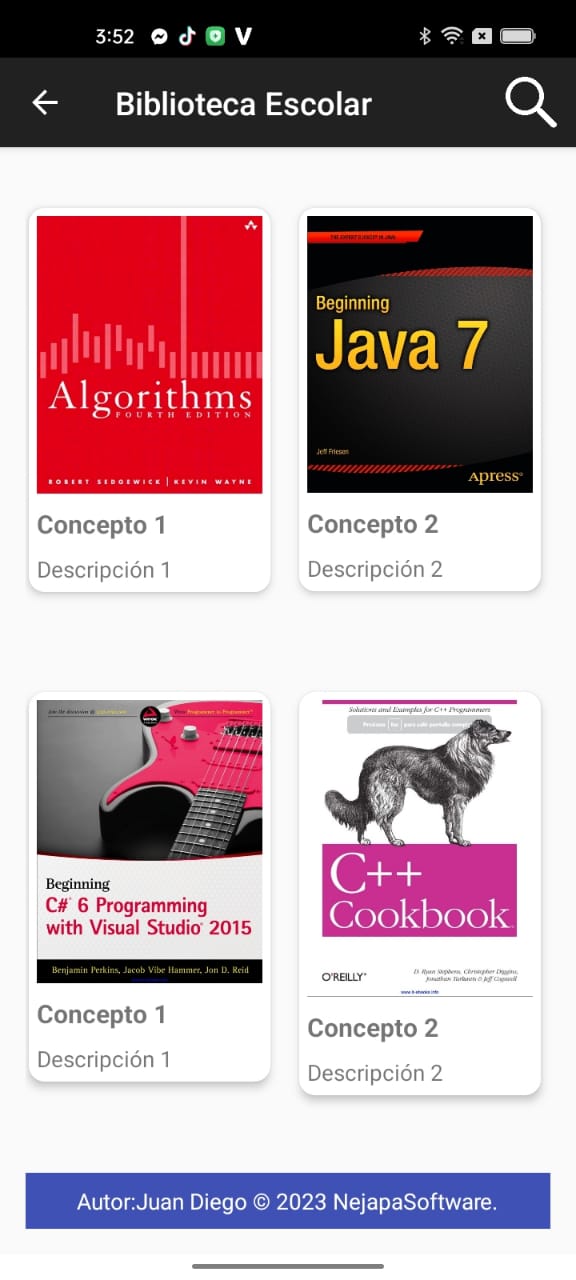
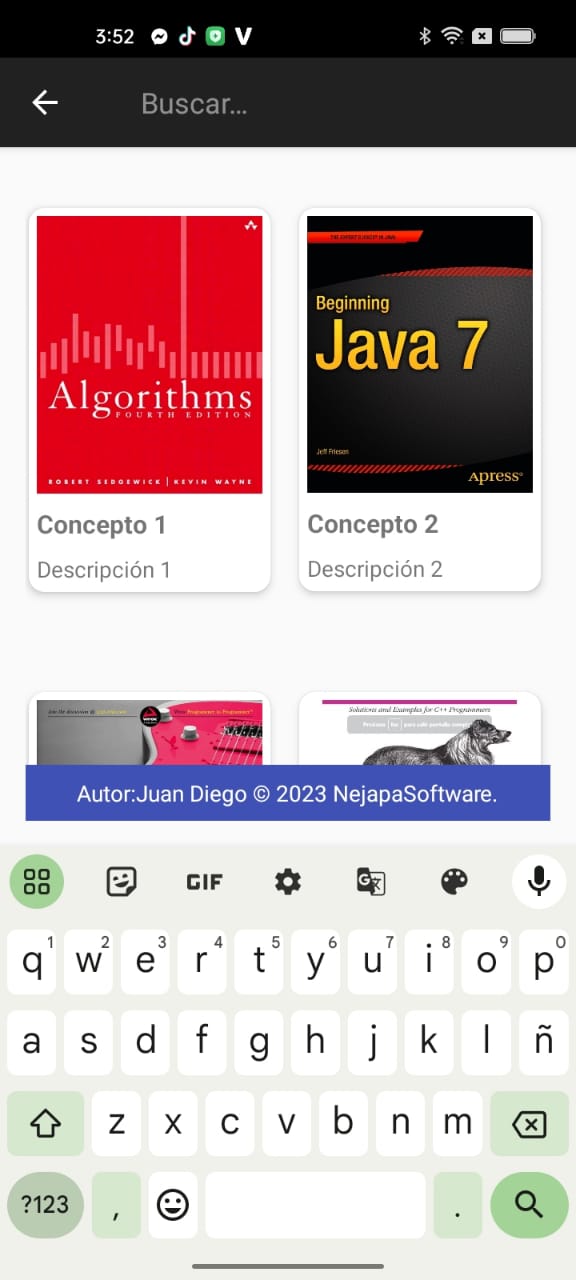
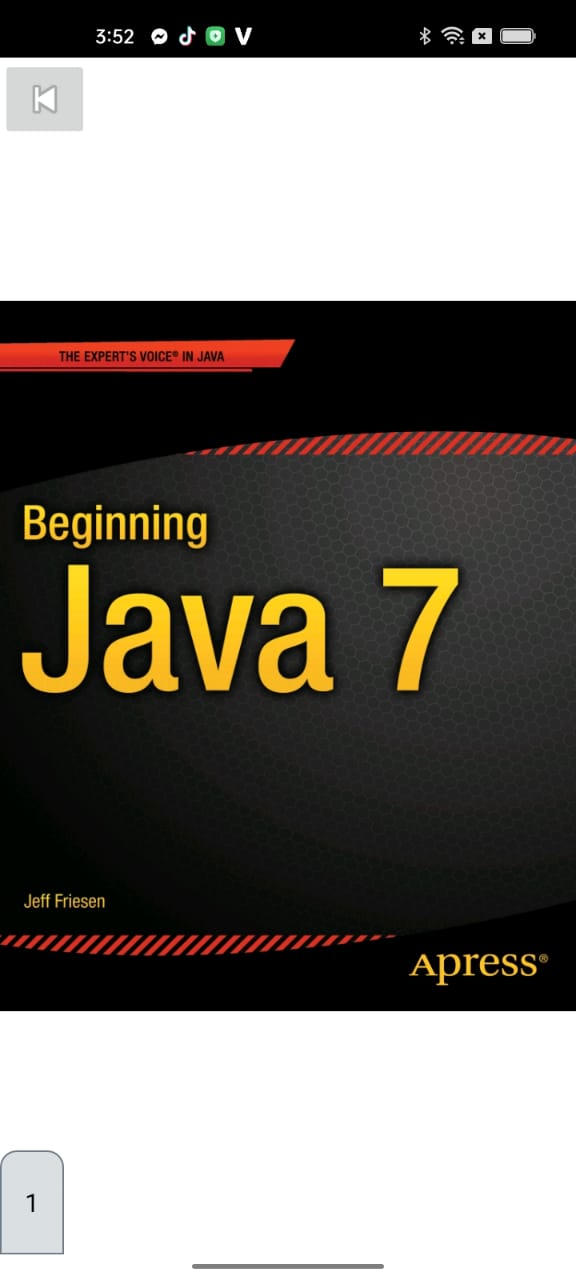
**Documentación de la aplicación ”Biblioteca Alumnos”**

**usuario:***admin*

**contraseña:***admin*

**Ejecución final**



**MainActivity.java**

**package com.example.BibliotecaAlumnos;**

**import android.content.Intent;**

**import android.os.Bundle;**

**import android.widget.Button;**

**import android.widget.EditText;**

**import android.widget.Toast;**

**import androidx.annotation.Nullable;**

**import androidx.appcompat.app.AppCompatActivity;**

**import com.example.BibliotecaAlumnos.R;**

**public class MainActivity extends AppCompatActivity {**

**private static final String *USUARIO\_CORRECTO* = "admin";**

**private static final String *CONTRASEÑA\_CORRECTA* = "admin";**

**private EditText editTextUsuario;**

**private EditText editTextContraseña;**

**@Override**

**protected void onCreate(@Nullable Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.*activity\_main*);**

**// Oculta la barra de acción en MainActivity**

**if (getSupportActionBar() != null) {**

**getSupportActionBar().hide();**

**}**

**editTextUsuario = findViewById(R.id.*editTextUsuario*);**

**editTextContraseña = findViewById(R.id.*editTextContraseña*);**

**Button buttonIniciarSesion = findViewById(R.id.*buttonIniciarSesion*);**

**buttonIniciarSesion.setOnClickListener(view -> {**

**iniciarSesion();**

**});**

**}**

**private void iniciarSesion() {**

**String usuario = editTextUsuario.getText().toString();**

**String contraseña = editTextContraseña.getText().toString();**

**// Verificar las credenciales**

**if (usuario.equals(*USUARIO\_CORRECTO*) && contraseña.equals(*CONTRASEÑA\_CORRECTA*)) {**

**// Credenciales correctas, iniciar la siguiente actividad**

**toCategoriaActivity();**

**} else {**

**// Credenciales incorrectas, mostrar un mensaje de error**

**Toast.*makeText*(this, "Usuario o contraseña incorrectos", Toast.*LENGTH\_SHORT*).show();**

**}**

**}**

**private void toCategoriaActivity() {**

**Intent intent = new Intent(MainActivity.this, ViewInicio.class);**

**startActivity(intent);**

**finish();**

**}**

**}**

**ViewInicio.java**

**package com.example.BibliotecaAlumnos;**

**import android.content.Intent;**

**import android.content.res.Configuration;**

**import android.os.Bundle;**

**import android.view.LayoutInflater;**

**import android.view.Menu;**

**import android.view.MenuItem;**

**import android.view.View;**

**import android.view.ViewGroup;**

**import android.widget.ImageView;**

**import android.widget.TextView;**

**import androidx.annotation.NonNull;**

**import androidx.appcompat.app.ActionBar;**

**import androidx.appcompat.app.AppCompatActivity;**

**import androidx.appcompat.widget.SearchView;**

**import androidx.recyclerview.widget.GridLayoutManager;**

**import androidx.recyclerview.widget.RecyclerView;**

**import com.example.BibliotecaAlumnos.R;**

**import java.util.ArrayList;**

**import java.util.List;**

**public class ViewInicio extends AppCompatActivity {**

**private RecyclerView recyclerView;**

**private TarjetaAdapter tarjetaAdapter;**

**private List<Tarjeta> tarjetas; // Lista original**

**private List<Tarjeta> tarjetasFiltradas; // Lista filtrada**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.*view\_inicio*);**

**recyclerView = findViewById(R.id.*recyclerView*);**

**int spanCount = getResources().getConfiguration().orientation == Configuration.*ORIENTATION\_LANDSCAPE* ? 3 : 2;**

**recyclerView.setLayoutManager(new GridLayoutManager(this, spanCount));**

**// Crear lista original**

**tarjetas = new ArrayList<>();**

**tarjetas.add(new Tarjeta(R.drawable.*libro1*, "Concepto 1", "Descripción 1", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro2*, "Concepto 2", "Descripción 2", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro3*, "Concepto 1", "Descripción 1", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro4*, "Concepto 2", "Descripción 2", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro5*, "Concepto 1", "Descripción 1", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro6*, "Concepto 2", "Descripción 2", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro7*, "Concepto 1", "Descripción 1", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro8*, "Concepto 2", "Descripción 2", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro9*, "Concepto 1", "Descripción 1", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro10*, "Concepto 2", "Descripción 2", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro11*, "Concepto 1", "Descripción 1", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro12*, "Concepto 2", "Descripción 2", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro13*, "Concepto 1", "Descripción 1", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro14*, "Concepto 2", "Descripción 2", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro15*, "Concepto 1", "Descripción 1", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro16*, "Concepto 2", "Descripción 2", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro17*, "Concepto 1", "Descripción 1", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro18*, "Concepto 2", "Descripción 2", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro19*, "Concepto 1", "Descripción 1", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro20*, "Concepto 2", "Descripción 2", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro21*, "Concepto 1", "Descripción 1", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro22*, "Concepto 2", "Descripción 2", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro23*, "Concepto 1", "Descripción 1", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro24*, "Concepto 2", "Descripción 2", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro25*, "Concepto 1", "Descripción 1", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro26*, "Concepto 2", "Descripción 2", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro27*, "Concepto 1", "Descripción 1", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro28*, "Concepto 2", "Descripción 2", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro29*, "Concepto 1", "Descripción 1", 350, 600, 5));**

**tarjetas.add(new Tarjeta(R.drawable.*libro30*, "Concepto 2", "Descripción 2", 350, 600, 5));**

**// Inicializar lista filtrada con la lista original**

**tarjetasFiltradas = new ArrayList<>(tarjetas);**

**tarjetaAdapter = new TarjetaAdapter(tarjetasFiltradas);**

**recyclerView.setAdapter(tarjetaAdapter);**

**ActionBar actionBar = getSupportActionBar();**

**// Habilita el ActionBar y muestra el título**

**if (actionBar != null) {**

**actionBar.setDisplayHomeAsUpEnabled(true);**

**actionBar.setTitle("Biblioteca Escolar");**

**actionBar.setDisplayShowHomeEnabled(true);**

**}**

**}**

**@Override**

**public boolean onCreateOptionsMenu(Menu menu) {**

**getMenuInflater().inflate(R.menu.*menu\_search*, menu);**

**// Obtén la referencia al SearchView**

**MenuItem searchItem = menu.findItem(R.id.*action\_search*);**

**SearchView searchView = (SearchView) searchItem.getActionView();**

**// Configura el listener para el cambio en el texto de búsqueda**

**searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {**

**@Override**

**public boolean onQueryTextSubmit(String query) {**

**// Aquí puedes realizar acciones cuando se envía la consulta**

**return true;**

**}**

**@Override**

**public boolean onQueryTextChange(String newText) {**

**// Filtra los datos en el adaptador según el texto de búsqueda**

**filterList(newText);**

**return true;**

**}**

**});**

**return super.onCreateOptionsMenu(menu);**

**}**

**private void filterList(String query) {**

**// Limpiar la lista filtrada**

**tarjetasFiltradas.clear();**

**// Si el texto de búsqueda está vacío, mostrar todos los elementos**

**if (query.isEmpty()) {**

**tarjetasFiltradas.addAll(tarjetas);**

**} else {**

**// Filtrar la lista original en función del texto de búsqueda**

**for (Tarjeta tarjeta : tarjetas) {**

**if (tarjeta.getConcepto().toLowerCase().contains(query.toLowerCase()) ||**

**tarjeta.getDescripcion().toLowerCase().contains(query.toLowerCase())) {**

**tarjetasFiltradas.add(tarjeta);**

**}**

**}**

**}**

**// Notificar al adaptador que los datos han cambiado**

**tarjetaAdapter.notifyDataSetChanged();**

**}**

**@Override**

**public boolean onOptionsItemSelected(MenuItem item) {**

**switch (item.getItemId()) {**

**case android.R.id.*home*:**

**// Manejar el clic en la flecha de retroceso**

**finish(); // Cierra la actividad actual**

**return true;**

**default:**

**return super.onOptionsItemSelected(item);**

**}**

**}**

**public static class Tarjeta {**

**private int imagen;**

**private String concepto;**

**private String descripcion;**

**private int tarjetaWidth;**

**private int tarjetaHeight;**

**private int espacioEntreTarjetas;**

**public Tarjeta(int imagen, String concepto, String descripcion, int tarjetaWidth, int tarjetaHeight, int espacioEntreTarjetas) {**

**this.imagen = imagen;**

**this.concepto = concepto;**

**this.descripcion = descripcion;**

**this.tarjetaWidth = tarjetaWidth;**

**this.tarjetaHeight = tarjetaHeight;**

**this.espacioEntreTarjetas = espacioEntreTarjetas;**

**}**

**public int getImagen() {**

**return imagen;**

**}**

**public String getConcepto() {**

**return concepto;**

**}**

**public String getDescripcion() {**

**return descripcion;**

**}**

**public int getTarjetaWidth() {**

**return tarjetaWidth;**

**}**

**public int getTarjetaHeight() {**

**return tarjetaHeight;**

**}**

**public int getEspacioEntreTarjetas() {**

**return espacioEntreTarjetas;**

**}**

**}**

**public class TarjetaAdapter extends RecyclerView.Adapter<TarjetaAdapter.TarjetaViewHolder> {**

**private List<Tarjeta> tarjetas;**

**public TarjetaAdapter(List<Tarjeta> tarjetas) {**

**this.tarjetas = tarjetas;**

**}**

**@NonNull**

**@Override**

**public TarjetaViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {**

**View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*card\_item\_tarjeta*, parent, false);**

**return new TarjetaViewHolder(view);**

**}**

**@Override**

**public void onBindViewHolder(TarjetaViewHolder holder, int position) {**

**Tarjeta tarjeta = tarjetas.get(position);**

**ViewGroup.MarginLayoutParams layoutParams = (ViewGroup.MarginLayoutParams) holder.itemView.getLayoutParams();**

**// Configura el tamaño de la tarjeta**

**layoutParams.width = tarjeta.getTarjetaWidth();**

**layoutParams.height = tarjeta.getTarjetaHeight();**

**// Configura el espacio entre tarjetas**

**layoutParams.setMargins(0, 0, 0, tarjeta.getEspacioEntreTarjetas());**

**holder.imageView.setImageResource(tarjeta.getImagen());**

**holder.textViewConcepto.setText(tarjeta.getConcepto());**

**holder.textViewDescripcion.setText(tarjeta.getDescripcion());**

**// Configura el clic en la tarjeta para abrir el PDF correspondiente**

**holder.itemView.setOnClickListener(view -> {**

**// Crea un Intent para abrir la actividad PdfViewerLibro**

**Intent intent = new Intent(view.getContext(), PdfViewerLibro.class);**

**// Pasa el ID del libro como extra**

**intent.putExtra("LibroId", position + 1); // Suma 1 porque la posición en la lista comienza desde 0**

**// Inicia la actividad PdfViewerLibro**

**view.getContext().startActivity(intent);**

**});**

**}**

**@Override**

**public int getItemCount() {**

**return tarjetas.size();**

**}**

**public class TarjetaViewHolder extends RecyclerView.ViewHolder {**

**ImageView imageView;**

**TextView textViewConcepto;**

**TextView textViewDescripcion;**

**public TarjetaViewHolder(View itemView) {**

**super(itemView);**

**imageView = itemView.findViewById(R.id.*imageView*);**

**textViewConcepto = itemView.findViewById(R.id.*textViewConcepto*);**

**textViewDescripcion = itemView.findViewById(R.id.*textViewDescripcion*);**

**}**

**}**

**}**

**}**

**LibrosActivity.java**

**package com.example.BibliotecaAlumnos;**

**import android.content.Intent;**

**import android.os.Bundle;**

**import android.widget.Button;**

**import android.widget.EditText;**

**import android.widget.Toast;**

**import androidx.annotation.Nullable;**

**import androidx.appcompat.app.AppCompatActivity;**

**import com.example.BibliotecaAlumnos.R;**

**public class MainActivity extends AppCompatActivity {**

**private static final String *USUARIO\_CORRECTO* = "admin";**

**private static final String *CONTRASEÑA\_CORRECTA* = "admin";**

**private EditText editTextUsuario;**

**private EditText editTextContraseña;**

**@Override**

**protected void onCreate(@Nullable Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.*activity\_main*);**

**// Oculta la barra de acción en MainActivity**

**if (getSupportActionBar() != null) {**

**getSupportActionBar().hide();**

**}**

**editTextUsuario = findViewById(R.id.*editTextUsuario*);**

**editTextContraseña = findViewById(R.id.*editTextContraseña*);**

**Button buttonIniciarSesion = findViewById(R.id.*buttonIniciarSesion*);**

**buttonIniciarSesion.setOnClickListener(view -> {**

**iniciarSesion();**

**});**

**}**

**private void iniciarSesion() {**

**String usuario = editTextUsuario.getText().toString();**

**String contraseña = editTextContraseña.getText().toString();**

**// Verificar las credenciales**

**if (usuario.equals(*USUARIO\_CORRECTO*) && contraseña.equals(*CONTRASEÑA\_CORRECTA*)) {**

**// Credenciales correctas, iniciar la siguiente actividad**

**toCategoriaActivity();**

**} else {**

**// Credenciales incorrectas, mostrar un mensaje de error**

**Toast.*makeText*(this, "Usuario o contraseña incorrectos", Toast.*LENGTH\_SHORT*).show();**

**}**

**}**

**private void toCategoriaActivity() {**

**Intent intent = new Intent(MainActivity.this, ViewInicio.class);**

**startActivity(intent);**

**finish();**

**}**

**}**

**PdfViwerLibro.java**

**package com.example.BibliotecaAlumnos;**

**import android.content.Intent;**

**import android.os.Bundle;**

**import android.view.View;**

**import android.view.ViewGroup;**

**import android.widget.ImageButton;**

**import android.widget.TextView;**

**import androidx.appcompat.app.AppCompatActivity;**

**import com.github.barteksc.pdfviewer.PDFView;**

**import com.github.barteksc.pdfviewer.listener.OnLoadCompleteListener;**

**import com.github.barteksc.pdfviewer.listener.OnPageChangeListener;**

**import com.github.barteksc.pdfviewer.scroll.DefaultScrollHandle;**

**import com.shockwave.pdfium.PdfDocument;**

**import java.io.InputStream;**

**import java.util.ArrayList;**

**import java.util.List;**

**public class PdfViewerLibro extends AppCompatActivity implements OnPageChangeListener, OnLoadCompleteListener {**

**private PDFView pdfView;**

**private Integer pageNumber = 0;**

**private String pdfFileName;**

**private int positionLibro;**

**private Libro libro;**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**// Configurar un diseño vacío directamente en el código**

**pdfView = new PDFView(this, null);**

**setContentView(pdfView);**

**ArrayList<Libro> libroArrayList = Libro.*LIBROS*;**

**Intent intent = getIntent();**

**if (intent != null && intent.hasExtra("LibroId")) {**

**positionLibro = intent.getIntExtra("LibroId", -1);**

**}**

**libro = libroArrayList.stream().filter(libro1 -> libro1.getId() == positionLibro).findFirst().get();**

**// Establecer límites de zoom**

**pdfView.setMinZoom(1f);**

**pdfView.setMaxZoom(5f);**

**// Carga y muestra el PDF**

**InputStream inputStream = getResources().openRawResource(libro.getPdfResource());**

**displayFromStream(inputStream);**

**// Botón de retroceso**

**ImageButton backButton = new ImageButton(this);**

**backButton.setImageResource(android.R.drawable.*ic\_media\_previous*);**

**backButton.setOnClickListener(view -> finish());**

**// Título del libro**

**TextView tituloLibro = new TextView(this);**

**tituloLibro.setSelected(true);**

**tituloLibro.setText(libro.getNombreLibro());**

**// Ocultar la barra de título**

**getSupportActionBar().hide();**

**// Ocultar los botones de zoom (puedes personalizar según tus necesidades)**

**ImageButton zoomInButton = new ImageButton(this);**

**ImageButton zoomOutButton = new ImageButton(this);**

**zoomInButton.setVisibility(View.*GONE*);**

**zoomOutButton.setVisibility(View.*GONE*);**

**// Añadir vistas al diseño**

**// Puedes personalizar el diseño según tus necesidades**

**setContentView(pdfView);**

**addContentView(backButton, new ViewGroup.LayoutParams(**

**ViewGroup.LayoutParams.*WRAP\_CONTENT*,**

**ViewGroup.LayoutParams.*WRAP\_CONTENT*));**

**addContentView(tituloLibro, new ViewGroup.LayoutParams(**

**ViewGroup.LayoutParams.*WRAP\_CONTENT*,**

**ViewGroup.LayoutParams.*WRAP\_CONTENT*));**

**addContentView(zoomInButton, new ViewGroup.LayoutParams(**

**ViewGroup.LayoutParams.*WRAP\_CONTENT*,**

**ViewGroup.LayoutParams.*WRAP\_CONTENT*));**

**addContentView(zoomOutButton, new ViewGroup.LayoutParams(**

**ViewGroup.LayoutParams.*WRAP\_CONTENT*,**

**ViewGroup.LayoutParams.*WRAP\_CONTENT*));**

**}**

**@Override**

**public void loadComplete(int nbPages) {**

**PdfDocument.Meta meta = pdfView.getDocumentMeta();**

**printBookmarksTree(pdfView.getTableOfContents(), "-");**

**}**

**@Override**

**public void onPageChanged(int page, int pageCount) {**

**pageNumber = page;**

**// Elimina esta línea para evitar que aparezca el contador en la parte superior**

**// setTitle(String.format("%s %s / %s", pdfFileName, page + 1, pageCount));**

**}**

**private void displayFromStream(InputStream inputStream) {**

**pdfView.fromStream(inputStream)**

**.defaultPage(pageNumber)**

**.enableSwipe(true)**

**.swipeHorizontal(true) // Establece el desplazamiento horizontal**

**.onPageChange(this)**

**.enableAnnotationRendering(true)**

**.onLoad(this)**

**.scrollHandle(new DefaultScrollHandle(this))**

**.load();**

**}**

**public void printBookmarksTree(List<PdfDocument.Bookmark> tree, String sep) {**

**for (PdfDocument.Bookmark b : tree) {**

**// Log.e(TAG, String.format("%s %s, p %d", sep, b.getTitle(), b.getPageIdx()));**

**if (b.hasChildren()) {**

**printBookmarksTree(b.getChildren(), sep + "-");**

**}**

**}**

**}**

**}**

**OlntemClicklistener.java**

**package com.example.BibliotecaAlumnos;**

**public interface OnItemClickListener {**

**void onItemClick(int position);**

**}**

**activity\_main.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"**

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent">**

**<LinearLayout**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:orientation="vertical"**

**android:padding="16dp"**

**android:layout\_centerInParent="true">**

**<androidx.cardview.widget.CardView**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**app:cardCornerRadius="8dp"**

**app:cardElevation="4dp">**

**<LinearLayout**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:orientation="vertical"**

**android:padding="16dp">**

**<!-- Campo de entrada para el usuario -->**

**<EditText**

**android:id="@+id/editTextUsuario"**

**android:layout\_width="220dp"**

**android:layout\_height="wrap\_content"**

**android:layout\_gravity="center\_horizontal"**

**android:hint="Usuario"**

**android:inputType="text"**

**android:textSize="30sp" />**

**<!-- Campo de entrada para la contraseña -->**

**<EditText**

**android:id="@+id/editTextContraseña"**

**android:layout\_width="223dp"**

**android:layout\_height="wrap\_content"**

**android:layout\_gravity="center\_horizontal"**

**android:hint="Contraseña"**

**android:inputType="textPassword"**

**android:textSize="30sp" />**

**</LinearLayout>**

**</androidx.cardview.widget.CardView>**

**<Button**

**android:id="@+id/buttonIniciarSesion"**

**android:layout\_width="250dp"**

**android:layout\_height="wrap\_content"**

**android:layout\_gravity="center\_horizontal"**

**android:layout\_marginTop="40dp"**

**android:clickable="true"**

**android:focusable="true"**

**android:text="Ingresar"**

**android:textSize="30sp"**

**android:background="#00FF00" /> <!-- Cambia #00FF00 al código de color verde que desees -->**

**</LinearLayout>**

**</RelativeLayout>**

**view\_inicio.xml**

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

**<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"**

**xmlns:card\_view="http://schemas.android.com/apk/res-auto"**

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent"**

**android:padding="16dp">**

**<androidx.recyclerview.widget.RecyclerView**

**android:id="@+id/recyclerView"**

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent"**

**android:layout\_alignParentStart="true"**

**android:layout\_marginStart="-10dp"**

**android:clipToPadding="false"**

**android:paddingTop="8dp"**

**android:paddingBottom="0dp" />**

**<include**

**layout="@layout/footer\_layout"**

**android:id="@+id/footer"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:layout\_alignParentBottom="true"/>**

**</RelativeLayout>**

**card\_item\_tarjeta.xml**

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

**<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"**

**xmlns:card\_view="http://schemas.android.com/apk/res-auto"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:padding="5dp">**

**<androidx.cardview.widget.CardView**

**android:id="@+id/cardView"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_centerHorizontal="true"**

**card\_view:cardCornerRadius="10dp"**

**card\_view:cardElevation="4dp"**

**card\_view:cardUseCompatPadding="true">**

**<LinearLayout**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:padding="5dp"**

**android:orientation="vertical">**

**<ImageView**

**android:id="@+id/imageView"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:adjustViewBounds="true"**

**android:scaleType="fitXY"**

**android:src="@drawable/libro1" />**

**<TextView**

**android:id="@+id/textViewConcepto"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:text="Concepto:"**

**android:textSize="16sp"**

**android:textStyle="bold"**

**android:layout\_marginTop="8dp"/>**

**<TextView**

**android:id="@+id/textViewDescripcion"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:text="Descripción de la imagen que uno desee utilizando el cardView como contenedor y el relative layout."**

**android:textSize="14sp"**

**android:layout\_marginTop="8dp"/>**

**</LinearLayout>**

**</androidx.cardview.widget.CardView>**

**</RelativeLayout>**

**footer\_layout.xml**

**<!-- footer\_layout.xml -->**

**<LinearLayout**

**xmlns:android="http://schemas.android.com/apk/res/android"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:orientation="vertical"**

**android:background="@color/colorPrimary">**

**<!-- Contenido del pie de página -->**

**<TextView**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_gravity="center"**

**android:padding="8dp"**

**android:text="Autor:Juan Diego © 2023 NejapaSoftware."**

**android:textColor="@android:color/white" />**

**</LinearLayout>**

**activity\_libros.xml**

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

**<androidx.coordinatorlayout.widget.CoordinatorLayout**

**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">**

**<androidx.recyclerview.widget.RecyclerView**

**android:id="@+id/recyclerViewLibros"**

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent" />**

**</androidx.coordinatorlayout.widget.CoordinatorLayout>**

**styles.xml**

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

**<resources>**

**<color name="colorPrimary">#3F51B5</color>**

**<color name="colorPrimaryDark">#303F9F</color>**

**<color name="colorAccent">#FF4081</color>**

**<style name="MyActionBarStyle" parent="Widget.AppCompat.Light.ActionBar.Solid.Inverse">**

**<item name="android:titleTextStyle">@style/MyTitleTextStyle</item>**

**</style>**

**<style name="MyTitleTextStyle" parent="TextAppearance.AppCompat.Widget.ActionBar.Title">**

**</style>**

**<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">**

**<item name="colorPrimary">@color/colorPrimary</item>**

**<item name="colorPrimaryDark">@color/colorPrimaryDark</item>**

**<item name="colorAccent">@color/colorAccent</item>**

**<item name="android:actionBarStyle">@style/MyActionBarStyle</item>**

**</style>**

**<style name="Theme.barra" parent="Theme.AppCompat.Light.DarkActionBar">**

**</style>**

**<style name="AppTheme.NoActionBar" parent="Theme.AppCompat.Light.NoActionBar">**

**</style>**

**</resources>**

**Explicación:**

**mainActivity.java**

**1. Layout y Vistas:**

* La interfaz de usuario está definida en el archivo de diseño **`activity\_main.xml`**
* Se utilizan dos `EditText` para que el usuario ingrese su nombre de usuario y contraseña.
* Hay un botón **(buttonIniciarSesion)** que el usuario puede hacer clic para intentar iniciar sesión.

**2. Variables y Constantes:**

* Se definen dos constantes USUARIO\_CORRECTO y CONTRASEÑA\_CORRECTA que representan las credenciales válidas para iniciar sesión.
* Se declaran dos variables `EditText` (editTextUsuario y editTextContraseña) para obtener el texto ingresado por el usuario.

**3. Método `onCreate`:**

* Este método se llama cuando la actividad se crea.
* Se oculta la barra de acción si está presente.
* Se asignan las vistas (`EditText` y `Button`) a las variables correspondientes.
* Se agrega un OnClickListener al botón de inicio de sesión para manejar los clics.

**4. Método `iniciarSesion`:**

* Este método se llama cuando el usuario hace clic en el botón de inicio de sesión.
* Obtiene el nombre de usuario y la contraseña ingresados por el usuario.
* Compara estas credenciales con las credenciales correctas (`USUARIO\_CORRECTO` y `CONTRASEÑA\_CORRECTA`).
* Si las credenciales son correctas, se llama al método `toCategoriaActivity` para iniciar la siguiente actividad (`ViewInicio`).
* Si las credenciales son incorrectas, se muestra un mensaje de error mediante un `Toast`.

**ViewInicio.java**

1. Layout:

* La interfaz de usuario está definida en el archivo `view\_inicio.xml`.

2. Inicialización:

* Se inicializa el `RecyclerView` y se configura con un `GridLayoutManager`. El número de columnas en el `GridLayoutManager` depende de la orientación del dispositivo (2 columnas en retrato, 3 en paisaje).
* Se crea una lista de objetos `Tarjeta` (`tarjetas`) que contiene información sobre cada tarjeta (imagen, concepto, descripción, tamaño de tarjeta, etc.).
* Se inicializa una lista filtrada (`tarjetasFiltradas`) con la lista original.

3. ActionBar:

* Se configura el `ActionBar` para mostrar un título ("Biblioteca Escolar") y habilita la navegación hacia atrás.

4. SearchView:

* Se infla un menú de búsqueda en el `onCreateOptionsMenu`.
* Se obtiene una referencia al `SearchView` y se configura un `OnQueryTextListener` para filtrar la lista de tarjetas mientras el usuario escribe.

5. Filtrado de Tarjetas:

* El método `filterList(String query)` filtra las tarjetas en función del texto de búsqueda ingresado por el usuario. La lista filtrada se actualiza y se notifica al adaptador.

6. Manejo de clics:

* El método `onOptionsItemSelected` maneja el clic en el ícono de la flecha hacia atrás en la barra de acción para cerrar la actividad actual.

7. Clase `Tarjeta`:

* Es una clase simple que representa una tarjeta con sus propiedades (imagen, concepto, descripción, tamaño de tarjeta, espacio entre tarjetas).

8. Clase `TarjetaAdapter`:

* Es una clase interna que extiende `RecyclerView.Adapter` y maneja la creación y visualización de las tarjetas en el `RecyclerView`.
* En el método `onBindViewHolder`, se configuran los datos de la tarjeta en cada vista y se establece un clic en la tarjeta para abrir un visor de PDF correspondiente.

**PdfViwerLibro.java**

1. Inicialización de la interfaz de usuario:

* La actividad crea un objeto `PDFView` y lo configura como el contenido principal de la actividad.
* Se obtiene la posición del libro que se debe mostrar desde el intent recibido. El libro correspondiente se obtiene de una lista estática de libros (`Libro.LIBROS`).
* Se configuran límites de zoom para el visor de PDF.

2. Carga y visualización del PDF:

* Se abre un `InputStream` desde el recurso de PDF asociado al libro.
* Se llama al método `displayFromStream` para cargar y mostrar el PDF utilizando la biblioteca `barteksc.pdfviewer`.
* Se habilita la navegación por páginas, el desplazamiento horizontal y se configuran manejadores de eventos para cambios de página y carga completa del documento.

3. Controles adicionales:

* Se añaden controles adicionales a la interfaz, como un botón de retroceso, un título de libro y botones de zoom (aunque se han configurado para que inicialmente estén ocultos).
* Se oculta la barra de acción (`getSupportActionBar().hide()`), probablemente para personalizar el diseño de la interfaz.

4. Manejo de eventos de carga y cambio de página:

* La actividad implementa las interfaces `OnPageChangeListener` y `OnLoadCompleteListener` para manejar eventos relacionados con el cambio de página y la carga completa del PDF.

5. Método `displayFromStream`:

* Este método utiliza la biblioteca `barteksc.pdfviewer` para cargar y mostrar el PDF desde un `InputStream`.
* Se configuran varias opciones, como la página predeterminada, la capacidad de desplazamiento horizontal y el manejo del desplazamiento.

**Libreria utilizada(**biblioteca de Barteksc**)**

La biblioteca `android-pdf-viewer` proporciona un visor de PDF personalizable para aplicaciones Android.

Estos son los elementos que ocupe;

1-`PDFView`:

* Este es el componente principal que se utiliza para mostrar el PDF en tu interfaz de usuario.

2- `OnPageChangeListener` y `OnLoadCompleteListener`:

* Estas interfaces son utilizadas para manejar eventos relacionados con la carga de páginas y la carga completa del PDF.

3- Métodos como `fromStream`, `defaultPage`, `enableSwipe`, etc.:

* Estos métodos son proporcionados por la biblioteca para configurar y personalizar la visualización del PDF en el `PDFView`.

4- Clases como `DefaultScrollHandle` y `PdfDocument`:

* Estas clases son parte de la biblioteca y se utilizan para manejar el desplazamiento y obtener información sobre el documento PDF, respectivamente.