

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="aula08_ler_temperatura_umidade_dht_bluetooth.android.aula08_pdm.android.pdm.aula08_ler_temperaturaumidadeth11bluetooth" >

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".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>
```

res/layout/activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/conectar"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="Conectar" />

    <Button
        android:id="@+id/desconectar"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="Desconectar" />

    <Button
        android:id="@+id/btnMedirTemperatura"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="Medir Temperatura" />

    <Button
        android:id="@+id/btnMedirUmidade"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="Medir Umidade" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="text"
        android:height="100dp"
        android:id="@+id/edtTxtResultadoMedicao"/>

</LinearLayout>
```

MainActivity.java

```
package aula08_ler_temperatura_umidade_dht_bluetooth.android.aula08_pdm.android.pdm.aula08_lertemperaturaumidadeth11bluetooth;

import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.os.SystemClock;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.UUID;

public class MainActivity extends Activity {
    Button conectar;
    Button desconectar;
    Button receberDadosTemperatura;
    Button receberDadosUmidade;
    EditText medicoesRecebidas;

    // Represents a remote Bluetooth device.
    private BluetoothDevice dispositivoBluetoothRemoto;

    // Represents the local device Bluetooth adapter.
    private BluetoothAdapter meuBluetoothAdapter = null;

    // A connected or connecting Bluetooth socket.
    private BluetoothSocket bluetoothSocket = null;
    private static final String endereco_MAC_do_Bluetooth_Remoto = "98:D3:31:40:31:E8";
    public static final int CÓDIGO_PARA_ATIVAÇÃO_DO_BLUETOOTH = 1;
    private static final UUID MEU_UUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");
    private InputStream inputStream = null;
    private OutputStream outputStream = null;

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

    public void fazerConexoesDoLayout_e_Listeners(){

        conectar = (Button)findViewById(R.id.conectar);
        desconectar = (Button)findViewById(R.id.desconectar);
        receberDadosTemperatura = (Button)findViewById(R.id.btnMedirTemperatura);
        receberDadosUmidade = (Button)findViewById(R.id.btnMedirUmidade);
        medicoesRecebidas = (EditText)findViewById(R.id.edtTxtResultadoMedicao);

        conectar.setOnClickListener(new Conectar());
        desconectar.setOnClickListener(new Desconectar());
        receberDadosTemperatura.setOnClickListener(new ReceberDados());
        receberDadosUmidade.setOnClickListener(new ReceberDados());
    }

    public void verificarCondiçãoDoBluetooth() {

        // Get a handle to the default local Bluetooth adapter.
        meuBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

        // Verifica se o celular tem Bluetooth
        if(meuBluetoothAdapter == null){
            Toast.makeText(getApplicationContext(), "Dispositivo não possui adaptador Bluetooth", Toast.LENGTH_LONG).show();

            // Finaliza a aplicação.
            finish();
        } else {
            // Verifica se o bluetooth está desligado. Se sim, pede permissão para ligar.
            if(!meuBluetoothAdapter.isEnabled()){
                Intent novolIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(novolIntent, CÓDIGO_PARA_ATIVAÇÃO_DO_BLUETOOTH);
            }
        }
    }
}
```

```

    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    switch(requestCode){
        case CÓDIGO_PARA_ATIVAÇÃO_DO_BLUETOOTH:
            if(resultCode == Activity.RESULT_OK){
                Toast.makeText(getApplicationContext(), "Bluetooth foi ativado", Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText(getApplicationContext(), "Bluetooth não foi ativado", Toast.LENGTH_LONG).show();
            }
            break;
    }
}

@Override
protected void onPause() {
    super.onPause();
}

@Override
protected void onResume() {
    super.onResume();
}

public class Conectar implements View.OnClickListener {

    @Override
    public void onClick(View v) {

        // Validate a Bluetooth address, such as "00:43:A8:23:10:F0" (Alphabetic characters must be uppercase to be valid)
        if(BluetoothAdapter.checkBluetoothAddress(endereco_MAC_do_Bluetooth_Remoto)){
            dispositivoBluetooohRemoto = meuBluetoothAdapter.getRemoteDevice(endereco_MAC_do_Bluetooth_Remoto);
        } else{
            Toast.makeText(getApplicationContext(), "Endereço MAC do dispositivo Bluetooth remoto não é válido",
Toast.LENGTH_SHORT).show();
        }

        try{
            bluetoothSocket = dispositivoBluetooohRemoto.createInsecureRfcommSocketToServiceRecord(MEU_UUID);
            bluetoothSocket.connect();
            medicoesRecebidas.setText("");
            Toast.makeText(getApplicationContext(), "Conectado", Toast.LENGTH_SHORT).show();
        } catch(IOException e){
            Log.e("ERRO AO CONECTAR", "O erro foi" + e.getMessage());
            Toast.makeText(getApplicationContext(), "Conexão não foi estabelecida", Toast.LENGTH_SHORT).show();
        }
    }
}

public class Desconectar implements View.OnClickListener {

    @Override
    public void onClick(View v) {
        medicoesRecebidas.setText("");
        if(bluetoothSocket != null) {
            try{
                // Immediately close this socket, and release all associated resources.
                bluetoothSocket.close();
                bluetoothSocket = null;
                Toast.makeText(getApplicationContext(), "Conexão encerrada", Toast.LENGTH_SHORT).show();
            } catch(IOException e){
                Log.e("ERRO AO DESCONECTAR", "O erro foi" + e.getMessage());
                Toast.makeText(getApplicationContext(), "Erro - A conexão permanece estabelecida", Toast.LENGTH_SHORT).show();
            }
        } else{
            Toast.makeText(getApplicationContext(), "Não há nenhuma conexão estabelecida a ser desconectada",
Toast.LENGTH_SHORT).show();
        }
    }
}

public class ReceberDados implements View.OnClickListener {

    private void sendData(String message) {
        byte[] msgBuffer = message.getBytes();

        try {

```

```

        outputStream.write(msgBuffer);
    } catch (IOException e) {
        Toast.makeText(getApplicationContext(), "Erro - Ao enviar dados", Toast.LENGTH_SHORT).show();
    }
}

@Override
public void onClick(View v) {

    // Verifica se há conexão estabelecida com o Bluetooth.
    if(bluetoothSocket != null){

        medicoesRecebidas.setText("");
        switch (v.getId()) {
            case R.id.btnMedirTemperatura:
                try{
                    //Envia Temperatura e umidade
                    outputStream = bluetoothSocket.getOutputStream();
                    sendData("t");
                    SystemClock.sleep(1000);

                    // Get the input stream associated with this socket.
                    inputStream = bluetoothSocket.getInputStream();

                    // Reads bytes from this stream and stores them in the byte array
                    byte[] msgBuffer = new byte[1024];
                    inputStream.read(msgBuffer);

                    medicoesRecebidas.setText(new String(msgBuffer));

                } catch (IOException e){
                    Log.e("ERROR", "O erro foi" + e.getMessage());
                    Toast.makeText(getApplicationContext(), "Mensagem não recebida", Toast.LENGTH_LONG).show();
                }
                break;

            case R.id.btnMedirUmidade:
                try{
                    //Envia Temperatura e umidade
                    outputStream = bluetoothSocket.getOutputStream();
                    sendData("u");
                    SystemClock.sleep(1000);

                    // Get the input stream associated with this socket.
                    inputStream = bluetoothSocket.getInputStream();

                    // Reads bytes from this stream and stores them in the byte array
                    byte[] msgBuffer = new byte[1024];
                    inputStream.read(msgBuffer);

                    medicoesRecebidas.setText(new String(msgBuffer));

                } catch (IOException e){
                    Log.e("ERROR", "O erro foi" + e.getMessage());
                    Toast.makeText(getApplicationContext(), "Mensagem não recebida", Toast.LENGTH_LONG).show();
                }
                break;

            default:
                Toast.makeText(getApplicationContext(), "Botão de medição indisponível", Toast.LENGTH_LONG).show();
                break;
        }
    } else {
        Toast.makeText(getApplicationContext(), "Bluetooth não está conectado", Toast.LENGTH_LONG).show();
    }
}
}
}

```