**qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm**

|  |
| --- |
| Manual de Programador  Ingeniería de Software  11° Semestre |

CURSO: Ingeniería de software

DOCENTE: Ing. Iván de León

PLAN PROYECTO FINAL

ESTUDIANTES: CARNE:

María José de León Fuentes 0903-10-5603

Miguel González Mérida 0903-08-4011

Claver Mijaíl Velásquez 0903-10-8106

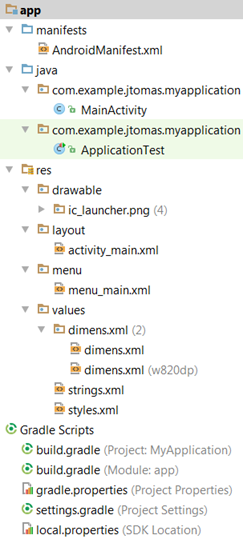
11º. Semestre Ingeniería en Sistemas

San Pedro Sacatepéquez, San Marcos 09 de Junio 2015

**PROYECTO REALIZADO EN ANDROID STUDIO**

Un proyecto Android está formado básicamente por un descriptor de la aplicación (*AndroidManifest.xml*), el código fuente en Java y una serie de ficheros con recursos. Cada elemento se almacena en una carpeta específica. Basandonos en el proyecto que acabamos de crear definiremos la estructura de un proyecto Android. Mas adelante se dará más detalles sobre la finalidad de cada fichero. Aunque muy similar existen algunas diferencias entre un proyecto creado en **Android Studio** y en **Eclipse**.

**Android Studio**

****

***AndroidManifest.xml***: Este fichero describe la aplicación Android. En él se indican las *actividades*, las *intenciones*, los *servicios* y los *proveedores de contenido* de la aplicación. También se declaran los permisos que requerirá la aplicación. Se indica la versión mínima de Android para poder ejecutarla, el paquete Java, la versión de la aplicación, etc.

***java***: Carpeta que contiene el código fuente de la aplicación. Como puedes observar los ficheros Java se almacenan en carpetas según el nombre de su paquete.

***MainActivity***: Clase Java con el código de la actividad inicial.

***ApplicationTest***: Clase Java pensada para insertar código de testeo de la aplicación utilizando el API JUnit.

***res***: Carpeta que contiene los recursos usados por la aplicación.

***drawable***: En esta carpeta se almacenan los ficheros de imágenes (JPG o PNG) y descriptores de imágenes en XML. En el proyecto se ha incluido el fichero *ic\_launcher.png* que será utilizado como icono de la aplicación. Observa como este recursdo se ha añadido en cuatro versiones diferentes. Como veremos en el siguiente capítulo, usaremos un sufijo especial si queremos tener varias versiones de un recurso, de forma que solo se cargue al cumplirse una determinada condición. Por ejemplo: (hdpi) significa que solo ha de cargar los recursos contenidos en esta carpeta cuando el dispositivo donde se instala la aplicación tiene una densidad gráfica alta (180-  dpi); (mdpi) se utilizará con densidad gráfica alta (180-  dpi). Si pulsas sobre las diferentes versiones del recurso observarás como se trata del mismo icono pero con más o menos resolución, de forma que en función de la densidad gráfica del dispositivo se ocupe un tamaño similar en la pantalla. Véase el apartado 2.6 del anexo E para más detalles.

***layout***: Contiene ficheros XML con vistas de la aplicación. Las vistas nos permitirán configurar las diferentes pantallas que compondrán la interfaz de usuario de la aplicación. Se utiliza un formato similar al HTML usado para diseñar páginas web. Se tratarán en el siguiente capítulo.

***menu***: Ficheros XML con los menús de cada actividad.

***values***: También utilizaremos ficheros XML para indicar valores usados en la aplicación, de esta manera podremos cambiarlos desde estos ficheros sin necesidad de ir al código fuente. En*dimens.xml*, se ha definido el margen horizontal y vertical por defecto. Observa como hay dos ficheros, el usado por defecto y el etiquetado como (w820dp)que será utilizado en dispositivos con ancho superior a 820 dp, esto ocurrirá en tabletas. En el fichero *strings.xml*, tendrás que definir todas las cadenas de caracteres de tu aplicación. Creando recursos alternativos resultará muy sencillo traducir una aplicación a otro idioma. Finalmente en *styles.xml*, podrás definir los estilos y temas de tu aplicación. Se estudian en el siguiente capítulo.

***anim***: Contiene ficheros XML con animaciones de vistas (Tween). Las animaciones se describen al final del capítulo 4.

***animator***: Contiene ficheros XML con animaciones de propiedades.

***xml***: Otros ficheros XML requeridos por la aplicación.

***raw***: Ficheros adicionales que no se encuentran en formato XML.

***Gradle Scripts***: En esta carpeta se almacenan una serie de ficheros Gradle que permiten construir la aplicación. Podrás definir aspectos importantes como la verisón del sdk de compilación (targetSdkVersion) y la versión mínima (minSdkVersion).

**APP MNS\_IP**

**Modulo Registro**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**AndroidManifest.xml**

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

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

*package="com.javaya.Registro"*

*android:versionCode="1"*

*android:versionName="1.0" >*

*<uses-sdk*

*android:minSdkVersion="8"*

*android:targetSdkVersion="17" />*

*<application*

*android:allowBackup="true"*

*android:icon="@drawable/ic\_launcher"*

*android:label="@string/app\_name"*

*android:theme="@style/AppTheme" >*

*<activity*

*android:name="com.javaya.Registro.MainActivity"*

*android:label="@string/app\_name" >*

*<intent-filter>*

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

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

*</intent-filter>*

*</activity>*

*</application>*

*</manifest>*

**USO DE SQLITE**

**AdminSQLiteOpenHelper.java**

**(Creación de la Base de Datos y sus Tablas si no existen)**

*package com.javaya.Registro;*

*import android.content.Context;*

*import android.database.sqlite.SQLiteDatabase;*

*import android.database.sqlite.SQLiteDatabase.CursorFactory;*

*import android.database.sqlite.SQLiteOpenHelper;*

*public class AdminSQLiteOpenHelper extends SQLiteOpenHelper {*

*public AdminSQLiteOpenHelper(Context context, String nombre, CursorFactory factory, int version) {*

*super(context, nombre, factory, version);*

*}*

*@Override*

*public void onCreate(SQLiteDatabase db) {*

*db.execSQL("create table agendacont(tel integer primary key, nombre text, dire text, ext integer)");*

*}*

*@Override*

*public void onUpgrade(SQLiteDatabase db, int versionAnte, int versionNue) {*

*db.execSQL("drop table if exists agendacont");*

*db.execSQL("create table agendacont(tel integer primary key, nombre text, dire text, exte integer)");*

*}*

*}*

**MainActivity.java**

**(Donde insertamos los datos a la Agenda)**

package com.javaya.Registro;

import android.app.Activity;

import android.content.ContentValues;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import android.view.Menu;

import android.view.View;

import android.widget.EditText;

import android.widget.Toast;

public class MainActivity extends Activity {

private EditText et1, et2, et3, et4;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

et1 = (EditText) findViewById(R.id.editText1);

et2 = (EditText) findViewById(R.id.editText2);

et3 = (EditText) findViewById(R.id.editText3);

et4 = (EditText) findViewById(R.id.editText4);

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.activity\_main, menu);

return true;

}

**//GUARDAMOS LOS DATOS A LA BASE DE DATOS**

public void alta(View v) {

AdminSQLiteOpenHelper admin = new AdminSQLiteOpenHelper(this,

"Agenda", null, 1);

SQLiteDatabase bd = admin.getWritableDatabase();

String tel = et1.getText().toString();

String nombre = et2.getText().toString();

String dire = et3.getText().toString();

String ext = et4.getText().toString();

ContentValues registro = new ContentValues();

registro.put("tel", tel);

registro.put("nombre", nombre);

registro.put("dire", dire);

registro.put("ext", ext);

bd.insert("agendacont", null, registro);

bd.close();

et1.setText("");

et2.setText("");

et3.setText("");

et4.setText("");

Toast.makeText(this, "Se cargaron los datos de la persona",

Toast.LENGTH\_SHORT).show();

}

**//BUSCAMOS CONTACTO POR NUMERO DE TELEFONO**

public void consulta(View v) {

AdminSQLiteOpenHelper admin = new AdminSQLiteOpenHelper(this,

"Agenda", null, 1);

SQLiteDatabase bd = admin.getWritableDatabase();

String Tel = et1.getText().toString();

Cursor fila = bd.rawQuery(

"select nombre,dire,ext from agendacont where tel=" + tel

+ "", null);

if (fila.moveToFirst()) {

et2.setText(fila.getString(0));

et3.setText(fila.getString(1));

et4.setText(fila.getString(2));

} else

Toast.makeText(this, "No existe una persona con dicho Telefono",

Toast.LENGTH\_SHORT).show();

bd.close();

}

**//ELIMINAMOS CONTACTO DE LA BASE DE DATOS**

public void baja(View v) {

AdminSQLiteOpenHelper admin = new AdminSQLiteOpenHelper(this,

"Agenda", null, 1);

SQLiteDatabase bd = admin.getWritableDatabase();

String dni = et1.getText().toString();

int cant = bd.delete("agendacont", "tel=" + tel + "", null);

bd.close();

et1.setText("");

et2.setText("");

et3.setText("");

et4.setText("");

if (cant == 1)

Toast.makeText(this, "Se borró la persona de Agenda",

Toast.LENGTH\_SHORT).show();

else

Toast.makeText(this, "No existe una persona con dicho Numero",

Toast.LENGTH\_SHORT).show();

}

**//MODIFICAMOS/EDITAMOS DATOS DEL CONTACTO**

public void modificacion(View v) {

AdminSQLiteOpenHelper admin = new AdminSQLiteOpenHelper(this,

"Agenda", null, 1);

SQLiteDatabase bd = admin.getWritableDatabase();

String tel = et1.getText().toString();

String nombre = et2.getText().toString();

String dire = et3.getText().toString();

String ext = et4.getText().toString();

ContentValues registro = new ContentValues();

registro.put("Nombre", nombre);

registro.put("Direccion", dire);

registro.put("Extencion", ext);

int cant = bd.update("Agenda", registro, "Telefono=" + tel, null);

bd.close();

if (cant == 1)

Toast.makeText(this, "Se modificaron los datos", Toast.LENGTH\_SHORT)

.show();

else

Toast.makeText(this, "¡No existe una persona con dicho Numero!",

Toast.LENGTH\_SHORT).show();

}

}

**MODULO MNS\_IP LLAMADAS**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**AndroidManifest.xml**

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

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

package="com.MSN\_IP\_Llamadas.calling"

android:versionCode="1"

android:versionName="1.0">

<uses-feature

android:name="android.hardware.microphone"

android:required="false"/>

<uses-permission android:name="android.permission.INTERNET"/>

<uses-permission android:name="android.permission.RECORD\_AUDIO"/>

<uses-permission android:name="android.permission.MODIFY\_AUDIO\_SETTINGS"/>

<uses-permission android:name="android.permission.READ\_PHONE\_STATE"/>

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"/>

<uses-sdk

android:minSdkVersion="9"

android:targetSdkVersion="17"/>

<application

android:allowBackup="true"

android:icon="@drawable/icon"

android:label="@string/app\_name"

android:theme="@style/AppTheme">

<service android:name=".SinchService"></service>

<activity

android:name=".LoginActivity"

android:label="@string/app\_name"

android:screenOrientation="portrait">

<intent-filter>

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

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

</intent-filter>

</activity>

<activity android:name=".CallScreenActivity" android:screenOrientation="portrait"/>

<activity android:name=".IncomingCallScreenActivity" android:screenOrientation="portrait"

android:noHistory="true"/>

<activity android:name=".PlaceCallActivity" android:screenOrientation="portrait"/>

</application>

</manifest>

**Layout: PantallaLlamadaActivity**

package com.MSN\_Ip.Llamadas;

import com.sinch.android.rtc.PushPair;

import com.sinch.android.rtc.calling.Call;

import com.sinch.android.rtc.calling.CallEndCause;

import com.sinch.android.rtc.calling.CallListener;

import android.media.AudioManager;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

import java.util.List;

import java.util.Locale;

import java.util.Timer;

import java.util.TimerTask;

public class CallScreenActivity extends BaseActivity {

static final String TAG = CallScreenActivity.class.getSimpleName();

private AudioPlayer mAudioPlayer;

private Timer mTimer;

private UpdateCallDurationTask mDurationTask;

private String mCallId;

private long mCallStart = 0;

private TextView mCallDuration;

private TextView mCallState;

private TextView mCallerName;

private class UpdateCallDurationTask extends TimerTask {

@Override

public void run() {

CallScreenActivity.this.runOnUiThread(new Runnable() {

@Override

public void run() {

updateCallDuration();

}

});

}

}

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.callscreen);

mAudioPlayer = new AudioPlayer(this);

mCallDuration = (TextView) findViewById(R.id.callDuration);

mCallerName = (TextView) findViewById(R.id.remoteUser);

mCallState = (TextView) findViewById(R.id.callState);

Button endCallButton = (Button) findViewById(R.id.hangupButton);

endCallButton.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

endCall();

}

});

mCallStart = System.currentTimeMillis();

mCallId = getIntent().getStringExtra(SinchService.CALL\_ID);

}

@Override

public void onServiceConnected() {

Call call = getSinchServiceInterface().getCall(mCallId);

if (call != null) {

call.addCallListener(new SinchCallListener());

mCallerName.setText(call.getRemoteUserId());

mCallState.setText("Llamando");

//mCallState.setText(call.getState().toString());

} else {

Log.e(TAG, "Started with invalid callId, aborting.");

finish();

}

}

@Override

public void onPause() {

super.onPause();

mDurationTask.cancel();

mTimer.cancel();

}

@Override

public void onResume() {

super.onResume();

mTimer = new Timer();

mDurationTask = new UpdateCallDurationTask();

mTimer.schedule(mDurationTask, 0, 500);

}

@Override

public void onBackPressed() {

// User should exit activity by ending call, not by going back.

}

private void endCall() {

mAudioPlayer.stopProgressTone();

Call call = getSinchServiceInterface().getCall(mCallId);

if (call != null) {

call.hangup();

}

finish();

}

private String formatTimespan(long timespan) {

long totalSeconds = timespan / 1000;

long minutes = totalSeconds / 60;

long seconds = totalSeconds % 60;

return String.format(Locale.US, "%02d:%02d", minutes, seconds);

}

private void updateCallDuration() {

if (mCallStart > 0) {

mCallDuration.setText(formatTimespan(System.currentTimeMillis() - mCallStart));

}

}

private class SinchCallListener implements CallListener {

@Override

public void onCallEnded(Call call) {

CallEndCause cause = call.getDetails().getEndCause();

Log.d(TAG, "Call ended. Reason: " + cause.toString());

mAudioPlayer.stopProgressTone();

setVolumeControlStream(AudioManager.USE\_DEFAULT\_STREAM\_TYPE);

String endMsg = "Call ended: " + call.getDetails().toString();

Toast.makeText(CallScreenActivity.this, endMsg, Toast.LENGTH\_LONG).show();

endCall();

}

@Override

public void onCallEstablished(Call call) {

Log.d(TAG, "Call established");

mAudioPlayer.stopProgressTone();

mCallState.setText("Llamada establecida");

//mCallState.setText(call.getState().toString());

setVolumeControlStream(AudioManager.STREAM\_VOICE\_CALL);

mCallStart = System.currentTimeMillis();

}

@Override

public void onCallProgressing(Call call) {

Log.d(TAG, "Call progressing");

mAudioPlayer.playProgressTone();

}

@Override

public void onShouldSendPushNotification(Call call, List<PushPair> pushPairs) {

// Send a push through your push provider here, e.g. GCM

}

}

}

**Layout: BaseActivity**

package com.MSN\_Ip.Llamadas;

import android.app.Activity;

import android.content.ComponentName;

import android.content.Intent;

import android.content.ServiceConnection;

import android.os.Bundle;

import android.os.IBinder;

public abstract class BaseActivity extends Activity implements ServiceConnection {

private SinchService.SinchServiceInterface mSinchServiceInterface;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

getApplicationContext().bindService(new Intent(this, SinchService.class), this,

BIND\_AUTO\_CREATE);

}

@Override

public void onServiceConnected(ComponentName componentName, IBinder iBinder) {

if (SinchService.class.getName().equals(componentName.getClassName())) {

mSinchServiceInterface = (SinchService.SinchServiceInterface) iBinder;

onServiceConnected();

}

}

@Override

public void onServiceDisconnected(ComponentName componentName) {

if (SinchService.class.getName().equals(componentName.getClassName())) {

mSinchServiceInterface = null;

onServiceDisconnected();

}

}

protected void onServiceConnected() {

// for subclasses

}

protected void onServiceDisconnected() {

// for subclasses

}

protected SinchService.SinchServiceInterface getSinchServiceInterface() {

return mSinchServiceInterface;

}

}

**Layou: LoginActvity**

package com.MNS\_Ip.Llamadas;

import com.sinch.android.rtc.SinchError;

import android.app.ProgressDialog;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

public class LoginActivity extends BaseActivity implements SinchService.StartFailedListener {

private Button mLoginButton;

private EditText mLoginName;

private ProgressDialog mSpinner;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.login);

mLoginName = (EditText) findViewById(R.id.loginName);

mLoginButton = (Button) findViewById(R.id.loginButton);

mLoginButton.setEnabled(false);

mLoginButton.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

loginClicked();

}

});

}

@Override

protected void onServiceConnected() {

mLoginButton.setEnabled(true);

getSinchServiceInterface().setStartListener(this);

}

@Override

protected void onPause() {

if (mSpinner != null) {

mSpinner.dismiss();

}

super.onPause();

}

@Override

public void onStartFailed(SinchError error) {

Toast.makeText(this, error.toString(), Toast.LENGTH\_LONG).show();

if (mSpinner != null) {

mSpinner.dismiss();

}

}

@Override

public void onStarted() {

openPlaceCallActivity();

}

private void loginClicked() {

String userName = mLoginName.getText().toString();

if (userName.isEmpty()) {

Toast.makeText(this, "Please enter a name", Toast.LENGTH\_LONG).show();

return;

}

if (!getSinchServiceInterface().isStarted()) {

getSinchServiceInterface().startClient(userName);

showSpinner();

} else {

openPlaceCallActivity();

}

}

private void openPlaceCallActivity() {

Intent mainActivity = new Intent(this, PlaceCallActivity.class);

startActivity(mainActivity);

}

private void showSpinner() {

mSpinner = new ProgressDialog(this);

mSpinner.setTitle("Logging in");

mSpinner.setMessage("Please wait...");

mSpinner.show();

}

}

**Layout: LlamadaEntrandoActivity**

package com.MNS\_Ip.Llamadas;

import com.sinch.android.rtc.PushPair;

import com.sinch.android.rtc.calling.Call;

import com.sinch.android.rtc.calling.CallEndCause;

import com.sinch.android.rtc.calling.CallListener;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.TextView;

import java.util.List;

public class IncomingCallScreenActivity extends BaseActivity {

static final String TAG = IncomingCallScreenActivity.class.getSimpleName();

private String mCallId;

private AudioPlayer mAudioPlayer;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.incoming);

Button answer = (Button) findViewById(R.id.answerButton);

answer.setOnClickListener(mClickListener);

Button decline = (Button) findViewById(R.id.declineButton);

decline.setOnClickListener(mClickListener);

mAudioPlayer = new AudioPlayer(this);

mAudioPlayer.playRingtone();

mCallId = getIntent().getStringExtra(SinchService.CALL\_ID);

}

@Override

protected void onServiceConnected() {

Call call = getSinchServiceInterface().getCall(mCallId);

if (call != null) {

call.addCallListener(new SinchCallListener());

TextView remoteUser = (TextView) findViewById(R.id.remoteUser);

remoteUser.setText(call.getRemoteUserId());

} else {

Log.e(TAG, "Started with invalid callId, aborting");

finish();

}

}

private void answerClicked() {

mAudioPlayer.stopRingtone();

Call call = getSinchServiceInterface().getCall(mCallId);

if (call != null) {

call.answer();

Intent intent = new Intent(this, CallScreenActivity.class);

intent.putExtra(SinchService.CALL\_ID, mCallId);

startActivity(intent);

} else {

finish();

}

}

private void declineClicked() {

mAudioPlayer.stopRingtone();

Call call = getSinchServiceInterface().getCall(mCallId);

if (call != null) {

call.hangup();

}

finish();

}

private class SinchCallListener implements CallListener {

@Override

public void onCallEnded(Call call) {

CallEndCause cause = call.getDetails().getEndCause();

Log.d(TAG, "Call ended, cause: " + cause.toString());

mAudioPlayer.stopRingtone();

finish();

}

@Override

public void onCallEstablished(Call call) {

Log.d(TAG, "Call established");

}

@Override

public void onCallProgressing(Call call) {

Log.d(TAG, "Call progressing");

}

@Override

public void onShouldSendPushNotification(Call call, List<PushPair> pushPairs) {

// Send a push through your push provider here, e.g. GCM

}

}

private OnClickListener mClickListener = new OnClickListener() {

@Override

public void onClick(View v) {

switch (v.getId()) {

case R.id.answerButton:

answerClicked();

break;

case R.id.declineButton:

declineClicked();

break;

}

}

};

}

**MODULO MNS\_IP MENSAJES**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Android manifest.xml**

Layout: MensajesActivity

package com.MNS\_Ip.Mensajes;

import com.sinch.android.rtc.PushPair;

import com.sinch.android.rtc.messaging.Message;

import com.sinch.android.rtc.messaging.MessageClient;

import com.sinch.android.rtc.messaging.MessageClientListener;

import com.sinch.android.rtc.messaging.MessageDeliveryInfo;

import com.sinch.android.rtc.messaging.MessageFailureInfo;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ListView;

import android.widget.Toast;

import java.util.List;

public class MessagingActivity extends BaseActivity implements MessageClientListener {

private static final String TAG = MessagingActivity.class.getSimpleName();

private MessageAdapter mMessageAdapter;

private EditText mTxtRecipient;

private EditText mTxtTextBody;

private Button mBtnSend;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.messaging);

mTxtRecipient = (EditText) findViewById(R.id.txtRecipient);

mTxtTextBody = (EditText) findViewById(R.id.txtTextBody);

mMessageAdapter = new MessageAdapter(this);

ListView messagesList = (ListView) findViewById(R.id.lstMessages);

messagesList.setAdapter(mMessageAdapter);

mBtnSend = (Button) findViewById(R.id.btnSend);

mBtnSend.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

sendMessage();

}

});

}

@Override

public void onDestroy() {

if (getSinchServiceInterface() != null) {

getSinchServiceInterface().removeMessageClientListener(this);

getSinchServiceInterface().stopClient();

}

super.onDestroy();

}

@Override

public void onServiceConnected() {

getSinchServiceInterface().addMessageClientListener(this);

setButtonEnabled(true);

}

@Override

public void onServiceDisconnected() {

setButtonEnabled(false);

}

private void sendMessage() {

String recipient = mTxtRecipient.getText().toString();

String textBody = mTxtTextBody.getText().toString();

if (recipient.isEmpty()) {

Toast.makeText(this, "Usuario no agregado", Toast.LENGTH\_SHORT).show();

return;

}

if (textBody.isEmpty()) {

Toast.makeText(this, "Ningún mensaje", Toast.LENGTH\_SHORT).show();

return;

}

getSinchServiceInterface().sendMessage(recipient, textBody);

mTxtTextBody.setText("");

}

private void setButtonEnabled(boolean enabled) {

mBtnSend.setEnabled(enabled);

}

@Override

public void onIncomingMessage(MessageClient client, Message message) {

mMessageAdapter.addMessage(message, MessageAdapter.DIRECTION\_INCOMING);

}

@Override

public void onMessageSent(MessageClient client, Message message, String recipientId) {

mMessageAdapter.addMessage(message, MessageAdapter.DIRECTION\_OUTGOING);

}

@Override

public void onShouldSendPushData(MessageClient client, Message message, List<PushPair> pushPairs) {

// Left blank intentionally

}

@Override

public void onMessageFailed(MessageClient client, Message message,

MessageFailureInfo failureInfo) {

StringBuilder sb = new StringBuilder();

sb.append("Fallo en el envio: ")

.append(failureInfo.getSinchError().getMessage());

Toast.makeText(this, sb.toString(), Toast.LENGTH\_LONG).show();

Log.d(TAG, sb.toString());

}

@Override

public void onMessageDelivered(MessageClient client, MessageDeliveryInfo deliveryInfo) {

Log.d(TAG, "onDelivered");

}

}