

INSTITUTO FEDERAL  
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
Bahia

Campus  
Vitória da Conquista



# **COORDENAÇÃO DE ENGENHARIA ELÉTRICA - COEEL**

## **PROJETO FINAL DE CURSO - PFC**

Controle de acionamento e velocidade de um motor de indução trifásico via IHM remota utilizando o protocolo Modbus

**MARIANA OLIVEIRA DO ROSÁRIO**

Vitória da Conquista-BA  
30 de novembro de 2023

**MARIANA OLIVEIRA DO ROSÁRIO**

**Controle de acionamento e velocidade de um motor  
de indução trifásico via IHM remota utilizando o  
protocolo Modbus**

Projeto Final de Curso apresentado ao Curso de Graduação em Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia da Bahia, *campus* Vitória da Conquista, como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica.

**Orientador:** Dr. Wesley de Almeida Souto

Vitória da Conquista-BA  
30 de novembro de 2023

FICHA CATALOGRÁFICA ELABORADA PELO SISTEMA DE BIBLIOTECAS DO IFBA, COM OS  
DADOS FORNECIDOS PELO(A) AUTOR(A)

R789c Rosário, Mariana Oliveira do

Controle de acionamento e velocidade de um motor de indução trifásico via IHM remota utilizando o protocolo Modbus: / Mariana Oliveira do Rosário; orientador Prof. Dr. Wesley de Almeida Souto -- Vitória da Conquista : IFBA, 2023.

65 p.

Trabalho de Conclusão de Curso (Engenharia Elétrica) -- Instituto Federal da Bahia, 2023.

1. CODESYS. 2. Modbus. 3. Conversor de frequência. 4. Motor de indução trifásico. 5. RS-485. I. de Almeida Souto, Prof. Dr. Wesley, orient. II. TÍTULO.

CDD/CDU

## TERMO DE ACEITE DE ORIENTAÇÃO PFC

### FOLHA DE APROVAÇÃO PFC

#### CONTROLE DE ACIONAMENTO E VELOCIDADE DE UM MOTOR DE INDUÇÃO TRIFÁSICO VIA IHM REMOTA UTILIZANDO O PROTOCOLO MODBUS

#### MARIANA OLIVEIRA DO ROSÁRIO

A presente Monografia de Projeto Final de Curso (PFC), apresentada em sessão realizada em **30 de novembro de 2023**, foi avaliada como adequada para a obtenção do Grau de Engenharia Eletricista, julgada **aprovada** em sua forma final pela Coordenação do Curso de Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia da Bahia, Campus Vitória da Conquista.

#### BANCA EXAMINADORA

Prof. Dr. Wesley de Almeida Souto (Orientador)

IFBA, *campus* Vitória da Conquista

Prof. Esp. Pablo Martins de Oliveira

IFBA, *campus* Vitória da Conquista

Prof. Esp. Danilo Brito Almeida

IFBA, *campus* Vitória da Conquista

Vitória da Conquista - Bahia

Em 28 de novembro de 2023.



Documento assinado eletronicamente por **WESLEY DE ALMEIDA SOUTO, Professor Efetivo**, em 30/11/2023, às 19:29, conforme decreto nº 8.539/2015.



Documento assinado eletronicamente por **DANILO BRITO ALMEIDA, Professor Efetivo**, em 30/11/2023, às 19:35, conforme decreto nº 8.539/2015.

---



Documento assinado eletronicamente por **PABLO MARTINS DE OLIVEIRA, Professor Efetivo**, em 12/12/2023, às 20:29, conforme decreto nº 8.539/2015.

---



A autenticidade do documento pode ser conferida no site [http://sei.ifba.edu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&acao\\_origem=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ifba.edu.br/sei/controlador_externo.php?acao=documento_conferir&acao_origem=documento_conferir&id_orgao_acesso_externo=0) informando o código verificador **3266004** e o código CRC **739B12C0**.

---

*Dedico esta obra à minha família e aos meus amigos.*

*O sucesso é a soma de pequenos esforços repetidos dia após dia. [Robert Collier]*

# AGRADECIMENTOS

Gostaria de expressar meus sinceros agradecimentos às pessoas e instituições que foram fundamentais para a realização deste trabalho de conclusão de curso.

Primeiramente, quero agradecer ao Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA) por proporcionar um ambiente propício ao aprendizado e à pesquisa. Agradeço pela infraestrutura, recursos e suporte acadêmico que me permitiram desenvolver este projeto.

Ao meu orientador, o Professor Dr. Wesley de Almeida Souto, devo uma profunda gratidão. Sua orientação, conhecimento e apoio foram essenciais em cada etapa desta jornada acadêmica. Suas valiosas sugestões e orientações foram cruciais para o sucesso deste trabalho.

Aos meus colegas e amigos, quero expressar meu reconhecimento pelo apoio mútuo, pelas discussões enriquecedoras e pelo espírito colaborativo que compartilhamos ao longo desses anos de estudo. Juntos, enfrentamos desafios e celebramos conquistas, e estou imensamente grata por isso.

Por fim, minha gratidão mais profunda vai para minha família. Suas palavras de incentivo, amor incondicional e apoio emocional foram o alicerce que sustentou minha jornada acadêmica. Sem o encorajamento e a compreensão de vocês, este trabalho não teria sido possível.

A todos que de alguma forma contribuíram para este trabalho, meu muito obrigada. Este é um marco importante em minha vida acadêmica, e sou grata por ter compartilhado essa jornada com cada um de vocês.

# RESUMO

A Interface Homem-Máquina (IHM) desempenha um papel crucial na interação entre humanos e sistemas computacionais. Este trabalho se concentra na criação de uma IHM remota para controlar um Motor de Indução Trifásico (MIT) por meio de um conversor de frequência CFW500, utilizando o protocolo de comunicação Modbus RTU e a interface serial RS-485.

Este trabalho foca no uso do CFW500 como forma de acionamento de um MIT, por oferecer vantagens como controle de velocidade e proteção contra sobrecargas e picos de corrente. Contudo, as IHMs integradas nesses dispositivos têm limitações, como acesso limitado a parâmetros e dificuldade de automação.

O objetivo principal deste estudo foi desenvolver uma IHM remota capaz de controlar o acionamento e a velocidade de um MIT por meio do CFW500 via protocolo Modbus. Para isso, foram estabelecidos objetivos específicos, como o desenho do sistema, a ligação dos componentes, a programação do CODESYS, a parametrização do CFW500, o estabelecimento da comunicação entre os componentes via Modbus, a criação de uma interface gráfica e a realização de testes.

Os resultados mostram que foi possível realizar o controle da velocidade do MIT e que os parâmetros de frequência, corrente e tensão puderam ser monitorados de forma eficaz. A implementação dessa IHM aborda limitações das IHMs integradas em conversores de frequência, permitindo um controle mais flexível e acesso simplificado a parâmetros. A utilização do padrão RS-485 e do protocolo Modbus facilita a comunicação entre dispositivos, tornando o sistema mais eficiente e de fácil integração. Ao cumprir os objetivos estabelecidos, este estudo contribui para melhorar a eficiência e usabilidade no controle de MITs.

**Palavras-chave:** CODESYS, Modbus, Conversor de frequência, Motor de indução trifásico, RS-485

# ABSTRACT

The Human-Machine Interface (HMI) plays a crucial role in the interaction between humans and computer systems. This work focuses on the creation of a remote HMI for controlling a Three-Phase Induction Motor (TIM) through a CFW500 frequency converter, utilizing the Modbus RTU communication protocol and the RS-485 serial interface.

This study centers on the use of the CFW500 as a mean to drive a TIM, because it offers advantages such as speed control and protection against overloads and current spikes. However, the integrated HMIs in these devices have limitations, such as restricted parameter access and automation challenges.

The primary objective of this study was to develop a remote HMI capable of controlling the activation and speed of a TIM through the CFW500 using the Modbus protocol. To achieve this goal, specific objectives have been established, such as drawing the system, connecting the components, CODESYS programming, parameterizing the frequency converter, establishing communication between the components, creating a graphical interface and running tests.

The results show that it was possible to control the TIM's speed while its frequency, current and tension were monitored. The implementation of this HMI addresses limitations of integrated HMIs in frequency converters, enabling more flexible control and simplified parameter access. The utilization of the RS-485 interface and the Modbus protocol facilitates the communication between devices, making the system more efficient and easily integrable. By fulfilling the set objectives, this study contributes to enhancing efficiency and usability in the control of TIMs.

**Keywords:** CODESYS, Modbus, Frequency converter, Three-phase induction motor, RS-485

# Lista de Figuras

1.1	Distribuição do consumo de energia elétrica em 2021 por classe. . . . .	3
2.1	PLC+HMI Samba™ series da Unitronics. . . . .	6
2.2	Visão explodida do W22 da WEG. . . . .	7
2.3	Partida direta . . . . .	9
2.4	Partida estrela-triângulo . . . . .	9
2.5	Partida compensadora . . . . .	10
2.6	Partida com <i>soft-starter</i> . . . . .	11
2.7	Partida com conversor de frequência . . . . .	12
2.8	Modelo Mestre/Escravo . . . . .	13
2.9	Camadas do modelo OSI . . . . .	16
3.1	Conversor de frequência CFW500 . . . . .	20
3.2	Motor de indução trifásico Weg . . . . .	22
3.3	Placa de identificação do motor . . . . .	23
3.4	Terminais de ligação do motor . . . . .	23
3.5	Cabo conversor USB serial RS-485 . . . . .	24
3.6	Diagrama de ligação dos componentes . . . . .	25
4.1	Conexões do CFW500 . . . . .	27
4.2	Ligações dos terminais do MIT . . . . .	28
4.3	Ligações do conversor Serial para USB . . . . .	28
4.4	Sistema completo . . . . .	29
4.5	Menu <i>Devices</i> . . . . .	31
4.6	Caixa de diálogo inicial do <i>Standard Project</i> . . . . .	31
4.7	Configuração da porta serial . . . . .	32
4.8	Configuração do mestre da rede . . . . .	32
4.9	Configuração do escravo da rede . . . . .	32
4.10	Criando um canal Modbus . . . . .	33
4.11	Bits do parâmetro P0680 . . . . .	36
4.12	Bits do parâmetro P0682 . . . . .	36
4.13	Mapeamento dos canais de comunicação Modbus . . . . .	37

4.14 Interface gráfica . . . . .	38
4.15 CFW500 com velocidade armazenada . . . . .	39
4.16 Interface gráfica com motor girando . . . . .	40
4.17 Gráficos do motor rodando na frequência 3 Hz . . . . .	40
4.18 Gráficos do motor rodando na frequência 30 Hz . . . . .	41
4.19 Gráficos do motor rodando na frequência 60 Hz . . . . .	41
4.20 Gráficos do motor acelerando . . . . .	42
4.21 Gráficos do motor desacelerando . . . . .	42

# Lista de Tabelas

2.1	Estrutura de um telegrama . . . . .	13
2.2	Public Function Codes (ORGANIZATION, 2012) . . . . .	15
4.1	Parâmetros CFW500 para inserção dos dados do motor . . . . .	29
4.2	Parametrização Modbus RTU . . . . .	30
4.3	Elementos visuais e suas respectivas variáveis . . . . .	37

# Lista de Códigos

4.1	Declaração de variáveis . . . . .	34
4.2	Código principal . . . . .	35

# Glossário: Símbolos e Siglas

<b>Notação</b>	<b>Descrição</b>	<b>Páginas</b>
$\Delta$	Configuração Triângulo ou Delta	9
$f_e$	Frequência elétrica aplicada (Hz)	8
$n_s$	Velocidade Síncrona (rpm)	8
ASCII	<i>American Code for Information Interchange</i>	13
CA	<i>Corrente Alternada</i>	7
CC	<i>Corrente Contínua</i>	7
CLP	<i>Controlador Lógico Programável</i>	18
CRC	<i>Cycling Redundance Check</i>	15
EIA	<i>Electronic Industries Association</i>	17
EPE	<i>Empresa de Pesquisa Energética</i>	2
FBD	<i>Function Block Diagram</i>	18
GND	<i>Ground</i>	17
IHM	<i>Interface Homem-Máquina</i>	1
IL	<i>Instruction List</i>	18
ISO	<i>International Standards Organization</i>	16
LD	<i>Ladder</i>	18

<b>Notação</b>	<b>Descrição</b>	<b>Páginas</b>
MIT	<i>Motor de Indução Trifásico</i>	1
N	Número de polos do estator	8
OSI	<i>Open Systems Interconnection</i>	16
PWM	<i>Pulse Width Modulation</i>	11
RFC	<i>Reserved Function Code</i>	14
RTU	<i>Remote Terminal Unit</i>	13
SFC	<i>Sequential Flow Chart</i>	18
ST	<i>Structured Text</i>	18
TIA	<i>Telecommunications Industry Association</i>	17
Y	Configuração Estrela	9

# Sumário

<b>Folha de Rosto</b> . . . . .	<b>ii</b>
<b>Ficha Catalográfica</b> . . . . .	<b>iii</b>
<b>Folha de Aprovação</b> . . . . .	<b>iv</b>
<b>Resumo</b> . . . . .	<b>viii</b>
<b>Abstract</b> . . . . .	<b>ix</b>
<b>Lista de Figuras</b> . . . . .	<b>x</b>
<b>Lista de Tabelas</b> . . . . .	<b>xii</b>
<b>Lista de Códigos</b> . . . . .	<b>xiii</b>
<b>Glossário: Símbolos e Siglas</b> . . . . .	<b>xiv</b>
<b>1 Introdução</b> . . . . .	<b>1</b>
1.1 Objetivo Geral . . . . .	2
1.1.1 Objetivos Específicos . . . . .	2
1.2 Justificativa . . . . .	2
<b>2 Referencial Teórico</b> . . . . .	<b>5</b>
2.1 Levantamento do Estado da Arte . . . . .	5
2.1.1 IHMs na automação industrial . . . . .	5
2.1.1.1 Soluções em <i>hardware</i> . . . . .	6
2.1.1.2 Soluções em <i>software</i> . . . . .	6
2.2 Motor de Indução Trifásico (MIT) . . . . .	7
2.2.1 Tipos de acionamento . . . . .	8
2.2.1.1 Partida Direta . . . . .	8
2.2.1.2 Partida Estrela-Triângulo . . . . .	9
2.2.1.3 Partida com chave compensadora . . . . .	10

---

2.2.1.4	Soft-Starter	10
2.2.1.5	Conversor de Frequência	11
2.3	Protocolo Modbus RTU	12
2.3.1	Modos de Transmissão	13
2.3.2	Estrutura das mensagens	13
2.3.2.1	Endereço	13
2.3.2.2	Código de Função	14
2.3.3	Dados da Requisição	15
2.3.4	CRC	15
2.4	Interface serial RS-485	16
2.5	CODESYS	18
<b>3</b>	<b>Metodologia</b>	<b>19</b>
3.1	Materiais e recursos	19
3.1.1	Notebook Asus	19
3.1.2	Conversor de frequência CFW500	20
3.1.3	Motor de indução trifásico Weg	22
3.1.4	Conversor RS-485 para USB	24
3.2	Diagrama de ligação	25
3.3	Desenvolvimento do código	25
3.4	Criação da interface gráfica	25
3.5	Parametrização do conversor de frequência	25
3.6	Ligação entre os componentes	26
3.7	Protocolo Modbus	26
<b>4</b>	<b>Desenvolvimento</b>	<b>27</b>
4.1	Conexão dos componentes	27
4.2	Parametrização do CFW500	29
4.3	Programação do CODESYS	30
4.3.1	Iniciando um novo projeto	30
4.3.2	Estabelecendo a comunicação entre o CFW500 e o CODESYS	31
4.3.3	Criando os canais de comunicação mestre-escravo	33
4.3.4	Desenvolvimento do código	34
4.3.5	Criação da interface gráfica	37
4.4	Resultados, Ensaios e Testes do Protótipo	38
<b>5</b>	<b>Considerações Finais</b>	<b>43</b>
<b>6</b>	<b>Sugestões para Trabalhos Futuros</b>	<b>45</b>

---

---

<b>REFERÊNCIAS</b> .....	<b>46</b>
--------------------------	-----------

# Capítulo 1

## Introdução

A Interface Homem-Máquina (IHM) é como os humanos interagem com um sistema de computador, desempenhando um papel importante na usabilidade, eficiência e experiência do usuário. Pode conter telas, teclados, botões, *sliders*, menus, ícones, indicadores visuais e outros elementos. Outros métodos de interação, como comandos de voz, gestos e outros tipos de dispositivos de entrada, também podem ser relevantes, dependendo do contexto e dos requisitos do sistema.

Em contexto industrial, a IHM desempenha um papel fundamental. Ela permite que o operador visualize e opere informações relevantes sobre as máquinas elétricas do sistema, auxiliando o monitoramento do seu desempenho e a tomada de decisões apropriadas. De forma geral, uma IHM para supervisão de motores elétricos conta com uma tela de exibição, botões e outros elementos de entrada. Além de permitir a visualização de dados importantes do motor, como velocidade, corrente elétrica, tensão, dentre outros, ela também pode permitir o seu controle, como iniciar, parar e alterar a velocidade.

Uma variedade de abordagens e tecnologias estão disponíveis para implementação de IHMs para monitoramento de motores elétricos, incluindo telas sensíveis ao toque, telas gráficas, interfaces web, dentre outras. Neste trabalho, foi desenvolvida uma IHM no *software* CODESYS, utilizando elementos para que o usuário possa controlar o acionamento, a parada e a velocidade de um motor de indução trifásico (MIT) por meio do envio de comandos para o conversor de frequência CFW-500, da Weg. A IHM criada também dispõe de uma tela que exhibe dados de velocidade de saída do motor, corrente do motor, frequência de saída do motor, tensão de saída, além de contar também com indicação de estados de falha,

alarme e subtensão do conversor de frequência.

## 1.1 Objetivo Geral

Criar uma IHM remota capaz de controlar o acionamento e a velocidade de um MIT através de um conversor de frequência utilizando o protocolo de comunicação Modbus RTU e interface serial RS-485.

### 1.1.1 Objetivos Específicos

- 1) Desenhar os circuitos de força e comando do MIT;
- 2) Realizar a ligação do MIT com o conversor de frequência e o computador, de acordo com o circuito desenhado;
- 3) Fazer a programação do *software* CODESYS;
- 4) Parametrizar o conversor de frequência;
- 5) Estabelecer a comunicação entre o CODESYS e o conversor de frequência via protocolo Modbus RTU;
- 6) Criar uma IHM remota e
- 7) Testar o sistema.

## 1.2 Justificativa

De acordo com dados apresentados pela Empresa de Pesquisa Energética (EPE) no Anuário Estatístico de Energia Elétrica 2022 (ano base 2021) (EPE, 2022), em 2021 a distribuição do consumo de eletricidade se deu conforme a Figura 1.1. O que demonstra que o setor industrial é, atualmente, o maior consumidor de energia elétrica do país.

Um dos maiores contribuintes para o enorme consumo de energia no setor industrial é o MIT. Existem diversas maneiras de acioná-lo: partida direta, estrela-triângulo, chave compensadora e chaves de partida eletrônica. Dentre elas, as chaves de partida eletrônica são opções que vêm ganhando muito espaço por trazerem diversos benefícios para o motor, proteção contra picos de corrente elétrica,



**Figura 1.1** – Distribuição do consumo de energia elétrica em 2021 por classe.

paradas inesperadas, sobrecarga, sobreaquecimento, redução da corrente de partida, dentre outros. As duas chaves eletrônicas mais utilizadas são o *soft-starter* (pode ser traduzido do inglês como "partida suave"), e o conversor de frequência, popularmente conhecido como inversor de frequência. O *soft-starter* tem a capacidade de reduzir sua corrente de partida e definir o tempo de aceleração e desaceleração. Já o conversor de frequência é capaz de realizar todas essas tarefas e, além disso, controlar a velocidade do motor.

O controle do motor de indução a partir de um conversor de frequência pode ser feito de maneira local ou remota. Localmente, o acionamento se dá através da IHM do conversor. Já remotamente, utilizam-se as suas saídas digitais e analógicas ou interface serial, como USB, RS-232 e RS-485.

A leitura dos dados através da IHM do conversor não é muito prática, pois só mostra um dado de cada vez e é necessário utilizar o próprio teclado do equipamento para selecionar o dado a ser lido, que é exibido apenas como um valor numérico. O controle do motor utilizando esses dispositivos através da sua IHM integrada também é trabalhoso, pois é necessário buscar cada parâmetro a ser mudado utilizando seu próprio teclado. Além disso, é necessário ter contato direto com o equipamento para poder editar seus parâmetros e o seu uso dificulta qualquer processo de automação. A comunicação via saídas digitais e analógicas também apresenta algumas dificuldades como, por exemplo, precisar realizar a conexão de diversos cabos.

Assim, surgiu o interesse em estudar um mecanismo para facilitar esse processo através da implementação de uma IHM remota via protocolo de comunicação Modbus RTU, utilizando o padrão RS-485, que consiste em apenas um cabo de 2 ou 4 vias. O sistema proposto deve ser capaz de permitir que o usuário controle o acionamento e a velocidade do MIT a partir de um computador ligado ao conversor de frequência responsável pelo acionamento do MIT e, além disso, mostrar na tela os dados do motor obtidos pelo conversor de frequência.

# Capítulo 2

## Referencial Teórico

Este capítulo será destinado a fazer uma fundamentação teórica que contribuirá com a implementação e desenvolvimento do trabalho. Serão abordados o estado da arte, o Motor de Indução Trifásico, seu princípio de funcionamento, aspectos construtivos e métodos de acionamento, o protocolo Modbus RTU, a interface serial RS-485 e o *software* CODESYS.

### 2.1 Levantamento do Estado da Arte

Este capítulo tem como objetivo discutir o cenário atual de pesquisa e desenvolvimento da interface homem-máquina para controle remoto de motores de indução trifásicos. O levantamento do estado da arte visa vincular a pesquisa às contribuições existentes na área.

#### 2.1.1 IHMs na automação industrial

As IHMs são componentes muito importantes em sistemas de automação industrial, facilitando o monitoramento, o controle e a tomada de decisões. Diversas tecnologias como telas sensíveis ao toque, interfaces gráficas e comandos de voz têm sido aplicadas para melhorar a usabilidade e a eficiência das IHMs em ambientes industriais.

### 2.1.1.1 Soluções em *hardware*

Existe uma enorme diversidade de painéis desenvolvidos para visualização industrial. Há os painéis embutidos, que são aqueles fisicamente incorporados na máquina ou armário de controle. Eles podem variar no que diz respeito ao tamanho de ecrã, processador, memória interna e protocolos de comunicação (PEREIRA, 2018). Os painéis embutidos podem ter protocolo de comunicação proprietário, otimizado para utilizar em conjunto com dispositivos do mesmo fabricante, protocolos de comunicação abertos/normalizados, que podem trabalhar com dispositivos de diferentes fabricantes, e aqueles que possuem CLP integrado, acumulando funções de monitoramento e controle, denominados HMI-PLC (PEREIRA, 2018), que são os mais relevantes para este trabalho.

A Unitronics, empresa que desenvolve e comercializa soluções para automação e controle, possui modelos de HMI-PLC com *on board I/Os*, ou seja, possuem entradas e saídas integradas. A Figura 2.1 mostra o PLC+HMI da série Samba™, que possui uma tela plana com *touchscreen* que pode ter 3,5", 4,3" ou 7". Esse dispositivo é indicado para projetos pequenos e possui mini USB para as versões de 4,3" e 7" e serial RS232 para a de 3,5". Além disso, é possível adicionar uma porta serial/Ethernet e uma porta CANbus, suporta vários protocolos, como Modbus TCP, por exemplo, e possui algumas funcionalidades como envio de SMS e *e-mail* e suporte para modem 3G (UNITRONICS, 2023).



Figura 2.1 – PLC+HMI Samba™ series da Unitronics.

### 2.1.1.2 Soluções em *software*

Além das opções em *hardware*, também existem soluções de software disponíveis em computadores convencionais e industriais para que desempenhem funções tradicionalmente confiadas a CLPs e IHMs, chamados de *SoftPLC* e *SoftHMI*.

O *SoftPLC* consiste em utilizar um PC como a plataforma de suporte de *hard-*

ware e usar *software* para alcançar as funções básicas de um CLP tradicional. As funções do CLP estão encapsuladas no *software*, sendo executado no ambiente do PC (LIANG; LI, 2011). Ele é dividido em duas partes: o sistema de desenvolvimento e o sistema de execução. O CODESYS Control, utilizado neste trabalho, é uma solução em *software* popularmente utilizada. A sessão 2.5 trata dele com mais detalhes.

## 2.2 Motor de Indução Trifásico (MIT)

O MIT é uma máquina elétrica rotativa de corrente alternada (CA). Ao compará-lo a motores de corrente contínua (CC), ele possui grandes vantagens, como: inexistência de comutador, menor custo em relação a um motor CC de mesma potência, manutenção mais simples, menor consumo de energia nos processos de aceleração e frenagem e possibilidade de obter velocidades maiores (CASTRO, 2003). A Figura 2.2 mostra um exemplo de MIT.

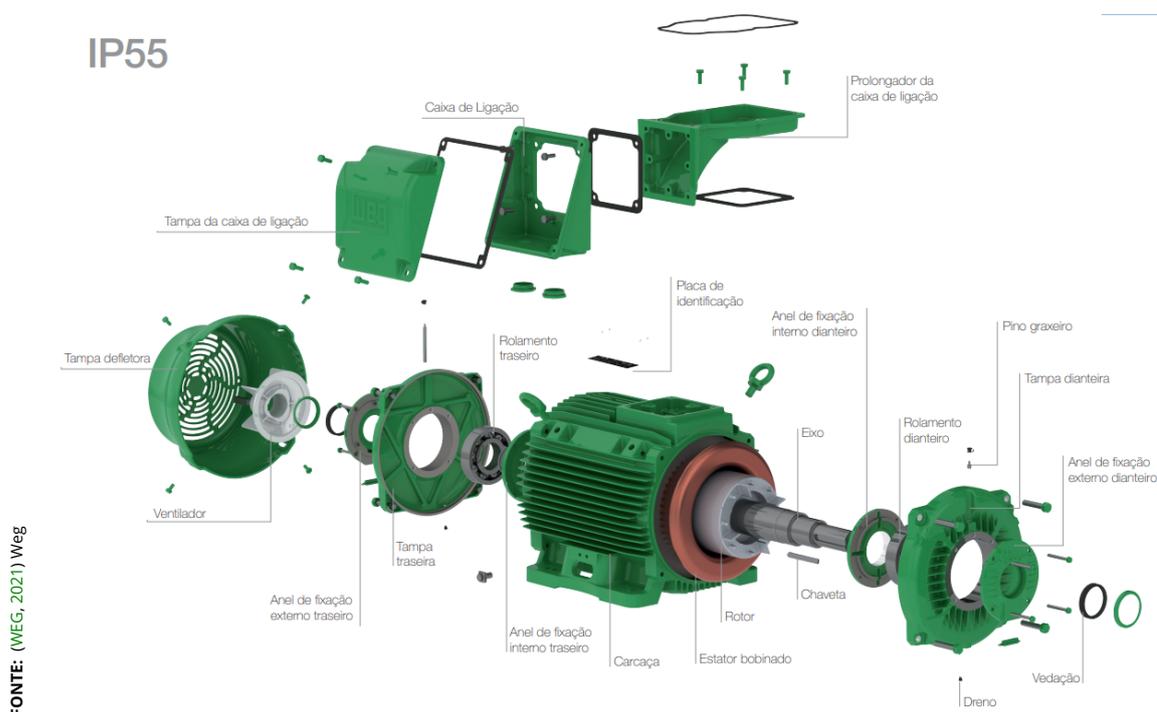


Figura 2.2 – Visão explodida do W22 da WEG.

Os principais componentes de um MIT são o estator e o rotor. O estator, também chamado de armadura, é a parte fixa do motor e o rotor é a parte girante. Em um motor de indução, os enrolamentos do estator produzem uma onda de fluxo eletromagnético que atravessa o rotor e induz correntes em seus enrolamen-

tos curto-circuitados. O rotor atua de forma assíncrona, sendo possível classificar os MITs como máquinas elétricas assíncronas que "produzem conjugado apenas quando a velocidade do rotor é diferente da velocidade síncrona" (UMANS, 2014, p. 199).

No MIT, cada fase da rede de alimentação trifásica é ligada a um dos três enrolamentos do estator. O campo magnético produzido no entreferro gira na velocidade síncrona, que é determinada pelo número de polos do estator e pela frequência aplicada a ele (UMANS, 2014), de acordo com a Equação 2.1.

$$n_s = \left(\frac{120}{N}\right)f_e \quad (2.1)$$

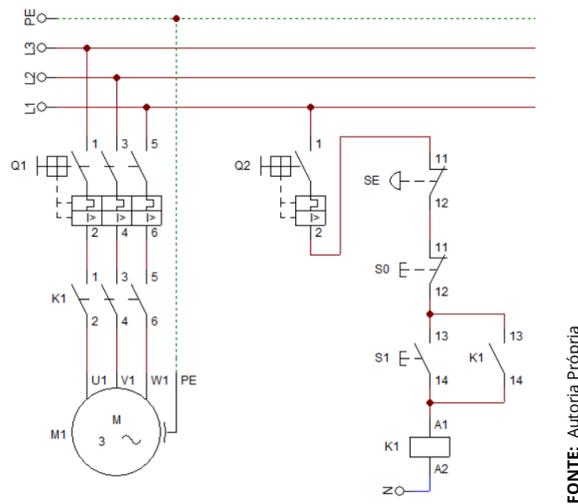
Onde  $n_s$  é a velocidade síncrona, em rpm (rotações por minuto),  $N$  refere-se ao número de polos do estator e  $f_e$  é a frequência elétrica aplicada, em Hertz.

### 2.2.1 Tipos de acionamento

As principais formas de acionamento de um MIT são: partida direta, partida estrela-triângulo, partida com chave compensadora e partida com chave eletrônica, que pode ser *soft-starter* ou conversor de frequência. Para cada tipo de partida apresentado a seguir, foi feito um diagrama elétrico utilizando o *software* CADE SIMU. Como os diagramas foram desenvolvidos com fim puramente didático, proteções além os disjuntor motor, como fusíveis e contatos auxiliares, foram suprimidas.

#### 2.2.1.1 Partida Direta

Dentre os diferentes métodos de dar acionar um motor, a partida direta é com certeza o mais simples e comum. Nele, o motor é conectado diretamente à fonte de alimentação por meio de um contator, permitindo que a corrente elétrica flua diretamente para o motor. A partida direta, embora simples, produz altos picos de corrente durante a partida, o que pode afetar tanto a rede quanto o próprio motor. A Figura 2.3 mostra o diagrama de ligação da partida direta.

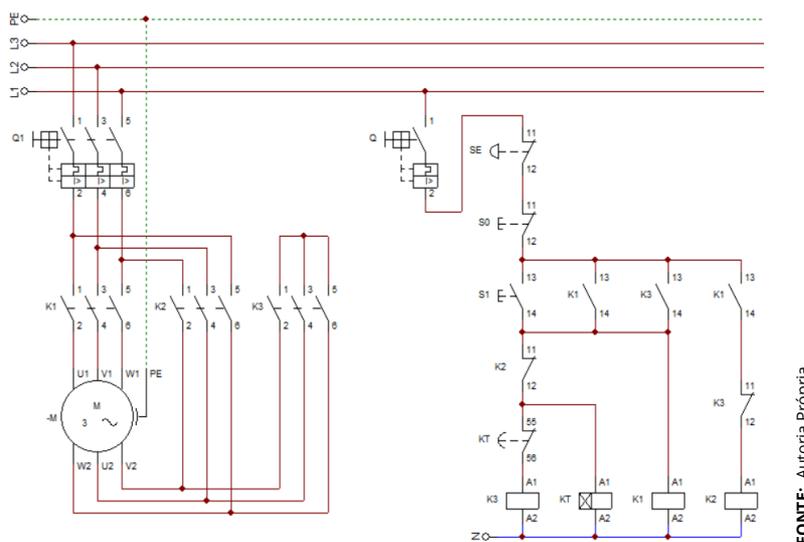


FONTE: Autoria Própria

Figura 2.3 – Partida direta

### 2.2.1.2 Partida Estrela-Triângulo

A partida estrela-triângulo é um método usado para reduzir a corrente de partida em motores trifásicos. Nesta partida, o motor é inicialmente ligado em configuração estrela (ou "Y"), o que reduz a tensão aplicada aos enrolamentos do motor. Após um determinado período de tempo, os contatos mudam para uma configuração triângulo, também chamada de delta (ou " $\Delta$ "), alimentando o motor com tensão nominal. Este método reduz picos de corrente e cargas no sistema elétrico durante a inicialização. A Figura 2.4 mostra o diagrama de ligação da partida estrela-triângulo.

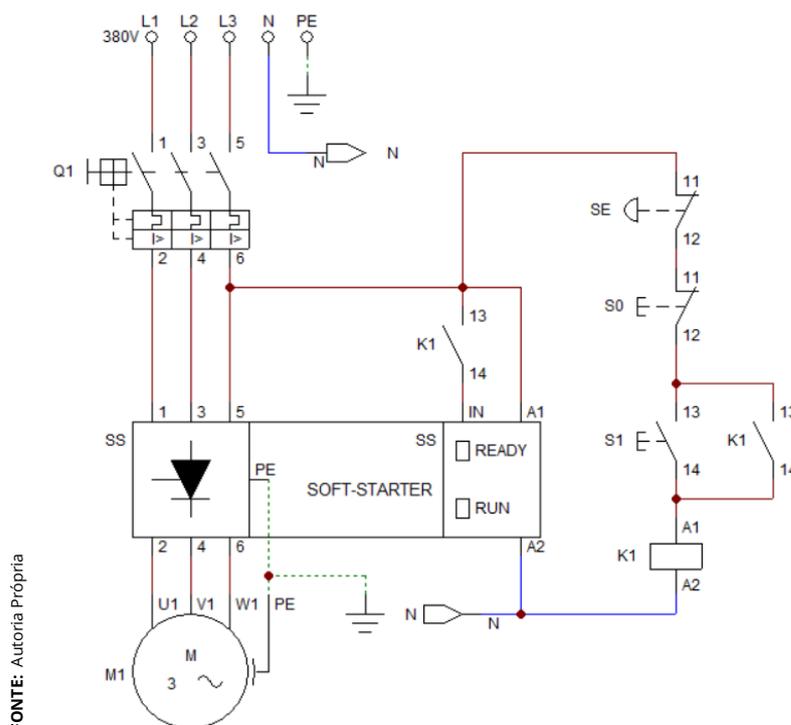


FONTE: Autoria Própria

Figura 2.4 – Partida estrela-triângulo



*Soft-starters* também possuem funcionalidades para aumentar sua segurança, como proteção contra sobrecarga, curto-circuito e sobretensão.



FONTE: Autoria Própria

Figura 2.6 – Partida com soft-starter

### 2.2.1.5 Conversor de Frequência

O conversor de frequência, popularmente conhecido como inversor de frequência, é um dispositivo eletrônico utilizado para converter uma frequência de entrada em uma frequência de saída desejada. Com ele, além de realizar o acionamento de um motor elétrico de corrente alternada de forma suave e segura, é possível obter uma frequência de saída ajustável para controlar a sua velocidade.

Seu princípio de funcionamento envolve um processo chamado de modulação por largura de pulso (**PWM - Pulse Width Modulation**), que varia a largura dos pulsos de tensão fornecidos ao motor. Ao variar a largura dos pulsos, a frequência efetiva da alimentação elétrica é modificada, permitindo controlar a velocidade do motor (**MOHAN; UNDELAND; ROBBINS, 2002**).

Assim como a *soft-starter*, os conversores também fornecem funcionalidades importantes de proteção contra sobrecorrente, sobretensão e subtensão. A Figura 2.7 mostra o diagrama de um acionamento convencional de um motor trifásico utilizando um conversor de frequência.

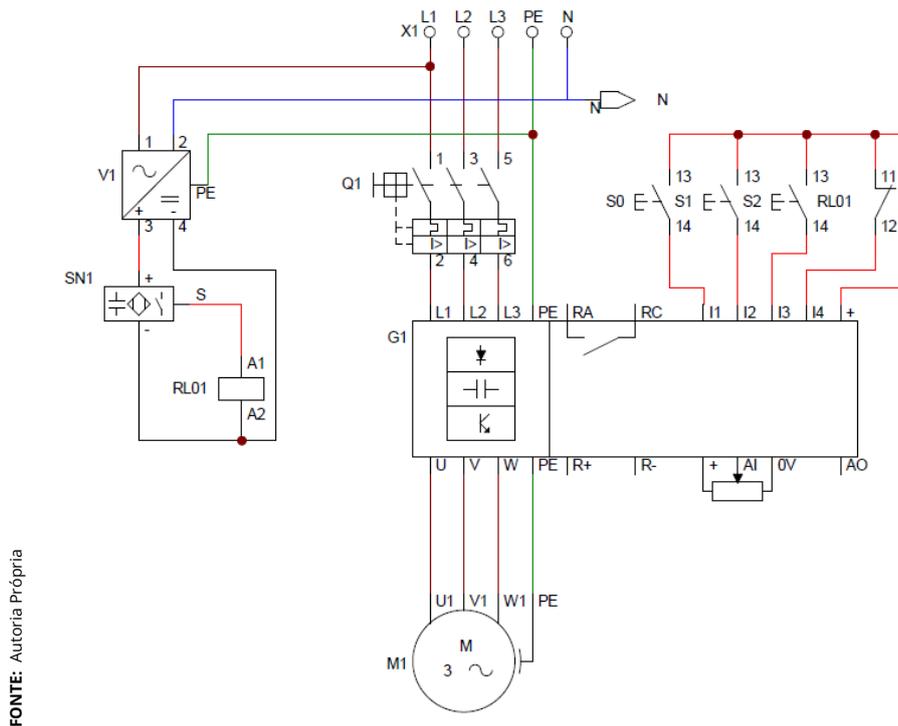
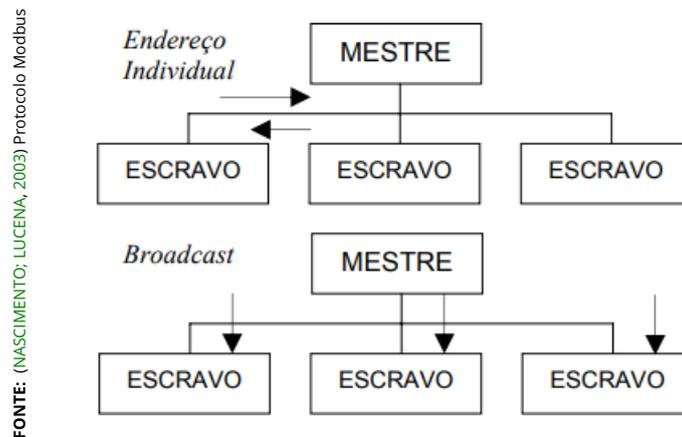


Figura 2.7 – Partida com conversor de frequência

## 2.3 Protocolo Modbus RTU

Modbus é um protocolo de troca de mensagens criado em 1979 pela Modicon, marca que hoje pertence à Schneider Electric. Seu objetivo é estabelecer comunicação do tipo cliente/servidor (ou mestre/escravo) entre dispositivos conectados em diferentes tipos de barramentos ou redes (ORGANIZATION, 2012). É um protocolo bastante popular em redes industriais, pois é utilizado por diversos dispositivos empregados em automação industrial.

A comunicação do tipo mestre/escravo estabelece que apenas um dos dispositivos conectados, denominado mestre, pode iniciar a comunicação (conhecida como *query*). Os dispositivos escravos, então, respondem de acordo com o pedido do mestre, seja enviando os dados solicitados ou executando alguma ação. O mestre pode se comunicar com apenas um escravo ou com vários. No último caso, ele tem a capacidade de endereçar cada dispositivo individualmente ou acessar todos simultaneamente, enviando mensagens em *broadcast* (NASCIMENTO; LUCENA, 2003), como ilustrado na Figura 2.8.



**Figura 2.8** – Modelo Mestre/Escravo

### 2.3.1 Modos de Transmissão

Dois modos de transmissão foram definidos para o protocolo Modbus: *American Code for Information Interchange*, [ASCII](#), e *Remote Terminal Unit*, [RTU](#). Os modos servem para especificar como os bytes da mensagem são transmitidos, sendo que não há a possibilidade de utilizar mais de um modo na mesma rede ([WEG, 2011](#)). O modo utilizado neste trabalho é o modo RTU.

### 2.3.2 Estrutura das mensagens

No modo Modbus RTU as mensagens, ou telegramas, trocadas entre mestre e escravo, possuem a seguinte estrutura: Endereço, Código de Função, Dados e CRC. O único campo de tamanho variável é o de dados. A Tabela 2.1 mostra como é a estrutura de um telegrama de requisição ou resposta.

**Tabela 2.1** – Estrutura de um telegrama

Endereço (1 byte)	Função (1 byte)	Dados da Requisição (n bytes)	CRC (2bytes)
----------------------	--------------------	----------------------------------	-----------------

FONTE: Autoria Própria

#### 2.3.2.1 Endereço

Para iniciar a comunicação, o mestre envia um byte com o endereço do escravo para o qual a mensagem é destinada. Quando o escravo responde, o início da sua mensagem também será o seu próprio endereço. Caso haja mais de um es-

cravo e o mestre deseje enviar uma mensagem em *broadcast* para todos, ele deve utilizar o endereço 0.

### 2.3.2.2 Código de Função

O *byte* deste campo especifica qual o serviço ou função solicitada pelo mestre. Existem três tipos de códigos de funções descritos na especificação da aplicação do Modbus:

**Public Function Codes (Códigos de Função Públicos):** São bem definidos, há a garantia de serem únicos, validados pela comunidade MODBUS.org, publicamente documentados, têm teste de conformidade disponível e incluem tanto os códigos de função públicos atribuídos definidos, quanto os códigos de funções não atribuídos reservados para uso futuro.

**User-Defined Function Codes (Códigos de Função Definidos pelo usuário):** As faixas de códigos de função desse tipo são duas: de 65 a 72 e de 100 a 110 decimal. O usuário pode selecionar e implementar um código de função que não é suportado pela especificação, mas não há garantia de que o código selecionado seja único e se o usuário quiser reposicionar a funcionalidade como um código de função público, ele deve iniciar um [RFC](#) (Reserved Function Code, ou Código de Função Reservado) para introduzir a mudança na categoria pública e ter um novo código de função atribuído. A MODBUS Organization, Inc reserva expressamente o direito de desenvolver o RFC proposto.

**Reserved Function Codes:** São códigos de função atualmente utilizados por algumas empresas para produtos legados e que não estão disponíveis para uso público.

Neste trabalho, apenas os códigos de função públicos foram utilizados. A Tabela [2.2](#) relaciona os códigos de função presentes na especificação de aplicação do protocolo modbus.

Foram utilizadas apenas duas funções: *Read Holding Registers*, que lê o conteúdo de blocos de registradores do tipo *holding* de um dispositivo remoto, e *Write Multiple Registers*, que realiza a escrita em registradores do tipo *holding*.

**Tabela 2.2** – Public Function Codes ([ORGANIZATION, 2012](#))

Função	Código	Sub código
Read Discrete Inputs	02	
Read Coils	01	
Write Single Coil	05	
Write Multiple Coils	15	
Read Input Register	04	
Read Holding Registers	03	
Write Single Register	06	
Write Multiple Registers	16	
Read/Write Multiple Registers	23	
Mask Write Register	22	
Read FIFO queue	24	
Read File record	20	
Write File record	21	
Read Exception status	07	
Diagnostic	08	00-18,20
Get Com event counter	11	
Get Com Event Log	12	
Report Server ID	17	
Read device Identification	43	14
Encapsulated Interface	43	13, 14
CANopen General Reference	43	13

FORTE: Autoria Própria

### 2.3.3 Dados da Requisição

O campo de dados tem tamanho variável. Seu formato e conteúdo dependem da função utilizada e dos valores transmitidos.

### 2.3.4 CRC

Este é o campo responsável pela checagem de erros na mensagem. A sigla **CRC** vem do inglês *Cycling Redundance Check* e utiliza polinômios geradores que estão presentes tanto do lado do remetente quanto do receptor ([GEEKSFORGEEKS, 2023](#)).

## 2.4 Interface serial RS-485

Quando computadores são interligados de modo a permitir a troca de informações entre eles, é formada uma rede. A fim de reduzir a complexidade das redes, a ISO (*International Standards Organization*) desenvolveu, na década de 1970, o padrão OSI (*Open Systems Interconnection*) (MENDES, 2007 apud LOPES; BENEVENUTO; OLIVEIRA, 2015). A Figura 2.9 relaciona as sete camadas que constituem esse modelo.

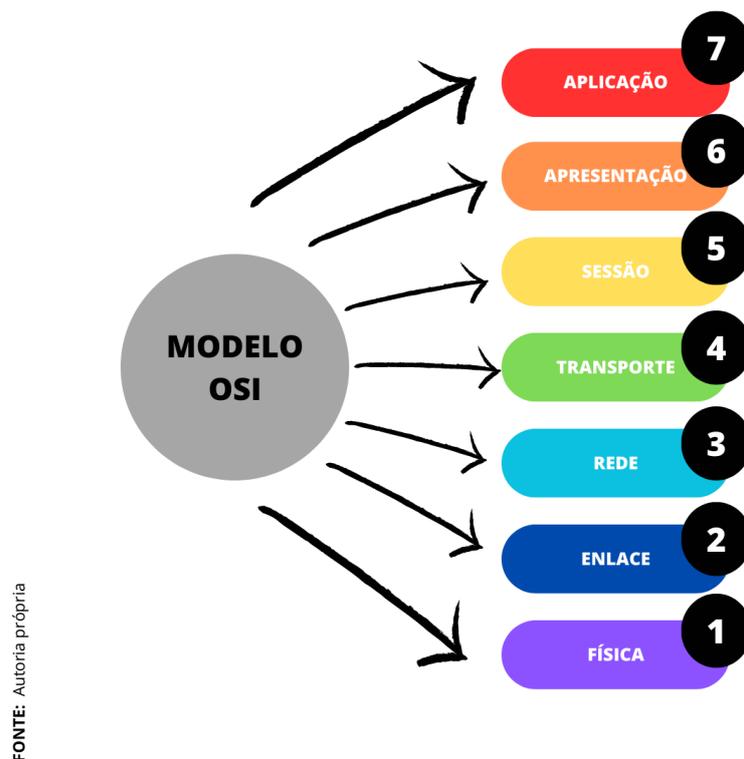


Figura 2.9 – Camadas do modelo OSI

A camada física é a que determina "como serão realizadas as transferências de bits através de um canal de comunicação" (LOPES; BENEVENUTO; OLIVEIRA, 2015, p. 271), ou seja, ela lida com sinais elétricos. A próxima camada é a de enlace, que recebe os dados do meio físico e realiza uma verificação para detectar possíveis erros, além de controlar o fluxo da transmissão de dados. A camada 3, rede, contém o endereçamento das máquinas e é responsável por decidir os caminhos que os dados irão seguir. A camada de transporte garante o envio e o recebimentos dos dados vindos da camada 3. A camada de sessão é a que estabelece e encerra conexões entre *hosts*, como são chamadas as máquinas conectadas a uma rede. Já a camada de apresentação tem como objetivo tratar os dados para que sejam usados pela sétima e última camada, a de aplicação, que irá, finalmente,

consumir os dados (MATHEUS, 2023).

O protocolo Modbus, descrito na sessão anterior, está na camada 7, a de aplicação. Ele necessita de um meio físico para estabelecer a comunicação entre os dispositivos. Assim, a interface serial RS-485 escolhida para este trabalho está no primeiro nível do modelo OSI. Essa interface segue o padrão EIA/TIA-485, em que EIA é a *Electronic Industries Association*, que formula padronizações técnicas nos Estados Unidos da América, e TIA é a *Telecommunications Industry Association*. Segundo (MENDES, 2007), "o comitê EIA/TIA especifica o sistema de cabeamento estruturado utilizado nas redes de computadores".

De acordo com (WEG, 2011), algumas das características da interface RS-485 são:

- ▶ Possibilita comunicação utilizando taxas de 9600 até 38400 Kbit/s.
- ▶ Interface isolada galvanicamente e com sinal diferencial, conferindo maior robustez contra interferência eletromagnética.
- ▶ Permite a conexão de até 32 dispositivos no mesmo segmento. Ao utilizar repetidores, pode conectar até 246 dispositivos.
- ▶ Comprimento máximo do barramento de 1000 metros.

A sua conexão é feita utilizando um cabo, preferencialmente de par trançado, de dois ou quatro fios, mais um fio 0 V, geralmente chamado de GND (*ground* ou terra). Quando são utilizados dois pares de fios, a operação é chamada de *full-duplex* e pode transmitir e receber dados ao mesmo tempo. Já na conexão com apenas um par, a operação é *half-duplex*, em que há apenas a transmissão ou o recebimento de dados de cada vez. No presente trabalho foi utilizado o modo *half-duplex*.

O par de fios é composto por um fio de dados e um fio de dados invertidos, ou seja "enquanto um fio transmite o sinal original, o outro transporta sua cópia inversa", o que significa que a transmissão de dados é diferencial (balanceada) (WEIS, 2021), oferecendo resistência à interferências. Ainda segundo Weis, o comprimento máximo de um cabo RS-485 pode chegar a 4000 pés (cerca de 1200 metros).

## 2.5 CODESYS

O CODESYS é um *software* de código aberto desenvolvido pela empresa alemã CODESYS GmbH, previamente chamada 3S-Smart Software Solutions GmbH. Seu nome vem do inglês *Controller Development System* e seu desenvolvimento foi baseado na norma internacional IEC 61131, que normaliza o projeto de controladores lógicos programáveis (CLP), onde IEC é a sigla da *International Electrotechnical Commission* (Comissão Internacional Eletrotécnica). O CODESYS é uma plataforma de automação, utilizada em aplicações de controle.

Todas as cinco linguagens de programação definidas na IEC 61131 estão presentes no ambiente de desenvolvimento do CODESYS, são elas:

- ▶ **ST** (*Structured Text*) - Texto Estruturado, similar às linguagens Pascal e C.
- ▶ **IL** (*Instruction List*) - Lista de Instruções, similar à linguagem Assembly.
- ▶ **SFC** (*Sequential Flow Chart*) - Método gráfico para sistemas sequenciais, também conhecido como Grafcet.
- ▶ **LD** (*Ladder*) - Linguagem que utiliza chaves e bobinas.
- ▶ **FBD** (*Function Block Diagram*) - Diagrama de Bloco.

Sua tecnologia é bastante versátil, o que permite a utilização de equipamentos de diversos fabricantes para automação no CODESYS. Ele possui simulador de CLP, além do CODESYS Control, um sistema de tempo de execução, que transforma um PC com sistema operacional Microsoft Windows em um CLP (GROUP, 2023). Um sistema de tempo de execução, também conhecido como ambiente de tempo de execução, é descrito por (ROUSE, 2013) como a coleção de recursos de *software* e *hardware* que permitem que um programa seja executado em sistema computacional. Neste trabalho, foi utilizado o CODESYS Control Win V3 x64 como CLP.

# Capítulo 3

## Metodologia

Neste capítulo serão abordadas todas as etapas para o desenvolvimento do protótipo do sistema de controle de acionamento e velocidade de um MIT. Primeiro será abordado sobre quais os materiais e recursos utilizados no desenvolvimento do projeto, depois será apresentado o diagrama de ligação dos componentes que foi concebido antes de realizar a implementação propriamente dita, como foi feita a programação do CLP, a criação da interface gráfica, a parametrização do conversor de frequência e, finalmente, o passo a passo das ligações feitas entre os equipamentos.

### 3.1 Materiais e recursos

Este trabalho foi desenvolvido na plataforma CODESYS, instalada em um *notebook* Asus, utilizando um conversor de frequência CFW500 da Weg, ligado ao computador através de um cabo conversor RS-485 para USB, e o motor controlado foi um MIT da marca Weg.

#### 3.1.1 *Notebook* Asus

O computador utilizado foi um *notebook* da marca Asus com as seguintes características:

- ▶ Processador Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz 2.59 GHz

- ▶ Memória RAM de 8,00 GB
- ▶ Sistema Operacional Windows 10 Pro de 64 bits

### 3.1.2 Conversor de frequência CFW500

O conversor de frequência CFW500 da Weg, Figura 3.1, conta com diversas características e recursos relevantes para a indústria. Em relação à conectividade, o CFW500 pode ser interligado às principais redes de comunicação industriais rápidas, como Fieldbus, com protocolos globalmente difundidos como CANopen, Profibus-DP, EtherNet/IP, Modbus-TCP, PROFINET IO e DeviceNet. Uma característica importante que possibilitou a sua utilização neste trabalho foi a de que qualquer módulo plug-in do CFW-500 possui interface serial RS485 Modbus RTU incorporada. (WEG, 2023).



**Figura 3.1** – Conversor de frequência CFW500

Suas principais características são:

- ▶ Corrente de saída de 1,0 a 211 A (0,25 a 175 cv / 0,18 a 132 kW);
- ▶ Tensão de alimentação monofásica ou trifásica 200-240 V ou 380-480 V;
- ▶ Controle escalar (V/F), Controle Vetorial (VWV), Vetorial Sensorless ou com Encoder, Controle de motores de Ímãs Permanentes (VWV PM);
- ▶ Módulos plug-in selecionáveis;
- ▶ Filosofia Plug & Play;

- ▶ Interface de operação e programação (IHM) incorporada;
- ▶ Porta RS-485 incorporada (em qualquer módulo plug-in selecionado);
- ▶ Pump Genius: funções dedicadas para o acionamento de bombas;
- ▶ Funções de Segurança: STO e SS1 nível SIL 3 / PLe;
- ▶ SoftPLC - funcionalidades de CLP incorporadas;
- ▶ IGBT de frenagem (disponível a partir do tamanho B);
- ▶ Proteção classe 3C2 ou 3C3 para aplicações em ambientes com agentes químicos corrosivos;
- ▶ Grau de proteção IP20 (padrão), NEMA1 ou IP66 (opcional);
- ▶ Ventilador com sistema de troca rápida;
- ▶ Filtro RFI interno (opcional);
- ▶ Porta de comunicação USB (acessório CFW500-CUSB);
- ▶ Softwares de programação gratuitos.

E como principais recursos, podemos citar:

- ▶ Senha para a proteção da programação
- ▶ Unidades especiais de engenharia (RPM, °C, Nm, mA, %, kW, kWh, entre outros)
- ▶ Backup de todos os parâmetros (via software WPS, cartão de memória ou memória do CFW500)
- ▶ Possibilidade de salvar até 2 programações diferentes na memória do CFW500
- ▶ Ajuste da frequência de chaveamento conforme a necessidade da aplicação
- ▶ Referência de velocidade via potenciômetro eletrônico
- ▶ Multispeed com até 8 velocidades programáveis
- ▶ Compensação de escorregamento
- ▶ Boost de torque manual ou automático (modo escalar V/F) ou autoajuste (modo vetorial VVW)

- ▶ Rampas de aceleração/desaceleração
- ▶ Rampa tipo "S"
- ▶ Frenagem CC
- ▶ Frenagem reostática
- ▶ Regulador PID para controle de processos em malha fechada
- ▶ Flying start / ride through
- ▶ Estado sleep
- ▶ Frequências ou faixas de frequências evitadas
- ▶ Proteção de sobrecarga e sobretemperatura no motor e nos IGBTs
- ▶ Proteção de sobrecorrente
- ▶ Supervisão da tensão do link CC
- ▶ Histórico de falhas

### 3.1.3 Motor de indução trifásico Weg

Foi utilizado um MIT da marca Weg (Figura 3.2).



**Figura 3.2** – Motor de indução trifásico Weg

A Figura 3.3 traz a placa de identificação do motor e a Figura 3.4 mostra seus terminais de ligação e os diagramas de ligação em delta (220V) ou estrela (380V). As principais características do motor são:

**Potência nominal** - 0,37 kW ou 0,50 cv

**Velocidade de rotação nominal** - 1720 RPM

**Fator de serviço** - 1,15

**Razão entre corrente de pico de corrente nominal** -  $I_p/I_n = 5,0$

**Grau de proteção** - IP55

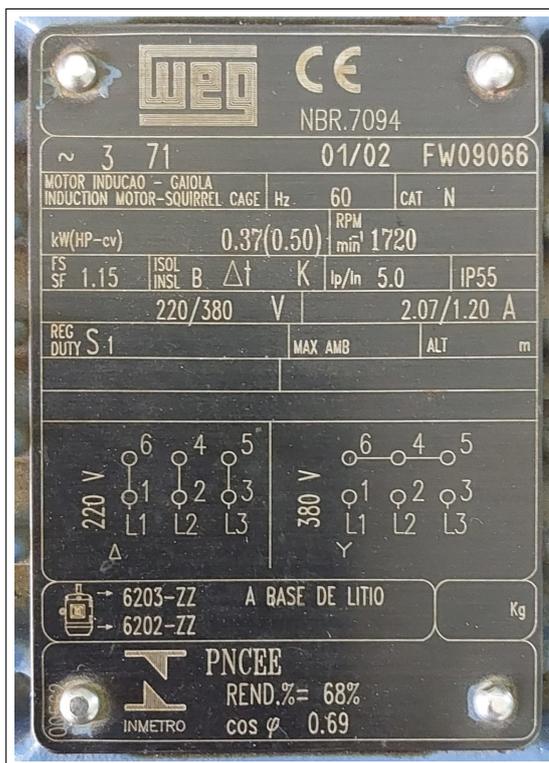
**Tensão nominal** - 220/380 V

**Corrente nominal** - 2,07/1,20 A

**Regime de serviço** -  $S_1$

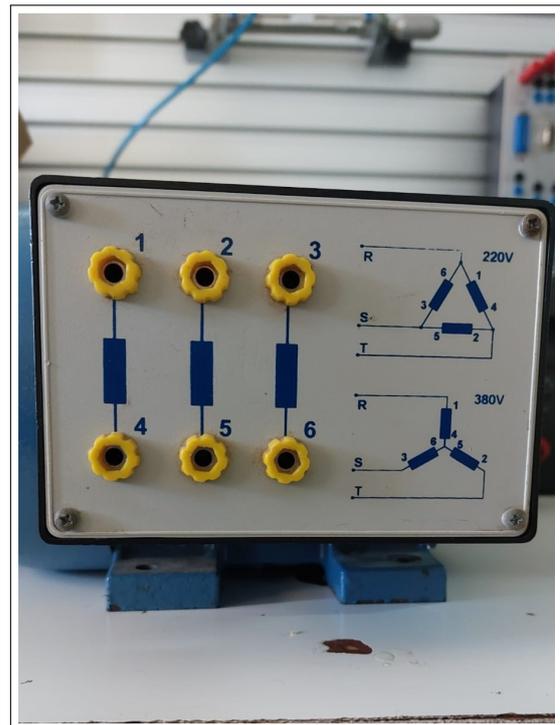
**Rendimento** - 68

**Fator de potência** -  $\cos(\phi) = 0,69$



FONTE: Autoria própria

**Figura 3.3** – Placa de identificação do motor



FONTE: Autoria própria

**Figura 3.4** – Terminais de ligação do motor

### 3.1.4 Conversor RS-485 para USB

Para realizar a comunicação entre a interface serial RS-485 do CFW500 com o computador, foi necessário utilizar um conversor. O modelo escolhido foi o mostrado na Figura 3.5.



**Figura 3.5** – Cabo conversor USB serial RS-485

Segundo (MAKERHERO, 2023), suas principais características são:

- ▶ Permite criar uma conexão Serial RS485 a partir de uma porta USB
- ▶ Placa de 6 pinos com terminais para facilitar conexão com jumpers
- ▶ Transferência de dados até 10Mb/s
- ▶ Proteção anti-estática
- ▶ Conexão Serial RS485 tipo macho com rosca para fixação
- ▶ Comprimento do cabo 60cm

## 3.2 Diagrama de ligação

A Figura 3.6 mostra o diagrama de ligação dos componentes do sistema.

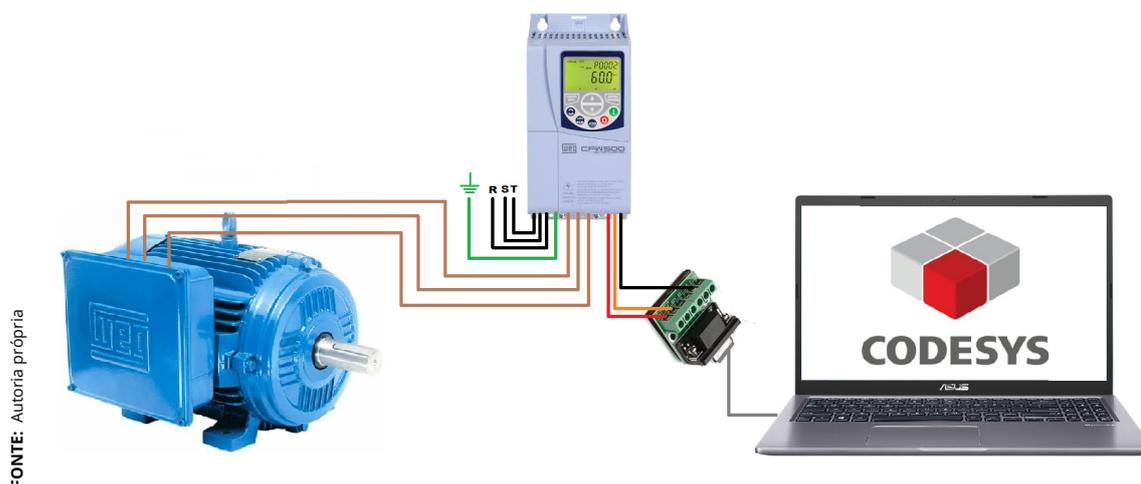


Figura 3.6 – Diagrama de ligação dos componentes

## 3.3 Desenvolvimento do código

O código foi desenvolvido em texto estruturado (ST).

## 3.4 Criação da interface gráfica

A interface gráfica foi criada no CODESYS, utilizando o recurso de Visualização disponível na plataforma. As ferramentas utilizadas foram: botões, *slider*, mostradores, luzes sinalizadoras e textos.

## 3.5 Parametrização do conversor de frequência

O conversor de frequência foi parametrizado conforme manuais do fabricante (Weg). O manual de programação (WEG, 2016) possui informações acerca de todos os parâmetros disponíveis no dispositivo e há também um manual específico para a utilização do protocolo Modbus com o CFW500 (WEG, 2011).

## 3.6 Ligação entre os componentes

Para ligar o motor ao CFW500, foram utilizados cabos com plugues do tipo banana-banana. A ligação do motor foi feita em delta, pois o conversor de frequência que, por sua vez, foi ligado à rede elétrica através de um plugue com três pinos (fase, terra e neutro), possui alimentação em 220 V. Foi utilizado também um disjuntor entre a rede e o conversor de frequência. Para conectar o CFW500 ao conversor USB serial RS-485 foram utilizados três *jumpers*<sup>1</sup>: um para o sinal positivo, outro para o sinal invertido e outro para o GND. A extremidade USB foi ligada diretamente ao computador.

## 3.7 Protocolo Modbus

Para estabelecer a comunicação entre o CODESYS e o inversor de frequência através do protocolo Modbus, foi utilizado como meio físico a interface RS-485 por meio de um cabo conversor serial para USB. No ambiente do CODESYS é possível adicionar um dispositivo Modbus e configurar de acordo com as necessidades específicas do projeto.

---

<sup>1</sup>Apesar de o ideal para a utilização da interface serial RS-485 ser o cabo de par trançado, neste trabalho a distância entre os equipamentos é curta e a utilização de *jumpers* não tem grande influência no resultado.

# Capítulo 4

## Desenvolvimento

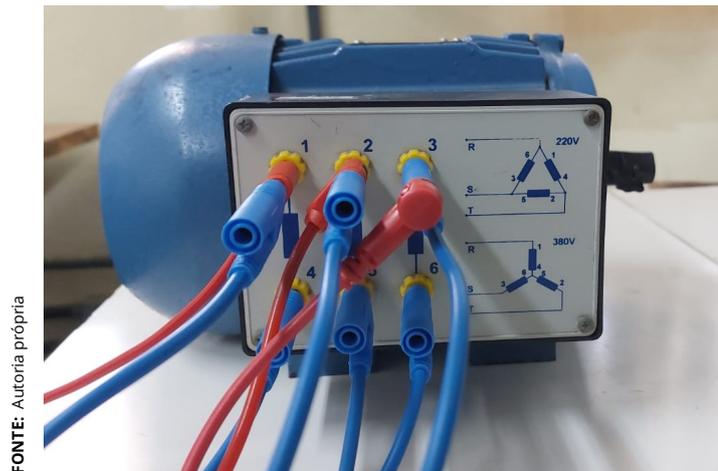
### 4.1 Conexão dos componentes

O conversor de frequência CFW500 é o equipamento central do sistema. Ele é alimentado pela rede com tensão 220 V, através de um disjuntor, e conectado ao motor trifásico a partir de três cabos. Ele possui três pontos para ligação para a interface RS485, numerados da seguinte forma: 12 é para o sinal positivo, 14 para o sinal invertido e 16 para o GND. A Figura 4.1 mostra o CFW500 já com os cabos conectados.



**Figura 4.1** – Conexões do CFW500

Os terminais do motor foram ligados em delta como visto na Figura 4.2.



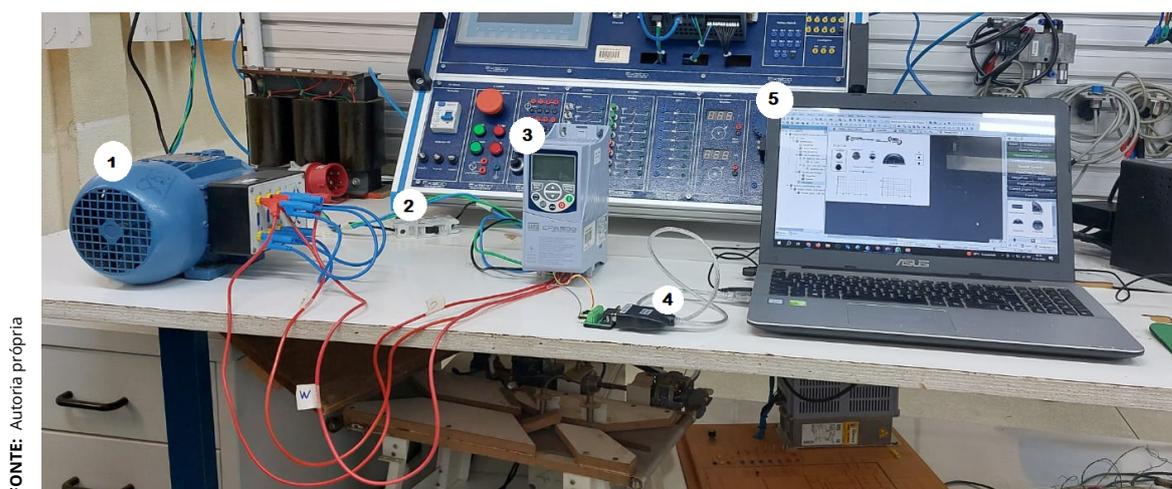
**Figura 4.2** – Ligações dos terminais do MIT

O conversor RS485 para USB foi ligado de acordo com a Figura 4.3.



**Figura 4.3** – Ligações do conversor Serial para USB

Por fim, o cabo USB do conversor foi ligado ao computador e o sistema completo é mostrado na Figura 4.4, onde 1 é o MIT, 2 é o disjuntor, 3 é o CFW500, 4 é o conversor serial para USB e 5 é o *notebook* com o CODESYS.



FONTE: Autoria própria

Figura 4.4 – Sistema completo

## 4.2 Parametrização do CFW500

Antes de comunicar o conversor de frequência com o CODESYS foi necessário configurar vários parâmetros. Primeiramente, é preciso parametrizar o CFW500 de acordo com o motor utilizado. Os parâmetros foram configurados de acordo com a Tabela 4.1.

Tabela 4.1 – Parâmetros CFW500 para inserção dos dados do motor

Parâmetro	Descrição	Valor
P0296	TENSÃO DE REDE	0 = 200 - 240 V
P0398	FATOR DE SERVIÇO DO MOTOR	1.15
P0399	RENDIMENTO NOMINAL DO MOTOR	68%
P0400	TENSÃO NOMINAL DO MOTOR	220 V
P0401	CORRENTE NOMINAL DO MOTOR	2.01
P0403	FREQUÊNCIA NOMINAL DO MOTOR	60
P0404	POTÊNCIA NOMINAL DO MOTOR	3 = 0,50 HP (0,37 kW)
P0407	FATOR DE POTÊNCIA DO MOTOR	0.69

FONTE: (WEG, 2016) Manual de Programação do CFW500

Depois, é preciso fazer a parametrização para habilitar a comunicação serial via protocolo Modbus. A Tabela 4.2 traz todos os parâmetros configurados.

Tabela 4.2 – Parametrização Modbus RTU

Parâmetro	Descrição	Valor
P0105	SELEÇÃO 1ª/2ª RAMPA	3 = Serial/USB
P0220	SELEÇÃO FONTE LOCAL/REMOTO	6 = Serial/USB (REM)
P0221	SELEÇÃO REFERÊNCIA LOCAL	9 = Serial/USB
P0222	SELEÇÃO REFERÊNCIA REMOTA	9 = Serial/USB
P0223	SELEÇÃO GIRO LOCAL	5 = Serial/USB (H)
P0224	SELEÇÃO GIRA/PARA LOCAL	2 = Serial/USB
P0225	SELEÇÃO JOG LOCAL	3 = Serial/USB
P0226	SELEÇÃO GIRO REMOTO	5 = Serial/USB (H)
P0227	SELEÇÃO GIRA/PARA REMOTO	2 = Serial/USB
P0228	SELEÇÃO JOG REMOTO	3 = Serial/USB
P0308	ENDEREÇO SERIAL	1
P0310	TAXA DE COMUNICAÇÃO SERIAL	0 = 9600 bits/s
P0311	CONFIGURAÇÃO DOS BYTES DA INTERFACE SERIAL	0 = 8 bits de dados, sem paridade, 1 stop bit
P0312	PROTOCOLO SERIAL	2 = Modbus RTU (1)
P0313	AÇÃO PARA ERRO DE COMUNICAÇÃO	1 = Para por Rampa

FONTE: (WEG, 2011) Manual do usuário Modbus do CFW500

## 4.3 Programação do CODESYS

### 4.3.1 Iniciando um novo projeto

Ao iniciar um novo projeto no CODESYS, são disponibilizados alguns modelos. Neste trabalho, o escolhido foi o *Standard Project*, ou Projeto Padrão, que começa já com um dispositivo, uma aplicação, e uma implementação vazia para o código do CLP, chamado PLC\_PGR, como visto na Figura 4.5.

Como foi dito na sessão 2.5, o CODESYS possui um recurso chamado CODESYS Control, um sistema de tempo de execução capaz de fazer com que um dispositivo haja como um controlador. Neste caso, foi utilizado o CODESYS Control Win V3 x64. A sua seleção é a primeira escolha a ser feita ao iniciar um novo projeto no CODESYS, juntamente com a linguagem de programação principal que será utilizada, como visto na Figura 4.6.

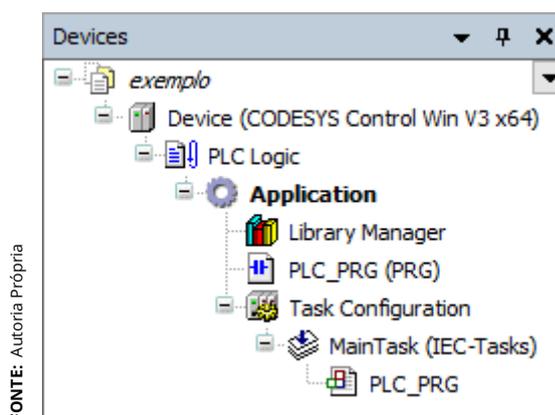


Figura 4.5 – Menu Devices

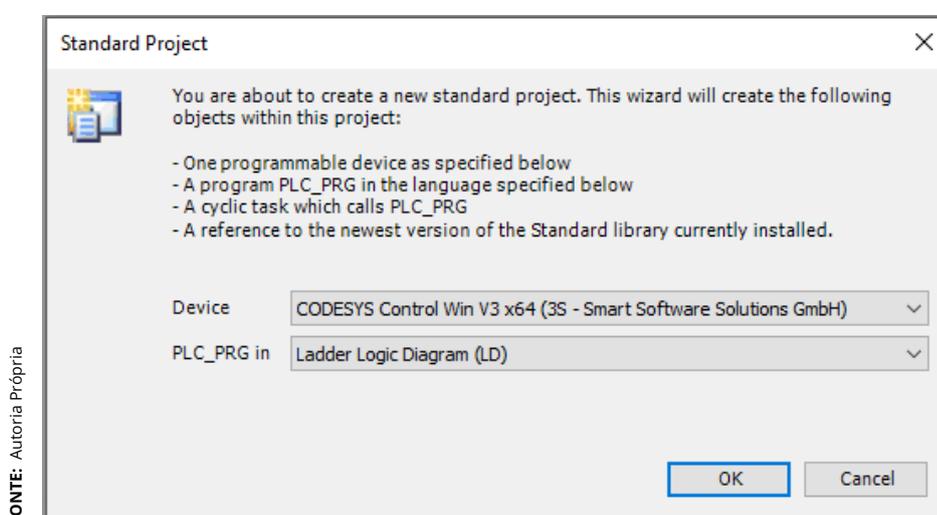


Figura 4.6 – Caixa de diálogo inicial do Standard Project

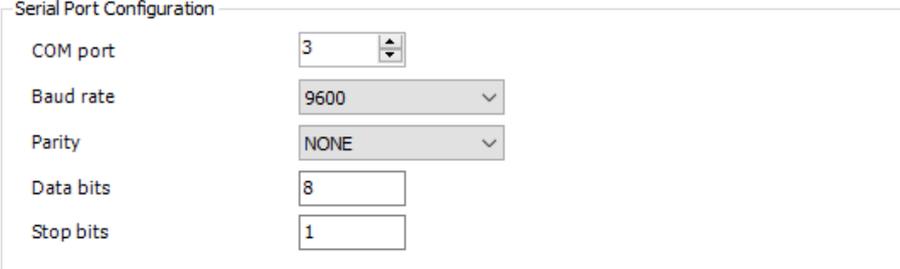
### 4.3.2 Estabelecendo a comunicação entre o CFW500 e o CODESYS

A plataforma CODESYS suporta de forma nativa a utilização do protocolo Modbus para transmitir mensagens entre componentes. Neste trabalho, o mestre da rede é o CLP configurado na plataforma CODESYS e o escravo é o conversor de frequência CFW500.

Para fazer uso do protocolo Modbus, primeiro é preciso adicionar e configurar a interface RS-485 ao dispositivo CODESYS Control Win. Isso pode ser feito através do menu *Add device*, depois escolher a opção *Modbus* e, abaixo dela, a opção *Modbus Serial Port* e *Modbus COM*. Como visto na Figura 4.7, é possível configurar a porta serial, selecionando em qual porta do computador está conectado o RS-485 em *COM port* que foi a porta COM 3 no computador utilizado, a velocidade de

transmissão em *Baud rate*, a paridade em *Parity* (par, ímpar ou sem paridade), a quantidade de bits de dados em *Data bits* e a quantidade de bits de parada em *Stop bits*. Tal configuração deve ser feita de acordo com a opção escolhida no momento da parametrização do CFW500.

FONTE: Autoria Própria



Field	Value
COM port	3
Baud rate	9600
Parity	NONE
Data bits	8
Stop bits	1

**Figura 4.7** – Configuração da porta serial

Depois, é preciso adicionar e configurar o dispositivo mestre da rede. A única configuração a ser feita nele é a escolha do modo RTU, que é o modo de operação utilizado neste trabalho. Além disso, é possível modificar o tempo de resposta em *Response timeout* e o tempo entre os quadros em *Time between frames*, ambos dados em ms, como visto na Figura 4.8.

FONTE: Autoria Própria



Field	Value
Transmission mode	RTU
Response timeout (ms)	1000
Time between frames (ms)	10
Auto-restart communication	<input type="checkbox"/>

**Figura 4.8** – Configuração do mestre da rede

Por último, o escravo é adicionado. Para o escravo, deve-se indicar o seu endereço de acordo com o que foi parametrizado no conversor de frequência (Figura 4.9).

FONTE: Autoria Própria



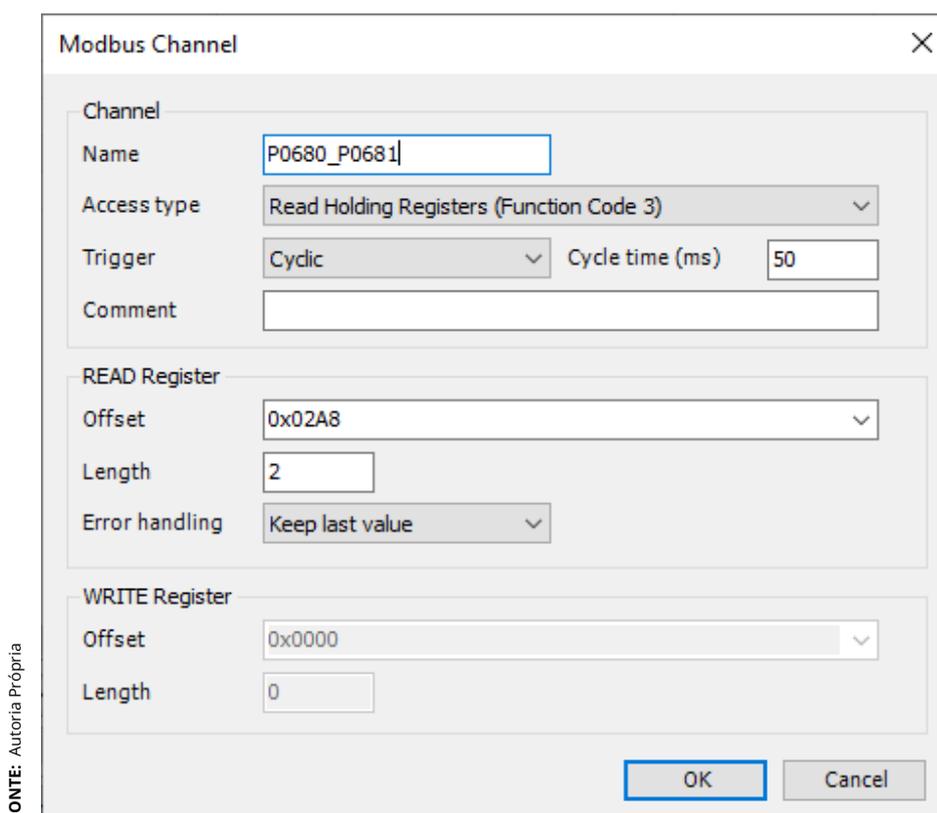
Field	Value
Slave address [1..247]	1
Response timeout [ms]	1000

**Figura 4.9** – Configuração do escravo da rede

Após a realização destes passos, já é possível iniciar a comunicação entre os dispositivos.

### 4.3.3 Criando os canais de comunicação mestre-escravo

Como visto na seção 2.3, a comunicação entre mestre e escravo de uma rede Modbus é feita através de troca de mensagens. No CODESYS, deve ser criado um canal para cada mensagem a ser enviada. A Figura 4.10 mostra uma tela de configuração de um canal, onde é possível colocar um nome, escolher qual o código de função utilizado em *Access type*, qual o gatilho para o envio da mensagem, que pode ser cíclico, borda de subida ou aplicação, em *Trigger*, o tempo do ciclo em *Cycle time*, em ms, o endereço do registrador, o tamanho, no caso em que o código de função pode ler ou escrever múltiplos registradores (quando esse é o caso, o registrador a ser indicado deve ser o primeiro, pois, ao utilizar múltiplos registradores, eles devem ter endereços subsequentes), e, quando é uma função de leitura, escolher o que fazer quando houver um erro, onde as opções são manter o último valor lido ou igualar a zero.



FONTE: Autoria Própria

Figura 4.10 – Criando um canal Modbus

No CFW500, os registradores do tipo *holding* tem seu endereço igual ao número do parâmetro, só que em hexadecimal. Por exemplo, o registrador inicial na Figura 4.10 foi o parâmetro P0680, sendo que 680 em hexadecimal é igual a 0x02A8. No CODESYS, não é necessário fazer essa conversão, pois, ao escrever

o número em decimal, a própria plataforma converte para hexadecimal. Como o objetivo desse canal era fazer a leitura de dois registradores, o código de função escolhido foi o 3, *Read Holding Registers*, que lê múltiplos registradores, e o tamanho foi modificado para 2.

### 4.3.4 Desenvolvimento do código

Por padrão, o arquivo onde é feita a escrita do código no CODESYS é o PLC\_PRG. Ele é dividido em duas partes: um para a declaração das variáveis e outra para o desenvolvimento da lógica de programação. Neste trabalho, como dito no Capítulo 3, a linguagem de programação escolhida foi ST. Foram declaradas variáveis para cada parâmetro do CFW500 a ser lido ou escrito, para os botões que habilita o sistema e o que faz com que o motor gire ou pare, assim como variáveis auxiliares para o controle da velocidade do motor, leitura de dados e estado do inversor, como visto no trecho de código 4.1.

**Código 4.1** – Declaração de variáveis

```
1 PROGRAM PLC_PRG
2 VAR
3     BT_LIGA_H       : BOOL ;
4     BT_HABILITA     : BOOL ;
5     P680            : WORD ;
6     P682            : WORD ;
7     P683            : WORD ;
8     P003            : WORD ;
9     P005            : WORD ;
10    P007            : WORD ;
11    SET_VEL         : WORD ;
12    CORRENTE        : REAL ;
13    FREQUENCIA      : REAL ;
14    TENSAO          : WORD ;
15    HABILITADO      : BOOL ;
16    GIRANDO         : BOOL ;
17    EM_ALARME      : BOOL ;
18    EM_FALHA       : BOOL ;
19 END_VAR
```

A lógica desenvolvida para o controle e monitoramento do MIT foi a mostrada no código 4.2.

**Código 4.2** – Código principal

```
1 P682 .0      := BT_LIGA_H ;
2 P682 .1      := BT_HABILITA ;
3 P683         := SET_VEL ;
4 CORRENTE     := WORD_TO_REAL(P003)/10 ;
5 FREQUENCIA   := WORD_TO_REAL(P005)/10 ;
6 TENSAO       := P007 ;
7 HABILITADO   := P680 .9 ;
8 GIRANDO      := P680 .8 ;
9 EM_ALARME    := P680 .7 ;
10 EM_FALHA    := P680 .15 ;
```

Ao adicionar os canais para leitura/escrita de parâmetros no CODESYS, o tipo de dado é, por padrão, WORD. No entanto, observa-se que os dados de corrente e frequência são lidos como um número inteiro sem separação de casas decimais, ou seja, é como se o valor real estivesse multiplicado por 10. Para resolver esse problema, é necessário converter os dados para REAL (através do operador WORD\_TO\_REAL) e dividir o resultado por 10. Então, por exemplo, se a corrente do motor lida é 1,4 A, o programa receberá 14, no formato WORD, na variável P003, que será convertido para REAL e dividido por 10, resultando em 1,4, valor que é atribuído à variável CORRENTE. O parâmetro P007, que lê a tensão de saída do motor, é lida pelo CFW500 como um número inteiro, então não foi necessário fazer quaisquer operações para corrigir seus dados.

Alguns parâmetros do CFW500 são compostos por vários bits que precisam ser acessados separadamente, a depender da necessidade. Para acessar cada bit de um parâmetro no CODESYS, basta indicar no código qual o bit desejado com a utilização de um ponto. Por exemplo, se o objetivo for ler o bit 0 do parâmetro P0680, cuja variável relacionada é P680, basta escrever P680.1. Neste trabalho, os estados de interesse foram: Em Falha (bit 15), Em Alarme (bit 7), Habilitado Geral (bit 9) e Girando (bit 8).

O parâmetro P0680 do CFW500 diz respeito ao seu estado lógico. Ele tem 15 bits e cada um representa um estado, como visto na Figura 4.11.

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3 a 0
Função	Em Falha	Automático (PID)	Subtensão	LOC/REM	JOG	Sentido de Giro	Habilitado Geral	Motor Girando	Em Alarme	Em modo de configuração	Segunda Rampa	Parada Rápida Ativa	Reservado

Figura 4.11 – Bits do parâmetro P0680

O parâmetro P0682 é a palavra de comando via serial. Ele também possui 15 bits, sendo que cada um deles representa um comando que pode ser executado pelo CFW500 (Figura 4.12). No código, os comandos transmitidos para o conversor de frequência são o Habilita Geral (bit 1) e o Gira/Para (bit 0).

Bits	15 a 8	7	6	5	4	3	2	1	0
Função	Reservado	Reset de Falhas	Parada Rápida	Utiliza Segunda Rampa	LOC/REM	JOG	Sentido de Giro	Habilita Geral	Gira/Para

Figura 4.12 – Bits do parâmetro P0682

A velocidade é definida através da variável SET\_VEL, que transmite o valor para a variável P683, que está ligada ao parâmetro P0683 do CFW500, que é a referência de velocidade via serial.

Para que o código funcione de forma correta, é preciso mapear os canais criados para a comunicação Modbus (seção 4.3.3). Isso é feito acessando a configuração do escravo no CODESYS, mais especificamente a seção *ModbusGenericSerialSlave I/O Mapping*. Lá estarão listados todos os canais criados, sendo possível relacionar uma variável do seu código a cada um deles. A Figura 4.13 mostra os canais já mapeados. No caso dos parâmetros P0682 e P0683, foi usado um só código de função para escrever em ambos os registradores (*Write Multiple Registers*). Por isso, eles ficaram aninhados em um só canal.

Variable	Mapping	Channel	Address	Type	Unit	Description
Application.PLC_PRG.P...		P0003	%IW0	ARRAY [0..0] OF WORD		Read Holding Registers
Application.PLC_PRG.P...		P0005	%IW1	ARRAY [0..0] OF WORD		Read Holding Registers
Application.PLC_PRG.P...		P0007	%IW2	ARRAY [0..0] OF WORD		Read Holding Registers
Application.PLC_PRG.P...		P0680	%IW3	ARRAY [0..0] OF WORD		Read Holding Registers
		P0682_P0683	%QW0	ARRAY [0..1] OF WORD		Write Multiple Registers
Application.PLC_PR...		P0682_P0683[0]	%QW0	WORD		0x02AA
Application.PLC_PR...		P0682_P0683[1]	%QW1	WORD		0x02AB

FONTE: Autoria própria

Figura 4.13 – Mapeamento dos canais de comunicação Modbus

### 4.3.5 Criação da interface gráfica

A interface gráfica também foi desenvolvida no CODESYS, utilizando o recurso *Visualization*. Foram utilizados LEDs para indicar o estado do conversor de frequência, um botão do tipo *pushbutton* para a função Habilita Geral, um botão do tipo alavanca para os comandos Gira e Para, um *slider* para a referência de velocidade, um mostrador para indicar a frequência de saída do motor e dois gráficos para monitorar a corrente do motor e a tensão de saída do motor.

Cada elemento da visualização foi relacionado a uma variável do programa. A tabela 4.3 relaciona cada variável com um item da interface gráfica.

Tabela 4.3 – Elementos visuais e suas respectivas variáveis

Variável	Descrição	Elemento
BT_LIGA_H	Gira/Para	
BT_HABILITA	Habilita Geral	
SET_VEL	Referência de Velocidade	
HABILITADO	Habilitado Geral	
GIRANDO	Girando	
EM_ALARME	Em Alarme	
EM_FALHA	Em Falha	

FONTE: Autoria própria

## 4.4 Resultados, Ensaios e Testes do Protótipo

A interface gráfica final é mostrada na Figura 4.14.

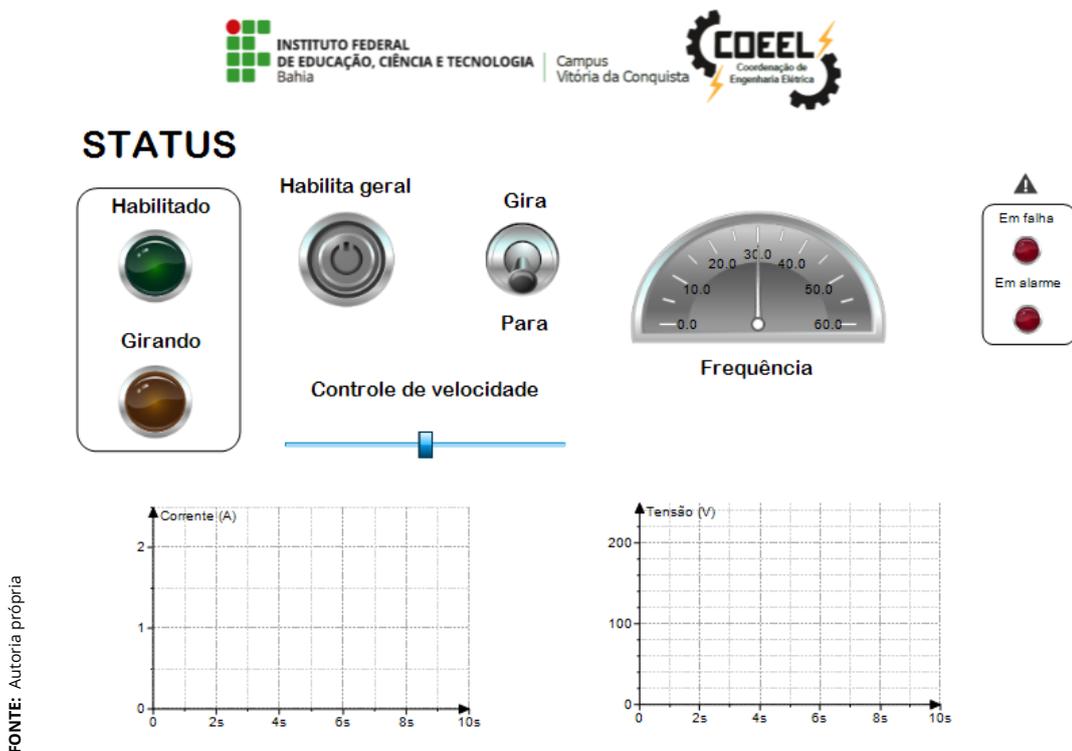


Figura 4.14 – Interface gráfica

O botão Habilita Geral é o que habilita o CFW500 para receber comandos e enviar dados. Ao acioná-lo, o equipamento envia uma mensagem para o CODESYS informando que ele está habilitado, o que acende o LED Habilitado da interface. Ao colocar a alavanca do comando Gira/Para na posição Gira, o motor começa a girar em uma frequência de 3 Hz, que é o padrão do equipamento. A frequência mínima inicial pode ser mudada nos parâmetros do conversor de frequência. Quando o motor está girando, o CODESYS recebe essa informação e acende o LED Girando. O mostrador com o rótulo Frequência mostra a frequência atual em que o motor está girando. Na interface existem também dois gráficos com a grandeza tempo (inicialmente em segundos, mas, a medida em que avança após 60 s, a escala é ajustada automaticamente para minutos) no eixo horizontal. O gráfico da esquerda tem em seu eixo vertical a corrente elétrica do motor, em Amperes, enquanto o da direita tem a tensão de saída do motor, em Volts.

Para o controle de velocidade do motor é utilizado o *slider* azul. Caso o motor seja iniciado sem alterar a posição desse *slider*, o motor irá girar na frequência

mínima configurada, como explicado anteriormente. Ao deslizar o cursor, a velocidade do motor é alterada de forma praticamente simultânea. Caso o cursor seja colocado em uma posição diferente da inicial com o motor ainda parado, essa velocidade fica armazenada para ser a velocidade inicial quando o comando Gira for acionado. A Figura 4.15 mostra um exemplo disso, quando temos a frequência atual do motor igual a 0.0, pois está parado, mas é possível ver no canto superior direito o número 30.0, que representa a velocidade que foi selecionada no *slider* quando o motor ainda estava parado, igual a 30 Hz.



**Figura 4.15** – CFW500 com velocidade armazenada

Para validar os resultados do trabalho, além de verificar se o funcionamento do motor ocorreu de acordo com o esperado, também foi possível verificar os dados vistos na IHM integrada do CFW500. Na tela inicial da IHM é exibida a frequência atual do motor e o *setpoint* de frequência, ou seja, qual a frequência a ser atingida. Também é possível navegar pelos parâmetros de leitura através do teclado integrado, sendo possível verificar qual a corrente, a tensão e a velocidade do motor, o que facilita o processo de validação dos dados mostrados na IHM remota.

A Figura 4.16 mostra a tela no momento em que o motor está girando em uma frequência de aproximadamente 30 Hz. As luzes de sinalização Girando, Habilitado, Em Alarme e Em Falha estão ligadas à palavra de estado lógico do CFW500. É possível atestar o seu funcionamento correto observando o *status* Girando, pois o conversor de frequência está configurado para realizar a desaceleração do mo-

tor por rampa, então quando a alavanca é movida para a posição Para, o LED não apaga instantaneamente, ele só apaga quando o motor, de fato, não está mais girando.

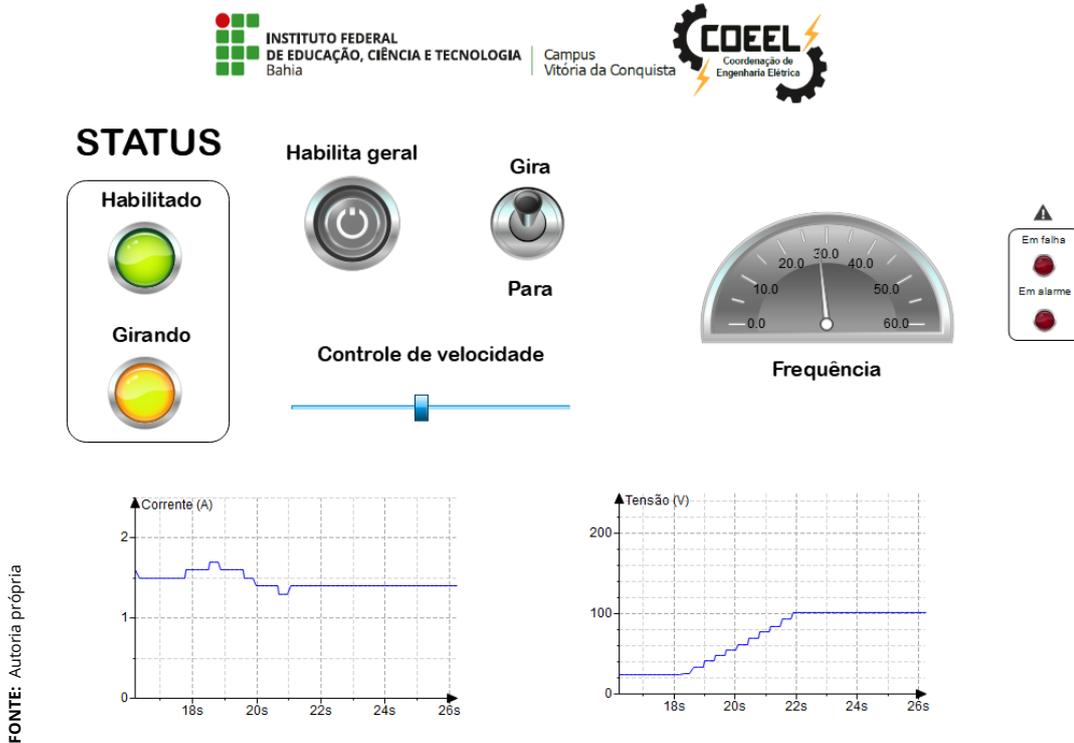


Figura 4.16 – Interface gráfica com motor girando

Foi possível monitorar o comportamento da corrente elétrica do motor e da tensão de saída do motor através dos gráficos. A Figura 4.17 mostra os gráficos quando o motor está rodando na sua frequência mínima de funcionamento.

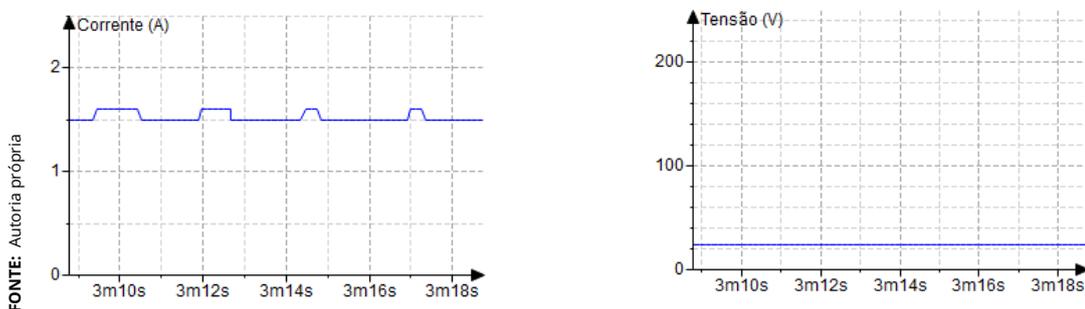


Figura 4.17 – Gráficos do motor rodando na frequência 3 Hz

A Figura 4.18 mostra os gráficos quando o motor está girando com 50% da sua velocidade nominal e a Figura 4.19 mostra quando a velocidade é máxima. É possível observar que a corrente do motor se mantém estável, aproximadamente 1,4 A. Já a tensão varia com a frequência. Observa-se que a corrente não é com-

pletamente linear quando o motor está rodando em sua frequência mínima configurada, mas em frequências mais altas, não é possível detectar oscilações, até mesmo quando ele está acelerando, como é possível ver na Figura 4.19, onde é possível ver rampa de aceleração no gráfico da tensão. A imagem foi registrada enquanto o motor acelerava desde 50% da sua velocidade nominal até o seu máximo. É possível observar que a tensão de saída do motor acompanha a proporção, pois o seu valor nominal é igual a 220 V e verifica-se que quando a frequência é 30 Hz, a tensão está em torno de 110 V e quando a frequência é 60 Hz, a tensão chega a 220 V.

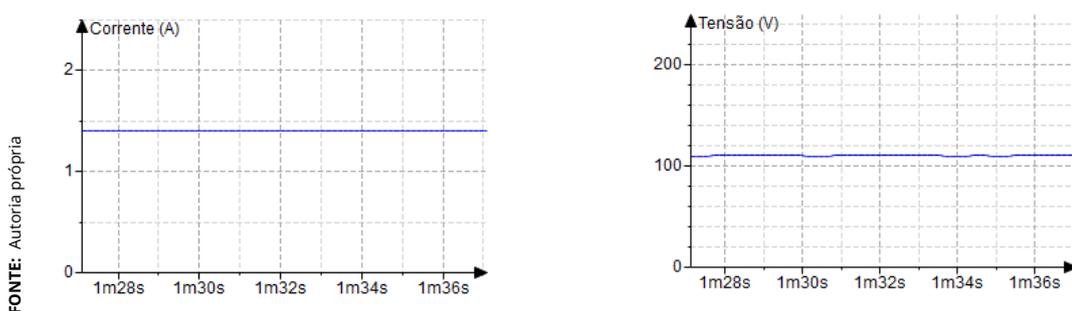


Figura 4.18 – Gráficos do motor rodando na frequência 30 Hz

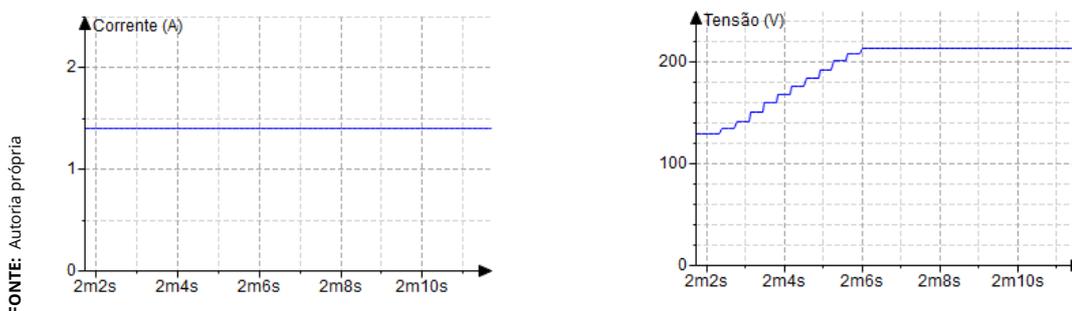


Figura 4.19 – Gráficos do motor rodando na frequência 60 Hz

É possível verificar que o motor acelera e desacelera em rampa através do gráfico da tensão de saída do motor. A Figura 4.20 mostra os gráficos quando o motor está acelerando até sua frequência máxima de 60 Hz. Já a Figura 4.21 mostra o caminho inverso, ou seja, a desaceleração de 60 Hz até o motor parar completamente. Estes testes foram feitos colocando o *slider* de controle de velocidade no máximo com o motor parado e acionando o comando Girar e, depois que o motor chegou a sua velocidade máxima, foi acionado o comando Para. Em ambos é possível observar um pico de corrente. No caso da aceleração, ele acontece na partida, enquanto na desaceleração ele ocorreu quando o motor estava na iminência da parada completa. No entanto, observa-se que a variação da corrente não

foi grande e isso se deu por conta da utilização do conversor de frequência, que limita a corrente de partida. A corrente média verificada durante o experimento foi de 1,4 A. A maior corrente registrada na partida foi de aproximadamente 1,7 A. Em um acionamento via partida direta, a corrente do motor pode atingir 8 vezes o seu valor nominal, o que pode gerar problemas na rede e sobrecarga no sistema.

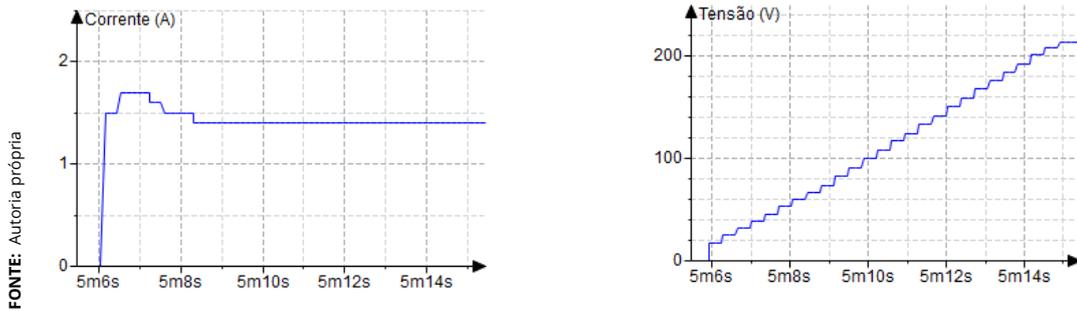


Figura 4.20 – Gráficos do motor acelerando

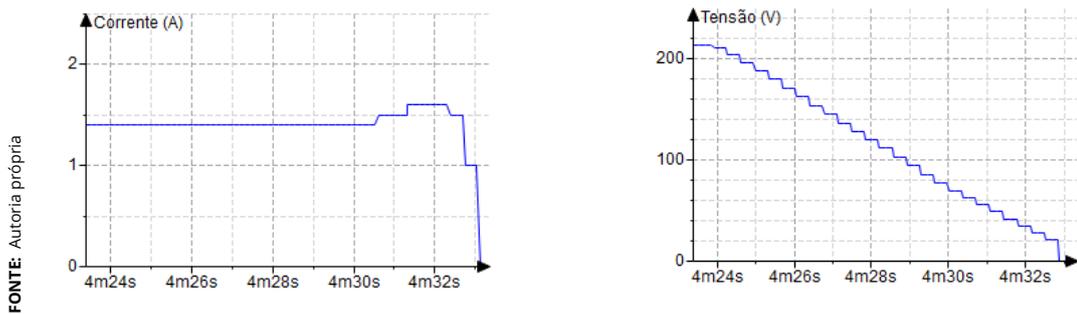


Figura 4.21 – Gráficos do motor desacelerando

# Capítulo 5

## Considerações Finais

Neste trabalho, foram alcançados os principais objetivos propostos. Foi desenvolvida com sucesso uma Interface Homem-Máquina (IHM) remota que possibilita o acionamento, controle da velocidade e monitoramento da frequência, corrente e tensão de um motor de indução trifásico. O uso do conversor de frequência CFW500 da Weg e a comunicação via Modbus RTU provaram ser uma abordagem eficaz para a automação e controle do sistema.

Foi detalhada a implementação da IHM remota, descrevendo como o conversor de frequência CFW500 da Weg foi integrado ao sistema, bem como a configuração da comunicação Modbus RTU via interface RS485. Além disso, foram apresentados os componentes da plataforma CODESYS utilizados no controle e monitoramento do sistema.

Os resultados obtidos durante o desenvolvimento e testes do sistema foram satisfatórios. A IHM permitiu o controle preciso da velocidade do motor de indução, enquanto os parâmetros de frequência, corrente e tensão puderam ser monitorados de forma eficaz. Os dados coletados demonstraram que o sistema opera conforme o esperado, atendendo aos requisitos de desempenho e confiabilidade.

A automação industrial é um campo em constante evolução, impulsionado pela necessidade de maior eficiência, controle preciso e redução de custos operacionais. Nesse contexto, a implementação bem-sucedida desta IHM remota demonstrou que a integração de tecnologias avançadas, como a comunicação Modbus RTU e a plataforma CODESYS, pode trazer benefícios tangíveis para a indústria.

O projeto oferece uma solução eficaz para o controle e monitoramento de

---

motores de indução trifásicos, permitindo ajustes precisos de velocidade e fornecendo informações em tempo real sobre a frequência, corrente e tensão. Isso não só otimiza os processos industriais, mas também melhora a segurança, pois permite o monitoramento remoto de operações críticas.

Além disso, a flexibilidade dessa IHM remota abre portas para uma variedade de aplicações em diferentes setores da indústria, contribuindo para a competitividade e a inovação. A capacidade de controle e monitoramento à distância proporciona uma maior agilidade na tomada de decisões e na manutenção preditiva, minimizando tempos de parada não programados e reduzindo custos operacionais.

Em resumo, este projeto demonstra a relevância da automação industrial moderna e destaca o potencial transformador da tecnologia quando aplicada de forma inteligente e eficaz.

# Capítulo 6

## Sugestões para Trabalhos Futuros

Para aprimorar ainda mais a eficácia e a abrangência deste projeto, algumas direções para trabalhos futuros podem ser consideradas:

1. **Integração com banco de dados:** Uma possível extensão deste trabalho seria a implementação de um sistema que envie os dados coletados, como frequência, corrente e tensão, para um banco de dados em tempo real. Isso permitiria o uso de técnicas de análise de dados e aprendizado de máquina para realizar a manutenção preditiva do motor. Ao monitorar tendências e anomalias nos dados, seria possível identificar potenciais problemas antes que ocorram falhas, economizando recursos e aumentando a disponibilidade do sistema.

2. **Testes de falha do motor:** Para garantir a robustez e a confiabilidade do sistema, seria interessante realizar testes de falha do motor em condições controladas. Isso poderia envolver a simulação de falhas, como sobrecargas, desequilíbrios ou variações abruptas de carga, e avaliar como a IHM remota responde a essas situações. Os resultados desses testes contribuiriam para o aprimoramento da segurança e da capacidade de resposta do sistema em situações críticas.

Essas sugestões representam oportunidades valiosas para expandir o escopo e a utilidade deste projeto, explorando novas dimensões da automação industrial e da manutenção de motores de indução trifásicos

# REFERÊNCIAS

CASTRO, A. C. de. *Motor de Indução Trifásico (MIT)*. 2003. <<http://www.ifba.edu.br/PROFESSORES/castro/MIT.pdf>> [Acessado em 16 de Maio de 2023]. ( Citado na página 7.)

EPE. *Anuário Estatístico de Energia Elétrica 2022 (ano base 2021)*. 2022. <<https://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/anuario-estatistico-de-energia-eletrica>> [Acessado em 13 de Maio de 2023]. ( Citado 2 vezes nas páginas 2 and 3.)

FRAZÃO, M. R. L. *Acionamento de Máquinas Elétricas: Partidas Estáticas*. São Paulo: Érica, 2012. ( Citado na página 10.)

GEEKSFORGEES. *Cyclic Redundancy Check and Modulo-2 Division*. 2023. <<https://www.geeksforgeeks.org/modulo-2-binary-division/>>, Acessado em 08 de Junho 2023. ( Citado na página 15.)

GROUP, C. *CODESYS Control*. 2023. <<https://www.codesys.com/products/codesys-runtime/control.html>>, Acessado em 10 de Agosto de 2023. ( Citado na página 18.)

HECKLER, M. V.; VENANCIO, P. A. *Soft-Starters: Princípios e Aplicações*. Porto Alegre: Bookman, 2016. ( Citado na página 10.)

LIANG, Q.; LI, L. The study of soft plc running system. *Procedia Engineering*, v. 15, 2011. ( Citado na página 7.)

LOPES, I. M.; BENEVENUTO, F. M.; OLIVEIRA, F. M. de. As camadas do modelo osi: Revisitando suas funcionalidades e respectivos protocolos. *Revista Interdisciplinar do Pensamento Científico*, v. 1, 2015. ( Citado na página 16.)

MAKERHERO. *Cabo Conversor USB Serial RS485*. 2023. <<https://www.makehero.com/produto/cabo-conversor-usb-serial-rs485/>>, Acessado em 27 de junho de 2023. ( Citado na página 24.)

MATHEUS, Y. *O modelo OSI e suas camadas*. 2023. <<https://www.alura.com.br/artigos/conhecendo-o-modelo-osi/>>, Acessado em 23 de Julho 2023. ( Citado na página 17.)

MENDES, D. R. *Redes de computadores: Teoria e Prática*. São Paulo: Novatec, 2007. (Citado 2 vezes nas páginas 16 and 17.)

MOHAN, N.; UNDELAND, T. M.; ROBBINS, W. P. *Power Electronics: Converters, Applications, and Design*. 2. ed. Nova Iorque: John Wiley Sons, 2002. (Citado na página 11.)

NASCIMENTO, J. M. A. do; LUCENA, P. B. de. Protocolo modbus. *UFRN - Redes para Automação Industrial*, LECA-DCA-UFRN Universidade Federal do Rio Grande do Norte, v. 1, p. 4, 2003. (Citado 2 vezes nas páginas 12 and 13.)

ORGANIZATION, M. *MODBUS Application Protocol Specification V1.1b3*. [S.l.], 2012. <[https://www.modbus.org/docs/Modbus\\_Application\\_Protocol\\_V1\\_1b3.pdf](https://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf)>, Acessado em 8 de Junho de 2023. (Citado 3 vezes nas páginas xii, 12, and 15.)

PEREIRA, A. M. M. P. *Soluções industriais para visualização de dados em tempo-real*. Tese (Doutorado) — Universidade do Porto, 2018. (Citado na página 6.)

ROUSE, M. *Runtime System*. 2013. <<https://www.techopedia.com/definition/24023/runtime-system>>, Acessado em 10 de Agosto de 2023. (Citado na página 18.)

UMANS, S. D. *Máquinas elétricas de Fitzgerald e Kingsley*. 7. ed. Porto Alegre: AMGH, 2014. (Citado na página 8.)

UNITRONICS. *Samba™ series*. 2023. <<https://www.unitronicsplc.com/programmable-controllers-samba-series/>>, Acessado em 14 de Agosto de 2023. (Citado na página 6.)

WEG. *Manual do Usuário Modbus RTU*: Série: Cfw500. Jaraguá do Sul – SC, São Paulo – SP, 2011. (Citado 4 vezes nas páginas 13, 17, 25, and 30.)

WEG. *Manual de Programação*: Série: Cfw500. Jaraguá do Sul – SC, São Paulo – SP, 2016. (Citado 4 vezes nas páginas 20, 25, 29, and 36.)

WEG. *Vista Explodida do Motor Trifásico W22*. 2021. <<https://static.weg.net/medias/downloadcenter/he2/hd3/WEG-vista-explodida-do-motor-trifasico-w22-premium-efficiency-ie3-50009253-banner-porpdf>> [Acessado em 16 de Maio de 2023]. (Citado na página 7.)

WEG. *Inversor de Frequência CFW500*. 2023. <[https://www.weg.net/catalog/weg/BR/pt/Automa%C3%A7%C3%A3o-e-Controle-Industrial/Drives/Inversores-de-Frequ%C3%A4ncia/Drives-para-OEMs-e-Uso-Geral/Inversor-de-Frequ%C3%A4ncia-CFW500/Inversor-de-Frequ%C3%A4ncia-CFW500/p/MKT\\_WDC\\_BRAZIL\\_PRODUCT\\_INVERTER\\_CFW500](https://www.weg.net/catalog/weg/BR/pt/Automa%C3%A7%C3%A3o-e-Controle-Industrial/Drives/Inversores-de-Frequ%C3%A4ncia/Drives-para-OEMs-e-Uso-Geral/Inversor-de-Frequ%C3%A4ncia-CFW500/Inversor-de-Frequ%C3%A4ncia-CFW500/p/MKT_WDC_BRAZIL_PRODUCT_INVERTER_CFW500)>, Acessado em 28 de maio de 2023. (Citado na página 20.)

WEIS, O. *Guia da comunicação RS485*. 2021. <<https://www.eltima.com/pt/article/rs485-communication-guide/>>, Acessado em 23 de Julho 2023. (Citado na página 17.)