
    }
    ▲ joseju *
    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
        if (convertView == null) {
            convertView = LayoutInflater.from(getContext()).inflate(R.layout.list_item_figure, parent, attachToRoot: false);
        }
        GeometricFigure figure = getItem(position);

        ImageView imageFigure = convertView.findViewById(R.id.imageFigure);
        TextView textFigureName = convertView.findViewById(R.id.textFigureName);
        TextView textPerimeter = convertView.findViewById(R.id.textPerimeter);
        TextView textArea = convertView.findViewById(R.id.textArea);
        TextView textVolume = convertView.findViewById(R.id.textVolume);

        if (figure != null) {
            imageFigure.setImageResource(figure.getImageResource());
            textFigureName.setText(figure.getName());
            textPerimeter.setText("Perímetro: " + figure.getPerimeterFormula());
            textArea.setText("Área: " + figure.getAreaFormula());

            if ("Cubo".equals(figure.getName())) {
                textVolume.setText("Volumen: " + figure.getVolumeFormula());
                textVolume.setVisibility(View.VISIBLE);
            } else {
                textVolume.setVisibility(View.GONE);
            }
        }

        return convertView;
    }
}

```

- **Extensión de ArrayAdapter:**
 - La clase `GeometricFigureAdapter` extiende `ArrayAdapter<GeometricFigure>`, lo que significa que es un adaptador diseñado para trabajar con un conjunto de datos compuesto por objetos de tipo `GeometricFigure`.
- **Constructor:**
 - `public GeometricFigureAdapter(Context context, int resource, ArrayList<GeometricFigure> objects)`: Es el constructor de la clase. Recibe un contexto, un identificador de recurso y una lista de objetos `GeometricFigure`. Llama al constructor de la clase base (`ArrayAdapter`) para inicializar el adaptador.
- **Método `getView`:**
 - `public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent)`: Este método se llama cuando se necesita una vista para mostrar un elemento en la posición específica dentro del conjunto de datos. Aquí se infla la vista a partir del archivo de diseño `list_item_figure.xml` si aún no se ha creado. Luego, se asignan los valores adecuados de la figura geométrica a los elementos de la vista.
- **Asignación de Datos:**
 - Se obtiene el objeto `GeometricFigure` en la posición actual.
 - Se encuentran las vistas dentro del elemento de la lista (`ImageView`, `TextView` para nombre, perímetro, área y volumen).

- Se asignan los valores correspondientes a estas vistas utilizando los métodos `get` de la clase `GeometricFigure`.
- Se realiza un control de visibilidad para el campo de volumen. En este caso, se muestra solo si la figura es un "Cubo".

4. FigureDetailsActivity.java

- Declaración de Variables:

```

2 usages
private EditText editTextLado;
2 usages
private EditText editTextAltura;
2 usages
private EditText editTextThirdSide; // Nuevo EditText para el tercer lado
2 usages
private Button buttonCalcular;
4 usages
private TextView textResultado;
2 usages
private TextView textTituloFigura;

```

- `editTextLado`, `editTextAltura`, `editTextThirdSide`: Representan los campos de entrada para el lado, altura y tercer lado (en el caso del triángulo escaleno) respectivamente.
- `buttonCalcular`: Es el botón que el usuario presiona para realizar los cálculos.
- `textResultado`: Es el campo de texto donde se mostrarán los resultados de los cálculos.
- **Método `onCreate`:**

```

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

    editTextLado = findViewById(R.id.editTextLado);
    editTextAltura = findViewById(R.id.editTextAltura);
    editTextThirdSide = findViewById(R.id.editTextThirdSide);
    buttonCalcular = findViewById(R.id.buttonCalcular);
    textResultado = findViewById(R.id.textResultado);
    textTituloFigura = findViewById(R.id.textTituloFigura);

    String figuraSeleccionada = getIntent().getStringExtra( name: "selectedFigureName");
    setTitle("Detalles de " + figuraSeleccionada);

    joseju
    buttonCalcular.setOnClickListener(new View.OnClickListener() {
        joseju
        @Override
        public void onClick(View v) {
            calcularResultado(figuraSeleccionada);
        }
    });

    // Establece el título de la figura
    textTituloFigura.setText("Detalles de " + figuraSeleccionada);
}

```

- Este método se ejecuta cuando la actividad se crea.

- Se establece el contenido de la actividad desde el archivo XML `activity_figure_details`.
- Se obtienen referencias a los elementos de la interfaz de usuario.
- Se configura un `OnClickListener` para el botón de cálculo.
- **Método `calcularResultado`:**

```
1 usage  + joseju
private void calcularResultado(String figuraSeleccionada) {
    double lado = Double.parseDouble(editTextLado.getText().toString());
    double altura = 0;
    double tercerLado = 0;

    if (figuraSeleccionada.equals("Triángulo Escaleno")) {
        String alturaStr = editTextAltura.getText().toString();
        String tercerLadoStr = editTextThirdSide.getText().toString();

        if (!alturaStr.isEmpty() && !tercerLadoStr.isEmpty()) {
            altura = Double.parseDouble(alturaStr);
            tercerLado = Double.parseDouble(tercerLadoStr);
        } else {
            textResultado.setText("Ingrese valores para la altura y el tercer lado del triángulo escaleno");
            return;
        }
    }

    // Realizar cálculos según la figura seleccionada
    double area = 0;
    double perimetro = 0;
    double volumen = 0; // Agregado para el caso del cubo

    switch (figuraSeleccionada) {
        case "Hexágono":
            area = calcularAreaHexagono(lado);
            perimetro = calcularPerimetroHexagono(lado);
            break;
        case "Triángulo Equilátero":
            area = calcularAreaTrianguloEquilatero(lado);
            perimetro = calcularPerimetroTrianguloEquilatero(lado);
            break;
        case "Cuadrado":
    
```

- Este método se llama cuando el usuario presiona el botón de cálculo.
- Se obtienen los valores ingresados por el usuario para el lado, altura y tercer lado (si es un triángulo escaleno).
- Se realiza un conjunto de cálculos específicos según la figura seleccionada usando un `switch`.
- Se muestra el resultado en el campo `textResultado`. Para el caso del cubo, también se muestra el volumen.
- **Métodos de Cálculo Específicos para Cada Figura:**

```

private double calcularAreaCuadrado(double lado) {
    return lado * lado;
}

1 usage  joseju
private double calcularPerimetroCuadrado(double lado) {
    // Perimetro del cuadrado
    return 4 * lado;
}

// Funciones de cálculo específicas para cada figura
1 usage  joseju
private double calcularAreaHexagono(double lado) {
    return (3 * Math.sqrt(3) / 2) * Math.pow(lado, 2);
}

1 usage  joseju
private double calcularPerimetroHexagono(double lado) {
    return 6 * lado;
}

1 usage  joseju
private double calcularAreaTrianguloEquilatero(double lado) {
    return (Math.sqrt(3) / 4) * Math.pow(lado, 2);
}

1 usage  joseju
private double calcularPerimetroTrianguloEquilatero(double lado) {
    return 3 * lado;
}

```

- Se tienen métodos específicos para calcular el área, perímetro y volumen de cada figura geométrica.
- En particular, se muestra cómo calcular el área y el perímetro para el triángulo escaleno, donde se utiliza la fórmula de Herón.
- **Interfaz de Usuario (activity_figure_details.xml):**

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="16dp">

    <!-- TextView para el título de la figura -->
    <TextView
        android:id="@+id/textTituloFigura"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Detalles de la Figura"
        android:textSize="20sp"
        android:textStyle="bold"/>

    <EditText
        android:id="@+id/editTextLado"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/textTituloFigura"
        android:layout_marginTop="16dp"
        android:hint="Ingrese el lado"
        android:inputType="numberDecimal"/>

    <EditText
        android:id="@+id/editTextAltura"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/editTextLado"
        android:layout_marginTop="16dp"
        android:hint="Ingrese la altura"
        android:inputType="numberDecimal"/>

```

- Este archivo define la interfaz de usuario para la actividad. Se compone de varios elementos **EditText** para la entrada de datos, un botón para

calcular y un campo de texto para mostrar los resultados.

```
        android:hint="Ingrese la altura"
        android:inputType="numberDecimal"
        android:visibility="gone"/>

<!-- Nuevo EditText para el tercer lado del Triángulo Escaleno -->
<EditText
    android:id="@+id/editTextThirdSide"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editTextAltura"
    android:layout_marginTop="16dp"
    android:hint="Ingrese el tercer lado"
    android:inputType="numberDecimal"
    android:visibility="gone"/>

<Button
    android:id="@+id/buttonCalcular"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/editTextThirdSide"
    android:layout_marginTop="16dp"
    android:text="Calcular"/>

<TextView
    android:id="@+id/textResultado"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/buttonCalcular"
    android:layout_marginTop="16dp"
```

1. Activity_Main.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      tools:context=".MainActivity">
7
8      <!-- Encabezado -->
9      <TextView
10         android:id="@+id/textViewHeader"
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:text="Operaciones Geométricas"
14         android:textSize="24sp"
15         android:textColor="@color/white"
16         android:gravity="center"
17         android:paddingTop="16dp"
18         android:paddingBottom="16dp"/>
19
20      <!-- ListView Personalizado -->
21      <ListView
22         android:id="@+id/listViewFigures"
23         android:layout_width="match_parent"
24         android:layout_height="match_parent"
25         android:layout_below="@id/textViewHeader"
26         android:divider="@android:color/darker_gray"
27         android:dividerHeight="1dp"
28         android:padding="16dp"/>
29
30  </RelativeLayout>
31

```

Este archivo XML define la interfaz de usuario para la actividad principal de una aplicación Android. Aquí hay una descripción rápida de sus partes:

- **Encabezado (TextView):**
 - Muestra el texto "Operaciones Geométricas" centrado en la parte superior de la pantalla con un tamaño de texto de 24sp y color blanco.
- **ListView Personalizado:**
 - Es un componente de lista que ocupa el resto de la pantalla debajo del encabezado.
 - Muestra una lista de figuras geométricas con un divisor entre ellas.
 - Tiene un relleno de 16dp en todos los lados.
- **Namespace de Herramientas (xmlns:tools):**
 - Define un espacio de nombres para herramientas que se utilizan en el diseño y la vista previa.
- **Declaración XML (<?xml version="1.0" encoding="utf-8"?>):**
 - Indica que el archivo sigue la especificación XML 1.0 y está codificado en UTF-8.

```

1 <?xml version="1.0" encoding="utf-8"?>
2
3 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="wrap_content"
6     android:padding="16dp">
7
8     <ImageView
9         android:id="@+id/imageFigure"
10        android:layout_width="50dp"
11        android:layout_height="50dp"
12        android:layout_marginRight="16dp" />
13
14    <TextView
15        android:id="@+id/textFigureName"
16        android:layout_width="wrap_content"
17        android:layout_height="wrap_content"
18        android:text="Nombre de la Figura"
19        android:textSize="18sp"
20        android:layout_toRightOf="@id/imageFigure" />
21
22    <TextView
23        android:id="@+id/textPerimeter"
24        android:layout_width="wrap_content"
25        android:layout_height="wrap_content"
26        android:text="Perimetro:"
27        android:textSize="14sp"
28        android:layout_below="@id/textFigureName"
29        android:layout_marginTop="8dp"
30        android:layout_toRightOf="@id/imageFigure" />
31

```

```

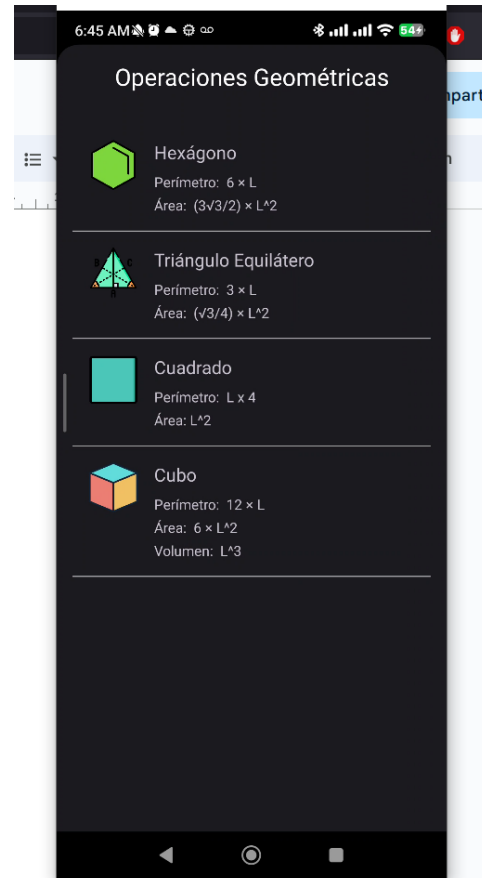
18        android:text="Nombre de la Figura"
19        android:textSize="18sp"
20        android:layout_toRightOf="@id/imageFigure" />
21
22    <TextView
23        android:id="@+id/textPerimeter"
24        android:layout_width="wrap_content"
25        android:layout_height="wrap_content"
26        android:text="Perimetro:"
27        android:textSize="14sp"
28        android:layout_below="@id/textFigureName"
29        android:layout_marginTop="8dp"
30        android:layout_toRightOf="@id/imageFigure" />
31
32    <TextView
33        android:id="@+id/textArea"
34        android:layout_width="wrap_content"
35        android:layout_height="wrap_content"
36        android:text="Area:"
37        android:textSize="14sp"
38        android:layout_below="@id/textPerimeter"
39        android:layout_marginTop="4dp"
40        android:layout_toRightOf="@id/imageFigure" />
41
42    <TextView
43        android:id="@+id/textVolume"
44        android:layout_width="wrap_content"
45        android:layout_height="wrap_content"
46        android:text="Volumen:"
47        android:textSize="14sp"
48        android:layout_below="@id/textArea"
49        android:layout_marginTop="4dp"

```

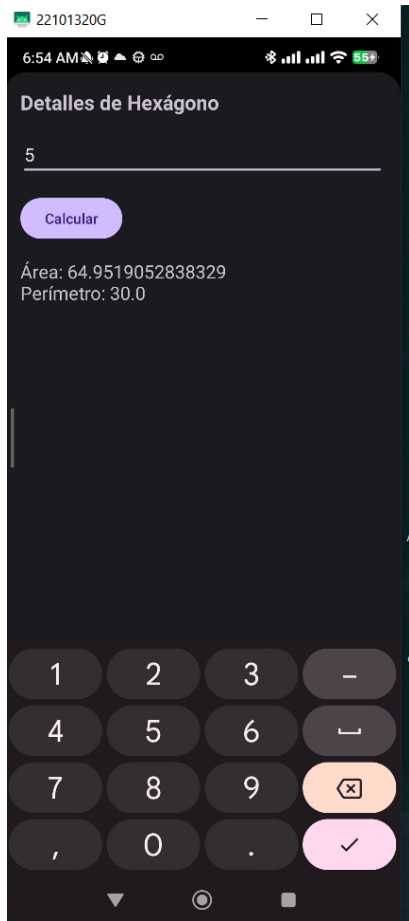
EJECUCIÓN

1. inicio

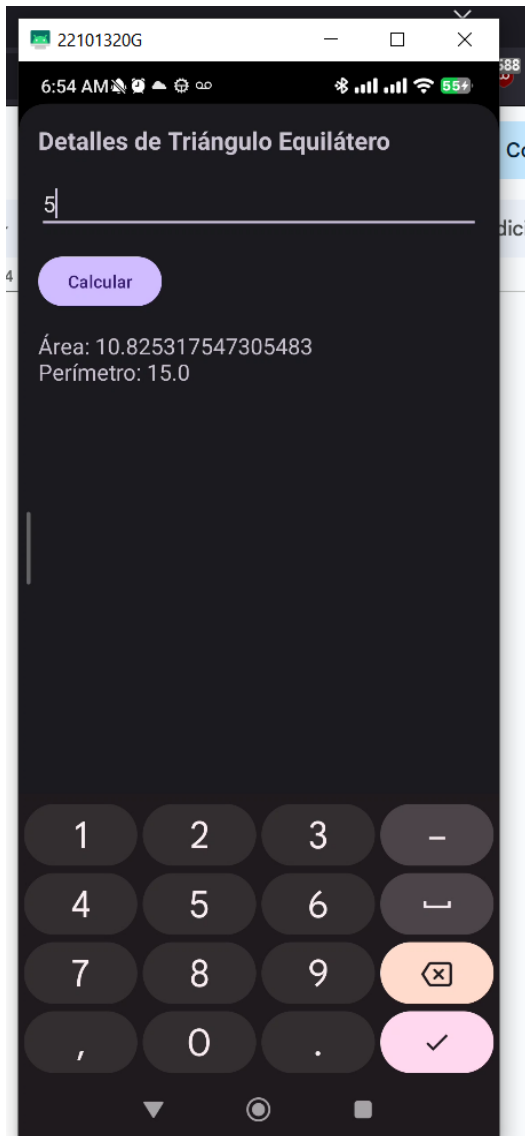
list view de unicin con las figuras.



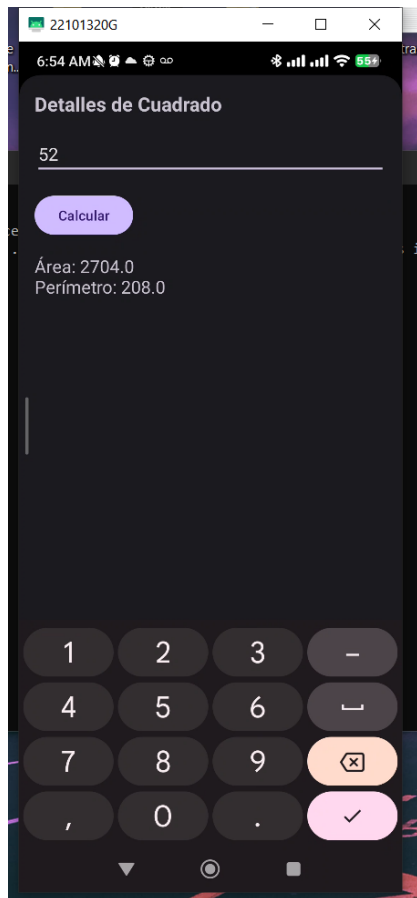
2. Area y perimetro del hexagono:



3. Area y perimetro del triangulo:



4. Area y perimetro de un cuadrado:



5. Area, perimetro y volumen de un cubo:

