



Unidad Didáctica 4: EJEMPLO 4

Programación Dirigida por Eventos

1. Enunciado

En este ejercicio práctico, desarrollaremos una aplicación Android que expanda la experiencia del usuario mediante el uso de fragmentos y widgets, siguiendo los Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030, específicamente el ODS 4: Educación de Calidad. La aplicación ayudará a los estudiantes a gestionar y acceder a recursos educativos de manera más eficiente.

Ejercicio 4: Plataforma de Recursos Educativos

Introducción

La educación de calidad es esencial para el desarrollo personal y profesional. Las aplicaciones móviles pueden mejorar la experiencia de aprendizaje proporcionando acceso rápido y fácil a recursos educativos. Este ejercicio se centra en desarrollar una aplicación que utilice fragmentos para organizar el contenido y widgets para proporcionar acceso rápido a información importante.

Enunciado del Problema

Desarrollar una aplicación Android que permita a los usuarios:

1. Navegar por diferentes secciones de recursos educativos mediante fragmentos.
2. Utilizar widgets para acceso rápido a recursos destacados.
3. Visualizar detalles y descripciones de los recursos educativos.
4. Guardar y actualizar el progreso de estudio.

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.fragment:fragment:1.3.4'
    implementation 'androidx.lifecycle:lifecycle-livedata-ktx:2.3.1'
    implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.3.1'
    implementation 'androidx.recyclerview:recyclerview:1.2.0'
}

```

Paso 2: Diseño de la Interfaz de Usuario

Crear un archivo XML para la actividad principal (activity_main.xml) que incluya un FrameLayout para cargar fragmentos dinámicamente y un BottomNavigationView para la navegación.

```

<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">

    <FrameLayout
        android:id="@+id/fragment_container"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toTopOf="@id/bottom_navigation"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottom_navigation"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        app:menu="@menu/bottom_navigation_menu"
        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 navegación y cargar fragmentos dinámicamente.

```
package com.example.educationalresources;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

import android.os.Bundle;
import android.view.MenuItem;

import com.google.android.material.bottomnavigation.BottomNavigationView;

public class MainActivity extends AppCompatActivity {

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

        BottomNavigationView bottomNavigationView = findViewById(R.id.bottom_navigation);
        bottomNavigationView.setOnNavigationItemSelectedListener(new
        BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(MenuItem item) {
                Fragment selectedFragment = null;
                switch (item.getItemId()) {
                    case R.id.nav_home:
                        selectedFragment = new HomeFragment();
                        break;
                    case R.id.nav_resources:
                        selectedFragment = new ResourcesFragment();
                        break;
                    case R.id.nav_profile:
                        selectedFragment = new ProfileFragment();
                        break;
                }
                loadFragment(selectedFragment);
                return true;
            }
        });
    }
}
```

```
// Cargar el fragmento inicial
if (savedInstanceState == null) {
    bottomNavigationView.setSelectedItemId(R.id.nav_home);
}
}

private void loadFragment(Fragment fragment) {
    FragmentManager fragmentManager = getSupportFragmentManager();
    FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
    fragmentTransaction.replace(R.id.fragment_container, fragment);
    fragmentTransaction.commit();
}
}
```

Paso 4: Creación de los Fragmentos

Crear los fragmentos HomeFragment, ResourcesFragment, y ProfileFragment con su respectivo diseño XML.

HomeFragment.java

```
package com.example.educationalresources;

import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class HomeFragment extends Fragment {
    public HomeFragment() {
        // Constructor vacío requerido
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        // Inflar el layout del fragmento y devolverlo como la vista raíz
        return inflater.inflate(R.layout.fragment_home, container, false);
    }
}
```

fragment_home.xml

```
<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=".HomeFragment">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Bienvenido a la Plataforma de Recursos Educativos"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Paso 5: Implementación de los Widgets

Crear un widget para mostrar recursos educativos destacados.

res/xml/widget_info.xml

```
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
    android:minWidth="250dp"
    android:minHeight="60dp"
    android:updatePeriodMillis="86400000"
    android:initialLayout="@layout/widget_layout"
    android:widgetCategory="home_screen">
</appwidget-provider>
```

widget_layout.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="8dp">

    <TextView
        android:id="@+id/widget_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Recursos Destacados" />

    <TextView
        android:id="@+id/widget_content"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Cargando..." />
</LinearLayout>
```

MyAppWidgetProvider.java

```
package com.example.educationalresources;
```

```
import android.app.PendingIntent;
import android.appwidget.AppWidgetManager;
import android.appwidget.AppWidgetProvider;
import android.content.Context;
import android.content.Intent;
import android.widget.RemoteViews;

public class MyAppWidgetProvider extends AppWidgetProvider {
    @Override
    public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[]
appWidgetIds) {
        for (int appWidgetId : appWidgetIds) {
            Intent intent = new Intent(context, MainActivity.class);
            PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, intent, 0);

            RemoteViews views = new RemoteViews(context.getPackageName(),
R.layout.widget_layout);
            views.setOnClickPendingIntent(R.id.widget_title, pendingIntent);

            views.setTextViewText(R.id.widget_content, "Recurso: Curso de Matemáticas");

            appWidgetManager.updateAppWidget(appWidgetId, views);
        }
    }
}
```

Paso 6: Configuración del Manifiesto

Actualizar el manifiesto para incluir el provider del widget.

```
<receiver android:name=".MyAppWidgetProvider">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
    </intent-filter>
    <meta-data android:name="android.appwidget.provider"
android:resource="@xml/widget_info" />
</receiver>
```

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 4: Educación de Calidad. A través de esta actividad, los estudiantes desarrollan habilidades en la creación de interfaces de usuario interactivas, el uso de fragmentos y la implementación de widgets, mejorando así la accesibilidad y eficiencia en la gestión de recursos educativos.

WELCOME
TO
UAX

UAX

Universidad
Alfonso X el Sabio

GRACIAS

UAX.COM