



Unidad Didáctica 2: EJEMPLO 2

Programación Dirigida a Eventos

1. Enunciado

En este ejercicio práctico, desarrollaremos una aplicación Android que funcione con tareas en segundo plano, siguiendo los Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030, específicamente el ODS 7: Energía asequible y no contaminante. La aplicación ayudará a los usuarios a monitorear y optimizar el consumo energético en sus hogares.

Ejercicio 2: Monitoreo de Consumo Energético en el Hogar

Introducción

La eficiencia energética es fundamental para la sostenibilidad y la reducción del impacto ambiental. Las aplicaciones móviles pueden ayudar a los usuarios a monitorear y gestionar su consumo de energía, proporcionando información valiosa y recomendaciones para mejorar la eficiencia. Este ejercicio se centra en desarrollar una aplicación que realice tareas en segundo plano para recopilar datos de consumo energético y presentar esta información de manera accesible.

Enunciado del Problema

Desarrollar una aplicación Android que permita a los usuarios:

1. Registrar el consumo diario de energía.
2. Monitorear el uso de energía en tiempo real.
3. Recibir recomendaciones para optimizar el consumo energético.
4. Visualizar un resumen diario del consumo energético.

2. Solución

A continuación, se presenta una solución detallada para desarrollar la aplicación propuesta.

Paso 1: Configuración del Proyecto

1. Iniciar un nuevo proyecto en Android Studio con una "Actividad Vacía".
2. Configurar los archivos build.gradle para asegurarse de tener las dependencias necesarias.

```
// build.gradle (Project level)
allprojects {
```

```
repositories {
    google()
    mavenCentral()
}

// build.gradle (Module level)
dependencies {
    implementation 'androidx.appcompat:appcompat:1.3.0'
    implementation 'com.google.android.material:material:1.4.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'androidx.lifecycle:lifecycle-livedata-ktx:2.3.1'
    implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.3.1'
}
```

Paso 2: Diseño de la Interfaz de Usuario

Crear un archivo XML para la actividad principal (activity_main.xml) que incluya campos de entrada para el consumo de energía, una gráfica para el monitoreo en tiempo real, y un TextView para mostrar el resumen diario.

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<EditText
    android:id="@+id/editTextEnergyConsumption"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Consumo de energía (kWh)"
    android:inputType="numberDecimal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/buttonSave"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Guardar"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editTextEnergyConsumption" />

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

```

        android:text="Resumen diario"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/buttonSave" />

<com.github.mikephil.charting.charts.LineChart
    android:id="@+id/lineChart"
    android:layout_width="Odp"
    android:layout_height="Odp"
    app:layout_constraintTop_toBottomOf="@+id/textViewSummary"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>

```

Paso 3: Implementación de la Actividad Principal

Implementar la lógica en MainActivity.java para manejar la entrada de datos y realizar tareas en segundo plano usando AsyncTask para simular el monitoreo en tiempo real del consumo energético.

```

package com.example.energymonitor;

import androidx.appcompat.app.AppCompatActivity;
import androidx.lifecycle.ViewModelProvider;

import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;

import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {

    private EditText editTextEnergyConsumption;
    private TextView textViewSummary;
    private Button buttonSave;
    private LineChart lineChart;

    private EnergyViewModel energyViewModel;

```

```

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

    editTextEnergyConsumption = findViewById(R.id.editTextEnergyConsumption);
    textViewSummary = findViewById(R.id.textViewSummary);
    buttonSave = findViewById(R.id.buttonSave);
    lineChart = findViewById(R.id.lineChart);

    energyViewModel = new ViewModelProvider(this).get(EnergyViewModel.class);

    buttonSave.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            saveData();
        }
    });

    energyViewModel.getEnergyData().observe(this, data -> updateChart(data));
    energyViewModel.getSummary().observe(this, summary -> textViewSummary.setText(summary));

    new EnergyMonitorTask().execute();
}

private void saveData() {
    String energy = editTextEnergyConsumption.getText().toString();

    if (energy.isEmpty()) {
        Toast.makeText(this, "Por favor, complete el campo de consumo de energía",
            Toast.LENGTH_SHORT).show();
    } else {
        energyViewModel.saveEnergyData(Float.parseFloat(energy));
    }
}

private void updateChart(List<Entry> data) {
    LineDataSet lineDataSet = new LineDataSet(data, "Consumo Energético");
    LineData lineData = new LineData(lineDataSet);
    lineChart.setData(lineData);
    lineChart.invalidate();
}

private class EnergyMonitorTask extends AsyncTask<Void, List<Entry>, Void> {
    @Override
    protected Void doInBackground(Void... voids) {
        List<Entry> data = new ArrayList<>();
        for (int i = 0; i < 24; i++) {
            data.add(new Entry(i, (float) (Math.random() * 10)));
            publishProgress(data);
        }
        try {

```

```

        Thread.sleep(1000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
return null;
}

@Override
protected void onProgressUpdate(List<Entry>... values) {
    super.onProgressUpdate(values);
    energyViewModel.setEnergyData(values[0]);
}
}
}

```

Paso 4: Implementación del ViewModel

Crear una clase EnergyViewModel.java para manejar los datos de consumo energético y proporcionar un resumen diario.

```

package com.example.energymonitor;

import androidx.lifecycle.LiveData;
import androidx.lifecycle.MutableLiveData;
import androidx.lifecycle.ViewModel;

import com.github.mikephil.charting.data.Entry;

import java.util.ArrayList;
import java.util.List;

public class EnergyViewModel extends ViewModel {

    private final MutableLiveData<List<Entry>> energyData = new MutableLiveData<>();
    private final MutableLiveData<String> summary = new MutableLiveData<>();

    public LiveData<List<Entry>> getEnergyData() {
        return energyData;
    }

    public LiveData<String> getSummary() {
        return summary;
    }

    public void saveEnergyData(float energy) {
        List<Entry> currentData = energyData.getValue();
        if (currentData == null) {
            currentData = new ArrayList<>();
        }
        currentData.add(new Entry(currentData.size(), energy));
    }
}

```

```
        energyData.setValue(currentData);

        updateSummary(currentData);
    }

    public void setEnergyData(List<Entry> data) {
        energyData.setValue(data);
        updateSummary(data);
    }

    private void updateSummary(List<Entry> data) {
        float total = 0;
        for (Entry entry : data) {
            total += entry.getY();
        }
        summary.setValue("Consumo total de energía: " + total + " kWh");
    }
}
```

3. Conclusión

Este ejercicio práctico permite a los estudiantes aplicar conceptos de desarrollo de aplicaciones Android para crear una herramienta útil que contribuye al ODS 7: Energía asequible y no contaminante. A través de esta actividad, los estudiantes desarrollan habilidades en la creación de interfaces de usuario interactivas, el manejo de tareas en segundo plano y la optimización del consumo energético.

WELCOME
TO
UAX

UAX

Universidad
Alfonso X el Sabio

GRACIAS

UAX.COM