

DESARROLLO DE APLICACIONES ANDROID 1

DAM – M08: PROGRAMACIÓN MULTIMEDIA
Y DISPOSITIVOS MÓVILES

UF1: DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES

RAQUEL LLORENTE ARMIJO

2021/2022

CONTENIDO

CÓDIGO	3
<i>Clases principales.....</i>	<i>3</i>
<i>activity_main.xml</i>	<i>3</i>
<i>MainActivity.java</i>	<i>6</i>
<i>Clases Spinner Personalizado.....</i>	<i>10</i>
<i>ic_baseline_arrow_drop_down_24.xml</i>	<i>10</i>
<i>spinner_shape.xml.....</i>	<i>10</i>
<i>spinner_item_layout.xml.....</i>	<i>10</i>
<i>CustomAdapter.java</i>	<i>11</i>
<i>Clases Personalización Aplicación.....</i>	<i>13</i>
<i>edittext_importe_design.xml</i>	<i>13</i>
<i>colors.xml</i>	<i>13</i>
<i>strings.xml.....</i>	<i>13</i>
<i>values\themes.xml.....</i>	<i>14</i>
<i>night\themes.xml.....</i>	<i>14</i>
CAPTURAS DE PANTALLA.....	15
<i>Funciones básicas</i>	<i>15</i>
<i>Errores y avisos.....</i>	<i>18</i>
<i>Misma divisa</i>	<i>18</i>
<i>Importe vacío</i>	<i>19</i>
<i>Valor importe inválido.....</i>	<i>20</i>
<i>Formato importe con decimales incorrecto</i>	<i>21</i>
<i>Importe negativo</i>	<i>22</i>
<i>Light/dark theme</i>	<i>23</i>

CÓDIGO

CLASES PRINCIPALES

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Define el formato de cada control -->
<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"
    tools:context=".MainActivity">

    <!-- TextView "Convertir de" -->
    <TextView
        android:id="@+id/tv_actual_rla"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="30dp"
        android:layout_marginTop="24dp"
        android:layout_marginBottom="8dp"
        android:text="@string/de"
        android:textSize="20sp"
        app:layout_constraintBottom_toTopOf="@+id/sp_actual_rla"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <!-- Spinner divisa actual -->
    <Spinner
        android:id="@+id/sp_actual_rla"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="24dp"
        android:layout_marginEnd="24dp"
        android:layout_marginBottom="48dp"
        android:background="@drawable/spinner_shape"
        android:minHeight="48dp"
        android:spinnerMode="dialog"
        app:layout_constraintBottom_toTopOf="@+id/sp_nueva_rla"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tv_actual_rla" />

    <!-- TextView "a" -->
    <TextView
        android:id="@+id/tv_nueva_rla"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="30dp"
        android:layout_marginBottom="8dp"
        android:text="@string/a"
        android:textSize="20sp"
        app:layout_constraintBottom_toTopOf="@+id/sp_nueva_rla"
```

```

        app:layout_constraintStart_toStartOf="parent" />

<!-- Spinner divisa nueva -->
<Spinner
    android:id="@+id/sp_nueva_rla"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginEnd="24dp"
    android:layout_marginBottom="82dp"
    android:background="@drawable/spinner_shape"
    android:minHeight="48dp"
    android:spinnerMode="dialog"
    app:layout_constraintBottom_toTopOf="@+id/et_importe_rla"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/sp_actual_rla" />

<!-- TextView "Importe" -->
<TextView
    android:id="@+id/tv_importe_rla"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="30dp"
    android:layout_marginBottom="8dp"
    android:text="@string/importe"
    android:textSize="20sp"
    app:layout_constraintBottom_toTopOf="@+id/et_importe_rla"
    app:layout_constraintStart_toStartOf="parent" />

<!-- EditText "Importe" -->
<EditText
    android:id="@+id/et_importe_rla"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginEnd="24dp"
    android:layout_marginBottom="24dp"
    android:background="@drawable/edittext_importe_design"
    android:inputType="textPersonName"
    android:maxLength="20"
    android:minHeight="48dp"
    android:padding="15dp"
    android:textSize="20sp"
    app:layout_constraintBottom_toTopOf="@+id/bt_convertir_rla"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/sp_nueva_rla" />

<!-- Button "Convertir" -->
<Button
    android:id="@+id/bt_convertir_rla"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="68dp"
    android:paddingLeft="30dp"
    android:paddingTop="10dp"
    android:paddingRight="30dp"
    android:paddingBottom="10dp"

```

```

        android:text="@string/convertir"
        android:textAllCaps="true"
        android:textSize="20sp"
        app:layout_constraintBottom_toTopOf="@+id/tv_result2_rla"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/et_importe_rla" />

<!-- TextView texto "Resultado:" -->
<TextView
    android:id="@+id/tv_result1_rla"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="40dp"
    android:layout_marginTop="16dp"
    android:layout_marginBottom="4dp"
    android:text="@string/resultado"
    android:textSize="14sp"
    android:visibility="invisible"
    app:layout_constraintBottom_toTopOf="@+id/tv_result2_rla"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/bt_convertir_rla"
    app:layout_constraintVertical_bias="0.73" />

<!-- TextView con el resultado obtenido de hacer la conversión -->
<TextView
    android:id="@+id/tv_result2_rla"
    android:layout_width="331dp"
    android:layout_height="62dp"
    android:layout_marginStart="42dp"
    android:layout_marginTop="56dp"
    android:layout_marginEnd="42dp"
    android:autoSizeTextType="uniform"
    android:padding="0dp"
    android:textAlignment="center"
    android:textStyle="bold"
    android:visibility="invisible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/bt_convertir_rla" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

MainActivity.java

```
/* Clase que contiene la funcionalidad de la aplicación */
package com.example.dam_m08_act1_raquelllorete;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;

import java.math.BigDecimal;
import java.math.RoundingMode;

public class MainActivity extends AppCompatActivity {
    // Declarar variables
    private EditText et_importe_rla;
    private TextView tv_result1_rla, tv_result2_rla, tv_importe_rla;
    private Button bt_convertir_rla;
    private Spinner sp_actual_rla, sp_nueva_rla;
    private CustomAdapter adapter_rla;

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

        // Inicializa variables
        et_importe_rla = (EditText) findViewById(R.id.et_importe_rla);
        tv_result1_rla = (TextView) findViewById(R.id.tv_result1_rla);
        tv_result2_rla = (TextView) findViewById(R.id.tv_result2_rla);
        tv_importe_rla = (TextView) findViewById(R.id.tv_importe_rla);
        bt_convertir_rla = (Button)
        findViewById(R.id.bt_convertir_rla);
        sp_actual_rla = (Spinner) findViewById(R.id.sp_actual_rla);
        sp_nueva_rla = (Spinner) findViewById(R.id.sp_nueva_rla);

        // Array formado por las 11 divisas disponibles
        String[] divisas_rla = {"EUR - Euro", "USD - Dólar
estadounidense", "GBP - Libra esterlina", "INR - Rupia india", "CHF -
Franco Suizo",
        "JPY - Yen japonés", "MXN - Peso mexicano", "SEK -
Corona sueca", "RUB - Rublo ruso", "KRW - Won surcoreano", "THB - Bath
tailandés"};

        // Array formado por las imágenes de las banderas
        int[] banderas_rla = {R.drawable.eur, R.drawable.usa,
        R.drawable.gbp, R.drawable.inr, R.drawable.chf,
        R.drawable.jpy, R.drawable.mxn, R.drawable.sek,
        R.drawable.rub, R.drawable.krw, R.drawable.thb};

        // Enlaza el formato del CustomAdapter con cada spinner
        adapter_rla = new CustomAdapter(this, divisas_rla,
        banderas_rla);
        sp_actual_rla.setAdapter(adapter_rla);
    }
}
```

```

        sp_nueva_rla.setAdapter(adapter_rla);

        // Selecciona el valor por defecto del segundo spinner para
        // evitar que ambos spinners comiencen con el mismo valor
        sp_nueva_rla.setSelection(1);

        // Array con las tasas de cambio de cada divisa en euros
        Double[] ratio_rla = {1.0, 0.86, 1.18, 0.011, 0.93, 0.0077,
        0.042, 0.099, 0.012, 0.00072, 0.026};

        // Evento que se activará al seleccionar un elemento del
        // primer spinner
        sp_actual_rla.setOnItemClickListener(new
        AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView,
            View view, int i, long l) {
                // Si se selecciona la misma opción en los dos
                // spinners:
                if(sp_actual_rla.getSelectedItemPosition() ==
                sp_nueva_rla.getSelectedItemPosition()) {
                    // Se deshabilita el botón y el EditText para
                    // escribir el importe
                    bt_convertir_rla.setEnabled(false);
                    et_importe_rla.setEnabled(false);

                    // Aparece un mensaje de alerta avisando de que no
                    // se puede seleccionar la misma opción en ambos spinners
                    AlertDialog.Builder alerta_rla = new
                    AlertDialog.Builder(MainActivity.this);
                    alerta_rla.setMessage("Por favor, seleccione una
                    opción distinta").show();
                }else{
                    // Si no, se habilita el botón y el EditText para
                    // escribir el importe
                    bt_convertir_rla.setEnabled(true);
                    et_importe_rla.setEnabled(true);
                }
            }
            @Override
            public void onNothingSelected(AdapterView<?> adapterView)
        {
        }
        });

        // Evento que se activará al seleccionar un elemento del
        // segundo spinner
        sp_nueva_rla.setOnItemClickListener(new
        AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView,
            View view, int i, long l) {
                // Si se selecciona la misma opción en los dos
                // spinners:
                if(sp_actual_rla.getSelectedItemPosition() ==
                sp_nueva_rla.getSelectedItemPosition()) {
                    // Se deshabilita el botón y el EditText para
                    // escribir el importe
                    bt_convertir_rla.setEnabled(false);
                    et_importe_rla.setEnabled(false);

```

```

        // Aparece un mensaje de alerta avisando de que no
        se puede seleccionar la misma opción en ambos spinners
        AlertDialog.Builder alerta_rla = new
AlertDialog.Builder(MainActivity.this);
        alerta_rla.setMessage("Por favor, seleccione una
opción distinta").show();
    }else{
        // Si no, se habilita el botón y el EditText para
        escribir el importe
        bt_convertir_rla.setEnabled(true);
        et_importe_rla.setEnabled(true);

    }
}
@Override
public void onNothingSelected(AdapterView<?> adapterView)
{
}
});

// Evento que se activará al hacer clic sobre el botón
bt_convertir_rla.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        // Bloque try/catch para capturar excepciones
        try{
            // Obtiene la posición del elemento seleccionado
            en cada spinner
            int pos_sp_actual_rla =
sp_actual_rla.getSelectedItemPosition();
            int pos_sp_nueva_rla =
sp_nueva_rla.getSelectedItemPosition();

            // Obtiene el valor escrito por el usuario en el
            EditText en formato String
            String tx_importe_rla =
et_importe_rla.getText().toString();
            // Parsea el String obtenido a Double
            Double num_importe_rla =
Double.parseDouble(tx_importe_rla);

            // Condición para evitar importes negativos
            if(num_importe_rla>0) {
                // Calcula la conversión: Primero,
                independientemente de la divisa actual elegida, convierte el
                // importe obtenido en euros. A continuación,
                divide la cantidad entre el cambio en euros de la
                // nueva divisa. De esta forma, teniendo
                solamente las conversiones de todas las divisas a euros,
                // podemos calcular el cambio de cualquier
                divisa

                Double result_rla = num_importe_rla *
ratio_rla[pos_sp_actual_rla] / ratio_rla[pos_sp_nueva_rla];
                // Redondea el resultado double a dos
                decimales

                BigDecimal bigDecimal_rla = new
BigDecimal(result_rla).setScale(2, RoundingMode.HALF_UP);
                result_rla = bigDecimal_rla.doubleValue();

```



```
        // Atribuye el valor obtenido a la propiedad
        "Text" del TextView
        tv_result2_rla.setText(result_rla.toString() +
" " + divisas_rla[pos_sp_nueva_rla].toString().substring(0,3));
        // Muestra los dos TextView con la palabra
        "Resultado:" y el valor del resultado
        tv_result1_rla.setVisibility(View.VISIBLE);
        tv_result2_rla.setVisibility(View.VISIBLE);

    } else{
        // Mensaje de error que aparecerá en caso de
        escribir un importe negativo
        et_importe_rla.setError("ERROR: No admite
valores negativos");
    }

    } catch (Exception ex1){
        // Mensaje de error que aparecerá en caso de
        producirse alguna excepción
        et_importe_rla.setError("ERROR: Valor inválido");
    }

    }
    });
}
}
```

CLASES SPINNER PERSONALIZADO

ic_baseline_arrow_drop_down_24.xml

```
<!-- Personalización del tamaño y color del vector flecha determinado de Android Studio -->
<vector android:height="24dp"
    android:viewportHeight="24" android:viewportWidth="24"
    android:width="48dp"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <path android:fillColor="@color/blue_600"
        android:pathData="M7,10l5,5 5,-5z"/>
</vector>
```

spinner_shape.xml

```
<?xml version="1.0" encoding="utf-8"?>
<i>
```

```

        android:layout_height="wrap_content">

        <ImageView
            android:id="@+id/iv_bandera_rla"
            android:layout_width="0dp"
            android:layout_height="60dp"
            android:layout_weight="1"
            android:padding="3dp"
            android:src="@drawable/usa" />

        <TextView
            android:id="@+id/tv_divisa_rla"
            android:layout_width="0dp"
            android:layout_height="60dp"
            android:layout_weight="3"
            android:gravity="left|center"
            android:text="USD - Dólar estadounidense"
            android:textSize="16sp" />

    </LinearLayout>

```

CustomAdapter.java

```

/* Clase que define el formato del Adapter del Spinner personalizado */
package com.example.dam_m08_act1_raquelllorente;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

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

public class CustomAdapter extends ArrayAdapter<String> {
    // Declara las variables que formarán la estructura del spinner personalizado
    Context contexto_rla;
    String[] texto_rla;
    int[] imagenes_rla;

    // Contructor del adapter personalizado
    public CustomAdapter(@NonNull Context contexto_rla, String[] texto_rla, int[] imagenes_rla) {
        super(contexto_rla, R.layout.spinner_item_layout, texto_rla);
        this.contexto_rla = contexto_rla;
        this.texto_rla = texto_rla;
        this.imagenes_rla = imagenes_rla;
    }

    @Override
    // Establece cómo se mostrará la lista de valores al desplegar el spinner

```

```

    public View getDropDownView(int position, @Nullable View
convertView, @NonNull ViewGroup parent) {
        // LayoutInflater es un tipo de variable que sirve para
instanciar un archivo XML
        LayoutInflater inflater_rla = (LayoutInflater)
contexto_rla.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        // En este caso, instanciará el archivo XML con el spinner
personalizado
        View row_rla =
inflater_rla.inflate(R.layout.spinner_item_layout, null);
        // Crea dos variables para guardar los TextView e ImageView
definidos en el "spinner_item_layout.xml"
        TextView tv_divisa_rla = (TextView)
row_rla.findViewById(R.id.tv_divisa_rla);
        ImageView iv_bandera_rla = (ImageView)
row_rla.findViewById(R.id.iv_bandera_rla);
        tv_divisa_rla.setText(texto_rla[position]);
        iv_bandera_rla.setImageResource(imagenes_rla[position]);

        // Devuelve la fila del spinner personalizado
        return row_rla;
    }

    @Override
    // Establece cómo se mostrarán los valores en el spinner sin
desplegar
    public View getView(int position, @Nullable View convertView,
@NonNull ViewGroup parent) {
        // LayoutInflater para instanciar el archivo
"spinner_item_layout.xml"
        LayoutInflater inflater_rla = (LayoutInflater)
contexto_rla.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View row_rla =
inflater_rla.inflate(R.layout.spinner_item_layout, null);
        // Variables para guardar los TextView e ImageView definidos
en el "spinner_item_layout.xml"
        TextView tv_divisa_rla = (TextView)
row_rla.findViewById(R.id.tv_divisa_rla);
        ImageView iv_bandera_rla = (ImageView)
row_rla.findViewById(R.id.iv_bandera_rla);
        tv_divisa_rla.setText(texto_rla[position]);
        iv_bandera_rla.setImageResource(imagenes_rla[position]);

        // Devuelve la fila del spinner personalizado
        return row_rla;
    }
}

```

CLASES PERSONALIZACIÓN APLICACIÓN

edittext_importe_design.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Crea nueva forma para personalizar el formato del EditText -->
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item><layer-list>
        <item><shape>
            <stroke android:width="2dp"
android:color="@color/blue_600"/>
            <corners android:radius="6dp"/>
            <padding android:bottom="3dp" android:left="3dp"
android:right="0dp" android:top="3dp" />
        </shape></item>
    </layer-list></item>
</selector>
```

colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="blue_100">#BBDEFB</color>
    <color name="blue_600">#1E88E5</color>
    <color name="blue_700">#197ED2</color>
    <color name="blue_800">#1565C0</color>
    <color name="grey_400">#BDBDBD</color>
    <color name="grey_900">#212121</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
</resources>
```

strings.xml

```
<resources>
    <string name="app_name">CONVERSION DIVISAS</string>
    <string name="importe">Importe</string>
    <string name="convertir">Convertir</string>
    <string name="de">Convertir de</string>
    <string name="a">a</string>
    <string name="resultado">Resultado:</string>
</resources>
```

values\themes.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.DAM_M08_Act1_RaquelLlorente"
parent="Theme.MaterialComponents.DayNight.DarkActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/blue_700</item>
        <item name="colorPrimaryVariant">@color/blue_800</item>
        <item name="colorOnPrimary">@color/white</item>
        <!-- Secondary brand color. -->
        <item name="colorSecondary">@color/grey_400</item>
        <item name="colorSecondaryVariant">@color/blue_100</item>
        <item name="colorOnSecondary">@color/black</item>
        <!-- Status bar color. -->
        <item name="android:statusBarColor"
tools:targetApi="1">?attr/colorPrimaryVariant</item>
        <!-- Customize your theme here. -->
        <item name="android:textColor">@color/black</item>
    </style>
</resources>
```

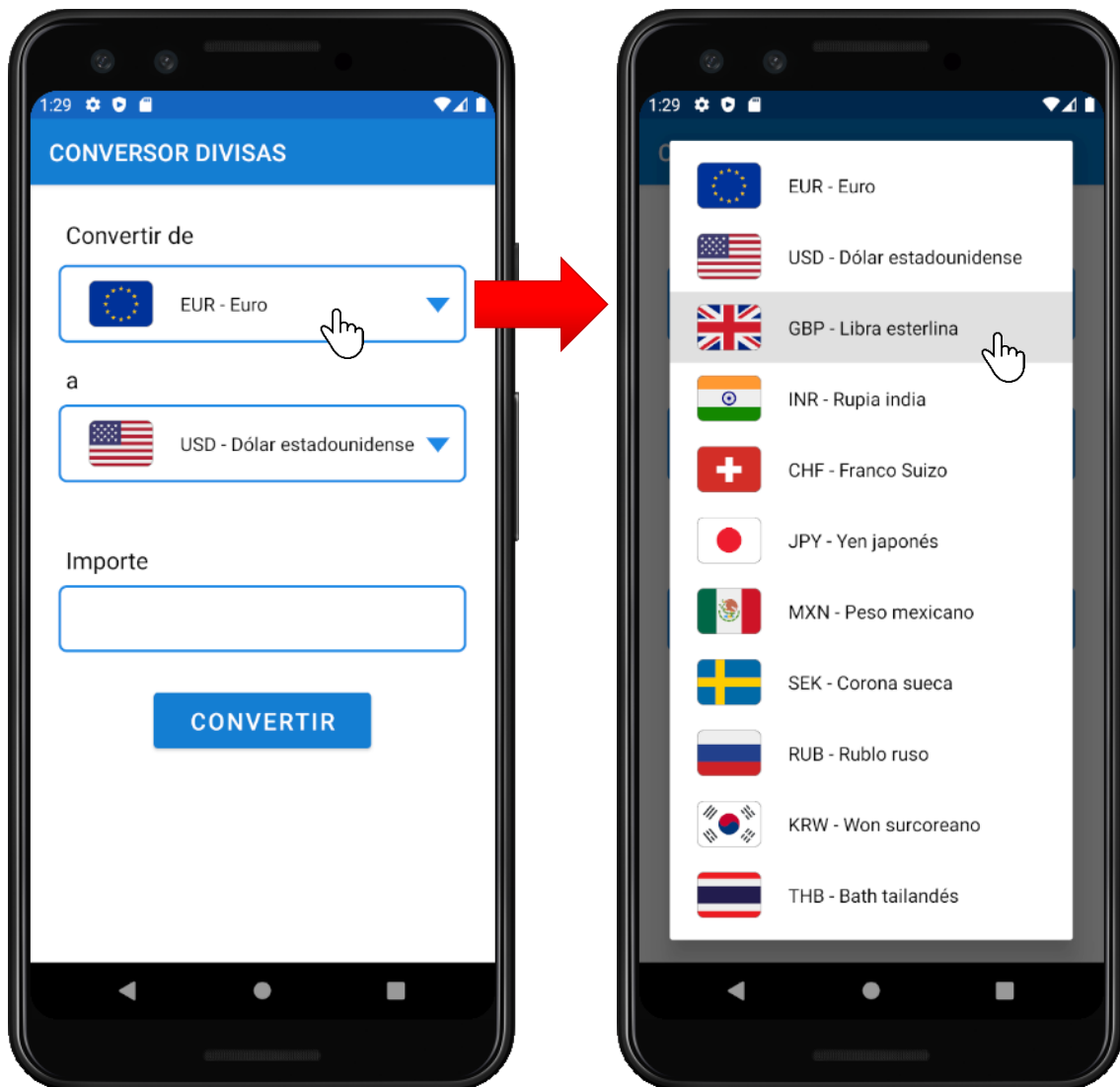
night\themes.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.DAM_M08_Act1_RaquelLlorente"
parent="Theme.MaterialComponents.DayNight.DarkActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/blue_800</item>
        <item name="colorPrimaryVariant">@color/black</item>
        <item name="colorOnPrimary">@color/white</item>
        <!-- Secondary brand color. -->
        <item name="colorSecondary">@color/grey_400</item>
        <item name="colorSecondaryVariant">@color/grey_900</item>
        <item name="colorOnSecondary">@color/black</item>
        <!-- Status bar color. -->
        <item name="android:statusBarColor"
tools:targetApi="1">?attr/colorPrimaryVariant</item>
        <!-- Customize your theme here. -->
        <item name="android:textColor">@color/white</item>
        <item name="android:colorBackground">@color/grey_900</item>
    </style>
</resources>
```

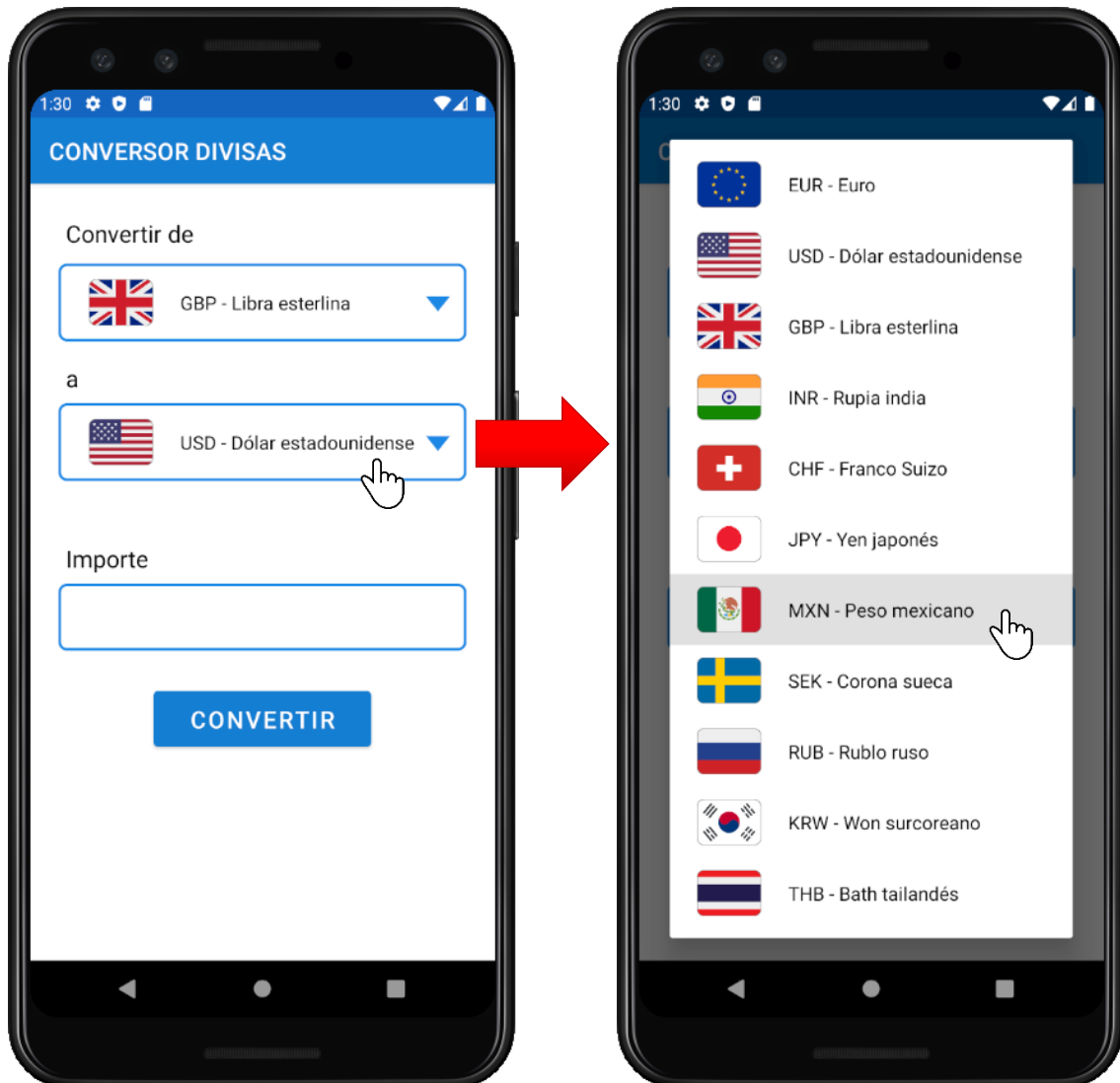
CAPTURAS DE PANTALLA

FUNCIONES BÁSICAS

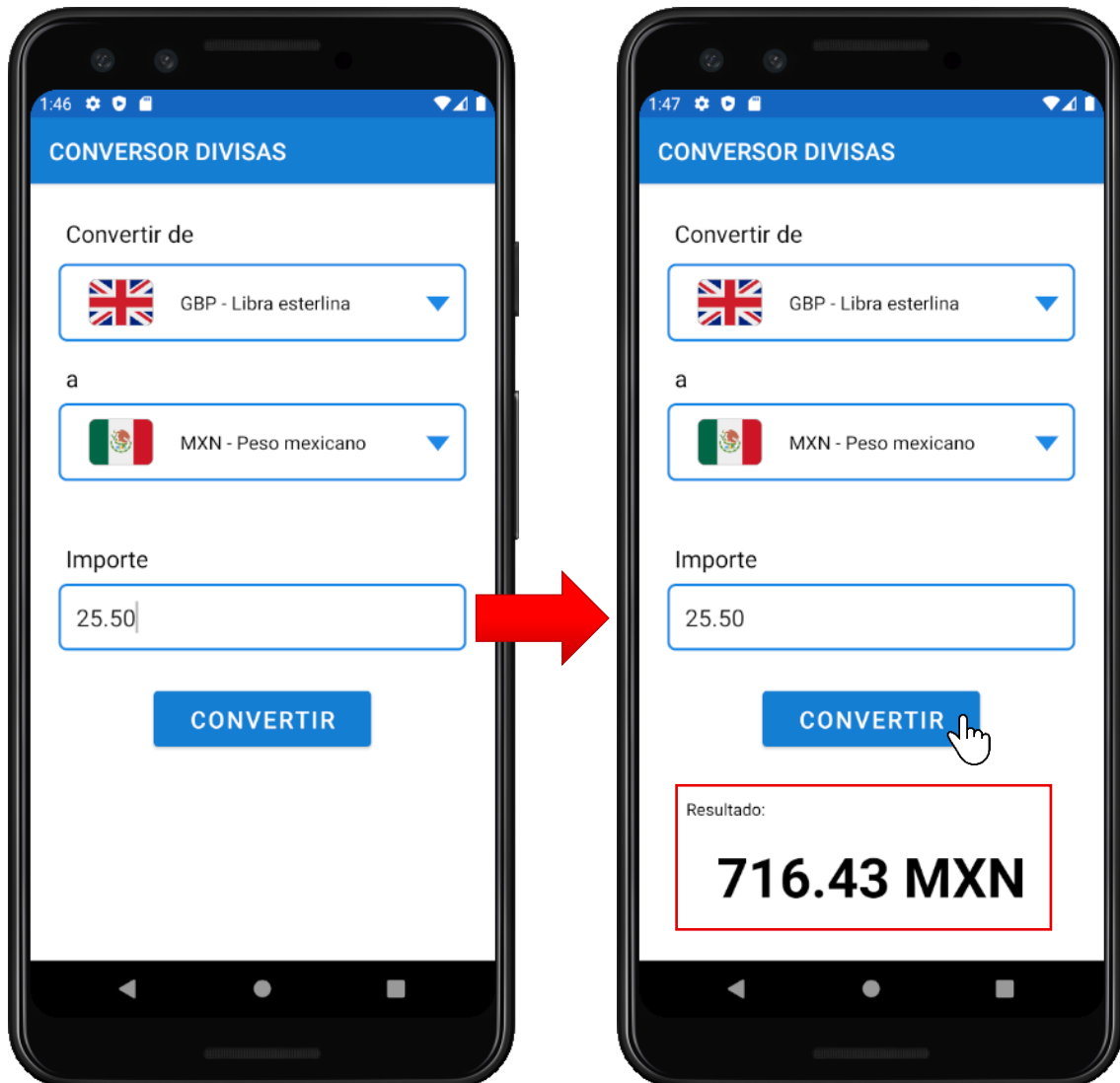
Desplegar el primer *Spinner* y seleccionar la divisa actual:



Desplegar el segundo *Spinner* y seleccionar la divisa a la que se quiere convertir:



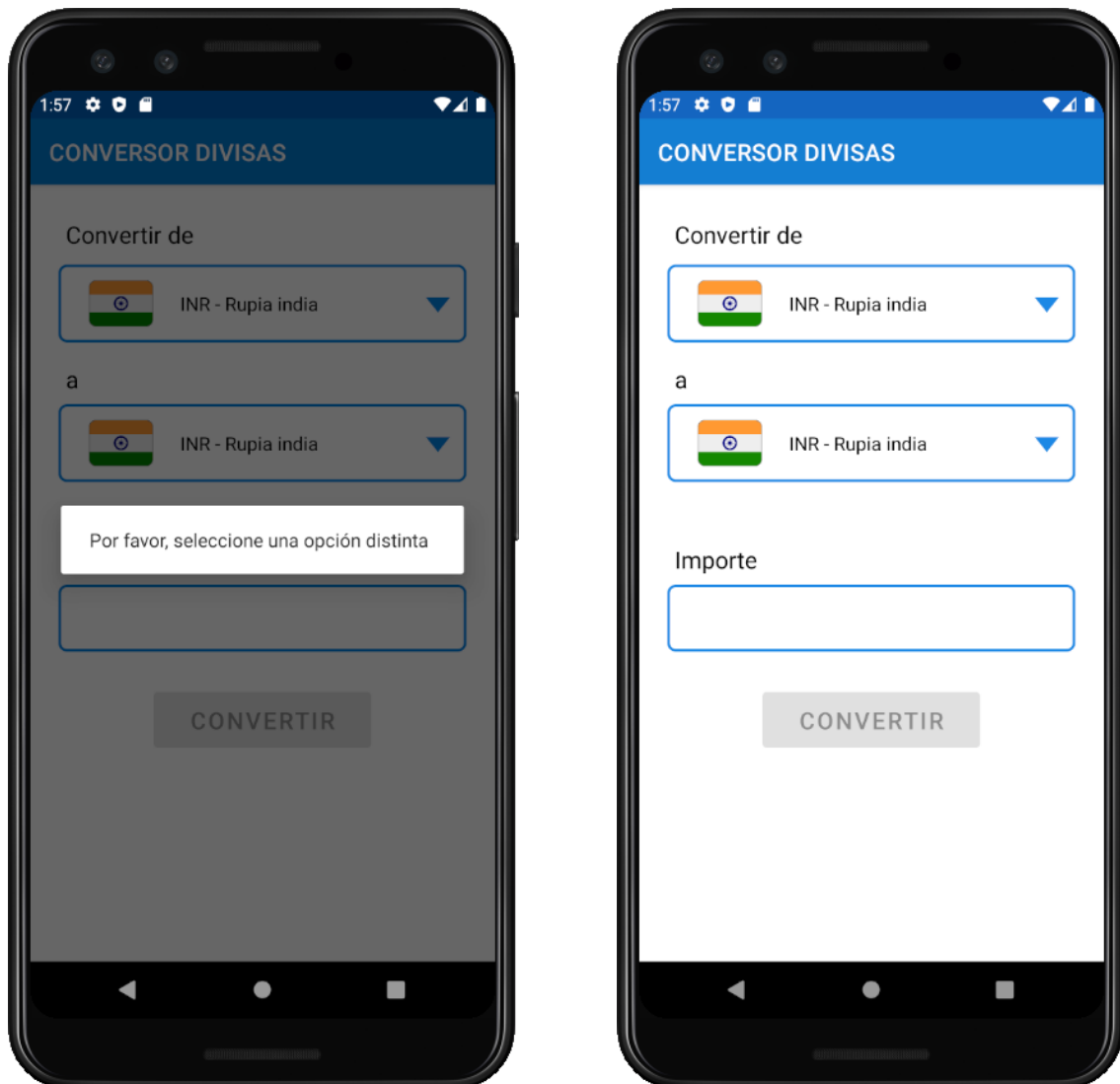
Escribir el importe y pulsar en el botón *CONVERTIR*:



ERRORES Y AVISOS

Misma divisa

Si se selecciona la misma divisa en ambos *Spinners*, aparece un mensaje de alerta advirtiéndolo que no se puede elegir la misma divisa. Además, se deshabilita el botón *CONVERTIR* y el cuadro de texto para escribir el importe.



Importe vacío

Si se pulsa en el botón *CONVERTIR* y el cuadro de texto para el importe está vacío, aparece un mensaje de error.



Valor importe inválido

Si se hace clic en el botón *CONVERTIR* y el cuadro de texto contiene un valor no numérico, aparece un mensaje de error.



Formato importe con decimales incorrecto

Si se quiere escribir un valor con decimales, se debe escribir un punto “.” para separar la parte entera de la parte decimal. En caso de escribir una coma “,” aparecerá un mensaje de error.



Importe negativo

En caso de escribir un importe negativo en el cuadro de texto, al pulsar en el botón *CONVERTIR*, aparece un mensaje advirtiéndolo que no se aceptan valores negativos.



LIGHT/DARK THEME



** Light Theme*



** Dark Theme*