

```

package com.dai.db;

import org.json.simple.JSONArray; import java.sql.PreparedStatement; import
java.sql.ResultSet; import java.sql.SQLException; import com.dai.webServer.Conexao.Conexao;
import com.dai.webServer.Objects.Analytics;

import org.json.simple.JSONObject;

public class AnalyticsDB {
    java.sql.Connection con;

    Analytics a = new Analytics();

    public AnalyticsDB() {
        con = Conexao.fazConexao();
    }

    public String readEmail(String topic){
        String value = null;

        PreparedStatement stmt = null;
        ResultSet rs = null;

        try {

            stmt = con.prepareStatement("SELECT email from v_contas where sensor_id =?");

            stmt.setString(1, topic);

            rs = stmt.executeQuery();

            System.out.println(rs);

            while (rs.next()) {

                System.out.println(rs.getString("email"));

                value = rs.getString("email");
                System.out.println(value);
            }

            return value;
            /* return value; */

        } catch (SQLException ex) {

```

```

    } finally {
        Conexao.fechaConexao(con, stmt, rs);
        return value;
    }

}

public String updateName(String name, String sensor, Long id_division){
    String value = null;
    System.out.println(name);

    System.out.println(id_division);
    PreparedStatement stmt = null;
    ResultSet rs = null;

    try {

        stmt = con.prepareStatement("UPDATE division SET name=? , sensor_id= ? where id_d

        stmt.setString(1, name);
        stmt.setString(2, sensor);

        stmt.setLong(3, id_division);

        System.out.println(stmt);
        stmt.executeUpdate();

    return value;
        /* return value; */

    } catch (SQLException ex) {
        System.out.println("deu erro "+ ex);

    } finally {
        Conexao.fechaConexao(con, stmt, rs);
        return value;
    }

}

public String updateArm(String armed, Long id_division){ String value = null;
System.out.println(armed);

```

```

        System.out.println(id_division);
        PreparedStatement stmt = null;
        ResultSet rs = null;

        try {

            stmt = con.prepareStatement("UPDATE division SET armed=? where id_division = ?");

            stmt.setString(1, armed);

            stmt.setLong(2, id_division);

            System.out.println(stmt);
            stmt.executeUpdate();

            return value;
            /* return value; */

        } catch (SQLException ex) {
            System.out.println("deu erro caralho"+ ex);
        } finally {
            Conexao.fechaConexao(con, stmt, rs);
            return value;
        }
    }
}

```

```

public String updateDivision(String id_sensor, String id_division){

```

```

String value = null;
System.out.println(id_sensor);

System.out.println(id_division);
PreparedStatement stmt = null;
ResultSet rs = null;

try {

    stmt = con.prepareStatement("UPDATE division SET sensor_id=? where id_division = ?");

    stmt.setString(1, id_sensor);

    stmt.setString(2, id_division);

    System.out.println(stmt);
    stmt.executeUpdate();

return value;
    /* return value; */

} catch (SQLException ex) {
    System.out.println("deu erro caralho"+ ex);

} finally {
    Conexao.fechaConexao(con, stmt, rs);
    return value;
}

}

public JSONObject isArmed(String topic){
    String value = null;
    JSONObject end = new JSONObject();

    PreparedStatement stmt = null;
    ResultSet rs = null;

    try {

        stmt = con.prepareStatement("SELECT * from division where sensor_id =?");
    
```

```

        stmt.setString(1, topic);

        rs = stmt.executeQuery();

        System.out.println(rs);

        while (rs.next()) {

            System.out.println(rs.getString("armed"));

            System.out.println(rs.getString("id_division"));

            end.put("armed", rs.getString("armed"));

            end.put("id_division", rs.getString("id_division"));

        }

        return end;
        /* return value; */

    } catch (SQLException ex) {

    } finally {
        Conexao.fechaConexao(con, stmt, rs);
    }
    return end;
}

```

```

public Long findIdDivision(String topic){
    Long value = null;

    PreparedStatement stmt = null;
    ResultSet rs = null;

    try {

```

```

        stmt = con.prepareStatement("SELECT id_division from v_contas where sensor_id = ?");

        stmt.setString(1, topic);

        rs = stmt.executeQuery();

        System.out.println(rs);

        while (rs.next()) {

            System.out.println(rs.getLong("id_division"));

            value = rs.getLong("id_division");
            System.out.println(value);
        }

        return value;
        /* return value; */

    } catch (SQLException ex) {

    } finally {
        Conexao.fechaConexao(con, stmt, rs);
        return value;
    }

}

```

```

public String armed(String armed){
    String value = null;

    PreparedStatement stmt = null;

```

```

ResultSet rs = null;

try {

    stmt = con.prepareStatement("SELECT armed from v_contas where sensor_id = ?");

    stmt.setString(1, armed);

    rs = stmt.executeQuery();

    System.out.println(rs);

    while (rs.next()) {

        System.out.println(rs.getString("armed"));

        value = rs.getString("armed");
        System.out.println(value);
    }

    return value;
    /* return value; */

} catch (SQLException ex) {

} finally {
    Conexao.fechaConexao(con, stmt, rs);
    return value;
}

}

```

```

public String approve(String message, String topic){
    String value = null;

    PreparedStatement stmt = null;
    ResultSet rs = null;

    try {

        stmt = con.prepareStatement("SELECT username from v_contas where sensor_id =? and id =?");

        stmt.setString(1, topic);

        stmt.setString(2, message);

        rs = stmt.executeQuery();

        System.out.println(rs);

        while (rs.next()) {

            System.out.println(rs.getString("username"));

            value = rs.getString("username");
            System.out.println(value);
        }

        return value;
        /* return value; */

    } catch (SQLException ex) {

    } finally {
        Conexao.fechaConexao(con, stmt, rs);
        return value;
    }
}

```



```

    }

}

public String read(String device, String dataI, String dataF) {
    String value = null;

    PreparedStatement stmt = null;
    ResultSet rs = null;

    try {

        stmt = con.prepareStatement("SELECT AVG(JSON_EXTRACT(attr, '$.measurements.temperature')) AS avgTemp FROM sensor_data WHERE device = ? AND dataI = ? AND dataF = ?");

        stmt.setString(1, device);
        stmt.setString(2, dataI);
        stmt.setString(3, dataF);

        rs = stmt.executeQuery();

        System.out.println(rs);

        while (rs.next()) {

            System.out.println(rs.getString("temp"));

            value = rs.getString("temp");
            System.out.println(value);
        }
        return value;
        /* return value; */

    } catch (SQLException ex) {

    } finally {
        Conexion.fechaConexion(con, stmt, rs);
        return value;
    }

}

public String readHumidade(String device, String dataI, String dataF) {

```

```

String value = null;

PreparedStatement stmt = null;
ResultSet rs = null;

try {

    stmt = con.prepareStatement("SELECT AVG(JSON_EXTRACT(attr, '$.measurements.humidity

    stmt.setString(1, device);
    stmt.setString(2, dataI);
    stmt.setString(3, dataF);

    rs = stmt.executeQuery();

    System.out.println(rs);

    while (rs.next()) {

        System.out.println(rs.getString("AVG(JSON_EXTRACT(attr, '$.measurements.humidity

        value = rs.getString("AVG(JSON_EXTRACT(attr, '$.measurements.humidity')"));
        System.out.println(value);
    }

return value;
    /* return value; */

} catch (SQLException ex) {
} finally {
    Conexao.fechaConexao(con, stmt, rs);
    return value;
}

}

public JSONObject returnGraph(String device, String dataI, String dataF) {

    PreparedStatement stmt = null;
    ResultSet rs = null;
    JSONObject end = new JSONObject();
    JSONArray temp = new JSONArray();

```

```

JSONArray data = new JSONArray();
Integer i = 0;

try {

    stmt = con.prepareStatement("SELECT JSON_EXTRACT(attr, '$.measurements.temperature')");

    stmt.setString(1, device);
    stmt.setString(2, dataI);
    stmt.setString(3, dataF);

    rs = stmt.executeQuery();

    while (rs.next()) {
        temp.add(i, rs.getString("temp"));

        data.add(i, rs.getString("data"));
        System.out.println(rs.getString("temp"));
    }
    end.put("temp" , temp);

    end.put("data" , data);
    //return onalua;

} catch (SQLException ex) {

} finally {
    Conexao.fechaConexao(con, stmt, rs);

    return end;
}

}

//Gravar entradas em casa na BD

public void insertDB(String message , String outcome){

    PreparedStatement stmt = null;

    try {
        stmt = con.prepareStatement("INSERT INTO entrance (tag,account_id)VALUES(?,?)");

        stmt.setString(1, message);

```

```

        stmt.setString(2, outcome);

        stmt.executeUpdate();

    } catch (SQLException ex) {
        System.out.println(ex);
    } finally {
        Conexao.fechaConexao(con, stmt);
    }
}

public void addCard(Long id_user , String tag){

    System.out.println(id_user);

    System.out.println(tag);
    PreparedStatement stmt = null;

    try {
        stmt = con.prepareStatement("UPDATE account set id_card = ? where id_user= ?");

        stmt.setString(1, tag);
        stmt.setLong(2, id_user);

        stmt.executeUpdate();

    } catch (SQLException ex) {
        System.out.println(ex);
    } finally {
        Conexao.fechaConexao(con, stmt);
    }
}

//Gravar entradas NÃO AUTORIZADAS em casa na BD
public void insertDBNot(String message , String outcome){

    PreparedStatement stmt = null;

    try {
        stmt = con.prepareStatement("INSERT INTO entrance (tag,account_id)VALUES(?,?)");

        stmt.setString(1, message);
        stmt.setString(2, "Conta não autorizada");
    }
}

```

```

        stmt.executeUpdate();

    } catch (SQLException ex) {
        System.out.println(ex);
    } finally {
        Conexao.fechaConexao(con, stmt);
    }
}

//visualizar entradas NAO ESTA COMPLETAMENTE A FUNCIONAL(ENVIAR DADOS NUM ARRAY CORRETO)

public JSONObject findHouse(Long account) {
    System.out.println(account);
    String value = null;

    PreparedStatement stmt = null;

    ResultSet rs = null;

    JSONObject end = new JSONObject();

    JSONArray id_houses = new JSONArray();

    JSONArray houses = new JSONArray();

    Integer i = 0;

    try {

        stmt = con.prepareStatement("SELECT * FROM house where account_id= ?");

        stmt.setLong(1, account);

        rs = stmt.executeQuery();

```

```

        System.out.println(rs);

        while (rs.next()) {

            id_houses.add(i, rs.getString("id_house"));

            houses.add(i, rs.getString("name"));
        }

        end.put("id_houses" , id_houses);

        end.put("houses" , houses);

    } catch (SQLException ex) {
    } finally {
        Conexao.fechaConexao(con, stmt, rs);
        return end;
    }
}

```

```

public JSONObject findDivision(String email) {
    String value = null;

    PreparedStatement stmt = null;

    ResultSet rs = null;

    JSONObject end = new JSONObject();

    JSONObject end2 = new JSONObject();

    JSONArray id_house = new JSONArray();
    JSONArray id_division = new JSONArray();

    JSONArray division = new JSONArray();

    JSONArray house = new JSONArray();

    Integer i = 0;

    try {

        stmt = con.prepareStatement("SELECT * FROM v_contas where email= ?");

        stmt.setString(1, email);

        rs = stmt.executeQuery();

        System.out.println(rs);

        while (rs.next()) {

            id_house.add(i, rs.getString("id_house"));

            id_division.add(i, rs.getString("id_division"));

```

```

        division.add(i,rs.getString("division"));

        house.add(i, rs.getString("house"));
    }

    end.put("id_house" , id_house);

    end.put("id_division" , id_division);

    end.put("division" , division);

    end.put("house" , house);
end2.put("array", end);


    } catch (SQLException ex) {
    } finally {
        Conexao.fechaConexao(con, stmt, rs);
        return end;
    }

}

//Envia o regesto de todas as entradas
public JSONObject readEntradas(String idHouse) {

    String value = null;

    PreparedStatement stmt = null;
    ResultSet rs = null;

    JSONObject end = new JSONObject();
    JSONArray idaccount = new JSONArray();
    JSONArray nome = new JSONArray();
    JSONArray data = new JSONArray();
    Integer i = 0;

    try {

        System.out.println(idHouse);

```



```

        stmt = con.prepareStatement("SELECT * FROM v_entrance where ID_HOUSE= ?");

        stmt.setString(1, idHouse);
        rs = stmt.executeQuery();

        System.out.println(rs);

        while (rs.next()) {

            idaccount.add(i, rs.getString("username"));
            data.add(i, rs.getString("reg_date"));
            nome.add(i, rs.getString("name"));

        }

        end.put("username" , idaccount);
        end.put("reg_date" , data);
        end.put("name" , nome);

    } catch (SQLException ex) {
    } finally {
        Conexao.fechaConexao(con, stmt, rs);
        return end;
    }

}

// NAO ESTA A FUNCIONAR, ENVIA TODOS OS SENSORES (ENVIAR DADOS NUM ARRAY CORRETO)
public JSONObject readSensor() {

    String value = null;

    PreparedStatement stmt = null;
    ResultSet rs = null;

    JSONObject end = new JSONObject();
    JSONArray teste = new JSONArray();
    JSONArray id = new JSONArray();
    JSONArray type = new JSONArray();
    JSONArray activ = new JSONArray();
    Integer i = 0;

```

```

try {

    stmt = con.prepareStatement("SELECT * from sensor");

    rs = stmt.executeQuery();

    while (rs.next()) {

        id.add(i, rs.getString("id_sensor"));
        type.add(i, rs.getString("type"));
        activ.add(i, rs.getString("activ"));
    }

    end.put("id_sensor" , id);
    end.put("type" , type);
    end.put("activ" , activ);

} catch (SQLException ex) {
} finally {
    Conexao.fechaConexao(con, stmt, rs);
    return end;
}

}

}

```