**UNIVERSIDAD ACADEMICA DE LA REGION NORTE DE GUERRERO UT**

**Docente**; Mtro.; **Abel Jerónimo Vargas**

**Alumno:**

* Roberto Chauteco Bello

**Redacción Proyecto Final Android Studio**

Chilapa De Álvarez Guerrero, diciembre 2023



## **MANIFEST**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
 <uses-permission android:name="android.permission.INTERNET" />  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Proyecto\_Final"  
 tools:targetApi="31">  
 <activity  
 android:name=".ViewUrlsActivity"  
 android:label="@string/app\_name">  
 </activity>  
  
 <activity  
 android:name=".PdfsActivity"  
 android:exported="false" />  
 <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>

## **Main.Java** nombre de usuario "admin" y una contraseña "1234".

package com.example.proyecto\_final;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity {  
  
 private EditText editTextUsername;  
 private EditText editTextPassword;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 // Inicializar los EditText  
 editTextUsername = findViewById(R.id.*editTextUsername*);  
 editTextPassword = findViewById(R.id.*editTextPassword*);  
 }  
  
 // Método para el botón de inicio de sesión  
 public void login(View view) {  
 String username = editTextUsername.getText().toString();  
 String password = editTextPassword.getText().toString();  
   
 if (isValidCredentials(username, password)) {  
 Intent siguiente = new Intent(this, PdfsActivity.class);  
 startActivity(siguiente);  
 } else {  
 // Credenciales inválidas, mostrar un mensaje de error  
 Toast.*makeText*(this, "Nombre de usuario o contraseña incorrectos", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 private boolean isValidCredentials(String username, String password) {  
   
  
 // Ejemplo: Usuario: admin, Contraseña: 1234  
 return username.equals("admin") && password.equals("1234");  
 }  
}

## 

## **ADAPTADOR**

package com.example.proyecto\_final;  
  
import android.content.Context;  
import android.content.Intent;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.util.List;  
  
public class Adaptador extends RecyclerView.Adapter<Adaptador.ViewHolder> {  
  
 List<String> titles;  
 List<Integer> images;  
 List<String> urls;  
 LayoutInflater inflater;  
  
 public Adaptador(Context ctx, List<String> titles, List<Integer> images, List<String> urls) {  
 this.titles = titles;  
 this.images = images;  
 this.inflater = LayoutInflater.*from*(ctx);  
 this.urls = urls;  
 }  
  
 @NonNull  
 @Override  
 public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = inflater.inflate(R.layout.*custom*, parent, false);  
 return new ViewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull ViewHolder holder, int position) {  
 holder.title.setText(titles.get(position));  
 holder.gridIcon.setImageResource(images.get(position));  
  
 // Agrega un OnClickListener para manejar el clic en el elemento  
 holder.itemView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 // Obtiene la posición del adaptador utilizando holder.getAdapterPosition()  
 int adapterPosition = holder.getAdapterPosition();  
  
 // Verifica si la posición es válida  
 if (adapterPosition != RecyclerView.*NO\_POSITION*) {  
 // Obtiene la URL correspondiente a la posición actual  
 String url = getUrlForPosition(adapterPosition);  
  
 // Abre el nuevo activity con WebView  
 Intent intent = new Intent(view.getContext(), ViewUrlsActivity.class);  
 intent.putExtra("url", url);  
 view.getContext().startActivity(intent);  
 }  
 }  
 });  
 }  
  
 // Método para obtener la URL según la posición  
 private String getUrlForPosition(int position) {  
 // Verifica si la posición está dentro del rango de la lista de imágenes  
 if (position >= 0 && position < urls.size()) {  
 // Obtiene la URL correspondiente a la posición actual  
 String url = urls.get(position);  
  
 // Imprime la URL en el registro de la consola para verificar  
 Log.*d*("Adapter", "URL: " + url);  
  
 return url;  
 } else {  
 // Devuelve una URL predeterminada o maneja la situación según tu lógica  
 return "https://google.com";  
 }  
 }  
  
 @Override  
 public int getItemCount() {  
 return titles.size();  
 }  
  
 public class ViewHolder extends RecyclerView.ViewHolder {  
 TextView title;  
 ImageView gridIcon;  
  
 public ViewHolder(@NonNull View itemView) {  
 super(itemView);  
 title = itemView.findViewById(R.id.*textView2*);  
 gridIcon = itemView.findViewById(R.id.*imageView2*);  
  
 itemView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Toast.*makeText*(v.getContext(), "Clicked -> " + getAdapterPosition(), Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
 }  
}

## **PdfsActivity.Java**

package com.example.proyecto\_final;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.GridLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
import android.content.Intent;  
import android.content.res.Configuration;  
import android.os.Bundle;  
import java.util.ArrayList;  
import java.util.List;  
import android.view.View;  
  
public class PdfsActivity extends AppCompatActivity {  
 RecyclerView dataList;  
 List<String> titles;  
 List<Integer> images;  
 List<String> urls;  
 Adapter adapter;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_pdfs*);  
  
 dataList = findViewById(R.id.*dataList*);  
 titles = new ArrayList<>();  
 images = new ArrayList<>();  
 urls = new ArrayList<>();  
 //NOMBRE DE LOS LIBROS   
 titles.add("Eloquent\_JavaScript ");  
 titles.add("HTML Y CSS3");  
 titles.add("manual-de-html-css");  
 titles.add("Python Para todos");  
 titles.add("Manual de Php");  
 titles.add("Pro Git");  
 titles.add("Introduccion a TypeScript");  
 titles.add("Aprendizaje Jquery");  
 titles.add("Algoritmos y Programacion");  
 titles.add("Problemario de Algoritmos de Flujo y Pseudocodigo");  
 titles.add("Fundamentos de Programación");  
 titles.add("Problemas y Algoritmos");  
 titles.add("Estructura de Datos en JavaScript");  
 titles.add("Html y css Front");  
 titles.add("El gran libro de HTML5, CSS3 y Javascript ");  
 titles.add("Diseño Web con Html y css");  
 titles.add("Hackin the art of exploitation");  
 titles.add("The Web Application Hacker’s Handbook");  
 titles.add("Base de Datos");  
 titles.add("Analis y Diseño de Base de Datos");  
 titles.add("Fundamentos a SQL");  
 titles.add("Gestores de Base de Datos");  
 titles.add("Mtedologia Scrum Master");  
 titles.add("Manual Basico de Andriod Studio");  
 titles.add("Manual Programacion Andriod Studio");  
 titles.add("Desarrollo de apps para Android ");  
 titles.add("Manual Netbeans");  
 titles.add("Curso de Java");  
 titles.add("Php y Networking");  
 titles.add("Tutorial Modelo Vista Controlador");  
   
 //Direccion de las imagenes de cada una de las portadas  
 images.add(R.drawable.*libro1*);  
 images.add(R.drawable.*libro11*);  
 images.add(R.drawable.*libro3*);  
 images.add(R.drawable.*libro4*);  
 images.add(R.drawable.*libro5*);  
 images.add(R.drawable.*libro6*);  
 images.add(R.drawable.*libro7*);  
 images.add(R.drawable.*libro8*);  
 images.add(R.drawable.*libro9*);  
 images.add(R.drawable.*libro10*);  
 images.add(R.drawable.*libro11*);  
 images.add(R.drawable.*libro12*);  
 images.add(R.drawable.*libro13*);  
 images.add(R.drawable.*libro14*);  
 images.add(R.drawable.*libro15*);  
 images.add(R.drawable.*libro16*);  
 images.add(R.drawable.*libro17*);  
 images.add(R.drawable.*libro18*);  
 images.add(R.drawable.*libro19*);  
 images.add(R.drawable.*libro20*);  
 images.add(R.drawable.*libro21*);  
 images.add(R.drawable.*libro22*);  
 images.add(R.drawable.*libro23*);  
 images.add(R.drawable.*libro24*);  
 images.add(R.drawable.*libro25*);  
 images.add(R.drawable.*libro26*);  
 images.add(R.drawable.*libro27*);  
 images.add(R.drawable.*libro28*);  
 images.add(R.drawable.*libro29*);  
 images.add(R.drawable.*libro30*);  
  
// Direccion de las urls de los libros  
 urls.add("https://eloquentjs-es.thedojo.mx/Eloquent\_JavaScript.pdf");  
 urls.add("https://gutl.jovenclub.cu/wp-content/uploads/2013/10/El+gran+libro+de+HTML5+CSS3+y+Javascrip.pdf");  
 urls.add("https://aapp.files.wordpress.com/2010/07/manual-de-html-css.pdf");  
 urls.add("https://persoal.citius.usc.es/eva.cernadas/informaticaparacientificos/material/libros/Python%20para%20todos.pdf");  
 urls.add("https://waltercarnero.com/cfp/tpprgweb/Manual\_Basico\_PHP.pdf");  
 urls.add("https://progit2.s3.amazonaws.com/es/2015-03-10-11982/progit-es.388.pdf");  
 urls.add("https://blog.educalix.com/wp-content/uploads/2023/03/Manual-TypeScript.pdf");  
 urls.add("https://riptutorial.com/Download/jquery-es.pdf");  
 urls.add("https://digitk.areandina.edu.co/bitstream/handle/areandina/1281/Algoritmos%20y%20programacio%CC%81n.pdf?sequence=1");  
 urls.add("https://editorial.uaa.mx/docs/algoritmos.pdf");  
 urls.add("https://elhacker.info/manuales/Lenguajes%20de%20Programacion/Fundamentos\_de\_programaci%C3%B3n\_4ta\_Edici%C3%B3n\_Luis\_Joyanes\_Aguilar\_2.pdf");  
 urls.add("https://editorial.unrn.edu.ar/media/data/lecturas/problemas\_algoritmos\_eunrn.pdf");  
 urls.add("libro de estructura de datos y algoritmos pdf");  
 urls.add("https://www.autentia.com/wp-content/uploads/libros/Front\_HTMLyCSS-Autentia.pdf");  
 urls.add("https://gutl.jovenclub.cu/wp-content/uploads/2013/10/El+gran+libro+de+HTML5+CSS3+y+Javascrip.pdf");  
 urls.add("https://elhacker.info/manuales/Desarrollo%20web/Dise%C3%B1o%20Web%20con%20HTML%20&%20CSS,%20USERS%20-%20Gustavo%20Carballeiro.PDF");  
 urls.add("https://edu.anarcho-copy.org/Against%20Security%20-%20Self%20Security/Dafydd%20Stuttard,%20Marcus%20Pinto%20-%20The%20web%20application%20hacker's%20handbook\_%20finding%20and%20exploiting%20security%20flaws-Wiley%20(2011).pdf");  
 urls.add("https://bdigital.uvhm.edu.mx/wp-content/uploads/2020/05/Bases-de-Datos.pdf");  
 urls.add("https://munayi.uleam.edu.ec/wp-content/uploads/2022/07/Ana%CC%81lisis-y-disen%CC%83o-de-base-de-datos.pdf");  
 urls.add("https://pedrobeltrancanessa-biblioteca.weebly.com/uploads/1/2/4/0/12405072/fundamentos\_de\_sql\_3edi\_oppel.pdf");  
 urls.add("https://www.mheducation.es/bcv/guide/capitulo/8448148797.pdf");  
 urls.add("https://www.scrummanager.com/files/scrum\_master.pdf");  
 urls.add("https://cursoslared.com/recursoslibre/TutorialAndroidPrincipiantes.pdf");  
 urls.add("https://aluzardo.github.io/trabajo-fin-de-grado/Tutoriales/Manual%20Programacion%20Android.pdf");  
 urls.add("https://www.uma.es/media/tinyimages/file/android\_ed2.pdf");  
 urls.add("https://www.unibarranquilla.edu.co/docs/33-L-de-Avila-Manual-de-NETBEANS.pdf");  
 urls.add("https://www.cursodejava.com.mx/descargas/CursoJava.pdf");  
 urls.add("http://www.upd.edu.mx/PDF/Libros/PhpNetwork.pdf");  
 urls.add("https://www.codigonexo.com/wp-content/uploads/2014/06/Curso-completo-MVC.pdf");  
  
 adapter = new Adapter(this, titles, images, urls);  
  
 int orientation = getResources().getConfiguration().orientation;  
  
 int spanCount;  
 if (orientation == Configuration.*ORIENTATION\_LANDSCAPE*) {  
 spanCount = 3;  
 } else {  
 spanCount = 2;  
 }  
  
 GridLayoutManager gridLayoutManager = new GridLayoutManager(this, spanCount, GridLayoutManager.*VERTICAL*, false);  
 dataList.setLayoutManager(gridLayoutManager);  
 dataList.setAdapter(adapter);  
 }  
  
}

## **ViewUrlsActivity.java**

package com.example.proyecto\_final;  
  
import android.content.ActivityNotFoundException;  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;  
import android.webkit.WebView;  
import android.webkit.WebViewClient;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class ViewUrlsActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_viewurls*);  
  
 Intent intent = getIntent();  
 if (intent != null && intent.hasExtra("url")) {  
 String url = intent.getStringExtra("url");  
  
 if (isPDFUrl(url)) {  
 // Si la URL es un enlace a un PDF, abrirlo con una intención del sistema  
 openPDFUsingIntent(url);  
 } else {  
 // Si no es un enlace a un PDF, cargar la URL en WebView  
 loadUrlInWebView(url);  
 }  
 }  
 }  
  
 private boolean isPDFUrl(String url) {  
 // Agrega lógica para determinar si la URL es un enlace a un PDF  
 return url.toLowerCase().endsWith(".pdf");  
 }  
  
 private void openPDFUsingIntent(String url) {  
 Intent intent = new Intent(Intent.*ACTION\_VIEW*);  
 intent.setDataAndType(Uri.*parse*(url), "application/pdf");  
 intent.setFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP*);  
  
 try {  
 startActivity(intent);  
 } catch (ActivityNotFoundException e) {  
 Toast.*makeText*(this, "No se encontró una aplicación para abrir archivos PDF", Toast.*LENGTH\_SHORT*).show();  
 }  
 finish(); // Cierra la actividad actual después de iniciar la intención  
 }  
  
 private void loadUrlInWebView(String url) {  
 WebView webView = findViewById(R.id.*webView*);  
  
 // Agrega un WebViewClient para manejar la carga de la página y los errores  
 webView.setWebViewClient(new WebViewClient() {  
 });  
  
 // Habilita JavaScript para páginas que lo requieran  
 webView.getSettings().setJavaScriptEnabled(true);  
  
 // Carga la URL en el WebView  
 webView.loadUrl(url);  
 }  
}

## **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="@android:color/background\_light"  
 tools:context=".MainActivity">  
  
 <ImageView  
 android:id="@+id/imageViewLogo"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="200dp"  
 android:src="@drawable/logo"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <com.airbnb.lottie.LottieAnimationView  
 android:id="@+id/animationView"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 app:layout\_constraintDimensionRatio="16:9"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageViewLogo"  
 app:layout\_constraintBottom\_toTopOf="@+id/guideline" />  
  
 <androidx.constraintlayout.widget.Guideline  
 android:id="@+id/guideline"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 app:layout\_constraintGuide\_percent="0.5" />  
  
 <EditText  
 android:id="@+id/editTextUsername"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="4dp"  
 android:hint="Nombre de Usuario"  
 android:padding="12dp"  
 android:textColor="@android:color/black"  
 android:textSize="16sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/guideline" />  
  
 <EditText  
 android:id="@+id/editTextPassword"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:hint="Contraseña"  
 android:inputType="textPassword"  
 android:padding="12dp"  
 android:textColor="@android:color/black"  
 android:textSize="16sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editTextUsername" />  
  
 <Button  
 android:id="@+id/buttonLogin"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:backgroundTint="@color/design\_default\_color\_primary"  
 android:onClick="login"  
 android:text="Iniciar Sesión"  
 android:textAppearance="@style/TextAppearance.AppCompat.Large"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editTextPassword" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

## **activity\_pdfs.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=".PdfsActivity">  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/dataList"  
 android:layout\_width="403dp"  
 android:layout\_height="664dp"  
 android:layout\_marginTop="48dp"  
 app:layout\_constraintTop\_toTopOf="parent"  
 tools:layout\_editor\_absoluteX="4dp" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

## **activity\_viewsurls.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout 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=".ViewUrlsActivity">  
  
 <WebView  
 android:id="@+id/webView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
</RelativeLayout>

## **Custom.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="wrap\_content"  
 android:orientation="vertical"  
 android:padding="10dp">  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 app:cardElevation="10dp"  
 app:contentPadding="10dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.6">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <ImageView  
 android:id="@+id/imageView2"  
 android:layout\_width="150dp"  
 android:layout\_height="100dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:textAlignment="center"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@mipmap/ic\_launcher\_round" />  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:textSize="15sp"  
 android:layout\_marginBottom="50dp"  
 android:textStyle="italic"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView2" />  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
  
 </androidx.cardview.widget.CardView>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

## **COLORS.XML**

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="black">#FF000000</color>  
 <color name="white">#FFFFFFFF</color>  
</resources>

## **STRINGS.XML**

<resources>  
 <string name="app\_name">Proyecto\_Final</string>  
  
</resources>

## **Main.Java—Explicación**

Dentro de este archivo esta toda la estructura de la aplicación dentro de él se encuentra un formulario de inicio de sesión básico, es decir funciona cuando el usuario ingresa un nombre de usuario y una contraseña, la aplicación verifica sí coinciden con un conjunto de credenciales predefinido.

Es decir, si son válidas, se inicia una nueva actividad que presumiblemente muestra archivos PDF (como indicado por PdfsActivity), de lo contrario, se muestra un mensaje de error.

* Inicialización de variables:
* Se declaran dos variables privadas editTextUsername y editTextPassword para representar los campos de entrada de texto en la interfaz de usuario.
* onCreate method:
* Este método se llama cuando la actividad se crea por primera vez.
* Establece el diseño de la actividad usando setContentView(R.layout.activity\_main), donde activity\_main es el archivo XML que define la interfaz de usuario para esta actividad.
* Inicializa los EditText a través de findViewById para referenciar los campos de entrada en el diseño.
* login method, este método se llama cuando se presiona el botón de inicio de sesión en la interfaz de usuario (posiblemente configurado en el archivo XML con android:onClick="login").
* Recupera el texto ingresado en los campos de usuario y contraseña.
* Llama a la función isValidCredentials con el nombre de usuario y la contraseña ingresados.
* Si las credenciales son válidas, se crea un nuevo Intento para abrir la actividad PdfsActivity y se inicia esa actividad.
* Si las credenciales no son válidas, se muestra un mensaje de error utilizando Toast.
* isValidCredentials method:
* Este método toma un nombre de usuario y una contraseña como parámetros y verifica si coinciden con un conjunto de credenciales predefinido.
* En este caso, las credenciales válidas son un nombre de usuario "admin" y una contraseña "1234".
* Retorna true si las credenciales son válidas, y false en caso contrario.

## **ADPTADOR**

Dentro del adaptador se utiliza para vincular datos a un RecyclerView que muestra elementos con un título, una imagen y una URL asociada, es decir Cuando se hace clic en un elemento, se inicia una nueva actividad (ViewUrlsActivity) para mostrar la URL correspondiente en un WebView.

* Asi como también Implementa la lógica para descargar el PDF desde la URL.
* Puedes hacer esto en un AsyncTask o mediante el uso de bibliotecas de descarga como Retrofit.
* Esta es una implementación básica para mostrar un PDF utilizando la biblioteca AndroidPdfViewer. Asegúrate de manejar la descarga del PDF según tus requisitos específicos. Ten en cuenta que este ejemplo utiliza la versión 3.2.0-beta.1 de la biblioteca; puedes verificar si hay versiones más recientes en el repositorio de GitHub del proyecto
* El constructor toma tres listas (titles, images, y urls) y un contexto (ctx) como parámetros.
* Inicializa las variables de instancia con los valores proporcionados.
* onCreateViewHolder:
* Crea y devuelve una nueva instancia de ViewHolder inflando el diseño definido en R.layout.custom.
* onBindViewHolder:
* Se llama cuando se vincula un conjunto de datos a una vista específica en el RecyclerView.
* Establece el texto y la imagen de la vista según la posición del elemento en las listas proporcionadas.
* Agrega un OnClickListener a cada elemento del RecyclerView. Cuando se hace clic en un elemento, se obtiene la URL correspondiente a la posición y se inicia una nueva actividad (ViewUrlsActivity) pasando esa URL como un extra en el Intent.
* getUrlForPosition:
* Un método auxiliar que toma una posición como parámetro y devuelve la URL correspondiente de la lista de URLs. También imprime la URL en el registro de la consola para propósitos de depuración.
* getItemCount:
* Retorna el número total de elementos en el conjunto de datos. En este caso, es la longitud de la lista de títulos.
* ViewHolder Clase Interna:
* Representa un elemento individual en el RecyclerView.

**PdfsActivity.Java**

Dentro de este archivo contiene funciones para mostrar una lista de libros en un formato de cuadrícula (RecyclerView). Cada elemento de la lista tiene un título, una imagen de portada y una URL asociada. Cuando se hace clic en un libro, se inicia una nueva actividad para mostrar la URL del libro en un WebView o una vista similar.

* Se inicializan las variables necesarias, incluyendo dataList que representa el RecyclerView, y listas (titles, images, y urls) que contienen información sobre los libros, como títulos, imágenes de portada y URLs.
* Llenado de listas:
* Se agregan títulos, imágenes de portada y URLs de libros a las listas correspondientes.
* Creación del adaptador:
* Se crea una instancia de la clase Adaptador, pasando las listas de títulos, imágenes y URLs.
* Configuración del RecyclerView:
* Se determina la orientación del dispositivo (vertical u horizontal) y se ajusta el número de columnas en el RecyclerView en consecuencia.
* Se crea un GridLayoutManager con el número de columnas calculado y se establece en el RecyclerView.
* Se establece el adaptador en el RecyclerView.
* Acción del clic en el elemento del RecyclerView:
* Cuando se hace clic en un elemento del RecyclerView, se obtiene la URL correspondiente al elemento seleccionado y se inicia una nueva actividad (ViewUrlsActivity) para mostrar esa URL.

## **ViewUrls**

Este archivo contiene este código proporciona una manera de manejar diferentes tipos de enlaces (PDF o páginas web) en una aplicación Android, abriendo el enlace en un WebView o una aplicación externa según el tipo de enlace.

* Obtención de la URL desde la intención (onCreate):
* En el método onCreate, la actividad recibe la intención que la inició y extrae la URL del extra "url" si existe.
* Verificación del tipo de URL (isPDFUrl):
* El método isPDFUrl verifica si la URL es un enlace a un archivo PDF. En este caso, simplemente compara si la URL termina con ".pdf" (es insensible a mayúsculas y minúsculas).
* Manejo de PDFs (openPDFUsingIntent):
* Si la URL es un enlace a un archivo PDF, se utiliza la intención del sistema para abrir el PDF. Se crea una nueva Intent con la acción ACTION\_VIEW y se establece el tipo de datos como "application/pdf". La URL se pasa como datos a esta intención y se inicia. Si no hay una aplicación disponible para manejar archivos PDF, se muestra un mensaje de tostada.
* Carga de URL en WebView (loadUrlInWebView):
* Si la URL no es un enlace a un PDF, se carga en un WebView. Se inicializa un WebView, se habilita JavaScript si es necesario y se carga la URL en el WebView. Se utiliza un WebViewClient para manejar la carga de la página y los errores.
* WebViewClient:
* En el código proporcionado, el WebViewClient está vacío, pero normalmente se usaría para personalizar el comportamiento del WebView, como manejar redirecciones, interceptar enlaces o mostrar una barra de progreso.
* Cierre de la actividad (finish):
* Después de abrir el PDF o cargar la URL en el WebView, la actividad se cierra utilizando finish().

## **Main.xml**

Este diseño utiliza restricciones (app:layout\_constraintStart\_toStartOf, app:layout\_constraintEnd\_toEndOf, app:layout\_constraintTop\_toBottomOf, etc.) para definir la posición relativa de los elementos en el diseño. La Guideline se utiliza para dividir verticalmente el diseño en dos partes iguales. Además, se especifica un logotipo (@drawable/logo) y se utiliza una animación Lottie en animationView.

* El diseño principal de la interfaz está contenido en un ConstraintLayout, que es un diseño flexible y poderoso para crear interfaces de usuario en Android.
* ImageView (imageViewLogo):
* Un ImageView que muestra un logotipo. Se utiliza app:layout\_constraintTop\_toTopOf="parent" para anclar la parte superior del ImageView a la parte superior del ConstraintLayout.
* LottieAnimationView (animationView):
* Guideline (guideline):
* Una Guideline que actúa como una línea guía en el diseño. Se posiciona verticalmente en el centro (app:layout\_constraintGuide\_percent="0.5") del ConstraintLayout.
* EditText (editTextUsername y editTextPassword):
* Dos EditText para ingresar el nombre de usuario y la contraseña. Ambos están configurados con app:layout\_constraintTop\_toBottomOf="@+id/guideline", lo que significa que se colocan debajo de la guideline.
* Button (buttonLogin)

## **Activity.pdfs**

Dentro de este archivo ConstraintLayout es el diseño indica que la actividad PdfsActivity mostrará una lista de elementos utilizando un RecyclerView. El contenido específico y el diseño de los elementos en la lista estarán determinados por el adaptador que se establecerá en el RecyclerView mediante programación en la actividad correspondiente (PdfsActivity).

* El diseño principal de la interfaz está contenido en un ConstraintLayout, que es un diseño flexible y poderoso para crear interfaces de usuario en Android.
* RecyclerView (dataList):
* Un RecyclerView que se utiliza para mostrar una lista de elementos en formato de cuadrícula o lista. Se le asigna el ID dataList y se especifica una altura y ancho específicos (403dp x 664dp). Además, se establece un margen superior (48dp) y se ancla en la parte superior del ConstraintLayout.

## **ViewsUrls**

Este diseño indica que la actividad PdfsActivity mostrará una lista de elementos utilizando un RecyclerView. El contenido específico y el diseño de los elementos en la lista estarán determinados por el adaptador que se establecerá en el RecyclerView mediante programación en la actividad correspondiente (PdfsActivity).

* Este es el diseño principal de la interfaz está contenido en un ConstraintLayout, que es un diseño flexible y poderoso para crear interfaces de usuario en Android.
* RecyclerView (dataList):
* Un RecyclerView que se utiliza para mostrar una lista de elementos en formato de cuadrícula o lista.

## **CUSTOM**

Dentro de este diseño es utilizado como un elemento de lista en un RecyclerView, y el contenido específico (imagen y texto) se proporcionará dinámicamente utilizando un adaptador, es decir cada elemento de la lista se mostrará con un contorno elevado gracias al uso de CardView.

* El diseño principal está contenido en un ConstraintLayout. Este es el contenedor principal que contiene los elementos de la tarjeta.
* CardView:
* Se utiliza un CardView para proporcionar un contorno elevado a la vista del elemento de lista. Se especifican márgenes y elevación (cardElevation) para darle un aspecto visual atractivo.
* Otro ConstraintLayout dentro del CardView:
* Hay otro ConstraintLayout anidado dentro del CardView. Esto puede contener varios elementos secundarios y se utiliza para organizar la imagen (ImageView) y el texto (TextView) dentro de la tarjeta.
* ImageView (imageView2):
* Un ImageView que muestra una imagen. En este caso, se establece un recurso fuente (@mipmap/ic\_launcher\_round). Puedes cambiar esto a la imagen específica que desees mostrar en cada elemento de la lista.
* TextView (textView2):
* Un TextView que muestra algún texto. La propiedad app:layout\_constraintTop\_toBottomOf="@+id/imageView2" coloca este TextView debajo de la ImageView. Puedes cambiar el texto y el estilo según sea necesario.

## **RESULTADOS FINALES**

