



ESCUELA SUPERIOR DE INGENIERÍA

Programación en Internet

Grado en Ingeniería Informática

Implementación de una aplicación Android que invoque
servicios web Rest

Autores:

Antonio Ruiz Rondán y Andrés Martínez Gavira

Supervisores:

Mercedes Rodríguez García y Guadalupe Ortiz Bellot

Cádiz, 23 de enero de 2018

Índice General

Introducción	5
1. Instalación del software necesario	5
1.1. Android Studio	6
1.2. Android SDK	6
1.3. Android AVD Manager	7
2. Creación e implementación del proyecto	9
2.1. Creación de OftApp	9
2.2. Implementación de OftApp	13
2.3. El Cliente App Android + Servicios Web REST	21
3. Probando la aplicación	22
Bibliografía	35
Anexos	37

Introducción

En este documento se va a abordar, a modo de tutorial, cómo instalar, configurar, crear y probar una aplicación android a través del software Android Studio, principalmente, junto a otro tipo de software de apoyo, para poderla observar en Smartphones. Así nuestra App versa sobre el “**I Congreso de Oftalmología de Cádiz**” en el mes de Octubre.

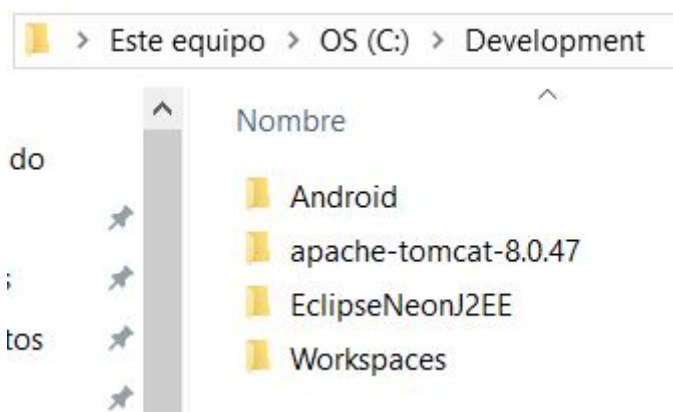
1. Instalación del software necesario

Para poder realizar un seguimiento satisfactorio de este tutorial es recomendable tener instalado el siguiente software:

1. Eclipse Neon v1 [1]
2. Apache Tomcat v8.0.47 [2]
3. Android Studio v2.3.2 [3]
4. Java Development Kit(JDK) & Java Runtime Environment(JRE) [4]
5. Android Software Development Kit(SDK)
6. WampServer u similares(Opcional) [5]

Dicho software debe de instalarse en la unidad del sistema operativo, normalmente C:\ en Windows dentro de la carpeta C:\Development, creando a su vez más carpetas para separar la parte de propia del software de la de los workspaces de android y eclipse.

Nos debe de quedar una estructura final de directorios así:



Se recomienda que se trabaje sobre un sistema operativo Windows en sus últimas versiones 7, 8, 8.1 o 10. Asimismo el uso de WampServer u similares nos es indiferente, concretamente para este tutorial, no así para los anteriores donde ha jugado su papel.

1.1. Android Studio

Para instalar Android Studio en su versión 2.3.2 debemos de dirigirnos a la siguiente página:

<https://developer.android.com/studio/index.html>

El proceso de instalación es sencillo e intuitivo. No debemos de olvidar que debemos de instalarlo dentro de development, en la carpeta Android.

Para configurarlo debemos de tener especial cuidado con lo que hacemos, pues podemos estropearlo y con ello volver a tener que instalarlo de nuevo.

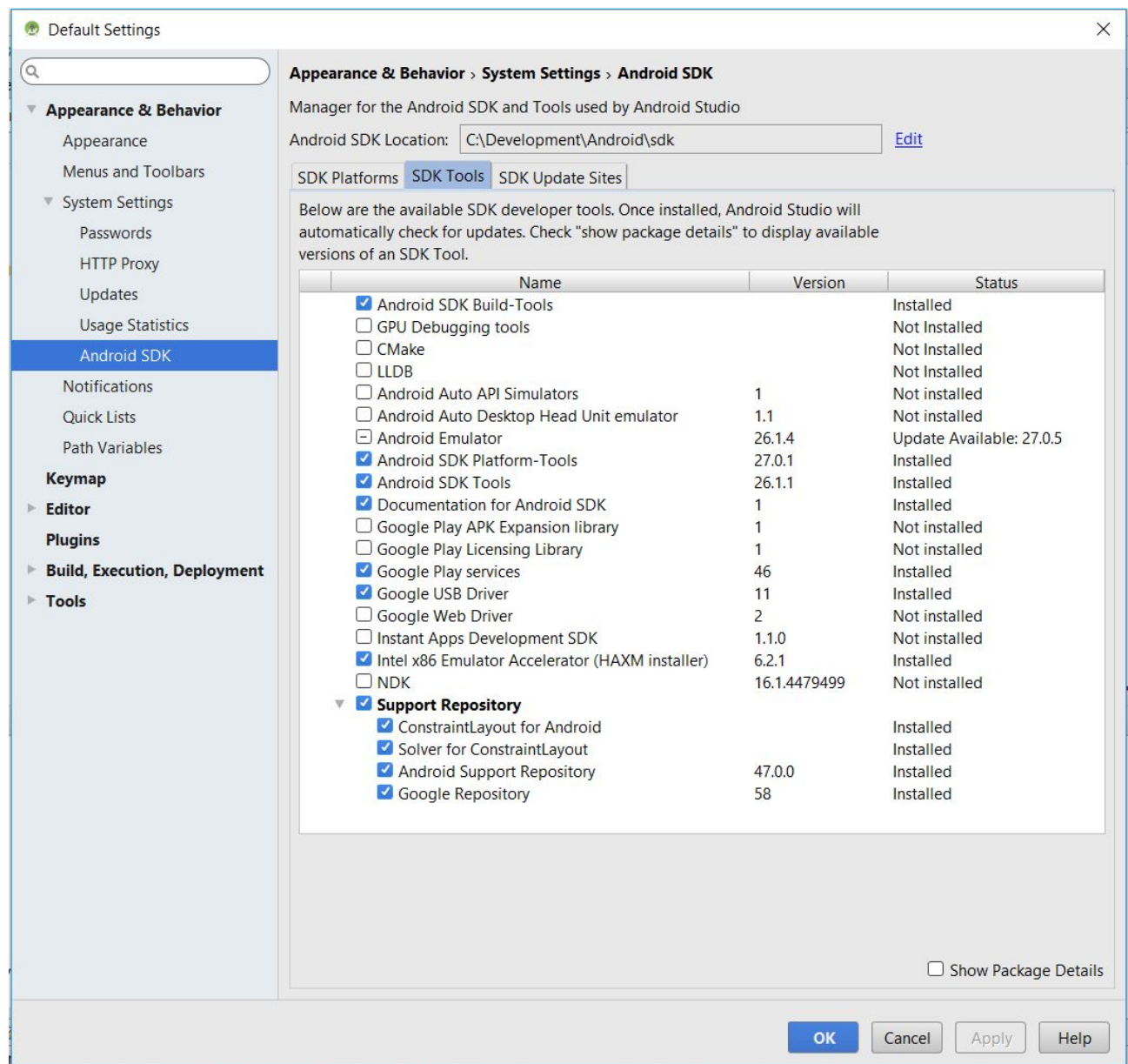
1.2. Android SDK

Este software es el que viene destinado a la parte de desarrollo, emulación y soporte de la gran multitud de dispositivos que existen dentro del SO Android. Por el hecho de que existen gran variedad de dispositivos de diferentes características es bueno saber que la App a desarrollar debe de ser programada para la mayoría de ellos. Nosotros hemos acordado seguir a día de hoy la API de Jelly Bean para Android 4.1 con acceso a los servicios de Google Play.

Todo esto se realiza a través de la interfaz de Android SDK repartida entre platforms y tools.

Estos sería lo que debemos de instalar o nuestros requisitos:

1. Imagen de Jelly Bean con SO Android 4.1 (con Google Play Services).
2. Android SDK Build Tools.
3. Android SDK Platform-Tools. (v27.0.1)
4. Android SDK Tools.(v26.1.1)
5. Documentación for Android SDK.(v1)
6. Google Play Services.(v46)
7. Google USB Driver.(v11)
8. Intel x86 Emulator Accelerator(HAXM installer).(v6.2.1)
9. Support Repository
 - 9.1. Constraint Layout for Android.
 - 9.2. Solver for Constraint Layout.
 - 9.3. Android Support Repository. (v47.0.0)
 - 9.4. Google Repository. (v58)



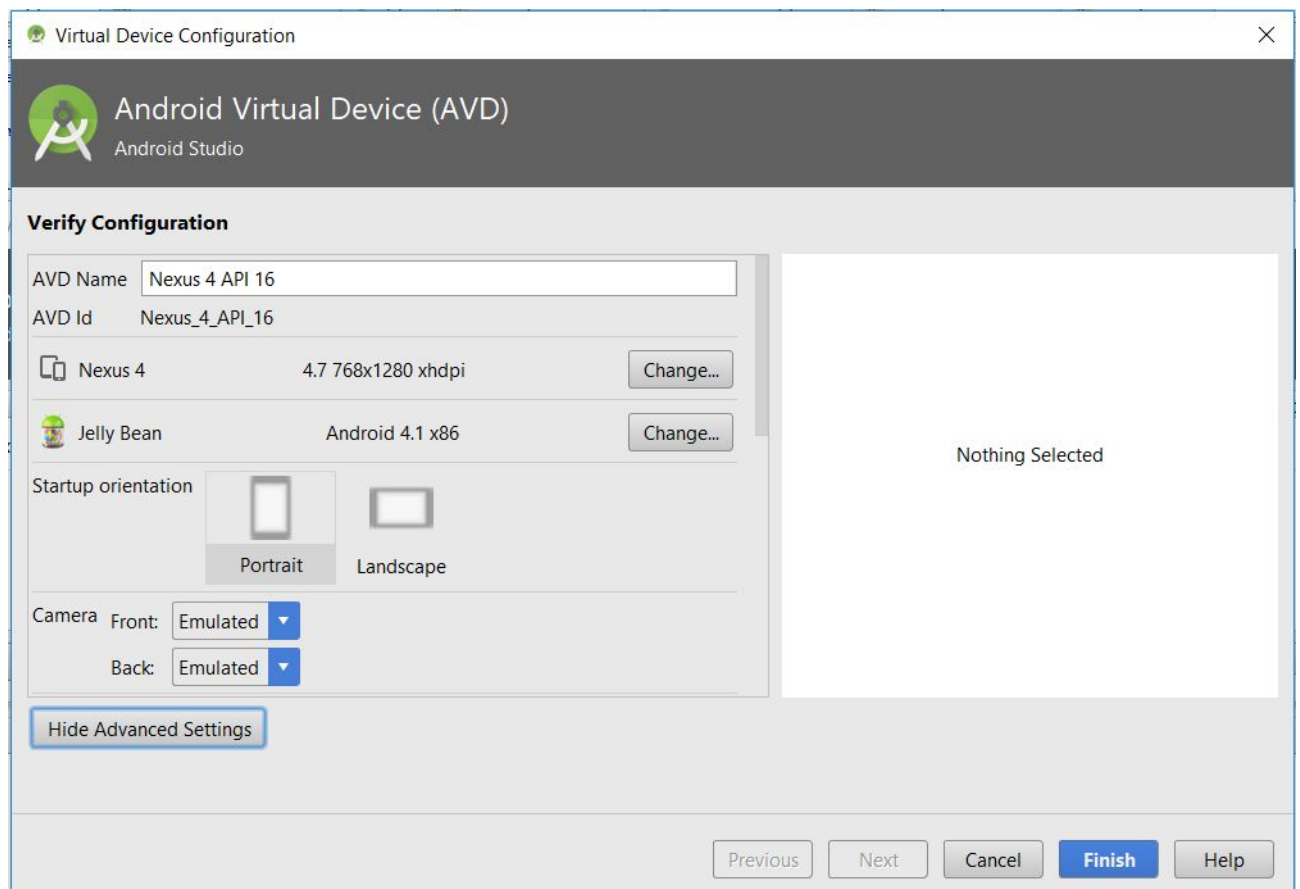
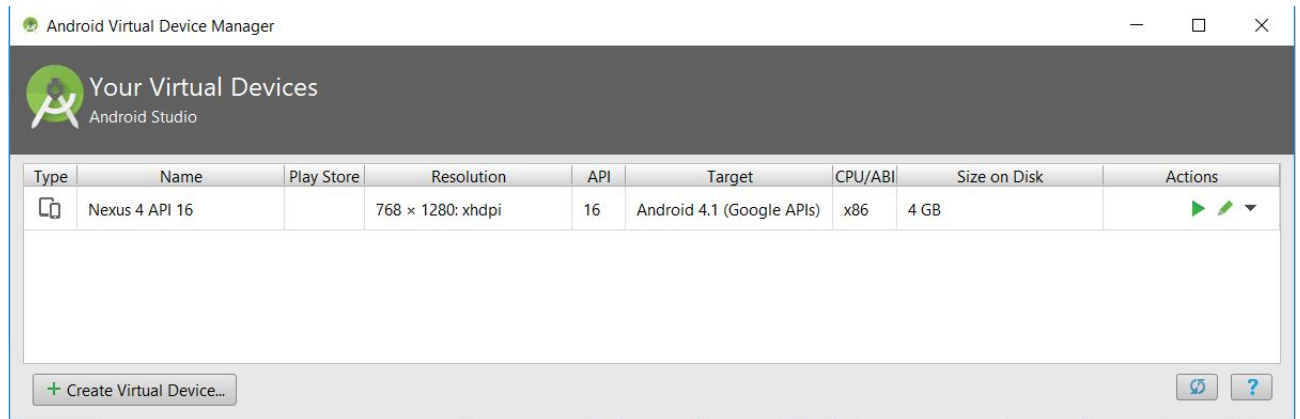
1.3. Android AVD Manager

Para poder ejecutar y probar nuestra App debemos de tener incorporado a nuestro IDE la herramienta Android Virtual Device Manager (AVD Manager).

Consiste en un conjunto de máquinas virtuales de diferentes dispositivos y que emulan su comportamiento.

Veámos cómo nos debe de quedar según los requisitos arriba expuestos:

Implementación de una aplicación Android que invoque servicios web Rest



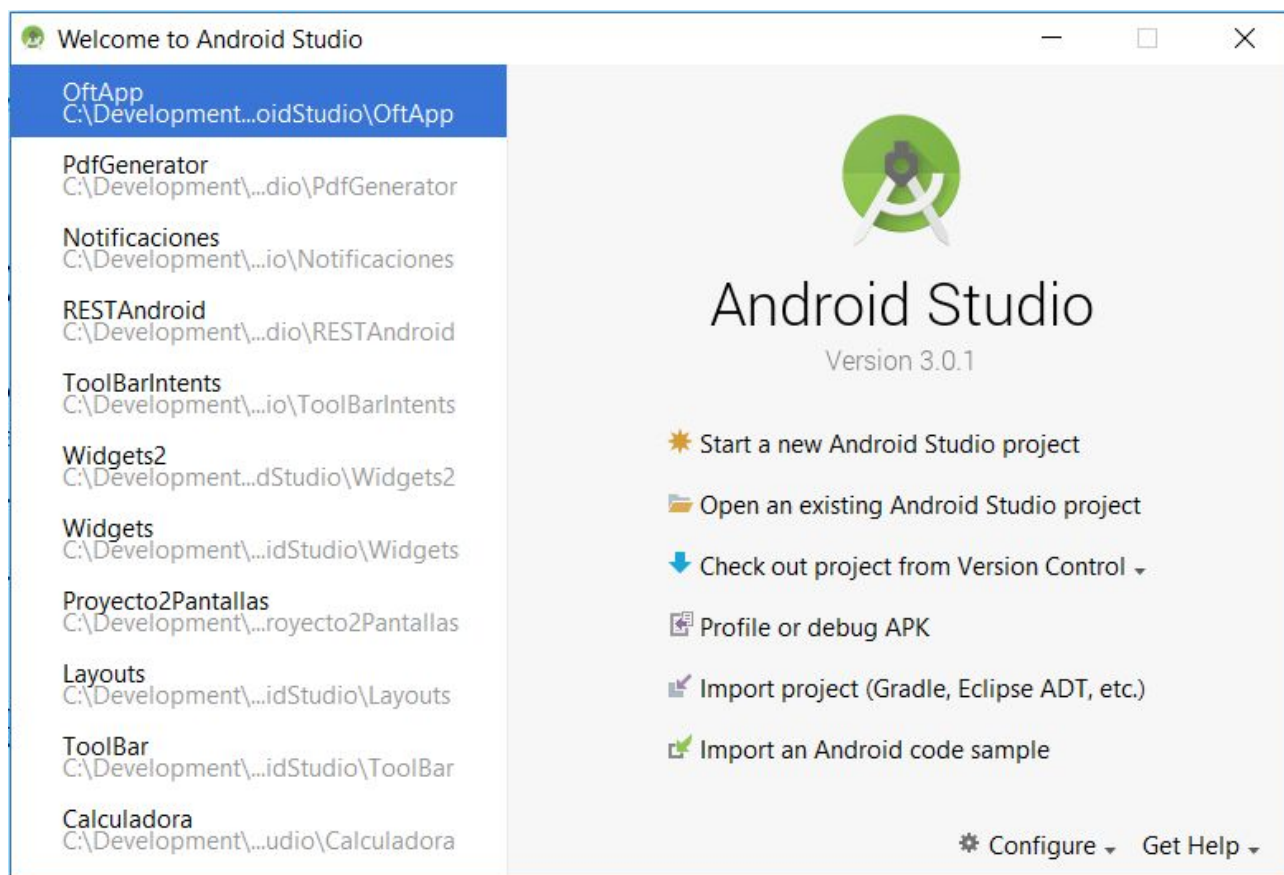
Y listo, con esto ya podemos ejecutar nuestra App sin ningún tipo de problema, ya sólo nos quedaría probarla en un dispositivo final o físico.

2. Creación e implementación del proyecto

“*I Congreso de Oftalmología de Cádiz*”, es una App que para su creación deberemos de dar los pasos siguientes:

2.1. Creación de OftApp

1. Abrimos el Android Studio y clickeamos en “Start a new Android Studio Project” con el botón izquierdo.



2. Ponemos un nombre al proyecto, que por convención debe de empezar con una letra en mayúscula. Por ejemplo **“OftApp”**.

Create New Project

Create Android Project

Application name
OftApp

Company domain
uca.es

Project location
C:\Development\Workspaces\WS_AndroidStudio\OftApp

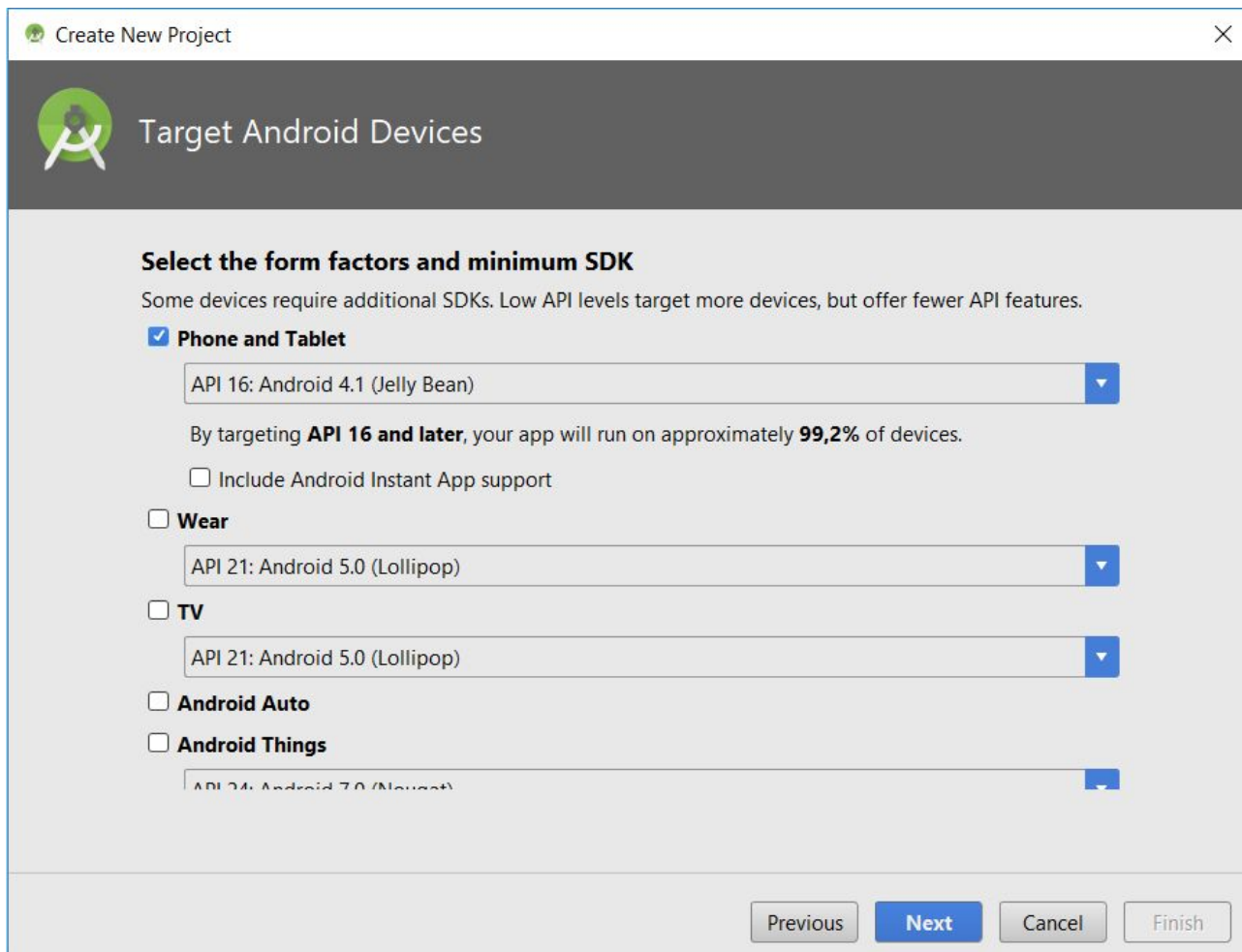
Package name
es.uca.oftapp Edit

☐ Include C++ support
☐ Include Kotlin support

⚠ 'OftApp' already exists at the specified project location.

Previous Next Cancel Finish

3. Damos click en siguiente y elegimos la API 16: Android 4.1 (Jelly Bean) (99.2% de compatibilidad de los dispositivos que usan android como SO). Y click en siguiente.



Create New Project

Target Android Devices

Select the form factors and minimum SDK

Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

☒ **Phone and Tablet**

API 16: Android 4.1 (Jelly Bean)

By targeting **API 16 and later**, your app will run on approximately **99,2%** of devices.

☐ Include Android Instant App support

☐ **Wear**

API 21: Android 5.0 (Lollipop)

☐ **TV**

API 21: Android 5.0 (Lollipop)

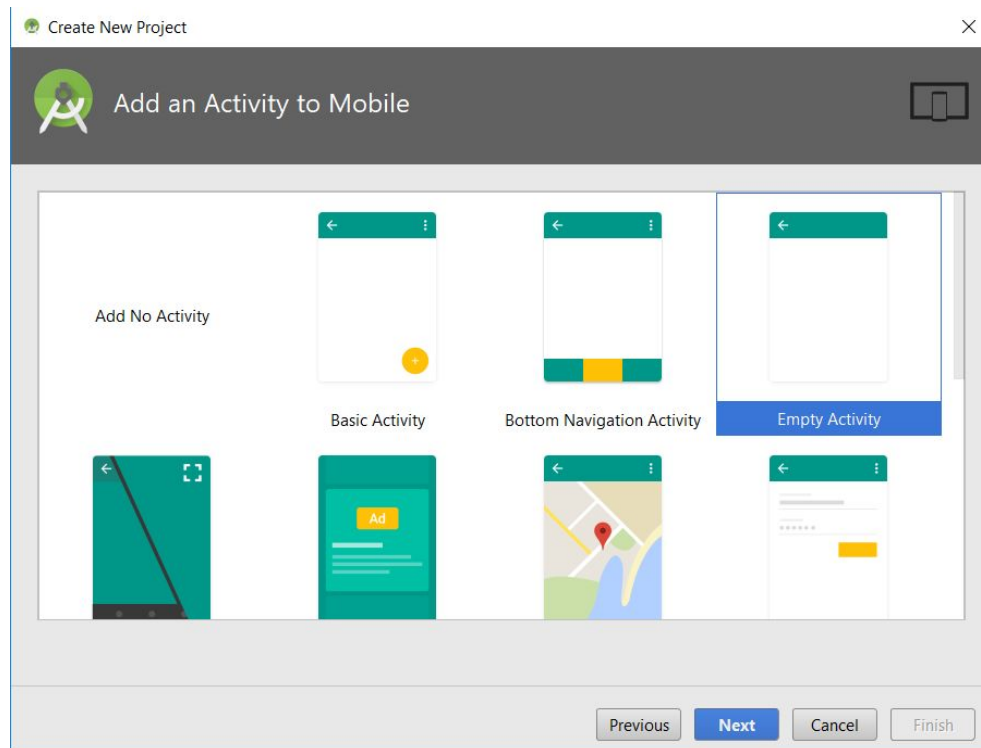
☐ **Android Auto**

☐ **Android Things**

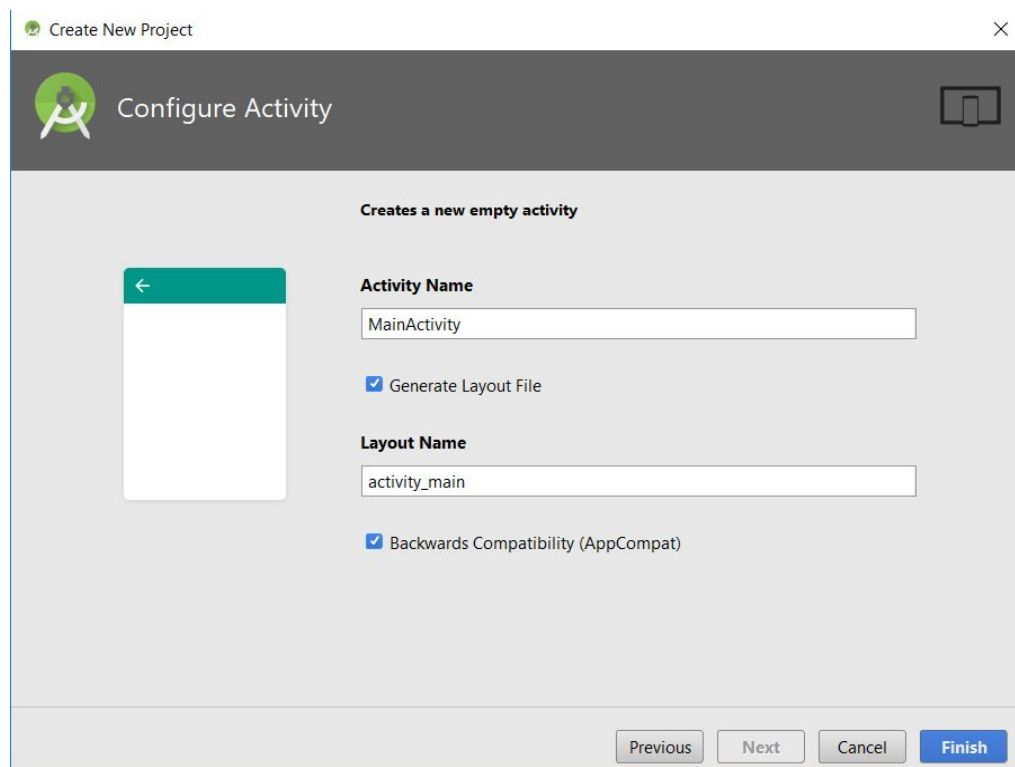
API 24: Android 7.0 (Nougat)

Previous Next Cancel Finish

4. Ahora elegimos Empty Activity y click en siguiente:



5. Finalmente dejamos la actividad principal con el nombre que viene por defecto y su xml asociado también, estos son respectivamente, MainActivity.java y activity_main.xml.



6. Click en Finish y listo.

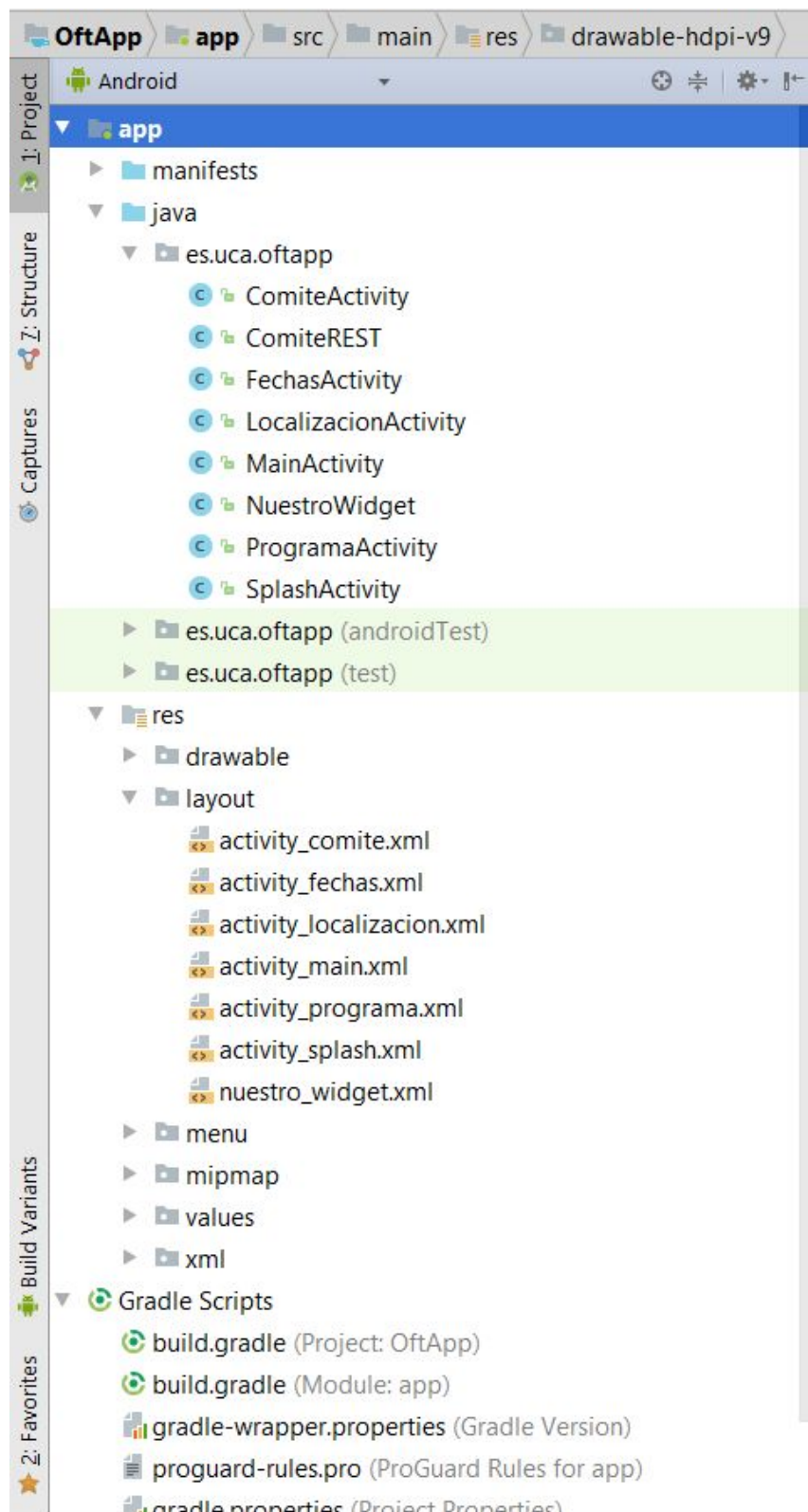
Ahora lo que tenemos que hacer es crear el resto de las clases y xml asociados a las mismas de la misma forma, teniendo en cuenta que siempre debemos respetar las convenciones del lenguaje de programación orientado a objetos Java. Por ello debemos tener cuidado con el uso inadecuado de mayúsculas o minúsculas al principio de cada nombre o característica.

También hay que tener en cuenta que además, que el procedimiento de creación del resto de objetos que componen el proyecto **“OftApp”** es relativamente sencillo y dependiendo de si es un paquete, una clase, un fichero xml, se creará de forma más o menos semi-automática.

2.2. Implementación de OftApp

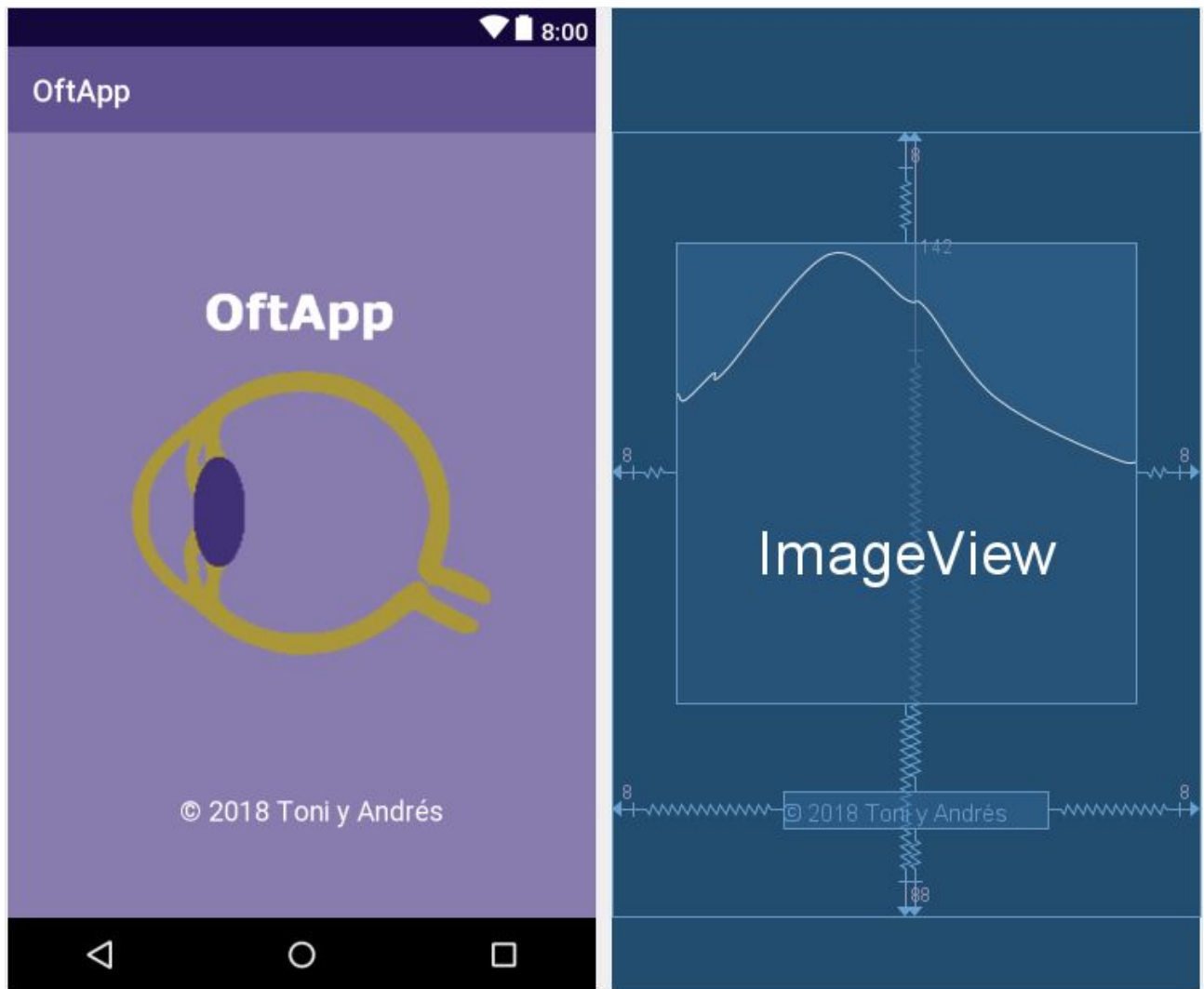
Para implementar nuestra App debemos ser muy metódicos, de tal forma que el resultado de la codificación de las clases de Java, la configuración de los ficheros XML y los asociados a la compilación del proyecto Gradle junto al fichero AndroidManifest.xml no interfieran en absoluto ninguno de ellos entre sí, es decir no haya ningún tipo de conflictos o incompatibilidades.

De esta forma, nuestro proyecto **“OftApp”** constará de las siguientes partes y tendrá la siguiente estructura:

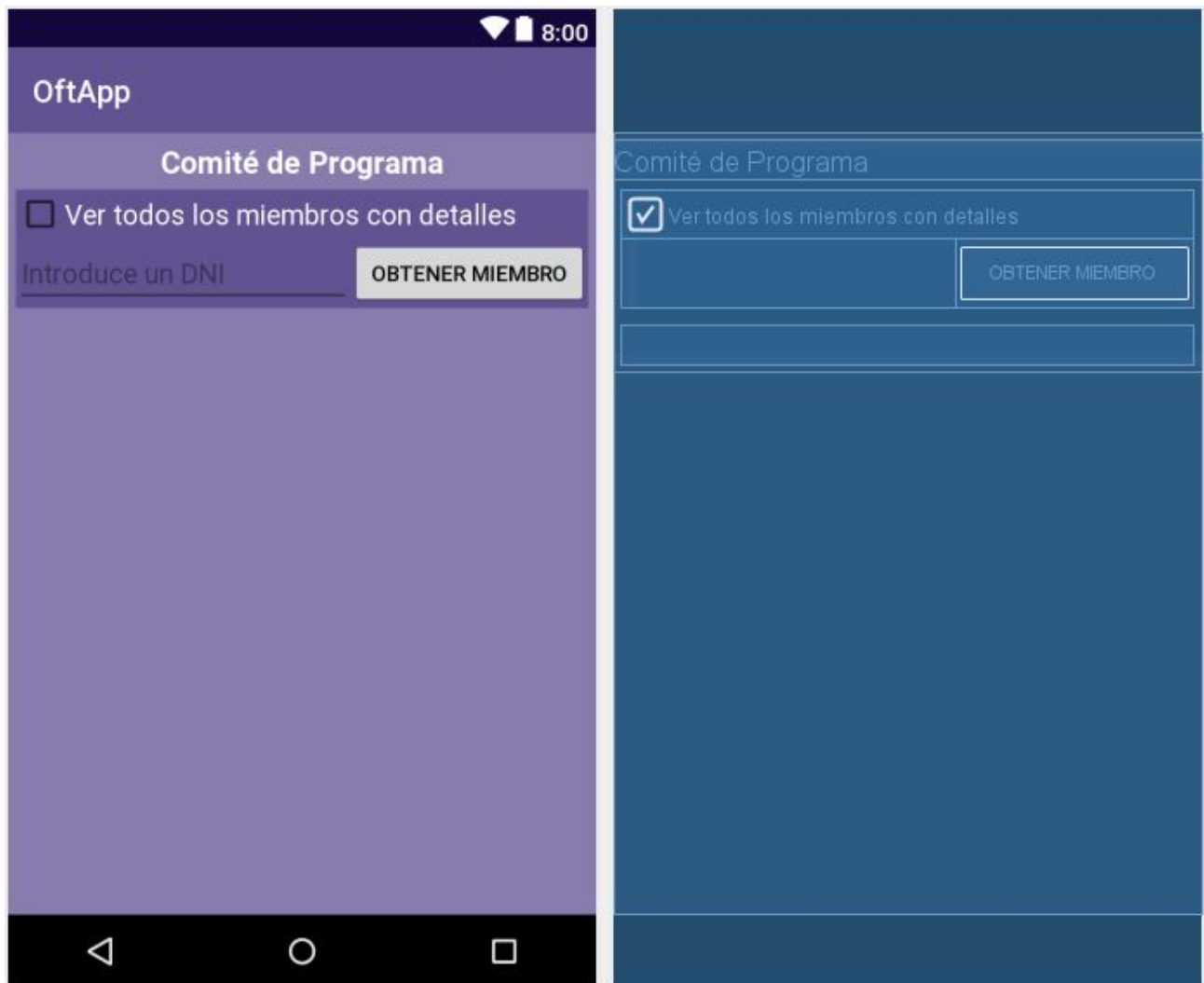


Veamos el diseño, de los layouts o esqueletos de las vistas de la aplicación:

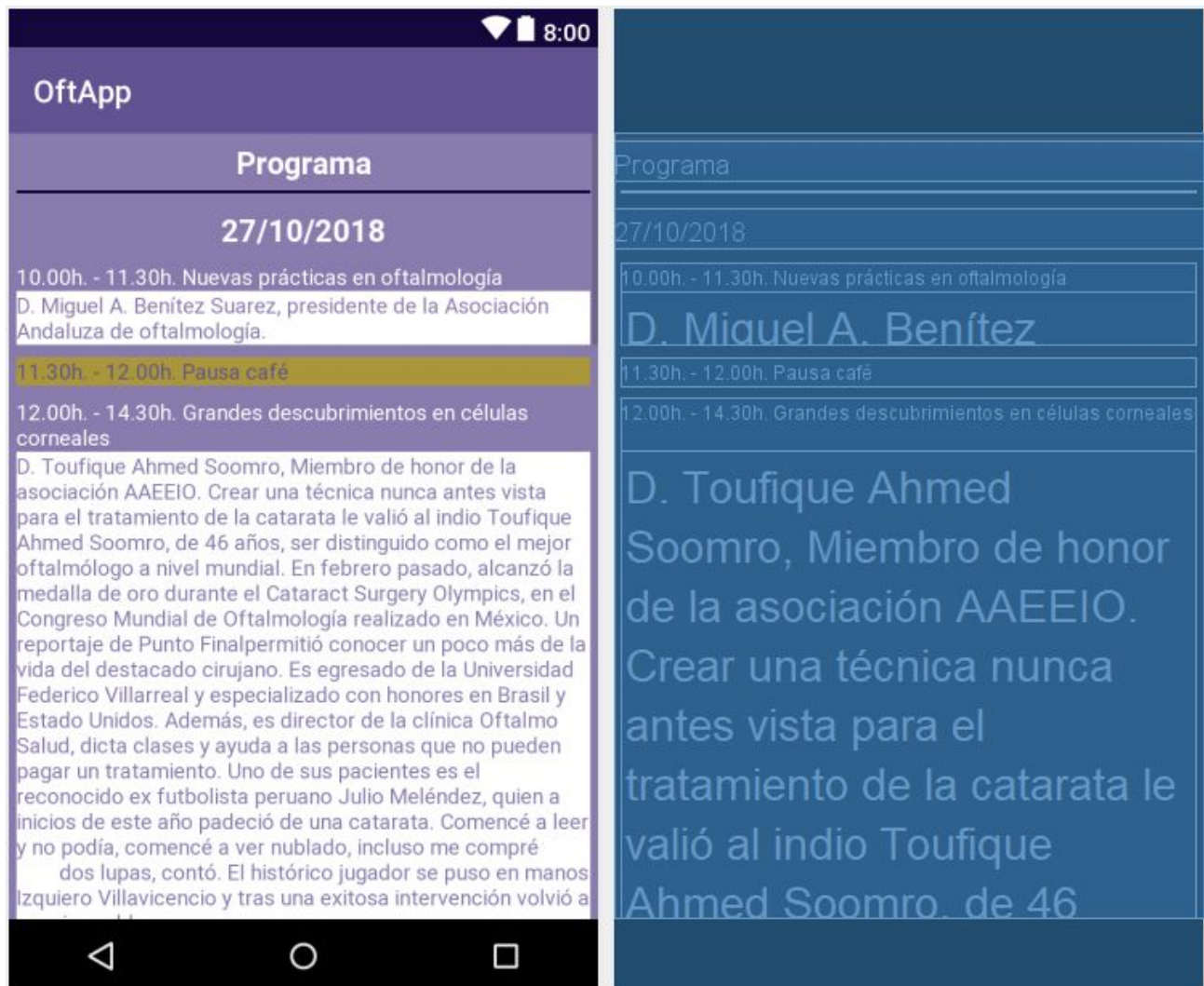
1. activity_main.xml: Layout correspondiente a la vista principal de la App.



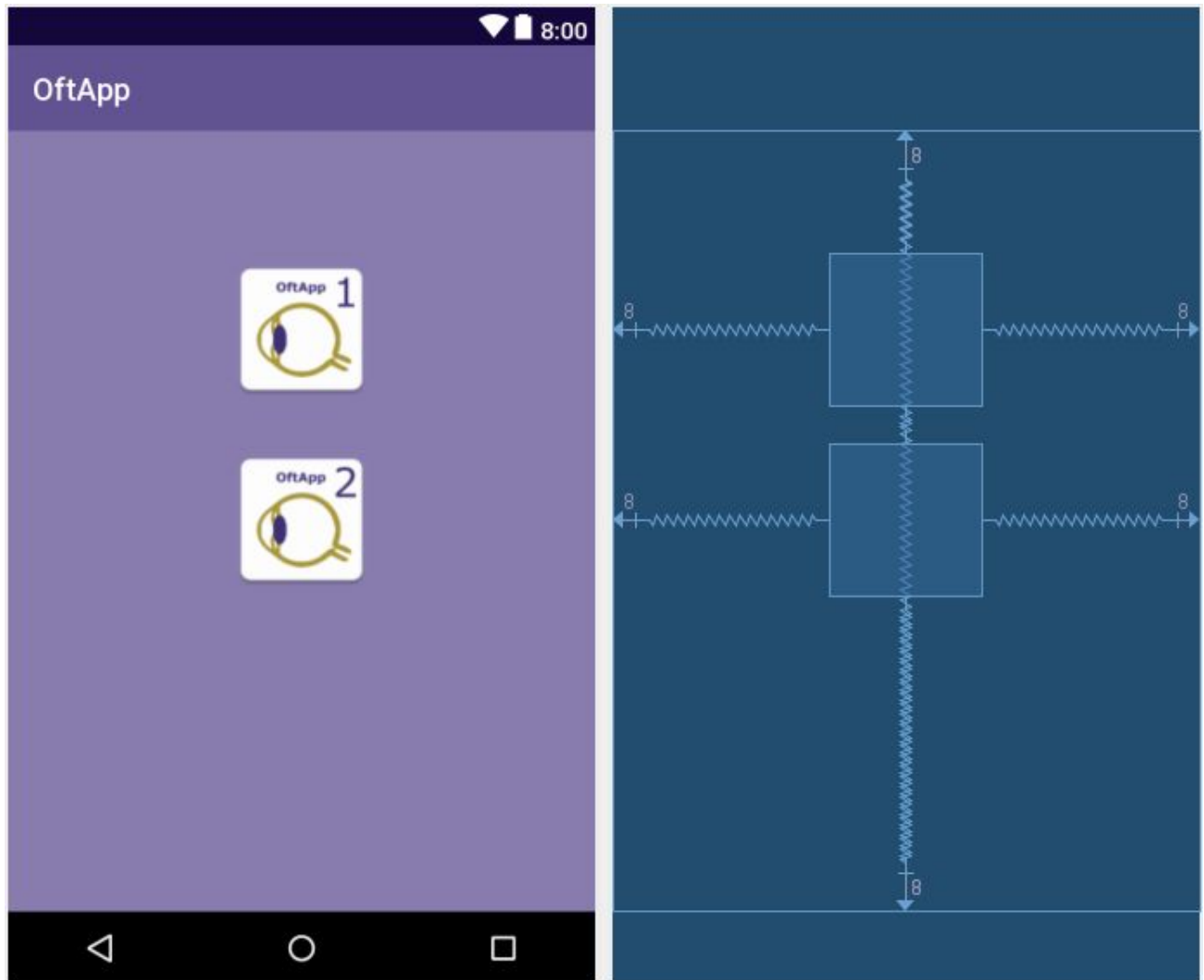
2. activity_comite.xml: Layout correspondiente a la vista del comité del programa.



3. activity_programa.xml: Layout correspondiente a la vista del programa de la App.



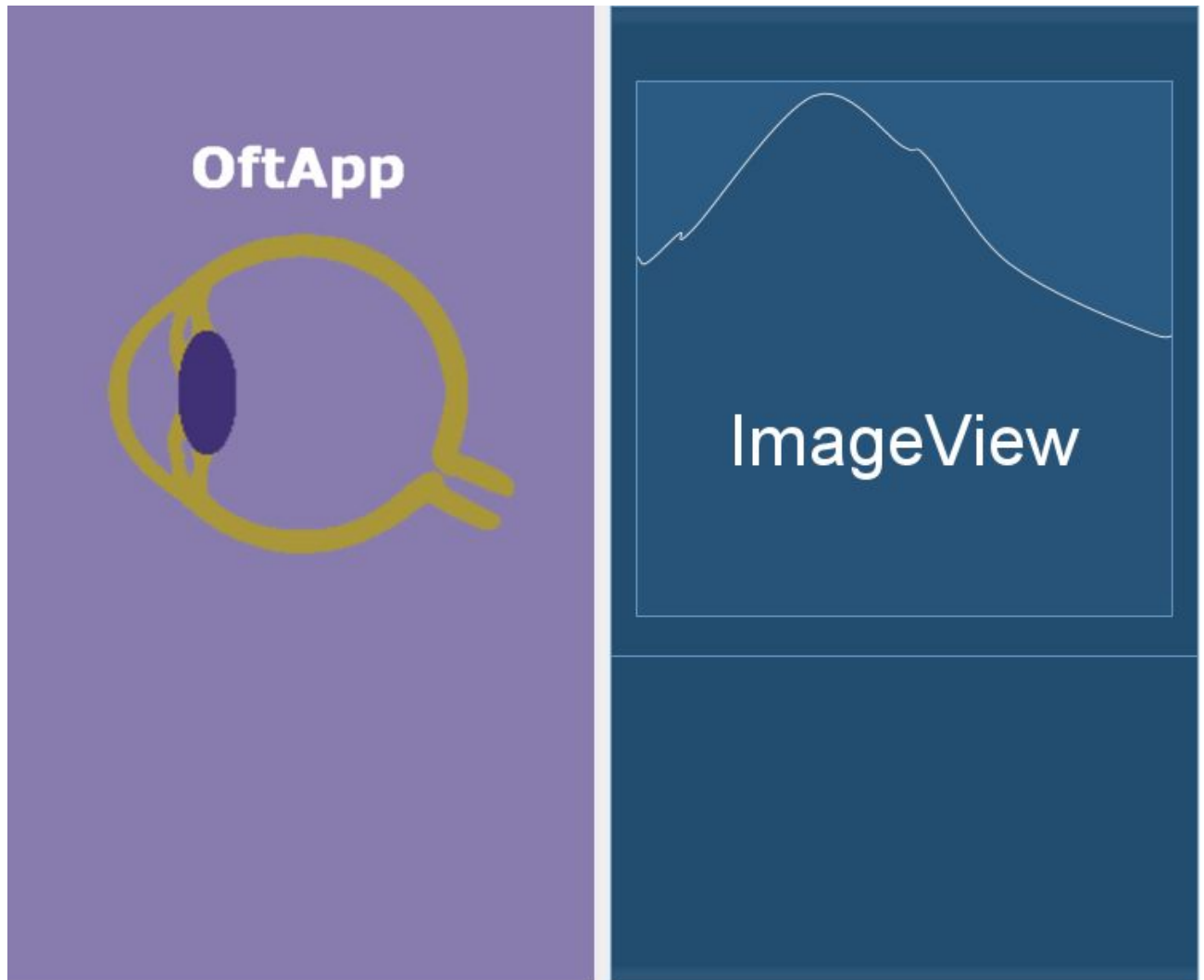
4. activity_fechas.xml: Layout correspondiente a la vista de las fechas importantes.



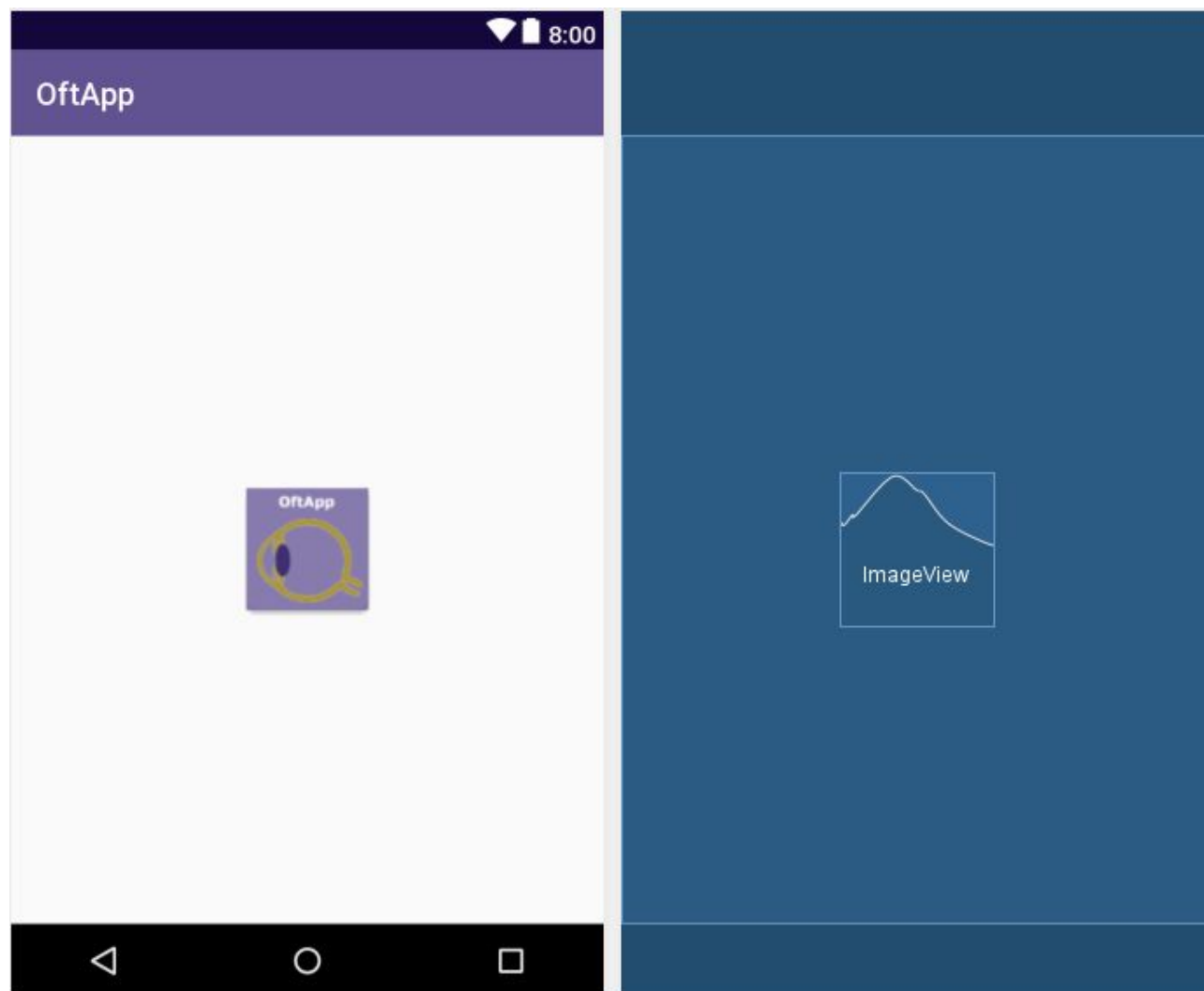
5. activity_localizacion.xml: Layout correspondiente a la vista de localización del congreso.



6. activity_splash.xml: Layout correspondiente a la vista de arranque de la App.



7. nuestro_widget.xml: El widget que proporcionamos en nuestra App.



2.3. El Cliente App Android + Servicios Web REST

El cliente App android es el encargado de llamar a los servicios web REST que teníamos implementados en el servidor de apache Tomcat.

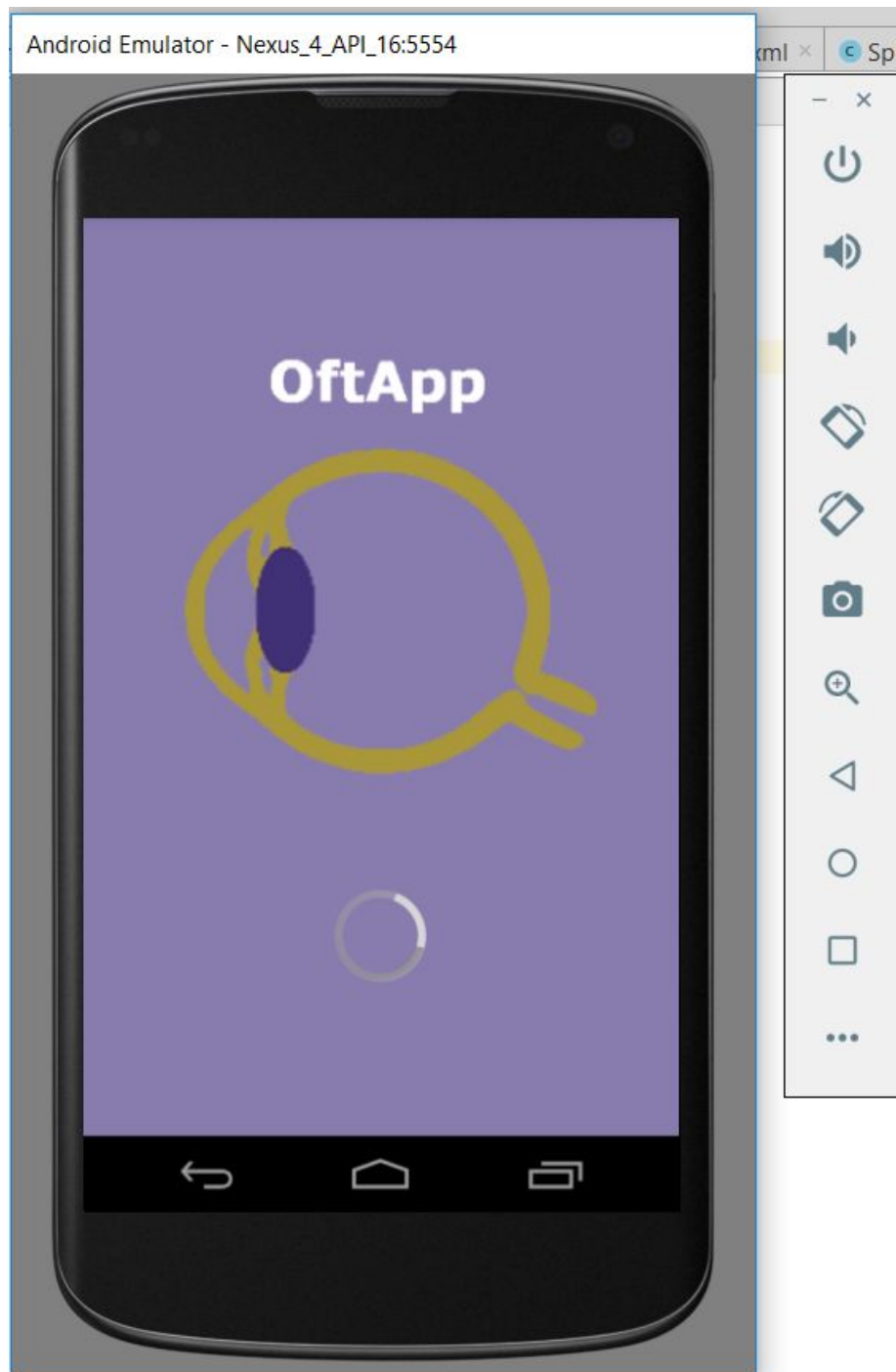
Para ello hacemos uso de la clase ComiteREST que hereda de AsyncTask para así poder hacer peticiones asíncronas a nuestro servicio web REST **Servicio** ayudados por sus métodos doInBackground, que ejecutado en segundo plano, es el encargado de comunicarse con el servicio web REST para tratar los datos obtenidos y pasárselos a onPostExecute que será el otro método encargado de comunicar esos datos con la App.

En los anexos se adjuntará todo el código fuente.

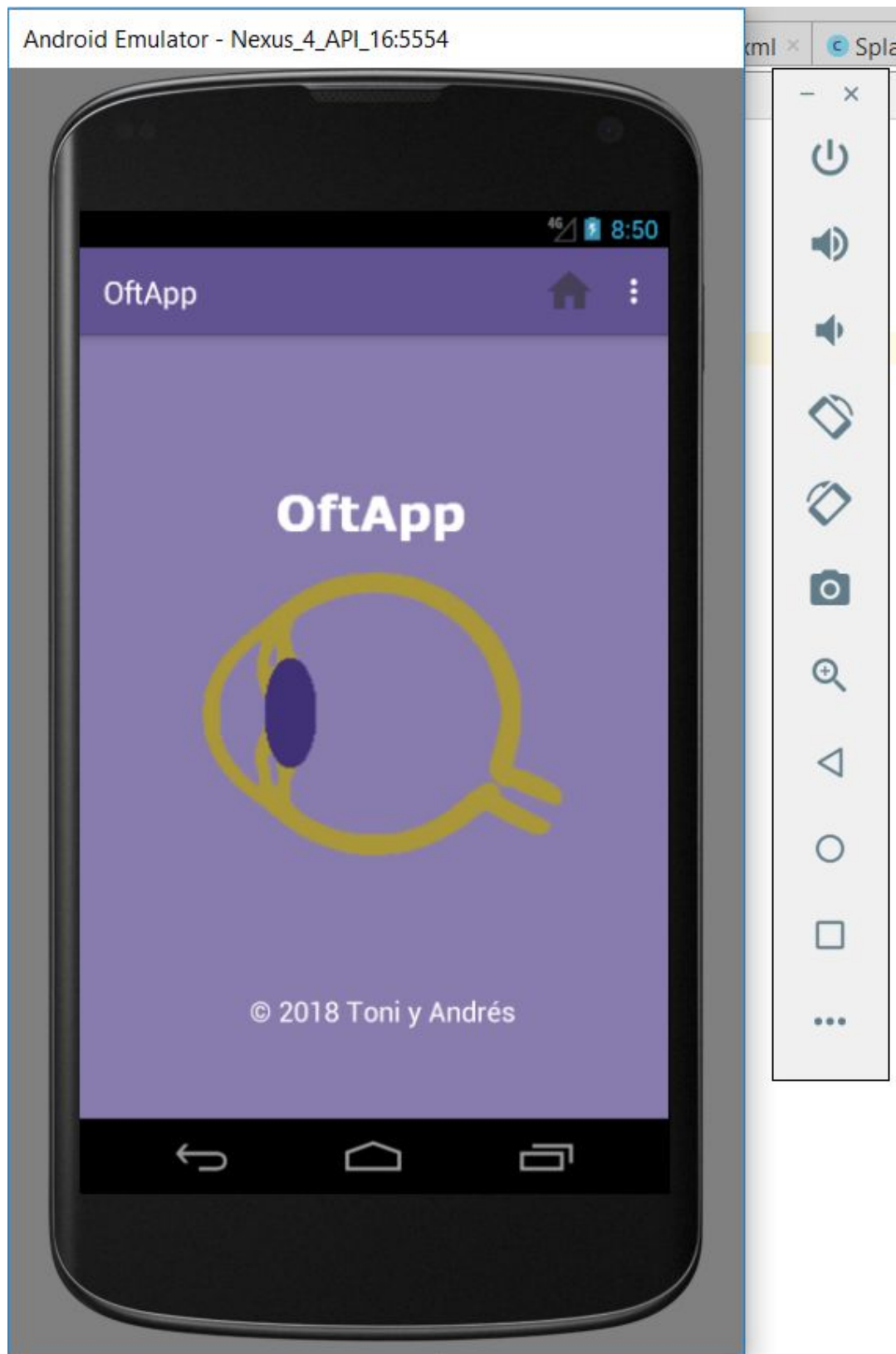
3. Probando la aplicación

En este apartado mostraremos cómo funciona nuestra aplicación a través de capturas de pantalla.

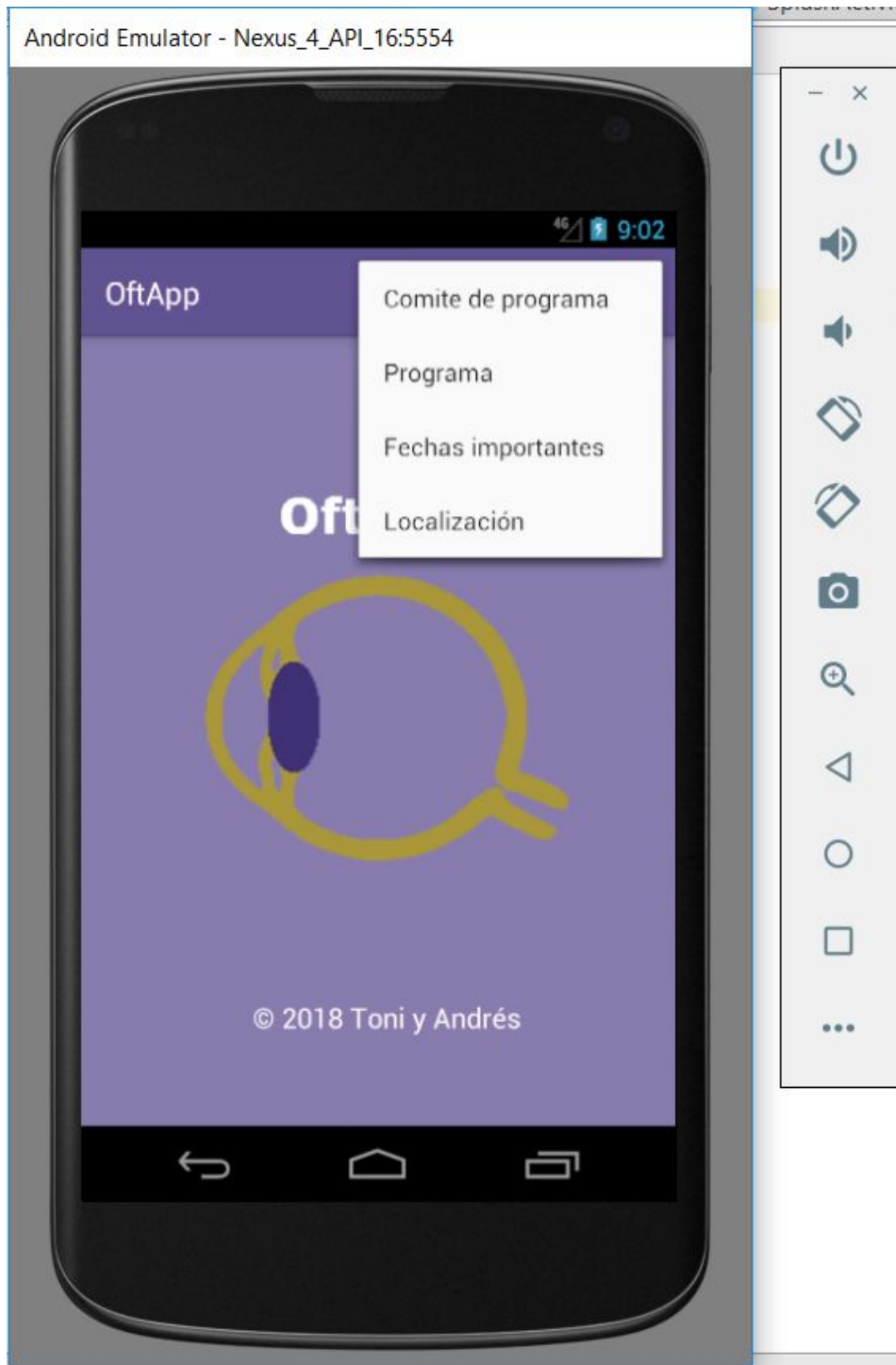
1. SplashScreen: vista de entrada a la aplicación android “*OftApp*”.



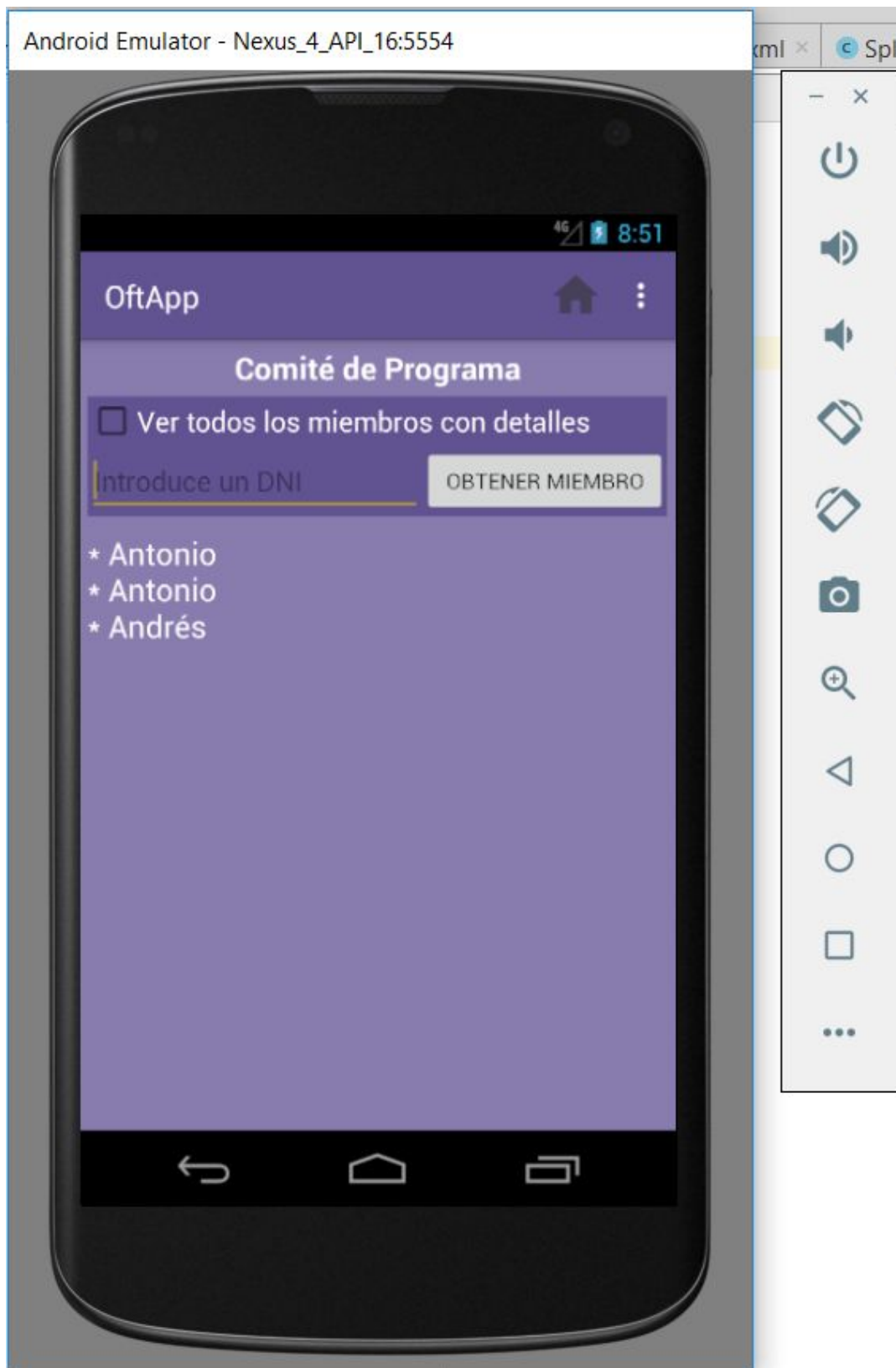
2. Página principal o de inicio de la aplicación.



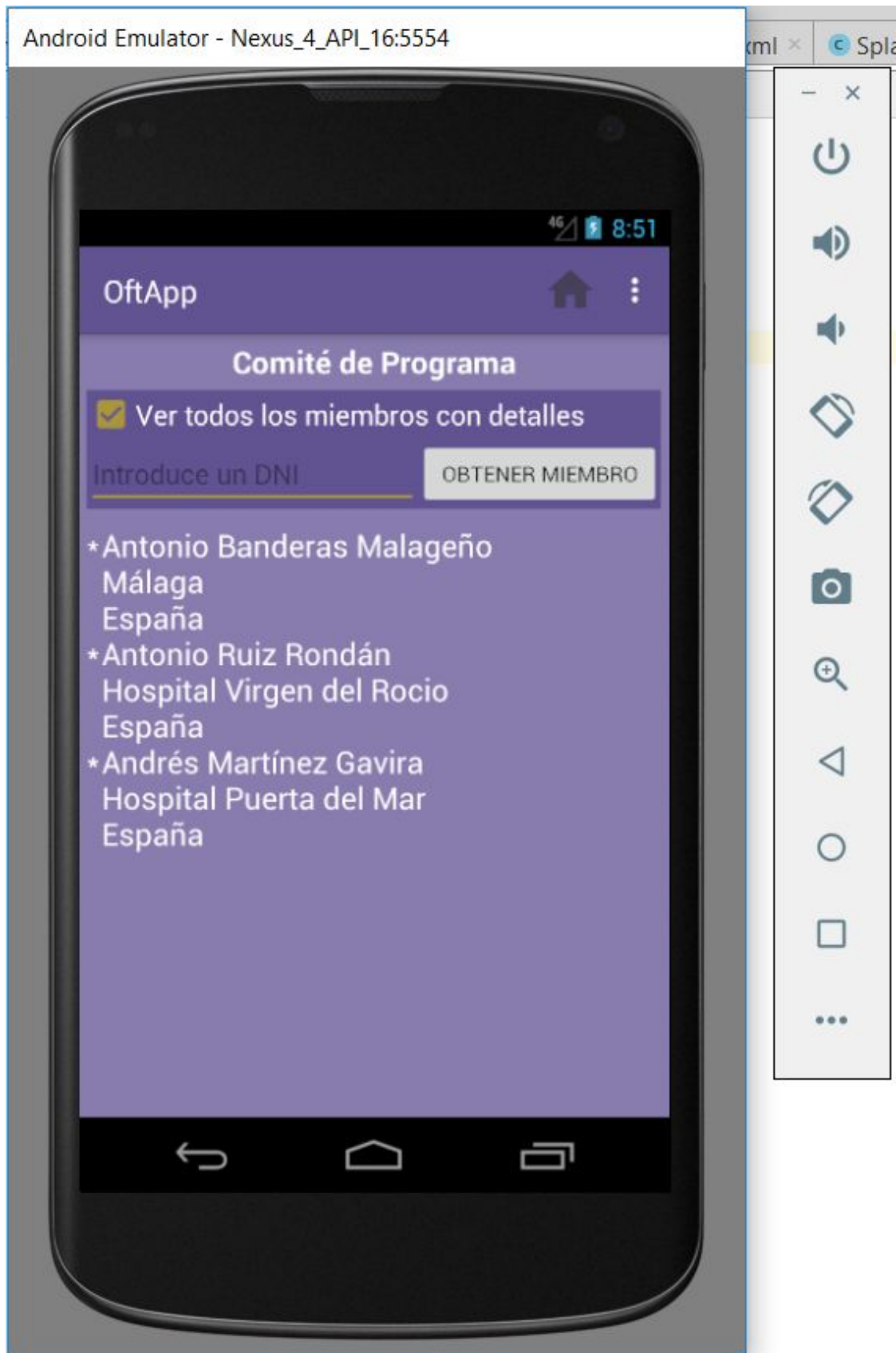
3. Opciones del menú de overflow.



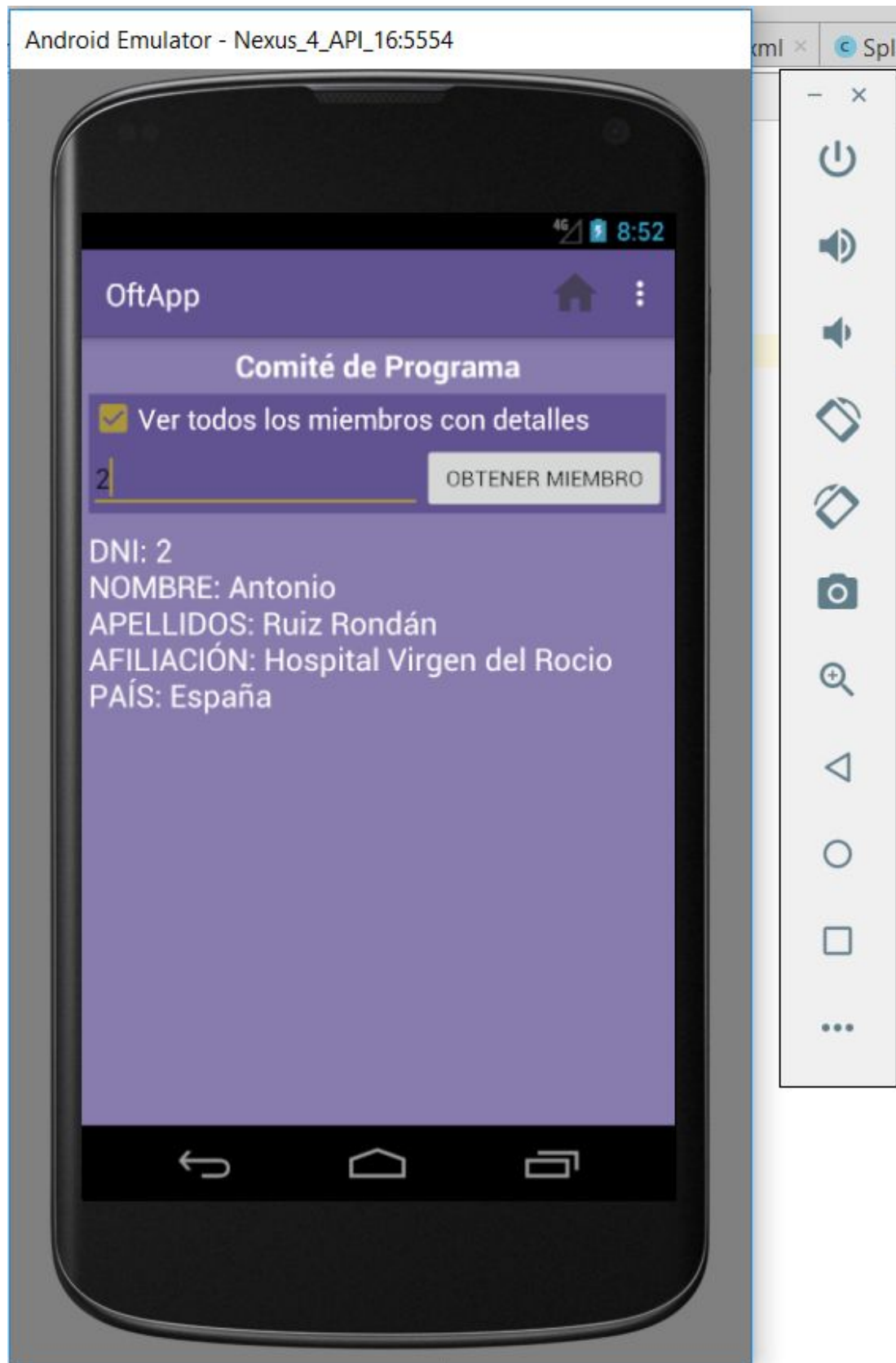
4. Vista correspondiente al comité del programa:
 - 4.1. Mostrando sólo los nombres de los miembros del comité.



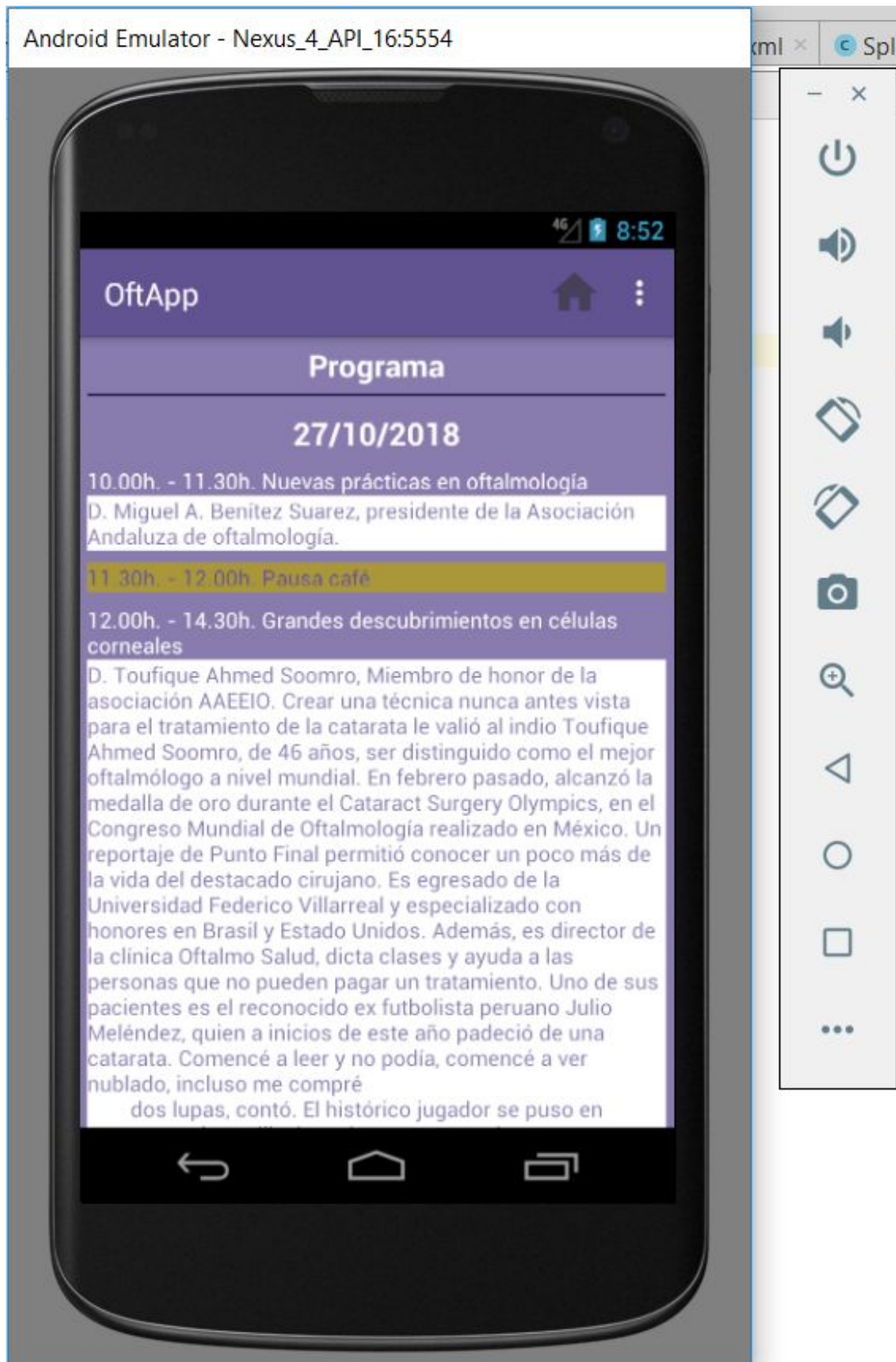
4.2. Mostrando además sus apellidos, afiliación y país.



4.3. Consultando por DNI como criterio de filtro de búsqueda.

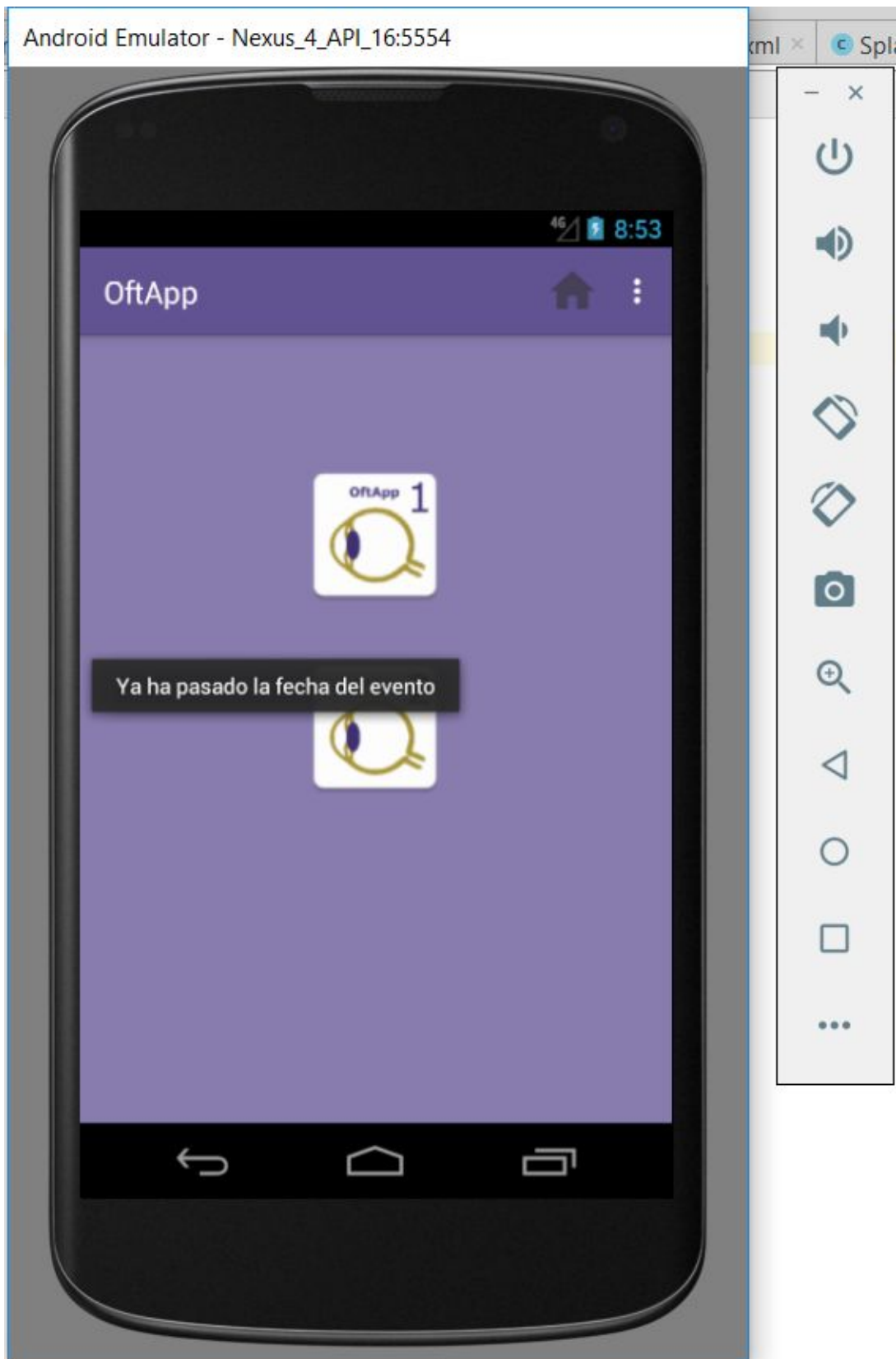


5. Vista del programa.



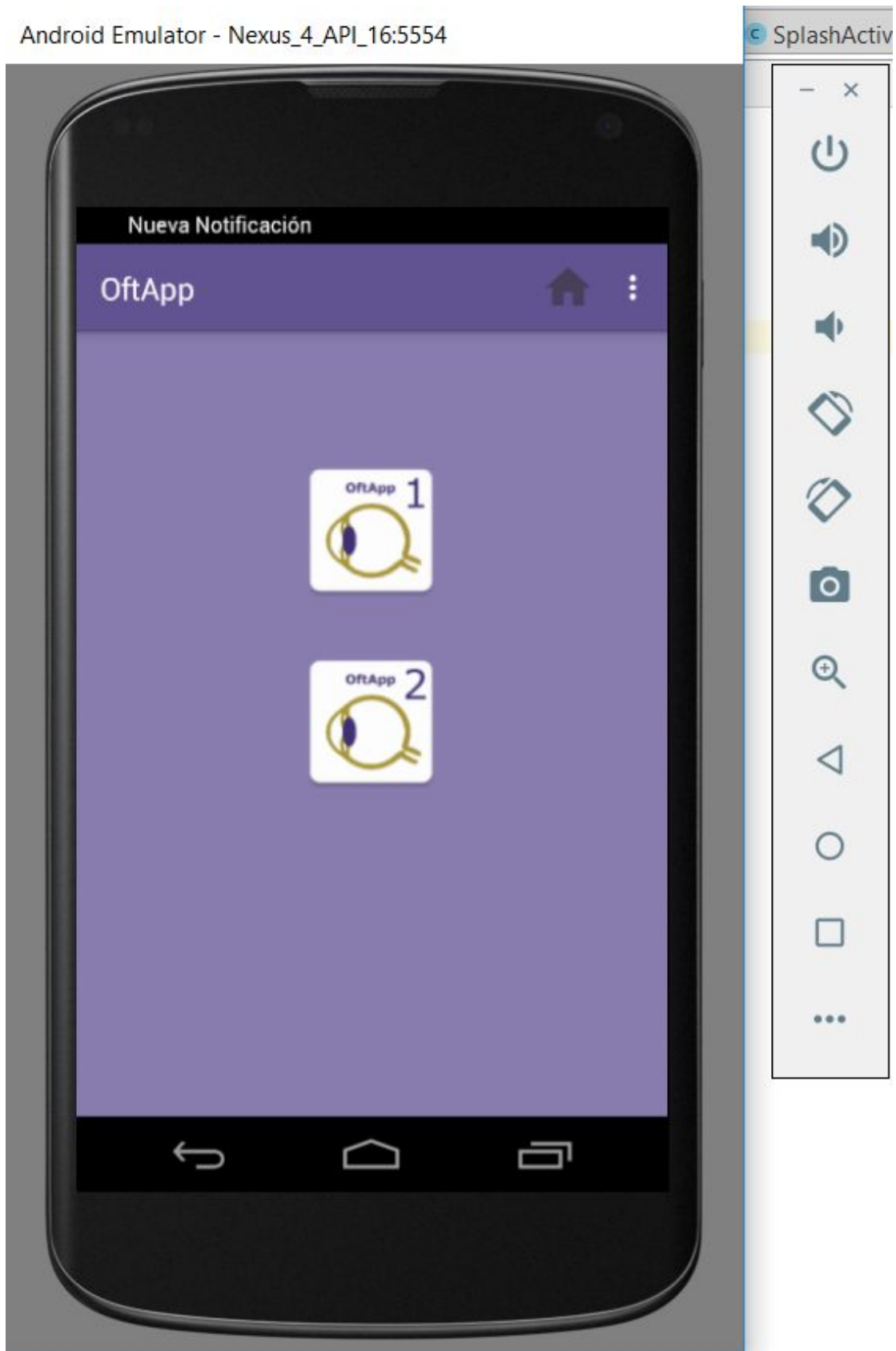
6. Vista de fechas importantes:

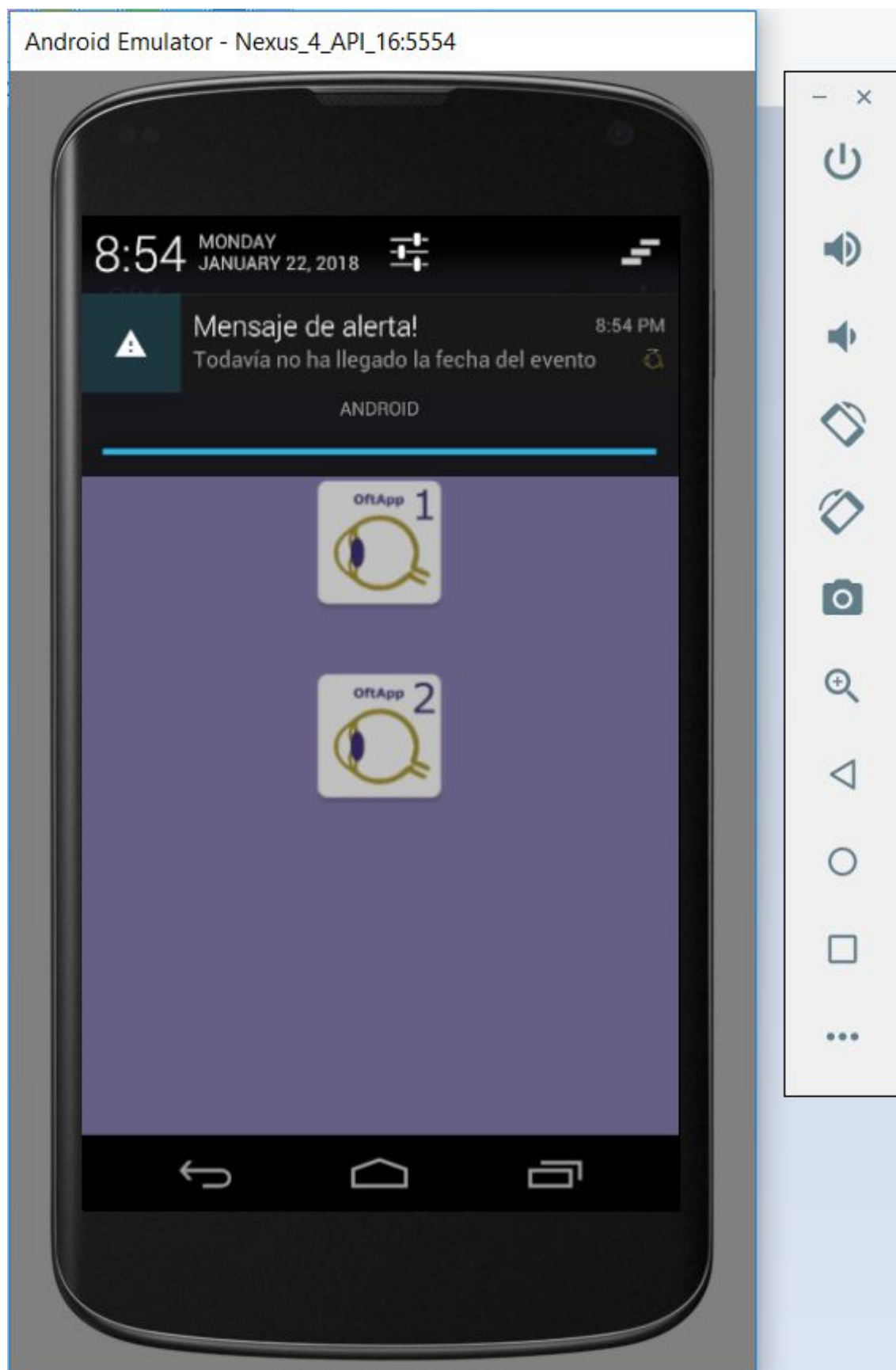
6.1. Fechas pasadas.



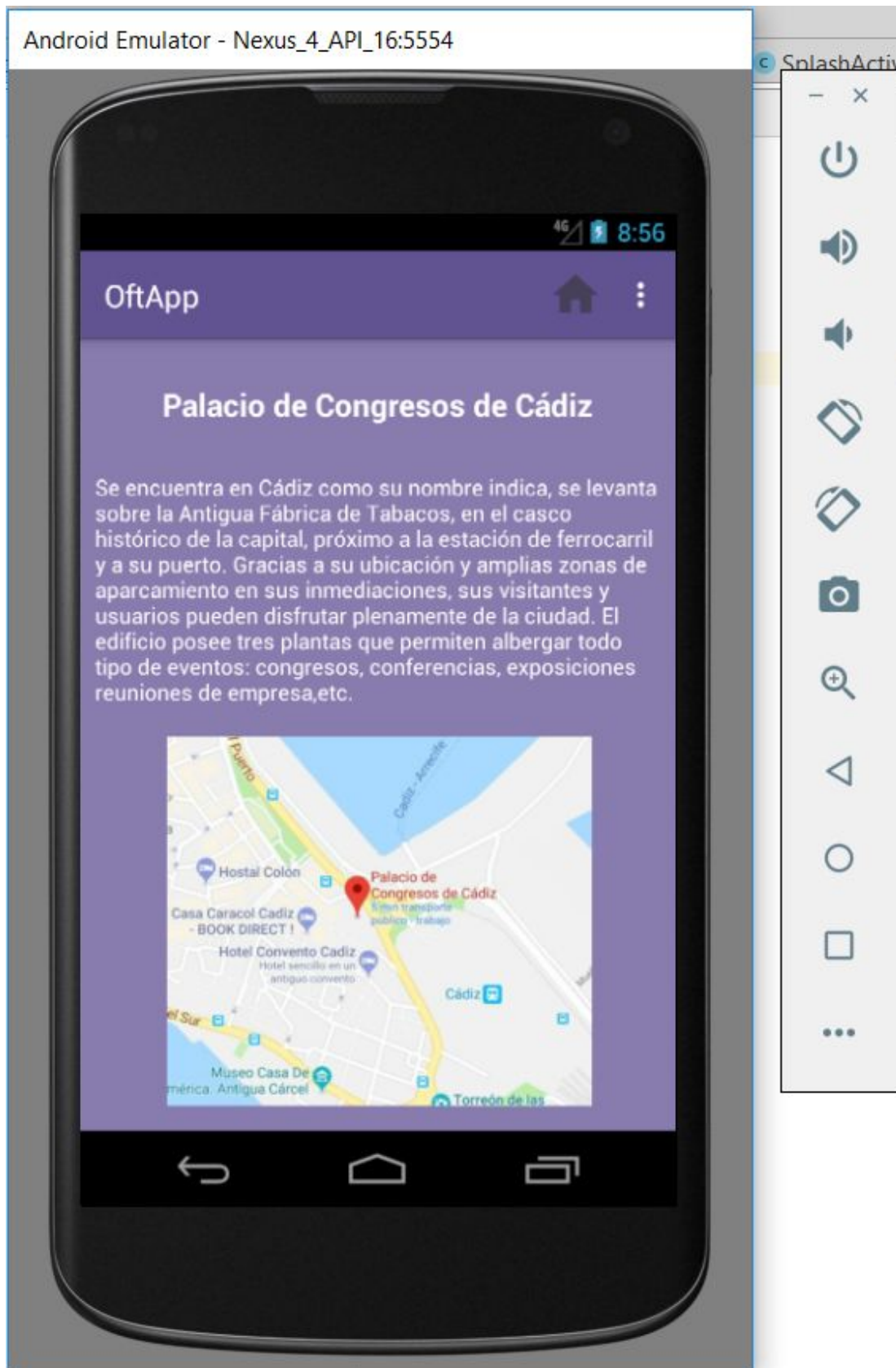
6.2. Fechas por llegar.

Android Emulator - Nexus_4_API_16:5554

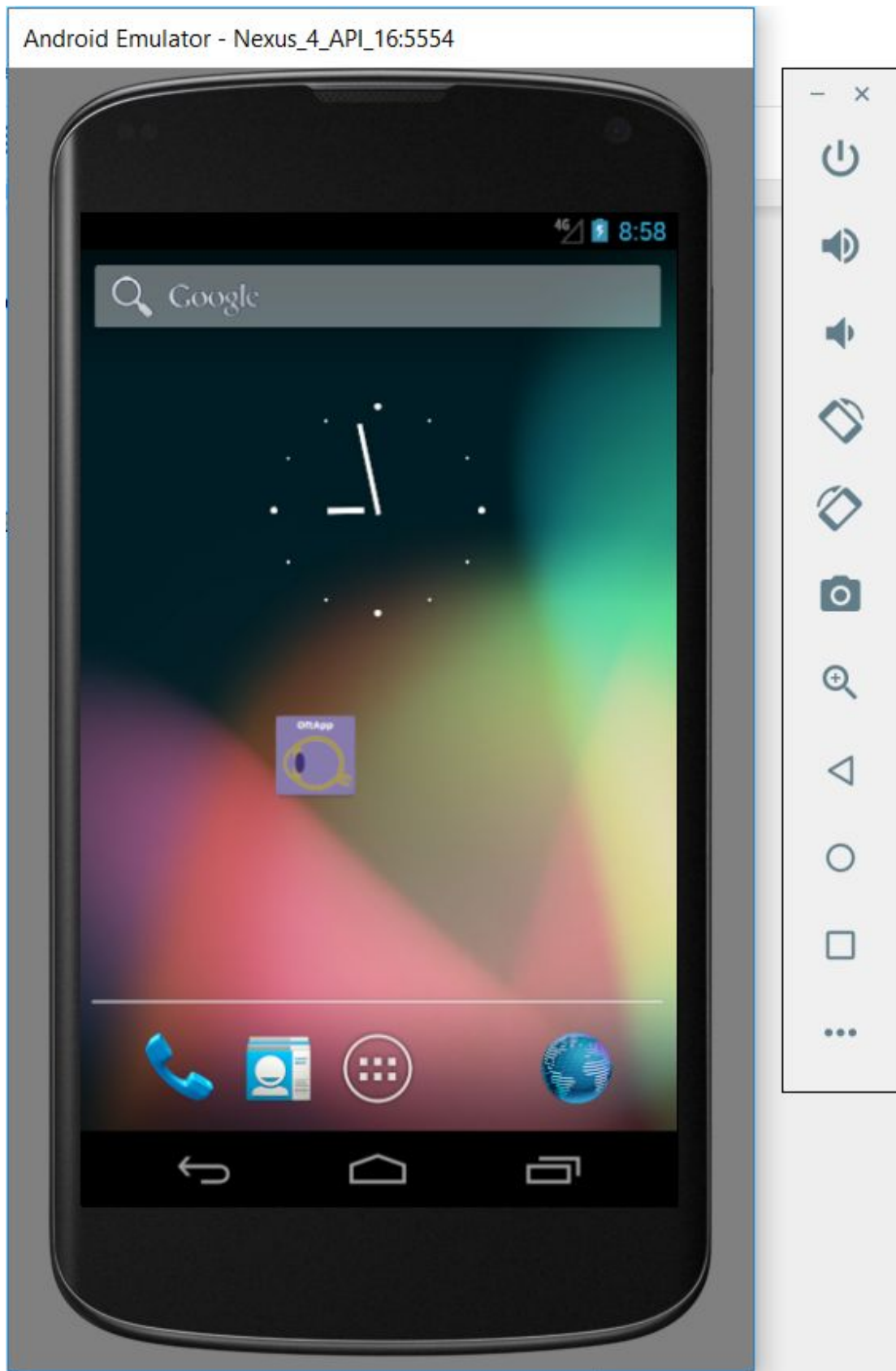




7. Vista de localización.



8. Vista del widget.



Bibliografía

- [1] Eclipse Neon 1, <http://www.eclipse.org/home/index.php>
- [2] Apache Tomcat 8.0.47, <https://tomcat.apache.org/download-80.cgi>
- [3] Android Studio 2.3.2, <https://developer.android.com/studio/index.html>
- [4] Oracle Java JDK & JRE, <https://www.oracle.com/es/java/index.html>
- [5] WampServer, <http://www.wampserver.com/en/>

Anexos

AndroidManifest.xml

```
1.  <?xml version="1.0" encoding="utf-8"?>
2.  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3.    package="es.uca.oftapp">
4.
5.    <uses-permission android:name="android.permission.INTERNET" />
6.    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
7.
8.    <application
9.        android:allowBackup="true"
10.       android:icon="@mipmap/ic_logo"
11.       android:label="@string/app_name"
12.       android:roundIcon="@mipmap/ic_launcher_round"
13.       android:supportsRtl="true"
14.       android:theme="@style/AppTheme">
15.        <activity
16.            android:name=".SplashActivity"
17.            android:theme="@style/AppTheme.NoActionBar">
18.            >
19.            <intent-filter>
20.                <action android:name="android.intent.action.MAIN" />
21.
22.                <category android:name="android.intent.category.LAUNCHER" />
23.            </intent-filter>
24.        </activity>
25.        <activity android:name=".ComiteActivity" />
26.        <activity android:name=".ProgramaActivity" />
27.        <activity android:name=".FechasActivity" />
28.        <activity android:name=".LocalizacionActivity" />
29.        <activity android:name=".MainActivity" />
30.
31.        <receiver android:name=".NuestroWidget">
32.            <intent-filter>
33.                <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
34.            </intent-filter>
35.
36.            <meta-data
37.                android:name="android.appwidget.provider"
38.                android:resource="@xml/nuestro_widget_info" />
39.        </receiver>
40.    </application>
41.
42. </manifest>
```

ComiteActivity.java

```
1. package es.uca.oftapp;
2.
3. import android.content.Intent;
4. import android.support.v7.app.AppCompatActivity;
5. import android.os.Bundle;
6. import android.view.Menu;
7. import android.view.MenuItem;
8. import android.view.View;
9. import android.widget.Button;
10. import android.widget.CheckBox;
11. import android.widget.EditText;
12. import android.widget.TextView;
13.
14. import org.json.JSONException;
15. import org.json.JSONObject;
16.
17. import java.util.Iterator;
18. import java.util.concurrent.ExecutionException;
19.
20. public class ComiteActivity extends AppCompatActivity {
21.
22.     private TextView txt1;
23.     private CheckBox chk1;
24.     private EditText edt1;
25.     private Button btn1;
26.
27.     @Override
28.     protected void onCreate(Bundle savedInstanceState) {
29.         super.onCreate(savedInstanceState);
30.         setContentView(R.layout.activity_comite);
31.
32.         txt1 = (TextView) findViewById(R.id.txt1);
33.         chk1 = (CheckBox) findViewById(R.id.chk1);
34.         edt1 = (EditText) findViewById(R.id.edt1);
35.         btn1 = (Button) findViewById(R.id.btn1);
36.
37.         //Some url endpoint that you may have
38.         final String urlTodosPonentes =
39.             "http://10.0.2.2:8080/ServicioWeb/oftalmologia/todosPonentes";
40.         final String urlUnPonente =
41.             "http://10.0.2.2:8080/ServicioWeb/oftalmologia/obtenerPonente/";
42.         //String to place our result in
43.         String resultados = getResultados(urlTodosPonentes);
44.         imprimirSinDetalles(resultados);
45.
46.         chk1.setOnClickListener(new View.OnClickListener() {
47.             @Override
48.             public void onClick(View v) {
49.                 String resultados = getResultados(urlTodosPonentes);
50.
51.                 if(chk1.isChecked()) {
52.                     if (resultados != null) {
53.                         JSONObject a ;
54.                         StringBuilder nombre = new StringBuilder();
55.                         try {
56.                             a = new JSONObject (resultados) ;
57.                             Iterator<String> iter = a.keys();
58.                             while (iter.hasNext()) {
```

```
57.         String key = iter.next();
58.         JSONObject value = a.getJSONObject(key);
59.
60.         nombre.append("*\t").append(value.getString("nombre")).append(" ")
61.         .append(value.getString("apellidos")).append("\n\t")
62.         .append(value.getString("afiliacion")).append("\n\t")
63.         .append(value.getString("pais")).append("\n");
64.         txt1.setText(nombre);
65.     } catch (JSONException e) {
66.         e.printStackTrace();
67.     }
68. }
69. } else {
70.     imprimirSinDetalles(resultados);
71. }
72. }
73. });
74.
75.
76. btn1.setOnClickListener(new View.OnClickListener() {
77.     @Override
78.     public void onClick(View v) {
79.         String resultado = getResultados(urlUnPonente + edt1.getText());
80.         String eliminar1, eliminar2, eliminar3;
81.         eliminar1 = "</div>";
82.         eliminar2 = "<div style='color: red; font-weight: bold;'>";
83.         eliminar3 = "<div style='color: orange; font-weight: bold;'>";
84.         resultado = resultado.replaceAll(eliminar1, "")
85.             .replaceAll(eliminar2, "")
86.             .replaceAll(eliminar3, "")
87.             .replaceAll("NOMBRE", "\nNOMBRE")
88.             .replaceAll("APELLIDOS", "\nAPELLIDOS")
89.             .replaceAll("AFILIACIÓN", "\nAFILIACIÓN")
90.             .replaceAll("PAÍS", "\nPAÍS");
91.         txt1.setText(resultado);
92.     }
93. });
94.
95.
96.
97. }
98.
99. private void imprimirSinDetalles(String resultados) {
100.     if (resultados != null) {
101.         JSONObject a;
102.         StringBuilder nombre = new StringBuilder();
103.         try {
104.             a = new JSONObject(resultados);
105.             Iterator<String> iter = a.keys();
106.             while (iter.hasNext()) {
107.                 String key = iter.next();
108.                 JSONObject value = a.getJSONObject(key);
109.                 nombre.append("*
110.                 ").append(value.getString("nombre")).append("\n");
111.                 txt1.setText(nombre);
112.             } catch (JSONException e) {
113.                 e.printStackTrace();
114.             }
```

```
115.     }
116. }
117.
118.
119. private String getResultados(String URL) {
120.     String r = null;
121.
122.     ComiteREST getRequest = new ComiteREST();
123.     //Perform the doInBackground method, passing in our url
124.     try {
125.         r = getRequest.execute(URL).get();
126.     } catch (InterruptedException e) {
127.         e.printStackTrace();
128.         r = null;
129.     } catch (ExecutionException e) {
130.         e.printStackTrace();
131.         r = null;
132.     }
133.
134.     return r;
135. }
136.
137.
138. @Override
139. public boolean onCreateOptionsMenu(Menu menu) {
140.     // Inflate the menu; this adds items to the action bar if it is present.
141.     getMenuInflater().inflate(R.menu.menu_main, menu);
142.     return true;
143. }
144.
145.
146. @Override
147. public boolean onOptionsItemSelected(MenuItem item) {
148.     // Handle action bar item clicks here. The action bar will
149.     // automatically handle clicks on the Home/Up button, so long
150.     // as you specify a parent activity in AndroidManifest.xml.
151.     int id = item.getItemId();
152.
153.     //noinspection SimplifiableIfStatement
154.     switch (id) {
155.         case R.id.action_comite:
156.             Intent intentComite = new Intent(ComiteActivity.this,
ComiteActivity.class);
157.             startActivity(intentComite);
158.             return true;
159.         case R.id.action_programa:
160.             Intent intentPrograma = new Intent(ComiteActivity.this,
ProgramaActivity.class);
161.             startActivity(intentPrograma);
162.             return true;
163.         case R.id.action_fechas:
164.             Intent intentFechas = new Intent(ComiteActivity.this,
FechasActivity.class);
165.             startActivity(intentFechas);
166.             return true;
167.         case R.id.action_localizacion:
168.             Intent intentLocalizacion = new Intent(ComiteActivity.this,
LocalizacionActivity.class);
169.             startActivity(intentLocalizacion);
170.             return true;
171.         case R.id.action_inicio:
172.             Intent intentMain = new Intent(ComiteActivity.this,
```



```
MainActivity.class);
173.         startActivity(intentMain);
174.         return true;
175.     default:
176.         return super.onOptionsItemSelected(item);
177.     }
178. }
179. }
```

ComiteREST.java

```
1. package es.uca.oftapp;
2.
3. import android.content.Context;
4. import android.os.AsyncTask;
5. import android.widget.TextView;
6.
7. import java.io.BufferedReader;
8. import java.io.IOException;
9. import java.io.InputStreamReader;
10. import java.net.HttpURLConnection;
11. import java.net.MalformedURLException;
12. import java.net.URL;
13.
14.
15. public class ComiteREST extends AsyncTask<String, Void, String> {
16.
17.     static final String REQUEST_METHOD = "GET";
18.     public static final int READ_TIMEOUT = 15000;
19.     public static final int CONNECTION_TIMEOUT = 15000;
20.
21.
22.     @Override
23.     protected String doInBackground(String... params) {
24.         String urlString = params[0];
25.         String result;
26.         String inputLine;
27.         HttpURLConnection connection = null;
28.
29.         try {
30.             //Create a URL object holding our url
31.             URL myUrl = new URL(urlString);
32.             //Create a connection
33.             connection =(HttpURLConnection) myUrl.openConnection();
34.             //Set methods and timeouts
35.             connection.setRequestMethod(REQUEST_METHOD);
36.             connection.setReadTimeout(READ_TIMEOUT);
37.             connection.setConnectTimeout(CONNECTION_TIMEOUT);
38.
39.             //Connect to our url
40.             connection.connect();
41.             //Create a new InputStreamReader
42.             InputStreamReader streamReader = new
43.                 InputStreamReader(connection.getInputStream());
44.             //Create a new buffered reader and String Builder
45.             BufferedReader reader = new BufferedReader(streamReader);
46.             StringBuilder stringBuilder = new StringBuilder();
47.             //Check if the line we are reading is not null
```

```
48.         while((inputLine = reader.readLine()) != null){
49.             stringBuilder.append(inputLine);
50.         }
51.         //Close our InputStream and Buffered reader
52.         reader.close();
53.         streamReader.close();
54.         //Set our result equal to our stringBuilder
55.         result = stringBuilder.toString();
56.     } catch (MalformedURLException e) {
57.         e.printStackTrace();
58.         result = null;
59.     } catch (IOException e) {
60.         e.printStackTrace();
61.         result = null;
62.     } finally {
63.         if (connection != null)
64.             connection.disconnect();
65.     }
66.
67.     return result;
68. }
69.
70. @Override
71. protected void onPostExecute(String results){
72.     super.onPostExecute(results);
73. }
74. }
```

FechasActivity.java

```
1. package es.uca.oftapp;
2.
3. import android.app.NotificationManager;
4. import android.app.PendingIntent;
5. import android.content.Context;
6. import android.content.Intent;
7. import android.graphics.BitmapFactory;
8. import android.support.v4.app.NotificationCompat;
9. import android.support.v7.app.AppCompatActivity;
10. import android.os.Bundle;
11. import android.view.Gravity;
12. import android.view.Menu;
13. import android.view.MenuItem;
14. import android.view.View;
15. import android.widget.ImageButton;
16. import android.widget.Toast;
17.
18. import java.util.Calendar;
19.
20. public class FechasActivity extends AppCompatActivity {
21.
22.     private ImageButton btnEdicion1, btnEdicion2;
23.     private static final int NOTIFIC_ALERTA_ID = 1;
24.
25.     private Calendar ahora = Calendar.getInstance();
26.     private Calendar Edicion1 = Calendar.getInstance();
27.     private Calendar Edicion2 = Calendar.getInstance();
28. }
```

```
29.     public FechasActivity() {
30.         Edicion1.set(2017, 10, 27);
31.         Edicion2.set(2018, 10, 27);
32.     }
33.
34.
35.     @Override
36.     protected void onCreate(Bundle savedInstanceState) {
37.         super.onCreate(savedInstanceState);
38.         setContentView(R.layout.activity_fechas);
39.
40.         btnEdicion1 = (ImageButton) findViewById(R.id.btnEdicion1);
41.         btnEdicion2 = (ImageButton) findViewById(R.id.btnEdicion2);
42.
43.         btnEdicion1.setOnClickListener(new View.OnClickListener() {
44.             @Override
45.             public void onClick(View v) {
46.                 if (ahora.compareTo(Edicion1) < 0) {
47.                     NotificationCompat.Builder notificacion = new
NotificationCompat.Builder(FechasActivity.this)
48.                         .setSmallIcon(R.drawable.logo)
49.                         .setLargeIcon(BitmapFactory.decodeResource(getResources(),
R.drawable.ic_stat_warning))
50.
51.                         .setContentTitle(getResources().getString(R.string.mensaje_alerta))
52.                         .setContentText(getResources().getString(R.string.todavia_no))
53.                         .setTicker(getResources().getString(R.string.nueva_notif));
54.                     Intent intent = new Intent(FechasActivity.this,
LocalizacionActivity.class);
55.                     PendingIntent pIntent =
PendingIntent.getActivity(FechasActivity.this, 0, intent, 0);
56.                     notificacion.setContentIntent(pIntent); //Asociamos el PendingIntent a
La notificación
57.                     NotificationManager mNotMan = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
58.                     mNotMan.notify(NOTIFIC_ALERTA_ID, notificacion.build()); //Genera La
notificación
59.                 } else {
60.                     Toast toast1 = Toast.makeText(getApplicationContext(),
61.                         getResources().getString(R.string.ya_paso),
Toast.LENGTH_SHORT);
62.                     toast1.setGravity(Gravity.CENTER | Gravity.LEFT, 0, 0);
63.                     toast1.show();
64.                 }
65.             }
66.         });
67.
68.
69.         btnEdicion2.setOnClickListener(new View.OnClickListener() {
70.             @Override
71.             public void onClick(View v) {
72.                 if (ahora.compareTo(Edicion2) < 0) {
73.                     NotificationCompat.Builder notificacion = new
NotificationCompat.Builder(FechasActivity.this)
74.                         .setSmallIcon(R.drawable.logo)
75.                         .setLargeIcon(BitmapFactory.decodeResource(getResources(),
R.drawable.ic_stat_warning))
76.
77.                         .setContentTitle(getResources().getString(R.string.mensaje_alerta))
```

```

78.         .setContentText(getResources().getString(R.string.todavia_no))
79.         .setTicker(getResources().getString(R.string.nueva_notif));
80.         Intent intent = new Intent(FechasActivity.this,
LocalizacionActivity.class);
81.         PendingIntent pIntent =
PendingIntent.getActivity(FechasActivity.this, 0, intent, 0);
82.         notificacion.setContentIntent(pIntent); //Asociamos el PendingIntent a
la notificación
83.         NotificationManager mNotMan = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
84.         mNotMan.notify(NOTIFIC_ALERTA_ID, notificacion.build()); //Genera la
notificación
85.     } else {
86.         Toast toast1 = Toast.makeText(getApplicationContext(),
87.         getResources().getString(R.string.ya_paso),
Toast.LENGTH_SHORT);
88.         toast1.setGravity(Gravity.CENTER | Gravity.LEFT, 0, 0);
89.         toast1.show();
90.     }
91. }
92. });
93. }
94.
95.
96. @Override
97. public boolean onCreateOptionsMenu(Menu menu) {
98.     // Inflate the menu; this adds items to the action bar if it is present.
99.     getMenuInflater().inflate(R.menu.menu_main, menu);
100.    return true;
101. }
102.
103.
104. @Override
105. public boolean onOptionsItemSelected(MenuItem item) {
106.     // Handle action bar item clicks here. The action bar will
107.     // automatically handle clicks on the Home/Up button, so long
108.     // as you specify a parent activity in AndroidManifest.xml.
109.     int id = item.getItemId();
110.
111.     //noinspection SimplifiableIfStatement
112.     switch (id) {
113.         case R.id.action_comite:
114.             Intent intentComite = new Intent(FechasActivity.this,
ComiteActivity.class);
115.             startActivity(intentComite);
116.             return true;
117.         case R.id.action_programa:
118.             Intent intentPrograma = new Intent(FechasActivity.this,
ProgramaActivity.class);
119.             startActivity(intentPrograma);
120.             return true;
121.         case R.id.action_fechas:
122.             Intent intentFechas = new Intent(FechasActivity.this,
FechasActivity.class);
123.             startActivity(intentFechas);
124.             return true;
125.         case R.id.action_localizacion:
126.             Intent intentLocalizacion = new Intent(FechasActivity.this,
LocalizacionActivity.class);
127.             startActivity(intentLocalizacion);
128.             return true;

```

```
129.         case R.id.action_inicio:
130.             Intent intentMain = new Intent(FechasActivity.this,
131.                 MainActivity.class);
132.             startActivity(intentMain);
133.             return true;
134.         default:
135.             return super.onOptionsItemSelected(item);
136.     }
137. }
```

LocalizaciónActivity.java

```
1. package es.uca.oftapp;
2.
3. import android.content.Intent;
4. import android.support.v7.app.AppCompatActivity;
5. import android.os.Bundle;
6. import android.view.Menu;
7. import android.view.MenuItem;
8.
9. public class LocalizacionActivity extends AppCompatActivity {
10.
11.     @Override
12.     protected void onCreate(Bundle savedInstanceState) {
13.         super.onCreate(savedInstanceState);
14.         setContentView(R.layout.activity_localizacion);
15.     }
16.
17.
18.     @Override
19.     public boolean onCreateOptionsMenu(Menu menu) {
20.         // Inflate the menu; this adds items to the action bar if it is present.
21.         getMenuInflater().inflate(R.menu.menu_main, menu);
22.         return true;
23.     }
24.
25.
26.     @Override
27.     public boolean onOptionsItemSelected(MenuItem item) {
28.         // Handle action bar item clicks here. The action bar will
29.         // automatically handle clicks on the Home/Up button, so long
30.         // as you specify a parent activity in AndroidManifest.xml.
31.         int id = item.getItemId();
32.
33.         //noinspection SimplifiableIfStatement
34.         switch (id) {
35.             case R.id.action_comite:
36.                 Intent intentComite = new Intent(LocalizacionActivity.this,
37.                     ComiteActivity.class);
38.                 startActivity(intentComite);
39.                 return true;
40.             case R.id.action_programa:
41.                 Intent intentPrograma = new Intent(LocalizacionActivity.this,
42.                     ProgramaActivity.class);
43.                 startActivity(intentPrograma);
44.                 return true;
45.         }
46.         return false;
47.     }
48. }
```

```
43.         case R.id.action_fechas:
44.             Intent intentFechas = new Intent(LocalizacionActivity.this,
FechasActivity.class);
45.             startActivity(intentFechas);
46.             return true;
47.         case R.id.action_localizacion:
48.             Intent intentLocalizacion = new Intent(LocalizacionActivity.this,
LocalizacionActivity.class);
49.             startActivity(intentLocalizacion);
50.             return true;
51.         case R.id.action_inicio:
52.             Intent intentMain = new Intent(LocalizacionActivity.this,
MainActivity.class);
53.             startActivity(intentMain);
54.             return true;
55.         default:
56.             return super.onOptionsItemSelected(item);
57.     }
58. }
59. }
```

MainActivity.java

```
1. package es.uca.oftapp;
2.
3. import android.content.Intent;
4. import android.support.v7.app.AppCompatActivity;
5. import android.os.Bundle;
6. import android.view.Menu;
7. import android.view.MenuItem;
8.
9. public class MainActivity extends AppCompatActivity {
10.
11.     @Override
12.     protected void onCreate(Bundle savedInstanceState) {
13.         super.onCreate(savedInstanceState);
14.         setContentView(R.layout.activity_main);
15.     }
16.
17.
18.     @Override
19.     public boolean onCreateOptionsMenu(Menu menu) {
20.         // Inflate the menu; this adds items to the action bar if it is present.
21.         getMenuInflater().inflate(R.menu.menu_main, menu);
22.         return true;
23.     }
24.
25.
26.     @Override
27.     public boolean onOptionsItemSelected(MenuItem item) {
28.         // Handle action bar item clicks here. The action bar will
29.         // automatically handle clicks on the Home/Up button, so long
30.         // as you specify a parent activity in AndroidManifest.xml.
31.         int id = item.getItemId();
32.
33.         //noinspection SimplifiableIfStatement
34.         switch (id) {
```

```
35.         case R.id.action_comite:
36.             Intent intentComite = new Intent(MainActivity.this,
ComiteActivity.class);
37.             startActivity(intentComite);
38.             return true;
39.         case R.id.action_programa:
40.             Intent intentPrograma = new Intent(MainActivity.this,
ProgramaActivity.class);
41.             startActivity(intentPrograma);
42.             return true;
43.         case R.id.action_fechas:
44.             Intent intentFechas = new Intent(MainActivity.this,
FechasActivity.class);
45.             startActivity(intentFechas);
46.             return true;
47.         case R.id.action_localizacion:
48.             Intent intentLocalizacion = new Intent(MainActivity.this,
LocalizacionActivity.class);
49.             startActivity(intentLocalizacion);
50.             return true;
51.         case R.id.action_inicio:
52.             Intent intentMain = new Intent(MainActivity.this, MainActivity.class);
53.             startActivity(intentMain);
54.             return true;
55.         default:
56.             return super.onOptionsItemSelected(item);
57.     }
58. }
59. }
```

NuestroWidget.java

```
1. package es.uca.oftapp;
2.
3. import android.app.PendingIntent;
4. import android.appwidget.AppWidgetManager;
5. import android.appwidget.AppWidgetProvider;
6. import android.content.Context;
7. import android.content.Intent;
8. import android.graphics.drawable.Drawable;
9. import android.media.Image;
10. import android.widget.RemoteViews;
11.
12. /**
13.  * Implementation of App Widget functionality.
14.  */
15. public class NuestroWidget extends AppWidgetProvider {
16.
17.     static void updateAppWidget(Context context, AppWidgetManager appWidgetManager,
18.         int appWidgetId) {
19.
20.         CharSequence widgetText = context.getString(R.string.appwidget_text);
21.
22.         // Construct the RemoteViews object
23.         RemoteViews views = new RemoteViews(context.getPackageName(),
R.layout.nuestro_widget);
24.         views.setTextViewText(R.id.appwidget_image, widgetText);
```

```
25.
26.         // Instruct the widget manager to update the widget
27.         appWidgetManager.updateAppWidget(appWidgetId, views);
28.     }
29.
30.     @Override
31.     public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[]
appWidgetIds) {
32.         // There may be multiple widgets active, so update all of them
33.         for (int appWidgetId : appWidgetIds) {
34.             //updateAppWidget(context, appWidgetManager, appWidgetId);
35.             Intent intent = new Intent(context, ProgramaActivity.class);
36.             PendingIntent pIntent = PendingIntent.getActivity(context, 0, intent, 0);
37.             //establece un canal de comunicacion entre nuestra aplicacion y el widget
38.             RemoteViews views = new RemoteViews(context.getPackageName(),
R.layout.nuestro_widget);
39.             //cuando se realice un click sobre el widget se lance el Intent
40.             views.setOnClickPendingIntent(R.id.appwidget_image, pIntent);
41.             //actualizamos nuestro widget
42.             appWidgetManager.updateAppWidget(appWidgetId, views);
43.         }
44.     }
45.
46.     @Override
47.     public void onEnabled(Context context) {
48.         // Enter relevant functionality for when the first widget is created
49.     }
50.
51.     @Override
52.     public void onDisabled(Context context) {
53.         // Enter relevant functionality for when the last widget is disabled
54.     }
55. }
```

ProgramaActivity.java

```
1. package es.uca.oftapp;
2.
3. import android.content.Intent;
4. import android.os.Bundle;
5. import android.support.v7.app.AppCompatActivity;
6. import android.view.Menu;
7. import android.view.MenuItem;
8.
9.
10. public class ProgramaActivity extends AppCompatActivity {
11.
12.     @Override
13.     protected void onCreate(Bundle savedInstanceState) {
14.         super.onCreate(savedInstanceState);
15.         setContentView(R.layout.activity_programa);
16.     }
17.
18.
19.     @Override
20.     public boolean onCreateOptionsMenu(Menu menu) {
21.         // Inflate the menu; this adds items to the action bar if it is present.
22.         getMenuInflater().inflate(R.menu.menu_main, menu);
23.     }
24. }
```



```
23.         return true;
24.     }
25.
26.
27.     @Override
28.     public boolean onOptionsItemSelected(MenuItem item) {
29.         // Handle action bar item clicks here. The action bar will
30.         // automatically handle clicks on the Home/Up button, so long
31.         // as you specify a parent activity in AndroidManifest.xml.
32.         int id = item.getItemId();
33.
34.         //noinspection SimplifiableIfStatement
35.         switch (id) {
36.             case R.id.action_comite:
37.                 Intent intentComite = new Intent(ProgramaActivity.this,
ComiteActivity.class);
38.                 startActivity(intentComite);
39.                 return true;
40.             case R.id.action_programa:
41.                 Intent intentPrograma = new Intent(ProgramaActivity.this,
ProgramaActivity.class);
42.                 startActivity(intentPrograma);
43.                 return true;
44.             case R.id.action_fechas:
45.                 Intent intentFechas = new Intent(ProgramaActivity.this,
FechasActivity.class);
46.                 startActivity(intentFechas);
47.                 return true;
48.             case R.id.action_localizacion:
49.                 Intent intentLocalizacion = new Intent(ProgramaActivity.this,
LocalizacionActivity.class);
50.                 startActivity(intentLocalizacion);
51.                 return true;
52.             case R.id.action_inicio:
53.                 Intent intentMain = new Intent(ProgramaActivity.this,
MainActivity.class);
54.                 startActivity(intentMain);
55.                 return true;
56.             default:
57.                 return super.onOptionsItemSelected(item);
58.         }
59.     }
60.
61.
62. }
```

SplashActivity.java

```
1. package es.uca.oftapp;
2.
3. import android.content.Intent;
4. import android.content.pm.ActivityInfo;
5. import android.os.Handler;
6. import android.support.v7.app.AppCompatActivity;
7. import android.os.Bundle;
8. import android.view.WindowManager;
9.
10. public class SplashActivity extends AppCompatActivity {
```

```
11.
12.     private final int DURACION_SPLASH = 3000;
13.
14.     @Override
15.     protected void onCreate(Bundle savedInstanceState) {
16.         super.onCreate(savedInstanceState);
17.
18.         setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
19.         this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
20.             WindowManager.LayoutParams.FLAG_FULLSCREEN);
21.
22.         setContentView(R.layout.activity_splash);
23.
24.         new Handler().postDelayed(new Runnable(){
25.             public void run(){
26.                 Intent intent = new Intent(SplashActivity.this, MainActivity.class);
27.                 startActivity(intent);
28.                 finish();
29.             }, DURACION_SPLASH);
30.     }
31. }
```

strings.xml

```
1. <resources>
2.     <string name="app_name">OftApp</string>
3.     <string name="copy">© 2018 Toni y Andrés</string>
4.     <string name="comite">Comite de programa</string>
5.     <string name="programa">Programa</string>
6.     <string name="fechas">Fechas importantes</string>
7.     <string name="localizacion">Localización</string>
8.     <string name="inicio">Inicio</string>
9.     <string name="mensaje_alerta">Mensaje de alerta!</string>
10.    <string name="nueva_notif">Nueva Notificación</string>
11.    <string name="ya_paso">Ya ha pasado la fecha del evento</string>
12.    <string name="todavia_no">Todavía no ha llegado la fecha del evento</string>
13.    <string name="txt_titulo_palacio">Palacio de Congresos de Cádiz</string>
14.    <string name="txt_localizacion">Se encuentra en Cádiz como su nombre indica, se
15.        levanta sobre
16.        la Antigua Fábrica de Tabacos, en el casco histórico de la capital, próximo a la
17.        estación
18.        de ferrocarril y a su puerto. Gracias a su ubicación y amplias zonas de
19.        aparcamiento en sus
20.        inmediaciones, sus visitantes y usuarios pueden disfrutar plenamente de la
21.        ciudad. El edificio
22.        posee tres plantas que permiten albergar todo tipo de eventos: congresos,
23.        conferencias,
24.        exposiciones reuniones de empresa,etc.</string>
25.    <string name="txt_titulo_programa">Programa</string>
26.    <string name="txt_titulo_comite">Comité de Programa</string>
27.    <string name="dia1">27/10/2018</string>
28.    <string name="dia2">28/10/2018</string>
29.    <string name="dia3">29/10/2018</string>
30.    <string name="hora1dia1">10.00h. - 11.30h. Nuevas prácticas en oftalmología</string>
31.    <string name="hora1dia1Ev">D. Miguel A. Benítez Suarez, presidente de la Asociación
32.        Andaluza de oftalmología.</string>
33.    <string name="hora2">11.30h. - 12.00h. Pausa café</string>
```

28. <string name="hora3dia1">12.00h. - 14.30h. Grandes descubrimientos en células corneales</string>
29. <string name="hora3dia1Ev">D. Toufique Ahmed Soomro, Miembro de honor de la asociación AAEEIO.
30. Crear una técnica nunca antes vista para el tratamiento de la catarata le valió al indio
31. Toufique Ahmed Soomro, de 46 años, ser distinguido como el mejor oftalmólogo a nivel mundial.
32. En febrero pasado, alcanzó la medalla de oro durante el Cataract Surgery Olympics, en el Congreso
33. Mundial de Oftalmología realizado en México. Un reportaje de "Punto Final" permitió conocer
34. un poco más de la vida del destacado cirujano. Es egresado de la Universidad Federico Villarreal
35. y especializado con honores en Brasil y Estado Unidos. Además, es director de la clínica Oftalmo
36. Salud, dicta clases y ayuda a las personas que no pueden pagar un tratamiento. Uno de sus
37. pacientes es el reconocido ex futbolista peruano Julio Meléndez, quien a inicios de este año
38. padeció de una catarata. "Comencé a leer y no podía, comencé a ver nublado, incluso me compré
39. dos lupas", contó. El histórico jugador se puso en manos Izquiero Villavicencio y tras una exitosa
40. intervención volvió a ver sin problemas.</string>
41. <string name="hora1dia2">10.00h. - 11.30h. Curugía ocular no invasiva</string>
42. <string name="hora1dia2Ev">D. Jose M. Sánchez San Emeterio, oftalmólogo titular en el Hospital Virgen del Rocío.</string>
43. <string name="hora3dia2">12.00h. - 14.30h. Aproximación a la cirugía en pacientes con presbicia</string>
44. <string name="hora3dia2Ev">Dña. Sarah Zelikovitz, Profesora del Graduate Center of CUNY. Es
45. autora de más de 200 trabajos de investigación y descubridora de uno de los modelos experimentales
46. de uveitis que mas luz ha arrojado en la comprensión de los mecanismos de producción de esta enfermedad.
47. El pasado año 2010 recibió el reconocimiento de la Academia Americana de Investigación oftalmológica(ARVO)
48. a la mejor trayectoria en investigación oftalmológica. En un porcentaje considerable de casos, a
49. pesar de estudios exhaustivos, no se llega a identificar el motivo de esta inflamación ocular y
50. en ocasiones es la primera manifestación de una enfermedad general que se manifiesta incluso años
51. después de iniciados los problemas oculares. Cada año casi un tercio de los pacientes que padecen
52. uveitis sufrirá un episodio que agravamiento que dañará su visión, en algunos casos de manera irreversible.
53. Por todo esto, unido a un comportamiento crónico en muchos de los casos, la uveitis siempre supuso
54. un importante reto para los Oftalmólogos dada la complejidad de estos pacientes y su edad relativamente
55. joven (la edad media a la que se inicia son los 40 años). El primer motivo por el que una persona
56. joven, en edad laboral, puede quedarse ciega en un país sanitariamente desarrollado como es el nuestro
57. es la UVEÍTIS. Esta enfermedad consiste en la inflamación de una de las capas del globo ocular y puede
58. tener diversas causas: infecciosa, autoinmune, secundariaa traumatismos o relacionada con tumores.</string>
59. <string name="hora1dia3">10.00h. - 11.30h. Presentación de la nueva Asociación Gaditana de Oftalmología</string>

```

60.     <string name="hora1dia3Ev">Dña. Teresa Martínez Génova, oftalmóloga del Hospital
        Puerta del Mar.</string>
61.     <string name="hora3dia3">12.00h. - 14.30h. A Survey of auditory display in
        image-guided interventions</string>
62.     <string name="hora3dia3Ev">Prof. Dr. Ron Kikinis, Director fundador del Laboratorio
        de Planificación Quirúrgica,
63.         Boston, MA. El Dr. Kikinis es el Director fundador del Laboratorio de
        Planificación Quirúrgica,
64.         Departamento de Radiología, Hospital Brigham y de Mujeres, Facultad de Medicina
        de Harvard, Boston,
65.         MA, y Profesor de Radiología en la Escuela de Medicina de Harvard. Este
        laboratorio fue fundado en
66.         1990. En 2004 fue nombrado profesor de radiología en la Facultad de Medicina de
        Harvard. En 2009,
67.         fue el destinatario inaugural de la Sociedad MICCAI "Enduring Impact Award". El
        24 de febrero de 2010
68.         fue nombrado Director Distinguido de Robert Greenes de Informática Biomédica en
        el Departamento de
69.         Radiología del Hospital Brigham and Womens. El 1 de enero de 2014, fue nombrado
        "Institutsleiter"
70.         de Fraunhofer MEVIS y Profesor de Informática Médica de la Universidad de Bremen.
        A mediados de los años
71.         80, el Dr. Kikinis desarrolló un interés científico en los algoritmos de
        procesamiento de imágenes
72.         y su uso para extraer información relevante de datos de imágenes médicas. Desde
        entonces, este tema ha
73.         madurado desde un tema bastante exótico a un campo de la ciencia. Esto se debe al
        aumento explosivo
74.         tanto de la cantidad como de la complejidad de los datos de imágenes. El Dr.
        Kikinis ha dirigido y ha
75.         participado en investigaciones en diferentes áreas de la ciencia. Sus actividades
        incluyen investigación
76.         tecnológica (segmentación, registro, visualización, computación de alto
        rendimiento), desarrollo de
77.         sistemas de software e investigación biomédica en una variedad de especialidades
        biomédicas. La mayoría
78.         de su investigación es de naturaleza interdisciplinaria y está dirigida por
        equipos multidisciplinarios.
79.         Los resultados de su investigación han sido reportados en una variedad de
        artículos de revistas revisadas
80.         por colegas. Es autor y coautor de 316 artículos revisados por pares</string>
81.     <string name="appwidget_text">OFTAPP</string>
82.     <string name="add_widget">Add widget oftapp</string>
83.     <string name="txtChk">Ver todos los miembros con detalles</string>
84.     <string name="txtEdit">Introduce un DNI</string>
85.     <string name="txtBtn">Obtener Miembro</string>
86. </resources>
    
```

styles.xml

```

1. <resources>
2.
3.     <!-- Base application theme. -->
4.     <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
5.         <!-- Customize your theme here. -->
6.         <item name="colorPrimary">@color/colorPrimary</item>
7.         <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
8.         <item name="colorAccent">@color/colorAccent</item>
    
```

```
9.     </style>
10.
11.     <style name="AppTheme.NoActionBar">
12.         <item name="windowActionBar">false</item>
13.         <item name="windowNoTitle">true</item>
14.     </style>
15.
16. </resources>
```

colors.xml

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <resources>
3.     <color name="colorPrimary">#615292</color>
4.     <color name="colorPrimaryDark">#13073b</color>
5.     <color name="colorAccent">#aa9739</color>
6.     <color name="colorAzulito">#887caf</color>
7. </resources>
```