



Roteiro de Laboratório – Aula 03

Tema: Comunicação de um Sistema Supervisório, Armazenamento de Dados e IHM

Motores e1, e2, e3, e4 e e5

Obs.: Equipamentos/Softwares Simulados em Python

Material Necessário ao Laboratório:

1) Softwares:

- *Python 3.8 ou superior instalado;*
- *Navegador Web: Chrome, Edge, Firefox ou outro;*
- *Software DBeaver*

2) Bibliotecas Python:

- *pyModbusTCP, pysqlite3, flask*

Instalação: pip install <nome_da_biblioteca>

1) Utilize como referência os arquivos da pasta LAB04

- ***e1_ModbusServer.py***
- ***e2_ModbusClient_Monitoring.py***
- ***e3_ModbusClient_ChangeParameter.py***
- ***e4_ReadModbus_And_Insert_Into_DB.py***
- ***e5_WebMonitoring.py***

Estes arquivos são os motores e1, e2, e3, e4 e e5 que serão iniciados nesta sequência.



Para iniciar cada um dos motores, execute em um prompt (cada prompt em uma janela):

`python eXXX_nome_do_motor.py`

Obs.: Executar um por vez, seguindo-se de e1 até e5.

2) Descrição dos motores:

- e1_ModbusServer.py

Este motor inicia um servidor Modbus disponibilizando uma área Holding Register (vetor com 10 posições).

- e2_ModbusClient_Monitoring.py

Este motor se conecta ao servidor Modbus e mostra os dados monitorados das 10 posições.

- e3_ModbusClient_ChangeParameter.py

Este motor se conecta ao servidor Modbus e altera a segunda posição do vetor Holding Register.

- e4_ReadModbus_And_Insert_Into_DB.py

Este motor se conecta ao servidor Modbus, lê o valor da segunda posição e armazena este valor em uma tabela do banco de dados, assim como o momento (timestamp) em que este dado foi lido.

e5_WebMonitoring.py

Este motor é um servidor web. Ele se conecta ao banco de dados, lê o último valor inserido, bem como o momento (timestamp) e disponibiliza estes dados em um Dashboard Web, funcionando no seguinte endereço:

<http://localhost:8090>

3) Visualização do Dashboard (IHM Web)

a) Abra um navegador web no seguinte endereço (veja a Figura 1):

<http://localhost:8090>

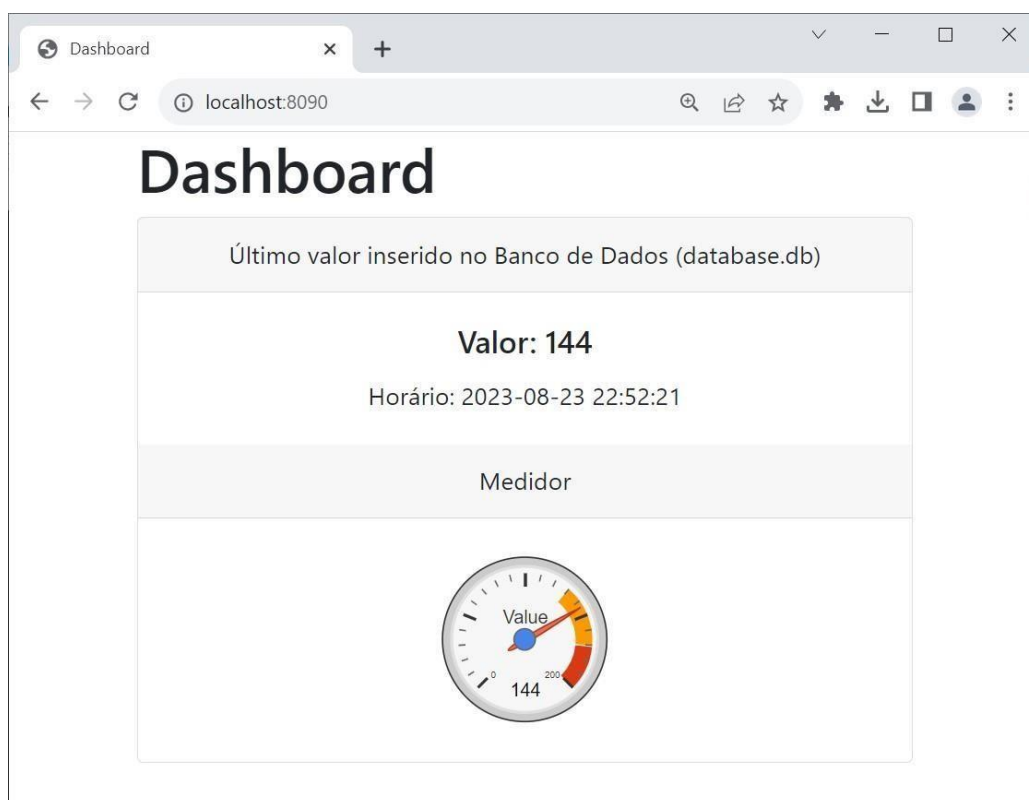


Figura 1

b) Verifique os valores lidos e atualizados a cada 500ms.

4) Execução dos Comandos (CRUD) na Ferramenta DBeaver:

a) Abra a Ferramenta DBeaver.

b) Vá até o menu Banco de dados -> Nova Conexão e selecione SQLite Veja a Figura 2.

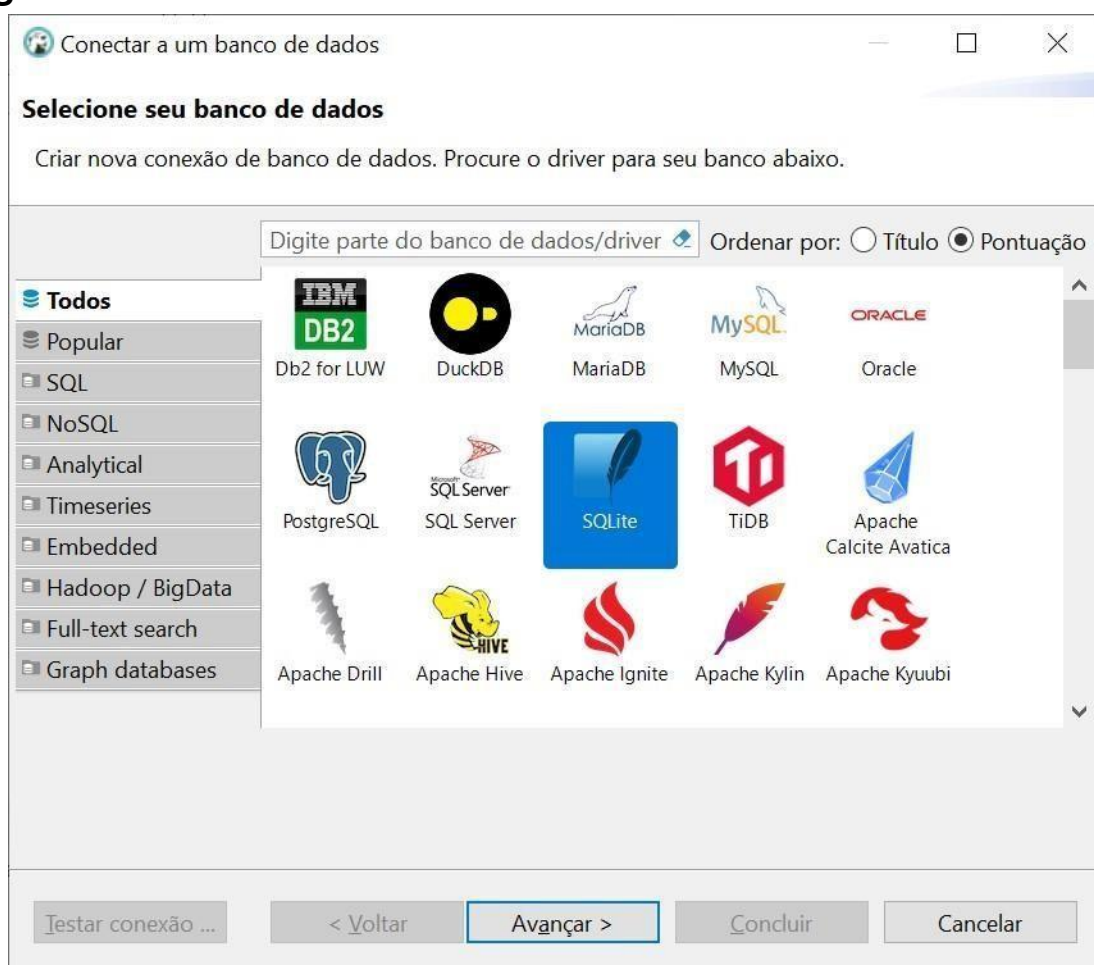


Figura 2

c) Clique em Avançar e escolha o caminho onde se encontra o arquivo “database.db”. Veja Figura 3.

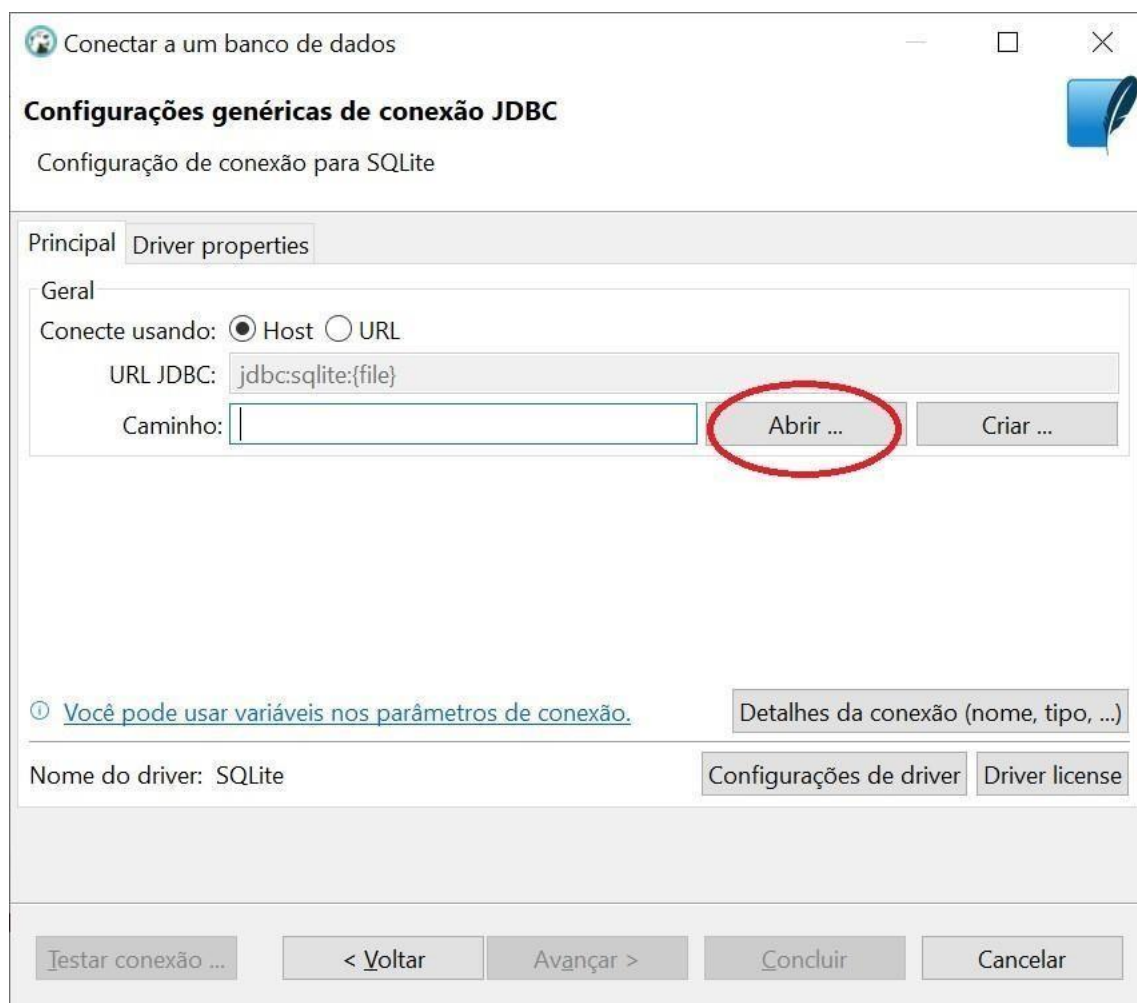


Figura 3

d) Após selecionado o arquivo “database.db”, clique em Avançar e faça o download automático do driver compatível com a base de dados SQLite, caso ele já não esteja instalado. Veja a Figura 4.

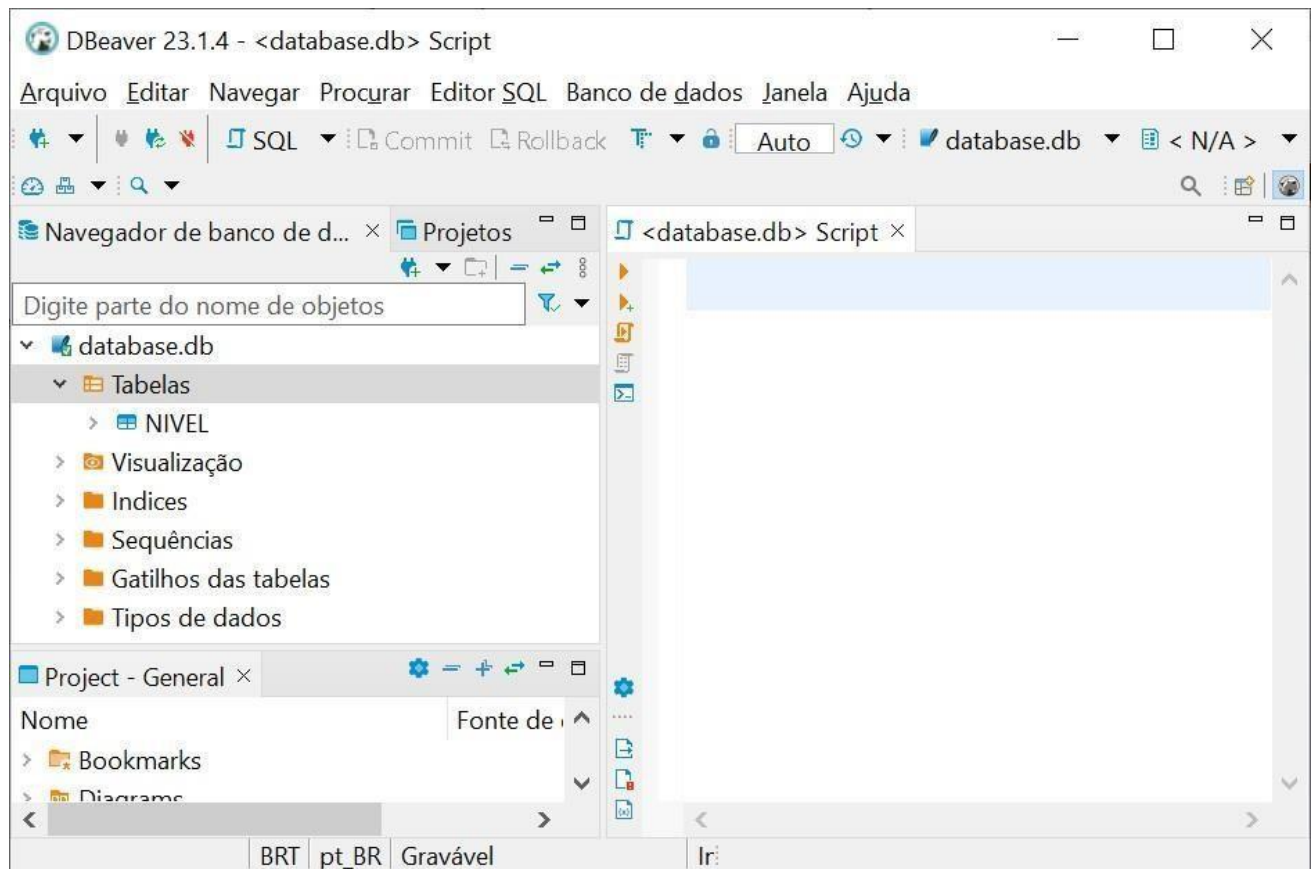


Figura 4

e) Com a ferramenta conectada ao Banco de Dados (arquivo: database.db) faça um envio de comandos para o Banco de Dados, utilizando-se dos conceitos do CRUD (Create, Read, Update e Delete).

Para isso, utilize os comandos SQL:

INSERT, SELECT, UPDATE e DELETE.

Execute variações nos comandos e verifique se os dados foram alterados. Veja a Figura 5.

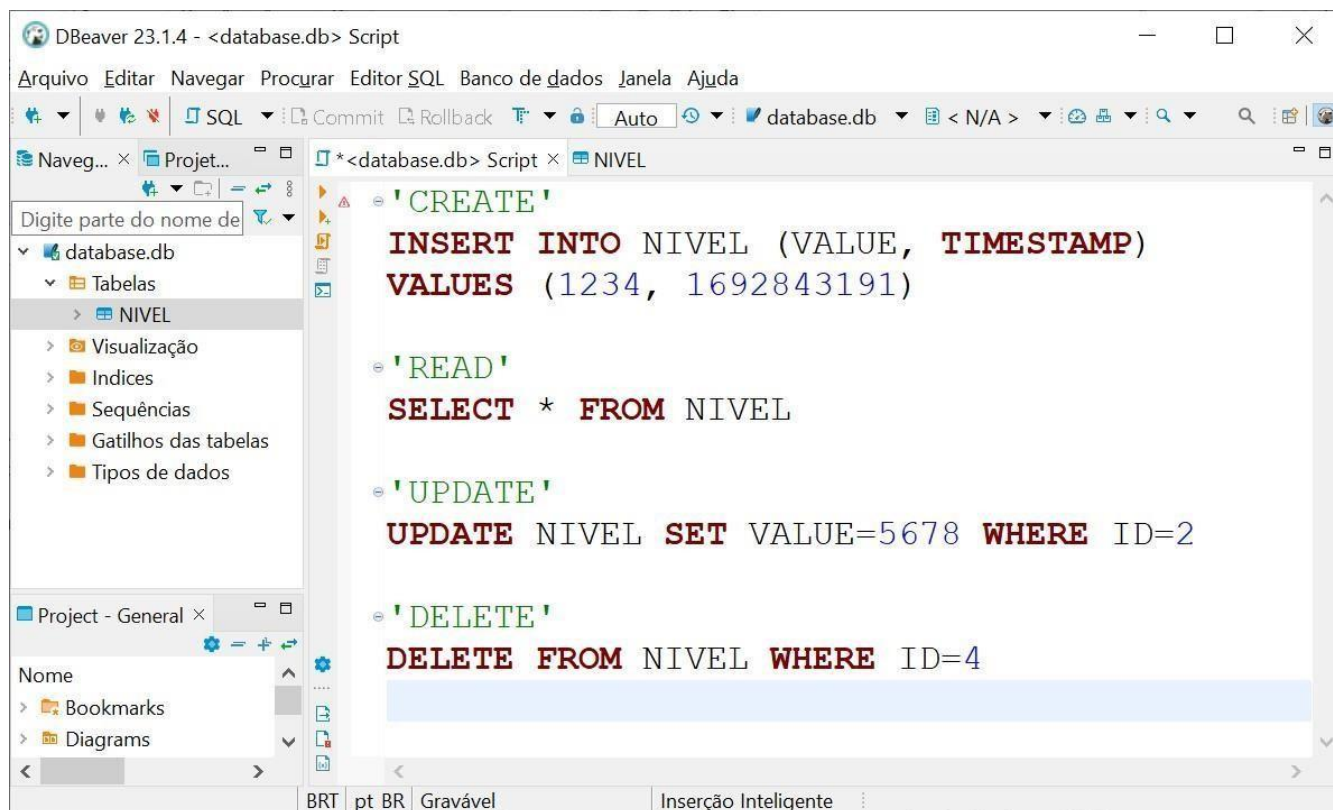


Figura 5

f) Após executar os comandos, feche cada uma das janelas, finalizando-se, dessa forma, todas as tarefas executadas pelos motores (e1, e2, e3, e4 e e5).

5) Entrega:

- Faça um relatório com os prints das telas de execução e uma breve descrição de cada uma das etapas executadas.**
- Entregar o relatório (em formato PDF) na Tarefa do MS Teams até o dia **10/07/2025** às **23:59h**.**