Roteiro de Laboratório – Aula 02
Tema: Comunicação OPC UA/Interfaces de Comunicação/Programação Script

Sistema de Controle de Nível (Simulado em Python) com Interface OPC UA e Armazenamento em Bancos de Dados

---

Servidor OPC e Cliente OPC

Todas as especificações OPC são baseadas no modelo de OPC cliente/servidor.
A especificação descreve a relação entre duas aplicações executadas em um sistema operacional no qual um aplicativo – o ***Cliente OPC*** faz uma solicitação de serviço de outro aplicativo o ***Servidor OPC***, que atende à solicitação.

Embora o modelo cliente/servidor OPC possa ser usado dentro de um único computador, quando usado em uma rede, fornece uma infraestrutura versátil e modular que oferece flexibilidade, interoperabilidade e escalabilidade. Este modelo é diferente de outras arquiteturas distribuídas como Master/Slave ou Redes Peer-to-peer.

Um Servidor OPC é, portanto, um aplicativo (software), que foi escrito segundo os padrões da especificação OPC. Um Servidor OPC irá responder aos pedidos e fornecer dados a um ou mais clientes OPC em um padrão, de forma consistente. Qualquer cliente OPC compatível pode interagir e solicitar acesso aos dados, independentemente do fornecedor ou o sistema que fornece os dados.

Uma aplicação OPC Cliente geralmente é utilizada por sistemas SCADA para acesso aos dados industriais provenientes de CLPs, instrumentos de campo, sensores e atuadores.

As principais especificações utilizadas por sistemas SCADA são:

- OPC Data Access (OPC DA);
- OPC Historical Data Access (OPC HDA); - OPC Alarms and Events (OPC AE).
- OPC UA (Unified Architecture).

Fonte: http://www.matrikonopc.com

Materiais necessários:

1) ***Interpretador Python (https://www.python.org) – Versão igual a superior 3.8***
2) ***Kassl-dOPCDAClient (Cliente OPC) (https://www.kassl.de/opc/download.shtml)***
3) ***Visual Studio Code (https://code.visualstudio.com)***
4) ***DBeaver Community (https://dbeaver.io/download/)***

Etapas de Execução:

Etapa 1 (Sistema de Controle de Nível)

1) Descompacte a pasta LAB02 e utilize a Ferramenta Visual Studio Code para abrir a pasta

2) Instale o Interpretador Python (versão 3.8 ou superior 64bit)

3) Instale as bibliotecas Python.
   a) Abrir um prompt de comando com status de administrador.
   b) Utilize o comando pip install -r requirements.txt
   (este comando irá instalar as bibliotecas: opcua, pygame, flask e pysqlite3)
   Obs.: É possível também executar individualmente
   pip install opcua
   pip install <biblioteca>

4) Abra o Visual Studio Code e selecione a opção Arquivo -> Abrir pasta. Selecione a pasta LAB03, conforme a Figura 1.

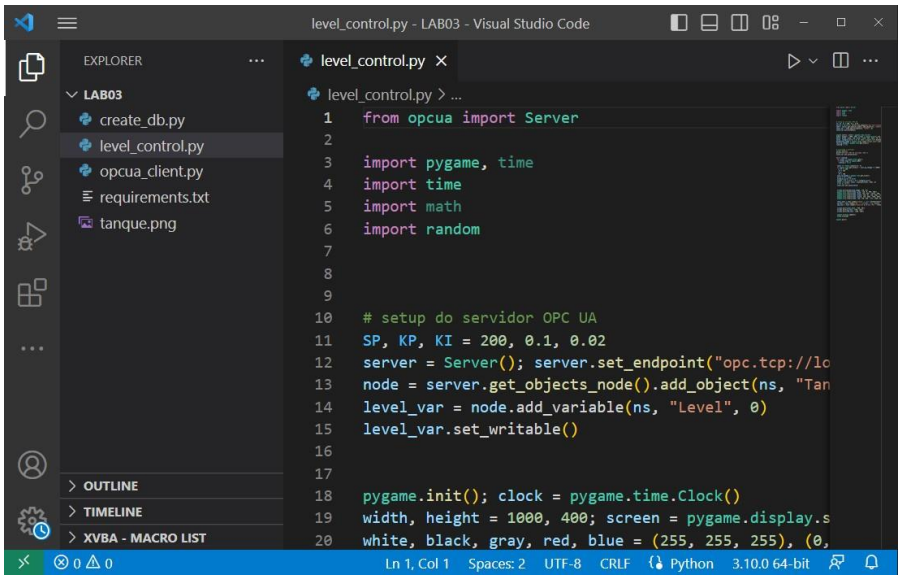

Figura 1

5) Abra um prompt de comando e execute o simulador por meio do arquivo level_control.py. Utilize o comando: **python level_control.py**

6) Será aberta a janela do Processo de Controle de Nível (em ambiente simulado), conforme a Figura 2. Este ambiente simulado faz o Controle de Nível de um tanque por meio de um Controlador PI. O valor da variável nível é disponibilizado também por meio de um Servidor OPC UA.

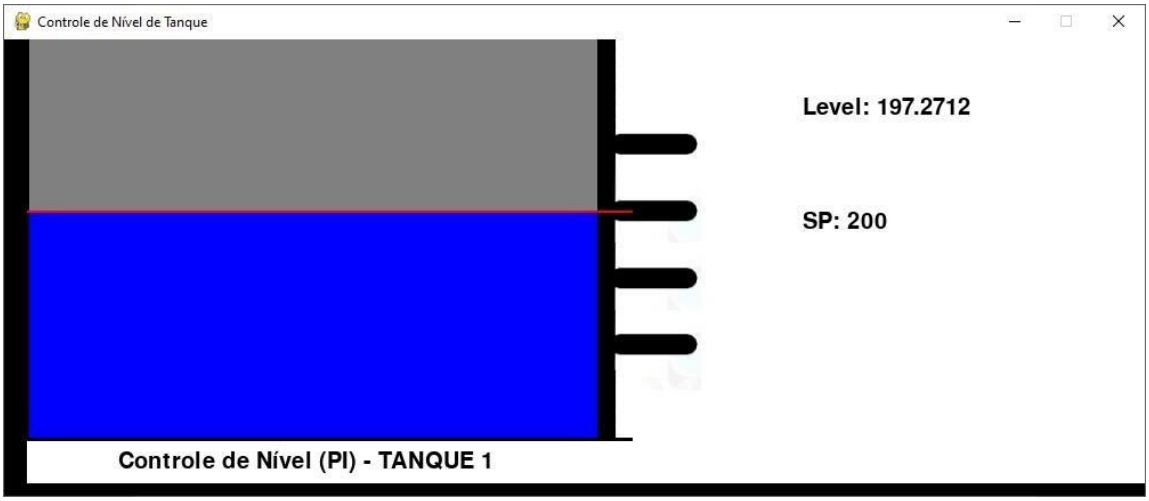

Figura 2

Etapa 2 (Monitoramento do Nível utilizando-se OPC UA)

1) Para monitorar o nível, será utilizada uma interface de supervisão baseada em um cliente OPC UA (Kassl-dOPCDAClient). Para isso, abra o Software, Selecione a opção File -> Connect to Server e selecione o Servidor conforme a Figura 3.

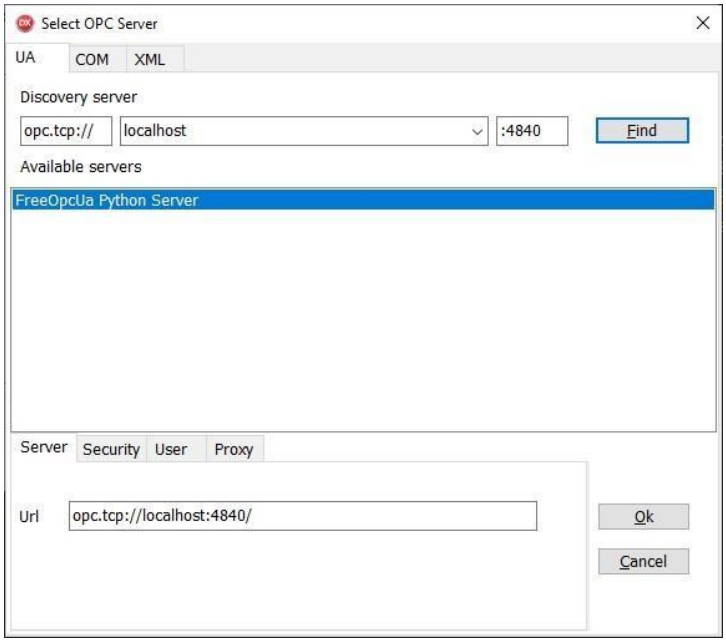

Figura 3

2) Após conectado, adicione um Grupo (Grupo 1), utilize as configurações padrão, e selecione dentro da Árvore **OPC UA Root/Tank/Level**. As Figuras 4, 5, 6 e 7 ilustram esta etapa.

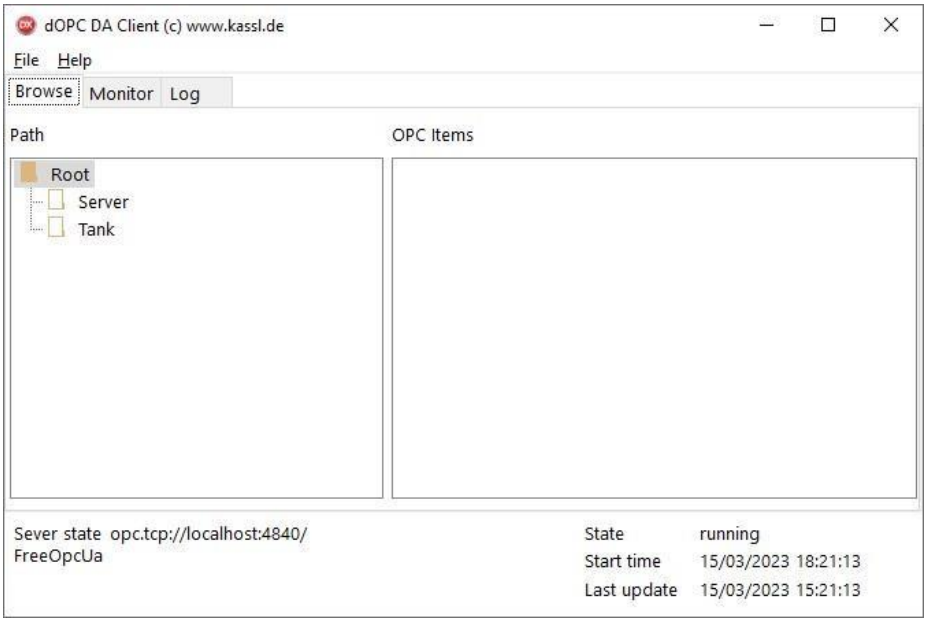

Figura 4

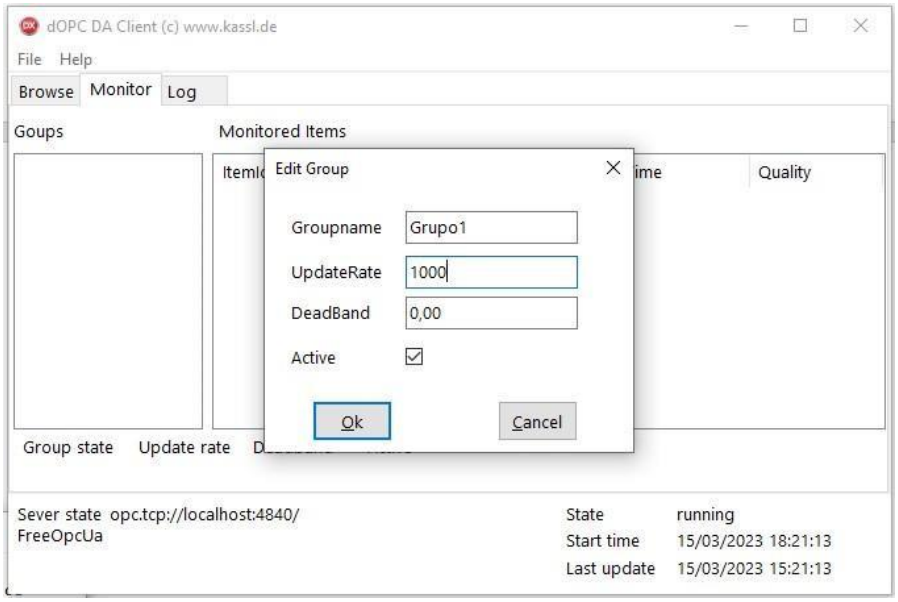

Figura 5

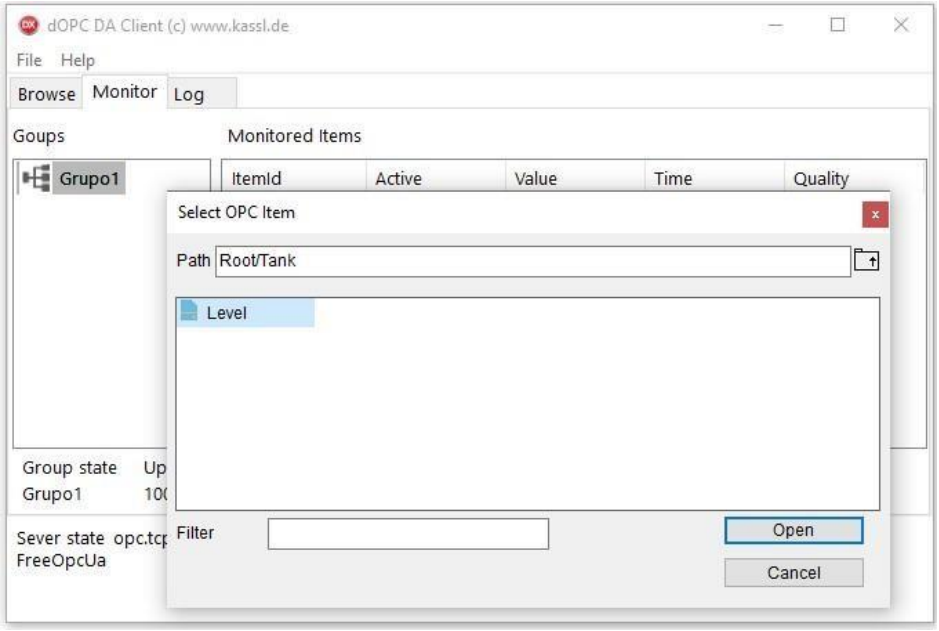

Figura 6

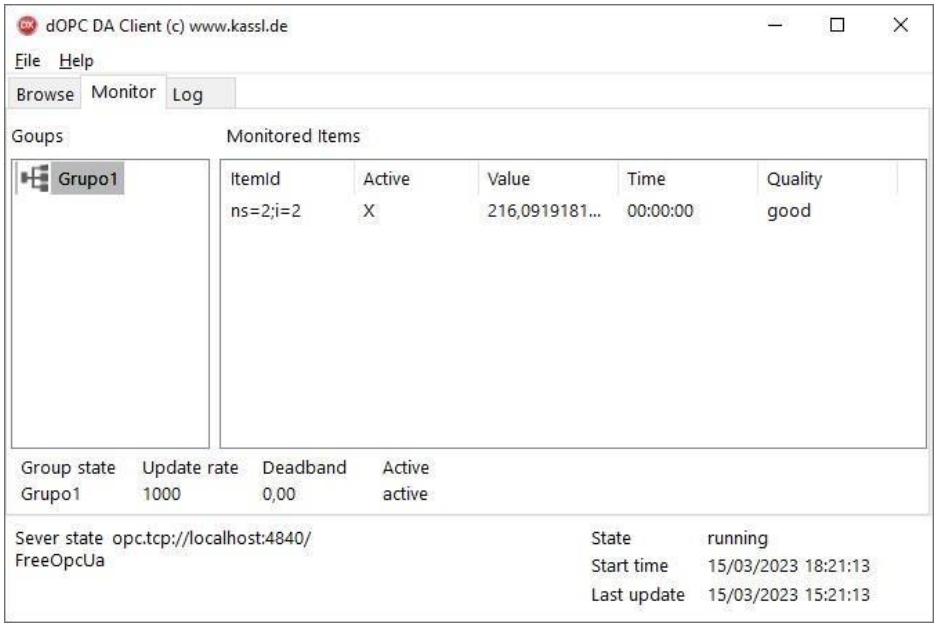

Figura 7

Etapa 3 (Criação do Banco de Dados – pv_database.db)

1) Dentro do Visual Studio, abra o arquivo **create_db.py**, conforme a Figura 8.

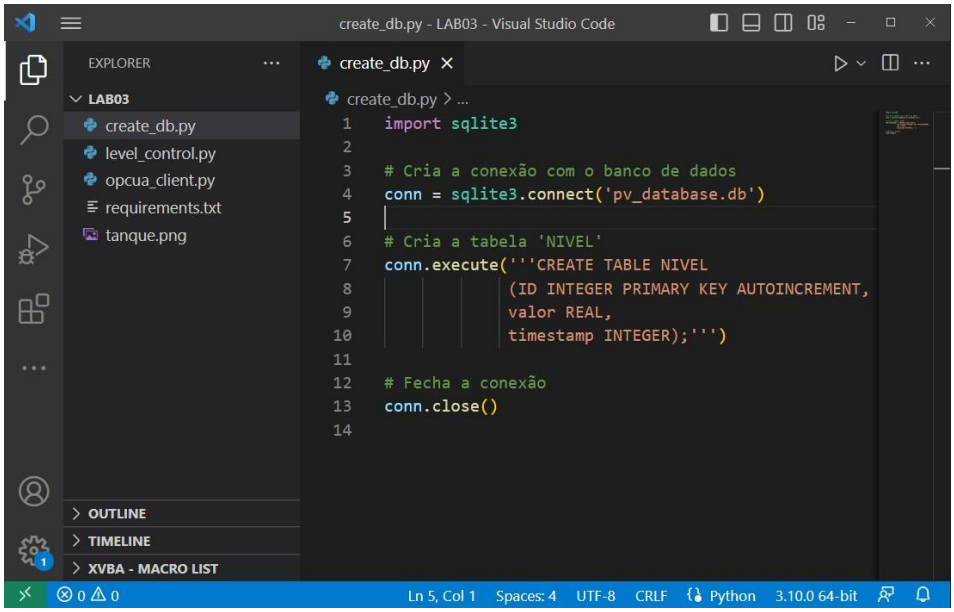

Figura 8

2) Abra um prompt de comando e execute o script que cria o banco de dados Utilize o comando: **python create_db.py**

3) Certifique-se de que o arquivo **pv_database.db** foi criado na mesma pasta.

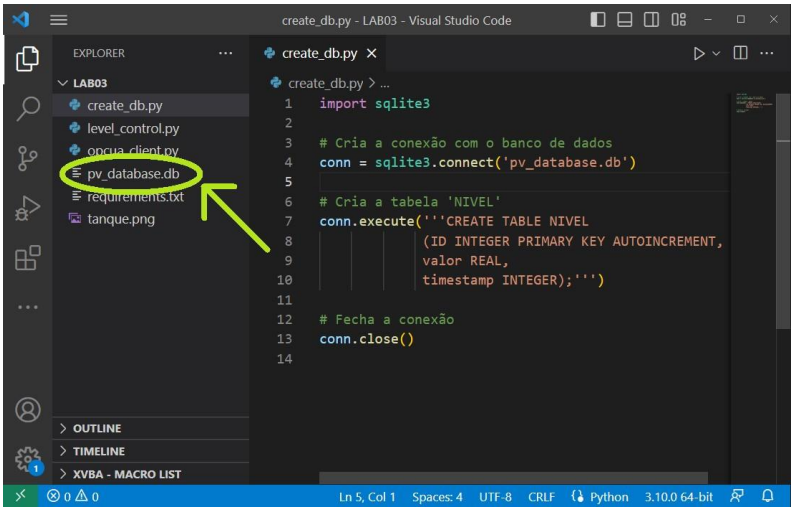

Figura 9

Etapa 4 (Conexão com o Banco de Dados por meio da IDE DBeaver)

1) Para Conexão com o banco de dados, utilize a Ferramenta DBeaver Community. Abra a ferramenta, conforme a Figura 10.

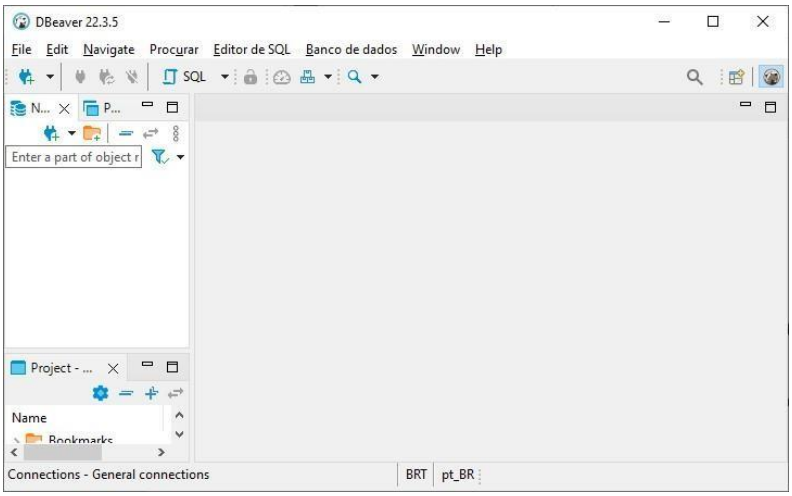

Figura 10

2) Selecione o Menu Banco de dados -> Nova Conexão de Banco de Dados e selecione SQLite e clique em Avançar, conforme a Figura 11.

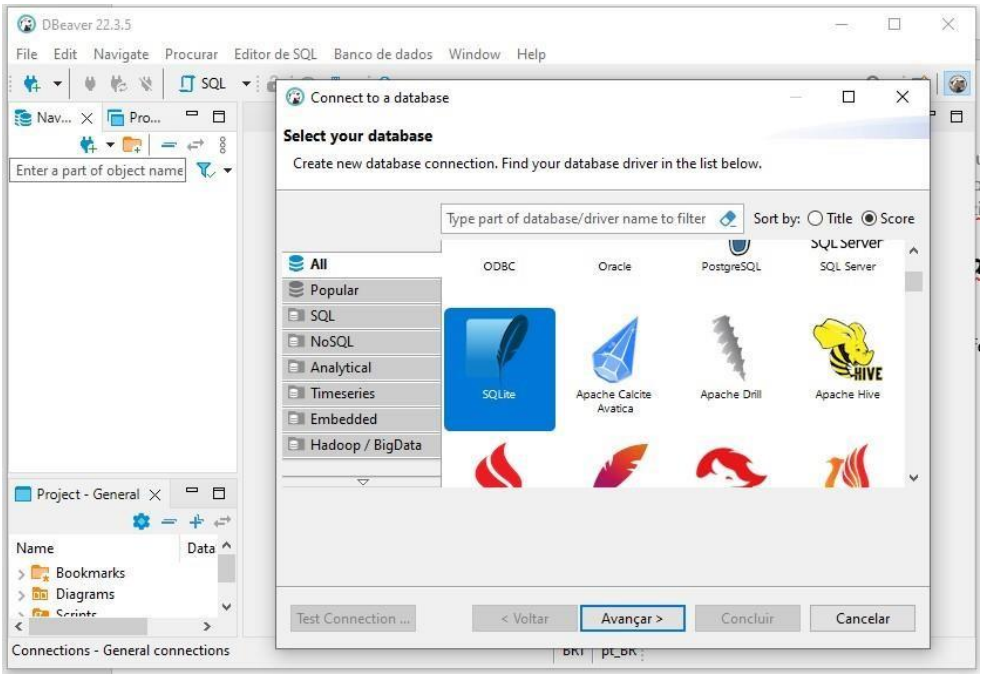

Figura 11

3) Selecione o arquivo **pv_database.db**, dentro da pasta LAB02 e clique em Abrir, conforme ilustrado na Figura 12.

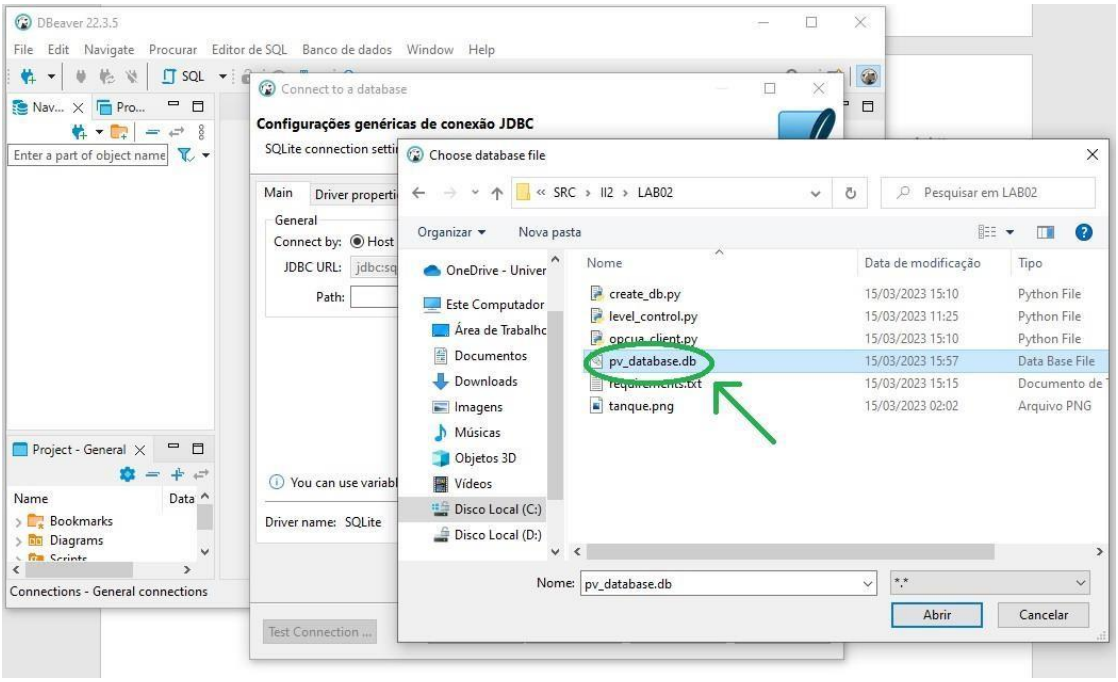

Figura 12

4) Expanda a árvore do lado esquerdo e execute o comando de leitura da tabela por meio do comando SQL: **SELECT * FROM NIVEL** , conforme ilustrado na Figura 13.

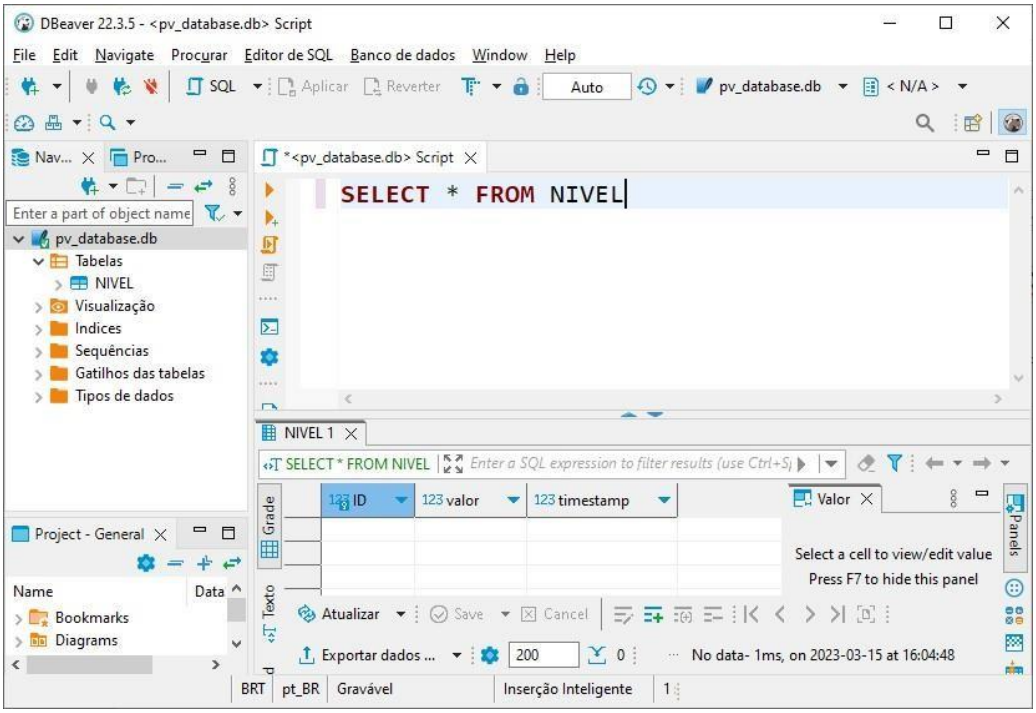

Figura 13

Etapa 5 (Criação da Interface de Comunicação Cliente OPCUA, Publicação Web e Inserção do valor lido no Banco de Dados)

1) Dentro do Visual Studio, abra o arquivo **opcua_client.py**, conforme a Figura 14.

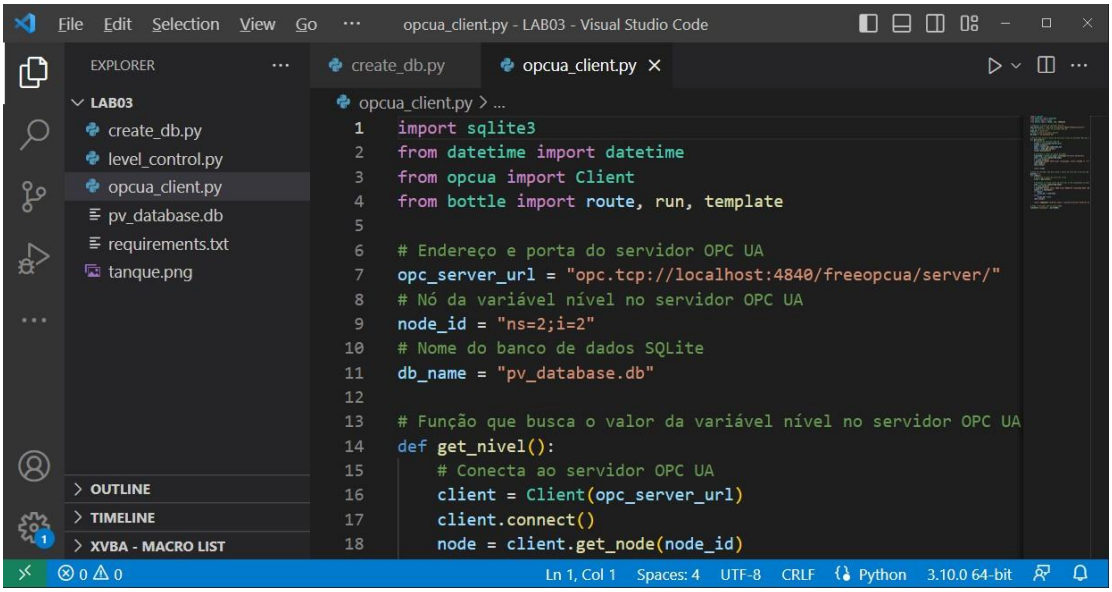

Figura 14

2) Abra um prompt de comando e execute o script que cria o banco de dados Utilize o comando: **python opcua_client.py**

3) Em seguida, abra um navegador web e acesse o site: **http://localhost:8082/** , conforme a Figura 15.

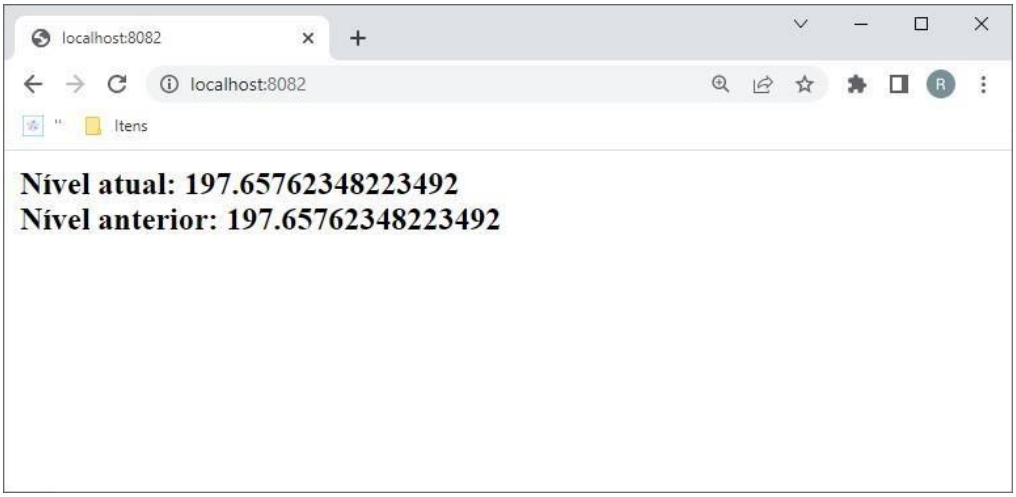

Figura 15

4) Clique algumas vezes no botão **Refresh** do navegador, conforme a Figura 16.

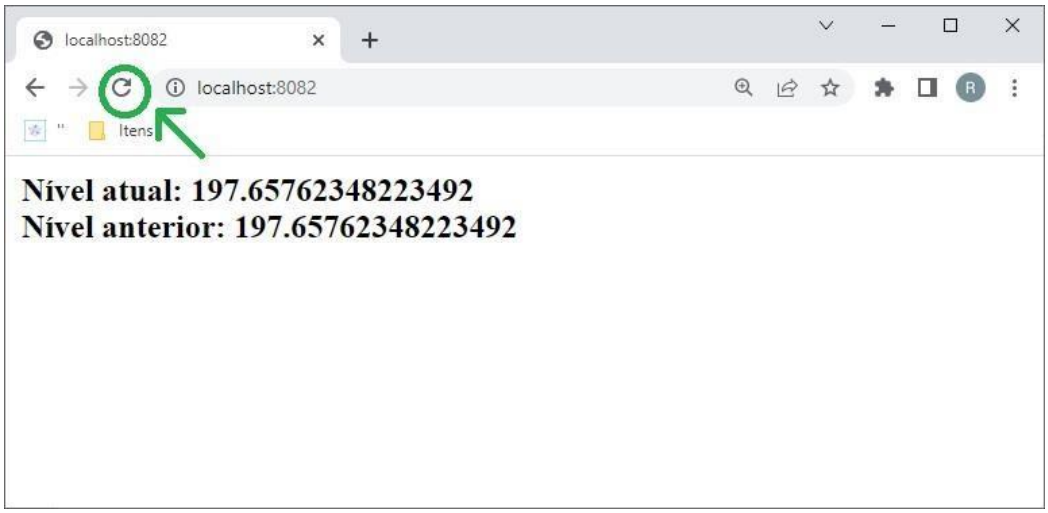

Figura 16

5) Retorne à Ferramenta DBeaver e execute novamente o comando de leitura da tabela no Banco de Dados, comando SQL: **SELECT * FROM NIVEL** , conforme a Figura 17.

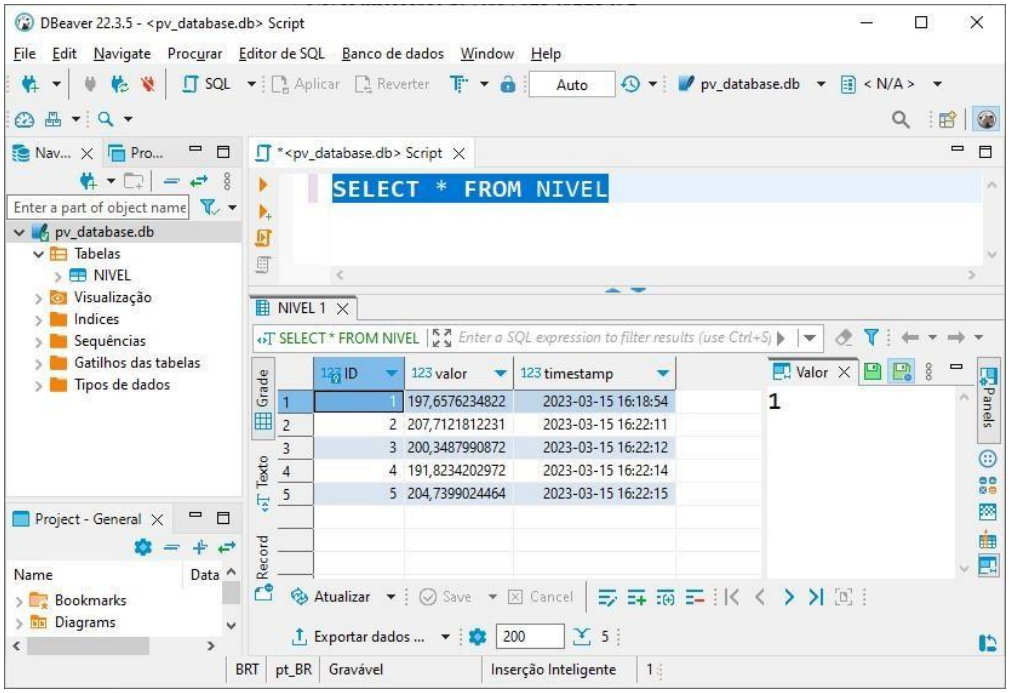

Figura 17

6) Dessa forma, encerra-se o ciclo de monitoramento.

A Figura 18, a seguir, mostra a arquitetura do sistema:

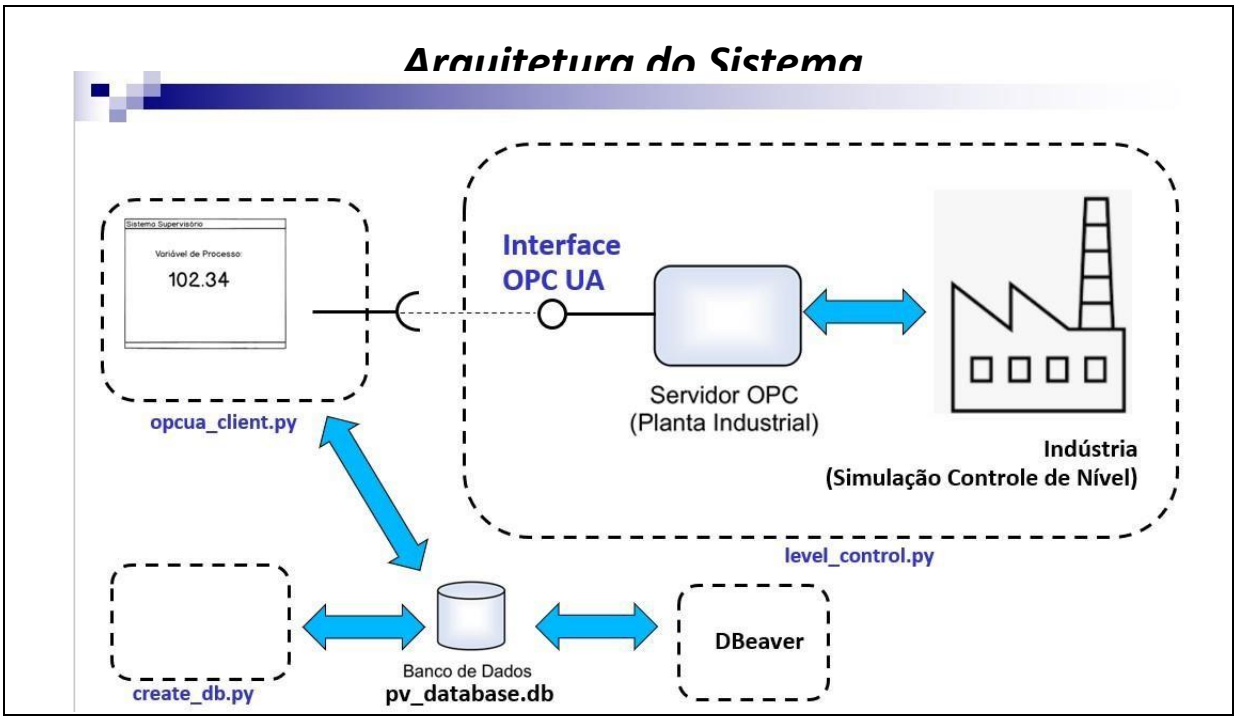

Figura 18

Etapa 6 (Usando-se o DBeaver para fazer alterações no Banco de Dados)

1) Utilizando-se como referência a Figura 16, faça uma captura (amostragem de 50 dados). Dessa forma, a tabela NIVEL deverá conter 50 registros. Certifique-se de que não há mais e nem menos registros, mas exatamente 50. Caso você tenha capturado mais registros, utilize dentro do DBeaver o comando SQL: **DELETE.....** para manter a amostra com exatamente 50 registros.

2) Dentro do DBeaver, utilize o comando SQL para criar uma tabela ESTUDANTE:

   CREATE TABLE ESTUDANTE (ID INTEGER PRIMARY KEY AUTOINCREMENT, NOME TEXT, MAT TEXT);

   Veja a Figura 19.

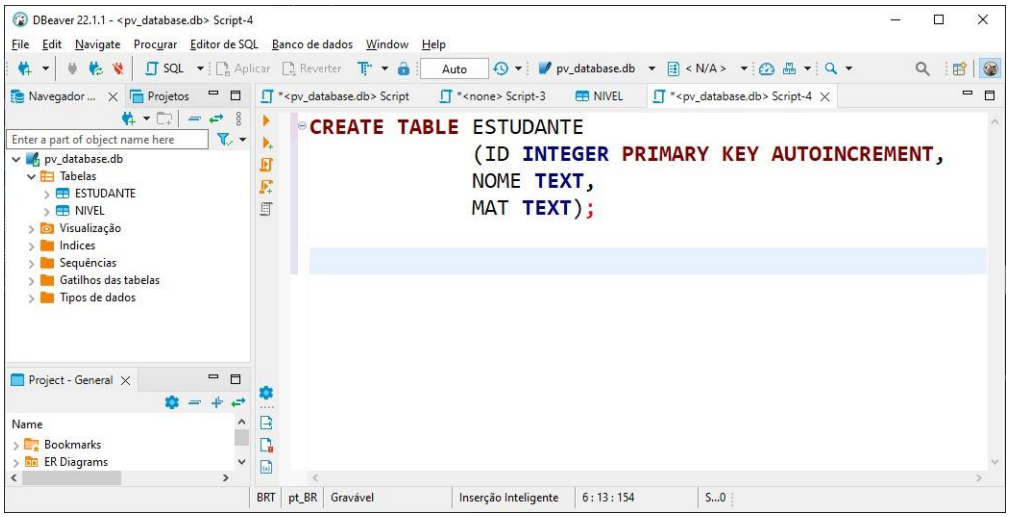

Figura 19

3) Utilize o comando INSERT para inserir o seu Nome e Número de Matrícula

   INSERT INTO ESTUDANTE (nome, MAT) VALUES ('Nome Sobrenome', '123MAT456')

Feche o DBeaver, e envie o arquivo **pv_database.db** no seu local de entrega no MSTeams.