



TRABAJO AUTÓNOMO 2

APLICACIÓN MÓVIL CON INTEGRACIÓN DE BASE DE DATOS EXTERNAS Y
PRIVADAS UTILIZANDO REST API

ELABORADO POR: ADRIANA COLLAGUAZO JARAMILLO

ITINERARIO: APLICACIONES MÓVILES Y SISTEMAS TELEMÁTICOS

CARRERA DE INGENIERÍA EN TELEMÁTICA

FIEC-ESPOL

Trabajo Autónomo:

Objetivo de Aprendizaje: Desarrollar aplicaciones móviles sencillas considerando las características de la programación de dispositivos móviles.

Recursos: Android Studio, REST API del curso, REST API privada, bases de datos PostgreSQL usando formato Json.

Duración: 10 horas

INSTRUCCIONES:

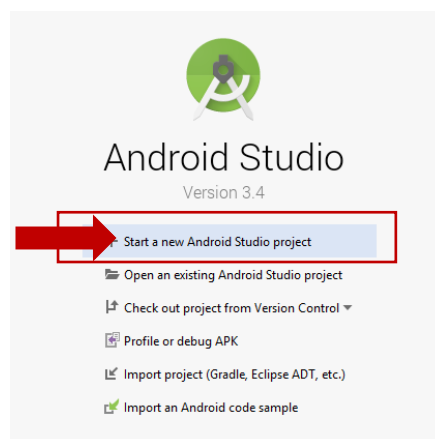
Crear una aplicación que interactúe con 3 tipos de base de datos utilizadas actualmente durante el desarrollo de aplicaciones móviles:

1. Base de datos externa PostgreSQL para el inicio de sesión y la consulta de datos
2. Uso de token de autorización
3. Base de datos privadas con REST APIS disponibles para desarrollo.

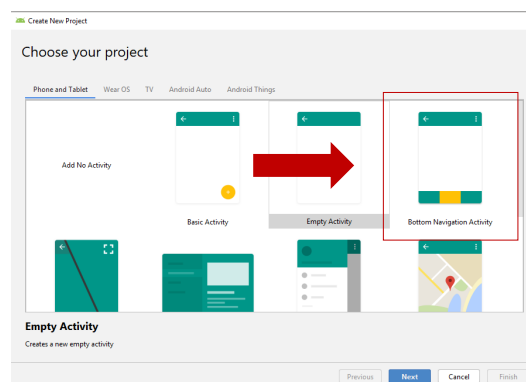
Para lograr la integración de aplicaciones móviles con bases de datos externas y privadas utilizando REST API y autorización de seguridad mediante web tokens.

ACTIVIDADES:**Paso 1: Crear un nuevo proyecto en Android Studio**

1. Al abrir Android Studio, podemos crear, abrir o importar proyectos. Seleccione "Start a new Android Studio project".

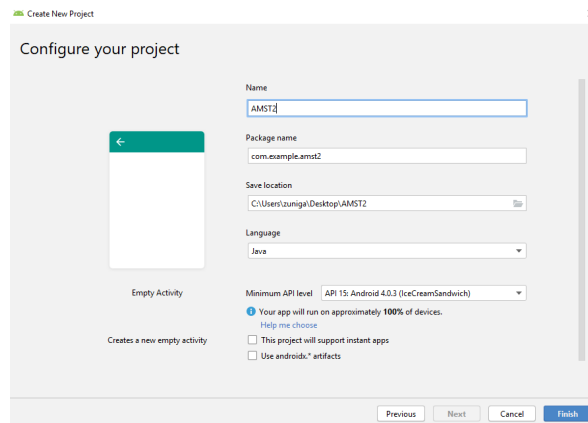


2. Seleccionar el tipo de proyecto: Para esta práctica escogeremos la pestaña **Phone and Tablet > Empty Activity**. *Nota: Otro tipo de actividades viene por defecto con componentes no necesarios para este taller.*



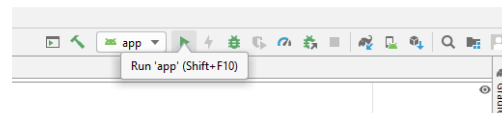
3. Configuración inicial del proyecto.

- [Name]: Colocaremos el nombre de nuestra app. (Recuerde que este nombre será reflejado en el PlayStore al momento de publicarlo). Para este taller, usaremos **AMST[numeroGrupo]**.
 - [PackageName]: Paquete principal de código java, se obtiene automáticamente del nombre.
 - [Save Location]: Dirección donde se ubica el proyecto en nuestra PC.
 - [Lenguaje]: java.
4. Seleccionamos **FINISH**. Como resultado se creará un proyecto vacío, solo presentado el mensaje "Hello World".

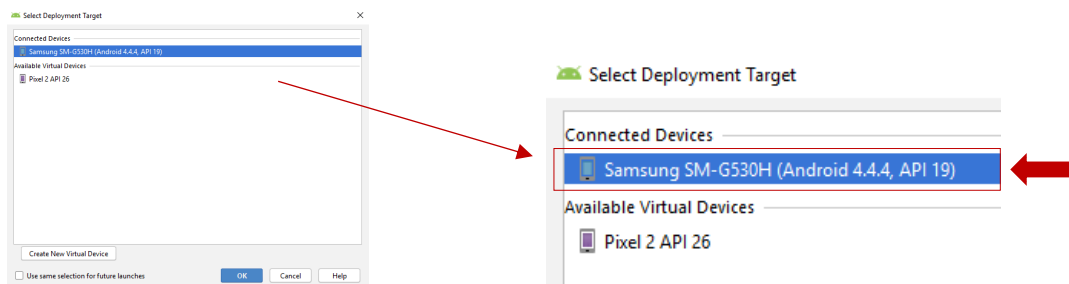


Paso 2: Ejecutar nuestra aplicación en el teléfono celular.

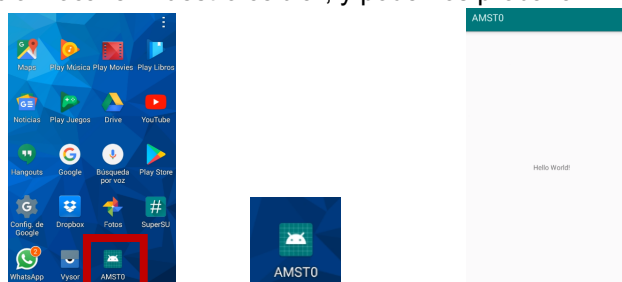
- Del lado superior derecho de AndroidStudio aparecer la barra de ejecución ("RUN").
- Buscar el icono play para ejecutar nuestra app (o usar el atajo Shift+F10).



- Ahora aparecerá la ventana "Select Deployment target" (Seleccionar dispositivo a ejecutar). Ahora seleccionamos en "**Connected devices**" > [modelo del teléfono conectado]: **Samsung SM G530**.



- Esto creará una aplicación local en nuestro celular, y podemos probarla.



Paso 3: Crear interfaz de usuario

Para esta aplicación crearemos una interfaz de usuario de inicio de sesión y un menú principal. Al momento de abrir la aplicación, se ejecutará MainActivity.java y activity_main.xml

1. Dentro del archivo activity_main.xml que se encuentra en la ruta app > res > layout, vamos a colocar los siguientes elementos:
 - a. (1 Layout vertical) para el arreglo de los elementos
 - b. (1 EditText) un título
 - c. (2 Inputs) Ingresar usuario y contraseña
 - d. (1 button) Botón de ingreso

Vista de inicio de sesión

```
<?xml version="1.0" encoding="utf-8" ?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="395dp"
        android:layout_height="340dp"
        android:orientation="vertical"
        tools:layout_editor_absoluteX="8dp"
        tools:layout_editor_absoluteY="8dp">

        <TextView
            android:id="@+id/textView"
            android:layout_width="match_parent"
            android:layout_height="179dp"
            android:gravity="center"
            android:text="Bienvenido"
            android:textSize="36sp" />

        <EditText
            android:id="@+id/editText"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Usuario"
            android:inputType="text" />

        <EditText
            android:id="@+id/editText2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Contraseña"
            android:inputType="textPassword" />

        <Button
            android:id="@+id/button"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Inicio de Sesión" />

    </LinearLayout>
```



En el título (TextView) utilizamos propiedades como textsize y gravity.

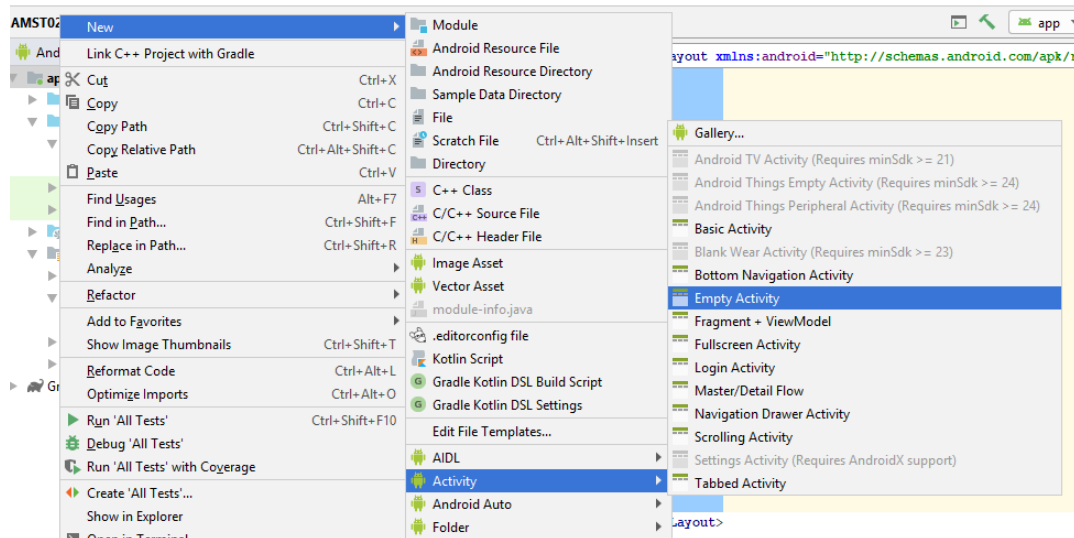
En los inputs usamos propiedades como hint y input type para personalizar el formulario.

Las propiedades customizan nuestra aplicación.

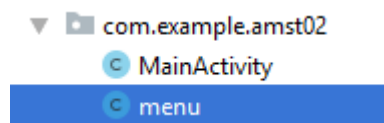
Preguntas de desafío:

- a) ¿Qué atributo me permitiría cambiar el tipo de letra de mi ventana? (Escriba la línea de código que usaría).
- b) Que hace el atributo type (mencione otros 5 valores que puede tener el atributo type diferentes al que se usa en el código).

2. Crear una segunda actividad a mi aplicación para redirigir al usuario cuando este ha completado con éxito el inicio de sesión.
 - a. Para crear una nueva actividad, damos clic derecho en "app".
 - b. Seleccionar *New > Activity > Empty Activity*.



- c. Ingresamos un el nombre para la nueva actividad [menú], lo que generara dos archivos:
 - i. Menu.java
 - ii. Menu.xml



3. Creamos un menú básico (15 minutos):
 - a. (1 Layout vertical) para el arreglo de los elementos
 - b. (1 EditText) un título
 - c. (4 Botones) un título



4. Creamos una transición de una actividad a otra para (De inicio de sesión -> Menú principal)
 - a. Seleccionamos el botón de Iniciar sesión > atributo(onClick)
 - b. Escribimos el nombre de la función **irMenuPrincipal**
 - c. Creamos la función irMenuPrincipal dentro de MainActivity.java

```

package com.example.amst02;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;           //Es necesario importar las librerías
import android.content.Intent;
import android.content.Intent;

public class MainActivity extends AppCompatActivity {

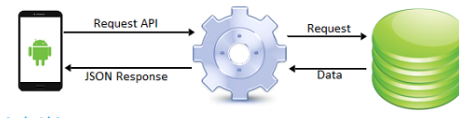
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    //IrMenuPrincipal desplaza la vista del teléfono entre actividades
    public void irMenuPrincipal(View v){
        Intent menuPrincipal = new Intent(getBaseContext(), menu.class);
        startActivity(menuPrincipal);
    }
}

```

Paso 4: Inicio de sesión con Base de datos externa (REST API).

Una REST API o API de desarrollo permite interactuar con una base de datos externa por medio de llamados HTTP, devolviendo información en formato Json



La REST API que usaremos para este taller será <http://amstdb.herokuapp.com/db/>. Esta API está desarrollada en Django Python con autenticación JWT. De esta manera, el teléfono puede acceder a grandes cantidades de información sin tener que alojarla de forma interna. A cambio requiere conexión a internet (lo cual en casos puede llegar a ser costosa).

1. Dar permisos para el uso de internet. Nuestra aplicación debe permitir conectarse a internet por medio del teléfono. Para ello, agregamos la siguiente línea en el archivo manifiesto (El archivo de configuraciones generales)

- a. El archivo se encuentra en app > manifests > AndroidManifest.xml
- b. Agregamos la siguiente línea (para conceder permisos de Internet):
`<uses-permission android:name="android.permission.INTERNET" />`

Se agrega antes del tag aplicaciones

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.amst02">
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"

```

2. Instalamos las dependencias necesarias (Librería Volley para request http).
 - a. Agregamos la librería Volley en el archivo **build.gradle** que se encuentra en "Grandle Scripts" > build.gradle
 - b. Agregamos la siguiente línea:

```
implementation 'com.android.volley:volley:1.1.1'
```

Se agrega dentro del grupo de dependencias.

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-
layout:1.1.3'
    implementation 'com.android.volley:volley:1.1.1'
}
```

- a) Una vez ingresada se actualizará el Gradle (Paquete de librerías) [!En caso de no sincronizarse puede sincronizarlo manualmente con File > Sync Project with Gradle files].

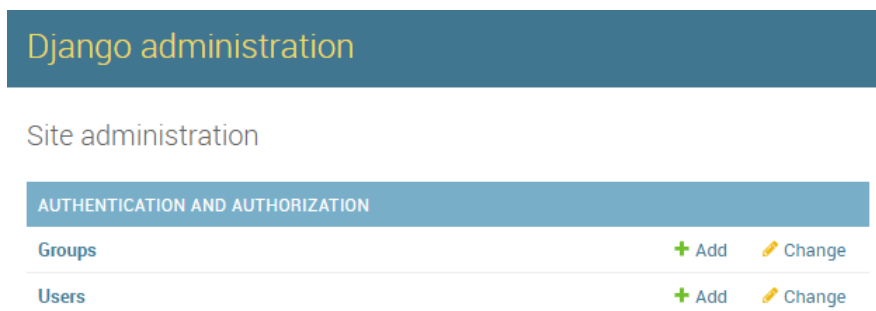
Paso 5 : Inicio de sesión con Base de datos externa (REST API).

Creamos la función Iniciar sesión en MainActivity.java

Ahora creamos la función que va a recibir como parametros nuestro usuario y contraseña y retornara el token de seguridad o el mensaje de que el usuario no es valido.

1. Creamos una variable lista (queue) de las solicitudes (request) a ejecutar.

```
public class MainActivity extends AppCompatActivity {
    private RequestQueue mQueue = null;
```
2. Debe crearse un usuario, accediendo al portal <https://amstadb.herokuapp.com/admin/>
 Usuario: practica02, contraseña practica02.
 Cree un usuario, y esas serán las credenciales que usara para su aplicación (tabla user > +Add).



3. Creamos la función IniciarSesion(). Para realizar la siguiente llamada HTTP:

Url: <https://amstadb.herokuapp.com/db/nuevo-jwt>

Método: POST

Cuerpo: {usuario, password}

Retorno: {token de usuario}

MainActivity.java

Revise los imports necesarios!

```
package com.example.amst02;

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.TextView;
```

```

import org.json.JSONException;
import org.json.JSONObject;

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 java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class MainActivity extends AppCompatActivity {

    private RequestQueue mQueue;
    private String token = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mQueue = Volley.newRequestQueue(this);
    }

    public void irMenuPrincipal(View v){
        final EditText usuario = (EditText) findViewById(R.id.tx_usuario);
        final EditText password = (EditText) findViewById(R.id.tx_pass);
        String str_usuario = usuario.getText().toString();
        String str_password = password.getText().toString();

        iniciarSesion(str_usuario, str_password);
    }

    private void iniciarSesion(String usuario, String password){
        Map<String, String> params = new HashMap();
        params.put("username", usuario);
        params.put("password", password);
        JSONObject parametros = new JSONObject(params);

        String login_url = "https://amstdb.herokuapp.com/db/nuevo-jwt";
        JsonObjectRequest request = new JsonObjectRequest(
            Request.Method.POST, login_url, parametros,
            new Response.Listener<JSONObject>() {
                @Override
                public void onResponse(JSONObject response) {
                    System.out.println(response);
                    try {
                        token = response.getString("token");
                        Intent menuPrincipal = new
Intent(getBaseContext(), menu.class);
                        menuPrincipal.putExtra("token", token);
                        startActivity(menuPrincipal);

                    } 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);
}
}

```

Obtenemos el token en el menú

Archivo: Menu.java

```

package com.example.amst02;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class menu extends AppCompatActivity {

    String token = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menu);
        Intent login = getIntent();
        this.token = (String)login.getExtras().get("token");
    }

    public void Salir(View v){
        this.finish();
        System.exit(0);
    }

    public void revisarSensores(View v){
        Intent red_sensores = new Intent(getApplicationContext(),
red_sensores.class);
        red_sensores.putExtra("token", token);
        startActivity(red_sensores);
    }
}

```

redes_sensores.java

Utilizaremos imágenes para indicar los diferentes sensores. Puede utilizar imágenes de internet: Formato png menos a 500mb. Estas imágenes deben ser agregadas al siguiente directorio: app > res > drawable

Redes_sensores.java

```

package com.example.amst02;

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.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 red_sensores extends AppCompatActivity {

    private RequestQueue mQueue;

    private String token = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_red_sensores);
        mQueue = Volley.newRequestQueue(this);

        Intent login = getIntent();
        this.token = (String) login.getExtras().get("token");
        revisarSensores();
    }

    private void revisarSensores() {

        final TextView tempValue = (TextView) findViewById(R.id.tempVal);
        final TextView pesoValue = (TextView) findViewById(R.id.pesoVal);
        final TextView humedadValue = (TextView)
        findViewById(R.id.humedadVal);

        String url_temp = "https://amstdb.herokuapp.com/db/logUno/1";
        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 {
tempValue.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);

String url_humedad = "https://amstdb.herokuapp.com/db/logUno/2";
JsonObjectRequest request_humedad = new JsonObjectRequest(
    Request.Method.GET, url_humedad, null,
    new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {
            System.out.println(response);
            try {
                humedadValue.setText(response.getString("humedad") + " 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_humedad);

String url_peso = "https://amstdb.herokuapp.com/db/logUno/3";
JsonObjectRequest request_peso = new JsonObjectRequest(
    Request.Method.GET, url_peso, null,
    new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {
            System.out.println(response);
            try {
                pesoValue.setText(response.getString("peso") + "
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);
}
}

```

Actividad de redes de sensores (XML)

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".red_sensores">

    <LinearLayout
        android:layout_width="395dp"
        android:layout_height="715dp"
        android:orientation="vertical"
        tools:layout_editor_absoluteX="8dp"
        tools:layout_editor_absoluteY="8dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="85dp"
            android:orientation="horizontal">

            <ImageView
                android:id="@+id/img1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                app:srcCompat="@drawable/temp" />

            <TextView
                android:id="@+id/tempTx"
                android:layout_width="wrap_content"
                android:layout_height="63dp"
                android:layout_weight="1"
                android:gravity="center_vertical"
                android:text="Temperatura Actual"
                android:textSize="24sp" />

            <TextView
                android:id="@+id/tempVal"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:gravity="right|center_vertical"
                android:text="00"
                android:textSize="24sp" />

        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="81dp"
            android:orientation="horizontal">

            <ImageView
                android:id="@+id/img2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                app:srcCompat="@drawable/hume" />

            <TextView
                android:id="@+id/humedadTx"
                android:layout_width="wrap_content"
                android:layout_height="63dp"
                android:layout_weight="1"
                android:gravity="center_vertical"
                android:text="Humedad"

```

```

        android:textSize="24sp" />

<TextView
    android:id="@+id/humedadVal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:gravity="right|center_vertical"
    android:text="0C"
    android:textSize="24sp" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <ImageView
        android:id="@+id/img3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/peso" />

    <TextView
        android:id="@+id/pesotx"
        android:layout_width="wrap_content"
        android:layout_height="63dp"
        android:layout_weight="1"
        android:gravity="center_vertical"
        android:text="Peso"
        android:textSize="24sp" />

    <TextView
        android:id="@+id/pesoVal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="right|center_vertical"
        android:text="hume"
        android:textSize="24sp" />

</LinearLayout>

</LinearLayout>

</android.support.constraint.ConstraintLayout>

```

Tareas de Desafío:

1. Realizar una actividad que me permita enviar una nueva temperatura a la base de datos actual (DEBERA UTILIZAR EL METODO POST como en el inicio de sesión, pero también enviar los parámetros y token de una nueva temperatura).

FORMATO DEL TRABAJO

El trabajo autónomo será desarrollado en el siguiente formato:

- Nombre del archivo: AMST_P01_Trabajo Autónomo A_Grupo B_Apellido1_Apellido2_Apellido3...
- (*) Siendo A el número del trabajo y B el número del grupo
- Nombre de la materia y paralelo 1
- Título del trabajo: Ejemplo: Trabajo Autónomo A - Tema
- Nombre de la profesora

- Número de grupo
- Nombres/Apellidos de los integrantes del grupo que hayan desarrollado el trabajo
- Fecha de inicio y fin del trabajo
- Resultados de las actividades planteadas: Explicación de las actividades ejecutadas, incluyendo las imágenes del proceso.
- Conclusiones y Recomendaciones: Respecto a lo aprendido durante el desarrollo del trabajo.
- Referencias bibliográficas: Colocar los documentos, enlaces web o libros consultados.