

REDE DOCTUM DE ENSINO
FACULDADE INTEGRADAS DE CARATINGA
CURSO SUPERIOR DE ENGENHARIA ELÉTRICA

**DESENVOLVIMENTO DE SISTEMA SUPERVISÓRIO DIDÁTICO PARA CONTROLE
DE INVERSOR DE FREQUÊNCIA ACIONANDO MOTOR DE INDUÇÃO TRIFÁSICO**

ARTHUR SOUZA DE MORAES

Trabalho de Conclusão de Curso

Caratinga/MG

2016

ARTHUR SOUZA DE MORAES

DESENVOLVIMENTO DE SISTEMA SUPERVISÓRIO DIDÁTICO PARA CONTROLE DE INVERSOR DE FREQUÊNCIA ACIONANDO MOTOR DE INDUÇÃO TRIFÁSICO

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso Superior de Engenharia Elétrica das Faculdades Integradas de Caratinga da DOCTUM Caratinga como requisito parcial para obtenção do Grau de Bacharel em Engenharia Elétrica.

Professor Orientador: Vinicius Murilo Lima Rodrigues.

Caratinga/MG

2016

TERMO DE APROVAÇÃO

TÍTULO DO TRABALHO

DESENVOLVIMENTO DE SISTEMA SUPERVISÓRIO DIDÁTICO PARA CONTROLE DE INVERSOR DE FREQUÊNCIA ACIONANDO MOTOR DE INDUÇÃO TRIFÁSICO

Nome completo do aluno: ARTHUR SOUZA DE MORAES

Este Trabalho de Conclusão de Curso foi apresentado perante a Banca de Avaliação composta pelos professores Daniel Mageste Butters, Joildo Fernandes Costa Junior e Vinicius Murilo Lima Rodrigues, às 21:40 horas do dia 12 de dezembro de 2016, como requisito parcial para a obtenção do título de bacharel em Engenharia Elétrica. Após a avaliação de cada professor e discussão, a Banca Avaliadora considerou o trabalho: APROVADO (aprovado ou não aprovado), com a qualificação: ÓTIMO (Excelente, Ótima, Bom, Satisfatório ou Insatisfatório).

Trabalho indicado para publicação: () SIM (X) NÃO

Caratinga, 12 de dezembro de 2016

Vinicius M. Lima
Professor Orientador e Presidente da Banca

[Assinatura]
Professor Avaliador 1

Daniel M. Butters
Professor Avaliador 2

Arthur Souza Moraes
Aluno(a)

[Assinatura]
Coordenador(a) do Curso

AGRADECIMENTOS

Agradeço a minha família pelo apoio não só durante esta jornada mas durante todas que passaram e todas que virão, aos meus pais Celso e Ivone e a minha irmã Alice Moraes, pois sem eles eu não estaria aqui.

Agradeço a minha noiva Suelen Martins, que vem conseguindo me aturar a 9 incansáveis anos rs, e que me deu tanta força durante esta jornada, me levantando nos momentos em que estava no chão (e não foram poucos) com suas palavras de apoio e com seu jeito meigo e carinhoso de ser, te amo.

Agradeço aos amigos que tive o prazer de conhecer durante estes anos que passei em Caratinga e dos quais dividi os cansaços e apertos da vida acadêmica, as frustrações e os prazeres de uma cerveja gelada.

Agradeço o Prof. Vinicius Murilo, que foi meu orientador e sem ele esse projeto não existiria.

Agradeço aos demais professores, que também tive o prazer de conhecer e pelo quais sou grato pela disposição que tiveram em compartilhar deste mundo do saber.

“Haja hoje para tanto ontem. E amanhã para tanto hoje. Sobretudo isso.”
(Paulo Leminski)

SOUZA DE MORAES, Arthur. **Desenvolvimento de Sistema Supervisório Didático para Controle de Inversor de Frequência acionando Motor de Indução Trifásico**. Caratinga, 2016. 62 f. Trabalho de Conclusão de Curso Superior de Engenharia Elétrica - Curso de Engenharia Elétrica. Faculdades Integradas de Caratinga, Rede DOCTUM, Caratinga, 2016.

RESUMO

Este trabalho apresenta, dentro de um contexto acadêmico, as funcionalidades e a importância dos sistemas supervisórios de controle e aquisição de dados, também conhecido como sistema SCADA, tanto para o setor industrial moderno quanto para as áreas do ensino da engenharia. A proposta do projeto veio a partir dos estudos com inversor de frequência no Laboratório de Máquinas Elétricas do Instituto Tecnológico de Caratinga – ITC Doctum, analisando as limitações da IHM acoplada ao inversor, notou-se que a construção de um supervisório abriria uma gama de opções interessantes para estudo e melhor entendimento dos comportamentos do inversor, do motor, bem como, compreender o funcionamento de um supervisório na prática. O objetivo foi construir um exemplo simplificado para fins didáticos de um controle de processo, através de um supervisório que controla um inversor de frequência que por sua vez aciona um motor de indução trifásico. O projeto proposto permite ainda a geração de gráficos e monitoramento de dados do sistema, isso foi possível por meio de uma interface gráfica criada através do software ScadaBR, instalado em um PC e se comunicando através do protocolo modbus-rtu com o inversor. Os resultados obtidos no projeto demonstraram a eficácia de um sistema supervisório no controle e monitoramento de processos automatizados, sendo este criado de maneira que fosse simples e intuitiva aos usuários.

Palavras-chave: Inversor de Frequência. SCADA. Sistema Supervisório. Motor de Indução Trifásico

SOUZA DE MORAES, Arthur. **Desenvolvimento de Sistema Supervisório Didático para Controle de Inversor de Frequência acionando Motor de Indução Trifásico**. Caratinga, 2016. 62 f. Trabalho de Conclusão de Curso Superior de Engenharia Elétrica - Curso de Engenharia Elétrica. Faculdades Integradas de Caratinga, Rede DOCTUM, Caratinga, 2016.

ABSTRACT

This work presents, within an academic context, how the functionalities and control and data acquisition control systems, also known as SCADA system, both for the industrial sector and for the areas of engineering teaching. The project proposal came from a comparative study with the frequency inverter in the Laboratory of Electrical Machines of the Technological Institute of Caratinga - ITC Doctum, analyzing the limitations of the HMI coupled to the inverter, it was noticed that a construction of a supervisor opened a range of Interesting options to study and better understand the behavior of the inverter, motor, as well as the odor of a practice supervisor. The objective was to construct a simplified example for didactic purposes of a process control, through a supervisor that controls a frequency inverter that in turn drives a three-phase induction motor. The proposed project also allows a generation of graphics and a monitoring of data of the system, this is possible through a graphic interface created by the software. The results obtained without design demonstrate the effectiveness of an uncontrolled supervision system and automated process monitoring, which is created in a way that is simple and intuitive to users.

Key-words: Frequency Inverter. SCADA. Supervisory System. Three Phase Induction Motor.

LISTA DE ILUSTRAÇÕES

Figura 1	– Sala de Monitoramento e Controle.(Fonte: Automation Primer, 2015)	16
Figura 2	– Quadro de Prefixos das Faixas de Dados.(Fonte: National Instruments Corporation, 2014)	21
Figura 3	– Estrutura das Mensagens em Modbus RTU.(Fonte: WEG, 2011)	22
Figura 4	– Diagrama de blocos do processo de obtenção da tensão e frequência em um inversor.(Fonte: WEG, 2016)	24
Figura 5	– Bancada do Inversor CFW09, Laboratório de Máquinas Elétricas/FIC.(Fonte: AUTOR)	27
Figura 6	– Kit WEG Comunicação Serial.(Fonte: WEG, 2011)	28
Figura 7	– Quadro para ajustes de PID.(Fonte: WEG, 2011)	29
Figura 8	– Esquema de MIT.(Fonte: WEG, 2016)	30
Figura 9	– Cabo Conversor (Esquerda) e Cabo Serial (Direita) .(Fonte: AUTOR)	32
Figura 10	– IHM conectada ao módulo de comunicação serial.(Fonte: AUTOR)	33
Figura 11	– Motor de Indução e placa de dados.(Fonte: AUTOR)	34
Figura 12	– Tela de Login do ScadaBR.(Fonte: AUTOR)	36
Figura 13	– Data Source no ScadaBR.(Fonte: AUTOR)	37
Figura 14	– Porta USB-Serial COM4 no PC.(Fonte: AUTOR)	38
Figura 15	– Lista de Data Points no ScadaBR.(Fonte: AUTOR)	43
Figura 16	– Data Point de Status Coil e Data Point de Registrador Holding.(Fonte: AUTOR) .	44
Figura 17	– Watch List do ScadaBR.(Fonte: AUTOR)	45
Figura 18	– Criação da Interface no ScadaBR.(Fonte: AUTOR)	46
Figura 19	– Janelas de configuração do componente Script Para Servidor.(Fonte: AUTOR) .	47
Figura 20	– Janelas de configuração do componente Botão de Escrita.(Fonte: AUTOR) . .	47
Figura 21	– Janelas de configuração do componente GIF Binário.(Fonte: AUTOR)	48
Figura 22	– Janelas de configuração do componente GIF Dinâmico.(Fonte: AUTOR) . . .	48
Figura 23	– Janelas de configuração do componente Data Point Simples.(Fonte: AUTOR) .	48
Figura 24	– Barra rolante de regulação alternativa.(Fonte: AUTOR)	49
Figura 25	– Janela de configuração do componente Gráficos.(Fonte: AUTOR)	50
Figura 26	– Sistema SCADA do projeto.(Fonte: AUTOR)	52
Figura 27	– Interface Gráfica do projeto.(Fonte: AUTOR)	53
Figura 28	– FONTE: AUTOR	61
Figura 29	– FONTE: AUTOR	61
Figura 30	– FONTE: AUTOR	62
Figura 31	– FONTE: AUTOR	62

LISTA DE TABELAS

Tabela 1	– Parâmetros do Motor	34
Tabela 2	– Parâmetros de Comunicação	35
Tabela 3	– Parâmetros do Status Coil	40
Tabela 4	– Parâmetros de Leitura	41
Tabela 5	– Parâmetros de Regulação	41
Tabela 6	– Parâmetros Especiais PID	42

LISTA DE ABREVIATURAS E SIGLAS

A - Corrente

BPS - Bits por Segundo

CLP - Controlador Logico Programavel

CRC - Cycling Redundance Cheking

CV - Cavalo

D - Derivativo

DC - Corrente Continua

DNP3 - Distributed Network Protocol

EIA - Electronic Industries Alliance

GIF - Graphics Interchange Format

HZ - Hertz

I - Intregal

IGBT - Transistor Bipolar de Porta Isolada

IHM - Interação Homem-Maquina

IP - Internet Protocol

LED - Diodo Emissor de Luz

LTDA - Limitada

MIT - Motor de Indução Trifásico

OPC - Oracle Certified Professional

P - Proporcional

PC - Computador Pessoal

PHP - Personal Home Page

PID - Proporcional, integral e derivativo

RJ11 - Registered Jack

RPM - Rotações por Minuto

RS - Recommended Standard

RTU - Remote Unit

SCADA - Sistema de Controle e Aquisição de Dados

TCP - Protocolo de Controle de Transmissão

UDP - User Datagram Protocol

USB - Universal Serial Bus

V - Volt

VVW - voltage Vector Weg

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Contextualização	13
1.2 PROBLEMA E JUSTIFICATIVA	14
1.2.1 OBJETIVOS	14
1.2.1.1 OBJETIVOS GERAL	14
1.2.1.2 OBJETIVOS ESPECIFICOS	14
1.3 Estrutura do Trabalho	15
2 REVISÃO DA LITERATURA	16
2.1 SISTEMAS SUPERVISÓRIOS	16
2.1.1 ScadaBR	17
2.2 PROTOCOLOS DE COMUNICAÇÃO INDUSTRIAL	18
2.2.1 Unidades Físicas de Comunicação Serial	19
2.2.2 Conceito Mestre/Escravo	20
2.2.3 Protocolo Modbus	20
2.2.3.1 Acesso aos Dados do Modbus	20
2.2.3.2 Modbus ASCII	21
2.2.3.3 Modbus RTU	22
2.2.3.4 Formato de Mensagem no Modo Modbus RTU	22
2.3 INVERSORES DE FREQUÊNCIA	23
2.3.1 Relação V/F Constante	24
2.3.2 Parâmetros	25
2.3.3 Inversor de Frequência WEG CFW09	26
2.3.3.1 KIT WEG Comunicação RS232 para PC	27
2.3.3.2 Função Especial - Regulador PID	28
2.4 MOTORES ELÉTRICOS	29
2.4.1 Motor de Indução Trifásico	29
2.4.1.1 Relação Velocidade do Motor e Escorregamento	31
3 DESENVOLVIMENTO DO PROJETO	32
3.1 CARACTERÍSTICAS TÉCNICAS DO PROJETO	32
3.2 CONFIGURAÇÃO DOS PARÂMETROS DE ACESSO	34
3.2.1 Parâmetros de Configuração do Motor	34
3.2.2 Parâmetros de Comunicação	35
3.3 SOFTWARE DE CONTROLE E AQUISIÇÃO DE DADOS	36
3.3.1 SCADABR	36
3.3.2 Configuração de Data Source	36

3.3.3 Data Point	38
3.3.3.1 Parâmetros de Controle e Monitoramento do Sistema	39
3.3.3.1.1 <i>Status do Coil</i>	39
3.3.3.1.2 <i>Registrador Holding</i>	40
3.3.3.2 Configuração dos Data Points	42
3.3.3.3 Watch List	45
3.4 INTERFACE PARA SISTEMA SUPERVISÓRIO	45
3.4.1 Configuração dos Componentes De Visualização	47
3.4.2 Configuração de Interface Gráfica	49
4 RESULTADOS E DISCUSSÕES	51
4.1 INTERFACE DO SUPERVISÓRIO SCADA	51
4.2 Erros de Comunicação	53
4.3 Análise dos Componentes da Interface	54
4.4 Análises do Comportamento do Sistema	55
5 CONCLUSÃO	56
5.1 SUGESTÕES PARA TRABALHOS FUTUROS	56
6 REFERÊNCIAS BIBLIOGRÁFICAS	57
7 Apêndice A	59
8 Apêndice B	61

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

O avanço dos processos industriais que estão a cada dia mais complexos, necessitam cada vez mais de serem controlados e monitorados em tempo real, pois, dessa forma é possível garantir maior confiabilidade e qualidade nos processos de produção, bem como, proporcionar maior segurança e fluidez aos operadores. Este fator levou as indústrias a investirem em sistemas de acionamentos e controle mais modernos impulsionados pelo rápido desenvolvimento das tecnologias nas áreas da informática e da transmissão de dados.

A ideia da supervisão dos processos surge desta necessidade, com a tecnologia se tornando cada vez mais rápida e eficiente, ferramentas computacionais reconhecidas mundialmente como SCADA se tornaram populares pela sua capacidade de obter aquisições de dados e representá-los de forma amigável ao operador.

Os sistemas de supervisório SCADA, são sistemas diversificados não antedendo somente os processos industriais. Possui também um papel fundamental em diversas áreas, como, as áreas de geração de energia, áreas de tratamento de água e esgoto, ou em pequenas áreas, como as de automação residencial e predial.

A vantagem da utilização de um software supervisório no monitoramento de um processo se dá a partir das suas funcionalidades frente as IHMs convencionais, além da superior facilidade de operação ainda conta com a geração de relatórios, banco de dados para armazenamento das informações além de maiores capacidades de processamento. Tendo em vista a importância dos supervisórios, pode-se concluir a importância do conhecimento dos processos pelos profissionais da área.

A necessidade de profissionais no sistema produtivo, com conhecimentos em processos industriais, capazes de compreender as interações entre os sistemas de controle e o processo é cada vez maior(...) É essencial que as escolas de formação de profissionais adéquem-se e treinem nestes tipos de aplicações industriais.(FRANÇA JR., PEREIRA E BATISTA, 2011)

Em relação a importância do supervisório nos mais variados processos, e a importância de se conhecer os softwares SCADA, pode-se se dizer que o ScadaBR (software livre) pelo baixo ou nenhum custo é ideal para ser instalado faculdades. De acordo com Silva et al. (2013) o ScadaBR é uma das ferramentas livres que pode ser utilizada por docentes para ensinar a seus alunos sobre sistemas supervisórios de controle e aquisição de dados.

Ainda segundo Silva et al. (2011) sobre softwares livres,

O software livre é algo extremamente viável para o ensino, uma vez que, permite que todos tenham acesso ao conhecimento científico e tecnológico. Partindo deste princípio, as instituições de ensino utilizam destas ferramentas para poder descentralizar o conteúdo das disciplinas ofertadas. Nas instituições de ensino superior o software livre é a base formadora de conhecimento, pois permite que o aluno continue a desenvolver suas atividades em seu próprio computador pessoal. (SILVA et.al., 2011)

1.2 PROBLEMA E JUSTIFICATIVA

Diante de algumas limitações e dificuldades encontradas na interface homem-máquina (IHM) dos inversores de frequência, em âmbito acadêmico, no que diz respeito a uma visualização mais clara das funções, ao acesso dos dados e parâmetros, à regulação do sistema e controle do motor, surgiu-se a ideia de criar um sistema supervisor didático que pudesse não só controlar e monitorar componentes básicas do inversor e do motor mas também auxiliar na compreensão do sistema com um todo e dos equipamentos envolvidos por meio de uma interface intuitiva e amigável aos usuários.

1.2.1 OBJETIVOS

1.2.1.1 OBJETIVOS GERAL

Desenvolver um sistema supervisor através do ScadaBR que faça o controle de um inversor de frequência WEG modelo CFW09 possibilitando o monitoramento e ajuste de um motor de indução trifásico, de modo que o sistema proposto possa servir como material didático e venha facilitar o entendimento dos alunos de engenharia quando estudarem por esta área de conhecimento.

1.2.1.2 OBJETIVOS ESPECIFICOS

Para atingir o objetivo geral foi necessário obedecer a objetivos específicos, estes detalhados abaixo:

- Realizar pesquisa bibliográfica do projeto;
- Levantar os parâmetros do Inversor;
- Configurar inversor de frequência;
- Configurar comunicação dos dispositivos;
- Configurar ScadaBR;
- Criar interface gráfica;

1.3 ESTRUTURA DO TRABALHO

O trabalho foi dividido em 5 capítulos:

- **No capítulo 1** foi feita uma introdução ao trabalho por meio de uma contextualização do tema, depois foi descrito o problema junto a justificativa do projeto, e por fim os objetivos propostos para a conclusão do trabalho.
- **O capítulo 2** traz uma revisão da literatura sobre os assuntos abordados no trabalho e necessários para a compreensão do projeto proposto. É feita uma descrição do sistema SCADA, das redes de comunicação industrial, dos inversores de frequência e motores de indução trifásico
- **O capítulo 3** trata sobre o desenvolvimento do supervisório realizado no software ScadaBR abordando os parâmetros e explicando todos os pontos de maneira detalhada com figuras e tabelas referentes a construção do projeto.
- **O capítulo 4** aborda o supervisório pronto onde se explica todos os pontos da interface pronta, também aborda problemas ocorridos no projeto e outras discussões a cerca das configurações do supervisório.
- **O capítulo 5** é o último, neste capítulo é feita a conclusão do trabalho e acrescenta-se sugestões para trabalhos futuros.

2 REVISÃO DA LITERATURA

2.1 SISTEMAS SUPERVISÓRIOS

Os sistemas supervisórios podem ser considerados como o nível mais alto de IHM, pois mostram o que está acontecendo no processo e permitem ainda que se atue neste. A evolução dos equipamentos industriais, com a introdução crescente de sistemas de automação industrial, tornou complexa a tarefa de monitorar, controlar e gerenciar esses sistemas. (MACHADO MARTINS, 2012)

Os sistemas supervisórios permitem que sejam rastreadas e monitoradas diversas informações de um processo produtivo ou instalação física. Essas informações são coletadas através de equipamentos de aquisição de dados e, em seguida, analisados, manipulados, armazenados e, posteriormente, apresentados ao operador por meio de um software.

Dentre os muitos sistemas utilizados, os que mais tem se difundido são os PCS (Sistemas de Controle de Processos ou Process Control Systems), os SCADA (Sistemas de Controle Supervisório e Aquisição de Dados ou Supervisory Control & Data Acquisition Systems) e os DCS (Sistemas de Controle Distribuído ou Distributed Control Systems). (MACHADO MARTINS, 2012)

Os sistemas SCADA são utilizados em larga escala na área industrial principalmente, são uma importante ferramenta capaz de gerenciar equipamentos ou processos industriais dos mais variados tipos, sejam eles, controladores automáticos, máquinas industriais e sensores, por exemplo.

Sistemas SCADA possuem como característica uma interface amigável de modo a facilitar a percepção e a atuação do operador sobre o resultado de processos complexos em “tempo real”, além de oferecer grande confiabilidade, flexibilidade e conectividade. Segundo a Fundação CERTI et al. (2010), entre as funções mais utilizadas em sistemas SCADA tem-se a geração de gráficos e relatórios com o histórico do processo, detecção de alarmes e registro de eventos em sistemas automatizados, controle de processos, acionamento e comando de equipamentos.



Figura 1: Sala de Monitoramento e Controle.(Fonte: Automation Primer, 2015)

Ainda segundo a Fundação CERTI et al. (2010), um SCADA convencional oferece drivers de comunicação com equipamentos, um sistema para registro contínuo de dados (datalogger) e uma interface gráfica para o usuário (IHM ou Interface Homem-Máquina).

Sendo assim a partir de computadores posicionados junto ao processo ou em salas de monitoramento e controle que podem estar situadas a quilômetros de distância, um software de controle e aquisição de dados disponibilizará janelas aos usuários, que podem ser de controle para comandos como: ligar/desligar, seleção de modo de operação – manual/automático ou local/remoto, manutenção e indicação de status e erro, podendo ser feitas de maneira individual e/ou para todos os equipamentos (escravos) numa rede.mesma

Há no mercado em destaque uma série de softwares para controle e aquisição de dados que atendem a demanda nacional, alguns deles são:

- **ScadaBR** (open source) da MCA Sistemas chocolate
- **InduSoft Web Studio** da InduSoft
- **Elipse** da Elipse Software
- **SIMATIC Wincc** da Siemens
- **Vijeo Citect** da Schneider Electric

2.1.1 ScadaBR

Segundo a MCA Sistemas (2016), O ScadaBR é um Sistema para Aquisição de Dados e Controle Supervisório sob licença de Software Livre, atualmente em sua versão 1.0, o sistema possibilita aos integradores ter uma solução aberta para desenvolvimento de sistemas Supervisórios para vários nichos de indústrias, entre eles a de Saneamento, Energia, Automação Predial e Industrial.

De acordo com a Fundação CERTI et al (2010). O ScadaBR por ser um modelo open source de licença gratuita, toda a documentação e o código-fonte do sistema estão à disposição, sendo permitido modificar e redistribuir o software se necessário. Computadores rodando o Windows, Linux e outros sistemas operacionais podem executar o software pelo fato deste ser uma aplicação multiplataforma baseada em Java.

O ScadaBR funciona a partir de um servidor de aplicações (ou servidor web) sendo o Apache Tomcat a escolha padrão. Ao executar o aplicativo, ele pode ser acessado a partir de um navegador de Internet, preferencialmente o Mozilla Firefox ou o Google Chrome onde será possível, após a configuração das variáveis (tags) de uma aplicação automatizada, montar interfaces de operador Web utilizando o próprio navegador

Os passos iniciais para a utilização e configuração do software se dá na prévia definição do protocolo e do tipo de comunicação com os equipamentos que serão monitorados e controlados. Segundo MCA Sistemas (2016) o ScadaBR possui drivers de aquisição de dados para utilização com os protocolos de comunicação mais difundidos atualmente na Indústria (por exemplo, OPC, Modbus Serial & TCP, DNP3, IEC101, Bacnet.)

Sendo assim, partindo-se da definição do tipo de comunicação e protocolo e iniciando-se o software. Segundo a Fundação CERTI et al. (2010) a operação do ScadaBR baseia-se em:

- **Tipos de dados:** São suportados valores binários (ou booleanos), valores de estados múltiplos, valores numéricos (ou analógicos), valores alfanuméricos e valores em imagens.
- **Data sources:** Um data source é um "lugar" de onde os dados são recebidos. Virtualmente, qualquer coisa pode ser um data source, desde que o protocolo de comunicação seja suportado pela aplicação. Valores de dados recebidos ou coletados por um data source são armazenados em data points.
- **Data point:** Um data point é uma coleção de valores históricos associados, por exemplo, um ponto pode ser uma leitura de temperatura de um quarto (valores numéricos), bem como podem ser valores de controle, como um indicador para ligar ou desligar um equipamento (valores binários).
- **Monitoramento:** O monitoramento de pontos dentro do sistema pode ser feito por meio de uma watch list (lista dinâmica de pontos com seus valores) ou de representações gráficas onde em ambos os Valores e gráficos são atualizados em tempo real sem ter que atualizar a janela do navegador.
- **Controle:** O controle de sistemas externos pode ser obtido para pontos que podem ser "setados"(ou que permitem escrita ou saída). Um ponto setável pode ser "setado" para um valor definido pelo usuário, como controle de velocidade de um motor.
- **Eventos:** Um evento é a ocorrência de uma condição definida no sistema. Existem tanto eventos definidos pelo sistema (erros de operação de data sources, logins de usuários, etc.) como definidos pelo usuário (detectores de valor, eventos agendados, etc.)

2.2 PROTOCOLOS DE COMUNICAÇÃO INDUSTRIAL

Segundo Machado Martins (2012). Os protocolos de comunicação são um conjunto de regras, procedimentos e leis que governam a troca de informação entre dois ou mais processos. Os protocolos industriais mais conhecidos são: Interbus, Modbus, Devicenet, Profibus, Hart, Ethernet entre outros. Todos estes destinados a comunicação entre sistemas de automação por meio de barramento de campo serial.

Os protocolos são fundamentais para a comunicação dos instrumentos de campo aos sistemas de controle e sistemas de monitoramento, sem protocolos, uma rede não funciona, pois ele determina como um programa deve preparar os dados para serem enviados para o estágio seguinte do processo de comunicação. A utilização das redes de comunicação industrial permite uma comunicação rápida e confiável, a análise das peculiaridades de cada plataforma disponível e sua relação custo-benefício são fatores importantes no que diz respeito à produtividade industrial.

2.2.1 Unidades Físicas de Comunicação Serial

A comunicação serial é utilizada em redes industriais, onde os padrões mais comuns são: RS-232, RS-422/RS-485. Em uma interface serial os bits de dados são enviados sequencialmente através de um canal de comunicação ou barramento.

A sigla RS significa *Recommended Standard* (padrão recomendado), sua nomenclatura foi substituída por EIA-232 na década de 90, no entanto a nomeação antiga do padrão ainda é a mais utilizada na prática hoje em dia.

As Características das interfaces seriais segundo a National Instruments Corporation (2013), são:

- **RS-232**, a interface serial mais comum, utilizado como componente padrão na maioria dos computadores de mesa compatíveis com Windows. O RS-232 permite apenas um transmissor e um receptor em cada linha.
- **RS-422** (Norma EIA RS-422-A) é a conexão serial utilizada nos computadores Apple. Ela fornece um mecanismo para transmitir dados a até 10 Mbits/segundo. O RS-422 envia cada sinal utilizando 2 fios para aumentar a máxima taxa de transmissão (baud rate) e comprimento do cabo.
- **RS-485** é uma melhoria do RS-422 e expande nas capacidades. O RS-485 foi feito para consignar a limitação multiponto do RS-422 (possui limite de 10 pontos), permite até 32 dispositivos para comunicar através da mesma linha de dados.

A taxa de transferência ou *baud rate*, que é a velocidade com que os dados são enviados através de um canal, é menor na interface serial RS232 (ponto-a-ponto), segundo a National Instruments Corporation (2013), a maioria dos dispositivos são limitados em 115,2 kbits/segundo e uma distância de comunicação de até 15 metros. Nos padrões RS 485/422 (multi-ponto) a taxa de transferência é superior aos do padrão RS232, isso possibilita a transmissão a uma distância de até 1200 metros, e uma comunicação eficiente com diverso equipamento na mesma rede.

2.2.2 Conceito Mestre/Escravo

Define-se como mestre o equipamento responsável por toda a iniciativa do processo de comunicação, ou seja, uma troca de dados entre os equipamentos, quer para transmissão ou recepção, é iniciado e finalizado pelo mestre. (ALFA INSTRUMENTOS, 2000)

Define-se como escravo o equipamento que, no processo de comunicação, só realiza alguma função sob solicitação e controle do mestre. (ALFA INSTRUMENTOS, 2000)

Segundo Seixas Filho (2003). Um modelo de comunicação mestre-escravo, é quando um único dispositivo, o mestre, inicia transações denominadas *queries*. Os demais dispositivos da rede (escravos) respondem, suprindo os dados requisitados pelo mestre ou executando uma ação por ele comandada.

2.2.3 Protocolo Modbus

O protocolo Modbus foi desenvolvido pela Modicon Industrial Automation Systems, hoje Schneider Eletric, para comunicar um dispositivo mestre com outros dispositivos escravos. (SEIXAS FILHO, 2003). Este protocolo basicamente define uma estrutura de mensagens composta por bytes, que os mais diversos tipos de dispositivos são capazes de reconhecer, independentemente do tipo de rede utilizada. (ALFA INSTRUMENTOS, 2000). Tipicamente, o mestre é uma interface homem-máquina (IHM) ou sistema SCADA (Supervisory Control and Data Acquisition) e o escravo é um sensor, controlador lógico programável (CLP) ou controlador programável para automação (CPA). (NATIONAL INSTRUMENTS CORPORATION, 2014)

O protocolo Modbus é um protocolo aberto que emprega duas variações nos modos de transmissão (ASCII e RTU) que definem o conteúdo de bit da mensagem que será transmitida na rede e como a informação da mensagem será enviada e recebida. No entanto, embora o protocolo modbus seja utilizado normalmente sobre conexões seriais, segundo a National Instruments Corporation (2014), outras unidades de dados de aplicação foram sendo introduzidas ao longo dos anos a fim de alterar o formato dos pacotes utilizados sobre o serial ou permitir o uso de TCP/IP sobre Ethernet e redes UDP (User Datagram Protocol).

2.2.3.1 Acesso aos Dados do Modbus

No protocolo Modbus existem 4 blocos de dados ou faixas de endereço, estes blocos armazenam os dados que podem ser acessados pelo Modbus, são eles: coils (ou saídas digitais/discretas), entradas discretas, registradores holding (ou registrados de saída) e registradores de entrada.

Bancos de dados definem o tipo e os direitos de acesso dos dados contidos. Os dispositivos escravo têm acesso direto a esses dados, que são hospedados localmente nos dispositivos. Os dados que podem ser acessados pelo Modbus

são de forma geral um subconjunto da memória principal do dispositivo. Por outro lado, os mestres Modbus precisam solicitar acesso a esses dados, utilizando diversos códigos de função.(...)Esses blocos dão a você a capacidade de restringir ou permitir acesso a diferentes elementos de dados e também fornecer mecanismos simplificados na camada de aplicação para o acesso a diferentes tipos de dados. (NATIONAL INSTRUMENTS CORPORATION, 2014)

As faixas de endereçamento de cada bloco são especificadas por um prefixo que indicará qual bloco se deseja ter acesso, os prefixos são classificados de 1 a 4 como sendo: 1 para coils, 2 para entradas discretas, 3 para registradores de entrada e 4 para registradores holding. Os números que seguem após o prefixo são os valores que indicam um endereçamento, ou um parâmetro armazenado no escravo no caso de um inversor, por exemplo.

Bloco de dados	Prefixo
Coils	0
Entradas discretas	1
Registradores de entrada	3
Registradores holding	4

Figura 2: Quadro de Prefixos das Faixas de Dados.(Fonte: National Instruments Corporation, 2014)

Um endereço ou parâmetro de número 1 pode ser referenciado, como 4.0001 ou 40.0001 para um registrador do tipo holding, ou para um coil como 0.0001 ou 00.0001. Segundo a National Instruments Corporation (2014), a diferença entre 4XXX, 4XXXX e 4XXXXXX depende do espaço de endereços utilizado pelo dispositivo, neste caso, se apenas alguns registradores forem usados, uma prática comum é usar a faixa de 4.001 a 4.999.

2.2.3.2 Modbus ASCII

Segundo Alfa Instrumentos (2000). Para cada palavra de dados da mensagem são enviados dois caracteres no padrão ASCII, onde a principal vantagem do modo ASCII é a possibilidade de haver intervalos grandes entre o envio dos dados de uma mesma mensagem.

Os dados transmitidos sobre codificação em caracteres ASCII contem 7 bits, no entanto, ao considerar-se os parâmetros de comunicação a quantidade de bits por palavra de dados sempre será igual a 10. Ou seja, além dos 7 bits de dados, cada palavra conterá 1 bit de Start, 1 bit de Paridade (Par ou Impar) e 1 bit de Parada (Stop). Caso não haja bit de paridade, considera-se 2 bits de Parada (Stop), ou seja, um bit adicional para manter a quantidade de bits da palavra de dados.

2.2.3.3 Modbus RTU

De acordo com Alfa Instrumentos (2000). Para cada palavra de dados da mensagem em modo RTU é enviado apenas um caractere no padrão HEXADECIMAL, onde a principal vantagem do modo RTU em relação ao ASCII é a maior densidade de caracteres que é enviada numa mesma mensagem, aumentando o desempenho da comunicação.

O modo RTU permite a compactação dos dados em pequenos pacotes, podendo os endereços e valores ser representados em formato binário e os números inteiros podem variar entre -32768 e 32767 sendo representados por 2 bytes. Este modo possui uma mensagem duas vezes menor que o comprimento da mensagem ASCII equivalente, e conseqüentemente, tem um menor consumo de dados.

No Modbus RTU as palavras possuem 8 bits, todavia, os 8 bits de dado da mensagem RTU são na verdade 11 bits quando adicionados os bits de start, paridade e stop. Do mesmo modo que no Modbus ASCII, caso não haja bit de paridade, adiciona-se 1 stop bit para manter a quantidade da palavra de dados (10 bits para ASCII e 11 bits para RTU).

2.2.3.4 Formato de Mensagem no Modo Modbus RTU

O formato do pacote de dados enviados na pergunta (mestre) e na resposta (escravo), possuem a mesma estrutura utilizada: Endereço do Dispositivo, Código da Função, Dados e CRC - Verificação de Erro. Cada mensagem deve ser transmitida numa seqüência de “palavras”.

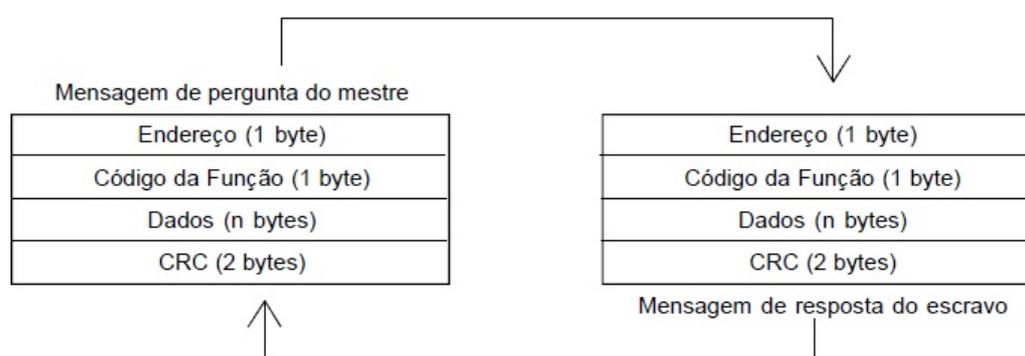


Figura 3: Estrutura das Mensagens em Modbus RTU.(Fonte: WEG, 2011)

Segundo Seixas Filho (2003) e Alpha Instrumentos (2000) a estrutura da mensagem é definida da seguinte forma:

- **Endereço do Dispositivo:** O mestre inicia a comunicação enviando um byte com o endereço do escravo para o qual se destina a mensagem, a faixa de endereços válidos vai de 0 a 247 sendo o endereço 0 para *broadcast* (todos escravos) e os números 1 a 247 são

usados para endereçar os escravos. O escravo coloca na resposta seu próprio endereço no campo de endereço da mensagem para sinalizar ao mestre qual escravo está respondendo.

- **Código da Função:** Onde o mestre estabelece o tipo de serviço ou função solicitada ao escravo (leitura e escrita, etc.). A faixa de valores varia de 1 a 255, os números 0 a 127 são utilizados para definir as funções e os números 128 a 255 são utilizados para informar os erros na transmissão. Na resposta do escravo este campo é usado para informar ao mestre se a ação foi executada com sucesso ou se ocorreu algum tipo de erro (exception).
- **Dados:** São os dados necessários para a execução da função, nos Bytes de Dados vão as informações adicionais necessárias, os endereços de memória, a quantidade de itens transmitidos, e a quantidade de bytes do campo.
- **CRC:** O campo CRC (Cyclic redundancy check) realiza a verificação de erros na mensagem. No caso de erro, este campo conterá um código de exception, identificando o motivo que causou o erro e orientando o mestre na execução do próximo comando. Tem o tamanho de 16 bits, os bits de configuração: start, paridade e stop bits, não são utilizados no cálculo do CRC, apenas os bits do caractere propriamente dito.

No protocolo Modbus RTU existem dois tipos de mecanismo de checagem de erro, são eles: checagem da paridade, aplicada para cada caractere da mensagem, e o frame check sequence (checagem do framing) aplicado a todos os campos de uma mensagem, que é do tipo CRC no modo RTU.

Os bits para verificação de paridade são opcionais, e as configurações podem ser paridade par, ímpar ou sem paridade.

Um *bit* extra é adicionado a cada caractere de modo que ele contenha um número par ou ímpar de *bits* ativos, caso sejam adotadas respectivamente paridade par ou ímpar. A principal desvantagem desse método é sua incapacidade de detectar números pares de inversões de *bit*.(SEIXAS FILHO, 2003)

2.3 INVERSORES DE FREQUÊNCIA

Inversores de frequência são equipamentos que transformam a tensão da rede, de amplitude e frequência constantes, em uma tensão de amplitude e frequência variáveis. É um equipamento atualmente utilizado em larga escala de aplicação nos setores comerciais e industriais com objetivo de controlar e variar a velocidade de motores de indução trifásicos, pois, como os inversores são capazes de gerar tensões e frequências trifásicas ajustáveis, possibilitam um controle preciso do torque e da velocidade. Além disso, proporcionam também vantagens como um maior nível de segurança, melhor aproveitamento de energia, controle a distância, redução de custos e aumento na qualidade e produtividade dos processos.

Conforme exposto sobre o funcionamento dos inversores de frequência em WEG (2016), O processo de obtenção da tensão e frequência desejadas por meio de tais equipamentos passa por três estágios, sendo eles, a ponte de diodos, que é responsável pela retificação do sinal alternado proveniente da rede de alimentação; o filtro ou link DC, responsável pelo alisamento/regulação da tensão retificada com armazenamento de energia por meio de um banco de capacitores; e os transistores IGBT, que por sua vez são responsáveis pela inversão da tensão contínua proveniente do link DC num sinal alternado, com tensão e frequência variáveis.

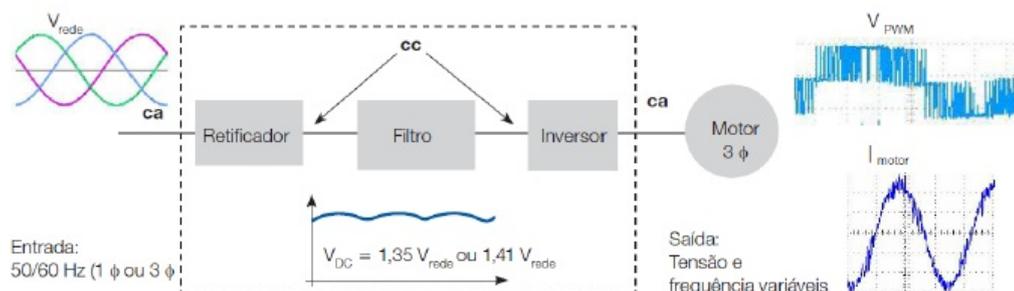


Figura 4: Diagrama de blocos do processo de obtenção da tensão e frequência em um inversor. (Fonte: WEG, 2016)

Ainda segundo a WEG (2016), o controle realizado pelos inversores eletrônicos podem ser de dois tipos:

- **Controle escalar:** Impõe no motor uma determinada relação tensão/frequência, visando manter o fluxo magnético do motor aproximadamente constante. Aplicável quando não há necessidade de respostas rápidas a comandos de torque e velocidade ou quando há conexão de múltiplos motores a um único inversor. É o mais utilizado devido à sua simplicidade e ao fato de que a grande maioria das aplicações não requer alta precisão e/ou rapidez no controle da velocidade.
- **Controle vetorial:** Possibilita atingir um elevado grau de precisão no controle do torque e velocidade do motor. O controle decompõe a corrente do motor em dois vetores: um que produz o fluxo magnetizante e outro que produz torque, regulando separadamente o torque e o fluxo. Pode ser realizado em malha aberta (“sensorless”) ou em malha fechada (com realimentação).

2.3.1 Relação V/F Constante

O funcionamento dos inversores de frequência com controle escalar está baseado numa estratégia de comando chamada “V/F constante”.

Para Capelli (2002). A função do inversor de frequência não é apenas controlar a velocidade de um motor, ele também precisa manter o torque (conjugado) constante para não

provocar alterações na rotação, quando o motor estiver com carga. Ou seja, o inversor deve ser capaz de controlar a velocidade da máquina e manter o torque constante mesmo nos momentos em que uma carga mecânica é imposta ao motor.

Para que o torque no motor fique constante o inversor deve manter a razão V/F constante, isto é, caso haja variação na frequência de alimentação do motor, deve haver uma variação proporcional da tensão. A curva V/F constante varia para se obter no estator uma corrente constante da ordem da corrente nominal do motor, pela equação $I \cong V/F = Cte$.

No motor em baixas velocidades (frequências abaixo de 30 hz), segundo Mascheroni, Llichtblau e Gerardi (2004) e WEG (2016), a resistência do estator começa a ter influência no cálculo da corrente, deste modo, a corrente e conseqüentemente o torque do motor diminuem bastante. Além do método de compensação $I \times R$ para aumentar a tensão e evitar este problema, de maneira que a capacidade de torque do motor seja mantida, alguns inversores possuem funções especiais como a compensação de escorregamento (que atenua a variação da velocidade em função da carga) e o boost de tensão (alteração da curva característica V/F para compensar o efeito da queda de tensão na resistência estática com objetivo de aumentar torque de partida).

2.3.2 Parâmetros

Um parâmetro do inversor de frequência é um valor de leitura ou escrita, através do qual o usuário pode ler ou programar valores que mostrem, sintonizem ou adéquem o comportamento do inversor e motor em uma determinada aplicação. (MASCHERONI; LICHTBLAU; GERARDI, 2004)

De acordo com Capelli (2002). Para que o inversor funcione a contento, não basta instalá-lo corretamente. É preciso “informar” a ele em que condições de trabalho irá operar. Essa tarefa é justamente a parametrização do inversor.

Os inversores disponíveis no mercado possuem parâmetros programáveis similares, independente do fabricante, o que se observa é a quantidade de recursos que cada modelo de inversor disponibiliza ao operador, assim, quanto maior o número de recursos que o inversor oferece devido a sua sofisticação, maior também será o número de parâmetros disponíveis.

Estes parâmetros são normalmente acessíveis através de uma interface chamada de interface Homem-Máquina (IHM), que dispõem de um mostrador digital e um teclado por onde o operador configura os parâmetros.

Os parâmetros podem ser agrupados pelas seguintes características, segundo Mascheroni, Llichtblau e Gerardi (2004):

- **Parâmetros de Leitura:** permitem visualizar os valores programados nos parâmetros de regulação, de configuração, do motor, mas não podem ser alteradas pelo usuário, por exemplo, velocidade do motor.

- **Parâmetros de Regulação:** Consistem em valores ajustáveis a serem utilizados pelas funções do inversor, por exemplo, tempo de aceleração e desaceleração do motor.
- **Parâmetros de Configuração:** Definem as características do inversor e as funções a serem executadas, por exemplo, definir modo de controle do inversor se local ou remoto.
- **Parâmetros do motor:** Definem as características nominais do motor a ser ajustado e utilizado em conjunto com o inversor, por exemplo, corrente nominal, tensão nominal, etc.
- **Parâmetros Especiais:** Consistem em funções especiais que podem ser habilitadas e configuradas pelo usuário como o regulador PID de um sistema.

2.3.3 Inversor de Frequência WEG CFW09

De acordo com a WEG (2016), o inversor de frequência CFW09 é destinado para o acionamento de motores trifásicos com potência de 1,5 a 1500 cv. Possui as mais avançadas tecnologias disponíveis para variação de velocidade em motores de indução trifásicos, pois incorpora em um único produto, técnicas de controle escalar (V/F), vetorial sensorless e vetorial com encoder.

As principais aplicações do inversor se dão principalmente em bombas centrífugas, bombas dosadoras de processo, ventiladores, exaustores, agitadores, misturadores, extrusoras, esteiras transportadoras, mesas de rolos, granuladores, secadores, fornos rotativos, máquinas de corte e solda, elevadores, sistemas multibombas, etc.

O inversor CFW09 ainda permite a possibilidade de comunicação entre outros dispositivos através de redes de comunicação “Fieldbus” que podem ser via serial RS-232 utilizando a comunicação Modbus RTU ou RS-485 utilizando internamente um cartão de rede (EBA ou EBB), também permite comunicação via Profibus DP e DeviceNet. Possui também disponibilidade de 100 parâmetros configuráveis pelo usuário via software SuperDrive ou IHM facilitando o acesso às informações e um manual de usuário com informações detalhadas a cerca do equipamento e suas funções.



Figura 5: Bancada do Inversor CFW09, Laboratório de Máquinas Elétricas/FIC.(Fonte: AUTOR)

2.3.3.1 KIT WEG Comunicação RS232 para PC

Para se fazer a supervisão do CFW-09 da WEG utilizando a interface serial RS-232, é necessário o kit de comunicação serial, disponível como um dispositivo opcional pela WEG, onde o mestre pode ser um computador, um controlador lógico programável (CLP) entre outros.

Este módulo é colocado no lugar da IHM e permite a comunicação via RS-232 utilizando o conector RJ11 para DB9. Além disso, caso seja necessária a utilização da IHM, o equipamento dispõe de uma conexão frontal onde esta passa a funcionar em conjunto com a comunicação serial, pode ser visto na Figura 10.

O software SuperDrive para Windows é um software gratuito desenvolvido e disponibilizado pela WEG que permite a programação, operação e monitoração do CFW-09. No entanto, o kit de comunicação serial RS232 permite ainda a comunicação com outros softwares de controle e supervisão através do protocolo Modbus RTU.

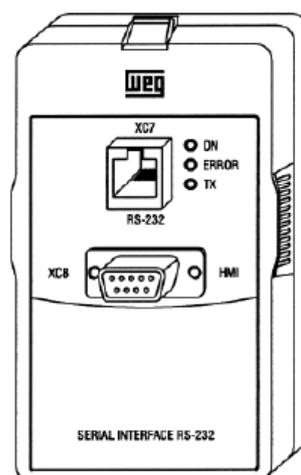


Figura 6: Kit WEG Comunicação Serial.(Fonte: WEG, 2011)

2.3.3.2 Função Especial - Regulador PID

O inversor CFW09 dispõe de diversas funções para o controle da velocidade do motor, uma dessas funções é o regulador PID, que faz o papel de regulador proporcional, integral e derivativo e pode ser aplicada no controle de um processo em malha fechada.

Segundo a WEG (2016), a velocidade será variada de modo a manter a variável de processo (aquela que se deseja controlar, por exemplo: nível de água de um reservatório, temperatura, dosagem, etc.) no valor desejado, ajustado na referência (setpoint).

Em um sistema de controle típico, a variável do processo é o parâmetro do sistema que precisa ser controlado como temperatura ($^{\circ}\text{C}$), pressão (psi) ou fluido (litros/minuto). Um sensor é usado para medir a variável de processo e fornecer feedback para o sistema de controle. O set point é o valor desejado ou comando para a variável de processo, tais como 100°C , no caso de um sistema de controle de temperatura. A qualquer momento, a diferença entre a variável de processo e o set point é usada pelo algoritmo do sistema de controle (compensador), para determinar a saída desejada do atuador, que por sua vez, acionará o sistema (planta). (NATIONAL INSTRUMENTS CORPORATION, 2011)

Ainda sobre a teoria PID o National Instruments Corporation (2001) explana que, a ideia básica por trás de um controlador PID é ler um sensor, calcular a resposta de saída do atuador através do cálculo proporcional, integral e derivativo e então somar os três componentes para calcular a saída.

O processo de configuração ideal para os ganhos P, I e D para obter uma resposta ideal de um sistema de controle é chamado ajuste, como exemplo, podemos considerar o ajuste de sintonia manual de PID, por tentativa e erro com mínima análise analítica, e o mais comumente utilizado, método de Ziegler-Nichols, tratando-se de um método mais analítico para obtenção dos ganhos PID.

No manual do fabricante para o inversor CFW09, são sugeridos alguns ajustes para os ganhos do regulador PID. Na quadro da Figura 7 são apresentados os valores sugeridos para controle de algumas grandezas.

Grandeza	Ganhos			Tempo
	Proporcional P520	Integral P521	Derivativo P522	Rampa PID P523
Pressão em sistema pneumático	1	0.043	0.000	3.0
Vazão em sistema pneumático	1	0.037	0.000	3.0
Pressão em sistema hidráulico	1	0.043	0.000	3.0
Vazão em sistema hidráulico	1	0.037	0.000	3.0
Temperatura	2	0.004	0.000	3.0

Figura 7: Quadro para ajustes de PID.(Fonte: WEG, 2011)

2.4 MOTORES ELÉTRICOS

Segundo a WEG (2016), motor elétrico é a máquina destinada a transformar energia elétrica em energia mecânica. Os dois tipos de motores mais comuns são, os motores de corrente contínua, que possuem um custo elevado de instalação e manutenção, até mesmo pelo fato de necessitarem de uma fonte de corrente contínua, ou de um dispositivo conversor para seu funcionamento. E os motores de corrente alternada, sendo estes os mais utilizados devido a sua grande simplicidade, robustez e baixo custo, sendo adequado para quase todos os tipos de máquinas acionadas, encontradas na prática.

Os principais tipos de motores CA são:

- **Motor síncrono:** funcionam com velocidade fixa, ou seja, sem interferência do escorregamento; utilizado normalmente para grandes potências (devido ao seu alto custo em tamanhos menores).
- **Motor de indução:** funcionam normalmente com uma velocidade constante, que varia ligeiramente com a carga mecânica aplicada ao eixo. É possível o controle da velocidade dos motores de indução com o auxílio de inversores de frequência.

2.4.1 Motor de Indução Trifásico

Para análise de funcionamento pode se considerar o motor de indução como um transformador, onde o enrolamento primário deste transformador é formado pelo estator e o enrolamento secundário pelo rotor. O próprio nome “motor

de indução” se deve ao fato de que toda a energia requerida pelo rotor para a geração de torque é “induzida” pelo primário do transformador (estator) no secundário (rotor). Como existem dois campos magnéticos, um no estator e outro no rotor, aparecerá uma força entre o rotor e o estator que fará com que o rotor gire, já que é o único que pode se movimentar pois está montado sobre rolamentos, disponibilizando assim energia mecânica (torque) no seu eixo. (MASCHERONI; LICHTBLAU; GERARDI, 2004)

De acordo com WEG (2016), o motor de indução trifásico, representado pela Figura 8, é composto fundamentalmente por um estator e um rotor.

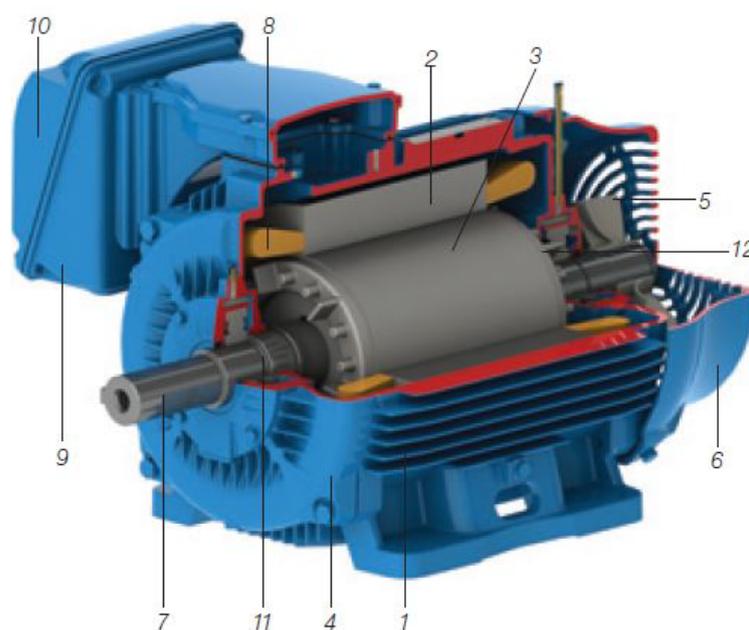


Figura 8: Esquema de MIT.(Fonte: WEG, 2016)

Rotor:

- (1) Carcaça: estrutura suporte do conjunto de construção robusta em ferro fundido, aço ou alumínio injetado, resistente à corrosão e normalmente com aletas.
- (2) Núcleo de chapas: chapas de aço magnético.
- (8) Enrolamento trifásico: três conjuntos iguais de bobinas, formando um sistema trifásico equilibrado ligado à rede trifásica de alimentação.

Estator:

- (3) Núcleo de chapas: mesmas características das chapas do estator.
- (12) Barras e anéis de curto-circuito: alumínio injetado sob pressão numa única peça.

Outras partes:

- (4) Tampa, (5) Ventilador, (6) Tampa defletora, (9) Caixa de ligação, (10) Terminais, (11) Rolamentos

2.4.1.1 Relação Velocidade do Motor e Escorregamento

Ainda de acordo a WEG (2016), a velocidade síncrona do motor é definida pela velocidade de rotação do campo girante, que depende do número de polos do motor e da frequência (f) da rede, em Hertz, é dada pela Equação (2.1). Se o motor gira a uma velocidade diferente da velocidade síncrona, devido a uma carga em seu eixo que faz a rotação diminuir, a diferença entre a velocidade do motor e a velocidade síncrona chama-se escorregamento e pode ser expresso em rotações por minuto (rpm), como fração da velocidade síncrona, ou como ainda, porcentagem desta (2.2).

$$n_s = \frac{60 \times f}{P} (rpm) \quad (2.1)$$

$$s(rpm) = n_s - n; \quad s = \frac{n_s - n}{n_s}; \quad s(\%) = \frac{n_s - n}{n_s} \times 100 \quad (2.2)$$

$$n = n_s \times \left(1 - \frac{s(\%)}{100}\right) \quad (2.3)$$

Sendo assim, a velocidade do motor, para um dado escorregamento s , será dado pela equação (2.3).

3 DESENVOLVIMENTO DO PROJETO

3.1 CARACTERÍSTICAS TÉCNICAS DO PROJETO

Para o início do projeto foi necessário conhecer e definir os dados referentes aos equipamentos que seriam utilizados na implementação do supervisor.

A comunicação do inversor de frequência, WEG modelo CFW09, foi realizada por meio do cabo serial RS232 utilizando a entrada RJ11. O cabo serial, por sua vez, foi conectado a um outro cabo conversor, onde sua saída DB9 (9 pinos) foi transformada para USB, possibilitando então a conexão Inversor-PC.



Figura 9: Cabo Conversor (Esquerda) e Cabo Serial (Direita) .(Fonte: AUTOR)

A descrição da interface serial RS232 é dada segundo a WEG (2011) como:

- Utilizada para conexão ponto a ponto (entre um único escravo e o mestre).
- Distância máxima: 10 metros.
- Níveis de sinal seguem a EIA STANDARD RS-232C.
- Três fios: transmissão (TX), recepção (RX) e retorno (0 V).
- Deve-se utilizar o módulo RS-232 Serial Interface.

O inversor de frequência utiliza o módulo (Kit) de comunicação, disponibilizado pela WEG como dispositivo opcional, para se comunicar com o PC através do protocolo Modbus

RTU, sendo o inversor o escravo definido com o endereço 1 na rede, e o PC o mestre, responsável pelo controle.



Figura 10: IHM conectada ao módulo de comunicação serial.(Fonte: AUTOR)

O motor utilizado é um motor de indução trifásico assíncrono rotor de gaiola de alto rendimento, do grupo Voges Metalurgia LTDA, modelo VTOP 80a, 4 polos. Possuindo os seguintes dados:

- Corrente Nominal: 2,86 A
- Frequência: 60 Hz
- Potência: 220 V
- Rotação: 1730 rpm

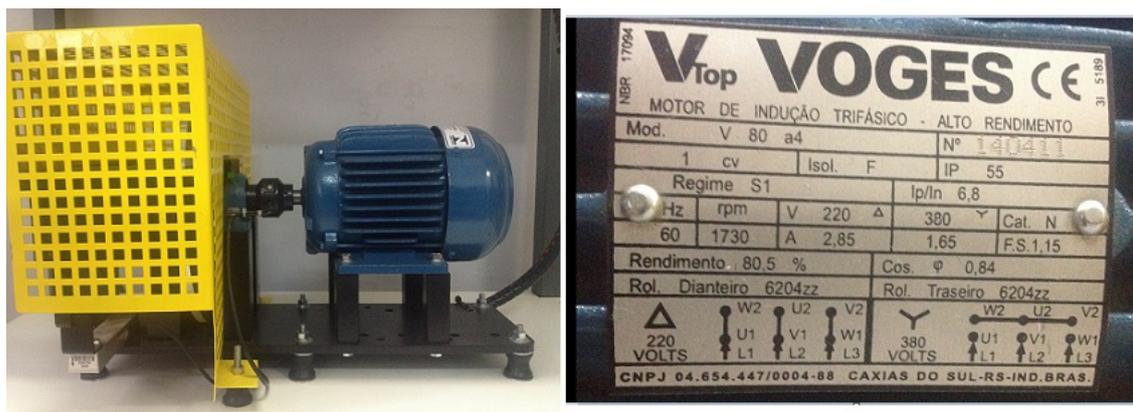


Figura 11: Motor de Indução e placa de dados.(Fonte: AUTOR)

3.2 CONFIGURAÇÃO DOS PARÂMETROS DE ACESSO

3.2.1 Parâmetros de Configuração do Motor

Ao dar início a uma nova atividade no inversor de frequência é importante efetuar o reset da máquina, isso permite ao usuário iniciar os ajustes a partir das configurações de fábrica e regular de acordo com as necessidades de cada projeto.

Todos os ajustes no inversor são feitos através de parâmetros¹ que são indicados no display através da letra P seguida de um número.

Primeiramente é necessário habilitar o acesso para alteração do conteúdo dos parâmetros, o ajuste é feito do P000 = 0 para P000= 5, feito isto, já é possível realizar o reset do equipamento no parâmetro P204 = 5, o valor 5 carrega os parâmetros atuais do inversor com os ajustes de fábrica para 60 Hz.

Apos o reset, é automaticamente requerido pela IHM a configuração dos dados do motor em uso. As informações contidas na placa do motor foram configuradas no inversor de frequência de acordo com a Tabela 1.

Tabela 1: Parâmetros do Motor

MOTOR	PARÂMETROS	DADOS
Tensão Nominal	P400	220 V
Corrente Nominal	P401	2,85 A
Velocidade Nominal	P402	1730 RPM
Frequência Nominal	P403	60 HZ
Potencial Nominal	P404	1 CV
Ventilação	P406	Auto Ventilado

Fonte: Autor

¹ Parâmetros: são aqueles existentes nos inversores cuja visualização e alteração é possível através da HMI (Interface Homem - Máquina).(WEG, 2011)

3.2.2 Parâmetros de Comunicação

Para a utilização do protocolo de comunicação Modbus-RTU no inversor de frequência, é necessário realizar algumas configurações.

O parâmetro P308 define o endereço do inversor (escravo) na rede, o parâmetro P312 seleciona o tipo de protocolo serial a ser utilizado na comunicação. No projeto foi definido P312 = 1, a opção é, baud rate de 9600 bps, sem paridade. A opção sem paridade faz com que sejam necessários 2 bits de parada (Stop Bit) ao contrário de 1.

É fundamental que todos os dispositivos relacionados a comunicação do projeto estejam configuradas com as mesmas especificações, ou seja, tanto o PC, quanto o software e o inversor devem obrigatoriamente possuir a mesma configuração de P312, relacionado ao protocolo Modbus-RTU.

Os demais parâmetros (P220 a P228) concederão permissões para que certos comandos sejam dados ao inversor através da comunicação serial ou através da IHM e devem ser alterados conforme a necessidade do usuário.

Na Tabela 2, estão dispostos alguns parâmetros e configurações do inversor de frequências relacionadas à comunicação serial e as demais funções de comando inicial.

Tabela 2: Parâmetros de Comunicação

FUNÇÃO	PARÂMETROS	AJUSTE
Senha de Acesso	P000	5
Tipo de Protocolo Serial	P312	1 = Modbus-RTU, 9600 bps, No Parity
Endereço Serial	P308	1
Seleção Local/Remoto	P220	5 = Serial (Default Local)
Seleção Referência Veloc. Local	P221	0 = Teclas da HMI
Seleção Referência Veloc. Remota	P222	9 = Serial
Seleção Sentido de Giro Local	P223	2 = Tecla da HMI (Default Horário)
Seleção Gira/Pára Local	P224	0 = Teclas I/O da HMI
Seleção do JOG Local	P225	1 = Tecla da HMI
Seleção Sentido de Giro Remoto	P226	5 = Serial (Default Horário)
Seleção Gira/Pára Remoto	P227	2 = Serial
Seleção do JOG Remoto	P228	3 = Serial

Fonte: Autor

A configuração entre os modos local e remoto ficaram relacionados de forma que todas as opções referentes à situação REMOTO foram habilitadas para comando via comunicação serial, e todas as opções referentes à situação LOCAL foram habilitadas para comando via IHM do próprio inversor. Ou seja, a opção de controle do inversor pelo usuário é facultativa, podendo ser tanto pelo supervisório quanto pela IHM local.

3.3 SOFTWARE DE CONTROLE E AQUISIÇÃO DE DADOS

3.3.1 SCADABR

O acesso ao ScadaBR se inicia por meio da ativação (start) do servidor web Java, Apache Tomcat (ver. 7.0.72). O endereço do aplicativo, foi configurado durante a instalação com porta 8085, sendo: <localhost:8085/ScadaBR/login.htm>.

O login (User ID) e a senha (Password) de acesso, são “admin” por padrão, a não ser que sejam alterados durante a instalação.

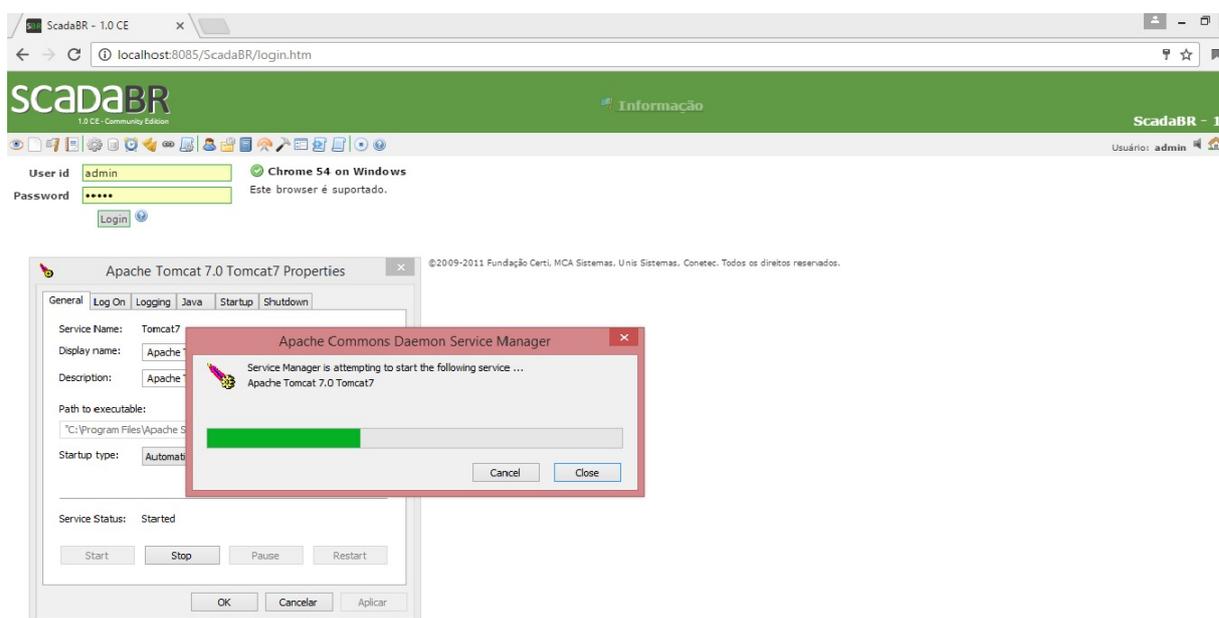


Figura 12: Tela de Login do ScadaBR.(Fonte: AUTOR)

3.3.2 Configuração de Data Source

A configuração do ScadaBR para se comunicar com o inversor se dá a partir da criação de um Data Source do tipo Modbus Serial. Após a seleção do modo modbus, é necessário configurar as opções de comunicação (Data Bits, Stop Bits, Paridade e Baud Rate), que como já citado, devem ser as mesmas do inversor.



Figura 13: Data Source no ScadaBR.(Fonte: AUTOR)

No campo *Nome* é definido o nome do data source, neste caso o mesmo recebeu o nome do modelo do inversor (CFW09). No campo *Período de Atualização* é determinado a frequência com que são solicitados os dados do equipamento Modbus. No campo *Timeout* é determinado o tempo (em milisegundos) que o data source aguardará a resposta de uma solicitação, caso não receba o data source tentará novamente quantas vezes estiver definido no campo de *Tentativas*.

Ou seja, os campos *Timeout* e *Tentativas* determinam o comportamento do sistema no caso de falhas de solicitação.

A configuração da Porta USB também foi configurada no PC com as mesmas especificações do data source e do inversor, como é demonstrado na figura 14.

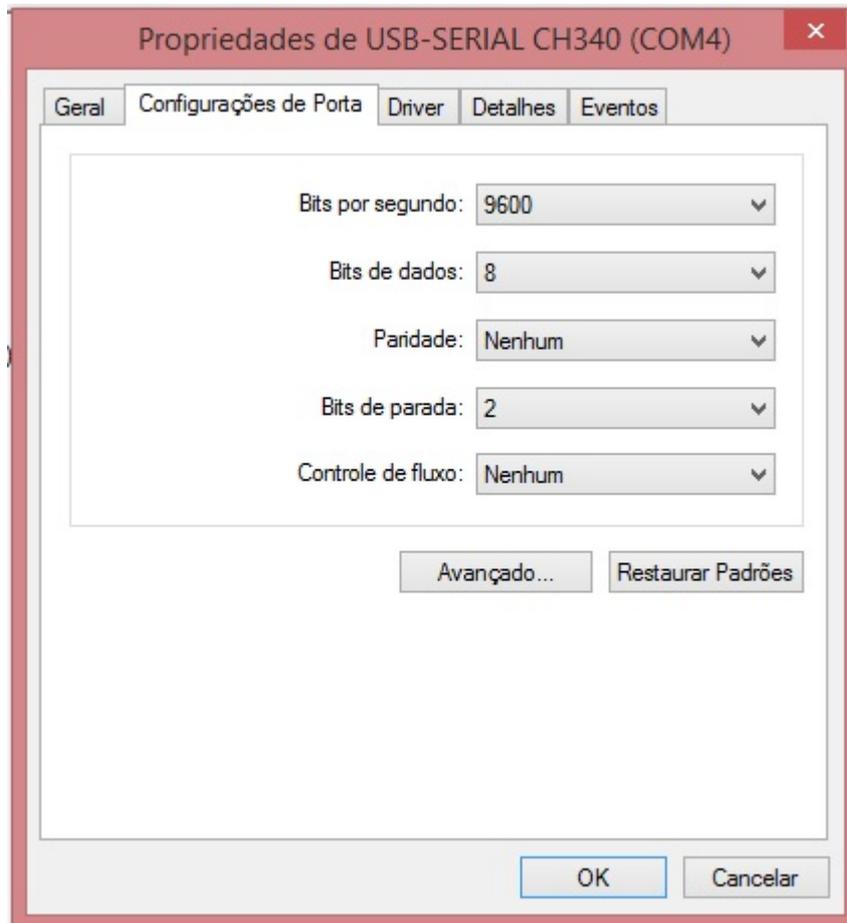


Figura 14: Porta USB-Serial COM4 no PC.(Fonte: AUTOR)

3.3.3 Data Point

Após configurar o Data Source do inversor, é necessário especificar quais dados serão monitorados e/ou controlados pelo ScadaBR no supervisório. Os Data Points são os dados de controle e monitoramento configurados dentro do programa, eles são criados a partir dos valores dos parâmetros do inversor.

Para a criação dos Data Points no ScadaBR, foi necessário compreender como são declarados os parâmetros do inversor e quais parâmetros seriam utilizados para o controle e monitoramento do projeto.

3.3.3.1 Parâmetros de Controle e Monitoramento do Sistema

De acordo com a WEG (2011) sobre o código de função, onde o mestre especifica o tipo de serviço ou função solicitada ao escravo (leitura, escrita, etc.), do protocolo Modbus RTU. Tem-se definido o acesso aos comandos da seguinte maneira no inversor;

No CFW09, os dados relativos aos parâmetros e variáveis básicas estão disponibilizados como registradores do tipo *holding* (referenciados a partir do endereço 40000 ou '4x'). Além destes registradores, o estado do inversor (habilitado/desabilitado, com erro/sem erro, etc.) e o comando para o inversor (girar / parar, girar horário / girar anti-horário, etc.), também podem ser acessadas através de funções para leitura/escrita de "coils" ou bits internos (referenciados a partir do endereço 00000 ou '0x'). (WEG, 2011)

A definição de variáveis básicas, registradores e bits internos é dada pela WEG (2011), como sendo:

- Variáveis Básicas: variáveis internas do inversor, e que somente podem ser acessadas via serial. É possível através das variáveis básicas, por exemplo, alterar referência de velocidade, ler o estado, habilitar ou desabilitar o inversor, etc
- Registrador: nomenclatura utilizada para representar tanto parâmetros quanto variáveis básicas durante a transmissão de dados.
- Bits internos: bits acessados somente pela serial, utilizados para comando e monitoração do estado do inversor.

Ou seja, o inversor de frequência possui duas faixas de registro para acesso aos dados de controle e monitoramento, os primeiros, tanto parâmetros quanto variáveis básicas foram definidos como sendo registradores do tipo *holding*, e o segundo, é possível acessando diretamente bits internos de comando e monitoramento, sendo definido como *coils*. Estas faixas de registro determinam onde os valores serão encontrados pelo mestre.

3.3.3.1.1 Status do Coil

Na Tabela 3, foram relacionados todos os valores dos bits disponíveis no inversor com seus respectivos comandos, seja apenas de leitura ou de controle/acionamento. Nos Coils, cada registrador contém um único e configurável bit, os valores nesta faixa são sempre binários (ou booleanos), ou seja, só possuem dois estados, referidos no sistema com os valores 0 ou 1.

Todos os parâmetros do Coil foram utilizados no projeto.

Tabela 3: Parâmetros do Status Coil

FUNÇÃO	PARÂMETROS	TIPO DE DADOS	CONFIGURÁVEL
Habilita Gira/Para	Bit 100	Binário	Sim - 0 = Gira 1 = Para
Habilita Geral	Bit 101	Binário	Sim - 0 = Inativo 0 = Ativo
Sentido de Rotação	Bit 102	Binário	Sim - 0 = A.H. 1 = Horário
Habilita JOG	Bit 103	Binário	Sim - 0 = Inativo 1 = Ativo
Habilita Local/Remoto	Bit 104	Binário	Sim - 0 = Local 1 = Remoto
Status Gira/Para	Bit 0	Binário	Não
Status Habilita Geral	Bit 1	Binário	Não
Status Sentido Rotação	Bit 2	Binário	Não
Status JOG	Bit 3	Binário	Não
Status Local/Remoto	Bit 4	Binário	Não
Status Erro	Bit 7	Binário	Não

Fonte: Autor

3.3.3.1.2 Registrador Holding

Nos registradores do tipo Holding cada registrador possui 2 bytes (ou uma “palavra”), pode ser com sinal ou sem sinal, e é configurável. Os valores nesta faixa são numéricos (ou analógicos) para os parâmetros do inversor, pois são valores decimais representados no sistema com uma variável de ponto flutuante. São valores numéricos no inversor por exemplo, a velocidade do motor, a corrente nominal, o tempo de rampa, a frequência etc.

Ao contrário da pequena quantidade de parâmetros binários do status coil, a faixa do registrador holding por sua vez abrange todos os parâmetros disponíveis no inversor, que vão desde o P000 até o P538 (existem algumas lacunas entre as numerações dos parâmetros, ou seja, alguns números nesse intervalo não existem como parâmetro).

A Tabela 4 e a Tabela 5 demonstram os parâmetros definidos para o projeto do supervisor. Na Tabela 4 estão os parâmetros de leitura de dados, seja do motor ou do próprio inversor, são variáveis que não podem ser alteradas pelo usuário. Na Tabela 5 estão os parâmetros de regulação do sistema, ou seja, que possuem valores que devem ser ajustados.

Tabela 4: Parâmetros de Leitura

FUNÇÃO	PARÂMETROS	FAIXA DO REGISTRO	TIPO DE DADOS
Velocidade do Motor	P002	Registrador Holding	Numérico
Corrente do Motor	P003	Registrador Holding	Numérico
Tensão do Link CC	P004	Registrador Holding	Numérico
Frequência do Motor	P005	Registrador Holding	Numérico
Torque no Motor	P009	Registrador Holding	Numérico
Último Erro Ocorrido	P014	Registrador Holding	Numérico
Penúltimo Erro Ocorrido	P015	Registrador Holding	Numérico
Contador Horas Energizado	P042	Registrador Holding	Numérico
Contador Horas Habilitado	P043	Registrador Holding	Numérico
Contador KWH	P044	Registrador Holding	Numérico

Fonte: Autor

Tabela 5: Parâmetros de Regulação

FUNÇÃO	PARÂMETROS	FAIXA DO REGISTRO	TIPO DE DADOS
Rampa Aceleração	P100	Registrador Holding	Numérico
Rampa Desaceleração	P101	Registrador Holding	Numérico
Velocidade JOG	P122	Registrador Holding	Numérico
Referência de Veloc. Min.	P133	Registrador Holding	Numérico
Referência de Veloc. Max.	P134	Registrador Holding	Numérico

Fonte: Autor

A Tabela 6 é referente aos parâmetros especiais de controle do PID que podem ser configurados a partir do parâmetro P520, estes parâmetros foram habilitados através do parâmetro P203=1, opção Regulação PID. Dentre os parâmetros especiais PID, foram selecionados para compor o supervisor os parâmetros de ajuste dos ganhos proporcional, integral e derivativo, o ajuste de Setpoint e a opção de escolha entre o setpoint automático ou manual.

Tabela 6: Parâmetros Especiais PID

FUNÇÃO	PARÂMETROS	AJUSTE
Seleção de Funções Especiais	P203	0 = Nenhuma 1 = Regulador PID
Ganho Proporcional	P520	1.000 ^a
Ganho Integral	P521	0.043 ^a
Ganho Diferencial	P522	0.000 ^a
Setpoint PID	P134	0.0% ^a
Tipo de Ação PID	P525	0 = Direto 1 = Reverso
Ajuste Automático de P525	P536	0 = Ativo 1 = Inativo
^a Valores pré-ajustados de fábrica.		

Fonte: Autor

3.3.3.2 Configuração dos Data Points

Tendo-se definido todos os parâmetros anteriormente, foi possível criar os data points (ou tags) um a um, nomeando-os a partir das suas respectivas funções e configurando-os de acordo com suas faixas de registro.

Data points						
Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)	
Ajuste Setpoint PID	Numérico		1	Registrador holding 536		
Ativar PID	Numérico		1	Registrador holding 203		
Contador de Horas Energ.	Numérico		1	Registrador holding 42		
Contador de Horas Habilitado	Numérico		1	Registrador holding 43		
Contador KWH	Numérico		1	Registrador holding 44		
Corrente Motor	Numérico		1	Registrador holding 3		
Frequência Motor	Numérico		1	Registrador holding 5		
Ganho Diferencial PID	Numérico		1	Registrador holding 522		
Ganho Integral PID	Numérico		1	Registrador holding 521		
Ganho Proporcional PID	Numérico		1	Registrador holding 520		
Gira/Para Motor	Binário		1	Status do coil	100	
Habilita Geral	Binário		1	Status do coil	101	
Habilita JOG	Binário		1	Status do coil	103	
Habilita local/Remoto	Binário		1	Status do coil	104	
Habilita Sentido	Binário		1	Status do coil	102	
Penúltimo Erro	Numérico		1	Registrador holding 15		
Rampa Aceleração	Numérico		1	Registrador holding 100		
Rampa Desaceleração	Numérico		1	Registrador holding 101		
Ref. Veloc. Max.	Numérico		1	Registrador holding 134		
Ref. Veloc. Min.	Numérico		1	Registrador holding 133		
Setpoint PID	Numérico		1	Registrador holding 525		
Status ERRO	Binário		1	Status do coil	7	
Status Gira/Para Motor	Binário		1	Status do coil	0	
Status Habilita Geral	Binário		1	Status do coil	1	
Status JOG	Binário		1	Status do coil	3	
Status Local/Remoto	Binário		1	Status do coil	4	
Status Sentido Motor	Binário		1	Status do coil	2	
Tensão do Link CC	Numérico		1	Registrador holding 4		
Torque Motor	Numérico		1	Registrador holding 9		
Último Erro	Numérico		1	Registrador holding 14		
Velocidade JOG	Numérico		1	Registrador holding 122		
Velocidade Motor	Numérico		1	Registrador holding 2		

Figura 15: Lista de Data Points no ScadaBR.(Fonte: AUTOR)

Na Figura 16, tem-se um exemplo de status coil (lado esquerdo) referente ao comando binário Habilitar Geral, as configurações que seguem são as mesmas para todos os outros coils diferenciando-se apenas na opção *Configurável*. Na mesma figura, tem-se um exemplo de registrador holding (lado direito) referente ao parâmetro de leitura da corrente no motor, os parâmetros do tipo holding assim como os do status coil possuem a mesma configuração, diferenciando-se um dos outros apenas na opção *Configurável*.

Detalhes do data point	
Nome	Habilita Geral
Export ID (XID)	DP_103976
Id do escravo	1
Faixa do registro	Status do coil
Tipo de dados modbus	Binário
Offset (baseado em 0)	101
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input checked="" type="checkbox"/>
Multiplicador	1
Aditivo	0

Detalhes do data point	
Nome	Corrente Motor
Export ID (XID)	DP_549683
Id do escravo	1
Faixa do registro	Registrador holding
Tipo de dados modbus	Inteiro de 2 bytes sem sinal
Offset (baseado em 0)	3
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input type="checkbox"/>
Multiplicador	0.1
Aditivo	0

Figura 16: Data Point de Status Coil e Data Point de Registrador Holding.(Fonte: AUTOR)

Alguns pontos importantes no entendimento das configurações dos DataPoint são:

- O campo Export ID(XID) é o nome que o ScadaBR atribui automaticamente para uso interno do software.
- A ID do escravo é correspondente ao endereço do escravo na rede, neste caso o inversor com endereço 1.
- A Faixa de Registro permite a escolha entre 4 faixas, sendo duas saídas (Coils e Holding) e duas entradas (Status de Entrada e Registrador de Entrada), no projeto usa-se apenas as saídas tendo em vista que o ScadaBR é o mestre na rede.
- O campo Tipos de Dados, no status coil é obrigatoriamente binário, não havendo outra opção para escolha. No holding, há 20 opções para escolha do tipo de dado (Binário, Inteiro de 2 bytes sem/com sinal, Inteiro de 4 bytes sem/com sinal, Float, BCD, etc.)
- Offset (baseado em 0), significa que sua contagem inicia em 0 (0-indexed), portanto, os parâmetros podem ser “setados” com seus valores originais, por exemplo, um parâmetro P006 é setado no offset como 6.
- O campo Configurável, determina se o parâmetro ajustado é de escrita ou apenas de leitura, e serve para tornar um ponto, que normalmente seria configurável (de acordo com sua faixa), em não configurável.
- O campo Multiplicador, é usado para determinar a escala de leitura de alguns parâmetros fazendo conversões quando necessário, no caso do inversor alguns dados de leitura e escrita, quando configurados no Scadabr, possuem o valor multiplicado por 10, ou seja, uma corrente que seria de 0.8A é mostrada como 8A, ou uma rampa de aceleração

configurada para 10.0s mostra 100.0s. Neste caso, usou-se o multiplicador para ajustar os valores de acordo com o valor real que se espera do sistema.

3.3.3.3 Watch List

Segundo o próprio software ScadaBR, uma Watch List é uma lista dos data points nos quais você tem interesse em observar em conjunto. Os dados são atualizados automaticamente — você não precisa atualizar a página (F5). (Inclusive, o sistema opera de maneira mais eficientemente caso você não atualize).

Portanto, após o cadastrado dos Data Points no sistema, para que o usuário possa interagir, é necessário escolher a opção Watch List. Na aba de Watch List os Data Points cadastrados encontram-se à esquerda, para ativá-los basta clicar sobre cada um e adicioná-los à Watch List como na Figura 17.



Figura 17: Watch List do ScadaBR.(Fonte: AUTOR)

É importante salientar que pela Watch List já é possível ter o controle total do inversor, monitorando os valores gerados pelo sistema e regulando os ajustes da maneira desejada.

3.4 INTERFACE PARA SISTEMA SUPERVISÓRIO

A criação de uma interface para o projeto tem o intuito de facilitar o controle e a obtenção de dados do sistema, fazendo com que todo o processo funcione de maneira dinâmica e possua

uma fácil interação com o usuário. Este sistema chamado de supervisorio, foi criado a partir da função Representações Gráficas, no ScadaBR.

A função Representações Gráficas permite ao usuário criar visualizações mais elaboradas dos dados, por meio de diversos componentes e funções características de sistemas supervisorios típicos.

O acesso à função se inicia com a criação de um nome pelo qual a interface será referenciada dentro das listas de visualizações gráficas, supondo que seja criada mais de uma janela de interface. Em sequência é possível inserir uma imagem de fundo sobre a qual os componentes da interface ficarão superpostos. Por último, a opção Acesso Anônimo setado no projeto como none (nenhum) significa que usuários anônimos não podem acessar a visualização.

A opção Componentes é a principal ferramenta na construção do supervisorio, nesta opção podem ser adicionados uma ação funcional por meio da seleção do DataPoint correspondente, podem ser inseridos por exemplo, botões (tipo toggle, tipo pulso), caixas contendo dados de leitura, imagens animadas com extensão GIF, gráficos animados com indicadores (ponteiros, nível, etc.). O scadabr além dos aplicativos personalizados já instaladas no software, permite o desenvolvimento de novas funções de controle, ou de novas opções de animação em qualquer linguagem de programação atual (JAVA, C++, PHP, JavaScript, etc.). Alguns comandos no supervisorio foram configurados por meio de script em JavaScript².

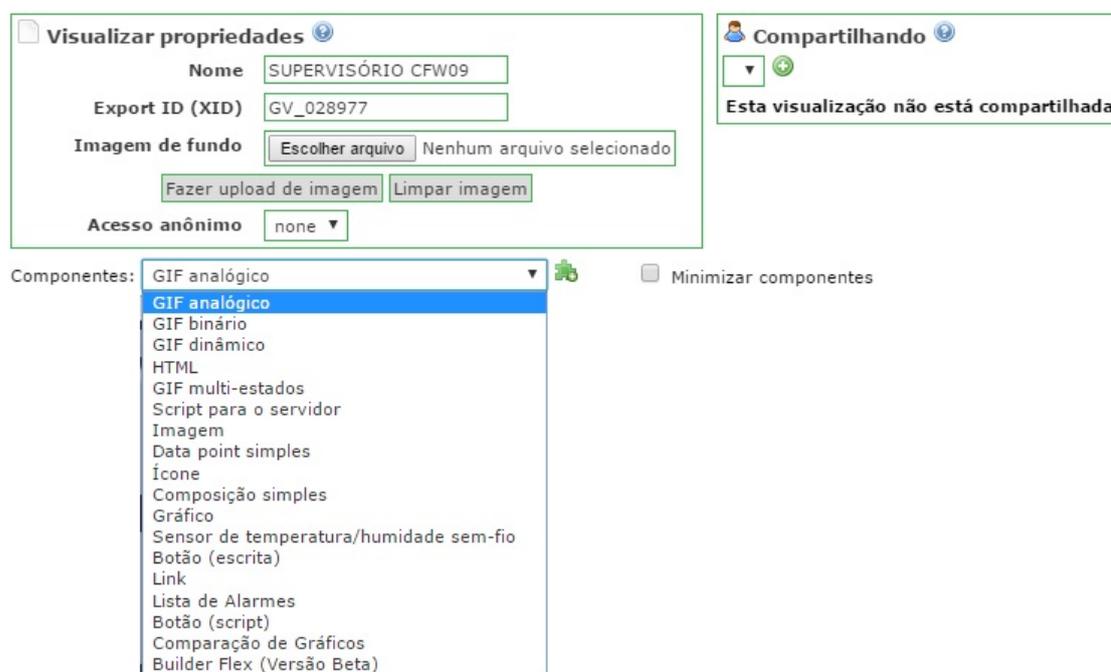


Figura 18: Criação da Interface no ScadaBR.(Fonte: AUTOR)

² É uma linguagem de script interpretada com base em objetos(trata todos os elementos da página como objetos distintos, facilitando a tarefa da programação),(...) usa uma sintaxe semelhante à de C e oferece suporte a construções estruturadas, como if...else, for e do...while. (Fonte: Microsoft, 2016)

3.4.1 Configuração dos Componentes De Visualização

O princípio de funcionamento de todos os componentes é basicamente o mesmo, primeiro o componente escolhido na caixa de opções é inserido na representação gráfica, segundo deve-se escolher o Data Point a qual o componente estará associado e então, após isto, realizar as configurações de processamento gráfico da imagem a ser exibida. (pode-se observar as opções de escolha componentes na Fig. 18)

Os Data Points responsáveis por habilitar/desabilitar o inversor, girar/parar o motor e inverter o giro entre horário/anti-horário foram configuradas através do componente **Script Para o Servidor**, esse componente permite que os comandos binários funcionem a partir de um botão animado, ao clicar na imagem do botão, muda-se o estado do comando e consequentemente a imagem do botão. Os scripts utilizados na configuração encontram-se no Apêndice A.

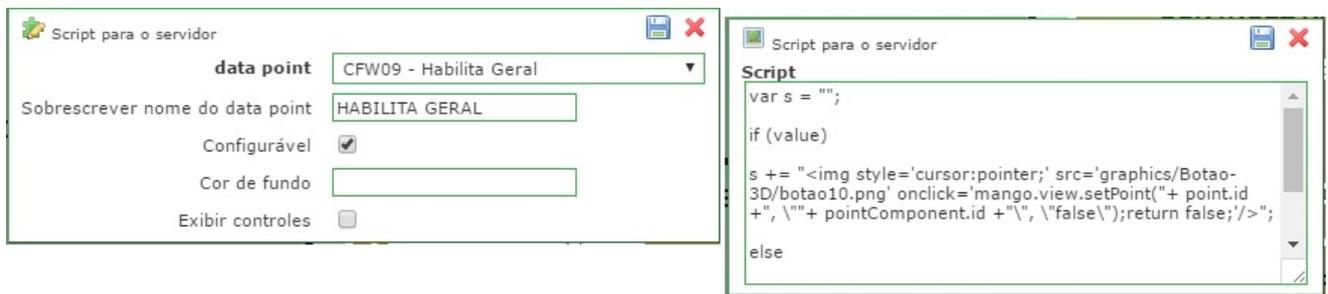


Figura 19: Janelas de configuração do componente Script Para Servidor.(Fonte: AUTOR)

Na seleção de modos Local/Remoto e acionamento de JOG, apesar de também serem comandos binários foi escolhido outro componente para representá-los, o **Botão de Escrita**, esta função funciona apenas com comandos binários, o botão permite escrever uma descrição para os modos, facilitando o entendimento do usuário sobre os comandos.

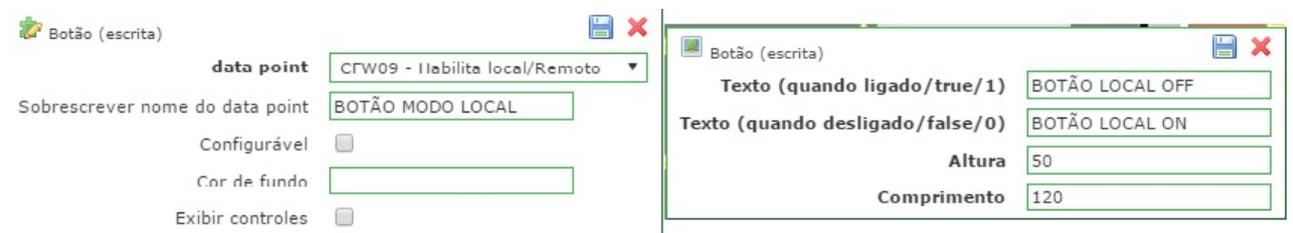


Figura 20: Janelas de configuração do componente Botão de Escrita.(Fonte: AUTOR)

No componente **GIF Binário**, alguns alarmes foram colocados na interface para sinalizar o que está habilitado ou se está habilitado, na seleção de modos Local/Remoto foi colocado dois leds, quando um está piscando com a cor verde (valor 0) o outro está com a cor vermelha

(valor 1), sinalizando assim que o primeiro está ativo e o segundo inativo. Também foi colocado um sinal luminoso para indicar se o JOG está em funcionamento ou não, e por último um LED na cor cinza para indicar algum erro no inversor, no momento do erro o LED muda para a cor vermelha.



Figura 21: Janelas de configuração do componente GIF Binário.(Fonte: AUTOR)

Apenas 1 Data Point foi configurado com a componente **GIF Dinâmico**, sendo o Data Point referente a leitura de velocidade do motor, neste componente o indicador do valor é um desenho dinâmico em que se pode visualizar um ponto numérico, pode ser um termômetro, um tanque de nível, ou como no projeto, um velocímetro.



Figura 22: Janelas de configuração do componente GIF Dinâmico.(Fonte: AUTOR)

O componente **Data Point Simples**, foi o mais utilizado na representação gráfica da interface, este componente está relacionado com todos os parâmetros de leitura e escrita, todos os Data Points de leitura (do motor e do inversor) foram configurados através desta componente, os Data Points de regulação (PID, Rampas, Velocidade do Motor e Tempo de JOG) também foram configurados como Data Point Simples. O componente abre uma pequena caixa contendo a informação requerida ou uma mesma caixa que permite a escrita.

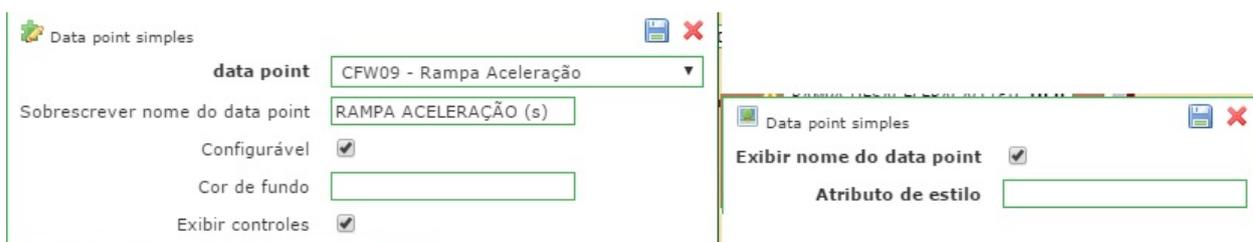


Figura 23: Janelas de configuração do componente Data Point Simples.(Fonte: AUTOR)

Foram criadas em modo de Script para Servidor 4 caixas de regulação, (1 para velocidade, 1 para rampa de aceleração, 1 para rampa de desaceleração e 1 para setpoint de PID), estas caixas possuem uma faixa de valores num modo de barra rolante, permitindo ao usuário “setar” o comando dos parâmetros citados de maneira alternativa ao Data Point Simples, a função pode ser observada na Figura 24.

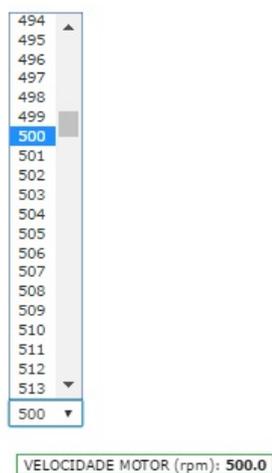


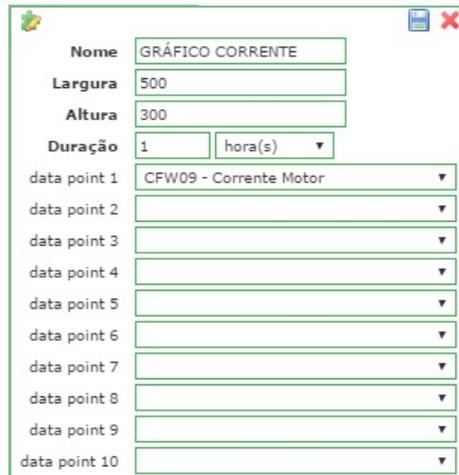
Figura 24: Barra rolante de regulação alternativa.(Fonte: AUTOR)

3.4.2 Configuração de Interface Gráfica

Além da interface de controle e leitura de dados do supervisor, também foi criada uma representação gráfica para o monitoramento do sistema através de gráficos. Uma segunda tela de representação gráfica foi necessária pois a primeira não poderia comportar os gráficos desejados para o projeto. Na segunda interface foram criados 4 gráficos a partir do componente **Gráfico**, os gráficos são: Gráfico de Corrente do Motor, Gráfico de Velocidade do Motor, Gráfico de Torque do Motor e Gráfico de Frequência do Motor.

A geração de gráficos é essencial para monitoramento de um sistema, as curvas nos gráficos são atualizadas sempre que o motor muda de estado, para análise do comportamento das variáveis e da relação motor/inversor foi feito um experimento didático, com o inversor funcionando por 2 horas e variando a sua velocidade (rpm) diversas vezes, neste intervalo de tempo.

Os gráficos com o comportamento das curvas das variáveis analisadas estão presentes no Apêndice B.



Nome	GRÁFICO CORRENTE
Largura	500
Altura	300
Duração	1 hora(s)
data point 1	CFW09 - Corrente Motor
data point 2	
data point 3	
data point 4	
data point 5	
data point 6	
data point 7	
data point 8	
data point 9	
data point 10	

Figura 25: Janela de configuração do componente Gráficos.(Fonte: AUTOR)

Além da opção de criar gráficos na interface do supervisão, o ScadaBR ainda disponibiliza a opção de criar modelos de relatórios dos data points que podem ser agendados para serem gerados automaticamente ou não, e que podem ser enviados por email.

Há também outra forma de monitorar o data point, na opção chamada detalhes do data point, através da watch list, é possível encontrar dados de estatísticas, gráfico, histórico etc. Essa opção é válida principalmente quando se quer analisar os dados de um determinado período, sendo ele longo ou curto.

4 RESULTADOS E DISCUSSÕES

4.1 INTERFACE DO SUPERVISÓRIO SCADA

A interface do supervisório foi construída a partir das informações detalhadas nas seções de desenvolvimento, a posição das componentes do supervisório foram estabelecidas de maneira bem acessível e intuitiva de modo a facilitar a compreensão do sistema pelos usuários e tornar o processo mais eficiente. Partindo-se deste princípio, a interface foi dividida em 5 blocos com funções distintas.

- O primeiro bloco chamado de **Acionamentos**, foi ajustado com parâmetros que exercem comandos como, habilitar/desabilitar inversor, girar/parar motor, alternar entre os modos local/remoto e acionar/parar JOG. Todos os comandos deste bloco são acionados por um botão tipo toggle.
- No segundo bloco, chamado de **Controle do Sistema**, foram colocados os parâmetros referentes à regulação das funções, ou seja, são os parâmetros que permitem ajustes através da escrita, são eles, o ajuste da velocidade do motor em rpm, o tempo de rampa (aceleração e desaceleração) em segundos e a velocidade de JOG em rpm. A função de inversão de giro do motor é acionada por botão e não permite escrita, porém, foi incluída neste bloco por se tratar de um parâmetro de controle.
- O terceiro bloco chamado de **Monitoramento** possui os parâmetros de leitura dos dados do sistema, este bloco foi dividido em duas partes, a primeira parte é constituída dos parâmetro de leitura do motor, sendo alguns como exemplo, a leitura da corrente, da energia consumida, da frequência etc. E a segunda parte, sendo constituída pelos parâmetros de leitura do inversor, composto por um contador de horas energizado, um contador de tempo habilitado e um medidor de tensão no barramento(link) de corrente contínua.
- O quarto bloco chamado de **Controlador PID**, como o próprio nome já diz, é composto por parâmetros responsáveis pelos controles PID do sistema, neste bloco o usuário deve ativar o comando para habilitar os parâmetros especiais PID, fazendo isso é possível realizar o ajuste através dos comandos de controle proporcional, integral e derivativo. Também é possível ajustar o setpoint, dado em %, estando no modo manual ou deixa-lo no modo automático.
- No quinto e último bloco chamado de **Erros do Sistema**, foram inseridos os parâmetros de alarme de erro do inversor, os últimos dois erros ocorrido no sistema é mostrado na interface através de um valor numérico, ao todo o inversor dispõe de 27 tipos de erros que podem se consultados pela sua numeração no manual de usuário do equipamento.

Os de planos de fundo do supervisório, tanto da interface SCADA quanto da interface gráfica, foram criados através de um software de edição de imagens e carregados posteriormente no ScadaBR para que fossem dispostos os componentes da interface sobre a imagem de fundo.

Abaixo as Figuras 26 e 27 são, respectivamente, do supervisório de controle e monitoramento, e dos gráficos do sistema.



Figura 26: Sistema SCADA do projeto.(Fonte: AUTOR)

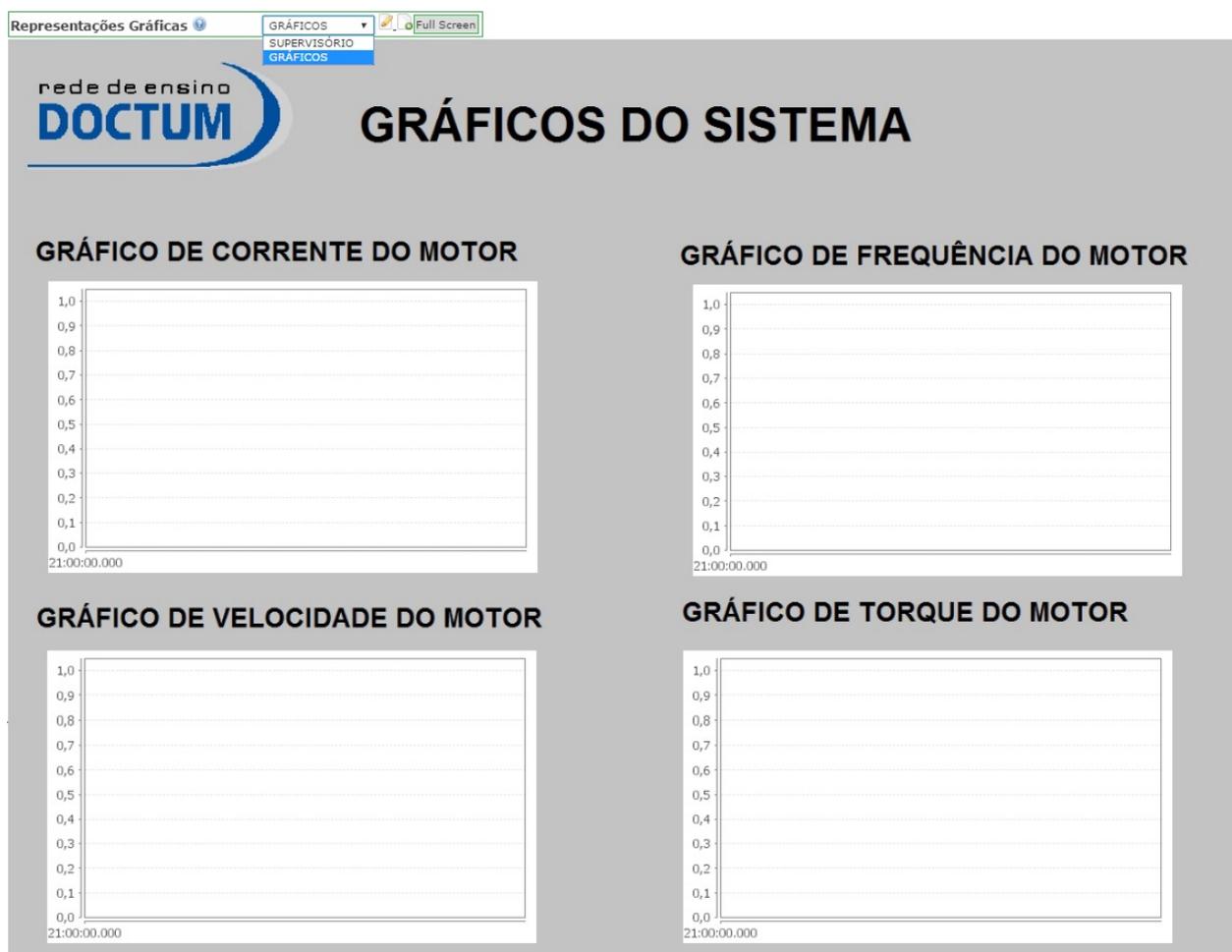


Figura 27: Interface Gráfica do projeto.(Fonte: AUTOR)

4.2 ERROS DE COMUNICAÇÃO

Apesar do desempenho satisfatório do sistema, alguns erros (chamados de Exception) relacionados a comunicação entre os dispositivos foram relatados durante todo o projeto.

Uma teoria plausível para alguns dos erros é o cenário que se percebe em relação a comunicação mestre-escravo da seguinte forma, a atualização ou "polling" acontece da maneira como foi configurado no baudrate (bps) e no período de atualização, neste ponto percebe-se que existe uma "colisão" de dados entre os dados coletados da interface gráfica (escrita) e as operações de "polling" (leitura). Essa colisão entre a troca de dados dados é observada principalmente quando a uma sobreposição de operações e o resultado disso é uma serie de alertas de erro relacionados a comunicação modbus e aos parâmetros configurados nos datapoints.

Como exemplo prático do problema, temos os datapoints de comando e leitura binários, quando é ativado apenas os datapoints de comando e os de leitura respectivos permanecem desabilitados não se nota o erro de comunicação. Porém ao ativar ambos os data points é causado

um conflito e é disparado o alarme de erros.

O problema, no entanto, não é apenas com os datapoints de coils, os datapoints que são holding também geram os mesmos erros de comunicação. Um exception mais recorrente é a mensagem Illegal Data Address (Endereço de Dados Ilegal) referente aos parâmetros do inversor. Uma possível causa do problema possa ser devido a falta de ordem numérica, neste caso o request do software provavelmente é feito para ler em ordem, ou seja, não é possível "pular" offset. No entanto, não foram encontradas nenhuma solução plausível e efetiva para os problemas, seja no fórum de dúvida e suporte do ScadaBR, ou seja nos manuais do ScadaBR e do inversor de frequência.

A consequência dos erros de comunicação pode ser vista na Fig. 17, nela pode-se observar a tela de watch list com alguns datapoints habilitados, em todos eles é possível ver a mensagem "**Valor do ponto pode não ser confiável**", esta mensagem não permite que os data points de leitura sejam atualizados automaticamente, sendo necessário atualizar cada um de forma manual após cada mudança de estado no sistema.

Contudo, o problema com a atualização dos dados não interfere no funcionamento do sistema, todos os parâmetros "setados" através do ScadaBR são lidos e aceitos pelo inversor independentemente do erro retornado. Outro fator interessante observado no sistema é que somente os parâmetros especiais de PID, referenciados a partir do parâmetro P520, não relataram erro em momento algum do projeto.

4.3 ANÁLISE DOS COMPONENTES DA INTERFACE

O ScadaBR por ser opensource, com linguagem de código aberta, é livre para que qualquer usuário que faça parte da comunidade possa contribuir (sem fins lucrativos) com atualizações para o melhoramento do sistema, partindo deste ponto, nota-se que em relação às configurações gráficas existem ainda pouquíssimos recursos gráficos em relação à quantidade de funções que podem ser exercidas no programa. Com isto, observa-se que muitas configurações e criações do conteúdo gráfico acabam sendo realizadas pelos usuários através de recursos externos ao programa, o que torna a configuração complexa e trabalhosa.

É importante mencionar que a cada atualização do ScadaBR, novas funções gráficas são inseridas no programa, porém se comparado a alguns softwares pagos ou até mesmo alguns gratuitos, o ScadaBR demonstra esse ponto negativo.

No projeto em questão, a construção dos componentes gráficos da interface (botões, alarmes, imagens, etc.) foram todos criados dentro dos próprios limites do programa, daquilo que o software oferece pronto ao usuário, sem recorrer a métodos externos. De acordo com o proposto para o projeto, ficou-se entendido que os componentes gráficos só agregam valor estético à interface do supervisor, ou seja, não provocam nenhuma interferência no processo

de controle e monitoramento.

4.4 ANÁLISES DO COMPORTAMENTO DO SISTEMA

Levando em consideração que o objetivo deste trabalho é abordar o funcionamento, de forma didática, de um sistema SCADA, bem como do inversor de frequência e da sua relação com motor. Tem-se que os acionamentos e comandos do inversor para com o motor foram todos realizados com este trabalhando a vazio, é óbvio dizer que numa situação prática isso não acontece pois o motor sempre trabalhará com alguma carga em seu eixo.

Com isto, devido falta de carga no motor não foi possível analisar algumas questões, por exemplo, analisar o inversor atuando de maneira a controlar e regular o sistema sobre uma variação de carga, analisar a relação torque x velocidade, tensão x frequência. Não foi possível acrescentar parâmetros com funções mais avançadas de controle, como, parâmetro de compensação de escorregamento, boost de torque, tempo e/ou velocidade de acomodação da partida, dentre outros. Também não foi possível analisar o comportamento dos controladores PID, pois não havia nenhum processo para controle no sistema, porém este comando foi habilitado e permitirá futuros trabalhos que venham a utilizar destes controles.

5 CONCLUSÃO

O presente trabalho alcançou os objetivos propostos de maneira satisfatória apesar dos erros relatados, o supervisório desenvolvido trouxe vantagens sobre a parametrização via IHM, pois melhorou o gerenciamento das informações e o controle das funções. O ScadaBR possui a grande vantagem de ser um software gratuito e aberto, isto o torna acessível tanto aos pequenos usuários, que o utilizam de maneira tímida, quanto para os grandes processos de controle seja na indústria ou na geração de energia por exemplo.

O software desenvolvido pode e deve ser utilizado para fins didáticos devido ao seu fácil manuseio e fácil entendimento das telas gráficas. O interessante que o usuário ainda tem a sua disposição, além das funções pré disponíveis, a opção de acrescentar novos parâmetros ao sistema e à interface de acordo com a sua necessidade.

Também foi possível observar com base nos dados analisados, que o fato do sistema se comportar de maneira eficaz, com confiabilidade e flexibilidade, justifica porque estes tipos de sistemas se tornaram uma peça fundamental no bom funcionamento dos processos industriais, gerando aprimoramento e conseqüentemente qualidade na produção. É possível observar ainda que a demanda por esse tipo de software só tende a aumentar ao ponto que os sistemas SCADA se tornem uma ferramenta essencial para todo tipo controle de processo feito atualmente.

5.1 SUGESTÕES PARA TRABALHOS FUTUROS

Como sugestão para trabalhos futuros, pode-se citar como possibilidades:

- Criar novas interfaces na qual cada uma pertencesse a um tipo de controle do inversor, por exemplo, este projeto foi feito em controle escalar V/F, a proposta é criar outra interface para o controle vetorial sensorless, outra para VVW (Voltage Vector Weg), e assim poder alternar entre os controles, visto que cada controle possui seus parâmetros exclusivos.
- Simular cargas variáveis no motor e analisar o comportamento do inversor de frequência para um monitoramento mais dinâmico e real.
- Criar um sistema controle seja de fluxo, de nível, temperatura e controlar via de parâmetros ajuste de PID.
- Procurar uma solução para as possíveis causas de erro na comunicação modbus do sistema.
- Melhorar a estética da interface gráfica utilizando de meios externos para a edição, como a API HMI Builder, que permite a criação de "widgets" flash.

6 REFERÊNCIAS BIBLIOGRÁFICAS

ALFA INSTRUMENTOS. **Protocolo de Comunicação Modbus RTU/ASCII**, Alfa Instrumentos, 2000.

CAPELLI, A. **Inversores de Frequência**, Revista Mecatrônica Atual nº 2. 2002.

FUNDAÇÃO CENTROS DE REFERÊNCIA EM TECNOLOGIAS INOVADORAS; FINANCIADORA DE ESTUDOS E PROJETOS; SERVIÇO DE APOIO ÀS MICRO E PEQUENAS EMPRESAS; CONSELHO NACIONAL DE DESENVOLVIMENTO CIENTÍFICO E TECNOLÓGICO. **ScadaBR: Manual do software**. Rio de Janeiro, 2010. 70 p.

FRANÇA JÚNIOR, M.C; PEREIRA, V.M; BATISTA, E.A. **Desenvolvimento de Software com Propósito Educacional para Parametrização, Monitoramento e Supervisão de Conversores de Frequência**. International Conference on Engineering and Computer Education - ICECE, 2011.

MASCHERONI, J.M.; LICHTBLAU, M.; GERARDI, D. **Guia de Aplicação de Inversores de Frequência**. WEG Automação, 2ª Edição, Florianópolis, 2005. 192p.

MACHADO MARTINS, G. **Princípios de automação industrial**. Santa Maria: UFSM. 2012

MCA SISTEMAS. **Soluções ScadaBR**.

Disponível em: <http://www.mcasistemas.com.br/solucoesscadabr.html>. Acesso em 30/11/2016.

NATIONAL INSTRUMENTS CORPORATION. **Explicando a Teoria PID**, 2011.

Disponível em: <http://www.ni.com/white-paper/3782/pt/>. Acesso em 30/11/2016

NATIONAL INSTRUMENTS CORPORATION. **O protocolo Modbus em detalhes**, 2014. 2014

Disponível em: <http://www.ni.com/white-paper/52134/pt/> Acesso. 30/11/2016

NATIONAL INSTRUMENTS CORPORATION. **Uma rápida comparação das Interfaces de Comunicação Serial RS-232, RS-422, e RS-485**, 2013.

Disponível em: <http://digital.ni.com/public.nsf/allkb/DE153F74C4BF3AD8862576AB006C1AAF>.

Acesso em: 30/11/2016

SEIXAS, Constantino. **Protocolos Orientados a Caracter**. UFMG – Departamento de Engenharia Eletrônica, Minas Gerais, 2003.

SILVA. M.R; OLIVEIRA, A; CARMO, M.J. **Importância da Ferramenta SCADABR Para o Ensino em Engenharia**. Congresso Brasileiro de Educação em Engenharia - COMBENGE, 2013.

WEG. **Manual do Inversor de Frequência: Série: CFW-09**. Software: versão 4.4x. Jaraguá do Sul, 2011.

WEG. **Guia de especificação de motores elétricos**. Jaraguá do Sul, 2016.

WEG. **Motores de indução alimentados por inversores de frequência PWM**. Jaraguá do Sul, 2016.

7 APÊNDICE A

Comandos usados para criar os botões na opção Script para Servidor.

Botão Liga/Desliga

```
var s = "";

if (value)

s += "<img style='cursor:pointer;' src='graphics/Botao-3D/botao10.png'
onclick='mango.view.setPoint("+ point.id +", \""+
pointComponent.id +"\", \"false\")';return false;'/>";

else

s += "<img src='graphics/Botao-3D/botao20.png' onclick='mango.view.setPoint
("+ point.id +", \""+ pointComponent.id +"\", \"true\")';return false;'/>";

return s;
}
```

Botão Horário

```
var s = "";

if (value)

s += "<img style='cursor:pointer;' src='graphics/Horario-on-off/Horario Off.gif'
onclick='mango.view.setPoint("+ point.id +", \""+ pointComponent.id +"\",
\"false\")';return false;'/>";

else

s += "<img src='graphics/Horario-on-off/Horario On.gif'
onclick='mango.view.setPoint("+ point.id +", \""+ pointComponent.id +"\",
\"true\")';return false;'/>";

return s;
```

Botão AntiHorário

```
var s = "";

if (value)

s += "<img style='cursor:pointer;' src='graphics/Anti-horario-on-off/Anti Horario On.gif'
onclick='mango.view.setPoint("+ point.id +",
\"" + pointComponent.id +"\", \"false\")';return false;'/>";

else

s += "<img src='graphics/Anti-horario-on-off/Anti Horario Off.gif'
onclick='mango.view.setPoint("+ point.id +",
\"" + pointComponent.id +"\", \"true\")';return false;'/>";
```

```
return s;
```

Barra Rolante - Velocidade do Motor (rpm)

```
var min = 90;
var max = 1800;
var lead = ""; //for leading zero or space
var s="";

s+="
```

Barra Rolante - Rampa Aceleração/Desaceleração

```
var min = 0;
var max = 60;
var lead = ""; //for leading zero or space
var s="";

s+="
```

8 APÊNDICE B

Gráficos gerados com valores a partir dos valores "setados" na velocidade. O gráfico gera a curva apenas quando o estado muda.

Graf. Corrente

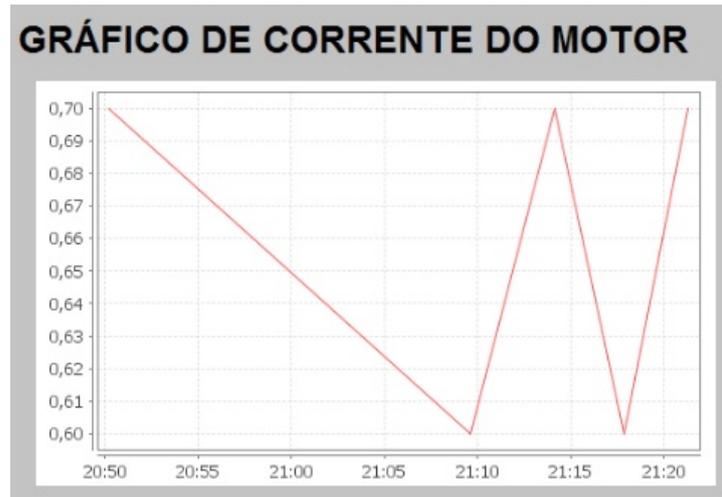


Figura 28: FONTE: AUTOR

Graf. Frequência



Figura 29: FONTE: AUTOR

Graf. Velocidade

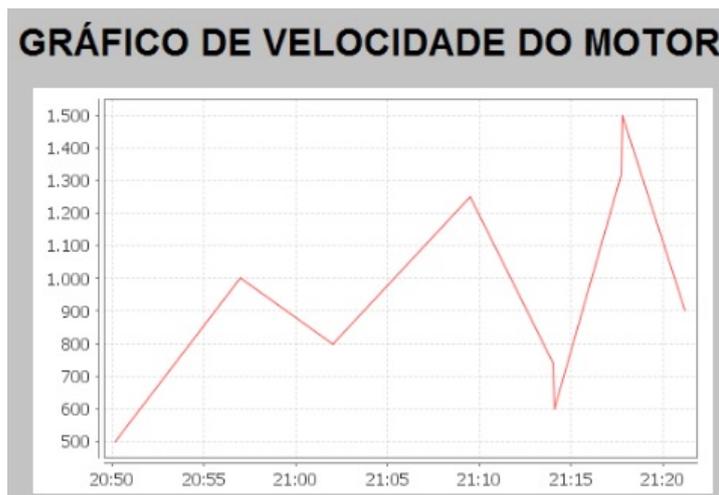


Figura 30: FONTE: AUTOR

Graf. Torque

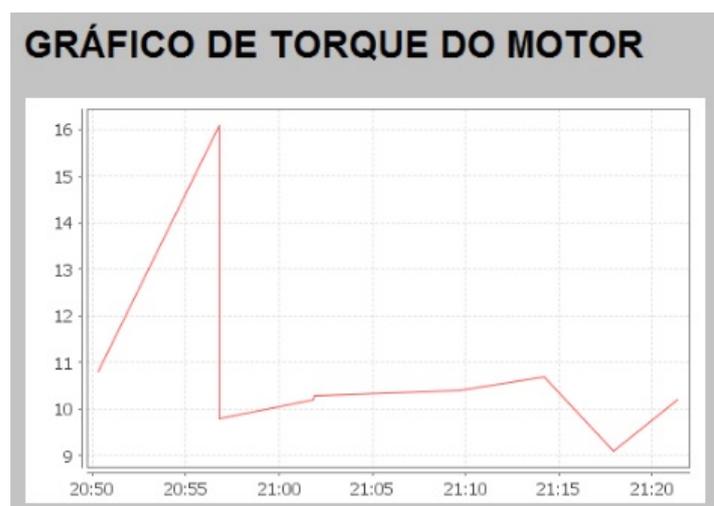


Figura 31: FONTE: AUTOR