

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

LUIS GUILHERME VITAL ROMANO

**PROPOSTA E DESENVOLVIMENTO DE UM LABORATÓRIO DE REDES
INDUSTRIAIS**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO
2019

LUIS GUILHERME VITAL ROMANO

**PROPOSTA E DESENVOLVIMENTO DE UM LABORATÓRIO DE REDES
INDUSTRIAIS**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina TCC 2, do curso de Engenharia de Controle e Automação da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Prof. Dr. Wagner Endo

CORNÉLIO PROCÓPIO
2019



Universidade Tecnológica Federal do Paraná
Campus Cornélio Procópio
Departamento Acadêmico de Elétrica
Curso de Engenharia de Controle e Automação



FOLHA DE APROVAÇÃO

Luis Guilherme Vital Romano

Proposta e desenvolvimento de um laboratório de redes industriais

Trabalho de conclusão de curso apresentado às 13:30hs do dia 25/06/2019 como requisito parcial para a obtenção do título de Engenheiro de Controle e Automação no programa de Graduação em Engenharia de Controle e Automação da Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Avaliadora composta pelos professores abaixo assinados. Após deliberação, a Banca Avaliadora considerou o trabalho aprovado.

Prof(a). Dr(a). Wagner Endo - Presidente (Orientador)

Prof(a). Dr(a). Emerson Ravazzi Pires da Silva - (Membro)

Prof(a). Dr(a). Paulo Rogério Scalassara - (Membro)

AGRADECIMENTOS

Primeiramente agradeço a Deus por conceder sabedoria e me guiar durante os obstáculos que a vida nos impõem.

Aos meus familiares, principalmente minha mãe Roseane, exemplo de batalhadora, pelo constante incentivo e apoio durante minha graduação, sempre me ensinando o valor do estudo e me guiando pelo melhor caminho possível.

Agradeço o Prof. Dr. Wagner Endo pela orientação, ensinamentos e conselhos durante o desenvolvimento deste trabalho. Aos Professores Dr. Paulo Scalassara e Dr. Emerson Ravazzi pela importante contribuição nesse trabalho acadêmico.

Aos colegas do curso de graduação em Engenharia de Controle e Automação e todos aqueles que direta ou indiretamente colaboraram com o desenvolvimento deste trabalho.

RESUMO

ROMANO, Luis Guilherme Vital. **Proposta e desenvolvimento de um laboratório de redes industriais**. 2019. 106 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia de Controle e Automação. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2019.

O presente trabalho apresenta uma proposta de demonstração de redes industriais de forma instrutiva, utilizando ferramentas presentes nas principais indústrias, evidenciando os protocolos de comunicação. Este trabalho discorre no desenvolvimento de um laboratório de redes industriais, onde computadores são interconectados utilizando softwares de supervisão e comunicação, de fabricantes distintos, na qual possuem funções de aquisições, acesso, armazenamento e monitoramento de dados, com o intuito de descrever o âmbito industrial visando facilitar a aprendizagem de conceitos relacionados a comunicação e supervisão de processos.

Palavras-chave: Rede industrial. Protocolo de comunicação. Supervisório.

ABSTRACT

ROMANO, Luis Guilherme Vital. **Proposal and development of an industrial networks laboratory**. 2019. 106 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia de Controle e Automação. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2019.

The present work presents a proposal for demonstration of the industrial networks in an instructive way, using tools present in the main industries, evidencing the communication protocols. This work addresses the development of a laboratory of industrial networks, where computers are interconnected using supervisory and communication software from different manufacturers, in which they have the functions of acquisition, access, storage and monitoring of data, in order to describe the to facilitate the learning of concepts related to communication and process supervision.

Keywords: Industrial networks. Communication protocol. Supervisory.

LISTA DE FIGURAS

FIGURA 1	- Componentes de um sistema de comunicação de dados.....	15
FIGURA 2	- Linha de comunicação ponto-a-ponto, representado por dois pontos de comunicação.....	15
FIGURA 3	- Linha de comunicação multiponto, representado por mais dispositivos de comunicação.....	16
FIGURA 4	- Esquema de uma rede local (<i>LAN</i>), conhecida como intranet...	20
FIGURA 5	- Hardware típico de um sistema <i>SCADA</i>	26
FIGURA 6	- Esquemático lógico da implementação do laboratório de redes industriais.....	30
FIGURA 7	- Aplicação de configuração automática de <i>DCOM</i> e <i>Firewall</i> para softwares <i>Elipse</i>	31
FIGURA 8	- Esquemático físico dos computadores do laboratório de rede industriais.....	32
FIGURA 9	- Visualização física dos computadores do laboratório de redes industriais.....	33
FIGURA 10	- Senha para ingressar no grupo doméstico dos computadores do laboratório de redes industriais.....	33
FIGURA 11	- Desenho da tela do sistema supervisorio implementando no <i>Elipse SCADA</i> , referente a um processo de Caldeira.....	34
FIGURA 12	- Identificação das <i>tags</i> para visualização das variáveis do processo de Caldeira.....	35
FIGURA 13	- Desenho da tela do sistema supervisorio implementado no <i>SCADABR</i> , referente a um processo de mistura.....	37
FIGURA 14	- Identificação dos botões e indicação de velocidade dos motores e bombas no processo de mistura.....	37
FIGURA 15	- Criação do <i>data source</i> para comunicação <i>MODBUS</i>	38
FIGURA 16	- Esquemático entre <i>OPC Server</i> (<i>Elipse SCADA</i>) e <i>OPC Client</i> (<i>Elipse E3</i>).....	39
FIGURA 17	- Primeira visualização dos valores gerados das variáveis no <i>OPC Server</i> (<i>Elipse SCADA</i>).....	40

FIGURA 18 - Primeiro monitoramento dos valores das variáveis recebidos no OPC Client (Elipse E3).....	41
FIGURA 19 - Segunda visualização dos valores gerados das variáveis no OPC Server (Elipse SCADA).....	41
FIGURA 20 - Segundo monitoramento dos valores das variáveis recebidos no OPC Client (Elipse E3).....	42
FIGURA 21 - Esquemático da comunicação analisada entre OPC Server (Elipse SCADA) e MODBUS Simulator.....	43
FIGURA 22 - Configuração dos parâmetros de comunicação no MODBUS Simulator.....	44
FIGURA 23 - Acionamento e setpoint dos motores e bombas no OPC Server (Elipse SCADA).....	45
FIGURA 24 - Simulação do CLP no MODBUS Simulador.....	45
FIGURA 25 - Esquemático da comunicação analisada entre OPC Server (Elipse SCADA) e OPC Client (MATLAB).....	46
FIGURA 26 - Comunicação ao OPC Servidor para compartilhamento das tags utilizando o OPCTOOL.....	47
FIGURA 27 - Gráfico do Nível e Água, com 50 amostras de tempo de 3 segundos.....	48
FIGURA 28 - Gráfico do Pressão da Linha, com 50 amostras de tempo de 3 segundos.....	48
FIGURA 29 - Esquemático da comunicação analisada entre OPC Server (SCADABR) e MODBUS Simulator.....	50
FIGURA 30 - Acionamento e setpoint dos motores e bombas no OPC Server (SCADABR).....	51
FIGURA 31 - Simulação do CLP para motores e bombas do OPC Server (SCADABR).....	51

LISTA DE TABELAS

TABELA 1	-	Camadas de unidades de protocolos do <i>TCP/IP</i>	17
TABELA 2	-	Camadas de unidades de protocolos do <i>OSI</i>	18
TABELA 3	-	<i>Tags</i> criadas para obtenção e simulação das variáveis.....	36
TABELA 4	-	<i>Data Points</i> criados para ações nos motores e bombas.....	38

SUMÁRIO

1	INTRODUÇÃO.....	11
1.1	OBJETIVOS.....	12
1.1.1	Objetivo Geral.....	12
1.1.2	Objetivo Específicos.....	12
2	FUNDAMENTAÇÃO TEORICA.....	14
2.1	REDE DE COMUNICAÇÃO.....	14
2.1.1	Padrões de comunicação.....	16
2.1.2	Modelos de rede.....	17
2.1.3	Arquitetura da rede.....	18
2.1.4	Intranet.....	20
2.2	REDES DE COMUNICAÇÃO INDUSTRIAL.....	21
2.3	PROTOCOLOS DE COMUNICAÇÃO INDUSTRIAL.....	21
2.3.1	COMUNICAÇÃO <i>OPC</i>	22
2.3.2	PROTOCOLO <i>MODBUS</i>	23
3	ESTUDOS DOS SISTEMAS SUPERVISORIOS.....	25
3.1	FERRAMENTAS DE CONTROLE E AQUISIÇÃO DE DADOS.....	25
3.1.1	Estudos dos softwares e suas aplicações.....	27
3.1.2	Implementação da biblioteca <i>OPC TOOL</i>	28
4	PROPOSTA DO LABORATÓRIO DE REDES INDUSTRIAIS.....	29
4.1	CONFIGURAÇÃO <i>DCOM</i> E <i>FIREWALL</i> E INSTALAÇÃO DOS PROGRAMAS.....	30
4.2	CONFIGURAÇÃO DA REDE LOCAL E GRUPO DOMÉSTICO ENTRE OS COMPUTADORES.....	32
4.3	ELABORAÇÃO DA TELA DO SUPERVISÓRIO (<i>ELIPSE SCADA</i>), IDENTIFICAÇÃO E CRIAÇÃO DAS <i>TAGS</i>	34
4.4	ELABORAÇÃO DA TELA DO SUPERVISÓRIO (<i>SCADABR</i>), IDENTIFICAÇÃO E CRIAÇÃO DOS <i>DATA POINTS</i>	36
5	RESULTADOS E DISCUSSÕES.....	39
5.1	COMUNICAÇÃO ENTRE <i>OPC SERVER</i> (<i>ELIPSE SCADA</i>) E <i>OPC CLIENT</i> (<i>ELIPSE E3</i>).....	39
5.2	COMUNICAÇÃO ENTRE <i>OPC SERVER</i> (<i>ELIPSE SCADA</i>) E <i>MODBUS SIMULATOR</i>	42
5.3	COMUNICAÇÃO <i>OPC SERVER</i> (<i>ELIPSE SCADA</i>) E <i>OPC CLIENT</i> (<i>MATLAB</i>).....	46
5.4	COMUNICAÇÃO ENTRE <i>OPC SERVER</i> (<i>SCADABR</i>) E <i>MODBUS SIMULATOR</i>	49
6	CONSIDERAÇÕES FINAIS.....	52
6.1	TRABALHOS FUTUROS.....	52

REFERÊNCIAS.....	54
APÊNDICE A – TUTORIAL DE CONFIGURAÇÃO <i>DCOM</i> E <i>FIREWALL</i>.....	56
APÊNDICE B – COMO CRIAR UM GRUPO DOMÉSTICO.....	79
APÊNDICE C – TUTORIAL PARA COMUNICAÇÃO ENTRE <i>OPC SERVER</i> E <i>OPC CLIENT</i>.....	82
APÊNDICE D – TUTORIAL PARA COMUNICAÇÃO ENTRE <i>OPC SERVER</i> E <i>MODBUS SIMULATOR</i>.....	90
APÊNDICE E – TUTORIAL PARA COMUNICAÇÃO ENTRE <i>OPC SERVER</i> E <i>MATLAB</i>.....	100

1 INTRODUÇÃO

Nas últimas décadas, com o aumento da demanda e da eficiência dos sistemas eletrônicos, ocorreram mudanças em todas as áreas que necessitavam do trabalho humano e também para os sistemas de controle industriais (GALLOWAY; HANCKE, 2013). Surgiu, em vista disso, a exigência de implantação de sistemas automáticos. Pode-se considerar que um sistema de controle e automação consiste em uma arquitetura de computação distribuída onde diversos fornecedores devem garantir interfaces bem definidas para que diversos elementos de campo e suas funcionalidades possam operar de forma global (LUGLI; SANTOS, 2005).

Essa demanda por automação na indústria e nos setores mais diferentes está associada à possibilidade de aumentar a velocidade do processo de informação, já que as operações são cada vez mais complexas e variáveis. Consequentemente são necessários um maior número de controladores e mecanismos para permitir decisões mais rápidas, visando aumentar os níveis de produtividade e eficiência nas premissas de excelência operacional (CASSIOLATO, 2017).

De todas as tecnologias associadas ao controle industrial, as redes de comunicação é a que sofreu maiores evoluções nos últimos tempos, seguindo a tendência global da evolução das comunicações, praticamente em todos os ramos de atividade, desde as telecomunicações móveis, à internet e comunicação sem fios (*wireless*) (SCHNEIDER, 2007).

A rede industrial se preocupa com a implementação de protocolos de comunicação entre grupo de equipamentos, controladores digitais, vários tipos de *softwares* e também para sistemas externos. Estes protocolos tem um ambiente de rede semelhante as redes convencionais no nível físico, mas que tem requisitos significativamente diferentes (GALLOWAY; HANCKE, 2013).

Simultaneamente a integração das redes de comunicação, é de extrema importância a implantação de sistemas supervisórios, para monitoramento de variáveis do processo. No mercado existem várias ferramentas computacionais, desenvolvidas para a obtenção e armazenamento de dados de equipamentos para

um devido controle. Esses softwares são instalados na arquitetura de computadores interconectados, ou através de uma conexão externa.

O desenvolvimento das soluções em softwares, permitem um gerenciamento em tempo real dos processos de diversas áreas, assim sendo possível a colocação de alarmes e análises das variáveis da planta. Desse modo se tem um monitoramento para verificar possíveis falhas em componentes desta, atuando de forma imediata como prevenção de riscos.

Desta forma, este trabalho propõe desenvolver um laboratório de redes industriais, interligando diferentes redes de comunicação integrados a softwares distintos de supervisão, devido a perspectiva de aumentar a velocidade do processo de informação. É resultado da necessidade de evidenciar os conceitos de redes industriais no âmbito acadêmico, conveniente à tendência global das evoluções das redes de comunicação, tendo em vista facilitar a aprendizagem na prática das questões relacionados a comunicação e supervisão de processos.

1.1 Objetivos

1.1.1 Objetivo Geral

Este trabalho tem como objetivo desenvolver um laboratório de redes industriais, onde os computadores são comunicados utilizando as ferramentas de sistemas supervisorio e simuladores de dispositivos, de modo a representar e identificar um ambiente industrial de forma instrutiva.

1.1.2 Objetivos específicos

- Estudar as redes de comunicação;
- Estudar o funcionamento dos protocolos de comunicação industriais;
- Estudar as ferramentas para supervisão;
- Realizar as instalações e configurações para aplicações de softwares;
- Realizar a comunicação entre as ferramentas computacionais;
- Verificar a comunicação dos protocolos industriais;

- Desenvolver a arquitetura da rede de computadores;
- Evidenciar a comunicação dos protocolos industriais;
- Implementar a rede industrial aos computadores.

2 FUNDAMENTAÇÃO TEORICA

Nesse capítulo, apresentam conceitos de redes de comunicação e de redes industriais abordados durante o desenvolvimento do trabalho.

2.1 Rede de comunicação

Consideramos as redes de comunicação como sendo os ambientes onde um conjunto de dispositivos, sistemas de comunicações e pacotes de software, permitem a troca de informações entre os equipamentos e o usuário (DANTAS, 2002). As informações apresentadas em qualquer forma entre as partes, são definidas como dados.

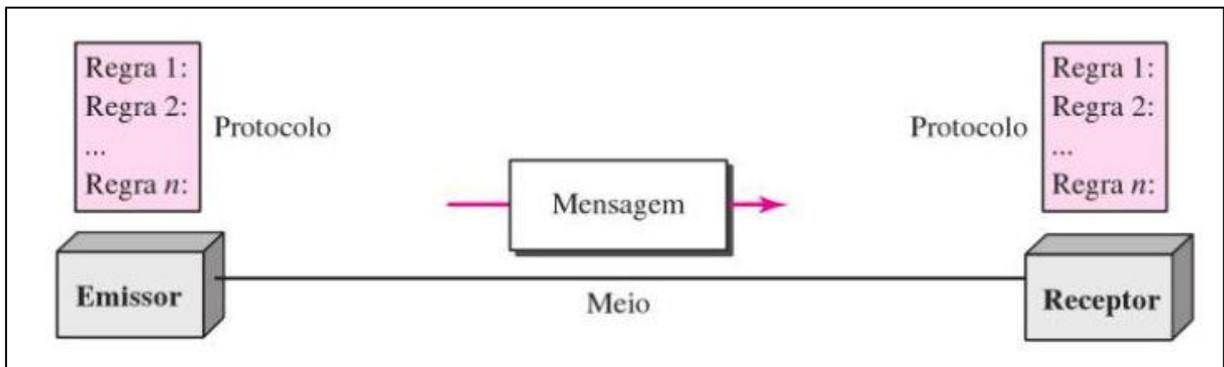
A comunicação dos dados é realizada entre dois dispositivos por intermédio de algum tipo de meio de transmissão, como um cabo condutor formado por fios. A eficácia de um sistema de comunicação de dados é dada pela entrega correta e com precisão das informações, no momento certo, sem atrasos significativos (FOROUZAN, 2007).

Um sistema de comunicação de dados é formado por cinco componentes, exemplificados na Figura 1:

- Mensagem: As mensagens são as informações a serem transmitidas, como por exemplo: texto, números, figuras, áudio e vídeo;
- Emissor: O emissor é o dispositivo que envia a mensagem de dados. Pode ser um computador, estação de trabalho, aparelho telefônico, televisão e assim por diante;
- Receptor: O receptor é o dispositivo que recebe a mensagem. Poder ser um computador, estação de trabalho, aparelho telefônico, televisão e assim por diante;
- Meio de transmissão. O meio de transmissão é o caminho físico pelo qual uma mensagem trafega do emissor ao receptor. Alguns exemplos de meio de transmissão são os seguintes: cabo de par trançado, cabo coaxial, cabo de fibra óptica e ondas de rádio;

- Protocolo: O protocolo é um conjunto de regras que controla a comunicação de dados. Representa um acordo entre os dispositivos de comunicação. Sem um protocolo, dois dispositivos podem estar conectados, mas sem se comunicar.

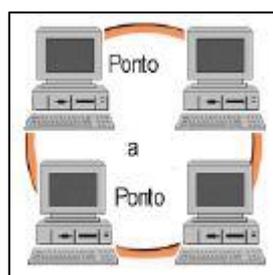
Figura 1 - Componentes de um sistema de comunicação de dados



Fonte: Forouzan (2007, p. 4)

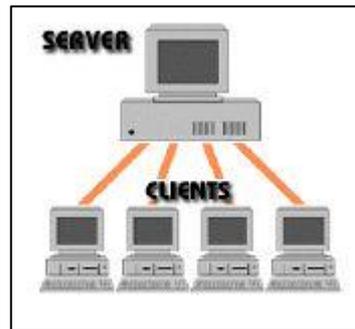
Ao organizar as interligações entre redes num sistema de comunicação, confrontamo-nos com diversas formas possíveis de utilização das linhas de comunicação. Elas podem ser por ponto-a-ponto, descrito na Figura 2, caracterizado por apenas dois pontos de comunicação, um em cada extremidade, ou multiponto, representado na Figura 3, que são as redes mais conhecidas como cliente-servidor, característico pela presença de mais dispositivos de comunicação (NOGUEIRA, 2009).

Figura 2 - Linha de comunicação ponto-a-ponto, representado por dois pontos de comunicação



Fonte: Nogueira (2009, p.30)

Figura 3 - Linha de comunicação multiponto, representado por mais dispositivos de comunicação



Fonte: Nogueira (2009, p.30)

Com o advento da comunicação e avanço das tecnologias dos computadores, surge o conceito para a interligação dos processamentos. A rede de computadores permitiu a migração do paradigma de processamento centralizado para o descentralizado, onde as atividades passariam a ser desenvolvidas distributivamente ao longo de vários mini e microcomputadores nos locais de trabalho. Com a crescente demanda de computadores nas empresas logo se percebeu a necessidade de interligá-los em rede, fortalecendo a convergência da informática com as telecomunicações (NOGUEIRA, 2009).

2.1.1 Padrões de comunicações

Padrões são necessários para criação e manutenção de um mercado competitivo para fabricantes de equipamentos, na garantia de interoperabilidade nacional e internacional de dados e da tecnologia de telecomunicações (FOROUZAN, 2007).

Os padrões que estabeleceram suas diretrizes de interconectividade são os:

- *International Organization for Standardization (ISO)*
- *International Telecommunication Union – Telecommunication Standards Sector (ITU – T)*

- *American National Standards Institute (ANSI)*
- *Institute of Electrical and Electronics Engineers (IEEE)*
- *Electronic Industries Association (EIA)*

2.1.2 Modelos de rede

Os modelos de rede são especificações padronizadas, detalhando as funções dos protocolos e seu inter-relacionamento, ou seja, representa uma abstração de como o ambiente deve funcionar. Os modelos de referência de protocolos permitem que um fabricante implemente de sua maneira um determinado conjunto de protocolos e, ainda assim, poderemos ter a interoperabilidade deste pacote de software com outro pacote padronizado desenvolvido por outro fabricante (DANTAS, 2002). Os dois protocolos de redes predominantes são o *OSI (Open System Interconnection)* e o modelo Internet (*TCP/IP*).

O modelo *TCP/IP* é o modelo de referência mais conhecido, o *TCP/IP (Transmission Control Protocol/Internet Protocol)* surgiu da rede *ARPANET*, rede de pesquisa criada pelo Departamento de Defesa do Governo Americano visando a conexão de inúmeras redes (DANTAS, 2002). Deste modo foi desenvolvida com a finalidade de interoperar diversos computadores com diferentes ambientes de software e hardware. O *IP* trata do roteamento de datagramas, ao passo que o *TCP* seria responsável por funções de mais alto nível, como segmentação, remontagem e detecção de erros (FOROUZAN, 2007). O modelo *TCP/IP* é baseado em cinco camadas de unidades de protocolos descritas na Tabela 1.

Tabela 1 – Camadas de unidades de protocolos do *TCP/IP*

Nível	Camada
5	Aplicação
4	Transporte
3	Rede
2	Enlace
1	Físico

Fonte: Adaptado de Dantas (2002, p.111)

Já o modelo *OSI (Open System Interconnection)* é um modelo de rede da *ISO (International Organization for Standardization)*, como diz seu acrônimo, é sistema aberto de conjunto de protocolos que permite que dois sistemas diferentes se comuniquem independentemente de suas arquiteturas subjacentes. Com a função de facilitar a comunicação entre sistemas diferentes sem a necessidade de realizar mudanças na lógica do hardware e software de ambos (FOROUZAN, 2007). O modelo *OSI* é composto por sete camadas ordenadas caracterizadas na Tabela 2.

Tabela 2 – Camadas de unidades de protocolos do OSI

Nível	Camada
7	Aplicação
6	Apresentação
5	Sessão
4	Transporte
3	Rede
2	Enlace
1	Físico

Fonte: Adaptado de Dantas (2002, p.114)

2.1.3 Arquitetura da rede

A função que permite a comunicação entre aplicações executadas em máquinas distintas envolve uma série de detalhes, que devem ser abordados cuidadosamente, afim de obter uma comunicação precisa, segura e livre de erros. Para reduzir a complexidade de projeto, a maioria das redes de computadores são estruturadas em camadas, onde cada camada desempenha uma função específica dentro do objetivo maior que é a tarefa de comunicação, assim neste contexto o conjunto das camadas de protocolos é definido com a Arquitetura de protocolos (DANTAS, 2002).

Essa arquitetura é baseada nos determinados modelos de referências descritos, podendo ser arquitetura proprietárias, que são ambientes implementados por fabricantes na comunicação de redes fechadas e abertas, onde se tem uma

ênfase, por ser uma arquitetura de projetos independentes de fabricantes e por suas definições serem estabelecidas em órgãos normativos (DANTAS, 2002).

Diante à modelos *OSI* e *TCP/IP*, o primeiro acreditava-se que tornaria o padrão final para comunicação de dados, entretanto isso não ocorreu, pois com a advinda dos conjuntos de protocolos *TCP/IP*, o modelo *OSI* não foram ricos o suficiente nas camadas mais baixas, sendo necessárias camadas extras, o que o *TCP/IP* supria essa falta, se tornando a arquitetura comercial predominante, sendo usado e testado na forma intensiva nas redes interconectadas (FOROUZAN, 2007).

A arquitetura *TCP/IP* é baseada em conjunto de protocolos hierárquicos, compostos por módulos interativos, cada um dos quais provendo funcionalidade específicas, denominada em 5 níveis, camada de aplicação, transporte, rede, enlaces e físico, que serão descritos abaixo (FOROUZAN, 2007).

A camada de aplicação é caracterizada por protocolos que solicitam à camada de transporte os serviços orientados e não orientados à conexão. Os protocolos que solicitam serviços ao *TCP*, são o *BGP* (*Border Gateway Protocol*), *FTP* (*File Transfer Protocol*), *HTTP* (*Hypertext Transfer Protocol*), *SMTP* (*Simple Mail Transfer Protocol*), *Telnet*, *DNS* (*Domain Name Server*) e *RPC* (*Remote Procedure Call*) (DANTAS, 2002).

Na camada de transporte são definidos três protocolos, o *TCP* (*Transmission Control Protocol*), *UDP* (*User Datagram Protocol*) e o *SCTP* (*Stream Control Trasmision Protocol*), na qual são responsáveis pela movimentação de dados. Vale a pena destacar que o *TCP* é conhecido como um protocolo “pessimista”, onde acredita que os envios dos segmentos sempre irão ocorrer perdas e fora de ordem, desta forma ele conta com uma implementação complexa (*Go-Back-n*), quando a transmissão perde pacotes de segmentos *n*, todos os pacotes de *n* até o final são retransmitidos. Já o *UDP*, é o protocolo otimista, enviam os pacotes de dados acreditando que todos vão chegar sem problemas, tendo uma característica de simplicidade na implementação (NOGUEIRA, 2009).

A camada de Rede, é a camada de ligação entre redes, composta pelo principal protocolo *IP* e por quatro protocolos auxiliares de suporte, *ARP*, *RARP*, *ICMP* e *IGMP*. O *IP* (*Internetworking Protocol*) é um protocolo sem conexão e não confiável, pois não dispõe de nenhuma verificação ou correção de erros, portanto isso não deve ser considerado um ponto fraco, visto que ele provém funções essenciais de

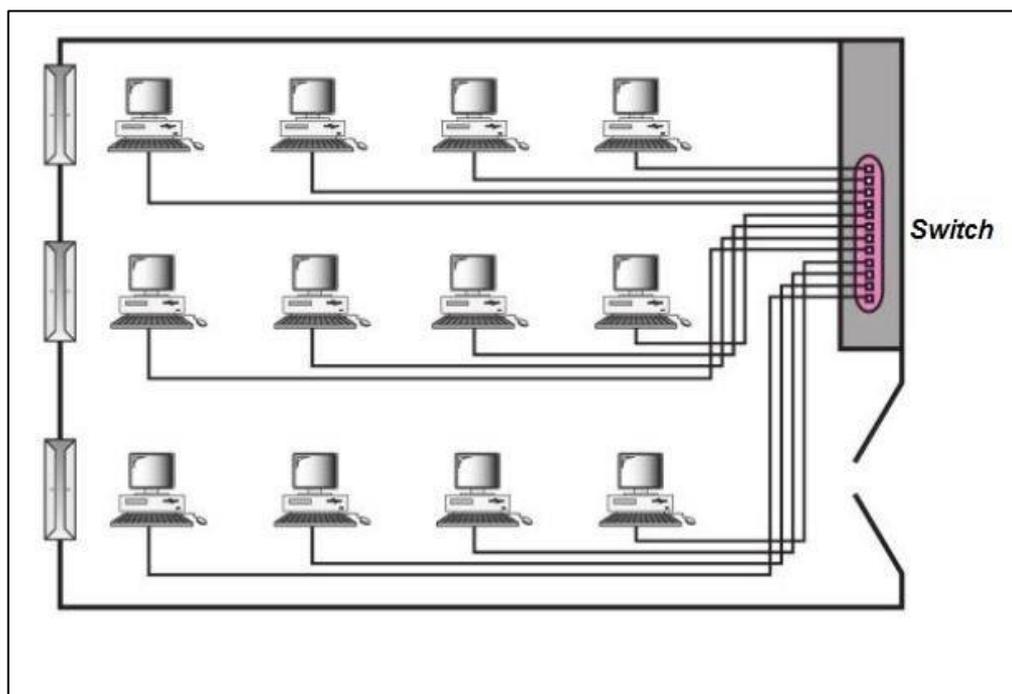
transmissão que dão liberdade aos usuários para acrescentar funcionalidades necessárias para dada aplicação (FOROUZAN, 2007).

Nas camadas de Enlace e Física não define nenhum protocolo específico, são definidos pelas redes subjacentes (*host-rede*), por exemplo podendo ser em uma rede local.

2.1.4 Intranet

Conhecida como Rede Local (*LAN – Area Network*), descrito na Figura 4, tem a função da comunicação entre os módulos do sistema *SCADA (Supervisory Control and Data Acquisition)* desenvolvidos estabelecendo uma rede local, privada e protegida baseada na arquitetura *Ethernet*, denominada Intranet. Esta arquitetura define o cabeamento, os sinais elétricos utilizados e atribui para cada ponto da rede um endereço físico único, chamado de *MAC (Media Access Control)*. O protocolo *Ethernet* está localizado na camada de enlace do modelo *OSI* e na camada física do modelo *TCP/IP* (PEREIRA, 2015).

Figura 4 - Esquema de uma rede local (*LAN*), conhecida como intranet.



Fonte: Adaptado Forouzan (2007, p.14)

Neste contexto, a intranet se refere a uma rede privada, sem acesso externo, dedicada ao transporte de dados entre os computadores, *workstations*, servidores de arquivos, Banco de Dados e dispositivos inteligentes do sistema SCADA, contidos em local com alguns quilômetros de extensão. A tecnologia de transmissão quase sempre consiste em um cabo, a quais todas as máquinas estão conectadas e apresentam uma velocidade que pode variar de 10 a 1.000 Mbps, tendo um baixo retardo e cometendo pouquíssimos erros (NOGUEIRA, 2009).

2.2 Redes de comunicação industrial

A rede industrial tem como objetivo principal a interoperabilidade e flexibilidade de operação, onde são constituídos por protocolos de comunicação (padronizados ou não) que transportam sinais os quais trafegam sob um barramento de comunicação comum, para a interligação de tarefas, a fim de controlar processos industriais (LUGLI; SANTOS, 2005). As padronizações dos protocolos auxiliam na interoperabilidade entre diferentes tecnologias de redes industriais. A escolha de um padrão de protocolo de comunicação, influencia diretamente no desempenho da rede e conseqüentemente no controle do processo (LUGLI; SANTOS, 2005).

Com o uso de redes de comunicação digitais característicos nas indústrias durante a última década, levou a melhoraria na precisão e na integridade do sinal digital de ponta a ponta (ELECTRICAL TECHNOLOGY, 2016).

As arquiteturas de redes industriais devem fazer com que os sinais trafeguem desde o chão de fábrica até o nível de informação gerencial (LUGLI; SANTOS, 2005).

2.3 Protocolos de Comunicação industrial

Com base nos padrões de comunicação, os protocolos de comunicação industrial tiveram seu avanço nas redes industriais devido ao seu alto custo/benefício, onde foram difundidos tipos de protocolos de chão de fábrica, podendo citar os principais protocolos: *Profibus*, *Devicenet*, *CAN*, *Fieldbus Foundation*, *MODBUS*, *OPC*, entre outros (LUGLI; SANTOS; FRANCO, 2008). Abaixo serão definidos e descritos os protocolos que serão abordados no desenvolvimento desse trabalho.

2.3.1 Comunicação OPC

A comunicação *OPC (OLE for Process Control)* foi desenvolvida com um intuito de ser um padrão baseado na tecnologia *COM/DCOM* para acesso à dados de tempo real dentro do sistema operacional *Windows*. Ela é gerenciada pela organização *OPC Foundation*, a qual possui um site disponível em www.opcfoundation.org (FONSECA, 2002).

O *OPC* é o padrão de interoperabilidade para a troca segura e confiável de dados no espaço de automação industrial. É independentemente da plataforma garante o fluxo contínuo de informações entre dispositivos de vários fornecedores. O padrão *OPC* é uma série de especificações desenvolvidas por fornecedores da indústria, usuários finais e desenvolvedores de softwares. Essas especificações definem a interface entre Clientes e Servidores, bem como Servidores e Servidores, incluindo acesso a dados em tempo real, monitoramento de alarmes e eventos, acesso a dados históricos e outras aplicações (OPC FOUNDATION, 2017).

A primeira especificação *OPC*, chamada *OPC Specification Version 1.0*, tinha como objetivo do grupo em atender as necessidades da indústria, assim com sua atualização, passou a ser chamada de *OPC Data Access Specification Version 1.0A*. Atualmente existem as seguintes especificações publicadas ou em processo de aprovação (FONSECA, 2002):

- *OPC Overview* – Descrição geral dos campos de aplicação das especificações *OPC*.
- *OPC Common Definitions and Interfaces* – Definição das funcionalidades básicas para as demais especificações.
- *OPC Data Access Specification* – Definição da interface para leitura e escrita de dados de tempo real.
- *OPC Alarms and Events Specification* – Definição da interface para monitoração de eventos.
- *OPC Historical Data Access Specification* – Definição da interface para acesso a dados históricos.

- *OPC Batch Specification* – Definição da interface para acesso aos dados de processos por batelada (batch). Esta especificação é uma extensão da *OPC Data Access Specification*.
- *OPC Security Specification* – Definição da interface para utilização de políticas de segurança.
- *OPC and XML* – Integração entre *OPC* e *XML* para aplicações via Internet (*web*).

2.3.2 Protocolo MODBUS

O *MODBUS* é um protocolo que se situa na camada de aplicação do modelo *OSI* para comunicação entre dispositivos em rede, essencialmente para troca de dados no campo da automação (GOMES, 2014). Desse modo estabelece comunicação entre dispositivos mestre-escravo/cliente-servidor. Em um relacionamento mestre-escravo, a comunicação sempre ocorre em pares, onde um dispositivo deve iniciar a requisição e então aguardar por uma resposta. Normalmente, o mestre é uma interface homem-máquina (*IHM*) ou sistema *SCADA* e o escravo é um sensor, controlador lógico programável (*CLP*) ou outro equipamento. O conteúdo das requisições e respostas das camadas de rede, pelas quais essas mensagens são enviadas, são definidos pelas diferentes camadas do protocolo (NATIONAL INSTRUMENTS, 2014).

Um dos modos em que os dados são formatados antes da transmissão é o *RTU (Remote Terminal Unit)*, sendo o mais utilizado pelo fato de compor o frame em binário, em que cada byte contém dois caracteres de 4 bits cada. Ele difere do formato *ASCII (American Standard Code for Information Interchange)*, ocupando menos recurso de rede, pois um byte do formato *ASCII* é formado por 7 bits e pode ser entendido somente por ler a sequência.

Os outros formatos são o *MODBUS/TCP*, onde os dados são encapsulados em pacote e enviados pela rede *Ethernet* via *TCP*, utilizando-se do *CSMA/CD (Carrier Sense Multiple Access with Collision Detection)* que gerencia as possíveis colisões que possam ocorrer, caso outros dispositivos estejam usando o mesmo canal, dando taxas de prioridade ao pacote enviado. E também o *MODBUS Plus*, que é de

propriedade da *Schneider Electric*, sendo que para utilizá-lo é necessário ter a licença de uso (INSTITUTO NEWTON C BRAGA, 2014).

3 ESTUDO DOS SISTEMAS SUPERVISÓRIOS

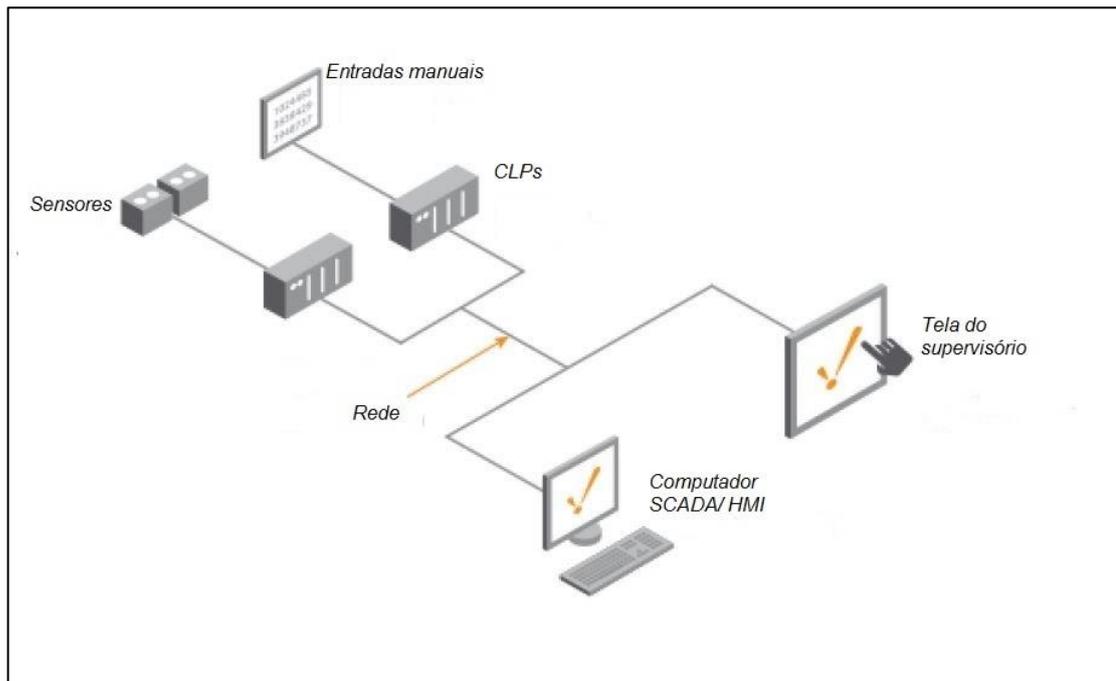
Nesse capítulo serão apresentados o estudo das ferramentas para supervisão e aquisições de dados.

3.1 Ferramentas de controle e aquisição de dados

Os sistemas *SCADA* (Supervisory Control and Data Acquisition), como o seu acrónimo indica, são sistemas desenvolvidos com o objetivo de supervisionar e monitorar as variáveis dos equipamentos ou plataformas industriais que se pretende controlar (GOMES, 2014). O sistema *SCADA* esquematizado na Figura 5 é constituído por:

- Um ou mais dispositivos de interface de dados, geralmente *CLPs*, ligados aos sensores do processo, que convertem os sinais analógicos recebidos dos mesmos, em sinais digitais entendidos pelo protocolo de comunicação que se esteja utilizando. Estes sinais são depois enviados para o sistema supervisão;
- Um sistema de comunicações usado para transferir dados entre os dispositivos de campo (*CLPs*) e o sistema supervisão;
- Um sistema de supervisão que faz a aquisição dos dados do processo e envia comandos de controle para o mesmo;
- Uma interface Homem-Máquina (*IHM*) que permite a supervisão e controle do processo em tempo real pelo operador humano.

Figura 5 - Hardware típico de um sistema SCADA



Fonte: Adaptado Gomes (2014, p.5)

O SCADA vêm sendo desenvolvido para diferentes áreas de atuação, tanto industriais como científicas, com o mesmo objetivo de apresentar ao observador os valores das variáveis ou parâmetros que estão sendo medidos, de modo a proporcionar níveis maiores de qualidade, redução dos custos operacionais, maior desempenho de produção e fundamentalmente, para facilitar a excelência operacional (CASSIOLATO, 2017).

Os sistemas SCADA são particularmente propícios para realizar tarefas em locais de operação ou manutenção críticas (FU; NI, 2015). Entre algumas das funções mais utilizadas desses sistemas estão (GOMES, 2014):

- Geração de gráficos e relatórios com o histórico do processo;
- Detecção de alarmes e registro de eventos em sistemas automatizados;
- Controle de processos incluindo envio remoto de parâmetros e *set-points*, acionamento e comando de equipamentos;
- Uso de linguagens de *script* para desenvolvimento de lógicas de automação.

3.1.1 Estudo dos softwares e suas aplicações

Os softwares disponibilizados pelos fabricantes para monitoramento e aquisição de dados em tempo real são poderosas ferramentas para o desenvolvimento de sistemas de supervisão e controle de processos abordada nesse trabalho. Através de uma interface com diversas aplicações, sendo totalmente configurável pelo usuário, permite o monitoramento de variáveis em quase tempo real, através de gráficos e objetos que estão relacionados com as variáveis físicas de campo.

Existem ferramentas também, desenvolvidas em modelo "*open-source*", ou seja, possui uma licença gratuita. Toda a documentação e o código-fonte do sistema estão à disposição aos usuários, sendo permitido modificar e redistribuir o software (COSTA; SOUZA, 2017).

Os sistemas supervisórios, são baseados nas aplicações chamadas *Tags*, Telas, Alarmes, Receitas, Históricos e Relatórios, que serão descritas abaixo.

As *tags* estão relacionadas a supervisão de um processo através da leitura de variáveis de processos no campo. Esta é associada aos valores dessas variáveis. São definidos tipos de *tags*, cada uma com sua determinada aplicação, como por exemplo, *tags* com drivers de comunicação (arquivos .DLL), responsáveis pela interligação do software com algum equipamento externo.

A função alarme permite a verificação dos valores das *tags*, podendo ser disparados quando os valores associados são ocorridos em quatro situações e prioridades diferentes: *LoLo* (baixo crítico), *Low* (baixo), *High* (alto), *HiHi* (alto crítico). Fora dessas faixas, os valores dos *tags* são considerados normais.

A ferramenta Receita é um conjunto de valores pré-definidos que podem ser carregados para um grupo de *tags* a fim de configurar um processo específico, com o objetivo de facilitar a tarefa do operador e evitando erros.

Os Históricos permitem o armazenamento da variação dos dados de um processo ao longo do tempo, possibilitando análises futuras. Estes Históricos podem

ser gerados de duas maneiras diferentes: de forma Contínua (armazena os dados continuamente) ou em Bateladas (armazena os dados em lotes).

Para verificar o comportamento dos dados monitorados, os sistemas contam com uma ferramenta de Diagnósticos, chamada Relatórios. Ela permite imprimir os dados de arquivos de Histórico ou de Alarmes ou ainda valores de *tags* em tempo real para serem avaliados (GOMES, 2014).

Os softwares de sistemas supervisórios são desenvolvidos com as mesmas finalidades, porem alguns tem uma abordagem mais avançada das aplicações, como exemplo a utilização de banco de dados e representações gráficas.

Para a comunicação com equipamentos de aquisição de dados, controladores, CLPs (Controladores Lógicos Programáveis), ou qualquer outro tipo de equipamento de campo de “chão de fábrica”, normalmente possuem interfaces que permitem sua conexão ao software como, padrões seriais, *TCP/IP*, entre outros. O software específico para essa comunicação utiliza a conexão *TCP/IP*, simulando valores para as aplicações compatíveis com os protocolos, *MODBUS RTU* e *MODBUS TCP*.

3.1.2 Implementação da biblioteca *OPC TOOL*

OPC Toolbox fornece acesso a dados *OPC online* e históricos diretamente do MATLAB e Simulink. Podendo ler, escrever e registrar dados *OPC* de dispositivos, como sistemas de controle distribuído, controle de supervisão e sistemas de aquisição de dados, e controladores lógicos programáveis. O *OPC Toolbox* permite que você trabalhe com dados de servidores ativos e historiadores de dados que estejam de acordo com o padrão *OPC Data Access (DA)*, com o padrão *OPC Historical Data Access (HDA)* e o padrão *OPC Unified Architecture (UA)* (MATHWORKS, 2017).

Essa biblioteca é empregada no trabalho proposto no intuito de comunicar os softwares de sistemas supervisórios descritos com o MATLAB, de modo a aumentar o campo de aplicações, sobre manuseio das variáveis obtidas.

4 PROPOSTA DO LABORATÓRIO DE REDES INDUSTRIAIS

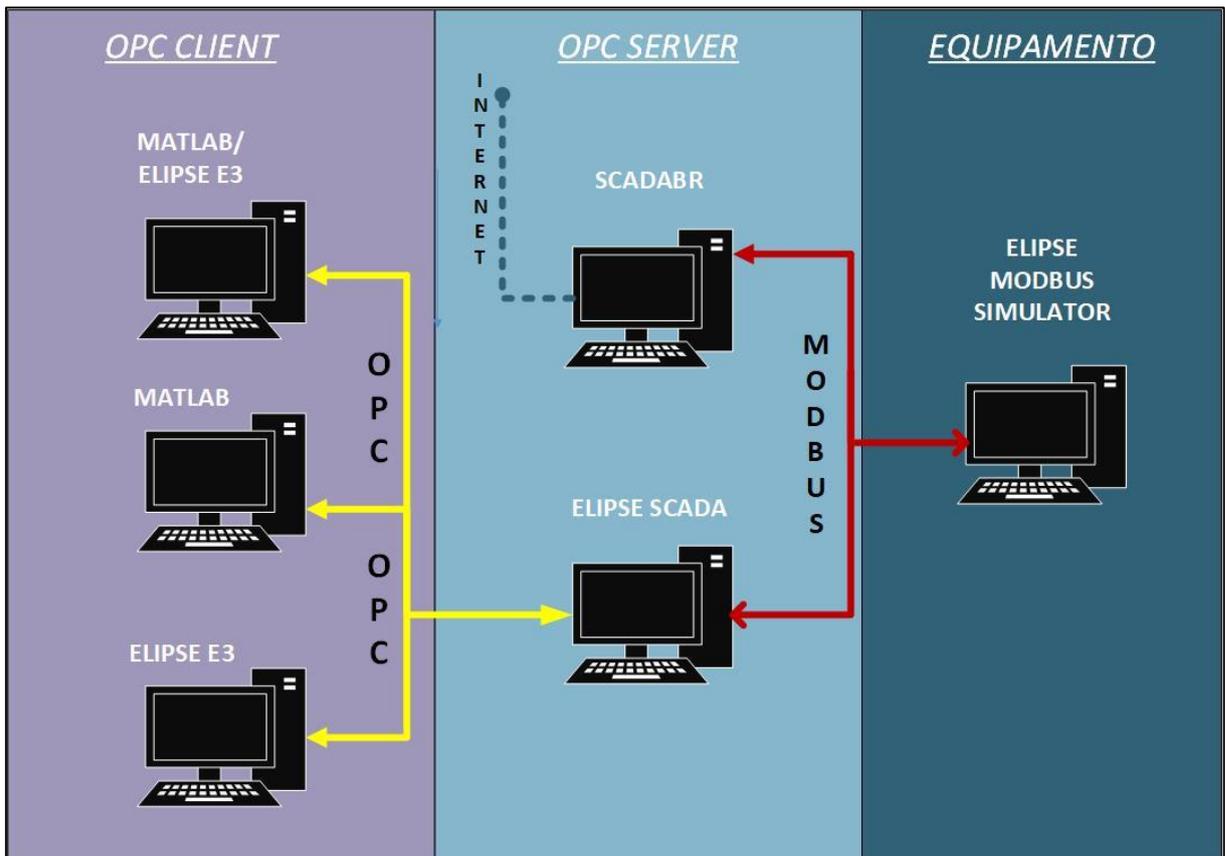
A proposta do trabalho consiste no desenvolvimento de uma arquitetura de rede de computadores, usando os protocolos comunicação *OPC* e *MODBUS*, integrados aos softwares de supervisão, com o intuito de descrever o âmbito industrial visando compreender os conceitos relacionados a comunicação e supervisão de processos.

O trabalho baseia-se na implementação de um sistema *SCADA OPC Server*, que obtêm dados de um simulador *MODBUS*, que funciona como um equipamento (*CLP*, inversores...) na rede, demonstrando a utilização do protocolo de comunicação *MODBUS*, e assim sendo capaz de endereça-los através do protocolo de comunicação *OPC* a três computadores *OPC Client*. Estes são compostos por um sistema *SCADA* e pela ferramenta computacional *MATLAB*, para análise e construção de gráficos. Além disso, a rede dispõe de outro sistema *SCADA OPC Server* com servidor *web*, que obtem e armazena os dados do processo. Para facilitar o entendimento, o esquema proposto é exemplificado na Figura 6.

O *OPC Client* tem a função de receber os dados das variáveis do *OPC Server*, referente aos equipamentos (*CLP*, sensores e etc), afim de monitorar as variáveis, para implementar e testar soluções com precisão. O computador *OPC Server*, tem a finalidade de definir, acessar e modificar as variáveis do processo, tendo um controle supervísório, e juntamente com sistema *web*, será possível a criação de alarmes e históricos, permitindo armazenar em banco de dados as variações das informações de um processo ao longo do tempo, para análises futuras.

A implementação do laboratório de redes industriais é muito além de definições de *setpoint*, e sim caracterizadas pela presença de distintos sistemas supervísórios e pelas interligações com diferentes tipos de redes de comunicação.

Figura 6 - Esquemático lógico de implementação do laboratório de redes industriais



Fonte: Autoria própria

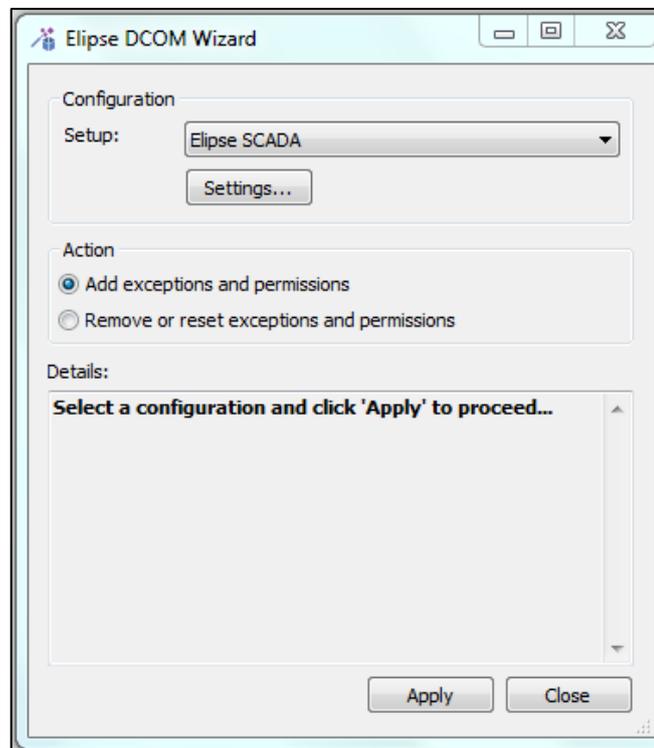
4.1 Configuração *DCOM* e *Firewall* e instalação dos programas

Em todos computadores presentes no laboratório de rede de comunicação industrial foram instalados o sistema operacional Microsoft Windows 7, devido a sua interface amigável, compatibilidade com os programas e grande utilização nos ambientes industriais. Com a determinação desse sistema, são importantes a realização de configurações do tipo *DCOM* e *Firewall* para êxito nas comunicações entre as aplicações SCADA e sistema operacional, com o intuito de dar permissões de acessos aos usuários e liberação de portas de comunicação.

Essas configurações foram realizadas para a comunicação *OPC* entre computadores da rede, porem nas aplicações Elipse, elas são desempenhadas por uma aplicação disponível no site da empresa fabricante, chamada "*EDCOMWizard*", onde tem a finalidade de realizar automaticamente, de maneira simples, todas as permissões para cada programa Elipse desejado (Figura 7). Com a finalidade de

mostrar que tipo de modificações são realizadas e poder fazê-las manualmente, são evidenciadas passo a passo no tutorial do Apêndice A.

Figura 7 – Aplicação de configuração automática de DCOM e Firewall para softwares Elipse



Fonte: Adaptado Elipse Software

Posteriormente as configurações, foram instalados em cada computador da rede de comunicação os respectivos programas, sendo disponíveis para qualquer usuário e aplicação:

- Elipse SCADA
- Elipse E3
- Elipse MODBUS Simulator
- SCADABR
- MATLAB

Com isso cada computador pode realizar qualquer função na rede industrial, sendo o *OPC Server* ou *OPC Client*, estando definido pela escolha do usuário, dependendo da operação que deseja utilizar.

É importante enfatizar que aplicações Elipse são utilizados em ambiente demo, que pode ocorrer limitações de uso de tarefas e o programa SCADABR é uma

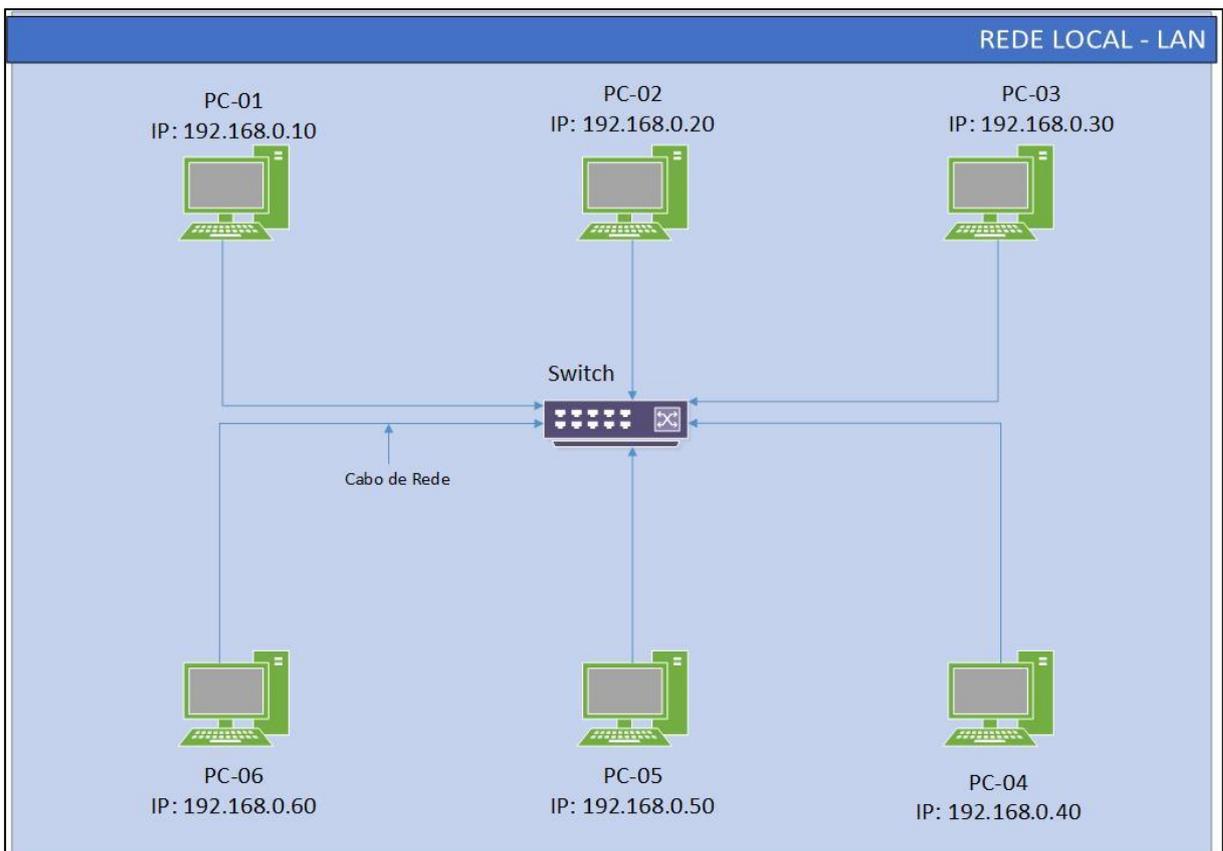
ferramenta open-source de protocolo aberto, assim tendo a disponibilidade de todas as funcionalidades.

Para instalação do Eclipse E3 é necessário ter instalado no computador a versão do *net framework* 4.0 ou superior e ao software SCADABR necessita da instalação do java na versão 6.0.

4.2 Configuração da Rede Local e Grupo Doméstico entre os computadores

Para a interconexão dos computadores, foi implantada uma rede privada e protegida, denominada rede local (*LAN*). Isso é efetuada através de um *switch*, que transmitem os dados para os computadores conectados, desta forma foram nomeados e definidos com *IP's* fixos, para melhor visualização e acesso na rede.

Figura 8 – Esquemático físico dos computadores do laboratório de redes industriais



Fonte: Autoria própria

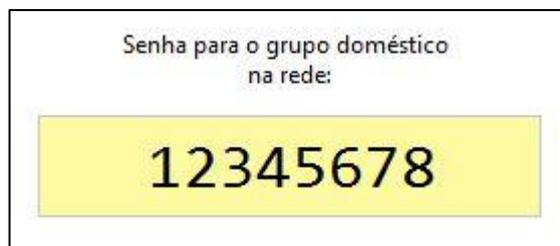
Figura 9 – Visualização física dos computadores do laboratório de redes industriais



Fonte: Autoria própria

De modo a permitir o compartilhamento dos dados e informações entre os computadores na rede de maneira segura e confiável, foi criado um grupo doméstico, onde foram ingressados cada computador a ele, sendo necessária uma senha (Figura 10). O passo a passo para criação de uma rede doméstica é descrito no Apêndice B.

Figura 10 – Senha para ingressar no grupo da rede doméstica dos computadores do laboratório de redes industriais



Fonte: Autoria própria

Com essas configurações do grupo doméstico solucionamos e evitamos qualquer problema que possa ocorrer através dos protocolos de comunicações industriais, relacionado ao sistema operacional *Microsoft Windows*. Para controle de acesso dos

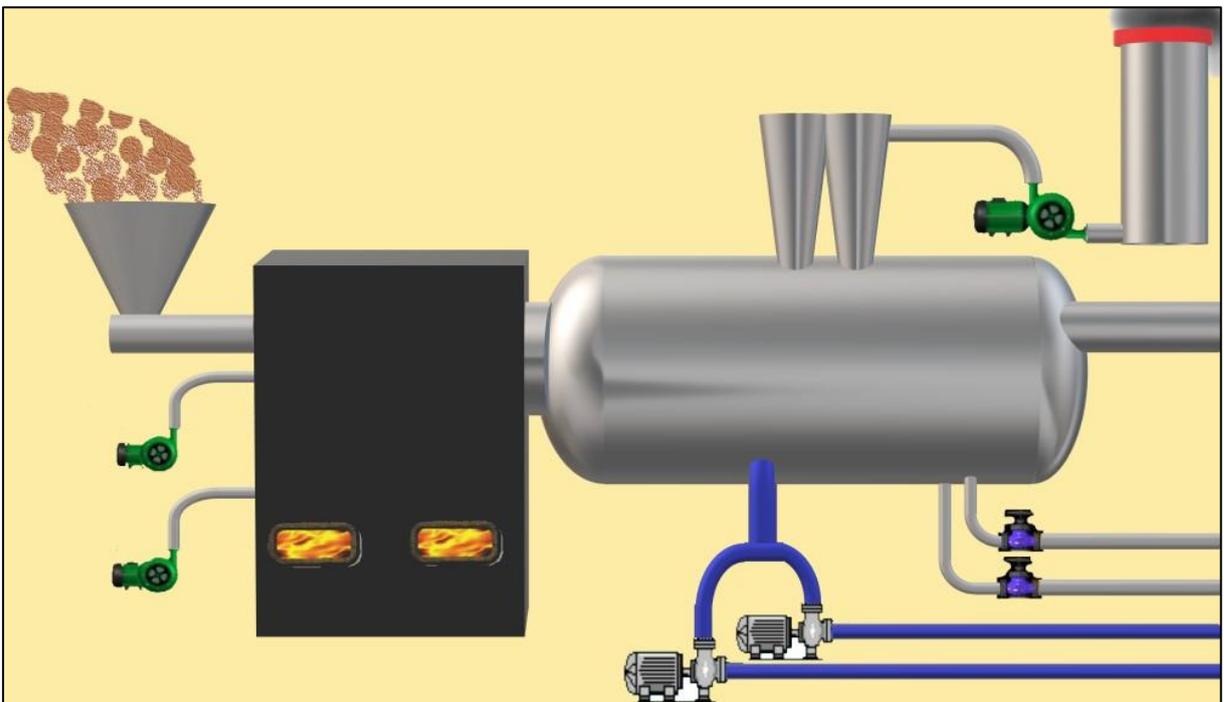
computadores do laboratório de redes industriais, a fim de estudos de aplicações industriais, foi definida uma senha aos usuários.

- Senha: 123456

4.3 Elaboração da tela do supervisorio (Elipse SCADA), identificação e criação das Tags

Com o intuito de evidenciar a comunicação em âmbito industrial, simulando um processo real com seus respectivos monitoramentos de variáveis, foi elaborado telas para o supervisorio *OPC Server*. Onde a primeira tela descreve um processo de uma caldeira industrial e a outra um processo de Mistura que será descrito no próximo tópico.

Figura 11 – Desenho da tela do sistema supervisorio implementado no Elipse SCADA, referente a um processo de Caldeira



Fonte: Autoria própria

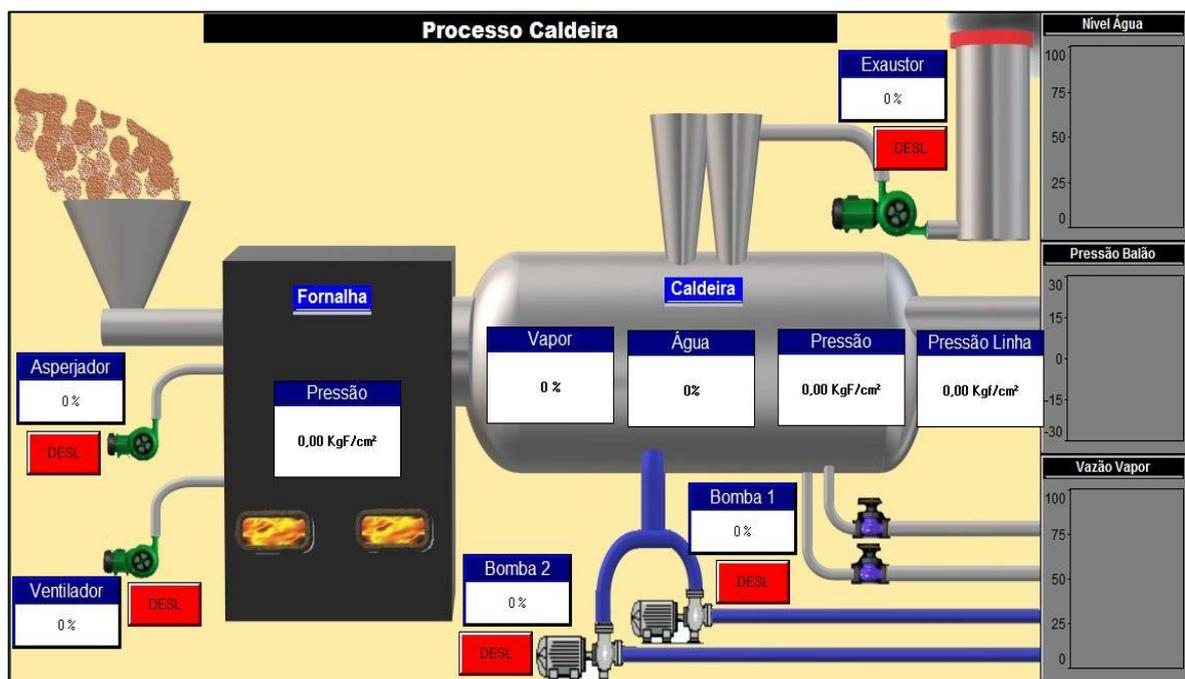
No processo de Caldeira, da Figura 11, ilustra a representação da alimentação da Fornalha, a Caldeira e o chaminé do Exaustor, onde são importantes o monitoramento das variáveis, tais como pressão, vazão e nível e do mesmo modo

o acionamentos e controle da velocidade de rotação dos motores e bombas. Na Figura 12, foram identificados os pontos de captação das variáveis, tal como as pressões na Fornalha, Balão e na linha, dadas em Kgf/cm , o nível de água dada em porcentagem referente a capacidade total do balão e a vazão do vapor. Nos motores e bombas são apresentados visualmente o estado de operação e sua porcentagem de rotação.

Para simulação dessas variáveis são adicionados *tags* de modo representar sinais obtidos por sensores, CLPs, inversores, dentre outros equipamentos de controle e obtenção de dados do chão de fábrica (Tabela 3). Com relação as variáveis do processo foram referenciadas a *tags demo*, que geram valores aleatórios, com o intuito de simular variações dos parâmetros que pode ocorrer em um âmbito industrial. Nas *tags demo* foram definidos limites de variações, para se obter parâmetros aceitáveis de valores, de acordo com determinada variável do processo.

Nos motores e bombas são criadas *tags PLC*, simulando a inserção de um equipamento conectado à rede, onde este tem a ação de liga/desliga e determinação de valores. Essas *tags* têm a função de definir a velocidade de rotação dos motores e bombas, dentro de limites de porcentagem de 0 a 100%, sendo 0% o equipamento totalmente parado e 100%, a máxima capacidade de operação.

Figura 12 – Identificação das *tags* para visualização das variáveis do processo de Caldeira



Fonte: Autoria própria

Tabela 3 – Tags criadas para obtenção e simulação das variáveis

Tags	Descrição
Asperjador	Motor do Asperjador
Bomba 1	Bomba primária de abastecimento de água
Bomba 2	Bomba reserva de abastecimento de água
Exaustor	Motor do Exaustor
Ventilador	Motor do Ventilador
Vel_Aasperjador	Velocidade de rotação do Motor do Asperjador
Vel_Bomba1	Velocidade de rotação da Bomba primária
Vel_Bomba2	Velocidade de rotação da Bomba reserva
Vel_Exaustor	Velocidade de rotação do Motor do Exaustor
Vel_Ventilador	Velocidade de rotação do Motor do Exaustor
NívelÁgua	Variável do Nível de Água
PressãoBalão	Variável de Pressão da Caldeira
PressãoLinha	Variável de Pressão na Linha
PressãoFornalha	Variável de Pressão da Fornalha
VazãoVapor	Variável de Vazão de Vapor

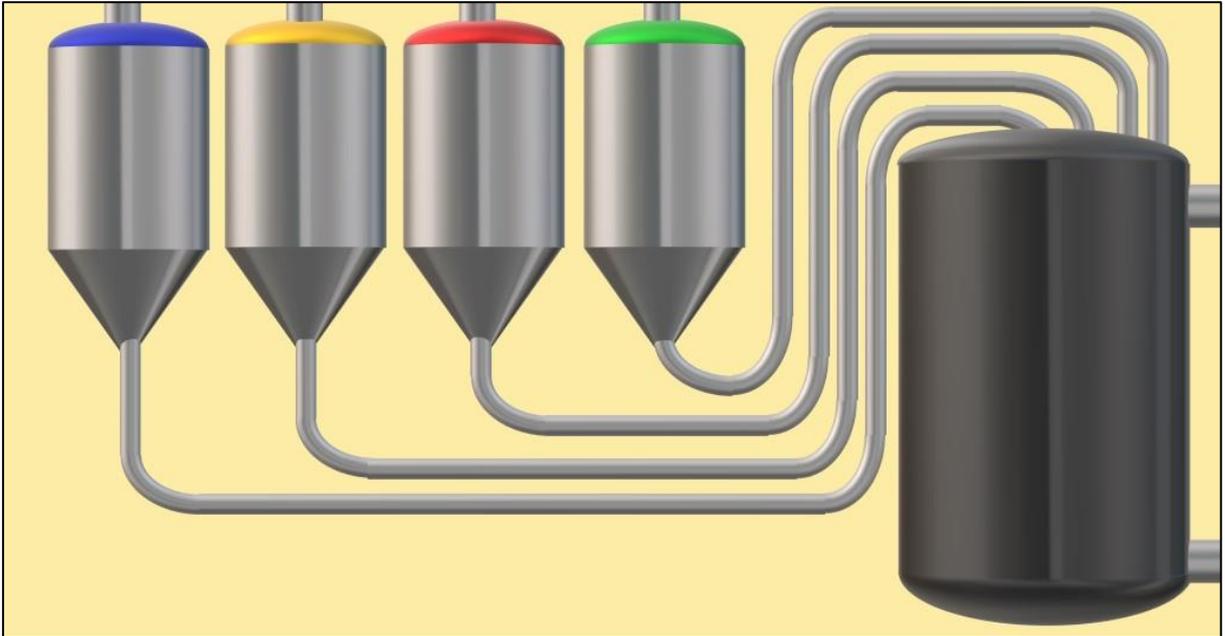
Fonte: Autoria própria

4.4 Elaboração da tela do supervisor (SCADABR), identificação e criação dos *Data Points*.

Como já descrito o processo de Caldeira anteriormente, utilizada no supervisor do Eclipse SCADA, a outra tela desenvolvida foi a de um processo de Mistura. Esta é definida no OPC Server SCADABR.

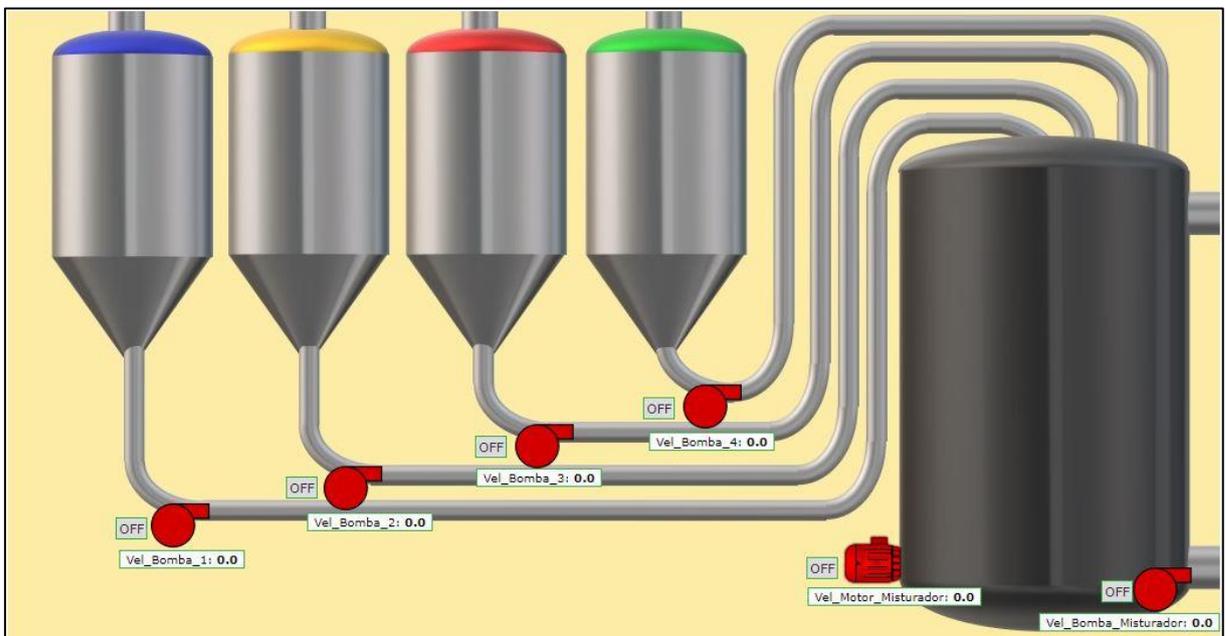
Na figura 13, o processo de Mistura é ilustrado por quatro tanques submetidos ao tanque Misturador, sendo eles operados pelas ações das bombas dos tanques e dos motores alimentadores e misturadores, de modo a se ter o controle de acionamento e velocidade de rotação. Na figura 14, foram evidenciados os botões de *on/off* e a indicação de velocidade dos respectivos motores e bombas. Foram definidos o *data source MODBUS IP* utilizado para conexão *TCP* (Figura 15) e os *data points* para a realização da comunicação com o equipamento, visto na Tabela 4.

Figura 13 – Desenho da tela do sistema supervisorio implementado no SCADABR, referente a um processo de Mistura



Fonte: Autoria própria

Figura 14 – Identificação dos botões e indicação de velocidades dos motores e bombas no processo de Mistura



Fonte: Autoria própria

Figura 15 – Criação do *data source* para comunicação *MODBUS*

Propriedades do modbus IP

Nome: Supervisório Mistura

Export ID (XID): DS_182579

Período de atualização: 1 segundo(s)

Quantificação:

Timeout (ms): 500

Tentativas: 2

Apenas quantidades contínuas:

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: 192.168.0.30

Porta: 503

Encapsulado:

Níveis de alarme de eventos

Exceção de data source: Urgente

Exceção de leitura de data point: Urgente

Exceção de escrita em data point: Urgente

Fonte: Autoria própria

Tabela 4 – *Data points* para ações nos motores e bombas

Data Points	Descrição
Bomba_1	Bomba do tanque 1
Bomba_2	Bomba do tanque 2
Bomba_3	Bomba do tanque 3
Bomba_4	Bomba do tanque 4
Bomba_Misturador	Bomba do tanque Misturador
Motor_Misturador	Motor do tanque Misturador
Vel_Bomba_1	Velocidade de rotação da Bomba do tanque 1
Vel_Bomba_2	Velocidade de rotação da Bomba do tanque 2
Vel_Bomba_3	Velocidade de rotação da Bomba do tanque 3
Vel_Bomba_4	Velocidade de rotação da Bomba do tanque 4
Vel_Bomba_Misturador	Velocidade de rotação da Bomba do Misturador
Vel_Motor_Misturador	Velocidade de rotação do Motor do Misturador

Fonte: Autoria própria

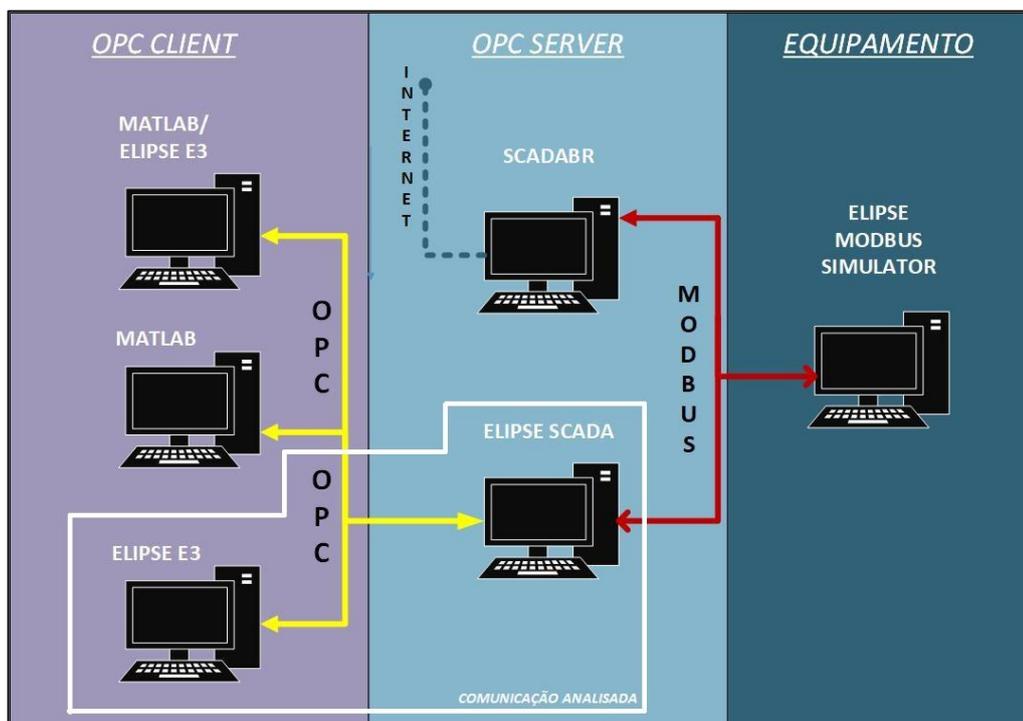
5 RESULTADOS E DISCUSSÕES

Neste capítulo serão apresentados resultados obtidos com a implementação da rede industrial, a fim de comprovar a execução da proposta de trabalho. Esses resultados são referentes ao protocolo de comunicação *OPC*, com comunicações entre Cliente e Servidor, e a utilização da *toolbox OPC*, e bem como o protocolo de comunicação *MODBUS*. As instruções para realização dessas comunicações são exemplificadas nos Apêndices C, D e E.

5.1 Comunicação entre *OPC SERVER* (Elipse SCADA) e *OPC CLIENT* (Elipse E3)

Utilizando o software Elipse SCADA, foi inserida a imagem criada para tela do supervisor, identificadas e criadas as *tags*, tendo assim aquisição dos parâmetros. O Elipse SCADA funciona como um *OPC Server*, onde os dados são enviados utilizando o protocolo *OPC*, para a ferramenta Elipse E3, que terá a função *OPC Client*, de modo a monitorar esses dados obtidos, visto na Figura 16.

Figura 16 – Esquemático da comunicação analisada entre *OPC Server* (Elipse SCADA) e *OPC Client* (Elipse E3)

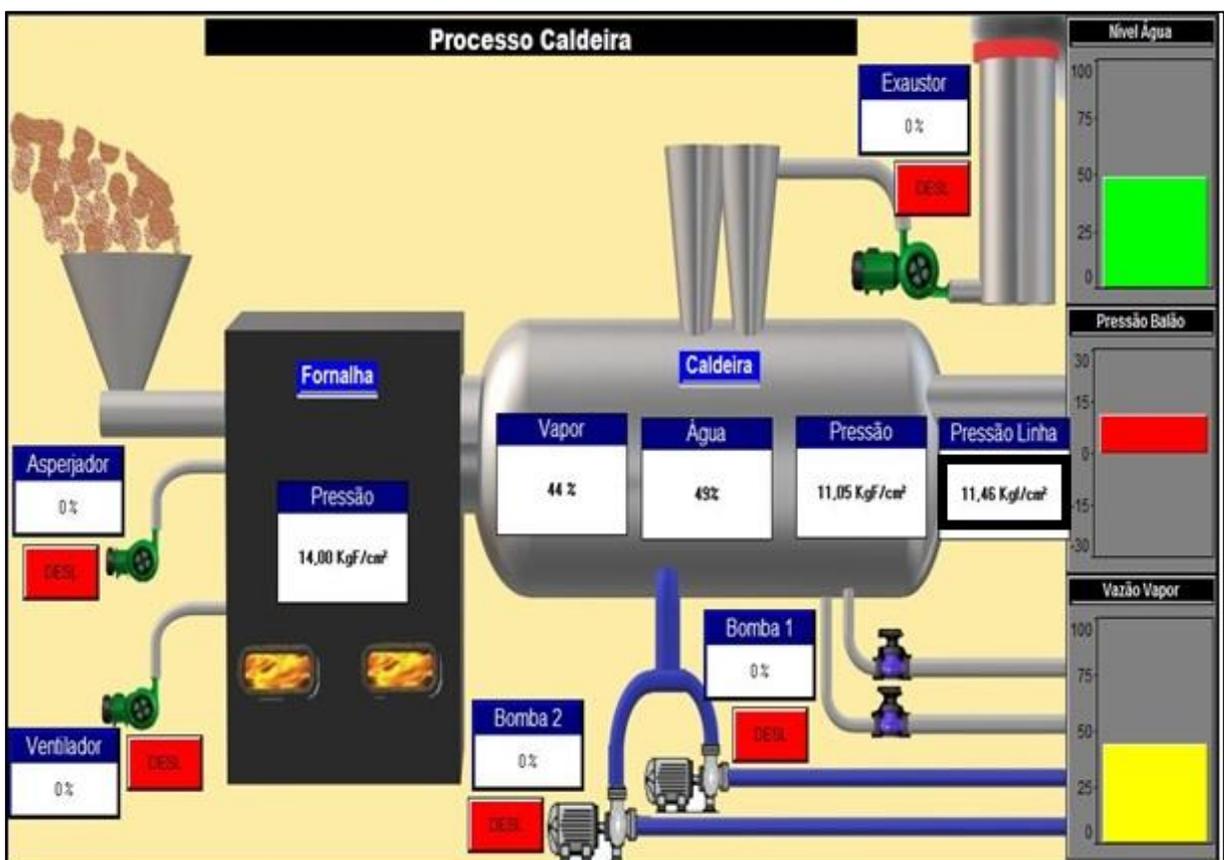


Fonte: Autoria própria

O supervisor do processo, Elipse SCADA, ilustra a geração de valores aleatórios das variáveis vazão, nível e pressão, bem como apresentados os valores em formato digital e em gráficos. Para inicialização da comunicação OPC entre OPC Server e OPC Client, são feitas as importações das tags para o Elipse E3.

Com a execução da aplicação pode se observar que a cada variação do valor da variável no supervisor, é atualizada no monitoramento do OPC Client. Na Figura 17 verificamos os valores gerados pelas variáveis no Elipse SCADA, e na Figura 18, o monitoramento dessas variáveis no Elipse E3. Analisando a região demarcada da Pressão de Linha de 11,46 Kgf/cm² observamos o seu correto monitoramento.

Figura 17 – Primeira visualização dos valores gerados das variáveis no OPC Server (Elipse SCADA)



Fonte: Autoria própria

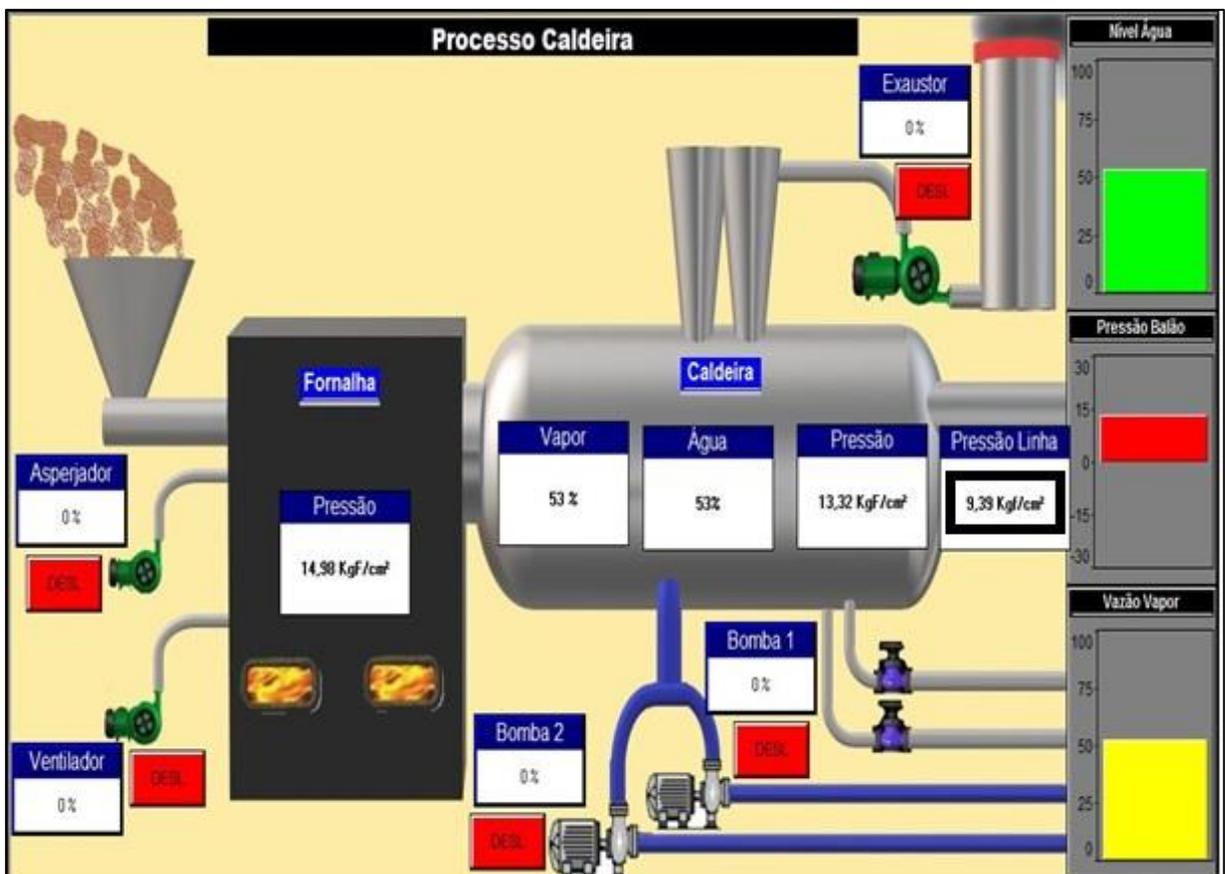
Figura 18 – Primeiro monitoramento dos valores das variáveis recebidos no *OPC Client* (Eclipse E3)

Nome	ID do Item	Var...	Valor	Qualid...	Estampa de tempo	Valor (sem escala)
DriverOPC3						
GrupoOPC1		1000				
Tags						
Motores_e_Bombas						
Variaveis						
NivelÁgua	Tags.Variaveis.NivelÁgua	g	48,9939346516312	192	24/05/2019 17:21:26,324	g 48,9939346516312
PressãoBalão	Tags.Variaveis.PressãoBalão	g	11,0548531730705	192	24/05/2019 17:21:26,324	g 11,0548531730705
PressãoFornalha	Tags.Variaveis.PressãoFornalha	g	13,9977597117527	192	24/05/2019 17:21:26,324	g 13,9977597117527
PressãoLinha	Tags.Variaveis.PressãoLinha	g	11,4648094119083	192	24/05/2019 17:21:26,324	g 11,4648094119083
VazãoVapor	Tags.Variaveis.VazãoVapor	g	44,2158236396374	192	24/05/2019 17:21:26,324	g 44,2158236396374

Fonte: Autoria própria

Do mesmo modo, observamos a troca dos dados das aplicações na Figura 19 e 20, validando a ocorrência da comunicação OPC.

Figura 19 –Segunda visualização dos valores gerados das variáveis no *OPC Server* (Eclipse SCADA)



Fonte: Autoria própria

Figura 20 – Segundo monitoramento dos valores das variáveis recebidos no *OPC Client* (Eclipse E3)

Nome	ID do Item	Var...	Valor	Qualid...	Estampa de tempo	Valor (sem escala)
DriverOPC3						
GrupoOPC1		1000				
Tags						
Motores_e_Bombas						
Variaveis						
NivelÁgua	Tags.Variaveis.NivelÁgua	g	53,373389591052	192	24/05/2019 17:22:53,108	g 53,373389591052
PressãoBalão	Tags.Variaveis.PressãoBalão	g	13,3234044629658	192	24/05/2019 17:22:53,108	g 13,3234044629658
PressãoFornalha	Tags.Variaveis.PressãoFornalha	g	14,9777111532945	192	24/05/2019 17:22:53,108	g 14,9777111532945
PressãoLinha	Tags.Variaveis.PressãoLinha	g	9,39325400140385	192	24/05/2019 17:22:53,108	g 9,39325400140385
VazãoVapor	Tags.Variaveis.VazãoVapor	g	52,6867472511979	192	24/05/2019 17:22:53,108	g 52,6867472511979

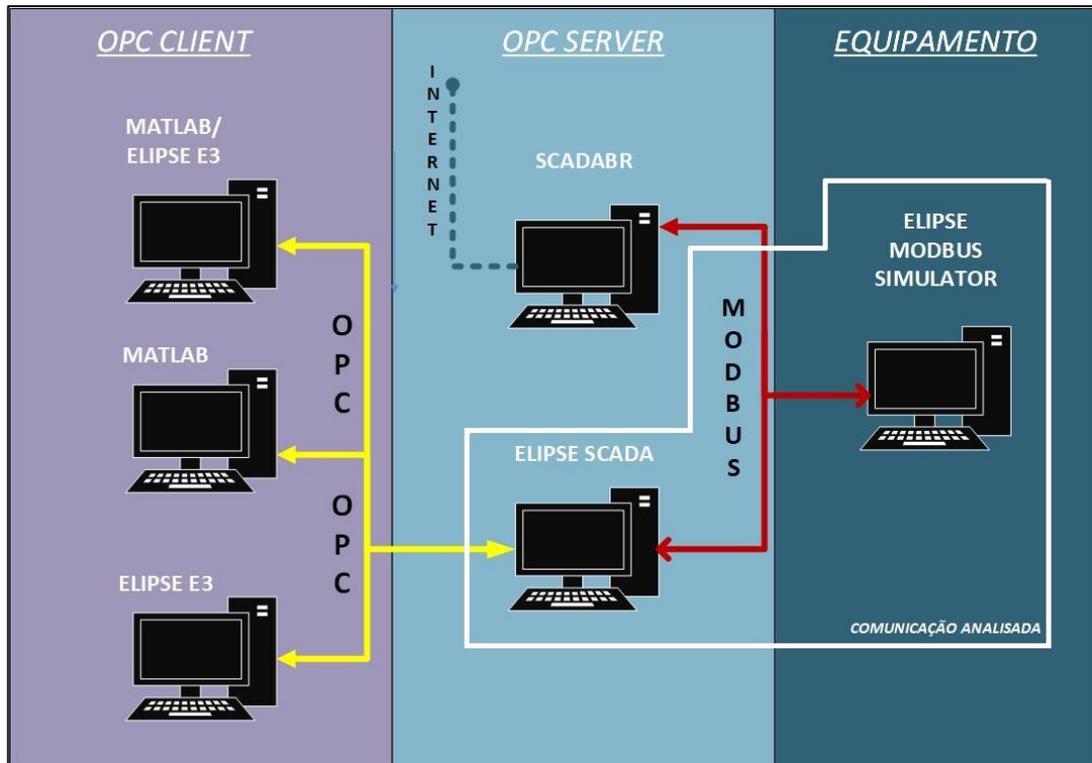
Fonte: Autoria própria

Por meio disso podemos entender que o computador definido como *OPC Server*, é equivalente a tela de um sistema supervisório de um processo fabril, como ocorre na maioria das indústrias de ponta, estando responsável por receber dados de sensores e pode-los endereçar a outro computador. Esse sendo delimitado *OPC Client*, para fins de monitoramento, não precisando necessariamente utilizar o mesmo programa para receber os dados, mas sim apenas o protocolo de comunicação industrial *OPC*.

5.2 Comunicação entre *OPC Server* (Eclipse *SCADA*) e *MODBUS Simulator*

Para simular o envio de dados de equipamentos de chão de fábrica para o *OPC Server*, definido pelo Eclipse *SCADA*, foi utilizado um simulador que opera como um equipamento CLP. Esse é o Eclipse *MODBUS Simulator*, que para tal foi utilizada o protocolo de comunicação industrial *MODBUS*, visto na Figura 21.

Figura 21 - Esquemático da comunicação analisada entre *OPC Server (Elipse SCADA)* e *MODBUS Simulator*

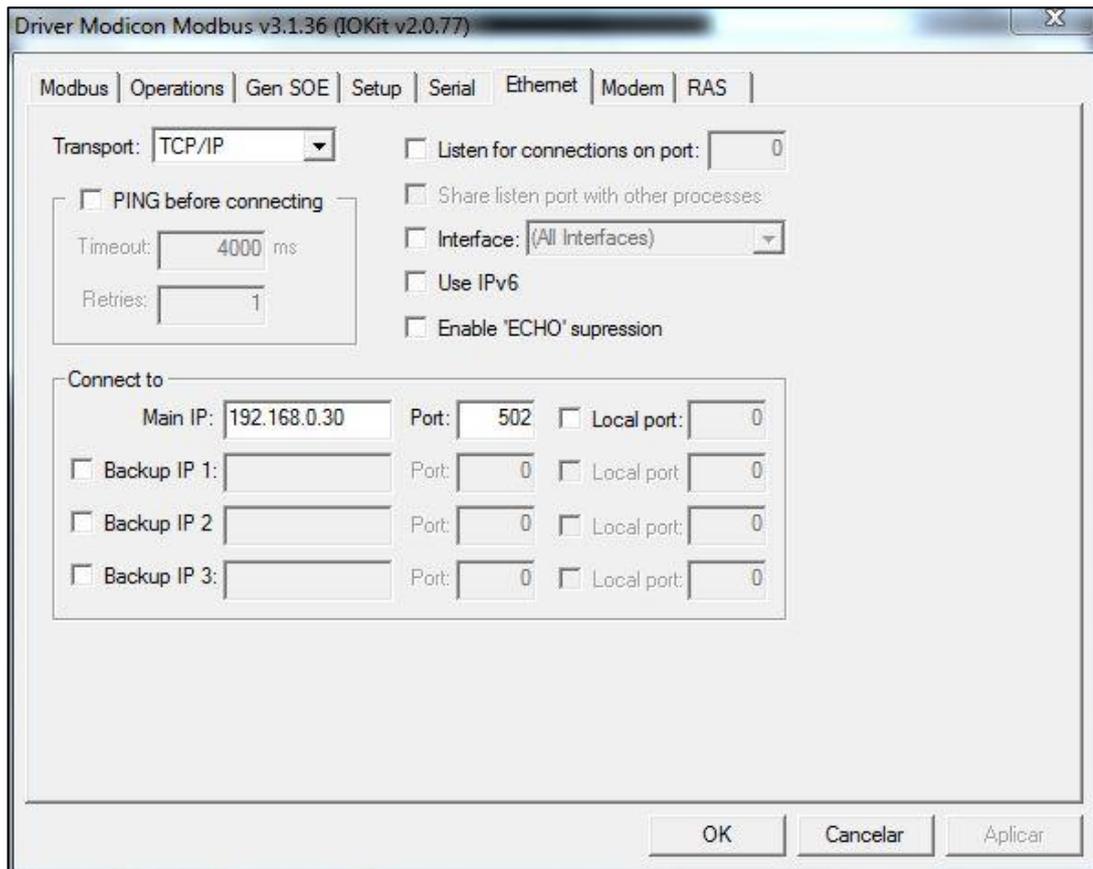


Fonte: Autoria própria.

São realizadas configurações no *OPC server*, para efetuar essa comunicação, tal como a inserção do *driver, modbuss.dll*, que indexados as *tags PLC* reconhecem como um equipamento conectado ao servidor. Essa *driver* é disponibilizado no site do fabricante.

Também na aba do *driver* precisam efetuar definições de comportamento, que nesse laboratório de rede industrial, o simulador se comportará como um equipamento conectado em *Ethernet*, operando com o protocolo *MODBUS RTU*. Para isso foi necessário especificar o *IP* do computador que equivalera a um *CLP*, para comunicação com o *OPC Server* (Figura 22).

Figura 22 - Configuração dos parâmetros de comunicação no MODBUS Simulator

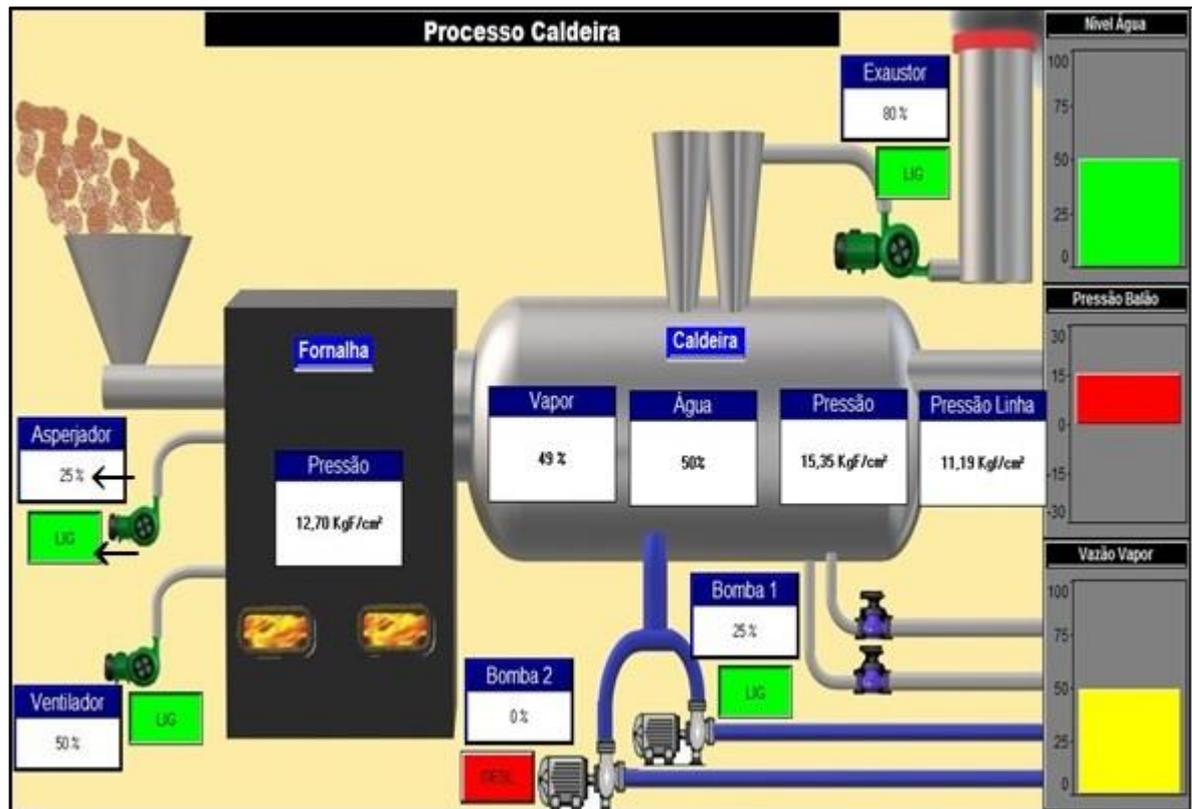


Fonte: Autoria própria.

No simulador foi definido a utilização de um *CLP*, para os motores e bombas. O *CLP* tem a função de acionamento dos seus respectivos elementos (motores e bombas), bem como a definição do *setpoint* da velocidade de rotação. O *CLP* é composto por 5 bobinas e 5 registradores, responsáveis pelas ações dos motores do Asperjador, Ventilador e Exaustor, e para as bombas 1 e 2 (reserva).

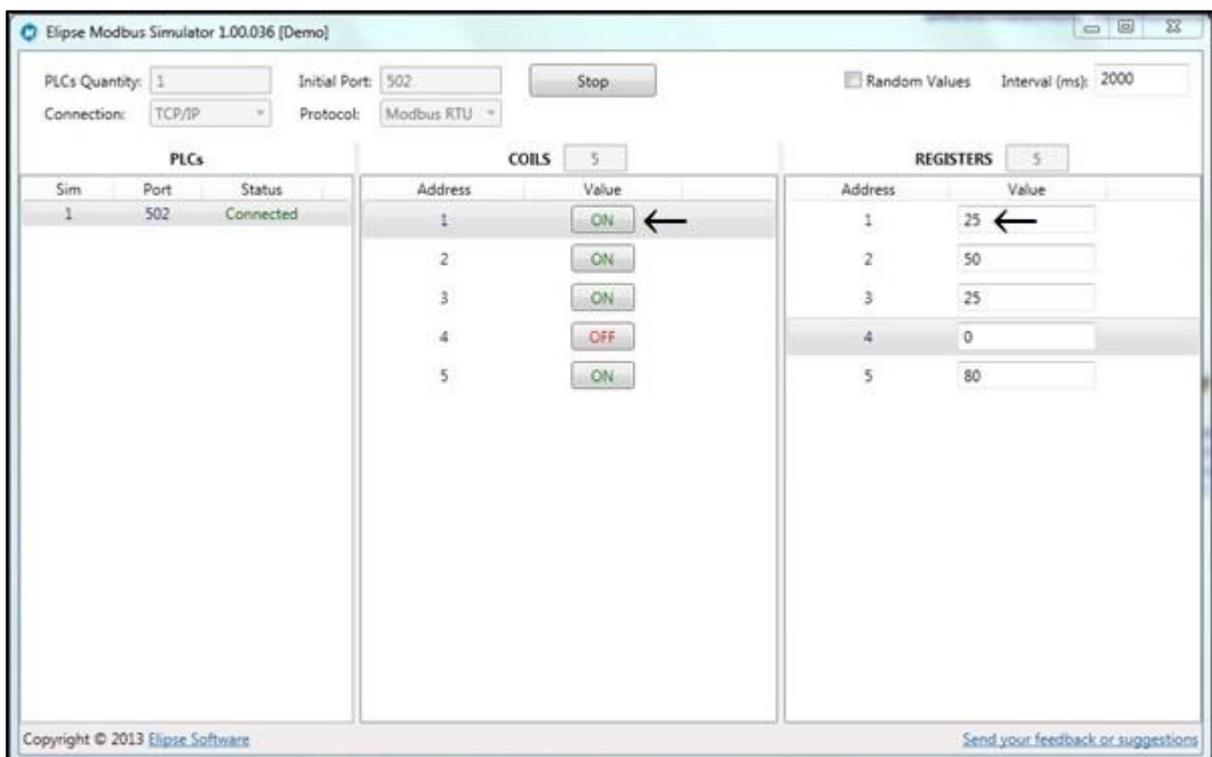
Podemos visualizar na Figura 24 que cada ação feita no simulador, é alterada na tela do sistema supervisório da Figura 23, bem com o comportamento de dados binários e envios de valores inteiros, validando a protocolo de comunicação *MODBUS* entre equipamento e *OPC Server*.

Figura 23 – Acionamento e *setpoint* dos motores e bombas no OPC Server (Elipse SCADA)



Fonte: Autoria própria.

Figura 24 – Simulação do CLP no MODBUS Simulator



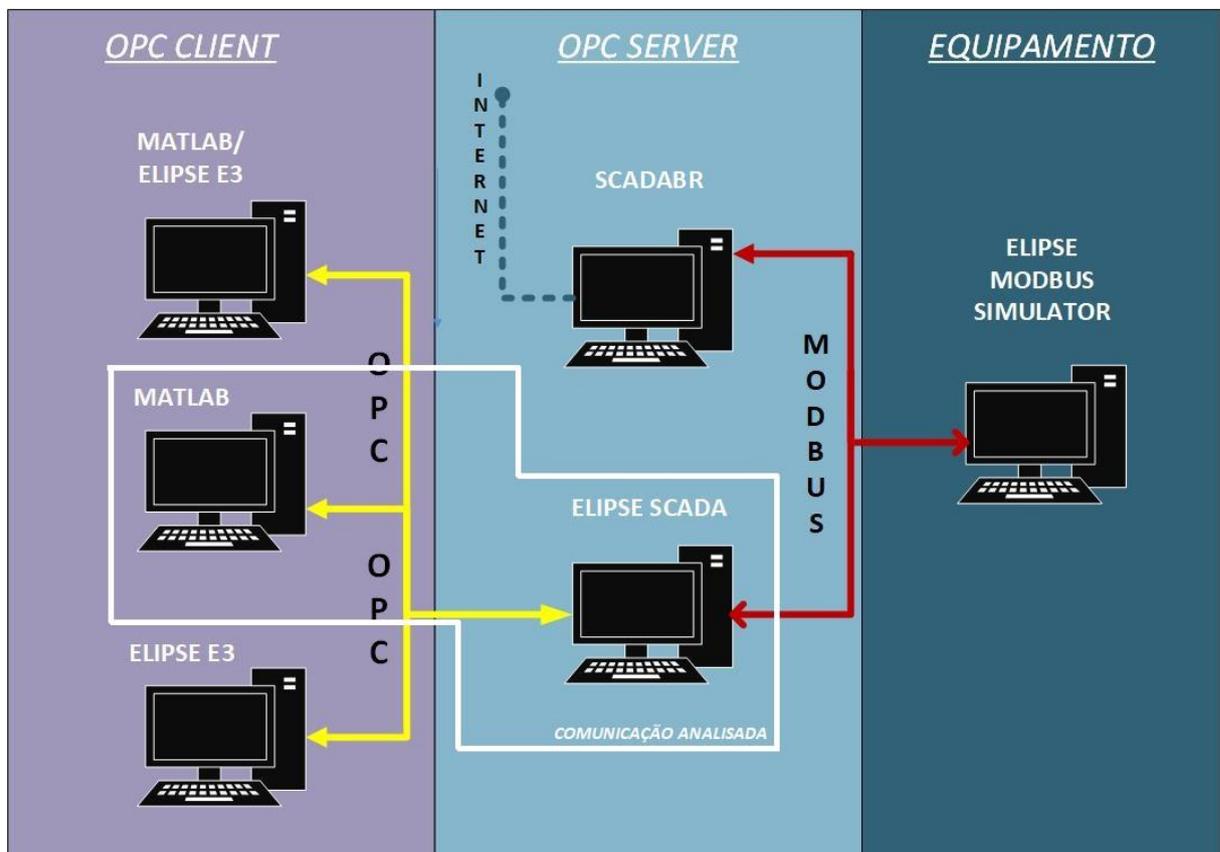
Fonte: Autoria própria.

Com base no experimento dessa comunicação, mesmo simulando um computador como um equipamento, pode se verificar que qualquer componente de chão de fábrica, responsável por controle de dispositivos, é capaz de trocar dados via protocolo *MODBUS* com *OPC Server* desta proposta de laboratório de rede de comunicação industrial, sendo necessário apenas a inserção do *driver* do dispositivo a ser conectado.

5.3 Comunicação entre *OPC Server* (Elipse SCADA) e *OPC Client* (MATLAB)

Utilizando o mesmo software Elipse SCADA, como um *OPC Servidor*, porem evidenciando o envio e o uso dos dados, usando uma ferramenta muito comum na engenharia, o MATLAB. Na comunicação *OPC* ele terá a função de ser mais um *OPC Client* na rede de comunicação industrial (Figura 25).

Figura 25 - Esquemático da comunicação analisada entre *OPC Server* (Elipse SCADA) e *OPC Client* (MATLAB)

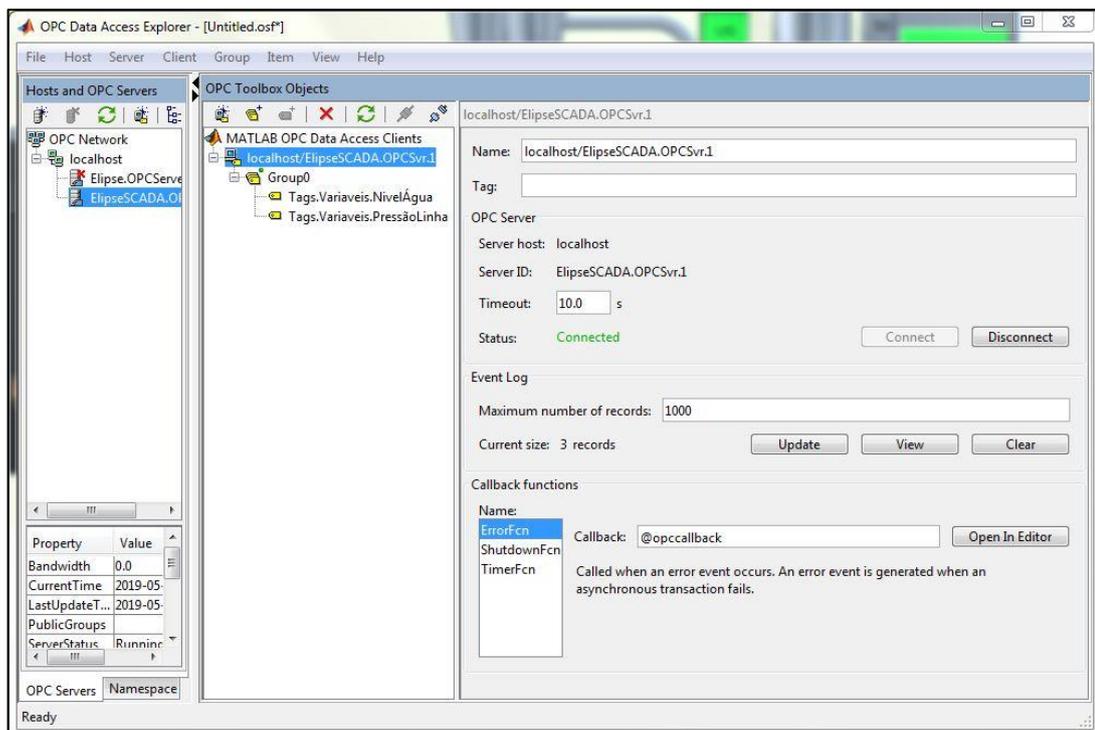


Fonte: Autoria própria.

O MATLAB foi definido nessa rede pela sua grande capacidade de manipulação de dados, de modo a analisar comportamento de variáveis do processo e podendo aplicar técnicas de controle. Nesse laboratório de comunicação industrial o intuito é evidenciar como será feita a inserção dessa ferramenta para efetuar a comunicação *OPC*. Para isso ocorrer é utilizada uma biblioteca no MATLAB chamada *OPCTOOL*, disponíveis nas versões mais completas.

Essa função tem como objetivo conectar ao *OPC Servidor* e assim transferir os dados das variáveis que deseja manipular. Ao chamar a função no MATLAB, foi adicionado o *Host localhost* para listar o *OPC Servidor* da aplicação. Posteriormente são adicionados grupo onde serão movimentadas as *tags* (Figura 26). Para se analisar parâmetros muito importantes no processo de caldeira, foi comunicada com o MATLAB via *OPC* a variável do nível e da pressão da linha, assim podendo obter o seu comportamento em um determinado tempo, afim de gerar alguns estudos.

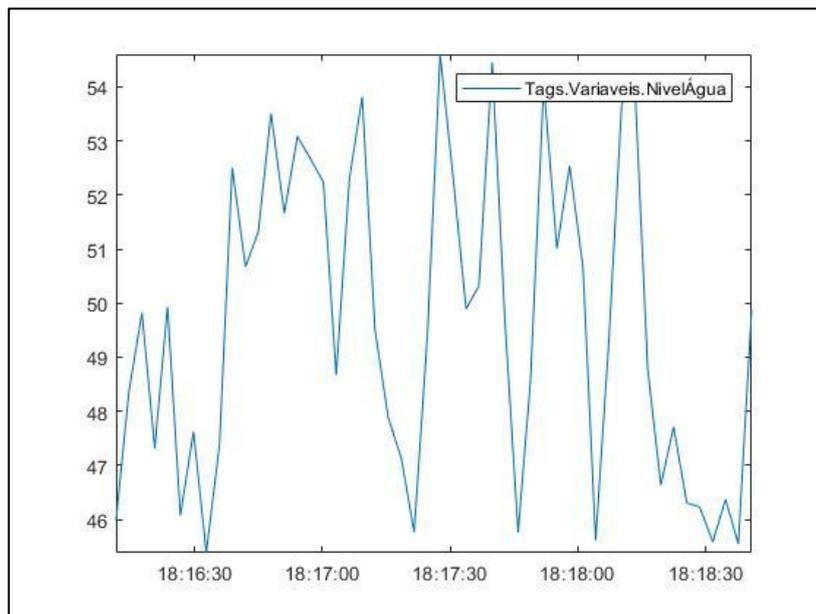
Figura 26 – Conectando ao OPC Servidor para compartilhamento das tags utilizando o OPCTOOL



Fonte: Autoria própria.

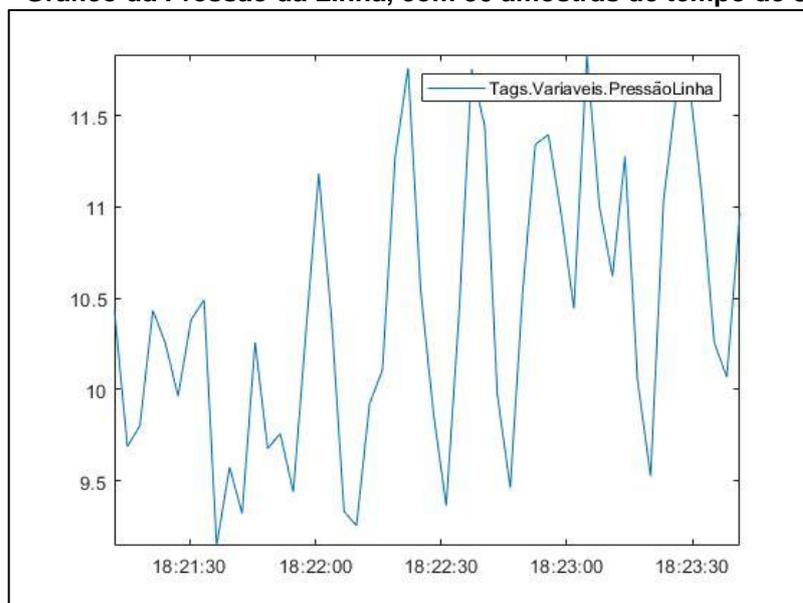
Na Figura 27, foi obtido um gráfico do Nível de água da caldeira, definido por 50 amostras a cada 3 segundos e na Figura 28 mostra o gráfico da Pressão da Linha, dado por 50 amostras a cada 3 segundos. Os gráficos foram gerados através da utilização das *tags demo*, que produzem variações dos parâmetros dentro de limites definidos, afim de evidenciar a ocorrência da comunicação *OPC* com o MATLAB e que dados estão sendo monitorados.

Figura 27 – Gráfico do Nível de Água, com 50 amostras de tempo de 3 segundos



Fonte: Autoria própria.

Figura 28 - Gráfico da Pressão da Linha, com 50 amostras de tempo de 3 segundos



Fonte: Autoria própria.

Enfim desse modo podemos verificar que, com a utilização da biblioteca *OPCTOOL*, temos outra ferramenta com um grande abrangente de funcionalidades, para conecta-la à supervisórios afim de manipular os dados para estudo, assim tornando mais eficaz e seguro o processo a ser aplicado.

O intuito da utilização da ferramenta *MATLAB* era ter acesso aos dados do *OPC Server* em rede, porém ao tentar acessar o *Host* colocando o *IP* fixo do computador com a aplicação *OPC Server*, aparece “Acesso negado”. Foram realizadas as configurações *DCOM* para permissões de acesso das aplicações e liberação de portas no *Firewall* seguindo o tutorial elaborado pela fabricante *MathWorks*, mas não se obteve êxito nas comunicações em rede, sendo apenas em *localhost*, ficando assim uma sugestão para trabalhos futuros.

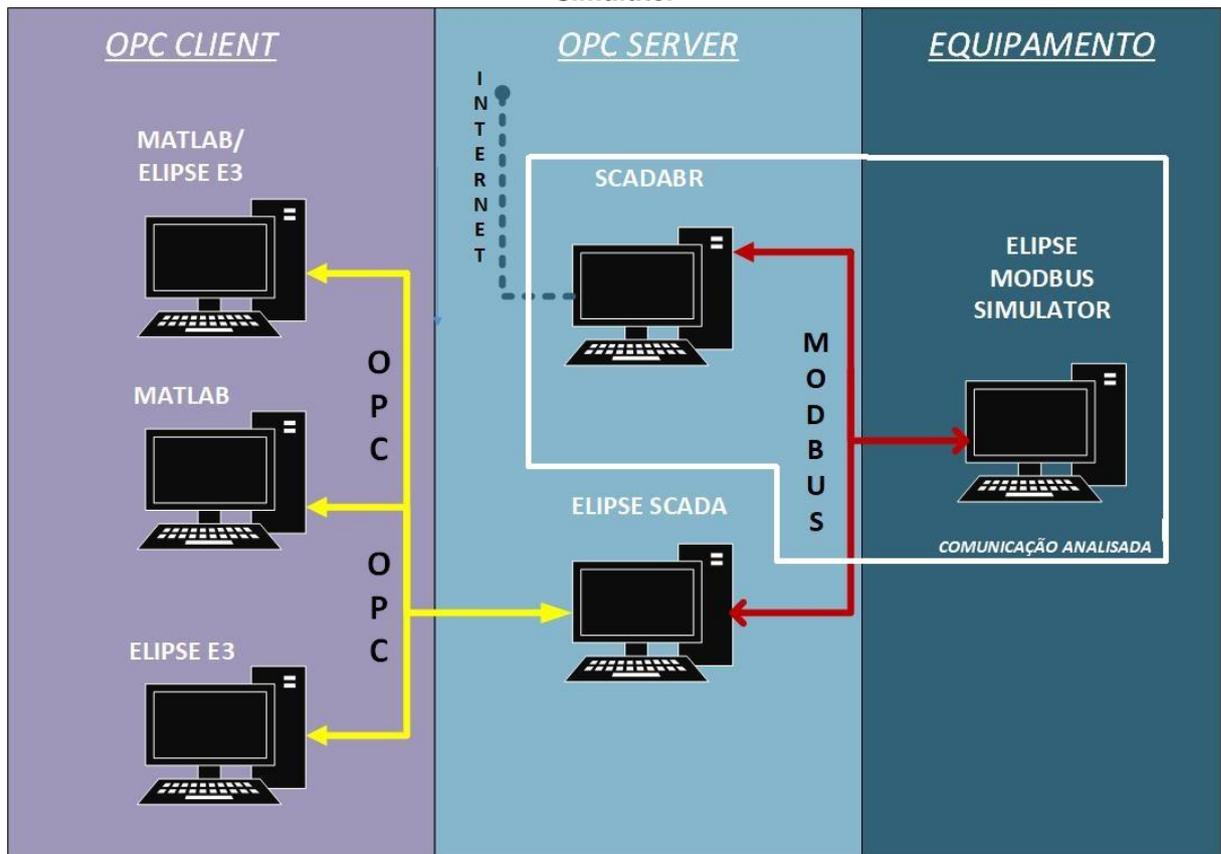
5.4 Comunicação entre *OPC Server* (SCADABR) e *MODBUS SIMULATOR*

Com o mesmo intuito de simular o envio de dados de equipamentos de chão de fábrica para o *OPC Server*, como visto anteriormente, porem agora evidenciando a utilizando do protocolo *MODBUS* com um *OPC Server* com servidor web, o *SCADABR*. Para a simulação dos dados foi aplicado o mesmo simulador, o *Eclipse MODBUS Simulator*, como visto na Figura 29.

Nesse *OPC server*, foi definido no data source o *IP* do computador que simulara um equipamento na rede. O simulador se comportará como um equipamento conectado usando o protocolo *MODBUS TCP*.

No simulador foi definido a utilização de um *CLP*, para os motores e bombas. O *CLP* tem a função de acionamento dos seus respectivos elementos (motores e bombas), bem como a definição do *setpoint* da velocidade de rotação. O simulador é composto por 9 bobinas e 9 registradores, responsáveis pelas ações dos motores do Alimentador e Misturador, e para as bombas dos tanques.

Figura 29 - Esquemático da comunicação analisada entre *OPC Server (SCADABR)* e *MODBUS Simulator*

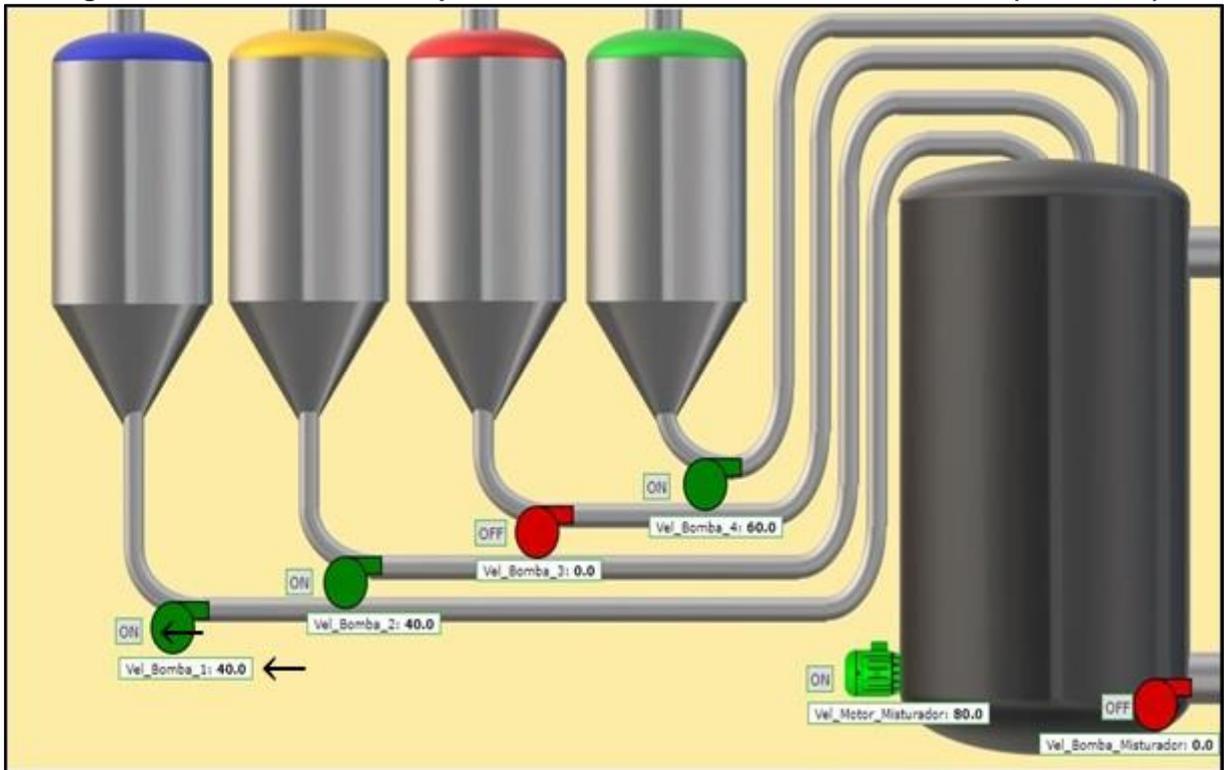


Fonte: Autoria própria.

Podemos visualizar na Figura 31 que cada ação feita no simulador, é alterada na tela do sistema supervisório da Figura 30, bem com o comportamento de dados binários e envios de valores inteiros, validando a protocolo de comunicação *MODBUS* entre equipamento e *OPC Server*.

Com base no experimento dessa comunicação, verificamos as grandes aplicações desse sistema *SCADA*, que trabalha com diferentes tipos de comunicações e tem seu diferencial de ser um servidor web, sendo acessado a partir de um navegador de Internet. Dessa maneira todos os modos de visualização das variáveis, protocolos de comunicação e construção de telas são empregados utilizando o próprio navegador.

Figura 30 - Acionamento e *setpoint* dos motores e bombas no OPC Server (SCADABR)



Fonte: Autoria própria.

Figura 31 – Simulação do CLP para motores e bombas do OPC Server (SCADABR)

Eclipse Modbus Simulator 1.00.036 [Demo]

PLCs Quantity: 1 Initial Port: 503 Stop Random Values Interval (ms): 2000

Connection: TCP/IP Protocol: Modbus TCP

PLCs			COILS 6		REGISTERS 6	
Sim	Port	Status	Address	Value	Address	Value
1	503	Waiting	1	ON ←	1	40 ←
			2	ON	2	40
			3	OFF	3	0
			4	ON	4	60
			5	OFF	5	0
			6	ON	6	80

Copyright © 2013 Elipse Software [Send your feedback or suggestions](#)

Fonte: Autoria própria.

6 CONSIDERAÇÕES FINAIS

Nesse trabalho foi desenvolvido um laboratório de redes industriais. Pode-se observar através dos resultados obtidos, que as utilizações dos softwares de supervisão vigentes nas indústrias tiveram êxito nas comunicações ao utilizar os protocolos comunicação, *OPC* e *MODBUS*.

A criação das telas dos supervisórios possibilitou evidenciar como ocorre a comunicação em um processo industrial real, simulando o envio dos dados de sensores ou *CLP* para o *OPC Server*, e dele endereçados aos *OPC Client* para um monitoramento. Foi analisada a comunicação *MODBUS* através de ações de acionamento e definição de *setpoints* do simulador de *MODBUS* com os sistemas *SCADA*, e a comunicação *OPC*, por meio de visualização do compartilhando dos dados em quase tempo real entre *OPC Server* e *OPC Client*, onde ambas foram validadas atingindo os objetivos propostos.

Durante o desenvolvimento deste trabalho, foi necessário fazer a estruturação, instalação e configuração do sistema operacional nos computadores definidos na rede industrial, de modo a mostrar todo o processo para iniciar a intercomunicação dos protocolos de comunicação industrial.

Contudo, o laboratório de rede de comunicação industrial mostra os conceitos teóricos e práticos, em torno das redes de comunicação, como padrões e modelos de redes, e a aplicação dos protocolos de comunicação industriais integrados com os sistemas de controle e aquisição de dados (*SCADA*).

6.1 Trabalhos Futuros

Como sugestão para trabalhos futuros podemos destacar os seguintes temas:

- Utilização da rede de comunicação industrial com equipamentos para obtenção de dados reais.
- Inserção de outros protocolos de comunicação vigentes nas indústrias.

- Utilização de novos softwares de sistemas *SCADA*, bem como os que usam tecnologias de rede *Wireless*.
- Criação de alarmes, históricos, armazenamento entre outras manipulações das variáveis descritas.
- Comunicação em rede dos sistemas *SCADA* com o *MATLAB*, utilizando a biblioteca *OPC Toolbox*.

REFERÊNCIAS

CASSIOLATO, César. **Redes Industriais**. Disponível em: <<http://www.smar.com/brasil/artigo-tecnico/redes-industriais>>. Acesso em 07 de Out. de 2017.

COSTA, Lucas B; SOUZA, Max M. F. **Automação e Supervisão de uma Planta Didática Industrial Utilizando um Sistema *Open-Source***. 69 f. Trabalho de Conclusão de Curso (Graduação) – Curso de Engenharia de Controle e Automação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio-PR, 2017.

DANTAS, Mario. **Tecnologias de Redes de Comunicação e Computadores**. Rio de Janeiro: Axcel Books do Brasil, 2002.

ELECTRICAL TECHNOLOGY. **An Overview of Industrial Communication Systems & Networks**. 2016. Disponível em: <<https://www.electricaltechnology.org/2016/12/industrial-communication-networks-systems.html>>. Acesso em 13 de Out de 2017.

FONSECA, Marcos de O. **Comunicação OPC – Uma abordagem prática**. Vitoria, 2002.

FOROUZAN, Behrouz A. **Comunicação de Dados e Redes de Computadores**. 4. ed. Porto Alegre: AMGH Lta, 2007.

FU, Chenchen; NI, Zurong. **The Application of embedded System in Supervisory Control and Data Acquisition System (SCADA) Over Wireless Sensor and GPRS Networks, IEEE**. Xiamen, China, p.1-2, 2015.

GALLOWAY, Brendan; HANCKE, Gerhard P. Introduction to industrial control networks. **Communications Surveys & Tutorials, IEEE**. New Jersey, v. 15, n.2, p. 860-880, 2013.

GOMES, J. R. R. **Automação Industrial com recurso a ferramentas Open Source**. Tese (Doutorado) — Universidade do Porto, 2014.

INSTITUTO NEWTON C. BRAGA. **Como funciona o protocolo Modbus (R0001)**. 2014. Disponível em: < <http://www.newtoncbraga.com.br/index.php/como-funciona/12089-como-funciona-o-protocolo-modbus-r0001>>. Acesso em 12 de Out de 2017.

LUGLI, Alexandre B.; SANTOS, Max M. D; FRANCO, Lucia R. H. R. **Uma visão dos protocolos para redes Ethernet industriais e suas aplicações**. p.1-19, 2008.

LUGLI, Alexandre B.; SANTOS, Max M. D. **Redes industriais: evolução, motivação e funcionamento**. Intech América do Sul. 2005.

MATHWORKS. **OPC Toolbox**, 2017. Disponível em :<<https://www.mathworks.com/products/opc.html>>. Acesso em 15 de Out de 2017.

NATIONAL INSTRUMENTS. **O protocolo Modbus em detalhes**. 2014. Disponível em: < <http://www.ni.com/white-paper/52134/pt/>> Acesso em 12 de Out de 2017.

NOGUEIRA, Márcio L. M. **Apostila de fundamentos e práticas em rede de computadores**. Versão 1.0. 2009 p15-30.

PEREIRA, João M. B. **Desenvolvimento de Sistema SCADA Intranet para a operação de um Laser de Elétrons Livres**. 2015. 95 f. Dissertação de Mestrado Profissional em Física com ênfase em Instrumentação Científica - Centro Brasileiro de Pesquisas Físicas CBPF, Rio de Janeiro, 2015.

SCHNEIDER ELECTRIC. **Redes de Comunicação Industrial**. 2007. Disponível em:<https://www.schneider-electric.pt/documents/product-services/training/doctecnico_redes.pdf>. Acesso em 13 de Out de 2017.

OPC FOUNDATION. **WHAT IS OPC?** Disponível em: <<https://opcfoundation.org/about/what-is-opc/>>. Acesso em 07 de Out. de 2017.

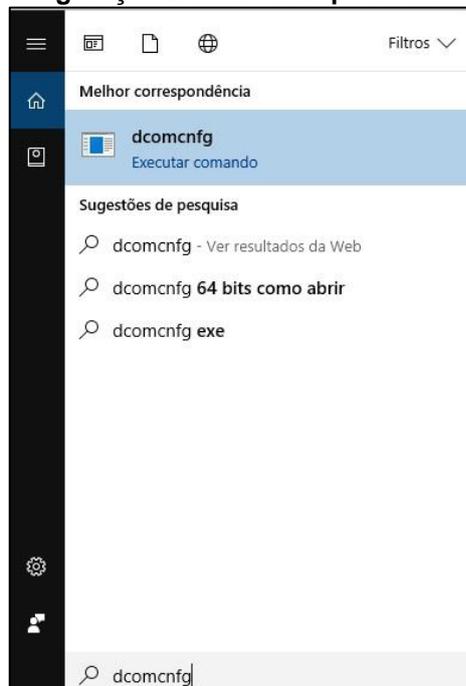
APÊNDICE A – TUTORIAL PARA CONFIGURAÇÕES *DCOM* E *FIREWALL*

No presente apêndice será apresentado um tutorial de configuração necessária para utilização dos *softwares* Elipse, como as aplicações Elipse *SCADA* e *E3*. Essa configuração tem o intuito de obter um bom funcionamento das comunicações com o sistema operacional *Microsoft Windows*. O tutorial baseia nas configurações de *DCOM* e *Firewall*, dando permissões as aplicações.

Para realizar o tutorial, as configurações foram realizadas na plataforma *Windows* 10, sendo equivalente nas versões *Windows*7/8.

Para realizar a configuração *DCOM*, o primeiro passo é acessar a *DCOM* do *Microsoft Windows*, digitando “*dcomcnfg*” no campo de busca, como na figura abaixo.

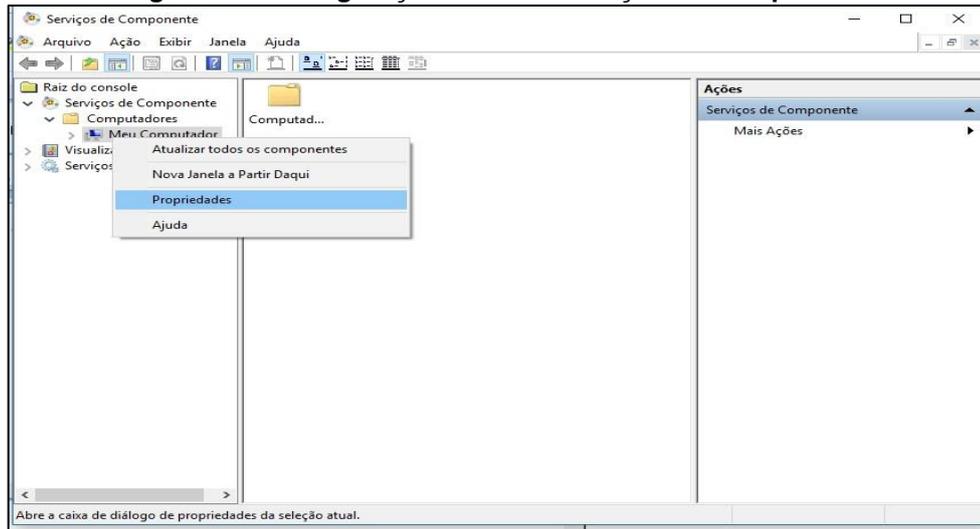
Figura 32 – Configuração *DCOM* - Campo de busca “*dcomcnfg*”



Fonte: Autoria própria.

Abrirá a janela abaixo, de Serviços de Componente, assim acessar Serviços de Componente > Computadores > botão direito em Meu Computador > Propriedades.

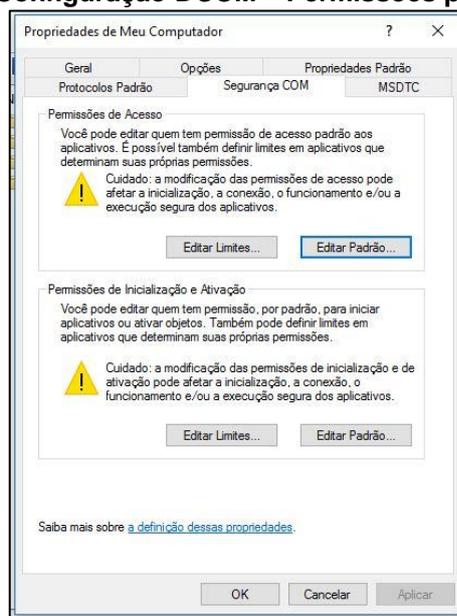
Figura 33 - Configuração DCOM - Serviços de Componente



Fonte: Autoria própria.

Em Propriedades de Meu Computador, clicamos em segurança COM, onde nessa tela iremos dar permissões aos usuários.

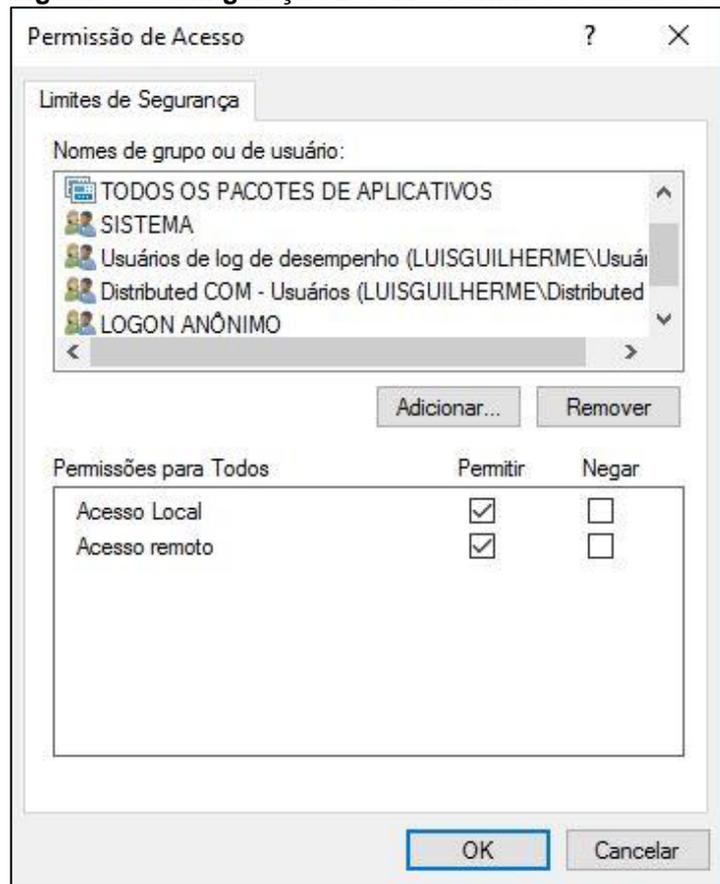
Figura 34 - Configuração DCOM – Permissões para o usuário



Fonte: Autoria própria.

Em Permissões de Acesso > Editar Limites. Na tela que aparecerá, configure os seguintes usuários, TODOS, SISTEMA E LOGON ANONIMO. Caso algum desses usuários não estejam, devemos adicionar.

Figura 35 - Configuração DCOM – Permissões de acesso

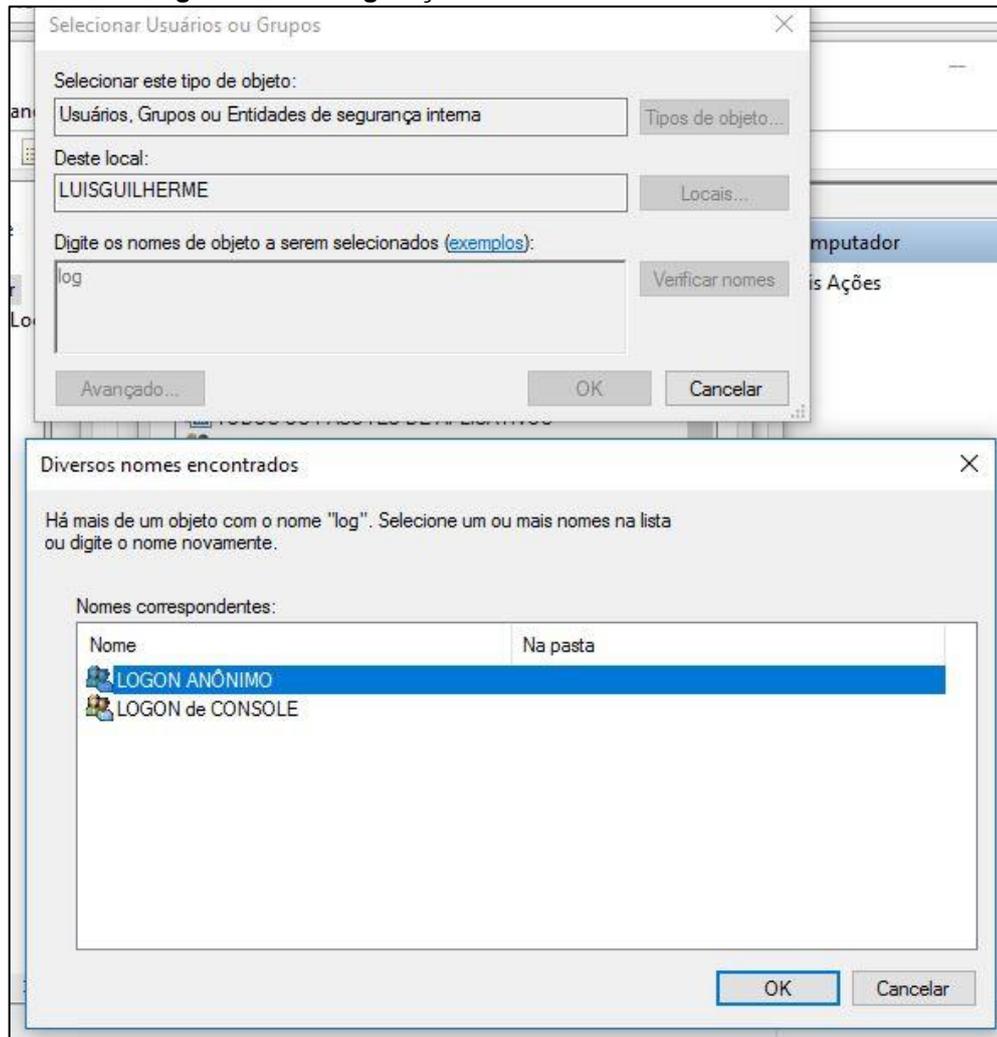


Fonte: Autoria própria.

Para adicionar usuários, clicamos em Adicionar > digitamos o nome do usuário para pesquisa > verificar nomes > e selecionamos o usuário, como no exemplo abaixo para adicionar o LOGON ANONIMO. Esse método é utilizado da mesma forma para adicionar os outros usuários.

Com todos os usuários adicionados, damos Permissões de Acesso Local e Acesso Remoto.

Figura 36 - Configuração DCOM – Adicionando usuários

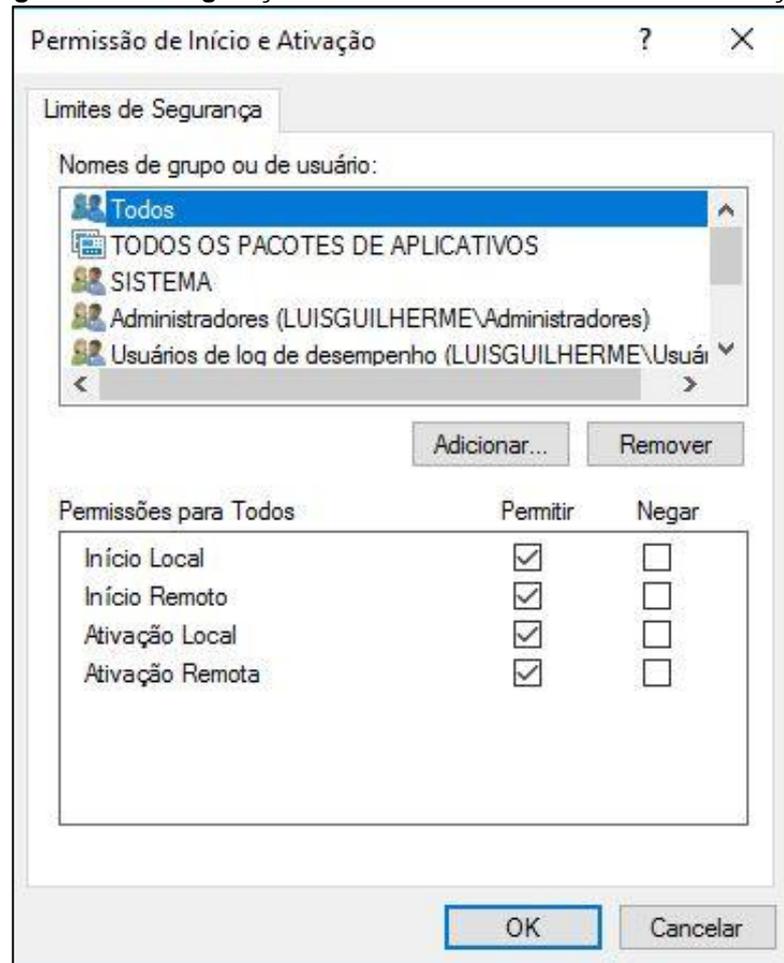


Fonte: Autoria própria.

Posteriormente fazemos as mesmas configurações para as Permissões de Início e Ativação > Editar Limites, onde deve ser adicionado os usuários TODOS, SISTEMA e LOGON ANONIMO, como descrito anteriormente.

Para os usuários atribuímos permissões de Início Local, Início Remoto, Ativação Local e Ativação Remota.

Figura 37 - Configuração DCOM – Permissão de Início e Ativação

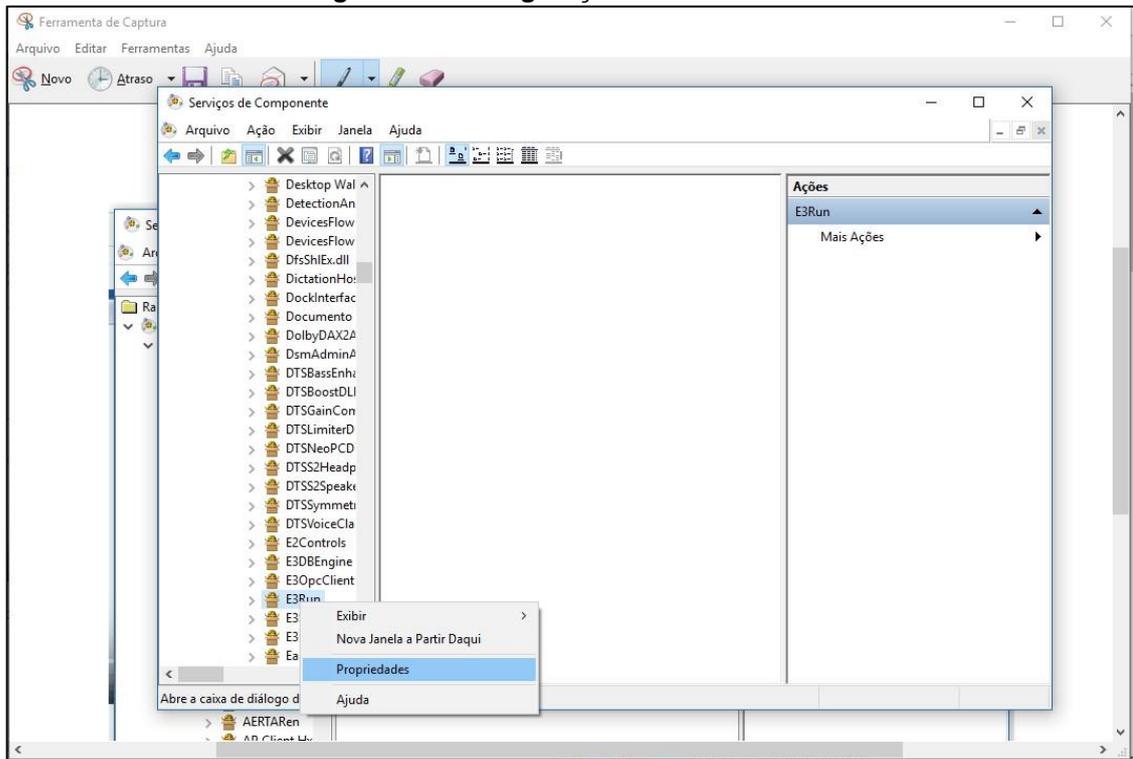


Fonte: Autoria própria.

Após o procedimento para adicionar e dar permissões aos usuários, é necessário realizar as configurações de DCOM e permissões para as aplicações E3 Server, E3 Run, OpcEnum e Elipse SCADA.

Acessamos em Serviços de Componente > Computadores > Meu Computador > *Config DCOM* > *E3 Run* > botão direito em Propriedades.

Figura 38 - Configuração DCOM - E3 Run

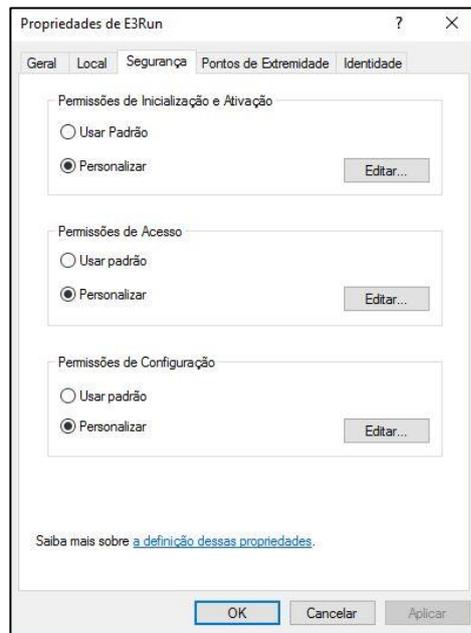


Fonte: Autoria própria.

Ao abrir a tela abaixo, clicamos em Segurança > Permissões de Inicialização e Ativação > Editar e adicionamos os usuários TODOS, SISTEMA e LOGON ANONIMO e atribuímos as permissões.

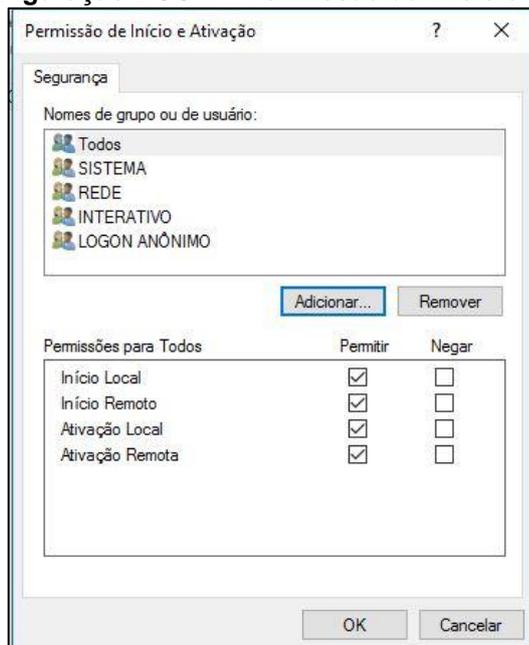
Como nos passos realizados anteriormente executamos o mesmo procedimento para, Permissão de Acesso > Editar.

Figura 39 - Configuração DCOM – Propriedades de E3 Run



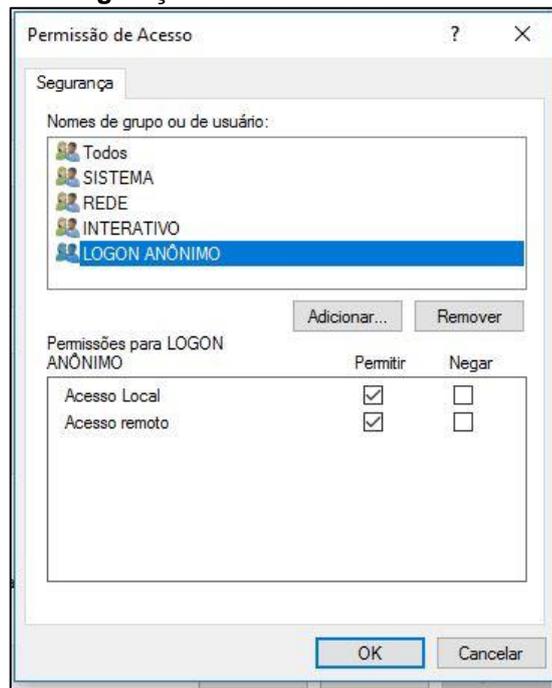
Fonte: Autoria própria.

Figura 40 - Configuração DCOM – Permissão de Início e Ativação E3 Run



Fonte: Autoria própria.

Figura 41 - Configuração DCOM- Permissão de Acesso E3 Run

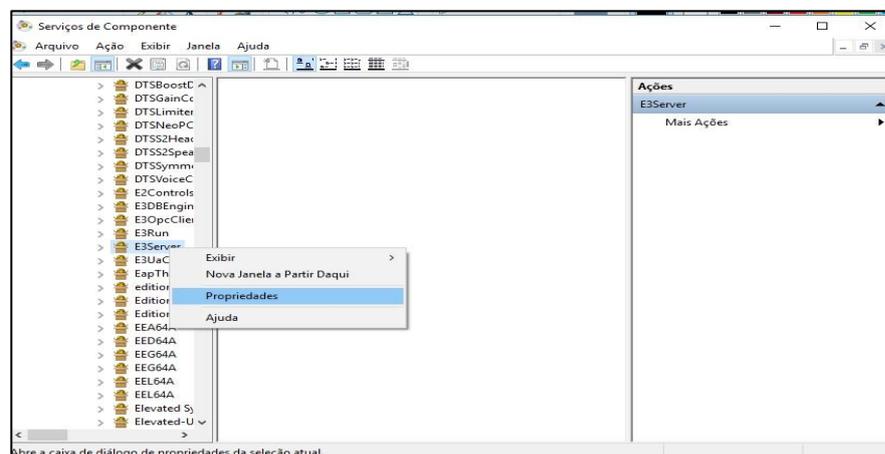


Fonte: Autoria própria.

Repetimos o processo descrito acima para o E3 Server, OpcEnum e Elipse SCADA.

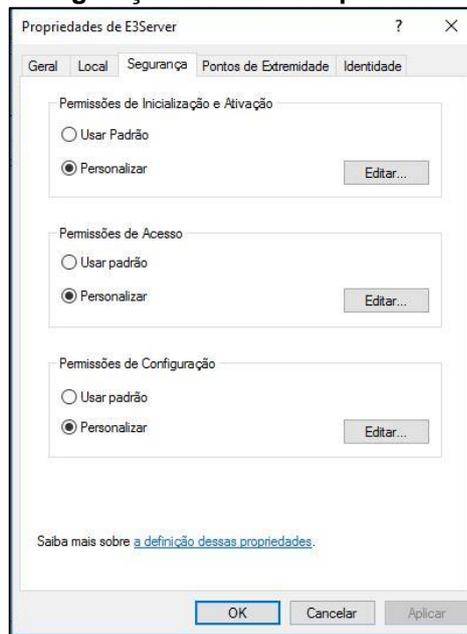
Serviços de Componente > Computadores > Meu Computador > Config DCOM > E3 Server > botão direito em Propriedades. Adicionar os usuários e dar permissões como já descrito.

Figura 42 - Configuração DCOM – E3 Server



Fonte: Autoria própria.

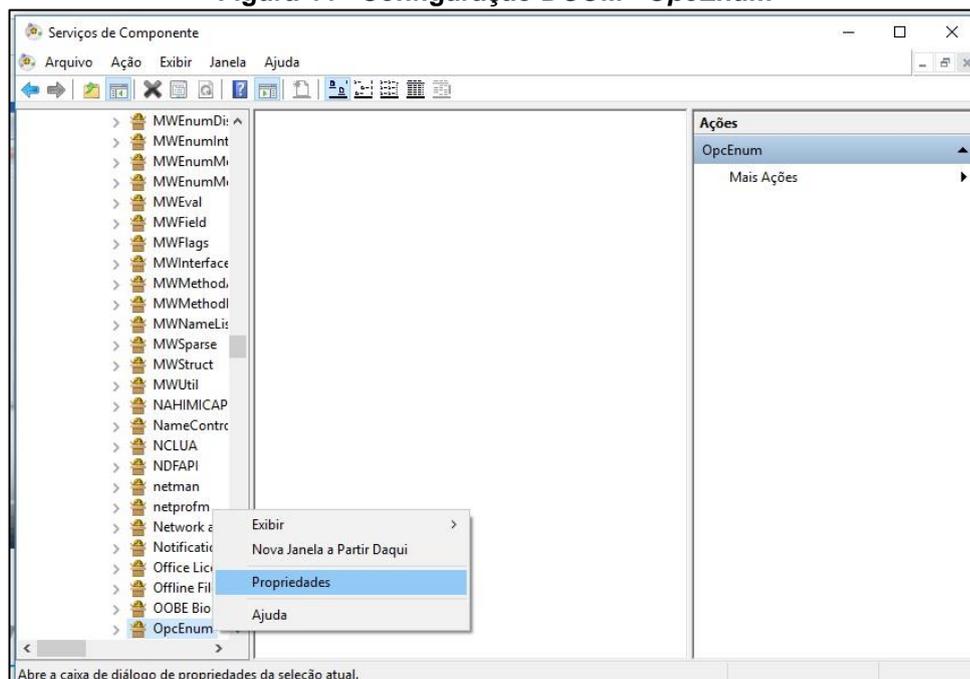
Figura 43- Configuração DCOM – Propriedade de E3 Server



Fonte: Autoria própria.

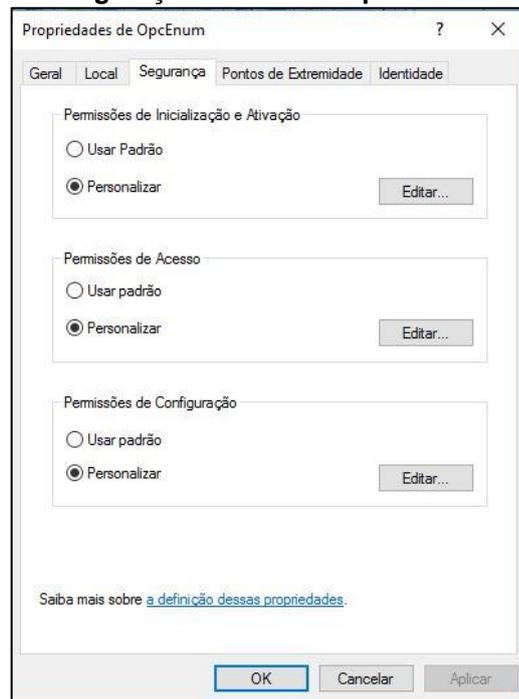
Serviços de Componente > Computadores > Meu Computador > Config DCOM > OpcEnum > botão direito em Propriedades. Adicionar os usuários e dar permissões como já descrito.

Figura 44 - Configuração DCOM - OpcEnum



Fonte: Autoria própria.

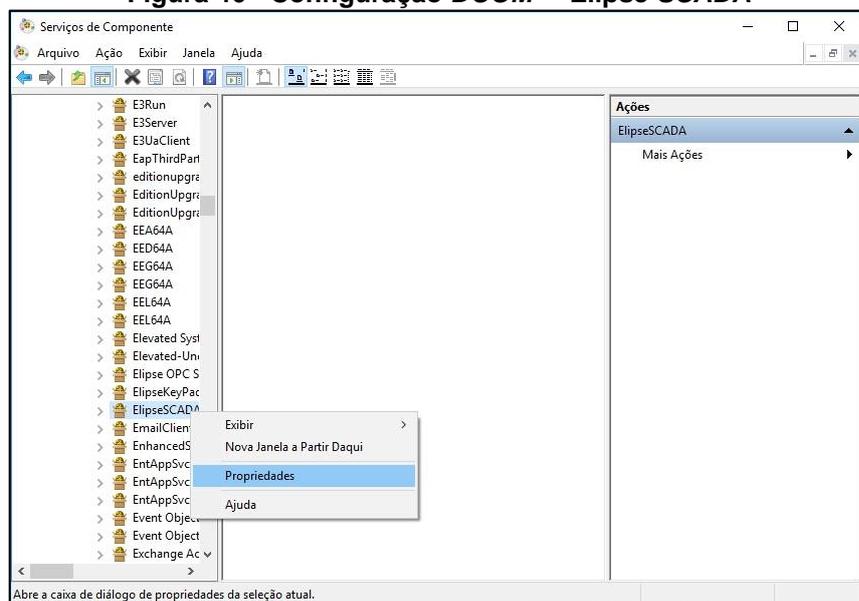
Figura 45 - Configuração DCOM – Propriedades de OpcEnum



Fonte: Autoria própria.

Serviços de Componente > Computadores > Meu Computador > *Config DCOM* > Elipse SCADA > botão direito em Propriedades. Adicionar os usuários e dar permissões já descrito.

Figura 46 - Configuração DCOM – Elipse SCADA



Fonte: Autoria própria.

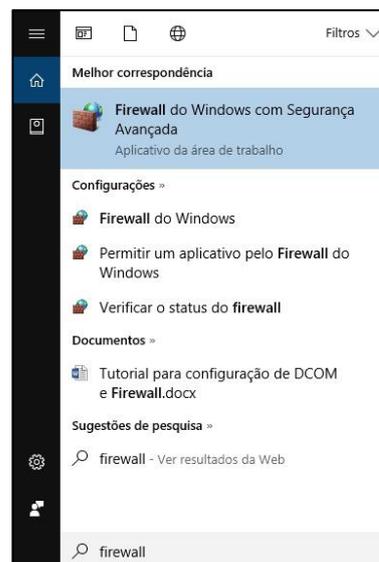
Figura 47 - Configuração DCOM – Propriedades de Elipse SCADA



Fonte: Autoria própria.

Ao realizarmos os passos acima, agora vamos efetuar a configuração de *Firewall* do *Microsoft Windows* para as aplicações. Precisamos acessar a *Segurança Avançada* do *Windows*, então no campo de busca, digitar *Firewall* > *Firewall* do *Windows* com *Segurança Avançada*

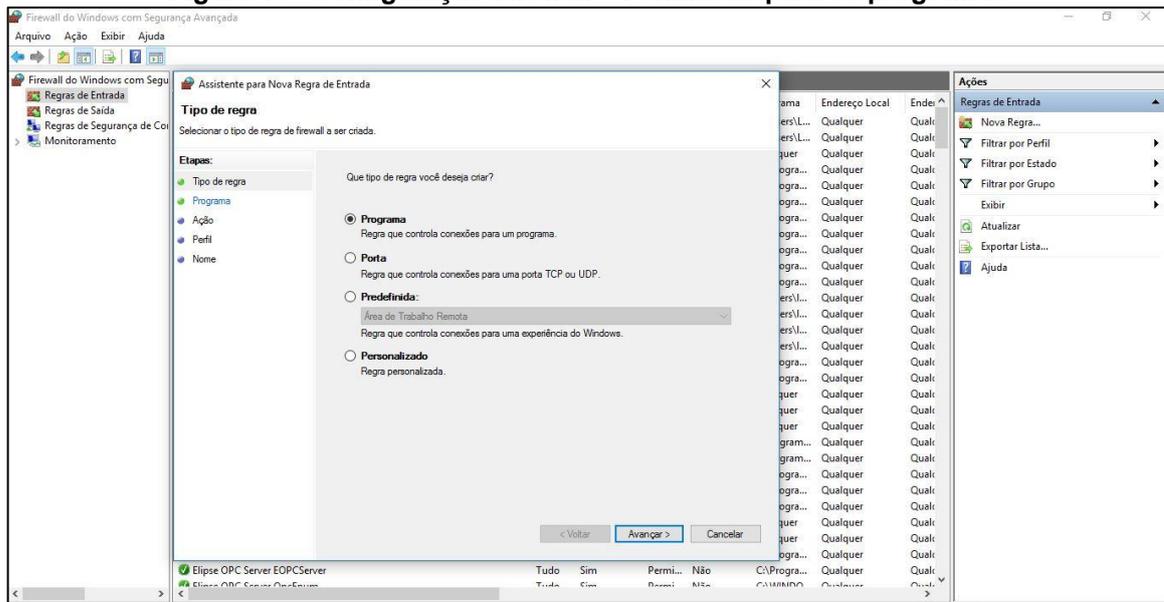
Figura 48 - Configuração Firewall – Campo de busca Firewall do Windows com Segurança Avançada



Fonte: Autoria própria.

Ao acessar aparecera a janela abaixo, onde definimos a permissão para os programas E3 Run, E3 Server, E3 Studio e o Elipse SCADA. Ir em Regras de Entrada > Nova Regra > Programa > Avançar.

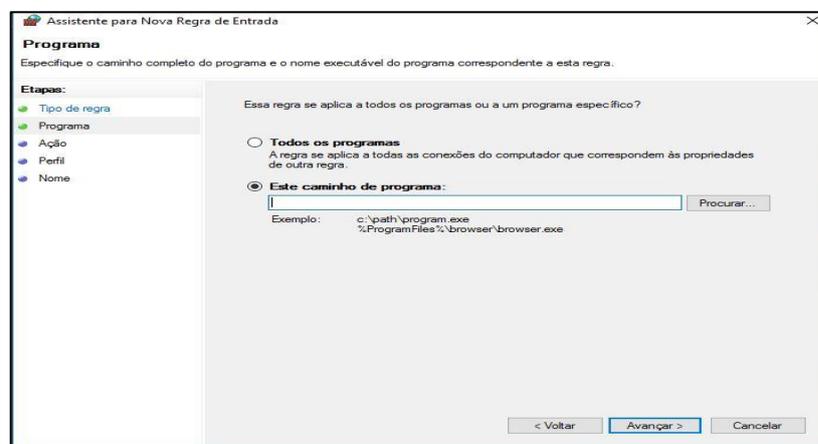
Figura 49 - Configuração Firewall – Permissão para os programas



Fonte: Autoria própria.

Na tela abaixo adicionamos o diretório dos arquivos que queremos adicionar.

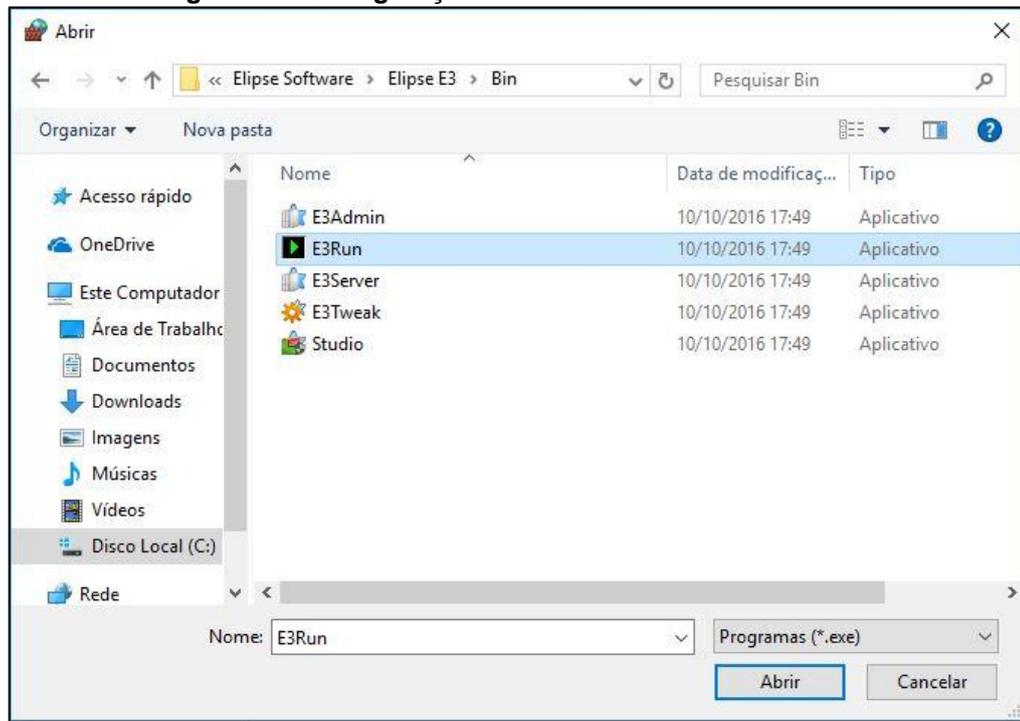
Figura 50 - Configuração Firewall – Diretório dos arquivos



Fonte: Autoria própria.

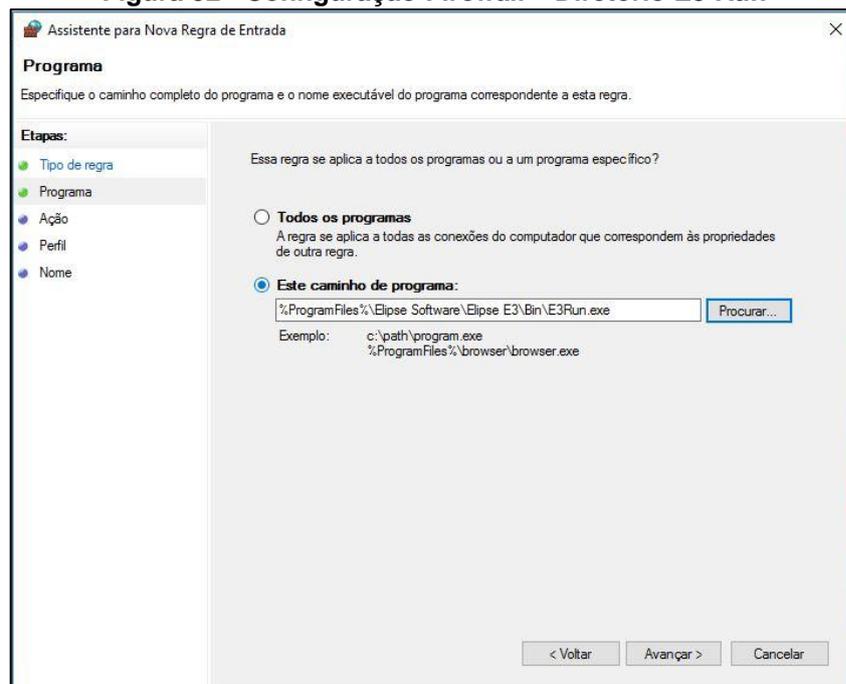
Adicionando o programa E3 Run.

Figura 51 - Configuração Firewall – Adicionando E3 Run



Fonte: Autoria própria.

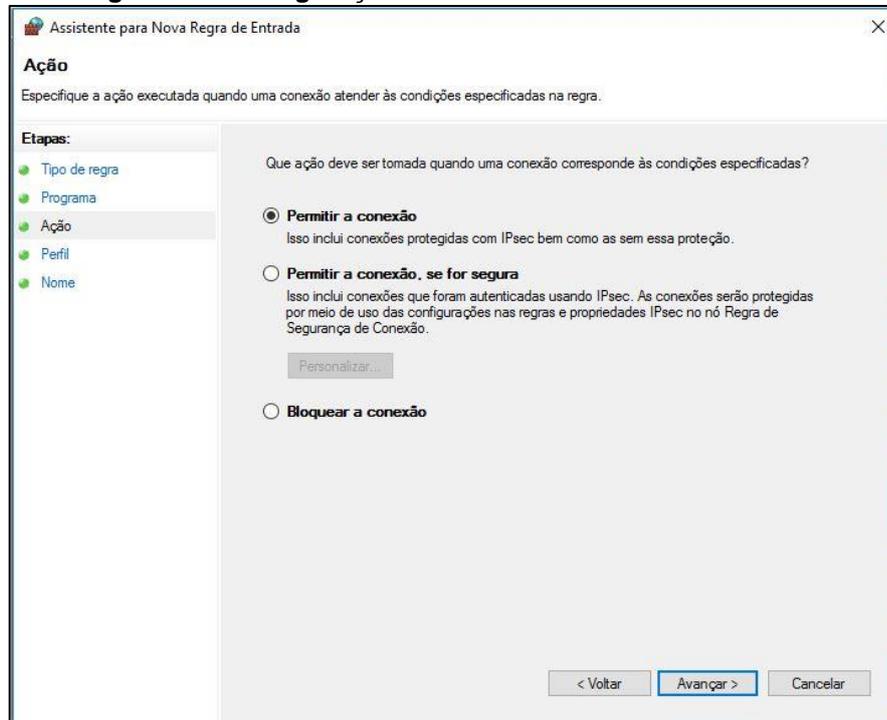
Figura 52 - Configuração Firewall – Diretório E3 Run



Fonte: Autoria própria.

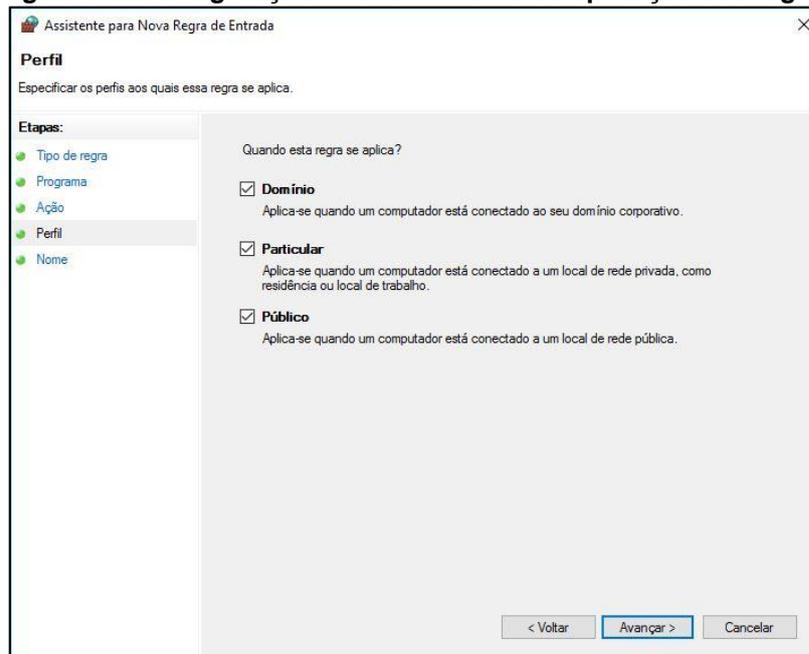
Permitimos a Conexão > Avançar > Domínios > Avançar.

Figura 53 - Configuração Firewall – Permissão de conexão



Fonte: Autoria própria.

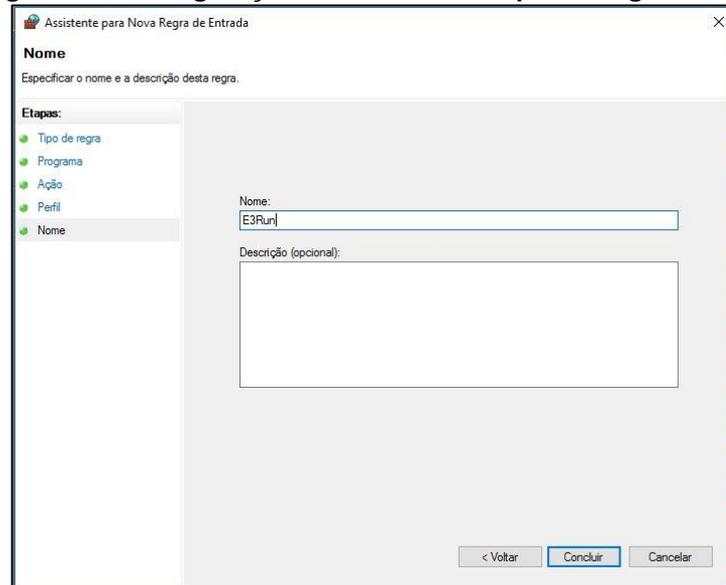
Figura 54 - Configuração Firewall – Perfis de aplicação das regras



Fonte: Autoria própria.

Nomeamos a regra do programa que acabamos de adicionar.

Figura 55 - Configuração Firewall – Nome para a regra E3 Run

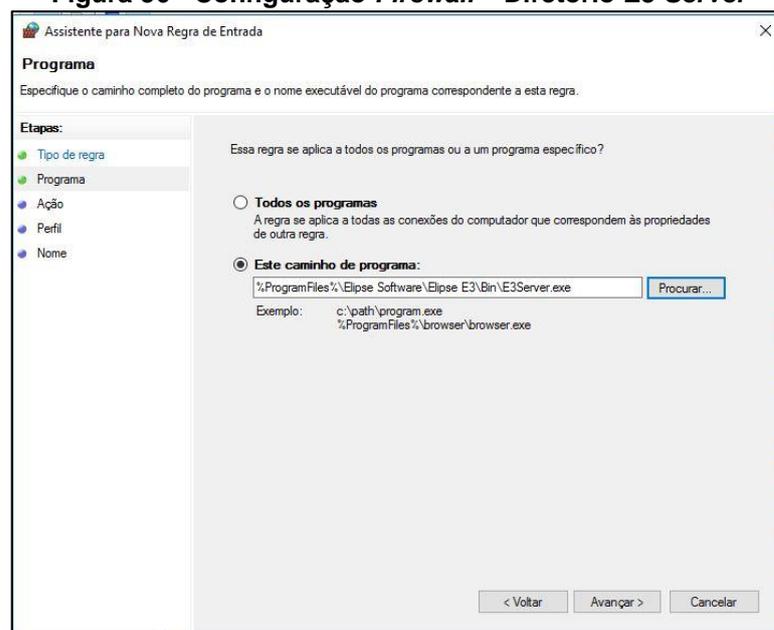


Fonte: Autoria própria.

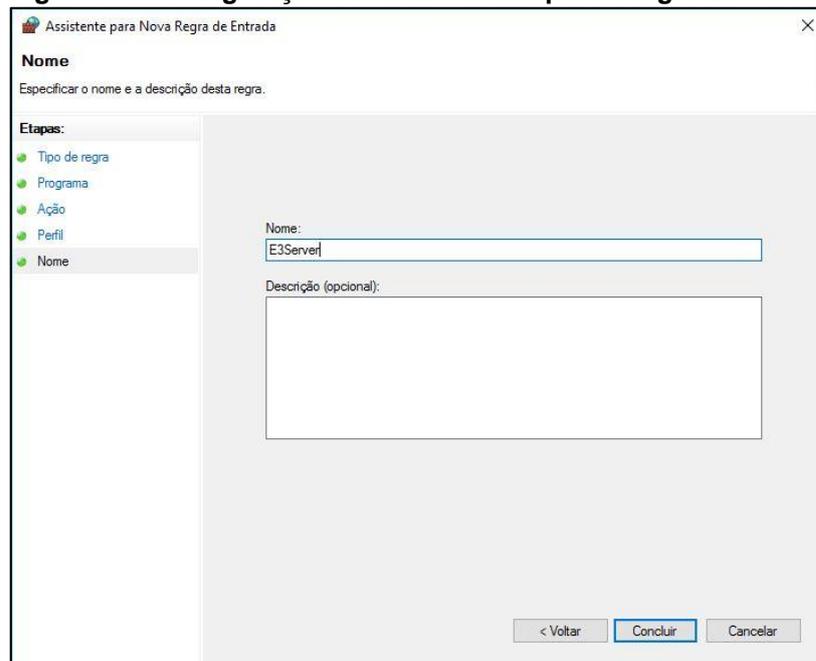
Fazemos o mesmo procedimento anterior para adicionar os arquivos do E3 Server, E3 Studio, Elipse SCADA e OpcEnum.

Para o E3 Server.

Figura 56 - Configuração Firewall – Diretório E3 Server

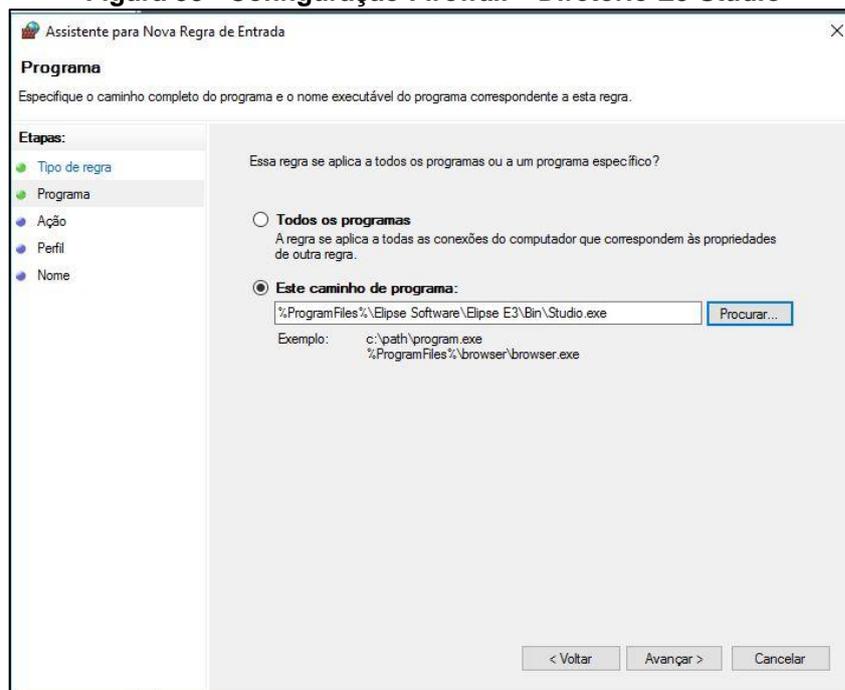


Fonte: Autoria própria.

Figura 57 - Configuração Firewall – Nome para a regra E3 Server

Fonte: Autoria própria.

Para o E3 Studio.

Figura 58 - Configuração Firewall – Diretório E3 Studio

Fonte: Autoria própria.

Figura 59 - Configuração Firewall – Nome para a regra E3 Studio

The screenshot shows the 'Assistente para Nova Regra de Entrada' (New Incoming Rule Wizard) window. The title bar reads 'Assistente para Nova Regra de Entrada'. The main heading is 'Nome' (Name). Below the heading, it says 'Especificar o nome e a descrição desta regra.' (Specify the name and description of this rule). On the left, there is a 'Etapas:' (Steps) list with five items: 'Tipo de regra' (Rule type), 'Programa' (Program), 'Ação' (Action), 'Perfil' (Profile), and 'Nome' (Name). The 'Nome' step is currently selected. The main area contains a 'Nome:' label followed by a text box containing the text 'Studio'. Below that is a 'Descrição (opcional):' (Optional description) label followed by a larger empty text box. At the bottom right, there are three buttons: '< Voltar' (Back), 'Concluir' (Finish), and 'Cancelar' (Cancel).

Fonte: Autoria própria.

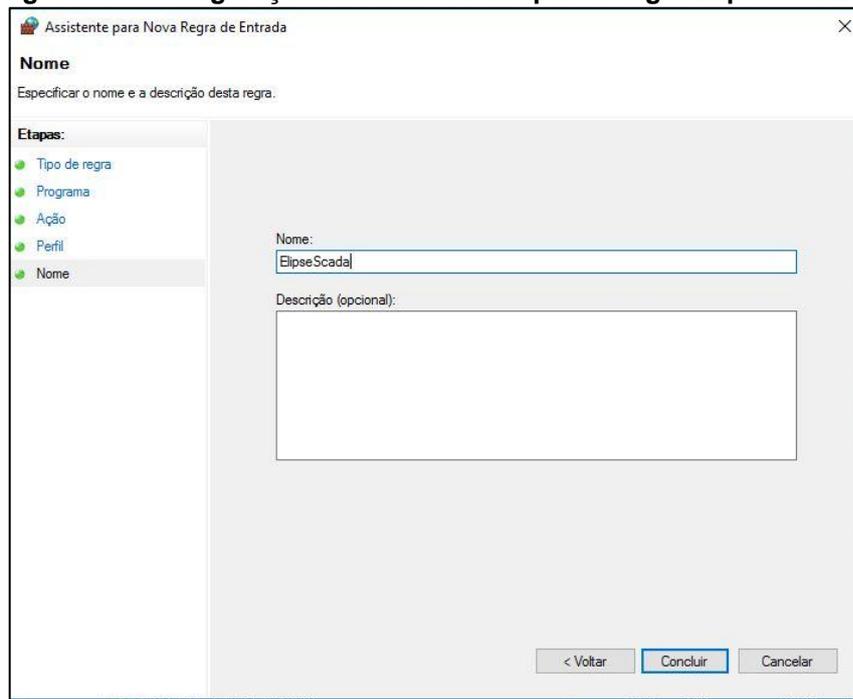
Para a aplicação Elipse SCADA.

Figura 60 - Configuração Firewall – Diretório Elipse SCADA

The screenshot shows the 'Assistente para Nova Regra de Entrada' (New Incoming Rule Wizard) window. The title bar reads 'Assistente para Nova Regra de Entrada'. The main heading is 'Programa' (Program). Below the heading, it says 'Especifique o caminho completo do programa e o nome executável do programa correspondente a esta regra.' (Specify the full path of the program and the executable name of the program corresponding to this rule). On the left, there is a 'Etapas:' (Steps) list with five items: 'Tipo de regra' (Rule type), 'Programa' (Program), 'Ação' (Action), 'Perfil' (Profile), and 'Nome' (Name). The 'Programa' step is currently selected. The main area contains the question 'Essa regra se aplica a todos os programas ou a um programa específico?' (Does this rule apply to all programs or to a specific program?). There are two radio button options: 'Todos os programas' (All programs) and 'Este caminho de programa:' (This program path:). The 'Este caminho de programa:' option is selected. Below this, there is a text box containing the path '%ProgramFiles%\x86\Elipse SCADA\Elipse32.exe' and a 'Procurar...' (Browse...) button. Below the text box, there are two examples: 'Exemplo: c:\path\program.exe' and 'Exemplo: %ProgramFiles%\browser\browser.exe'. At the bottom right, there are three buttons: '< Voltar' (Back), 'Avançar >' (Next), and 'Cancelar' (Cancel).

Fonte: Autoria própria.

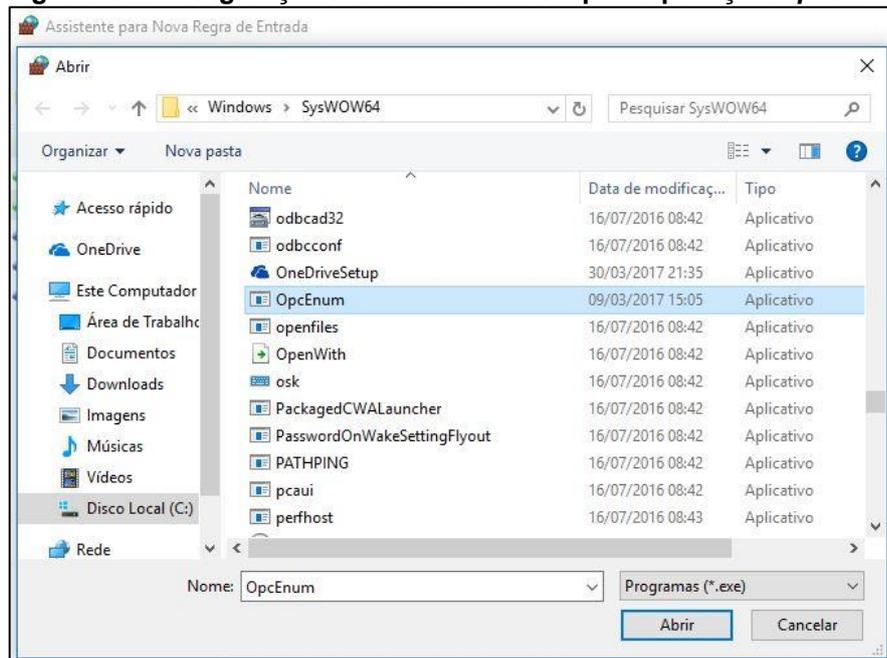
Figura 61 - Configuração Firewall – Nome para a regra Elipse SCADA



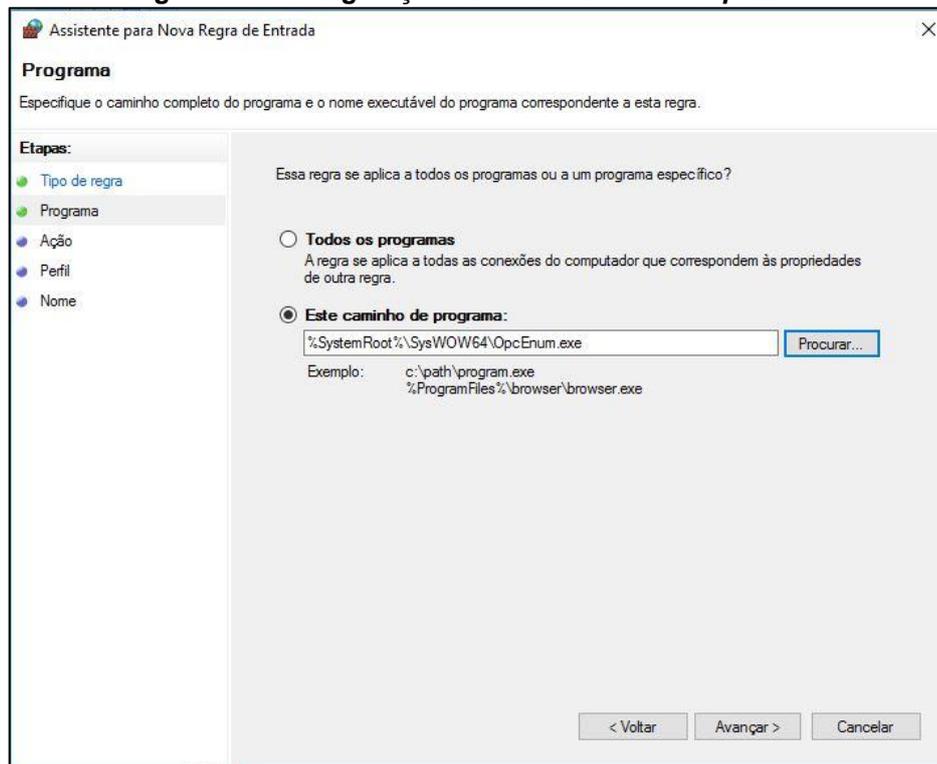
Fonte: Autoria própria.

Para *OPCEnum*.

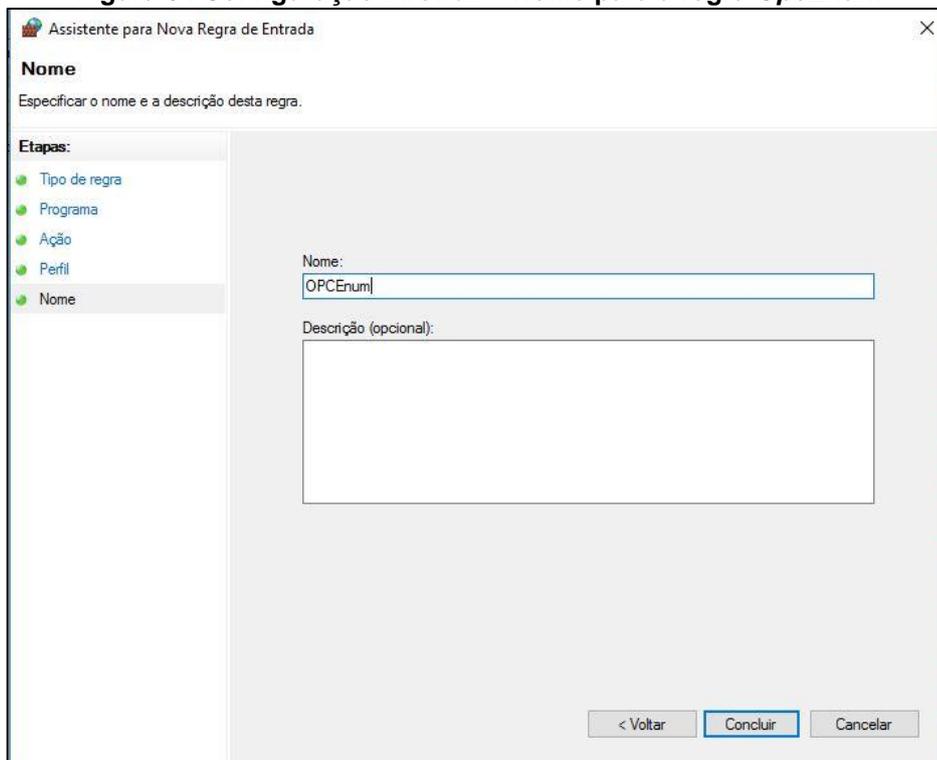
Figura 62 - Configuração Firewall – Caminho para aplicação OpcEnum



Fonte: Autoria própria.

Figura 63 - Configuração Firewall – Diretório OpcEnum

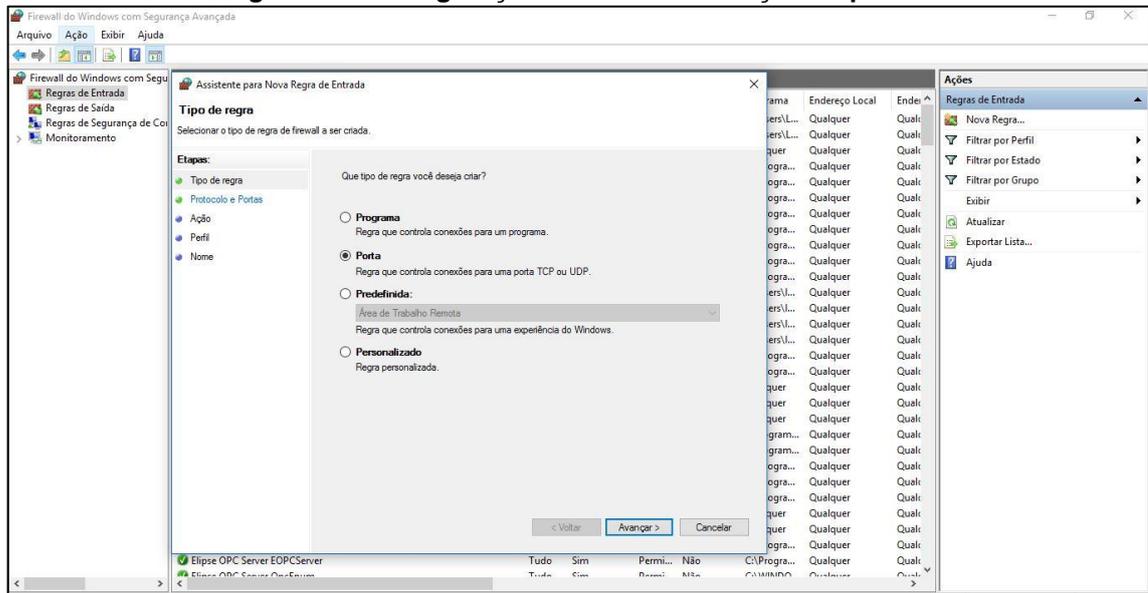
Fonte: Autoria própria.

Figura 64 Configuração Firewall – Nome para a regra OpcEnum

Fonte: Autoria própria.

Além disso devemos liberar as portas que serão utilizadas para conexões Elipse. Regras de Entrada > Nova Regra > Porta

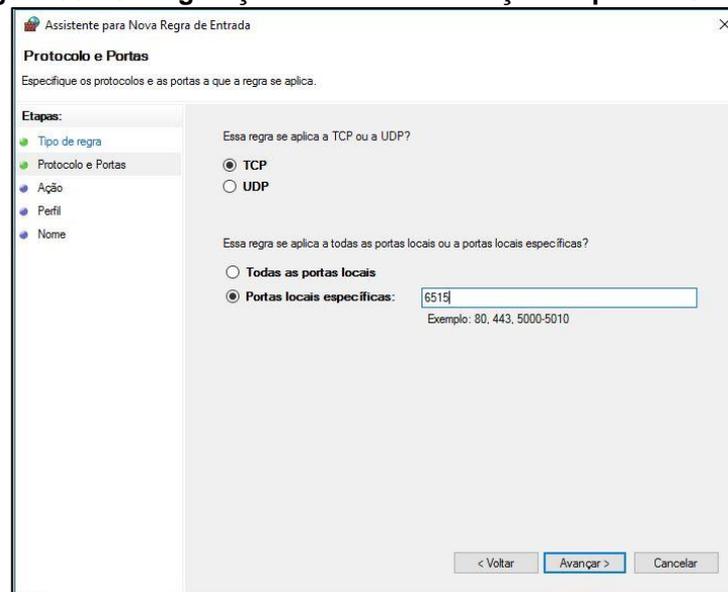
Figura 65 - Configuração Firewall – Liberação de portas



Fonte: Autoria própria.

Na porta TCP colocamos 6515, que é a porta responsável pela conexão do Viewer com o E3 Server e da conexão do Studio com o server remoto.

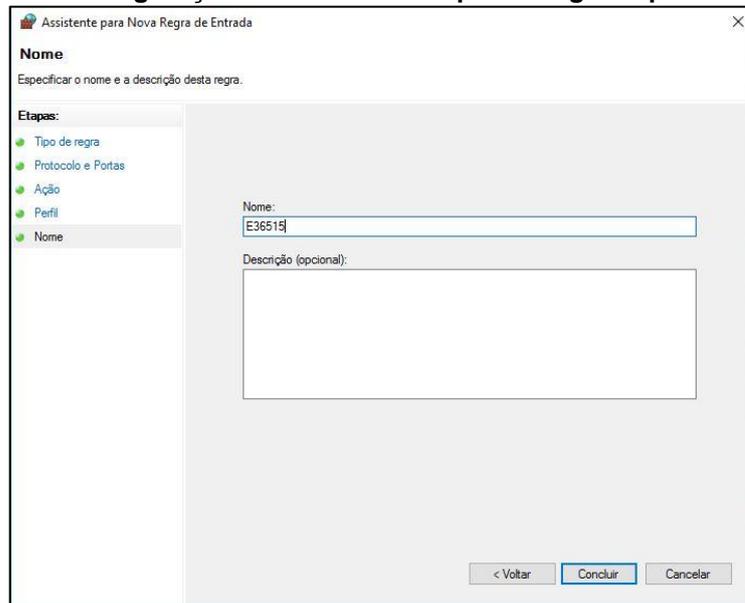
Figura 66 - Configuração Firewall – Liberação da porta TCP 6515



Fonte: Autoria própria.

Nomeamos como E36515.

Figura 67 - Configuração Firewall – Nome para a regra porta TCP 6515

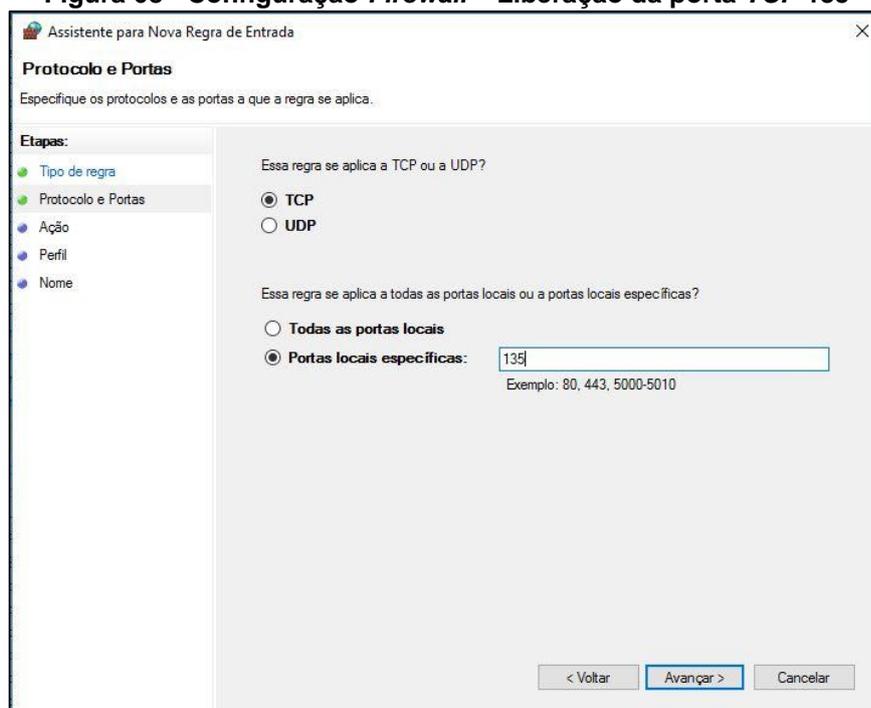


The screenshot shows the 'Assistente para Nova Regra de Entrada' (New Inbound Rule Wizard) window. The current step is 'Nome' (Name), where the user is prompted to specify the name and description of the rule. The 'Nome' field contains the text 'E36515'. The 'Descrição (opcional):' (Optional description) field is empty. The 'Etapas' (Steps) pane on the left shows the following steps: Tipo de regra, Protocolo e Portas, Ação, Perfil, and Nome (which is currently selected). At the bottom of the window, there are three buttons: '< Voltar' (Back), 'Concluir' (Finish), and 'Cancelar' (Cancel).

Fonte: Autoria própria.

Fazemos o mesmo processo, permitindo a porta 135, que é utilizada pelo DCOM. Devemos dar permissão tanto pelo TCP, quanto a via UDP.

Figura 68 - Configuração Firewall – Liberação da porta TCP 135



The screenshot shows the 'Assistente para Nova Regra de Entrada' (New Inbound Rule Wizard) window. The current step is 'Protocolo e Portas' (Protocol and Ports), where the user is prompted to specify the protocols and ports to which the rule applies. The 'Essa regra se aplica a TCP ou a UDP?' (Does this rule apply to TCP or UDP?) section has the 'TCP' radio button selected. The 'Essa regra se aplica a todas as portas locais ou a portas locais específicas?' (Does this rule apply to all local ports or specific local ports?) section has the 'Portas locais específicas:' (Specific local ports) radio button selected, and the text box next to it contains the number '135'. Below the text box, there is an example: 'Exemplo: 80, 443, 5000-5010'. The 'Etapas' (Steps) pane on the left shows the following steps: Tipo de regra, Protocolo e Portas (which is currently selected), Ação, Perfil, and Nome. At the bottom of the window, there are three buttons: '< Voltar' (Back), 'Avançar >' (Next), and 'Cancelar' (Cancel).

Fonte: Autoria própria.

Figura 69 - Configuração Firewall – Nome para a regra da porta TCP 135

The screenshot shows a dialog box titled "Assistente para Nova Regra de Entrada" with a close button (X) in the top right corner. The main heading is "Nome" and the instruction is "Especificar o nome e a descrição desta regra." On the left, a vertical list of steps is shown: "Tipo de regra", "Protocolo e Portas", "Ação", "Perfil", and "Nome", with "Nome" selected. The main area contains a "Nome:" label followed by a text input field containing "DCOM_135_TCP". Below it is a "Descrição (opcional):" label followed by a larger empty text area. At the bottom right, there are three buttons: "< Voltar", "Concluir", and "Cancelar".

Fonte: Autoria própria.

Figura 70 - Configuração Firewall – Liberação da porta UDP 135

The screenshot shows the same dialog box, now at the "Protocolo e Portas" step. The instruction is "Especifique os protocolos e as portas a que a regra se aplica." The left-hand list of steps now has "Protocolo e Portas" selected. The main area contains two questions. The first is "Essa regra se aplica a TCP ou a UDP?" with radio buttons for "TCP" and "UDP", where "UDP" is selected. The second is "Essa regra se aplica a todas as portas locais ou a portas locais específicas?" with radio buttons for "Todas as portas locais" and "Portas locais específicas:", where "Portas locais específicas:" is selected. Below this is a text input field containing "135" and an example text "Exemplo: 80, 443, 5000-5010". At the bottom right, there are three buttons: "< Voltar", "Avançar >", and "Cancelar".

Fonte: Autoria própria.

Figura 71 - Configuração Firewall – Nome para a regra da porta UDP 135

Assistente para Nova Regra de Entrada

Nome

Especificar o nome e a descrição desta regra.

Etapas:

- Tipo de regra
- Protocolo e Portas
- Ação
- Perfil
- Nome

Nome:
DCOM_135_UDP

Descrição (opcional):

< Voltar Concluir Cancelar

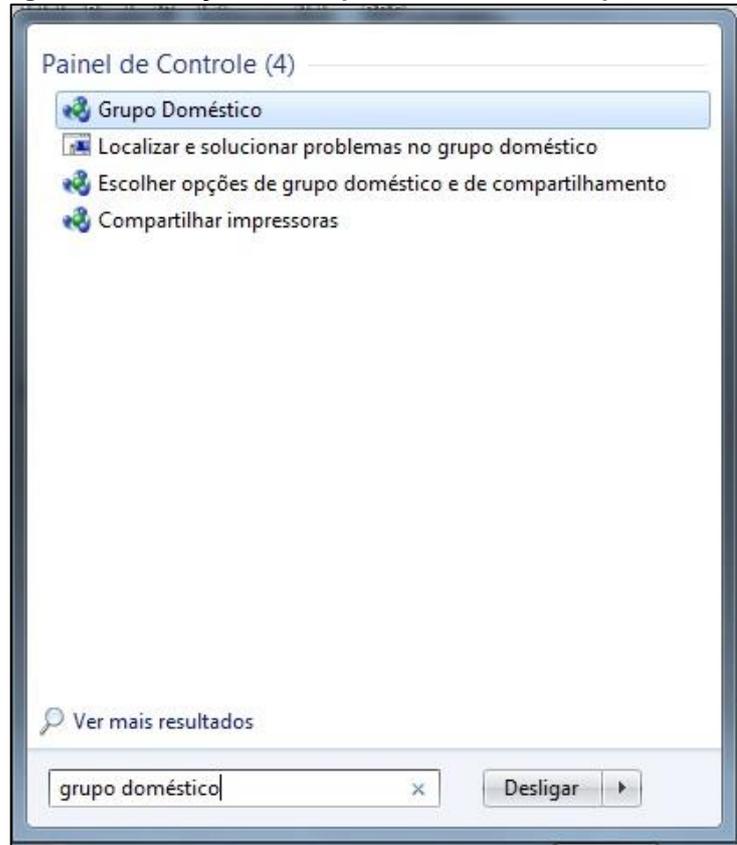
Fonte: Autoria própria.

APÊNDICE B – COMO CRIAR UMA REDE DOMÉSTICA

Esse apêndice apresenta um tutorial detalhado de como criar um grupo doméstico.

O primeiro passo para abrir um Grupo Doméstico, digite grupo doméstico na caixa de pesquisa na barra de tarefas e escolha Grupo Doméstico.

Figura 72 – Criação do Grupo Doméstico – campo de busca

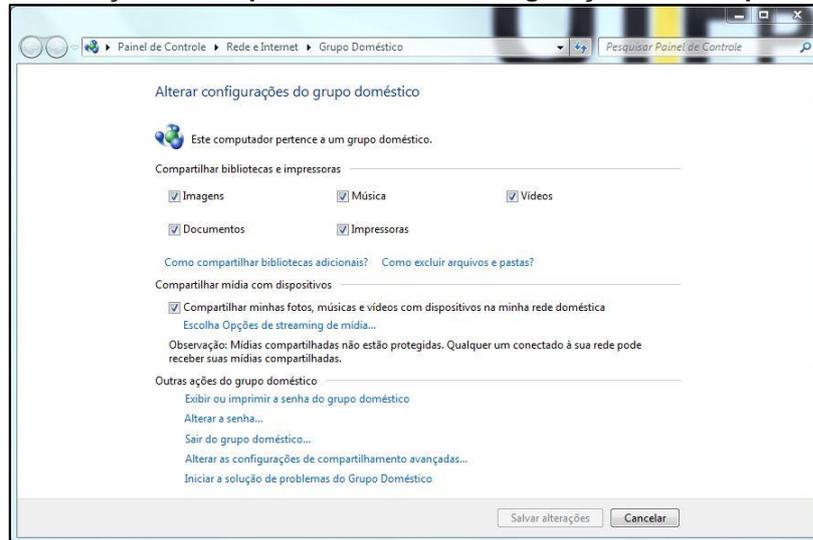


Fonte: Autoria própria.

Selecione Criar um grupo doméstico > Avançar.

Escolha as bibliotecas e os dispositivos que você quer compartilhar com o grupo doméstico e selecione Avançar.

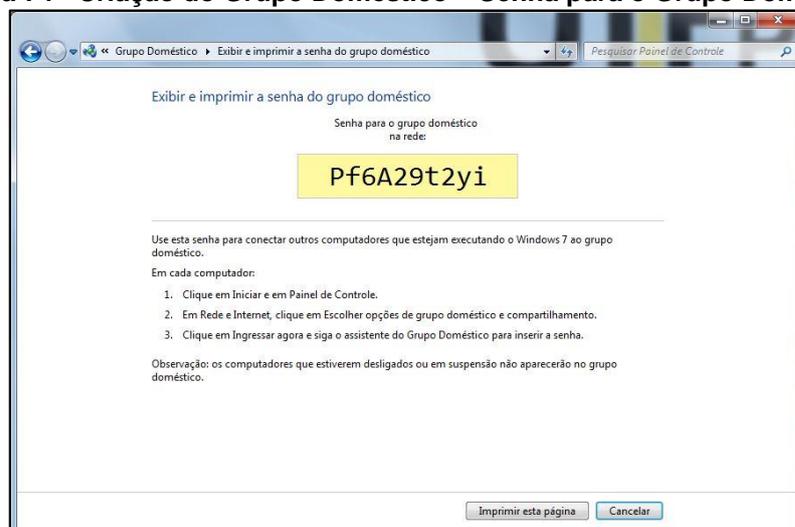
Figura 73 - Criação do Grupo Doméstico – Configurações de compartilhamento



Fonte: Autoria própria.

Aparecerá uma senha que você deve imprimir ou anotar. Ela será necessária para adicionar outros computadores ao grupo doméstico. Selecione Concluir

Figura 74 - Criação do Grupo Doméstico – Senha para o Grupo Doméstico



Fonte: Autoria própria.

Para adicionar os outros computadores ao grupo doméstico, depois que alguém da sua rede cria um grupo doméstico, o próximo passo é ingressar nesse grupo. Você precisará da senha do grupo doméstico, que pode ser obtida com algum membro desse grupo. Todas as contas de usuário, exceto a conta Convidado, pertencerão ao grupo doméstico. Cada pessoa controla o acesso às suas próprias bibliotecas.

Para entrar em um grupo doméstico, siga estas etapas no computador que você quer adicionar ao grupo doméstico:

Se não houver um lugar para inserir a senha, pode ser que não haja um grupo doméstico disponível. Confira se alguém criou um grupo doméstico, ou você mesmo poderá criá-lo.

Para abrir um Grupo Doméstico, digite grupo doméstico na caixa de pesquisa na barra de tarefas e escolha Grupo Doméstico, Ingressar agora > Avançar.

Escolha as bibliotecas e os dispositivos que você quer compartilhar com o grupo doméstico e selecione Avançar.

Digite a senha do grupo doméstico na caixa e selecione Avançar e selecione Concluir.

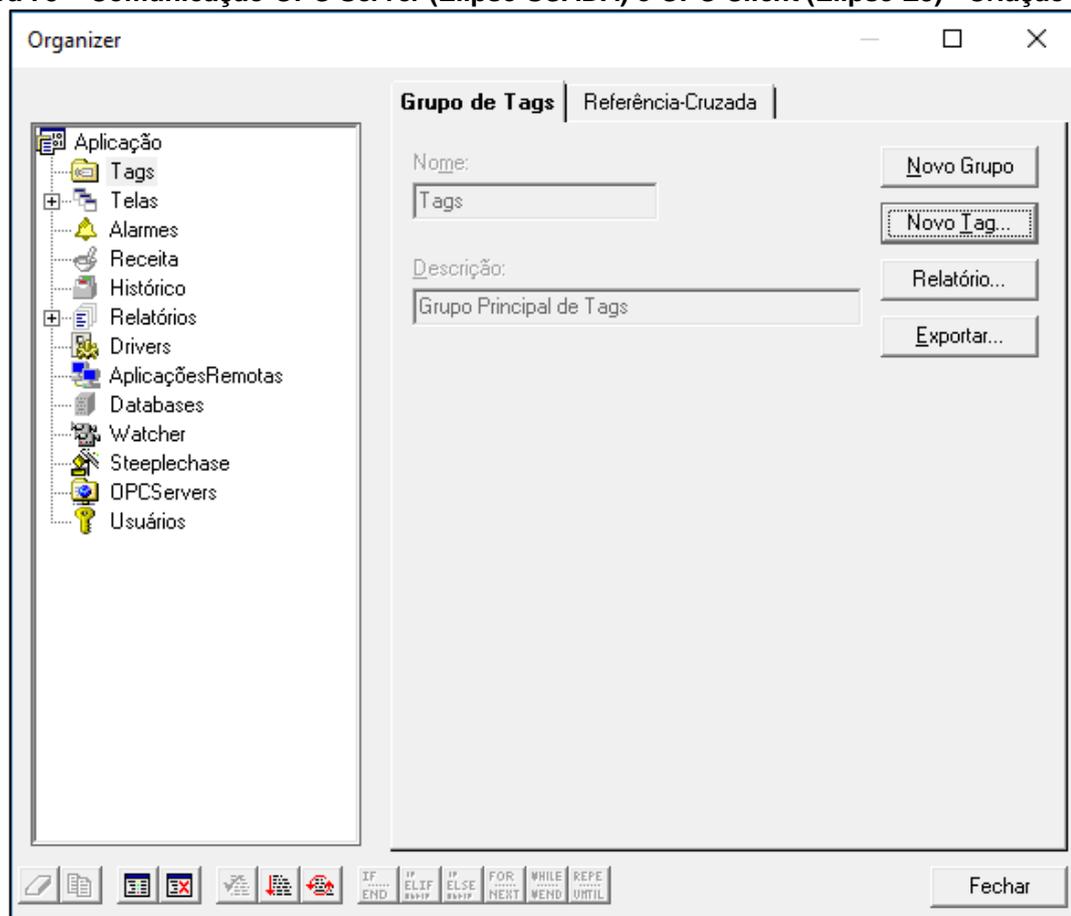
APÊNDICE C – TUTORIAL PARA COMUNICAÇÃO ENTRE OPC SERVER E OPC CLIENT

Esse apêndice apresenta um tutorial detalhado de como realizar a comunicação OPC entre OPC Server e OPC Client, utilizando os softwares Elipse SCADA e Elipse E3.

O primeiro passo é a instalação dos programas Elipse SCADA e Elipse E3, e assim definir qual aplicação será o OPC Server e OPC Client. Nesse tutorial foi definido o sistema supervisório Elipse SCADA como Servidor e o E3 como Cliente.

Abrindo o programa Elipse SCADA precisamos criar as tags para obtenção dos dados. Para isso acessamos Arquivo > Organizer > Tags > Novo Tag

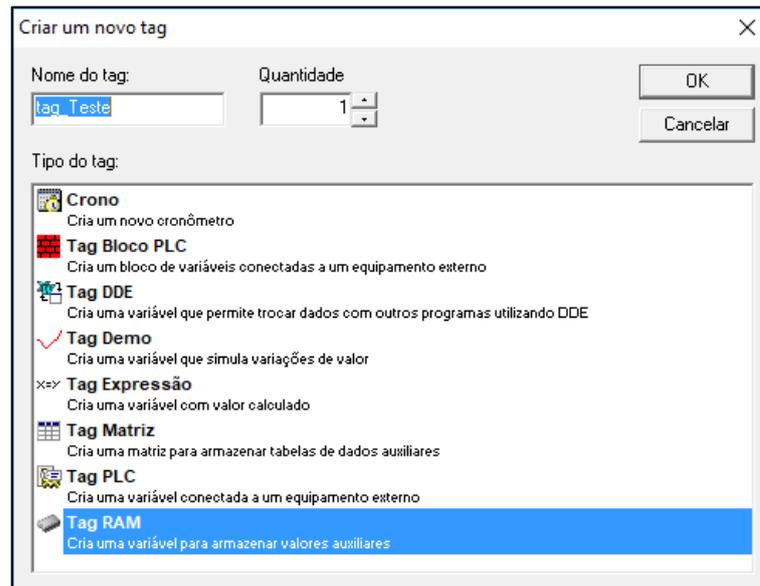
Figura 75 – Comunicação OPC Server (Elipse SCADA) e OPC Client (Elipse E3) - Criação de tag



Fonte: Autoria própria.

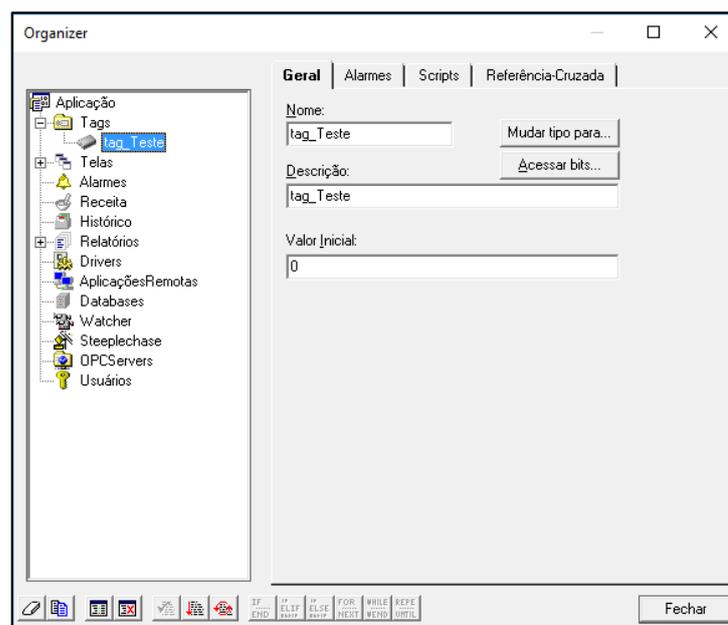
Abrirá uma janela para escolha do tipo da *tag*, onde nesse tutorial foi definida com *tag RAM*, que tem o objetivo de armazenar valores. Após isso, nomeamos ela como *tag_Teste* e damos um OK para criação da *tag*.

Figura 76 - Comunicação OPC Server (Elipse SCADA) e OPC Client (Elipse E3) – Tipo de tag



Fonte: Autoria própria.

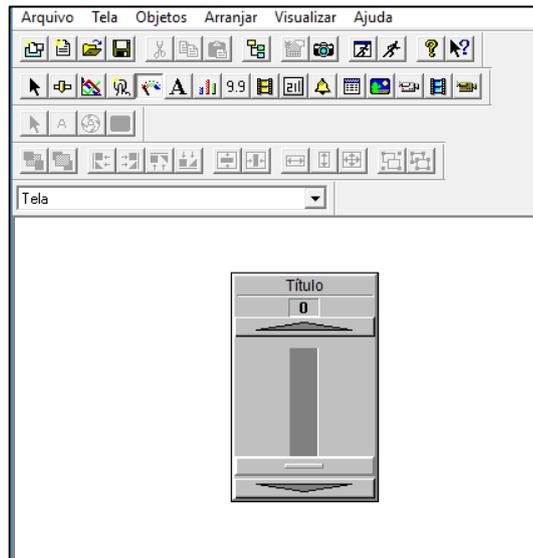
Figura 77 - Comunicação OPC Server (Elipse SCADA) e OPC Client (Elipse E3) – Visualização da tag criada



Fonte: Autoria própria.

Posteriormente precisamos adicionar o visualizador que será vinculado a *tag*, de modo a gerar os dados. Foi adicionado um *Slider*, que altera os valores ao movimentar a barra.

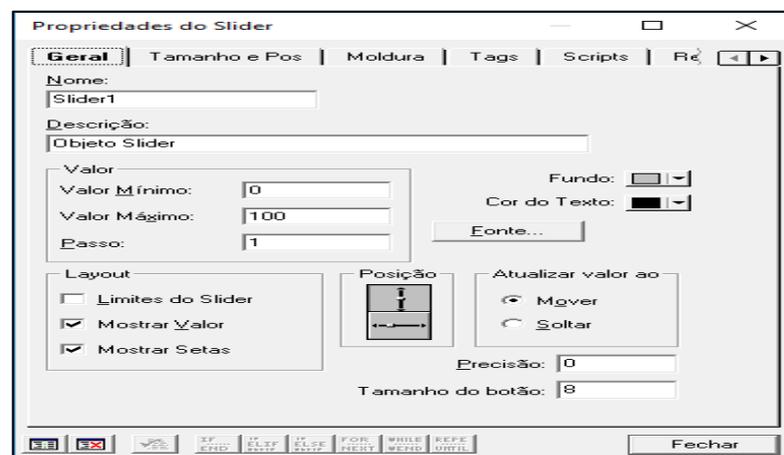
Figura 78 - Comunicação OPC Server (Elipse SCADA) e OPC Client (Elipse E3) – Criação do Slider



Fonte: Autoria própria.

Para melhor visualização dos valores, foram feitas configurações no *Slider* de modo a mostrar e limitar o valor ao arraste da barra.

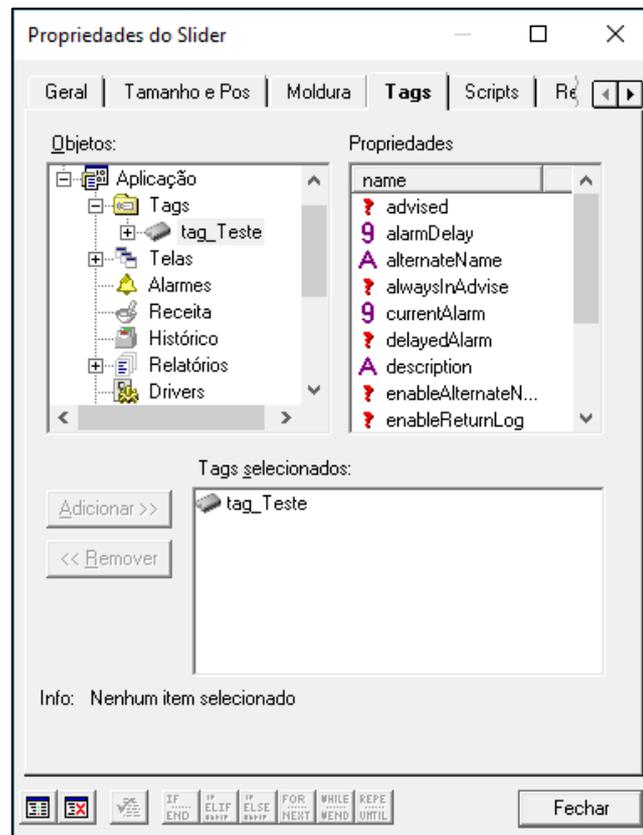
Figura 79 - Comunicação OPC Server (Elipse SCADA) e OPC Client (Elipse E3) – Definições do Slider



Fonte: Autoria própria.

Assim podemos vincular a *tag_Test* ao visualizador *Slider*.

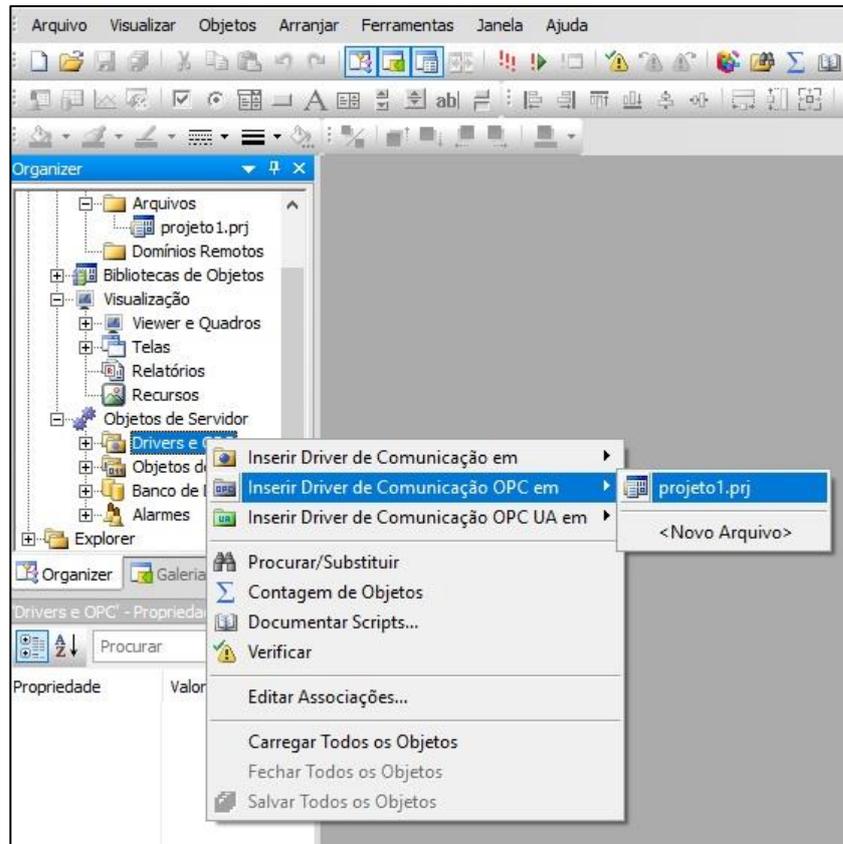
Figura 80 - Comunicação OPC Server (Elipse SCADA) e OPC Client (Elipse E3) – Vinculando a tag com o Slider



Fonte: Autoria própria.

Depois de criar e vincular a *tag*, precisamos fazer as configurações no *OPC Client*. O Elipse E3 foi definido nesse tutorial como *OPC Client* para monitoramento dos dados. Abrindo a aplicação E3 *Studio*, são definidos nome do projeto e opções de configuração. Para comunicação com o *OPC Server* precisamos clicar na aba do *Organizer* com o botão direito em *Driver* e *OPC > Inserir Driver de Comunicação OPC* em > e selecionamos o nome do projeto que foi criado.

Figura 81 - Comunicação OPC Server (Eclipse SCADA) e OPC Client (Eclipse E3) – Inserindo Driver de Comunicação OPC



Fonte: Autoria própria.

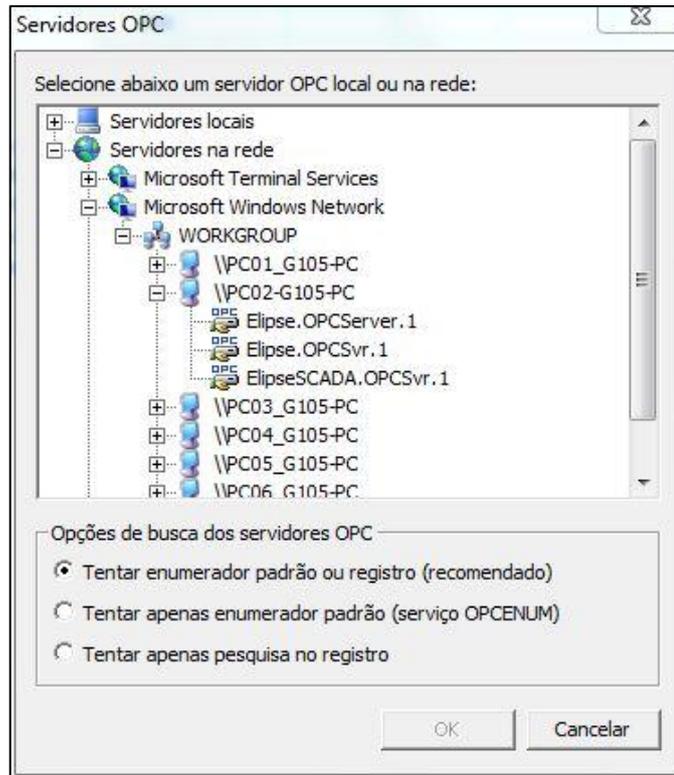
Selecionamos o Servidor OPC, com a opção de Servidores Locais ou Servidores na rede. Para esse tutorial realizamos a comunicação com o servidor OPC na rede.

Figura 82 - Comunicação OPC Server (Eclipse SCADA) e OPC Client (Eclipse E3) – Selecionando servidor



Fonte: Autoria própria.

Figura 83 - Comunicação OPC Server (Elipse SCADA) e OPC Client (Elipse E3) – Servidores OPC



Fonte: Autoria própria.

Ativamos a comunicação

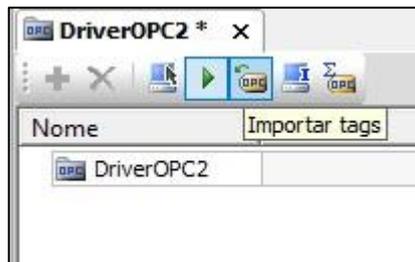
Figura 84 - Comunicação OPC Server (Elipse SCADA) e OPC Client (Elipse E3) – Ativando a comunicação



Fonte: Autoria própria.

Posteriormente realizamos a importação das *tags*, onde o sistema pede a permissão para criar grupo de *tags*.

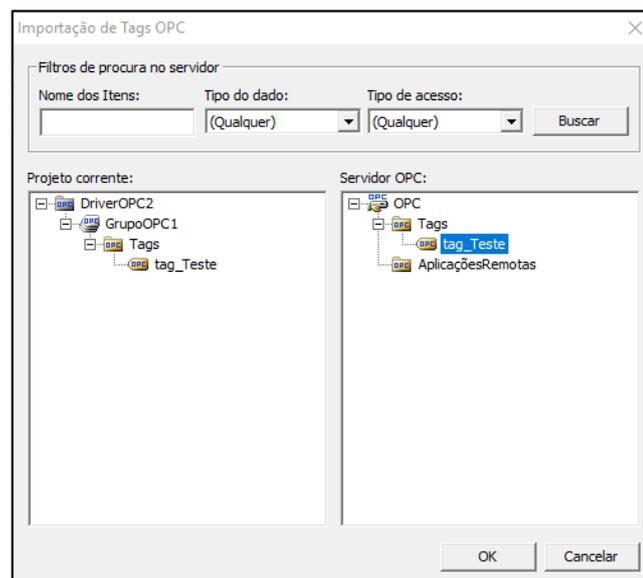
Figura 85 - Comunicação OPC Server (Elipse SCADA) e OPC Client (Elipse E3) – Importação das tags



Fonte: Autoria própria.

Na janela de Importação de *Tags OPC*, clicamos em *Buscar*, e a aplicação pesquisa as *tags* dos Servidor *OPC*. Assim clicamos nas *tags* que desejamos importar e arrastamos para o grupo do Projeto corrente.

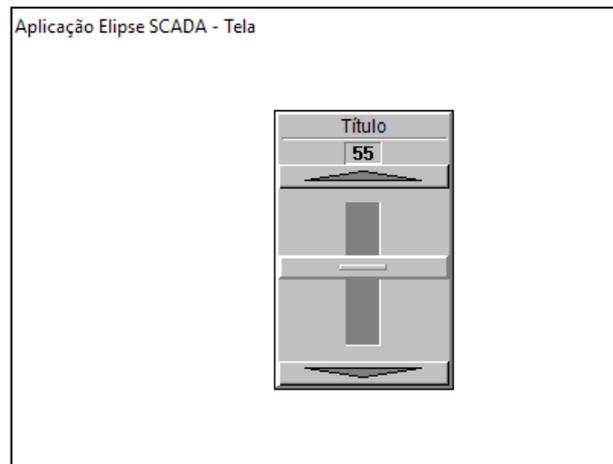
Figura 86 - Comunicação OPC Server (Elipse SCADA) e OPC Client (Elipse E3) – Importação das tags do servidor OPC para o cliente



Fonte: Autoria própria.

Feitas as importações das *tags*, executamos a aplicação do *OPC Server* Elipse SCADA, apertando a tecla *F10*. Arrastamos o *Slider* ao um valor.

Figura 87 - Comunicação *OPC Server* (Elipse SCADA) e *OPC Client* (Elipse E3) – Aplicação Elipse SCADA com o *Slider*



Fonte: Autoria própria.

Verificando no Elipse E3 que funciona como *OPC Client*, o monitoramento dos dados do *OPC Server*, afirmando a ocorrência da comunicação *OPC* nesse tutorial.

Figura 88 - Comunicação *OPC Server* (Elipse SCADA) e *OPC Client* (Elipse E3) – Monitorando do valor do *Slider*

Nome	ID do Item	Var...	Valor
DriverOPC2			
GrupoOPC1		1000	
Tags			
tag_Testes	Tags.tag_Testes	9	55

Fonte: Autoria própria.

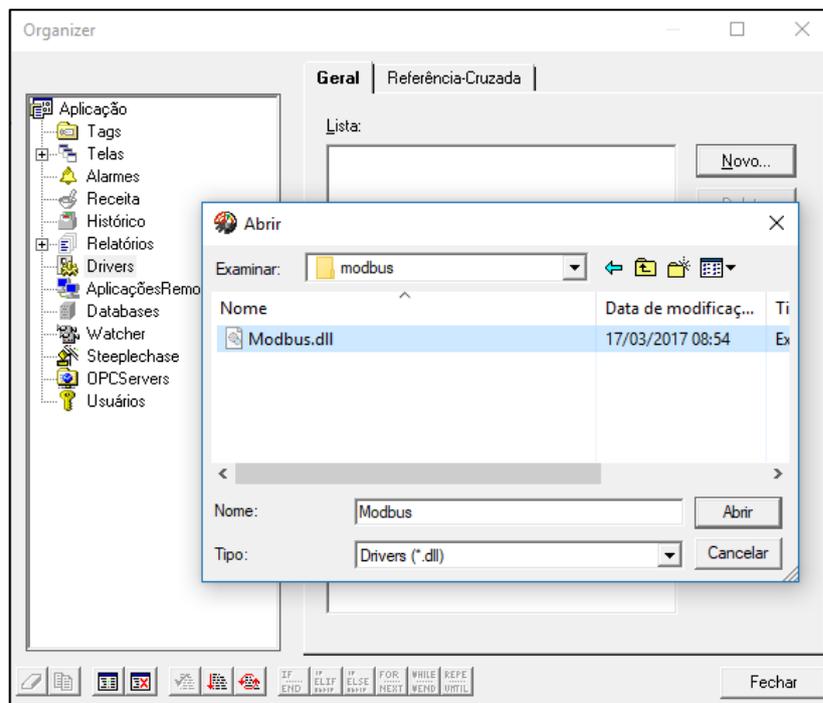
APÊNDICE D – TUTORIAL PARA COMUNICAÇÃO ENTRE OPC SERVER E MODBUS SIMULATOR

Esse apêndice apresenta um tutorial detalhado de como realizar a comunicação *MODBUS* entre *OPC Server* e o *MODBUS Simulator*, utilizando os softwares *Elipse SCADA* e o *Elipse MODBUS Simulator*.

O primeiro passo é ter feito a instalação dos programas *Elipse SCADA* e do *Elipse MODBUS Simulator*. Precisamos adicionar no *Elipse SCADA* o driver do equipamento para realizar a comunicação com o Simulador. Esse driver é o *modbus.dll* disponibilizado no site da *Elipse Software* para download.

Após baixado o arquivo, abrimos a aplicação *Elipse SCADA* e acessamos *Arquivo > Organizer > Drivers > Novo*, então procuramos no diretório o local em que foi salvo o arquivo.

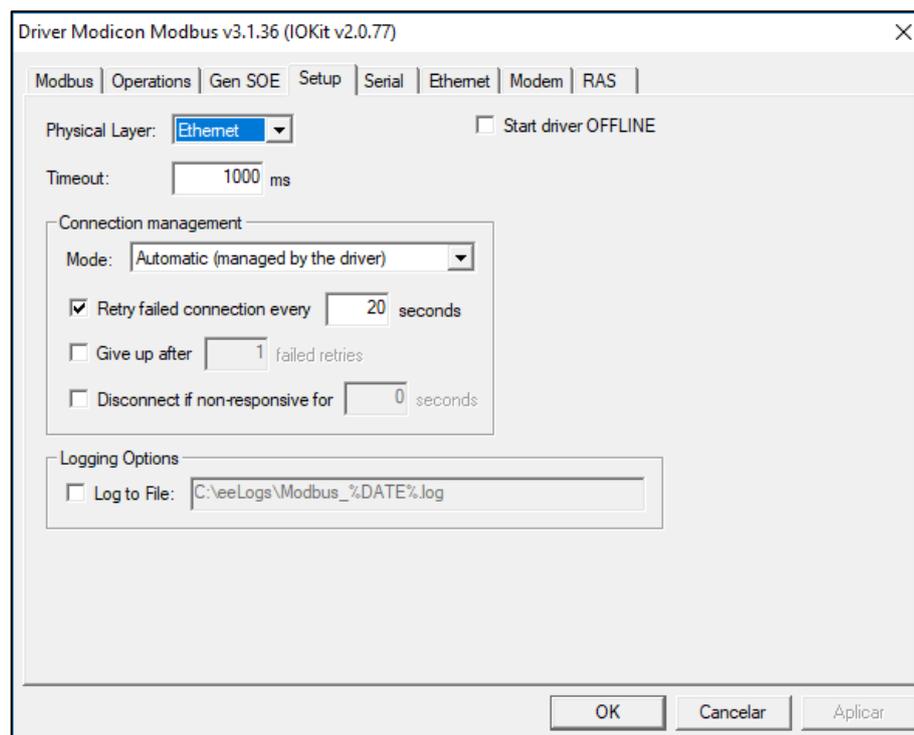
Figura 89 - Comunicação OPC Server (Elipse SCADA) e Simulador MODBUS – Adicionando o driver modbus.dll



Fonte: Autoria própria.

Após adicionar o arquivo, são necessárias algumas configurações no drivers para comunicação. Para isso clicamos em Arquivo > *Organizer* > *Driver* > Configurar > Extras e na aba *setup*. No índice *Physical Layer*, escolher a opção *Ethernet*.

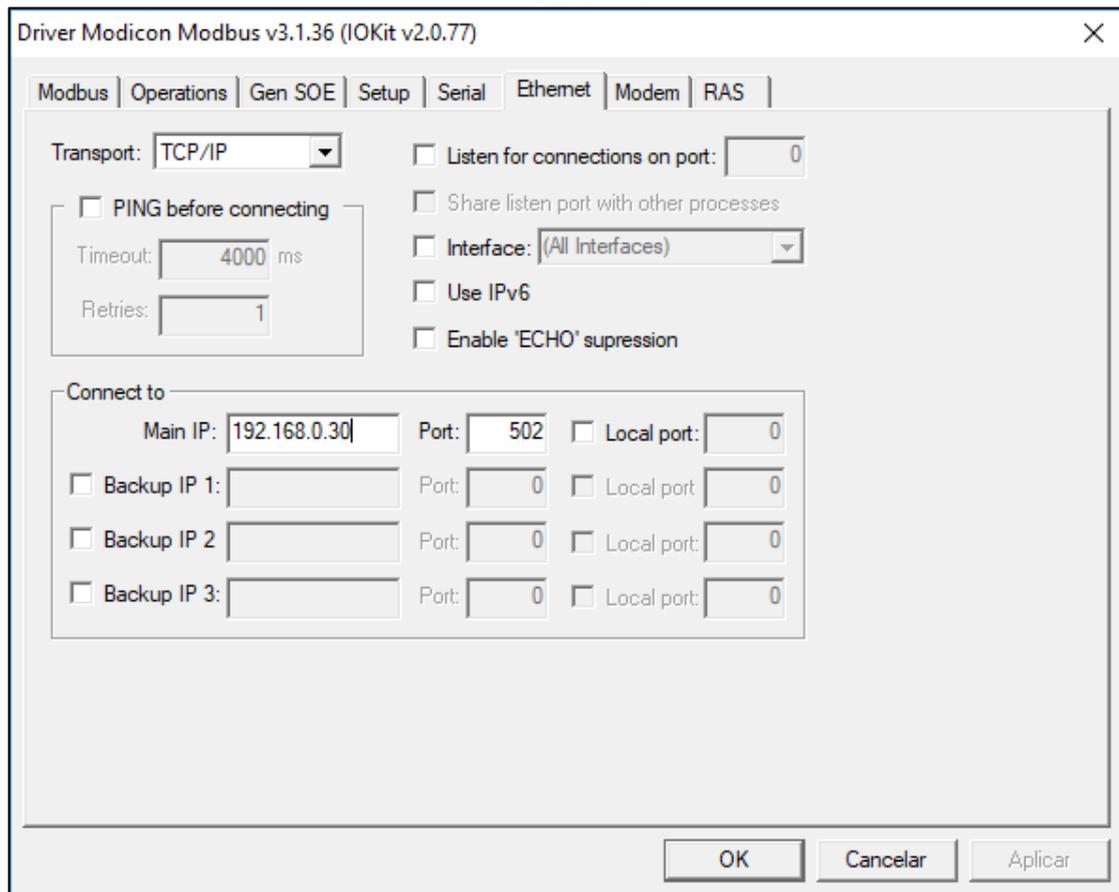
Figura 90 - Comunicação OPC Server (Eclipse SCADA) e Simulador MODBUS – Configuração no driver de comunicação



Fonte: Autoria própria.

Na aba Ethernet definir o IP onde se encontra o Simulador, se localhost (127.0.0.1) ou em rede. No tutorial o MODBUS Simulator foi definido em rede.

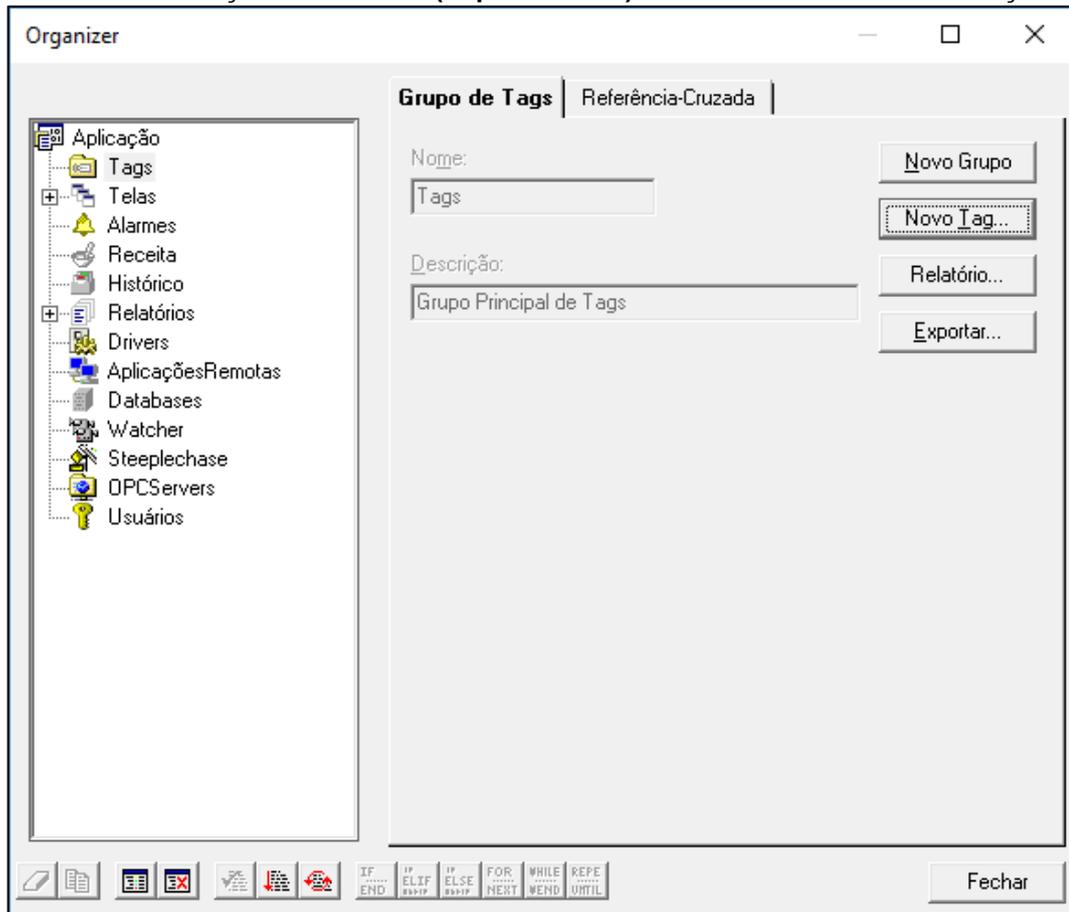
Figura 91 - Comunicação OPC Server (Elipse SCADA) e Simulador MODBUS – Definição de IP do Simulador



Fonte: Autoria própria.

Após as configurações, criamos as *tags* para obtenção dos dados. Para isso acessamos Arquivo > Organizer > Tags > Novo Tag

Figura 92 - Comunicação OPC Server (Eclipse SCADA) e Simulador MODBUS – Criação da tag

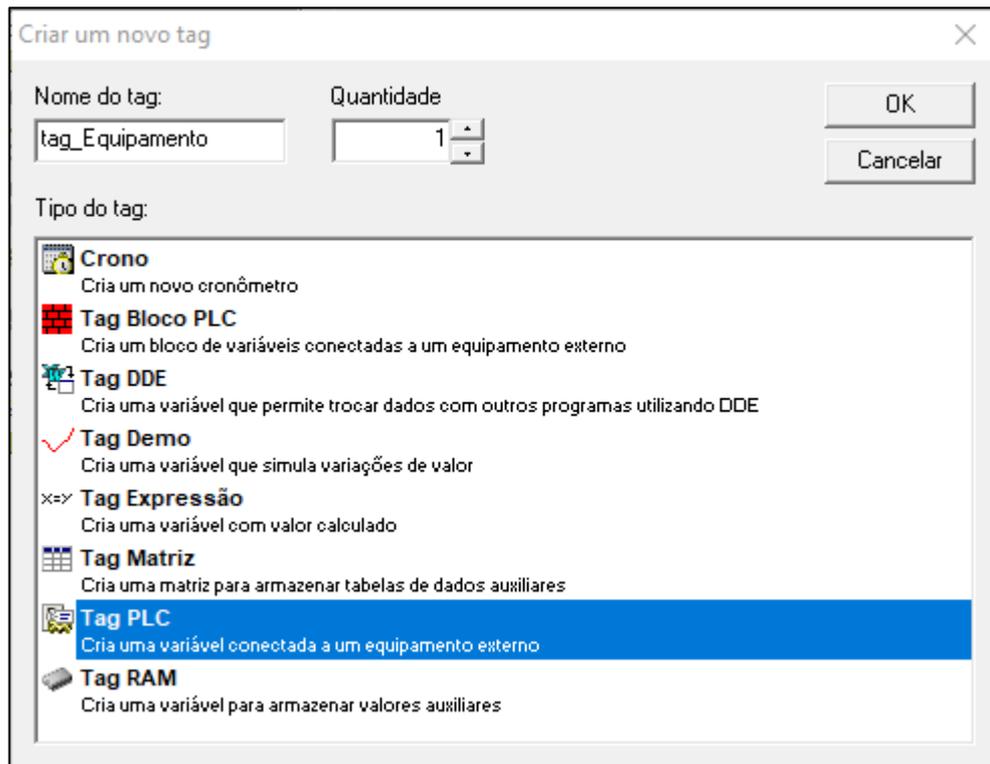


Fonte: Autoria própria.

Abrirá uma janela para escolha do tipo da *tag*, onde precisamos definir a *tag* como *tag PLC*, que tem o objetivo de obter valores de um equipamento externo.

Após isso, nomeamos ela como *tag_Equipamento* e damos um OK para criação da *tag*.

Figura 93 - Comunicação OPC Server (Elipse SCADA) e Simulador MODBUS – Criação da tag PLC

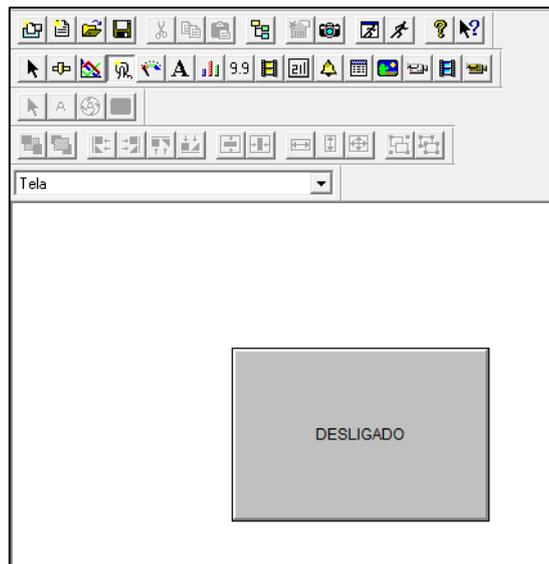


Fonte: Autoria própria.

Posteriormente precisamos adicionar o visualizador que será vinculado a *tag*, de modo a gerar os dados.

Foi adicionado um Botão, que recebe valores binários e apresenta a mensagem Ligado ou Desligado.

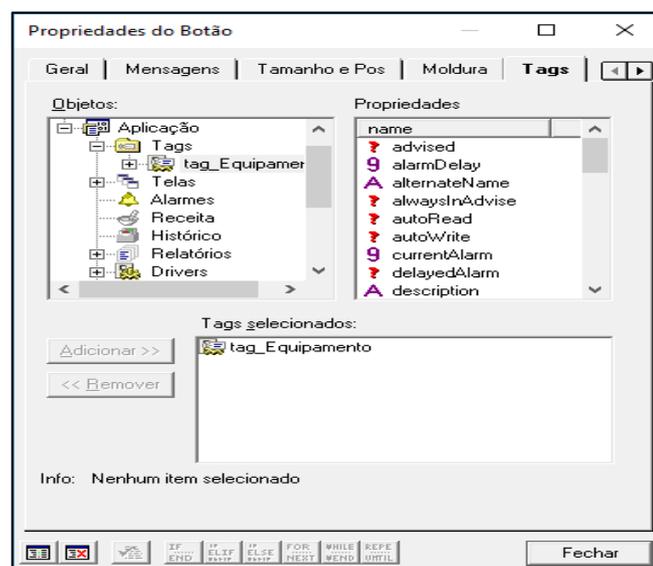
Figura 94 - Comunicação OPC Server (Elipse SCADA) e Simulador MODBUS – Visualizador Botão



Fonte: Autoria própria.

Assim podemos vincular a *tag_Equipamento* ao botão.

Figura 95 - Comunicação OPC Server (Elipse SCADA) e Simulador MODBUS – Vincular a tag PLC ao botão



Fonte: Autoria própria.

Depois de criar e vincular a *tag*, precisamos fazer as configurações dos parâmetros de leitura do equipamento. Precisamos atribuir a *tag* o endereço que se encontra no simulador. São definidos 3 parâmetros.

N1 – Qual o número do CLP que está relacionado a variável

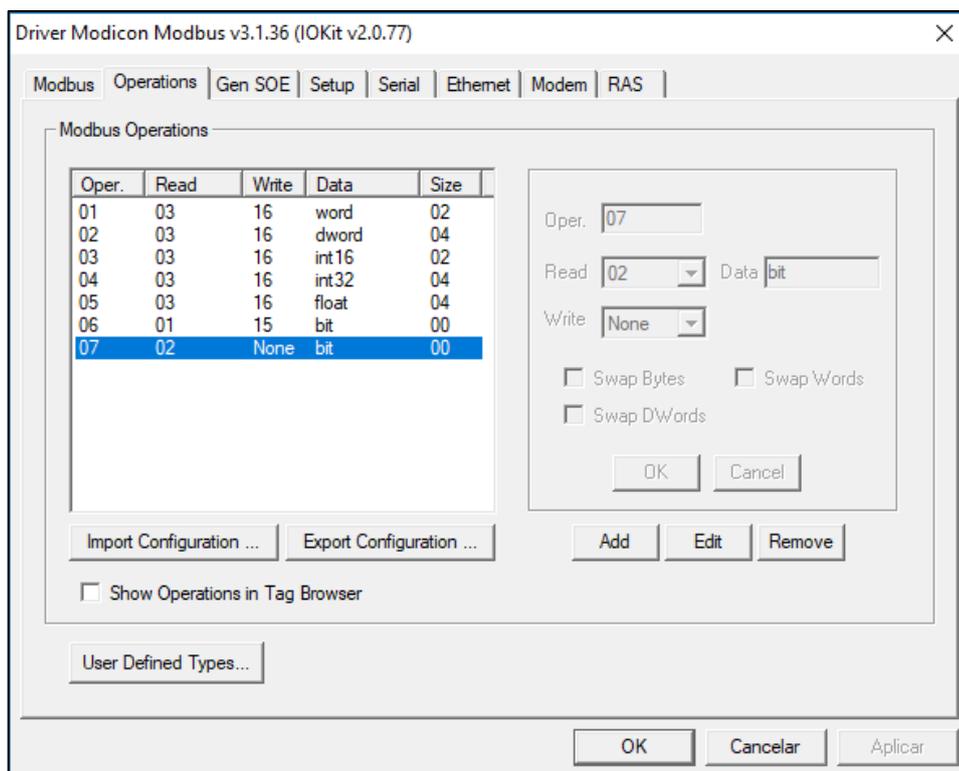
N2 – Operação a ser utilizada para a variável

N3 – Não se aplica, sempre 0

N4 – Endereço da variável dentro do CLP

A própria aplicação é composta por operações prontas para serem definidas. Arquivo > *Organizer* > *Driver* > Configurar > Extras e na aba *Operations*.

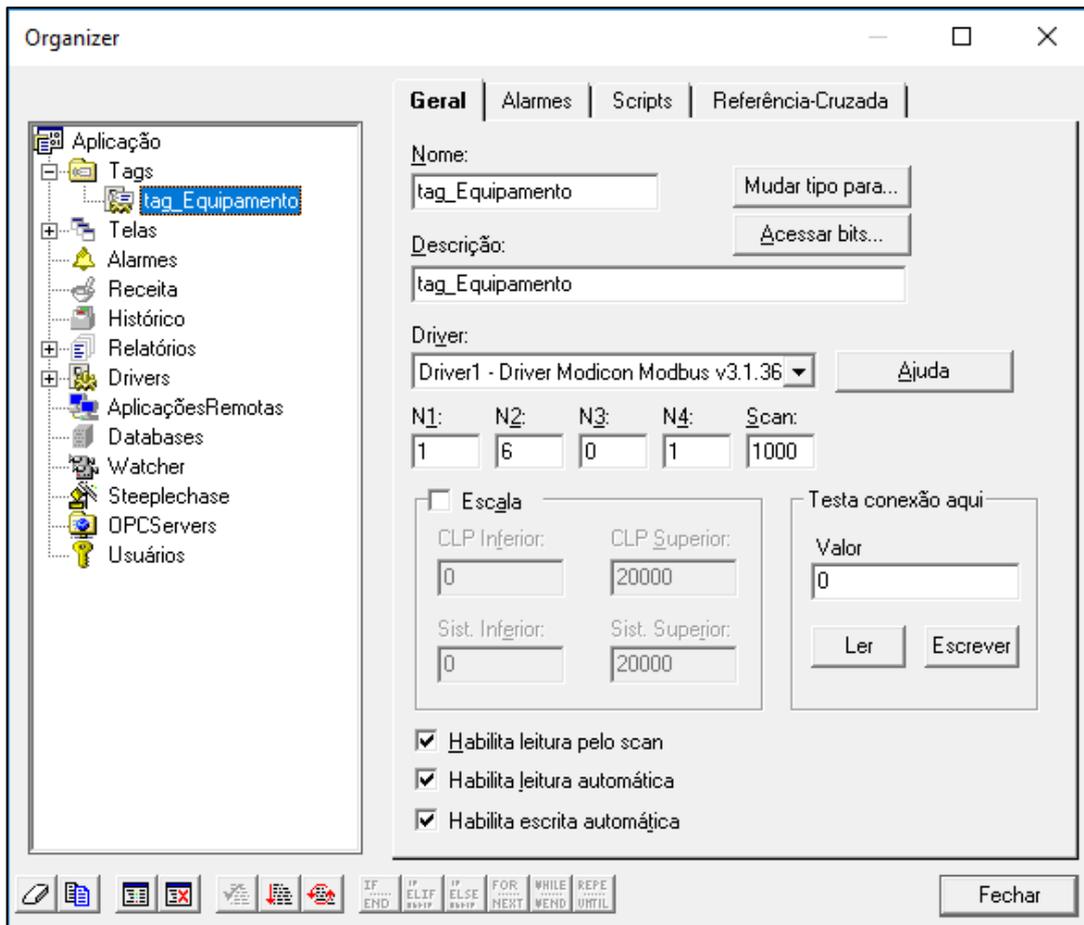
Figura 96 - Comunicação OPC Server (Elipse SCADA) e Simulador MODBUS – Operações MODBUS



Fonte: Autoria própria.

No tutorial foi definido um CLP, com operações binárias e que se encontra no primeiro endereço do CLP.

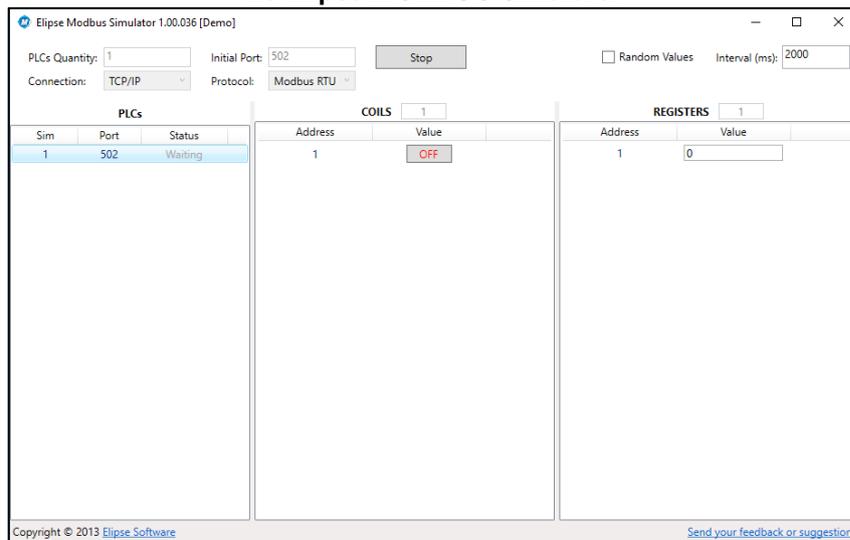
Figura 97 - Comunicação OPC Server (Elipse SCADA) e Simulador MODBUS – Configuração dos parâmetros de comunicação da tag



Fonte: Autoria própria.

Abrindo a aplicação Elipse MODBUS Simulator, são definidos quantos CLP's, porta e protocolos, bem como o número de bobinas e registradores de cada CLP. Para fim, nesse tutorial foi definido um CLP, com uma bobina e um registrador e iniciado o simulador.

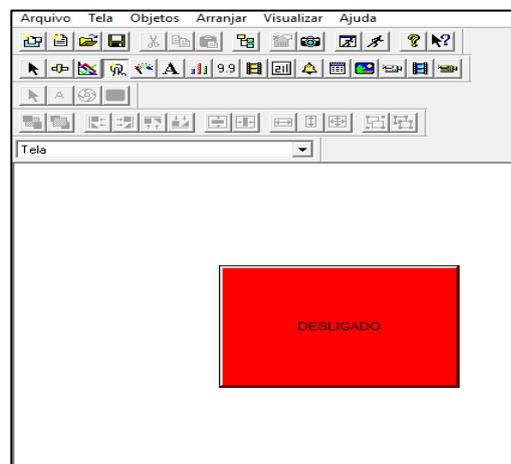
Figura 98 - Comunicação OPC Server (Elipse SCADA) e Simulador MODBUS – Configurações no Elipse MODBUS Simulator



Fonte: Autoria própria.

Executamos a aplicação do OPC Server Elipse SCADA apertando a tecla F10.

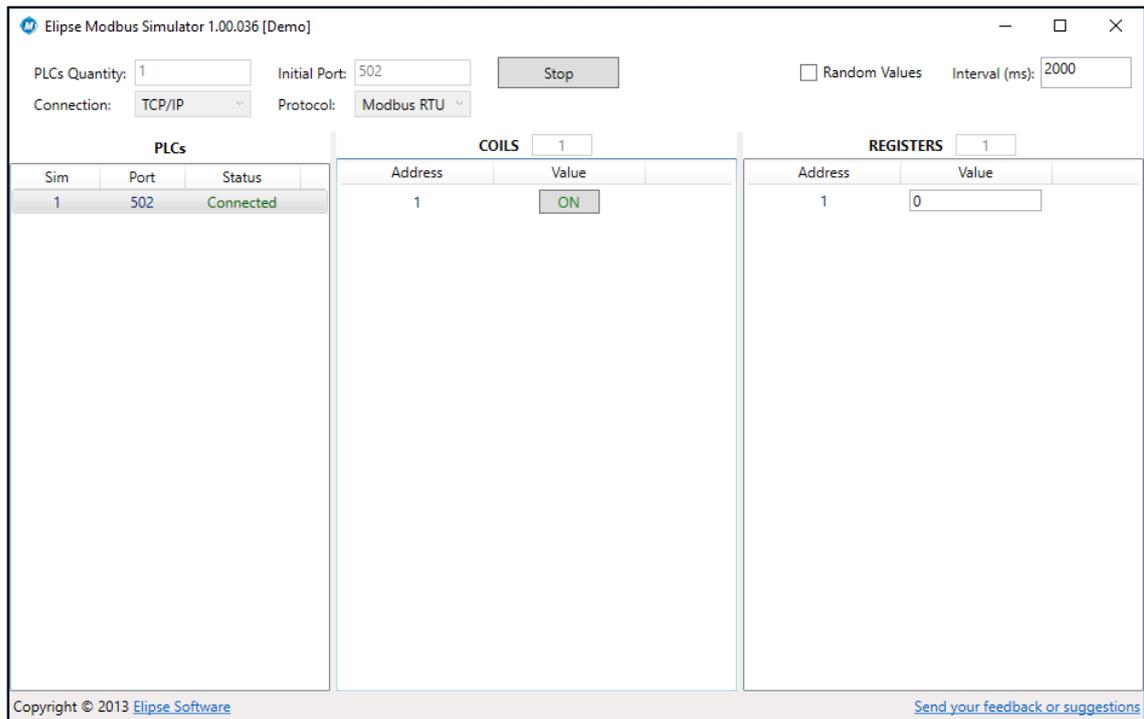
Figura 99 - Comunicação OPC Server (Elipse SCADA) e Simulador MODBUS – Executando a aplicação



Fonte: Autoria própria.

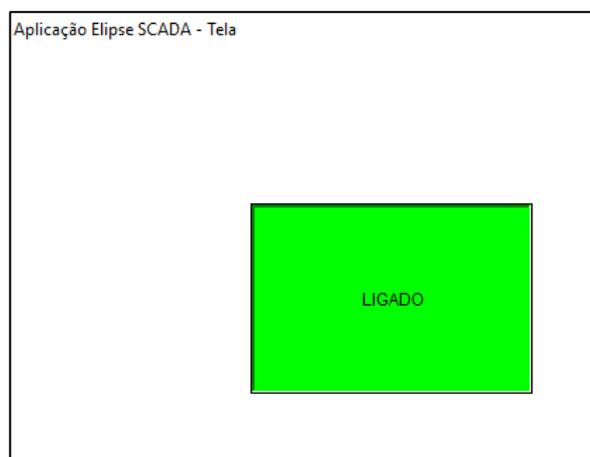
Ao acionarmos a bobina do simulador MODBUS para ON, o status do botão é alterado para ligado na tela do supervisor. Afirmando a ocorrência da comunicação *MODBUS* nesse tutorial.

Figura 100 - Comunicação OPC Server (Elipse SCADA) e Simulador MODBUS – Acionamento da bobina no Simulador



Fonte: Autoria própria.

Figura 101 - Comunicação OPC Server (Elipse SCADA) e Simulador MODBUS – Indicação de acionamento do botão referente ao Simulador



Fonte: Autoria própria.

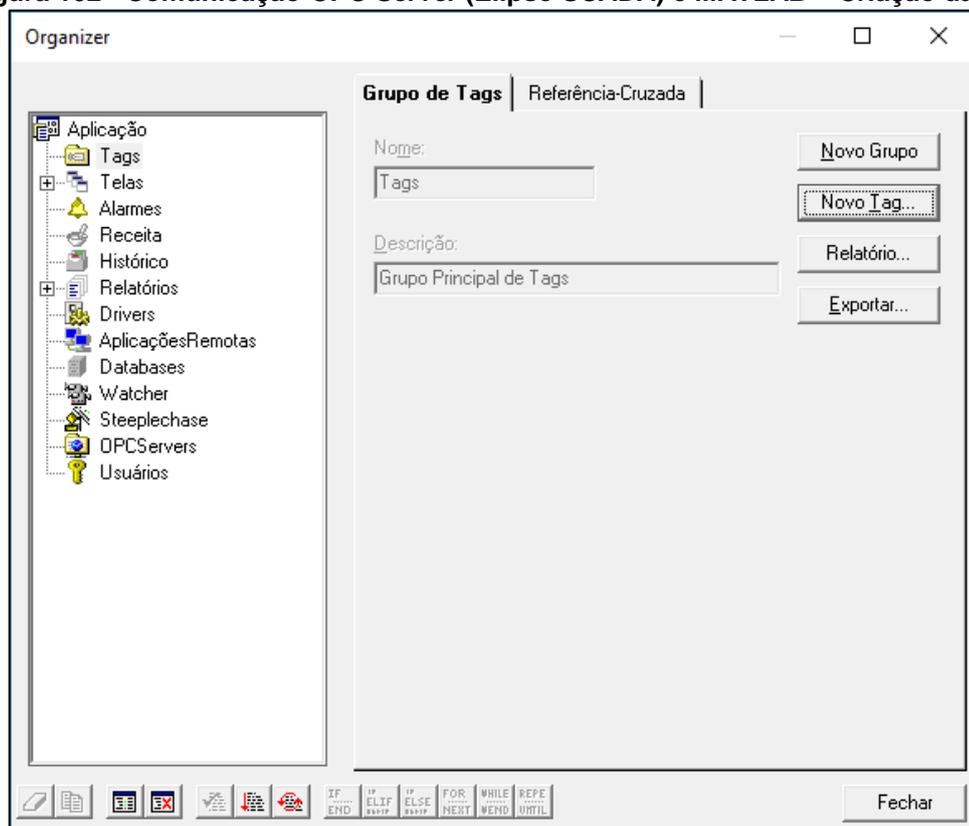
APÊNDICE E – TUTORIAL PARA COMUNICAÇÃO ENTRE OPC SERVER E MATLAB

Esse apêndice apresenta um tutorial detalhado de como realizar a comunicação OPC entre OPC Server e OPC Client, utilizando os softwares Elipse SCADA e a ferramenta computacional MATLAB.

O primeiro passo é a instalação dos programas Elipse SCADA e MATLAB. Para essa comunicação é necessária a utilização da biblioteca OPC toolbox. Ela é disponibilizada nas versões mais completas do MATLAB, senão possui-la, precisará realizar o download no site da MathWorks. Nesse tutorial o sistema supervisor Elipse SCADA é um Servidor e o MATLAB é um Cliente.

Abrindo o programa Elipse SCADA precisamos criar as tags para obtenção dos dados. Para isso acessamos Arquivo > Organizer > Tags > Novo Tag

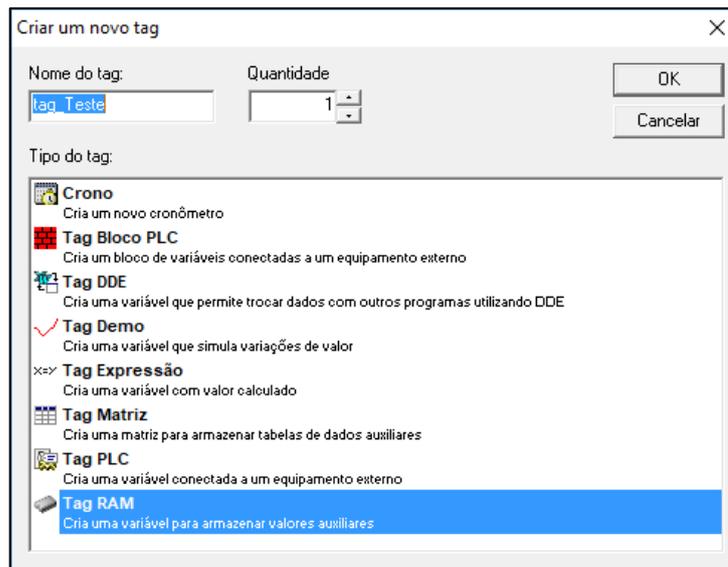
Figura 102 - Comunicação OPC Server (Elipse SCADA) e MATLAB – Criação da tag



Fonte: Autoria própria.

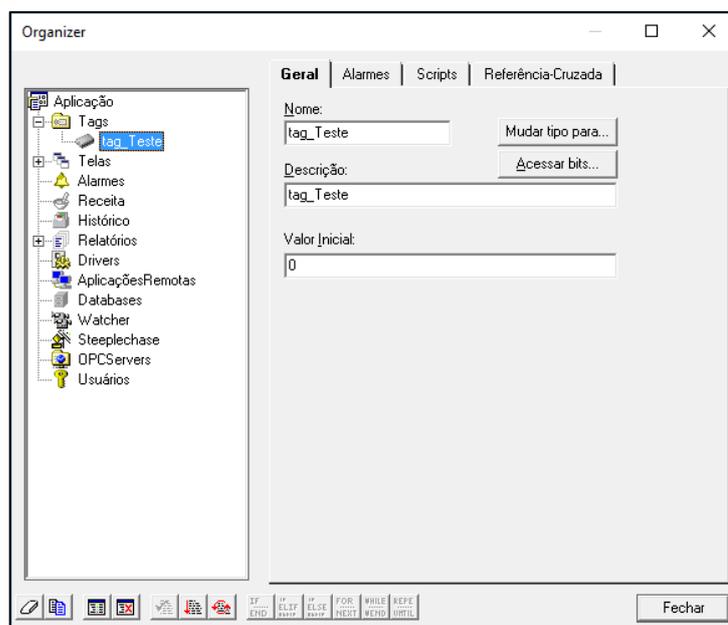
Abrirá uma janela para escolha do tipo da *tag*, onde nesse tutorial foi definida com *tag RAM*, que tem o objetivo de armazenar valores. Após isso, nomeamos ela como *tag_Teste* e damos um OK para criação da *tag*.

Figura 103 - Comunicação OPC Server (Elipse SCADA) e MATLAB – Criação da tag RAM



Fonte: Autoria própria.

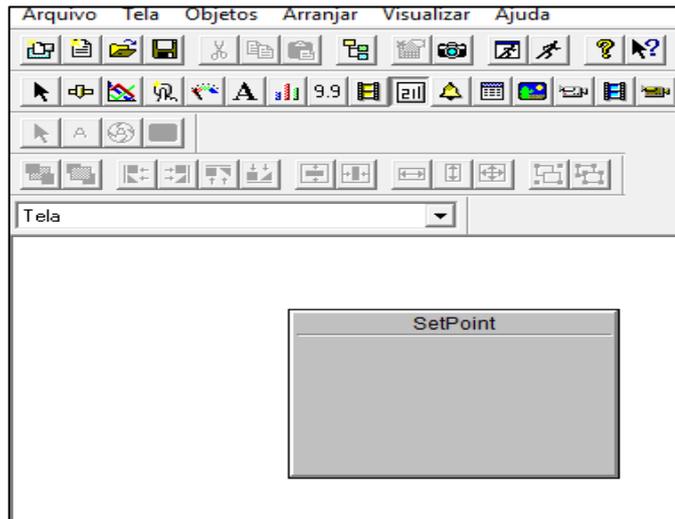
Figura 104 - Comunicação OPC Server (Elipse SCADA) e MATLAB – Visualização da tag criada



Fonte: Autoria própria.

Posteriormente precisamos adicionar o visualizador que será vinculado a *tag*, de modo a gerar os dados. Foi adicionado um *Setpoint*, que são definidos os valores pelo usuário.

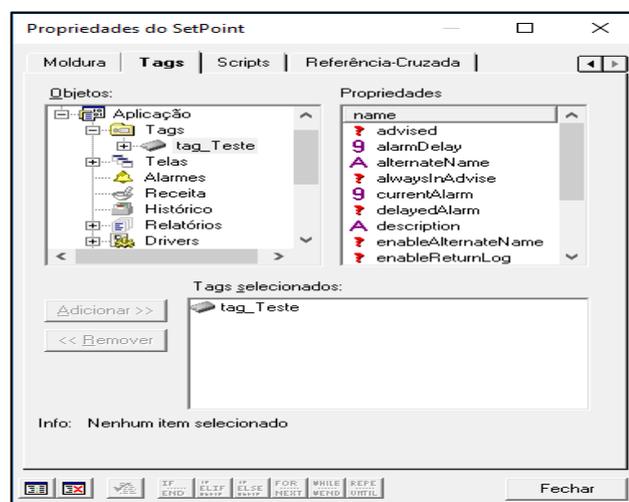
Figura 105 - Comunicação OPC Server (Elipse SCADA) e MATLAB – Visualizador SetPoint



Fonte: Autoria própria.

Assim podemos vincular a *tag_Test* ao visualizador *Setpoint*

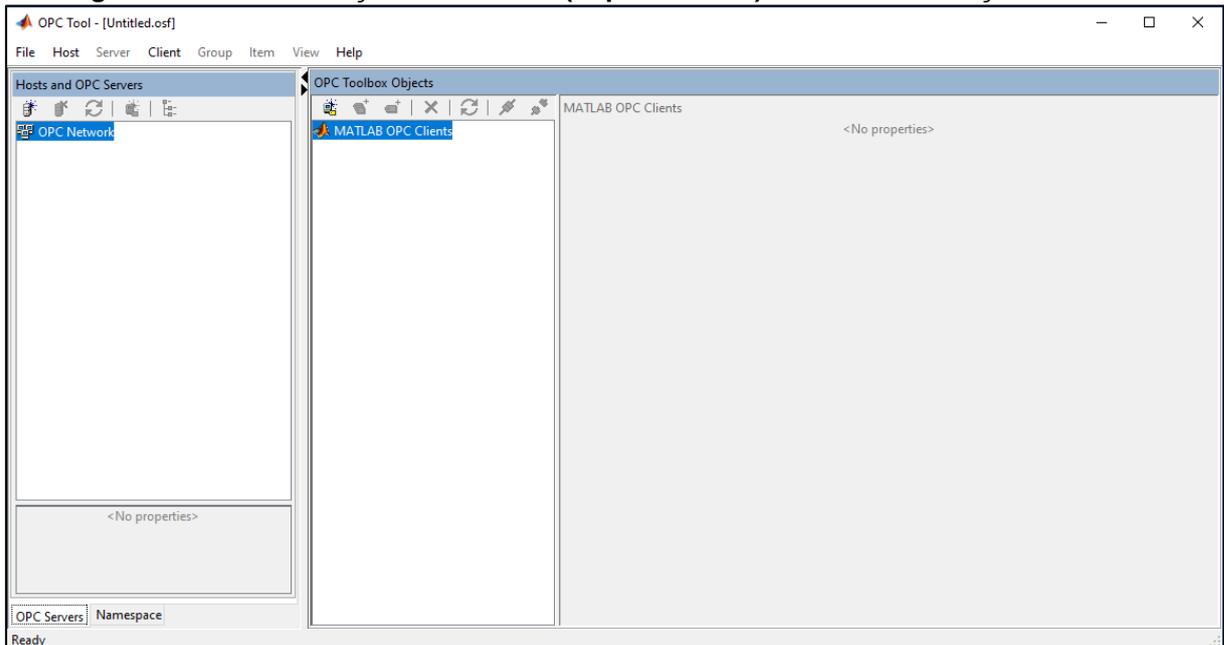
Figura 106 - Comunicação OPC Server (Elipse SCADA) e MATLAB – Vincular a tag com o SetPoint



Fonte: Autoria própria.

Depois de criar e vincular a *tag*, precisamos fazer abrir o *software* MATLAB e digitar na tela de comando a função *opctool* e em seguida abriremos a tela para comunicação.

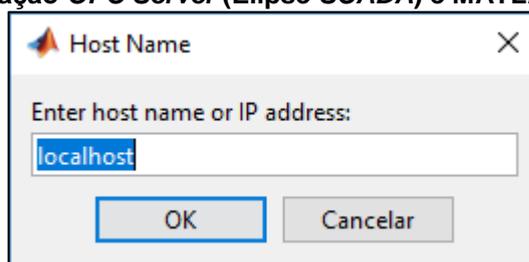
Figura 107 - Comunicação OPC Server (Elipse SCADA) e MATLAB - Função OPC Tool



Fonte: Autoria própria.

Em *Hosts and OPC Server*, clicamos em *Add Host*, com o intuito de pesquisar os *OPC Server* disponíveis. Precisamos digitar o nome do *Host* ou o endereço de *IP*. Nesse tutorial o Servidor está disponível em *localhost*.

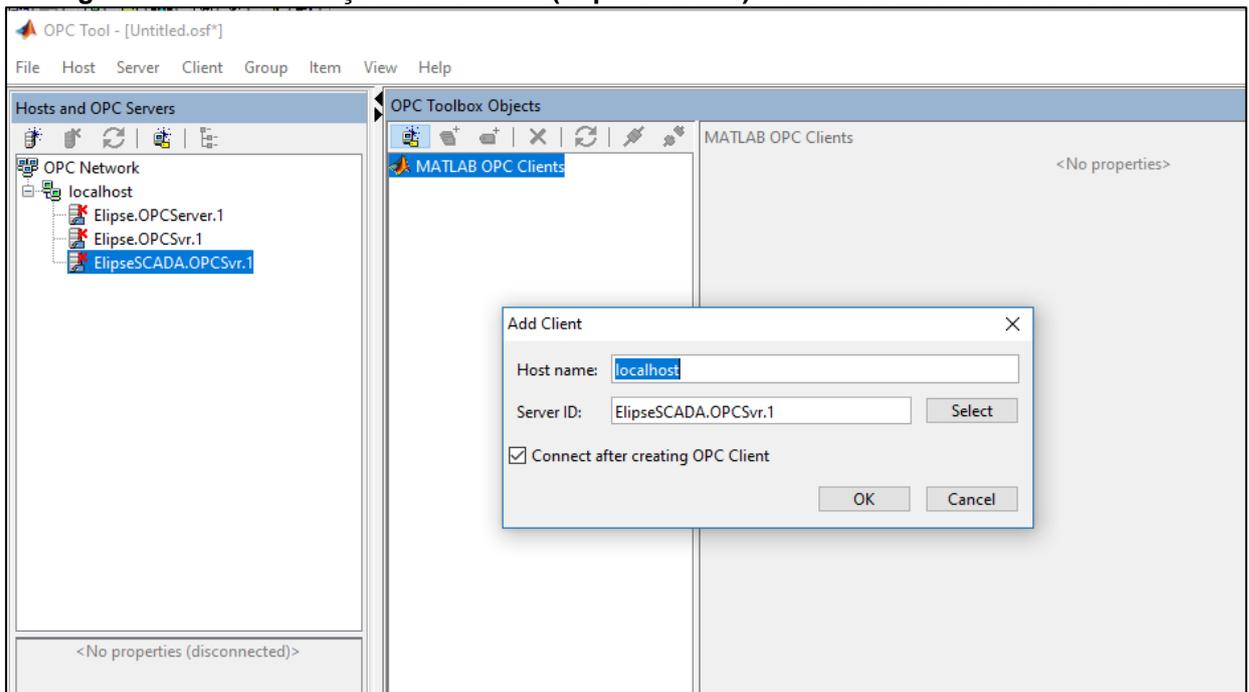
Figura 108 - Comunicação OPC Server (Elipse SCADA) e MATLAB – Adicionando Host



Fonte: Autoria própria.

No campo *Hosts and OPC Server*, aparece os OPC Servers disponíveis, então selecionamos ElipseSCADA.OPCSvr.1. Em OPC Toolbox Objects, clicamos em *Add Client* e depois em OK.

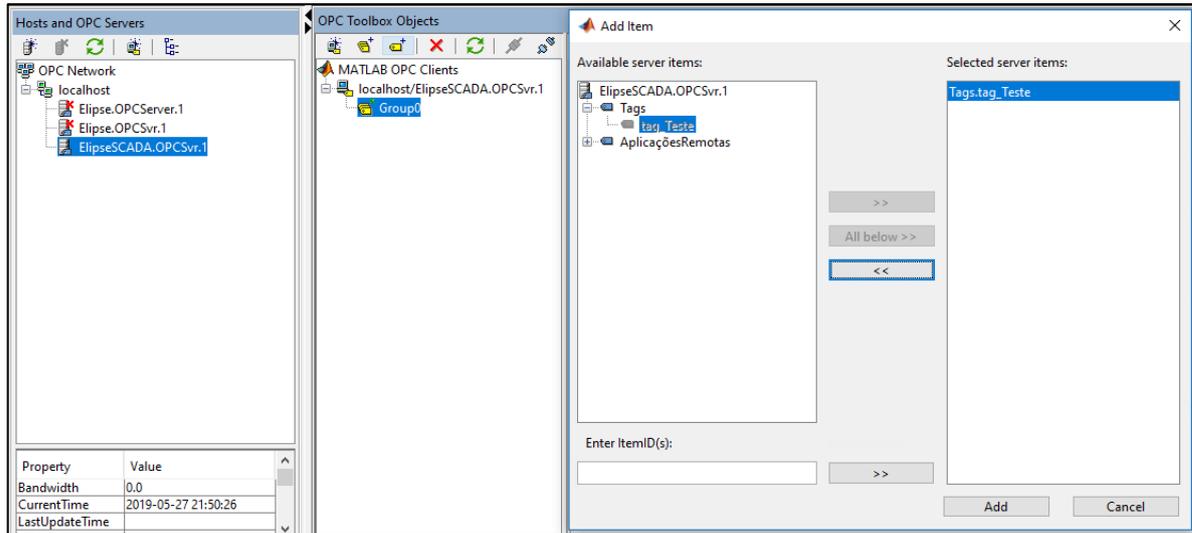
Figura 109 - Comunicação OPC Server (Elipse SCADA) e MATLAB – Adicionando Client



Fonte: Autoria própria.

Posteriormente clicamos no ícone *Add Group > Add Item* e adicionamos a *tag* do OPC Server Elipse SCADA, clicando sobre a *tag_Teste*, depois em *>>* e *Add*, assim foi adicionada a *tag* para o OPC Client MATLAB.

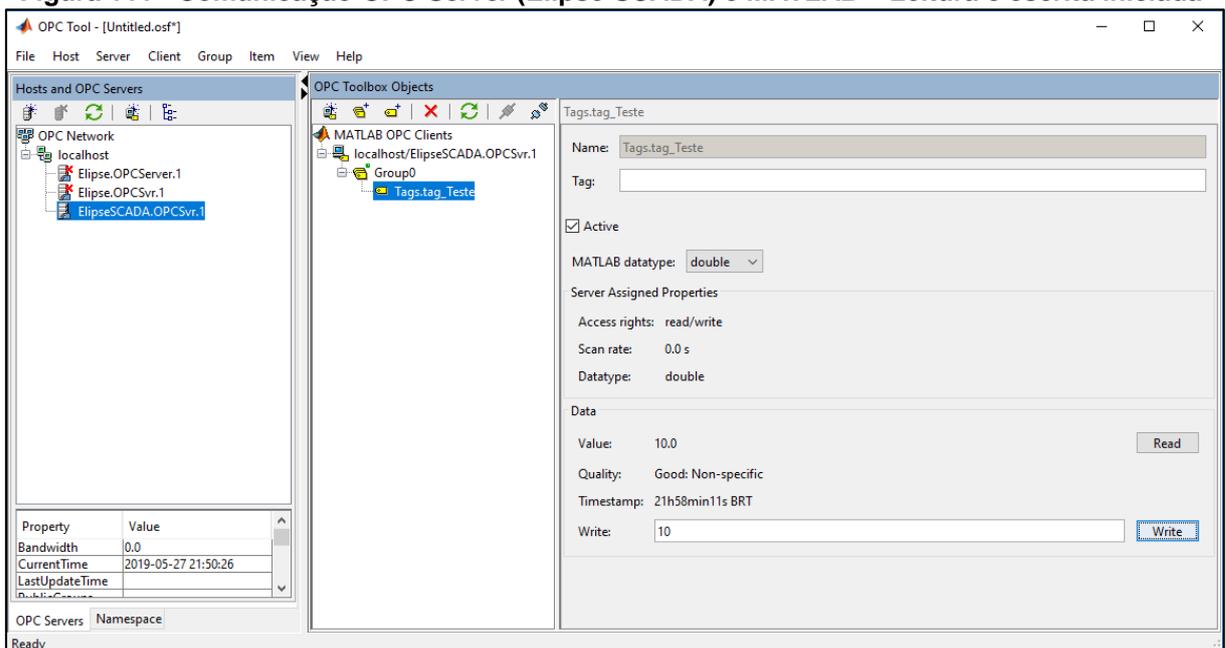
Figura 110 - Comunicação OPC Server (Elipse SCADA) e MATLAB – Importando tags do servidor



Fonte: Autoria própria.

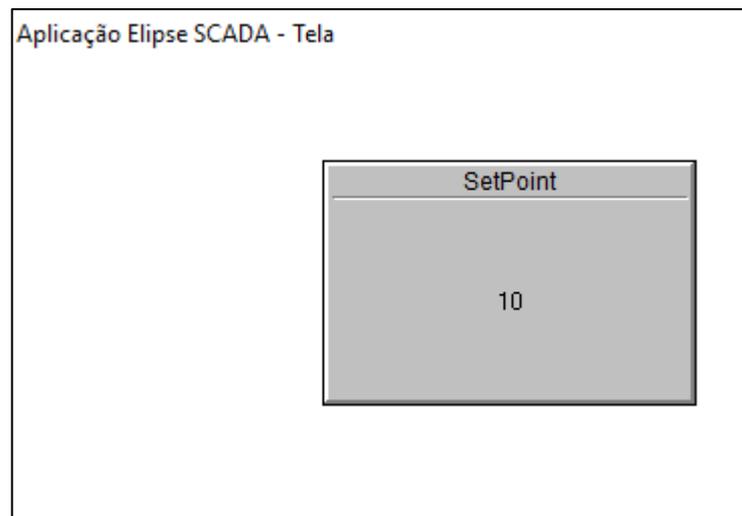
Rodando a aplicação do Elipse SCADA, podemos fazer a escrita colocando valores no MATLAB, quanto a leitura dos dados. Verificando a comunicação OPC utilizando a biblioteca *OPC toolbox*.

Figura 111 - Comunicação OPC Server (Elipse SCADA) e MATLAB – Leitura e escrita iniciada



Fonte: Autoria própria.

Figura 112 - Comunicação OPC Server (Elipse SCADA) e MATLAB – Leitura e escrita do valor do SetPoint



Fonte: Autoria própria.