

Tracking de productos congelados – Pingüi

Aplicaciones moviles y servicios telematicos

Luis Macas Lopez - Richard Ruales | 2019-1s | 30/06/2019

# Enlace del GitHub del repositorio

<https://github.com/luianmac/AMST_E1_G8.git>

# Código Fuente Java y XML

MainActivity.java

Esta clase obtendrá los datos de los editText usuario y password ingresados por el usuario para colocarlos dentro en un HashMap y pasarlos a un objeto JSONObject con objetivo de enviarlo como parámetro a un JsonObjectRequest el cual será una solicitud a la base de datos para que se le asigne un token para el acceso a los datos almacenados en la base de datos. En respuesta se envía el token en caso de que la solicitud fue aceptada.

package com.example.adita.app\_pingui\_g8;  
  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.support.v7.app.AlertDialog;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.android.volley.Request;  
import com.android.volley.RequestQueue;  
import com.android.volley.Response;  
import com.android.volley.VolleyError;  
import com.android.volley.toolbox.JsonObjectRequest;  
import com.android.volley.toolbox.Volley;  
  
import org.json.JSONObject;  
  
import java.util.HashMap;  
import java.util.Map;  
  
public class MainActivity extends AppCompatActivity {  
 private RequestQueue mQueue=null;  
 private String token=null;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 mQueue= Volley.newRequestQueue(this);  
 }  
  
 */\*\*  
 \* Método que captura los datos ingresados por la persona en la interfaz  
 \* gráfica y llama a la función iniciarSesión para validar dichos datos  
 \** ***@param*** *v para reconocer que el método es para ser usado por los botones  
 \* método 'OnClick'  
 \*/* public void login(View v){  
 final EditText dt1=(EditText) findViewById(R.id.txtuser);  
 final EditText dt2=(EditText) findViewById(R.id.txtpassword);  
 String usuario=dt1.getText().toString();  
 String contrasena=dt2.getText().toString();  
 iniciarSesion(usuario,contrasena);  
  
 }  
  
 */\*\*  
 \* Método que se encarga de validar el usuario y las contraseñas  
 \* ingresadas para logonearse al menú de la APP y enviar el token a la  
 \* siguiente actividad  
 \** ***@param*** *usuario es una cadena de texto con la que se identifica la persona  
 \** ***@param*** *contrasena es una cadena de texto que solo conoce la persona  
 \*/* private void iniciarSesion(String usuario, String contrasena) {  
 Map<String, String> params=new HashMap<>();  
 params.put("username", usuario);  
 params.put("password", contrasena);  
  
 JSONObject parametros=new JSONObject(params);  
 String URL = "https://amstdb.herokuapp.com/db/nuevo-jwt";  
  
 JsonObjectRequest request = new JsonObjectRequest(Request.Method.POST, URL, parametros, new Response.Listener<JSONObject>() {  
 @Override  
 public void onResponse(JSONObject response) {  
 System.out.println(response);  
 try {  
 token = response.getString("token");  
 Intent i = new Intent(getBaseContext(), parametrosControl.class);  
 i.putExtra("token", token);  
 startActivity(i);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 }  
 }, new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 AlertDialog alertDialog = new AlertDialog.Builder(MainActivity.this).create();  
 alertDialog.setTitle("Alerta");  
 alertDialog.setMessage("Credenciales Incorrectas");  
 alertDialog.setButton(AlertDialog.BUTTON\_NEUTRAL, "OK",  
 new DialogInterface.OnClickListener() {  
 public void onClick(DialogInterface dialog, int  
 which) {  
 dialog.dismiss();  
 }  
 });  
 alertDialog.show();  
 }  
 });  
 mQueue.add(request);  
 }  
}

parametrosControl.java

Esta clase en primer lugar obtiene el token que fue enviado por la clase MainActivity, además esta clase contiene las funciones salir, mostrarTemperatura y mostrarRecorrido las cuales están asignadas a cada uno de los botones para que se ejecuten cuando estos sean presionados. En caso de ejecutarse mostrarTemperatura y mostrarRecorrido, estas enviarán el token a la clase temperatura y recorrido respectivamente y redirigirán a una nueva activity.

package com.example.adita.app\_pingui\_g8;  
  
import android.content.Intent;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
  
import com.android.volley.RequestQueue;  
import com.android.volley.toolbox.Volley;  
  
public class parametrosControl extends AppCompatActivity {  
 String token = "";  
  
 */\*\*  
 \* Método que permite la ejecución de la Activity y captura el token de  
 \* la anterior actividad  
 \** ***@param*** *savedInstanceState guarda el estado de la instancia  
 \*/* @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_parametros\_control);  
 Intent login = getIntent();  
 this.token = (String)login.getExtras().get("token");  
 }  
  
 */\*\*  
 \* Método para finalizar la ejecución de la app  
 \** ***@param*** *v para reconocer que el método es para ser usado por el boton 'SALIR'  
 \* método 'OnClick'  
 \*/* public void Salir(View v){  
 Intent intent = new Intent(this,MainActivity.class);  
 startActivity(intent);  
 stopService(intent);  
 finish();  
 //System.exit(0);  
 }  
  
 */\*\*  
 \* Método para mostrar el registro de temperatura almacenado en la base de datos DJANGO  
 \* y envía el token a la siguiente actividad  
 \** ***@param*** *v para reconocer que el método es para ser usado por el boton 'TEMPERATURA'  
 \* método 'OnClick'  
 \*/* public void mostrarTemperatura(View v){  
 Intent temperatura = new Intent(getBaseContext(),temperatura.class);  
 temperatura.putExtra("token", token);  
 startActivity(temperatura);  
 }  
 */\*\*  
 \* Método para mostrar el registro de los recorridos realizados almacenados en la base  
 \* de datos DJANGO y envía el token a la siguiente actividad.  
 \** ***@param*** *v para reconocer que el método es para ser usado por el boton 'RECORRIDO'  
 \* método 'OnClick'  
 \*/* public void mostrarRecorrido(View v){  
 Intent recorrido = new Intent(getBaseContext(),recorrido.class);  
 recorrido.putExtra("token", token);  
 startActivity(recorrido);  
 }  
}

recorrido.java

Esta clase hará una solicitud a la base de datos por medio JsonObjectRequest especificando el método a usar, GET, y el URL <https://amstdb.herokuapp.com/db/recorrido> de donde queremos obtener los datos de recorrido, en caso de una respuesta acertada se nos responderá con los datos obteniéndolos mediante response.getString("nombre de la variable"), para establecer estos strings en los TextView para que sean vistos por el usuario. Es muy importante que se escriba dentro de response.getString el mismo nombre de la variable que se encuentra en la base de datos.

package com.example.adita.app\_pingui\_g8;  
  
import android.content.Intent;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.widget.TextView;  
  
import com.android.volley.AuthFailureError;  
import com.android.volley.Request;  
import com.android.volley.RequestQueue;  
import com.android.volley.Response;  
import com.android.volley.VolleyError;  
import com.android.volley.toolbox.JsonObjectRequest;  
import com.android.volley.toolbox.Volley;  
  
import org.json.JSONObject;  
  
import java.lang.reflect.Array;  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.Map;  
  
public class recorrido extends AppCompatActivity {  
  
 private RequestQueue mQueue;  
 private String token = "";  
  
 */\*\*  
 \* Método que permite la ejecución de la Activity y captura el token de  
 \* la anterior actividad  
 \** ***@param*** *savedInstanceState guarda el estado de la instancia  
 \*/* @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_recorrido);  
 mQueue = Volley.newRequestQueue(this);  
 Intent login = getIntent();  
 this.token = (String)login.getExtras().get("token");  
  
 final TextView tv = (TextView) findViewById(R.id.txtRecorrido);  
 final TextView tv2 = (TextView) findViewById(R.id.txtRecorrido2);  
 final TextView tv3 = (TextView) findViewById(R.id.txtRecorrido3);  
 final TextView tv4 = (TextView) findViewById(R.id.txtRecorrido4);  
 ArrayList<TextView> a = new ArrayList<>();  
 a.add(tv);  
 a.add(tv2);  
 a.add(tv3);  
 a.add(tv4);  
 for(int i=0; i<4; i++) {  
 String url = "https://amstdb.herokuapp.com/db/recorrido/" + i;  
 revisarRecorrido(url,a.get(i));  
 }  
 }  
  
 */\*\*  
 \* Método que establece conexión con la base de datos y obtiene los datos detallados  
 \* en formato JSON  
 \** ***@param*** *url link de la base da datos a establecer conexióm  
 \** ***@param*** *a cuadro de texto de la interfaz en donde se van a escribir los datos obtenidos  
 \*/* private void revisarRecorrido(String url, final TextView a) {  
 //final TextView tv = (TextView) findViewById(R.id.txtRecorrido);  
 //String url\_Recorrido = "https://amstdb.herokuapp.com/db/recorrido";  
 JsonObjectRequest request = new JsonObjectRequest(  
 Request.Method.GET, url, null,  
 new Response.Listener<JSONObject>() {  
 @Override  
 public void onResponse(JSONObject response) {  
 System.out.println(response);  
 try {  
 a.setText("Origen: "+response.getString("origen")+"\n"+"Destino: "+response.getString("destino")+"\n"+"Fecha de Salida: "+response.getString("fecha\_inicio"));  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }, new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 }  
 }){  
 @Override  
 public Map<String, String> getHeaders() throws AuthFailureError  
 {  
 Map<String, String> params = new HashMap<String, String>();  
 params.put("Authorization", "JWT " + token);  
 System.out.println(token);  
 return params;  
 }  
 };;  
 mQueue.add(request);  
 }  
}

temperatura.java

Esta clase hará una solicitud a la base de datos por medio JsonObjectRequest especificando el método a usar, GET, y el URL <https://amstdb.herokuapp.com/db/registroDeFrios/20> de donde queremos obtener los datos de temperatura, en caso de una respuesta acertada se nos responderá con los datos obteniéndolos mediante response.getString("temperatura"), para establecer estos strings en los TextView para que sean vistos por el usuario.

package com.example.adita.app\_pingui\_g8;  
  
import android.content.Intent;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.widget.TextView;  
  
import com.android.volley.AuthFailureError;  
import com.android.volley.Request;  
import com.android.volley.RequestQueue;  
import com.android.volley.Response;  
import com.android.volley.VolleyError;  
import com.android.volley.toolbox.JsonObjectRequest;  
import com.android.volley.toolbox.Volley;  
  
import org.json.JSONObject;  
  
import java.util.HashMap;  
import java.util.Map;  
  
public class temperatura extends AppCompatActivity {  
 private RequestQueue mQueue;  
 private String token = "";  
  
 */\*\*  
 \* Método que permite la ejecución de la Activity y captura el token de  
 \* la anterior actividad  
 \** ***@param*** *savedInstanceState guarda el estado de la instancia  
 \*/* @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_temperatura);  
 mQueue = Volley.newRequestQueue(this);  
 Intent login = getIntent();  
 this.token = (String)login.getExtras().get("token");  
 revisarTemperatura();  
 }  
  
 */\*\*  
 \* Método que captura el valor temperatura almacenado en la base de datos  
 \* y los muetra en el cuadro de texto de la interfaz gráfica  
 \*/* private void revisarTemperatura() {  
 final TextView tv = (TextView) findViewById(R.id.txtTemperatura);  
 String url\_Temp = "https://amstdb.herokuapp.com/db/registroDeFrios/20";  
 JsonObjectRequest request = new JsonObjectRequest(  
 Request.Method.GET, url\_Temp, null,  
 new Response.Listener<JSONObject>() {  
 @Override  
 public void onResponse(JSONObject response) {  
 System.out.println(response);  
 try {  
 tv.setText(response.getString("temperatura")+ " C");  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }, new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 }  
 }){  
 @Override  
 public Map<String, String> getHeaders() throws AuthFailureError  
 {  
 Map<String, String> params = new HashMap<String, String>();  
 params.put("Authorization", "JWT " + token);  
 System.out.println(token);  
 return params;  
 }  
 };;  
 mQueue.add(request);  
 }  
}

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout 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"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
  
 <!-- Control EditText  
 Campo de texto en donde el usuario puede editar su contenido  
 Campo de texto para indicar el usuario  
 identificador 'txtuser'-->  
 <EditText  
 android:id="@+id/txtuser"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="usuario"  
 android:inputType="textPersonName" />  
  
 <!-- Control EditText  
 Campo de texto en donde el usuario puede editar su contenido  
 Campo de texto para indicar la contraseña  
 identificador txtpassword-->  
 <EditText  
 android:id="@+id/txtpassword"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="contraseña"  
 android:inputType="textPassword" />  
  
 <!-- Control Button  
 Boton que permite capturar el evento descrito en el código Java  
 para poder logoneare al menu del aplicativo  
 identificador btnlogin-->  
 <Button  
 android:id="@+id/btnlogin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:onClick="login"  
 android:text="login" />  
  
</LinearLayout>

activity\_parametros\_control.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout 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"  
 android:orientation="vertical"  
 tools:context=".parametrosControl">  
  
 <!-- Control Button  
 Boton que permite capturar el evento descrito en el código Java  
 para poder cambiar de actividad y mostrar los valores de temperatura  
 identificador btnTemp-->  
 <Button  
 android:id="@+id/btnTemp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="150dp"  
 android:onClick="mostrarTemperatura"  
 android:text="Temperatura" />  
  
 <!-- Control Button  
 Boton que permite capturar el evento descrito en el código Java  
 para poder cambiar de actividad y mostrar los valores de recorrido  
 identificador btnRecorre-->  
 <Button  
 android:id="@+id/btnRecorr"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:onClick="mostrarRecorrido"  
 android:text="Recorrido" />  
  
 <!-- Control Button  
 Boton que permite capturar el evento descrito en el código Java  
 para poder finalizar la activity establecida en ese instante  
 identificador btnSalir-->  
 <Button  
 android:id="@+id/btnSalir"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:onClick="Salir"  
 android:text="SALIR" />  
</LinearLayout>

activity\_recorrido.xml

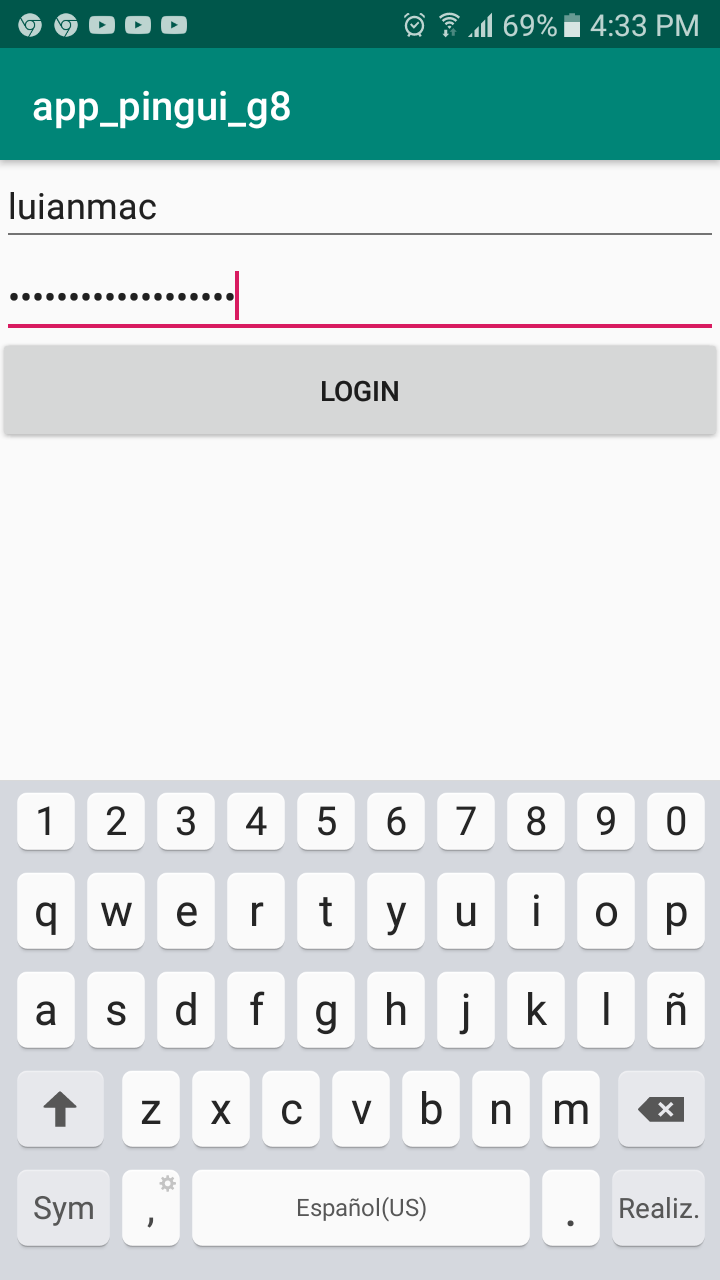
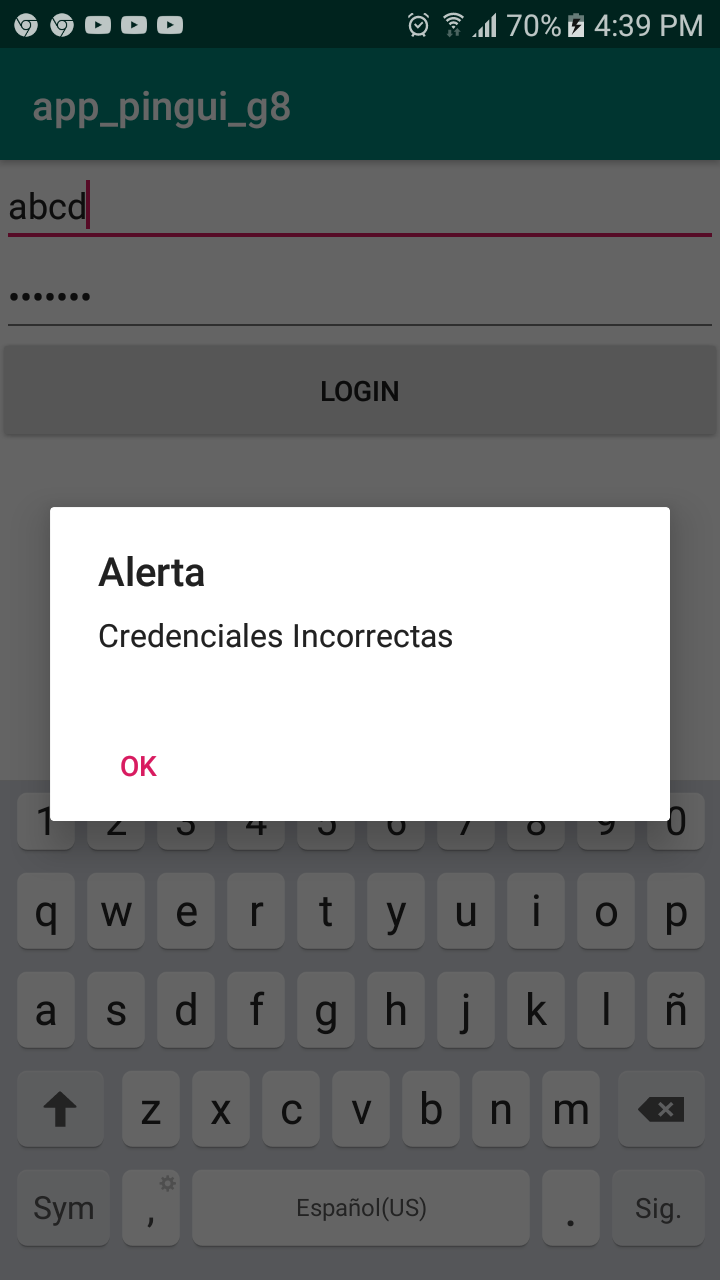
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".recorrido">  
  
 <!-- Control EditText  
 Campo de texto en donde el usuario puede editar su contenido  
 Campo de texto para indicar el Recorrido  
 identificador 'editText2'-->  
 <EditText  
 android:id="@+id/editText2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:text="Recorrido" />  
  
 <!-- Control TextView  
 Campo de texto en donde se puede mostrar datos  
 Campo de texto para indicar el recorrido capturado desde la DB  
 identificador 'txtRecorrido'-->  
 <TextView  
 android:id="@+id/txtRecorrido"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Recorrido"  
 android:textSize="24sp" />  
  
 <!-- Control TextView  
 Campo de texto en donde se puede mostrar datos  
 Campo de texto para indicar el recorrido capturado desde la DB  
 identificador 'txtRecorrido2'-->  
 <TextView  
 android:id="@+id/txtRecorrido2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="TextView" />  
  
 <!-- Control TextView  
 Campo de texto en donde se puede mostrar datos  
 Campo de texto para indicar el recorrido capturado desde la DB  
 identificador 'txtRecorrido3'-->  
 <TextView  
 android:id="@+id/txtRecorrido3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="TextView" />  
  
 <!-- Control TextView  
 Campo de texto en donde se puede mostrar datos  
 Campo de texto para indicar el recorrido capturado desde la DB  
 identificador 'txtRecorrido4'-->  
 <TextView  
 android:id="@+id/txtRecorrido4"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="TextView" />  
  
</LinearLayout>

activity\_temperatura.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout 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"  
 android:orientation="vertical"  
 tools:context=".temperatura">  
  
 <!-- Control EditText  
 Campo de texto en donde el usuario puede editar su contenido  
 Campo de texto para indicar el usuario  
 identificador 'Temperatura'-->  
 <EditText  
 android:id="@+id/editText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:text="Temperatura" />  
  
 <!-- Control TextView  
 Campo de texto en donde se puede mostrar datos  
 Campo de texto para indicar el recorrido capturado desde la DB  
 identificador 'txtTemperatura'-->  
 <TextView  
 android:id="@+id/txtTemperatura"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Temperatura"  
 android:textSize="24sp" />  
</LinearLayout>

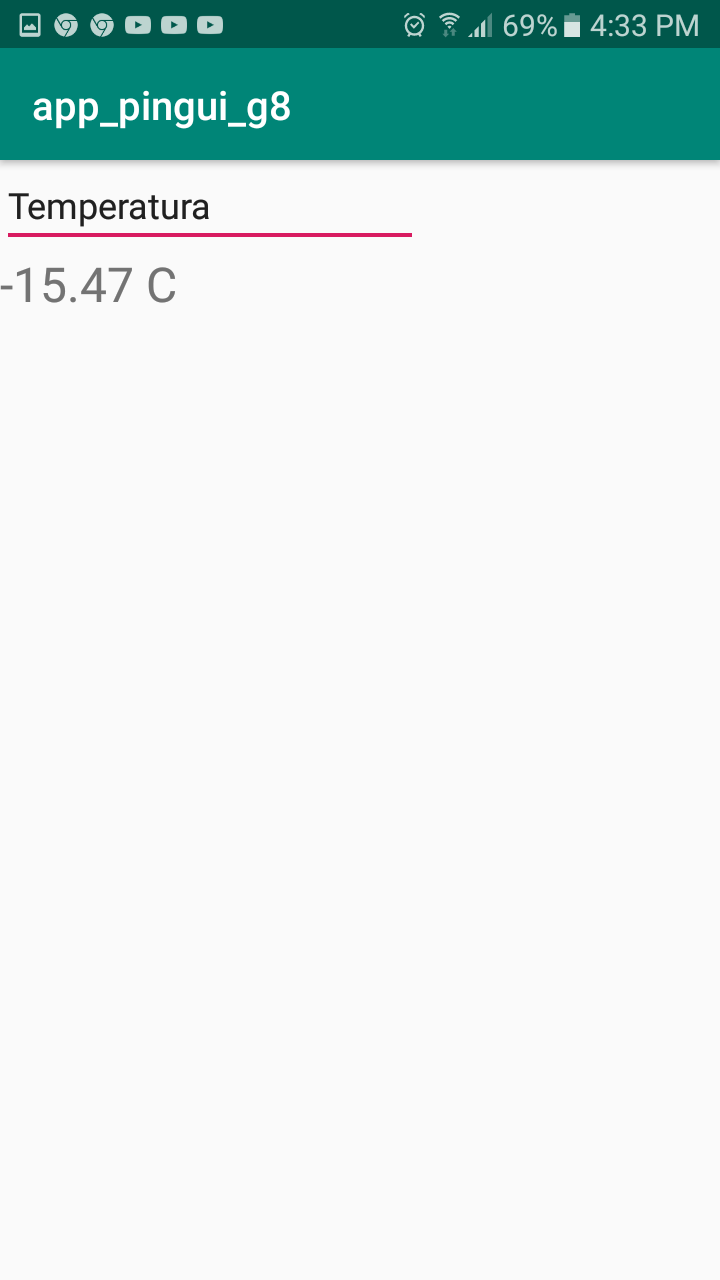
## Vistas del aplicativo

Una vez pulsado el icono de la aplicación se presentará esta ventana para realizar el inicio de sesión, donde se debe ingresar un usuario y una contraseña valida y luego clic sobre el botón login, si las credenciales no están registradas en la base de datos que se va a usar entonces, se presentará un cuadro de dialogo indicando que las credenciales son incorrectas.



Inmediatamente luego de presionar el botón login la app redirige a una nueva ventana que nos mostrará un menú con tres opciones Temperatura, Recorrido y Salir, las cuales permitirán al usuario obtener los datos de temperatura y recorrido que se encuentran almacenados en la base de datos y salir de la aplicación.

Si se presiona la opción “temperatura” la app redirige a una nueva ventana donde se mostrará los datos de temperatura que han tenido los productos.



Si se presiona la opción “recorrido” la app redirige a una nueva ventana donde se mostrará las ubicaciones de origen y destino por donde ha estado los productos

Finalmente, si se presione la opción “salir” la app redirige a la ventana del menú de login.