

DESARROLLO DE APLICACIONES ANDROID 2

DAM – M08: PROGRAMACIÓN MULTIMEDIA
Y DISPOSITIVOS MÓVILES

UF1: DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES

RAQUEL LLORENTE ARMIJO

2021/2022

CONTENIDO

<i>CÓDIGO</i>	3
AndroidManifest.xml	3
MainActivity.java	4
activity_main.xml	12
SecondActivity.java	15
activity_second.xml	16
AdminSQLiteOpenHelper.java	17
<i>CAPTURAS DE PANTALLA</i>	18
Con conexión a Internet	18
Sin conexión a Internet	21

CÓDIGO

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.actconversordivisasrta" >

    <!-- Permiso para permitir a la aplicación acceder a Internet -->
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.ActConversorDivisasRLA" >
        <activity
            android:name=".SecondActivity"
            android:exported="true"
            android:parentActivityName=".MainActivity"/> <!--
Establece 'MainActivity' como padre de 'SecondActivity' y, por tanto,
aparece un botón
en
forma de flecha en 'SecondActivity' que nos permite volver a
'MainActivity' -->
        <activity
            android:name=".MainActivity"
            android:exported="true" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

MainActivity.java

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

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

import android.content.ContentValues;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.AsyncTask;
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 android.widget.Toast;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

import java.math.BigDecimal;
import java.math.RoundingMode;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;
import java.util.Locale;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

public class MainActivity extends AppCompatActivity {
    /* Declaración de variables */
    private EditText et_importe_rta;
    private TextView tv_result1_rta, tv_result2_rta, tv_importe_rta;
    private Button bt_convertir_rta, bt_tasas_rta;
    private Spinner sp_actual_rta, sp_nueva_rta;
    public static CustomAdapter adapter_rta;

    // public y static para que puedan ser usados por todo el java
    public static ArrayList<String> currency_rta = new
ArrayList<String>();
    public static ArrayList<String> rate_rta = new
ArrayList<String>();
    public static ArrayList<String> currency_rate_rta = new
ArrayList<String>();

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

```

        // Inicializa variables (Relación entre la parte lógica y la
        parte gráfica)
        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);
        bt_tasas_rla = (Button) findViewById(R.id.bt_tasas_rla);
        sp_actual_rla = (Spinner) findViewById(R.id.sp_actual_rla);
        sp_nueva_rla = (Spinner) findViewById(R.id.sp_nueva_rla);

        // AsyncTask que se conecta a un servidor
        new
        ConexionAsyncTask().execute("https://www.ecb.europa.eu/stats/eurofxref
/eurofxref-daily.xml");

        // 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);

            // Parsear ratios
            Double [] num_rate_rla = new
Double[rate_rla.size()];
            for(int i = 0; i< rate_rla.size(); i++) {
                num_rate_rla[i] =
Double.parseDouble(rate_rla.get(i));
            }

            // Calcula la conversión: Multiplica el ratio de
la nueva moneda seleccionada por el importe escrito por
            // el usuario y divide el resultado entre el

```

```

ratio de la moneda actual
        Double result_rla =
(num_rate_rla[pos_sp_nueva_rla]*num_importe_rla) /
num_rate_rla[pos_sp_actual_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() + " "
+ currency_rla.get(pos_sp_nueva_rla));
        // 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);

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

// La clase AsyncTask es una interfaz que permite ejecutar tareas
en segundo plano dentro del contexto
// del ciclo de vida de una aplicación
private class ConexionAsyncTask extends AsyncTask<String, Integer,
String> {

    // Array formado por las imágenes de las banderas
    public int[] banderas_rla = {R.drawable.usd, R.drawable.jpy,
R.drawable.bgn, R.drawable.czg, R.drawable.dkk, R.drawable.gbp,
R.drawable.huf, R.drawable.pln, R.drawable.ron,
R.drawable.sek, R.drawable.chf, R.drawable.isk, R.drawable.nok,
R.drawable.hrk, R.drawable.rub, R.drawable.tr,
R.drawable.aud, R.drawable.brl, R.drawable.cad, R.drawable.cny,
R.drawable.hkd, R.drawable.idr, R.drawable.ils,
R.drawable.inr, R.drawable.krw, R.drawable.mxn, R.drawable.myr,
R.drawable.nzd, R.drawable.php, R.drawable.sgd,
R.drawable.thb, R.drawable.zar};

    // Método que se encarga de realizar una tarea en segundo
plano
    @Override
    protected String doInBackground(String... direccion_rla) {
        String servidor_rla = direccion_rla[0];
        String resultado_rla = new String();
        HttpURLConnection conexion_rla = null;
        try {
            // Conexión a Internet
            URL url = new URL(servidor_rla);
            conexion_rla = (HttpURLConnection)
url.openConnection();
            conexion_rla.setRequestMethod("GET");
            conexion_rla.connect();

```

```

        // Si consigue establecer la conexión, establece
        '¡Conectado!' como valor de la variable 'resultado_rla'
        resultado_rla = "¡Conectado!";

        // borra los elementos de los 3 ArrayList ya que va a
        actualizar los valores con los datos obtenidos de Internet
        currency_rla.clear();
        rate_rla.clear();
        currency_rate_rla.clear();

        // lee el XML devuelto por la URL y lo guarda en un
        objeto de tipo Document
        DocumentBuilderFactory factory_rla =
        DocumentBuilderFactory.newInstance();
        DocumentBuilder builder_rla =
        factory_rla.newDocumentBuilder();
        Document doc_rla = builder_rla.parse(new
        URL(servidor_rla).openStream());

        // recorre 'doc_rla' y guarda las etiquetas llamadas
        'Cube' en un objeto de tipo NodeList ya que las etiquetas 'Cube'
        // son las que contienen los atributos con el
        nombre de la moneda (currency) y el valor del ratio (rate)
        NodeList nodeList_rla = (NodeList)
        doc_rla.getElementsByTagName("Cube");

        // recorre la lista de nodos 'nodeList_rla'
        for(int i=0; i<nodeList_rla.getLength(); i++){
            Node node_rla = nodeList_rla.item(i);
            Element element_rla = (Element) node_rla;
            // guarda los valores de los atributos 'currency'
            y 'rate' en 2 variables
            String varCurrency_rla = (String)
            element_rla.getAttribute("currency");
            String varRate_rla = (String)
            element_rla.getAttribute("rate");

            /* No todas las etiquetas llamadas 'Cube' tienen
            los atributos 'currency' y 'rate' */
            // selecciona solo las etiquetas 'Cube' cuyo
            atributo 'currency' devuelva algún valor
            if (varCurrency_rla != "") {
                currency_rla.add(varCurrency_rla);
                rate_rla.add(varRate_rla);
                currency_rate_rla.add("Currency: " +
                varCurrency_rla + " - Rate: " + varRate_rla);
            }
        }

        // Si no puede establecer la conexión, es decir, si se
        captura alguna excepción, establece 'error' como valor de la variable
        'resultado_rla'
    } catch (Exception e) {
        resultado_rla = "error";

        // Se produzca o no un error siempre devolverá el valor de
        la variable 'resultado_rla'
    } finally {
        return resultado_rla; // pasa el resultado al método
        'postExecute()'
    }

```



```

    }
}

// Método que se ejecuta una vez termina la ejecución de la
// tarea en segundo plano
@Override
protected void onPostExecute(String resultado_rla) {
    super.onPostExecute(resultado_rla);

    // Abre una instancia de la base de datos en modo lectura
    // y escritura
    AdminSQLiteOpenHelper admin_rla = new
    AdminSQLiteOpenHelper(MainActivity.this, "administracion", null, 1);
    SQLiteDatabase baseDatos_rla =
    admin_rla.getWritableDatabase();

    // si se ha establecido la conexión
    if(resultado_rla.equals(";Conectado!")){
        // borra todos los datos de la tabla 'tablaDivisasRLA'
        // para poder actualizarla con los datos obtenidos de Internet
        baseDatos_rla.delete("tablaDivisasRLA", null, null);

        for(int i = 0; i< currency_rla.size(); i++) {
            // crea un objeto de la clase 'ContentValues'.
            // Este objeto representará una nueva fila(registro) en la tabla de la BD
            ContentValues registro_rla = new ContentValues();
            // rellena cada columna (id, currency, rate) de la
            // nueva fila con el valor correspondiente
            registro_rla.put("id", i);
            registro_rla.put("currency", currency_rla.get(i));
            registro_rla.put("rate", rate_rla.get(i));
            // inserta la nueva fila en la tabla que hemos
            // creado en la clase 'AdminSQLiteOpenHelper'
            baseDatos_rla.insert("tablaDivisasRLA", null,
            registro_rla);
        }
        // cierra la BD
        baseDatos_rla.close();

        // enlaza el objeto de la clase CustomAdapter con cada
        // uno de los Spinners
        adapter_rla = new CustomAdapter(MainActivity.this,
        currency_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);

        // muestra un mensaje indicando que hay conexión a
        // Internet
        Toast.makeText(MainActivity.this, ";Conexión
        establecida correctamente!", Toast.LENGTH_SHORT).show();
    }

    // si NO se ha podido establecer la conexión
    if(resultado_rla.equals("error")) {

        // realiza una consulta sobre la BD 'baseDatos_rla'
        // para obtener los valores de las columnas 'currency' y 'rate' de
        // la tabla 'tablaDivisasRLA'. Guarda los resultados
    }
}

```

```

en la variable 'fila_rla' de tipo Cursor
        Cursor fila_rla = baseDatos_rla.rawQuery("select
currency, rate from tablaDivisasRLA", null);

        /* la primera vez que se abre la app es necesario
tener conexión para poder rellenar la base de datos */
        // si la consulta no devuelve ningún resultado quiere
decir que es la primera vez que abre la app
        if(fila_rla.getCount() == 0){
            // muestra un mensaje de alerta avisando de que la
primera vez hay que tener conexión a Internet
            AlertDialog.Builder alerta_rla = new
AlertDialog.Builder(MainActivity.this);
            alerta_rla.setMessage("La primera vez que inicias
la app debes tener conexión para cargar los datos en la base de
datos");

            AlertDialog titulo = alerta_rla.create();
            titulo.setTitle("ERROR");
            titulo.show();

            // se deshabilita los spinners, los botones y el
EditText para escribir el importe
            sp_nueva_rla.setEnabled(false);
            sp_actual_rla.setEnabled(false);
            bt_tasas_rla.setEnabled(false);
            bt_convertir_rla.setEnabled(false);
            et_importe_rla.setEnabled(false);

            // si la consulta devuelve algún resultado quiere
decir que la base de datos tiene elementos
        } else {
            // borra los elementos de los 3 ArrayList ya que
va a actualizar los valores con los datos de la base de datos
            currency_rla.clear();
            rate_rla.clear();
            currency_rate_rla.clear();

            // recorre el conjunto de resultados devueltos por
la consulta
            fila_rla.moveToFirst();
            do {
                // añade el valor de cada columna en el
ArrayList correspondiente
                // 'toLowerCase()' simplemente es para ver
visualmente cuándo usa los datos de la BD (en minúscula) y cuándo
// los de Internet (que por defecto
están en mayúscula)

                currency_rla.add(fila_rla.getString(0).toString().toLowerCase(Locale.R
OOT));

                rate_rla.add(fila_rla.getString(1).toString());
            } while (fila_rla.moveToNext());

            // enlaza el objeto de la clase CustomAdapter con
cada uno de los Spinners
            adapter_rla = new CustomAdapter(MainActivity.this,
currency_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);

    // rellena el ArrayList 'currency_rate_rla'
    for(int i= 0; i<currency_rla.size(); i++){
        currency_rate_rla.add("Currency: " +
currency_rla.get(i) + " - Rate: " + rate_rla.get(i));
    }

    // muestra un mensaje indicando que no hay
conexión a Internet
    Toast.makeText(MainActivity.this, "AVISO: No hay
conexión, se cargarán los datos de la base de datos",
Toast.LENGTH_SHORT).show();
    }
}

// Método para llamar a la Activity "SecondActivity"
public void launchSecondActivity(View view){
    // Intent describe la Activity que se debe iniciar y contiene
los datos necesarios para ello
    Intent intent_rla = new Intent(this, SecondActivity.class);
    // Inicia la Activity definida en "intent_rla"
    startActivity(intent_rla);
}
}
```

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"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tv_actual_rla"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="24dp"
        android:text="@string/de"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="@+id/sp_actual_rla"
        app:layout_constraintTop_toTopOf="parent" />

    <Spinner
        android:id="@+id/sp_actual_rla"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="24dp"
        android:layout_marginEnd="8dp"
        android:background="@drawable/spinner_shape"
        android:minHeight="48dp"
        android:spinnerMode="dialog"
        app:layout_constraintEnd_toStartOf="@+id/sp_nueva_rla"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/sp_nueva_rla" />

    <Spinner
        android:id="@+id/sp_nueva_rla"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="58dp"
        android:layout_marginEnd="24dp"
        android:layout_marginBottom="15dp"
        android:background="@drawable/spinner_shape"
        android:minHeight="48dp"
        android:spinnerMode="dialog"
        app:layout_constraintBottom_toTopOf="@+id/bt_tasas_rla"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toEndOf="@+id/sp_actual_rla"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tv_nueva_rla"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
```

```

        android:layout_marginStart="16dp"
        android:layout_marginTop="24dp"
        android:text="@string/a"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="@+id/sp_nueva_rla"
        app:layout_constraintTop_toTopOf="parent" />

<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="11dp"
    android:background="@drawable/edittext_importe_design"
    android:inputType="numberDecimal"
    android:padding="15dp"
    android:textSize="24sp"
    app:layout_constraintBottom_toTopOf="@+id/bt_convertir_rla"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/bt_tasas_rla" />

<Button
    android:id="@+id/bt_convertir_rla"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingLeft="20dp"
    android:paddingTop="10dp"
    android:paddingRight="20dp"
    android:paddingBottom="10dp"
    android:text="@string/convertir"
    android:textAllCaps="true"
    android:textSize="24sp"
    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
    android:id="@+id/tv_result1_rla"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="12dp"
    android:text="@string/resultado"
    android:textSize="20sp"
    android:visibility="invisible"
    app:layout_constraintBottom_toTopOf="@+id/tv_result2_rla"
    app:layout_constraintStart_toStartOf="@+id/tv_result2_rla" />

<TextView
    android:id="@+id/tv_result2_rla"
    android:layout_width="0dp"
    android:layout_height="80dp"
    android:layout_marginStart="32dp"
    android:layout_marginTop="64dp"
    android:layout_marginEnd="32dp"
    android:autoSizeTextType="uniform"
    android:padding="0dp"
    android:textAlignment="center"

```

```
        android:textStyle="bold"
        android:visibility="invisible"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/bt_convertir_rla" />

<TextView
    android:id="@+id/tv_importe_rla"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginBottom="12dp"
    android:text="@string/importe"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toTopOf="@+id/et_importe_rla"
    app:layout_constraintStart_toStartOf="@+id/et_importe_rla" />

<Button
    android:id="@+id/bt_tasas_rla"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="84dp"
    android:onClick="launchSecondActivity"
    android:paddingLeft="20dp"
    android:paddingTop="10dp"
    android:paddingRight="20dp"
    android:paddingBottom="10dp"
    android:text="@string/tasas"
    android:textAllCaps="false"
    android:textSize="20sp"
    app:layout_constraintBottom_toTopOf="@+id/et_importe_rla"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/sp_nueva_rla" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

SecondActivity.java

```
/* Segunda Activity (Esta activity se ejecutará al pulsar el botón  
'bt_tasas_rla') */  
package com.example.actconversordivisasr1a;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.os.Bundle;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
  
public class SecondActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_second);  
  
        // Objeto ListView en el que se mostrará todas las  
        monedas(currency) con su ratio(rate) correspondiente  
        ListView lv_tasas_rla = (ListView)  
        findViewById(R.id.lv_tasas_rla);  
        // Define el ArrayAdapter indicando el formato y los valores  
        (ArrayList 'currency_rate_rla')  
        ArrayAdapter<String> adapter_rla = new  
        ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,  
        android.R.id.text1, MainActivity.currency_rate_rla);  
        // Enlaza el ArrayAdapter 'adapter_rla' con la ListView  
        'lv_tasas_rla'  
        lv_tasas_rla.setAdapter(adapter_rla);  
    }  
}
```

activity_second.xml

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

    <ListView
        android:id="@+id/lv_tasas_rla"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="6dp"
        android:divider="@color/blue_100"
        android:dividerHeight="2dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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


AdminSQLiteOpenHelper.java

```
/* Clase para administrar la base de datos SQLite */
package com.example.actconversordivisasrta;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

public class AdminSQLiteOpenHelper extends SQLiteOpenHelper {

    public AdminSQLiteOpenHelper(@Nullable Context context, @Nullable
String name, @Nullable SQLiteDatabase.CursorFactory factory, int
version) {
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase_rla) {
        // crea una tabla llamada 'tablaDivisasRLA' con 3 columnas
        ('id','currency','rate') en la base de datos 'sqLiteDatabase_rla'
        sqLiteDatabase_rla.execSQL("create table tablaDivisasRLA (id
int primary key, currency text, rate text)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int
i1) {
    }
}
```

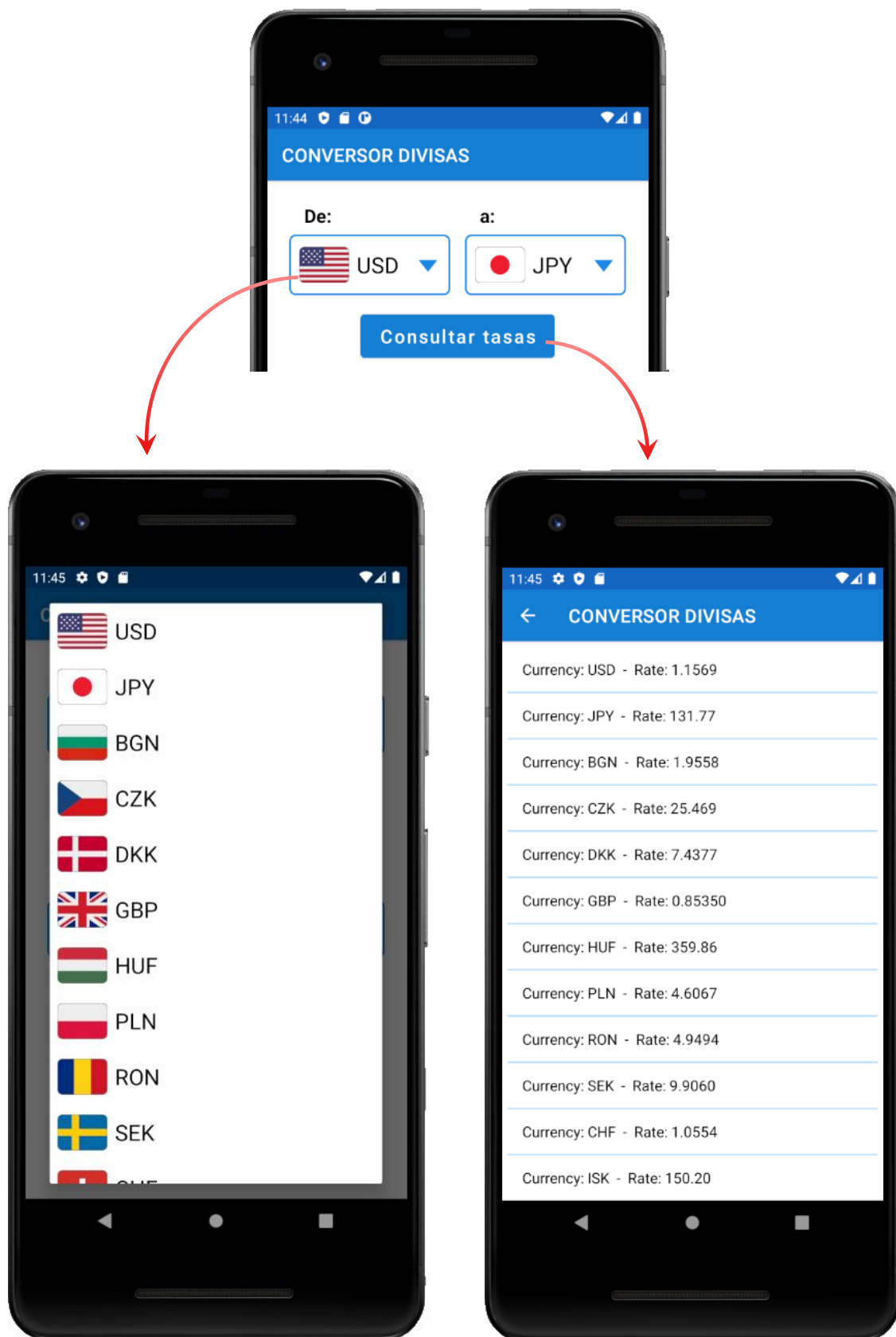
CAPTURAS DE PANTALLA

CON CONEXIÓN A INTERNET

Si hay conexión a Internet muestra el siguiente mensaje:



Rellena los dos Spinners y el ListView de la segunda Activity con los datos obtenidos de Internet:

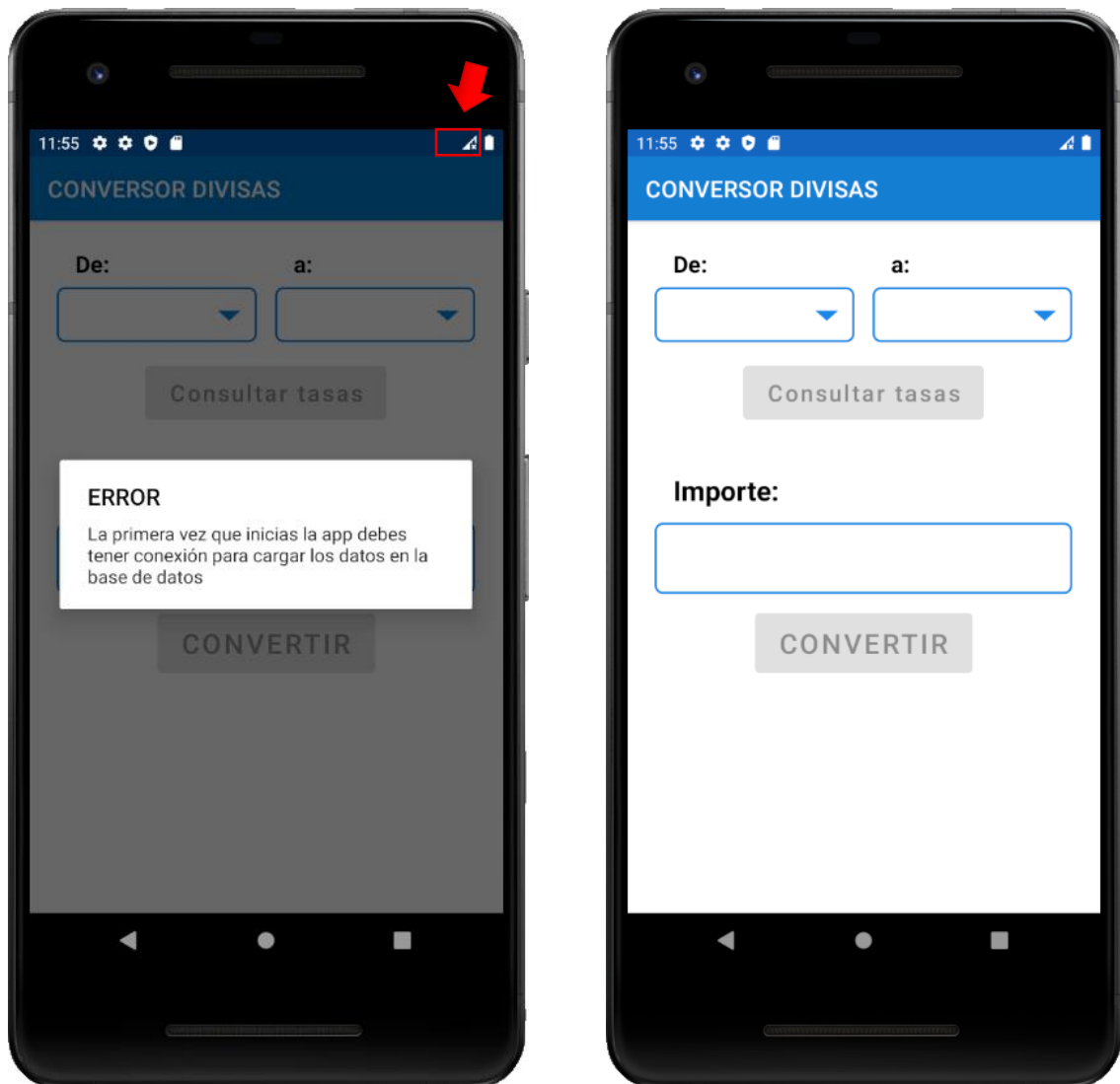


Permite hacer conversiones entre cualquiera de las monedas disponibles:



SIN CONEXIÓN A INTERNET

Si la primera vez que se inicia la aplicación no hay conexión, muestra un mensaje de error y deshabilita todas las funcionalidades de la aplicación:

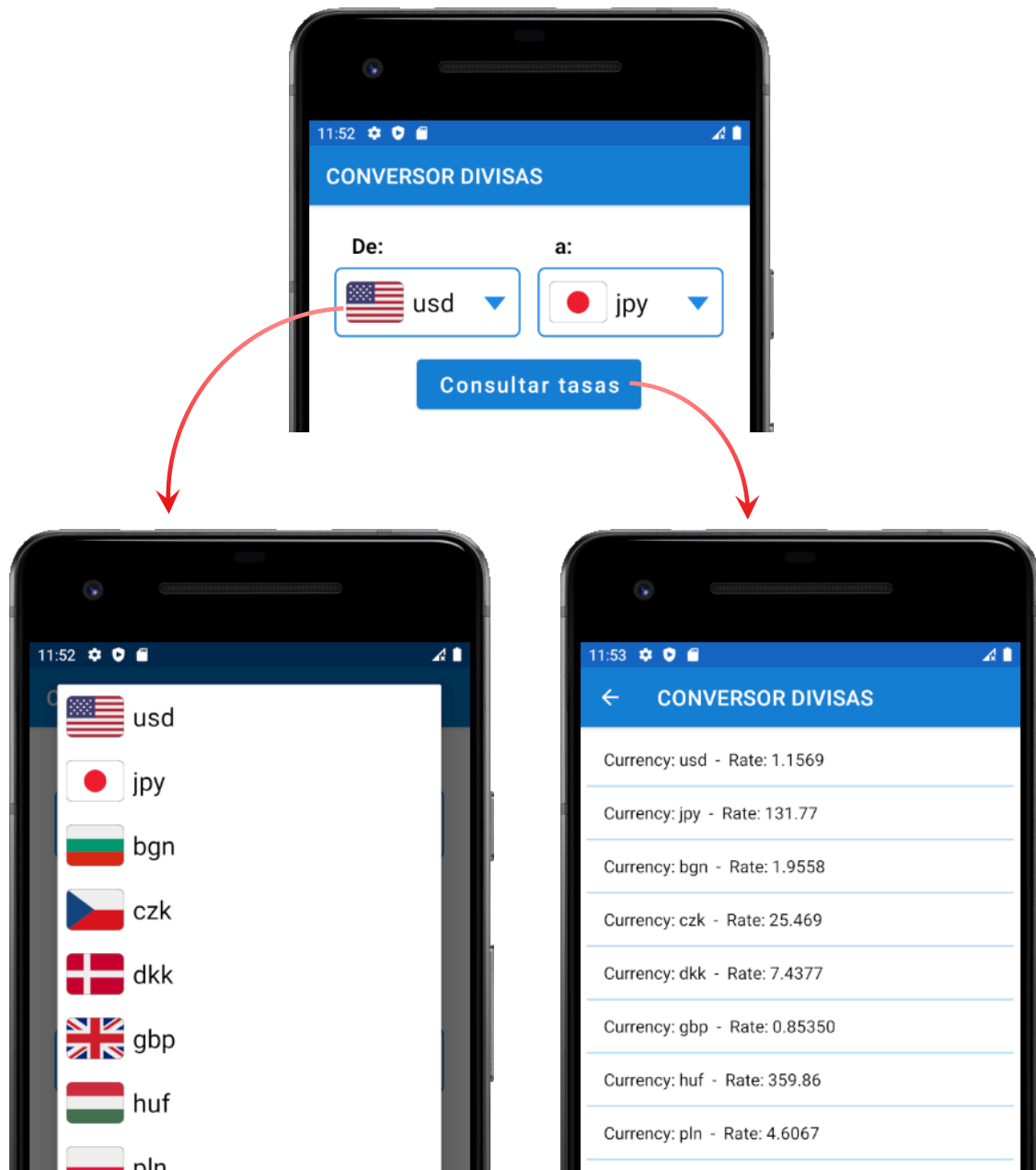


¡Este error no desaparecerá hasta obtener conexión a Internet!

Si la primera vez que se inicia la aplicación sí había conexión y, por tanto, se han cargado los datos en la base de datos, muestra un mensaje de aviso para que el usuario sepa que los datos con los que estará trabajando pueden no estar actualizados:



Rellena los dos Spinners y el ListView de la segunda Activity con los datos obtenidos de la base de datos:



Cuando carga los datos de la base de datos esta aplicación está programada para que muestre el valor de 'currency' en minúscula. Esta función simplemente se ha añadido para que sea más sencillo identificar cuándo se cargan los datos de Internet (por defecto, en mayúscula) y cuándo se cargan de la base de datos (en minúscula). No obstante, en una aplicación real no tendría mucho sentido.

Permite hacer conversiones entre cualquiera de las monedas disponibles:

