

Universidade Federal Fluminense

Tutorial Python-SCADABR com Modbus

Lizandro de Sousa (lizandrosousa@id.uff.br)

Sumário

1. Instalação das Bibliotecas Necessárias
2. Implementação Modbus no Python
3. Instalação do ScadaBR
4. Implementação da Comunicação do Servidor com o ScadaBR

1. Instalação das Bibliotecas Necessárias

Instalação das Bibliotecas Necessárias:

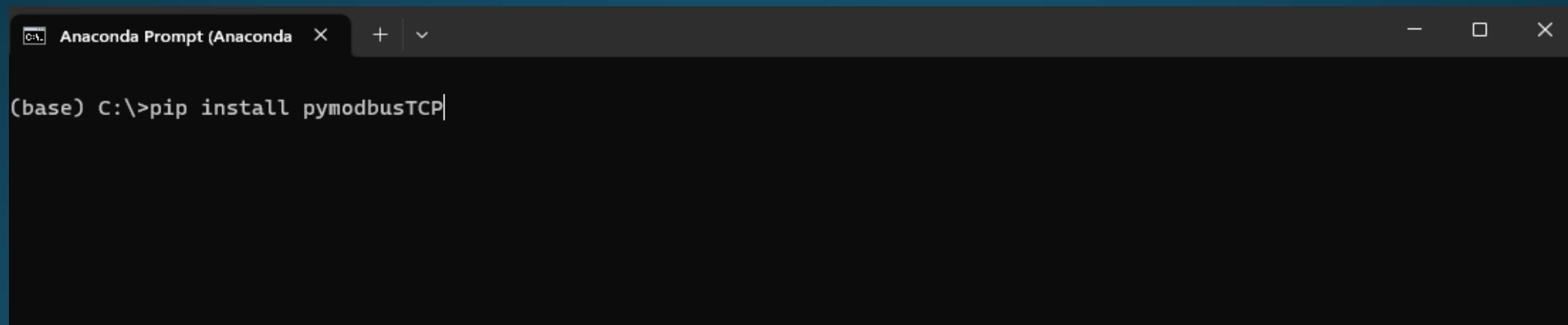
- a. PymodbusTCP
- b. Pymodbus
- c. Numpy
- d. Scipy

1. Instalação das Bibliotecas Necessárias

a. Instalação do PymodbusTCP:

Digitar no terminal do Anaconda o comando:

pip install pymodbusTCP



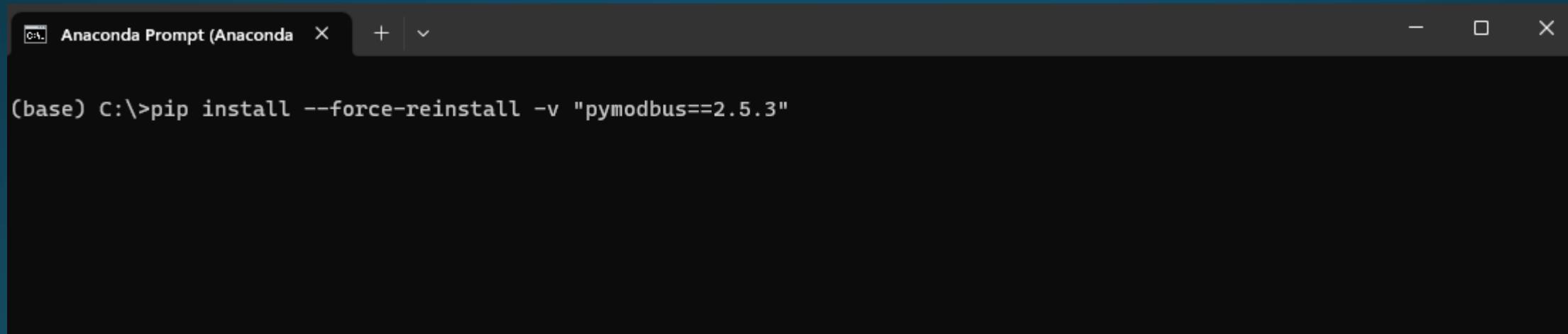
A screenshot of the Anaconda Prompt window. The title bar says "Anaconda Prompt (Anaconda)". The command "(base) C:\>pip install pymodbusTCP" is typed into the prompt area. The window has standard minimize, maximize, and close buttons at the top right.

1. Instalação das Bibliotecas Necessárias

b. Instalação do Pymodbus (versão 2.5.3):

Digitar no terminal do Anaconda o comando:

pip install --force-reinstall -v "pymodbus==2.5.3"



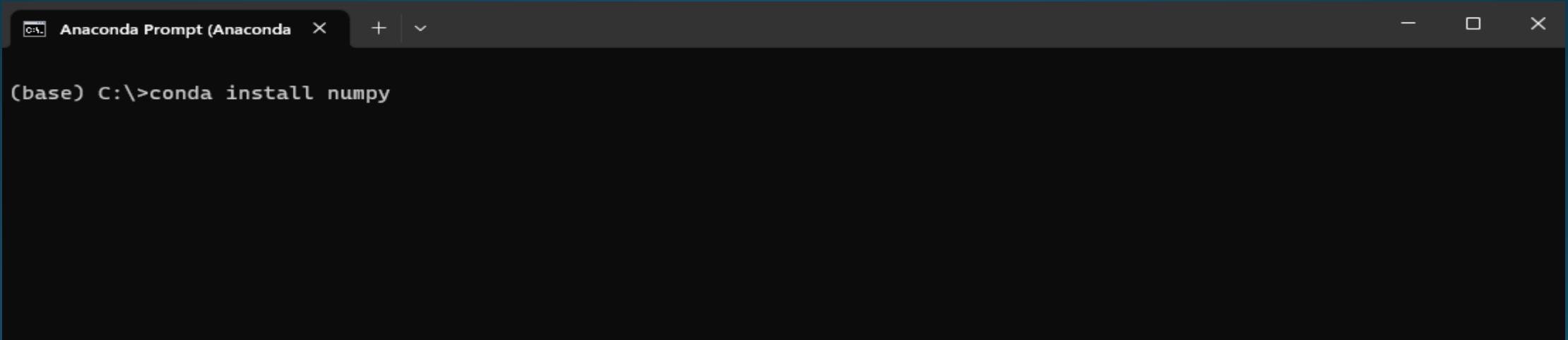
A screenshot of an Anaconda Prompt window titled "Anaconda Prompt (Anaconda)". The window shows a single command being typed in: "(base) C:\>pip install --force-reinstall -v "pymodbus==2.5.3"".

1. Instalação das Bibliotecas Necessárias

c. Instalação do numpy:

Digitar no terminal do Anaconda o comando:

conda install numpy



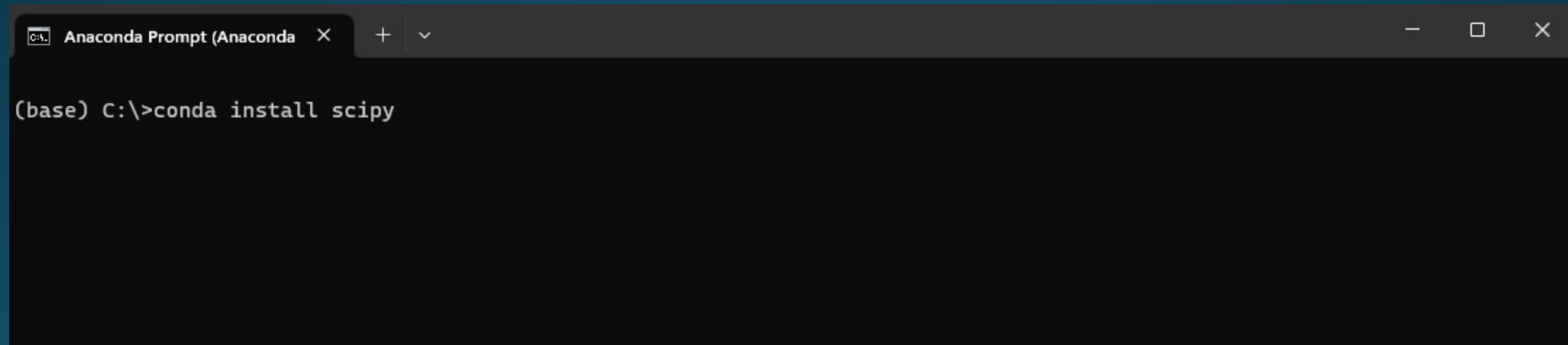
A screenshot of the Anaconda Prompt window. The title bar says "Anaconda Prompt (Anaconda)". The command line shows "(base) C:\>conda install numpy".

1. Instalação das Bibliotecas Necessárias

d. Instalação do scipy:

Digitar no terminal do Anaconda o comando:

conda install scipy



A screenshot of an Anaconda Prompt window titled "Anaconda Prompt (Anaconda)". The window has a dark theme. In the terminal area, the text "(base) C:\>conda install scipy" is visible, indicating the command being entered.

2. Implementação do Modbus no Python

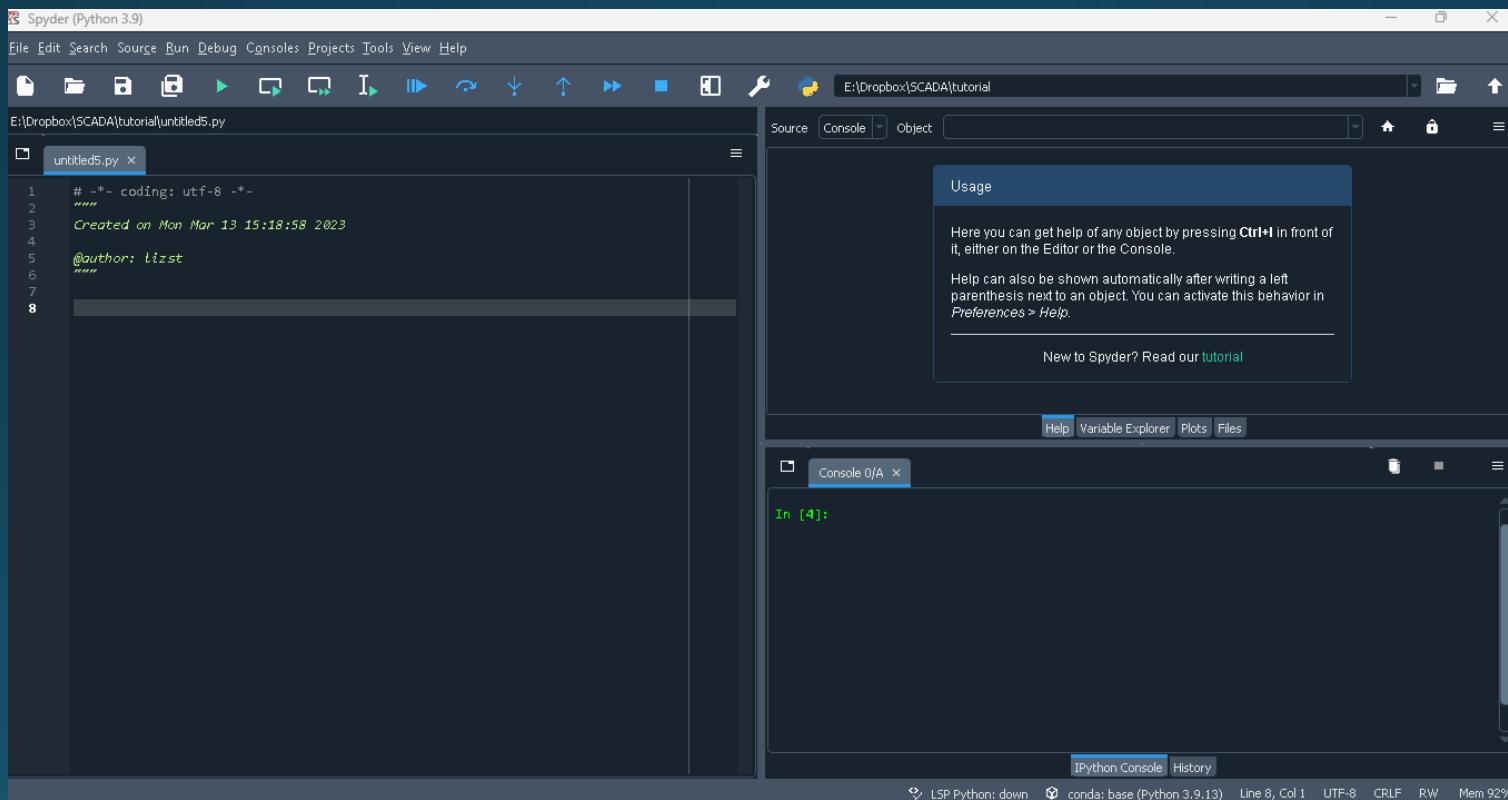
No Python, temos duas rotinas que serão necessárias para testar a comunicação Modbus:

- a. `servidor.py`
- b. `teste-scada.py`

As etapas da implementação serão mostradas a seguir.

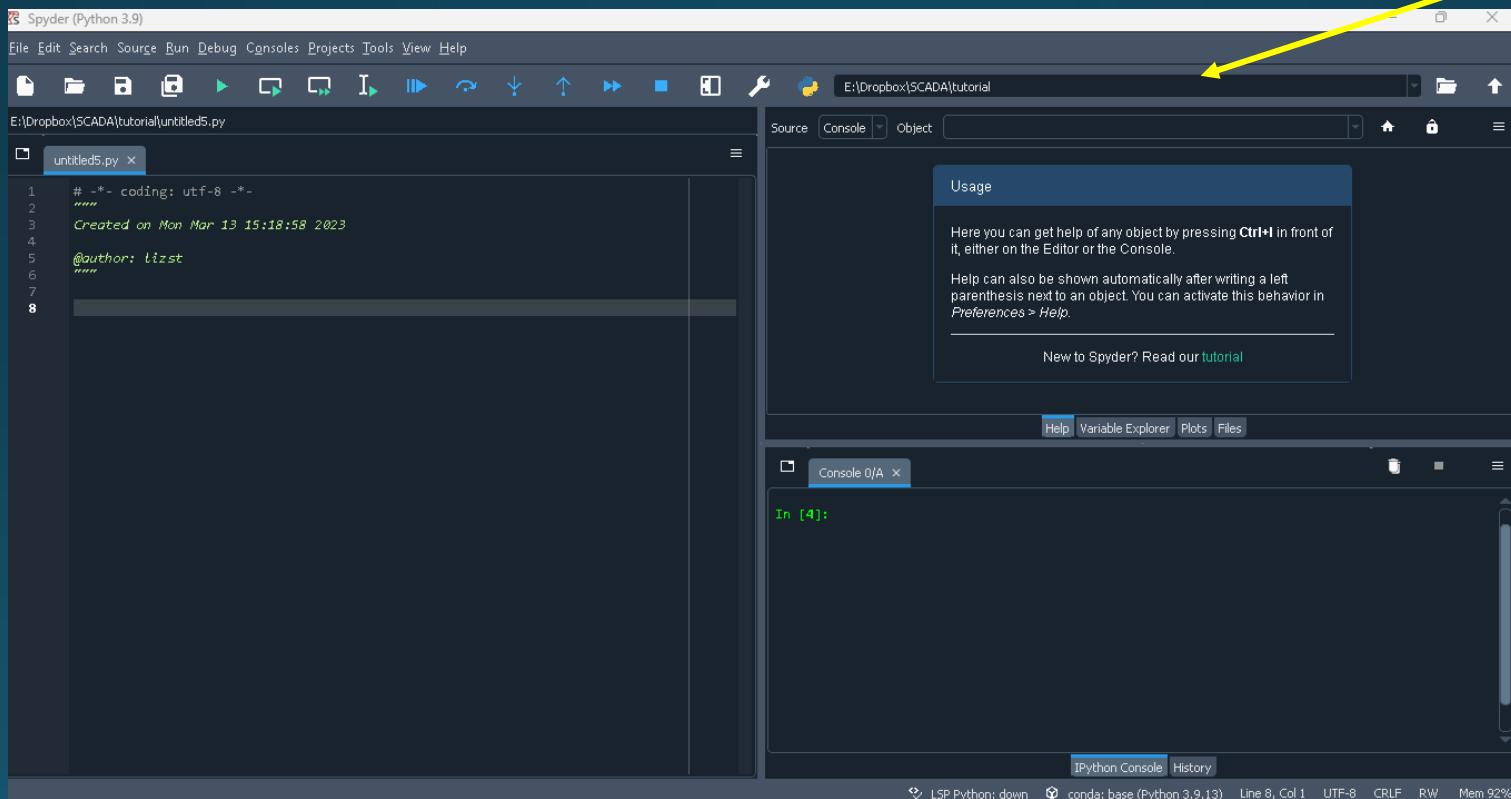
2. Implementação do Modbus no Python

(1) Abrir o Spyder:



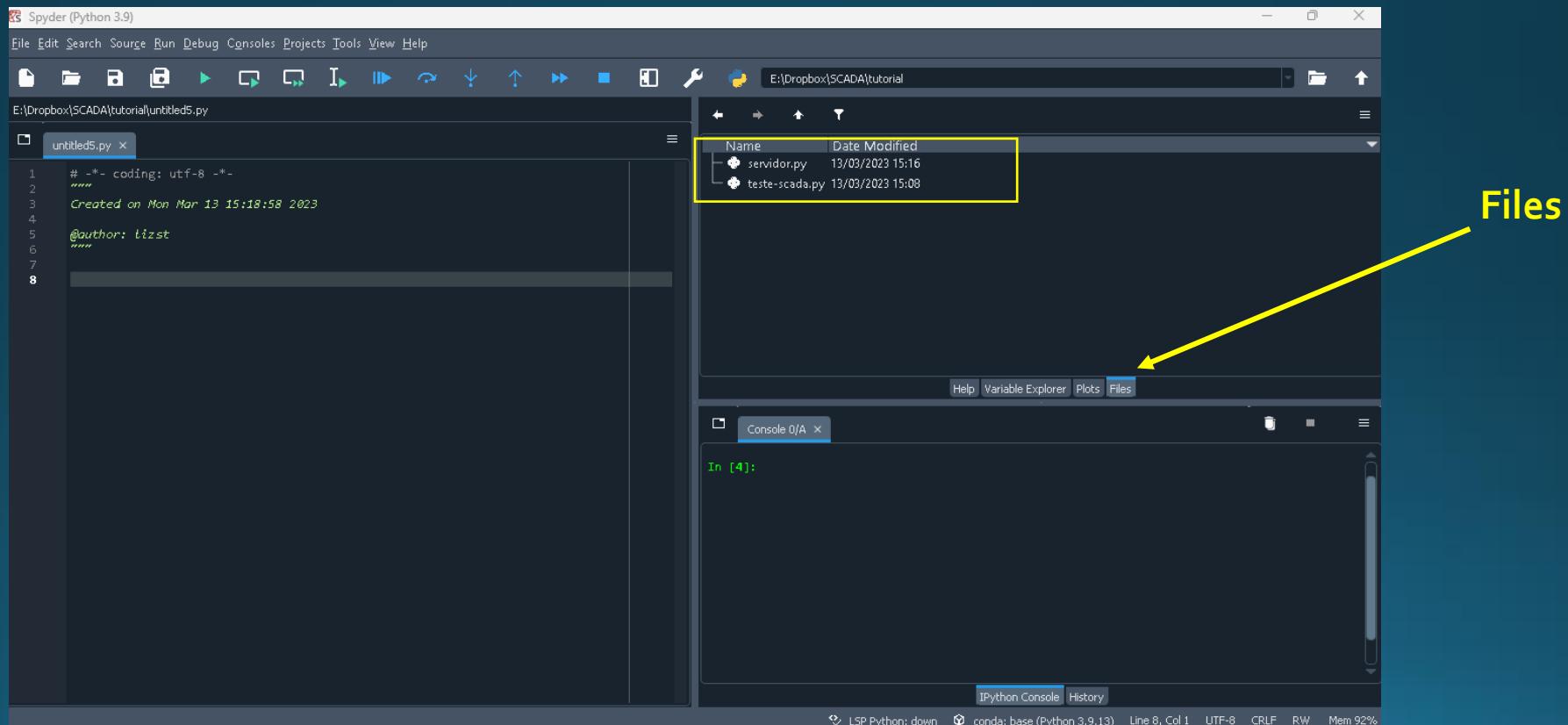
2. Implementação do Modbus no Python

(2) Apostar para a pasta de trabalho: ~SCADA/tutorial/: **Pasta de Trabalho**



2. Implementação do Modbus no Python

(3) Verificar os arquivos dentro da pasta, na Aba “files”:



Files

2. Implementação do Modbus no Python

(4) Abrir os arquivos da Aba “files”:

Arquivos abertos

```
# -*- coding: utf-8 -*-
"""
Servidor
"""

"""
Created on Mon Mar 29 17:52:54 2021
@author: lizst
"""

"""
Pymodbus Synchronous Server Example
-----m pip install -U name_pacote-----
The synchronous server is implemented in pure python without any third
party libraries (unless you need to use the serial protocols which require
pyserial). This is helpful in constrained or old environments where using
twisted is just not feasible. What follows is an example of its use:
# -
# import the various server implementations
# -
from pymodbus.version import version
from pymodbus.server.sync import StartTcpServer
from pymodbus.server.sync import StartTlsServer
from pymodbus.server.sync import StartUdpServer
from pymodbus.server.sync import StartSerialServer
from pymodbus.device import ModbusDeviceIdentification
from pymodbus.datastore import ModbusSequentialDataBlock, ModbusSparseDataBlock
from pymodbus.datastore import ModbusSlaveContext, ModbusServerContext
from pymodbus.transaction import ModbusRtuFramer, ModbusBinaryFramer
# -
```

E:\Dropbox\SCADA\tutorial

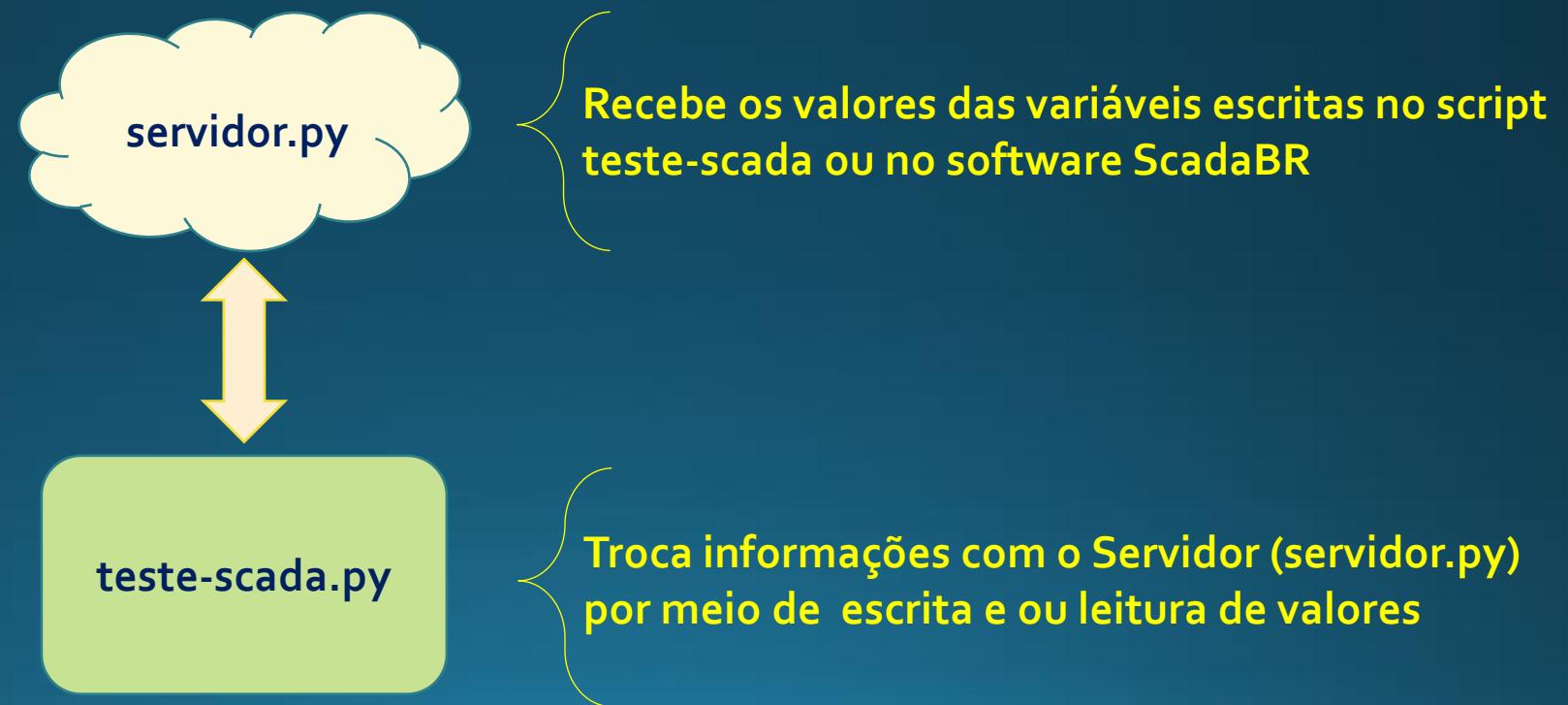
Name	Date Modified
servidor.py	13/03/2023 15:16
teste-scada.py	13/03/2023 15:08

In [4]:

LSP Python: down conda: base (Python 3.9.13) Line 1, Col 1 UTF-8 CRLF RW Mem 91%

2. Implementação do Modbus no Python

(5) Estrutura dos arquivos:



2. Implementação do Modbus no Python

(6) Estrutura dos scripts: (servidor.py)

Bibliotecas

```
"""
Servidor
"""

"""
Created on Mon Mar 29 17:52:54 2021

@author: tizst
"""

# -----
# import the various server implementations
# -----
from pymodbus.version import version
from pymodbus.server.sync import StartTcpServer
from pymodbus.server.sync import StartTlsServer
from pymodbus.server.sync import StartUdpServer
from pymodbus.server.sync import StartSerialServer

from pymodbus.device import ModbusDeviceIdentification
from pymodbus.datastore import ModbusSequentialDataBlock, ModbusSparseDataBlock
from pymodbus.datastore import ModbusSlaveContext, ModbusServerContext

from pymodbus.transaction import ModbusRtuFramer, ModbusBinaryFramer
# -----
# configure the service logging
# -----
import logging
FORMAT = ('%(asctime)-15s %(threadName)-15s'
          '%(levelname)-8s %(module)-15s:%(lineno)-8s %(message)s')
logging.basicConfig(format=FORMAT)
log = logging.getLogger()
log.setLevel(logging.DEBUG)
```

Execução

```
def run_server():

    store = ModbusSlaveContext(
        di=ModbusSequentialDataBlock(0, [17]*100),
        co=ModbusSequentialDataBlock(0, [17]*100),
        hr=ModbusSequentialDataBlock(0, [17]*100),
        ir=ModbusSequentialDataBlock(0, [17]*100))

    context = ModbusServerContext(slaves=store, single=True)

    identity = ModbusDeviceIdentification()
    identity.VendorName = 'Pymodbus'
    identity.ProductCode = 'PM'
    identity.VendorUrl = 'http://github.com/riptideio/pymodbus/'
    identity.ProductName = 'Pymodbus Server'
    identity.ModelName = 'Pymodbus Server'
    identity.MajorMinorRevision = version.short()

    # -----
    # run the server you want
    # -----
    # Tcp:
    StartTcpServer(context, identity=identity, address=( "", 5028))
    #

if __name__ == "__main__":
    run_server()
```

2. Implementação do Modbus no Python

(6) Estrutura dos scripts: (servidor.py)

Bibliotecas

```
"""
Servidor
"""

"""
Created on Mon Mar 29 17:52:54 2021
@author: tizst
"""

# -----
# import the various server implementations
# -----
from pymodbus.version import version
from pymodbus.server.sync import StartTcpServer
from pymodbus.server.sync import StartTlsServer
from pymodbus.server.sync import StartUdpServer
from pymodbus.server.sync import StartSerialServer

from pymodbus.device import ModbusDeviceIdentification
from pymodbus.datastore import ModbusSequentialDataBlock, ModbusSparseDataBlock
from pymodbus.datastore import ModbusSlaveContext, ModbusServerContext

from pymodbus.transaction import ModbusRtuFramer, ModbusBinaryFramer
# -----
# configure the service logging
# -----
import logging
FORMAT = ('%(asctime)-15s %(threadName)-15s'
          '%(levelname)-8s %(module)-15s:%(lineno)-8s %(message)s')
logging.basicConfig(format=FORMAT)
log = logging.getLogger()
log.setLevel(logging.DEBUG)
```

Execução

```
def run_server():

    store = ModbusSlaveContext(
        di=ModbusSequentialDataBlock(0, [17]*100),
        co=ModbusSequentialDataBlock(0, [17]*100),
        hr=ModbusSequentialDataBlock(0, [17]*100),
        ir=ModbusSequentialDataBlock(0, [17]*100))

    context = ModbusServerContext(slaves=store, single=True)

    identity = ModbusDeviceIdentification()
    identity.VendorName = 'Pymodbus'
    identity.ProductCode = 'PM'
    identity.VendorUrl = 'http://github.com/riptideio/pymodbus/'
    identity.ProductName = 'Pymodbus Server'
    identity.ModelName = 'Pymodbus Server'
    identity.MajorMinorRevision = version.short()

    # -----
    # run the server you want
    # -----
    # Tcp:
    StartTcpServer(context, identity=identity, address=( "", 5028))
    #

if __name__ == "__main__":
    run_server()
```

Endereço da porta do servidor



2. Implementação do Modbus no Python

(7) Estrutura dos scripts: (teste-scada.py)

Bibliotecas

```
# -*- coding: utf-8 -*-
"""
Created on Fri Mar 26 22:06:50 2021

@author: tizst
"""

# Importação das Bibliotecas
from pyModbusTCP.client import ModbusClient
from pyModbusTCP import utils
import numpy as np
from scipy.optimize import fsolve
import time
```

Funções para escrita e leitura

```
# Classe inicial para escrita e leitura
class FloatModbusClient(ModbusClient):
    def read_float(self, address, number=1):
        reg_1 = self.read_holding_registers(address, number * 2)
        if reg_1:
            return [utils.decode_ieee(f) for f in utils.word_list_to_long(reg_1)]
        else:
            return None

    def write_float(self, address, floats_list):
        b32_1 = [utils.encode_ieee(f) for f in floats_list]
        b16_1 = utils.long_list_to_word(b32_1)
        return self.write_multiple_registers(address, b16_1)
```

Execução de teste

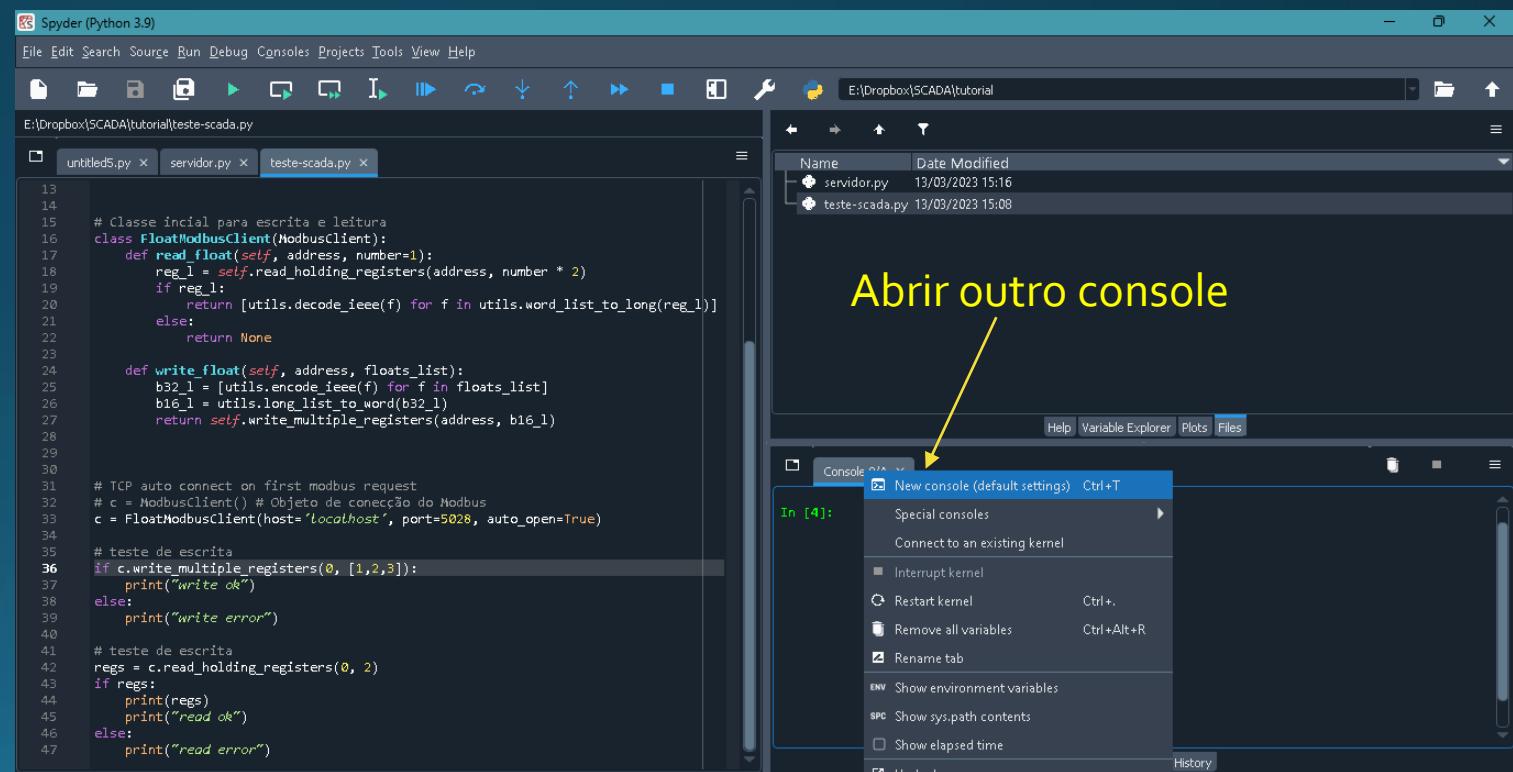
```
# TCP auto connect on first modbus request
# c = ModbusClient() # Objeto de conexão do Modbus
c = FloatModbusClient(host='localhost', port=5028, auto_open=True)

# teste de escrita
if c.write_multiple_registers(0, [1,2,3]):
    print("write ok")
else:
    print("write error")

# teste de escrita
regs = c.read_holding_registers(0, 2)
if regs:
    print(regs)
    print("read ok")
else:
    print("read error")
```

2. Implementação do Modbus no Python

(8) Execução dos scripts:
No Spyder, abrir dois consoles:



2. Implementação do Modbus no Python

(8) Execução dos scripts:
No Spyder, abrir dois consoles:

The screenshot shows the Spyder Python 3.9 IDE interface. On the left, there is a code editor with three tabs: 'untitled5.py', 'servidor.py', and 'teste-scada.py'. The 'teste-scada.py' tab is active, displaying Python code for a Modbus client. On the right, there is a file browser showing files in the directory 'E:\Dropbox\SCADA\tutorial'. Below the file browser, two yellow arrows point from the text 'Primeiro console' and 'Segundo console' to the tabs 'Console 0/A' and 'Console 1/A' respectively, which are located at the bottom of the interface.

```
# Classe inicial para escrita e leitura
class FloatModbusClient(ModbusClient):
    def read_float(self, address, number=1):
        reg_1 = self.read_holding_registers(address, number * 2)
        if reg_1:
            return [utils.decode_ieee(f) for f in utils.word_list_to_long(reg_1)]
        else:
            return None

    def write_float(self, address, floats_list):
        b32_l = [utils.encode_ieee(f) for f in floats_list]
        b16_l = utils.long_list_to_word(b32_l)
        return self.write_multiple_registers(address, b16_l)

# TCP auto connect on first modbus request
# c = ModbusClient() # Objeto de conexão do Modbus
c = FloatModbusClient(host='localhost', port=5028, auto_open=True)

# teste de escrita
if c.write_multiple_registers(0, [1,2,3]):
    print("write ok")
else:
    print("write error")

# teste de escrita
regs = c.read_holding_registers(0, 2)
if regs:
    print(regs)
    print("read ok")
else:
    print("read error")
```

2. Implementação do Modbus no Python

(9) Execução dos scripts:

Executar no primeiro console o script “servidor.py”:

The screenshot shows a Python IDE interface with three main components:

- Left Panel:** A code editor window titled "servidor.py" containing Python code for a Modbus server. The code includes imports, logging configuration, and a main function "run_server()". A yellow arrow points from the text "3 - Executar o servidor" to the run button in the toolbar.
- Middle Panel:** A file browser showing two files: "servidor.py" and "teste-scada.py". A yellow box labeled "2 - Clicar no servidor.py" surrounds the "servidor.py" file.
- Right Panel:** An IPython Console window titled "Console 0/A" with the status "In [4]:" and a yellow box labeled "1 - Clicar no Console" surrounding it.

The status bar at the bottom provides system information: LSP Python: down, conda: base (Python 3.9.13), Line 9, Col 15, UTF-8, CRLF, RW, Mem 88%.

```
File Edit Search Source Run Debug Consoles Projects Tools View Help  
E:\Dropbox\SCADA\tutorial  
servidor.py  
untitled5.py x servidor.py x teste-scada.py x  
32 logging.basicConfig(format=FORMAT)  
33 log = logging.getLogger()  
34 log.setLevel(logging.DEBUG)  
35  
36  
37 def run_server():  
38     store = ModbusSlaveContext(  
39         di=ModbusSequentialDataBlock(0, [17]*100),  
40         co=ModbusSequentialDataBlock(0, [17]*100),  
41         hr=ModbusSequentialDataBlock(0, [17]*100),  
42         ir=ModbusSequentialDataBlock(0, [17]*100))  
43  
44     context = ModbusServerContext(slaves=store, single=True)  
45  
46  
47     identity = ModbusDeviceIdentification()  
48     identity.VendorName = 'Pymodbus'  
49     identity.ProductCode = 'PM'  
50     identity.ProductUrl = 'https://github.com/riptideio/pymodbus/'  
51     identity.ProductName = 'Pymodbus Server'  
52     identity.ModelName = 'Pymodbus Server'  
53     identity.MajorMinorRevision = version.short()  
54  
55     # ---  
56     # run the server you want  
57     # ---  
58     # Tcp:  
59     StartTcpServer(context, identity=identity, address=("", 5028))  
60  
61     # ---  
62  
63 if __name__ == "__main__":  
64     run_server()  
65  
Name Date Modified  
servidor.py 13/03/2023 15:16  
teste-scada.py 13/03/2023 15:08  
Help Variable Explorer Plots Files  
Console 0/A x Console 1/A x  
In [4]:  
IPython Console History  
LSP Python: down conda: base (Python 3.9.13) Line 9, Col 15 UTF-8 CRLF RW Mem 88%
```

2. Implementação do Modbus no Python

(9) Execução dos scripts:

Executar no primeiro console o script “servidor.py”:

The screenshot shows a Python IDE interface with three main components:

- Left Panel:** A code editor window titled "servidor.py" containing Python code for a Modbus server. The code includes imports, logging configuration, and a main function "run_server()". A yellow arrow points from the text "3 - Executar o servidor" to the run button in the toolbar.
- Middle Panel:** A file browser showing two files: "servidor.py" and "teste-scada.py". A yellow box labeled "2 - Clicar no servidor.py" surrounds the "servidor.py" file.
- Right Panel:** An IPython Console window titled "Console 0/A" with the status bar showing "In [4]". A yellow box labeled "1 - Clicar no Console" surrounds the console tab.

The code in "servidor.py" is as follows:

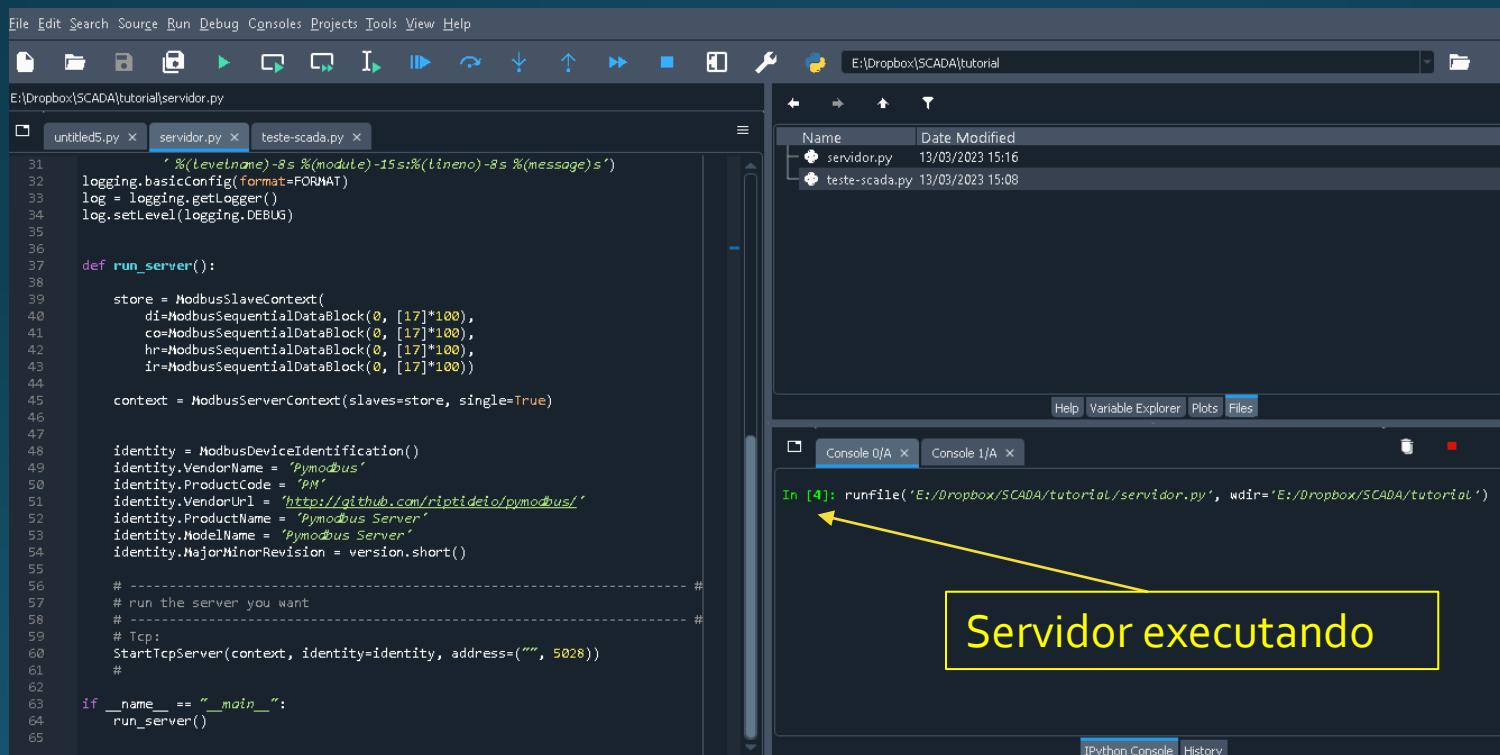
```
File Edit Search Source Run Debug Consoles Projects Tools View Help  
E:\Dropbox\SCADA\tutorial  
servidor.py  
untitled5.py x servidor.py x teste-scada.py x  
32 logging.basicConfig(format=FORMAT)  
33 log = logging.getLogger()  
34 log.setLevel(logging.DEBUG)  
35  
36  
37 def run_server():  
38     store = ModbusSlaveContext(  
39         di=ModbusSequentialDataBlock(0, [17]*100),  
40         co=ModbusSequentialDataBlock(0, [17]*100),  
41         hr=ModbusSequentialDataBlock(0, [17]*100),  
42         ir=ModbusSequentialDataBlock(0, [17]*100))  
43  
44     context = ModbusServerContext(slaves=store, single=True)  
45  
46  
47     identity = ModbusDeviceIdentification()  
48     identity.VendorName = 'Pymodbus'  
49     identity.ProductCode = 'PM'  
50     identity.ProductUrl = 'https://github.com/riptideio/pymodbus/'  
51     identity.ProductName = 'Pymodbus Server'  
52     identity.ModelName = 'Pymodbus Server'  
53     identity.MajorMinorRevision = version.short()  
54  
55     # ---  
56     # run the server you want  
57     # ---  
58     # Tcp:  
59     StartTcpServer(context, identity=identity, address=("", 5028))  
60  
61     # ---  
62  
63 if __name__ == "__main__":  
64     run_server()  
65
```

File E:\Dropbox\SCADA\tutorial
servidor.py
untitled5.py x servidor.py x teste-scada.py x
In [4]:
IPython Console History
LSP Python: down conda: base (Python 3.9.13) Line 9, Col 15 UTF-8 CRLF RW Mem 88%

2. Implementação do Modbus no Python

(9) Execução dos scripts:

Executar no primeiro console o script “servidor.py”:



The screenshot shows a Jupyter Notebook interface with the following components:

- File Bar:** File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, Help.
- Toolbar:** Includes icons for file operations like Open, Save, Run, Stop, and a Python logo.
- Code Editor:** Displays the content of `servidor.py`. The code defines a `run_server()` function that sets up a Modbus server context and runs it. It also includes configuration for device identity and a comment about running the server.
- File Explorer:** Shows files in the directory `E:\Dropbox\SCADA\tutorial`, specifically `servidor.py` and `teste-scada.py`.
- Console:** Shows the command `In [4]: runfile('E:/Dropbox/SCADA/tutorial/servidor.py', wdir='E:/Dropbox/SCADA/tutorial')` being executed.
- Text Overlay:** A yellow-bordered box contains the text "Servidor executando" (Server executing), with an arrow pointing from the text to the console output area.

2. Implementação do Modbus no Python

(9) Execução dos scripts:

Executar no segundo console o script “teste-scada.py”:

The screenshot shows a Python development environment with the following components:

- File Explorer:** Shows files in the directory `E:\Dropbox\SCADA\tutorial`. The files listed are `servidor.py` (modified 13/03/2023 15:16) and `teste-scada.py` (modified 13/03/2023 15:08).
- Code Editor:** Displays the content of `servidor.py`. A yellow arrow points from the text "3 - Executar o servidor" to the run button icon in the toolbar.
- Console 0/A:** Displays the Python 3.9.13 startup message and the IPython 7.31.1 welcome message.
- Console 1/A:** Displays the IPython 7.31.1 welcome message and the command `In [1]:`.

Annotations with yellow boxes and arrows:

- 3 - Executar o servidor**: Points to the run button icon in the code editor toolbar.
- 2 - Clicar no teste-scada.py**: Points to the `teste-scada.py` file in the file explorer.
- 1 - Clicar no Console**: Points to the IPython 7.31.1 console window.

```
File Edit Search Source Run Debug Consoles Projects Tools View Help
E:\Dropbox\SCADA\tutorial\servidor.py
untitled5.py x servidor.py x teste-scada.py x
31     '%(levelname)-8s %(module)-15s %(lineno)-8s %(message)s'
32     logging.basicConfig(format=FORMAT)
33     log = logging.getLogger()
34     log.setLevel(logging.DEBUG)
35
36     def run_server():
37
38         store = ModbusSlaveContext(
39             di=ModbusSequentialDataBlock(0, [17]*100),
40             co=ModbusSequentialDataBlock(0, [17]*100),
41             hr=ModbusSequentialDataBlock(0, [17]*100),
42             ir=ModbusSequentialDataBlock(0, [17]*100))
43
44         context = ModbusServerContext(slaves=store, single=True)
45
46
47         identity = ModbusDeviceIdentification()
48         identity.VendorName = 'Pymodbus'
49         identity.ProductCode = 'PM'
50         identity.VendorUrl = 'http://github.com/riptideio/pymodbus/'
51         identity.ProductName = 'Pymodbus Server'
52         identity.ModelName = 'Pymodbus Server'
53         identity.MajorMinorRevision = version.short()
54
55
56         # -----
57         # run the server you want
58         # -----
59         # Tcp:
60         StartTcpServer(context, identity=identity, address=("", 5028))
61
62
63         if __name__ == "__main__":
64             run_server()
65
E:\Dropbox\SCADA\tutorial
Name Date Modified
servidor.py 13/03/2023 15:16
teste-scada.py 13/03/2023 15:08
Help Variable Explorer Plots Files
Console 0/A x Console 1/A x
Python 3.9.13 (main, Aug 25 2023, 23:51:50) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.31.1 -- An enhanced Interactive Python.

In [1]:
# - - - - -
```

2. Implementação do Modbus no Python

(9) Execução dos scripts:

Executar no segundo console o script “teste-scada.py”:

The screenshot shows a development environment with two consoles. On the left, a code editor displays the script `teste-scada.py`. The code defines a class `FloatModbusClient` with methods for reading and writing floating-point values over Modbus. It also includes a test section where it writes to address 0 and reads from address 0. On the right, a file browser shows files in the directory `E:\Dropbox\SCADA\tutorial`, including `servidor.py` and `teste-scada.py`. Below the file browser is an "Execution Result" box containing the text "Resultado da Execução". The bottom right console window shows the output of running the script:

```
Python 3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.31.1 -- An enhanced Interactive Python
In [1]: runfile('E:/Dropbox/SCADA/tutorial/teste-scada.py', wdir='E:/Dropbox/SCADA/tutorial')
write ok
[1, 2]
read ok
```

2. Implementação do Modbus no Python

```
# TCP auto connect on first modbus request
# c = ModbusClient() # Objeto de conexão do Modbus
c = FloatModbusClient(host='localhost', port=5028, auto_open=True)

while 1<2:
    # teste de escrita
    if c.write_float(0, [10,20,30]):
        print("write ok")
    else:
        print("write error")

    # teste de leitura
    regs = c.read_holding_registers(0, 2)
    if regs:
        print(regs)
        print("read ok")
    else:
        print("read error")
```

Escrever no servidor os valor 10,20 e 30

Se a escrita funcionar: OK

Se a escrita não funcionar: "Erro"

Testar a leitura dos dois primeiros números escritos

Se a leitura funcionar: OK

Se a leitura não funcionar: "Erro"

3. Instalação do SCADABR

Para a instalação do SCADABR (versão 1.2), primeiramente fazer o download do programa no site:

<https://scadabr.org/>

Obs.: Verificar a necessidade de instalação do JAVA8 em sua máquina.

3. Instalação do SCADABR

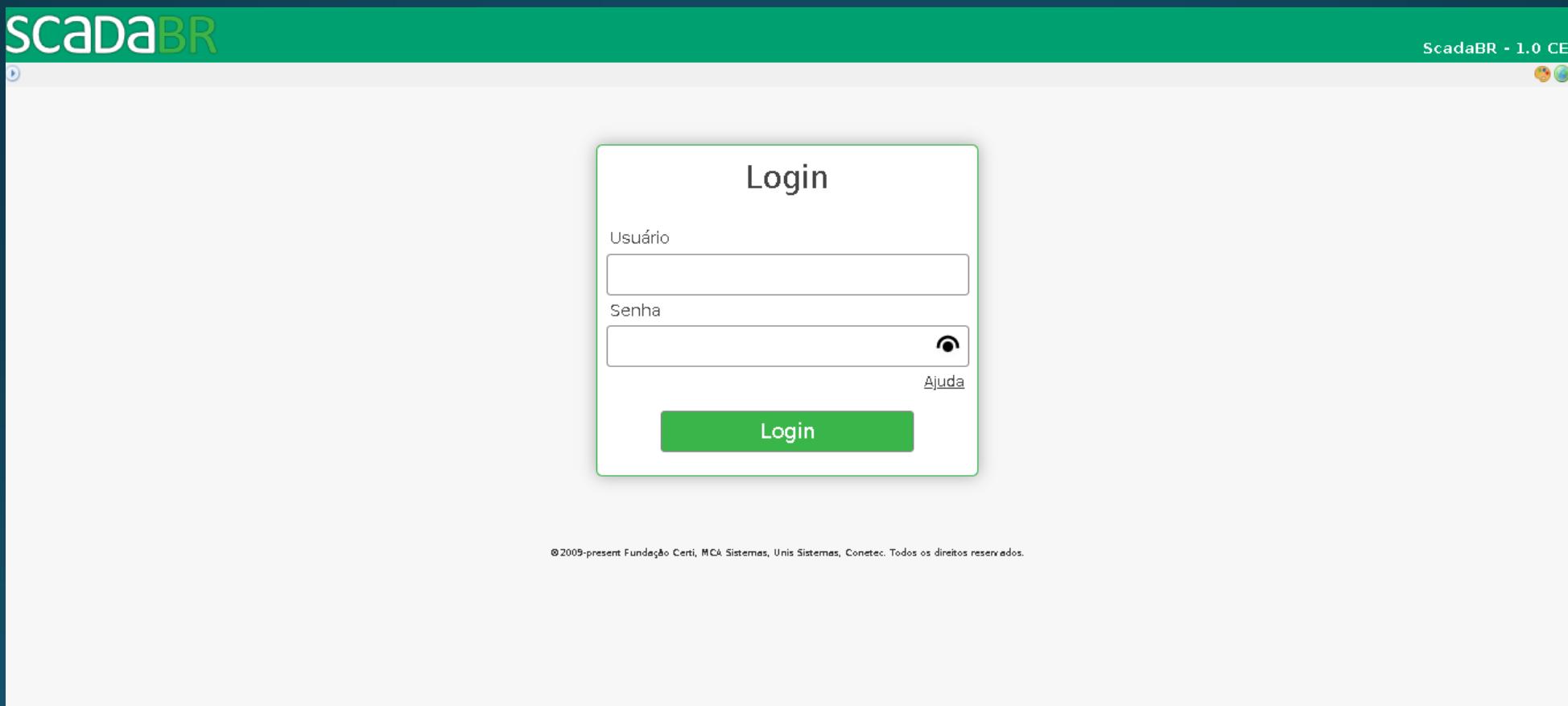
Após a instalação, o ícone do ScadaBR será inserido na área de trabalho do PC:



Para abrir o ScadaBR basta clicar no ícone e aguardar a abertura do ambiente de simulação no navegador de seu PC.

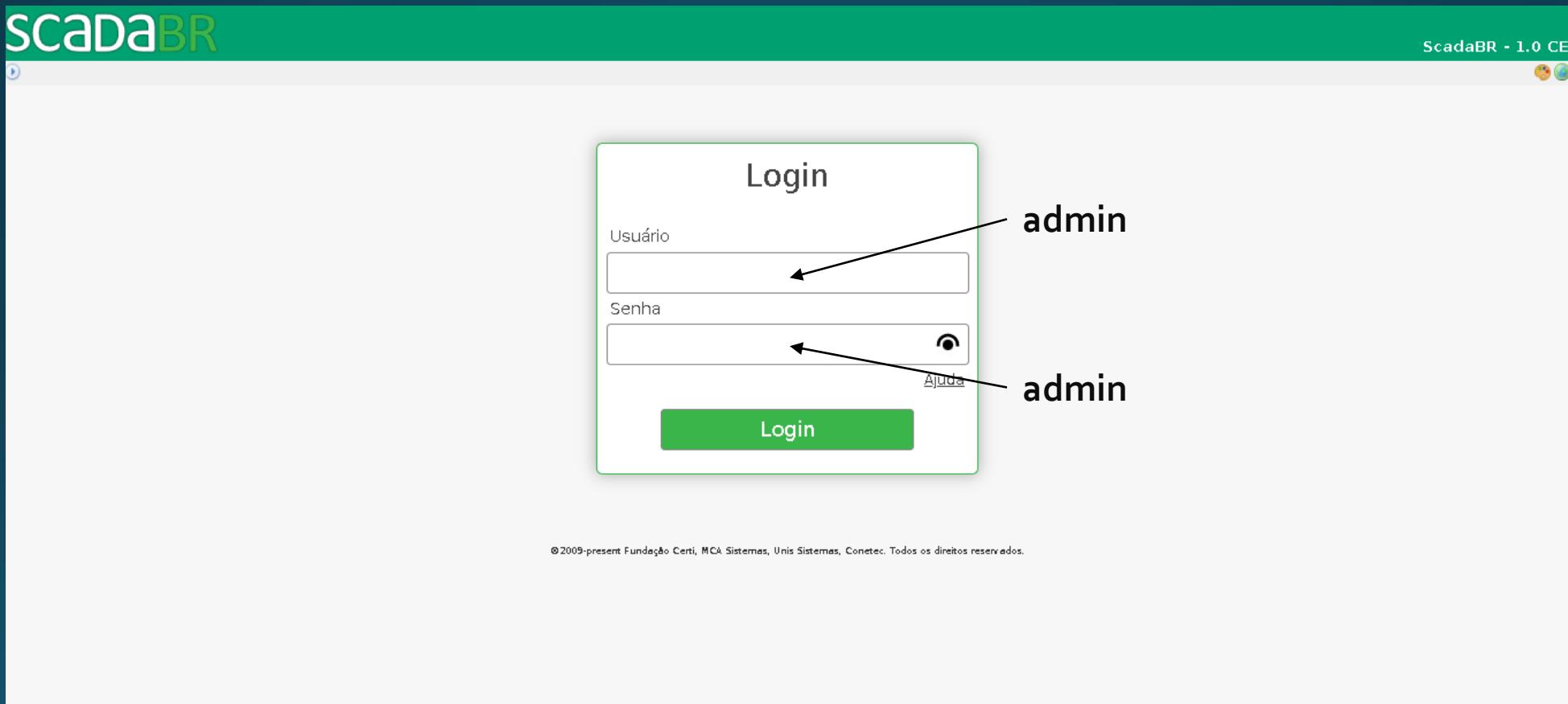
4. Implementação da Comunicação do Servidor com o ScadaBR

1. Abrir o ScadaBR



4. Implementação da Comunicação do Servidor com o ScadaBR

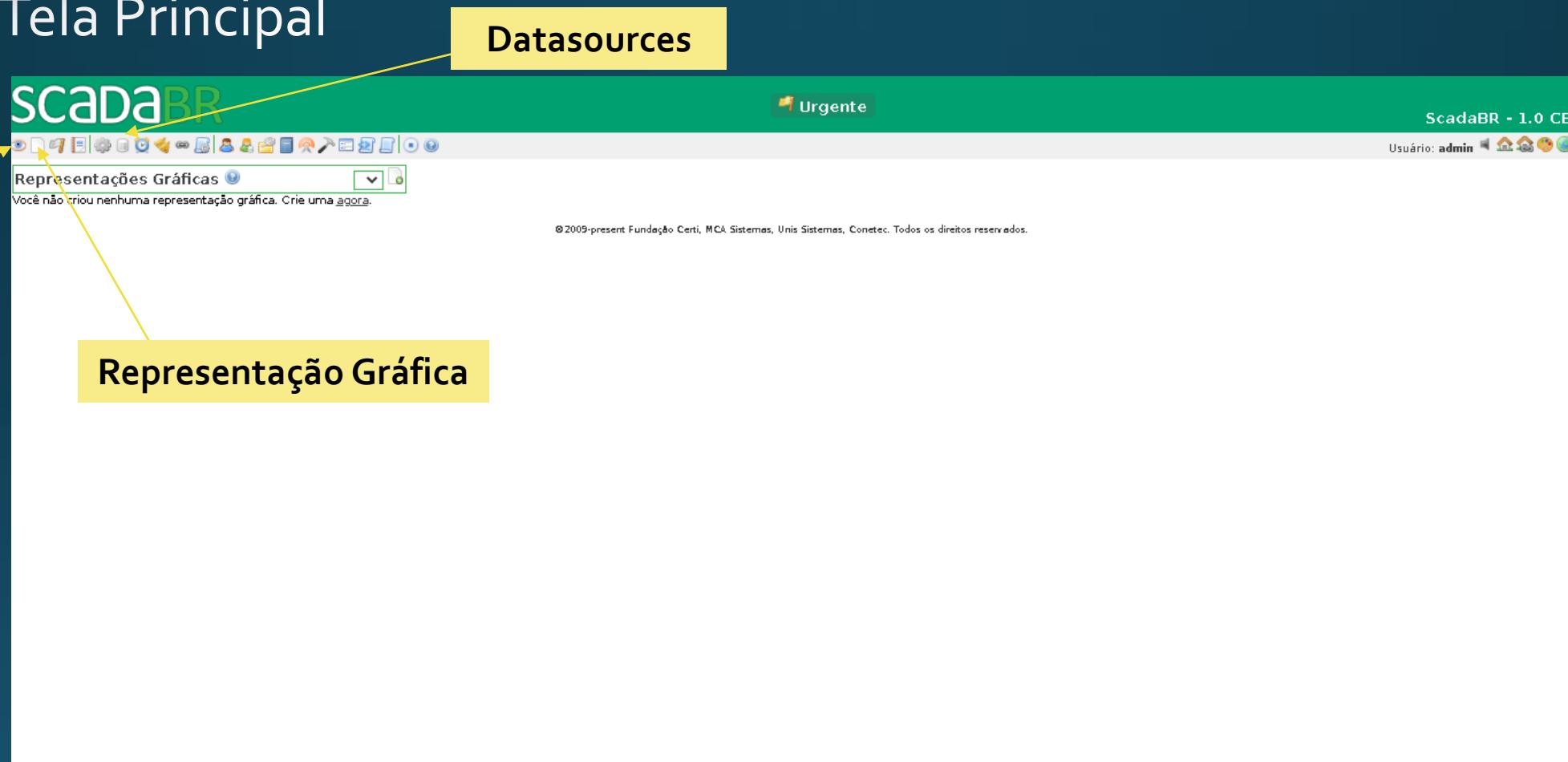
2. Entrar no sistema



©2009-present Fundação Certi, MCA Sistemas, Unis Sistemas, Conetec. Todos os direitos reservados.

4. Implementação da Comunicação do Servidor com o ScadaBR

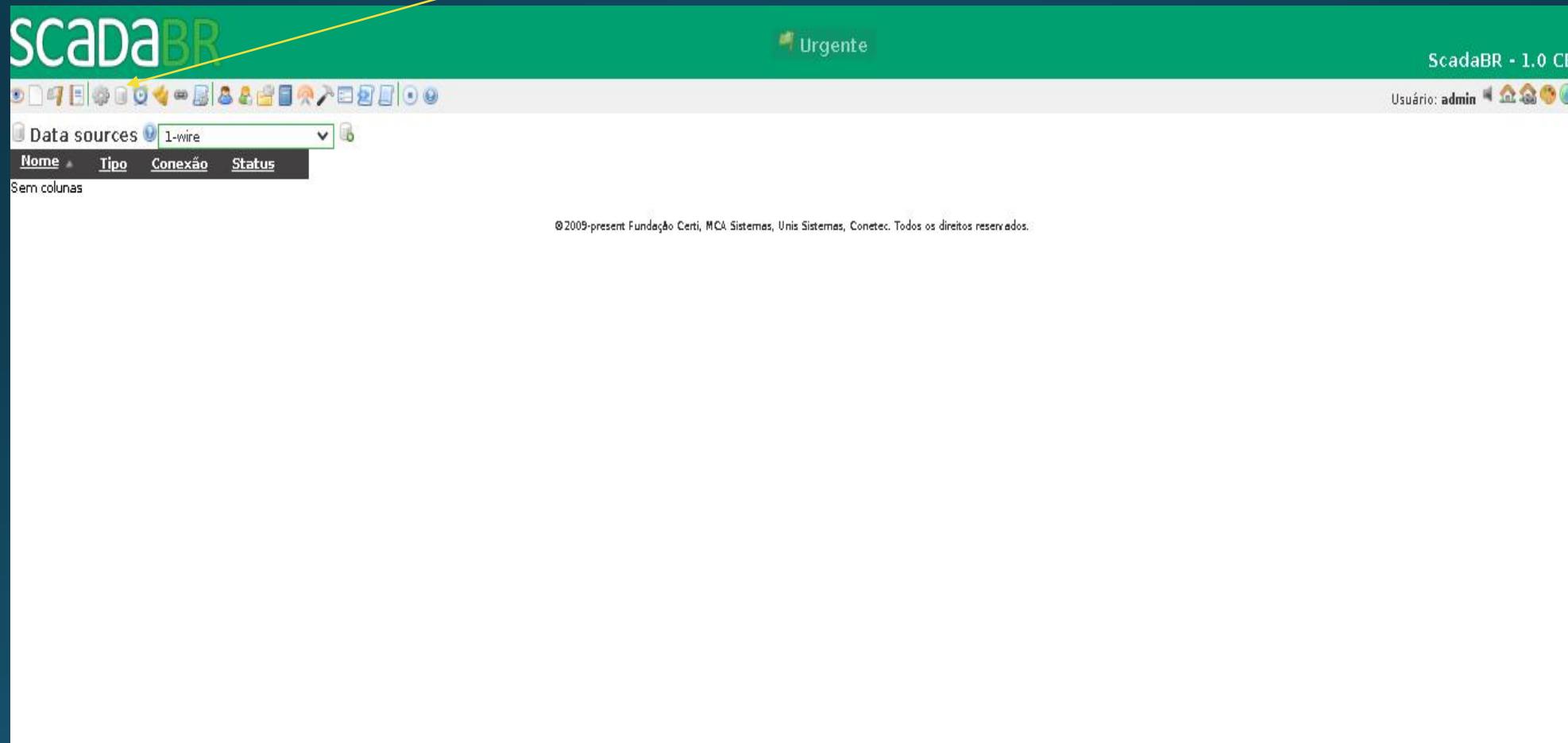
3. Tela Principal



4. Implementação da Comunicação do Servidor com o ScadaBR

4. Datasources

Selecionar Datasources



4. Implementação da Comunicação do Servidor com o ScadaBR

4. Datasources

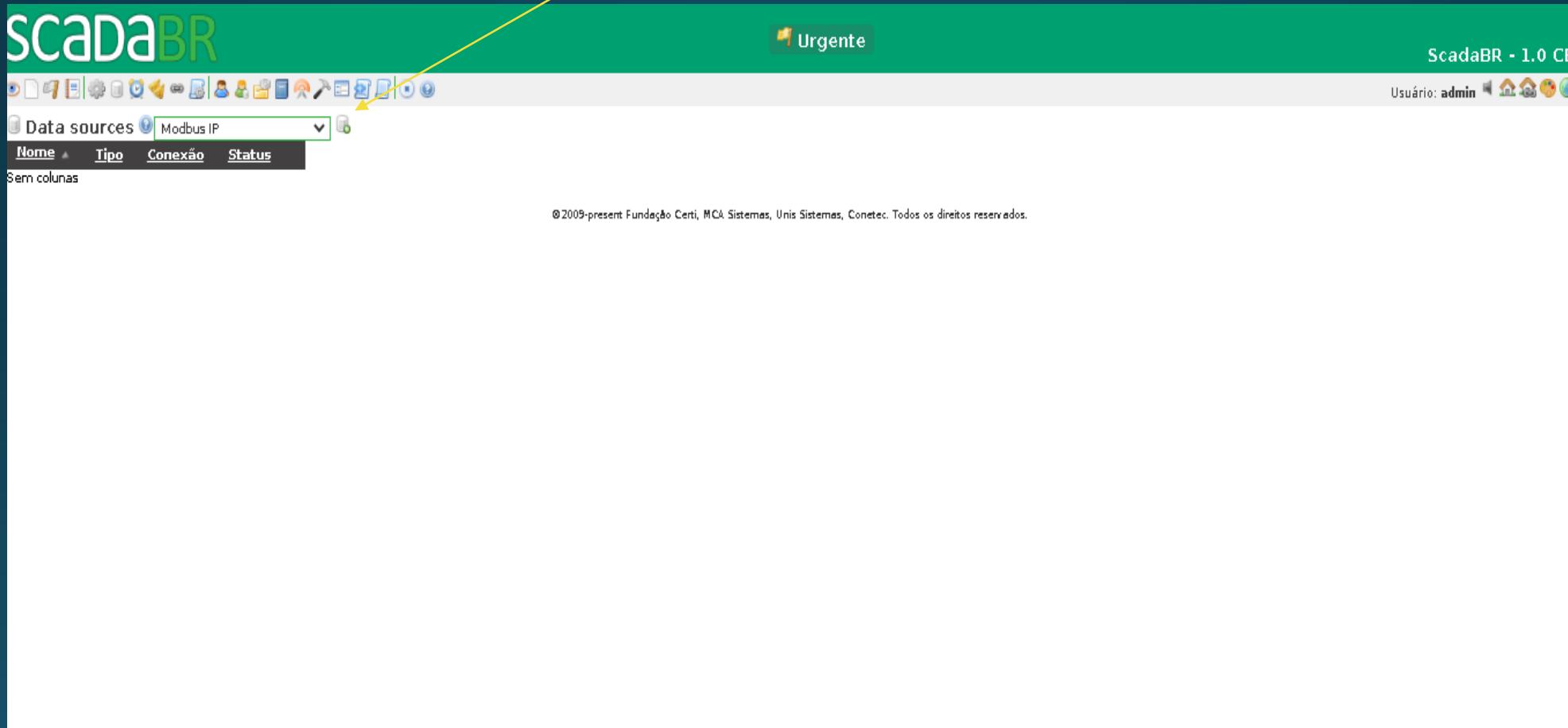
Selecionar ModbusIP

The screenshot shows the ScadaBR 1.0 CE software interface. At the top, there is a green header bar with the ScadaBR logo, an 'Urgente' button, and the text 'ScadaBR - 1.0 CE'. Below the header is a toolbar with various icons. The main area is titled 'Data sources' and has a dropdown menu set to 'Modbus IP'. A yellow arrow points from the text 'Selecionar ModbusIP' to this dropdown. Below the dropdown is a table with columns: Nome, Tipo, Conexão, and Status. The table is currently empty, showing the message 'Sem colunas'. At the bottom of the screen, there is a copyright notice: '©2009-present Fundação Certi, MCA Sistemas, Unis Sistemas, Conetec. Todos os direitos reservados.'

4. Implementação da Comunicação do Servidor com o ScadaBR

4. Datasources

Selecionar Adicionar



4. Implementação da Comunicação do Servidor com o ScadaBR

4. Datasources

No campo nome colocar “teste”

The screenshot shows the ScadaBR 1.0 CE software interface. On the left, there is a sidebar with various icons. The main area has two main sections:

- Propriedades do Modbus IP**:
 - Nome: (highlighted with a yellow arrow)
 - Export ID (XID): DS_894091
 - Período de atualização: 5 minuto(s)
 - Quantização:
 - Timeout {ms}: 500
 - Retentativas: 2
 - Apenas quantidades contíguas:
 - Criar pontos de monitor de escravo:
 - Máxima contagem de leitura de bits: 2000
 - Máxima contagem de leitura de registradores: 125
 - Máxima contagem de escrita de registradores: 120
 - Tipo de transporte: TCP
 - Host:
 - Porta: 502
 - Encapsulado:
 - Criar ponto monitor de conexão:
- Pesquisa de nós modbus**:
 - Pesquisar por nós:
 - Cancelar:
 - Nós encontrados:
- Leitura de dados Modbus**:
 - Id do escravo: 1
 - Faixa do registro: Bobina (Coil Status)
 - Offset (baseado em 0): 0
 - Número de registradores: 100
 - Ler dados:
- Teste de localizador de ponto**:
 - Id do escravo: 1
 - Faixa do registro: Bobina (Coil Status)
 - Tipo de dados modbus: Binário
 - Offset (baseado em 0): 0
 - Bit: 0
 - Número de registradores: 0
 - Codificação de caracteres: ASCII
 - Ler:
 - Adicionar ponto:

At the bottom, there is a footer with the text: © 2009-present Fundação Certi, MCA Sistemas, Unis Sistemas, Conetec. Todos os direitos reservados.

4. Implementação da Comunicação do Servidor com o ScadaBR

4. Datasources

No período de atualização colocar 5 segundos

The screenshot shows the ScadaBR 1.0 CE software interface. On the left, there is a sidebar with various icons. The main window has two main sections:

- Propriedades do Modbus IP**:
 - Nome: DS_894091
 - Export ID (XID): DS_894091
 - Período de atualização: 5 minuto(s) (highlighted with a yellow arrow)
 - Quantização:
 - Timeout {ms}: 500
 - Retentativas: 2
 - Apenas quantidades contíguas:
 - Criar pontos de monitor de escravo:
 - Máxima contagem de leitura de bits: 2000
 - Máxima contagem de leitura de registradores: 125
 - Máxima contagem de escrita de registradores: 120
 - Tipo de transporte: TCP
 - Host: (empty input field)
 - Porta: 502
 - Encapsulado:
 - Criar ponto monitor de conexão:
- Pesquisa de nós modbus**:
 - Pesquisar por nós
 - Cancelar
 - Nós encontrados: (list box)
- Leitura de dados Modbus**:
 - Id do escravo: 1
 - Faixa do registro: Bobina (Coil Status)
 - Offset (baseado em 0): 0
 - Número de registradores: 100
 - Ler dados
- Teste de localizador de ponto**:
 - Id do escravo: 1
 - Faixa do registro: Bobina (Coil Status)
 - Tipo de dados modbus: Binário
 - Offset (baseado em 0): 0
 - Bit: 0
 - Número de registradores: 0
 - Codificação de caracteres: ASCII
 - Ler
 - Adicionar ponto

At the bottom of the interface, there is a footer with the text: © 2009-present Fundação Certi, MCA Sistemas, Unis Sistemas, Conetec. Todos os direitos reservados.

4. Implementação da Comunicação do Servidor com o ScadaBR

4. Datasources

The screenshot shows the ScadaBR 1.0 CE software interface. On the left, there is a sidebar with various icons. The main window has a green header bar with the title "Urgente" and "ScadaBR - 1.0 CE". It displays two main panels:

- Propriedades do Modbus IP**: A configuration panel for a Modbus IP source. It includes fields for Name (teste), Export ID (DS_894091), Periodo de atualização (5 segundo(s)), Quantização (checkbox), Timeout (ms) (500), Retentativas (2), Apenas quantidades contíguas (checkbox), Criar pontos de monitor de escravo (checkbox), Máxima contagem de leitura de bits (2000), Máxima contagem de leitura de registradores (125), Máxima contagem de escrita de registradores (120), Tipo de transporte (TCP), Host (localhost), Porta (502), Encapsulado (checkbox), and Criar ponto monitor de conexão (checkbox). There are also sections for Níveis de alarme de eventos and Exceção de data source, Exceção de leitura de data point, and Exceção de escrita em data point, each with Urgente dropdowns.
- Pesquisa de nós modbus / Leitura de dados Modbus**: A panel for reading Modbus data. It shows a list of found nodes (Nós encontrados) and settings for a slave (Id do escravo: 1, Faixa do registro: Bobina (Coil Status), Offset (baseado em 0): 0, Número de registradores: 100). Below this is a "Teste de localizador de ponto" section with similar settings (Id do escravo: 1, Faixa do registro: Bobina (Coil Status), Tipo de dados modbus: Binário, Offset (baseado em 0): 0, Bit: 0, Número de registradores: 0, Codificação de caracteres: ASCII).

A yellow arrow points from the "Host" field in the Modbus IP properties panel to the "Host" field in the "Leitura de dados Modbus" panel, indicating that both fields represent the same physical host address.

Em host colocar “localhost”

© 2009-present Fundação Certi, MCA Sistemas, Unis Sistemas, Conetec. Todos os direitos reservados.

4. Implementação da Comunicação do Servidor com o ScadaBR

4. Datasources

The screenshot shows the ScadaBR 1.0 CE software interface. On the left, a configuration panel titled "Propriedades do Modbus IP" (Properties of Modbus IP) is displayed. It includes fields for "Nome" (Name: teste), "Export ID (XID)" (DS_894091), "Período de atualização" (Update period: 5 segundos), "Quantização" (Quantization), "Timeout (ms)" (Timeout: 500), "Retentativas" (Retries: 2), and "Porta" (Port: 5028). A yellow arrow points from the "Porta" field in this panel to the "Porta" field in the main configuration area. The main area contains two tabs: "Pesquisa de nós modbus" (Modbus node search) and "Leitura de dados Modbus" (Modbus data reading). The "Leitura de dados Modbus" tab is active, showing fields for "Id do escravo" (Slave ID: 1), "Faixa do registro" (Register range: Bobina (Coil Status)), "Offset (baseado em 0)" (Offset (based on 0): 0), "Número de registradores" (Number of registers: 100), and a "Ler dados" (Read data) button. Below these tabs is a "Teste de localizador de ponto" (Point locator test) section with similar fields for "Id do escravo", "Faixa do registro", "Tipo de dados modbus" (Modbus data type: Binário), "Offset (baseado em 0)", "Bit", "Número de registradores", and "Codificação de caracteres" (Character encoding: ASCII). The status bar at the bottom indicates "Urgente". The top right corner shows the user "admin" and the version "ScadaBR - 1.0 CE".

Em Porta colocar "5028"

© 2009-present Fundação Certi, MCA Sistemas, Unis Sistemas, Conetec. Todos os direitos reservados.

4. Implementação da Comunicação do Servidor com o ScadaBR

4. Datasources

Clicar no botão salvar

The screenshot shows the ScadaBR 1.0 CE software interface. On the left, there is a sidebar titled "Propriedades do Modbus IP" (Modbus IP Properties) with various configuration options like Name, Export ID, Period of update, Quantization, Timeout, Retries, and transport type (TCP). Below this are sections for alarm levels and exception handling. In the center, there are two main panels: "Pesquisa de nós modbus" (Modbus node search) and "Leitura de dados Modbus" (Modbus data reading). The "Leitura de dados Modbus" panel contains fields for Slave ID, Register range, Offset, Number of registers, and a "Ler dados" (Read data) button. Below these is a "Teste de localizador de ponto" (Point locator test) panel with similar fields for testing. A yellow arrow points from the text "Clicar no botão salvar" to the "Ler dados" button.

ScadaBR - 1.0 CE

Usuário: admin

Propriedades do Modbus IP

Nome: teste

Export ID (XID): DS_894091

Período de atualização: 5 segundos

Quantização:

Timeout (ms): 500

Retentativas: 2

Apenas quantidades contíguas:

Criar pontos de monitor de escravo:

Máxima contagem de leitura de bits: 2000

Máxima contagem de leitura de registradores: 125

Máxima contagem de escrita de registradores: 120

Tipo de transporte: TCP

Host: localhost

Porta: 5028

Encapsulado:

Criar ponto monitor de conexão:

Níveis de alarme de eventos

Exceção de data source: Urgente

Exceção de leitura de data point: Urgente

Exceção de escrita em data point: Urgente

Pesquisa de nós modbus

Pesquisar por nós:

Cancelar

Nós encontrados:

Leitura de dados Modbus

Id do escravo: 1

Faixa do registro: Bobina (Coil Status)

Offset (baseado em 0): 0

Número de registradores: 100

Ler dados

Teste de localizador de ponto

Id do escravo: 1

Faixa do registro: Bobina (Coil Status)

Tipo de dados modbus: Binário

Offset (baseado em 0): 0

Bit: 0

Número de registradores: 0

Codificação de caracteres: ASCII

Ler Adicionar ponto

© 2009-present Fundação Certi, MCA Sistemas, Unis Sistemas, Conetec. Todos os direitos reservados.

37

4. Implementação da Comunicação do Servidor com o ScadaBR

4. Datasources

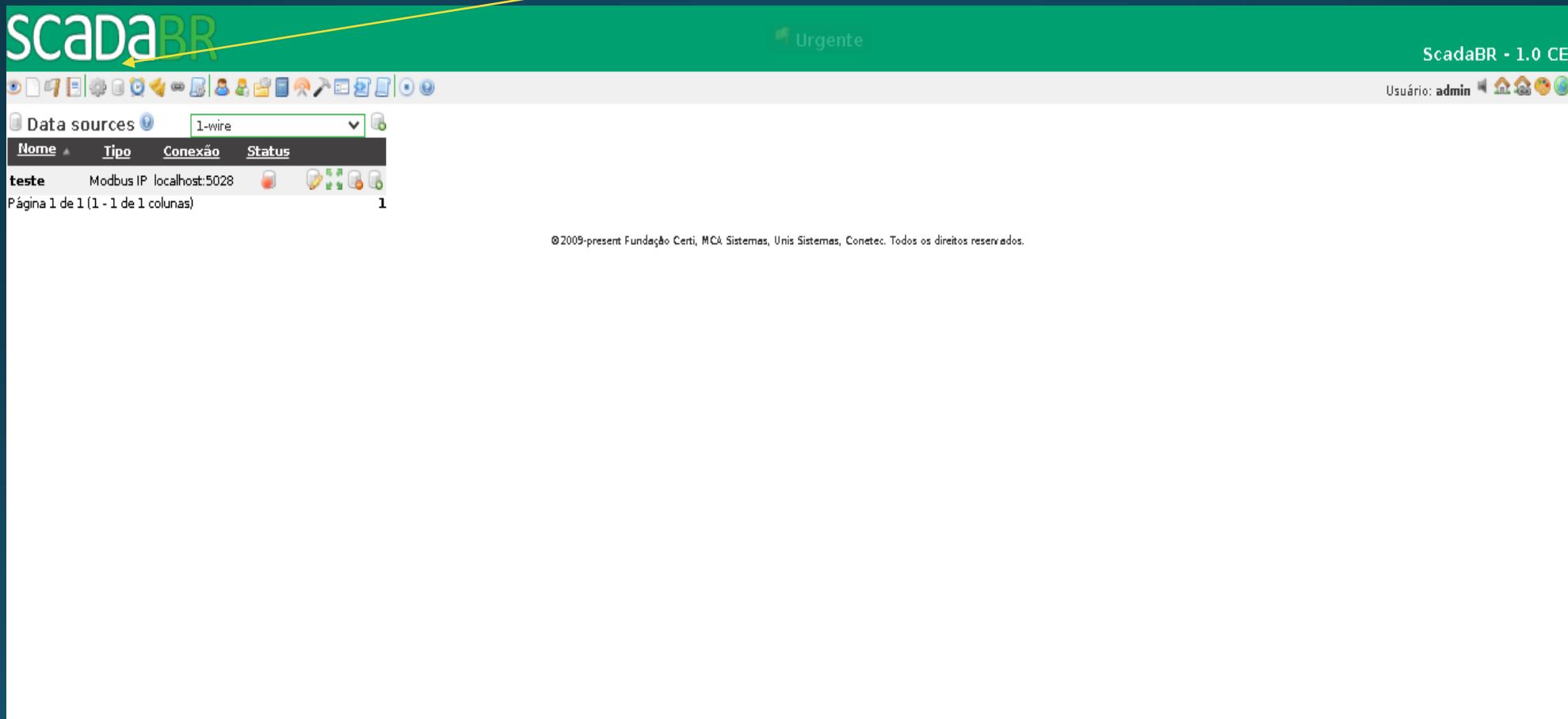
Clicar novamente em datasources

The screenshot shows the ScadaBR 1.0 CE software interface. On the left, there is a sidebar titled "Propriedades do Modbus IP" (Properties of Modbus IP) containing various configuration options for a Modbus connection. On the right, there are two main panels: "Pesquisa de nós modbus" (Modbus node search) and "Leitura de dados Modbus" (Modbus data reading). The "Leitura de dados Modbus" panel is active, showing fields for "Id do escravo" (Slave ID), "Faixa do registro" (Register range), "Offset (baseado em 0)" (Offset based on 0), and "Número de registradores" (Number of registers). Below these, there is a "Teste de localizador de ponto" (Point locator test) section with similar fields. A yellow arrow points from the "Datasources" section of the navigation bar at the top to the "Leitura de dados Modbus" panel. Another yellow arrow points from the "Leitura de dados Modbus" panel to the "Ler dados" (Read data) button.

4. Implementação da Comunicação do Servidor com o ScadaBR

4. Datasources

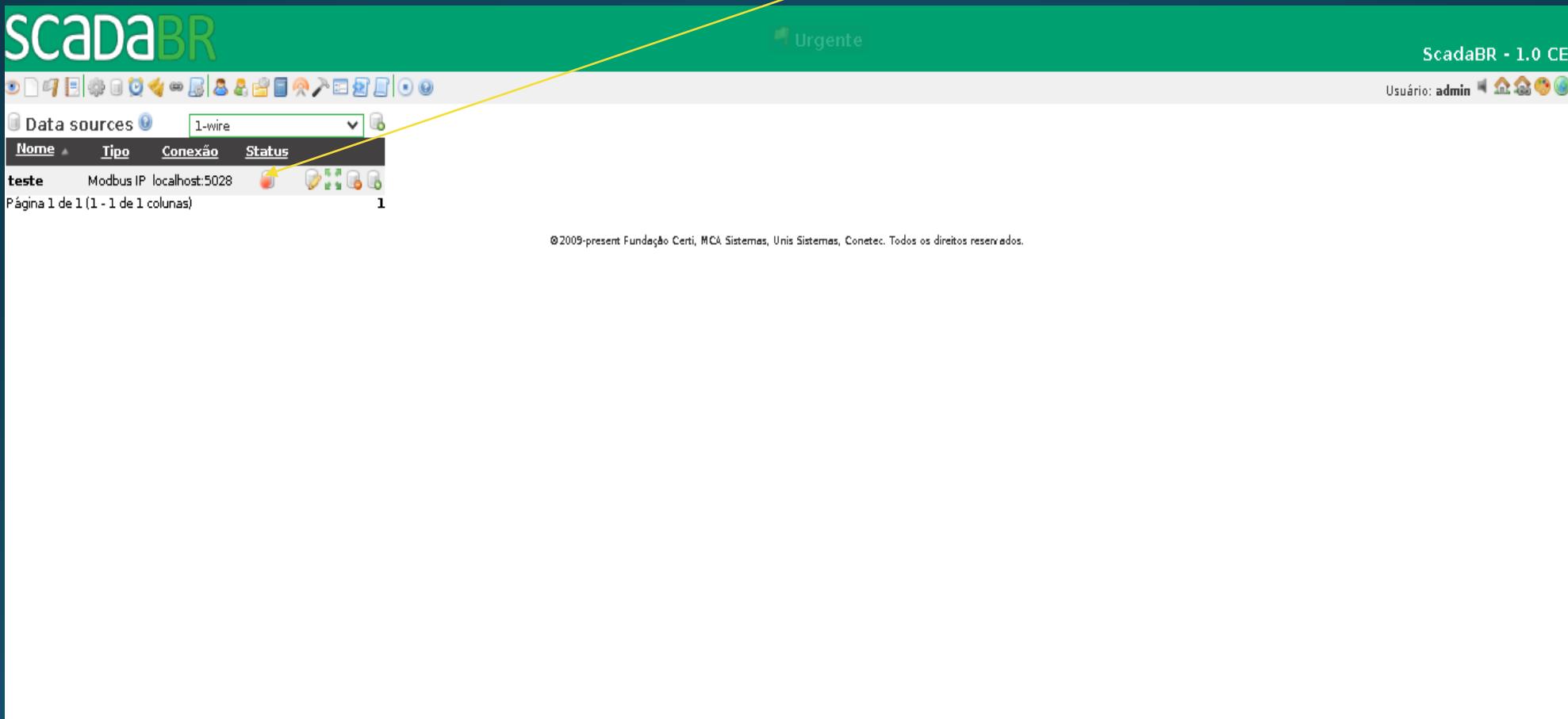
Clicar novamente em datasources



4. Implementação da Comunicação do Servidor com o ScadaBR

4. Datasources

Clicar em habilitar



4. Implementação da Comunicação do Servidor com o ScadaBR

4. Datasources

The screenshot shows the ScadaBR 1.0 CE software interface. The title bar reads "scadaBR" and "ScadaBR - 1.0 CE". The top menu bar includes icons for file operations, system settings, and monitoring. A status indicator "Urgente" is visible in the top right. The main content area is titled "Data sources" and displays a table with one row. The table columns are "Nome", "Tipo", "Conexão", and "Status". The single entry is "teste" (Type: Modbus IP, Connection: localhost:5028). A yellow box highlights the "Status" column for the "teste" entry. The bottom of the screen features a footer with copyright information: "©2009-present Fundação Certi, MC&Sistemas, Unis Sistemas, Conetec. Todos os direitos reservados."

Nome	Tipo	Conexão	Status
teste	Modbus IP	localhost:5028	

Página 1 de 1 (1 - 1 de 1 colunas) 1

©2009-present Fundação Certi, MC&Sistemas, Unis Sistemas, Conetec. Todos os direitos reservados.

4. Implementação da Comunicação do Servidor com o ScadaBR

4. Datasources

Clicar em editar

The screenshot shows the ScadaBR 1.0 CE application window. At the top, there's a toolbar with various icons. Below it is a header bar with the title 'scadaBR', a status message 'Urgente', and a user status 'Usuário: admin'. On the right side of the header, there are icons for home, system, and network. The main area is titled 'Data sources' and contains a table with one row. The table has columns for 'Nome', 'Tipo', 'Conexão', and 'Status'. The single entry is 'teste' (Type: Modbus IP, Connection: localhost:5028). To the right of the table, there are icons for creating a new source, deleting, and editing. A yellow arrow points from the text 'Clicar em editar' to the edit icon. At the bottom of the screen, there's a footer with the text '©2009-present Fundação Certi, MC&Sistemas, Unis Sistemas, Conetec. Todos os direitos reservados.' and a page number 'Página 1 de 1 (1 - 1 de 1 colunas)'.

Nome	Tipo	Conexão	Status
teste	Modbus IP	localhost:5028	

Página 1 de 1 (1 - 1 de 1 colunas)

©2009-present Fundação Certi, MC&Sistemas, Unis Sistemas, Conetec. Todos os direitos reservados.

4. Implementação da Comunicação do Servidor com o ScadaBR

5. Datapoints

The screenshot shows the ScadaBR - 1.0 CE software interface. On the left, there is a sidebar with various icons and a section titled "Propriedades do Modbus IP" containing fields for Name (teste), Export ID (DS_108973), Refresh period (5 segundos), Quantization, Timeout (500 ms), Retries (2), Contiguous quantities, Slave monitor creation, Maximum bit reading count (2000), Maximum register reading count (125), Maximum register writing count (120), Transport type (TCP), Host (localhost), Port (5028), Encapsulation, and Connection monitor creation. Below this is a section for "Níveis de alarme de eventos" with dropdown menus for Urgente, Risco, and Alerta for data source, reading, and writing exceptions.

In the center, there are two main configuration panels:

- Pesquisa de nós modbus**: A search interface for Modbus nodes, showing a list of found nodes and search filters for node name and address range.
- Leitura de dados Modbus**: A panel for reading Modbus data, with fields for Slave ID (1), Address range (Bobina (Coil Status)), Offset (0), Number of registers (100), and a "Ler dados" (Read data) button.

Below these is a "Teste de localizador de ponto" (Point locator test) panel, which is currently empty.

At the bottom right, the text "Adicionar Datapoint" is displayed in blue, with a blue arrow pointing from the text towards the "Data points" table at the bottom left.

Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)

4. Implementação da Comunicação do Servidor com o ScadaBR

5. Datapoints

The screenshot displays the ScadaBR configuration interface with several panels:

- Left Panel:** Configuration options for data points, including:
 - Apenas quantidades contíguas:
 - Criar pontos de monitor de escravo:
 - Máxima contagem de leitura de bits: 2000
 - Máxima contagem de leitura de registradores: 125
 - Máxima contagem de escrita de registradores: 120
 - Tipo de transporte: TCP
 - Host: localhost
 - Porta: 5028
 - Encapsulado:
 - Criar ponto monitor de conexão:
- Top Right Panel:** Teste de localizador de ponto (Point Locator Test) settings:
 - Id do escravo: 1
 - Faixa do registro: Bobina (Coil Status)
 - Tipo de dados modbus: Binário
 - Offset (baseado em 0): 0
 - Bit: 0
 - Número de registradores: 0
 - Codificação de caracteres: ASCII
 - Ler: Adicionar ponto:
- Bottom Left Panel:** Data points table:

Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)
- Bottom Right Panel:** Detalhes do data point (Data Point Details) configuration:
 - Nome: var_1
 - Export ID (XID): DP_726598
 - Id do escravo: 1
 - Faixa do registro: Bobina (Coil Status)
 - Tipo de dados modbus: Binário
 - Offset (baseado em 0): 0
 - Bit: 0
 - Número de registradores: 0
 - Codificação de caracteres: ASCII
 - Configurável:
 - Multiplicador: 1
 - Aditivo: 0

Annotations on the right side of the bottom-right panel:

- Nome: var_1**
- Registrador holding**
- Float de 4 bytes**

4. Implementação da Comunicação do Servidor com o ScadaBR

5. Datapoints

The screenshot displays the ScadaBR configuration interface with several panels:

- Left Panel (General Configuration):** Includes settings for contiguous quantities, master monitoring points, bit reading counts (2000), register reading counts (125), register writing counts (120), transport type (TCP), host (localhost), port (5028), encapsulation, and event alarm levels (Urgent for all three categories).
- Middle Panel (Test):** Shows a "Point Locator Test" configuration with fields for Slave ID (1), Register Range (Coil Status), Modbus Data Type (Binary), Offset (0), Bit (0), Number of Registers (0), Character Encoding (ASCII), and buttons for "Read" and "Add Point".
- Bottom Left Panel (Data Points List):** A table titled "Data points" with columns: Nome, Tipo de dado, Status, Escravo, Faixa, and Offset (baseado em 0). It currently shows one row with values: DP_726598, Binário, 1, Bobina (Coil Status), 0, and 0.
- Bottom Right Panel (Details):** A detailed configuration dialog for the selected data point, showing fields: Nome (DP_726598), Export ID (XID) (DP_726598), Slave ID (1), Register Range (Coil Status), Modbus Data Type (Binary), Offset (0), Bit (0), Number of Registers (0), Character Encoding (ASCII), Configurable (checked), Multiplier (1), and Additive (0).
- Bottom Center Panel (Preview):** A preview window showing the configuration details for the selected data point.

A blue arrow points from the "Salvar" button in the middle panel to the "Salvar" button in the bottom right panel, indicating the save operation.

4. Implementação da Comunicação do Servidor com o ScadaBR

5. Datapoints

The screenshot shows the ScadaBR software interface with several panels:

- Propriedades do Modbus IP**: Configuration for Modbus TCP connection to localhost port 5028.
- Pesquisa de nós modbus**: Search for Modbus nodes.
- Leitura de dados Modbus**: Read data from slave ID 1, register range Bobina (Coil Status), offset 0, and 100 registers.
- Teste de localizador de ponto**: Test point locator for slave ID 1, binary data type, offset 0, bit 0, and ASCII encoding.
- Níveis de alarme de eventos**: Event alarm levels for data source, read, and write exceptions.
- Data points**: List of configured data points, including var_1 (Numérico, Escravo 1, Registro holding 0).

A blue arrow points from the "Habilitar" button to the "Status" column in the Data points table.

Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)
var_1	Numérico		1	Registrador holding	0

Habilitar

4. Implementação da Comunicação do Servidor com o ScadaBR

5. Datapoints

The screenshot shows the ScadaBR software interface with several panels:

- Propriedades do Modbus IP**: Configuration for a Modbus TCP connection named "teste".
 - Nome: teste
 - Export ID (XID): DS_108973
 - Período de atualização: 5 segundo(s)
 - Quantização:
 - Timeout (ms): 500
 - Retentativas: 2
 - Apenas quantidades contíguas:
 - Criar pontos de monitor de escravo:
 - Máxima contagem de leitura de bits: 2000
 - Máxima contagem de leitura de registradores: 125
 - Máxima contagem de escrita de registradores: 120
 - Tipo de transporte: TCP
 - Host: localhost
 - Porta: 5028
 - Encapsulado:
 - Criar ponto monitor de conexão:
- Pesquisa de nós modbus**: A search panel for Modbus slaves.
- Leitura de dados Modbus**: A panel for reading data from a slave.
 - Id do escravo: 1
 - Faixa do registro: Bobina (Coil Status)
 - Offset (baseado em 0): 0
 - Número de registradores: 100
 - Ler dados
- Teste de localizador de ponto**: A panel for testing point locators.
 - Id do escravo: 1
 - Faixa do registro: Bobina (Coil Status)
 - Tipo de dados modbus: Binário
 - Offset (baseado em 0): 0
 - Bit: 0
 - Número: Salvo neste PC
 - Codificação de caracteres: ASCII
 - Ler
 - Adicionar ponto
- Data points**: A table of existing data points.

Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)
var_1	Numérico		1	Registrador holding	0

A blue arrow points from the text "Adicionar" (Add) to the "Adicionar ponto" button in the "Teste de localizador de ponto" panel.

4. Implementação da Comunicação do Servidor com o ScadaBR

5. Datapoints

The screenshot displays the ScadaBR configuration interface with several panels:

- Left Panel (Data Points Creation):** Contains fields for: "Apenas quantidades contíguas" (checkbox), "Criar pontos de monitor de escravo" (checkbox), "Máxima contagem de leitura de bits" (2000), "Máxima contagem de leitura de registradores" (125), "Máxima contagem de escrita de registradores" (120), "Tipo de transporte" (TCP dropdown), "Host" (localhost), "Porta" (5028), "Encapsulado" (checkbox), "Criar ponto monitor de conexão" (checkbox), and "Níveis de alarme de eventos" section with dropdowns for "Exceção de data source" (Urgente), "Exceção de leitura de data point" (Urgente), and "Exceção de escrita em data point" (Urgente).
- Top Right Panel (Modbus Test):** Titled "Teste de localizador de ponto". It includes fields for: "Id do escravo" (1), "Faixa do registro" (Bobina (Coil Status) dropdown), "Tipo de dados modbus" (Binário dropdown), "Offset (baseado em 0)" (0), "Bit" (0), "Número de registradores" (0), "Codificação de caracteres" (ASCII), and buttons "Ler" and "Adicionar ponto".
- Middle Left Panel (Data Points List):** A table titled "Data points" with columns: Nome, Tipo de dado, Status, Escravo, Faixa, and Offset (baseado em 0). One row is shown: "var_1" (Numérico), Escravo 1, Registrador holding 0.
- Bottom Right Panel (Data Point Details):** Titled "Detalhes do data point". It shows the configuration for "var_1": "Nome" (DP_820353), "Export ID (XID)" (DP_820353), "Id do escravo" (1), "Faixa do registro" (Bobina (Coil Status) dropdown), "Tipo de dados modbus" (Binário dropdown), "Offset (baseado em 0)" (2), "Bit" (0), "Número de registradores" (0), "Codificação de caracteres" (ASCII), "Configurável" (checkbox checked), "Multiplicador" (1), and "Aditivo" (0). Arrows from the text annotations point to specific fields: "var_2" points to the "Nome" field; "Registrador holding" points to the "Faixa do registro" dropdown; "Float de 4 bytes" points to the "Offset (baseado em 0)" field; and "Offset=2" points to the "Offset (baseado em 0)" value.

4. Implementação da Comunicação do Servidor com o ScadaBR

5. Datapoints

The screenshot displays the ScadaBR configuration interface with several panels:

- Left Panel (Configuration):** Contains settings for monitoring and data points:
 - Apenas quantidades contíguas:**
 - Criar pontos de monitor de escravo:**
 - Máxima contagem de leitura de bits:** 2000
 - Máxima contagem de leitura de registradores:** 125
 - Máxima contagem de escrita de registradores:** 120
 - Tipo de transporte:** TCP
 - Host:** localhost
 - Porta:** 5028
 - Encapsulado:**
 - Criar ponto monitor de conexão:**

Níveis de alarme de eventos:

 - Exceção de data source:** Urgente
 - Exceção de leitura de data point:** Urgente
 - Exceção de escrita em data point:** Urgente
- Central Panel (Teste de localizador de ponto):** A configuration dialog for testing a point:

Id do escravo: 1	Faixa do registro: Bobina (Coil Status)
Tipo de dados modbus: Binário	
Offset (baseado em 0): 0	
Bit: 0	
Número de registradores: 0	
Codificação de caracteres: ASCII	

Ler **Adicionar ponto**
- Bottom Left Panel (Data points list):** Shows a table of existing data points:

Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)
var_1	Numérico		1	Registrador holding	0
- Bottom Right Panel (Salvar - Details):** A configuration dialog for saving a data point:

Nome: DP_820353
Export ID (XID): DP_820353
Id do escravo: 1
Faixa do registro: Bobina (Coil Status)
Tipo de dados modbus: Binário
Offset (baseado em 0): 2
Bit: 0
Número de registradores: 0
Codificação de caracteres: ASCII
Configurável: <input checked="" type="checkbox"/>
Multiplicador: 1
Aditivo: 0

4. Implementação da Comunicação do Servidor com o ScadaBR

5. Datapoints

The screenshot shows the configuration interface for a Modbus data source. On the left, the 'Propriedades do Modbus IP' section includes fields for Name (teste), Export ID (DS_108973), Update Period (5 seconds), Quantization, Timeout (500 ms), Retries (2), Contiguous quantities, Slave monitor creation, Maximum bit reading count (2000), Maximum register reading count (125), Maximum register writing count (120), Transport type (TCP), Host (localhost), Port (5028), Encapsulation, and Connection monitor creation. Below these are alarm levels for events, data sources, data points, and data point writes. On the right, the 'Leitura de dados Modbus' section shows settings for Slave ID (1), Register range (Bobina (Coil Status)), Offset (0), Number of registers (100), and a 'Ler dados' (Read data) button. A 'Pesquisa de nós modbus' (Modbus node search) panel is also visible. At the bottom, a 'Teste de localizador de ponto' (Point locator test) section is shown with similar parameters. A 'Data points' table at the very bottom lists two entries: 'var_1' (Numérico, Slave 1, Register holding 0) and 'var_2' (Numérico, Slave 1, Register holding 2). An arrow points from the text 'Habilitar datapoint' to the enablement icon for 'var_2'.

Habilitar datapoint

Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)
var_1	Numérico		1	Registrador holding 0	
var_2	Numérico		1	Registrador holding 2	

4. Implementação da Comunicação do Servidor com o ScadaBR

5. Datapoints

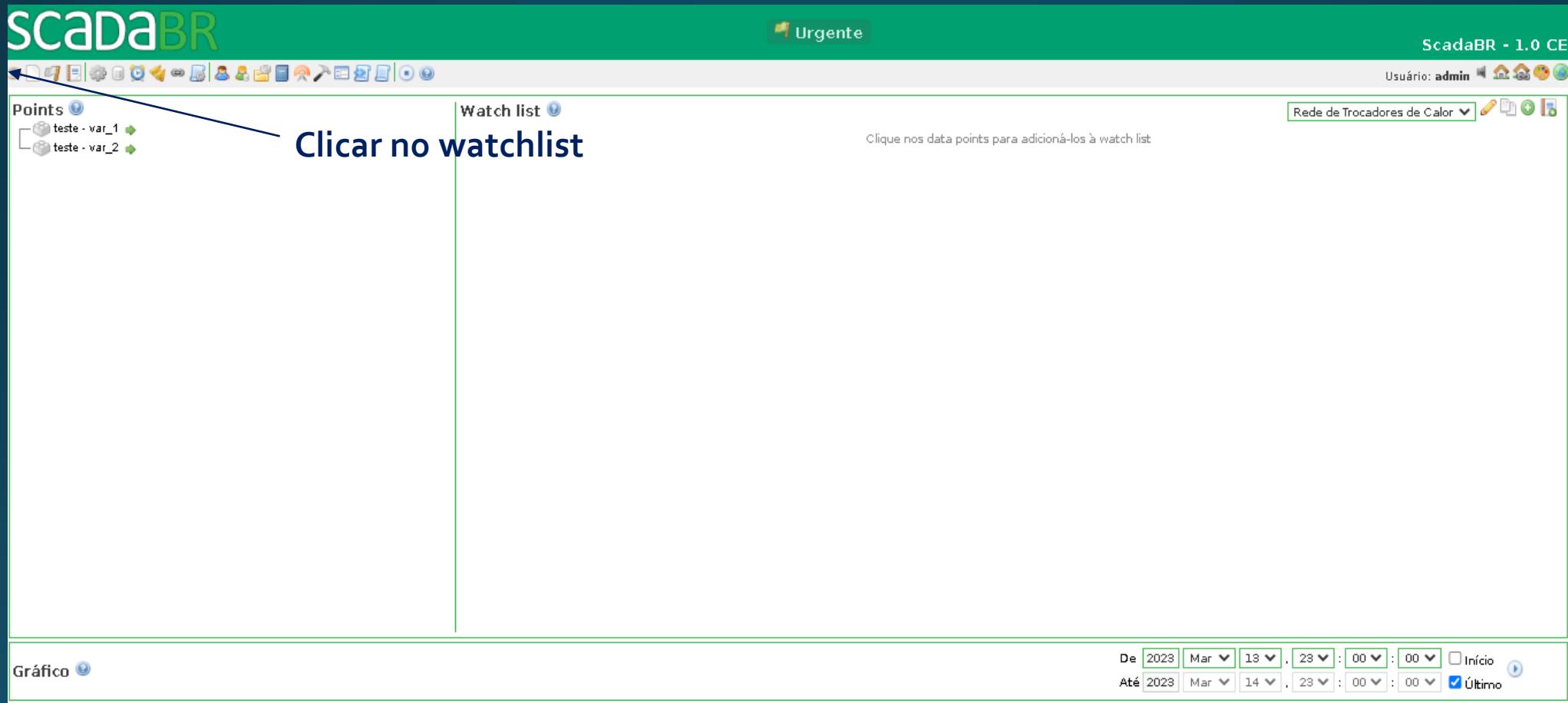
The screenshot displays the ScadaBR configuration interface with several panels:

- Propriedades do Modbus IP**: Configuration for a Modbus TCP connection named "teste".
 - Nome: teste
 - Export ID (XID): DS_108978
 - Período de atualização: 5 segundo(s)
 - Quantização: (checkbox)
 - Timeout (ms): 500
 - Retentativas: 2
 - Apenas quantidades contíguas: (checkbox)
 - Criar pontos de monitor de escravo: (checkbox)
 - Máxima contagem de leitura de bits: 2000
 - Máxima contagem de leitura de registradores: 125
 - Máxima contagem de escrita de registradores: 120
 - Tipo de transporte: TCP
 - Host: localhost
 - Porta: 5028
 - Encapsulado: (checkbox)
 - Criar ponto monitor de conexão: (checkbox)
- Pesquisa de nós modbus**: Search results for Modbus nodes.
 - Pesquisar por nós: (button)
 - Cancelar: (button)
 - Nós encontrados: (list box)
- Leitura de dados Modbus**: Data reading configuration for slave ID 1.
 - Id do escravo: 1
 - Faixa do registro: Bobina (Coil Status)
 - Offset (baseado em 0): 0
 - Número de registradores: 100
 - Ler dados: (button)
- Teste de localizador de ponto**: Point locator test configuration for slave ID 1.
 - Id do escravo: 1
 - Faixa do registro: Bobina (Coil Status)
 - Tipo de dados modbus: Binário
 - Offset (baseado em 0): 0
 - Bit: 0
 - Número de registradores: 0
 - Codificação de caracteres: ASCII
 - Ler: (button)
 - Adicionar ponto: (button)
- Data points**: Table of configured data points.

Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)
var_1	Numérico		1	Registrador holding	0
var_2	Numérico		1	Registrador holding	2

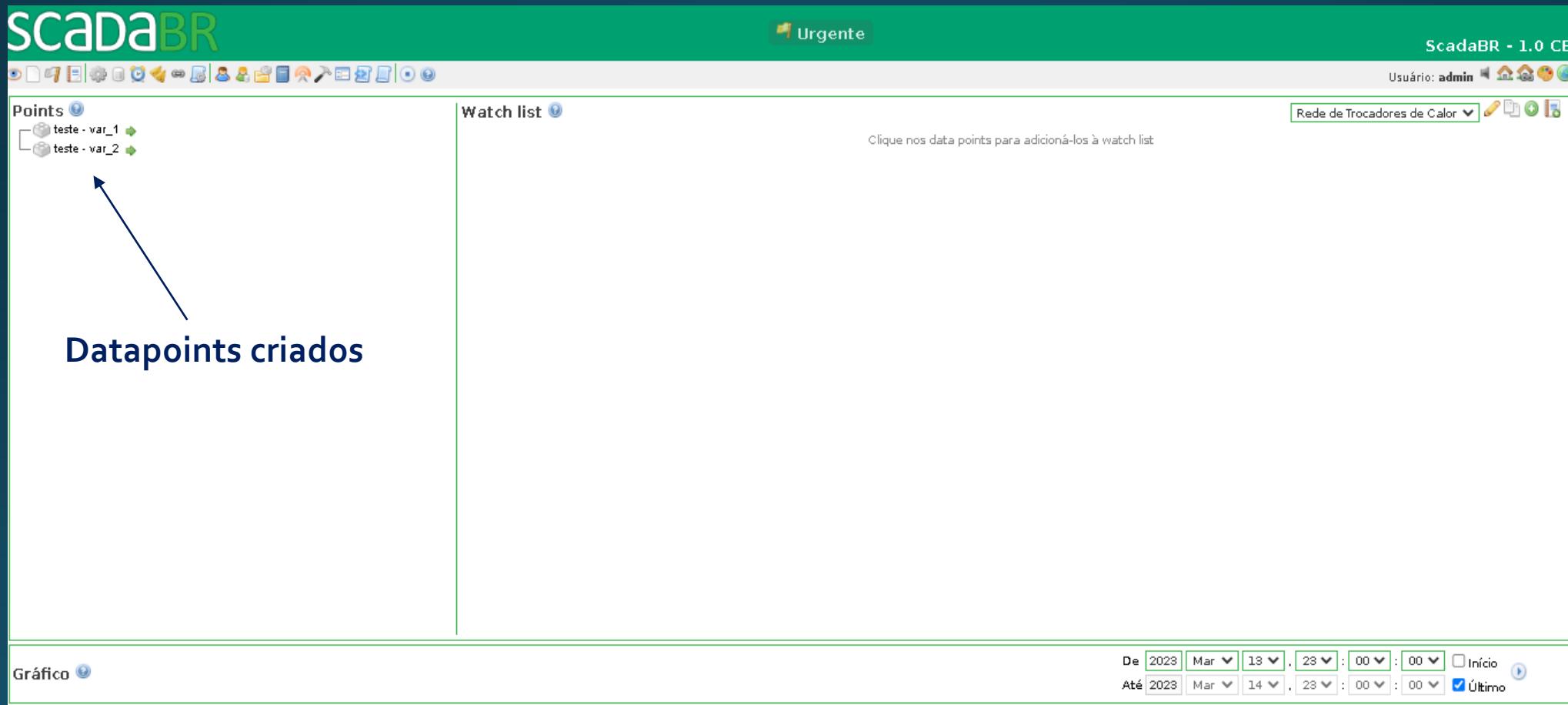
4. Implementação da Comunicação do Servidor com o ScadaBR

6. Visualização no Watchlist



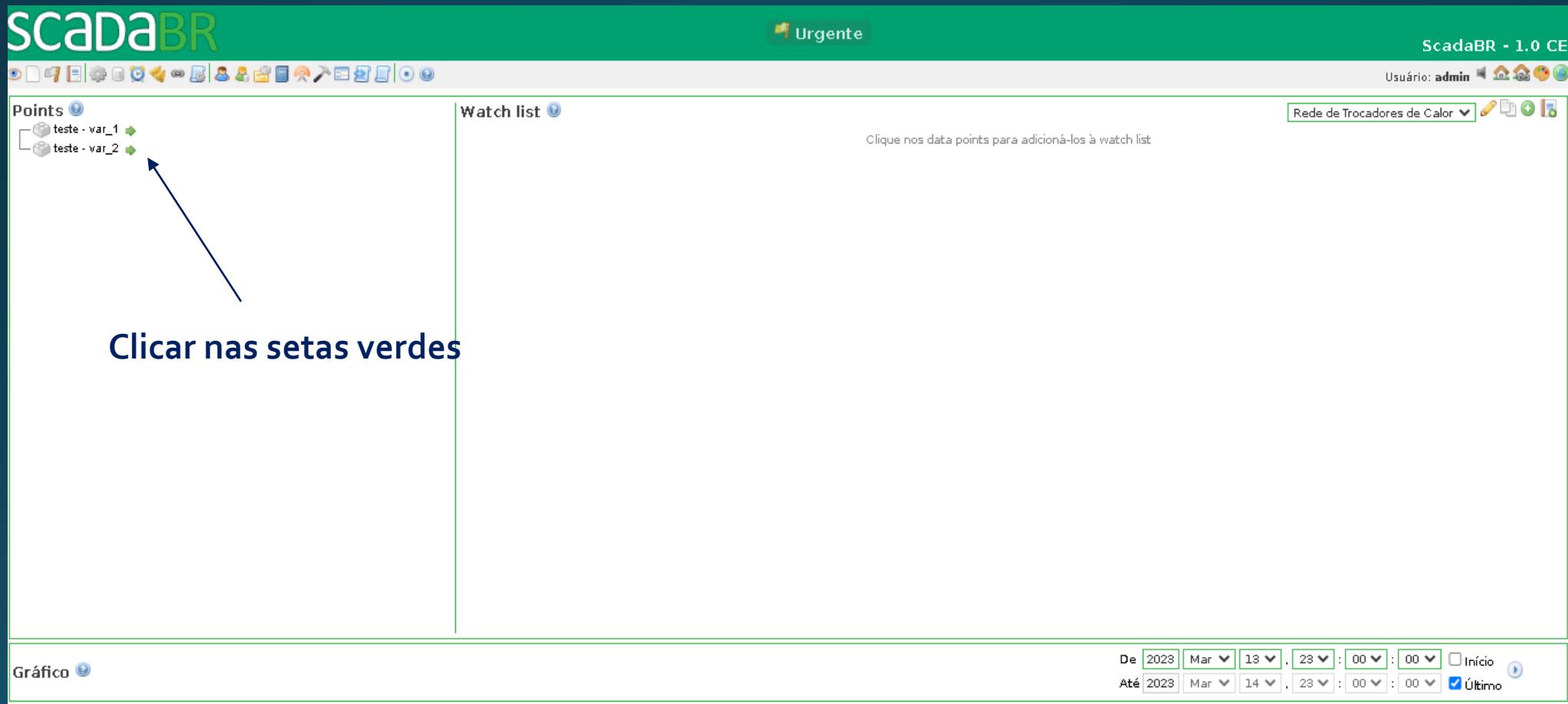
4. Implementação da Comunicação do Servidor com o ScadaBR

6. Visualização no Watchlist



4. Implementação da Comunicação do Servidor com o ScadaBR

6. Visualização no Watchlist



4. Implementação da Comunicação do Servidor com o ScadaBR

6. Visualização no Watchlist

