

Ficha do Aplicativo

App11 – Aplicativo Cadastro de Produtos com Banco de Dados

Resumo

Este é um novo aplicativo utilizando armazenamento em Banco de Dados.

Objetivos de Aprendizagem

1. Praticar o uso de Intents para navegar entre as telas
2. Compreender o conceito e usar Banco de Dados em aplicativos Android, ampliando os conceitos do aplicativo anterior para a inclusão, alteração e exclusão de registros no SQLite
3. Usar Adapter para popular o ListView a partir do Banco de Dados
4. Usar Adapter para popular um Spinner a partir de um vetor de objetos
5. Técnica para trabalhar com números reais, sem os problemas inerentes aos erros de arredondamento comuns ao usar tipos Double ou Float

Dinâmica do Aplicativo

Este aplicativo conterà e exibirá um cadastro de produtos do tipo vendidos em supermercados. Na atividade principal (MainActivity) estará visível uma lista dos produtos cadastrados. Ao tocar sobre um deles será aberta uma outra atividade (ProdutoActivity) que permitirá a visualização de todos os dados do produto selecionado. Essa atividade tem botões para editar e excluir o produto, bem como um terceiro botão para retornar à atividade principal.

Ao selecionar a edição do produto abre-se uma terceira atividade (IncluiActivity) com várias caixas de texto editáveis (EditText) através das quais pode-se alterar os dados do produto.

Lista de Activities do Aplicativo

Nome	Layout
MainActivity.java	activity_main.xml
ProdutoActivity.java	activity_produto.xml
IncluiActivity.java	activity_inclui.xml

Drawables deste Aplicativo

Nome do arquivo	Dimensões	Tipo	Imagem
imgedita.xml	36 x 36	XML – Vector	
imgexclui.xml	36 x 36	XML – Vector	
imgcancela.xml	36 x 36	XML – Vector	

Resources

strings.xml

```
<resources>
  <string name="app_name">Cadastro de Produtos</string>
  <string name="btn_inclui">Incluir Novo Produto</string>
  <string name="btn_salvar">Salvar</string>
  <string name="btn_cancelar">Cancelar</string>
  <string name="txt_inclui">Inclusão de Produto</string>
  <string name="txt_altera">Alteração de Produto</string>
</resources>
```

colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="purple_200">#FFBB86FC</color>
  <color name="purple_500">#9B25BA</color>
  <color name="purple_700">#FF3700B3</color>
  <color name="teal_200">#FFD746FD</color>
  <color name="teal_700">#FF018786</color>
  <color name="black">#FF000000</color>
  <color name="white">#FFFFFFFF</color>
  <color name="corfundo">#FF202050</color>
  <color name="corlista">#E2FBFF</color>
  <color name="cortexto">#FF0000AE</color>
  <color name="cordestaque">#8503A6</color>
  <color name="cordivisor">#FF778585</color>
</resources>
```

faixa_unidade.xml

Este layout é usado para personalizar a exibição das unidades no spinner contido no módulo IncluiActivity

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <TextView
    android:id="@+id/txtUnidade"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:paddingStart="4dp"
    android:paddingTop="4dp"
    android:textColor="@color/cortexto"
    android:textSize="18sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

  <View
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:background="@color/cordivisor"
    android:layout_marginStart="4dp"
    android:layout_marginTop="4dp"
    app:layout_constraintTop_toBottomOf="@+id/txtUnidade"
    tools:layout_editor_absoluteX="0dp" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Detalhamento das Activities

Classe BDHelper.java

```
public class BDHelper extends SQLiteOpenHelper {
    private static final String NOMEBD = "produtos";
    private static final int VERSAOBD = 1;

    public BDHelper(@Nullable Context context) {
        super(context, NOMEBD, null, VERSAOBD);
    }

    @Override
    public void onCreate(SQLiteDatabase bd) {
        criarEstrutura(bd);
        inserirDados(bd); // se necessário
    }

    @Override
    public void onUpgrade(SQLiteDatabase bd, int oldVersion, int newVersion) {
        // nada a fazer aqui.
    }

    private void criarEstrutura(SQLiteDatabase bd) {
        String sql;
        sql = "CREATE TABLE PRODUTOS (" +
            "_id INTEGER PRIMARY KEY AUTOINCREMENT, " +
            "CODIGO INTEGER, " +
            "NOME TEXT, " +
            "UNID TEXT, " +
            "QTDE INTEGER, " +
            "PCUNIT INTEGER " +
            ")";
        bd.execSQL(sql);
    }

    private void inserirDados(SQLiteDatabase bd) {
        insereProduto(bd,12107,"Vergalhão de aço", "m", 135, 4931);
        insereProduto(bd,12030,"Impermeabilizante", "Balde", 141, 16790);
        insereProduto(bd,12110,"Tinta PVA Branco Acetinado", "Lata", 89, 12631);
        insereProduto(bd,12100,"Tinta Acrílica Gelo Fosco", "Lata", 34, 19837);
        insereProduto(bd,12200,"Cola PVC", "Bisnaga", 328, 1095);
        insereProduto(bd,12410,"Areia Granel", "m3", 603, 11730);
        insereProduto(bd,12420,"Areia Ensacada - sc 5kg", "Saco", 52, 620);
        insereProduto(bd,12300,"Tijolo baiano 6 furos", "Peça", 7550, 109);
        insereProduto(bd,12310,"Tijolo maciço vermelho aparente", "Peça", 3340, 179);
        insereProduto(bd,12350,"Tijolo maciço branco aparente", "Peça", 2208, 236);
        insereProduto(bd,12420,"Pedra Brita 2", "m3", 1129, 7695);
        insereProduto(bd,13150,"Manta asfáltica - rolo c/ 10m2", "m2", 110, 3050);
        insereProduto(bd,11120,"Caibro 5x5 - peça com 3m", "m", 912, 2195);
    }

    private void insereProduto(SQLiteDatabase bd, int codigo, String nome, String unid, int qtde, int pcunit) {
        ContentValues cv = new ContentValues();
        cv.put("CODIGO", codigo);
        cv.put("NOME", nome);
        cv.put("UNID", unid);
        cv.put("QTDE", qtde);
        cv.put("PCUNIT", pcunit);
        bd.insert("PRODUTOS", null, cv);
    }
}
```

Classe Unidade.java

```
public class Unidade {  
  
    private final String unidade;  
  
    public Unidade(String unidade) {  
        this.unidade = unidade;  
    }  
  
    @Override  
    public String toString() {  
        return unidade;  
    }  
  
    public static int indexOf(String unidade) {  
        for (int i = 0; i < qtdeUnidades; i++)  
            if (unidades[i].toString().equals(unidade))  
                return i;  
        return 0;  
    }  
  
    public static int qtdeUnidades = 15;  
    public static Unidade[] unidades = {  
        new Unidade(""),  
        new Unidade("Balde"),  
        new Unidade("Bisnaga"),  
        new Unidade("Kg"),  
        new Unidade("Lata"),  
        new Unidade("Litro"),  
        new Unidade("m"),  
        new Unidade("m2"),  
        new Unidade("m3"),  
        new Unidade("Pacote"),  
        new Unidade("Peça"),  
        new Unidade("Rolo"),  
        new Unidade("Saco"),  
        new Unidade("Tubo"),  
        new Unidade("Vidro")  
    };  
}
```

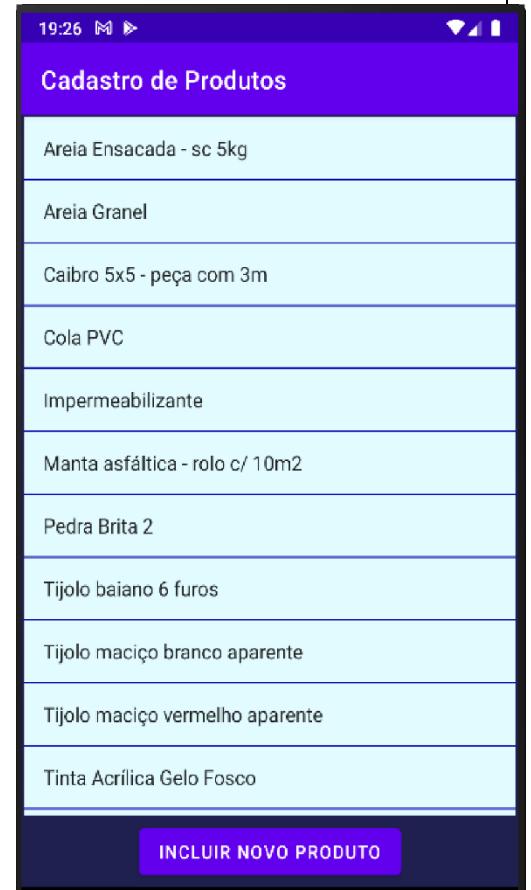
Layout: activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/corfundo"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btnInclui"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:onClick="btnIncluiOnClick"
        android:text="@string/btn_inclui"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/lvProdutos" />

    <ListView
        android:id="@+id/lvProdutos"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="2dp"
        android:layout_marginTop="2dp"
        android:layout_marginEnd="2dp"
        android:layout_marginBottom="56dp"
        android:background="@color/corlista"
        android:divider="@color/cortexto"
        android:dividerHeight="1dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



Código: MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    private ListView lvProdutos;

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

        lvProdutos = findViewById(R.id.lvProdutos);
        setLVProdutosOnClick();
    }

    @Override
    protected void onStart() {
        super.onStart();
        setLvProdutosAdapter();
    }

    private void setLvProdutosAdapter() {
        String sql = "select _id, NOME from PRODUTOS order by NOME";
        try {
            DBHelper dbHelper = new DBHelper(this);
            SQLiteDatabase bd = dbHelper.getReadableDatabase();
            Cursor cursor = bd.rawQuery(sql, null);
            SimpleCursorAdapter adaptador = new SimpleCursorAdapter(
                this,
                android.R.layout.simple_list_item_1,
                cursor,
                new String[] {"NOME"},
                new int[] {android.R.id.text1},
                0
            );
        }
    }
}
```

```

    );
    lvProdutos.setAdapter(adaptador);
}
catch (SQLException e) {
    Toast.makeText(this, "Falha no acesso aos dados - " + e.getMessage(), Toast.LENGTH_SHORT).show();
}
}

private void setLVProdutosOnItemClickListener() {
    lvProdutos.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> adapterView, View view, int position, long id) {
            Intent intent = new Intent(MainActivity.this, ProdutoActivity.class);
            intent.putExtra(ProdutoActivity.EXTRA_IDPRODUTO, id);
            startActivity(intent);
        }
    });
}

public void btnIncluiOnClick(View view) {
    Intent intent = new Intent(MainActivity.this, IncluiActivity.class);
    intent.putExtra(IncluiActivity.EXTRA_OP, IncluiActivity.OP_INCLUI);
    startActivity(intent);
}
}
}

```

Layout: activity_produto.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/corlista"
    tools:context=".ProdutoActivity">

    <TextView
        android:id="@+id/txtCodigo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="40dp"
        android:layout_marginEnd="8dp"
        android:text="TextView"
        android:textColor="@color/cortexto"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.501"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/txtNome"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="24dp"
        android:layout_marginEnd="8dp"
        android:text="TextView"
        android:textColor="@color/cortexto"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.501"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtCodigo" />

    <TextView
        android:id="@+id/txtUnid"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="76dp"
        android:layout_marginEnd="8dp"
        android:text="TextView"
        android:textColor="@color/cortexto"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.501"

```



```

app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txtCodigo" />

<TextView
    android:id="@+id/txtQtde"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="128dp"
    android:layout_marginEnd="8dp"
    android:text="TextView"
    android:textColor="@color/cortexto"
    android:textSize="24sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.501"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/txtCodigo" />

<TextView
    android:id="@+id/txtPcUnit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="184dp"
    android:layout_marginEnd="8dp"
    android:text="TextView"
    android:textColor="@color/cortexto"
    android:textSize="24sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.501"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/txtCodigo" />

<LinearLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:gravity="center_horizontal"
    android:orientation="horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent">

    <ImageButton
        android:id="@+id/btnAlterar"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_marginStart="2dp"
        android:layout_marginEnd="2dp"
        android:layout_weight="1"
        android:backgroundTint="@color/corlista"
        android:onClick="btnAlterarOnClick"
        app:srcCompat="@drawable/imgedita"
        app:tint="@color/purple_500"
        tools:layout_editor_absoluteX="46dp"
        tools:layout_editor_absoluteY="401dp" />

    <ImageButton
        android:id="@+id/btnExcluir"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_marginStart="2dp"
        android:layout_marginEnd="2dp"
        android:layout_weight="1"
        android:backgroundTint="@color/corlista"
        android:onClick="btnExcluirOnClick"
        app:srcCompat="@drawable/imgexclui"
        app:tint="@color/purple_500"
        tools:layout_editor_absoluteX="130dp"
        tools:layout_editor_absoluteY="401dp" />

    <ImageButton
        android:id="@+id/btnVoltar"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_marginStart="2dp"
        android:layout_marginEnd="2dp"

```

```

        android:layout_weight="1"
        android:backgroundTint="@color/corlista"
        android:onClick="btnVoltarOnClick"
        app:srcCompat="@drawable/imgcancela"
        app:tint="@color/purple_500"
        tools:layout_editor_absoluteX="220dp"
        tools:layout_editor_absoluteY="401dp" />

```

```
</LinearLayout>
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

Código: ProdutoActivity.java

```

public class ProdutoActivity extends AppCompatActivity {

    public static final String EXTRA_IDPRODUTO = "idproduto";

    private long idProduto;

    private BDHelper bdHelper;
    private SQLiteDatabase bd;

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

        carregaDados();
    }

    private void carregaDados() {
        TextView txt;
        String s;
        int valor;

        idProduto = getIntent().getExtras().getLong(EXTRA_IDPRODUTO);

        try {
            bdHelper = new BDHelper(this);
            bd = bdHelper.getReadableDatabase();
            String sql = "select * from PRODUTOS where _id = ?";
            Cursor cursor = bd.rawQuery(sql, new String[] {Long.toString(idProduto)} );
            if (cursor.moveToFirst()) {
                txt = findViewById(R.id.txtCodigo);
                txt.setText(cursor.getString(cursor.getColumnIndexOrThrow("CODIGO")));
                txt = findViewById(R.id.txtNome);
                txt.setText(cursor.getString(cursor.getColumnIndexOrThrow("NOME")));
                txt = findViewById(R.id.txtUnid);
                String unid = cursor.getString(cursor.getColumnIndexOrThrow("UNID"));
                txt.setText(unid);

                valor = cursor.getInt(cursor.getColumnIndexOrThrow("QTDE"));
                s = String.format(Locale.getDefault(), "Qtde. disponível: %,d", valor);
                txt = findViewById(R.id.txtQtde);
                txt.setText(s);

                valor = cursor.getInt(cursor.getColumnIndexOrThrow("PCUNIT"));
                txt = findViewById(R.id.txtPcUnit);
                txt.setText(String.format(Locale.getDefault(), "Valor: R$ %, .2f", valor/100.0));
            }
            cursor.close();
        }
        catch (SQLException e){
            Toast.makeText(this, "Falha no acesso aos dados - " + e.getMessage(), Toast.LENGTH_LONG).show();
        }
    }

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

    public void btnAlterarOnClick(View view) {
        Intent intent = new Intent(ProdutoActivity.this, IncluiActivity.class);
        intent.putExtra(IncluiActivity.EXTRA_OP, IncluiActivity.OP_ALTERA);
        intent.putExtra(ProdutoActivity.EXTRA_IDPRODUTO, idProduto);
        startActivity(intent);
        finish();
    }
}

```



```
public void btnExcluirOnClick(View view) {
    AlertDialog alertDialog = new AlertDialog.Builder(ProdutoActivity.this).create();
    alertDialog.setMessage("Tem certeza de que deseja excluir o produto selecionado?");
    alertDialog.setButton(Dialog.BUTTON_NEGATIVE, "Cancelar", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {

        }
    });
    alertDialog.setButton(Dialog.BUTTON_POSITIVE, "Confirmar", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int which) {
            bd.delete("PRODUTOS", "_id = ?", new String[] {Long.toString(idProduto)});
            finish();
        }
    });
    alertDialog.show();
}
}
```

Layout: activity_inclui.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/corlista"
    tools:context=".IncluiActivity">

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:orientation="vertical"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <TextView
            android:id="@+id/txtTitulo"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Inclusão de Produto"
            android:textColor="@color/cordestaque"
            android:textSize="24sp" />

        <View
            android:id="@+id/divider1"
            android:layout_width="match_parent"
            android:layout_height="4dp"
            android:background="?android:attr/listDivider" />

        <TextView
            android:id="@+id/textView1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Código do Produto"
            android:textColor="@color/cordestaque" />

        <EditText
            android:id="@+id/edtCodigo"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="20dp"
            android:ems="10"
            android:inputType="number|numberDecimal"
            android:text=""
            android:textColor="@color/cortexto" />

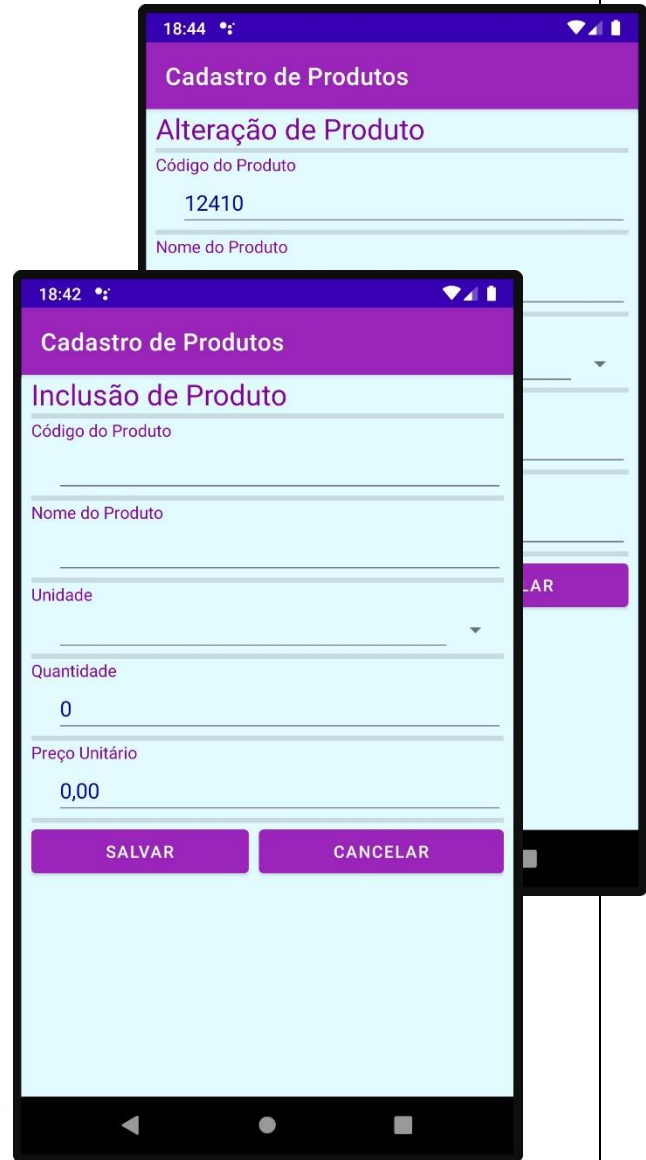
        <View
            android:id="@+id/divider2"
            android:layout_width="match_parent"
            android:layout_height="4dp"
            android:background="?android:attr/listDivider" />

        <TextView
            android:id="@+id/textView2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Nome do Produto"
            android:textColor="@color/cordestaque" />

        <EditText
            android:id="@+id/edtNome"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="20dp"
            android:ems="10"
            android:inputType="textPersonName"
            android:text=""
            android:textColor="@color/cortexto" />

        <View
            android:id="@+id/divider3"
            android:layout_width="match_parent"

```



```

        android:layout_height="4dp"
        android:background="?android:attr/listDivider" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Unidade"
    android:textColor="@color/cordestaque" />

<Spinner
    android:id="@+id/spnUnidade"
    style="@style/Widget.AppCompat.Spinner.DropDown"
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:layout_marginLeft="20dp"
    android:spinnerMode="dropdown"
    android:textColor="@color/cortexto" />

<View
    android:id="@+id/divider4"
    android:layout_width="match_parent"
    android:layout_height="4dp"
    android:background="?android:attr/listDivider" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Quantidade"
    android:textColor="@color/cordestaque" />

<EditText
    android:id="@+id/edtQtde"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:ems="10"
    android:inputType="number|numberDecimal"
    android:text=""
    android:textColor="@color/cortexto" />

<View
    android:id="@+id/divider5"
    android:layout_width="match_parent"
    android:layout_height="4dp"
    android:background="?android:attr/listDivider" />

<TextView
    android:id="@+id/textView5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Preço Unitário"
    android:textColor="@color/cordestaque" />

<EditText
    android:id="@+id/edtPcUnit"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:ems="10"
    android:inputType="number|numberDecimal"
    android:text=""
    android:textColor="@color/cortexto" />

<View
    android:id="@+id/divider6"
    android:layout_width="match_parent"
    android:layout_height="4dp"
    android:background="?android:attr/listDivider" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <Button
        android:id="@+id/btnSalvar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

```

```

        android:layout_marginEnd="4dp"
        android:layout_weight="1"
        android:onClick="btnSalvarOnClick"
        android:text="@string/btn_salvar" />

        <Button
            android:id="@+id/btnCancelar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="4dp"
            android:layout_weight="1"
            android:onClick="btnCancelarOnClick"
            android:text="@string/btn_cancelar" />
    </LinearLayout>

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

Código: IncluiActivity.java

```

public class IncluiActivity extends AppCompatActivity {

    public static final String EXTRA_OP = "operacao";
    public static final int OP_INCLUI = 1;
    public static final int OP_ALTERA = 2;

    private int operacao;
    private long idProduto;
    private int pcUnit;

    private EditText edtCodigo;
    private EditText edtNome;
    private Spinner spnUnidade;
    private EditText edtQtde;
    private EditText edtPcUnit;

    public void setOperacao(int operacao) {
        this.operacao = operacao;

        TextView txtTitulo = findViewById(R.id.txtTitulo);
        if (operacao == OP_INCLUI) {
            txtTitulo.setText(R.string.txt_inclui);
            setPcUnit(0);
        }
        else {
            txtTitulo.setText(R.string.txt_altera);
            carregaDados();
        }
    }

    public void setPcUnit(int pcUnit) {
        this.pcUnit = pcUnit;
        edtPcUnit.setText(String.format(Locale.getDefault(), "%.2f", pcUnit / 100.0));
    }

    private void setEdtPcUnitOnKeyListener() {
        edtPcUnit.setInputType(InputType.TYPE_CLASS_NUMBER | InputType.TYPE_NUMBER_FLAG_DECIMAL);
        edtPcUnit.setKeyListener(DigitsKeyListener.getInstance("0123456789"));
        edtPcUnit.setOnKeyListener(new View.OnKeyListener() {
            @Override
            public boolean onKey(View view, int qualTecla, KeyEvent evento) {
                //Toast.makeText(getApplicationContext(), "pc tecla = " + qualTecla, Toast.LENGTH_SHORT).show();
                if (evento.getAction() == KeyEvent.ACTION_DOWN) {
                    if (KeyEvent.KEYCODE_0 <= qualTecla && qualTecla <= KeyEvent.KEYCODE_9) // Teclas numéricas
                        setPcUnit(pcUnit * 10 + qualTecla - 7);
                    if (qualTecla == KeyEvent.KEYCODE_DEL) // BackSpace para apagar
                        setPcUnit(pcUnit / 10);
                    edtPcUnit.setSelection(edtPcUnit.getText().length());
                    return true;
                }
                return false;
            }
        });
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_inclui);

edtCodigo = findViewById(R.id.edtCodigo);
edtNome = findViewById(R.id.edtNome);
spnUnidade = findViewById(R.id.spnUnidade);
edtQtde = findViewById(R.id.edtQtde);
edtPcUnit = findViewById(R.id.edtPcUnit);

setSpnUnidadeAdapter();
setEdtPcUnitOnKeyListener();

setOperacao(getIntent().getExtras().getInt(EXTRA_OP));
}

private void setSpnUnidadeAdapter() {
    ArrayAdapter<Unidade> adaptador = new ArrayAdapter<>(
        this,
        R.layout.faixa_unidade,
        R.id.txtUnidade,
        Unidade.unidades
    );
    spnUnidade.setAdapter(adaptador);
}

private void carregaDados() {
    idProduto = getIntent().getExtras().getLong(ProdutoActivity.EXTRA_IDPRODUTO);
    String sql = "select _id, CODIGO, NOME, UNID, QTDE, PCUNIT from PRODUTOS where _id = ?";
    try {
        BDHelper bdHelper = new BDHelper(this);
        SQLiteDatabase bd = bdHelper.getReadableDatabase();
        Cursor cursor = bd.rawQuery(sql, new String[] {Long.toString(idProduto)});
        if (cursor.moveToFirst()) {
            Toast.makeText(this, "Cursor ok", Toast.LENGTH_LONG).show();
            edtCodigo.setText(cursor.getString(cursor.getColumnIndexOrThrow("CODIGO")));
            edtNome.setText(cursor.getString(cursor.getColumnIndexOrThrow("NOME")));
            String unid = cursor.getString(cursor.getColumnIndexOrThrow("UNID"));
            spnUnidade.setSelection(Unidade.indexOf(unid));
            edtQtde.setText(cursor.getString(cursor.getColumnIndexOrThrow("QTDE")));
            setPcUnit(cursor.getInt(cursor.getColumnIndexOrThrow("PCUNIT")));
        }
        cursor.close();
    }
    catch (SQLException e){
        Toast.makeText(this, "Falha no banco de dados - " + e.getMessage(), Toast.LENGTH_LONG).show();
    }
}

private boolean jaExiste(String codigo) {
    BDHelper bdHelper = new BDHelper(this);
    SQLiteDatabase bd = bdHelper.getReadableDatabase();
    String sql = "select CODIGO from PRODUTOS where CODIGO = ?";
    Cursor cursor = bd.rawQuery(sql, new String[] {codigo});
    if (cursor.moveToFirst()) {
        cursor.close();
        return true;
    }
    else {
        cursor.close();
        return false;
    }
}

private void dadosValidos() throws Exception {
    String s;
    s = edtCodigo.getText().toString();
    if (s.equals(""))
        throw new Exception("O Código deve ser preenchido");
    else if (operacao == OP_INCLUI && jaExiste(s))
        throw new Exception("Código já cadastrado");
    s = edtNome.getText().toString();
    if (s.equals(""))
        throw new Exception("O Nome do Produto deve ser preenchido");
    s = spnUnidade.getSelectedItem().toString();
    if (s.equals(""))
        throw new Exception("A Unidade deve ser preenchida");
    s = edtQtde.getText().toString();
    if (s.equals(""))
        throw new Exception("A Quantidade deve ser preenchida");
}

```

```

}

public void btnSalvarOnClick(View view) {
    try {
        dadosValidos();
        if (operacao == OP_INCLUI) {
            insereProduto();
        }
        else if (operacao == OP_ALTERA) {
            alteraProduto();
            finish();
        }
        finish();
    } catch (SQLiteException e) {
        Toast.makeText(this, "Inclusão falhou", Toast.LENGTH_LONG).show();
    }
    catch (Exception e) {
        Toast.makeText(this, e.getMessage(), Toast.LENGTH_LONG).show();
    }
}

private ContentValues carregaCV() {
    ContentValues cv = new ContentValues();
    cv.put("CODIGO", edtCodigo.getText().toString());
    cv.put("NOME", edtNome.getText().toString());
    cv.put("UNID", spnUnidade.getSelectedItem().toString());
    cv.put("QTDE", Integer.parseInt(edtQtde.getText().toString()));
    cv.put("PCUNIT", pcUnit);
    return cv;
}

private void insereProduto() {
    BDHelper bdHelper = new BDHelper(this);
    SQLiteDatabase bd = bdHelper.getWritableDatabase();
    ContentValues cvInsere = carregaCV();
    bd.insert("PRODUTOS", null, cvInsere);
}

private void alteraProduto() {
    BDHelper bdHelper = new BDHelper(this);
    SQLiteDatabase bd = bdHelper.getWritableDatabase();
    ContentValues cvAlterar = carregaCV();
    bd.update("PRODUTOS", cvAlterar, "_id = ?", new String[] {Long.toString(idProduto)});
}

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

```

Plano de Desenvolvimento deste Aplicativo

1. Crie um projeto novo com Empty Activity
2. Baixe e descompacte o arquivo de resources que está na minha página e copie o conteúdo na pasta *drawable*
3. Edite os arquivos de resources strings.xml e colors.xml acrescentando os elementos usados
4. Crie a classe BDHelper estendendo a classe abstrata SQLiteOpenHelper
5. Crie a classe Unidade
6. Crie o layout da atividade principal: activity_main.xml
7. Implemente o código Java da atividade principal: ActivityMain.java. Nesta classe haverá a necessidade de:
 - Usar um SimpleCursorAdapter para popular a ListView lvProdutos.
 - Criar o OnItemClickListener do ListView
 - Implementar o código do botão de inclusão de novos produtos
8. Crie uma nova atividade baseada no template Empty Activity → esta será ProdutoActivity. Seu propósito será exibir todos os campos de um produto usando componentes TextView
9. Monte o layout: activity_produto.xml
10. Implemente o código Java da atividade ProdutoActivity.java. Os destaques nessa classe são:
 - Implementação dos três botões que vão permitir as tarefas de alteração e exclusão de produtos, bem como o fechamento da Activity, retornando à MainActivity
 - Para exclusão deve ser executado um AlertDialog para confirmar a operação
11. Crie uma nova atividade baseada no template Empty Activity → esta será IncluiActivity
12. Monte o layout: activity_inclui.xml
13. Implemente o código Java da atividade IncluiActivity.java. Há bastante coisa a ser feita nesta classe:
 - Crie todas as suas propriedades privadas
 - Instancie todas os EditText e o Spinner usando findViewById()
 - Implemente os setters para *operacao*, *unid*, *qtde* e *pcUnit* que serão os responsáveis pela exibição dessas propriedades nos respectivos componentes
 - Implemente o ArrayAdapter do Spinner, que será o responsável pela exibição da unidade de venda do produto
 - Implemente o OnItemSelectedListener do Spinner. Nele, quando a unidade for trocada o setter setUnid() precisa ser acionado para substituir a unidade do produto, bem como ajustar a quantidade (a unidade "Kg" tem casas decimais e as outras não)
 - Implemente o OnKeyListener do componente edtQtde. Nele será ajustada a propriedade *qtde*. A necessidade disso são os problemas inerentes ao tratamento de números reais explicados em aula
 - Implemente o OnKeyListener do componente edtPcUni. Nele será ajustada a propriedade *pcUnit*. Aqui vale a mesma observação acima sobre os números reais
 - Implemente o método carregaDados() responsável por acessar o banco de dados e carregar os componentes visuais com os valores adequados
 - Implemente os métodos jaExiste() responsável por verificar a existência prévia de um código de produto e dadosValidos() que faz a validação dos dados digitados nos campos editáveis. As validações são: código não pode estar vazio; na inclusão o código não pode ser repetido, o campo nome não pode estar vazio, o spinner de unidade não pode estar vazio. Se alguma consistência for encontrada uma exceção será levantada
 - Implemente o código do botão salvar para gravar os dados digitados no banco de dados, considerando a dupla função que a activity possui: para a inclusão de novo produto deve ser usado insert e para a alteração de produto já cadastrado deve ser usado update