BIANCA LARISSA BARBOSA

PROPOSTA DE MODERNIZAÇÃO DE UM SISTEMA DE ESTEIRA TRANSPORTADORA DE CARGAS

UBERLÂNDIA

2021

BIANCA LARISSA BARBOSA

PROPOSTA DE MODERNIZAÇÃO DE UM SISTEMA DE ESTEIRA TRANSPORTADORA DE CARGAS

Trabalho de Conclusão de Curso da Engenharia de Controle e Automação da Universidade Federal de Uberlândia - UFU - Câmpus Santa Mônica, como requisito para a obtenção do título de Graduação em Engenharia de Controle e Automação.

Universidade Federal de Uberlândia – UFU
Faculdade de Engenharia Elétrica

Orientador: Prof. Dr. Renato Ferreira Fernandes Júnior

UBERLÂNDIA
2021

Barbosa, Bianca Larissa
Proposta de modernização de um sistema de esteira transportadora de cargas/ **Bianca Larissa Barbosa**. – **UBERLÂNDIA**, **2021**- 90p.: il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Renato Ferreira Fernandes

Trabalho de Conclusão de Curso – Universidade Federal de Uberlândia – UFU Faculdade de Engenharia Elétrica. **2021**.
Inclui bibliografia.

1. Modbus. 2. Esteira Transportadora. 3. Automação Industrial.4. Modernização.5.Inversor de Frequência.

I. Renato Ferreira Fernandes. II. Universidade Federal de Uberlândia. III. Faculdade de Engenharia Elétrica. IV. Engenharia de Controle e Automação.

Dedicatória

Aos meus pais, Cláudio e Lucilene, e à minha família por todo esforço, suporte e dedicação durante toda a minha graduação.

Às amizades pelo apoio e companhia nesse tempo.

E à todos os que me ajudaram até aqui, em especial a Daniel pelo incentivo incondicional durante essa jornada.

Agradecimentos

Expresso minha gratidão a todos os profissionais do curso de Engenharia de Controle e Automação da Universidade Federal de Uberlândia, por todo o suporte que me deram, não só ao longo da realização do meu trabalho mas também de toda graduação.

Agradeço ao meu orientador, o Professor Renato Ferreira Fernandes Júnior por ter aceitado acompanhar-me neste projeto, por todas as oportunidades dadas e pela disponibilidade em me ajudar com meu desenvolvimento durante todo esse período.

“Por vezes sentimos que aquilo que fazemos não é senão uma gota no oceano.
Mas sem ela o oceano seria menor.”,
(Madre Teresa de Calcutá)

Resumo

Visto o contexto recente de constantes inovações tecnológicas, a implementação de adaptações e melhorias se mostram como uma estratégia para fazer com que sistemas obsoletos possam ser absorvidos pelas exigências contemporâneas. Dessa maneira, o objetivo deste trabalho é desenvolver uma abordagem para disponibilizar uma planta didática de um sistema de esteira transportadora de carga de acordo com os equipamentos de uso recorrente no laboratório de Redes Industriais II do curso de Engenharia de Controle e Automação da Universidade Federal de Uberlândia. Na nova disposição, haveria um outro CLP como principal que, por meio do protocolo Modbus, seria o responsável por controlar o sistema, comunicar com os elementos de partida, e obter valores das entradas e saídas lidos pelo CLP original. A estratégia desenvolvida se mostrou funcional, sendo possível colocar a esteira de carga em movimento e acessar os valores dos IO's. Por meio da captura de linha foi possível uma análise mais aprofundada da rede. As análises das mensagens trocadas indicam que em um ciclo de *scan* há excesso de *requests* para um mesmo escravo e alguns momentos de inatividade.

Palavras-chave: Modbus. Esteira Transportadora. Automação Industrial. Modernização.

Abstract

Due to the recent context of constant technological innovations, adjustments and improvements reveals as a strategy to make obsolete systems accepted by the present-day demands. Thus, the goal of this project is to develop an approach to make available to the use a didactic plant for a load conveyor system according to the equipment of recurrent use in the Industrial Networking II laboratory of the Control and Automation Engineering program at the Federal University of Uberlândia. By studying the original system's composition, the purpose is to bring forward methods to modernize it, by replacing the controller. And so, based on the chosen solution, will be described the procedures performed to build up a communication network for the control of the conveyor's motor, using a frequency inverter over the Modbus protocol, in addition to the control of the load cell by local I/O. In order to reach it, based on the study of the original structure, a scenario was defined to modernize the system. In this new configuration, would have another PLC as the main controller of system, which using the Modbus protocol, would be responsible for controlling the system, for communicating with the electronic power devices and get the inputs and outputs values from the original PLC. In the tests, the developed approach was successful to put the system in operation. Using a sniffer software was possible to take a closer look at the communication network. The analysis of the exchanged packages shows that in a scan cycle was sent needless requests for one same slave and was noticed some moments of inactivity.

Keywords: Modbus. Conveyor. Industrial Automation. Modernization.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

APDU	*Application layer protocol data unit*
ASCII	*American Standard Code for Information Interchange*
CA	*Corrente Alternada*
CC	*Corrente Contínua*
CLP	*Controlador Lógico Programável*
CPU	*Central Process Unit (Unidade Central de Processamento)*
CRC	*Cyclical Redundancy Checking*
CV	*Cavalo-Vapor - Unidade de medida de potência de máquinas ou motores*
DCS	*Distributed Control Systems*
FDB	*Function Block Diagram*
Hz	*Hertz - Unidade de medida de frequência*
I/O	*Input/ Ouput*
ID	*Identificador*
IGBT	*Insulated Gate Bipolar Transistor*
IHM	*Interface Homem-Máquina*
In	*Corrente nominal*
IP	*Internet Protocol*
Ip	*Corrente de partida*
kg	*Quilograma - Unidade de medida de massa*
kW	*Quilowatts - Unidade de medida de potência*
LD	*Ladder Diagram*
LRC	*Longitudinal Redundancy Check*
m	*Metros - Unidade de medida de distância*
mA	*Miliampere - Unidade de medida de corrente*
MPa	*Mega Pascal - Unidade de medida de pressão*
ms	*Milisegundos - Unidade de medida de tempo*
mV	*Milivolt - Unidade de medida de tensão*
PCMCIA	*Personal Computer Memory Card International Association*
PLC	*Programmable Logic Controller*
Psi	*Libra-força por polegada quadrada - Unidade de medida de pressão*

RPM	*Rotações por minutos - Unidade de medida de velocidade angular*
RS-232	*Recommended Standard 232*
RS-485	*Recommended Standard 485*
SCADA	*Supervisory control and data acquisition*
SFC	*Sequential Function Chart*
Vcc	*Tensão em corrente contínua - Unidade de medida de potencial elétrico*
Vdc	*Tensão em corrente contínua - Unidade de medida de potencial elétrico*

Sumário

1 INTRODUÇÃO

No fim do século XX, os propulsores da nova revolução do desenvolvimento foram – e ainda continuam sendo, a tecnologia e globalização. O aperfeiçoamento da informática, dos transportes e das comunicações foram os principais alavancadores de mudanças (ROSÁRIO, 2009). E dado a velocidade que esse processo de evolução vem ocorrendo, é considerável os impactos nos últimos anos, principalmente no que tange aos parques industriais.

As indústrias passaram por grandes transformações para atender as novas demandas de qualidade, flexibilidade e produtividade. A forma de responder a esses padrões foi por meio da implementação de máquinas automatizadas, já que esses equipamentos envolvem um conjunto de técnicas e procedimentos de automação que possibilitavam o aumento da qualidade, redução de custos, processos mais seguros e mais eficientes, menor tempo de produção e preços mais acessíveis, permitindo assim a adaptação às exigências e competividade dos produtos no mercado (ROSÁRIO, 2009).

Com isso, o mercado passou a ressaltar investimentos em tecnologia, já que privilegiam a inovação como vantagem competitiva (ROSÁRIO, 2009). Isso fez com que as inovações tecnológicas de processos constantemente envolvessem a melhoria ou a criação de novas técnicas e o desenvolvimento de novos equipamentos (SILVA, S. 2018).

É o que Muniz (2000) caracteriza como difusão tecnológica, ou seja, é um processo incremental e contínuo de mudança tecnológica, que promove a adaptação da inovação original a um grande número de situações e o aperfeiçoamento contínuo das suas características e desempenho.

Nesse contexto, a necessidade de atualização tecnológica é cada vez mais presente e nunca foi tão constante como nos últimos tempos (ROSÁRIO, 2009).

Acompanhando essa tendência, as redes de comunicação industrial ou simplesmente redes industriais, vêm passando pela expansão de protocolos e aprimoramento de equipamentos para possibilitar implementações de aplicações mais complexas e eficientes (SANTOS et al., 2018).

Com base nas relevâncias citadas anteriormente, e num estudo aprofundado da questão, espera-se ter informações o suficiente para poder concluir se a adaptação da rede de comunicação de um sistema de esteira transportadora é viável. E se verificado que a solução alternativa é praticável, visa-se a executar o projeto de acordo com os métodos sugeridos e assim disponibilizar um sistema instrumentado de acordo com os equipamentos acessíveis ao

laboratório de Redes Industriais II da Universidade Federal de Uberlândia.

1.1 Justificativa

De acordo com Rosário (2009), o ciclo de vida dos produtos foi drasticamente reduzido e o ritmo de renovação é cada vez mais forte. A capacidade de identificar novas necessidades e lhes dar resposta é uma habilidade relevante, sendo a automação um dos caminhos para modificar e modernizar métodos de produção. Ainda, ele salienta que é indispensável uma lógica de inovação estratégica que permita ser capaz de conceber ou adaptar produtos de forma continuada.

Tendo isso em vista, o tema foi escolhido devido a importância da atualização tecnológica no contexto atual e dado que o laboratório de Redes industriais possui esse sistema de esteira de carga, mas seu uso, atualmente, está comprometido por não estar adaptado aos equipamentos habitualmente utilizados nos estudos da disciplina.

A perspectiva é aplicar os conhecimentos adquiridos através do curso na disciplina de Redes Industriais II sobre protocolos e toda parte de equipamentos como o CLP, sensores, atuadores, e também sobre sistemas supervisórios e assim disponibilizar uma planta de acordo com os equipamentos do laboratório de Redes Industriais para estudos futuros.

1.2 Objetivos

O objetivo principal deste trabalho é apresentar uma proposta de solução de modernização da automação de uma planta didática de uma esteira transportadora.

Para isso, como objetivos específicos, será estudado automação atual da planta e propor sua modernização através do controle do motor via inversor de frequência, utilizando o protocolo Modbus. Também será proposto o controle da célula de carga através de I/O local.

Essas análises incluem as seguintes etapas:

- Estudo do funcionamento e da lógica de controle da esteira de carga.

- Estudo das possibilidades de adaptação da instrumentação do sistema acordo com os equipamentos disponíveis.
- Desenvolvimento de um projeto de automação para a esteira.
- Montagem elétrica do sistema.
- Montagem da automação do sistema.
- Desenvolvimento de um sistema de supervisão adaptado.
- Modelagem do sistema de controle.
- Implementação e validação do sistema final.

Esse projeto está dividido em 5 capítulos. O atual capítulo aborda as considerações iniciais a respeito do tema deste trabalho. O capítulo dois aborda a revisão da literatura para aprofundar nos conhecimentos necessários para fazer o que fora proposto no primeiro capítulo. No capítulo três, é apresentada a metodologia formulada para o estudo do sistema e para implementação da solução. No capítulo quatro é descrito os resultados obtidos durante a execução do projeto. E por fim, o capítulo 5 discute as conclusões finais sobre o trabalho.

2 REFERENCIAL TEÓRICO

Neste capítulo serão descritos os conceitos básicos sobre os elementos principais envolvidos na automação de sistemas baseados em esteiras transportadoras. Estes elementos de automação inclui os controladores, os sensores e atuadores e também os protocolos de comunicação entre estes equipamentos.

2.1 Esteiras Transportadoras

As primeiras esteiras transportadoras datam meados do século XVII, quando eram utilizadas principalmente para utilizado para o transporte popular de materiais a granel, como por exemplo, sacos de grãos. Esse mecanismo era composto por correias, feitas de lona, couro ou borracha, que se deslocavam sobre a cama de madeira lisa (REDUTORES IBR, 2016).

A partir da Segunda Revolução Industrial, as esteiras tiveram grande influência no aumento da escala de produção e no processo fabril. Foi nesse período, que as esteiras passaram a ter seu uso relacionado com teorias que buscam diminuir o tempo e o custo da produção e consequentemente aumentarem os lucros da empresa (REDUTORES IBR, 2016).

Nesse contexto, os processos industriais tornaram-se automáticos e cada vez mais dependentes dinamismo na execução de tarefas e operações, veio o desenvolvimento de esteiras automatizadas para garantir a agilidade em diversas etapas dos processos industriais que envolvem movimentação e transporte de cargas.

Entre as vantagens de adquirir a esteira transportadora automatizada estão (LOGITEC SISTEMAS):

- Baixo custo com manutenção;
- Produto resistente e durável, estrutura forte para suportar cargas dependendo da necessidade;
- Disponível em vários tipos, atendendo diversas aplicações;
- Versatilidade, pois pode ser utilizada em vários segmentos industriais;

- Segurança, pois dificilmente há falhas de funcionamento;

- Agilidade no processo industrial com a presença da esteira transportadora automatizada;

Como constato por Magalhães (2010), os sistemas de movimentação, deslocamento e manuseio, onde são encontradas as esteiras transportadoras, são um dos principais sistemas motrizes que compõem o setor industrial brasileiro, com presença expressiva em plantas de alimentos e bebidas, minerais não-metálicos e metalurgia. As principais aplicações nesses setores são:

- Transporte de produtos em linhas de montagem e de manipulação para curtas distâncias, como montagens de eletrodomésticos e processamento manual de alimentos.

- Movimentação de materiais granulados, como areia e carvão em volumosas quantidades por extensas distâncias.

Nesse contexto, as esteiras vêm sendo utilizadas principalmente no setor industrial com a finalidade de grandes reduções de custos logísticos e aumento de agilidade nos transportes.

2.2 Automação nas indústrias

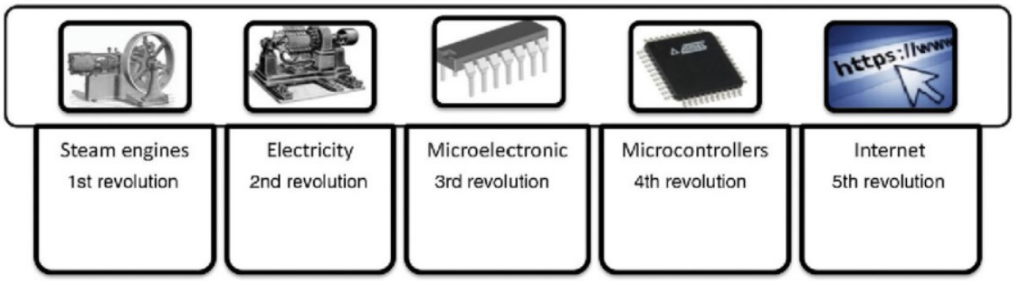
De acordo com, a palavra ‘Automação’ vem das palavras gregas ‘Auto’, que significa ‘que funciona por si mesmo’ e ‘Matos’, que significa movimento. Ou seja, ‘Automação’ quer dizer o mecanismo para sistemas funcionarem por si próprios’ (SEN, 2017).

Desde a Pré-História, o homem se preocupa em tentar fazer utensílios e ferramentas para reduzir o esforço em suas atividades operárias. No entanto, a automação industrial só ganhou espaço em meados do século XVIII, durante a Primeira Revolução Industrial, com processos semiautomáticos, como a máquina de fiar e a invenção de um mecanismo de regulagem de fluxo do vapor em máquinas. A partir daí, o foco deixou de ser a processo de produção artesanal e o mercado voltou seus esforços para o desenvolvimento industrial (SEIDL et al., 2014).

E desde aquele tempo, com maior intensidade a partir do século passado, com o uso da energia elétrica, que estimulou indústrias como a de aço, de química e de produção de ferramentas, de motores a combustão, e um intenso desenvolvimento dos transportes (SEIDL et al., 2014), a automação vem se consolidando como o conjunto de tecnologias para resultar na operação de máquinas e sistemas com o mínimo de intervenção humana e atingir níveis de performances superior a operação manual (DEY; SEN, 2020).

No século XX, com a difusão dos circuitos lógicos, passou-se a contar com computadores e controladores programáveis, sendo os primeiros pilares de todos sistemas de automação contemporâneos (SEIDL et al., 2014). Nesse cenário, e visto a tendência de globalização, as indústrias passaram por grandes transformações para atender as novas demandas a fim de manter a relevância de seus produtos.

Figura 1 – Evolução da automação do ponto de vista tecnológico.



Fonte: Adaptado de (MEHTA; REDDY, 2014)

Para as novas exigências do mercado, de acordo com Seidl et al. (2014), dentre os vários objetivos da automação das indústrias pode-se destacar alguns, como:

- **Qualidade:** necessidade de manter a qualidade dos produtos de forma que as deficiências sejam compensadas e a sofisticação do que for produzido seja mantido;

- **Flexibilidade:** inovações e mudanças são constantes nas linhas de produção atualmente, sendo assim é necessária flexibilidade para atender as especificações de clientes e variar a escala de produção de forma a evitar desperdício de material e custo operacional e assim conseguir manter a

relevância e a competitividade no mercado;

- **Produtividade:** é necessário o controle do equipamento de produção para garantir que os produtos estejam o mais próximo possível de um padrão aceitável, de forma a minimizar o refugo e controlar a produção;

- **Viabilidade Técnica:** foco no processamento imediato do grande volume de informações gerado pelas plantas atuais e automatizar de forma segura e organizada maior número de processos, visando reduzir custos e limitando o risco de falha humana.

E então, tem-se o contexto mais recente, com processos inteiramente automáticos, com reduzida necessidade de intervenção humana para pleno funcionamento e com grande quantidade de dispositivos e máquinas em funcionamento com pequenos tempos de parada, sendo monitorados constantemente por relativamente poucas pessoas.

2.3 Sistemas de automação atuais

Como dito, hoje em dia, a automação está associada a atividades atreladas aos computadores que substituem o trabalho humano em favor da segurança, da qualidade e da redução de custos, aperfeiçoando os complexos objetivos dos serviços e das indústrias.

Segundo Seidl et al. (2014), entre as principais características dos sistemas de automação estão:

- **Acionamento:** fornece ao sistema a energia para atingir determinado objetivo, por exemplo: motores, pistões hidráulicos, etc.;

- **Sensoriamento:** mede o desempenho do sistema de automação ou uma propriedade de alguns de seus componentes;

- **Controle:** utiliza as informações dos sensores para regular, controlar os dispositivos, por exemplo: controlar motores, válvulas, etc.;

- **Comparador:** elemento que permite comparar valores medidos com valores preestabelecidos e serve para tomada de decisão e de quando e como atuar, por

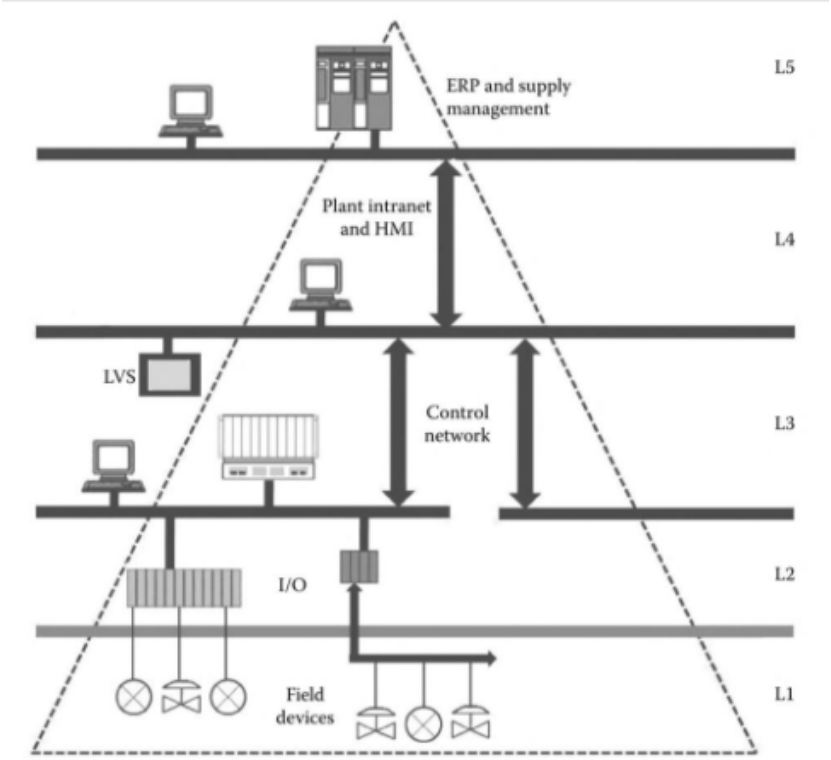
exemplo: termostato e sistemas de software;

- **Programas:** contêm as informações de processo e permitem controlar as interações entre os diversos componentes.

Essas características se referem a diferentes processos e máquinas, responsáveis por cada etapa de produção e em diferentes níveis de automação, mas que devem se comunicar ao longo de todo o processo de produção a fim de dar forma ao produto final.

Para um fluxo ordenado de informações e com objetivo de otimizar a performance do processo, a cadeia de informações em um sistema de automação industrial pode ser organizada em níveis de hierarquia, como pode ser visto na Figura 2.

Figura 2 – Níveis hierárquicos de processos industriais.



Fonte: Adaptado de (SEN, 2017)

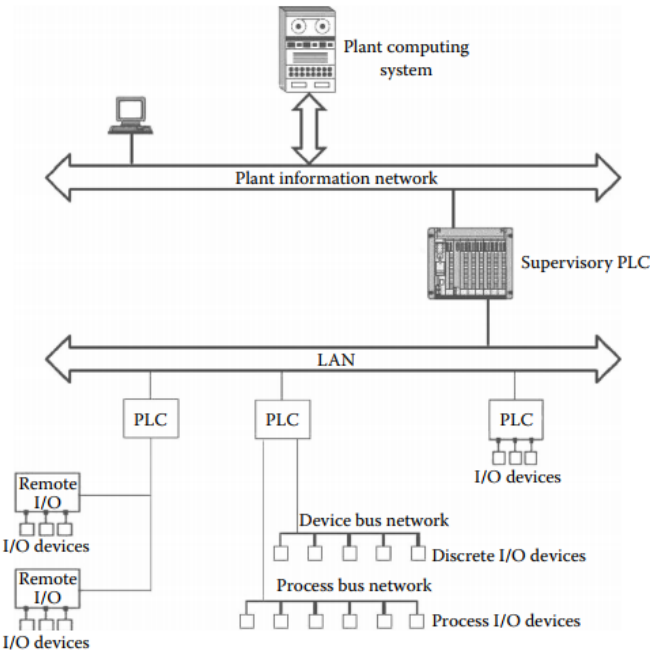
No nível mais inferior (L1), ou no chamado nível de campo, se encontra aos sensores, posicionadores e atuadores. Eles são os responsáveis por indicar as condições do processo, como temperatura, pressão, nível e fluxo (SEN, 2017).

No segundo nível (L2), o nível de I/O, ou entradas e saídas, comanda-se as entradas e saídas. A saída do sensor, por meio do nível de I/O é enviada para o controlador que então envia o devido sinal de volta para o atuador para o controle do processo (SEN, 2017).

No nível 3 (L3), são usados PLCs (*Programmable Logic Controllers*) ou DCSs (*Distributed Control Systems*) geram os sinais de controle. Esses sinais são enviados para os sensores e atuadores onde os sinais são recebidos e a ação de controle é realizada, que pode ser abrir ou fechar uma válvula, iniciar ou parar um motor, etc. Os PLCs, ou CLPs em português, são os elementos de controle que serão abordados em mais detalhes na seção 2.5. Os DCSs é como são chamados os sistemas remotos, onde a função de controle do sistema é compartilhada por mais de um CLP, formando assim uma rede de controle de I/Os distribuída pela planta.

É possível classificar um barramento de I/O's como rede de equipamentos, que comunicam dados e informações de status, ou como rede de processos, que enviam informações sobre as condições de máquinas ou processos. Na Figura 3 é mostrado o exemplo de uma estrutura que possui o controle descentralizado, onde há uma rede de equipamentos e processos conectadas a um CLP e I/O's remotos conectados a outros CLPs, além de um CLP que comunica o controle distribuído com as camadas superiores (SEN, 2017).

Figura 3 – Exemplo de um sistema de controle descentralizado.



Fonte: Adaptado de (SEN, 2017)

Dando sequência a hierarquia da Figura 2, no nível 4 (L4), é a área de interação homem-máquina, onde deve ser certificado que qualquer variável do processo em toda planta deve estar disponível para o operador, seja por meio de dispositivos que permitem interação direta entre usuário e o controlador (IHM - interfaces homem-máquinas) ou por meio de softwares usados para monitorar e controlar a planta (sistemas supervisórios), para que assim essas ferramentas possam auxiliar o colaborador e alertar quanto níveis estarem fora dos limites definidos, mudar configurações de equipamentos e outras configurações (SEN, 2017).

No primeiro nível (L5), o nível de gerenciamento corporativo, os dados são mandados para o ambiente administrativo e de tecnologia da informação para avaliação dos elementos envolvidos no processo como armazenamento de dados, análise de qualidade e quantidade dos produtos, controle de estoque, gestão de insumos, entre outros (SEN, 2017).

Os diferentes níveis têm que lidar com diferentes requisitos para este determinado nível. Por exemplo, no nível mais acima, há um grande volume de dados, mas não possuem grande exigências quanto a tempo e frequência. Já os três últimos níveis exigem características diferentes dado a constante troca de dados, pequeno tamanho do pacote de dados e tempo de resposta e comunicação determinística (SEN, 2017), isto é, informação em tempo real, mas sempre voltada para o processo interno, nunca ou quase nunca com informações ao ambiente externo da produção (SANTOS et al., 2018).

Dada as características necessárias para esses níveis, sendo os três últimos onde está essencialmente a automação industrial e o nível 4, o de supervisão, podendo ser considerado tanto de automação industrial quanto de tecnologia da informação industrial, para que todas as máquinas possam estar conectadas e se reconhecerem, é necessário o uso de redes de comunicação industrial, ou simplesmente redes industriais, com protocolos de comunicação específicos para esses ambientes (DEY; SEN, 2020).

A comunicação industrial é responsável por transportar a informação da planta para a sala de controle e assim controlar o processo de forma mais eficiente e segura.

2.4 Protocolos industriais

Como define SEIDL et al. (2014), "*Fieldbus* é o termo genérico empregado para

descrever tecnologias de comunicação industrial para controle em tempo real".

A transmissão de sinais em plantas de processo iniciou-se com a transmissão pneumática, de 3 a 15 Psi e foi sucedido pela transmissão de sinal por corrente, de 4 a 20 mA. Essa transmissão de sinais analógicos, eram direcionadas para o controle do nível mais inferior da pirâmide da automação. Os protocolos *fieldbuses,* como Foundation Fieldbus, Profibus, DeviceNet, ControlNet, Modbus, AS-i e CAN bus, são sucessores dessas tecnologias e têm como característica a comunicação que permite a convivência de um mecanismo de comunicação digital simultaneamente com um mecanismo analógico, podendo assim, fazer a leitura e escrita de dados nos dispositivos de campo (DEY; SEN, 2020).

Ao longo dos anos, as diversas modificações e mudanças de requisitos para a produção industrial fizeram com que os sistemas de automação se tornassem mais complexos. Isso provocou o desenvolvimento nos protocolos disponíveis no mercado e da diversidade de implementação possíveis (SEIDL et al., 2014). Segundo Zurawski (2017), embora os protocolos *fieldbuses* ainda sejam bastante presentes na automação, a plataforma de comunicação Ethernet vem ganhando espaço devido ao baixo custo, à flexibilidade, à escalabilidade e à performance. Protocolos baseados nessa tecnologia são: Ethernet IP, Foundation Fieldbus HSE, Profinet, Modbus TCP.

Como abordado, muitas tecnologias industriais foram implementadas desde a Terceira Revolução Industrial. Mesmo que hajam outras mais recentes, parte dessas tecnologias continuam a ser utilizadas nas empresas e, do ponto de vista produtivo, ainda apodem ser consideradas atuais e relevantes (SANTOS et al., 2018), que é o caso da tecnologia base para o desenvolvimento deste trabalho, o protocolo Modbus.

2.4.1 Modbus

Como afirma SEIDL et al. (2014), Modbus é o principal protocolo de comunicação de dados do mercado. Criado pela empresa Modicon em 1979, é usado principalmente para estabelecer a comunicação entre dispositivos do tipo mestre-escravo, onde o mestre, é responsável por iniciar as transmissões, e os outros, os escravos, respondem ao pedido do mestre, além de processar e enviar a informação para ele.

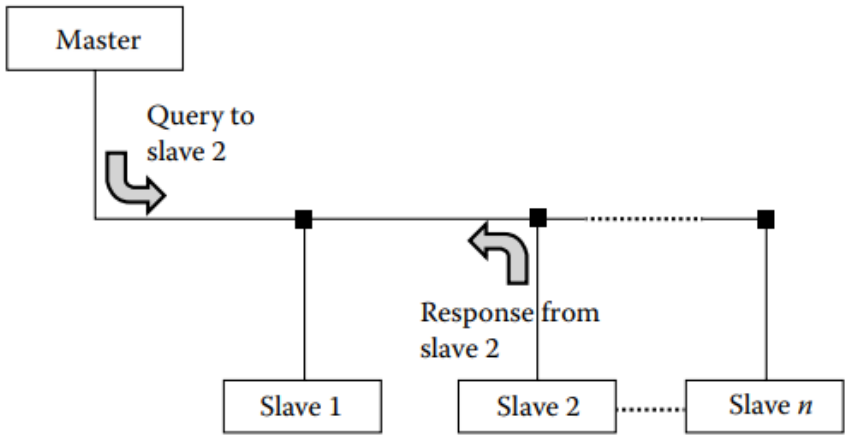
Embora inicialmente fosse um protocolo proprietário controlado pela Modicon, desde

2004 é administrado por uma comunidade de usuários de automação conhecida como Modbus-IDA, uma organização sem fins lucrativos que gerencia a o protocolo e procura mantê-lo relevante, continuando a distribuição aberta das especificações do protocolo e fornecendo infraestrutura para a certificação de compatibilidade de equipamentos (WILAMOWSKI; IRWIN, 2016).

Grande parte pela estrutura aberta e flexível, e de acordo com SEIDL et al. (2014), o Modbus é largamente utilizado na indústria, principalmente em PLCs, sendo suportado pela maioria dos softwares HMI e SCADA.

Modbus pode ser definido como uma aplicação da camada de mensagem, de acordo com o modelo OSI sendo a camada 7, para a comunicação do tipo mestre/escravo entre diferentes equipamentos conectados em vários tipos de redes. A Figura 4 mostra um esquema da troca de mensagens entre um mestre e um escravo.

Figura 4 – Representação da comunicação entre os elementos de uma rede Modbus.



Fonte: Adaptado de (SEN, 2017)

A troca de informações entre um mestre Modbus e um escravo Modbus é iniciado quando o mestre envia um pedido, que também pode ser chamado de *query* para o escravo para a troca de informações ou executar um comando. Depois que o escravo recebe o pedido, ele executa o comando ou prepara o dado requisitado. O escravo então envia uma mensagem de resposta (*response*) para o mestre reconhecendo que houve sucesso na execução do comando requisitado.

Um dispositivo pode ser um mestre, escravo ou ambos, dependendo das especificações do equipamento e da implementação. As configurações mais comuns são:

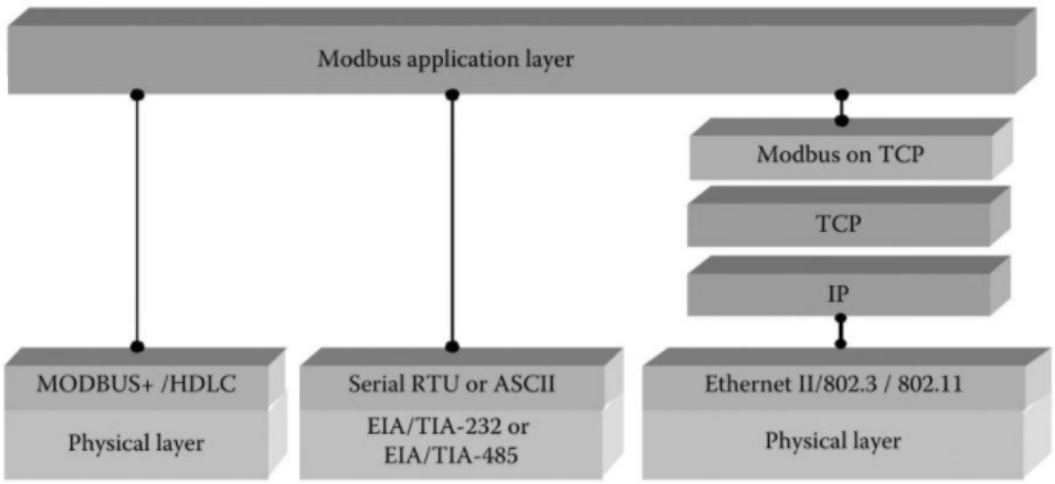
- Aplicações para HMI ou SCADA são implementadas como mestre para iniciar a comunicação com o PLC e outros dispositivos para coleta de informações;

- Um sensor, ou dispositivos de I/O em geral, são implementados como escravo para que assim os outros elementos possam ler e escrever os valores lidos nesses dispositivos.

- O PLC pode ser implementado tanto como mestre como escravo para que possa iniciar a comunicação com outros PLCs e dispositivos de I/O, e também responder a pedidos de outros PLCs, SCADA, HMIs e outros equipamentos.

Quanto à arquitetura, o Modbus é organizado como um protocolo de duas camadas, como pode ser visto na Figura 5.

Figura 5 – Camadas do protocolo Modbus.



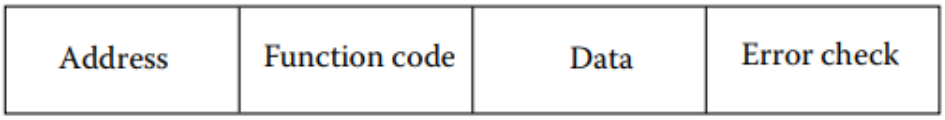
Fonte: Adaptado de (ZURAWSKI, 2017)

Na camada superior, chamada de camada de aplicação, é onde estão definidas as funções ou serviços que o mestre Modbus pode solicitar de um escravo Modbus. Esses pedidos são codificados como mensagens pela estrutura, ou *frame,* chamada APDU (*Application Protocol Data Unit*), que é a unidade de dados da aplicação do protocolo (WILAMOWSKI;

IRWIN, 2016).

Esse frame possui variações quanto ao modo de transmissão, mas basicamente, há três campos para transportar a mensagem, como mostrado na Figura 6.

Figura 6 – Formato do frame APDU Modbus.



Fonte: Adaptado de (SEN, 2017)

O campo *Address* é onde os dispositivos escravos são endereçados usando um número identificador (ID), no intervalo de 1 a 247 – embora essa não seja a disponibilidade real devido às limitações práticas. Pode ter dois significados (ZURAWSKI, 2017):

- Representar o endereço: esse ID precisa ser único entre todos os escravos acessíveis ao mestre;
- Representar parte do endereço indicando para um escravo: as outras partes estão representas no mecanismo de transporte.

Além desses, o valor do campo *Address* pode ter outros significados como descritos na Figura 7.

Figura 7 – Possíveis valores para o campo Address em um frame Modbus.

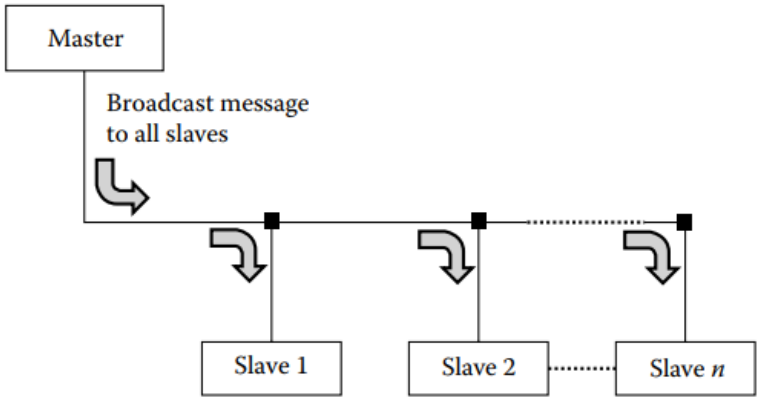
Broadcast	Server Addressing	Reserved	Address Fully Determined by Next Layer
0	From 1 to 247	From 248 to 254	255

Fonte: Adaptado de (ZURAWSKI, 2017)

Quando o *Address* é definido com o broadcast, o *request* é enviado para todos os escravos, mas nesse caso, não é enviada a resposta de volta. A representação desse modo de comunicação pode ser vista na Figura 8. Quando o *Address* possui o valor de 255 significa que

o escravo há está endereçado pela camada de transporte (ZURAWSKI, 2017).

Figura 8 – Modo de comunicação broadcast



Fonte: Adaptado de (SEN, 2017)

O próximo campo do *frame* Modbus, visto na Figura 6, é o campo do código de função, que onde fica o código do serviço pedido ao escravo em um *request*, e no caso da resposta, o indicador de sucesso ou falha. Esses códigos são valores que estão no intervalo de 0x01 a 0x7F. O intervalo entre 0x80 a 0xFF é reservado para o mecanismo de tratamento de exceções, que é o mecanismo que cuida do tratamento da ocorrência das condições que alteram o fluxo normal da aplicação. O êxito no processamento de um código é indicado pela reprodução desse código no mesmo campo na mensagem de resposta (ZURAWSKI, 2017). A relação dos códigos de função do protocolo Modbus pode ser vista na Figura 9.

Figura 9 – Funções Modbus para execução de comandos.

Memory Area	Function Name	Function Code (Hex)	Addressable Elements	Possible Response Error Codes
Discrete Inputs	Read discrete inputs	0x02	1–2000	01, 02, 03, 04
Coils	Read coils	0x01	1–2000	01, 02, 03, 04
Coils	Write single coil	0x05	1	01, 02, 03, 04
Coils	Write multiple coils	0x0F	1–1976	01, 02, 03, 04
Input registers	Read input registers	0x04	1–125	01, 02, 03, 04
Holding registers	Read holding registers	0x03	1–125	01, 02, 03, 04
Holding registers	Write single register	0x06	1	01, 02, 03, 04
Holding registers	Write multiple registers	0x10	1–123	01, 02, 03, 04
Holding registers	Read/write multiple registers	0x17	1–121 (write) 1–125 (read)	01, 02, 03, 04
Holding registers	Mask write register	0x16	1	01, 02, 03, 04
Holding registers	Read FIFO queue	0x18	1–32	01, 02, 03, 04
Files	Read file record	0x14		01, 02, 03, 04, 08
Files	Write file record	0x15		01, 02, 03, 04, 08

Fonte*:* Adaptado de (WILAMOWSKI; IRWIN, 2016)

As funções, indicadas pela representação hexadecimal como pode ser visto na terceira coluna da Figura 9, estão associadas ao acesso de áreas de memória (WILAMOWSKI; IRWIN, 2016).

Todas funções listadas, com exceção das funções 0x14, 0x15, 0x16 e 0x18, fazem o pedido de que algum valor seja lido ou escrito em uma área de memória específica. Já os códigos 0x05 e 0x06 são usados para escrever em um único elemento, *coil* ou *holding register*. Os códigos restantes permitem ao mestre ler ou escrever em múltiplos elementos da mesma área de memória (WILAMOWSKI; IRWIN, 2016).

Na Tabela 1, estão descritas, resumidamente, as características dos tipos de dados das áreas de memória do protocolo Modbus, vistos na Figura 9.

Tabela 1 – Tipos de dados Modbus.

Tipo de Dado Modbus	Tipo do Dado	Descrição
Discrete	Single bit	Do tipo *read-only*, ou seja, realiza a leitura. Esses valores são obtidos por meio de dispositivos de I/O.
Coil	Single bit	Do tipo *read-write* podendo realizar leitura e/ou escrita. Seu valor pode ser alterado por meio de uma aplicação mestre.
Input register	Word (16 bits)	Do tipo *read-only*, ou seja, realiza a leitura. Esses valores são obtidos por meio de dispositivos de I/O.
Holding Register	Word (16 bits)	Do tipo *read-write* podendo realizar leitura e/ou escrita. Seu valor pode ser alterado por meio de uma aplicação mestre.

Fonte: Adaptado de (ZURAWSKI, 2017)

Ainda na Figura 6, o campo *data* contém informações adicionais de um *request* e os resultados do processamento do código quando é uma resposta. Em caso de êxito, o conteúdo pode ser o que foi solicitado pelo código da função, em caso contrário, será um código de

exceção.

E por último, na Figura 6, está o campo de checagem de erro que permite que o mestre confirme a integridade da mensagem recebido do escravo. Ao receber a mensagem, o dispositivo calcula o erro o compara com o valor de verificação de erro recebido na mensagem. Se os dois valores estiverem de acordo, nenhum erro ocorreu e as ações foram tomadas adequadamente. A mensagem recebida é rejeitada se os dois valores forem diferentes O método de checagem depende do modo de transmissão utilizada, para o Modbus RTU é utilizado o CRC (*Cyclic Redundancy Check*) e para o Modbus ASCII é utilizado o LRC (*Longitudinal Redundancy Check*).

Como foi representado na Figura 5, na camada mais abaixo define como os *frames* da camada superior são encapsulados e codificados, enviados pelo meio físico. Existem três modos de transmissão para essa camada inferior, são eles:

- **ASCII** (*American Standard Code for Information*): realiza a transmissão serial de dados, ou seja, um bit por vez, e codifica a camada superior, a de aplicação, como caracteres ASCII, a cada byte são codificados como dois caracteres ASCII e cada caractere ASCII é transmitido usando 10 bits da seguinte forma (WILAMOWSKI; IRWIN, 2016):
    - 1 start bit (ST);
    - 7 bits de dados, para o caractere ASCII (0-9, A-F) com o bit menos significativo enviado primeiro;
    - 1 bit de paridade (PT);
    - 1 stop bit (SP).

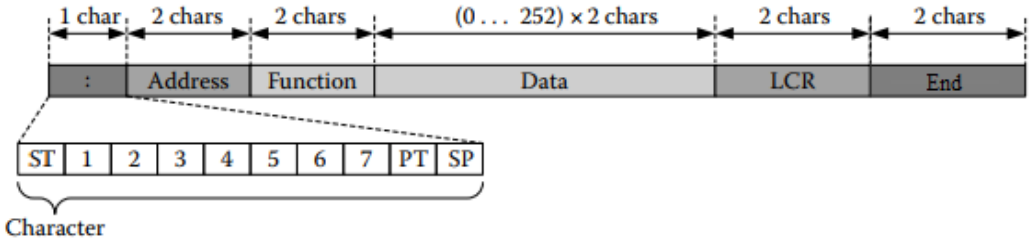
  O meio físico utilizado são os padrões EIA/TIA-232 ou EIA/TIA-485, também conhecidos como RS-232 e RS-485, respectivamente.

  O *frame do* MODBUS ASCII consiste em seis campos: *Start, Address, Function Code, Data, LRC, and End*. O *frame* começa com um *Start bit*, que é o dois pontos (:) na forma de caractere ASCII, 3Ah, onde h significando hexadecimal. O

frame termina com um caractere que indica que a aplicação pode analisar a próxima linha, indicado pelos caracteres ASCII 0Dh e 0Ah. O outro restante quatro campos intermediários têm caracteres de 0 a 9 e de A a F em hexadecimal (SEN, 2017).

O formato do *frame* ASCII pode ser visto na Figura 10.

Figura 10 – Formato do *frame* Modbus ASCII.



Fonte: Adaptado de (WILAMOWSKI; IRWIN, 2016)

- **RTU** (*Remote Terminal Unit*): realiza a transmissão serial de dados e codifica usando a representação de bits, tendo a identificação dos limites dos elementos baseada no tempo.

  Assim como no ASCII, a troca de informações ocorre pelos padrões como RS-232 e RS-485.

  Nesse modo, o *frame* começa com um intervalo de pausa correspondente a pelo menos 3,5 vezes o tamanho do caractere e termina com esse mesmo intervalo. Cada byte, no *frame*, é transmitido usando um caractere de 11 bits:

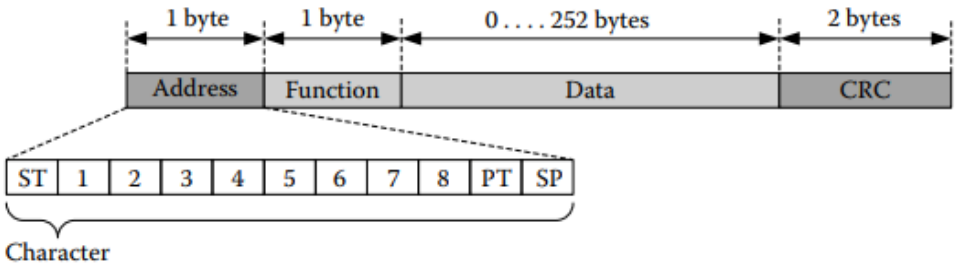
  - 1 start bit (ST), usado para sincronização inicial;
  - 8 bits de dados, em código binário com o bit menos significativo enviado primeiro;
  - 1 bit de paridade (PT), usado para detecção de erro. O tipo de paridade (ímpar, par ou sem paridade) pode ser escolhido pelo usuário, mas desde que todos os dispositivos da rede tenham o mesmo método de paridade,

- 1 stop bit (SP), para garantir um tempo mínimo ocioso entre a transmissão consecutiva de caracteres.

Cada caractere é transmitido com um intervalo de tempo de 1,5 vezes seu tamanho.

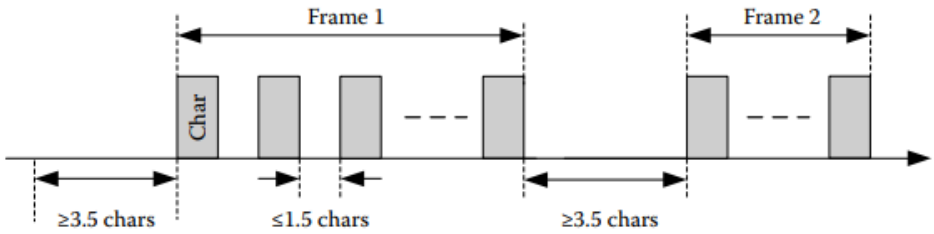
O formato do *frame* RTU e as especificações de tempo podem ser vistos nas Figuras 11 e 12, respectivamente.

Figura 11 – Formato do *frame* do Modbus RTU.



Fonte: Adaptado de (WILAMOWSKI; IRWIN, 2016)

Figura 12 – Requisitos de tempo para a transmissão de frames do Modbus RTU.



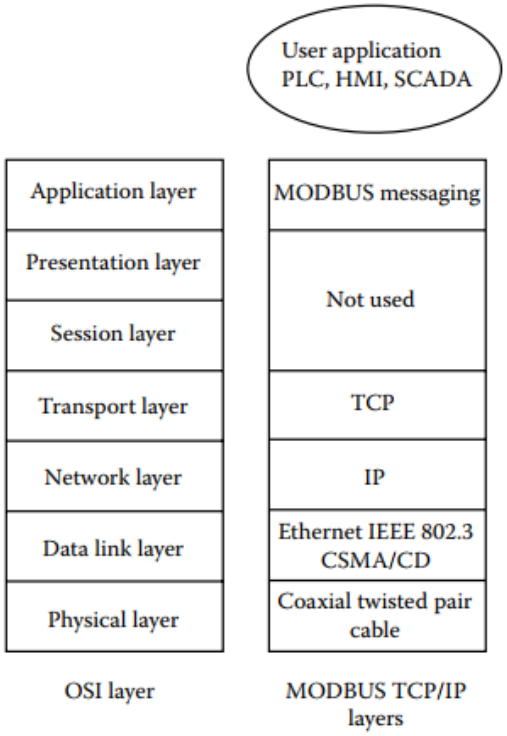
Fonte: Adaptado de (WILAMOWSKI; IRWIN, 2016)

- **TCP:** o envio de informações é feito por meio de conexões TCP estabelecidas por meio do protocolo Ethernet IP. A troca de informações ocorre pelo envio de f*rames* Modbus TCP por conexões previamente estabelecidas pelo protocolo TCP/IP, que pertence à camada de transporte, que ocorrem independentes do protocolo Modbus (WILAMOWSKI; IRWIN, 2016).

Na Figura 13, pode ser visto como é a organização em camadas do Modbus

TCP.

Figura 13 – Camadas do Modbus TCP/IP, pelo modelo ISO-OSI.



Fonte: Adaptado de (SEN, 2017)

A porta 502 é reservada para aplicações Modbus e é por onde ocorre a troca de *requests* para o estabelecimento de conexões.

O *frame* Modbus TCP possui um cabeçalho, o MBAP *(Modbus Application Protocol header) header*, além do ADPU visto na Figura 6. O cabeçalho possui os seguintes campos:

- *Transaction identifier:* em uma conexão TCP podem ser solicitadas trocas de informações seguidas, não é garantia que as respostas cheguem na mesma ordem que foram requisitadas. Então há um identificador para cada transação para preservar a ordem solicitada. Na solicitação esse campo é preenchido com um valor e na resposta esse mesmo número deve ser retornado;

- *Protocol identifier:* usado para identificar o protocolo utilizado. Deve

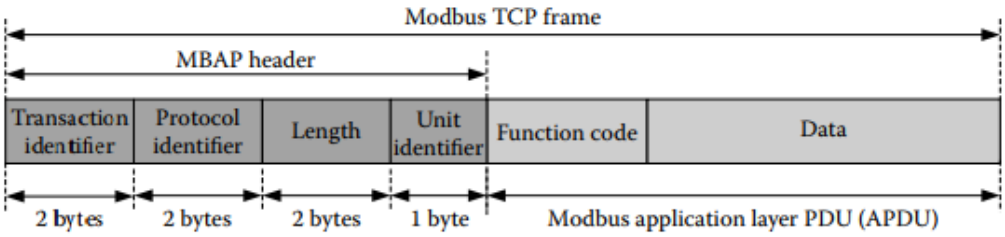
ser o valor 0;

- *Length:* indica o tamanho (em bytes) do cabeçalho e do APDU para assim identificar os limites do *frame*.

- *Unit identifier:* usado para identificar o dispositivo de destino, o escravo. Usado geralmente quando há gateways entre Modbus TCP e Modbus Serial, para ele consiga determinar qual elemento deve receber o *frame.* Quando não há gateways, esse campo é comumente ignorado.

Na Figura 14 pode ser visto o formato do *frame* Modbus TCP/IP.

Figura 14 – Formato do *frame* Modbus TCP/IP.



Fonte: Adaptado de (WILAMOWSKI; IRWIN, 2016).

Diferentemente do que ocorre nos modos Modbus Serial, o *frame* Modbus TCP não possui um campo para detecção de erros. Isso é devido à camada TCP/IP que já possui mecanismos de detecção de erros (WILAMOWSKI; IRWIN, 2016).

2.5 Controladores e Lógica de Programação

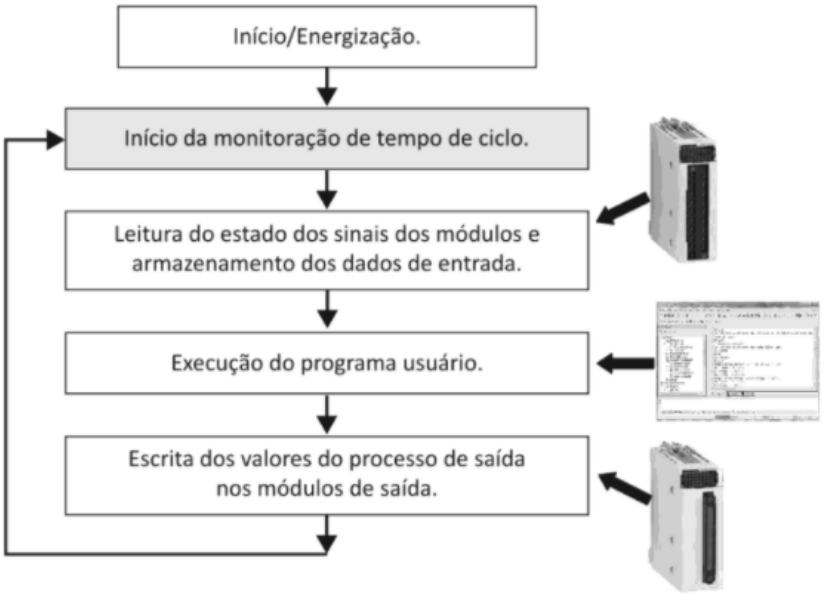
Como interpreta SILVA E. (apud INTERNACIONAL ELECTROTECHNICAL COMISSION, 1992, p.9), pela norma IEC 61131-1, um controlador lógico programável (CLP) é definido como:

"Sistema eletrônico digital, desenvolvido para uso em ambiente industrial, que usa

> uma memória programável para armazenamento interno de instruções do usuário, para implementação de funções específicas, como lógica, sequenciamento, temporização, contagem e aritmética, para controlar, por meio de entradas e saídas, vários tipos de máquinas e processos" (INTERNACIONAL ELECTROTECHNICAL COMISSION, 1992, p.9).

O controlador lógico programável executa um programa de controle, lê os estados de cada entrada e verifica se alguma foi acionada. Esse processo é chamado de ciclo de varredura ou *scan* e geralmente dura na ordem microssegundos. Após o ciclo de varredura, o CLP armazena os resultados obtidos e usa-os ao longo da execução do programa do usuário. A partir disso, a CPU do controlador lógico programável atualiza as saídas, de acordo com as instruções definidas no programa do usuário, escreve o valor contido na memória das saídas e atualiza as interfaces ou módulos de saída. Em seguida inicia-se um novo ciclo de varredura. Essa sequência está representada na Figura 15.

Figura 15 – Ciclo de *scan* de um CLP



Fonte: Adaptado de (SILVA, E. 2016)

A arquitetura básica de um CLP é composta pelos seguintes elementos:

- **Unidade de entrada:** recebe sinais elétricos dos equipamentos ou processo;
- **Unidade de saída:** recebe os sinais processados pelo CLP e disponibiliza um

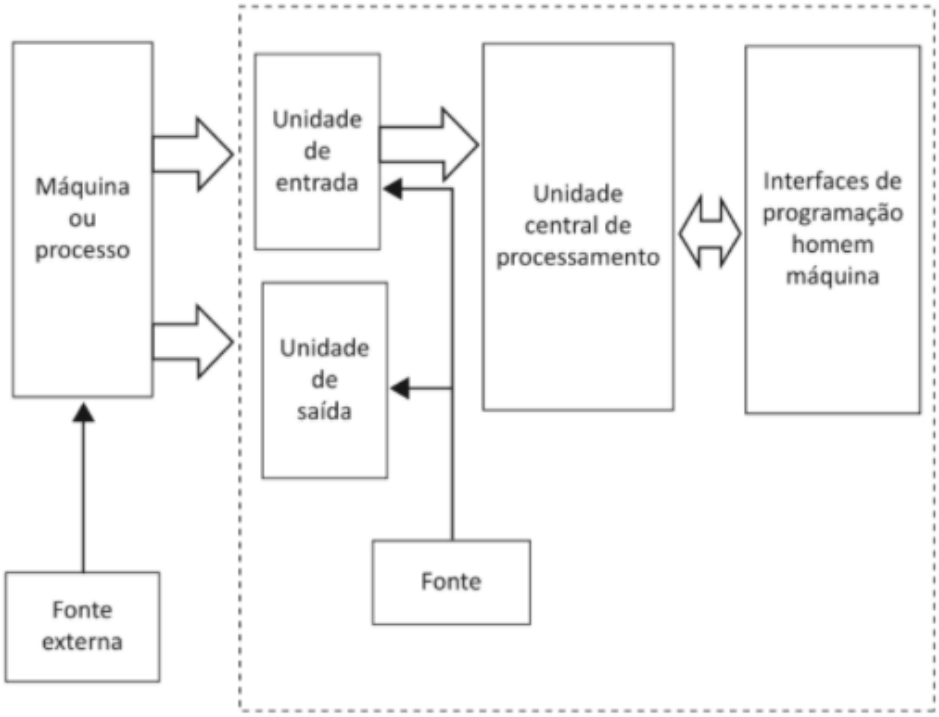
sinal elétrico para utilização na máquina ou processo;

- **Unidade de processamento:** responsável por administrar todas as funções, recebe sinais da unidade de entrada, executa a lógica do programa do usuário e envia o resultado para a unidade de saída;

- **Fonte de alimentação:** adapta a energia elétrica para o funcionamento do CLP.

A organização dessas estruturas pode ser vista na Figura 16.

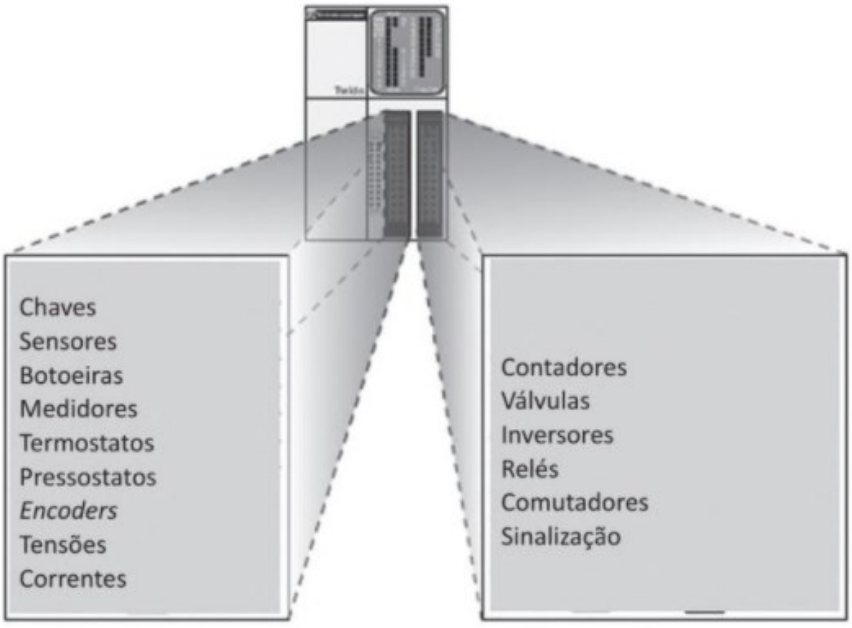
Figura 16 – Arquitetura básica do CLP.



Fonte: Adaptado de (SILVA, E. 2016)

As unidades de entradas e saídas, também chamados de módulos, são constituídos de cartões, assim como exemplificado na Figura 17.

Figura 17 – Entradas e saídas de um CLP.



Fonte: Adaptado de (SILVA, E. 2016)

Os módulos podem ser classificados como discreto ou analógico. As unidades de entrada e saída discreto admitem apenas dois estados:

- Ligado – *On;*
- Desligado – *Off.*

Na Figura 18 pode ser visto o comportamento de um sinal discreto.

Figura 18 – Sinal discreto.

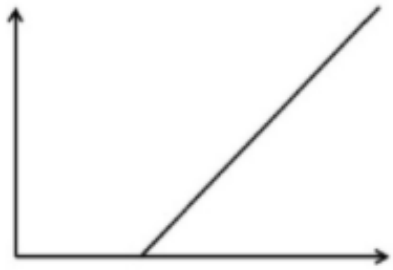
Fonte: Adaptado de (MEHTA; REDDY, 2014)

As unidades analógicas admitem mais de dois estados possíveis, ver Figura 19, entre dois limites determinados. As grandezas analógicas elétricas tratadas por um Controlador Lógico Programável são tensão e corrente.

- Corrente: 0 a 20 mA; 4 a 20 mA.
- Tensão: 0 a 10 Vcc; 0 a 5 Vcc; -5 Vcc a 5 Vcc; -10 Vcc a 10 Vcc.

As entradas analógicas recebem o sinal analógico e o converte em valores numéricos, alguns dispositivos usados nessas entradas são o *encoder* e transmissores de pressão, nível, temperatura etc. Já as saídas analógicas, converte valores numéricos em sinais de tensão ou corrente para controlar dispositivos como inversores de frequência e válvulas posicionadoras (SILVA, E. 2016).

Figura 19 – Sinal analógico.



Fonte: Adaptado de (MEHTA; REDDY, 2014)

Além desses, os CLPs podem aceitar módulos especiais como módulo Ethernet, módulo de comunicação com redes, módulos de medição de parâmetros elétricos, entre outros.

Para identificar objetos de dados cujo conteúdo pode mudar, como dados associados com entradas, saídas ou na memória do controlador são usadas variáveis que podem ter nomes atribuídos pelo usuário e uma identificação quanto ao seu endereço de memória a partir do seu escopo. Na Figura 20 pode ser visto como o endereço das variáveis podem ser representados.

Figura 20 – Formas de identificação do endereço de uma variável de um programa de CLP.

Sinal inicial	Identificação de memória	Tamanho do dado		Descrição
%	**M** (Acesso à memória)	**X**	(1 bit)	Acesso às variáveis booleanas.
	I (Entrada física)	**W**	(16 bits)	Acesso às variáveis com 16 bits de tamanho: INT, UINT e WORD.
	Q (Saída física)	**D**	(32 bits)	Acesso às variáveis com 32 bits de tamanho: DINT, UDINT, DWORD.
		T	(32 bits)	TIME, DATE, TOD e DATE_AND_TIME.
		R	(32 bits)	Acesso às variáveis com 32 bits de tamanho: REAL.
		A	----------	O conteúdo dessa região é definido pelo usuário conforme a necessidade do projeto.

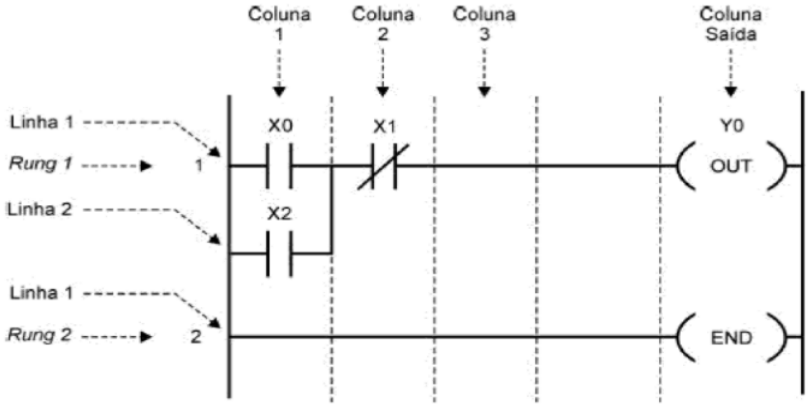
Fonte: Adaptado de (SILVA, E. 2016)

De acordo com SILVA E. (2016), a norma IEC61131-3 especifica a semântica e a sintaxe de um conjunto de linguagens de programação para um CLP. As mais comuns são: texto estruturado (*Structured Text*, ST), diagrama de blocos funcionais (*Function Block Diagram*, FBD), diagrama *Ladder* (*Ladder Diagram,* LD) e gráfico sequencial de função (*Sequential Function Chart*, SFC). Os elementos das linguagens são definidos na programação do controlador, e os recursos estabelecidos determinam a comunicação entre o CLP e outros componentes do sistema (SILVA, E. 2016)

A linguagem de diagrama *Ladde*r, também conhecida como diagrama de relés, diagrama escada ou diagrama de contatos, possui o objetivo principal de controlar o acionamento de saídas, dependendo da combinação lógica dos contatos de entrada, e é a linguagem de programação mais utilizada em CLPs sendo semelhante a um diagrama elétrico (ROGGIA; FUENTES, 2016).

O nome *Ladder* deve-se à representação da linguagem se parecer com uma escada, do inglês *ladder*, na qual duas barras verticais paralelas são interligadas pela lógica de controle, formando degraus (*rungs*) da escada. Cada lógica de controle existente na aplicação é chamada de *rung*, que é composta de linhas e colunas sob quais são colocadas as instruções a serem executadas, conforme pode ser visto na Figura 21 (GEORGINI, 2018).

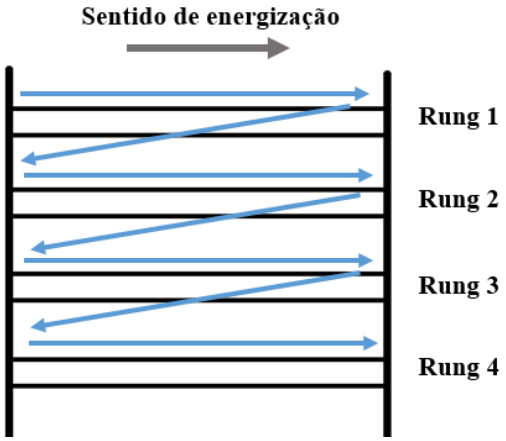
Figura 21 – Componentes da programação em linguagem *Ladder*.



Fonte: Adaptado de (GEORGINI, 2018)

Para as instruções serem executadas, elas devem ser "energizadas" a partir de um "caminho de corrente" entre as duas barras, em uma lógica flui no sentido da barra da esquerda para a barra da direta, como visto na Figura 22 (ROGGIA; FUENTES, 2016).

Figura 22 – Esquema do sentido da lógica *Ladder*.

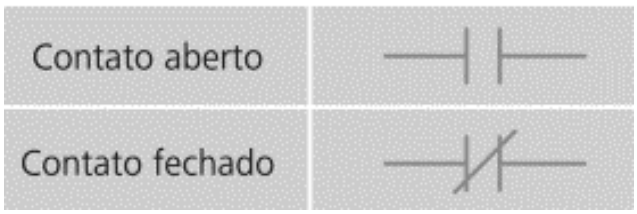


Fonte: Adaptado de (MEHTA; REDDY, 2014)

Os principais elementos na composição das instruções da lógica de controle são:

- **Contatos:** representação do dispositivo conectado ao módulo de entrada. Assume valores 0 ou 1 e pode ser representado como NA (normalmente aberto = 0) ou NF (normalmente fechado = 1). Na Figura 23 pode ser visto como são representados.

Figura 23 – Símbolos gráficos dos contatos em linguagem *Ladder*.



Fonte: Adaptado de (ROGGIA; FUENTES, 2016)

- **Bobinas:** representação das saídas da lógica de controle. É acionada somente quando todas as condições do *rung* são satisfeitas. Seu símbolo está na Figura 24

Figura 24 – Símbolos gráficos das saídas em linguagem *Ladder*.

Fonte: Adaptado de (ROGGIA; FUENTES, 2016)

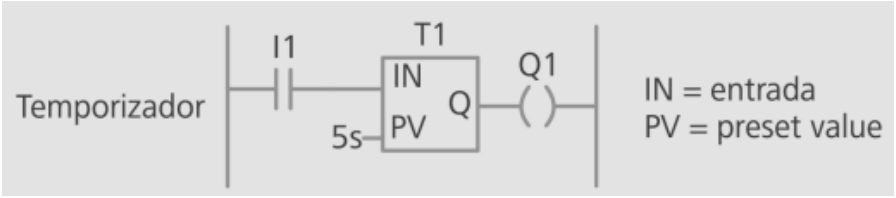
- **Temporizadores**: acionam ou desligam uma memória ou uma saída de acordo com um tempo especificado. Possui dois modos:

    - Retardo na energização: uma saída será ligada após decorrido um determinado tempo a partir do acionamento do temporizador.

    - Retardo na desenergização: uma saída será desligada após decorrido um determinado tempo a partir do acionamento do temporizador.

    Essas funções são utilizadas em chaves de partida de motores de indução, como a partida estrela-triângulo, por exemplo. Uma forma de como um temporizador pode ser implementado em uma lógica está na Figura 25.

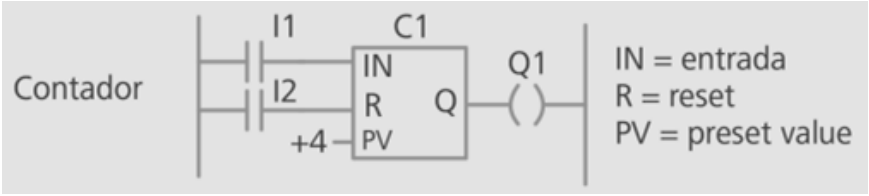
Figura 25 – Representação de uma lógica utilizando temporizador.



Fonte: Adaptado de (ROGGIA; FUENTES, 2016)

- **Contadores:** ativam uma memória ou uma saída após uma determinada contagem de eventos. Um contador crescente pode acionar uma saída após um botão ou um sensor ter sido acionado um determinado número de vezes previamente programado, conforme mostrado na Figura 26.

Figura 26 – Representação de uma lógica utilizando contador.



Fonte: Adaptado de (ROGGIA; FUENTES, 2016)

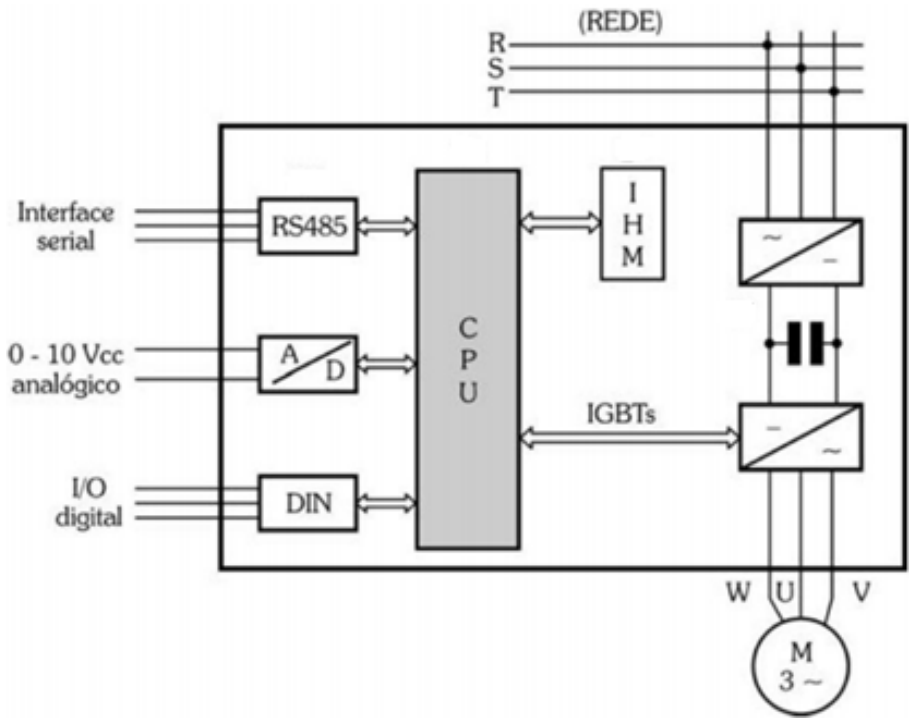
2.6 Inversor de Frequência

Um inversor de frequência é um dispositivo usado para acionamento, variação da frequência e tensão para controle da velocidade de giro e potência consumida de um motor elétrico trifásico.

Para controlar a potência consumida pela carga, o inversor de frequência varia a frequência fornecida pela rede, transformando a corrente e tensão alternada fixa em corrente e tensão alternada variável.

Os componentes de um inversor de frequência podem ser agrupados em blocos como pode ser visto na representação da Figura 27.

Figura 27 – Representação em blocos dos componentes de um inversor de frequência.



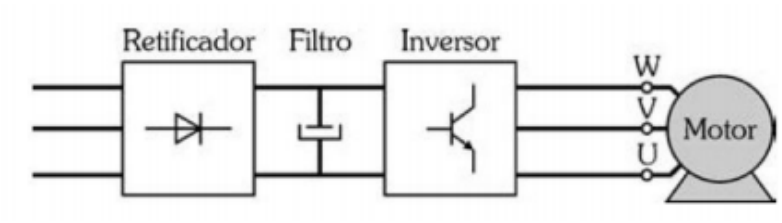
Fonte: Adaptado de (FRANCHI, 2013)

Pelo *display* da IHM é por onde pode-se visualizar o que está acontecendo com o inversor e pelas teclas é possível parametriza-lo de acordo com a aplicação.

Na CPU é onde fica armazenado todos os parâmetros e dados do sistema. O inversor possui um sistema de controle inclui o controle do inversor, a leitura da velocidade do motor, leitura de corrente e o sistema de interfaces que engloba o ajuste de parâmetros pelo usuário por configuração local ou via protocolos industriais, envio de informações para o operador e diagnósticos de falha por IHM, e leitura de entradas e saídas digitais e analógicas para receber sinais de controle, como de partida de parada, e enviar informações como status do motor, falhas e outros.

Além disso, a CPU é responsável por executar a lógica de geração de pulsos de disparo para os IGBTs (*Insulated Gate Bipolar Transistor*). Os IGBTs compõem o módulo do circuito de saída do inversor. Esse módulo é composto por um retificador, um filtro e um conversor inversor, como pode ser visto na Figura 28 e detalhados a seguir.

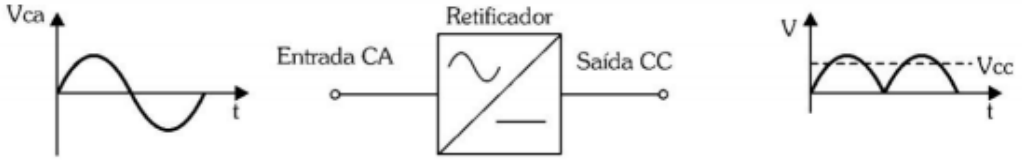
Figura 28 – Esquema de um conversor de frequência aplicado a um motor trifásico.



Fonte: Adaptado de (FRANCHI, 2013)

O retificador é um conversor que tem a função de transformar uma grandeza CA em CC.

Figura 29 – Conversor tipo retificador.

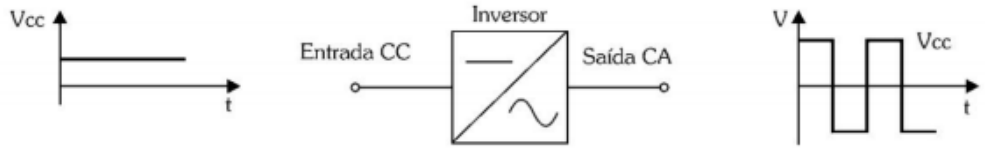


Fonte: Adaptado de (FRANCHI, 2013)

O filtro é um circuito intermediário, também chamado de link CC com filtro, para transformar a forma de onda em senoidal.

E por último, o inversor é um tipo de conversor que converte uma grandeza CC em CA.

Figura 30 – Conversor tipo inversor.



Fonte: Adaptado de (FRANCHI, 2013)

Dessa forma, o inversor de frequência altera a frequência na entrada do motor, que deixa de ser a frequência fixa da rede, e sim um valor definido pelo usuário. Isso implica com

que a velocidade do motor seja proporcional ao valor de frequência aplicada, já que como mostrado por Chapman (2013), a velocidade de rotação do motor, ou velocidade síncrona, é dada pela Equação 1.

$$V_s = \frac{120 \times f}{P} \qquad (1)$$

em que f é a frequência aplicada ao estator do motor, em hertz, e P é o número de polos da máquina.

2.7 Sistemas Supervisórios

Sistemas supervisórios são sistemas digitais de monitoração e operação de informações de plantas que gerenciam variáveis de processo de forma remota (SEIDL et al., 2014). Por meio desse recurso reúne toda a informação e disponibiliza-a em monitores que ficam na sala de controladores e exibe a forma concisa podendo usar gráficos, botões, controles, animações, além de poderem ser guardadas em bancos de dados locais ou remotos para fins de registro.

Esses sistemas intermediam a transferência de dados entre uma estação supervisória e as remotas com a leitura de valores atuais dos dispositivos que a eles são associados.

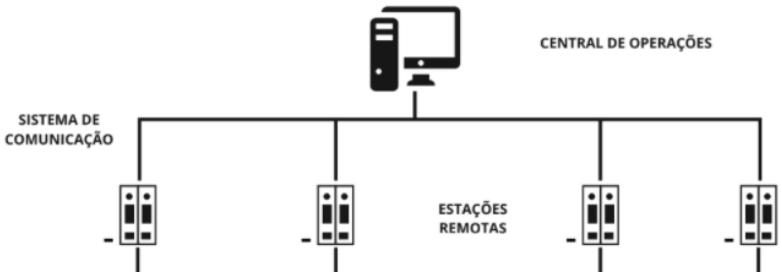
Os sistemas de supervisão estão localizados no topo da pirâmide dos sistemas de automação, ficando responsável pelas funções de supervisão e gerenciamento que incluem (GARCIA JR., 2019):

- **Supervisão:** operação, monitoramento, gerenciamento de alarmes e eventos, comandos e intervenções como mudança de *setpoint*, fechamento de válvulas, início de processos, entre outros.

- **Gerenciamento:** manipulação da base de dados, controle operacional, otimização dos processos, simulação, modelagem etc.

São formadas por uma central de operações e diversas unidades remotas, que são responsáveis pelo recolhimento de dados localmente para posterior envio para essa central. Por fim, um software na central fica responsável por correlacionar os dados, interpretá-los e gerenciá-los, conforme representado na Figura 31.

Figura 31 – Elementos de um sistema SCADA.



Fonte: Adaptado de (GARCIA JR., 2019)

3 METODOLOGIA

Neste capítulo será descrito as características do sistema, a proposta e a implementação do controle do motor de um sistema de esteira transportadora, por meio do inversor de frequência e do controle da célula de carga através de I/O local.

3.1 Sistema original

A esteira transportadora, que pode ser vista na Figura 32, trata-se de uma planta didática que permite acionamento por meio de três formas distintas: partida direta, *soft starter* ou inversor de frequência e controle da movimentação da esteira através da célula de carga, além de possibilitar a medição e análise da qualidade de energia.

Figura 32 – Sistema de esteira transportadora.

A representação desse sistema pode ser vista nas Figuras 33 e 34 indicando os elementos integrantes.

Figura 33 – Vista frontal da esteira transportadora.

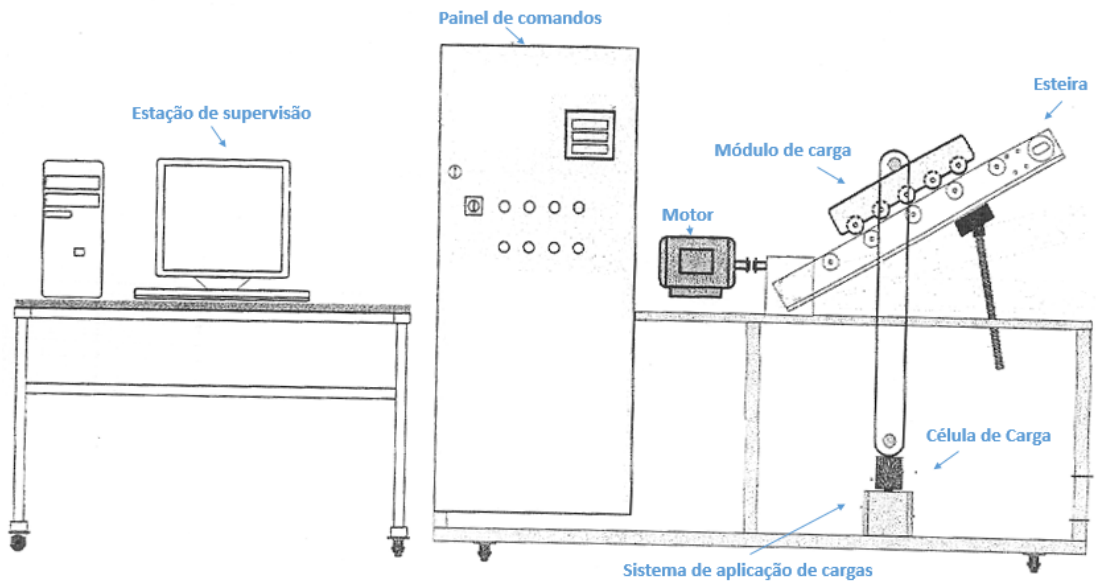
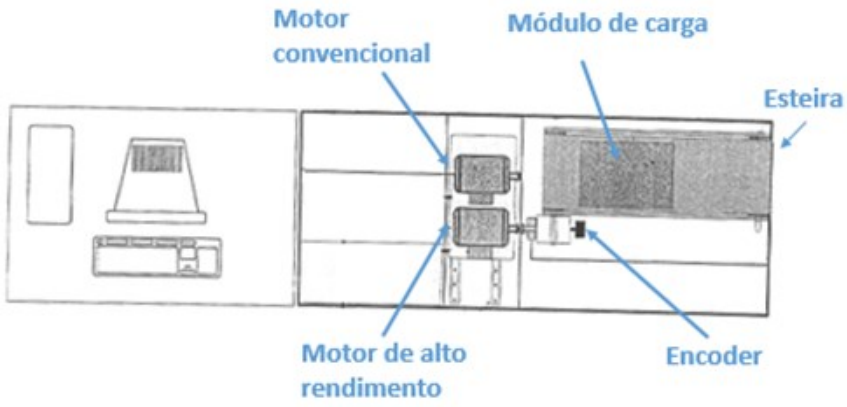


Figura 34 – Vista superior da esteira transportadora.



3.1.1 Equipamentos de Campo

Na estrutura central, a esteira – que permite uma inclinação de até 30º, pode ser movimentada pelo motor de alto rendimento ou o motor convencional, sendo que atualmente, a esteira está conectada ao motor de alto rendimento.

O motor de alto rendimento possui as seguintes especificações técnicas:

- Tipo: Motor de indução – Gaiola;
- Modelo: HB64548;

- Grau de Proteção: IP55;
- Isolação: F;
- Regime: S1;
- Potência Nominal: 1,10 kW (1,50 CV);
- Tensões Nominais: 220/380 V;
- Correntes Nominais: 4,00/2,32 A;
- Frequência Nominal: 60 Hz;
- Velocidade de Rotação Nominal: 3400 RPM;
- Razão da Corrente de Partida pela Corrente Nominal (Ip/In): 7,50;
- Categoria de desempenho: N;
- Fator de Serviço: 1,15;
- Rendimento Nominal: 83,00;
- Fator de Potência Nominal: 0,87.

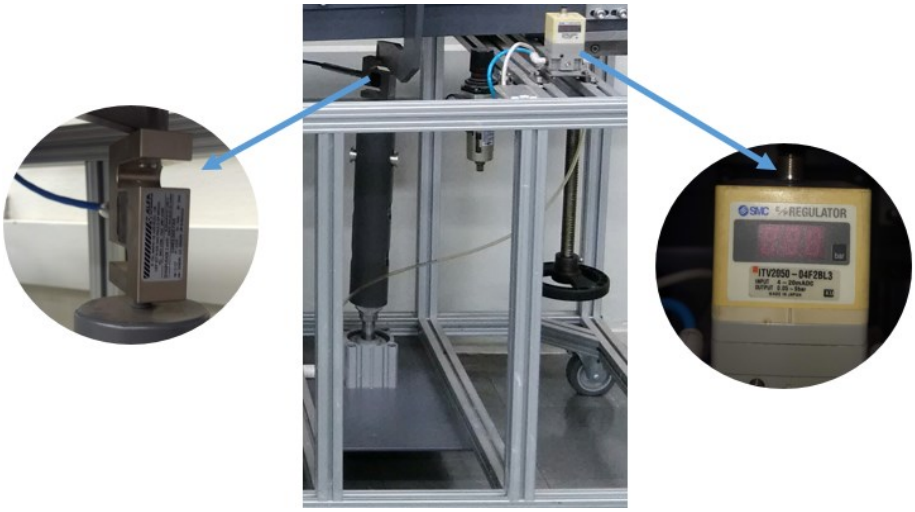
A velocidade da rotação do motor é indicada por um *encoder* taco gerador analógico (Hohner TH22R4500). Esse sensor possui eixo rígido e sinal de saída de 4 a 20 mA, alimentação de 24 Vcc e possui a escala máxima de velocidade de 4500 RPM.

Na estrutura de suporte da esteira há uma célula de carga (Alfa Z 2mV/V), alimentada pela tensão de 24 Vcc, com a capacidade nominal de 250 kg a 5000 kg e sensibilidade de 2mV/V, responsável por medir o peso que está sendo aplicado na esteira.

O regulador eletropneumático (SMC ITV2050 04F2BL3) varia a pressão da eletroválvula de alimentação de acordo com o valor do sinal elétrico de entrada aplicado, atuando no sistema de aplicação de cargas. O regulador é alimentado com tensão de 24 Vcc, possui pressão mínima de alimentação de 0,1 MPa e pressão máxima de alimentação é de 0,2 MPa e o sinal de entrada deve ser entre 4 e 20mA.

A célula de carga e o atuador pneumático podem ser vistos na Figura 35.

Figura 35 – Esquema localizando a célula de carga (à esquerda) e o regulador eletropneumático (à direita) no sistema da esteira (ao centro).



No painel de comando é onde estão os elementos para distribuição da energia e controle dos equipamentos.

Figura 36 – Painel de comando da esteira transportadora.

Nesse painel estão alocados os contatores, disjuntores, transformadores de corrente, fusíveis, bornes, cabos, fonte de alimentação, um medidor de energia, inversor de frequência, *soft starter*, partida direta e o controlador. Os elementos de partida do motor podem ser vistos na Figura 37.

Figura 37 – Da esquerda para direita, *soft starter* (Telemecanique Altistart 48), inversor de frequência (Telemecanique Altivar 31), partida direta (Telemecanique Tesys IUCB05BI e Power Logic PM850UMG).

Para esse trabalho, será focada na partida por meio do inversor de frequência.

O Telemecanique Altivar 31, também chamado de ATV31, é usado com máquinas simples de 0,18 kW a 15 kW de potência e possui dois tipos de alimentação:

- 200 V a 240 V monofásico, para máquinas de 0,18 kW a 2,2 kW;
- 380 V a 500 V trifásico, para máquinas de 0,37kW a 15 kW;

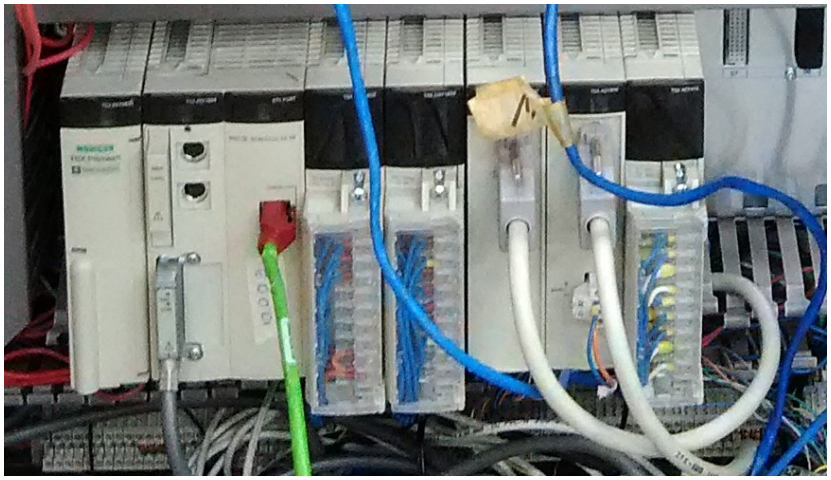
Além do terminal integrado para controle local, suporta também os protocolos industriais de comunicação Modbus e CANopen.

3.1.2 Sistema de Controle

O controlador responsável por realizar a comunicação da rede de campo, fica

localizado na parte inferior do painel de comandos e pode ser visto na Figura 38.

Figura 38 – Controlador (Schneider TSX Premium) da rede de comunicação da bancada da esteira.



O controlador é composto por módulos que estão representados pela Figura 39 e detalhados a seguir.

Figura 39 – Representação dos módulos do controlador.

PSY 2600	P57 1634m	ETY PORT	DEY 16D2	DSY 16R5	AEY 800	ASY 800	AEY 414
Fonte 24 VDC	CPU	Ethernet	16I 24VDC	16Q Relay	8AI 24VDC	AO	4I RTD

- P57 1634m: é a CPU do controlador; usa cartão PCMCIA para comunicação da CPU com o ambiente RS-485.

- ETY PORT: módulo Ethernet que possibilita a comunicação Modbus TCP/IP da estação de supervisão com o controlador.

- DEY16D2: módulo com 16 entradas discretas onde estão conectadas as entradas de status de qual motor está conectado, motor convencional ou de alto rendimento.

- DSY16R5: módulo com 16 saídas discretas onde estão conectados os contatores, entre eles K2 que aciona a partida direta, K3 e K4 que acionam o *soft starter*, o contatores K5 e K6 que acionam o inversor de frequência.

- AEY800: módulo com 8 entradas analógicas, que recebe os dados da célula de carga e do *encoder*.

- ASY800: módulo com 8 saídas analógicas, que envia os valores de para o sinal de entrada do regulador eletropneumático.

- AEY414: módulo com 4 entradas analógicas que possui conexões termoelétricas para medição de temperatura. Para cada entrada é usado um sensor de temperatura (PT100) para medição da temperatura das fases do motor.

3.1.3 Estação de Supervisão

A estação de supervisão é por onde o operador interage com o sistema. Por meio de um supervisório é possível ligar a esteira, selecionar a forma de partida, alterar velocidade e monitorar variáveis como rotação do motor, velocidade da esteira, temperatura do motor, tensão de cada fase, entre outras.

3.1.4 Desvantagens da automação original

O controlador do sistema (Schneider TSX Premium) é um equipamento que foi descontinuado pelo fabricante, e, portanto, não é oferecido suporte para esse modelo, impossibilitando por exemplo, a troca de cartões de I/Os ou a aquisição de módulos de comunicação, caso necessário.

Além disso, a lógica de controle é desenvolvida em um ambiente defasado, instalado em um sistema operacional Windows XP ou anterior, fazendo com que seja necessário o uso de máquinas virtuais para executar esse software de configuração.

Outra questão de obsolescência é quanto ao cartão de comunicação do controlador, baseado na tecnologia PCMCIA, que hoje em dia é considerada ultrapassada e de difícil acesso e manutenção (TECHLIB, 2021).

Em vista disso, uma adequação do sistema se faz pertinente, já que ele ainda é funcional

apesar de estar desatualizado. Assim sendo, é necessário buscar tecnologias mais recentes para tornar o sistema viável às práticas de laboratório nas disciplinas de Controle e Automação.

3.2 Proposta de automação para o sistema

Como dito, este trabalho busca viabilizar a planta para o uso no laboratório de Redes Industriais II do curso de Engenharia de Controle e Automação da Universidade Federal de Uberlândia, além de apresentar uma abordagem alternativa.

Dessa forma, este trabalho visa verificar se é possível substituir o controlador atual (Schneider TSX Premium) pelo controlador Siemens SIMATIC S71200, que é amplamente utilizado no laboratório em várias disciplinas do curso de Engenharia de Controle e Automação.

Atualmente, considera-se duas soluções possíveis: (1) Substituição total do controlador Schneider pelo controlador Siemens S71200; ou (2) utilização do S71200 em conjunto com o Schneider.

Considerando a primeira solução, o controlador Siemens S71200, mostrado na Figura 40, aceita até oito módulos de sinal para entradas digitais, saídas digitais, entradas analógicas e saídas analógicas.

Figura 40 – Controlador Siemens SIMATIC S7-1200-1214C do laboratório.

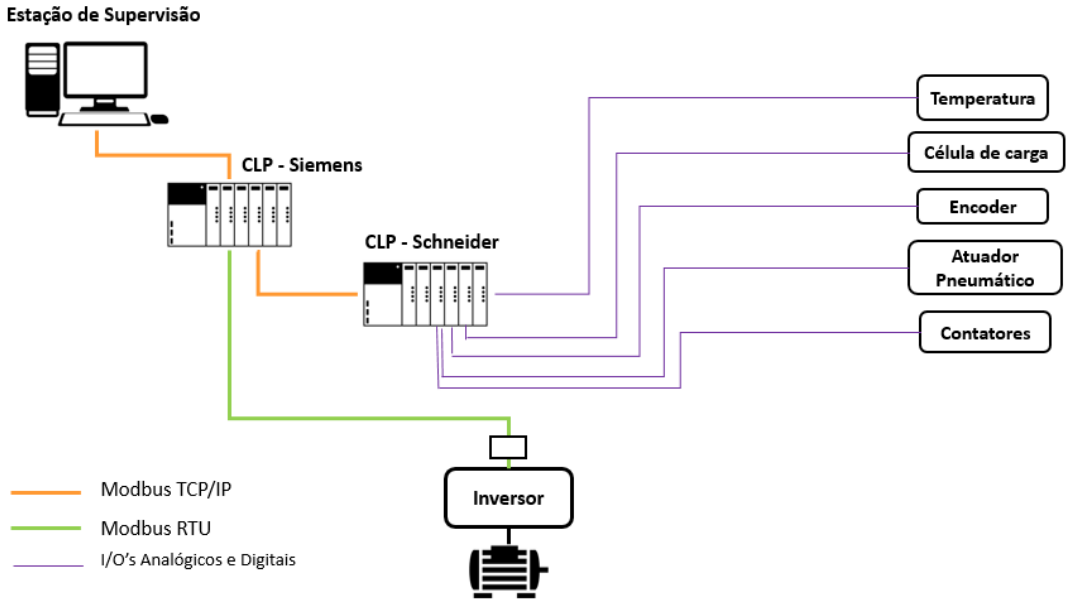
Os exemplares Siemens S71200 disponíveis no laboratório possuem o módulo CM 1214 para comunicação Modbus RS-485 e RS-232. Mas para execução dessa adaptação é

necessário realizar modificações na instalação do sistema, como um conversor de tensão de 24Vdc para 10Vdc para as entradas analógicas – o *encoder* e célula de carga, e a aquisição de um módulo de medição de temperatura (SB 1231 RTD).

No caso da segunda opção, iria fazer uso de parte da estrutura original e trabalhar com dois CLP's no sistema, onde CLP Siemens seria o principal e o CLP Schneider seja uma remota em que teria o papel de somente para disponibilizar os I/O's locais para o CLP principal através do protocolo Modbus. Já o CLP Siemens seria o responsável por acessar os I/O's do CLP Schneider, o inversor de frequência e controlar todo o sistema.

Com relação a comunicação, o CLP Siemens comunica com o inversor de frequência através do protocolo Modbus RTU. O CLP Siemens comunica com o CLP Schneider através do Modbus TCP. E o CLP Siemens comunica com o supervisório através de Modbus TCP ou Profinet. Na Figura 41 está ilustrada essa proposta de comunicação.

Figura 41 – Proposta de comunicação para o sistema de esteira transportadora.



Analisando as duas propostas, foi escolhida a segunda, onde são utilizados dois controladores devido à facilidade de implementação, proveito da instalação já existente e por utilizar a filosofia de sistema remoto por descentralizar o controle e o envio de informações. A seguir será apresentado as etapas para execução e características da proposta selecionada.

3.2.1 Sistema de Controle

Nesta seção será descrito a construção do controle do sistema, que tem por objetivo acionar e configurar a velocidade da esteira, de acordo com a velocidade de trabalho selecionada pelo usuário, através da estrutura representada na Figura 41.

Em relação ao funcionamento geral do sistema, para acionar a esteira utilizando o inversor de frequência são necessárias as seguintes etapas:

1. Energizar o elemento de partida;
2. Ligar o inversor de frequência;
3. Configuração da velocidade.

A Etapa 1 consiste em após o CLP Schneider obter os valores das entradas e saídas, ele recebe um comando do CLP Siemens, para ligar os contatores K3 e K4 responsáveis por energizar o inversor de frequência. A troca de dados entre os dois CLP's ocorre através do protocolo Modbus TCP.

As Etapas 2 e 3 estão associadas. Quando o inversor de frequência estiver energizado, fica disponível a troca de informações, através do protocolo Modbus RTU, entre o inversor e o CLP Siemens para que então seja possível ligá-lo e então fazer o ajuste da velocidade desejada.

Dessa forma, de acordo com o funcionamento descrito, o sistema de controle pode ser dividido em duas partes, a parte Modbus TCP e a parte Modbus RTU.

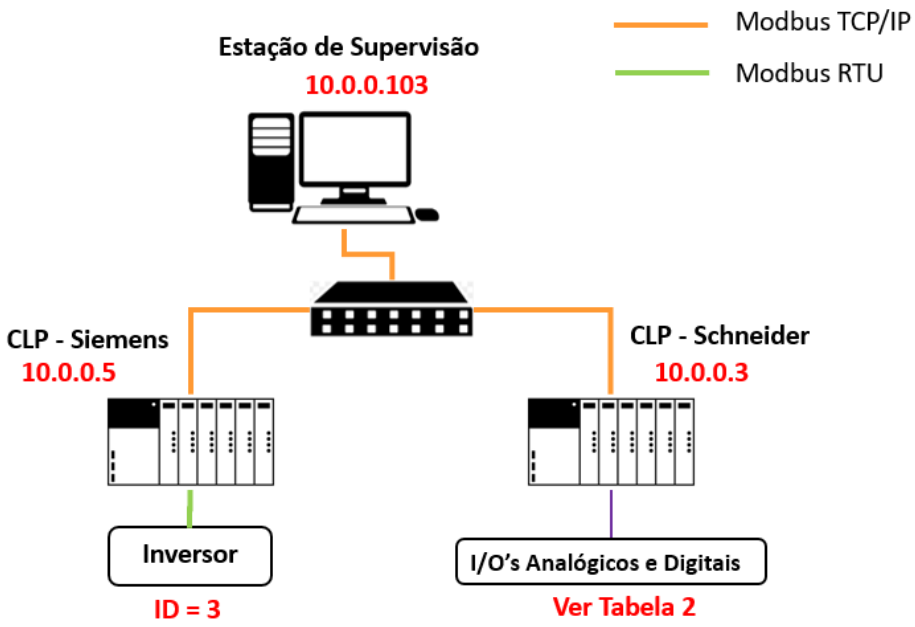
A seguir será detalhado os passos necessários para a execução das Etapas 1, 2 e 3.

3.2.1.1 Parametrização dos dispositivos

Considerando que a proposta seguida consiste em utilizar a estrutura existente, foram utilizados os equipamentos descritos na seção 3.1.1.

Na Figura 42 pode ser visto o endereço de cada elemento na rede montada de acordo com o esquema da Figura 41.

Figura 42 – Endereçamento dos equipamentos.



No CLP Schneider, as entradas e saídas analógicas e digitais são acessadas pelos endereços como descrito na Tabela 2.

Tabela 2 – Endereços da entradas e saídas digitais e analógicas.

Descrição da variável	Endereço
Contator K3	%Q0.3.1
Contator K4	%Q0.3.2
Contator K5	%Q0.3.3
Contator K6	%Q0.3.4
Leitura da célula de carga	%IW0.4.0
Leitura do *encoder*	%IW0.4.1
Sinal de entrada para o atuador pneumático	%QW0.5.1
Status do motor convencional	%I0.2.0
Status do motor de alto rendimento	%I0.2.1
Temperatura da carcaça do motor	%IW0.6.3
Temperatura da fase 1	%IW0.6.0
Temperatura da fase 2	%IW0.6.1
Temperatura da fase 3	%IW0.6.2

Ainda, os registros Modbus acessados para preparar o inversor de frequência para o acionamento do motor estão descritos vistos na Tabela 3.

Tabela 3 - Registros Modbus acessados para parametrizar o inversor de frequência.

Tipo	Endereço Modbus	Descrição
Read/Write	8501	Registro de controle – por onde são definidos os estados de operação do inversor
Read/Write	8502	Velocidade (frequência)
Read/Write	9001	Tempo de aceleração
Read/Write	9002	Tempo de desaceleração
Read-only	3201	Variável de monitoramento dos estados do inversor
Read-only	3206	Variável de monitoramento do motor

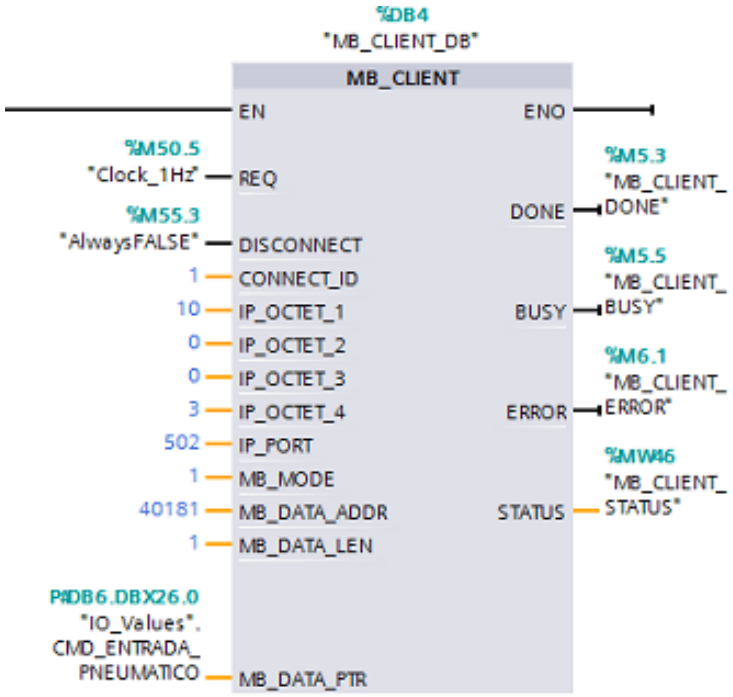
3.2.1.2 Rede Modbus TCP

Nesta seção será descrito a estratégia de controle desenvolvida para parte da rede Modbus TCP.

Para essa lógica, o CLP Siemens é configurado como mestre. No ambiente Siemens, chamado TIA Portal, essa configuração é feita por meio do bloco MB_CLIENT, onde são definidos parâmetros como o tempo de requisição (REQ), se será realizado leitura ou escrita (MB_MODE), o registro inicial a ser lido/escrito (MB_DATA_ADDR), a quantidade de registros a serem lidos/escritos a partir do registro inicial (MB_DATA_LEN), os dados a serem recebidos/enviados (MB_DATA_ADDR), além do endereço IP do escravo, nesse caso o CLP Schneider.

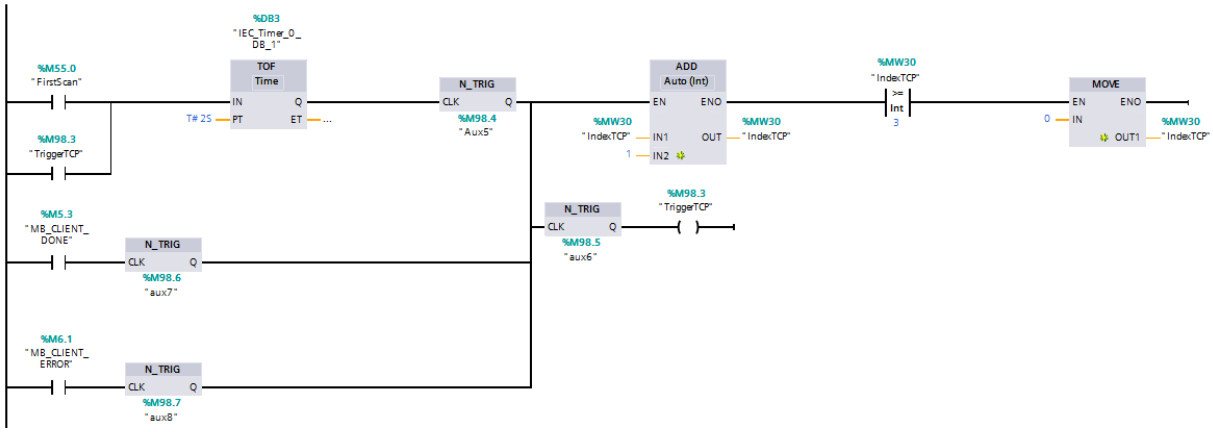
A configuração usada em um dos blocos MB_CLIENT pode ser vista na Figura 43.

Figura 43 – Configuração usada em um dos blocos MB_CLIENT.



Nesse caso, é necessário fazer múltiplos acessos ao mesmo escravo, ou seja, é necessário alternar chamadas de leitura e escrita. Por exemplo, para obter os valores das entradas e saídas lidos pelo CLP Schneider é feita uma leitura e para selecionar uma partida é preciso acionar os contatores correspondentes, então é feita uma escrita. Dessa forma, foram configurados blocos MB_CLIENT de leitura e escrita, de acordo com o necessário, e desenvolvida uma estratégia de modo que apenas uma destas chamadas seja executada por vez. Essa implementação pode ser vista na Figura 44.

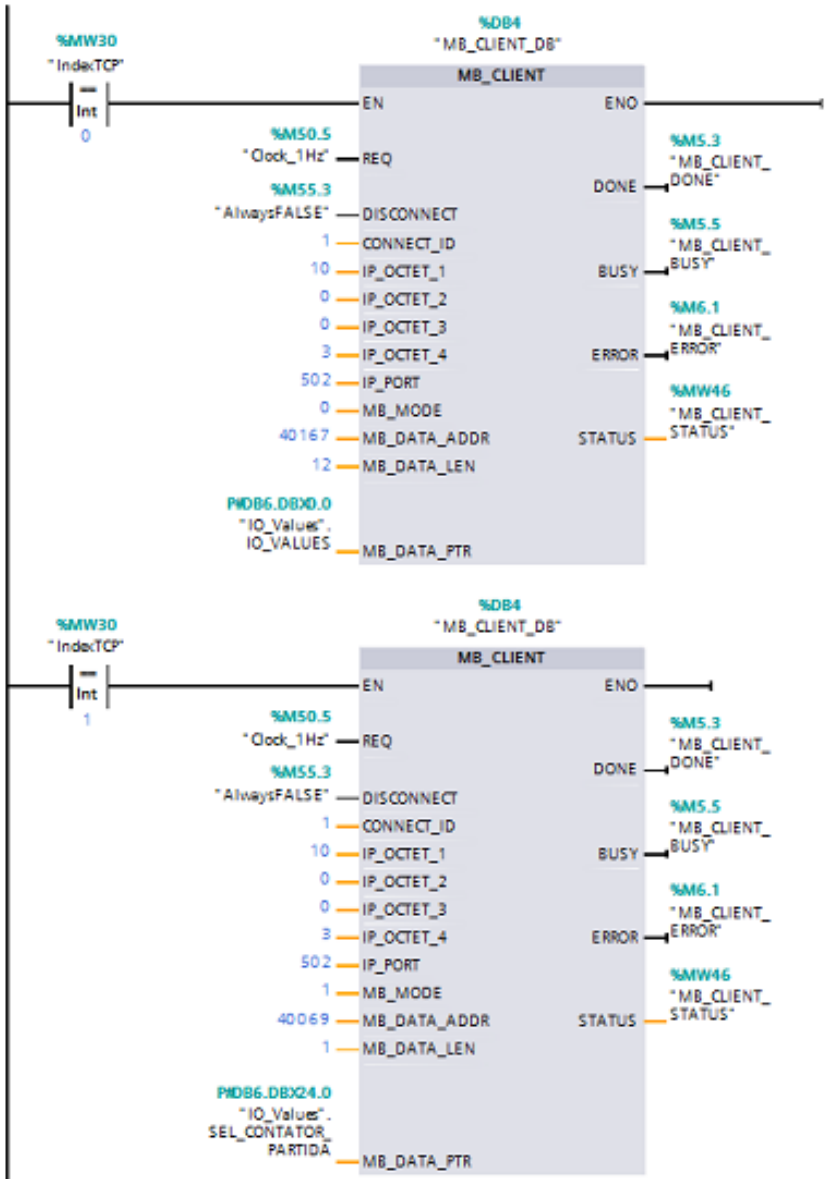
Figura 44 – Implementação para alternar chamadas de leitura e escrita Modbus TCP.



Nessa estratégia, são geradas as variáveis de controle para a chamada do bloco MB_CLIENT. A variável “IndexTCP” sofre um acréscimo de 1 quando uma requisição é concluída (seja com sucesso, seja com falha) ou quando o tempo de 2 segundos for esgotado. A cada valor de “IndexTCP”, habilita a execução de um bloco MB_CLIENT distinto. Seu valor varia até 3, a quantidade total de blocos MB_CLIENT usados para leituras e escritas. A variável “TriggerTCP” gera um pulso atrasado um ciclo após uma alteração de “IndexTCP”.

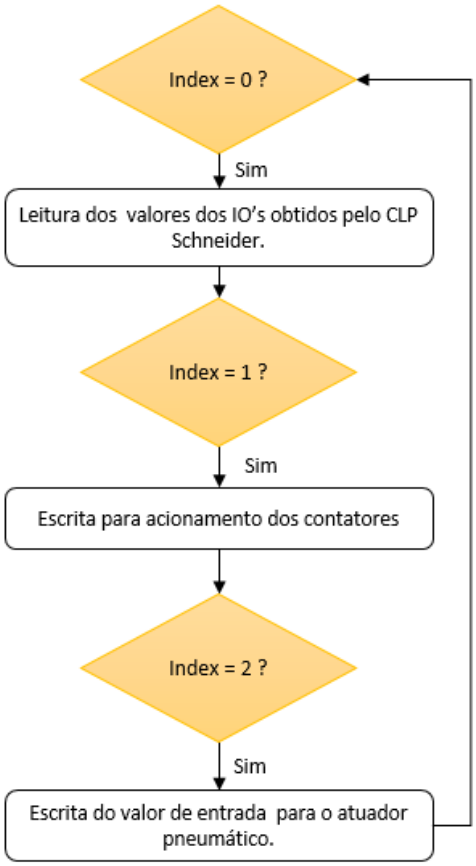
Para a execução de um bloco MB_CLIENT, como pode ser visto na Figura 45, é feita uma comparação de “IndexTCP” com um valor único. Os parâmetros do bloco MB_CLIENT são adaptados de acordo com cada tarefa.

Figura 45 – Sequência de algumas leituras e escritas Modbus TCP.



A Figura 46 mostra um esquemático que sintetiza o fluxo da rede Modbus TCP de acordo com a estratégia da implementação desenvolvida.

Figura 46 – Fluxograma da sequência de leituras e escritas da comunicação Modbus TCP.



3.2.1.3 Rede Modbus RTU

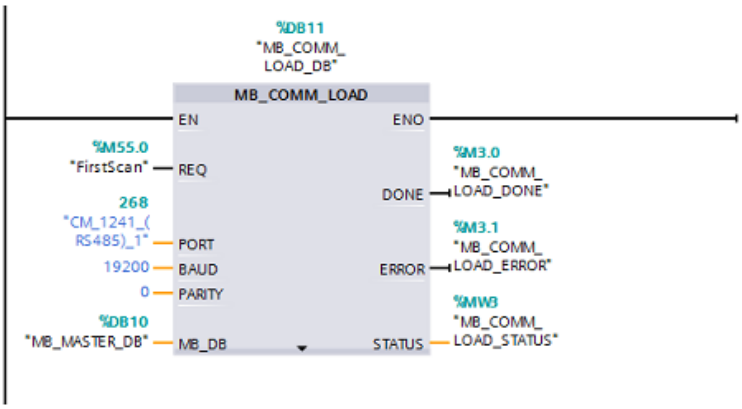
Nesta seção será descrito a estratégia de controle desenvolvida para parte da rede Modbus RTU.

A comunicação Modbus RTU no ambiente TIA PORTAL é configurada pelo bloco Modbus_Comm_Load. Nele, os seguintes parâmetros são ajustados:

- **REQ**: taxa de requisições;
- **BAUD**: a velocidade da rede, em bit/s.
- **PARITY**: tipo de paridade

Na Figura 47, pode ser visto os valores configurados para esse trabalho.

Figura 47 – Parametrização da comunicação Modbus RTU.

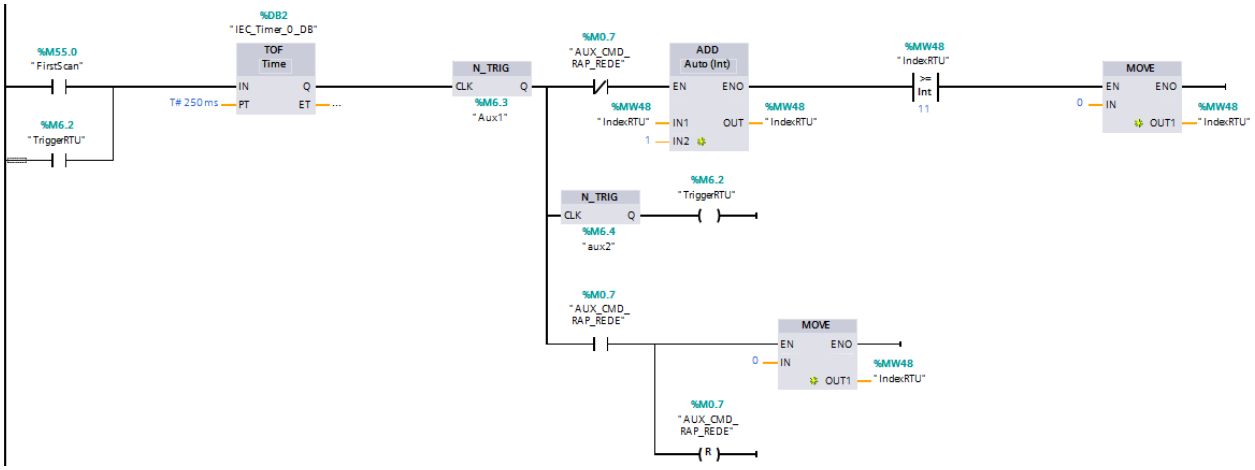


Além do bloco Modbus_Comm_Load, é usado o bloco Modbus_Master para que atue como um mestre na rede Modbus RTU.

Para parametrizar o inversor, é necessário fazer diversos acessos a esse escravo, ou seja, para definir uma velocidade é feita uma escrita no registro 8502 e para obter o status do motor é feita uma leitura no registro 3206.

Dessa forma, foram utilizados múltiplos blocos Modbus_Master, onde cada um foi responsável por uma leitura ou escrita. E associado a isso, foi desenvolvida uma estratégia onde cada leitura ou escrita fosse executada por vez. A implementação dessa estratégia pode ser vista na Figura 48.

Figura 48 – Implementação para alternar chamadas de leitura e escrita Modbus RTU.

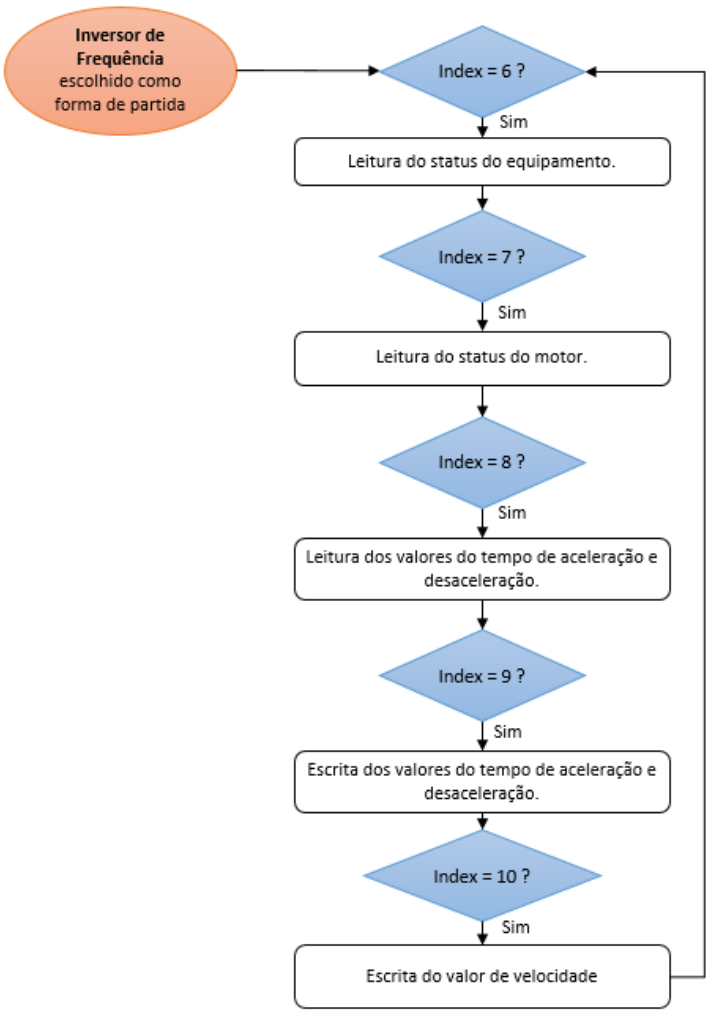


Por essa estratégia, são geradas as variáveis de controle para a chamada do bloco Modbus_Master. A variável “IndexRTU” sofre um acréscimo de 1 a cada 500 ms. A cada valor de “IndexRTU”, habilita a execução de um bloco Modbus_Master distinto. Seu valor varia até 15, a quantidade total de blocos Modbus_Master usados para leituras e escritas. A variável “TriggerRTU” gera um pulso atrasado um ciclo após uma alteração de “IndexRTU”.

Para a execução de cada bloco Modbus_Master, é feita uma comparação de “IndexRTU” com um valor único. Os parâmetros do bloco são adaptados de acordo com cada tarefa.

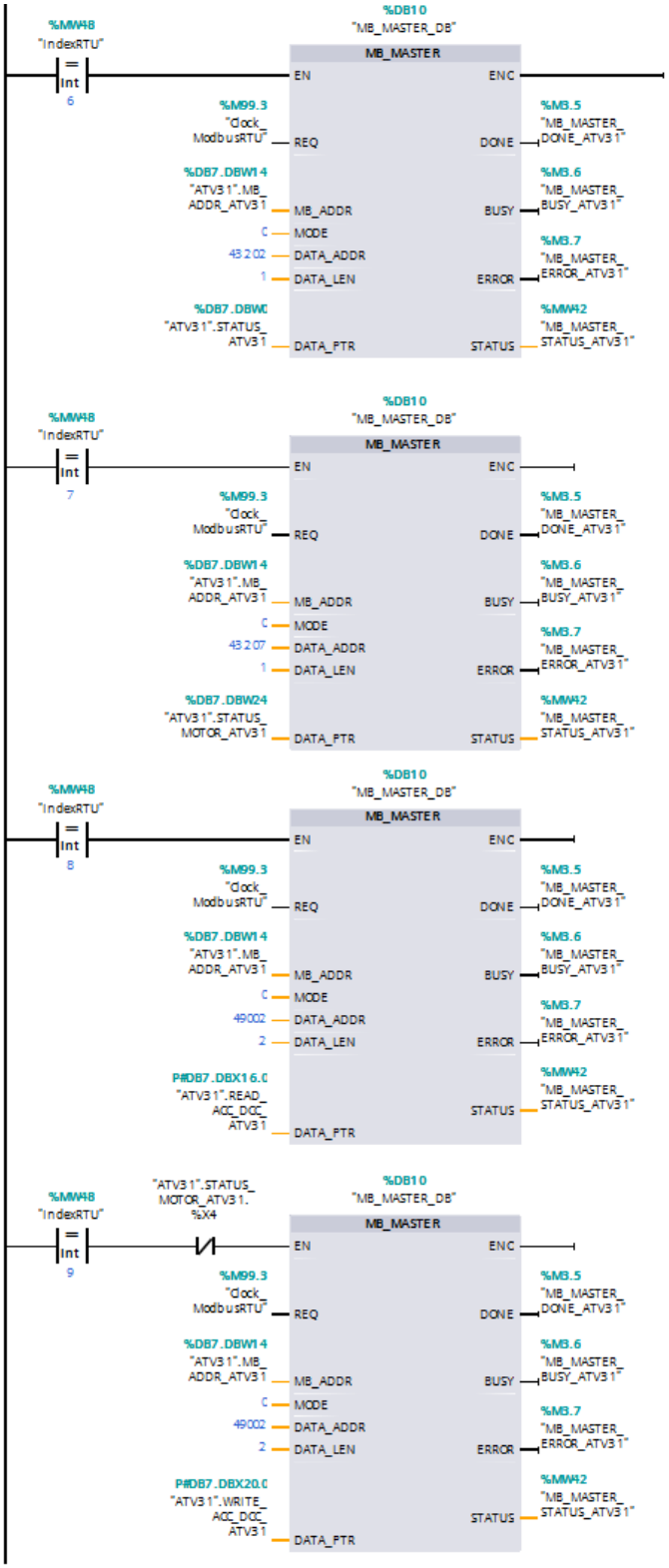
Na Figura 49, pode ser visto um esquema representando as leituras e escritas feitas para realizar a parametrização do inversor.

Figura 49 – Fluxograma da sequência de leituras e escritas da comunicação Modbus RTU.



Na Figura 50 estão a sequência de alguns dos blocos Modbus_Master utilizados.

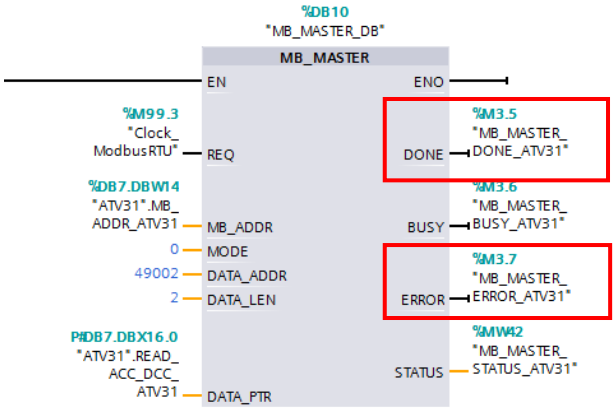
Figura 50 – Sequência de alguns blocos utilizados para leitura e escrita Modbus RTU.



Ainda, foi desenvolvida uma lógica para analisar a comunicação Modbus RTU e diagnosticar se há falha de comunicação com o inversor de frequência.

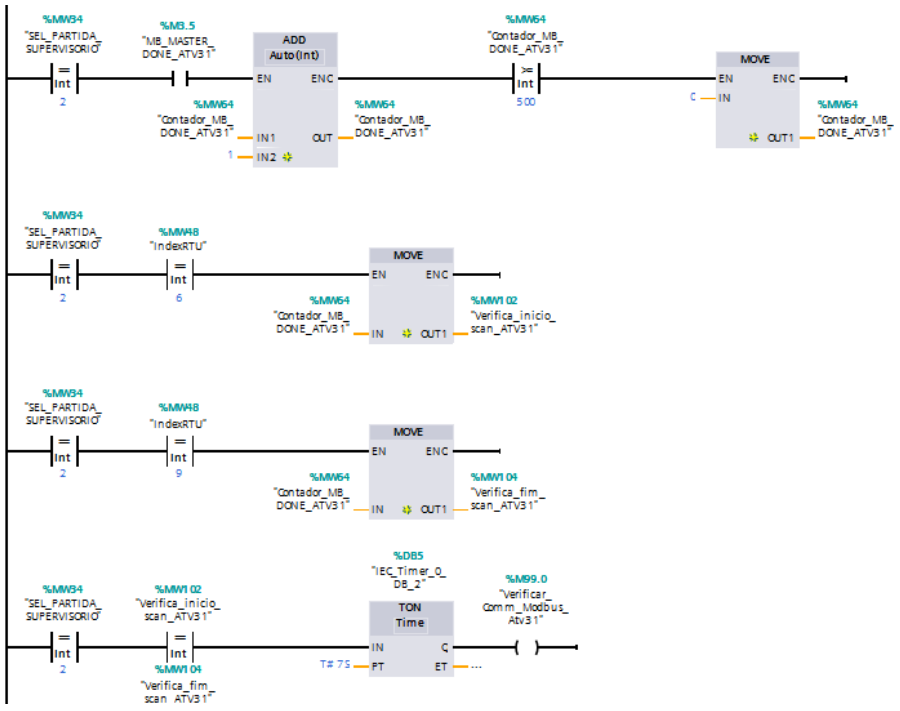
Essa lógica pode ser vista na Figura 52, onde são analisadas as saídas do bloco MB_CLIENT, *Done* e *Error* (Figura 51).

Figura 51 – Saídas de monitoramento do bloco MB_CLIENT.



Em um ciclo, uma requisição pode ser completada com sucesso, nesse caso a saída *Done* fica em *True*. Quando uma requisição não é completada, a saída *Error* fica em *True*.

Figura 52 – Estratégia de verificação de erro desenvolvida.



A cada requisição completada com sucesso é incrementado mais um no contador. No início do ciclo da rede RTU para o inversor, quando IndexRTU é igual a 6, e no fim do ciclo da rede RTU, quando IndexRTU é igual a 9, é armazenado o valor atual do contador. Se esses valores permanecerem o mesmo por mais de 7 segundos, significa que não houve mais requisições completadas com sucesso, indicando um erro na comunicação.

Dessa forma, quando há alguma inconsistência e não estiver mais ocorrendo troca de informações entre o CLP Siemens e o inversor de frequência, é enviada um valor para o supervisório e exibida uma mensagem de alerta para o usuário.

3.2.2 Sistema de Supervisão

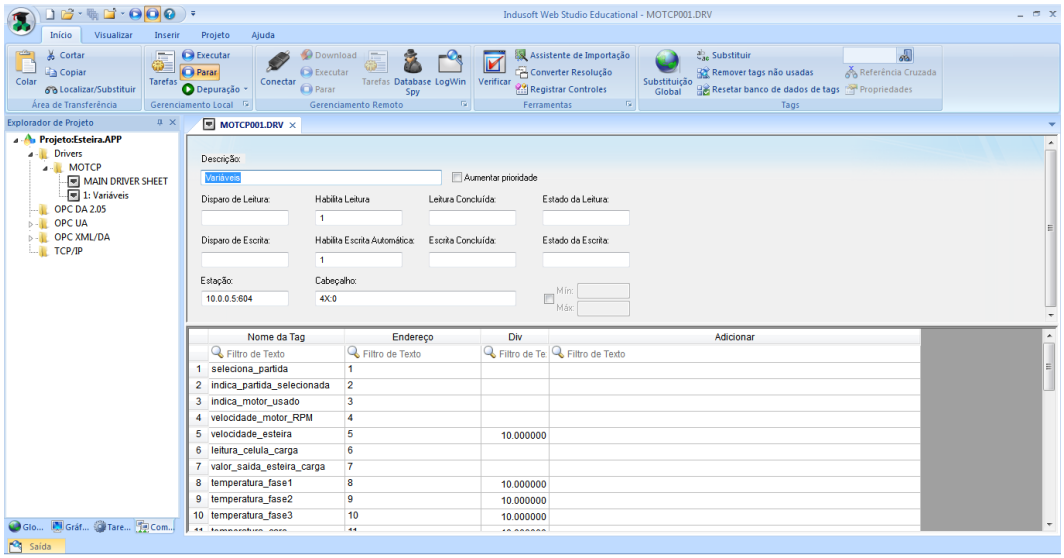
Nesta seção será mostrado o sistema de supervisão, responsável pelo monitoramento do processo e de todas as etapas do sistema de esteira de cara em tempo real.

O sistema supervisório foi desenvolvido através do software *InduSoft Web Studio*, que possui uma coleção de ferramentas de automação que possibilita o desenvolvimento de aplicações IHM e SCADA.

Como mostrado na Figura 41, o supervisório troca dados com o CLP Siemens via Modbus TCP/IP, sendo que nesse contexto, o supervisório é o mestre e CLP Siemens é o escravo. Isso ocorre por meio do driver de comunicação MOTCP, que implementa a protocolo Modbus. Na Figura 53, pode ser vista a tela de configuração do driver.

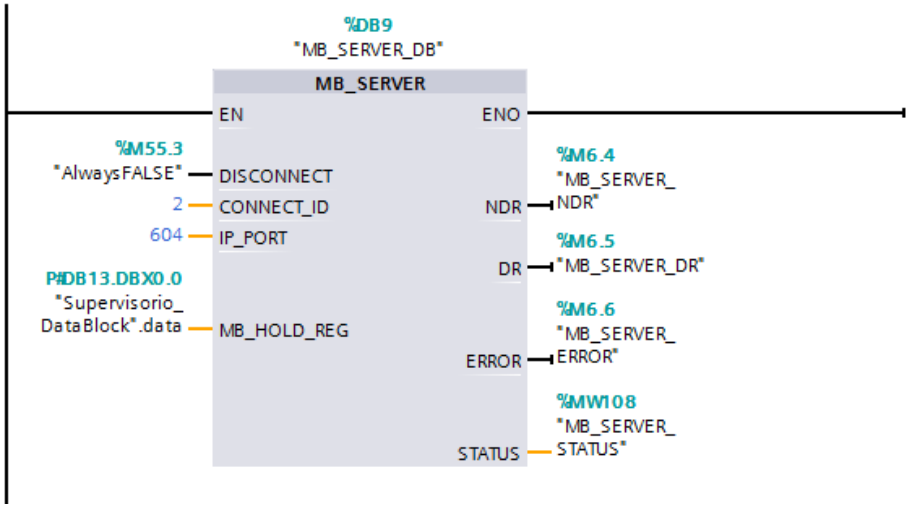
Na parte superior dessa tela estão os campos de onde são definidos o endereço do escravo, a porta de comunicação, os modos de leitura e a faixa de endereços. Na parte de baixo está o mapeamento de variáveis e seus respectivos endereços.

Figura 53 – Tela de configuração do driver de comunicação Modbus TCP



No lado do CLP Siemens, para a comunicação foi necessário utilizar o bloco MB_SERVER, mostrado na Figura 54, que estabelece as configurações do escravo.

Figura 54 – Configuração usada no bloco MB_SERVER.

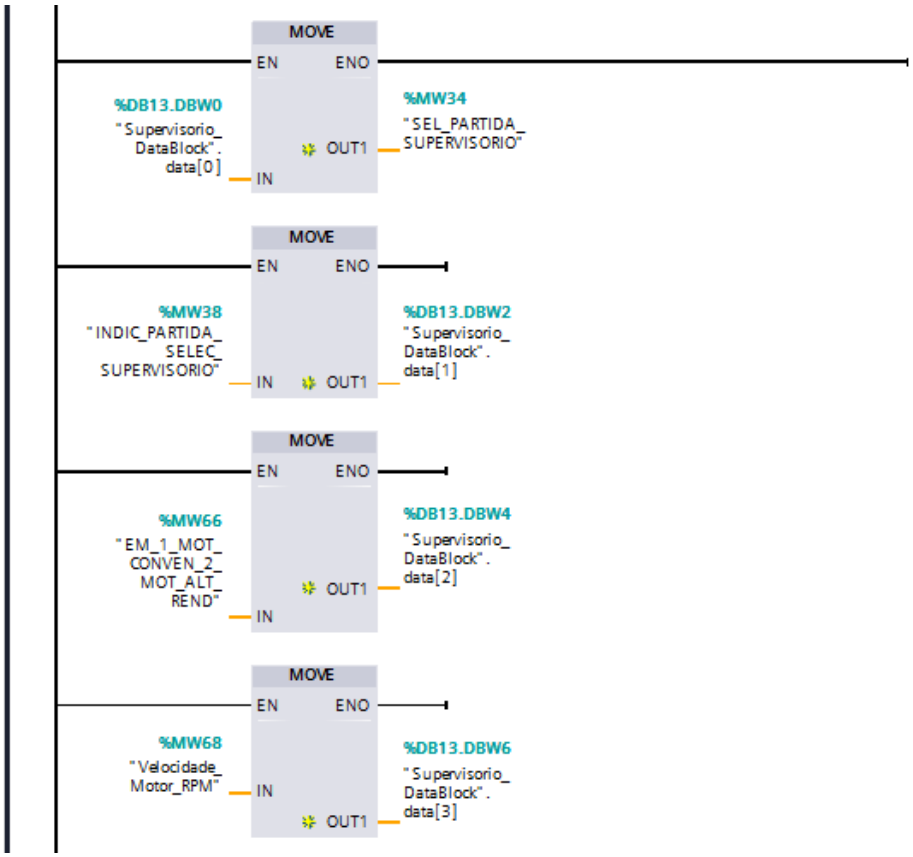


No bloco MB_SERVER é definido o ID do escravo, a porta de comunicação e o *datablock* que conterá os registros *holding register* a serem acessados pelo mestre, além de possuir as variáveis de monitoramento: *Error*, Status, NR (fica em '*True*' a cada escrita do mestre) e NDR (fica em '*True*' a cada leitura do mestre).

A Figura 55, mostra alguns valores do CLP sendo passados para um *datablock*, que é

uma área de memória onde pode ser definido qualquer tipo de dado. Para essa aplicação, no *datablock* 'Supervisorio_DataBlock' foi criado um vetor 'data', com 30 posições, em que cada uma poderia receber um valor. Dessa forma, o protocolo Modbus entende que esses são os registros a serem disponibilizados para a troca com o supervisório.

Figura 55 – Parte das variáveis sendo atribuídas a uma posição do *datablock*.



Com a comunicação estabelecida é possível elaborar a interface para monitorar e controlar as variáveis e os dispositivos do processo.

Na Figura 56 pode ser vista a tela inicial do sistema supervisório desenvolvido. Nessa tela, o usuário pode selecionar qual o elemento da partida do motor. Ele pode escolher entre a partida pelo *soft starter* ou pelo inversor de frequência através do pressionamento dos botões na tela principal. Para facilitar a explicação deste trabalho será utilizado o inversor de frequência. Porém a partida do *soft starter* também está funcionando.

Figura 56 – Página inicial do sistema supervisório.

Ao selecionar a partida, o usuário é direcionado a tela mostrada na Figura 57.

Figura 57 – Tela de comandos do sistema supervisório.

Por essa tela, é possível acompanhar valores lidos das entradas, como as temperaturas das fases do motor e definir valores para o sistema, como o valor de entrada para o atuador pneumático.

Na parte inferior estão os comandos referentes ao elemento de partida e ao motor. O elemento de partida é acionado por meio dos botões “Ligar”, “Desligar” e para caso de falha, há o botão “Reset”.

Após ligado, o tempo de aceleração, desaceleração e a velocidade do motor podem ser definidas. Quando o motor estiver em movimento, são exibidas as velocidades motor e da esteira.

Além disso, há indicadores de status para informar ao usuário a atual condição do sistema e das redes de comunicação. Na Figura 58, pode ser visto a mensagem exibida na tela no caso de erro.

Figura 58 – Mensagem de erro exibida na tela de comandos do supervisório, indicando a condição do sistema.

3.3 Validação do sistema proposto

Um dos objetivos desse projeto é analisar o desempenho da comunicação tendo o CLP Siemens S71200 como mestre da rede.

Para a análise da performance será analisada o ciclo de *scan* do sistema proposto desenvolvido, por meio de capturas de linha da rede. Essas capturas são feitas utilizando um software, também chamado de *sniffer*, que permite detectar o tráfego de mensagens trocadas entre os dispositivos que se comunicam em uma rede Modbus, além dos tempos da rede. Essa análise foi feita para a comunicação Modbus TCP e Modbus RTU em etapas distintas.

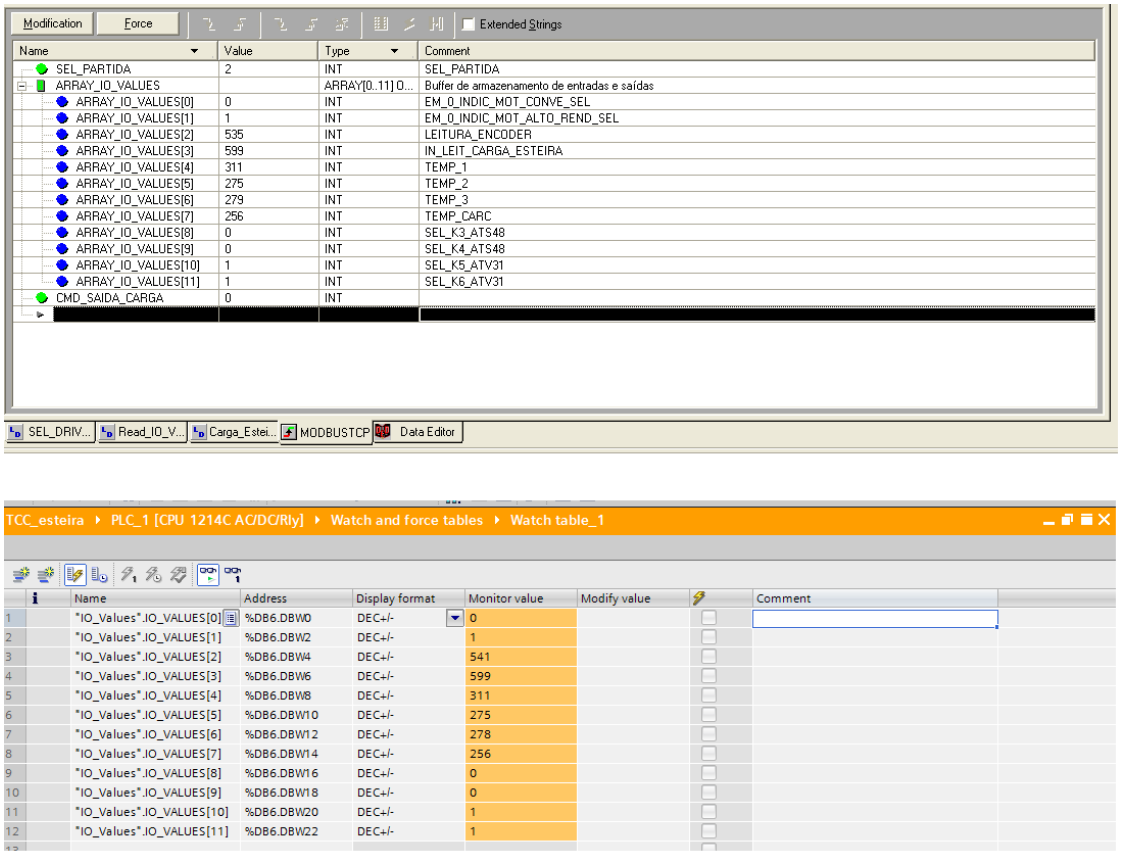
Dessa forma, por essa análise é possível obter informações detalhadas sobre o comportamento do sistema.

4 RESULTADOS OBTIDOS

Nessa seção será mostrado os resultados obtidos sobre os testes realizados e as análises de ciclo de *scan* dessa rede.

A primeira etapa de testes do sistema foi em relação às entradas e saídas, analógicas e digitais. Por meio da comunicação Modbus TCP foi possível obter os valores lidos pelo CLP Schneider. A Figura 59, mostra as capturas de tela das *watch tables* dos dois CLPs em instantes próximos durante o funcionamento do inversor. Nessa figura, na parte superior podem ser vistos os valores lidos no CLP Schneider através das entradas e saídas analógicas e digitais, mapeadas conforme a Tabela 2; e na parte inferior é mostrado esses valores recebidos pelo CLP Siemens.

Figura 59 – Capturas de tela da *watch table* de cada CLP's com os valores dos IO's.



Além disso, foi feito o teste de acionamento do atuador pneumático por meio do CLP Siemens, onde eram enviados valores de entrada variando de 4 a 20 mA. Foram feitas uma série de medições avaliando duas condições distintas, com a esteira em inclinação a 0º e 25º.

As sequências de valores obtidos estão compiladas nas Tabelas 4 e 5.

Tabela 4 – Valores obtidos no teste com o atuador pneumático com 0º de inclinação.

Entrada (mA)	Posicionador (bar)	Célula de Carga
4	1,48	636
4	1,48	636
4	2,23	523
5	2,22	636
6	2,55	644
8	3,77	697
8	4,42	714
8	4,43	713
10	5,57	764

Tabela 5 – Valores obtidos no teste com o atuador pneumático com 25º de inclinação.

Entrada (mA)	Posicionador (bar)	Célula de Carga
5	2,22	920
6	2,95	921
6	4,8	920
10	5,46	920
10	7,48	927
10	7,52	920

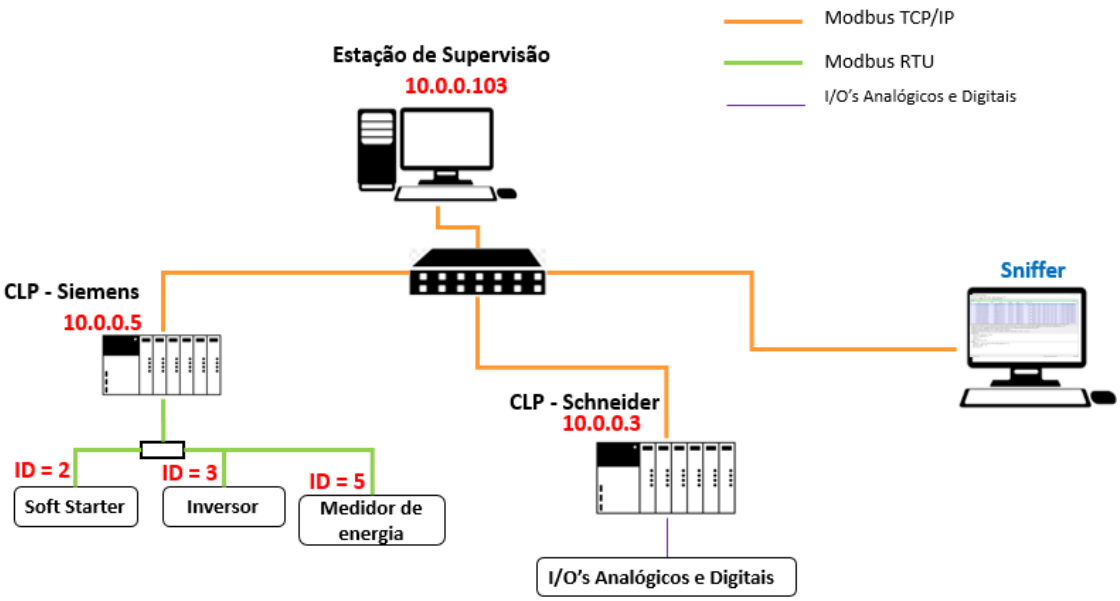
A Tabela 4 mostra que a conforme aumenta o valor de entrada, a pressão segue a tendência de aumento, o que também pode ser observado para o valor da carga aplicada na esteira. Já quando teve inclinação na esteira, aumentou-se o esforço incidente na esteira, porém o valor da carga manteve-se praticamente constante independente da pressão aplicada, como pode ser visto na Tabela 5.

Considerando os mecanismos desenvolvidos, descritos na seção 3.2.1, obteve-se sucesso em colocar a esteira em movimento por meio do inversor de frequência. Dessa forma, como este projeto buscou elaborar um sistema alternativo visando as mesmas funcionalidades do sistema original, foram replicadas as lógicas descritas na seção 3.2.1 para a comunicação com o *soft starter* e o medidor de energia, tendo assim três elementos na rede Modbus RTU.

A seguir será descrito a análise para a comunicação Modbus TCP e Modbus RTU.

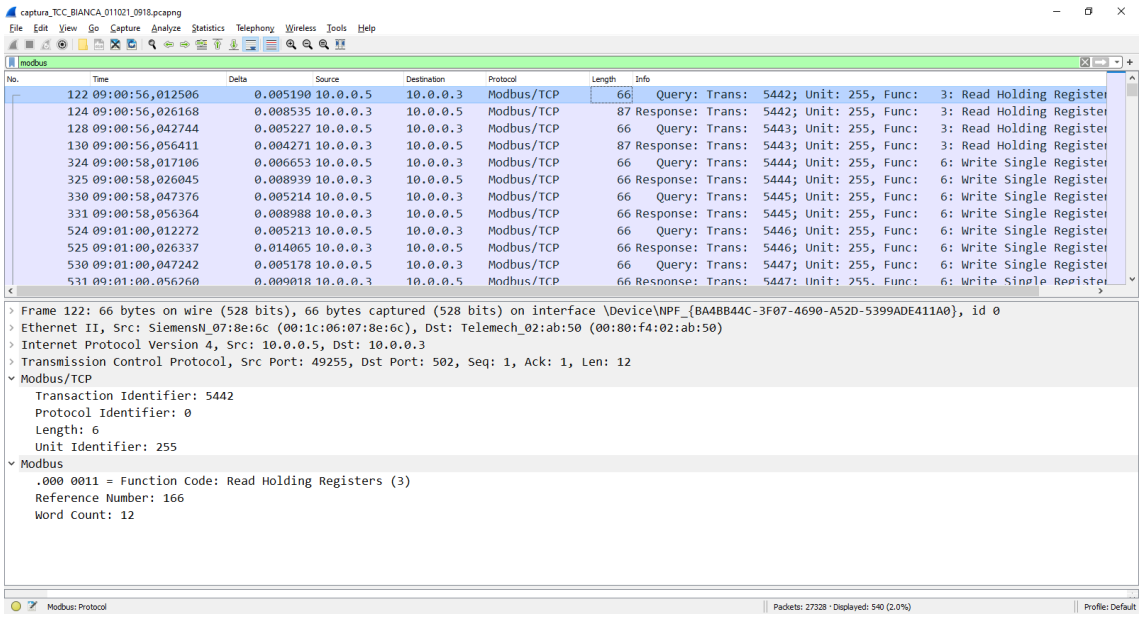
Para a validação da comunicação Modbus TCP foi montado o cenário de teste da Figura 60.

Figura 60 – Estrutura do cenário de teste da comunicação Modbus TCP.



Para analisar a comunicação foi utilizado para o software aberto *Wireshark*. Uma tela da captura é mostrada na figura a seguir.

Figura 61– Captura de tela do software de captura de linha Modbus TCP.



Os dados foram salvos no formato .csv e desta forma foi compilado uma tabela com os dados obtidos conforme mostrado na Tabela 6.

Tabela 6 – Parte da captura de linha da comunicação Modbus TCP.

No.	Time	Source	Destination	Length	Req/Res	Info	MSG
122	09:00:56,013	10.0.0.5	10.0.0.3	66	Query	Read Holding Registers 166 até 178	FF 03 00 A6 00 0C
124	09:00:56,026	10.0.0.3	10.0.0.5	87	Response	Read Holding Registers	FF 03 18 00 00 ...
128	09:00:56,043	10.0.0.5	10.0.0.3	66	Query	Read Holding Registers 166 até 178	FF 03 00 A6 00 0C
130	09:00:56,056	10.0.0.3	10.0.0.5	87	Response	Read Holding Registers	FF 03 18 00 00 ...
324	09:00:58,017	10.0.0.5	10.0.0.3	66	Query	Write Single Register 68	FF 06 00 44 00 00
325	09:00:58,026	10.0.0.3	10.0.0.5	66	Response	Write Single Register	FF 06 00 44 00 00
330	09:00:58,047	10.0.0.5	10.0.0.3	66	Query	Write Single Register 68	FF 06 00 44 00 00
331	09:00:58,056	10.0.0.3	10.0.0.5	66	Response	Write Single Register	FF 06 00 44 00 00
524	09:01:00,012	10.0.0.5	10.0.0.3	66	Query	Write Single Register 180	FF 06 00 B4 00 00
525	09:01:00,026	10.0.0.3	10.0.0.5	66	Response	Write Single Register	FF 06 00 B4 00 00
530	09:01:00,047	10.0.0.5	10.0.0.3	66	Query	Write Single Register 180	FF 06 00 B4 00 00
531	09:01:00,056	10.0.0.3	10.0.0.5	66	Response	Write Single Register	FF 06 00 B4 00 00
724	09:01:02,012	10.0.0.5	10.0.0.3	66	Query	Read Holding Registers 166 até 178	FF 03 00 A6 00 0C
726	09:01:02,026	10.0.0.3	10.0.0.5	87	Response	Read Holding Registers	FF 03 18 00 00 ...
730	09:01:02,047	10.0.0.5	10.0.0.3	66	Query	Read Holding Registers 166 até 178	FF 03 00 A6 00 0C
732	09:01:02,057	10.0.0.3	10.0.0.5	87	Response	Read Holding Registers	FF 03 18 00 00 ...
925	09:01:04,012	10.0.0.5	10.0.0.3	66	Query	Write Single Register 68	FF 06 00 44 00 00
927	09:01:04,026	10.0.0.3	10.0.0.5	66	Response	Write Single Register	FF 06 00 44 00 00
931	09:01:04,052	10.0.0.5	10.0.0.3	66	Query	Write Single Register 68	FF 06 00 44 00 00
935	09:01:04,066	10.0.0.3	10.0.0.5	66	Response	Write Single Register	FF 06 00 44 00 00
1126	09:01:06,017	10.0.0.5	10.0.0.3	66	Query	Write Single Register 180	FF 06 00 B4 00 00
1129	09:01:06,026	10.0.0.3	10.0.0.5	66	Response	Write Single Register	FF 06 00 B4 00 00
1132	09:01:06,047	10.0.0.5	10.0.0.3	66	Query	Write Single Register 180	FF 06 00 B4 00 00
1135	09:01:06,056	10.0.0.3	10.0.0.5	66	Response	Write Single Register	FF 06 00 B4 00 00
1329	09:01:08,012	10.0.0.5	10.0.0.3	66	Query	Read Holding Registers 166 até 178	FF 03 00 A6 00 0C
1331	09:01:08,016	10.0.0.3	10.0.0.5	87	Response	Read Holding Registers	FF 03 18 00 00 ...
1335	09:01:08,037	10.0.0.5	10.0.0.3	66	Query	Read Holding Registers 166 até 178	FF 03 00 A6 00 0C
1336	09:01:08,047	10.0.0.3	10.0.0.5	87	Response	Read Holding Registers	FF 03 18 00 00 ...

De acordo com a tabela acima, a coluna "*Timestamp*" mostra o tempo capturado pelo *software*. A coluna "*Source*" e "*Destination*" mostra os endereços IPs da origem e destino da mensagem. Neste caso é possível notar somente a comunicação entre o CLP S71200 e o CLP Schneider. A coluna "Len" representa o tamanho da mensagem. A coluna "REQ/RES" mostra se a mensagem é de pergunta do mestre (*Query*) ou a resposta do escravo (*Response*). A coluna

“Func” e a coluna “Info” representam o comando Modbus da mensagem. Baseado nestas informações foi criado a coluna “Latência” o tempo calculado entre a pergunta do mestre e a resposta do escravo, e “Delta REQ-REQ” o tempo entre a pergunta atual e a anterior.

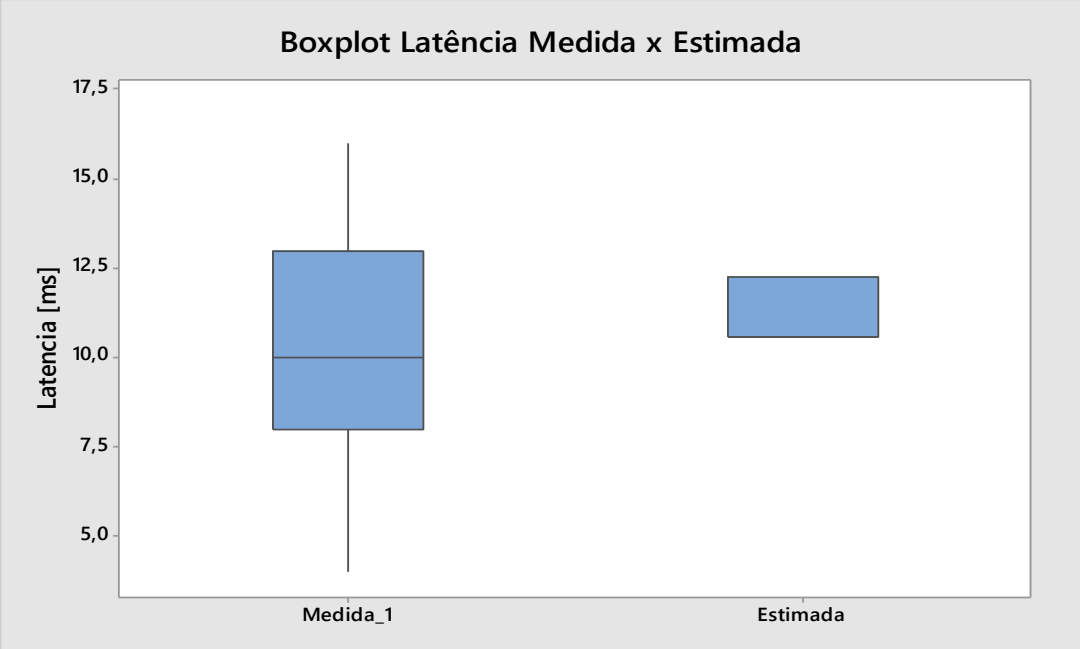
Para validação do sistema de análise foi feita uma comparação dos valores medidos e estimados da rede. Como o Modbus TCP utiliza a rede ethernet a 100Mbps, o tempo de byte considerado foi:

$$T_{Byte} = T_{Bit} * 8 = \frac{1}{100000} * 8 = 8e^{-5} \quad (3)$$

Desta forma, a latência da mensagem seria a quantidade de bytes multiplicada pelo T_{byte}.

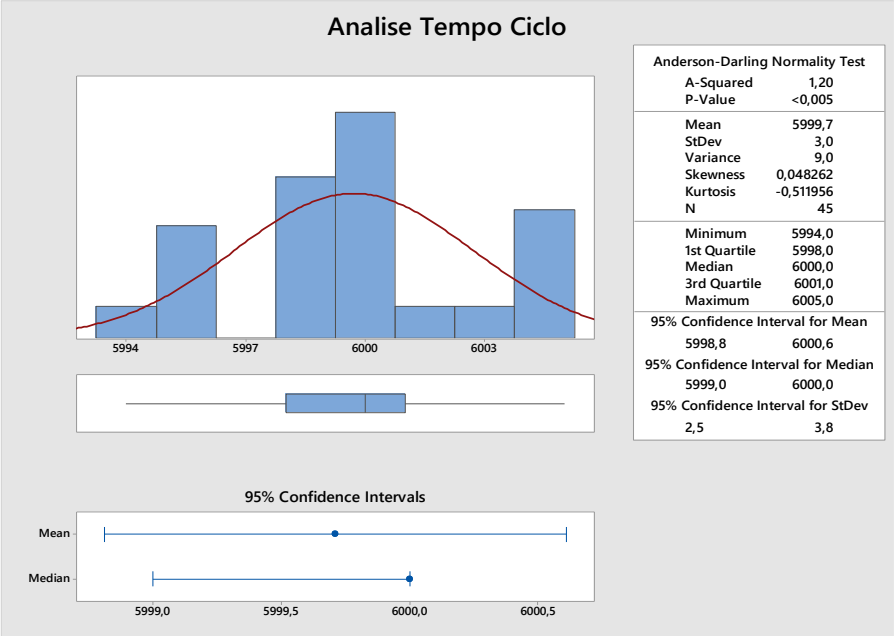
O gráfico *boxsplot* da Figura 62, mostra uma comparação entre os tempos estimado e medido. O valor estimado tem um valor médio de 11,12 ms com desvio padrão de 0,793. Enquanto que os valores medidos têm média de 10,26 ms com desvio padrão de 3,04. Com isso, é possível notar que o valor medido foi muito próximo do estimado, o que validaria a nossa análise com a ferramenta utilizada.

Figura 62 – Gráfico da latência medida e estimada da comunicação Modbus TCP.



Foi determinado o tempo de ciclo para os comandos e o resultado é mostrado no gráfico da Figura 63.

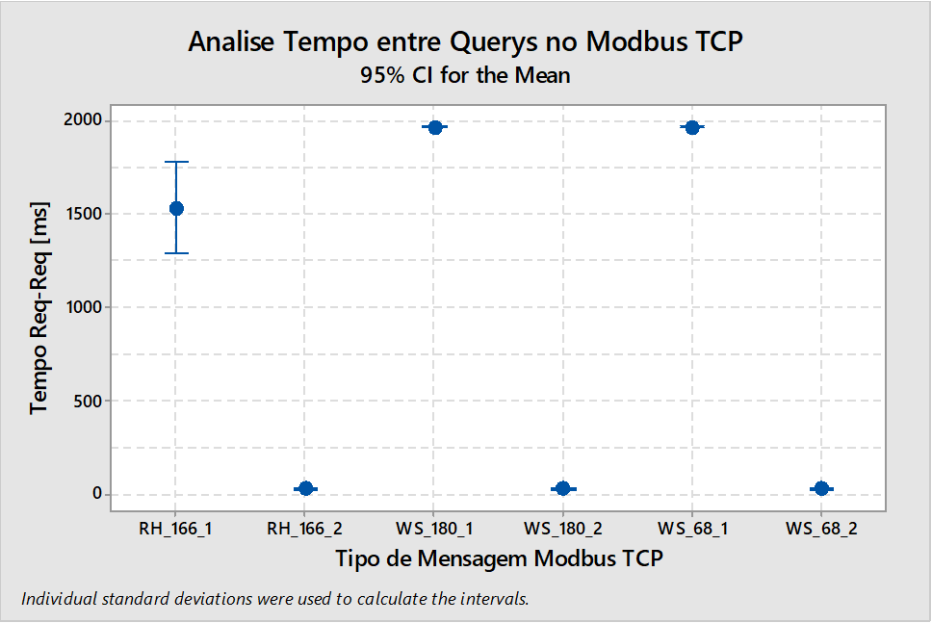
Figura 63 – Gráfico da análise do tempo de ciclo.



Através da análise da figura 63, é possível notar que o tempo de ciclo é de 6 segundos, com pequena variação do desvio padrão. Porém, esse tempo de ciclo pode ser considerado alto, já que há leitura e escrita de apenas três comandos Modbus TCP.

Analisando o tempo entre os comandos de *request* se tem a análise da Figura 64.

Figura 64 – Análise de tempo entre as requisições Modbus TCP.



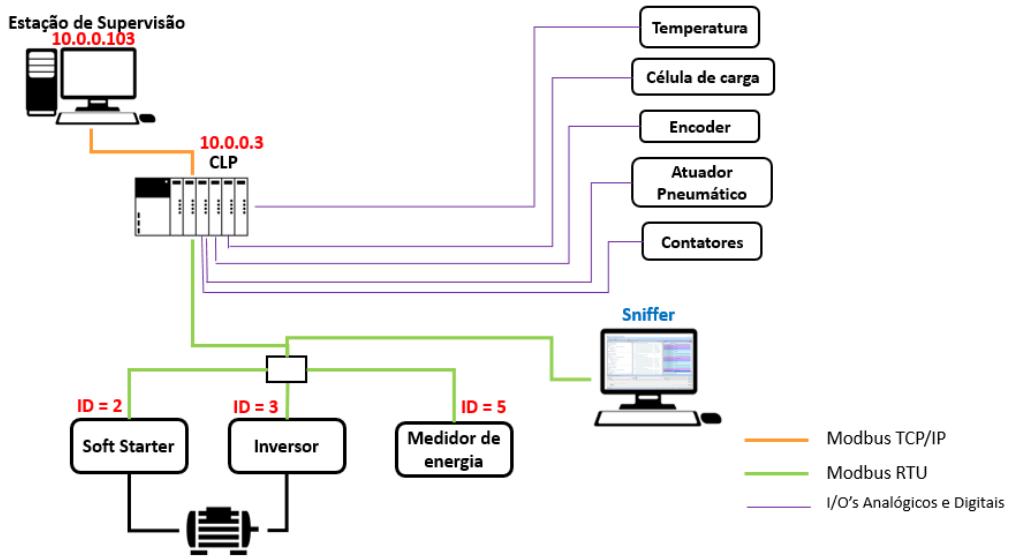
O gráfico da 64 mostra o tempo medido entre comandos do mestre (*requests*). Por esse gráfico é possível observar que em cada ciclo, havia 3 comandos, onde cada um era repetido 2 vezes da seguinte forma:

- RH_166_x era leitura de *Holding Registers* para endereço 166, e então teria o primeiro e segundo comando.
- WS_180_x era escrita em *Single Register* no endereço 180, e então teria o primeiro e segundo comando.
- WS_68_x representava a escrita no registro 68, e então teria o primeiro e segundo comando.

Ainda, a análise do gráfico da Figura 64 mostra que entre o final de um comando e o início do outro havia um tempo de atraso de 1,5s a 2,0s. E então, a repetição do comando duas vezes, onde o segundo comando era sempre rápido na ordem 30ms. Tendo assim, o tempo de 6 segundos do ciclo. Outra observação que pode ser feita é que o tempo era bem sincronizado, já que o desvio padrão foi muito pequeno entre os tempos.

Para a outra etapa da análise da rede, foi montado o cenário da Figura 65, visando a validação da comunicação Modbus RTU.

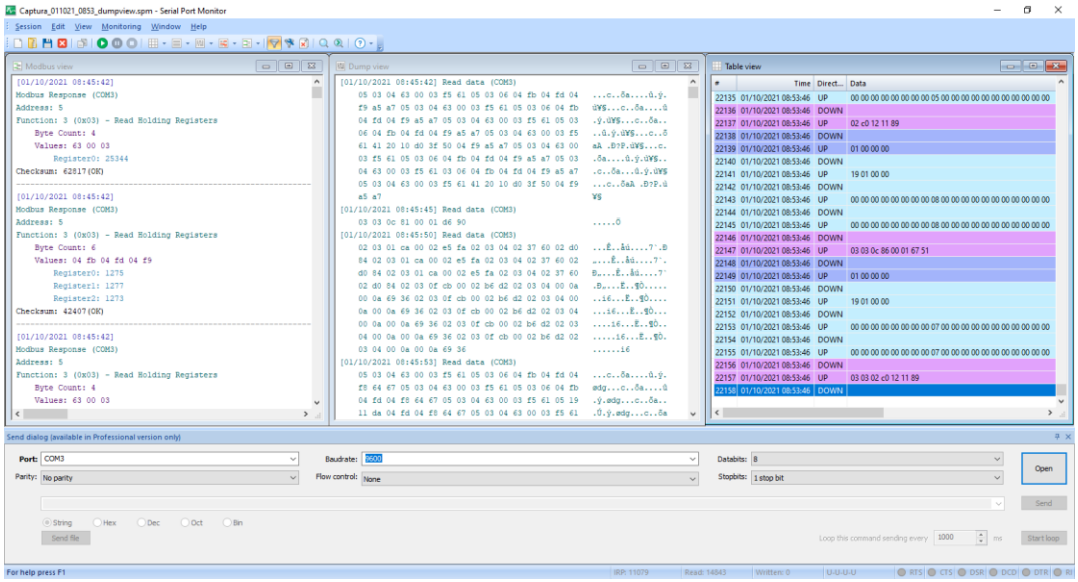
Figura 65 – Estrutura do cenário de teste da comunicação Modbus RTU.



Para analisar a comunicação foi utilizado o software *Serial Port Monitor* da empresa

Eltima Software para a escuta e captura de linha. Uma tela da captura é mostrada na figura a seguir.

Figura 66 – Captura de tela do software de captura de linha Modbus RTU.



Os dados obtidos foram salvos no formato .csv e compilados em uma tabela conforme mostrado na Tabela 7.

Tabela 7 – Parte da captura de linha da comunicação Modbus RTU, com 3 elementos na rede

(continua)

#	Timestamp	Mensagem linha	ID		Comando Modbus	Bytes	Latência REQ-RES	Delta REQ-REQ
7	00:00:00,000	05 03 04 63 00 03 f5 61	5	REQ	Leitura 40464 3 regs	8		
17	00:00:00,015	05 03 06 04 fb 04 fd 04 f9 a5 a7	5	RES		11	15,00	
27	00:00:00,031	05 03 04 63 00 03 f5 61	5	REQ	Leitura 40464 3 regs	8		31,00
37	00:00:00,046	05 03 06 04 fb 04 fd 04 f9 a5 a7	5	RES		11	15,00	
167	00:00:08,002	02 03 01 ca 00 02 e5 fa	2	REQ	Leitura 40459 - 2 regs	8		31,00
187	00:00:08,034	02 03 04 02 37 60 02 d0 84	2			9	32,00	
197	00:00:08,049	02 03 01 ca 00 02 e5 fa	2	REQ	Leitura 40459 2 regs	8		47,00
207	00:00:08,065	02 03 04 02 37 60 02 d0 84	2			9	16,00	
217	00:00:08,174	02 03 0f cb 00 02 b6 d2	2	REQ	Leitura 40459 - 2 regs	8		125,00
807	00:00:35,178	03 03 0c 81 00 01 d6 90	3	REQ	Leitura 43202 - 1 reg	8		4400,00
817	00:00:35,193	03 03 02 82 27 e1 3e	3	RES		7	15,00	
827	00:00:35,224	03 03 0c 81 00 01 d6 90	3	REQ	Leitura 43202 - 1 reg	8		46,00
837	00:00:35,240	03 03 02 82 27 e1 3e	3	RES		7	16,00	
847	00:00:35,256	03 03 0c 81 00 01 d6 90	3	REQ	Leitura 43202 - 1 reg	8		32,00

Tabela 8 – Parte da captura de linha da comunicação Modbus RTU, com 3 elementos na rede

(conclusão)

857	00:00:35,271	03 03 02 82 27 e1 3e	3	RES		7	15,00	
867	00:00:35,365	03 03 0c 86 00 01 67 51	3	REQ	Leitura 43207 - 1 reg	8		109,00
877	00:00:35,380	03 03 02 80 42 20 75	3	RES		7	15,00	
897	00:00:35,412	03 03 0c 86 00 01 67 51	3	REQ	Leitura 43207 - 1 reg	8		47,00
907	00:00:35,427	03 03 02 80 42 20 75	3	RES		7	15,00	
917	00:00:35,443	03 03 0c 86 00 01 67 51	3	REQ	Leitura 43207 - 1 reg	8		31,00
927	00:00:35,458	03 03 02 80 42 20 75	3	RES		7	15,00	
937	00:00:35,568	03 03 0c 86 00 01 67 51	3	REQ	Leitura 43207 - 1 reg	8		125,00
947	00:00:35,583	03 03 02 80 42 20 75	3	RES		7	15,00	
957	00:00:35,599	03 03 0c 86 00 01 67 51	3	REQ	Leitura 43207 - 1 reg	8		31,00

Como pode ser visto nas Tabelas 7 e 8, a coluna "*Timestamp"* mostra o instante de tempo capturado pelo software. A coluna "Mensagem" corresponde ao *frame* de dados Modbus RTU capturado na linha no formato hexadecimal. A coluna "ID" representa o primeiro byte da mensagem. É possível verificar que foram capturados dados dos *devices* 2, 3 e 5, sendo que correspondem ao *soft starter*, inversor de frequência e medidor de energia, respectivamente. A coluna "REQ/RES" mostra se a mensagem é de pergunta do mestre (REQ) ou a resposta do escravo (RES). A coluna "Comando" representa a interpretação do comando Modbus da mensagem, bem como o número de *bytes* da mensagem.

Baseado nestas informações foi criado a coluna Latência REQ-RES, que refere ao tempo calculado entre a pergunta do mestre e a resposta do escravo, e Delta REQ-REQ sendo o tempo entre a pergunta atual e a anterior.

Para validação do sistema de análise foi feita uma comparação dos valores medidos e estimados da rede. Como a comunicação serial foi configurada em 19200/1/*None*, o tempo de *byte* considerado foi:

$$T_{Byte} = T_{Bit} * 10 = \frac{1}{19200} * 10 = 0{,}000521 \quad (3)$$

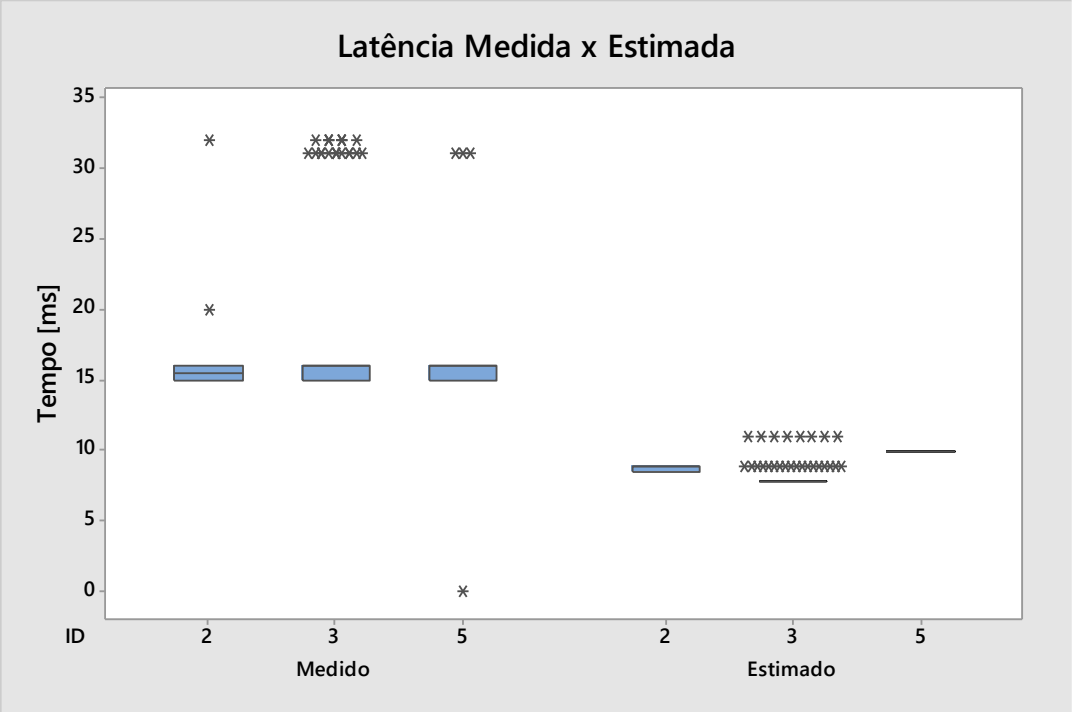
E desta forma, a latência da mensagem seria a quantidade de bytes multiplicada pelo T_{byte}. A Tabela 9 mostra os tempos médios obtidos para cada dispositivo da rede.

Tabela 9 - Tempo médio e estimado para cada dispositivo no cenário de teste.

ID	Tempo Medido Médio (ms)	Tempo Estimado Médio (ms)
2	16,75	8,73
3	17,99	8,12
5	17,47	9,90

A partir dessas informações, foi criado também um gráfico *boxsplot* comparando o tempo estimado e o tempo medido, como pode ser visto na Figura 67.

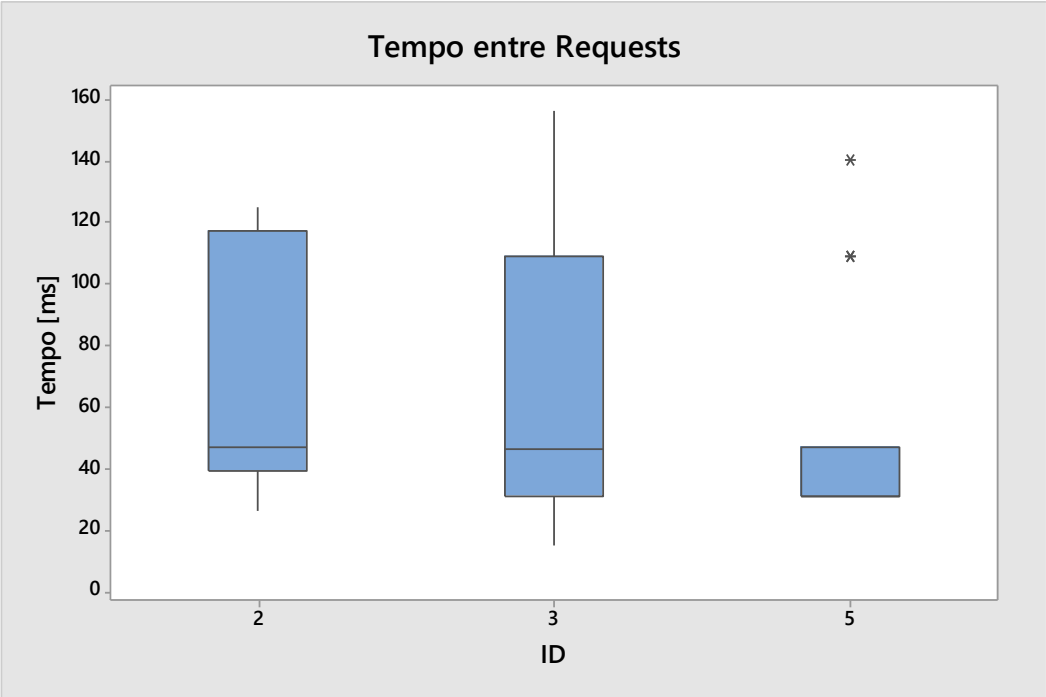
Figura 67 – Comparação do tempo estimado e o tempo medido.



Observando o gráfico da figura anterior, é possível concluir que o valor medido é ligeiramente superior ao valor estimado e que era esperado devido as características do sistema de captura não serem ideais para uma medição com maior acurácia. Porém é possível tirar conclusões com os dados coletados.

O gráfico da Figura 68 mostra o tempo entre *requests*. Ele mostra a rapidez no algoritmo em novas perguntas. Para esse gráfico foi retirado as amostras com tempo maior que 3 segundos pois será explicado na próxima análise.

Figura 68 – Tempo entre *Requests*.



Como pode ser visto na figura anterior, o valor médio é bastante próximo entre os dispositivos. O dispositivo 5 demonstra melhores resultados em relação aos outros dois pois o equipamento com ID igual a 5 possui poucas amostras em relação aos outros.

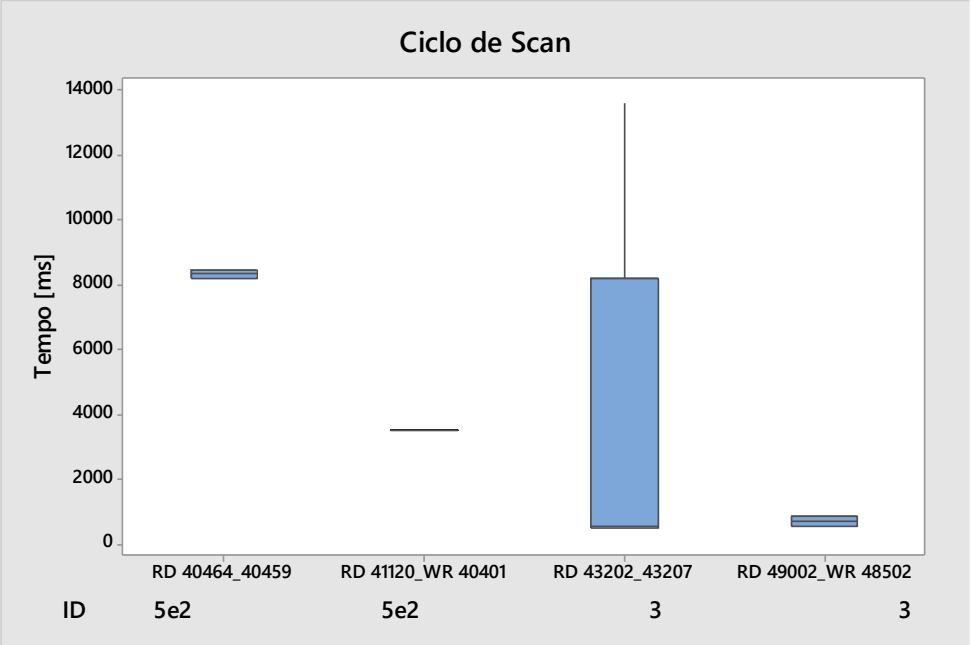
Com relação ao tempo de ciclo, foi obtido os seguintes dados da Tabela 9.

Tabela 10 – Dados do ciclo de *scan* da rede

ID	**Comandos**	**Nº de Comandos**	**Ciclo de *Scan***
5 e 2	RD 40464_40459	15	8439
5e 2	RD 40464_40459	10	8174
5e 2	RD 41120_WR 40401	9	3494
3	RD 43202_43207	8	436
3	RD 43202_43207	8	468
3	RD 49002_WR 48502	13	827
3	RD 43202_43207	17	9251
3	RD 43202_43207	8	452
3	RD 49002_WR 48502	13	546
3	RD 43202_43207	15	4898
3	RD 43202_43207	8	468
3	RD 43202_43207	8	562
3	RD 43202_43207	22	13572

Como pode ser visto na tabela anterior, a coluna ID refere-se ao dispositivo da rede, a coluna Comandos indica quais registros foram utilizados e se foram de leitura (RD) ou escrita (WR). A terceira coluna indica a quantidade de comandos trocados; e a quarta coluna indica o tempo gasto. Desta forma, com base nos dados da Tabela 9, foi gerado um gráfico *boxsplot* dos comandos, que pode ser visto na Figura 69.

Figura 69 – Gráfico do ciclo de *scan*.



Durante os testes, notou-se uma considerável demora para executar comandos, afetando principalmente o momento de colocar ou tirar o motor de movimentação.

Pelas análises de captura de linha, foi possível ver que, com relação as mensagens e ao ciclo de *scan*, o Modbus TCP é mais comportado que quando comparado ao Modbus RTU.

Ainda sobre o Modbus TCP, é notável que o sistema se beneficiaria se houvesse somente um comando de cada por ciclo, e que tentasse escrever em registros próximos, já que assim seria possível fazer a escrita no mesmo comando. Desta forma, o ciclo de *scan* poderia ter apenas dois comandos, um de leitura e outro de escrita, o que poderia resultar em um tempo de ciclo mais baixo, na ordem de centenas de milissegundos.

Já sobre o Modbus RTU, pode-se verificar que mesmo quando nenhum elemento de partida é selecionado, é feita a leitura do de todos elementos da rede. Porém, o sistema parte

com o *soft starter* ligado e o inversor desligado. Neste caso, quando o inversor não está selecionado, há uma espera de aproximadamente 5 segundos, afetando o tempo de *scan* do *soft starter*, como pode ser constatado no gráfico da Figura 60. Quando o inversor (ID = 3) é habilitado, não são feitas mais requisições para o (ID = 5) e o *soft starter* (ID = 2), então existe menos *timeouts* na comunicação.

Além disso, pode-se observar uma quantidade grande de mensagens repetidas dentro de um ciclo que faz com que a supervisão dos pontos não seja uniforme.

5 CONCLUSÃO E TRABALHOS FUTUROS

O objetivo principal deste trabalho foi o estudo e a análise de um sistema de esteira de cargas, para assim buscar uma solução alternativa que lidasse com a questão de equipamentos desatualizados, associado com a adequação do sistema a estrutura disponível no laboratório.

Em relação ao objetivo principal do projeto, foi possível criar uma solução alternativa com um CLP mais moderno e com pequenas mudanças na estrutura atual da planta, porém com uma solução que atendesse melhor o laboratório, onde vários CLPs poderão ser utilizados para controlar a planta. O sistema de esteira de carga foi colocado em funcionamento utilizando o CLP Siemens como controlador principal, em uma rede Modbus RTU com até três elementos, o inversor de frequência, o *soft starter* e o medidor de energia.

O desenvolvimento do presente trabalho possibilitou aplicar os conhecimentos adquiridos ao longo do curso e obteve-se êxito em todas as fases propostas, desde o entendimento da solução atual, através da analise do diagrama elétrico, estudo dos equipamentos, estudo da lógica antiga do CLP da Schneider, além da parte de desenvolvimento de uma solução para o problema, e do desenvolvimento da lógica de controle da planta. E por fim, dos testes de validação de cada uma das partes e do sistema como um todo.

Com relação aos testes finais, foi verificado que a esteira tinha um longo tempo de resposta, que não era sentido quando estava usando somente o CLP original. Desta forma, foi feita uma análise usando a ferramenta de escuta de linha onde foi possível analisar as particularidades do sistema. Por essas análises foi possível concluir que para o devido funcionamento da estratégia proposta deve-se ajustar o comparador limite do Index de acordo com o número de requisições necessárias, para evitar que durante o ciclo de *scan* haja instantes de inatividade, fazendo com que aumente o tempo de *scan*, tornando assim o sistema mais lento. Outro ponto que pode afetar o desempenho é em relação a grande quantidade de requisições repetidas dentro de um ciclo.

Em suma, com este trabalho foi possível obter um conhecimento mais aprofundado sobre elementos de partida de motores e de outros equipamentos elétricos, além de promover um estudo mais detalhado sobre as características do protocolo Modbus. E assim, viabilizando um sistema funcional para o laboratório.

Para estudos futuros, primeiramente teria de ser melhorado os tempos da comunicação Modbus. Aliado a isso poderia ser colocado endereços próximos no mesmo comando, e também desabilitar os comandos de um determinado equipamento quando o outro não for selecionado.

Outra melhoria no sistema seria trabalhar no desenvolvimento um sistema de controle que regule a velocidade da esteira de acordo com a carga aplicada. Além disso, poderia ser feito o estudo de uma rede equivalente a original, adicionando também o dispositivo de partida direta, ou ainda, comparar com outras estratégias de implementação da comunicação Modbus e explocar soluções que envolvam outros protocolos e fabricantes.

REFERÊNCIAS

CHAPMAN, S. J. *Fundamentos de máquinas elétricas*. Tradução: Anatólio Laschuk. – 5. ed. Porto Alegre: AMGH, 2013. Citado na página 47.

DEY, C; SEN S. K. *Industrial Automation Technologies*. CRC Press, 2020. Citado na página 21.

FRANCHI, C. M. *Inversores de Frequência: Teoria e Aplicações*. 2. Ed. São Paulo: Editora Érica, 2013. Citado na página 45.

GARCIA JR, E. *Introdução a Sistemas de Supervisão, Controle e Aquisição de Dados: SCADA*. Rio de Janeiro: Alta Books. 2019. Citado na página 47.

GEORGINI, J. M. *Automação Aplicada - Descrição e implementação de Sistemas Sequenciais com PLCs*. 9. ed. Editora Érica. 2018. Citado na página 41.

LOGITEC SISTEMAS. *Esteira Transportadora Automatizada*. [S.l.] Disponível em: http://logitecsistemas.com.br/informacoes/esteira-transportadora-automatizada. Acesso em 26 de jan. 2021. Citado na página 19.

MAGALHÃES, F. R. P. de. *Análise de eficiência energética para técnicas de Acionamento de correias transportadoras.* Tese (Mestrado) – Universidade Federal do Ceará, 2010. Citado na página 20.

MEHTA, B. R.; REDDY, Y. J. *Industrial Process Automation Systems: Design and Implementation.* Editora Butterworth-Heinemann. 2014. Citado na página 21.

MUNIZ, S. *Investimento recente, capacitação tecnológica e competitividade.* São Paulo: São Paulo Perspec. vol.14 no.3, 2000. Disponível em <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-88392000000300015>. Citado na página 16.

REDUTORES IBR. *Esteiras Rolantes*. [S.l.], 2016. Disponível em: http://www.redutoresibr.com.br/pt/Noticia/esteiras-rolantes. Acesso em 26 de jan. 2021. Citado na página 29.

ROGGIA, L; FUENTES, R. C. *Automação Industrial.* Santa Maria: Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria, Rede e-Tec Brasil, 2016. Citado na página 41.

ROSÁRIO, J. M. *Automação industrial*. [S.l.]: Editora Baraúna, 2009. Citado na página 16.

SANTOS, M. M. et al.; *Indústria 4.0: Fundamentos, Perspectivas e Aplicações*. São Paulo: Érica, 2018. Citado na página 16.

SEIDL, J. et al. *Segurança de Automação Industrial e SCADA*. [S.l.]: Elsevier Brasil, 2014. Citado na página 20.

SEN, S. K. *Fieldbus and Networking in Process Automation.* CRC Press, 2017. Citado na página 20.

SILVA, E. A. *Introdução às linguagens de programação para CLP*. São Paulo: Editora Blucher, 2016. Citado na página 36.

SILVA, S. E; GONÇALVES, C. A. *O que é inovação tecnológica: seu papel transformador nas empresas e nos mercados.* 1. ed. Curitiba: Editora Appris, 2018. Citado na página 16.

TECHLIB. *PCMCIA*. [S.l.]. Disponível em: https://techlib.wiki/definition/pcmcia.html. Acesso em 18 de fev. 2021. Citado na página 55.

WILAMOWSKI, B. M; IRWIN J. D. *Industrial Communication Systems.* 1. ed. CRC Press, 2017. Citado na página 27.

ZURAWSKI, R. *Industrial Communication Technology Handbook*. 2. ed. CRC Press, 2017.Citado na página 26.