

DESENVOLVIMENTO DE UM SISTEMA DE MONITORAMENTO PARA MÁQUINA EMPACOTADORA UTILIZANDO SUPERVISÓRIO EM PLATAFORMA ABERTA

Luan Hübner, Luiz Fernando Henning,
Instituto Federal de Santa Catarina
Câmpus Jaraguá do Sul – Rau – Curso de Bacharelado em Engenharia Elétrica
e-mail: luan.h26@aluno.ifsc.edu.br, luizh@ifsc.edu.br
Trabalho de Conclusão de Curso – TCC – 14/12/2022.

Resumo – O objetivo deste documento é apresentar a metodologia de desenvolvimento de um sistema de monitoramento em plataforma aberta para máquina empacotadora. Desenvolvido sobre um estudo de caso de uma máquina real em operação, onde será criada uma interface gráfica para acompanhamento e registro histórico. Ao término será proposto um software desenvolvido utilizando a plataforma livre ScadaBR integrado a ferramenta de *dashboard* Grafana, e instalado em um minicomputador do tipo *raspberry pi*. Possibilitando assim acompanhar o desempenho da máquina a qualquer hora de qualquer lugar de forma segura.

Palavras-Chave – Dashboard, Empacotadora, Histórico, Monitoramento, ScadaBR, Desempenho.

DEVELOPMENT OF A MONITORING SYSTEM FOR PACKAGING MACHINE USING SUPERVISORY IN OPEN PLATFORM

Abstract – The objective of this document is present the methodology for developing an open platform monitoring system for a packing machine. Developed on a case study of a real machine in operation, where a graphical interface will be created for monitoring and historical record. At the end will be proposed a software developed using the free platform ScadaBR integrated the Grafana dashboard tool, and installed on a minicomputer of the type *raspberry pi*. Thus enabling you to monitor machine performance anytime from anywhere safely.

Keywords – Dashboard, History, Monitoring, Packer, Performance, ScadaBR.

I. INTRODUÇÃO

Com a era da indústria 4.0 e os avanços tecnológicos exponenciais, cada vez mais vê-se a necessidade do monitoramento dos equipamentos seja para tomada de decisão, fechamento de um negócio ou até mesmo para um diagnóstico geral da empresa. A Indústria 4.0 é uma tendência que representa a quarta revolução industrial [1]. Essa revolução visa relacionar a tecnologia da informação e comunicação com tecnologias disruptivas contemporâneas,

tornando esses ambientes mais autônomos, competitivos e eficientes [2].

Este trabalho tem a finalidade de desenvolver um Sistema de Supervisão e Aquisição de Dados (SCADA) para monitorar e supervisionar as variáveis e os dispositivos de um processo de empacotamento em uma linha de produção. O trabalho também propõe coleta de dados de produção para visualização e gravação em banco de dados. O desenvolvimento de uma interface para monitoramento dos dados de produção. Um banco de dados com histórico de produção e o acesso remoto as informações de produção em tempo real de qualquer lugar.

O trabalho divide-se em seis seções. Na Seção II, é apresentada a definição e o funcionamento de uma empacotadora. Na Seção III, é apresentado e explanado sobre o que é um sistema de supervisão. Na Seção IV, é apresentado um estudo de caso. Na Seção V, são apresentados os métodos utilizados no desenvolvimento do sistema de supervisão e na Seção VI são apresentadas as conclusões.

II. DEFINIÇÃO DE EMPACOTADORA

Em uma empacotadora, a embalagem flexível (ou “filme”) refere-se ao material utilizado no pacote que envolverá o material, seja ele de plástico, papel, cartão ou laminados combinando vários materiais. O filme pode ser alimentado continuamente a partir de um rolo ou fornecido como folhas pré-cortadas, porém o mais comum no Brasil e utilizado na máquina estudada é o tipo rolo. Este por sua vez pode ainda ser impresso ou liso, dependerá da necessidade ou tipo de aplicação onde utilizam-se marcações para indicar início e fim de cada pacote ou até mesmo com a finalidade de contagem.

As máquinas empacotadoras verticais do termo em inglês “*Vertical form fill sealing machine*” (VFFS) são máquinas que formam a embalagem, enchem-na com um produto úmido ou seco e fecham-na. As máquinas VFFS são altamente versáteis e podem ser usadas para praticamente qualquer produto líquido, cremoso ou sólido. Em suma, quase qualquer produto que possa ser descartado ou despejado em um saco pode ser embalado em uma máquina empacotadora [3]. O pacote deste tipo de máquina normalmente caracteriza-se por vedações horizontais na parte superior e inferior e uma única costura longitudinal, geralmente no centro da parte de trás do saco.

A operação da empacotadora começa com o filme sendo retirado do rolo de suprimento e posto sobre uma série de rolos tracionadores que controlam a tensão do filme. A tensão do filme é fundamental para o bom funcionamento da máquina e

se não for controlado adequadamente, pode resultar em filme enrugado e comprimento inconsistente do saco.

Depois que o rolo tracionador puxa o filme, este passa por um funil de formação, que o molda em forma tubular. Geralmente é redondo, embora outras formas possam ser usadas dependendo da forma desejada do saco acabado [3]. À medida que o tubo é formado, as bordas longitudinais (verticais) do filme são unidas e seladas a quente por meio de uma solda.

Um único par de mandíbulas de vedação horizontal é usado para formar simultaneamente ambas as vedações de extremidade. Cada uma dessas mandíbulas tem duas vedações, uma das quais veda a parte superior do saco cheio e a outra que veda o fundo do próximo saco. Essas mandíbulas geralmente incorporam uma faca para cortar os sacos durante a vedação.

Depois que o fundo do saco é selado, o produto cai pelo centro do funil de formação e cai no fundo do saco. O produto sólido ou líquido pode ser introduzido usando qualquer tecnologia de enchimento apropriada. Mais à frente serão tratados os dosadores mais utilizados.

Uma vez que o saco esteja cheio, ele deve ser puxado para baixo, formando o próximo. As máquinas, empacotadoras do tipo mesa móvel, utilizam as próprias mordanças de vedação para fazer isso. Depois que as mordanças de vedação se fecham horizontalmente, movem-se para baixo puxando o filme para a posição para o próximo saco. Na parte inferior do curso, eles se abrem e se movem para cima para formar o próximo selo. Este tipo de máquina desempenha um papel brilhante no empacotamento de materiais que produzem muito pó, já que a solda do pacote é feita logo abaixo do dosador.

Em empacotadoras do tipo mesa fixa um par de correias, acionadas por servomotores, prende o filme entre a correia e o tubo de formação. Desta forma a mordança de vedação não precisa mover-se e permite uma velocidade muito maior da máquina. Este tipo de máquina é amplamente aplicada a produtos da família dos grãos, onde o produto permite uma queda maior dentro do pacote sem causar uma nuvem de poeira.

Ambos os tipos de empacotadora operam em modo descontínuo e possuem suas vantagens e desvantagens dependendo do material a ser embalado. O supervisor desenvolvido servirá para ambos os tipos independente do material a ser embalado. Quanto aos dosadores que alimentam a empacotadora, existem muitos, abaixo será citado os mais comuns utilizados no Brasil.

1) *Dosador de balança*: uma calha vibratória vibra com diferentes intensidades e despeja o material em um recipiente conectado a uma célula de carga até atingir o peso desejado.

2) *Dosador de rosca*: uma rosca sem fim empurra o material até o cone de alimentação do pacote, este sistema é orientado por volume, ou seja, a quantidade do material é medida pela quantidade de voltas da rosca.

3) *Dosador de caneco*: o material cai por um funil central dentro de um caneco com tampa, este caneco é acoplado a uma base circular giratória com vários canecos, quando o caneco está cheio a base gira e a tampa do caneco fecha dosando a quantidade certa.

4) *Dosador de multi-cabeçotes*: sistema rotativo (disco espalhador) que direciona o material para a çaçamba de

pesagem (cesto) geralmente á um número de 16 à 24 cestos configurados com pesos diferentes, onde o software define qual a melhor combinação de cestos liberar para formar o melhor peso.

5) *Dosador de gaveta*: similar ao dosador de caneco porém possui uma única gaveta com dois pistões superiores e inferiores. O superior atua para fechar a gaveta quando cheia e o inferior libera o material depois que a tampa superior fecha.

III. SISTEMA DE SUPERVISÃO

Um sistema de supervisão consiste em um software capaz de operar sob uma plataforma computacional na forma de sistema operacional, com a capacidade de gerar telas com um desenho esquemático que represente o processo físico a gerenciar e que está sendo controlado por um CLP ou outro sistema de controle [4].

Os sistemas SCADA (do inglês *Supervisory Control and Data Acquisition*) para processos industriais possibilitam a leitura e escrita de dados referente às grandezas físicas do processo a controlar por meio de unidades remotas industriais tais como UTR e/ou Controladores Logico Programáveis, que a partir de uma unidade de processamento são capazes de formatar esses dados e disponibilizá-los de forma visível ao usuário ou operador em uma multiplicidade de formas por meio de sistemas de supervisão [5]. O sistema de supervisão é assim uma representação virtual do processo ou planta física a controlar, que possui uma representação em tempo real das principais variáveis monitoradas ou controladas.

O objetivo principal dos sistemas SCADA é propiciar uma interface de alto nível do operador com o processo, informando-o em tempo real de todos os eventos de importância da planta e também possibilitando a ele uma intervenção direta no processo [5]. Possibilita assim ao usuário ou operador verificar, de forma gráfica, os valores das variáveis do processo que representam as grandezas físicas, observar tendências de variação, verificar os estados de equipamentos, entre outras características e propriedades, possibilitando também o envio de comandos e parâmetros para o processo inerente à sua operação.

Para permitir isso, os sistemas SCADA identificam os *tagnames*, que são assim todas as variáveis numéricas ou alfanuméricas envolvidas na aplicação, podendo executar funções computacionais (operações matemáticas, lógicas, com vetores ou *strings*, etc.) ou representar pontos de entrada/saída de dados do processo que está sendo controlado [4]. Neste caso, correspondem às variáveis do processo real (ex.: velocidade, peso, quantidade etc.), comportando-se como a ligação entre o controlador e o sistema.

As janelas ou telas construídas em um sistema de supervisão são comumente chamadas de sinóticos, um nome herdado dos tradicionais sinóticos elétricos utilizados durante muitos anos na indústria de sistema elétricos.

Um SCADA típico deve oferecer *drivers* de comunicação com equipamentos, um sistema para registro contínuo de dados ("*datalogger*") e uma interface gráfica para usuário, conhecida como "IHM" ou Interface Homem-Máquina. Na IHM são disponibilizados elementos gráficos como botões,

ícones e *displays*, representando o processo real que está sendo monitorado ou controlado [5].

Dentre as funções mais utilizadas em sistemas SCADA estão:

- Geração de gráficos e relatórios com o histórico do processo;
- Detecção de alarmes e registro de eventos em sistemas automatizados;
- Controle de processos incluindo envio remoto de parâmetros e *set-points*, acionamento e comando de equipamentos;
- Uso de linguagens de script para desenvolvimento de lógicas de automação ("algoritmos").

O software ScadaBR é desenvolvido sob licença do tipo *Open Source* (código aberto), onde todo o código é disponibilizado de forma gratuita tanto para utilização como para modificações [6].

O ScadaBR é uma aplicação multiplataforma baseada em Java, isso permite que o software seja executado em diferentes plataformas, necessitando apenas de um servidor de aplicação (sendo o Apache Tomcat a escolha padrão) [6].

Ao executar o aplicativo, ele pode ser acessado a partir de um navegador de Internet, preferencialmente o Firefox ou o Chrome. A interface principal do ScadaBR é de fácil utilização e já oferece visualização das variáveis, gráficos, estatísticas, configuração dos protocolos, alarmes, construção de telas tipo IHM e uma série de opções configuráveis.

Após configurar os protocolos de comunicação com os equipamentos e definir as variáveis (entradas e saídas, ou "*tags*") de uma aplicação automatizada, é possível montar interfaces de operador Web utilizando o próprio navegador. Também é possível criar aplicativos personalizados, em qualquer linguagem de programação moderna, a partir do código-fonte disponibilizado ou de sua API "*web-services*".

IV. ESTUDO DE CASO

A empacotadora do estudo em questão opera conforme descrito anteriormente. É uma máquina que tem por princípio empacotar determinado produto, através do filme plástico que passa pela máquina onde ele é moldado pelo colarinho que forma o pacote e é feito a solda inferior. Solda feita desce o produto dentro do pacote, ao mesmo tempo faz a solda vertical e a solda superior, fecha o pacote e já corta o mesmo.

A empacotadora deste trabalho em questão é do tipo mesa fixa com dosador de rosca e pode ser observada na Figura 1. Tem como característica comando pneumático a seco (sem lubrificação), dosador volumétrico tipo rosca e conta também com um datador, para datar as embalagens (datas fabricação, validade e lote).



Figura 1. Empacotadora mesa móvel dosador de rosca

Para realizar a operação desta máquina a mesma conta com um painel com componentes de potência e acionamento como: fonte, relés, disjuntores, disjuntores-motor, contadores, relés de estado sólido e inversores. O controle de todos os componentes é feito por um rack de CLP em conjunto com uma IHM, onde esta serve como interface para o operador ajustar parâmetros e observar a máquina trabalhar.

Um detalhe importante a ser frisado é a marca do CLP do fabricante Delta, esta informação é importante para consultar manuais, entender o método de programação e também verificar o mapa de endereços *modbus*. Na Figura 2 é apresentado a placa de montagem deste painel.

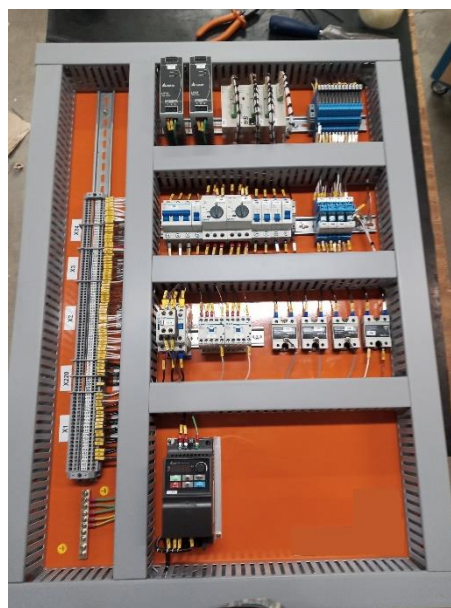


Figura 2. Placa de montagem do painel de controle da empacotadora

Máquinas deste tipo já operam desta forma há anos e possuem um bom funcionamento, todavia não existe nenhum tipo de métrica. É possível visualizar apenas informações tais como: se máquina está funcionando ou não, um contador de pacotes e uma célula de carga para pesar.

Não possui também nenhum tipo de dado histórico para verificar a efetividade. São feitas apenas anotações manuais, onde tem-se uma noção de quantos pacotes por dia ou peso é necessário empacotar. Não conta com a possibilidade de comparação de operadores ou até mesmo uma mensuração do período de trabalho.

Com estas imperfeições em mente e com a proposta de melhoria, propôs-se a criação de um sistema de monitoramento e supervisão de baixo custo e com métricas de produção.

V. METODOLOGIA DE DESENVOLVIMENTO DO SISTEMA DE SUPERVISÃO

Em função do que foi apresentado, propõe-se a criação de um sistema de supervisão, seguindo-se oito passos conforme ilustrado na Figura 3 e explanado nas subseções A-H, da presente Seção IV.

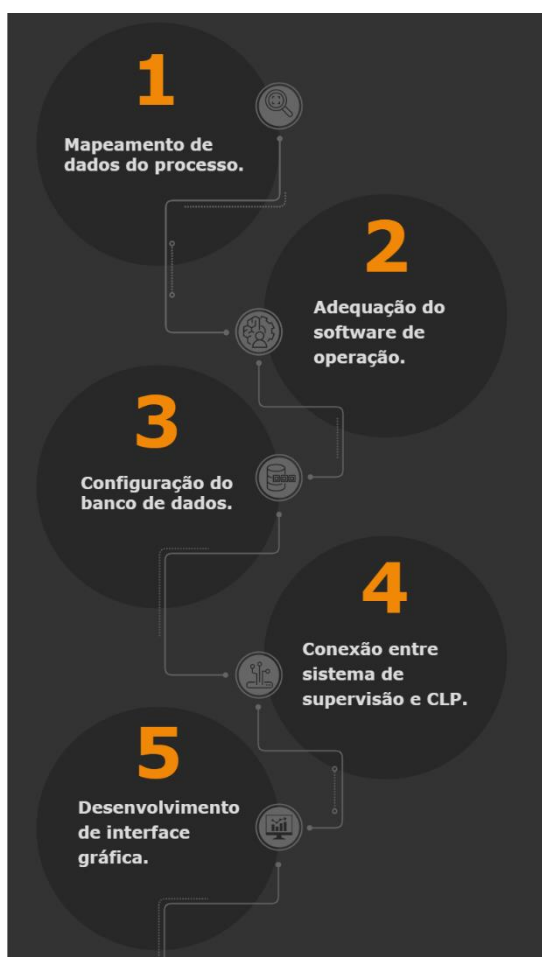


Figura 3. Fluxograma das etapas para desenvolvimento do sistema de monitoramento

A. Mapeamento de dados do processo

Esta etapa requer um levantamento das informações do processo que seriam interessantes serem observados neste tipo de máquina. Para definir quais dados seriam relevantes, foi realizada uma pesquisa com fabricantes deste tipo de máquina e a partir disto visou-se dados pertinentes ao desempenho da máquina sendo operada por um trabalhador em seu turno.

Estas informações foram ainda subdivididas em três categorias: produção atual, índices de indicadores e histórico de produção. Por fim, chegou-se na relação abaixo:

1) Produção atual:

- Peso do último pacote (em gramas);
- Velocidade da máquina (em pacotes por minuto);
- Operador (pessoa responsável pelo turno da máquina);
- Receita (produto que está sendo empacotado);
- Estado atual da máquina (Falha, Manual, Ciclo e Automático);
- Número de ciclos (cada pacote equivale a um ciclo);
- Performance instantânea (relação entre total já produzido e total programado para produzir);
- Índice de qualidade (relação entre total de pacotes empacotados e aqueles com peso correto);
- Número de pacotes com peso bom (permitido pelo INMETRO);
- Número de pacotes com peso ruim (fora da faixa permitida pelo INMETRO);
- Quantidade processada de produto (peso em kg do produto que está sendo embalado);
- Quantidade estimada de Embalagem (em kg);
- Evolução da produção (quantidade de pacotes produzidos no decorrer do dia);
- Tempo em falha (em minutos);
- Tempo em ciclo (em minutos);

- Tempo em manual (em minutos);
- Tempo em setup (em minutos);
- Pacotes com peso bom, peso ruim e total;
- Produto com peso bom, peso ruim e total;
- Embalagem com peso bom, peso ruim e total;

2) Indicadores:

- Disponibilidade (relação entre tempo produzindo e tempo programado para produzir);
- Qualidade (relação entre quantidade de pacotes com peso bom e quantidade total produzida de pacotes);
- Performance (relação entre velocidade média real de empacotamento e velocidade desejada para a tarefa);
- OEE Instantâneo (multiplicação dos 3 fatores anteriores);

3) Histórico de produção:

- Tabela de registro de operação onde é gravado o histórico de operação;
- Tabela de registro de peso onde é gravado o histórico de pesagem;
- Tabela de registro OEE onde é gravado o histórico de rendimento diário;

B. Adequação do software de operação

Esta etapa consistiu em adaptar o programa já existente da máquina de empacotar para coletar os dados pretendidos. Com este propósito foi utilizada uma área separada da área de processo interno do CLP, e com lógica de programação *ladder* foram realocando os dados para determinadas memórias.

Os CLP's da marca Delta utilizam o software chamado ISPSOFT para desenvolvimento e alteração de programas em *ladder* [7]. Nesta interface criou-se uma rotina de verificação diária e separação de dados, sendo utilizado blocos *move* e *buffer*.

Para medir os tempos dos estados de operação (Falha, Manual, Ciclo e Automático) implementou-se alguns temporizadores. Criou-se também um contador de ciclos (variável numérica crescente que não repete-se e nunca zero), podendo ser utilizada para medir vida útil da máquina e também servir como chave primária do banco de dados.

Como métrica de produção foi necessário medir tempo, velocidade e qualidade. Posteriormente com a multiplicação destas taxas de desempenho é calculado o OEE (*Overall Equipment Effectiveness*, ou Índice de Eficiência Global do equipamento). Trata-se de um método popular de avaliação da eficácia da capacidade utilizado no mundo todo [1].

Na Figura 4 é apresentado um exemplo da lógica utilizada para o cálculo da disponibilidade utilizando blocos matemáticos.

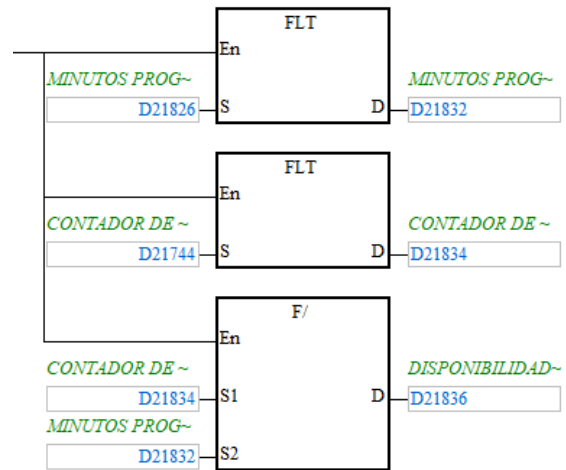


Figura 4. Exemplo de programação em linguagem *ladder* no CLP Delta

Com a separação interna das informações também houve necessidade de configurar gatilhos de envio de informações e salvamento de dados quando determinadas situações fossem cumpridas, como a pesagem de um novo pacote, por exemplo.

C. Configuração do banco de dados

Nesta etapa foi realizada a configuração do banco de dados para a aplicação ScadaBR, migrando do sistema nativo Derby para o MySQL. Os motivos para troca foram:

- O banco de dados MySQL cresce em um ritmo menor, ou seja é mais compacto, o armazenamento é mais otimizado [8];
- MySQL é mais rápido que o Derby na consulta, cerca de 10 vezes;
- Consolidação no mercado, o MySQL é utilizado mundialmente há muito tempo e possui bastante material de auxílio facilmente encontrado na internet;

Para vincular o ScadaBR ao MySQL primeiro é necessário criar um banco de dados com o nome desejado. Depois cria-se um usuário e senha que será utilizado pelo ScadaBR para acessar o banco. Depois de criado o usuário é necessário gerar as permissões de acesso.

Após a criação do banco de dados MYSQL é necessário efetuar a configuração do ScadaBR através do arquivos *env.properties*, onde são alteradas as informações referentes ao tipo de banco de dados e também endereçamento, usuário e senha.

Feito isso é só reiniciar o servidor do ScadaBR e este vincula-se automaticamente ao banco indicado. Também não é necessário criar tabelas no banco, o próprio ScadaBR faz isso quando vincula ao MySQL.

Após realizar as configurações relatadas anteriormente, observou-se que o formato de organização das tabelas no sistema ScadaBR apresentam limitações. As tabelas são criadas agrupando fatores numéricos separadamente de fatores alfanuméricos. O que acaba impossibilitando orientar de forma temporal em uma mesma tabela valores numéricos e caracteres do tipo texto.

No processo da máquina duas informações bastante importantes são o nome do operador e a receita que está sendo utilizada. Informações que com toda certeza também precisam ser armazenadas na mesma tabela que informações como número de pacotes produzidos, peso do último pacote, etc. Como solução para esta limitação decidiu-se implementar a utilização de outro software para gravação das informações no banco de dados.

O software escolhido foi o Node-Red, este possui uma fácil interface de iteração e possibilita uma versatilidade maior na gravação das informações. Depois de um estudo e posterior análise, desenvolveram-se 3 tabelas com registro histórico:

Tabela 1 - Registro de operação: Data e hora, Ciclo, Contador de pacotes, Velocidade de Empacotamento, Operador e Receita;

Tabela 2 - Registro de peso: Data e hora, Ciclo, Peso da Embalagem, Peso Inmetro, Peso medido e Receita;

Tabela 3 - Registro OEE: Data e hora, Tempo Produzindo, Tempo programado, Disponibilidade, Velocidade média, Velocidade programada, Performance, Pacotes com peso bom, Total de pacotes, Qualidade e OEE. Sendo que esta tabela registra sempre os dados do dia anterior;

Após a criação das tabelas no banco de dados MySQL, foi realizada a configuração do Node-Red. Para isso, foi necessário baixar algumas bibliotecas (*Modbus*, por exemplo) e a criação do fluxo dos blocos lógicos. O fluxo desenvolvido para esta aplicação pode ser observado no Anexo I.

Para facilitar a gravação de informações e otimizar o tamanho do banco de dados, criou-se gatilhos de gravação em memórias específicas do CLP. Para as tabelas 1 e 2 estas memórias foram configuradas de forma a serem ativadas apenas quando algum dos valores sofrer alteração, já para a tabela 3 a memória é ativada com a troca da data (passagem de um dia para outro).

Utilizando blocos MOV na programação LADDER criou-se também um conjunto de memórias sequenciais com as informações de cada tabela, facilitando assim a aquisição do protocolo *Modbus* via Node-Red, já que este recebe os dados em um único pacote.

O Node-Red consulta as memórias gatilho e verifica seu estado, se verdadeiro (*true*) coleta as informações do conjunto de memórias e salva no banco de dados, após salvar desabilita o gatilho. Se falso (*false*), não executa nenhuma função evitando informações repetidas no banco e deixando o canal *Modbus* livre.

D. Conexão entre sistema de supervisão e Controlador Lógico Programável

Consistiu em estabelecer a conexão do ScadaBR com o CLP, isto foi feito fisicamente conectando ao computador via porta ethernet e virtualmente via um componente chamado “*Data Source*” traduzindo seria uma fonte de dados. Neste componente escolhe-se o protocolo de comunicação desejado, para este trabalho *Modbus IP*.

Após esta etapa, foi efetuada a configuração do protocolo de comunicação de acordo com as informações necessárias, tais como:

- Período de atualização: Velocidade com que o supervisor fará a consulta dos dados. Como existe a

possibilidade da máquina trabalhar com 60 pacotes por minuto, ou seja, 1 pacote por segundo, optou-se por um período de atualização de 1 segundo. É preciso tomar certo cuidado com esta velocidade, pois quanto mais rápido for mais processamento exigirá do hardware.

- Host: Este campo deve ser preenchido com o endereço de IP do CLP, neste caso 10.100.4.10

- Os demais campos não necessitam ser alterados e podem ser mantidos conforme padrão.

Com os campos devidamente preenchidos basta salvar a configuração, neste ponto se tudo foi corretamente configurado, o ScadaBR faz o teste de comunicação e exibe mensagens de erro caso tenha alguma inconsistência.

Após efetuar a conexão, são incluídos os pontos de dados (data points), que são as informações de endereçamento de memória no protocolo *modbus*. Os parâmetros destes pontos devem ser preenchidos da seguinte forma:

- Nome: Nome do ponto de dados. Para facilitar a visualização e futuras alterações, neste trabalho a sintaxe usada é “*ENDEREÇO_DA_MEMÓRIA_DO_CLP*”- “*DESCRIÇÃO_ABREVIADA*”. Por exemplo: Memória D20062 que contém o valor do peso da embalagem preenchido pelo operador na IHM, sua nomeação no ScadaBR D20062-PES.EMB.

- *Export ID* (XID): Este ID é utilizado caso o usuário deseje realizar alguma lógica de programação interna em JavaScript. Cada data point possui um, seria equivalente a um CPF e o sistema gera números aleatórios automaticamente, não sendo necessário colocar qualquer valor.

- ID do escravo: endereço de escravo do CLP na rede *modbus*, essa é uma característica do protocolo de comunicação. Na rede *modbus* existe um mestre, que geralmente é a IHM, e os demais equipamentos conectados a rede são escravos e possuem cada um número de identificação configurável. Neste caso o endereço de escravo do CLP é igual a 1.

- Faixa de registro: Esta configuração também é uma característica do protocolo *modbus* e está relacionada com o código da função. Cada função é utilizada para acessar um tipo específico de dado, onde o mestre especifica o tipo de serviço ou função solicitada ao escravo (leitura, escrita, etc). No ScadaBR para este protocolo existem 4 opções:

1 - Bobina (*Coil Status*) = Equivalente a saída digital

2 - Entrada Discreta (*Input Status*) = Equivalente a entrada digital

3 - Registrador *holding* = Equivalente a saída analógica

4 - Registrador de entrada (*Input Register*) = Equivalente a entrada analógica

Como neste caso os dados que precisam ser coletados são números reais não binários, os pontos de dados apresentam-se como Registrador *Holding*.

- Tipo de dados *modbus*:

Neste campo configura-se o tipo de dado que deseja-se ler, os CLP's da Delta possuem algumas particularidades e após alguns testes chegou-se à seguinte conclusão:

Tabela 1 - Relação entre memória do CLP e configuração no ScadaBR

Memória do CLP	Valor que contém	Tipo de dado no ScadaBR
Tipo M	Variável do tipo binária (0 ou 1)	Bobina (Coil Status)
Tipo D	Número inteiro de 16 bits	Inteiro de 2 bytes (com ou sem sinal)*
Tipo D	Número inteiro de 32 bits	Inteiro de 4 bytes (com ou sem sinal)*
Tipo D	Número decimal de 32 bits	Float trocado de 4 bytes
Tipo D	Caracteres alfanuméricos	String de comprimento fixo

*Obs.: A escolha de com ou sem sinal vai depender se o número em questão tem ou não valor negativo.

- Offset (baseado em 0): Neste campo deve-se colocar o endereço *modbus* da memória que deseja-se ler. No caso da Delta no manual de programação do CLP existe uma tabela com estes endereços Anexo II Tabela 2. Como exemplo novamente será analisando a configuração utilizada para a memória D20062. Segundo a tabela do fabricante as memórias do tipo D possuem um range de D0~D29999 e o equivalente em *modbus* seria 400001~430000, como o ScadaBR baseia seu offset em zero, o endereço para a memória D20062 é igual a 20062. Detalhe que o primeiro dígito do range “4” não precisa ser preenchido já que este se refere ao código da função explicado anteriormente, e possui seu próprio campo. Mais detalhes sobre cada código do CLP Delta podem ser observados no Anexo II tabela 3.

- Número de registradores: Este campo só deve ser preenchido no caso de algum dado ocupar mais de uma memória do CLP.

- Codificação de caracteres: Por padrão é ASCII, característica do protocolo *modbus* nos CLP's Delta.

- Configurável: Este *checkbox* serve para permitir ao ScadaBR alterar o dado na memória do CLP. Marcar caso desejar trabalhar dessa forma.

- Multiplicador: Campo utilizado para multiplicar o valor do *datapoint* por algum fator. Utilizado para linearização.

- Aditivo: Campo utilizado para somar algum valor o valor do *datapoint*. Também utilizado para linearização.

Na Figura 5 é apresentado o modelo de *datapoints* que foi configurado para este projeto.



Figura 5. Exemplo de configuração de ponto de dados

E. Desenvolvimento de interface gráfica

Nesta etapa foram projetados os layouts de tela e representações gráficas. Para formação do conceito das telas baseou-se em um artigo “Metodologia para Desenvolvimento de IHMs de Alta Performance Visual” [9] do fabricante de supervisórios Elipse.

Para criação dos layouts de fundo utilizou-se o software Photopea, por suas ferramentas disponíveis, utilização via navegador e por ser um software gratuito. Baseando-se na resolução Full HD as telas criadas no photopea possuem tamanho de 1920x1080 pixels, e foram exportadas em formato PNG para importação no ScadaBR.

Já na interface do ScadaBR para adicionar a tela é necessário navegar até a aba “Representação gráfica” onde preenche-se o campo nome e realiza-se a importação da imagem de fundo pelo botão escolher arquivo. É importante clicar no botão salvar sempre que for feita alguma alteração, pois caso a página atualize todo o processo depois do último salvamento será perdido.

Na *droplist* “Componentes” encontram-se os elementos que podem ser adicionados a tela. Basta escolher o desejado e clicar no ícone de quebra-cabeça para que o componente seja inserido no centro da tela. Diferentes componentes possuem diferentes funcionalidades e configurações. Neste trabalho foram muito utilizados dois tipos, “HTML” e “Script para o servidor”. Estes dois componentes são bastante versáteis e possuem infinitas possibilidades.

1) *HTML*: O funcionamento deste componente é bastante simples, basta colocar o código HTML que deseja-se utilizar e clicar no disquete para salvar. Serve para adicionar textos, imagens, links e até mesmo para fazer a navegação entre as telas criadas. No Anexo III são apresentados alguns exemplos de códigos HTML utilizados.

É um componente bastante poderoso basta conhecer um pouco de linguagem HTML. Por ser uma linguagem bastante popular também é possível encontrar muitas informações e auxílio na internet. Uma limitação deste componente seria a impossibilidade de conecta-lo a um *datapoint*.

2) *Script para servidor*: O componente “Script para servidor” serve justamente para sanar esta necessidade da conexão com o *datapoint*. A linguagem que é executada dentro deste componente é JavaScript, e sua funcionalidade é bastante similar ao componente HTML. Basta colocar o código JSON que deseja-se utilizar e clicar no disquete para salvar.

A diferença deste componente seria um parâmetro que este possui em relação ao HTML. Este parâmetro permite escolher um ponto de dados dentre aqueles que estão cadastrados e utiliza-lo no script JSON.

Muito útil para exibir valores numéricos ou texto de um ponto de dados, além de imagens dinâmicas que alteram-se com mudanças de valores. Serve também para inserção de gráficos temporais relacionados a mudança de valor de um ou mais pontos de dados. No Anexo III são apresentados alguns códigos com maiores detalhes.

Assim como o HTML por ser uma linguagem bastante popular também é possível encontrar muitas informações e auxílio na internet. O restante dos componentes utilizados nas telas possui código semelhante aos apresentados com algumas variações. A inserção destes componentes foi feita em todas as telas e vários testes de posição e alinhamento foram feitos até chegar no resultado final.

F. Integração de ferramenta de dashboard a interface gráfica

Consistiu em integrar o ScadaBR ao Grafana. Trata-se de uma ferramenta de dashboard muito potente, que possibilita obter uma infinidade de dados existentes e visualizá-los como quiser, sem necessidade de ingestão de dados na lógica de *back-end*.

Visando uma fácil apresentação de dados e também a possibilidade de pesquisa por tempo “*time picker*”, o Grafana se encaixa perfeitamente. Além de possuir uma versão de software livre, possui incontáveis recursos de aquisição de informações de bancos de dados.

Para a ferramenta acessar as informações do banco de dados é necessário configurar sua conexão. Navegando até a aba *Data Source* é preciso encontrar em sua lista de fonte de dados o tipo de banco desejado, neste trabalho o utilizado foi MySQL.

Depois de selecionar o tipo de banco de dados é necessário configurá-lo. Com o endereço do banco, usuário e senha conforme momento de criação do banco. Após o preenchimento correto dos campos basta salvar a conexão, caso ocorra alguma incompatibilidade o Grafana irá apresentar uma mensagem de erro.

Feito a conexão com o banco de dados já é possível criar as dashboards. Após adicionar um novo painel é necessário configurá-lo, no canto superior direito da tela há uma *droplist* onde define-se qual o tipo de visualização, gráfico temporal, gráfico de barras, *gauge*, tabela, etc.

Para a dashboard buscar os dados do banco MySQL é necessário configurar sua consulta. A sintaxe é a mesma utilizada em comandos do MySQL. No campo Fonte de dados escolhe-se o banco configurado, e no campo UMA deve ser inserido o comando MySQL onde:

- SELECT seleciona quais colunas deseja-se visualizar;

- FROM informa de qual tabela deve buscar estas informações;

- WHERE especifica qual *datapoint* buscar;

- ORDER BY serve para organizar os valores baseado em algum parâmetro, como por exemplo tempo;

Na Figura 6 é possível verificar o código utilizado para consultar e apresentar os dados da tabela de registro de operação.

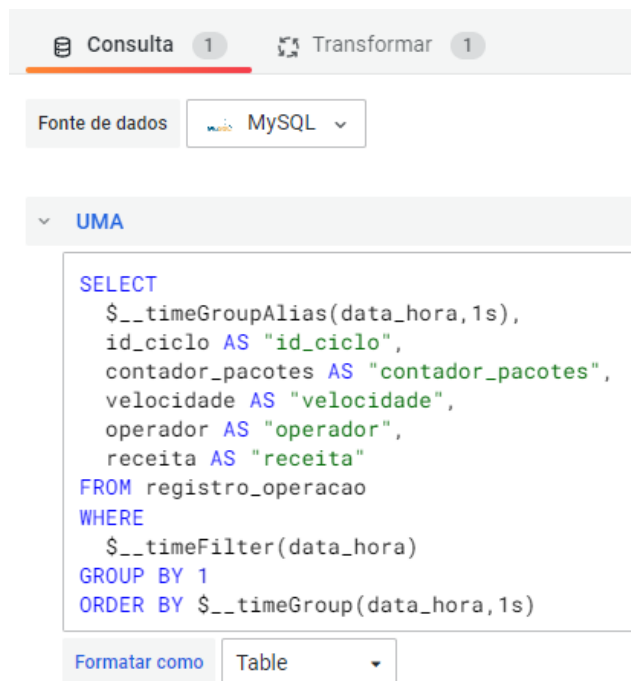


Figura 6. Exemplo de estrutura para leitura de valores do banco de dados via Grafana

E desta forma é possível obter os dados do banco MySQL no Grafana. Título da tabela ou gráfico, nome da coluna, alinhamento, cores, etc. podem ser totalmente customizados.

Depois de configuradas as tabelas e gráficos, para inseri-los ao ScadaBR é necessário habilitar a função *embed* do Grafana em seu arquivo de configurações. Com esta função ativa é possível utilizar um *iframe* no componente HTML do ScadaBR e inserir uma janela na representação gráfica.

G. Instalação do sistema de supervisão em hardware

Consistiu na instalação do sistema desenvolvido em um *Raspberry Pi*, trata-se de um minicomputador em uma única placa do tamanho bastante reduzido que executa o sistema operacional Linux. Acoplado em conjunto com o *Raspberry* foi utilizado uma expansão que acrescenta ao hardware algumas funcionalidades como duas portas ethernet, portas HDMI, bateria, ventilador, entre outras. A relação do hardware utilizado e seu custo pode ser observado na Tabela 2.

O modelo de *Raspberry* definido neste trabalho possui capacidade de processamento satisfatório para a aplicação além de memória e espaço de armazenamento adequados para a quantidade de informação. Seu hardware pode ser observado na Figura 7.

Tabela 2 – Hardware utilizado na configuração do minicomputador

Modelo	Descrição	Valor
CM4004032	Módulo de computador	R\$ 780,60*
CM4-DUAL-ETH-BOX-A	Placa base	R\$ 459,05 *

*Obs.: Valor de mercado em Agosto de 2022.

O processo de instalação foi feito via interface gráfica da própria placa, onde foram instalados todos os softwares que formam o sistema de supervisão: ScadaBR, MySQL, Grafana, Node-Red.

Além disso o minicomputador também foi configurado como uma VPN (*Virtual Private Network*) é uma conexão segura baseada em criptografia. O objetivo é transportar informação de forma segura combinando autenticação e tunelamento [10]. Desta forma apenas quem possui um certificado autenticado terá acesso as informações do equipamento.

Na Rede Privada Virtual é possível configurar certificados com níveis de segurança diferentes. Desta forma um técnico pode ter um acesso que permite a ele alterar de forma remota o programa do CLP ou algum parâmetro do sistema de supervisão, ao mesmo tempo que o certificado do usuário pode permitir que este apenas visualize o processo de monitoramento.



Figura 7. Imagem do hardware do minicomputador com Raspberry

H. Conexão do sistema de supervisão ao controle da empacotadora

Consistiu primeiro em carregar as versões alteradas do programa do CLP e da IHM no equipamento existente. Para que assim possa ser feita a comunicação com o minicomputador e a posterior coleta de dados.

Para o sistema funcionar de forma correta é necessário implementá-lo ao painel conforme Figura 8. Posteriormente um ponto de internet deve ser conectado à porta WAN do minicomputador. E a porta LAN por sua vez deve ser conectada ao CLP e IHM, como na maioria das vezes cada

equipamento possui apenas uma porta ethernet, para tudo ficar na mesma rede, se utiliza um switch industrial. Precisa ser industrial, pois em ambientes fabris existem muitas interferências que podem acabar afetando a comunicação correta entre equipamentos.

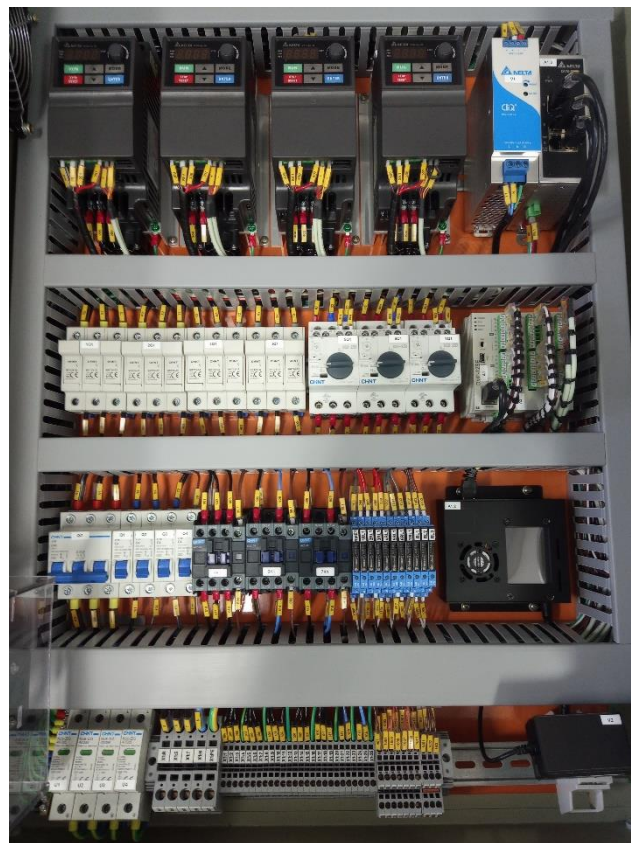


Figura 8. Placa de montagem da empacotadora com minicomputador já implementado

Com os endereços de IP configurados de forma correta no supervisor, a partir do momento que os equipamentos são interligados e o painel é energizado a coleta de informações inicia instantaneamente.

Para visualizar o sistema de supervisão há duas formas, a primeira seria utilizar a porta HDMI do próprio minicomputador conectado a um monitor. A segunda seria através da conexão VPN com instalação do certificado no dispositivo pretendido.

O resultado deste trabalho pode ser observado nas 6 figuras do Anexo IV onde é apresentado as telas da interface gráfica de supervisão.

VI. CONCLUSÕES

Conforme já destacado é de grande importância acompanhar a produção de uma máquina deste tipo para verificar seu desempenho e com base nisso estimar métricas. Neste escopo fica evidente a importância de um sistema de supervisão de fácil acesso e seguro. Sendo essencial para garantir um aumento da produtividade, eficiência, economia e segurança nas fábricas.

O objetivo principal deste trabalho foi alcançado integralmente, visto que foi possível desenvolver um sistema de supervisão satisfatório. Em testes com a máquina funcionando foi possível verificar sua eficácia, e possibilitou ao empresário implementar melhorias para aumentar a produção.

Com o sistema funcionando foi possível analisar o índice OEE e medir a eficiência da máquina, a média semanal ficou em torno de 82,578 %. Valor bastante satisfatório tendo em vista que a média do mercado está em 80% e um OEE de Classe Mundial (parâmetro de mercado para indústrias de alto nível) é de 85%. A utilização do supervisor permitiu esse acompanhamento direto, com dados produtivos de máquina que auxiliam no processo de tomada de decisão.

Uma importante redução nos custos do sistema desenvolvido foi a utilização do *Raspberry*. Como o ScadaBR possui uma interface relativamente leve, não é necessário adquirir um computador dedicado para isso. Se comparado a outros sistemas existentes no mercado, em que é necessário um hardware bastante robusto para rodar aplicações deste tipo.

O custo de investimento médio de uma licença de um supervisor pago, consolidado no mercado (AVEVA Edge, Elipse, ADISRA, etc.) varia em torno de R\$ 7.000,00 a R\$ 15.000,00 dependendo do número de *TAG's* e configurações escolhidas. Com a utilização de um sistema de plataforma livre, como o ScadaBR, houve redução dos custos possibilitando empresas de menor poder aquisitivo também possuírem um sistema de supervisão e iniciarem na indústria 4.0.

Em contrapartida foram necessárias muitas horas de desenvolvimento para criar o sistema. Foram gastas horas para aprender a utilizar as ferramentas ScadaBR, Node-RED, Grafana, MySQL Workbench além de horas em pesquisa com profissionais da área, fabricantes de máquinas e operadores.

Este trabalho foi muito importante para a formação acadêmica, pois, foi possível verificar na prática a utilização de conteúdos vislumbrados em diversas disciplinas durante a graduação, em especial as atividades que englobam acionamentos elétricos, projetos elétricos e automação industrial. Todavia foi necessário agregar muitos outros conhecimentos que não puderam ser abordados no curso, como por exemplo, banco de dados, linguagens de programação, além de desenvolver habilidades mais aprofundadas em áreas como programação *ladder*.

Um dos pontos analisados, e que apresentou desempenho superior ao previsto inicialmente, é relacionado ao espaço de armazenamento das informações. Após o sistema de supervisão operar por algum tempo, mediu-se o espaço ocupado pelo banco de dados e foi possível identificar um valor de 10MB por turno de 8 horas. Considerando o espaço restante disponível no minicomputador após a instalação do sistema, e dois turnos por dia em um ano com 250 dias úteis, o minicomputador teria capacidade de salvar aproximadamente 4 anos de informação.

Como o objetivo inicial deste projeto compreendia a visualização e acompanhamento, futuramente pode-se estudar a possibilidade da substituição da IHM por uma tela para operar a empacotadora pelo próprio supervisor. Podendo assim reduzir ainda mais os custos, tendo a possibilidade tanto

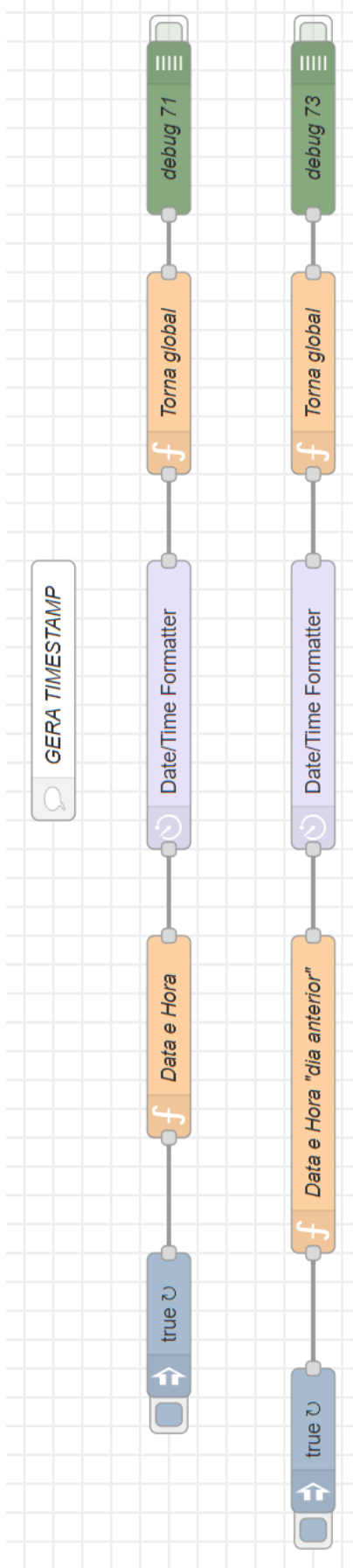
de visualizar os dados como também operar a máquina via supervisor.

Outra possibilidade também para trabalhos futuros, seria a implementação de um backup em nuvem, de todo o banco de dados e informações ligadas ao processo. Assim se por algum motivo o hardware local sofrer alguma avaria todas as informações estarão espelhadas e seguras em um servidor remoto.

REFERÊNCIAS

- [1] SLACK, N.; CHAMBERS, S.; JOHNSTON, R.; BETTS, A. Gerenciamento de Operações e de Processos. Porto Alegre: Bookman, 2013.
- [2] LUGLI, A.B.; SANTOS, M.M.D. REDES INDUSTRIAIS PARA AUTOMAÇÃO INDUSTRIAL - AS-I, PROFIBUS E PROFINET. São Paulo: Editora Saraiva, 2019.
- [3] HENRY, J. R. Packaging Machinery Handbook. Createspace Independent Publishing Platform, 2012.
- [4] SANTOS, M.M. D. Supervisão de Sistemas - Funcionalidades e Aplicações. São Paulo: Editora Saraiva, 2014.
- [5] ROQUE, L.A.O. L. Automação de Processos com Linguagem Ladder e Sistemas Supervisórios. Rio de Janeiro: Grupo GEN, 2014.
- [6] SENSORWEB. Manual do Software SCADABR. Florianópolis: 2010.
- [7] DELTA ELECTRONICS. AS Series Module Manual. Taiwan: 2021.
- [8] MANZANO, J.A.N. G. MySQL 5.5 Interativo: Guia Essencial de Orientação e Desenvolvimento. São Paulo: Editora Saraiva, 2011.
- [9] GOETZ, H. F. Metodologia para Desenvolvimento de IHMs de Alta Performance Visual. Disponível em: <https://kb.elipse.com.br/metodologia-para-desenvolvimento-de-ihms-de-alta-performance-visual/>. Acesso em: 22 out. 2022
- [10] MORAES, A.F. D. Segurança em Redes - Fundamentos. São Paulo: Editora Saraiva, 2010.

ANEXO I – ESTRUTURA DE BLOCOS DESENVOLVIDA NO NODE-RED



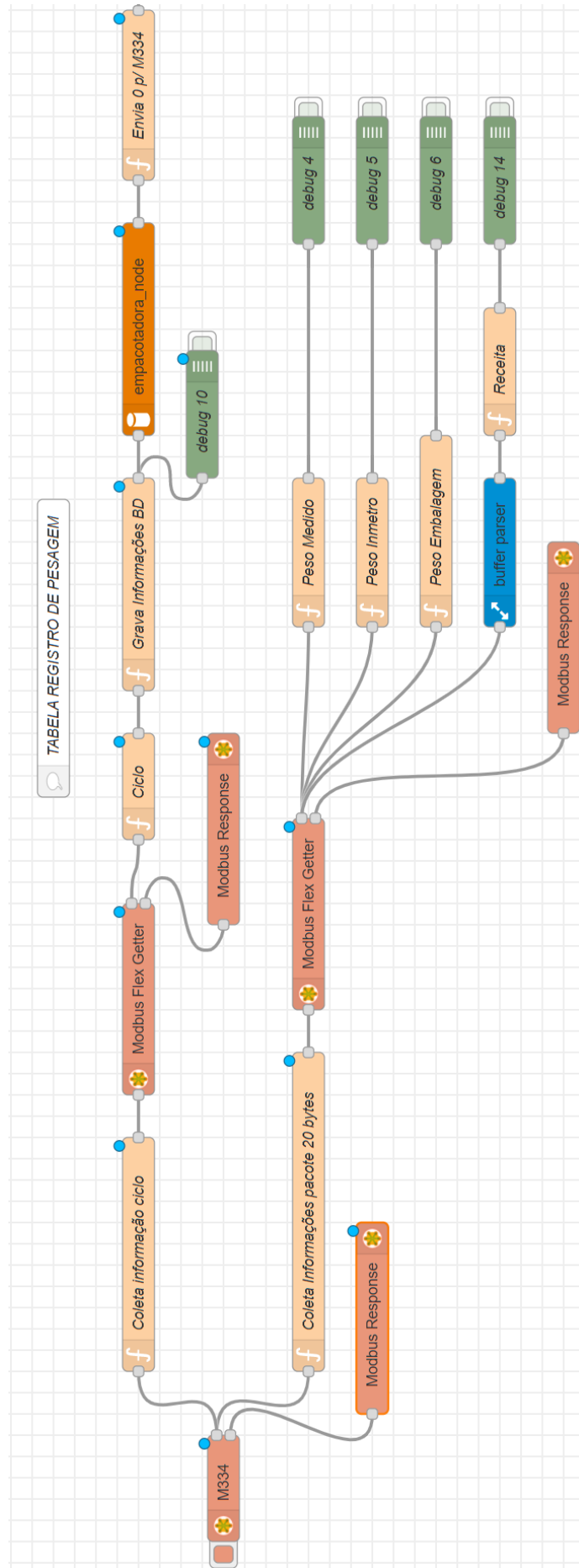
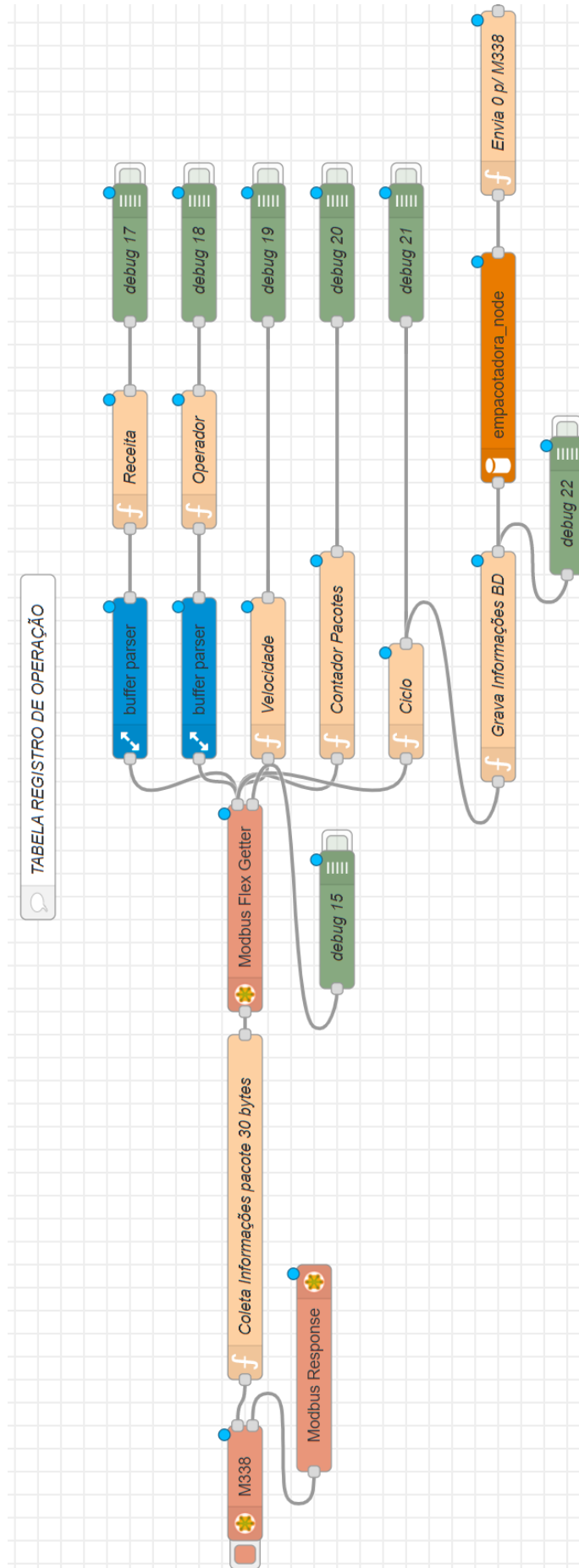
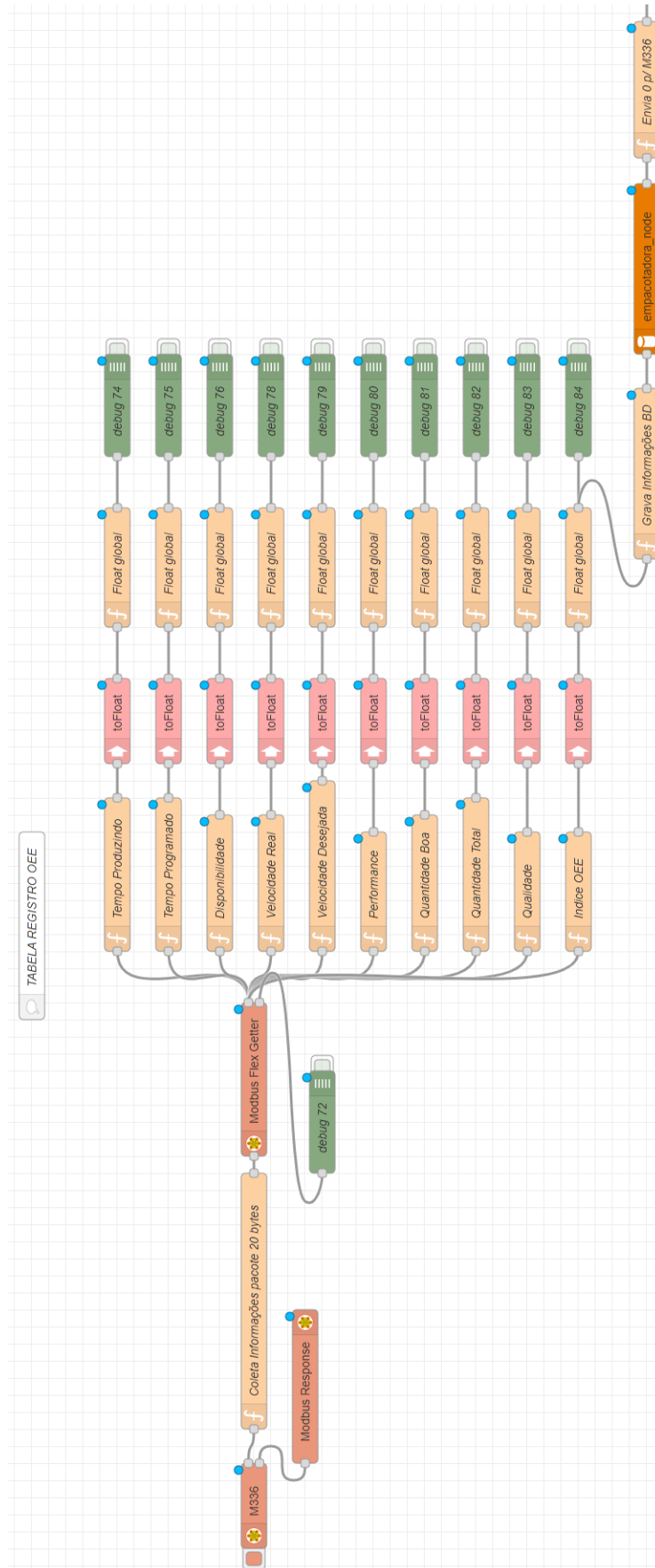


TABELA REGISTRO DE PESAGEM





ANEXO II – MAPA MODBUS DOS CONTROLADORES LÓGICOS PROGRAMÁVEIS DA MARCA DELTA

Chapter 10 Function Cards

10.2.10.9 Modbus TCP

● **Modbus TCP Specifications**

Item	Specification
Device type	Modbus TCP Server
Communication port	502
Maximum connection number	8
Maximum data length	200 words

● **Standard Modbus device address**

Device	Type	Format	Range	Modbus address (Dec)	AS Series Address (Hex)
X	Bit	DD.DD	X0.0~X63.15	124577~125600	6000~63FF
	Word	DD	X0~X63	332769~332832	8000~803F
Y	Bit	DD.DD	Y0.0~Y63.15	040961~041984	A000~A3FF
	Word	DD	Y0~Y63	440961~441024	A000~A03F
M	Bit	DDDD	M0~M8191	000001~008192	0000~1FFF
SM	Bit	DDDD	SM0~SM4095	016385~020480	4000~4FFF
SR	Word	DDDD	SR0~SR2047	449153~451200	C000~C7FF
D	Word	DDDDD	D0~D29999	400001~430000	0000~752F
S	Bit	DDDD	S0~S2047	020481~022528	5000~57FF
T	Bit	DDD	T0~T511	057345~057856	E000~E1FF
	Word	DDD	T0~T511	457345~457856	E000~E1FF
C	Bit	DDD	C0~C511	061441~061952	F000~F1FF
	Word	DDD	C0~C511	461441~461952	F000~F1FF
HC	Bit	DDD	HC0~HC255	064513~064768	FC00~FCFF
	DWord	DDD	HC0~HC255	464513~464768	FC00~FCFF
E	Word	DD	E0~E9	465025~465039	FE00~FE09

AS Series Module Manual

- **Standard Modbus function codes and range**

Function code	Description	Applicable to devices	Supported device range
01	Read multiple bit devices	X, Y, M, SM, S, T, C, HC	1~1600
02	Read multiple bit devices	X, Y, M, SM, S, T, C, HC	1~1600
03	Read multiple word devices	X, Y, SR, D, T, C, HC, E	1~100, but for HC: 1~50
04	Read multiple word devices	X	1~100
05	Write the status in a single bit device	Y, M, SM, S, T, C, HC	1
06	Write data in a single word device	Y, SR, D, T, C, HC, E	1
0F	Write the status in multiple bit devices	Y, M, SM, S, T, C, HC	1~1600
10	Write the status in multiple word devices	Y, SR, D, T, C, HC, E	1~100, but for HC: 1~50
17	Read/write the status from/in multiple word devices	Y, SR, D, T, C, HC, E	1~100, but for HC: 1~50

10.2.10.10 Network Security

To enhance security and performance of the system, it is suggested to use closed network or LAN with firewall protection to prevent cyber-attacks.

ANEXO III – CÓDIGOS UTILIZADOS NA INTERFACE GRÁFICA DO SISTEMA DE SUPERVISÃO

➔ EXEMPLOS HTML

```

```

Com esta linha de código é possível inserir uma imagem e também ajustar seu tamanho pelos parâmetros width e height. Lembrando que para este código funcionar é necessário que adicionar previamente a imagem na pasta “graphics” do ScadaBR.

```
<span style="white-space: nowrap; color: #363636; font-weight: bold; font-size: 24px; font-family:'Roboto', sans-serif;">EMPACOTADORA MESA FIXA DOSADOR DE GAVETA</span>
```

Com o código acima é possível inserir um texto a tela e em seu parâmetro “style” configurar diversos aspectos como cor, tamanho, propriedade, etc.

Também com o componente HTML é possível realizar o link para as outras telas do supervisor, abaixo é apresentado o código para isto.

```
<a href="/empacotadora/views.shtm?viewId=3">  
<button type="button" display="none" style="width:370px; height:69px; cursor:pointer;  
background:transparent; border:none; white-space: nowrap; color: #fff; font-weight: none; font-size:  
24px; font-family:'Roboto', sans-serif;">Registro de Produção</button>  
</a>
```

➔ EXEMPLOS SCRIPT PARA SERVIDOR

```
var s = "<div style=\"border:groove ; border-color:lightgray; padding: 2px; border-width: thin;  
color:#363636; background-color: none; width:500px; font-size: 28px; font-family:'Roboto', sans-serif;  
text-align: center; font-weight: bold;\"> " + "<span style=\"font-weight:bold;\">" + renderedText +  
"</span></div>"; return s;
```

Este script permite exibir o valor de um data point, este por exemplo foi utilizado para exibir o nome da receita. O script para mostrar variáveis numéricas é uma variação deste.

```
var descricao = "";  
var imagem_padrao = "";  
var usar_valor_exato = false;  
// Configurações de aparência  
var height = 105;  
var width = 90;  
// Conjunto de imagens a serem exibidas p/ cada valor  
// Use o formato >> "imagem:valor", <<  
var imagens = [  
    "/empacotadora/graphics/images/alarm.png:0",  
    "/empacotadora/graphics/images/manual.png:1",  
    "/empacotadora/graphics/images/ciclo.png:2",  
    "/empacotadora/graphics/images/automatic.png:3",  
];  
var s = "";  
s += "<img ";  
if (descricao.length > 0) {  
    s += "title=\"" + descricao + """;  
}  
var src = imagem_padrao;  
for (var i in imagens) {  
    var index = Number(i);  
    var valor_atual = Number(imagens[index].substring((imagens[index].lastIndexOf(":") + 1));  
    var proximo_valor = undefined;  
    if (typeof imagens[(index + 1)] == "string") {
```

```

        proximo_valor = Number(imagens[(index + 1)].substring((imagens[(index +
1)].lastIndexOf(":") + 1)));
    }
    if (usar_valor_exato == true) {
        if (valor_atual == value) {
            src = imagens[index].substring(0, imagens[index].lastIndexOf(":"));
        }
    } else {
        if ((value >= valor_atual) && ((value < proximo_valor) || (typeof proximo_valor ==
"undefined"))) {
            src = imagens[index].substring(0, imagens[index].lastIndexOf(":"));
        }
    }
}
s += "style='height:" + height + "px; width:" + width + "px;' ";
s += "src='" + src + "'";
s += "> </img>";
return s;

```

Este script serve para exibir uma imagem diferente dependendo do valor do data point. Neste caso foi utilizado para representar o estado da máquina, que possui quatro possibilidades.

O último script e talvez mais complexo exibe os valores de um determinado data point com o passar do tempo.

```

// Points to be included in chart (ID or XID)
var reading_points = [ point.id ];

```

```

// Color of each series in the graph
// (the position of each color corresponds
// to a point in "reading_array" variable)
var series_colors = [ '#05C5F080' ];

```

```

// Description of each series in the graph
// (each description corresponds to a point
// in "reading_array" variable)
var descriptions = [ 'Pacotes' ];

```

```

// Time unit adopted in chart:
// 0 -> Seconds
// 1 -> Minutes
// 2 -> Hours
var time_unit = 2;

```

```

// Time period adopted in chart (in the unit
// specified by "time_unit" variable)
var time_value = 80;

```

```

// Customize chart size (in pixels)
var height = 350;
var width = 702;

```

```

// Configure chart title
var show_title = false;
var title = 'Evolução da Produção';

```

```

// Start Y-axis always at 0
var begin_Y_at_zero = true;

```

```

// Enable chart animations

```

```
var enable_animations = false;

var invalid_message = "is not a numeric, multistate or binary datapoint";
var invalid_html = "";

// Get datapoint identifiers (ID/XID)
function getDataPointIds(identifier) {
    var dpDAO = new com.serotonin.mango.db.dao.DataPointDao();
    var dp = dpDAO.getDataPoint(identifier);

    var point_id = dp.getId();
    var point_xid = String(dp.getXid());
    return { id: point_id, xid: point_xid };
}

// Get data point type
function getDataPointType(identifier) {
    var types = {
        0: "UNKNOWN",
        1: "BINARY",
        2: "MULTISTATE",
        3: "NUMERIC",
        4: "ALPHANUMERIC",
        5: "IMAGE"
    }

    var dpDAO = new com.serotonin.mango.db.dao.DataPointDao();
    var dp = dpDAO.getDataPoint(identifier);
    var locator = dp.getPointLocator();
    return types[locator.getDataTypeId()];
}

// Get data points' values and times
function readPoints(id) {
    // Units: Second, minute, hour
    var unit_values = [ 1000, 60000, 360000 ];
    var index = (time_unit > 3) ? 0 : time_unit;
    var since = new Date().getTime() - (time_value * unit_values[index]);

    var val = new com.serotonin.mango.db.dao.PointValueDao();
    return val.getPointValues(id, since);
}

// Create a JSON array with a point value history
function createDataArray(obj, is_binary) {
    var foo = '[';
    var size = obj.size() - 1;

    for (var i = size; i >= 0; i--) {
        var time = obj.get(i).time;
        var value = obj.get(i).value;
        if (is_binary)
            value = String(value) == "true" ? 1 : 0;

        foo += '{"x":' + time + ',';
        foo += '"y":' + value + '}'

        if (i != 0) {
            foo += ',';
        }
    }
}
```

```
}

foo += ']';
return foo;
}

// Create a JSON object compatible with Chart.js "datasets"
function createJSONDatasets() {
    var size = reading_points.length;
    var foo = '[';

    for (var i = 0; i < size; i++) {
        var is_binary = false;
        var dp_id = getDataPointIds(reading_points[i]).id;
        var dp_type = getDataPointType(dp_id);
        var point_values = readPoints(dp_id);

        // Don't include non numeric datapoints in array
        if (dp_type == "BINARY") {
            is_binary = true;
        } else if (dp_type != "NUMERIC" && dp_type != "MULTISTATE") {
            invalid_html += descriptions[i] + ": " + invalid_message + "<br>";
            continue;
        }

        if (foo != '[')
            foo += ',';

        var reading_array = createDataArray(point_values, is_binary);
        foo += '{';
        foo +=     "label":'" + descriptions[i] + "',';
        foo +=     "data":' + reading_array + ',';
        foo +=     "backgroundColor":'" + series_colors[i] + "',';
        foo +=     "borderColor":'" + series_colors[i] + ""';
        foo += '};'

    }

    foo += ']';
    return foo;
}

// Create a JSON string that can be interpreted
// by FUScaBR library
function createFinalJSON() {
    var foo = "";

    foo += '{';
    // Specific FUScaBR options
    foo +=     "height":'" + height + "',';
    foo +=     "width":'" + width + "',';
    foo +=     "beginAtZero":' + begin_Y_at_zero + ',';
    foo +=     "timeBased":true,';
    foo +=     "animations":' + enable_animations + ',';
    foo +=     "showTitle":' + show_title + ',';
    foo +=     "title":'" + title + "',';
    // Chart.js options
    foo +=     "type":'"line",';
    foo +=     "data":{';
```



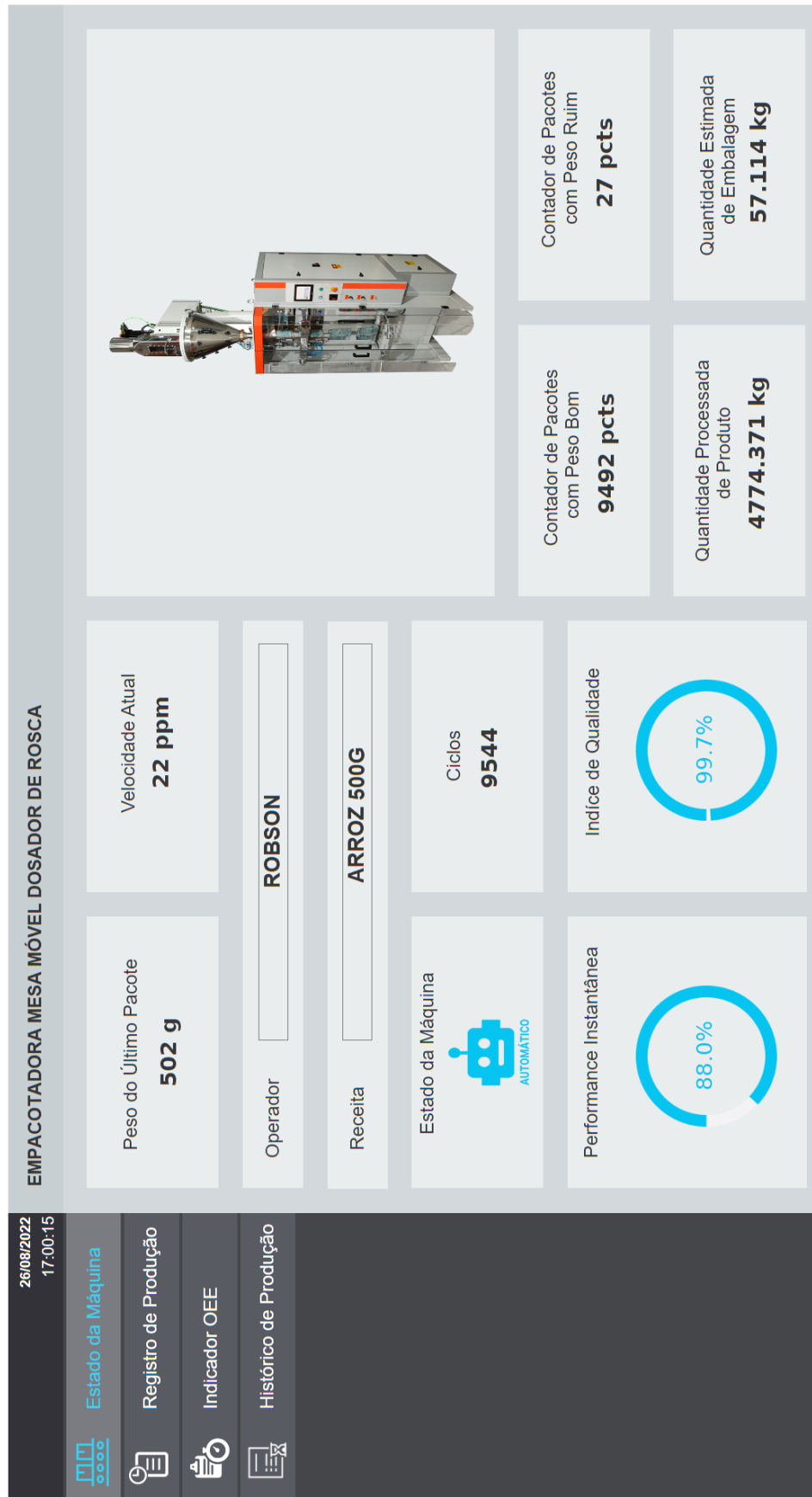
```
    foo +=                "'datasets':" + createJSONDatasets());
    foo +=                '};
    foo +=                '};

    return foo;
}

var s = "<input class='fuscabr-chart-data' type='hidden' value='" + createFinalJSON() + "'>";
if (invalid_html.length)
    s += invalid_html;

/* Uncomment the line below to show the generated JSON code */
//s += "<div style='width: 500px;'>" + createFinalJSON() + "</div>";
return s;
```

ANEXO IV – INTERFACE GRÁFICA DO SISTEMA DE SUPERVISÃO



EMPACOTADORA MESA MÓVEL DOSADOR DE ROSCA

Tempo em Falha
0 minutos

Tempo em Ciclo
434 minutos

Tempo em Manual
0 minutos

Tempo em Setup
0 minutos



Evolução da Produção

Registro de Peso Diário

Peso Bom	9498	Peso Ruim	27
Pacotes		Embalagem	
4761.477 kg		56.994 kg	
15.907 kg		0.162 kg	
4777.384 kg		57.150 kg	
Total		Total	

26/08/2022
17:00:32

- Estado da Máquina
- Registro de Produção
- Indicador OEE
- Histórico de Produção

©2009-present Fundação Cen. IMA Sistemas. Uma Sistemas. Conetec. Todos os direitos reservados.



26/08/2022
17:00:47

Estado da Máquina

Registro de Produção

Indicador OEE

Histórico de Produção

EMPACOTADORA MESA MÓVEL DOSADOR DE ROSCA

Disponibilidade

Tempo produzindo	=	435 min
Tempo programado para produzir	=	528 min

82.4%

Qualidade

Quantidade de Pacotes Bons	=	9505 pcts
Quantidade Total Produzida	=	9532 pcts

99.7%

Performance

Velocidade Média Real	=	21 ppm
Velocidade Desejada para Tarefa	=	25 ppm

84.0%

OEE Instantâneo

Disponibilidade	X	Qualidade	X	Performance
-----------------	---	-----------	---	-------------

69.0%

© 2009-presente Fundação Cenit - IMA Sistemas, Unia Sistemas, Conetec. Todos os direitos reservados.

Instituto Federal de Santa Catarina – Jaraguá do Sul - RAU
 Rua dos Imigrantes, 445 | Rau | Jaraguá do Sul /SC | CEP: 89254-430
 Fone: (47) 3276-9600 | www.gw.ifsc.edu.br | CNPJ 11.402.887/0005-94



26/08/2022
17:01:02

Estado da Máquina

Registro de Produção

Indicador OEE

Histórico de Produção

Registro operação

Registro peso

Registro OEE

EMPACOTADORA MESA MÓVEL DOSADOR DE ROSCA

HISTÓRICO DE PRODUÇÃO / Registro Operação

Registro de Operação					
Data e Hora	Ciclo	Contador de Pacotes	Velocidade de Empacotamento	Operador	Recetta
2022-08-26 17:00:57	116762	9561	22	ROBSON	ARROZ 500G
2022-08-26 17:00:54	116761	9560	22	ROBSON	ARROZ 500G
2022-08-26 17:00:51	116760	9559	22	ROBSON	ARROZ 500G
2022-08-26 17:00:49	116759	9558	22	ROBSON	ARROZ 500G
2022-08-26 17:00:46	116758	9557	22	ROBSON	ARROZ 500G
2022-08-26 17:00:43	116757	9556	22	ROBSON	ARROZ 500G
2022-08-26 17:00:40	116756	9555	21	ROBSON	ARROZ 500G
2022-08-26 17:00:38	116755	9554	22	ROBSON	ARROZ 500G
2022-08-26 17:00:35	116754	9553	22	ROBSON	ARROZ 500G
2022-08-26 17:00:32	116753	9552	21	ROBSON	ARROZ 500G

Velocidade de Empacotamento

22 ppm

Evolução da Produção

— Pacotes

©2009-present Fundação Cen: MCA Sistemas, Unia Sistemas, Conetec. Todos os direitos reservados.



26/08/2022
17:01:18

Estado da Máquina

Registro de Produção

Indicador OEE

Histórico de Produção

Registro operação

Registro peso

Registro OEE

EMPACOTADORA MESA MÓVEL DOSADOR DE ROSCA

HISTÓRICO DE PRODUÇÃO / Registro Peso

Registro de Peso					
Data e Hora ↓	Ciclo	Peso da Embalagem	Peso Inmetro	Peso Medido	Recetta
2022-08-26 17:01:13	116768	6.00 g	500 g	503 g	ARROZ 500G
2022-08-26 17:01:10	116767	6.00 g	500 g	502 g	ARROZ 500G
2022-08-26 17:01:07	116766	6.00 g	500 g	502 g	ARROZ 500G
2022-08-26 17:01:05	116765	6.00 g	500 g	502 g	ARROZ 500G
2022-08-26 17:01:02	116764	6.00 g	500 g	502 g	ARROZ 500G
2022-08-26 17:00:59	116763	6.00 g	500 g	502 g	ARROZ 500G
2022-08-26 17:00:57	116762	6.00 g	500 g	502 g	ARROZ 500G
2022-08-26 17:00:54	116761	6.00 g	500 g	502 g	ARROZ 500G
2022-08-26 17:00:51	116760	6.00 g	500 g	502 g	ARROZ 500G
2022-08-26 17:00:49	116759	6.00 g	500 g	502 g	ARROZ 500G

Aferição de Peso

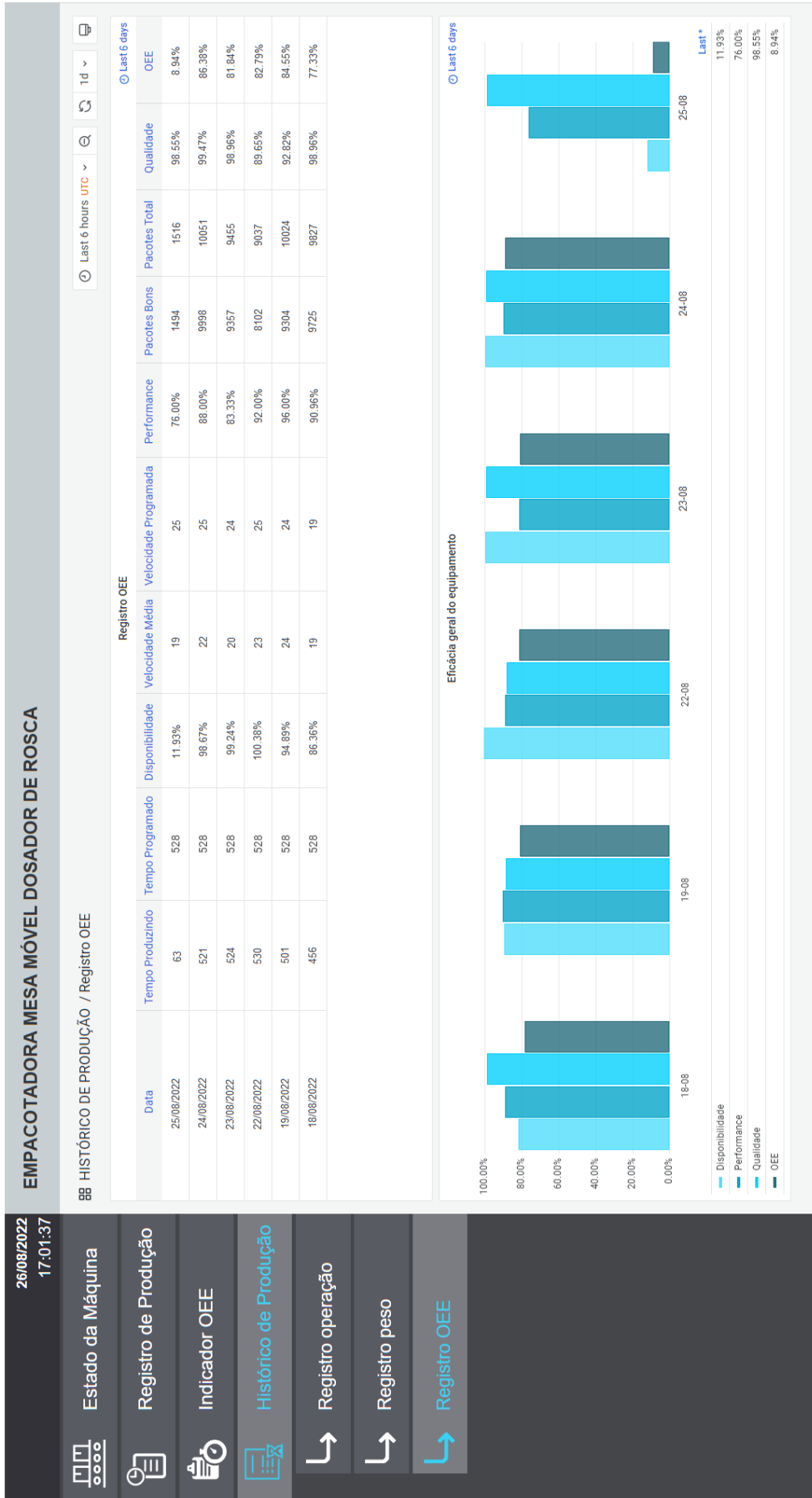
503.0
502.5
502.0
501.5
501.0
500.5
500.0

16:56:30 16:56:45 16:57:00 16:57:15 16:57:30 16:57:45 16:58:00 16:58:15 16:58:30 16:58:45 16:59:00 16:59:15 16:59:30 16:59:45 17:00:00 17:00:15 17:00:30 17:00:45 17:01:00 17:01:15

— Peso Inmetro — Peso Medido

ⓘ Last 5 minutes

©2009-present Fundação Cent. MCA Sistemas. Unia Sistemas. Conex. Todos os direitos reservados.



26/08/2022
17:01:37

Estado da Máquina

Registro de Produção

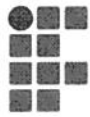
Indicador OEE

Histórico de Produção

Registro operação

Registro peso

Registro OEE



NOME DO ALUNO:
Luan Hubner

TÍTULO DO TRABALHO DE CONCLUSÃO DO CURSO:
Desenvolvimento De Um Sistema De Monitoramento Para Máquina Empacotadora Utilizando Supervisório Em Plataforma Aberta

Este trabalho foi julgado adequado para obtenção do título em Bacharel em Engenharia Elétrica, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

Jaraguá do Sul, 10 de dezembro de 2022.

Prof. Dr. Luiz Fernando Henning
Orientador

IFSC – Campus Jaraguá do Sul – Rau

Prof. MSc Edilson Hipólito da Silva
IFSC – Campus Jaraguá do Sul – Rau

Edgard Vieira Ribeiro
P&R Automação Industrial