

Aurum  
EDITORA

# COMUNICAÇÃO EM AUTOMAÇÃO INDUSTRIAL:

INTEGRAÇÃO MODBUS-ETHERNET E IMPLEMENTAÇÃO  
DE SISTEMAS SUPERVISÓRIOS

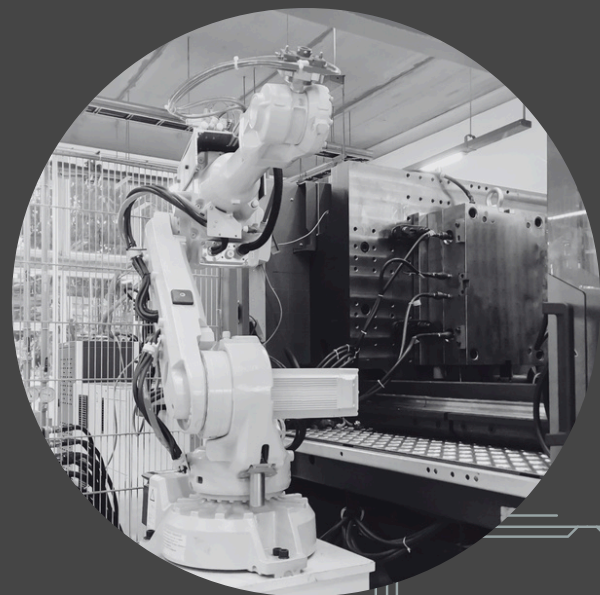


MATHEUS HORTA DE PAULA  
MÁRIO CUPERTINO DA SILVA JÚNIOR  
MARCOS VINÍCIUS LOPES PEREIRA  
MICHEL CARLO RODRIGUES LELES

Aurum  
EDITORA

# COMUNICAÇÃO EM AUTOMAÇÃO INDUSTRIAL:

INTEGRAÇÃO MODBUS-ETHERNET E IMPLEMENTAÇÃO  
DE SISTEMAS SUPERVISÓRIOS



MATHEUS HORTA DE PAULA  
MÁRIO CUPERTINO DA SILVA JÚNIOR  
MARCOS VINÍCIUS LOPES PEREIRA  
MICHEL CARLO RODRIGUES LELES

## **AURUM EDITORA LTDA - 2025**

Curitiba – Paraná - Brasil

### **EDITOR CHEFE**

Lucas Gabriel Vieira Ewers

### **AUTORES DO LIVRO**

Matheus Horta de Paula

Mário Cupertino da Silva Júnior

Marcos Vinícius Lopes Pereira

Michel Carlo Rodrigues Leles

Copyright © Aurum Editora Ltda

Texto Copyright © 2025 Os Autores

Edição Copyright © 2025 Aurum Editora Ltda

### **EDIÇÃO DE TEXTO**

Stefanie Vitoria Garcia de Bastos



Este trabalho está licenciado sob uma licença Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

### **EDIÇÃO DE ARTE**

Aurum Editora Ltda

### **IMAGENS DA CAPA**

Freepik, Canva.

### **BIBLIOTECÁRIA**

Eliane de Freitas Leite

### **ÁREA DE CONHECIMENTO**

Engenharias

A responsabilidade pelo conteúdo, precisão e veracidade dos dados apresentados neste texto é inteiramente do autor, não refletindo necessariamente a posição oficial da Editora. O trabalho pode ser baixado e compartilhado, desde que o crédito seja dado ao autor, mas não é permitida a modificação do conteúdo de qualquer forma ou seu uso para fins comerciais.

A Aurum Editora se compromete a manter a integridade editorial em todas as fases do processo de publicação, prevenindo plágio, dados ou resultados fraudulentos, e assegurando que interesses financeiros não afetem os padrões éticos da publicação. Qualquer suspeita de má conduta científica será verificada com atenção aos princípios éticos e acadêmicos. Todos os manuscritos passaram por uma avaliação duplo-cega, realizada pelos membros do Conselho Editorial, e foram aprovados para publicação com base em critérios de imparcialidade e objetividade acadêmica.

## **CORPO EDITORIAL**

- Adriano Rosa da Silva - Mestre em História Social pela Universidade Federal Fluminense
- Alessandro Sathler Leal da Silva - Doutor em Educação pela Universidade do Estado do Rio de Janeiro
- Alex Lourenço dos Santos - Doutorando em Geografia pela Universidade Federal de Catalão
- Ayla de Jesus Moura - Mestra em Educação Física pela Universidade Federal do Vale do São Francisco
- Camila Aparecida da Silva Albach - Doutoranda em Ciências Sociais Aplicadas pela Universidade Estadual de Ponta Grossa
- Carina Mandler Schmidmeier - Mestranda em Direito pela Pontifícia Universidade Católica do Paraná
- Carolline Nunes Lopes - Mestra em Psicologia pela Universidade Federal do Rio de Janeiro
- Daniel da Rocha Silva - Mestre em Letras pela Universidade Federal de Sergipe
- Daniel Rodrigues de Lima - Mestre em História pela Universidade Federal do Amazonas.
- Fabio José Antonio da Silva - Doutor em Educação Física pela Universidade Estadual de Londrina.
- Fabricio do Nascimento Moreira - Doutorando em Administração pela Universidade Federal do Rio de Janeiro
- Felipe Antônio da Silva - Graduado em Direito pelo Centro Universitário Unihorizontes
- Francisco Welton Machado - Editor Independente - Graduado em Geografia pela Universidade Estadual do Piauí
- Gabriella de Moraes - Doutora em Direito pela Universidade Federal de Minas Gerais
- Gleyson Martins Magalhães Reymão - Mestre Profissional em Propriedade Intelectual e Transferência de Tecnologia para a Inovação pelo Instituto Federal do Pará
- Gustavo Boni Minetto - Mestrando em Educação, Linguagens e Tecnologia pela Universidade Estadual de Goiás
- João Vitor Silva Almeida - Graduado em Gestão de Cooperativas pela Universidade Federal do Tocantins
- José Bruno Martins Leão - Doutor em Sistema Constitucional de Garantia de Direitos pela Instituição Toledo de Ensino
- José Cláudio da Silva Júnior - Mestrando em Ciências da Saúde pela Universidade de Pernambuco
- José Leonardo Diniz de Melo Santos - Mestre em Educação, Culturas e Identidades pela Universidade Federal Rural de Pernambuco



José Marciel Araújo Porcino - Graduado em Pedagogia pela Universidade Federal da Paraíba, UFPB, Brasil.

José Neto de Oliveira Felipe - Doutorando em Ensino de Ciências Exatas - PPGECE - Universidade do Vale do Taquari - UNIVATES, UNIVATES, Brasil.

Luan Brenner da Costa - Editor Independente - Graduado em Enfermagem pela Fundação Herminio Ometto

Lucas Matheus Araujo Bicalho - Mestrando em Historia pela Universidade Estadual de Montes Claros, UNIMONTES, Brasil.

Luciano Victor da Silva Santos - Mestrando em Hotelaria e Turismo pela Universidade Federal de Pernambuco, UFPE, Brasil.

Luzia Eleonora Rohr Balaj - Doutoranda em Música pela Universidade Federal do Estado do Rio de Janeiro

Magno Fernando Almeida Nazaré - Mestre em Educação Profissional e Tecnológica pelo Instituto Federal de Educação, Ciência e Tecnologia do Maranhão

Maickon Willian de Freitas - Mestre em Ciências Biológicas pela Universidade Estadual Paulista Júlio de Mesquita Filho

Maikon Luiz Mirkoski - Mestre Profissional em Matemática em Rede Nacional pela Universidade Estadual de Ponta Grossa

Mailson Moreira dos Santos Gama - Doutorando em História pela Universidade Federal de Minas Gerais

Marcela da Silva Melo - Mestre em Avaliação de Políticas Públicas pela Universidade Federal do Ceará

Marcos Scarpioni - Doutorando em Ciência da Religião pela Universidade Federal de Juiz de Fora

Marilha da Silva Bastos - Mestranda em Educação Brasileira pela Universidade Federal do Ceará

Mario Marcos Lopes - Doutorando em Educação pela Universidade Federal de São Carlos

Mirna Liz da Cruz - Editora Independente - Graduada em Odontologia pela Universidade Federal de Goiás

Newton Ataíde Meira - Mestrando em Desenvolvimento Social pela Universidade Estadual de Montes Claros

Plinio da Silva Andrade - Mestrando em Ciências da Educação pela Universidade Leonardo Da Vinci

Priscila da Silva de Souza Bertotti - Editora Independente - Graduada em Biomedicina pelo Centro Universitário UniOpet

Rafael José Kraisch - Doutorando em Neurociências pela Universidade Federal de Santa Catarina

Ryan Dutra Rodrigues - Editor Independente - Graduado em Psicologia pelo Centro Universitário das Faculdades Metropolitanas Unidas



Salatiel Elias de Oliveira - Doutor em Apostilamento de Reconhecimento de Título pela Universidade do Oeste Paulista

Sebastião Lacerda de Lima Filho - Doutorando em Medicina Translacional pela Universidade Federal do Ceará

Silvio de Almeida Junior - Doutor em Promoção de Saúde pela Universidade de Franca

Swelen Freitas Gabarron Peralta - Doutoranda em Educação pela Universidade Tuiuti do Paraná

Talita Benedcta Santos Künast - Doutoranda em Biodiversidade e Biotecnologia pela Universidade Federal de Mato Grosso

Tályta Carine da Silva Saraiva - Mestra em Agronomia pela Universidade Federal do Piauí

Thiago Giordano de Souza Siqueira - Doutor em Ciência da Informação pela Universidade Estadual Paulista Júlio de Mesquita Filho

Thiago Silva Prado - Doutor em Educação pela Universidade Estadual de Maringá

Vinicius Valim Pereira - Doutor em Zootecnia pela Universidade Estadual de Maringá, UEM, Brasil.

Wilson Moura - Doutor em Psicologia pela Christian Business School

Yohans de Oliveira Esteves - Doutor em Psicologia pela Universidade Salgado de Oliveira



## Dados Internacionais de Catalogação na Publicação (CIP) (Câmara Brasileira do Livro, SP, Brasil)

Comunicação em automação industrial [livro eletrônico] : integração Modbus-Ethernet e implementação e sistemas supervisórios / Matheus Horta de Paula...[et al.]. --  
-- Curitiba, PR : Aurum Editora, 2026.  
PDF

Outros autores: Mário Cupertino da Silva Júnior,  
Marcos Vinícius Lopes Pereira, Michel Carlo  
Rodrigues Leles.

Bibliografia.

ISBN 978-65-83849-78-6

1. Arquitetura de rede de computador  
2. Automação industrial 3. Comunicação  
4. Engenharia 5. Tecnologia I. Paula, Matheus Horta  
de. II. Silva Júnior, Mário Cupertino da.  
III. Pereira, Marcos Vinícius Lopes. IV. Leles,  
Michel Carlo Rodrigues.

26-346736.0

CDD-670.427

### Índices para catálogo sistemático:

1. Automação industrial : Tecnologia 670.427

Eliane de Freitas Leite - Bibliotecária - CRB 8/8415

**DOI:** 10.63330/livroautoral412026-

**Aurum Editora Ltda**

CNPJ: 589029480001-12

[contato@aurumeditora.com](mailto:contato@aurumeditora.com)

(41) 98792-9544

Curitiba - Paraná



## AUTORES

### **Matheus Horta de Paula**

Engenheiro Mecatrônico formado pela Universidade Federal de São João del-Rei (UFSJ). Participou de vários projetos de iniciação científica envolvendo Automação de Processos e Redes Industriais. Foi Bolsista IC-IT CNPq (Chamada PIBITI - FUNTTEL). Atua nas áreas de robótica, automação e redes industriais de computadores. Também possui vasta experiência de trabalho na área de informática. Atualmente é Senior Development Analyst na Solinftec onde dedica-se à criação de soluções inteligentes para a Agricultura, unindo experiências prévias com sistemas de informática e a formação tecnológica.

E-mail: matheushp@live.com

### **Mário Cupertino da Silva Júnior**

Possui graduação em Engenharia de Controle e Automação pela Pontifícia Universidade Católica de Minas Gerais (2004), mestrado e doutorado em Engenharia Agrícola pela Universidade Federal de Viçosa (2006 e 2010), com estágio de doutoramento na Universidade de Valladolid, em Valladolid, Espanha (2008-2009). Atualmente, está se especializando em Energias Renováveis pela UFMG e é professor Associado na Universidade Federal de Sergipe. Tem experiência e interesse em Automação de Processos e Energia, atuando principalmente nos seguintes temas: Automação Agrícola, Energias Renováveis e Processamento Digital de Imagens.

E-mail: mariocupertino@academico.ufs.br

### **Marcos Vinícius Lopes Pereira**

Possui graduação em Engenharia de Controle e Automação (2003), mestrado em Engenharia Elétrica (2006) e doutorado em Administração (2020), todas essas obtidas na Universidade Federal de Minas Gerais (UFMG). Atualmente é professor adjunto Universidade Federal de São João del-Rei (UFSJ), Campus Alto Paraopeba, atuando, principalmente, nos cursos de Engenharia Mecatrônica e Engenharia de Telecomunicações. Tem experiência na área de Engenharia de Controle e Automação, e possui as seguintes áreas de interesse: controle de processos industriais, processamento de sinais, identificação de sistemas dinâmicos, otimização de sistemas e inteligência computacional (redes neurais e sistemas nebulosos).

E-mail: marcos.vinicius@ufsj.edu.br

### **Michel Carlo Rodrigues Leles**

Professor Associado da Universidade Federal de São João del-Rei, Campus Alto Paraopeba, em que está lotado no Departamento de Tecnologia (DTECH/CAP/UFSJ). Engenheiro de Controle e Automação pela Universidade Federal de Minas Gerais. Possui Mestrado e Doutorado em Engenharia Elétrica pela mesma instituição. Docente nos Programas de Pós Graduação em: i) Ciência da Computação (PPGCC/UFSJ); e ii) Pesquisa Operacional (PPGPO/ITA-UNIFESP). Tem interesse na grande área de Data Science, especialmente na análise de séries temporais, processamento de linguagem natural, e finanças computacionais.

E-mail: mleles@ufsj.edu.br



## RESUMO

A comunicação entre dispositivos industriais constitui um dos pilares da automação moderna, possibilitando a integração entre sensores, controladores, sistemas supervisórios e infraestruturas computacionais. Protocolos como Modbus e redes baseadas em Ethernet são amplamente utilizados em ambientes industriais devido à sua robustez, simplicidade e ampla compatibilidade entre equipamentos de diferentes fabricantes. Este livro apresenta o desenvolvimento e a implementação de uma plataforma didática destinada ao estudo, experimentação e demonstração de comunicação entre dispositivos industriais utilizando os padrões Modbus e Ethernet. A proposta consiste na construção de uma estrutura física contendo equipamentos industriais reais — incluindo um FieldLogger, um controlador de processos N1200 e dispositivos de rede — interligados por barramento Modbus RS-485 e integrados à rede Ethernet por meio de um *gateway* capaz de converter requisições entre os dois protocolos. Além da descrição detalhada da construção física da plataforma, são abordados os aspectos de configuração lógica dos dispositivos, incluindo a parametrização da rede Modbus, a criação de variáveis e alarmes, o registro de dados e a implementação de serviços de rede como HTTP, FTP e SMTP. O livro também apresenta a integração com sistemas supervisórios, utilizando *softwares* SCADA para monitoramento e controle dos dispositivos. O módulo ou plataforma proposta, demonstra na prática conceitos fundamentais de redes industriais, comunicação entre dispositivos e integração de sistemas de automação, constituindo uma ferramenta valiosa para ensino, pesquisa e desenvolvimento tecnológico na área de automação industrial.

**Palavras-chave:** Arquitetura de rede; Gateway de protocolo; Aquisição de dados industriais; Integração de dispositivos; Monitoramento de processos.



## ABSTRACT

Communication between industrial devices constitutes one of the fundamental pillars of modern automation, enabling the integration of sensors, controllers, supervisory systems, and computational infrastructures. Protocols such as Modbus and Ethernet-based networks are widely used in industrial environments due to their robustness, simplicity, and broad compatibility among devices from different manufacturers. This book presents the development and implementation of an experimental platform designed for the study, experimentation, and demonstration of communication between industrial devices using the Modbus and Ethernet standards. The proposed platform consists of a physical structure incorporating real industrial equipment—including a FieldLogger, an N1200 process controller, and network devices—interconnected through a Modbus RS-485 bus and integrated with an Ethernet network by means of a gateway capable of converting requests between the two communication protocols.

In addition to providing a detailed description of the physical construction of the platform, the book addresses the logical configuration aspects of the devices, including Modbus network parameterization, creation of variables and alarms, data logging, and the implementation of network services such as HTTP, FTP, and SMTP. The integration with supervisory systems is also presented, using SCADA software for monitoring and control of the devices connected to the network. The proposed platform demonstrates, in a practical manner, fundamental concepts of industrial networks, device communication, and automation system integration, constituting a valuable tool for teaching, research, and technological development in the field of industrial automation.

**Keywords:** Net architecture; Protocol gateway; Industrial data acquisition; Device integration; Process monitoring.



## SUMÁRIO

APRESENTAÇÃO.....	13
1 INTRODUÇÃO.....	14
2 PARTE FÍSICA 1 – ESTRUTURA.....	17
2.1 MÓDULO DE MADEIRA.....	17
2.2 EQUIPAMENTOS INTERNOS.....	17
2.3 CABEAMENTO INTERNO – ELÉTRICA.....	19
2.4 BARRAMENTO MODBUS.....	21
2.5 ADESIVOS – ACABAMENTO E LAYOUT.....	22
3 PARTE FÍSICA 2 – REDE.....	24
3.1 TOPOLOGIA DE REDE PONTO A PONTO.....	24
3.2 REDE CRIADA PARA TESTE DO MÓDULO (PLATAFORMA).....	24
4 PARTE LÓGICA 1 – PROGRAMAS.....	26
4.1 FLCONFIG – GERAL.....	26
4.2 FLCONFIG – MODBUS.....	26
4.3 FLCONFIG - ANALÓGICAS.....	28
4.4 FLCONFIG - DIGITAIS E RELÉS.....	29
4.5 FLCONFIG - CANAIS REMOTOS.....	30
4.6 FLCONFIG – CANAIS VIRTUAIS.....	33
4.7 FLCONFIG - ALARMES.....	34
4.8 FLCONFIG – REGISTROS.....	36
4.9 NCONFIG – GERAL.....	37
4.10 RSLINX - SERVIDOR OPC.....	38
5 PARTE LÓGICA 2 – SERVIÇOS DE REDE.....	49
5.1 SMTP – STUNNEL.....	49
5.2 FTP – FILEZILLA.....	51
5.3 HTTP.....	53
5.4 MODBUS TCP.....	54
6 PARTE LÓGICA 3 – INTEGRAÇÃO.....	56
6.1 SISTEMA SUPERVISÓRIO.....	56
6.2 DRIVER MODBUS DO ELIPSE SCADA.....	58
6.3 EXEMPLOS.....	61
6.3.1 Primeiro exemplo: ler dado escravo Modbus.....	62
6.3.2 Segundo exemplo: receber valor de uma entrada analógica do FieldLogger.....	66



6.3.3 Terceiro exemplo: alterar estado de um relé do N1200.....	68
6.4 ACESSO REMOTO.....	72
7 CONSIDERAÇÕES FINAIS.....	75
REFERÊNCIAS BIBLIOGRÁFICAS.....	76



## LISTA DE FIGURAS

Figura 1 - Vistas com as medidas do módulo e uma vista isométrica.....	17
Figura 2 – FieldLogger da empresa Novus.....	18
Figura 3 – Controlador N1200 da empresa Novus.....	18
Figura 4 - Switch Ethernet da empresa Allen-Bradley.....	19
Figura 5 - Fonte de 24 V da empresa Allen-Bradley.....	19
Figura 6 - Esquema elétrico interno.....	20
Figura 7 - Foto cabeamento sendo montado.....	21
Figura 8 - Barramento Modbus em placa de circuito impresso.....	21
Figura 9 - Barramento Modbus.....	21
Figura 10 - Layout do painel frontal do módulo.....	23
Figura 11 - Layouts laterais esquerdo e direito do módulo.....	23
Figura 12 - Topologia Ponto a ponto entre o FieldLogger e o computador.....	24
Figura 13 - Conector RJ45 Fêmea utilizado na extensão Ethernet.....	24
Figura 14 - Rede criada para teste do (módulo).....	25
Figura 15 - FLConfig: tela inicial.....	26
Figura 16 - FLConfig: interface RS485.....	28
Figura 17 - FLConfig: canais analógicos.....	28
Figura 18 - FLConfig: canais digitais.....	29
Figura 19 - Divisor de tensão.....	30
Figura 20 - FLConfig: canais remotos.....	31
Figura 21 - Editando Tags.....	32
Figura 22 - Conexões e tempos.....	33
Figura 23 - FLConfig: canais virtuais.....	33
Figura 24 - Criando uma tag.....	34
Figura 25 - FLConfig: alarmes.....	35
Figura 26 - Criando alarme.....	36
Figura 27 - FLConfig: registros.....	37
Figura 28 - NConfig: instalação.....	37
Figura 29 - RSLinx: adicionar OPC.....	38
Figura 30 - RSLinx: adicionar tópicos a cada CLP.....	39
Figura 31 - Captura RSLinx e supervisão.....	39
Figura 32 - Adicionar usuário.....	40
Figura 33 - Novo usuário.....	40
Figura 34 - Novo grupo.....	41
Figura 35 – Adicionar usuário ao grupo criado.....	42
Figura 36 - Configurações DCOM.....	42
Figura 37 - Aba <i>Security</i> .....	43
Figura 38 - Adicionando o grupo OPCUsers.....	43
Figura 39 - Permissões de padrão.....	44
Figura 40 - Acertando as permissões.....	44
Figura 41 - Usuário e senha configurados.....	45
Figura 42 - Propriedades DCOM.....	46
Figura 43 - Segurança COM.....	46
Figura 44 - Central de rede e compartilhamento.....	47
Figura 45 - Ativar descoberta da rede.....	48
Figura 46 - Alterar política de acesso.....	48
Figura 47 - SMTP: o caminho dos e-mails.....	49
Figura 48 - OpenSSL: criando a certificação para criptografia.....	50



Figura 49 - SMTP: configuração do Stunnel.....	50
Figura 50 - FTP: serviços.....	51
Figura 51 - FTP: FieldLogger como servidor.....	52
Figura 52 - FTP: FieldLogger como cliente; à esquerda a captura pelo Wireshark e à direita o registro do FileZilla.....	53
Figura 53 – FLConfig: HTTP.....	53
Figura 54 - Modbus TCP: formato do datagrama.....	54
Figura 55 - Modbus TCP: caminho dos dados.....	55
Figura 56 – Tela do Novus SuperView.....	56
Figura 57 -Tela do ScadaBr.....	57
Figura 58 – Tela do Elipse Scada.....	58
Figura 59 - Download do driver Modbus no site do Elipse Scada.....	58
Figura 60 - Adicionando driver Modbus.dll.....	59
Figura 61 - Menu de configuração do driver Modbus.....	59
Figura 62 - Aba de configuração Modbus.....	60
Figura 63 - Aba de configuração <i>Operations</i> .....	60
Figura 64 - Aba de configuração <i>Setup</i> .....	61
Figura 65 - Aba de configuração Ethernet.....	61
Figura 66 - Adicionar <i>tags</i> .....	62
Figura 67 - Adicionar <i>tags</i> .....	63
Figura 68 - <i>Tag RHT1_temp</i> .....	63
Figura 69 - Gauge Temp_RHT1.....	65
Figura 70 - Aba <i>Tags</i> do <i>Gauge</i> .....	66
Figura 71 - <i>Tag</i> da entrada analógica 8 do FieldLogger.....	67
Figura 72 - Adicionando a <i>Tag</i> ao <i>Display</i> .....	68
Figura 73 - Criando a operação de estado do relé.....	69
Figura 74 - Criando o <i>tag</i> de estado do relé.....	69
Figura 75 - Criando o botão de estado do relé.....	70
Figura 76 - Criando o botão de estado do relé.....	71
Figura 77 - Adicionando o <i>tag</i> ao botão de estado do relé.....	71
Figura 78 - VNC: servidor e cliente.....	73
Figura 79 - <i>TeamViewer</i> .....	74



## APRESENTAÇÃO

A automação industrial moderna depende fortemente da capacidade de comunicação entre diferentes dispositivos e sistemas. Sensores, controladores lógicos programáveis (CLPs), registradores de dados, atuadores e sistemas supervisórios precisam trocar informações continuamente para garantir o funcionamento eficiente, seguro e confiável dos processos industriais.

Nesse cenário, os protocolos de comunicação industrial desempenham um papel fundamental ao padronizar a forma como os dados são transmitidos entre os equipamentos. Entre os diversos protocolos existentes, o Modbus permanece como um dos mais utilizados, principalmente em aplicações de controle e monitoramento distribuído.

Com o avanço das redes de comunicação e a crescente utilização de infraestruturas baseadas em Ethernet, tornou-se cada vez mais comum a integração entre redes industriais tradicionais e redes de computadores. Essa integração possibilita o monitoramento remoto de processos, a centralização de dados operacionais e a implementação de sistemas supervisórios avançados.

Este livro apresenta o desenvolvimento de um módulo experimental projetado para estudar e demonstrar a comunicação entre dispositivos industriais utilizando os protocolos Modbus e Ethernet. A proposta deste módulo é oferecer um ambiente didático que permita visualizar e compreender, de forma prática, o funcionamento das redes industriais e suas diferentes arquiteturas.

A obra está estruturada de forma progressiva, iniciando pela descrição da construção física do módulo, passando pela configuração lógica dos dispositivos e chegando à integração com sistemas supervisórios e mecanismos de acesso remoto.

Dessa forma, o leitor poderá acompanhar todas as etapas envolvidas no desenvolvimento de um sistema de comunicação industrial completo, desde sua concepção até sua operação em ambiente de rede.



# 1 INTRODUÇÃO

A automação industrial moderna depende fundamentalmente da capacidade de comunicação entre dispositivos distribuídos em sistemas de controle e monitoramento. Sensores, controladores, atuadores e sistemas supervisórios precisam trocar informações continuamente para garantir a operação eficiente, segura e confiável de processos industriais. Nesse contexto, os protocolos de comunicação industrial desempenham papel central ao definir as regras que permitem a interoperabilidade entre equipamentos de diferentes fabricantes e tecnologias (KOCAMUFTUOGLU, 2022).

Historicamente, os sistemas industriais eram baseados em arquiteturas centralizadas, nas quais os dispositivos de campo estavam diretamente conectados a controladores locais por meio de sinais analógicos ou digitais. Com o avanço das tecnologias de informação e comunicação, esses sistemas evoluíram para arquiteturas distribuídas, nas quais múltiplos dispositivos inteligentes são interligados por redes industriais capazes de transmitir dados em tempo real. Essa evolução permitiu maior flexibilidade, escalabilidade e capacidade de monitoramento remoto dos processos industriais (ZURAWSKI, 2015).

Entre os diversos protocolos utilizados na automação industrial, o Modbus destaca-se como um dos mais difundidos. Desenvolvido originalmente pela empresa Modicon em 1979 para comunicação entre controladores lógicos programáveis (CLPs), o Modbus tornou-se um padrão amplamente adotado devido à sua simplicidade, robustez e ao fato de ser um protocolo aberto e livre de royalties. Atualmente, ele é utilizado em uma grande variedade de aplicações industriais, incluindo sistemas de controle, aquisição de dados e integração de dispositivos eletrônicos industriais (MODBUS ORGANIZATION, 2012).

O protocolo Modbus opera tradicionalmente em redes seriais utilizando interfaces físicas como RS-232 ou RS-485, adotando uma arquitetura de comunicação mestre-escravo. Nesse modelo, um dispositivo mestre inicia todas as requisições de comunicação na rede, enquanto os dispositivos escravos respondem às solicitações enviadas pelo mestre. Essa arquitetura permite a comunicação entre múltiplos dispositivos conectados a um mesmo barramento, tornando o protocolo adequado para aplicações industriais distribuídas (ENCODER PRODUCTS COMPANY, 2019).

Com a evolução das redes de computadores e a crescente adoção da tecnologia Ethernet no ambiente industrial, surgiu uma nova abordagem de comunicação baseada no encapsulamento de mensagens Modbus em pacotes TCP/IP. Esse mecanismo, conhecido como Modbus TCP, permite que dispositivos industriais utilizem infraestruturas de rede baseadas em Ethernet para comunicação de dados, possibilitando a integração entre redes industriais e redes de tecnologia da informação. A utilização do Modbus TCP facilita a interconexão de equipamentos industriais com sistemas supervisórios, servidores de dados e aplicações de monitoramento remoto (ACROMAG, 2004).

A integração entre redes Modbus e redes Ethernet representa um dos pilares da transformação digital na indústria e da chamada Indústria 4.0, na qual dispositivos físicos e sistemas computacionais são integrados em arquiteturas ciberfísicas capazes de coletar, processar e analisar grandes volumes de dados

operacionais. Nesse cenário, a comunicação eficiente entre dispositivos torna-se essencial para viabilizar aplicações como monitoramento remoto, manutenção preditiva e otimização de processos produtivos. Estudos recentes demonstram que protocolos tradicionais como Modbus continuam desempenhando papel relevante nesse contexto, especialmente quando integrados a tecnologias modernas de rede e Internet das Coisas Industrial (IIoT) (NIKOLAJEW & EICHELBERGER, 2024).

Entretanto, a integração entre diferentes protocolos e tecnologias de comunicação ainda representa um desafio importante no desenvolvimento de sistemas de automação industrial. Muitas instalações industriais utilizam equipamentos que operam em redes seriais, enquanto novos sistemas são desenvolvidos utilizando infraestruturas Ethernet baseadas em TCP/IP. A coexistência dessas tecnologias exige o uso de dispositivos intermediários, conhecidos como *gateways* de protocolo, responsáveis por traduzir e encaminhar mensagens entre diferentes redes de comunicação (GAITAN, 2026).

Diversos estudos têm investigado arquiteturas capazes de integrar redes seriais baseadas em Modbus com redes Ethernet industriais, demonstrando que essa abordagem permite ampliar significativamente a capacidade de monitoramento e controle dos sistemas industriais. Sistemas de aquisição de dados baseados em Modbus sobre Ethernet, por exemplo, possibilitam a coleta e armazenamento de variáveis de processo provenientes de dispositivos industriais distribuídos, permitindo sua visualização e análise por meio de aplicações computacionais e sistemas supervisórios (LE et al., 2024).

Além da integração entre protocolos, outro aspecto fundamental nos sistemas modernos de automação industrial é a utilização de sistemas supervisórios, também conhecidos como sistemas SCADA (*Supervisory Control and Data Acquisition*). Esses sistemas permitem o monitoramento em tempo real de variáveis de processo, a visualização gráfica de estados operacionais e o envio de comandos para dispositivos de campo. A utilização de sistemas supervisórios integrados às redes industriais amplia significativamente a capacidade de controle e análise dos processos produtivos.

Diante desse contexto, torna-se evidente a importância de módulos ou plataformas experimentais que permitam o estudo prático das tecnologias de comunicação industrial. Embora muitos conceitos possam ser abordados teoricamente, a compreensão completa do funcionamento das redes industriais exige a interação direta com equipamentos reais, permitindo observar o comportamento da rede, analisar o tráfego de dados e experimentar diferentes configurações de comunicação.

Nesse sentido, o presente livro apresenta o desenvolvimento de um módulo experimental destinado ao estudo da comunicação entre dispositivos industriais utilizando os protocolos Modbus e Ethernet. O módulo foi projetado para integrar dispositivos de automação industrial em uma arquitetura de rede capaz de demonstrar diferentes formas de comunicação, incluindo redes seriais baseadas em Modbus RTU, redes Ethernet utilizando Modbus TCP e a integração dessas redes com sistemas supervisórios. A proposta do módulo é fornecer um ambiente didático e experimental que permita compreender, de forma prática, os

princípios de funcionamento das redes industriais e das arquiteturas de comunicação utilizadas em sistemas de automação. Além disso, o módulo possibilita a realização de experimentos envolvendo aquisição de dados, controle de dispositivos, integração de sensores e implementação de serviços de rede para acesso remoto e monitoramento.

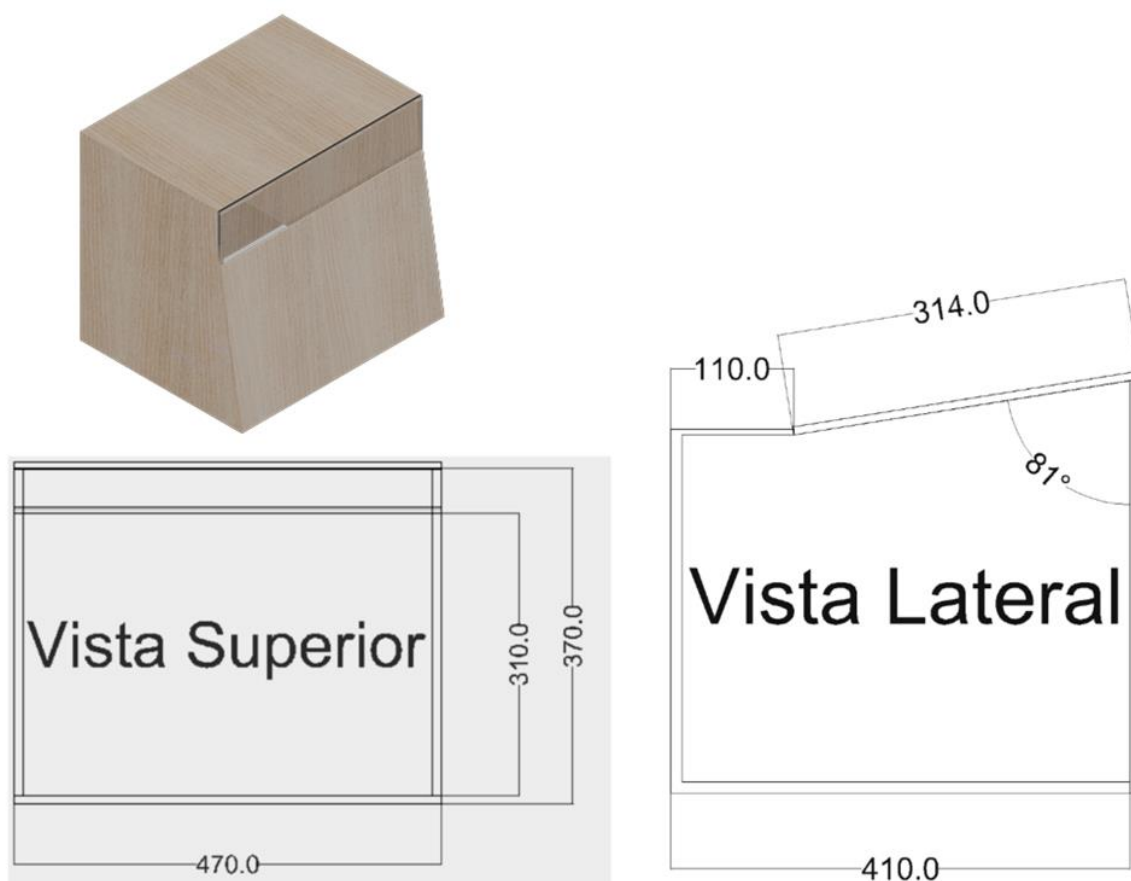
Dessa forma, o desenvolvimento do módulo não apenas contribui para o estudo das tecnologias de comunicação industrial, mas também fornece uma ferramenta importante para o ensino e a pesquisa em automação industrial, redes industriais e sistemas ciberfísicos.

## 2 PARTE FÍSICA 1 – ESTRUTURA

### 2.1 MÓDULO DE MADEIRA

O material escolhido para dar forma ao módulo foi madeira, pela facilidade de ser trabalhada e manuseada com ferramentas comuns, além de uma tampa frontal em acrílico para possibilitar a visão rápida dos equipamentos internamente e facilitar a detecção de erros mostrados pelos *LEDs* indicativos dos aparelhos que não puderam ser reproduzidos na parte frontal. Nesse caso, as melhorias estruturais sugeridas para as próximas versões seriam: a utilização de um material metálico como estrutura, para que o módulo possa ser mais robusto, seguro contra incêndios e acidentes externos, além de encontrar uma solução para a visualização dos *LEDs* indicativos dos seus equipamentos internos de forma mais eficiente. O formato, as dimensões e detalhes podem ser visualizados pela Figura 1.

Figura 1 - Vistas com as medidas do módulo e uma vista isométrica



Fonte: Próprio Autor.

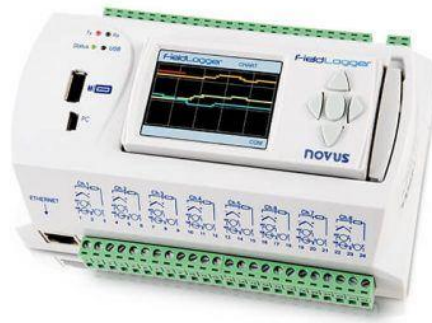
### 2.2 EQUIPAMENTOS INTERNOS

A lista de dispositivos que compõe o módulo pode ser visualizados pelas Figuras de 2 a 5, com suas respectivas descrições.

A Figura 2 representa o equipamento Fieldlogger da empresa Novus. Este equipamento é o principal componente do módulo, atua como mestre da rede Modbus e *gateway* para Ethernet, interligando as duas

redes. Como possui memória de armazenamento, também faz alguns registros de como os sensores plugados a ele estão funcionando.

Figura 2 – FieldLogger da empresa Novus



Fonte: [https://cdn.novusautomation.com/downloads/manual\\_fieldlogger\\_v19x\\_a\\_pt.pdf](https://cdn.novusautomation.com/downloads/manual_fieldlogger_v19x_a_pt.pdf)

A Figura 3 ilustra o dispositivo N1200, também da empresa Novus. Este dispositivo é um controlador universal de processos, funciona como escravo Modbus e possui uma saída em *PWM* (*Pulse Width Modulation* – Modulação por largura de pulso) que possibilita ajuste fino sobre um atuador.

Figura 3 – Controlador N1200 da empresa Novus



Fonte: [https://www.novus.com.br/downloads/arquivos/v20x\\_manual\\_n1200\\_port.pdf](https://www.novus.com.br/downloads/arquivos/v20x_manual_n1200_port.pdf)

A Figura 4 apresenta o dispositivo Switch Ethernet da empresa Allen-Bradley, que funciona como um extensor Ethernet, possibilitando a conexão de mais equipamentos à rede.

Figura 4 - Switch Ethernet da empresa Allen-Bradley



Fonte: <https://www.plchardware.com/Products/RA-1783-US05T.aspx>

A Figura 5 representa uma fonte de 24 volts, também da empresa Allen-Bradley, responsável por fornecer alimentação extra para sensores ou atuadores que precisarem ser energizados.

Figura 5 - Fonte de 24 V da empresa Allen-Bradley

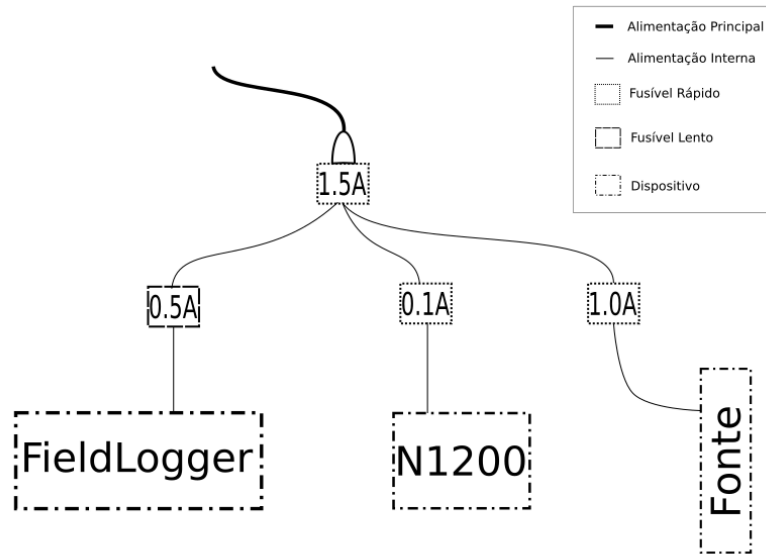


Fonte: <https://ddpartssolution.com/products/allen-bradley-1606-xle80e-din-rail-power-supply>

## 2.3 CABEAMENTO INTERNO – ELÉTRICA

A conexão elétrica, ilustrada pela Figura 6, foi feita sendo distribuída por um conector geral, para cada um dos dispositivos, associado a uma chave geral de liga/desliga e um fusível. Cada um dos dispositivos também possui uma chave de liga/desliga e um fusível associado à sua conexão, com exceção do Switch, que é alimentado internamente pela fonte de 24V, mas possuindo também uma chave de liga/desliga associado a sua ligação.

Figura 6 - Esquema elétrico interno



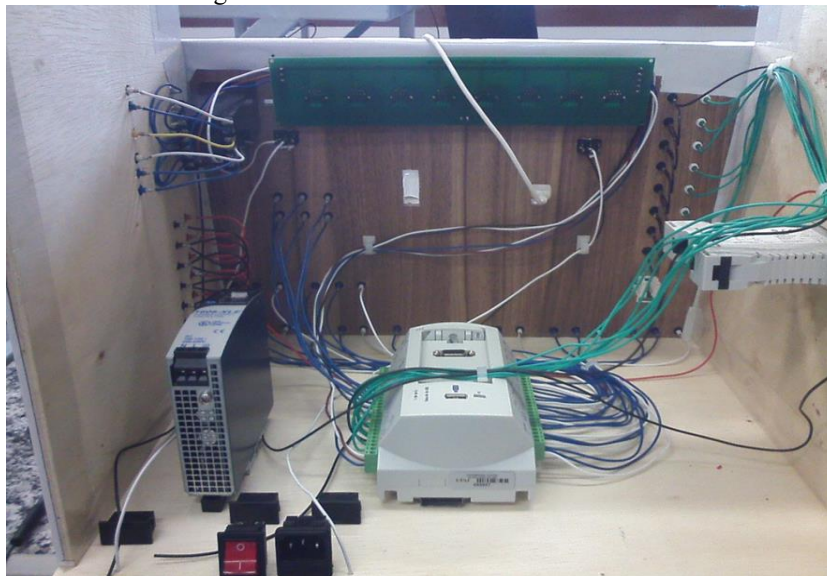
Fonte: Próprio Autor.

As conexões analógicas, digitais e relés dos dispositivos foram disponibilizadas em bornes de 2 mm de diâmetro, assim como as saídas de 24V da fonte auxiliar, perfazendo um tamanho padrão, utilizado também nos outros módulos presentes no laboratório.

Todos os cabos flexíveis utilizados nas conexões dos equipamentos estão dimensionados acima da especificação do fabricante para evitar quaisquer problemas elétricos (cabos flexíveis de 0.75mm de raio), além de terem cores correspondentes para cada tipo de ligação facilitando a organização e visualização, caso necessite de futuras manutenções. A Figura 7 ilustra o cabeamento sendo montado no módulo.

As conexões USB's foram disponibilizadas na frente do módulo, por cabos extensores, assim como a interface Ethernet do Fieldlogger, que também foi estendida utilizando um cabo de rede e um conector fêmea Ethernet.

Figura 7 - Foto cabeamento sendo montado

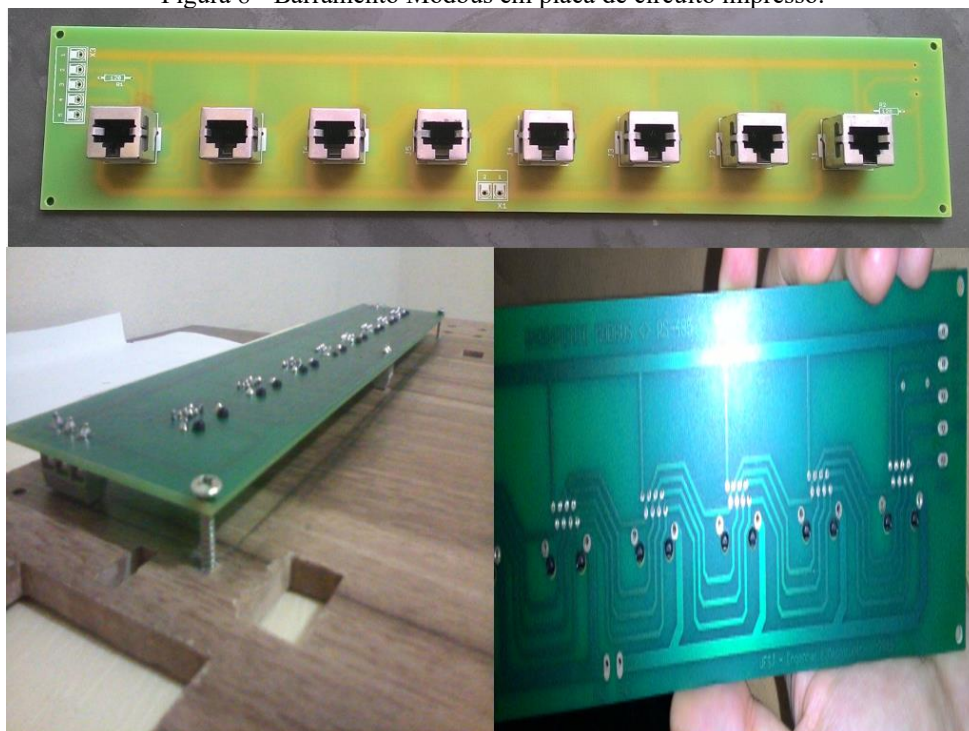


Fonte: Próprio Autor.

## 2.4 BARRAMENTO MODBUS

As conexões Modbus, ilustradas pela Figura 8, foram feitas por uma placa de circuito, desenvolvida especificamente para realizar as conexões internas entre o Fieldlogger e o N1200, além de proporcionar as conexões externas para os outros escravos que serão utilizados. Também foram deixados espaços na placa, para serem colocados resistores de terminação, caso seja exigido uma melhora na relação de sinal/ruído em projetos futuros.

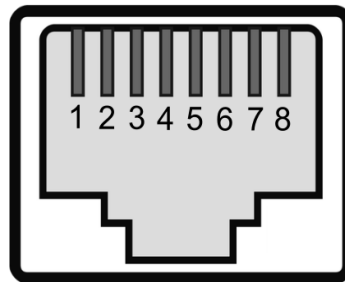
Figura 8 - Barramento Modbus em placa de circuito impresso.



Fonte: Próprio Autor.

A conexão dos dados e alimentação no barramento Modbus é baseada na pinagem do conector RJ45, ilustrada pela Figura 9. Esse padrão que deve ser seguido por todos os equipamentos que serão inseridos nesse barramento, mantendo o posicionamento de cada fio para a crimpagem de novos conectores, lembrando sempre que a crimpagem e a utilização devem ser feitas de forma correta, pois podem acarretar danos irreversíveis ao equipamento.

Figura 9 - Barramento Modbus



- 1 - Referência negativa da Fonte 24V
- 2 - Referência positiva da Fonte 24V
- 3 - Referência positiva da Fonte 24V
- 4 - Conexão D1 da Modbus RS-485
- 5 - Conexão D1 da Modbus RS-485
- 6 - Conexão D0 da Modbus RS-485
- 7 - Conexão D0 da Modbus RS-485
- 8 - Conexão C da Modbus RS-485

Fonte: Próprio Autor.

A fonte de 24V citada é a fonte interna do FieldLogger que deve ser utilizada apenas com equipamentos que necessitam de pouca potência para funcionar.

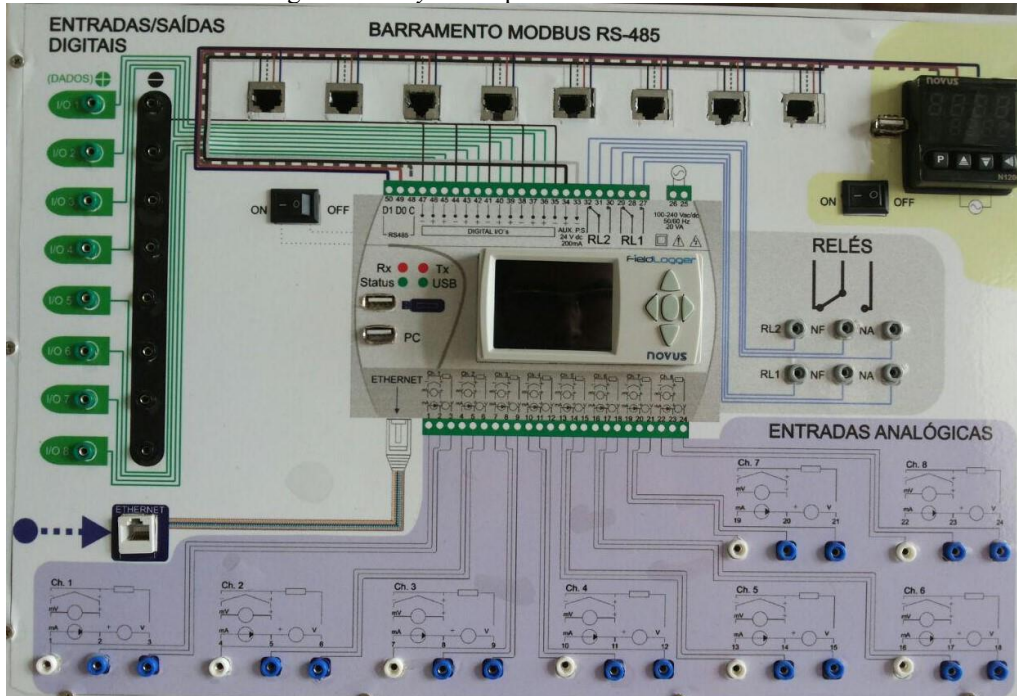
## 2.5 ADESIVOS – ACABAMENTO E *LAYOUT*

A construção do módulo se deu por um processo contínuo, sempre procurando a melhor localização dos conectores externos, com o mínimo de prejuízo para a organização dos componentes internos. Como a proposta de criação envolve a utilização do módulo na área acadêmica, várias tentativas de layout foram feitas até chegar no formato ilustrado pelas Figuras 10 e 11, sendo que a Figura 10 apresenta os layouts das laterais do módulo e a Figura 11 o layout frontal.

Depois de ter sua estrutura pronta, a escolha e desenvolvimento de um *layout* de posicionamento foi pensado, de modo que todos os componentes pudessem caber no módulo de madeira, sem que houvesse exagerada sobreposição de fios e que a parte externa pudesse ser utilizada, não só como extensão, mas também como exposição dos equipamentos que formam o módulo. Utilizando um *software* de desenho vetorial, chegou-se ao desenho do *layout* final, que, posteriormente, foi transformado em adesivo e colado nas laterais da plataforma. Depois de feitas as furações na caixa para encaixe de plugues e equipamentos,

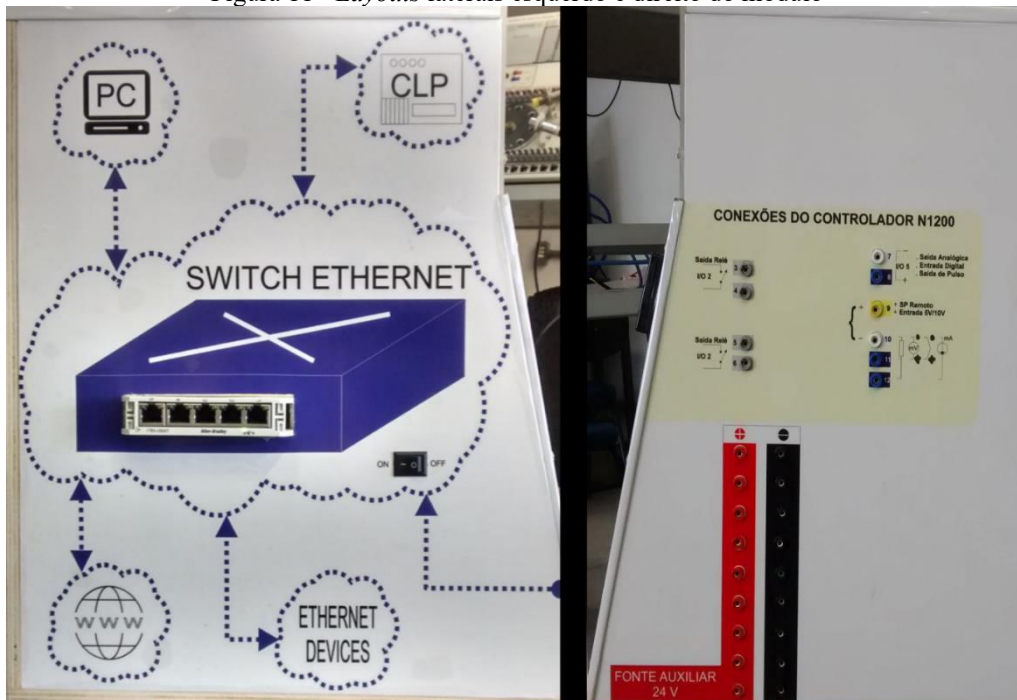
os adesivos foram colados e, por fim, uma capa adesiva transparente foi colocada para garantir maior durabilidade do equipamento e segurança para os desenhos.

Figura 10 - Layout do painel frontal do módulo



Fonte: Próprio Autor.

Figura 11 - Layouts laterais esquerdo e direito do módulo

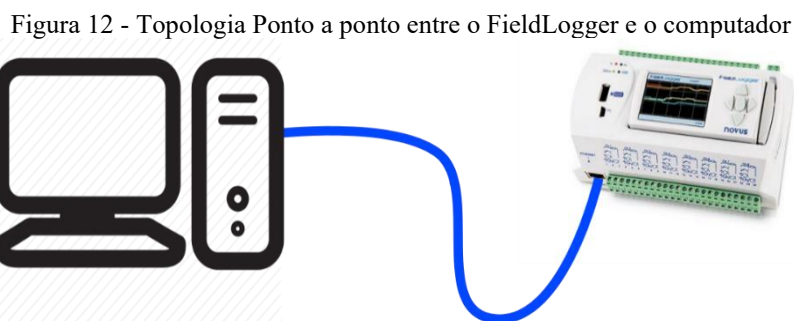


Fonte: Próprio Autor.

## 3 PARTE FÍSICA 2 – REDE

### 3.1 TOPOLOGIA DE REDE PONTO A PONTO

A primeira parte de testes e estudos do equipamento, mesmo antes do módulo estar fisicamente pronto, foi feita ligando-se o FieldLogger diretamente ao computador pela interface Ethernet, caracterizando uma ligação de rede ponto a ponto, conforme ilustra a Figura 12.



Fonte: Próprio Autor.

A topologia ponto a ponto continuou a ser usada em alguns testes, através da extensão Ethernet, instalada no painel frontal, após a montagem completa da plataforma, fixando uma ponta do cabo de rede ao conector RJ45 fêmea (Figura 13) e respeitando a ordem da fiação, conforme apresentado pela Figura 9.

Figura 13 - Conector RJ45 Fêmea utilizado na extensão Ethernet



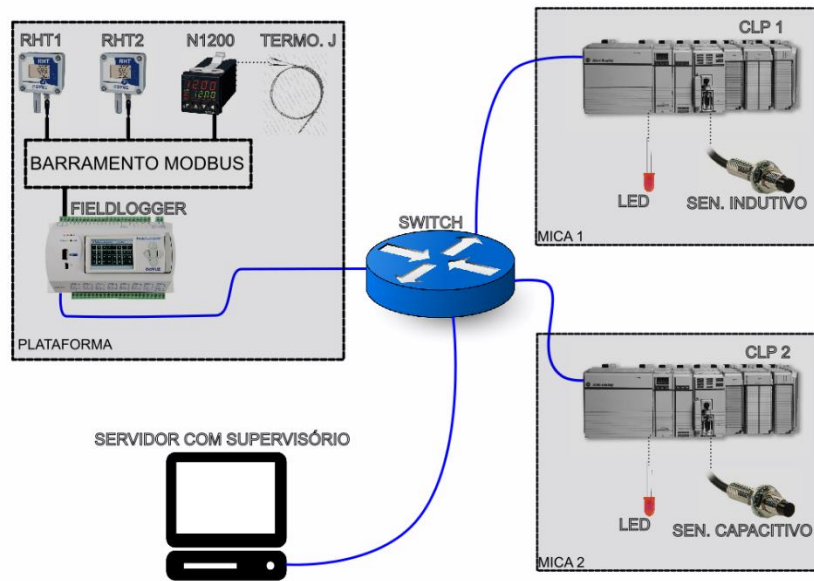
Fonte: <https://www.cablewholesale.com/products/network-phone/keystones/product-326-120iv.php>

### 3.2 REDE CRIADA PARA TESTE DO MÓDULO (PLATAFORMA)

Para testar a versatilidade do módulo operando com outros equipamentos, criou-se uma rede com outras topologias e arquiteturas, passando essa a ser de um tipo mista, onde, em vez de ponto a ponto apenas, tem agora a união das topologias de barramento e estrela se misturando e formando um aglomerado, conforme ilustra a Figura 14.

A arquitetura também sofreu mudanças, pois onde apenas o padrão Mestre-Escravo do Modbus operava, agora passou a existir Cliente-Servidor com a necessidade de instalação de um *software* supervisor para gerenciamento completo da rede.

Figura 14 - Rede criada para teste do (módulo)



Fonte: Próprio Autor.

A arquitetura Mestre-Escravo presente nos equipamentos Modbus funciona da seguinte maneira: O mestre da rede (geralmente o FieldLogger) faz a requisição de um dado para um escravo através de uma mensagem de *broadcast*, essa requisição é processada e o escravo devolve o valor requerido para o mestre, podendo esse valor ser usado, posteriormente, para ser visualizado em algum *software* ou sistema de controle.

A arquitetura Cliente-Servidor está relacionada com a interface Ethernet. Ela funciona de maneira um pouco diferente da Mestre-Escravo pois ao invés das requisições serem feitas por um servidor, agora elas ficam no servidor e são requisitadas pelo cliente, podendo o mesmo ser um dispositivo ou um programa que recebe esses valores e os manipula como for necessário.

## 4 PARTE LÓGICA 1 – PROGRAMAS

### 4.1 FLCONFIG – GERAL

Para que o módulo funcione, o FieldLogger deve ser configurado para desempenhar as suas funções de registro, controle e supervisão da rede. A configuração foi feita utilizando o *software* FLConfig, disponibilizado gratuitamente pelo fabricante, possibilitando a total personalização de algumas funções do equipamento como a forma de trabalho na rede Modbus, supervisão de *tags* e alarmes, entre outras que serão explicitadas logo a seguir. A primeira tela do FLConfig (Figura 15) exibe as quatro configurações disponíveis:

- a- Configuração: exibe as opções para alterar as configurações do FieldLogger;
- b- Coleta: exibe as configurações para coletar os dados armazenados no FieldLogger;
- c- Diagnóstico: permite a exibição gráfica de variáveis previamente configuradas;
- d- Preferências: exibe as opções configuráveis do FLConfig.

Figura 15 - FLConfig: tela inicial



Fonte: Próprio Autor.

### 4.2 FLCONFIG – MODBUS

A configuração da rede Modbus no FieldLogger segue os seguintes parâmetros:

- a- Modo de operação: o modo de operação é a principal característica de uma rede Modbus. Nesse ponto o FieldLogger pode estar desabilitado, modo mestre ou modo escravo. Quando desabilitado, não faz parte da rede Modbus, em modo escravo, simplesmente atua como um escravo da rede e responde às solicitações do mestre. Quando está em modo mestre, ele se

transforma no principal componente da rede e todas as opções que serão descritas abaixo devem ser configuradas nos escravos da rede para o mesmo padrão que for definido nele;

- b- Endereço: o campo endereço só permite alteração quando o FieldLogger está configurado em modo escravo, pois em modo mestre o valor deve seguir a regra padrão do Modbus ficando alocado no endereço 255. Esse valor pode ir de 1 a 247, que é o limite de dispositivos que a rede suporta;
- c- *Baud Rate*: a taxa em bits por segundo das transferências de dados na rede. Os valores são padronizados e podem ser escolhidos entre 1200 e 115200 bits por segundo, sendo sempre o valor do mestre seguido pelos escravos. Ex.: Se o mestre da rede (FieldLogger ou outro equipamento) foi configurado para operar em 9600 bits por segundo, todos os escravos que forem adicionados na rede devem ser configurados com a mesma taxa;
- d- Bits de parada: como a rede Modbus é uma rede síncrona, ela possui bits de parada que servem para sinalizar ao equipamento quando uma transferência de dados termina, melhorando a confiabilidade no fluxo de informações. Pode ser configurado utilizar nenhum, um ou dois bits de parada;
- e- Paridade: a checagem de erros é feita por verificação de paridade, podendo ser de dois tipos: par ou ímpar. Na paridade par é inserido um bit igual a “1” no fim do dado quando a quantidade de bits “1” que compõem esse mesmo dado é par, caso a quantidade for ímpar insere-se o bit “0”. Na paridade ímpar o bit “1” é inserido quando a quantidade de bits “1” do dado for ímpar e o bit “0” é inserido caso essa quantidade seja par.

Todas essas configurações relacionadas à rede Modbus são configuradas na aba chamada: “Interface RS485” do FLConfig, sendo RS485 o padrão físico da rede, apresentada na Figura 16:

Figura 16 - FLConfig: interface RS485

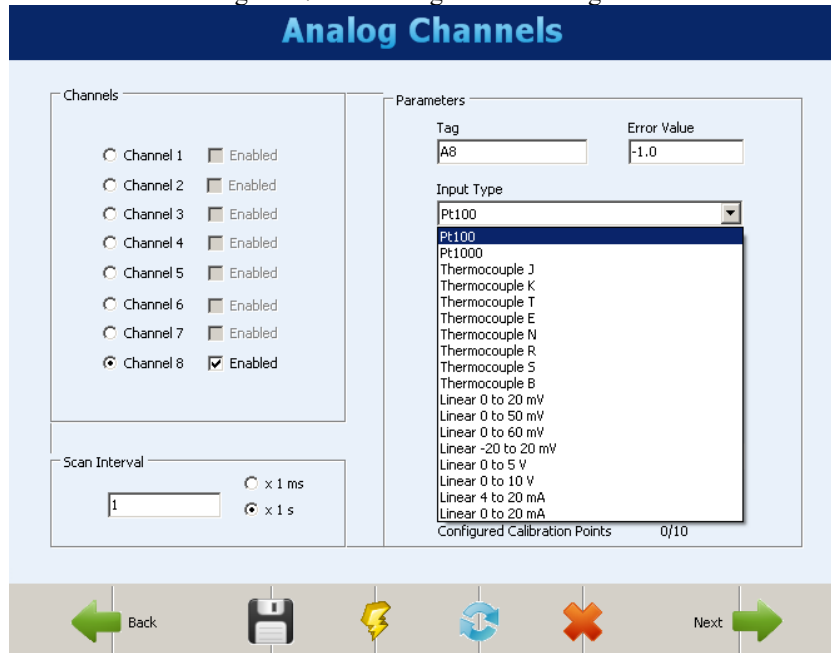


Fonte: Próprio Autor.

### 4.3 FLCONFIG - ANALÓGICAS

Os canais analógicos são compreendidos por oito entradas analógicas que podem ser configuradas para trabalhar com diversos tipos de sensores analógicos, de termopares a pt-100's, desde que a escala de trabalho dos sensores seja de algum dos tipos que o equipamento suporta, conforme ilustra a Figura 17.

Figura 17 - FLConfig: canais analógicos



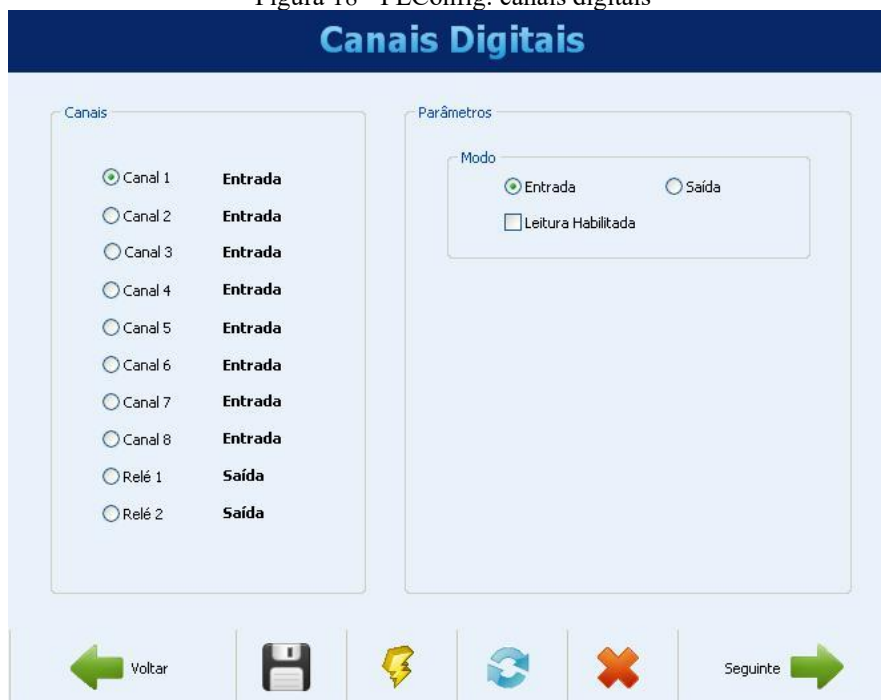
Fonte: Próprio Autor.

Com a entrada número oito habilitada, é possível notar todas as opções que o equipamento suporta, devendo o restante das configurações ser testada e modificada para cada tipo de sensor que o operador utilizará, sendo possível inclusive configurar até dez pontos de calibração para o sensor utilizado em uma forma de ajuste fino.

#### 4.4 FLCONFIG - DIGITAIS E RELÉS

Os canais digitais, ilustrados pela Figura 18, funcionam como entradas ou saídas de sinal, com exceção dos relés que funcionam apenas como saídas, onde uma lógica de programação ou comando modifica o valor desejado. Os valores lógicos de ligado ou desligado, representados por decimais “1” ou “0”, são obtidos nesses canais pelos valores de tensão de 5V ou 0V. Vale ressaltar que mesmo funcionando como saída, cada canal tem a função apenas de representação, ou seja, se for necessário o acionamento de algum aparelho externo utilizando o sinal de saída, o mesmo deve ser amplificado pois não tem potência suficiente para acionar cargas externas.

Figura 18 - FLConfig: canais digitais

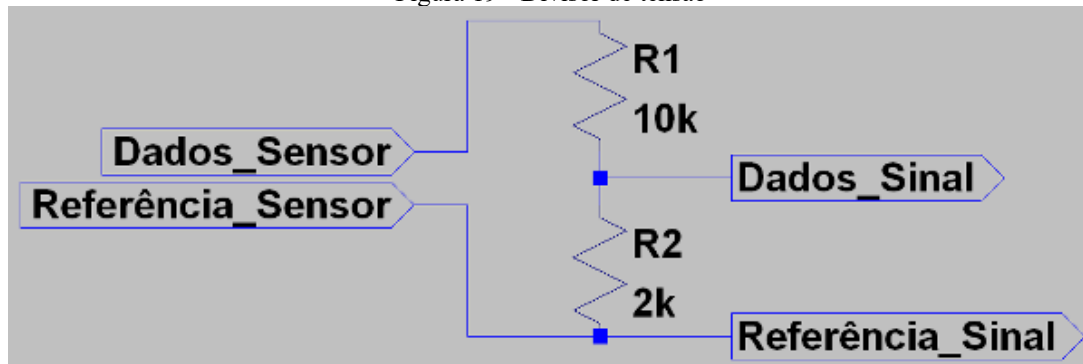


Fonte: Próprio Autor.

Quando os sensores digitais presentes no laboratório são utilizados, eles funcionam alimentados por alguma fonte de 24V, seja ela proveniente da plataforma ou outra externa. Isso faz com que a tensão de saída fique entre 24V quando está em nível alto e caia a 9V, quando em nível baixo. Sendo assim, se for colocado diretamente em alguma das entradas do módulo o valor nunca será lido corretamente, pois estará sempre em nível alto. Para contornar isso e poder usar os sensores corretamente, foi feito um divisor de

tensão, representado pela Figura 19, que trata o sinal que sai do sensor, antes de chegar na entrada do módulo, permitindo assim que o valor seja lido da forma desejada.

Figura 19 - Divisor de tensão



Fonte: Próprio Autor.

#### 4.5 FLCONFIG - CANAIS REMOTOS

A primeira etapa de configuração da rede Modbus fica reservada para a forma como a rede vai se comportar e os parâmetros que ela vai seguir de acordo com o mestre. Depois de feitos os acertos de configuração é possível cadastrar até 64 variáveis para que o FieldLogger faça a análise e/ou controle de dados dos escravos presentes na rede Modbus. Essa etapa não é obrigatória, tendo em vista que esses dados podem ser analisados, posteriormente, por um *software* de supervisão, mas pode ser da intenção do usuário que dados mais importantes sejam analisados diretamente na IHM do módulo, para isso é necessário que essa etapa seja realizada de forma correta, conforme ilustram as Figuras 20, 21 e 22.

Para exemplificar de forma prática faremos o cadastro de duas variáveis que o RHT (sensor) mede, a temperatura e a umidade. Primeiramente, é necessário dar um nome à variável, que faça sentido, facilitando a sua identificação posterior. Esse nome é colocado no campo “Tag”. No campo “Endereço Escravo Modbus”, devemos colocar o endereço Modbus que foi configurado previamente no RHT (escravo Modbus que será sendo monitorado). No campo “Comando Modbus” selecionamos o comando que o mestre enviará para o escravo para poder fazer a leitura da variável escolhida, no caso do RHT escolhemos o comando “03 – Read Holding Registers”, que é um comando de leitura de um registrador, um espaço de memória dentro do RHT que armazena o valor que está sendo medido pelo sensor analógico (nesse caso o sensor de temperatura e umidade). Se fosse utilizado outro dispositivo, possivelmente outro comando Modbus poderia ser utilizado, mas o FieldLogger apenas apresenta quatro opções que são mais do que suficientes para a todos os equipamentos que foram testados<sup>1</sup>. No campo “Registrador Inicial” é necessário colocar em qual registrador de memória (registrador do RHT) se encontra o dado que queremos observar,

<sup>1</sup> Para mais informações sobre os comandos Modbus possíveis é sugerida a leitura disponível em: <[http://modbus.org/docs/PI\\_MBUS\\_300.pdf](http://modbus.org/docs/PI_MBUS_300.pdf)>.

no caso do RHT temos 3 registradores que podemos utilizar: O “7” contém o valor da temperatura, o “8” contém o valor da umidade e o “9” que é o resultado do ponto de orvalho. Esses valores são o padrão dos registradores do RHT, para outros dispositivos possivelmente os valores serão diferentes<sup>2</sup>. O campo “Unidade” é preenchido com a unidade da variável que será observada (°C ou %). O campo “Valor de erro” é por padrão “-1”, valor esse que será exibido caso ocorra algum erro de comunicação entre o mestre e o escravo que está fornecendo o dado observado. No campo “Nº Casas Decimais” é configurado o número de casas decimais que serão utilizadas para mostrar o valor da variável recebida pelo mestre, que no caso do RHT, por exemplo, todas as variáveis que ele envia ao mestre vem no formato de três números, por exemplo, se ele estiver medindo uma temperatura de 28,5°C, enviará o valor “285” para o mestre (FieldLogger). Logo, para exibir esse valor corretamente na tela, coloca-se “1” como o valor de casas decimais. Salientando que, caso, outros escravos estiverem sendo analisados, podem apresentar configurações diferentes. A caixa seletora “Valor sem sinal” deve ser marcada quando a variável que estiver sendo medida não possuir valores negativos, como no caso da umidade, por exemplo. Por outro lado, quando estiver medindo temperatura, é possível que aconteçam valores negativos, nesse caso, a caixa deve se manter desmarcada.

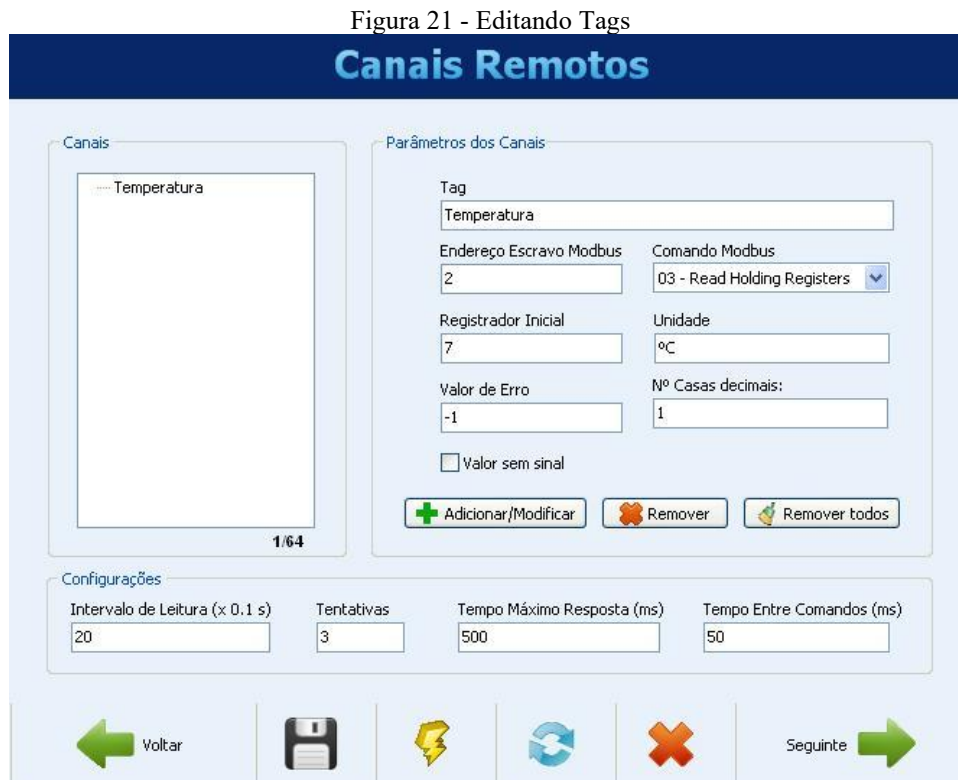
Figura 20 - FLConfig: canais remotos

Fonte: Próprio Autor.

Após ter todos os campos preenchidos corretamente, pressiona-se o botão “Adicionar/Modificar” para adicionar a *tag* e logo ela aparecerá na lista de canais ao lado. Caso queira adicionar outra *tag* é necessário apenas modificar o nome e os parâmetros que forem necessários, se apenas os parâmetros forem

<sup>2</sup> O mapa de registradores do RHT está disponível em:  
<[http://www.novus.com.br/downloads/Arquivos/v10x\\_b\\_manual\\_rht\\_wm-dm-p10\\_485\\_lcd\\_%20portuguese.pdf](http://www.novus.com.br/downloads/Arquivos/v10x_b_manual_rht_wm-dm-p10_485_lcd_%20portuguese.pdf)>.

modificados e o nome permanecer o mesmo, após clicar em “Adicionar/Modificar” a *tag* criada anteriormente será modificada. O botão “Remover” remove a *tag* que está selecionada na lista de canais e o botão “Remover todos” remove todas as *tags* criadas. Tudo isso pode ser visualizado pela Figura 21.



Fonte: Próprio Autor.

Os campos de configurações (Figura 22), contêm as opções de acesso aos dados dos canais, uma vez configurados funcionam igualmente para todos os canais adicionados anteriormente. Em “Intervalo de leitura” é configurado o tempo que o FieldLogger vai aguardar entre uma leitura e outra da lista de todos os canais catalogados. O campo “Tentativas” é responsável por limitar a quantidade de vezes que o FieldLogger vai tentar acessar um dado em um dos escravos na rede, antes de mostrar o dado com o valor de erro (caso exista algum erro na transmissão) e o complemento desse campo é o “Tempo Máximo de Resposta (ms)” que é o tempo em microssegundos onde o FieldLogger espera para receber uma resposta de um comando dado a algum dos escravos. Por último, o campo “Tempo Entre Comandos (ms)” é responsável por estabelecer o tempo que o FieldLogger vai aguardar para enviar cada comando Modbus quando a conexão é bem-sucedida, atualizando assim os valores dos canais para que possam ser lidos.

Figura 22 - Conexões e tempos

Intervalo de Leitura (x 0.1 s)	Tentativas	Tempo Máximo Resposta (ms)	Tempo Entre Comandos (ms)
20	3	500	50

Fonte: Próprio Autor.

#### 4.6 FLCONFIG – CANAIS VIRTUAIS

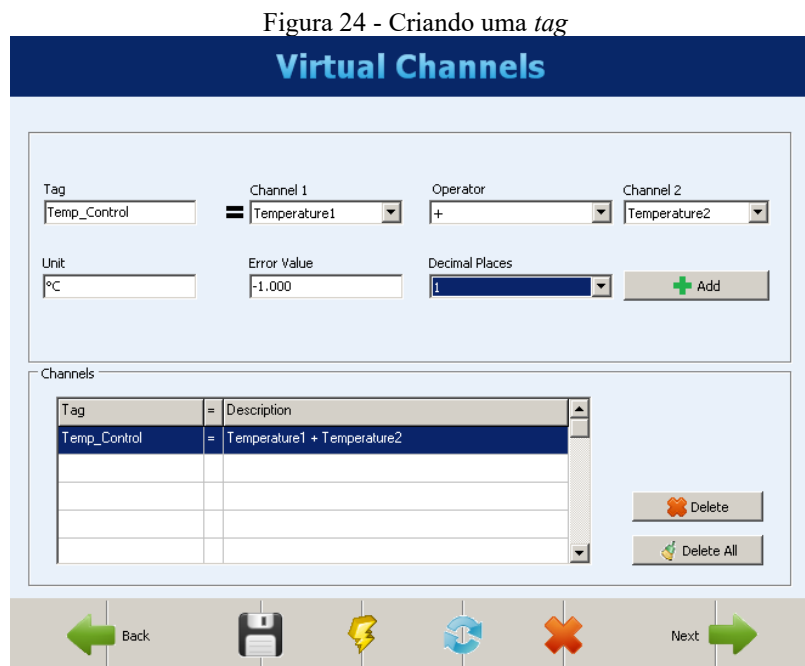
Assim como, a etapa anterior, a criação de Canais Virtuais (Figura 23) não é obrigatória, mas para o caso onde o operador não possui nenhum *software* supervisor disponível, o FieldLogger permite que sejam adicionados até 128 canais virtuais, que funcionam como variáveis, podendo ser utilizadas para quaisquer lógicas necessárias ao uso do operador. Os campos relativos a essa etapa de configuração são: “Tag” (contém o nome da variável que está sendo criada), “Canal 1” e “Canal 2” (seleção dos componentes que vão participar da lógica que já foram cadastrados anteriormente na etapa Canais Remotos), “Operador” (qual a lógica de operação que será implementada entre os dois canais), “Unidade” (corresponde a unidade final da operação entre os canais), “Valor de Erro” (Valor que será exibido caso algum erro aconteça) e “Casas Decimais” (a quantidade de casas decimais a considerar no resultado final). Depois de todas as configurações feitas e conferidas o botão de adicionar é pressionado e a *Tag* é cadastrada como um canal do equipamento, aparecendo na lista de canais logo abaixo. Caso algo esteja incorreto, os botões “Remover” (para remover apenas a *tag* selecionada) ou “Remover Todos” (para remover todas as *tags* criadas) podem ser utilizados para a remoção das *tags* desejadas.

Figura 23 - FLConfig: canais virtuais

Tag	Descrição

Fonte: Próprio Autor.

Um exemplo prático simples pode ser demonstrado através de um controle de temperatura, onde a soma de duas temperaturas não pode atingir um valor maior que o estipulado. Nesse caso, utiliza-se “Temp\_Control” como nome da *tag* e adiciona-se os dois canais que serão analisados junto do operador “+”, que estará sempre somando o valor lido pelo FieldLogger desses dois canais. Para a temperatura coloca-se a unidade em “°C” e o valor de erro mantém-se em “-1”, pois nesse caso não se trabalha com equipamentos que estejam em temperaturas negativas. Caso fosse necessário, esse valor deveria ser alterado pois poderia causar incoerências no momento de supervisão do funcionamento do equipamento. Como o valor lido pelos sensores é enviado para o FieldLogger em três algarismos, utiliza-se “1” como número de casas decimais. Caso o equipamento mude, podem ocorrer diferenças nesse valor. Depois de todos os campos editados e conferidos o botão “Adicionar” é pressionado e a *tag* é mostrada na lista de canais, como pode ser observado pela Figura 24, sendo possível alterar sua descrição para facilitar a identificação.



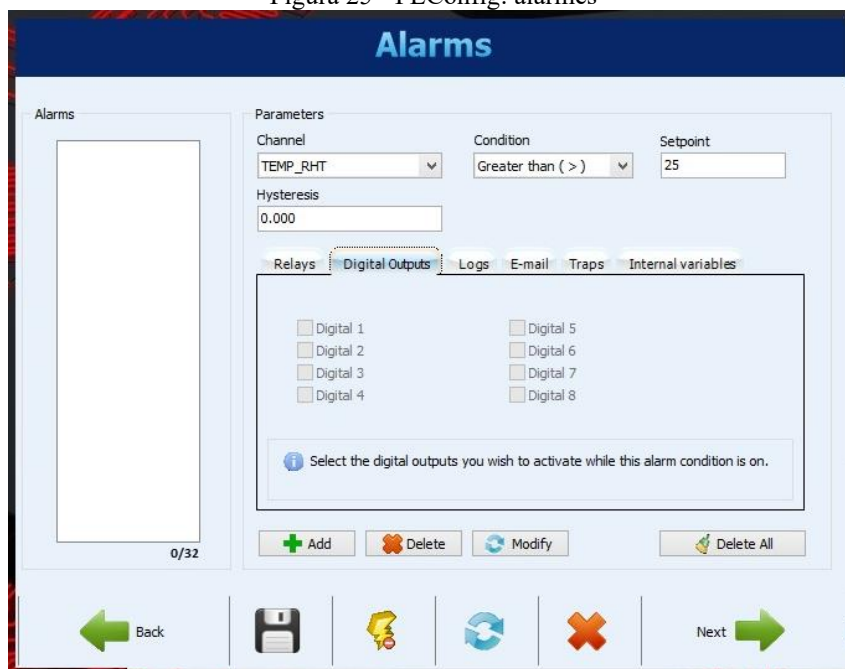
Fonte: Próprio Autor.

#### 4.7 FLCONFIG - ALARMES

Assim como, as etapas anteriores, essa etapa também não é obrigatória, mas caso o operador necessite que o FieldLogger tenha autonomia em tomadas de decisão, alarmes podem ser criados para cada canal existente, seja ele remoto ou virtual, ou também para entradas analógicas ou digitais, automatizando qualquer processo que for necessário. Se for necessário o operador pode incluir até 32 alarmes, onde os parâmetros comuns são: “Canal” (aqui é selecionado o canal que será monitorado), “Condição” (qual a condição lógica relacionada ao controle do canal), “SetPoint” (qual o valor relacionado à condição anterior) e “Histerese” (valor de histerese levado em consideração na leitura do valor do canal para evitar

chaveamentos rápidos no circuito). Logo abaixo dos parâmetros comuns, existem abas, que podem aparecer ou não, de acordo com o que foi criado nas etapas anteriores, por elas é possível determinar qual ação será tomada de acordo com a lógica proposta. A Figura 25 ilustra a tela de Alarmes.

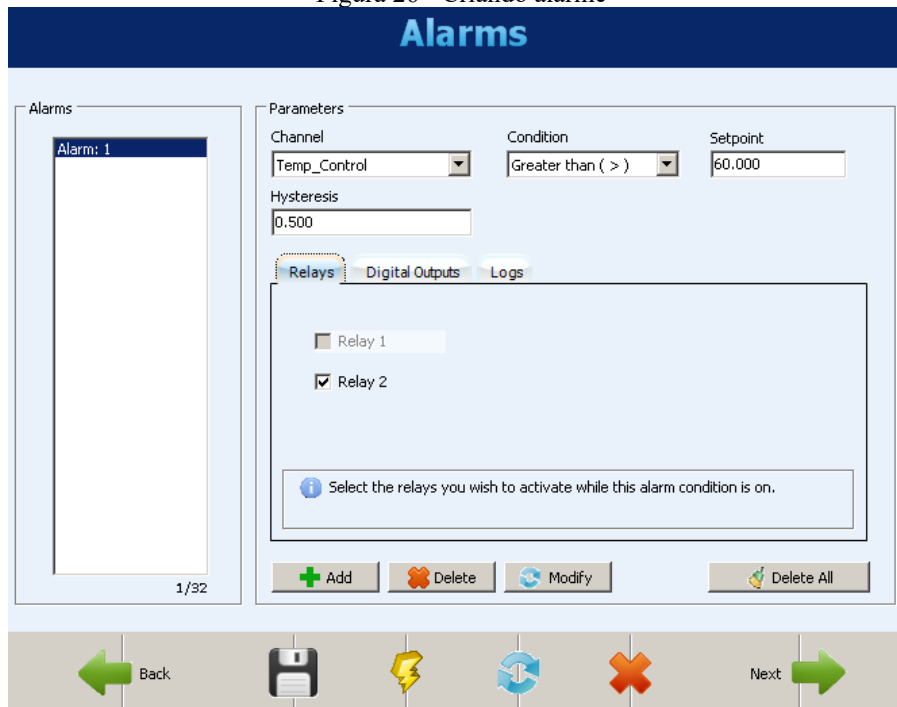
Figura 25 - FLConfig: alarmes



Fonte: Próprio Autor.

Continuando o exemplo prático anterior, na Figura 26, preencheu-se o parâmetro “Canal”, selecionando a *tag* virtual criada anteriormente com o nome “Temp\_Control”. A condição escolhida é: Maior (>) e o *Setpoint* igual a 60,000, e manteve-se a histerese de 0,500. Como ação dessa condição, seleciona-se o Relé 2 para ser acionado quando a soma ficar maior que 60. Clica-se no botão “Adicionar” e observa-se na lista de alarmes se o *Alarm:1* foi criado. Agora, sempre que a soma das temperaturas ultrapassar o valor 60, o Relé 2 do FieldLogger será acionado, podendo ter conectado a este um ventilador ou outro dispositivo qualquer proporcionando um controle simples a essas temperaturas.

Figura 26 - Criando alarme



Fonte: Próprio Autor.

#### 4.8 FLCONFIG – REGISTROS

Essa é a última etapa de configuração do equipamento (Figura 27) e, assim como as últimas, não é obrigatória para o funcionamento da rede, tendo em vista que o registro das variáveis também pode ser feito utilizando um sistema supervisor. Caso seja o desejo do operador, o FieldLogger possui a capacidade de registrar os valores dos canais cadastrados nas outras etapas e salvá-los criando uma tabela com essas informações. A primeira parte de configuração fica por conta de selecionar qual será o “Modo de Início”, que é quando o FieldLogger vai começar o registro dos dados, seguido pelo “Modo de Término”, que é quando param os registros. Além das opções a serem escolhidas, ainda é possível definir que o sistema supervisor ou outro dispositivo da rede Modbus inicie ou pare os registros, para isso basta selecionar as duas caixas de seleção com os títulos: “Permitir início dos registros via comando Modbus” e “Permitir término dos registros via comando Modbus”. Na seleção de canais é necessário escolher quais canais deverão ser registrados, passando os desejados para a coluna “Canais Selecionados”, utilizando os botões direcionais para fazer a seleção. No campo “Memória para Registros”, seleciona-se a memória que será utilizada para guardar a tabela com os valores registrados, lembrando que a opção “Cartão SD” só será válida caso o operador tenha inserido um cartão, corretamente formatado, no FieldLogger. Por fim, no campo “Intervalo Entre Registros” é possível definir o intervalo de tempo que o FieldLogger levará entre um registro e outro, sendo possível escolher entre duas escalas: “x1 s” (o valor digitado em segundos) e “x1 ms” (o valor digitado em milissegundos).

Figura 27 - FLConfig: registros

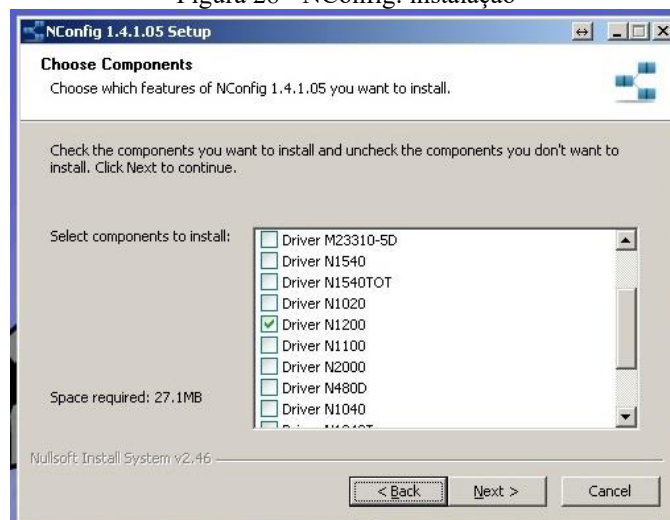


Fonte: Próprio Autor.

#### 4.9 NCONFIG – GERAL

O NConfig é utilizado para configurar o controlador N1200 da Novus, componente que integra o módulo e se conecta diretamente ao barramento Modbus. No momento da instalação do *software* é recomendado que a opção de *driver* do N1200 seja selecionada para que o *driver* seja atualizado e facilite o reconhecimento do dispositivo quando o mesmo for inserido na porta USB, como ilustra a Figura 28.

Figura 28 - NConfig: instalação

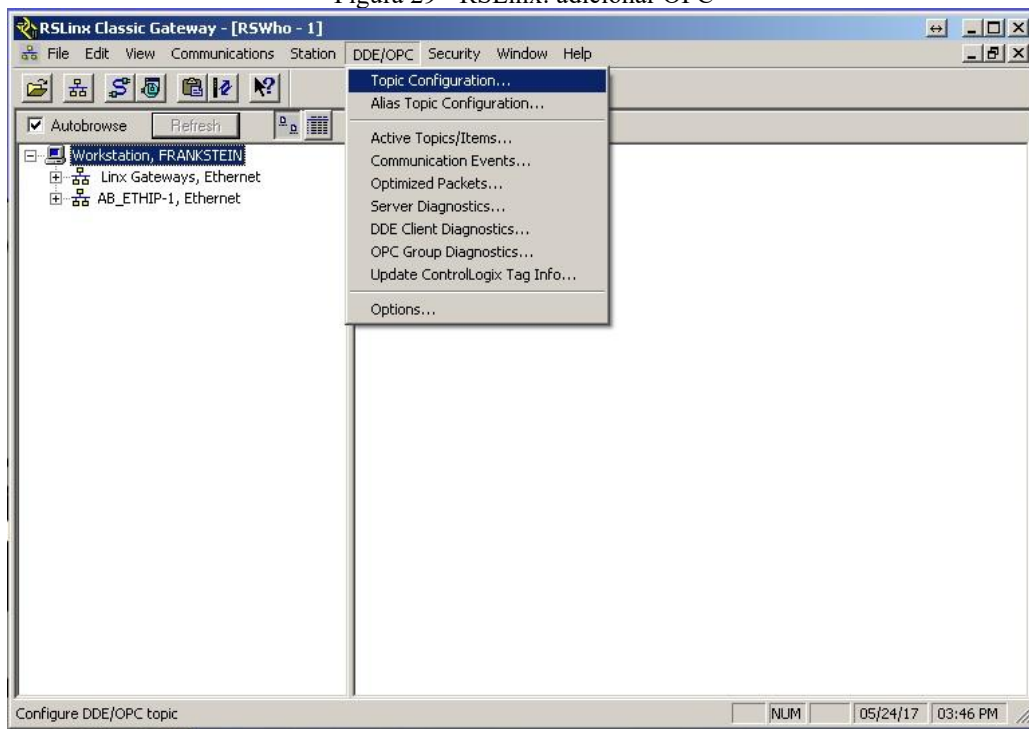


Fonte: Próprio Autor.

#### 4.10 RSLINX - SERVIDOR OPC

Os MICA's (Módulos de Instrumentação, Controle e Automação (MICA)), presentes no laboratório, possuem conexão à rede Ethernet/IP, sendo acessados de maneira diferente pelos sistemas supervisórios. Mas os *softwares* evoluíram ao longo do tempo e é possível que um mesmo sistema supervisório acesse diferentes tipos de rede, como é o caso que está sendo trabalhado. Para que os CLP's dos MICA's sejam acessados na rede é necessário a existência de um servidor *OPC*, que é um programa responsável por fazer a ponte entre o sistema supervisório e o equipamento. No caso dos MICA's a criação do servidor *OPC* pode ser feita diretamente pelo RSLinx, como mostra a Figura 29. Se todos os MICA's forem acessíveis pela rede é necessário apenas um computador com o RSLinx instalado, pois uma vez criado, o servidor *OPC* pode direcionar a requisição para cada CLP presente na rede.

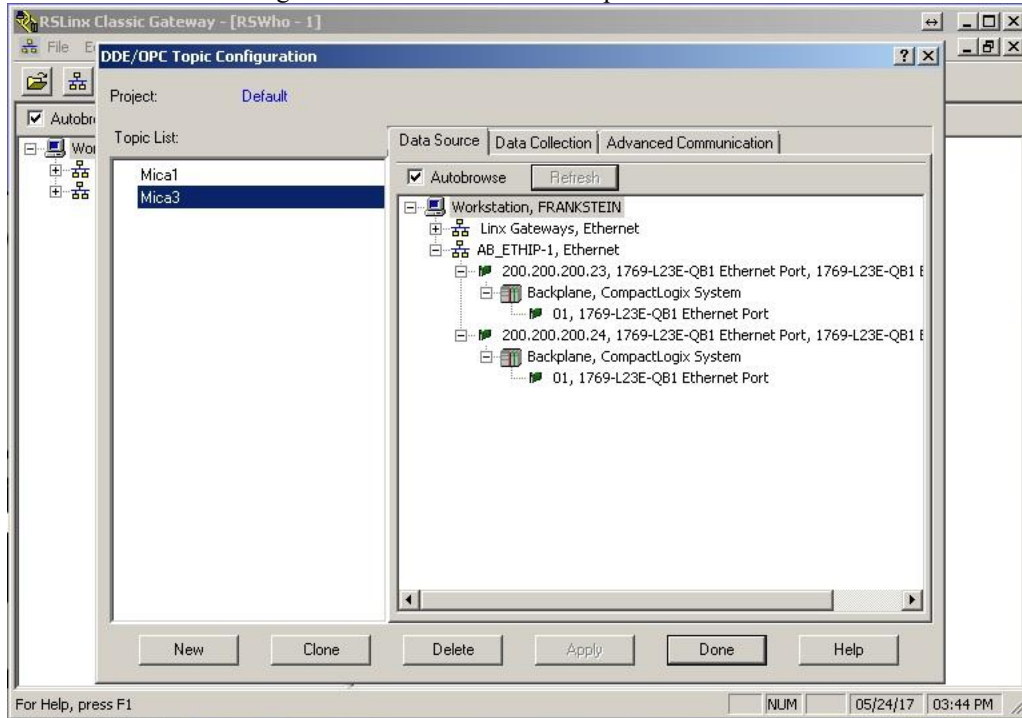
Figura 29 - RSLinx: adicionar OPC



Fonte: Próprio Autor.

Após clicar na opção *DDE/OPC*, tendo já adicionado a placa de rede para fazer a leitura dos dispositivos disponíveis, cria-se um tópico para cada CLP que estiver na rede e após criado basta selecionar o CLP que o tópico ficará responsável e clicar em “*Apply*”, a partir daí todo o comando enviado a este tópico será redirecionado para o CLP que estiver sido aplicado. Cada CLP deve ter seu tópico criado, para que não ocorram erros de transmissão ou falha de comunicação entre os MICA's e o sistema supervisório, como apresenta a Figura 30.

Figura 30 - RSLinx: adicionar tópicos a cada CLP



Fonte: Próprio Autor.

O protocolo utilizado pelo servidor OPC para comunicar-se com o CLP é por padrão o protocolo CIP. Em uma captura de comunicação entre o sistema supervisório e o RSLinx, apresentada pela Figura 31, utilizando o Wireshark e farejando a placa de rede, foi possível visualizar vários pacotes de comunicação com o protocolo em questão.

Figura 31 - Captura RSLinx e supervisório

Source	Destination	Protocol	Length	Info
192.168.1.110	192.168.1.103	CIP CM	114	Unconnected Send: Get Attribute All
192.168.1.103	192.168.1.110	TCP	60	44818->2069 [ACK] Seq=54 Ack=115 Win=4060 Len=0
192.168.1.103	192.168.1.110	CIP	130	Success
192.168.1.110	192.168.1.103	CIP CM	120	Unconnected Send: Unknown Service (0x4c)
192.168.1.103	192.168.1.110	CIP	174	Success
192.168.1.110	192.168.1.103	CIP CM	114	Unconnected Send: Get Attribute All
192.168.1.103	192.168.1.110	CIP	130	Success
192.168.1.110	192.168.1.103	CIP CM	120	Unconnected Send: Unknown Service (0x4c)
192.168.1.103	192.168.1.110	CIP	102	Success

Fonte: Próprio Autor.

Além da configuração interna, onde apenas o computador com o servidor OPC instalado pode fazer alterações nos MICAs, foi possível expandir o acesso ao servidor OPC para qualquer máquina da rede. Essa capacidade foi obtida graças à manipulações na política de usuários, liberando acesso externo ao sistema. Seguindo os passos do apanhado de tutoriais obtidos em:

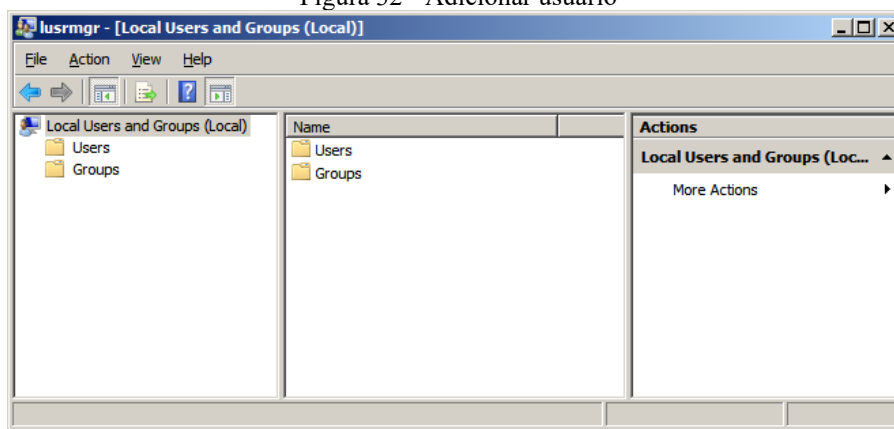
- a- <https://www.kepware.com/en-us/support/resource-library/connectivity-guides/remote-opc-dcom/> ;
- b- [https://www.automation.com/pdf\\_articles/Troubleshooting OPC and DCOM.pdf](https://www.automation.com/pdf_articles/Troubleshooting OPC and DCOM.pdf)

c- <ftp://ftp.softing.com/pub/outgoing/opc/DCOM/DCOM-Settings-en.pdf>

A primeira etapa se encontra na criação do usuário local, criado no servidor, que será utilizado pra fazer o acesso através dos sistemas supervisórios, que funcionarão nas máquinas remotas. Existe a possibilidade de fazer o acesso anônimo, mas já pensando em futuras melhorias de segurança da rede e para evitar demasiada exposição do servidor, foi realizada a configuração adequada e com segurança de acesso.

Para adicionar o usuário local que ficará responsável por disponibilizar a conexão *OPC* para todos os computadores da rede, basta abrir o gerenciador local de grupos e usuários, através do Painel de Controle ou em Iniciar → Executar, digite "lusrmgr.msc", como ilustra a Figura 32.

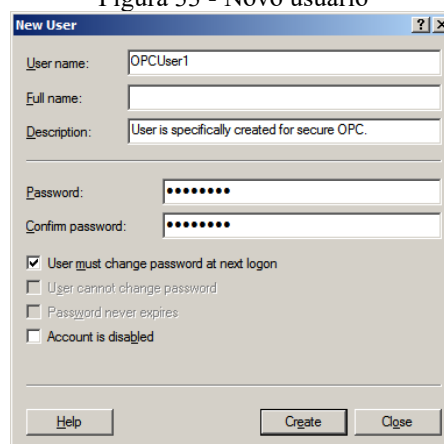
Figura 32 - Adicionar usuário



Fonte: Próprio Autor.

Depois, ao clicar em "Users" e selecionar o menu "Action" → "New User", aparecerá a tela da Figura 33.

Figura 33 - Novo usuário

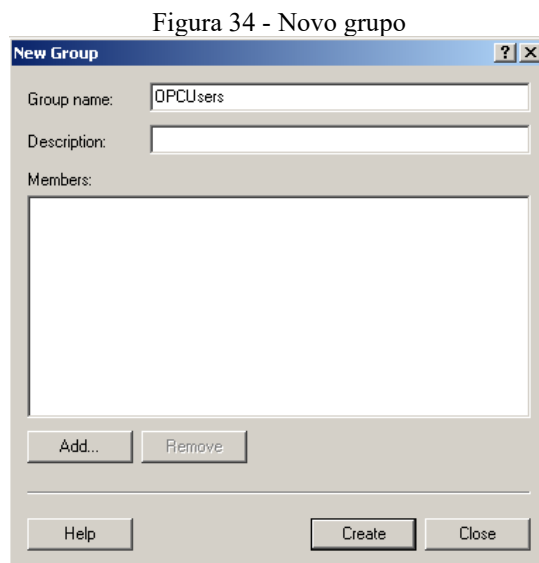


Fonte: Próprio Autor.

Assim que a tela para adicionar um novo usuário for aberta, é possível colocar o nome que o novo usuário terá, no exemplo usamos "OPCUser1". Uma descrição simples, digite a senha e confirme a senha

digitando novamente. Por fim, é necessário deixar marcada a opção “*User must change password at next logon*”, para simbolizar que o usuário deve mudar a senha a próxima vez que logar na máquina e todos os outros campos devem permanecer desmarcados. Feito isso, basta clicar em “*Create*” e a nova conta será criada.

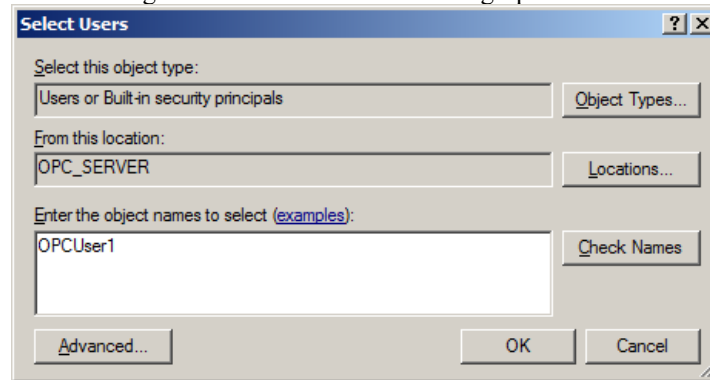
Ainda na janela de gerenciamento de usuários e grupos, basta clicar na opção “*Groups*” e selecionar “*Action*” → “*New Group*”, conforme apresentado na Figura 34.



Fonte: Próprio Autor.

Digite o nome do novo grupo como “OPCUsers” e pode-se colocar uma descrição adequada. Feito isso, basta clicar em “*Create*”. Com isso, o novo grupo de usuários OPC também foi criado. Agora que o usuário OPC e o grupo de usuários foi criado, pode-se adicionar o usuário ao grupo. Ao retornar à janela de gerenciamento de usuários e grupos, e abra-se o menu de grupos. Para selecionar o grupo “OPCUsers”, que foi criado a pouco, basta clicar com o botão direito. Sobre a opção “*All tasks*”, clicar em “*Add to group*” e “*Add*”. Será aberta a janela de seleção de usuários (Figura 35) para serem adicionados ao grupo, no espaço reservado para digitar “*Enter the object names to select (examples)*”, deve-se digitar o nome do usuário criado anteriormente, no caso do nosso exemplo: “OPCUser1”. Clique em “*Check Names*” para que o computador corrija o caminho do nome de usuário, se tudo estiver correto, clique no botão “*OK*”.

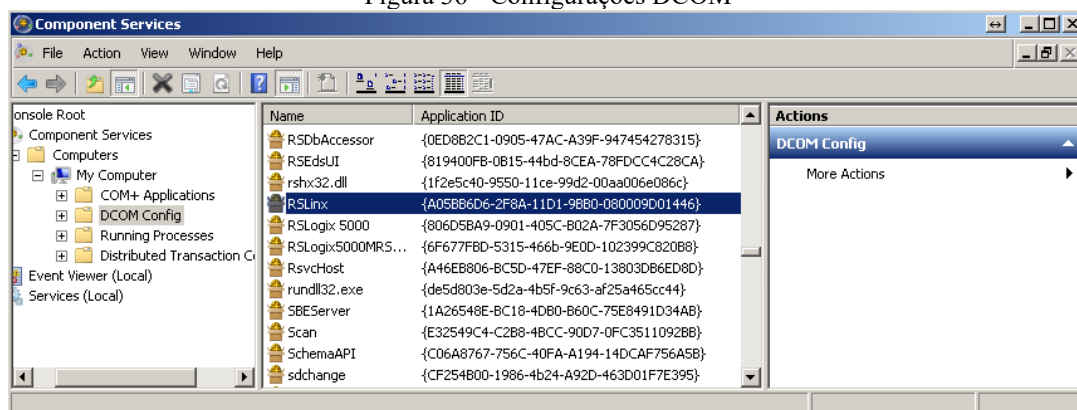
Figura 35 – Adicionar usuário ao grupo criado



Fonte: Próprio Autor.

Após a criação do usuário e o grupo que será responsável por liberar o acesso, parte-se então pra configuração da política de privacidade e acesso ao recurso pela rede. Para obter acesso a configuração DCOM, clique em iniciar → executar e digite: “dcomcnfg”. Na janela que abrir sobre a pasta “Console Root” expanda “Component Services”, “Computers”, “My Computer” e “DCOM Config”, como apresentado pela Figura 36.

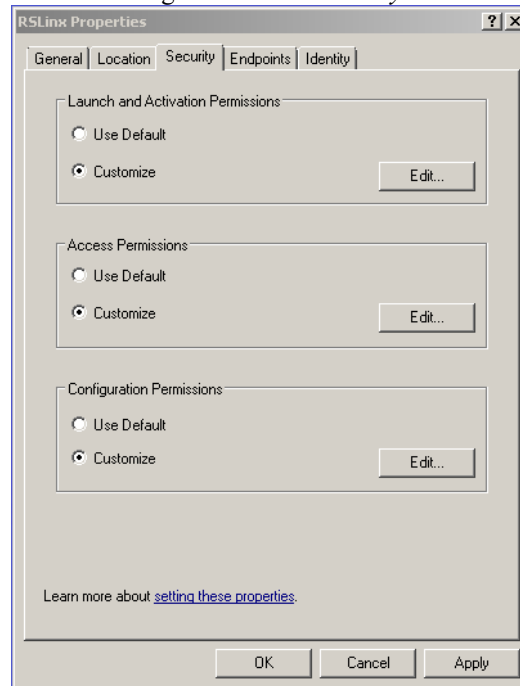
Figura 36 - Configurações DCOM



Fonte: Próprio Autor.

Procure pelos objetos DCOM habilitados e localize a aplicação de servidor OPC, que no caso do exemplo é RSLinx. Clique com o botão direito e selecione “Properties” (Figura 37). Na aba “General” a opção “Authentication Level” deve estar selecionada em modo “Default”. Na aba “Location” deve estar marcado apenas a opção: “Run application on this computer”, se a opção estiver esmaecida não altere nada. Na aba “Security” aparecem três modos de permissão e todos devem estar com a opção “Customize” marcadas além do grupo de usuários OPCGroup ser adicionado em cada uma delas.

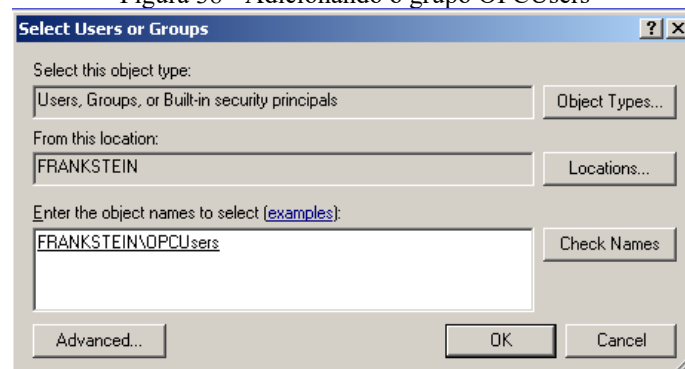
Figura 37 - Aba *Security*



Fonte: Próprio Autor.

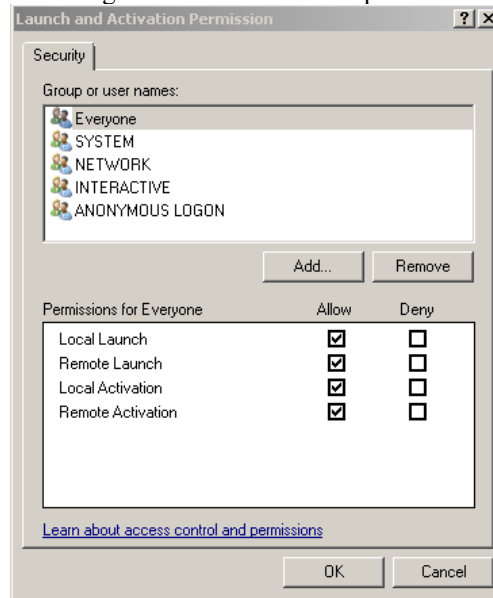
Para adicionar o grupo na primeira opção clique em “*Edit*” e a janela de permissões de cada usuário será aberta, logo clicando em “*Add*”. outra janela para selecionar usuários e grupos se abre, basta digitar o nome do grupo criado, no caso do exemplo: OPCUsers e clicar em “*Check Names*” pra que ele possa acertar o nome completo do grupo, conforme Figuras 38 e 39.

Figura 38 - Adicionando o grupo OPCUsers



Fonte: Próprio Autor.

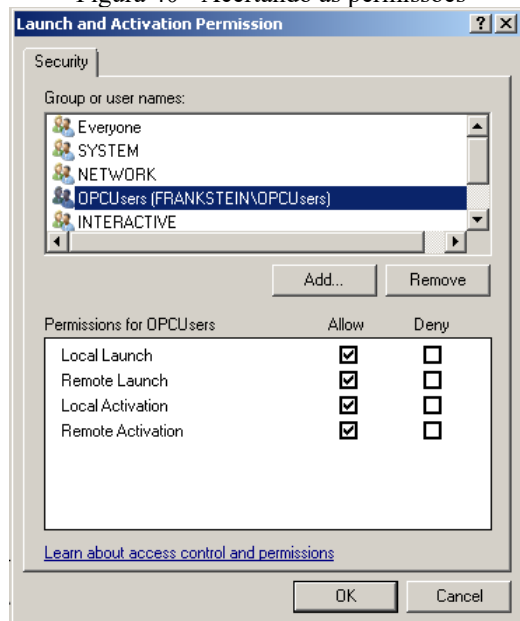
Figura 39 - Permissões de padrão



Fonte: Próprio Autor.

Feito isso, basta clicar em “OK”, e o nome do grupo aparecerá na tela de permissões. Marque a coluna “Allow” em todas as permissões e dê “OK”, conforme a Figura 40.

Figura 40 - Acertando as permissões

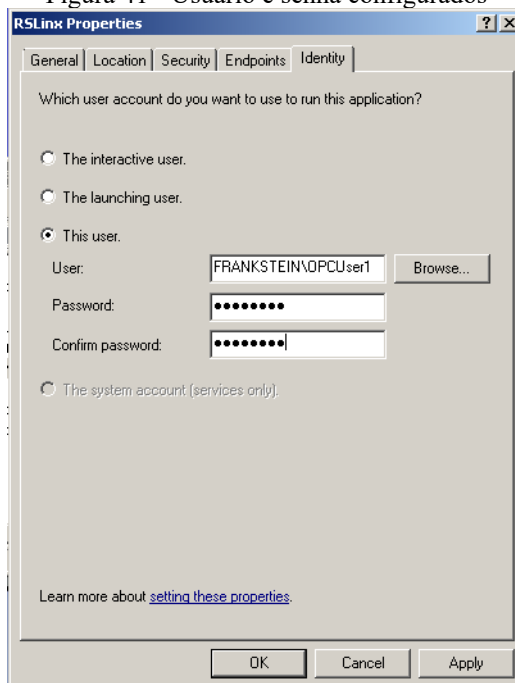


Fonte: Próprio Autor.

Agora repita o procedimento para os outros dois quadros permissões: “Access Permissions” e “Configuration Permissions”. Feito isso, a última aba a ser alterada é a aba “Identify”, deve ser selecionada a opção “This user” para configurar o usuário “OPCUser1” criado momentos atrás, conforme Figura 41.

Para facilitar, pode ser utilizado o botão “*Browse*” e apenas repetir o mesmo procedimento feito, para as permissões anteriores. A mesma senha deve ser informada e confirmada e, por fim, clique no botão “*Apply*”.

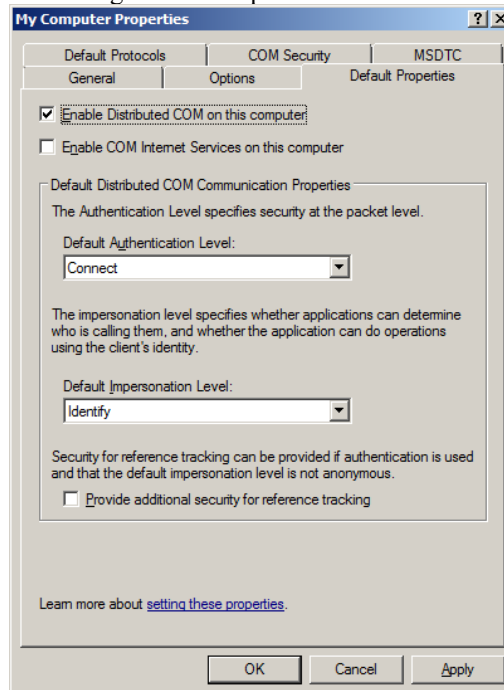
Figura 41 - Usuário e senha configurados



Fonte: Próprio Autor.

Agora que o usuário foi configurado para acessar o servidor *OPC*, é hora de configurar o sistema para distribuir o acesso. Na janela de configuração DCOM, clique com o botão direito em “*My Computer*” e selecione “*Properties*”. Nas propriedades que apareceram selecione a aba “*Default Properties*”, conforme apresentado pela Figura 42.

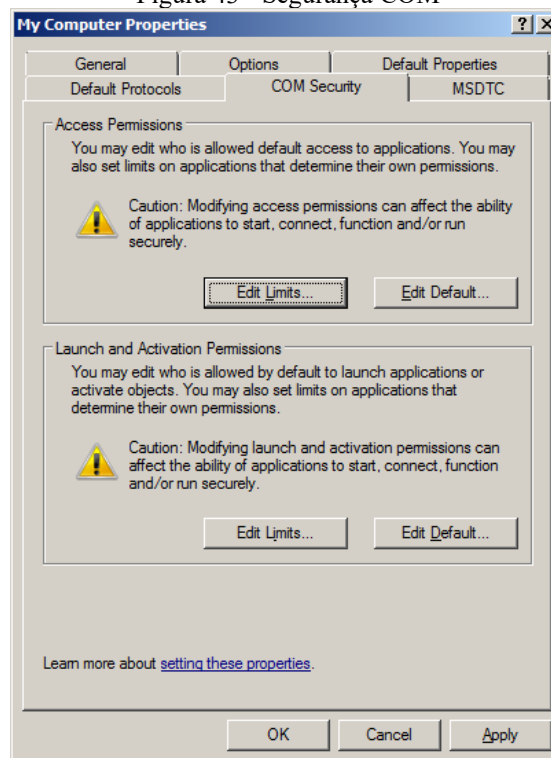
Figura 42 - Propriedades DCOM



Fonte: Próprio Autor.

Verifique que a opção “*Enable Distributed COM on this computer*” deve estar marcada e nas opções “*Default Authentication Level*” e “*Default Impersonation Level*” as opções “*Connect*” e “*Identify*” devem estar selecionadas respectivamente. Depois abra a aba “*COM Security*”, ilustrada pela Figura 43.

Figura 43 - Segurança COM



Fonte: Próprio Autor.

Nessa aba, as duas permissões devem ter seus limites alterados clicando em “*Edit Limits*”. Nos limites de “*Access Permissions*” selecione a opção “*ANONYMOUS LOGON*” e marque todas as permissões como “*Allow*”. Nos limites de “*Launch and Activation Permissions*” adicione o grupo OPCUsers como nos passos anteriores e marque todas as permissões com “*Allow*”.

Após ter-se o usuário e grupo de usuários *OPC* cadastrados e devidamente configurados, com as permissões dos programas, pode ser necessário aplicar a descoberta de rede para que o servidor fique visível na rede, para máquinas clientes, caso esteja instalado em uma máquina com Windows superior ao XP. Para fazer isso, basta ir até a “Central de Rede e Compartilhamento” (*Network and Sharing Center*), que está localizada dentro do Painel de Controle, conforme tela ilustrada da Figura 44, ou clicando com o botão direito em cima do ícone de rede, ao lado do relógio, no canto direito da barra de tarefas. Na janela aberta da Central, clique no link de nome “Alterar configurações avançadas de compartilhamento” (*Change advanced sharing settings*).

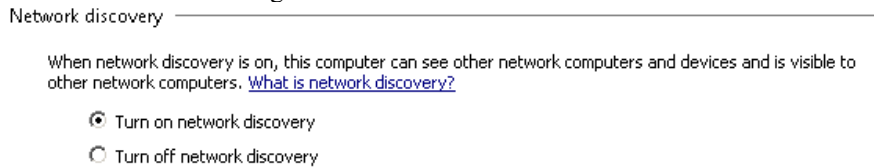
Figura 44 - Central de rede e compartilhamento



Fonte: Próprio Autor.

Nas opções avançadas de compartilhamento são exibidas opções relacionadas a cada tipo de rede, na rede que estiver trabalhando abra o menu “Descoberta de Rede” (*Network Discovery*) e selecione “Permitir descoberta de rede” (*Turn on Network Discovery*), conforme a Figura 45. A partir daí o computador será exibido no menu de rede dos computadores clientes.

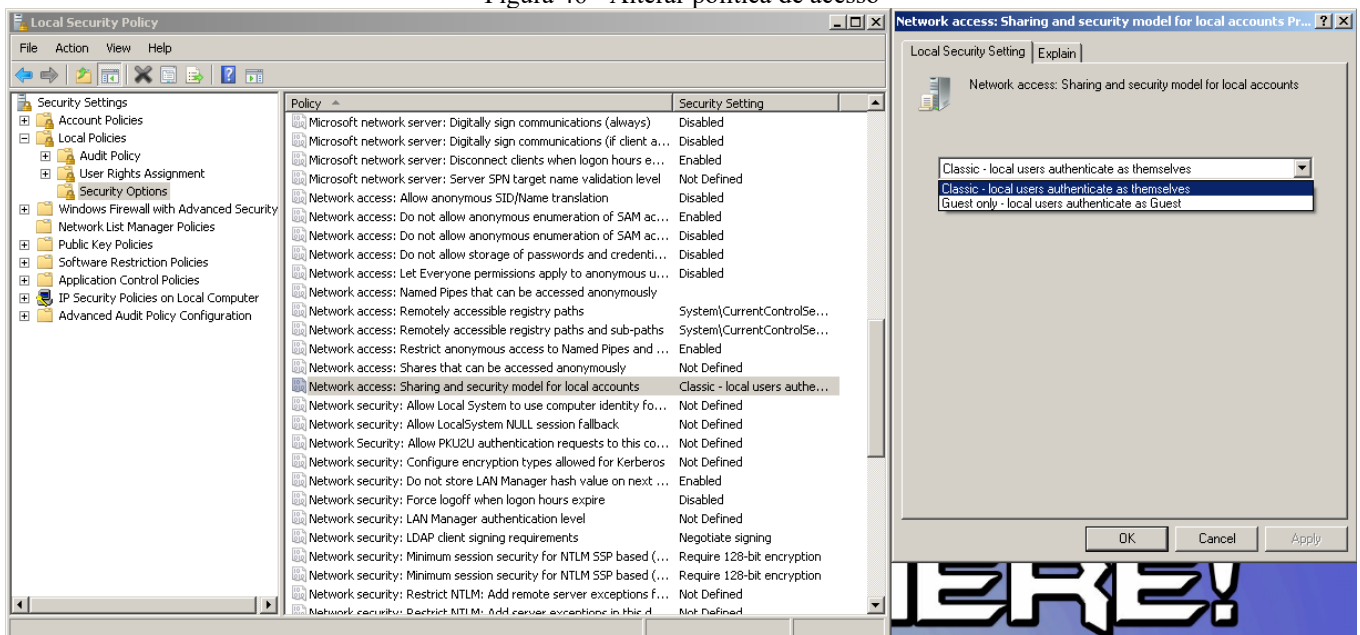
Figura 45 - Ativar descoberta da rede



Fonte: Próprio Autor.

O último passo para permitir que a máquina seja acessada, faz-se necessário alterar a política de segurança local, ilustrada pela Figura 46, para permitir o acesso completo do novo usuário a partir de qualquer computador da rede. Para acessar o menu de “*Network Security Policy*” basta abrir Iniciar → Executar e digitar: secpol.msc. Na janela aberta, posicione sobre “Opções de Segurança” (*Security Settings*) e expanda “Políticas Locais” (*Local Policies*). Depois selecione “Opções de Segurança” (*Security Options*). Na lista aberta, clique com o botão direito em “Acesso a rede: Modelo de segurança e compartilhamento para contas locais” (*Network access: Sharing and security model for local accounts*) e selecione “Propriedades”. Das opções disponíveis escolha “Classico – Usuários locais autenticam como eles próprios” (*Classic – local users authenticate as themselves*) e então, clique “OK”.

Figura 46 - Alterar política de acesso



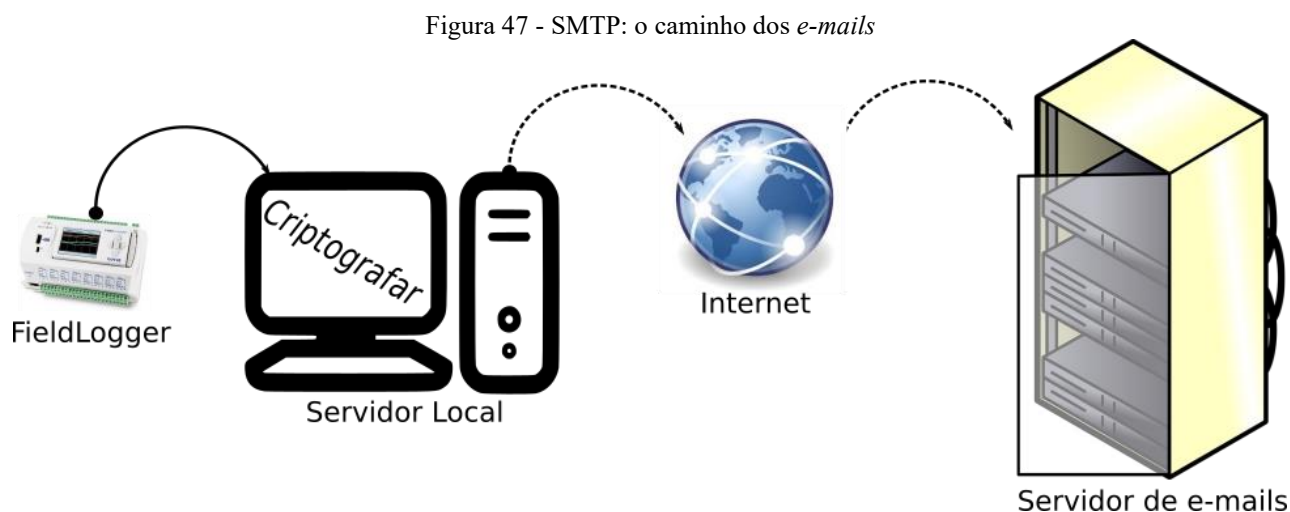
Fonte: Próprio Autor.

Feitos esses passos, apenas reinicie a máquina para carregar as novas configurações. Caso não consiga fazer o acesso ou queira configurar de forma diferente o servidor, reveja os passos ou leia os artigos citados anteriormente, pois contém informações sobre configurações diferentes que podem ser feitas.

## 5 PARTE LÓGICA 2 – SERVIÇOS DE REDE

### 5.1 SMTP – STUNNEL

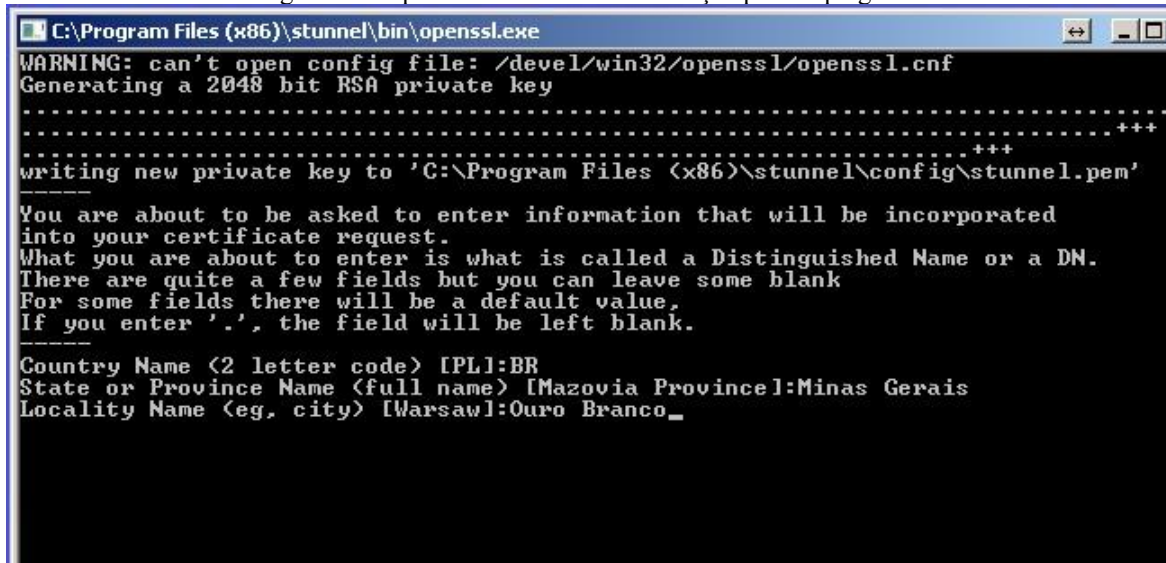
O protocolo SMTP é responsável por transmitir as mensagens de e-mail entre servidores. O FieldLogger possui a capacidade de enviar mensagens de e-mail, utilizando o protocolo SMTP, quando uma situação de alarme acontece. Um problema encontrado durante os testes é que todos os servidores de e-mail atuais, devido à necessidade cada vez maior de segurança na internet, precisam de criptografia para enviar mensagens e não aceitam outro método de envio. O FieldLogger não possui esse tipo de criptografia, ele se conecta ao servidor utilizando o padrão “AUTH LOGIN” do SMTP, isento de criptografia, e envia as mensagens diretamente para o servidor que estiver cadastrado. Uma solução encontrada nesse caso foi instalar um programa que faça essa criptografia antes de encaminhar as mensagens para um servidor externo de e-mails. A Figura 47 ilustra o caminho dos e-mails na rede criada.



Fonte: Próprio Autor.

Para criar esse servidor de criptografia, instalou-se o programa Stunnel, ele permite que seja configurado um servidor de e-mails no computador, que está ligado à rede. Esse servidor recebe as requisições de e-mail feitas pelo FieldLogger, criptografa as mensagens e as encaminha corretamente para o verdadeiro servidor de e-mails na internet (no caso o Gmail, servidor de e-mails da Google, que foi utilizado no referido exemplo). Lembrando que a criptografia do Stunnel é feita sobre a ferramenta de autenticação usando a biblioteca OpenSSL, que deve ser gratuitamente baixada e instalada através do link: [https://sourceforge.net/projects/gnuwin32/files/openssl/0.9.8h-1/openssl-0.9.8h-1-setup.exe/download?use\\_mirror=ufpr](https://sourceforge.net/projects/gnuwin32/files/openssl/0.9.8h-1/openssl-0.9.8h-1-setup.exe/download?use_mirror=ufpr), onde depois de instalada deverá ter as credenciais configuradas por um passo a passo bem simples, apenas preenchendo os espaços corretamente, conforme ilustração da Figura 48.

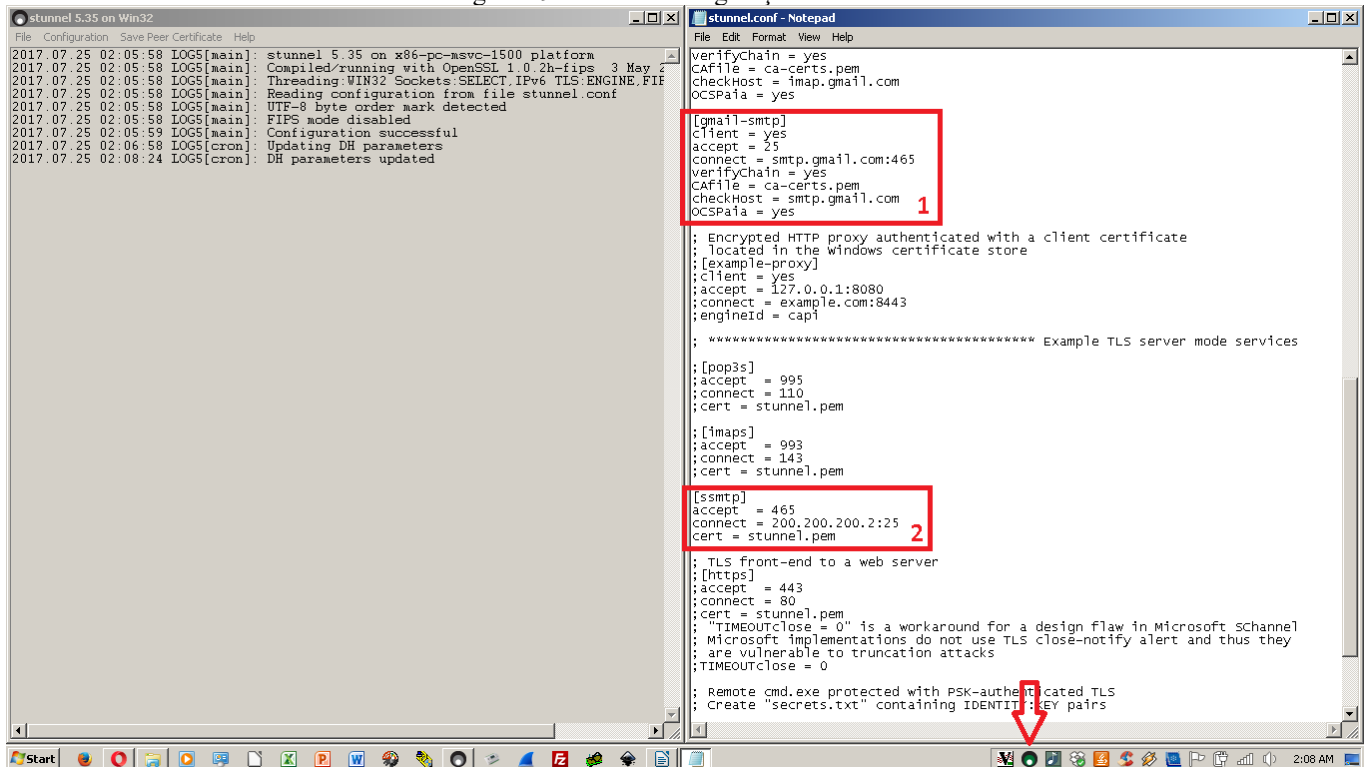
Figura 48 - OpenSSL: criando a certificação para criptografia



Fonte: Próprio Autor.

Depois de criada a certificação, o servidor de e-mails Stunnel já pode ser configurado normalmente (Figura 49), para receber os e-mails sem criptografia enviados pelo FieldLogger, criptografá-los e direcioná-los para o devido servidor externo (Gmail, Outlook, Yahoo, etc.).

Figura 49 - SMTP: configuração do Stunnel

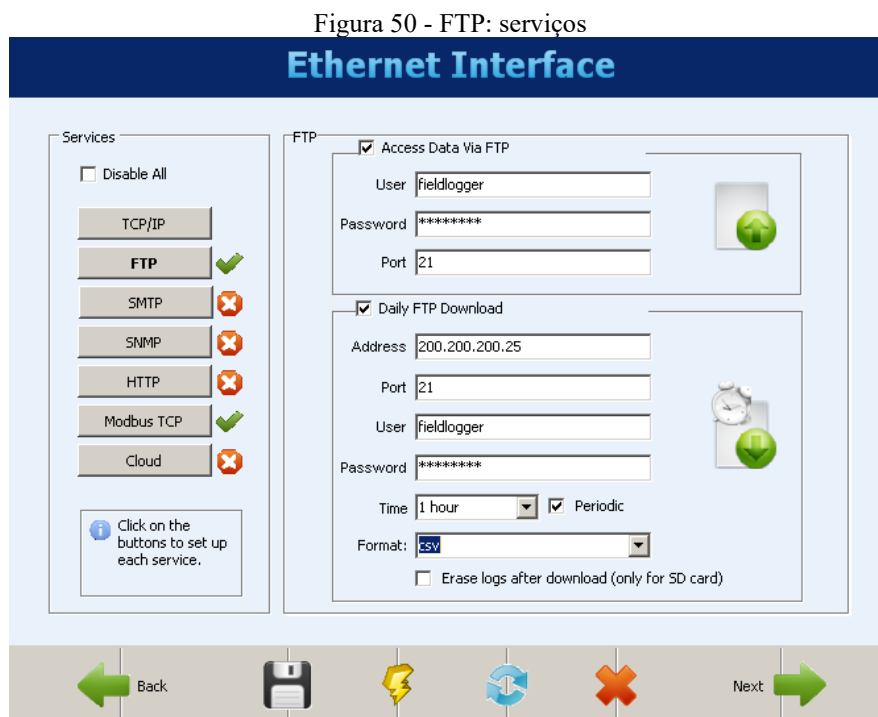


Fonte: Próprio Autor.

Na imagem da Figura 49 tem-se à esquerda, a tela principal do Stunnel, onde através do botão “*Configuration*” é possível editar o arquivo de configuração, que se encontra aberto à esquerda na imagem. Os dois quadros marcados representam a entrada e a saída de *e-mails* do servidor. O quadro 1 representa a configuração de saída do servidor, como ele vai se conectar ao servidor externo (Gmail) e qual o arquivo geral de certificação que utilizará na criptografia. Já o quadro 2 representa o funcionamento do servidor de *e-mails* local, mostrando de qual IP ele receberá as mensagens e qual arquivo de certificado utilizará para criptografar a mensagem. Como no caso do exemplo (200.200.200.2:25) ele está recebendo do FieldLogger no IP = 200.200.200.2 e na porta 25. A seta vermelha mostra o ícone do Stunnel na cor verde, mostrando que o servidor está ativo e aguardando a chegada de novas mensagens.

## 5.2 FTP – FILEZILLA

O protocolo FTP é utilizado para transferências de arquivo entre computadores ligados em rede. Atuando em uma arquitetura Cliente-Servidor, o protocolo pode ser utilizado para transferir arquivos entre o computador e o FieldLogger. Quando é necessário acessar e copiar os registros que o FieldLogger fez, o protocolo FTP pode ser utilizado, com o auxílio do programa FileZilla, como ilustra a Figura 50. Nesse caso, existem duas formas de acesso, uma quando o FieldLogger atua como servidor, tendo o FileZilla como seu cliente e outra onde o FieldLogger é o cliente do servidor FTP, criado pelo FileZilla.



Fonte: Próprio Autor.

Para habilitar as funções do protocolo FTP no FieldLogger, utiliza-se o FLConfig, e na etapa de configuração dos serviços da interface Ethernet é possível habilitar as duas funções do protocolo no FieldLogger. A primeira opção “Acesso dados via FTP” é relacionada ao FieldLogger atuando como servidor (Figura 51), onde é possível definir o nome de usuário, a senha e a porta de comunicação que será usada na interface Ethernet, que por padrão é utilizada a porta “21”. A segunda opção “Download FTP diário”, apresentada pela Figura 52, é relacionada ao FieldLogger atuando como cliente, ela é responsável por programar o envio dos dados de acordo com um período, podendo ser uma hora do dia ou em determinados intervalos de tempo, enviando os dados para o servidor criado com o programa FileZilla.

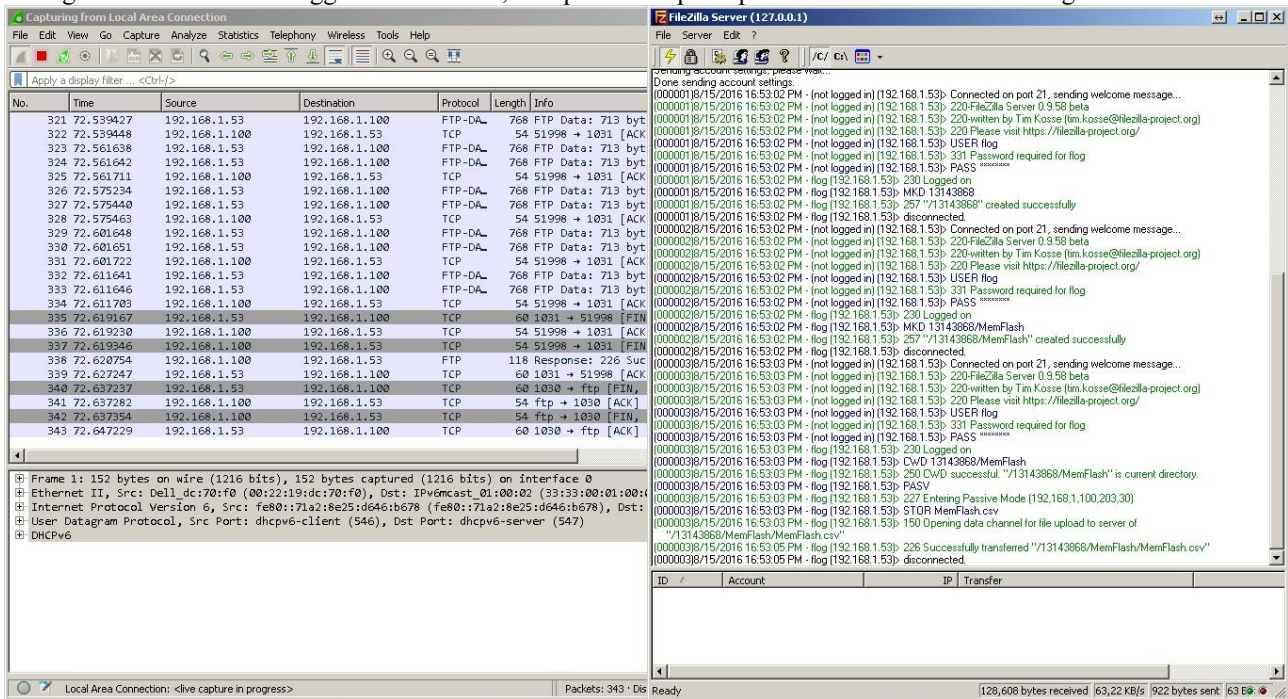
Figura 51 - FTP: FieldLogger como servidor

The screenshot shows a web browser window with a file named 'MemFlash.csv' displayed. Below the file icon is a table with the following data:

	A	B	C	D	E	F	G	H	I
1	FieldLogger								
2	DATE	TIME	A8	D7	RHT1-Temp	RHT2-Temp	RHT1-Umid	RHT2-Umid	N1200
3	03/09/2016	17:30:18	-1	0	30.6	33.2	44.7	38.5	33.2
4	03/09/2016	17:30:20	-1	0	30.6	33.1	44.7	38.5	33.2
5	03/09/2016	17:30:22	-1	0	30.6	33.2	44.7	38.5	33.3
6	03/09/2016	17:30:24	-1	0	30.6	33.2	44.7	38.5	33.3
7	03/09/2016	17:30:26	-1	0	30.6	33.2	44.8	38.5	33.3
8	03/09/2016	17:30:28	-1	0	30.6	33.2	44.8	38.5	33.3
9	03/09/2016	17:30:30	-1	0	30.6	33.2	44.8	38.5	33.1
10	03/09/2016	17:30:32	-1	0	30.6	33.2	44.9	38.5	33.1

Fonte: Próprio Autor.

Figura 52 - FTP: FieldLogger como cliente; à esquerda a captura pelo Wireshark e à direita o registro do FileZilla

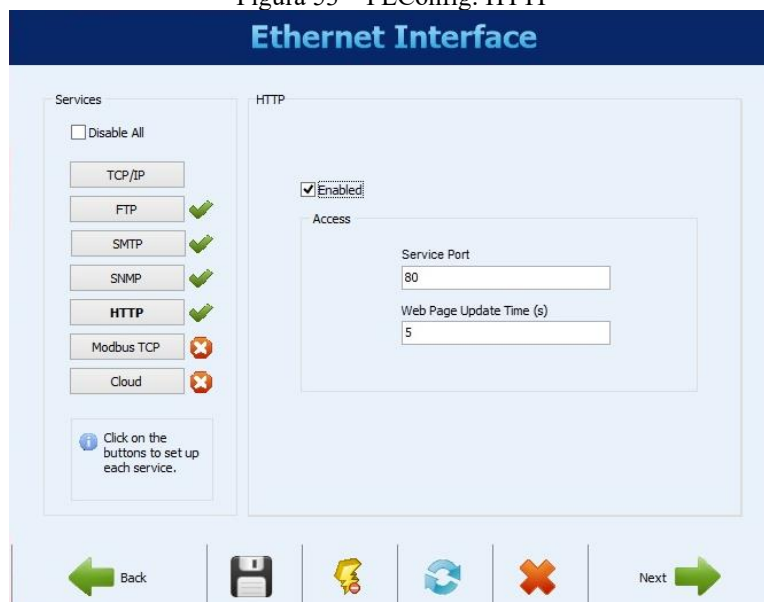


Fonte: Próprio Autor.

### 5.3 HTTP

Na configuração de HTTP (Figura 53), é possível configurar o FieldLogger para exibir uma página *web* com algumas informações simplificadas sobre alguns dos seus status. No FLConfig, basta selecionar HTTP nos serviços Ethernet e escolher a porta e a frequência de atualização dos dados da página, utilizando sempre como padrão a porta 80. A partir disso, digitando o IP do FieldLogger, em qualquer navegador, a página já será exibida.

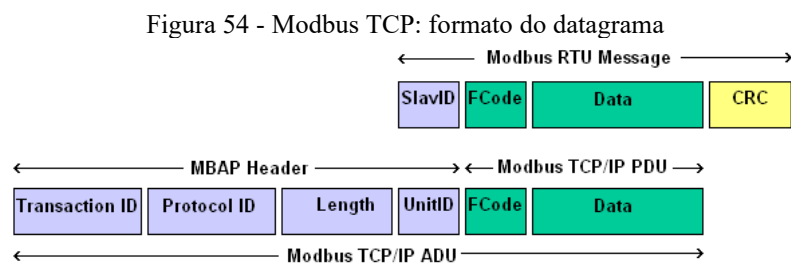
Figura 53 – FLConfig: HTTP



Fonte: Próprio Autor.

## 5.4 MODBUS TCP

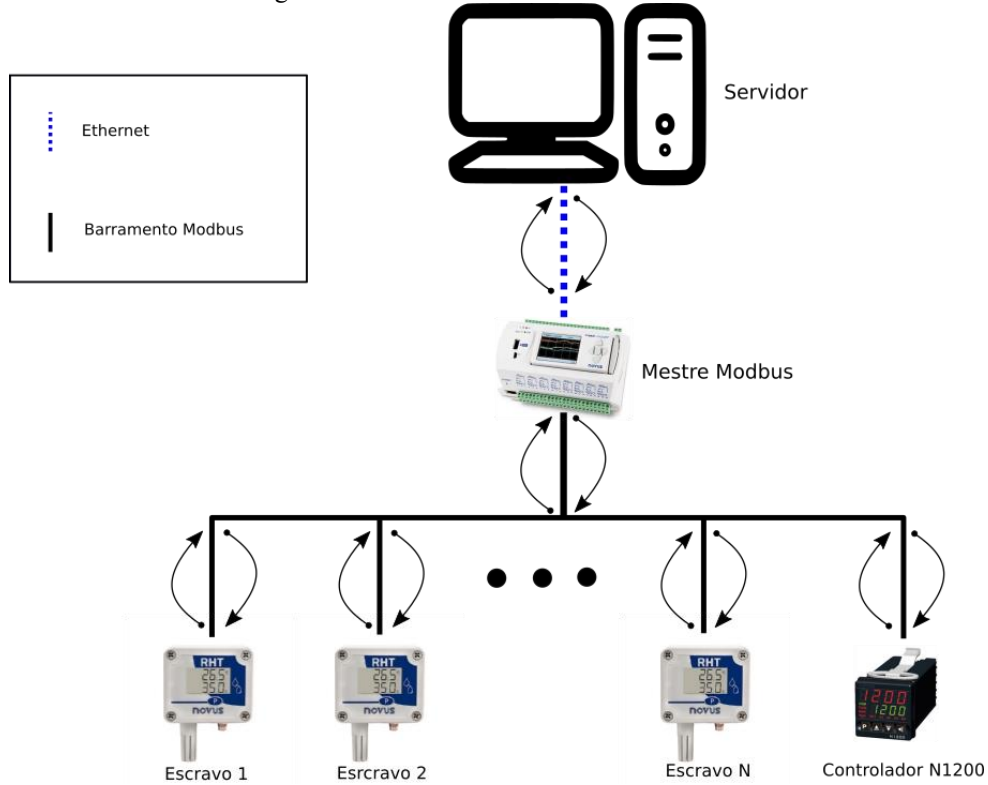
Como as redes Modbus e Ethernet são fisicamente separadas na plataforma e possuem maneiras distintas de transmitir dados, deve existir algo que torne possível essa comunicação. Nesse caso, o FieldLogger é o responsável pela comunicação e agindo como um *gateway*, transforma as requisições que chegam via rede Ethernet em datagramas Modbus. Tudo isso é feito graças a um protocolo especial chamado Modbus TCP, que encapsula comandos Modbus em pacotes Ethernet, permitindo que todo o controle dos dispositivos Modbus seja expandido para a rede Ethernet, possibilitando inclusive a utilização de sistemas supervisórios complexos que englobem a rede como um todo. A Figura 54 ilustra o formato do datagrama do Modbus TCP.



Fonte: [https://cdn.novusautomation.com/downloads/manual\\_fieldlogger\\_v19x\\_a\\_pt.pdf](https://cdn.novusautomation.com/downloads/manual_fieldlogger_v19x_a_pt.pdf)

Como a comunicação do *gateway* é bidirecional, ou seja, no caso do módulo, ela segue tanto no sentido da rede Modbus para a Ethernet, como o contrário, os pacotes estão a todo instante sendo encapsulados e desencapsulados pelo FieldLogger. No caso de um sistema onde está presente um computador, com um sistema supervisório instalado, controlando e monitorando a rede, tem-se a característica de arquitetura multi-mestre, onde mais de um mestre de rede envia comandos e recebe respostas dos escravos. A Figura 55 representa uma comunicação que está acontecendo em um instante de requisição do sistema supervisório para um determinado escravo da rede Modbus, os dois estão em redes separadas, mas o FieldLogger recebe os pacotes Ethernet encapsulados em Modbus TCP e desencapsula os pacotes em datagramas Modbus, para direcionar ao devido escravo o comando requerido pelo supervisório. A resposta é enviada seguindo o caminho inverso.

Figura 55 - Modbus TCP: caminho dos dados



Fonte: [https://cdn.novusautomation.com/downloads/manual\\_fieldlogger\\_v19x\\_a\\_pt.pdf](https://cdn.novusautomation.com/downloads/manual_fieldlogger_v19x_a_pt.pdf)

A característica multi-mestre fica mais evidente quando se precisa requisitar algo diretamente ao FieldLogger pelo supervisor. Como a rede Modbus só pode ter um mestre, para fazer uma requisição direta é necessário utilizar o endereço 255, diferente dos endereços dos escravos que podem ir de 1 até 247. Comandos assim podem ser úteis quando o operador deseja trabalhar com as entradas digitais ou analógicas do FieldLogger ou quando precisa alterar o estado dos relés. Os comandos Modbus são padronizados e podem ser comparados pela Tabela 1.

Tabela 1 – Comandos Modbus

01	Leitura de bloco de bits	Bobina (Coil)
02	Leitura de bloco de bits	Entradas discretas
03	Leitura de bloco de registradores	Registradores Retentivos (Holding Registers)
04	Leitura de bloco de registradores	Registradores de Entrada (Input Registers)
05	Escrita de bit único	Bobina (Coil)
06	Escrita de registrador único	Registradores Retentivos (Holding Registers)
15	Escrita em bloco de bits	Bobina (Coil)
16	Escrita em bloco de registradores	Registradores Retentivos (Holding Registers)

Fonte: MODBUS ORGANIZATION, 2012.

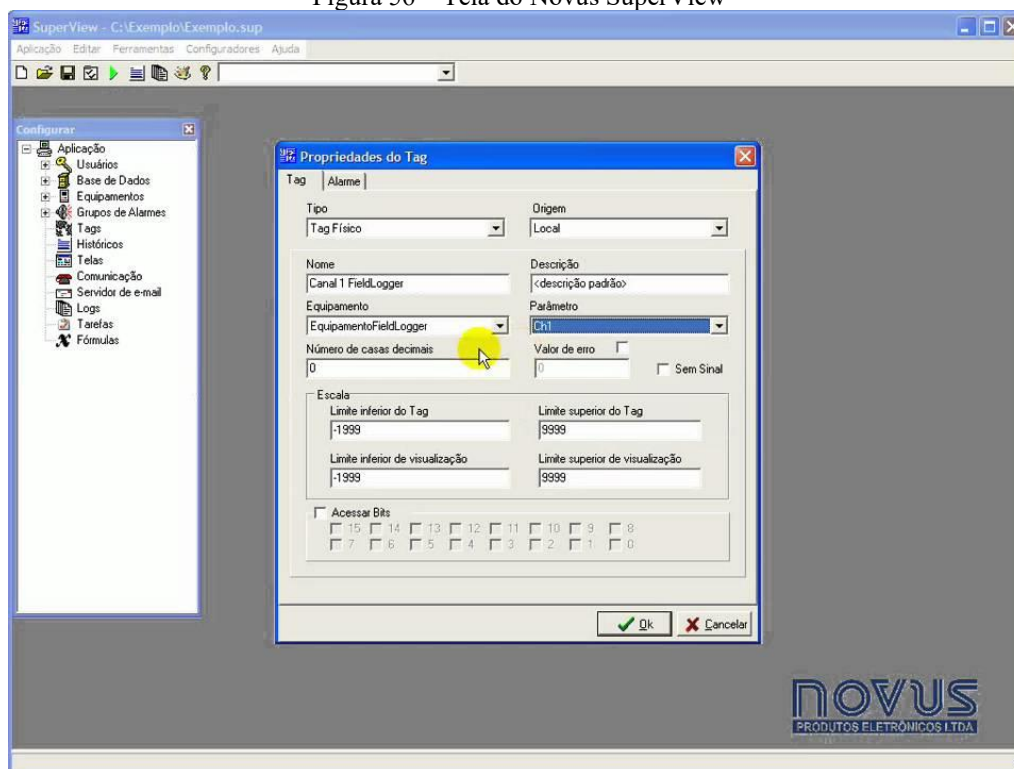
## 6 PARTE LÓGICA 3 – INTEGRAÇÃO

### 6.1 SISTEMA SUPERVISÓRIO

Como citado anteriormente, o sistema supervisório é responsável por combinar a supervisão e o controle dos equipamentos na rede, trabalhando junto com o protocolo Modbus TCP permite que o módulo seja utilizado e monitorado com os outros equipamentos presentes no laboratório. Durante os testes foram utilizados três *softwares* para criação de sistemas supervisórios sendo eles:

- a- Novus SuperView: o SuperView, ilustrado na Figura 56, é o *software* disponibilizado pela Novus, empresa fabricante da maioria dos componentes do módulo, que permite a criação e configuração de um sistema supervisório capaz de monitorar os equipamentos desenvolvidos por ela. Ele funcionou muito bem com todos os equipamentos do módulo, mas não é compatível com os MICA's, logo foi pouco utilizado.

Figura 56 – Tela do Novus SuperView

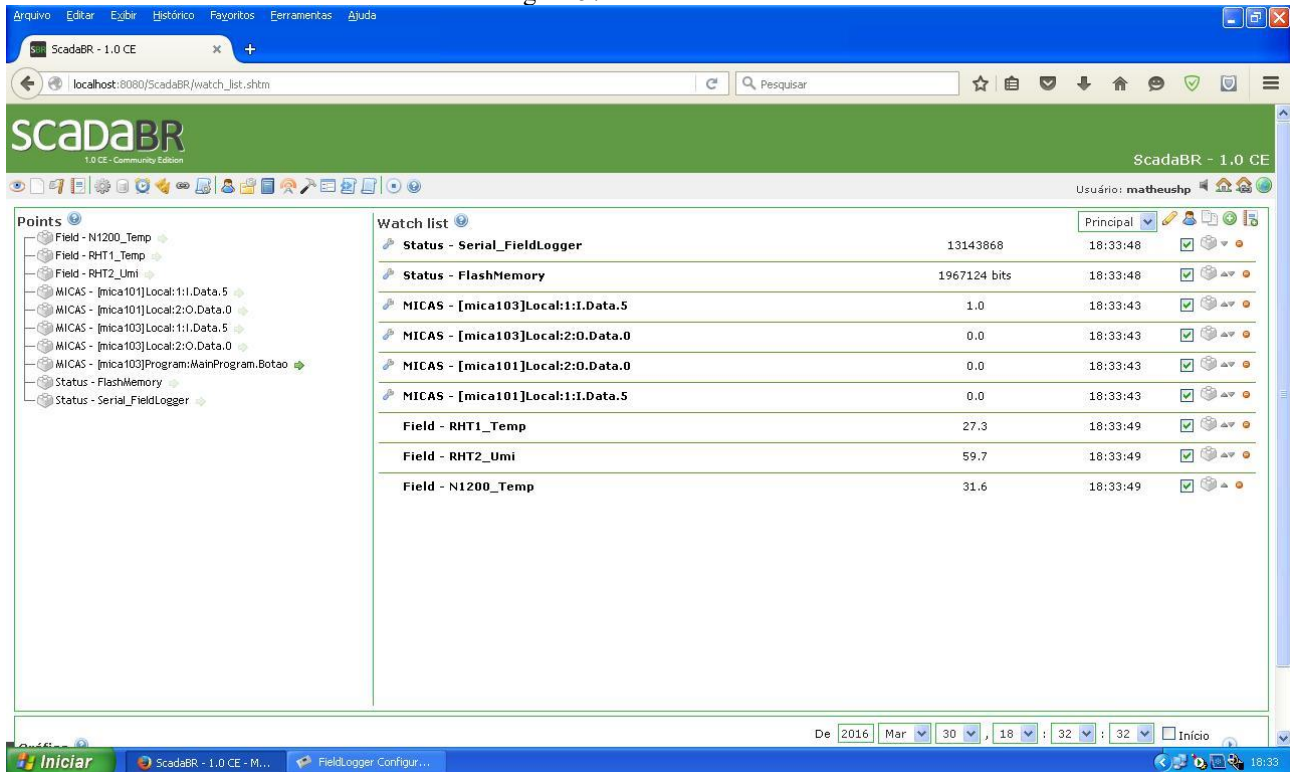


Fonte: [https://cdn.novusautomation.com/downloads/manual\\_fieldlogger\\_v19x\\_a\\_pt.pdf](https://cdn.novusautomation.com/downloads/manual_fieldlogger_v19x_a_pt.pdf)

- b- ScadaBR: o ScadaBR, ilustrado pela Figura 57, é um *software* nacional de código livre e independente de plataforma, funciona tanto em sistemas Microsoft quanto em sistemas livres como o Linux. Ele consegue isso por trabalhar via *browser*, sendo gerenciado pelo container de aplicações chamado TomCat. Seria a melhor opção em vista de custo e liberdade de sistema para o projeto, mas até a data dos testes, os desenvolvedores não tinham portado o programa para a versão mais atual da linguagem Java (o *software* funciona apenas na versão 7), com isso a funcionalidade de acesso ao servidor *OPC* (*OPC/DA*) apresentava extrema lentidão quando

tinha as *tags* dos CLP's inseridas para monitoramento, impedindo que fosse utilizado para trabalhar com os MICA's. Apesar a disso, os comandos Modbus funcionaram normalmente e alguns testes foram feitos, mas pela necessidade do projeto de integrar todos os equipamentos do laboratório, o ScadaBR não foi utilizado como *software* final.

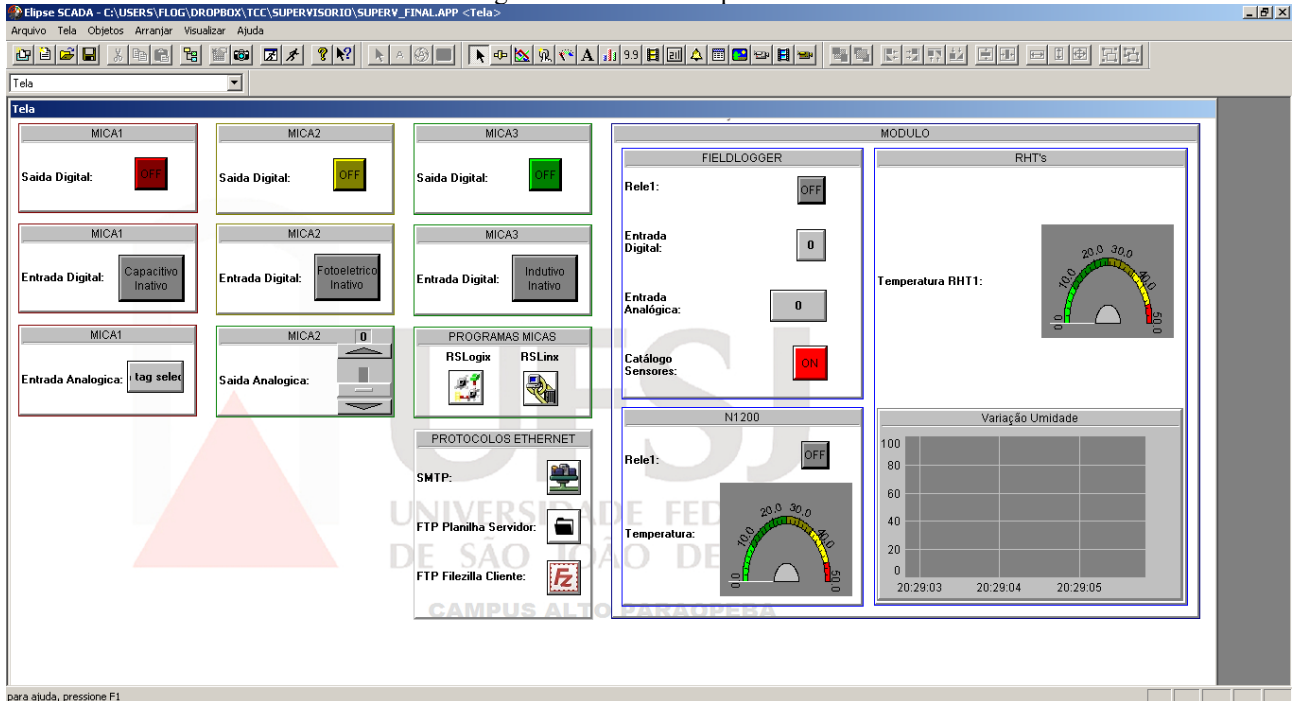
Figura 57 -Tela do ScadaBr



Fontes: [https://cdn.novusautomation.com/downloads/manual\\_fieldlogger\\_v19x\\_a\\_pt.pdf](https://cdn.novusautomation.com/downloads/manual_fieldlogger_v19x_a_pt.pdf)

- c- Elipse Scada: o Elipse Scada, ilustrado pela Figura 58, é um *software* proprietário e funciona apenas em sistemas Microsoft, mas foi o *software* que melhor se comportou quando a rede toda foi implementada. Ele é amplamente utilizado na indústria e também na academia, principalmente em aulas de disciplinas que desejam apresentar o conceito de sistema supervisorio. A versão de demonstração (demo) permite que sejam configuradas até 20 *tags* de equipamentos, sendo suficientes para permitir que uma rede simples seja criada e configurada com diferentes equipamentos e alguns sensores para ilustrar o seu funcionamento.

Figura 58 – Tela do Elipse Scada



Fonte: Próprio Autor.

## 6.2 DRIVER MODBUS DO ELIPSE SCADA

Esta etapa, contempla alguns passos na criação do sistema supervisorio utilizando o Elipse Scada. Depois da instalação simples do Elipse Scada, o primeiro passo consiste em instalar o mestre Modbus para que seja possível enviar e receber comandos Modbus do módulo. Primeiro, é necessário criar uma conta no site da empresa, gratuitamente, para em seguida, ser possível efetuar o *download* do *Driver* Modicon Modbus Master na página: <https://www.elipse.com.br/downloads/?key=modbus&language=ptbr#header-main>, ilustrada pela Figura 59.

Figura 59 - *Download* do *driver* Modbus no site do Elipse Scada

Drivers

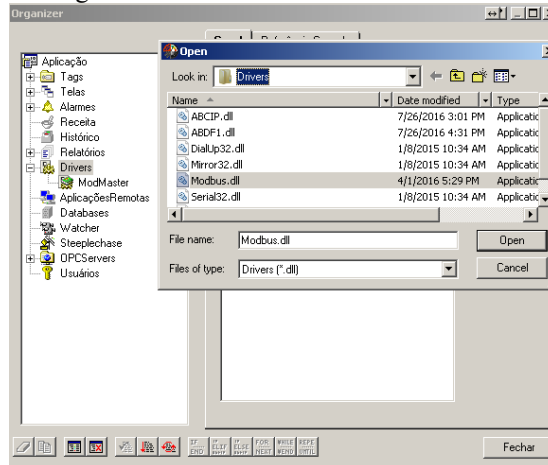
Descrição	Tamanho	Versão	Link
Driver Modicon Modbus Master (ASC/RTU/TCP) Modbus.dll	5.87 MB	v3.1.36	<a href="#">Download</a>
Fabricante: Modicon			
Protocolos: Modbus ASC, Modbus RTU, Modbus TCP			

Fonte: Próprio Autor.

Depois de feito o *download*, coloca-se o arquivo “Modbus.dll” em uma pasta no computador, para que fique fácil encontrá-lo novamente, no caso do exemplo, o arquivo foi colocado na própria pasta de instalação do Elipse Scada. Depois disso, basta iniciar o Elipse Scada e no novo projeto criado, abrir o Organizer, pelo menu ou pelo atalho “Alt+O”. Dentro do Organizer, clicar na opção “*Drivers*”, dentro dela a aba geral será aberta na lateral direita da janela, para tanto, basta apenas clicar em “Novo” e selecionar o

arquivo “Modbus.dll”, que foi baixado anteriormente. O novo *driver* agora aparecerá na lista de *drivers*, com isso, basta selecionar o novo *driver* Modbus e clicar em “Configurar” para poder alterar seu nome e o deixar pronto para ser utilizado, conforme ilustra a Figura 60.

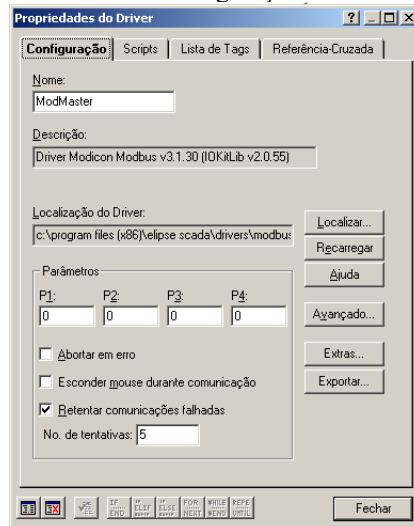
Figura 60 - Adicionando *driver* Modbus.dll



Fonte: Próprio Autor.

Depois que o menu Configurar for aberto, é necessário alterar o nome do *driver*. Depois de alterar o nome, clique no botão “Extras” para ter acesso às configurações de rede do *driver*, conforme ilustra a Figura 61.

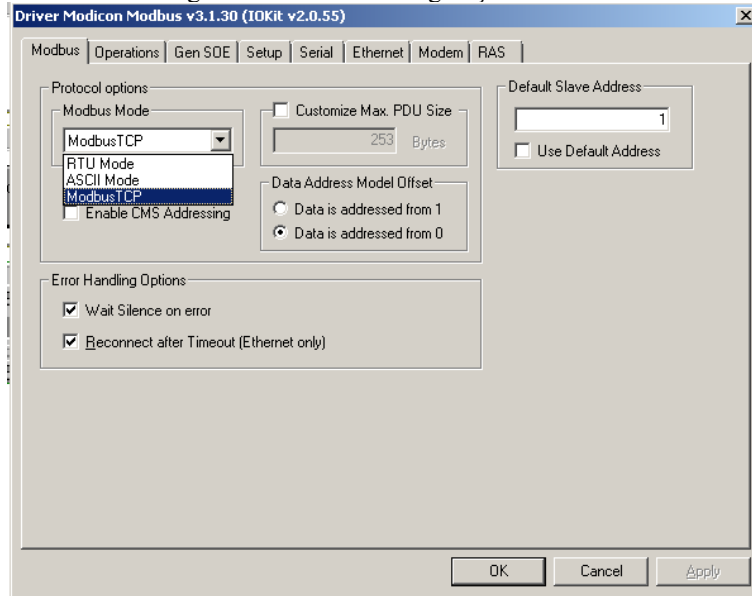
Figura 61 - Menu de configuração do *driver* Modbus



Fonte: Próprio Autor.

O próximo passo, consiste na realização da configuração de como o *driver* Modbus vai atuar em cada aba de configuração. Na primeira aba é necessário selecionar o modo “Modbus TCP”, as outras opções podem ser deixadas conforme a Figura 62.

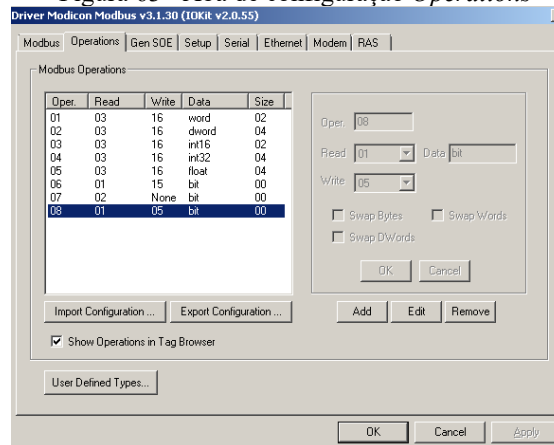
Figura 62 - Aba de configuração Modbus



Fonte: <https://www.elipse.com.br/downloads/?key=modbus&language=ptbr#header-main>,

Na aba “*Operations*”, ficam catalogadas as operações do padrão Modbus que podem ser realizadas para leitura e escrita em memória, por exemplo. Além das operações padrão, que podem ser confirmadas na Tabela 1, ainda foi criada a operação de número 8, necessária para fazer a escrita dos relés do controlador N1200, pois, não segue o padrão normal de acesso. Para criar operações basta clicar em “*Add*”, preencher os dados necessários e depois clicar em “*OK*”, como ilustra a Figura 63.

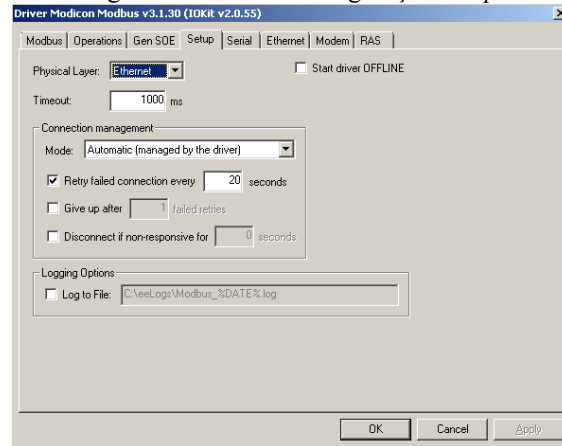
Figura 63 - Aba de configuração *Operations*



Fonte: Próprio Autor.

A aba “*Gen SOE*” contém apenas uma opção que por padrão fica marcada. Continuando na aba “*Setup*”, a opção de “*Physical Layer*” deve ser marcada Ethernet, pois é a camada física de rede que se está usando pra comunicar com o mestre Modbus da plataforma. A opção de *Timeout* foi deixada em 1000ms e o modo de conexão *Automatic*, conforme ilustra a Figura 64.

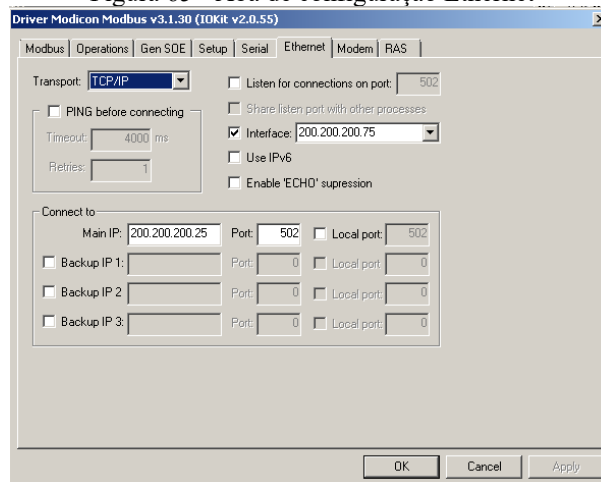
Figura 64 - Aba de configuração *Setup*



Fonte: Próprio Autor.

As quatro próximas abas são responsáveis por cada um dos tipos de rede possíveis de se trabalhar com o *driver* Modbus, como o acesso foi realizado via Ethernet, a única aba que será necessário alterar será a aba de mesmo nome. Assim, na aba Ethernet é necessário selecionar o modo *Transport* como TCP/IP, marcar a caixa “*Interface*” e selecionar a placa de rede com o IP que está sendo utilizado para acessar a plataforma (Figura 65). Dentro do quadro “*Connect to*”, no campo “*Main IP*”, deve-se colocar o IP do mestre Modbus, que por padrão trabalha na porta 502.

Figura 65 - Aba de configuração Ethernet



Fonte: Próprio Autor.

Feitas essas configurações é só clicar em “*Apply*”, a partir daqui o *driver* Modbus estará configurado para começar a enviar e receber comandos Modbus para o mestre e escravos da rede.

### 6.3 EXEMPLOS

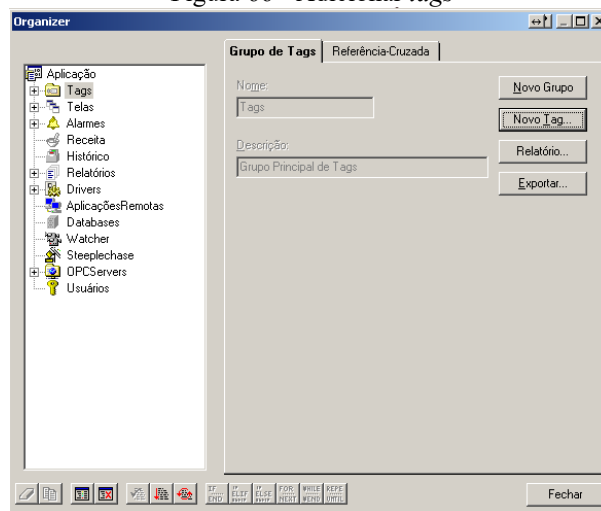
Depois de configurado completamente para trabalhar com a rede Modbus via Ethernet (sem fio ou cabeada), foram apresentados exemplos, baseados no sistema supervisorio criado, para demonstração dos

testes do módulo completo, dentro da rede criada no laboratório. Para cada função que será desempenhada dentro do supervisor é necessário criar uma *tag*, caso queira visualizar o dado de um escravo, dar um comando para alterar o estado de um relé, alterar ou visualizar o estado de uma porta digital, entre outras funções, sempre deve ser feito dessa forma, cria-se a *tag* referente a função que ela desempenhará, depois se cria um botão ou um quadro para visualizar o valor e associa-se a *tag* ao botão ou painel de visualização criado.

### 6.3.1 Primeiro exemplo: ler dado escravo Modbus:

Para fazer a leitura de um dado em um escravo Modbus, precisa-se primeiro criar a *tag* que será a referência para essa função de leitura dos dados do sensor. As *tags* podem ser criadas sozinhas ou em grupos, quando se tem vários dispositivos e necessita de uma organização mais completa. Nesse caso do exemplo, foi criada apenas uma *tag* simples. Para tanto, é necessário abrir o Organizer pelo menu do sistema ou pelo atalho “Alt+O”, e na janela aberta, clicar sobre o campo *Tags* e no painel direito a opção “Novo Tag”, para ser selecionada, conforme ilustração da tela na Figura 66.

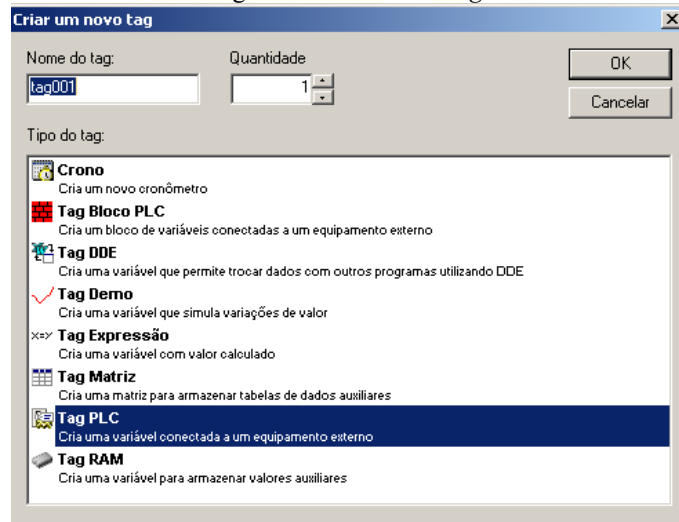
Figura 66 - Adicionar *tags*



Fonte: Próprio Autor.

Ao clicar na opção “Novo Tag”, uma janela se abrirá com as opções para criar um *tag* (Figura 67). É necessário editar o nome do novo *tag*, colocando algo que faça alusão à função que vai desempenhar para facilitar a localização, quando o *tag* for utilizada, e também selecionar “Tag PLC” como tipo da *tag* que será criada, pois representa uma variável localizada em um equipamento externo.

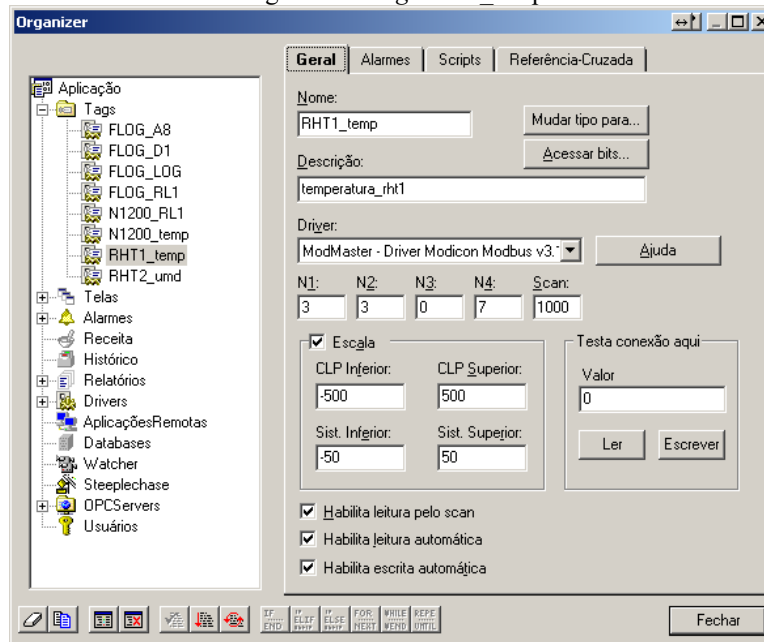
Figura 67 - Adicionar *tags*



Fonte: Próprio Autor.

Depois que for criada, a *tag* começa a aparecer na lista de *tags* e deve ser editada para exercer sua função de forma correta. Nesse exemplo, editou-se a *tag* responsável por mostrar no *gauge* o valor de temperatura do primeiro RHT (sensor de temperatura e umidade da empresa Novus), logo chamou-se o *tag* de “RHT1\_temp”, como pode ser visto na lista de *tags* da Figura 68.

Figura 68 - *Tag RHT1 temp*



Fonte: Próprio Autor.

Ao iniciar as opções de cima pra baixo, da tela da Figura 68, logo após o nome do *tag*, é possível colocar uma descrição para facilitar o entendimento do que o *tag* representa, logo abaixo a opção de qual *driver* será utilizado para trabalhar com o *tag*. Como a rede é do tipo Modbus é necessário escolher o *driver*

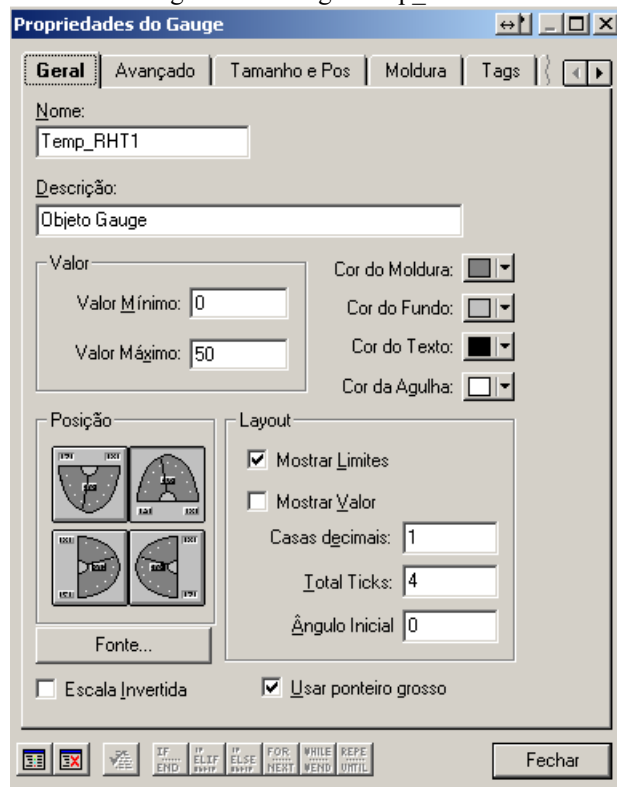
ModMaster que foi adicionado anteriormente. Depois tem-se cinco parâmetros de trabalho, N1, N2, N3, N4 e Scan:

- a- o parâmetro N1: É responsável por definir para qual endereço do dispositivo Modbus estamos fazendo a requisição;
- b- o parâmetro N2: É responsável por definir qual função Modbus estamos desempenhando na requisição;
- c- o parâmetro N3: É sempre nulo;
- d- o parâmetro N4: É responsável por definir para qual o endereço do registrador presente no dispositivo Modbus a requisição de dados será feita;
- e- o parâmetro Scan: É responsável por definir o intervalo de tempo entre uma requisição e outra.

No referido exemplo, N1 é igual a 3, pois o endereço de rede Modbus do primeiro RHT é 1. Como pretende-se fazer uma leitura de dados, a função Modbus de leitura de registradores (*Read Holding Register*), tem o parâmetro N2 é igual à função 3. O N3 é sempre zero e como o dado que se quer obter é o dado de temperatura, seta-se então N4 igual a 7, pois é o valor do registrador responsável por repassar o valor medido de temperatura do RHT. O Scan foi setado em 1000ms, valor que é o dobro do valor de atualização configurado no FieldLogger para evitar falhas de comunicação. É importante notar também, ainda na Figura 68, que no referido exemplo, alterou-se a faixa de escala, para que fique mais enxuta ao ser repassada ao *gauge*. Como os valores obtidos dos registradores dos escravos são em três casas, devido a sua resolução, informa-se os valores limites de medição nos dois campos superiores (CLP Inferior e CLP Superior) e nos dois campos inferiores, é necessário informar os valores que se quer mostrar no *gauge* de forma equivalente (Sist. Inferior e Sist. Superior). Para finalizar, todas as caixas de marcação devem estar marcadas, quando disponíveis, para que o sistema possa ter total controle do *tag* e também é possível testar a conexão para saber se está tudo configurado corretamente.

Depois de criado o *tag* de leitura da temperatura, é hora de criar o *gauge* que vai receber e repassar essas informações na tela. Para criar o *gauge* pode ser utilizado o botão de criação no menu, ao clicar na tela e arrastar ficará definido o tamanho. A primeira aba de configuração do novo *gauge*, aparece as opções de escala e *layout*, podendo ser ajustadas ao gosto de quem está criando o supervisão. No caso do exemplo, como não serão medidos valores negativos a escala foi escolhida de 0 a 50 e uma casa decimal para representação, conforme ilustra a Figura 69.

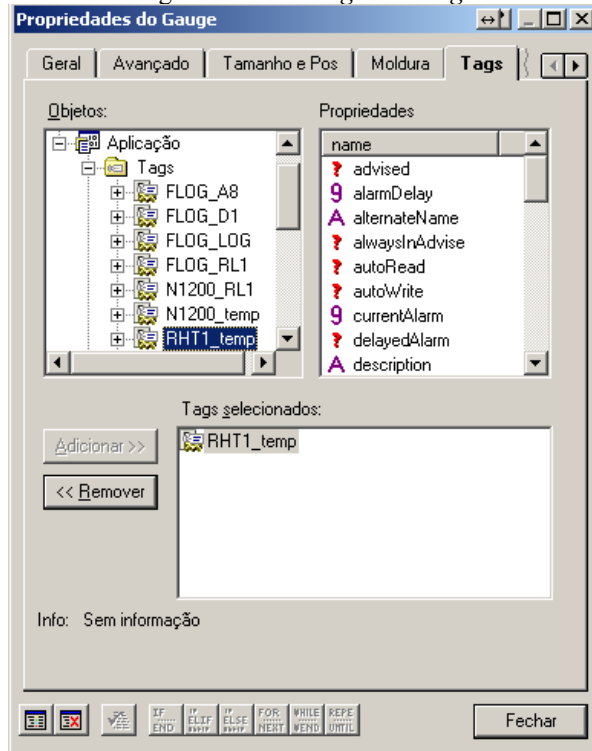
Figura 69 - Gauge Temp\_RHT1



Fonte: Próprio Autor.

As abas seguintes: “Avançado”, “Tamanho e Pos” e “Moldura” são referentes a opções gráficas e podem ser exploradas de acordo com o gosto de quem está desenvolvendo o supervisório. Já na aba “Tags”, deve-se abrir o menu “Aplicação”, chegar no menu “Tags” e abri-lo, tendo acesso ao “Tag RHT1\_temp”, criado anteriormente. Para relacionar o valor do “Tag RHT1\_temp” com o “gauge Temp\_RHT1”, basta seleccionar o tag e clicar em “Adicionar”. Pronto, agora quando o sistema estiver rodando será exibido o valor do tag pelo gauge, conforme apresenta a Figura 70.

Figura 70 - Aba Tags do Gauge

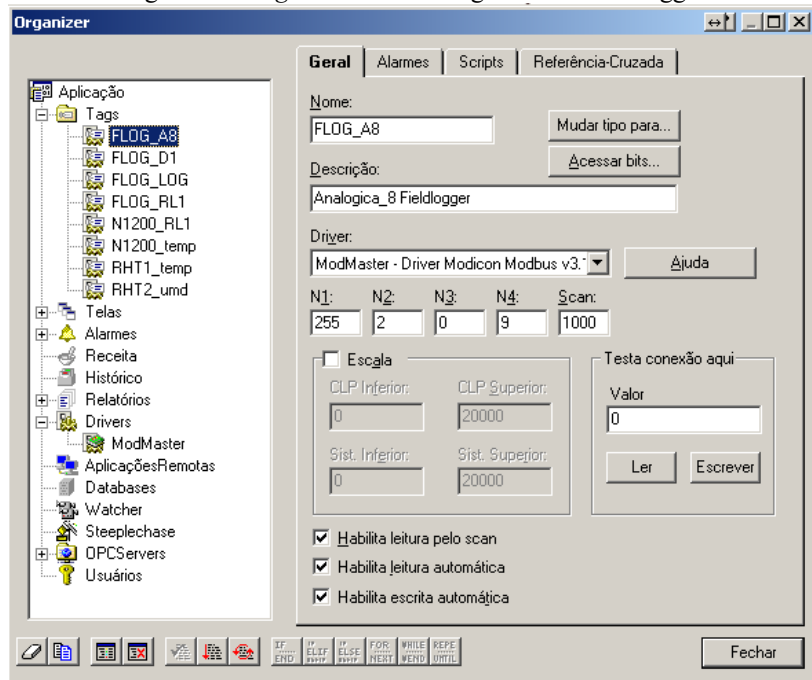


Fonte: Próprio Autor.

### 6.3.2 Segundo exemplo: receber valor de uma entrada analógica do FieldLogger:

O próximo passo consistiu na criação de uma *tag* para ler o valor de uma entrada analógica do FieldLogger. Para tanto, faz-se necessário abrir o Organizer e no menu de *tags*, criar um novo “*Tag PLC*”. Nesse novo *tag*, precisa-se definir o ModMaster como o *driver* a ser utilizado e nos parâmetros, definir N1 igual a 255. Toda vez que um comando Modbus cujo valor de endereço é igual a 255 é disparado na rede, fica entendido como um chamado ao mestre da rede, pois os escravos só podem ser endereçados de 1 a 247. O parâmetro N2 é igual a 2 pois representa a função de leitura de registrador com 4 bytes, quantidade suficiente para ler os dados armazenados no registrador (16 bits de dados). O N3 é sempre zero e o N4 é igual ao número do registrador que deseja visualizar, no caso o registrador 9 é o responsável por armazenar os valores escaneados pela entrada analógica 8. O intervalo de Scan é igual a 1000, o mesmo aplicado nos outros *tags*. Todos esses parâmetros podem ser visualizados pela tela da Figura 71.

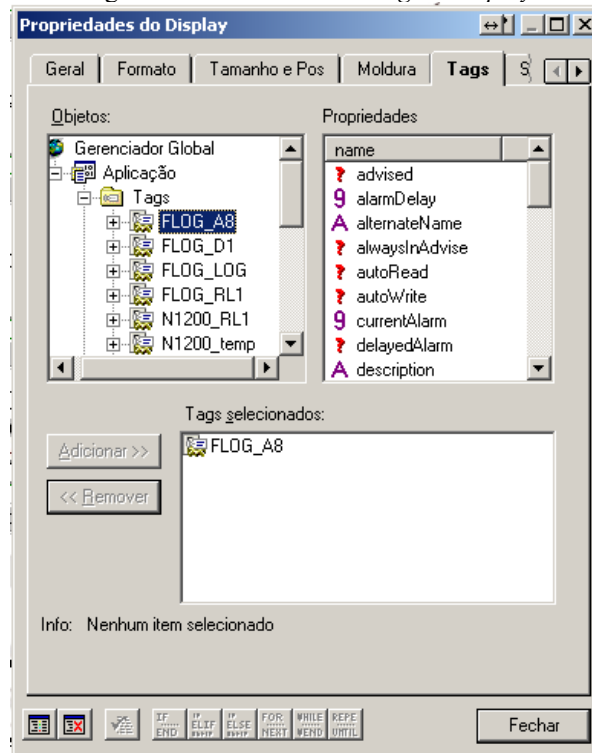
Figura 71 - Tag da entrada analógica 8 do FieldLogger



Fonte: Próprio Autor.

Depois de criada a *tag*, pode-se criar o *display* de visualização, para que o valor seja mostrado no supervisório. Para tanto, basta adicionar um novo *display* e arrastar para adequar o tamanho. Nas opções do novo *display* criado, a maior parte das abas são para personalização dos seus atributos visuais, mas na aba “Tags” (Figura 72), pode-se relacionar o *tag* criada com o *display*, simplesmente, expandindo o menu “Aplicação”, expandindo os *tags*, selecionando o *tag* criada no passo anterior e adicionando-se essa *tag* aos *tags* selecionados, como no exemplo anterior. A partir disso, sempre que rodar a aplicação, o *display* criado atualizará o valor lido, caso tenha algum sensor presente na entrada.

Figura 72 - Adicionando a *Tag* ao *Display*



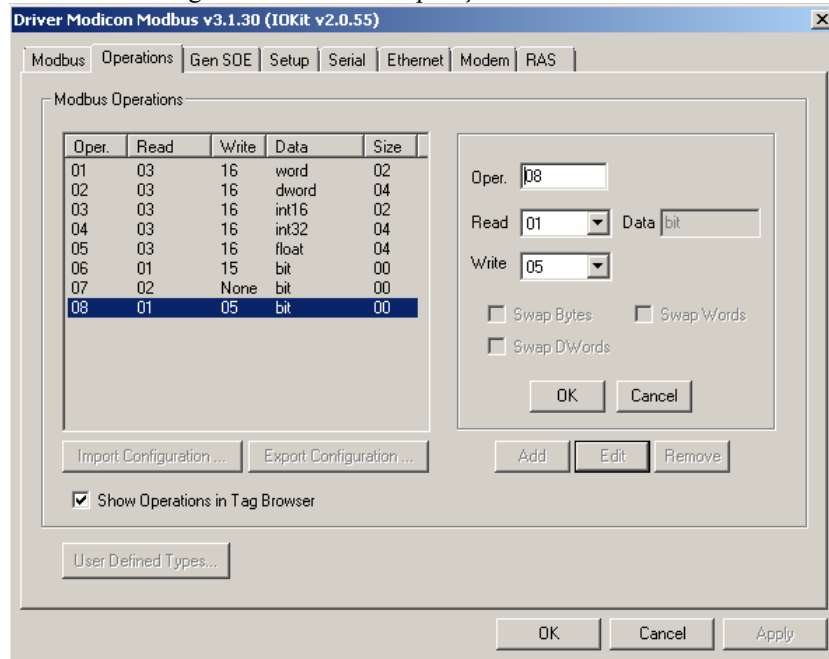
Fonte: Próprio Autor.

### 6.3.3 Terceiro exemplo: alterar estado de um relé do N1200

Nos exemplos anteriores foram tratadas apenas as leituras de dados, mas com o supervisório, também é possível escrever dados, ou seja, enviar comandos aos equipamentos para realizar tarefas. Nesse exemplo, será tratada a alteração do estado de um relé do controlador N1200, através de um botão simples, alocado na tela do supervisório.

A primeira etapa, consistiu na criação de uma nova operação Modbus, pois nos testes feitos quando estava sendo implementada a plataforma, nenhuma das operações padronizadas do *driver* Modbus funcionaram adequadamente. Para adicionar uma operação, basta abrir o Organizer, no menu lateral esquerdo expandir o menu de *drivers* e selecionar o *driver* ModMaster. Com o *driver* selecionado, basta clicar no botão “Extras”, no canto direito da janela. Dentro do menu “Extras”, abrir a aba “Operations”. Dentro da aba, clicar em “Add”, para acrescentar a nova operação que ficará com o número 08. A operação de leitura (*Read*) é de valor 1 e a operação de escrita (*Write*) é de valor 5. Preenchidos os valores, basta clicar em “OK” e a nova operação já está pronta pra ser utilizada, conforme ilustra a Figura 73.

Figura 73 - Criando a operação de estado do relé

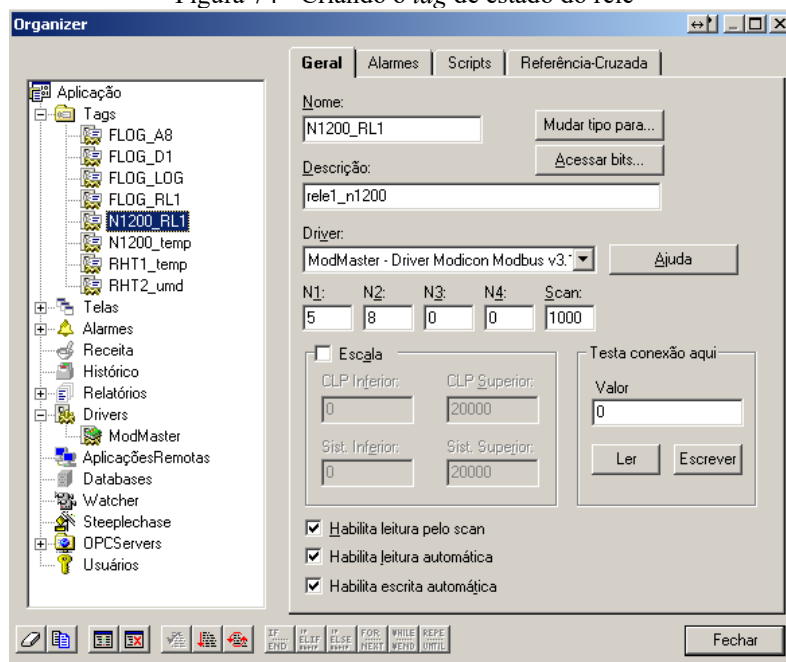


Fonte: Próprio Autor.

É possível também criar o *tag* relacionada ao relé do controlador N1200. Para isso, basta entrar novamente no Organizer e criar mais um *tag*, mas agora os valores dos parâmetros serão (Figura 74):

- a- N1 = 5, pois é endereço Modbus do controlador N1200;
- b- N2 = 8, pois é a função criada anteriormente;
- c- N3 = 0, sempre zero;
- d- N4 = 0, endereço da bobina responsável por manipular o estado do primeiro relé.

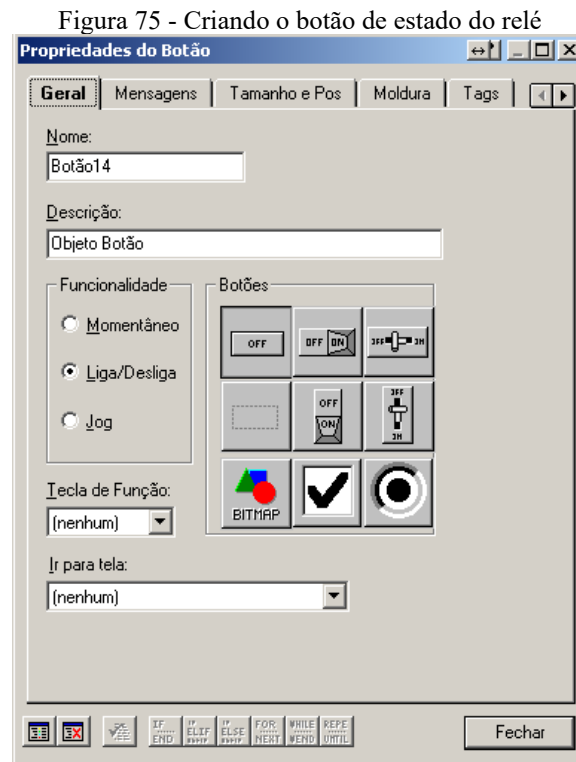
Figura 74 - Criando o *tag* de estado do relé



Fonte: Próprio Autor.

Sempre lembrando de marcar as três caixas para que seja habilitado o acesso ao *tag* pelo botão na página do supervisório.

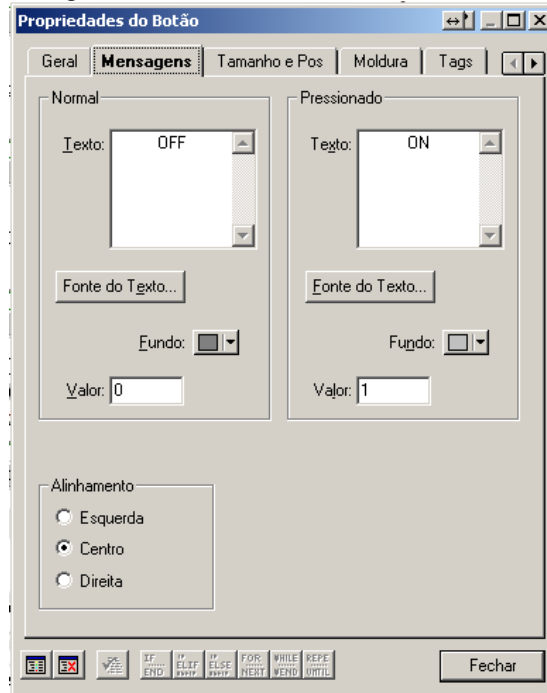
Por fim, a última etapa consistiu na criação do botão que fará as alterações de estado. No menu de criação, adicionar um botão e arrastá-lo para o tamanho desejado. Nas propriedades do botão, a primeira aba “Geral”, consiste na escolha do estilo do botão, para o caso do sistema de teste foi utilizada a primeira opção e a funcionalidade selecionada em modo “Liga/Desliga”, conforme ilustração da Figura 75.



Fonte: Próprio Autor.

Depois de selecionado o estilo, selecionar a aba “Mensagens” (Figura 76). O texto nesse caso deve fazer sentido, sempre combinando com o valor escolhido embaixo. Para o caso do botão de teste, foi escolhido o texto “OFF” para o valor 0 e “ON” para o valor 1, representando que o relé está aberto quando em valor zero e fechado em valor 1.

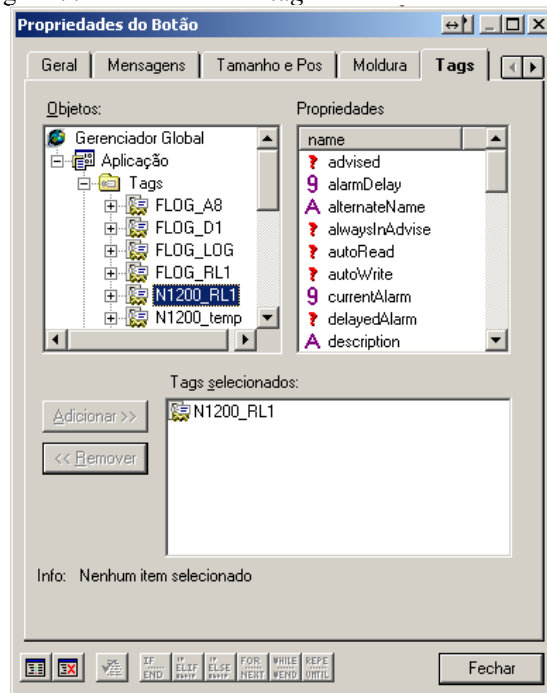
Figura 76 - Criando o botão de estado do relé



Fonte: Próprio Autor.

As duas próximas abas (Tamanho e Pos e Moldura) são apenas para escolhas decorativas, nesse caso pulou-se para a aba “Tags”. Dentro desta aba “Tags”, realizou-se o mesmo procedimento feito nos exemplos anteriores para adicionar a *tag*, com uma única diferença, selecionando a *tag* criada anteriormente com a função específica para trabalhar com o relé do controlador N1200, de acordo com a Figura 77.

Figura 77 - Adicionando o *tag* ao botão de estado do relé



Fonte: Próprio Autor.

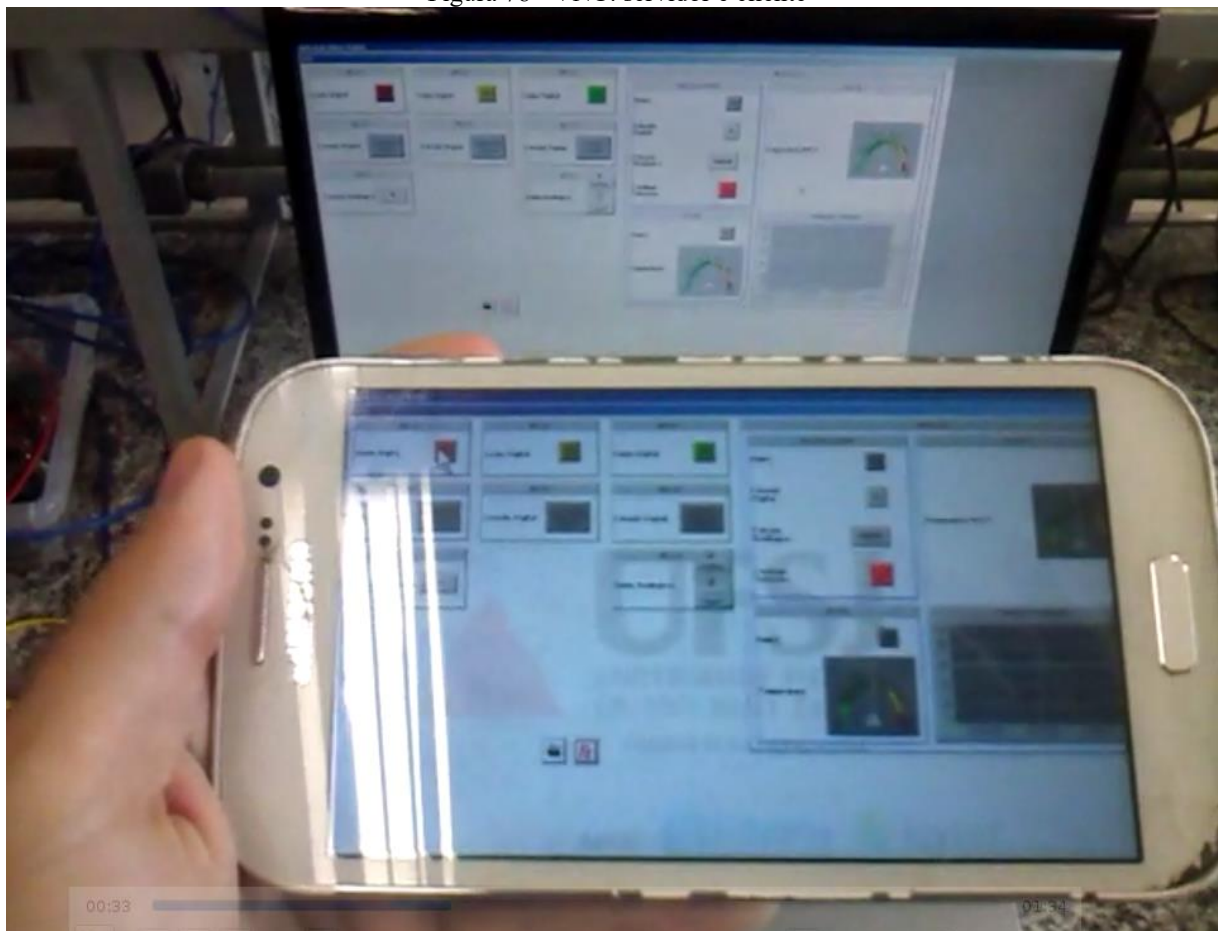
Finalmente, agora é possível dar comandos para abrir e fechar o contato dos relés através de um botão. Para criar um botão para um dos relés do FieldLogger o procedimento é similar, basta utilizar a função 3 e verificar o endereço do mesmo na tabela de registradores.

#### 6.4 ACESSO REMOTO

O acesso remoto é a possibilidade de enviar comandos e monitorar equipamentos à distância, permitindo mais segurança e maior conforto ao operador, além de maior velocidade na resolução de problemas, pois, independentemente da posição geográfica do operador, basta que ele tenha acesso remoto ao equipamento e, assim, poderá tomar medidas a fim de solucionar um problema qualquer. Na rede montada e testada com o sistema supervisorio, incluindo o módulo e os MICA's, existem duas possibilidades ao operador que devem ser consideradas e isso vai levar em conta qual *software* escolher:

Caso 1: se o operador estiver em uma área onde exista conexão direta com a rede do equipamento, seja ela via cabo ou sinal de rádio, por exemplo, e não seja possível o acesso à rede externamente, o melhor a se considerar é a utilização de uma conexão por protocolo VNC. Esse protocolo permite a realização de controle remoto baseado na arquitetura cliente-servidor e existem diversos *softwares* disponíveis e gratuitos que disponibilizam uma interface de conexão baseada no protocolo VNC. Nos testes realizados foi utilizado o “*TightVNC Server*” na máquina onde o sistema supervisorio está instalado, sendo essa considerada o servidor e nos clientes foi utilizado o mesmo programa, mas a sua versão cliente para poder realizar o acesso. Nos sistemas Android também existem programas que trazem uma interface para utilização do protocolo VNC como cliente. Nos testes foi instalado o “*VNC Viewer*” e, assim, foi possível controlar o servidor utilizando um smartphone, conforme ilustra a Figura 78.

Figura 78 - VNC: servidor e cliente

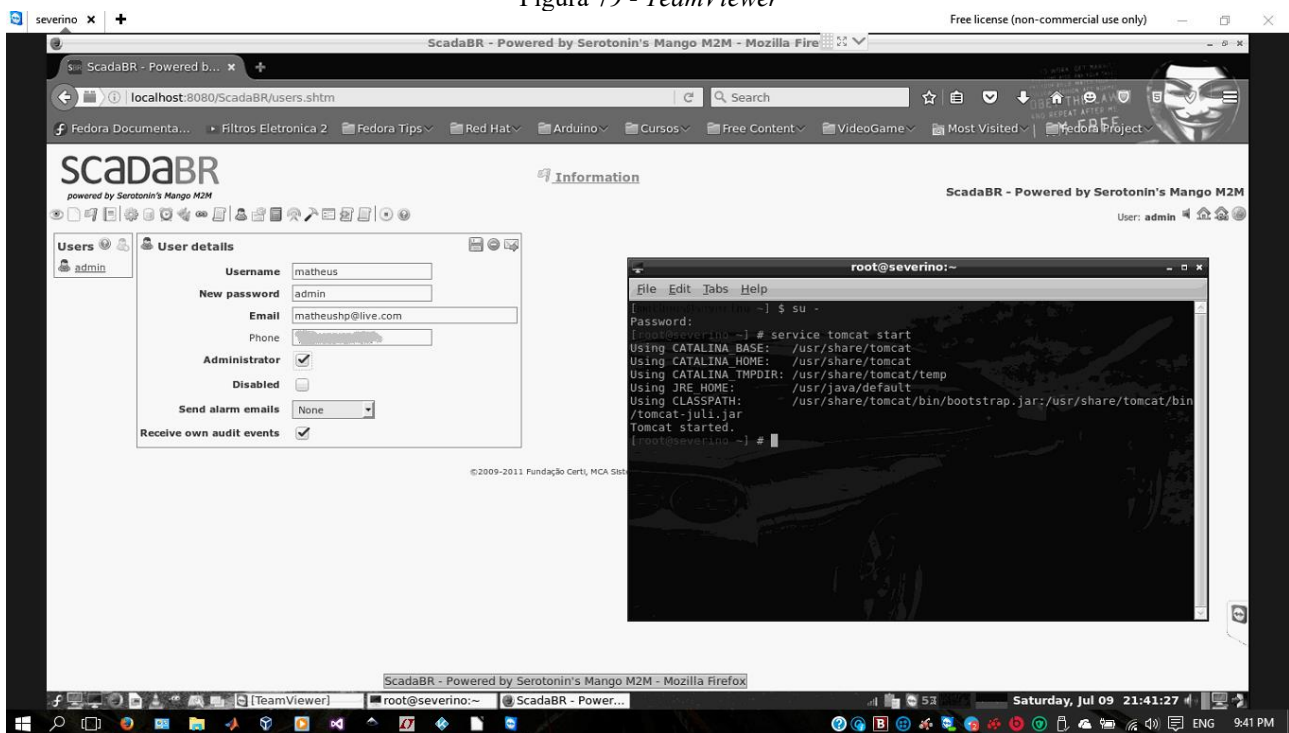


Fonte: Próprio Autor.

Caso 2: um segundo caso a se considerar é quando o operador necessita acessar o equipamento estando distante da rede ou impossibilitado de conectar-se diretamente a ela, utilizando o protocolo VNC. Considerando que a rede tenha acesso externo através de uma conexão de internet é possível fazer o controle, mas algumas coisas devem ser levadas em consideração. Em primeiro lugar, o provedor de acesso à internet deve disponibilizar um número de IP único externo, pois a rede deve ser acessível de qualquer lugar. Em alguns casos onde o provedor não fornece um IP externo, pode ser utilizado algum servidor externo de redirecionamento como os disponíveis no site NO-IP ([www.noip.com](http://www.noip.com)), onde faz-se o cadastro no site e instala o *software* que realizará o direcionamento, mas isso não é o foco nesse momento. Também é necessário verificar como está distribuída a estrutura da rede, sendo que para ter acesso externo direto em uma única máquina da rede (servidor com sistema supervisorio instalado) é preciso fazer um direcionamento de portas no roteador responsável por fornecer o acesso. Utilizando a rede do Campus Alto Paraopeba (CAP) da Universidade Federal de São João del-Rei (UFSJ), como exemplo, pode-se ter uma dimensão do problema, onde tem-se diversas camadas de rede, onde seria preciso ter acesso e fazer redirecionamento de portas até chegar no fornecimento principal de internet do Campus. Isso se torna inviável e arriscado, pois cada porta aberta em um roteador se torna uma brecha a mais de segurança que

pode ser explorada por pessoas mal-intencionadas. Para contornar esse problema, uma solução encontrada foi a utilização de um *software* específico de controle remoto, o “*TeamViewer*”. Esse *software*, da empresa com mesmo nome, pode ser encontrado em: [www.teamviewer.com](http://www.teamviewer.com). Ele resolve o problema dos direcionamentos de portas e a falta de IP único externo, pois, a conexão é feita diretamente com um servidor da empresa, antes de ser encaminhada para o cliente que está realizando o controle. Para ser realizado o controle remoto, basta que tanto o dispositivo controlador, quanto o controlado tenham acesso à internet. Além disso, é necessário ter o *software* instalado, uma conta de *e-mail* criada para acessar e, por fim, ter o cadastro do dispositivo que será controlado, conforme ilustra a Figura 79. O *TeamViewer* pode ser utilizado, gratuitamente, por pessoas físicas, com limite de dispositivos cadastrados para controle, e funciona em diferentes sistemas operacionais, permitindo que o controle seja efetuado também por um *smartphone*.

Figura 79 - *TeamViewer*



Fonte: Próprio Autor.

Na imagem da tela de “*prompt comando*” presente na Figura 79, pode-se observar que o acesso remoto acontece de um *notebook* pessoal na cidade de Ouro Branco, acessando um computador que está no laboratório do CAP-UFSJ. Sendo o *notebook* de onde parte a conexão com o *TeamViewer* instalado, e a tela em escala de cinza, o controle remoto. No caso do exemplo, foi utilizada escala de cinza para facilitar o destaque das máquinas envolvidas e também porque a conexão fica mais rápida quando está configurada dessa forma.

## 7 CONSIDERAÇÕES FINAIS

O desenvolvimento deste livro teve como propósito principal apresentar, de forma sistemática e aplicada, os fundamentos e a implementação de uma arquitetura de comunicação industrial, capaz de integrar diferentes dispositivos e tecnologias presentes em ambientes de automação. Ao longo dos capítulos foram discutidos conceitos fundamentais de redes industriais, protocolos de comunicação, integração entre dispositivos de campo e sistemas supervisórios, bem como os aspectos práticos envolvidos na construção de uma plataforma experimental para estudo e demonstração dessas tecnologias.

A proposta apresentada demonstra que a integração entre redes seriais baseadas em protocolos industriais tradicionais e infraestruturas de comunicação, baseadas em Ethernet, constitui um elemento essencial para a modernização dos sistemas de automação. A utilização de *gateways* de protocolo, mecanismos de encapsulamento de dados e *softwares* supervisórios permite ampliar significativamente a capacidade de monitoramento, controle e aquisição de dados em sistemas distribuídos. Nesse contexto, o módulo desenvolvido mostrou-se capaz de demonstrar de forma clara e didática os princípios de interoperabilidade entre diferentes camadas de comunicação, possibilitando a visualização prática do fluxo de informações entre dispositivos industriais.

Outro aspecto relevante deste trabalho refere-se ao seu potencial didático e acadêmico. A construção de um módulo experimental, baseado em equipamentos industriais reais, permite que estudantes e pesquisadores tenham contato direto com tecnologias amplamente utilizadas na indústria, aproximando o ambiente de ensino da realidade encontrada em sistemas produtivos. Dessa forma, o material apresentado contribui para a formação de profissionais mais preparados para atuar na área de automação industrial, especialmente em atividades relacionadas à integração de sistemas e redes industriais.

Além da contribuição educacional, a arquitetura proposta também pode servir como base para futuras aplicações em pesquisa e desenvolvimento tecnológico. A expansão da plataforma para ambientes conectados à Internet Industrial das Coisas (IIoT), a integração com sistemas de análise de dados e a implementação de mecanismos avançados de segurança cibernética representam caminhos promissores para a evolução do trabalho apresentado. A crescente digitalização dos processos industriais torna cada vez mais necessária a compreensão das infraestruturas de comunicação que sustentam esses sistemas, reforçando a importância de estudos e experimentações nessa área.

Por fim, espera-se que este livro contribua para o avanço do conhecimento e para a disseminação de boas práticas no estudo e implementação de redes de comunicação industrial. Ao combinar fundamentos teóricos com uma abordagem prática e experimental, a obra busca oferecer uma referência útil tanto para estudantes e professores, quanto para profissionais da engenharia que desejam compreender, projetar e implementar sistemas de comunicação em ambientes de automação industrial.

## REFERÊNCIAS BIBLIOGRÁFICAS

ACROMAG INC. **Introduction to Modbus TCP/IP**. Wixom: Acromag, 2004. Disponível em: [https://www.prosoft-technology.com/kb/assets/intro\\_modbustcp.pdf](https://www.prosoft-technology.com/kb/assets/intro_modbustcp.pdf)

ENCODER PRODUCTS COMPANY. **Industrial Ethernet communication protocols**. White Paper. Sandpoint: Encoder Products Company, 2019. Disponível em: [https://www.encoder.com/hubfs/white-papers/WP-2019\\_Industrial-Ethernet-Protocols/wp2019-industrial-ethernet-communication-protocols.pdf](https://www.encoder.com/hubfs/white-papers/WP-2019_Industrial-Ethernet-Protocols/wp2019-industrial-ethernet-communication-protocols.pdf)

GĂITAN, Nicolae C.; ZAGAN, Ionuț; GĂITAN, Vasile G. Evaluation of the performance of the Modbus RTU communication protocol for consecutive address transactions and improvements related to ModbusE. **Journal of Supercomputing**, [S.l.], 2026. Disponível em: <https://link.springer.com/article/10.1007/s11227-025-08207-5>

KOCAMUFTUOGLU, Alper. **A comparative study on industrial communication protocols using IoT platforms**. 2022. Disponível em: [https://www.researchgate.net/publication/359235554\\_A\\_Comparative\\_Study\\_on\\_Industrial\\_Communication\\_Protocols\\_Using\\_IoT\\_Platforms](https://www.researchgate.net/publication/359235554_A_Comparative_Study_on_Industrial_Communication_Protocols_Using_IoT_Platforms)

LE, L. H. et al. Integration of Modbus Ethernet communication for data acquisition and monitoring systems. **Research Square**, 2024. Disponível em: <https://assets-eu.researchsquare.com/files/rs-4708973/v1/0dff4538-6a91-41b9-a357-f112ee286c3b.pdf>

MODBUS ORGANIZATION. **Modbus application protocol specification V1.1b3**. 2012. Disponível em: [https://modbus.org/docs/Modbus\\_Application\\_Protocol\\_V1\\_1b3.pdf](https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf)

NIKOLAJEW, Christian; EICHELBERGER, Holger. Industry 4.0 connectors: a performance experiment with Modbus/TCP. **arXiv**, 2024. Disponível em: <https://arxiv.org/abs/2410.15813>

ZURAWSKI, Richard (ed.). **Industrial communication technology handbook**. 2. ed. Boca Raton: CRC Press, 2015.

REALIZAÇÃO:

**Aurum**  
EDITORA

CNPJ: 589029480001-12  
contato@aurumeditora.com  
(41) 98792-9544  
Curitiba - Paraná  
[www.aurumeditora.com](http://www.aurumeditora.com)