

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);
        DatePicker = (DatePicker)
view.findViewById(R.id.dialog_date_picker);
        DatePicker.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 = DatePicker.getYear();
                    int month = DatePicker.getMonth();
                    int day = DatePicker.getDayOfMonth();
                    Date date = new GregorianCalendar(year,
month, day).getTime();
                    setResult(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().siz
e()==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(),getStri
ng(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 inflater =
LayoutInflater.from(getActivity());
        return new ReminderHolder(inflater,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(), getStrin
g(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.setCompoundDrawablesWithIntrinsicBoun
ds(R.drawable.ic_button_add, 0, 0, 0);
}
else {
    mDoneCheckBox.setVisibility(View.VISIBLE);
    String id = intent.getStringExtra(REMINDER_ID);

```

```

        mReminder =
ReminderLab.getInstance(getApplicationContext()).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(getApplicationContext(),
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,c
ontentValues);
        }
        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,mF
ragment).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().getTargetConte
xt();
        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>
```