Nizar Sanchez(2018-6482)

Realizar una agenda de eventos móvil que me permita agregar, actualizar y borrar recordatorios.  Con los campos recordatorio, fecha (puede ser un calendario). Debe mostrarme los recordatorios en una lista o Gridview. Que yo pueda entrar a la app y tenga un botón que diga ver recordatorios para hoy, al presionarlo me muestre mensaje a mensaje los recordatorios de este día los que no son de ese día no los muestre.

Que pueda ver todos , editar cada recordatorios y eliminar los que no me interesen.

**AndroidManifest.xml**

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

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

package="com.example.tarea7" >

<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/AppTheme" >

<activity android:name=".MainActivity" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity android:name=".ReminderActivity">

</activity>

</application>

</manifest>

**ReminderBaseHelper.java**

**package com.example.tarea7.database;**

import android.content.Context;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

public class ReminderBaseHelper extends SQLiteOpenHelper {

private static final int *VERSION*= 1;

private static final String *DATABASE\_NAME* = "crimeBase.db";

public ReminderBaseHelper(Context context){

super(context,*DATABASE\_NAME*,null,*VERSION*);

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL("create table "+ReminderDbSchema.ReminderTable.*NAME*+"("+"\_id integer primary key autoincrement,"

+ ReminderDbSchema.ReminderTable.Cols.*UUID*+","+

ReminderDbSchema.ReminderTable.Cols.*TITLE*+","+

ReminderDbSchema.ReminderTable.Cols.*DATE*+","+

ReminderDbSchema.ReminderTable.Cols.*DONE*+")");

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

}

}

**ReminderCursorWrapper.java**

**package com.example.tarea7.database;**

import android.database.Cursor;

import android.database.CursorWrapper;

import com.example.tarea7.Reminder;

import java.util.Date;

import java.util.UUID;

public class ReminderCursorWrapper extends CursorWrapper {

public ReminderCursorWrapper(Cursor cursor){

super(cursor);

}

public Reminder getReminder(){

String uuid = getString(getColumnIndex(ReminderDbSchema.ReminderTable.Cols.*UUID*));

String title = getString(getColumnIndex(ReminderDbSchema.ReminderTable.Cols.*TITLE*));

long date = getLong(getColumnIndex(ReminderDbSchema.ReminderTable.Cols.*DATE*));

int isDone= getInt(getColumnIndex(ReminderDbSchema.ReminderTable.Cols.*DONE*));

Reminder reminder = new Reminder(UUID.*fromString*(uuid));

reminder.setTitle(title);

reminder.setDone(isDone!=0);

reminder.setDate(new Date(date));

return reminder;

}

}

**ReminderDbSchema.java**

**package com.example.tarea7.database;**

public class ReminderDbSchema {

public static final class ReminderTable{

public static final String *NAME* = "reminders";

public static final class Cols{

public static final String *UUID* = "uuid";

public static final String *TITLE*= "title";

public static final String *DATE* ="date";

public static final String *DONE* = "done";

}

}

}

**DatePickerFragment,java**

**package com.example.tarea7;**

import android.app.Activity;

import android.app.AlertDialog;

import android.app.Dialog;

import android.content.DialogInterface;

import android.content.Intent;

import android.os.Build;

import android.os.Bundle;

import android.view.LayoutInflater;

import android.view.View;

import android.widget.DatePicker;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.fragment.app.DialogFragment;

import androidx.fragment.app.Fragment;

import java.util.Calendar;

import java.util.Date;

import java.util.GregorianCalendar;

public class DatePickerFragment extends DialogFragment {

private static final String *ARG\_DATE* = "date";

public static final String *EXTRA\_DATE*="com.example.tarea7.date";

private DatePicker mDatePicker;

@NonNull

@Override

public Dialog onCreateDialog(@Nullable Bundle savedInstanceState) {

Date date = (Date) getArguments().getSerializable(*ARG\_DATE*); //new Date to insert

Calendar calendar = Calendar.*getInstance*();

calendar.setTime(date);

int year = calendar.get(Calendar.*YEAR*);

int month = calendar.get(Calendar.*MONTH*);

int day = calendar.get(Calendar.*DAY\_OF\_MONTH*);

View view = LayoutInflater.*from*(getActivity()).inflate(R.layout.*dialog\_date*,null);

mDatePicker = (DatePicker) view.findViewById(R.id.*dialog\_date\_picker*);

mDatePicker.init(year,month,day,null);

return new AlertDialog.Builder(getActivity())

.setView(view)

.setTitle(R.string.*button\_date*)

.setPositiveButton(R.string.*button\_ok*, new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

int year = mDatePicker.getYear();

int month = mDatePicker.getMonth();

int day = mDatePicker.getDayOfMonth();

Date date = new GregorianCalendar(year, month, day).getTime();

sendResult(Activity.*RESULT\_OK*, date);

}

})

.create();

}

public static DatePickerFragment newInstance(Date date){

Bundle args = new Bundle();

args.putSerializable(*ARG\_DATE*,date);

DatePickerFragment fragment = new DatePickerFragment();

fragment.setArguments(args);

return fragment;

}

private void sendResult(int resultCode, Date date){

if (getTargetFragment()==null){

return;

}

Intent intent = new Intent();

intent.putExtra(*EXTRA\_DATE*,date);

Fragment targetFragment = getTargetFragment();

targetFragment.onActivityResult(getTargetRequestCode(),resultCode,intent);

}

}

**MainActivity.java**

**package com.example.tarea7;**

import android.os.Bundle;

import androidx.fragment.app.Fragment;

public class MainActivity extends SingleFragmentActivity {

@Override

protected Fragment createFragment() {

return new MainFragment();

}

}

**MainFragment.java**

**package com.example.tarea7;**

import android.content.Intent;

import android.os.Bundle;

import android.view.LayoutInflater;

import android.view.Menu;

import android.view.MenuInflater;

import android.view.MenuItem;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.fragment.app.Fragment;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import java.util.Calendar;

import java.util.Date;

import java.util.GregorianCalendar;

import java.util.List;

public class MainFragment extends Fragment {

private RecyclerView mRecyclerView;

private ReminderAdapter mAdapter;

private Button mButtonAddReminder;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

View view;

if(ReminderLab.*getInstance*(getContext()).getReminders().size()==0){

view = inflater.inflate(R.layout.*empty\_reminders*,null,false);

mButtonAddReminder = view.findViewById(R.id.*button\_new\_reminder*);

mButtonAddReminder.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

goToReminderFragment();

}

});

return view;

}else{

view = inflater.inflate(R.layout.*fragment\_main\_reminder\_list*,null,false);

mRecyclerView = view.findViewById(R.id.*recycler\_view*);

mRecyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));

updateUI();

return view;

}

}

@Override

public void onCreateOptionsMenu(@NonNull Menu menu, @NonNull MenuInflater inflater) {

super.onCreateOptionsMenu(menu, inflater);

inflater.inflate(R.menu.*fragment\_main\_reminder\_list*,menu);

}

@Override

public void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setHasOptionsMenu(true);

}

private void updateUI(){

ReminderLab reminderLab = ReminderLab.*getInstance*(getActivity());

List<Reminder> reminders = reminderLab.getReminders();

mAdapter = new ReminderAdapter(reminders);

mRecyclerView.setAdapter(mAdapter);

}

@Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {

switch (item.getItemId()){

case R.id.*new\_reminder*:

goToReminderFragment();

return true;

case R.id.*today\_reminder*:

showTodayReminders();

return true;

default:

return super.onOptionsItemSelected(item);

}

}

private class ReminderHolder extends RecyclerView.ViewHolder implements RecyclerView.OnClickListener{

private Reminder mReminder;

private ImageView mImageView;

private TextView mTextViewTitle;

private TextView mTextViewDate;

public ReminderHolder(LayoutInflater inflater,ViewGroup group){

super(inflater.inflate(R.layout.*list\_item\_reminder*,group,false));

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

mTextViewTitle = itemView.findViewById(R.id.*text\_view\_title\_item*);

mTextViewDate = itemView.findViewById(R.id.*text\_view\_date\_item*);

itemView.setOnClickListener(this);

}

public void bind(Reminder reminder){

mReminder = reminder;

mTextViewDate.setText(mReminder.getDate().toString());

mTextViewTitle.setText(mReminder.getTitle());

if(mReminder.isDone()){

mImageView.setVisibility(View.*VISIBLE*);

}else{

mImageView.setVisibility(View.*INVISIBLE*);

}

}

@Override

public void onClick(View v) {

Intent intent =ReminderFragment.*createOrUpdateIntent*(getContext(),getString(R.string.*button\_update*),mReminder.getUUID().toString());

startActivity(intent);

}

}

private class ReminderAdapter extends RecyclerView.Adapter<ReminderHolder> {

private List<Reminder> mReminderList;

public ReminderAdapter(List<Reminder> reminders){

mReminderList = reminders;

}

@NonNull

@Override

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

LayoutInflater layoutInflater = LayoutInflater.*from*(getActivity());

return new ReminderHolder(layoutInflater,parent);

}

@Override

public void onBindViewHolder(@NonNull ReminderHolder holder, int position) {

holder.bind(mReminderList.get(position));

}

@Override

public int getItemCount() {

return mReminderList.size();

}

}

private void showTodayReminders(){

List<Reminder> reminders= ReminderLab.*getInstance*(getContext()).getReminders();

Date date = new Date();

if(reminders.size()==0){

return;

}

for(Reminder r : reminders){

Calendar currentCalendar = new GregorianCalendar();

currentCalendar.setTime(date);

int currentDay = currentCalendar.get(Calendar.*DAY\_OF\_MONTH*);

int currentMonth = currentCalendar.get(Calendar.*MONTH*);

int currentYear = currentCalendar.get(Calendar.*YEAR*);

Calendar reminderCalendar = new GregorianCalendar();

reminderCalendar.setTime(r.getDate());

int reminderDay = reminderCalendar.get(Calendar.*DAY\_OF\_MONTH*);

int reminderMonth = reminderCalendar.get(Calendar.*MONTH*);

int reminderYear = reminderCalendar.get(Calendar.*YEAR*);

if(reminderDay== currentDay && reminderMonth==currentMonth && currentYear==reminderYear)

{

Toast.*makeText*(getContext(),r.toString(),Toast.*LENGTH\_SHORT*).show();

}

}

}

private void goToReminderFragment(){

Intent intent = ReminderFragment.*createOrUpdateIntent*(getContext(),getString(R.string.*button\_add*), "");

startActivity(intent);

}

}

**Reminder.java**

**package com.example.tarea7;**

import androidx.annotation.NonNull;

import java.util.Date;

import java.util.UUID;

public class Reminder {

private String mTitle;

private Date mDate;

private boolean mIsDone;

private UUID mUUID;

public Reminder(String title, Date date, boolean isDone) {

mUUID=UUID.*randomUUID*();

mTitle = title;

mDate = date;

mIsDone = isDone;

}

public Reminder(UUID id) {

mUUID=id;

}

public void setTitle(String title) {

mTitle = title;

}

public void setDate(Date date) {

mDate = date;

}

public void setDone(boolean done) {

mIsDone = done;

}

public String getTitle() {

return mTitle;

}

public Date getDate() {

return mDate;

}

public UUID getUUID() {

return mUUID;

}

public boolean isDone() {

return mIsDone;

}

@NonNull

@Override

public String toString() {

return "Title:"+getTitle()+ "\n"

+"Date: "+getDate().toString();

}

}

**ReminderActivity.java**

**package com.example.tarea7;**

import androidx.fragment.app.Fragment;

public class ReminderActivity extends SingleFragmentActivity {

@Override

protected Fragment createFragment() {

return new ReminderFragment();

}

}

**ReminderFragment.java**

**package com.example.tarea7;**

import android.content.Context;

import android.content.Intent;

import android.os.Bundle;

import android.view.LayoutInflater;

import android.view.Menu;

import android.view.MenuInflater;

import android.view.MenuItem;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.CheckBox;

import android.widget.EditText;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.fragment.app.Fragment;

import androidx.fragment.app.FragmentManager;

import java.util.Date;

import java.util.UUID;

public class ReminderFragment extends Fragment {

private static final String *CREATE\_OR\_UPDATE\_REMINDER* = "com.example.tarea7.CreateOrUpdate";

private static final String *DATE\_PICKER*="DialogDate";

private static final String *REMINDER\_ID*="com.example.tarea7.ReminderId";

private static final int *REQUEST\_DATE*=0;

private Button mDateButton;

private Button mCreateOrUpdateButton;

private EditText mTitleEditText;

private CheckBox mDoneCheckBox;

private boolean mIsCreateMode;

private Reminder mReminder;

private ReminderLab mReminderLab;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.*fragment\_reminder*, null, false);

mReminderLab = ReminderLab.*getInstance*(getContext());

mTitleEditText = view.findViewById(R.id.*edit\_text\_title*);

mCreateOrUpdateButton = view.findViewById(R.id.*button\_create\_or\_update*);

mDateButton = view.findViewById(R.id.*button\_date*); //

mDoneCheckBox= view.findViewById(R.id.*check\_box\_done*);

Intent intent = getActivity().getIntent();

String title = mTitleEditText.getText().toString();

String buttonText = intent.getStringExtra(*CREATE\_OR\_UPDATE\_REMINDER*);

mIsCreateMode = buttonText.equals(getString(R.string.*button\_add*));

mCreateOrUpdateButton.setText(buttonText);

mReminder = new Reminder(title,new Date(),false);

mDateButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

FragmentManager fm = getFragmentManager();

DatePickerFragment dialog = DatePickerFragment.*newInstance*(mReminder.getDate());

dialog.setTargetFragment(ReminderFragment.this,*REQUEST\_DATE*);

dialog.show(fm,*DATE\_PICKER*);

//Pick the date;

}

});

mCreateOrUpdateButton.setText(intent.getStringExtra(*CREATE\_OR\_UPDATE\_REMINDER*));

mCreateOrUpdateButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if(mIsCreateMode){

//Create Reminder

mReminder.setTitle(mTitleEditText.getText().toString());

mReminder.setDone(false);

mReminderLab.addReminder(mReminder);

}else{

//Update Reminder

boolean isDone = mDoneCheckBox.isChecked();

String title = mTitleEditText.getText().toString();

mReminder.setDone(isDone);

mReminder.setTitle(title);

mReminderLab.updateReminder(mReminder);

}

getBack();

}

});

if(mIsCreateMode){

mDoneCheckBox.setVisibility(View.*INVISIBLE*);

setHasOptionsMenu(false);

mCreateOrUpdateButton.setCompoundDrawablesWithIntrinsicBounds(R.drawable.*ic\_button\_add*,0,0,0);

}

else {

mDoneCheckBox.setVisibility(View.*VISIBLE*);

String id = intent.getStringExtra(*REMINDER\_ID*);

mReminder = ReminderLab.*getInstance*(getContext()).getReminder(UUID.*fromString*(id));

mDoneCheckBox.setChecked(mReminder.isDone());

mDateButton.setText(mReminder.getDate().toString());

mTitleEditText.setText(mReminder.getTitle());

mCreateOrUpdateButton.setText(R.string.*button\_update*);

mCreateOrUpdateButton.setCompoundDrawablesWithIntrinsicBounds(R.drawable.*ic\_button\_update*,0,0,0);

setHasOptionsMenu(true);

}

return view;

}

public static Intent createOrUpdateIntent(Context context,String buttonText,String id){

Intent intent = new Intent(context,ReminderActivity.class);

intent.putExtra(*CREATE\_OR\_UPDATE\_REMINDER*,buttonText);

if(!id.equals("")){

intent.putExtra(*REMINDER\_ID*,id);

}else {

return intent;

}

return intent;

}

@Override

public void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

if (requestCode==*REQUEST\_DATE*){

Date date = (Date) data.getSerializableExtra(DatePickerFragment.*EXTRA\_DATE*);

mReminder.setDate(date);

updateDate();

}

}

private void updateDate(){

mDateButton.setText(mReminder.getDate().toString());

}

@Override

public void onCreateOptionsMenu(@NonNull Menu menu, @NonNull MenuInflater inflater) {

super.onCreateOptionsMenu(menu, inflater);

inflater.inflate(R.menu.*fragment\_reminder*,menu);

}

@Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {

switch (item.getItemId()){

case R.id.*delete\_reminder*:

mReminderLab.deleteReminder(mReminder);

getBack();

return true;

default:

return super.onOptionsItemSelected(item);

}

}

private void getBack(){

Intent i=new Intent(getContext(), MainActivity.class);

i.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP*);

startActivity(i);

}

}

**ReminderLab.java**

**package com.example.tarea7;**

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import com.example.tarea7.database.ReminderBaseHelper;

import com.example.tarea7.database.ReminderCursorWrapper;

import com.example.tarea7.database.ReminderDbSchema;

import java.util.ArrayList;

import java.util.List;

import java.util.UUID;

public class ReminderLab {

private Context mContext;

private static ReminderLab *sReminderLab*;

private SQLiteDatabase mDatabase;

private ReminderLab(Context context){

mContext = context;

mDatabase = new ReminderBaseHelper(mContext).getWritableDatabase();

}

public static ReminderLab getInstance(Context context){

if(*sReminderLab*==null){

*sReminderLab* = new ReminderLab(context);

}

return *sReminderLab*;

}

public void addReminder(Reminder reminder){

ContentValues contentValues = *getContentValues*(reminder);

mDatabase.insert(ReminderDbSchema.ReminderTable.*NAME*,null,contentValues);

}

public void updateReminder(Reminder reminder){

String uuidString = reminder.getUUID().toString();

ContentValues values = *getContentValues*(reminder);

mDatabase.update(ReminderDbSchema.ReminderTable.*NAME*,values,

ReminderDbSchema.ReminderTable.Cols.*UUID*+ " = ?",

new String []{uuidString});

}

public List<Reminder> getReminders(){

List<Reminder> reminders = new ArrayList<>();

ReminderCursorWrapper cursorWrapper = queryReminder(null,null);

try{

cursorWrapper.moveToFirst();

while (!cursorWrapper.isAfterLast()){

reminders.add(cursorWrapper.getReminder());

cursorWrapper.moveToNext();

}

} finally {

cursorWrapper.close();

}

return reminders;

}

public void deleteReminder(Reminder reminder){

String uuid= reminder.getUUID().toString();

mDatabase.delete(ReminderDbSchema.ReminderTable.*NAME*,

ReminderDbSchema.ReminderTable.Cols.*UUID*+" = ?",

new String[]{uuid});

}

public Reminder getReminder(UUID id){

ReminderCursorWrapper cursorWrapper = queryReminder(

ReminderDbSchema.ReminderTable.Cols.*UUID* +" = ?",

new String[] {id.toString()}

);

try{

if(cursorWrapper.getCount()==0){

return null;

}

cursorWrapper.moveToFirst();

return cursorWrapper.getReminder();

}finally {

cursorWrapper.close();

}

}

private static ContentValues getContentValues(Reminder reminder){

// Preparing the values to insert

ContentValues contentValues = new ContentValues();

contentValues.put(ReminderDbSchema.ReminderTable.Cols.*UUID*,reminder.getUUID().toString());

contentValues.put(ReminderDbSchema.ReminderTable.Cols.*DATE*,reminder.getDate().getTime());

contentValues.put(ReminderDbSchema.ReminderTable.Cols.*DONE*,reminder.isDone()? 1:0);

contentValues.put(ReminderDbSchema.ReminderTable.Cols.*TITLE*,reminder.getTitle());

return contentValues;

}

private ReminderCursorWrapper queryReminder(String whereClause, String[] whereArgs) {

Cursor cursor = mDatabase.query(

ReminderDbSchema.ReminderTable.*NAME*,

null, // columns - null selects all columns

whereClause,

whereArgs,

null, // groupBy

null, // having

null // orderBy

);

return new ReminderCursorWrapper(cursor);

}

}

**SingleFragmenActivity.java**

**package com.example.tarea7;**

import android.os.Bundle;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import androidx.fragment.app.Fragment;

import androidx.fragment.app.FragmentManager;

public abstract class SingleFragmentActivity extends AppCompatActivity {

protected Fragment mFragment;

protected abstract Fragment createFragment();

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_single\_fragment*);

FragmentManager fm = getSupportFragmentManager();

mFragment = fm.findFragmentById(R.id.*single\_fragment\_container*);

if(mFragment ==null){

mFragment = createFragment();

fm.beginTransaction().add(R.id.*single\_fragment\_container*,mFragment).commit();

}

}

}

**ExampleInstrumentedTest.java**

**package com.example.tarea7;**

import android.content.Context;

import androidx.test.platform.app.InstrumentationRegistry;

import androidx.test.ext.junit.runners.AndroidJUnit4;

import org.junit.Test;

import org.junit.runner.RunWith;

import static org.junit.Assert.\*;

*/\*\**

*\* Instrumented test, which will execute on an Android device.*

*\**

*\** ***@see*** *<a href="http://d.android.com/tools/testing">Testing documentation</a>*

*\*/*

@RunWith(AndroidJUnit4.class)

public class ExampleInstrumentedTest {

@Test

public void useAppContext() {

// Context of the app under test.

Context appContext = InstrumentationRegistry.*getInstrumentation*().getTargetContext();

*assertEquals*("com.example.tarea7", appContext.getPackageName());

}

}

**ExampleUnitTest.java**

**package com.example.tarea7;**

import org.junit.Test;

import static org.junit.Assert.\*;

*/\*\**

*\* Example local unit test, which will execute on the development machine (host).*

*\**

*\** ***@see*** *<a href="http://d.android.com/tools/testing">Testing documentation</a>*

*\*/*

public class ExampleUnitTest {

@Test

public void addition\_isCorrect() {

*assertEquals*(4, 2 + 2);

}

}

**activity\_single\_fragment.xml**

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

<FrameLayout

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"

android:id="@+id/single\_fragment\_container"

tools:context=".MainActivity"/>

**dialog\_date.xml**

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

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

android:orientation="vertical" android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:calendarViewShown="false"

android:id="@+id/dialog\_date\_picker"

>

</DatePicker>

**empty\_reminders.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">

<Button

android:id="@+id/button\_new\_reminder"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

android:text="@string/new\_reminder"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.664" />

<TextView

android:id="@+id/textView"

android:layout\_width="279dp"

android:layout\_height="51dp"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

android:text="@string/empty\_crimes"

android:textSize="18sp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.603"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.448" />

</androidx.constraintlayout.widget.ConstraintLayout>

**fragment\_main\_reminder\_list.xml**

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

<androidx.recyclerview.widget.RecyclerView

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:id="@+id/recycler\_view"

>

</androidx.recyclerview.widget.RecyclerView>

**fragment\_reminder.xml**

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

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

android:orientation="vertical" android:layout\_width="match\_parent"

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

<EditText

android:id="@+id/edit\_text\_title"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:ems="10"

android:hint="@string/title\_hint"

android:inputType="textPersonName" />

<Button

android:id="@+id/button\_date"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:drawableLeft="@drawable/ic\_calendar"

android:text="@string/button\_date" />

<CheckBox

android:id="@+id/check\_box\_done"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/checkbox\_done" />

<Button

android:id="@+id/button\_create\_or\_update"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

</LinearLayout>

**list\_item\_reminder.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="horizontal">

<ImageView

android:id="@+id/imageView"

android:layout\_width="47dp"

android:layout\_height="77dp"

android:src="@drawable/ic\_done"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.956"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.0" />

<TextView

android:id="@+id/text\_view\_title\_item"

android:layout\_width="301dp"

android:layout\_height="22dp"

android:text="@string/reminder\_title"

android:textColor="@android:color/black"

android:textSize="16sp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.21"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.238" />

<TextView

android:id="@+id/text\_view\_date\_item"

android:layout\_width="298dp"

android:layout\_height="24dp"

android:text="@string/reminder\_date"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.216"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.875" />

</androidx.constraintlayout.widget.ConstraintLayout>

**fragment\_main\_reminder\_list.xml**

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

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

xmlns:app="http://schemas.android.com/apk/res-auto">

<item android:title="@string/new\_reminder"

android:icon="@drawable/ic\_add\_reminder"

android:id="@+id/new\_reminder"

app:showAsAction="ifRoom|withText"

/>

<item android:title="@string/reminder\_for\_today"

android:icon="@drawable/ic\_reminder"

android:id="@+id/today\_reminder"

app:showAsAction="ifRoom|withText"

/>

</menu>

**fragment\_reminder.xml**

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

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

xmlns:app="http://schemas.android.com/apk/res-auto">

<item

android:title="@string/delete\_reminder"

android:id="@+id/delete\_reminder"

android:icon="@drawable/ic\_delete\_reminder"

app:showAsAction= "ifRoom|withText"

>

</item>

</menu>

**colors.xml**

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

<resources>

<color name="colorPrimary">#008577</color>

<color name="colorPrimaryDark">#00574B</color>

<color name="colorAccent">#D81B60</color>

</resources>

**strings.xml**

**<resources>**

<string name="app\_name">Tarea7</string>

<string name="new\_reminder">NEW REMINDER</string>

<string name="delete\_reminder">DELETE REMINDER</string>

<string name="title\_hint">Enter the reminder"'"s title </string>

<string name="button\_date">Pick date</string>

<string name="checkbox\_done">Done</string>

<string name="button\_add">Add</string>

<string name="button\_update">Update</string>

<string name="button\_ok">OK</string>

<string name="reminder\_title">Title: %1$d</string>

<string name="reminder\_date">Date: %1$d</string>

<string name="empty\_crimes">There are no reminders added,tap the button to add a reminder. </string>

<string name="reminder\_for\_today">Show reminders for today</string>

</resources>

**styles.xml**

**<resources>**

<!-- Base application theme. -->

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

<!-- Customize your theme here. -->

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

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

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

</style>

</resources>

**ic\_add\_reminder.xml**

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

android:width="24dp"

android:height="24dp"

android:viewportWidth="24"

android:viewportHeight="24"

android:tint="#FFFFFF"

android:alpha="0.8">

<path

android:fillColor="#FF000000"

android:pathData="M19,13h-6v6h-2v-6H5v-2h6V5h2v6h6v2z"/>

</vector>

**ic\_button\_add.xml**

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

android:width="24dp"

android:height="24dp"

android:viewportWidth="24"

android:viewportHeight="24"

android:tint="#37FF08"

android:alpha="0.8">

<path

android:fillColor="#FF000000"

android:pathData="M19,13h-6v6h-2v-6H5v-2h6V5h2v6h6v2z"/>

</vector>

**ic\_button\_update.xml**

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

android:width="24dp"

android:height="24dp"

android:viewportWidth="24"

android:viewportHeight="24"

android:tint="#FF090A"

android:alpha="0.8">

<path

android:fillColor="#FF000000"

android:pathData="M21,10.12h-6.78l2.74,-2.82c-2.73,-2.7 -7.15,-2.8 -9.88,-0.1 -2.73,2.71 -2.73,7.08 0,9.79 2.73,2.71 7.15,2.71 9.88,0C18.32,15.65 19,14.08 19,12.1h2c0,1.98 -0.88,4.55 -2.64,6.29 -3.51,3.48 -9.21,3.48 -12.72,0 -3.5,-3.47 -3.53,-9.11 -0.02,-12.58 3.51,-3.47 9.14,-3.47 12.65,0L21,3v7.12zM12.5,8v4.25l3.5,2.08 -0.72,1.21L11,13V8h1.5z"/>

</vector>

**ic\_delete\_reminder.xml**

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

android:width="24dp"

android:height="24dp"

android:viewportWidth="24"

android:viewportHeight="24"

android:tint="#FFFFFF"

android:alpha="0.8">

<path

android:fillColor="#FF000000"

android:pathData="M6,19c0,1.1 0.9,2 2,2h8c1.1,0 2,-0.9 2,-2V7H6v12zM19,4h-3.5l-1,-1h-5l-1,1H5v2h14V4z"/>

</vector>

**ic\_done.xml**

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

android:width="24dp"

android:height="24dp"

android:viewportWidth="24"

android:viewportHeight="24"

android:tint="#37FF08"

android:alpha="0.8">

<path

android:fillColor="#FF000000"

android:pathData="M9,16.17L4.83,12l-1.42,1.41L9,19 21,7l-1.41,-1.41z"/>

</vector>