



## DESENVOLVIMENTO DE UM PROGRAMA DE CLP PARA CONTROLE E SUPERVISÃO DE VÁLVULAS ATUADORAS DE UMA ESTAÇÃO DE TRATAMENTO DE EFLUENTES

Vitor Machado de Costa<sup>1</sup>

Wagner Fernandes Zeferino<sup>2</sup>

**Resumo:** Com a 4º revolução industrial em seu ápice, os CLPs (Controladores Lógicos Programáveis) são, cada vez mais, implantados nos processos industriais. Este trabalho tem como objetivo o desenvolvimento de um programa de CLP para o controle e supervisão das válvulas atuadoras de uma estação de tratamento, substituindo então todo o controle operacional das válvulas por um único ponto de controle e supervisão. Dessa forma as bibliografias principais do trabalho se baseiam em linguagens de programação, comunicação de redes industriais e sistemas de supervisão. Foi feito um estudo das válvulas atuadoras, definição do software para programação, modelo de CLP e criando uma base de parâmetros para o desenvolvimento do programa. Com isso foi possível o desenvolvimento completo para o programa em que, mesclando as linguagens de programação, utilizando as variáveis da rede de comunicação e criando as telas de supervisão, o operador consiga controlar e supervisionar as suas válvulas atuadoras de uma central de controle.

**Palavras-chave:** CLP, Modbus-RTU, válvula atuadora e sistema de supervisão.

### 1 INTRODUÇÃO

Após tantas mudanças advindas das revoluções industriais, passa-se pela 4º revolução industrial que se denomina Indústria 4.0, cujo enfoque está na informação e controle das máquinas.

O controle das máquinas é feito a partir de um CLP, equipamento eletrônico utilizado em sistemas de automação e controle industrial, com aplicação em alta escala no mercado. Uma característica que torna o CLP flexível é a possibilidade que ele oferece de desenvolver e alterar facilmente a lógica para o acionamento das saídas em função das entradas (MICALOSKI,2012).

---

<sup>1</sup> Graduando em Engenharia Elétrica, ano 2022: E-mail: vitormdecosta@gmail.com

<sup>2</sup> Prof. do Centro Universitário UniSATC. E-mail: wagner.zeferino@satc.edu.br



Atualmente, os sistemas de automação industrial utilizam tecnologias de computação e comunicação para automatizar o monitoramento e controle dos processos industriais, efetuando coleta de dados em ambientes complexos e a respectiva apresentação de modo amigável para o operador, com recursos gráficos elaborados, através de uma Interface Homem Máquina (GONÇALVES, 2005).

A empresa STEP Automação precisa desenvolver uma solução de automação de processo. A aplicação se dá ao controle remoto de 15 válvulas atuadoras devidamente separadas de uma estação de tratamento de efluentes onde o controle e monitoramento delas precisa ser feito através de uma única localização, representada por uma central de controle.

As máquinas estão localizadas em lugares distintos da estação. Dessa forma, o controle e os status das válvulas são monitorados por vários funcionários, propiciando confusão na troca de informação sobre as respectivas válvulas atuadoras.

O objetivo do trabalho visa desenvolver a programação do CLP, a partir da mescla das linguagens de programação texto estruturado e ladder, a fim de aplicar o controle e monitoramento das válvulas atuadoras, e conter um supervisório para fácil visualização e operação da planta pelo operador.

## **2 FUNDAMENTAÇÃO TEÓRICA**

### **2.1 INDÚSTRIA 4.0**

Indústria 4.0 é a revolução industrial que descreve a utilização de alta tecnologia em sua respectiva forma, utilizando um conjunto de tecnologias de ponta ligadas a internet e com o objetivo de tornar os sistemas de produção mais flexíveis, colaborativos e autônomos (BITKON et al., 2016).

Nesse cenário as máquinas empregam de forma automática, resultando na otimização e configuração delas, comunicam-se entre si e em alguns casos utilizam algoritmos com inteligência artificial para aprimoramento e execução da atividade, a fim de aumentar a eficiência dos processos (BAHRIN et al., 2016).

Com a utilização de sensores nos meios de fabricação, os âmbitos físico e virtual unem-se, levando a criação de sistemas físicos e cibernéticos. Como estão conectados à internet fica possível a interação uns com os outros por meio de protocolos de comunicação de rede, e através de seus algoritmos e programações

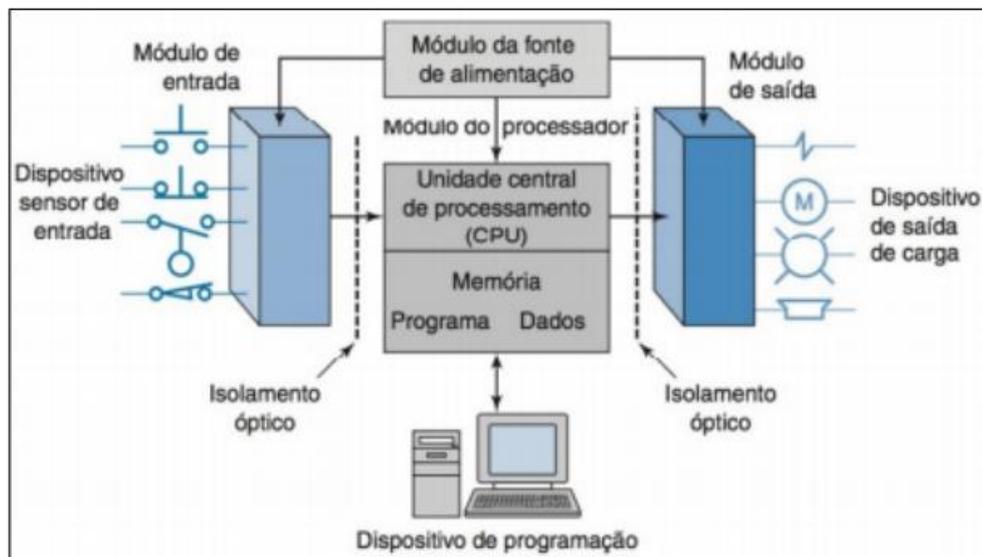
conseguem informar os dados de uma forma mais objetiva e ágil para os usuários e operadores (THE BOSTON CONSULTING GROUP, 2015).

## 2.2 CONTROLADOR LÓGICO PROGRAMÁVEL

Os controladores lógicos programáveis são atualmente a tecnologia mais utilizada para controle de máquinas e troca de dados com um supervisor. O CLP é nomeado informalmente de computador industrial podendo ser programado para efetuar funções de controle de máquina e verificação de informações por um meio físico no ambiente analisado (SOUSA, 2015).

Com a utilização desses controladores houve uma redução considerável no uso de fios para ligações das máquinas, também trouxeram outros benefícios como facilidade na instalação e manutenção, fácil programação, compatibilidade de rede, coleta e verificação de falhas e dados, tornando-se altamente confiável nos processos industriais (PETRUZELLA, 2014). A Fig. 1 mostra a estrutura de um CLP.

Figura 1: Estrutura de um CLP.



Fonte: Petruzella (2014).

Seu funcionamento ocorre através de um processo de laço. Após a inicialização, é feita a verificação das entradas e atualizações na memória, executa-se a programação e atualiza-se as saídas, depois retorna para a verificação das entradas fechando o laço. Tudo isso acontece de forma simultânea (SOUSA, 2015).

## 2.3 PROGRAMAÇÃO E LINGUAGEM DE PROGRAMAÇÃO

Linguagem é uma forma de transmissão de informações entre dois ou mais elementos com a capacidade de se comunicarem entre si. Programar significa abastecer uma série de instruções a um sistema computacional, em que se execute de forma automática as decisões de controle em função do estado atual, das entradas e das saídas do sistema (FRANCHI; CAMARGO, 2008).

### 2.3.1 Norma IEC 61131-3

A inexistência de normas em relação às linguagens de programação dos CLPs levou ao surgimento de inúmeras variantes destas ao longo do tempo, todas diferentes entre si. Felizmente a comunidade industrial internacional reconheceu que era necessário estabelecer um padrão aberto para os CLPs, visando a padronização de procedimentos de diversos fabricantes (FRANCHI; CAMARGO, 2008).

Para isso, foi criado um grupo de trabalho no IEC para estabelecer normas a todo o ciclo de desenvolvimento dos CLPs, incluindo o projeto de hardware, instalação, testes, documentação, programação e comunicação. No início da década de 1990, o IEC publicou várias partes da norma IEC 61131 que cobre o ciclo de vida completo dos CLPs (FRANCHI; CAMARGO, 2008).

De acordo com FRANCHI e CAMARGO 2008, alguns anos depois essa norma foi revisada e recebeu o número IEC 61131. Cujas terceira parte: IEC 61131-3 trata-se das linguagens de programação. Com o objetivo de simplificar a sua análise, é usual dividi-la em três seções:

- Generalidades;
- Elementos comuns;
- Linguagens de programação.

### 2.3.2 Entradas, saídas e memória do CLP

Os elementos mais importantes do CLP são as entradas, saídas e a memória interna. Somente através das entradas o CLP recebe os dados do ambiente

externo. De forma que o CLP só pode gerar o controle de algum dispositivo, através da conexão de suas saídas (FRANCHI; CAMARGO, 2008).

As variáveis que possibilitam o acesso direto das posições de memória dos CLPs. Uma posição de memória do CLP é identificada por três regiões lógicas (FRANCHI; CAMARGO, 2008). A primeira letra identifica a variável a ser mapeada, podendo ser a entrada, saída ou posição interna de memória, conforme a Qd. 1. A segunda letra demonstra o tipo de dado a ser coletado ou transferido, como mostra o Qd. 2. Os tipos de dados especificados aparecem no Qd. 3 juntamente com as suas respectivas faixas de valores.

Quadro 1: Mapeamento das posições de memória de um CLP.

Palavra-chave	Inglês	Português
I	Inputs	Entradas
Q	Outputs	Saídas
M	Memory	Memória

Fonte: Adaptado Franchi;Camargo (2008).

Quadro 2: Identificação do tipo de dado.

Segunda letra	Tipo do dado
X	Bit
B	Byte (8 bits)
W	Word (16 bits)
D	Double Word (32 bits)
I	Long Word (64 bits)

Fonte: Adaptado Franchi;Camargo (2008).

Quadro 3: Tipos de dado especificados pela norma IEC 61131-3.

Palavra-chave	Tipo de dado	Faixa de valores
BOOL	BOOLEANO	0 ou 1

SINT	CURTO INTEIRO	0 a 255
INT	INTEIRO	-32.768 a 32.767
DINT	DUPLO INTEIRO	- 2.147.483.648 a 2.147.483.647
UINT	INT. C/ SINAL	0 a 65.535
REAL	PONTO FLUT.	+/-2.9E-39 a +/-3.4E + 38
TIME	TEMPO REAL	Depende da implementação
STRING	STRING	Depende da implementação
BYTE	8 BITS	Faixa de valores não declarada
WORD	16 BTS	Faixa de valores não declarada

Fonte: Adaptado Franchi;Camargo (2008).

### 2.3.3 Linguagem de programação ladder

A linguagem Ladder foi a primeira que surgiu na programação dos CLPs, pois sua funcionalidade procura imitar os antigos diagramas elétricos. O objetivo é evitar uma quebra de padrão alta, permitindo assim a melhor aceitação do produto no mercado (DA SILVA, 2007).

O diagrama de contatos Ladder consiste em um desenho formado por duas linhas verticais, que representam os polos positivo e negativo de uma bateria, ou fonte de alimentação genérica. Entre as duas linhas verticais são desenhados ramos horizontais que possuem chaves. Estas podem ser normalmente abertas ou fechadas e representam os estados das entradas do CLP (DA SILVA, 2007). Como ilustrado na Fig. 2.

Figura 2: Lógica de programação ladder.



Fonte: Da Silva (2007).

### 2.3.4 Linguagem de programação texto estruturado

É uma linguagem textual de alto nível e muito robusta que contém essenciais elementos de uma linguagem de programação moderna, incluindo as



instruções condicionais (*THEN, ELSE, IF e CASE OF*) e instruções de iterações (*REPEAT, FOR e WHILE*) (FRANCHI; CAMARGO, 2008).

A programação estruturada é incentivada por ser excelente para o desenvolvimento de blocos funcionais complexos, os quais podem ser introduzidos em qualquer outra linguagem IEC. Das linguagens textuais é a mais potente, portanto a mais utilizada para aplicações complexas que envolvam um comportamento sequencial (FRANCHI; CAMARGO, 2008).

## 2.4 REDES INDUSTRIAIS

Redes industriais são sistemas distribuídos com diversos elementos que trabalham de forma simultânea com o objetivo de supervisionar e controlar um determinado processo. Uma troca rápida e precisa de informações entre sensores, atuadores, computadores, CLPs, entre outros (SANTOS, 2019).

As redes industriais surgem da necessidade de interligar computadores e controladores, permitindo o compartilhamento de recursos e bases de dados, que passam a ser únicos, conferindo segurança as informações (NOGUEIRA, 2009).

### 2.4.1 Modbus-RTU

Modbus-RTU é um dos protocolos de comunicação industrial mais utilizados, especialmente em aplicações de controle industrial, sistemas de gestão de energia e automação predial. Os sistemas implementados costumam usar o Modbus-RTU variante sobre barramentos seriais EIA/TIA-485 para interconectar a planta de dispositivos integrados via *gateways* com supervisão e sistemas de gestão (MODBUS-IDA, 2011).

Modbus-RTU é uma variante do popular protocolo de comunicação Modbus usado para transmitir comandos de controle industrial e dados no nível da planta. Suas especificações estão disponíveis em dois documentos técnicos elaborados pela *Modbus Organization*: (i) *Modbus Application Protocol*, que define a estrutura de mensagens em modo cliente/servidor na camada de aplicação; e (ii) *Modbus Serial Line Protocol*, que especifica uma arquitetura mestre/escravo para a camada de laço de dados, o *ASCII (American Standard Code for Information Interchange)* e modos de



transmissão RTU (unidade de terminal remoto), e os requisitos de implementação sobre as linhas EIA/TIA-485 e EIA/TIA232 (STEMMER, 2010).

De acordo com Dhinakaran (2016), em um modelo mestre escravo, a comunicação é sempre iniciada pelo mestre, e os nós escravos não se comunicam entre si. O mestre pode transmitir dois tipos de mensagens aos escravos, dentro de uma mesma rede:

- Mensagem tipo *unicast*: o mestre envia uma requisição ao escravo definido e este retorna uma mensagem-resposta ao mestre.
- Mensagem tipo *broadcast*: o mestre envia a requisição para todos os escravos, e não é enviada nenhuma resposta para o mestre. Existem dois modos de transmissão: ASCII (*American Code for Information Interchange*) e RTU (*Remote Terminal Unit*).

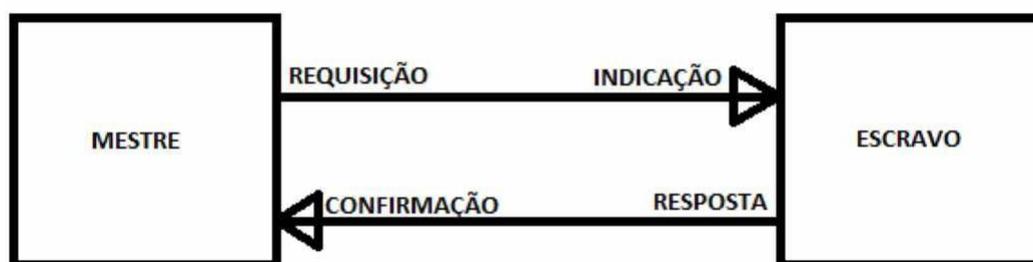
As comunicações mestre-escravo Modbus-RTU exigem que transações ocorram uma de cada vez. Especificamente, o mestre não envia uma nova solicitação até que uma resposta seja recebida de um escravo ou um problema de comunicação é detectado como resultado se não receber uma resposta (situação de tempo limite) (STEMMER, 2010).

Em um sistema Modbus-RTU típico, o mestre faz solicitações aos escravos de forma cíclica. Maneira em intervalos de tempo suficientemente longos para garantir que um escravo receba a informação, processe-a e envie sua resposta, que é então recebida e processada pelo mestre antes que o mestre faça uma nova solicitação a um escravo (REYNDERS; et al, 2005).

### 2.4.1.1 DADOS E ENDEREÇOS MODBUS RTU

O protocolo Modbus-RTU é baseado em quatro tipos de mensagens, são elas: requisição, confirmação, indicação e resposta. Representadas na Fig. 3 (MODBUS-IDA, 2006).

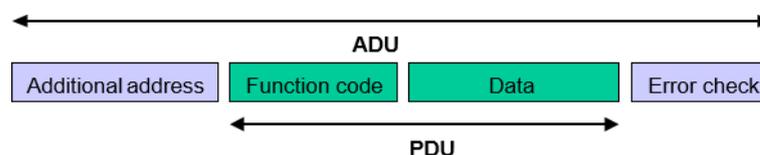
Figura 3: Tipos de mensagens Modbus-RTU.



Fonte: Adaptada de Modbus-IDA (2006).

Conforme Stemmer (2010), o protocolo Modbus define uma unidade de dados de protocolo (PDU) simples e independente das camadas de comunicação subjacentes. O mapeamento do protocolo Modbus em barramentos ou redes específicas pode introduzir alguns campos adicionais na unidade de dados da aplicação (ADU), visto na Fig. 4.

Figura 4: Application data Unit.



Fonte: Adaptada de Modbus-IDA (2006).

O código de função na requisição, informa ao dispositivo escravo endereçado qual o tipo de ação deve ser executado. Os bytes de dados contêm qualquer informação adicional que o escravo precisa para executar a função. O campo



de verificação de erros fornece um método para o escravo validar a integridade do conteúdo da mensagem (MODBUS-IDA, 2006).

Para resposta normal, o código de função na resposta é um eco do código de função na requisição consulta. O campo de verificação de erros permite ao mestre confirmar se o conteúdo da mensagem é válido. No nível da mensagem, o protocolo Modbus-RTU ainda aplica o princípio mestre-escravo, embora o método de comunicação de rede seja ponto-a-ponto (MODBUS-IDA, 2006).

## 2.5 SISTEMA SUPERVISORIO

Sistemas supervisórios servem como interface entre o computador e equipamentos eletrônicos como máquinas industriais, controladores automáticos e sensores dos mais variados tipos (ASSUNÇÃO, 2016).

O supervisório tem capacidade de ser multitarefa, essa característica faz com que ele execute várias tarefas ao mesmo tempo que são: sistema de aquisição de dados/banco de dados em tempo real e gerenciamento de alarmes (ZANNI, 2015).

Os componentes de um sistema físico de um sistema supervisório são classificados da seguinte forma: redes de comunicação, sensores e atuadores e estações remotas de aquisição/controle de dados (ASSUNÇÃO, 2016).

O controle e aquisição de dados dos equipamentos do processo, são advindos das estações remotas de entrada e saída. São compostos por diversos equipamentos como o CLP, para que o processo seja manipulado (ISA, 2016).

Essas estações de monitorização centrais são consideradas como unidades principais do sistema de supervisão, são responsáveis por reunir as informações geradas pelas estações e agir de acordo com os eventos acontecidos, podendo estar em uma unidade central de processamento ou distribuídos pela rede (GONÇALVES, 2005).

## 3 PROCEDIMENTO EXPERIMENTAL

Neste capítulo é feito o desenvolvimento do procedimento experimental para obter a base para os resultados do programa de CLP.

### 3.1 ESTUDO DA VÁLVULA ATUADORA COESTER

Foi feito um estudo minucioso da válvula atuadora a ser utilizada, demonstrada na Fig. 5, para através deste estudo, definir o modelo de CLP e consequentemente o software de programação que serão utilizados.

Figura 5: Válvula atuadora CSR6T.



Fonte: Do Autor (2022).

A válvula atuadora CSR6T permite conexão a uma rede ModBus-RTU ou ModBus ASCII utilizando uma placa específica de comunicação Modbus conforme Fig. 6. Esta placa possui duas conexões para a rede e funciona como um repetidor. Desta forma, é possível construir uma rede em topologia de anel.

Figura 6: Placa de comunicação Modbus.



Fonte: Do autor (2022)

### 3.1.1 Definição dos endereços do mapa Modbus-RTU para a aplicação

A interface do atuador é composta por duas áreas distintas de memória. Uma área para escrita e outra para leitura. Na área de leitura, estão disponíveis o estado atual do atuador, programação atual e as curvas de torque armazenadas na memória interna. Na área de escrita, estão disponíveis os comandos de atuação e posição desejada.

No Qd. 4, estão definidos os comandos de escrita da válvula, a parte de controle e comando, com seu respectivo tipo de dado e sua respectiva função.

Quadro 4: Endereço de comando de escrita da rede Modbus-RTU.

Word	Área
1	Bits de comando
	0 – abrir
	1 – fechar
	2 – parar
	3 – ESD
	4 – posicionar
	5 – spare
	6 – preset FC fechamento
	7 – preset FC abertura
8 – quitar alarmes	
2	Posição desejada
3-5	Reserva

Fonte: Do autor (2022).

No Qd. 5, estão definidos os comandos de leitura da válvula, a parte de entrada de dados, com seu respectivo tipo de dado e sua respectiva função.

Quadro 5: Endereço de comando de leitura da rede Modbus-RTU.

Word	Área
1	Bits de status 1 0 – modo de operação local 1 – modo de operação remoto 2 – modo de operação desligado 3 – posição aberta 4 – posição fechada 5 – posição intermediária 6 – movimento de abertura 7 – movimento de fechamento 8 – atuador (válvula) parado 9 – motor energizado 10 – movimento manual 11 – aberto com torque 12 – fechado com torque 13 – modo Modbus
2	Bits de status 2 0 – falta de fase 1 – sobreaquecimento do motor 2 – atuador sem programação 3 – parada local acionado
3	Bits de alarme 1 0 – modo de operação não remoto 1 – parada local de emergência 2 – parada antecipada por torque na abertura 3 – parada antecipada por torque na abertura 4 – atuador (válvula) travado 5 – operação incompleta 6 – sobreaquecimento do motor 7 – falha no motor ao energizar 8 – motor energizado após fim de curso 9 – motor energizado após comando de parada 10 – falha de comando 11 – falta de fase 12 – atuador não programado
4	Bits de alarme 2 0 – tensão de alimentação alta 1 – tensão de alimentação baixa 2 – tensão de bateria baixa 3 – 4 – fim de curso mecânico ultrapassado na abertura 5 – fim de curso mecânico ultrapassado no fechamento 6 – fim de curso programado ultrapassado na abertura 7 – fim de curso programado ultrapassado no fechamento 8 – cartão anybus não detectado
5	Posição atual

Fonte: Do autor (2022).

### 3.2 DEFINIÇÃO DO SOFTWARE

O software que melhor se adequou para essa aplicação foi o **e!COCKPIT** da marca Wago, que fornece espaços de trabalho modernos e ferramentas operacionais, relações complexas entre os dispositivos de rede, podendo ser identificadas de forma intuitiva e simplificada a interação dos dispositivos.

O **e!COCKPIT** oferece opções amplas de linguagens de programação para o desenvolvimento de software: Texto Estruturado (ST), Lista de Instruções (IL), Diagrama de Bloco de Funções (FBD) e Diagrama Ladder (LD) e todas as linguagens de programação podem ser combinadas umas com as outras.

Um dos pontos cruciais para escolha do software de programação seria o Webvisu, área do software **e!COCKPIT** que emprega a visualização das telas do programa com o operador em uma tela de computador pela interação com HTML5 que é suportado nesse software. A Fig. 7 demonstra a interface do operador, se comportando como um supervisor e não necessitando de uma ihm.

Figura 7: Demonstração do Supervisor integrado Wago (Webvisu)

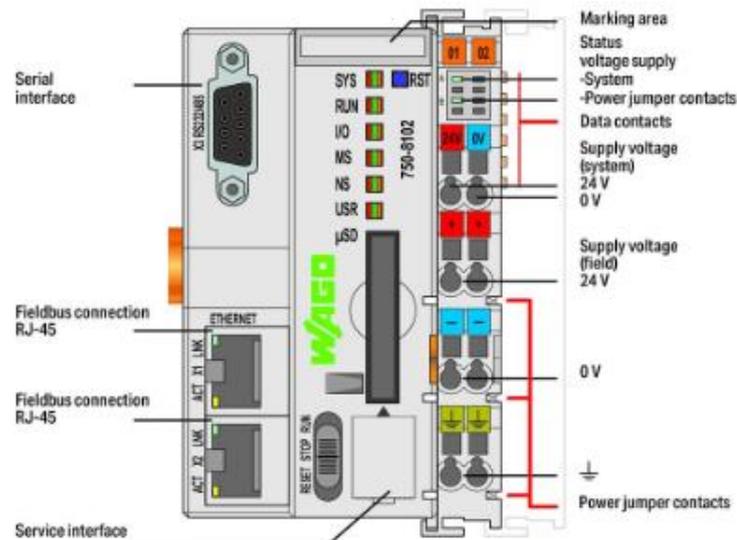


Fonte: Wago Br (2022).

### 3.2.1 Definição do modelo de CLP

A linha que foi utilizada de CLP, seria o modelo PFC100 750-8102 demonstrado na Fig. 8, é um CLP compacto modular I/O System da WAGO. Esse modelo é o que estava disponibilizado em estoque da empresa para implementação do software e que possui as características técnicas de suportabilidade para a comunicação de rede Modbus-RTU entre o CLP e controle das válvulas para o desenvolvimento do programa.

Figura 8: PFC100 750-8102

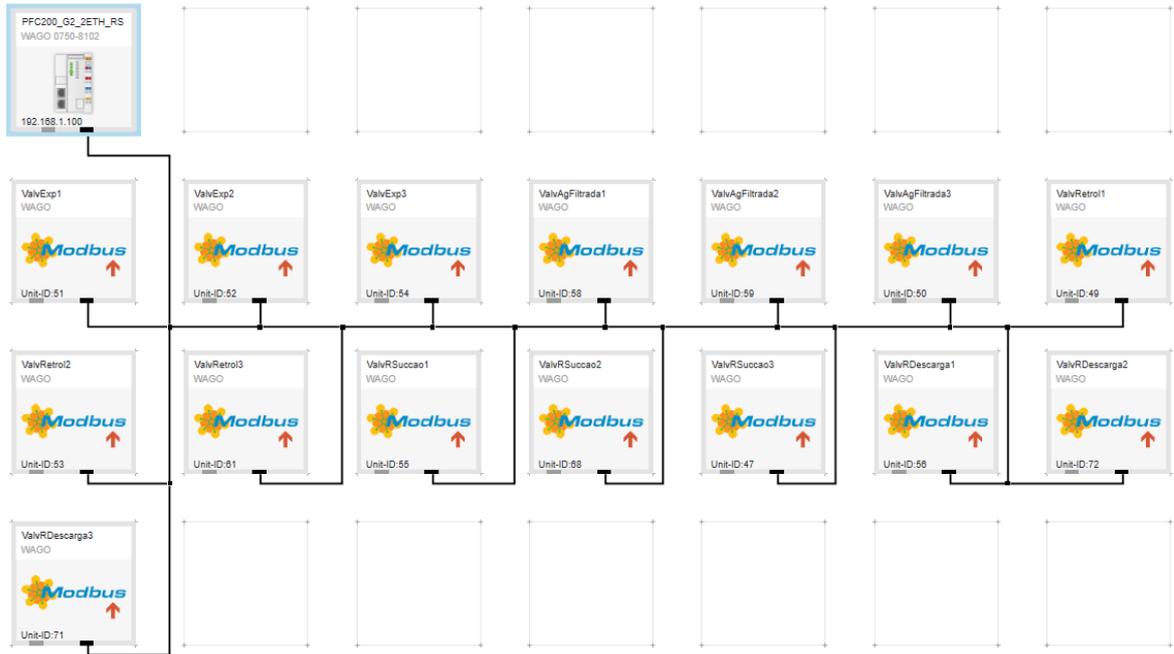


Fonte: Wago Br (2022).

### 3.3 MONTAGEM DO MAPA E ENDEREÇAMENTO MODBUS-RTU NO e!COCKPIT

Para coleta de informações, definições de variáveis de escrita e leitura e organização do programa, é preciso montar a topologia de rede Modbus-Rtu entre CLP e as válvulas atuadoras com os respectivos endereços e nomenclaturas de cada válvula. Na Fig. 9 temos a demonstração da topologia de rede Modbus-Rtu.

Figura 9: Topologia da rede Modbus-RTU no e!COCKPIT.



Fonte: Autor (2022).

Após a montagem do mapa Modbus-RTU para as válvulas atuadoras e CLP, foi preciso definir as variáveis a serem coletadas e transmitidas, descritas nos Qd. 4 e Qd. 5, para o desenvolvimento do programa de controle e supervisão das válvulas. Na fig. 10 observa-se as setes variáveis que foram utilizadas para a troca de dados entre o CLP e válvula.

Figura 10: Variáveis para troca de dados.

Network > ValvExp1

Data points

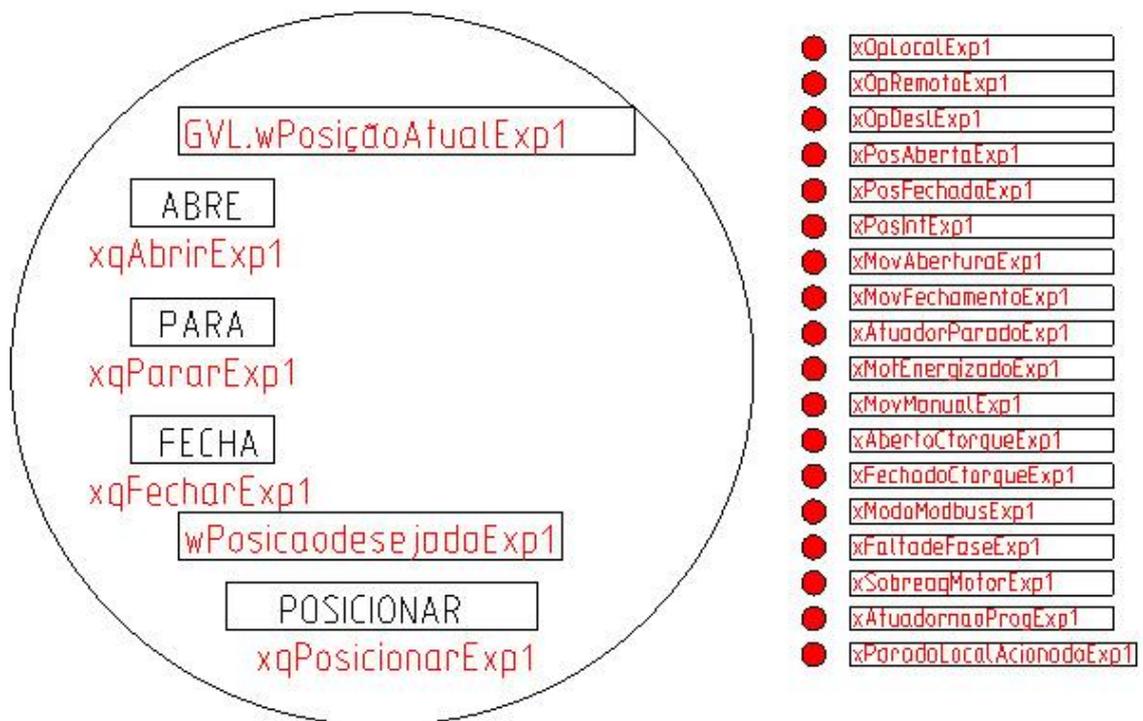
Name	Data type	Length of the array	Modbus address
Generic data points			
→ w1	WORD		write: register 0
→ w2	WORD		write: register 1
•→ w1Status	WORD		read: register 0
•→ w2Status	WORD		read: register 1
•→ w3Alarme	WORD		read: register 2
•→ w4Alarme	WORD		read: register 3
•→ w5PosicaoAtual	WORD		read: register 4

Fonte: Autor (2022).

### 3.5 ENDEREÇAMENTO E VARIÁVEIS PARA O SISTEMA DE SUPERVISÃO

Informações a serem expostas na tela do supervisor são imprescindíveis para o entendimento do operador e funções de comando executadas por ele. Dessa forma foi criado um esboço com tags (variáveis) e informações a serem usadas para o controle e status da válvula, conforme mostra a Fig. 11.

Figura 11: Esboço de telas para variáveis e informações do sistema de supervisão.



Fonte: Autor (2022).

## 4 RESULTADOS E DISCUSSÃO

No capítulo de resultados e discussões, se tem uma discussão em relação aos resultados obtidos e comparações aos objetivos a serem alcançados.

### 4.1 DECLARAÇÃO DE VARIÁVEIS NO PROGRAMA

Os elementos mais importantes do CLP são as entradas, saídas e a memória interna. Através das entradas o CLP recebe os dados do ambiente externo. De forma que o CLP só pode gerar o controle de algum dispositivo, através da

conexão de suas saídas. De forma similar, o CLP só pode controlar algum dispositivo, se este estiver conectado em uma de suas saídas. Para o desenvolvimento do programa ocorreu as definições de todas as variáveis, para o controle e supervisão das válvulas com os seus respectivos endereços. Com intuito de facilitar a visualização das variáveis foi feito uma sequência de descrição nominal para melhor visualização de cada variável, conforme Fig. 12.

Figura 12: Declaração das variáveis da Válvula de expurgo 1.

```

VAR_GLOBAL
  xqAbrirExpl:BOOL;
  xqFecharExpl:BOOL;
  xqPararExpl:BOOL;
  xqPosicionarExpl:BOOL;
  xqQuitaraLarmesExpl:BOOL;
  xqVerifFalhaComExpl:BOOL;
  xOpLocalExpl:BOOL;
  xOpRemotoExpl:BOOL;
  xOpDeslExpl:BOOL;
  xMovAberturaExpl:BOOL;
  xMovFechamentoExpl:BOOL;
  xAtuadorParadoExpl:BOOL;
  xModoModbusExpl:BOOL;
  xFaltaFaseExpl:BOOL;
  xSobreaqMotorExpl:BOOL;
  xAtuadorSemProgExpl:BOOL;
  xParadaLocalAcionadoExpl:BOOL;
  xModoOpNaoRemotoExpl:BOOL;
  xParadaLocalEmergExpl:BOOL;
  xParadaAntTorqAberExpl:BOOL;
  xParadaAntTorqFechExpl:BOOL;
  xAtuadorTravExpl:BOOL;
  xOpIncompExpl:BOOL;
  xSobreaqueMotorExpl:BOOL;
  xFalhaMotEnergExpl:BOOL;
  xMotEnergFimdeCursoExpl:BOOL;
  xMotEnergComParadaExpl:BOOL;
  xFalhaComandoExpl:BOOL;
  xFaltadeFaseExpl:BOOL;
  xAtuadornaoProgExpl:BOOL;
  xTensaoAliAltaExpl:BOOL;
  xTensaoAliBaixaExpl:BOOL;
  xTensaoBateriaBaixaExpl:BOOL;
  xFimdeCursoMecUltraAberExpl:BOOL;
  xFimdeCursoMecUltraFechExpl:BOOL;
  xFimdeCursoProgUltraAberExpl:BOOL;
  xFimdeCursoProgUltraFechExpl:BOOL;
  xCartAnyBusnaoDetecExpl:BOOL;
  wPosicaoAtualExpl:WORD;
  xPosAbertaExpl:BOOL;
  xPosFechadaExpl:BOOL;
  xPosIntExpl:BOOL;
  xMotEnergizadoExpl:BOOL;
  xMovManualExpl:BOOL;
  xAbertoCtorqueExpl:BOOL;
  xFechadoCtorqueExpl:BOOL;

```

Fonte: Autor (2021).

Pensando que possa ocorrer uma falha energética ou curto-circuito dentro da parte física do CLP, foi definida algumas variáveis persistentes, onde o programa do CLP define que mesmo se o controlador lógico for desligado ou ocorrer alguma falha, ele não perderá o valor daquela variável que está sendo utilizada, essas variáveis estão definidas na Fig. 13.

Figura 13: Declaração das variáveis persistente da Válvula de expurgo 1.

```

VAR_GLOBAL PERSISTENT RETAIN
  wPosicaodesejadaExpl:WORD;
  wFimdeCursoAberturaExpl:WORD;
  wFimdeCursoFechamentoExpl:WORD;
  tTempoReset: TIME;

END_VAR

```

Fonte: Autor (2022).

Como existe uma quantidade elevada de variáveis para cada atuadora, nas Fig. 12 e Fig. 13 representou-se as variáveis da Válvula de expurgo 1, entretanto foram replicadas para todas as outras 14 válvulas atuadoras. Em questão de nomenclatura e interpretação das variáveis estão descritas da seguinte forma:

- Entradas (x.....).
- Saídas (xq.....).
- Memória (xAux... ou xqAux...).
- Word (w.....).
- Time(t.....).

#### 4.2 PROGRAMAÇÃO PARA CRIAÇÃO DOS BLOCOS DE FUNÇÃO.

É uma linguagem gráfica de programação que permite um desenvolvimento hierárquico e modular no programa, uma vez que podem ser estruturados blocos de funções mais complexos a partir de outros menores e mais simples. São um meio de programação que permite especificar algoritmos ou conjunto de ações aplicados aos dados de entrada. No desenvolvimento do programa foram criados três blocos de função. Dois seriam para o tratamento de dados das informações que são recebidas e enviadas da válvula e o CLP. Isso foi criado devido à informação da válvula atuadora para CLP ser transmitida no tipo de dado *word* e ser controlado pelo tipo de dado booleano, respectivamente a válvula atuadora aceita apenas a informação em *word* se fazendo necessária a conversão dos dados booleanos para *word*. O nome definido no programa para o bloco de tratamento de entrada dos dados foi “WORD-AS-BIT”, definido na Fig. 14 e para saída dos dados foi “BIT-AS-WORD”, definido na Fig. 15.

Figura 14: Representação escrita e gráfica do bloco “WORD-AS-BIT”.

```

1  FUNCTION_BLOCK POU
2  VAR_INPUT
3      wIn:WORD;
4
5  END_VAR
6  VAR_OUTPUT
7      x1:BOOL;
8      x2:BOOL;
9      x3:BOOL;
10     x4:BOOL;
11     x5:BOOL;
12     x6:BOOL;
13     x7:BOOL;
14     x8:BOOL;
15     x9:BOOL;
16     x10:BOOL;
17     x11:BOOL;
18     x12:BOOL;
19     x13:BOOL;
20     x14:BOOL;
21     x15:BOOL;
22     x16:BOOL;
23 END_VAR
24 VAR
25 END_VAR
26

```

Fonte: Autor (2021).

Figura 15: Representação escrita e gráfica do bloco “BIT-AS-WORD”.

```

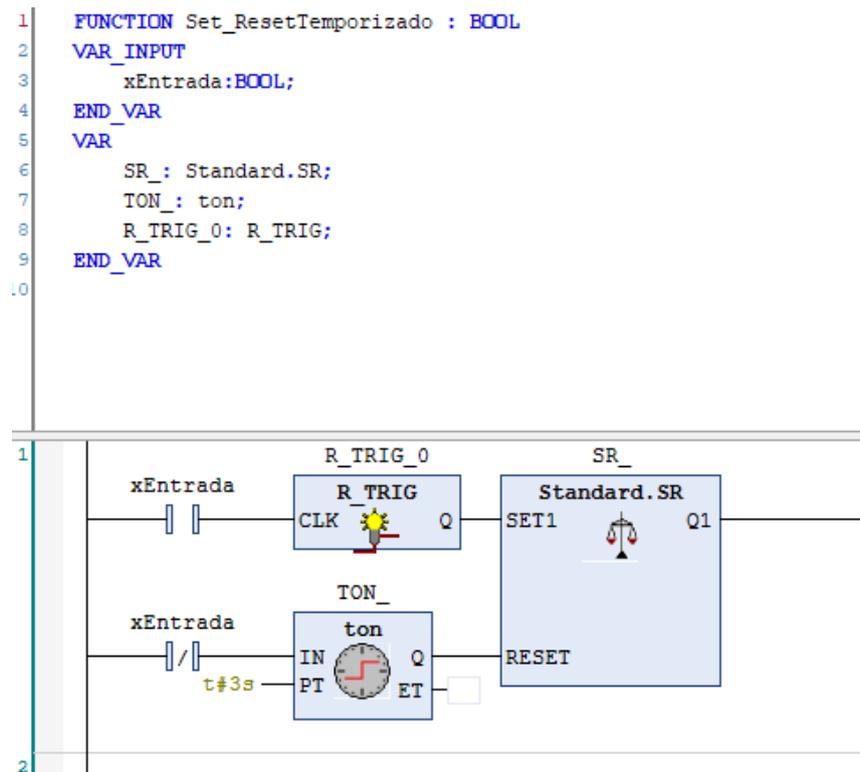
1  FUNCTION_BLOCK POU
2  VAR_INPUT
3      x1:BOOL;
4      xq2:BOOL;
5      xq3:BOOL;
6      xq4:BOOL;
7      xq5:BOOL;
8      xq6:BOOL;
9      xq7:BOOL;
10     xq8:BOOL;
11     xq9:BOOL;
12     xq10:BOOL;
13     xq11:BOOL;
14     xq12:BOOL;
15     xq13:BOOL;
16     xq14:BOOL;
17     xq15:BOOL;
18     xq16:BOOL;
19
20
21 END_VAR
22 VAR_OUTPUT
23     wOut:WORD;
24 END_VAR
25 VAR
26 END_VAR

```

Fonte: Autor (2022).

O terceiro bloco de função é um set e reset temporizado demonstrado na Fig. 16, criado com intuito de estipular um tempo de programação para executar a função *set* e *reset*. Que tem por objetivo acionar uma saída, e que ela fique acionada até o momento em que seja resetada (desligada), por um tempo estipulado pelo programador.

Figura 16: Representação escrita e gráfica do bloco “SET\_RESETTEMP”.



Fonte: Autor (2021).

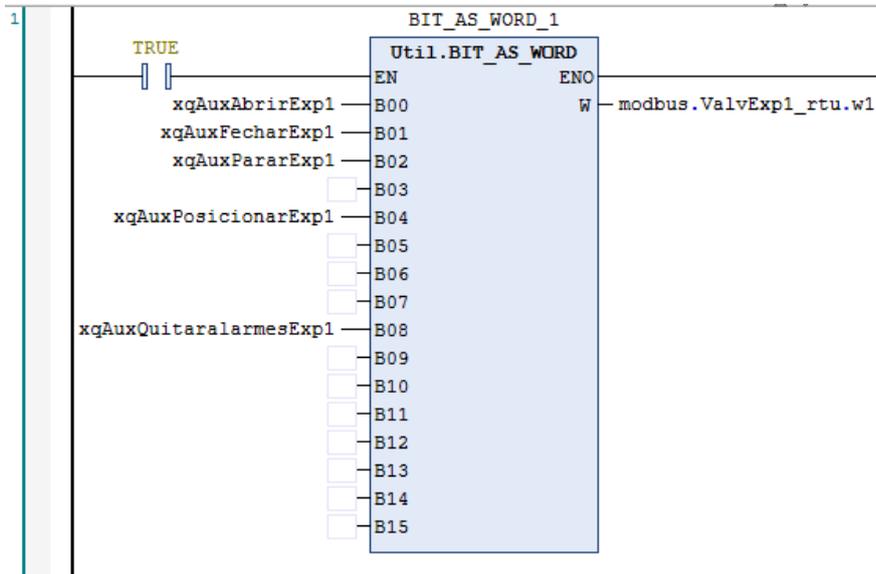
#### 4.3 PROGRAMAÇÃO PARA CONTROLE DAS VÁLVULA ATUADORAS.

Feitos os respectivos complementos para efetuar o controle completo da válvula, foram utilizados dois métodos de linguagem de programação. Ladder que tem a função de imitar os antigos diagramas elétricos, facilitando assim a interação e lógica para o programador, e a Linguagem Textual (texto estruturado), que contém essenciais elementos de uma linguagem de programação moderna, incluindo as instruções condicionais permitindo ao programador aplicar um processo complicado em alto nível de lógica, facilitando o controle por meio da escrita.

Nas Fig. 17, Fig. 18 e Fig. 19, temos o controle completo da válvula de expurgo 1, sendo que na Fig. 17 mostra-se o uso do bloco de função “BIT-AS-WORD”,

utilizado para o comando de saída de informações do CLP para controle das válvulas atuadoras.

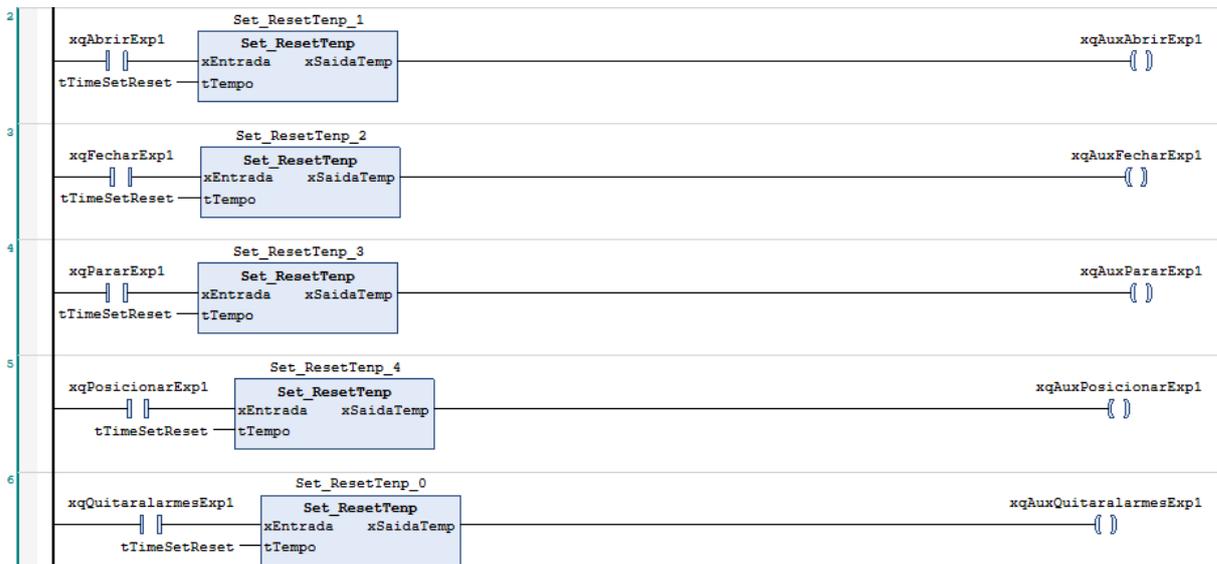
Figura 17: Programação de controle válvula com “BIT-AS-WORD”.



Fonte: Autor (2022).

Na Fig. 18 temos o controle das saídas booleanas que utilizamos o “SET\_RESETTEMP”, com o objetivo de efetuar um controle temporizado de abertura, parada e fechamento da válvula, pela possibilidade de uma taxa de erro da comunicação entre a válvula e CLP.

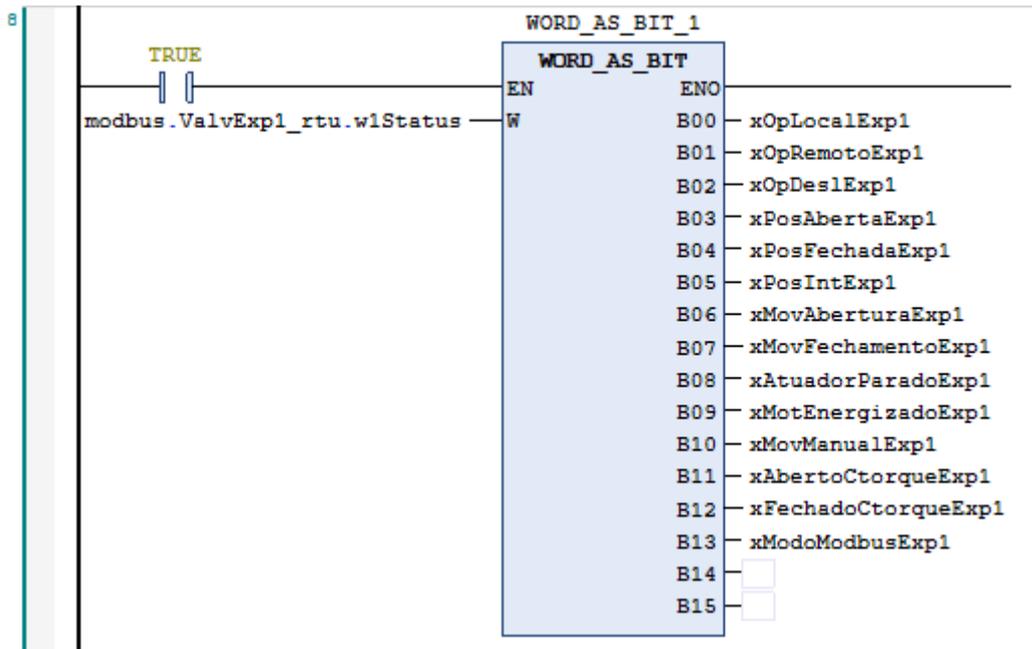
Figura 18: Programação de controle válvula com “SET\_RESETTEMP”.



Fonte: Autor (2022).

A Fig. 19 apresenta as entradas de informações, status, valores e alarmes da válvula que seria a parte de leitura de dados, entre o atuador e o CLP, utilizando o bloco de função “WORD-AS-BIT”.

Figura 19: Programação de controle válvula com “WORD-AS-BIT”.



Fonte: Autor (2022).

Visto acima nas imagens observou-se a mescla de linguagens de programações e interações para efetuar o controle das válvulas de uma maneira mais simples e lógica.

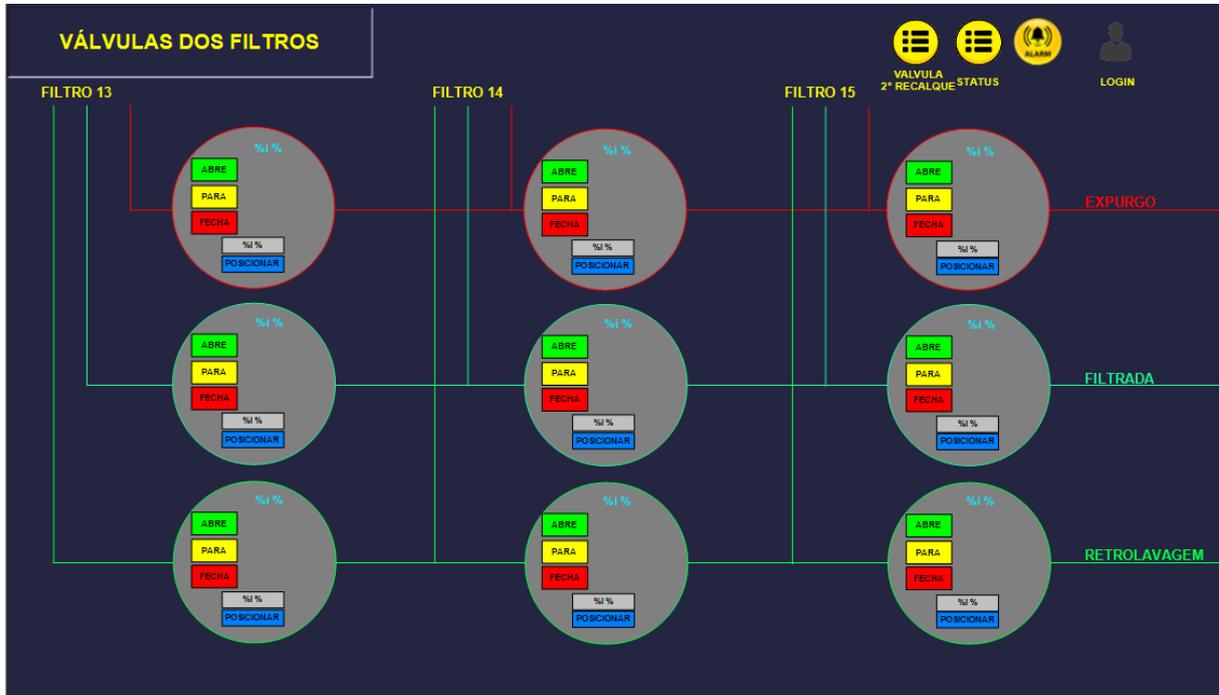
#### 4.4 SISTEMA DE SUPERVISÃO DO PROGRAMA DO CLP

Sistemas de supervisão servem como interface entre o operador e toda a lógica de programação e controle do processo. O principal desafio é a absorção de todo o programa desenvolvido e retransmissão através de uma forma amigável e intuitiva para o operador da válvula. Onde ele consiga exercer todos os controles e leituras das funções em que o CLP irá atribuir na válvula atuadora.

Pensando numa tela intuitiva para o operador foi recriado uma ilusão de semelhança a frente da válvula para se ter uma melhor interação entre e comando de fechamento, abertura, parada e posicionamento da válvula. Para que não houvesse

tanta troca de tela em relação ao comando de válvulas, foi exposto também a informação de posição nessa tela, como mostrado na Fig. 20.

Figura 20: Tela de controle e leitura para comando das válvulas.



Fonte: Autor (2022).

Em virtude da relação de status ter uma quantidade elevada de informações, foi feita uma tela que recria a lista de status em que ao lado de cada definição de informação tem uma luz indicando a situação de: ligada ou desligada para cada função. Dessa forma o operador tem uma sequência de leitura para acompanhar as situações em que a válvula opera, como mostra a Fig. 21.

Figura 21: Tela de status de das válvulas.



Fonte: Autor (2022).

## 5 CONCLUSÃO

Com o desenvolvimento do programa de CLP para o controle e supervisão das válvulas atuadora, foi demonstrada a possibilidade do operador poder controlar e supervisionar todo o sistema dele por uma única interface, acarretando a diminuição de erros na troca de informações e um melhor detalhamento de posições das válvulas no seu processo.

Em relação ao desenvolvimento do programa foi analisado que por meio de uma mescla de linguagens de programações pode se criar um controle de automação com uma menor quantidade de parâmetros e uma forma de lógica mais simples de entendimento para o programador. Onde a cada ponto do programa utiliza-se a melhor linguagem determinada à operação.

A supervisão do programa envolve um parecer criativo, que pretende alcançar uma melhor interface entre o operador e atuador. As telas do supervisório são intuitivas e ao mesmo tempo funcionais ao operador que a comanda.



Já a comunicação de rede utilizada, demonstra a inviabilidade do uso de fios de comando para o controle das válvulas atuadoras e o custo-benefício de apenas utilizar o CLP e não necessitar de cartões de entradas e saídas, devido todas as entradas e saídas serem utilizadas através da rede Modbus-RTU e o processo dela ganhar segurança através da troca de informações por meio da comunicação Modbus-RTU.

O trabalho apresentado abre possibilidades de estudos mais aprofundados na utilização da mescla de linguagens de programação para o desenvolvimento de programas de CLP para controle e supervisão. Por meio do uso de redes de comunicações industriais possibilita a obtenção de informações cada vez mais rápidas e seguras, levando novas possibilidades para profissionais que programam CLP para processos industriais de pequeno, médio e grande porte.

### REFERÊNCIAS

ASSUNÇÃO, F. N. O. **Controlador Fuzzy implementado em ScadaBR para o controle de pressão numa adutora de água em Florianópolis**. 2016.

BAHRIN, M.; OTHMAN, F.; AZLI, N.; TALIB, M. Industry 4.0: A review on industrial automation and robotic. **Journal Teknologi**, [s.l.], v. 78, n.6-13, p.137–143, 2016.

BITKOM; VDMA; ZVI. **Implementation strategy industrie 4.0**: report on the results of the industrie 4.0 platform. Frankfurt, Alemanha, 2016.

Controlador PFC100 (750-8102) | WAGO BR. Disponível em :<<https://www.wago.com/br/controlador/controlador-pfc100/p/750-8102>>. Acesso em: 3 nov. 2022.

DA SILVA, Edilson Alfredo. **Introdução às linguagens de programação para CLP**. Editora Blucher, 2016.

DA SILVA, Marcelo Eurípedes. **Controladores Lógico Programáveis-Ladder**. 2007.

DHINAKARAN, Vasanth; GUGAPRIYA, G. MODBUS TO M-BUS PROTOCOL CONVERSION GATEWAY WITH LOW CONVERSION LATENCY FOR BUILDING



MANAGEMENT SYSTEMS. **Pakistan Journal of Biotechnology**, v. 13, n. special issue 1, p. 199-202, 2016.

FRANCHI, Claiton Moro; CAMARGO, Valter Luís Arlindo de. **Controladores lógicos programáveis: sistemas discretos**. Saraiva Educação SA, 2008.

GONÇALVES DA SILVA, A. P.; SALVADOR, M. **O que são sistemas supervisórios?** Disponível em: <<http://www.drbrm.org/automacao/6SistemaSupervisorios.pdf>>. Acesso em: 28 maio 2022.

ISA. 2016. **Introdução à Norma ISA-101: Interfaces Homem-Máquina.** ” 2016. Disponível em: <<http://isasp.org.br/wp-content/uploads/2020/01/ISA-101-III-Simp%C3%B3sio-ISA-S%C3%A3o-Paulo-Sabesp-Nov2016.pdf>>. Acesso em: 22 maio de 2022.

MODBUS-IDA. **Modbus over serial line specification and implementation guide V1.02.** 2006. Disponível em: <[https://modbus.org/docs/Modbus\\_over\\_serial\\_line\\_V1\\_02.pdf](https://modbus.org/docs/Modbus_over_serial_line_V1_02.pdf)>. Acesso em 14 de maio de 2022.

PETRUZELLA, Frank D. **Controladores Lógicos Programáveis**. 4. ed. São Paulo: Bookman, 2014.

SANTOS, Stephan August Roehrig Domingues dos et al. **Programação e simulação de um sistema de automação do setor operacional de uma indústria de grãos**. 2019.

SOUSA, Rédytton Brenner. **Implementação de Controle de Riser, Validação Experimental e Análise Através de Processamento de Imagens**. 2015. 49 f. TCC (Graduação) - Curso de Engenharia de Controle e Automação, Faculdade de Tecnologia, Universidade de Brasília Faculdade de Tecnologia, Brasília, 2015.

STEMMER, Marcelo Ricardo. **Redes locais industriais: a integração da produção através das redes de comunicação**. Ed. da UFSC, Florianópolis, 2010.



REYNDERS, Deon, et al. **Practical Industrial Data Communications: Best Practice Techniques**. Elsevier Science, Amsterdã, 2004.

THE BOSTON CONSULTING GROUP (BCG). **Industry 4.0: the future of productivity and growth in manufacturing industries**. Alemanha, 2015.

ZANNI, Alessandro. 2015. **Sistemas Cyber-Físicos e Cidades Inteligentes**. 2015. Disponível em: <https://www.ibm.com/developerworks/br/library/ba-cyber-physical-systems-and-smart-cities-iot/index.html>. Acesso em: 30 maio de 2022.

