

UNICESUMAR - UNIVERSIDADE DE MARINGÁ
CENTRO DE CIÊNCIAS EXATAS TECNOLÓGICAS E AGRÁRIAS
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**APLICAÇÃO DE CONCEITOS DE ÍNDUSTRIA 4.0 À AUTOMAÇÃO DE
PROCESSOS**

ALEXANDRE DOS SANTOS COELHO

MARINGÁ – PR
2021

ALEXANDRE DOS SANTOS COELHO

**APLICAÇÃO DE CONCEITOS DE ÍNDUSTRIA 4.0 À AUTOMAÇÃO DE
PROCESSOS**

Artigo apresentado ao Curso de Graduação em Engenharia Elétrica da UNICESUMAR – Universidade de Maringá como requisito parcial para a obtenção do título de Bacharel(a) em Engenharia, sob a orientação do Prof. Me. Fábio Augusto Gentilin

MARINGÁ – PR

2021

FOLHA DE APROVAÇÃO
ALEXANDRE DOS SANTOS COELHO

**APLICAÇÃO DE CONCEITOS DE ÍNDUSTRIA 4.0 À AUTOMAÇÃO DE
PROCESSOS**

Artigo apresentado ao Curso de Graduação em Engenharia Elétrica da UNICESUMAR –
Universidade de Maringá como requisito parcial para a obtenção do título de Bacharel(a) em
Engenharia, sob a orientação do Prof. Me Fábio Augusto Gentilin.

Aprovado em: 25 de novembro de 2021.

BANCA EXAMINADORA

Msc. Fábio Augusto Gentilin

Msc. Murillo Vilela Magan

Msc. Leandro Thiago Loureiro

APLICAÇÃO DE CONCEITOS DE ÍNDUSTRIA 4.0 À AUTOMAÇÃO DE PROCESSOS

Alexandre dos Santos Coelho

RESUMO

Este trabalho tem como objetivo, a elaboração da automação de um processo industrial, simulado em software, com a integração de plataformas IIoT, visando obter os dados de tal simulação em tempo real, para otimizar o fluxo de informações para que, se aplicado definitivamente em uma empresa, possa dar apoio à diversos setores gerenciais de tal. O sistema é desenvolvido através de uma plataforma de programação de controladores lógicos programáveis, no qual, possibilita a criação da lógica de controle do processo e seu sistema supervisor, visando simplicidade na operação e uma série de recursos almejados por empresas no seu processo produtivo. Por sua vez, a plataforma de fluxo de informações com recursos IoT, faz o intermédio entre uma rede de informações local e um servidor cloud, que armazena tais dados em sua plataforma e disponibiliza o acesso dessas informações de maneira remota, com segurança e praticidade.

Palavras-chave: Controladores lógicos programáveis. Internet da Coisas. Produção Industrial.

APPLICATION OF INDUSTRY 4.0 CONCEPTS TO PROCESS AUTOMATION

ABSTRACT

This work aims to develop the automation of an industrial process, simulate in software, with an integration of IIoT platforms, it aiming to get the data of such simulation in real time, to optimize the flow of information, so that, if applied definitely in a company, the system can support various managerial sectors. The system is developed through a programmable logic controller platform, which enables the creation of the process control logic and its supervisor system, it aiming simplification of operation and several features accomplished by companies in their production process. In turn, the flow information platform with IIoT features intermediates between a local information network and a cloud server, which stores such files on its platform and provide the remote access of these information, safely and conveniently.

Keywords: Programmable Logic Controller, Internet of things. Industrial Production.

1 INTRODUÇÃO

Atualmente, com o grande consumo de produtos industrializados de diversos setores e seguimentos, buscam-se, cada vez mais, melhoramentos nos processos de fabricação, visando aumentar a produção, reduzir custos, aumentar a qualidade e o padrão dos itens fabricados.

Para atingir tais objetivos, as empresas recorrem ininterruptamente à recursos de automação industrial, todavia, muito dos recursos já consolidados, limitam-se à uma solução local, geralmente disposta de um CLP (Controlador lógico programável), ligado à uma rede de tecnologia de automação (T.A.) que irá conectar os demais dispositivos, como: servidores locais, sistemas supervisórios e vários outros tipos de *Clients*.

O que há de novo na Automação Industrial é a aplicação de conceitos de Internet das coisas (IoT) e Sistemas Ciber físicos, em que ambos já são diversamente utilizados em outros segmentos (automação residencial, sistemas de segurança, dispositivos de monitoramento de sinais vitais de saúde, etc.). Tais conceitos, quando unidos na indústria, formam o que é conhecido como indústria 4.0. O termo “4.0” faz referência à 4ª revolução industrial, que assim como as anteriores, marcam mudanças e aprimoramentos nos processos de manufatura. Pode-se observar a ordem cronológica das revoluções industriais na Figura 1.

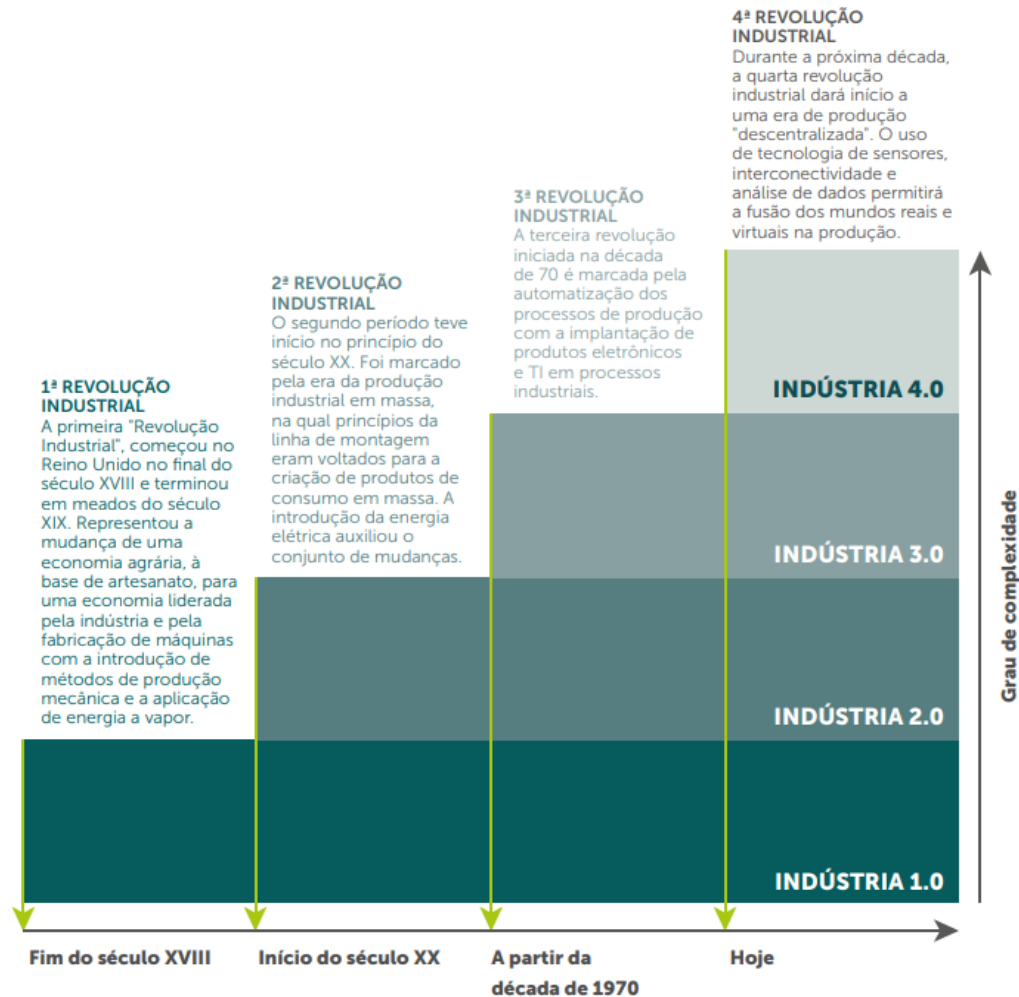
Para Heiner, et al (2014) o avanço da digitalização de informações, combinados com os conceitos de IoT e tecnologias de campo “smart”, guiam para novos paradigmas na produção industrial. Tais conceitos podem possibilitar a construção de sistemas com maior apoio na gestão empresarial, com alto fluxo de informações atuais do processo, podendo assim, auxiliar nas tomadas de decisões.

A Indústria 4.0 está alicerçada em tecnologias como a Internet das coisas e objetos inteligentes, construindo sistemas com maior capacidade de autogestão, possibilitando uma maior customização dos produtos sem perder as vantagens da produção em massa (LASI et al., 2014).

Para Tabaa et al. (2018) plantas inteligentes produzem recursos e os processos são caracterizados por sistemas ciber físicos, os quais oferecem benefícios significantes, em tempo real em termos de qualidade, tempo, recursos e custos melhores ao comparados com sistemas de produção convencionais.

De acordo com Drath & Horch, 2014 apud Pereira & Simonetto (2018) tais sistemas, podem ser conceituados em três hipóteses:

Figura 1 - Evolução da produção industrial



Fonte: Firjan (2016)

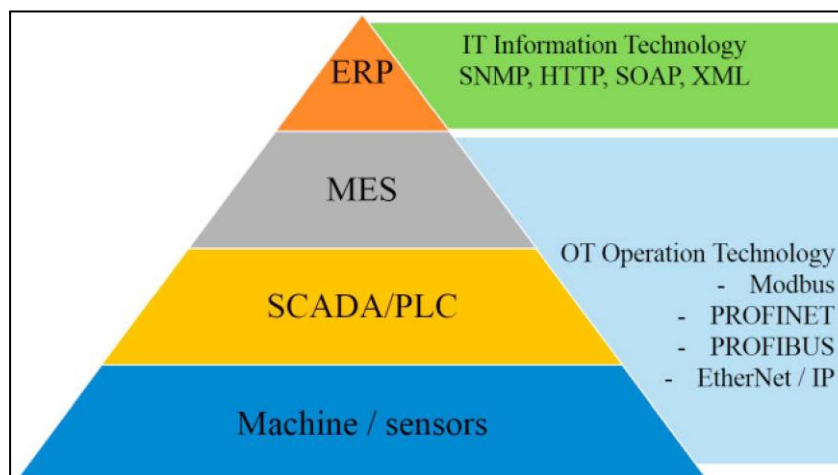
1. A primeira hipótese trata da comunicação em sistemas de produção, que está cada vez mais fácil devido à maior disponibilidade de infraestrutura. Assim, a comunicação pode ser feita em qualquer lugar, auxiliando serviços de engenharia, configuração, diagnósticos, operação, entre outros. Essa tendência, segundo os autores, é imparável, e não é forçada por ninguém.
2. Dispositivos de campo, máquinas, fábricas e até mesmo produtos individuais estarão conectadas a uma rede, como a Internet, e serão capazes de armazenar dados em tempo real. Esses dispositivos poderão ser acessados através da rede a partir de qualquer lugar, podendo ser buscados e analisados.

3. Dispositivos de campo, máquinas, fábricas e até mesmo produtos individuais terão capacidade de armazenar documentos e conhecimento sobre si dentro da rede, fora de sua estrutura física. Essas informações podem ser atualizadas com o passar do tempo e, a partir delas, algumas funcionalidades podem ser executadas, como negociações.

Essas hipóteses podem ser exemplificadas como, células de produção interligadas dentro à toda estrutura da rede T.A. da empresa, tendo uma rede de comunicação entre dispositivos de campo, que fornecem e armazenam variáveis primárias e secundárias internamente além de compartilha-las com os demais dispositivos.

Ilustra-se esses exemplos na Figura 2.

Figura 2 - Sistemas ciber físicos na indústria



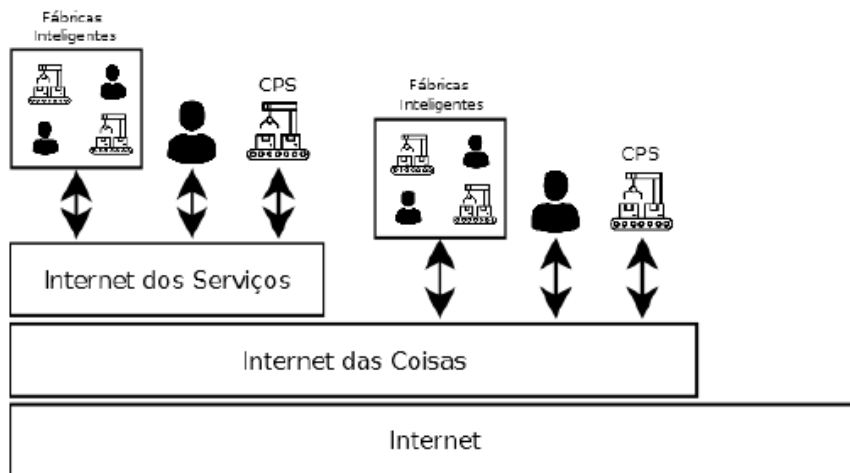
Fonte: Tabaa, Chouri, Saadaoui, & Alami (2018)

Segundo Hermann, Pentek e Otto (2016) apud Pereira & Simonetto, (2018) a Indústria 4.0 é composta por quatro componentes: (i) Sistemas Ciber-Físicos; (ii) Internet das Coisas; (iii) Internet de Serviços; e (iv) Fábricas Inteligentes. Os CPS são os componentes que integram o mundo físico ao virtual; são equipamentos que armazenam dados sobre o seu estado e realizam operações.

Internet das coisas (IoT) pode ser definida como o sistema de máquinas, objetos, sensores que podem transmitir e receber dados através de uma rede, sem que haja uma intervenção humana.

Assim, pode-se ilustrar a estrutura da indústria 4.0 de acordo com a Figura 3.

Figura 3 - Estrutura da indústria 4.0



Fonte: Pereira & Simonetto (2018)

Visto tais aspectos sobre Indústria 4.0, pretende-se, neste artigo, elaborar e simular a lógica de controle e supervisão de um processo industrial em batelada, utilizando um CLP virtual, com seus respectivos softwares necessários, do fabricante Rockwell Automation e migrar dados importantes do processo para um servidor *Cloud* através do software Node-RED e também armazenar dados importantes em um servidor SQL.

2 DESENVOLVIMENTO

A seguir, o método empregado para a elaboração da planta industrial simulada, bem como sua comunicação com a nuvem.

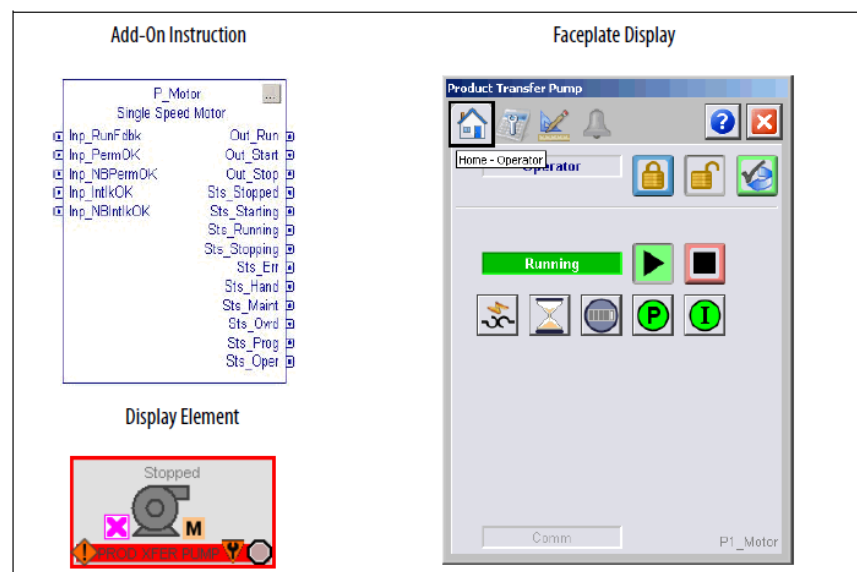
2.1 Conceitos iniciais para desenvolvimento da lógica e supervisório

Para o desenvolvimento da lógica e sistema supervisório, foi usada a biblioteca de objetos de processos *PlantPax* v3.5 da Rockwell Automation, pois, esta apresenta inúmeras

vantagens como: criação automática de tags, animações exclusivas no sistema supervisório, telas de engenharia, manutenção e operação, visualização de intertravamentos e permissivos, etc. Cabe ao projetista, adicionar as *add-ons* (rotinas de programação já elaboradas) à lógica do programa, e os objetos gráficos necessários para cada animação e tela no ambiente de configuração do sistema supervisório.

A Figura 4 ilustra um exemplo de animação e *add-on* de uma partida direta de um motor elétrico.

Figura 4 - Add-on e faceplates PlantPax



Fonte: Rockwell Automation, Inc. (2017)

Os softwares necessários para esta aplicação são:

1. Studio 5000 v.31: usado para elaborar a lógica de programação, sendo neste projeto, usado 4 linguagens diferentes (Ladder, Function Block, Structured Text e Sequential Function Chart) afim de otimizar ao máximo a lógica de processo, sendo que, cada tipo de linguagem, apresenta um certo conforto ao projetista de acordo com a necessidade.
2. RSEmulate 5000: necessário para “criar um CLP virtual” que irá se comunicar com o programa escrito no Studio 5000. Assim é possível simular, virtualmente, o processamento de um CLP utilizado o próprio sistema operacional do computador.
3. FactoryTalk View: utilizado para criar as telas, gráficos, alarmes e todo o log de eventos do sistema supervisório. Neste, são divididos em duas partes, sendo: FactoryTalk View Studio, onde são criadas as telas e feitas todas as configurações e

Factory Talk Client, onde todas as configurações são executadas após ser gerado um arquivo *Client* no software anterior.

4. RSLinx: responsável por gerenciar os drivers de comunicação (neste caso comunicação entre o computador e um CLP virtual).

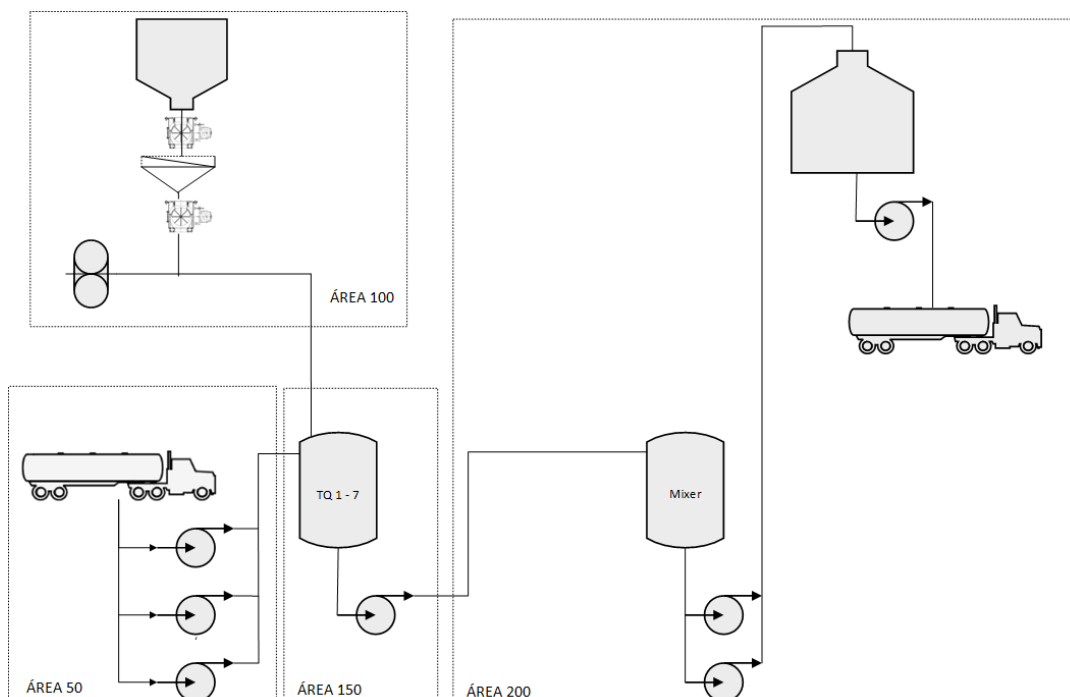
A automação do processo foi dividida em 4 áreas, nomeadas como área 50, 100, 150 e 200 respectivamente.

Trata-se de um processo de mistura de fluídos dosados com aditivos sólidos, os quais são muito comuns em processos onde a levedura de cana de açúcar é matéria-prima, por exemplo. Cada área, será designada para executar cada parte do processo, sendo:

- Área 50: Carregamento de matéria-prima para os tanques de processo;
- Área 100: Sistema de dosagem de aditivos sólidos através de transporte pneumático;
- Área 150: Composta pelos tanques de processo, onde a matéria-prima é aditivada e processada mediante à controle de temperatura;
- Área 200: Designada à mistura e bombeamento dos produtos aditivados nos tanques da área 150, para os tanques de armazenamento.

O processo pode ser melhor entendido de acordo com o fluxograma de processos ilustrado na Figura 5.

Figura 5 - Fluxograma de processo



Fonte: o autor

2.2 Interlocks e Permissivos

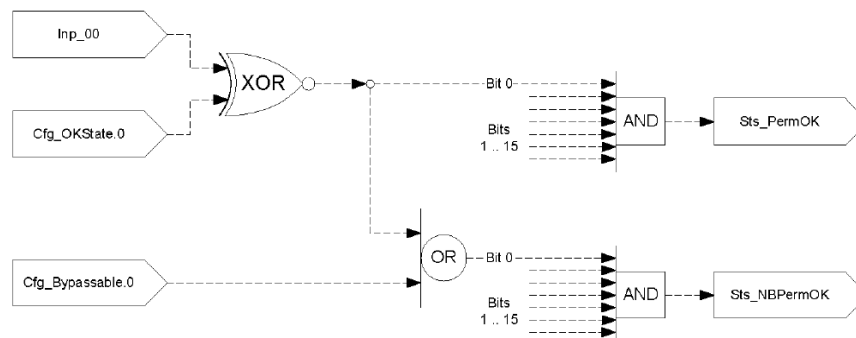
Uma grande vantagem ao usar essa biblioteca de processos é: poder atribuir outros blocos *add-on* ao operacional do dispositivo em questão. Dois deles são os blocos de Interlocks e Permissivos.

Segundo (Rockwell Automation, Inc., 2017), as instruções Perm (Permissives with Bypass) são usadas para coletar as informações que permitem que o equipamento seja energizado. Ainda segundo o autor, as condições permissivas são ignoradas quando o equipamento está em funcionamento.

Ou seja, os permissivos impedem, somente, a energização do dispositivo somente em seu estado inicial. Para fins operacionais e de manutenção, há a possibilidade de ativar o *Bypass*, ignorar as condições permissivas e iniciar o equipamento em modo manual.

A Figura 6 ilustra o diagrama lógico de funcionamento dessa instrução.

Figura 6 - Diagrama funcional da instrução P_perm

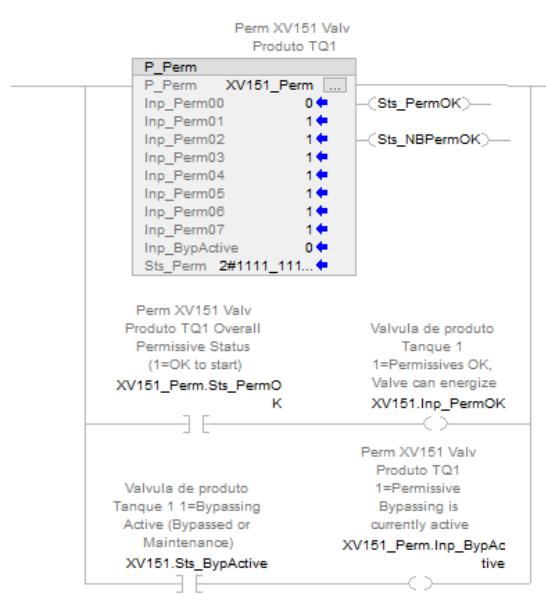


Fonte: Rockwell Automation, Inc. (2017)

Pode-se exemplificar, através da

Figura 7 o bloco lógico P_Perm em linguagem de programação ladder. Nesse caso, é necessário atribuir os sinais de entrada e saída do bloco permissivo e do bloco do dispositivo, informando através do estado lógico de [Tag_Perm]_Perm.Sts_PermOk para a entrada [Tag_device].Inp_PermOK. Caso ocorra algum evento não passível de permissão, o bloco do dispositivo entrará em estado inoperante ao iniciar novamente. Também é necessário linkar os bits de status de *Bypass* e do bloco do dispositivo e do bloco permissivo.

Figura 7 - Exemplo de bloco permissivo



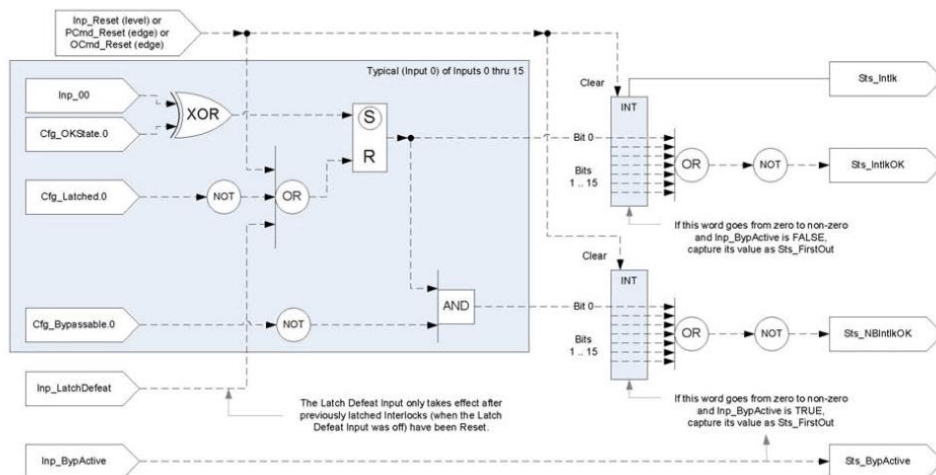
Fonte: O autor

Ao contrário da instrução permissiva, o bloco P_Interlock, avalia as entradas, e, caso haja alguma condição de contingência, o dispositivo desligará, imediatamente mesmo que esteja recebendo um comando para ligar.

O diagrama de funcionamento do bloco Interlock pode ser ilustrado na Figura 8.

Antes do desenvolvimento da lógica, foram formuladas as tabelas de intertravamentos e permissivos de cada dispositivo atuante no processo, sendo: válvulas e motores.

Figura 8 - Diagrama funcional da instrução P_Intlk



Fonte: Rockwell Automation, Inc., (2017)

A tabela 1 mostra um exemplo da organização feita para as condições permissivas para o acionamento de uma bomba.

Tabela 1 - Organização de permissivos

TABELA DE CONDIÇÕES PERMISSIVAS ÁREA 50			
EQUIPAMENTO (TAG)	DESCRIÇÃO	CONDIÇÕES	ESTADO DE OPERAÇÃO
M051	BOMBA DE ENVIO DE PRODUTO 01	Válvula de recebimento aberta	1
		Nível do tanque inferior à 80%	1
		Tanque disponível	1
M052	BOMBA DE ENVIO DE PRODUTO 02	Válvula de recebimento aberta	1
		Nível do tanque inferior à 80%	1
		Tanque disponível	1
M053	BOMBA DE ENVIO DE PRODUTO 03	Válvula de recebimento aberta	1
		Nível do tanque inferior à 80%	1
		Tanque disponível	1

Fonte: O autor

E assim, sucessivamente foi aplicada essa lógica para cada dispositivo no processo, analisando condições de contingência e erros operacionais. A Tabela 2 exemplifica uma tabela de interlocks para o acionamento de uma bomba.

Tabela 2 - Organização de interlocks

TABELA DE CONDIÇÕES PERMISSIVAS ÁREA 50			
EQUIPAMENTO (TAG)	DESCRIÇÃO	CONDIÇÕES	ESTADO DE OPERAÇÃO
M051	BOMBA DE ENVIO DE PRODUTO 01	Seleção de bomba para envio	0
		Status de válvula de recebimento	0
		Vibração do motor	0
		Proteção do motor	0
		Pressão de entrada da bomba	0
		Nível alto TQ selecionado	0

Fonte: O autor

Ao desenvolver a lógica, os sinais dos sensores, transmissores são manipulados e usados como entradas nestes blocos. Caso o sinal seja diferente do estado normal de operação,

pode ocorrer a parada imediata do equipamento (caso interlock) ou a inibição da sua partida (caso permissivo).

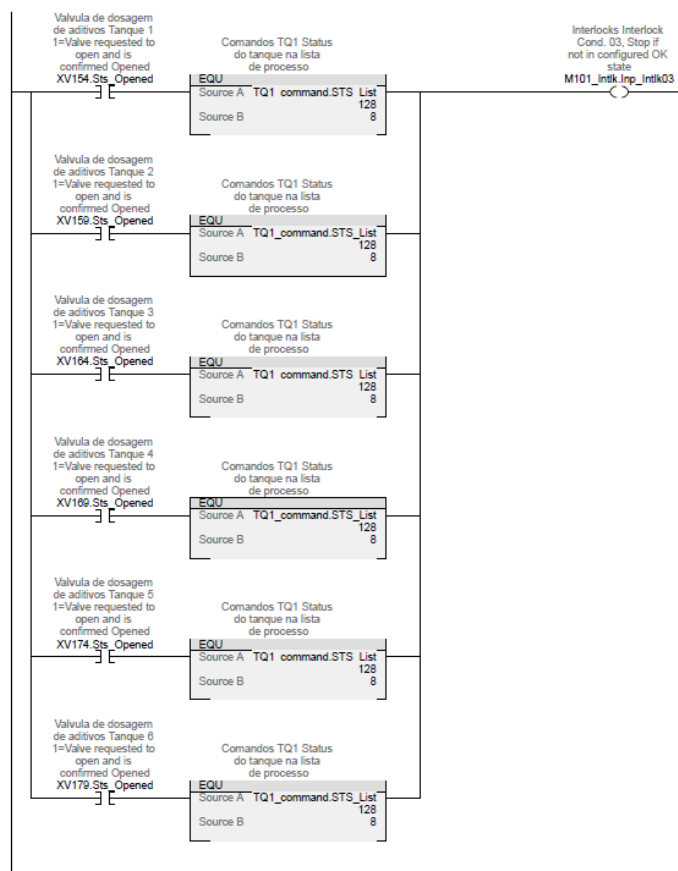
2.3 Desenvolvimento da lógica do programa

Já consideradas as condições de contingência, pode-se elaborar a lógica sem maiores preocupações, caso ocorram, devido ao fato de estas terem prioridade em relação ao comando de partida ou parada do dispositivo

Assim, avaliando qual tipo de linguagem de programação seria mais aplicável de acordo com a necessidade da lógica, foram escritas as rotinas para cada área:

- Linguagem Ladder, foram definidas as configurações do programa, como intertravamentos, contadores e temporizadores. A Figura 9 ilustra um trecho do código em Ladder. É vantajoso utilizar essa linguagem quando há necessidade de execução de tarefas simultâneas.

Figura 9 - Lógica para condição de acionamento do soprador de aditivos



Fonte: O autor

- Em linguagem FDB, foram parametrizados os blocos PID para controle de abertura das válvulas de vapor. Neste projeto, foi adotada essa linguagem pois o bloco fornece o recurso de auto sintonia do controlador com o processo, calculando automaticamente os ganhos proporcional, integral e derivativo.
- Em texto estruturado, foi definida a ordem de acionamento do sistema de dosagem, caso haja solicitações múltiplas dos tanques. A escolha dessa linguagem é devido à ordem da execução das instruções, onde, é feita linha por linha até o final do código. Assim, caso ocorra alguma solicitação simultânea, não há chances da linha alimentar dois tanques distintos, pois, a execução segue a ordem da escrita. A Figura 10 ilustra uma parte do código em linguagem estruturada.

Figura 10 - Rotina de prioridade da linha de dosagem

```

1 if M101.Sts_Stopped then
2
3   if STEP_BITS_PRIORIDADE.0 and M101.Sts_Stopped then      //avalia se tq1 solicita dosagem
4     // se sim, habilita prioridade para tq 1
5     // e permaneça assim até que a dosagem acabe
6     STEP_BITS_PRIORIDADE.6:=1;
7     STEP_BITS_PRIORIDADE.7:=0;
8     STEP_BITS_PRIORIDADE.8:=0;
9     STEP_BITS_PRIORIDADE.9:=0;
10    STEP_BITS_PRIORIDADE.10:=0;
11    STEP_BITS_PRIORIDADE.11:=0;
12  end_if;
13
14
15  if STEP_BITS_PRIORIDADE.1 and M101.Sts_Stopped then      //avalia se tq2 solicita dosagem
16    // se sim, habilita prioridade para tq 2
17    // e permaneça assim até que a dosagem acabe
18    STEP_BITS_PRIORIDADE.6:=0;
19    STEP_BITS_PRIORIDADE.7:=1;
20    STEP_BITS_PRIORIDADE.8:=0;
21    STEP_BITS_PRIORIDADE.9:=0;
22    STEP_BITS_PRIORIDADE.10:=0;
23    STEP_BITS_PRIORIDADE.11:=0;
24  end_if;
25
26  if STEP_BITS_PRIORIDADE.2 and M101.Sts_Stopped then      //avalia se tq3 solicita dosagem
27    // e permaneça assim até que a dosagem acabe
28    STEP_BITS_PRIORIDADE.6:=0;
29    STEP_BITS_PRIORIDADE.7:=0;
30    STEP_BITS_PRIORIDADE.8:=1;
31    STEP_BITS_PRIORIDADE.9:=0;
32    STEP_BITS_PRIORIDADE.10:=0;
33    STEP_BITS_PRIORIDADE.11:=0;
34  end_if;
35
--

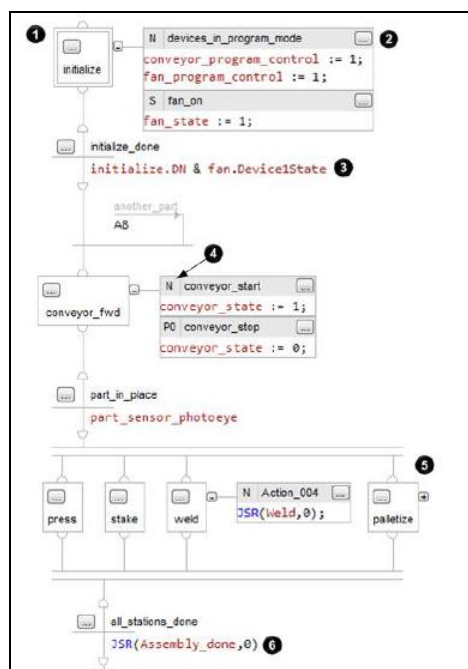
```

Fonte: O autor

Em linguagem SFC, foram programados os passos de cada tanque, respeitando cada condição para seguimento do programa. Há uma grande vantagem em utilizar essa linguagem quando há a necessidade de seguir etapas definidas no processo, onde, cada etapa executa uma ação específica atribuída a si. A Figura 11 ilustra a uma lógica em SFC, onde:

1. Representa a etapa do processo em questão;
2. As ações atribuídas a cada etapa;
3. As transações que dão sequência no processo quando a lógica é verdadeira;
4. Índice que determina quando as ações iniciam ou param;
5. Transação com caminhos diversos, podendo ser em configuração de ações paralelas ou com lógica por um único caminho;
6. Instrução JSR para chamar um sub rotina.

Figura 11 - Lógica em SFC



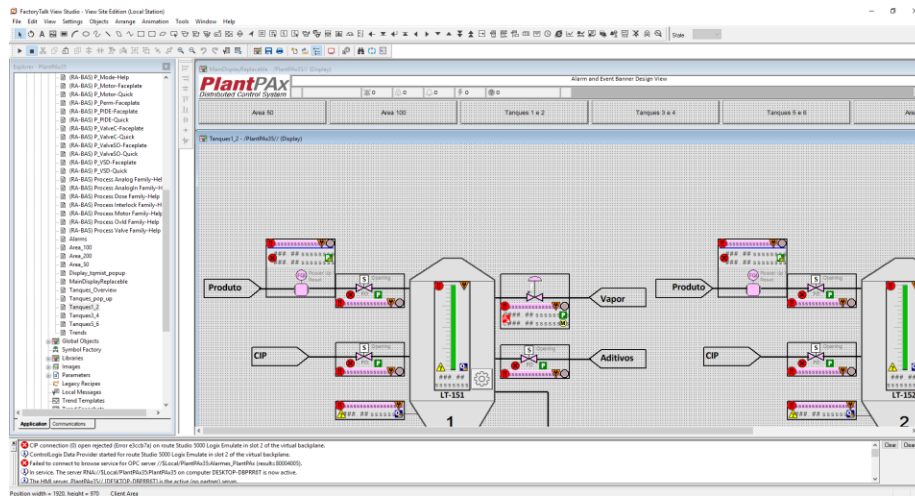
Fonte: (Rockwell Automation, Inc., 2017)

2.4 Desenvolvimento das telas do supervisor

O sistema supervisor foi pensado para ser simples, de fácil operação, intuitivo e com menor margem possível de erros operacionais. A Figura 12 ilustra o desenvolvimento de uma das telas do supervisor.

Para que haja uma operação otimizada, foi desenvolvido um menu de navegação e fixado na extremidade superior do monitor, para que possa ser chamadas outras telas através dos botões de navegação.

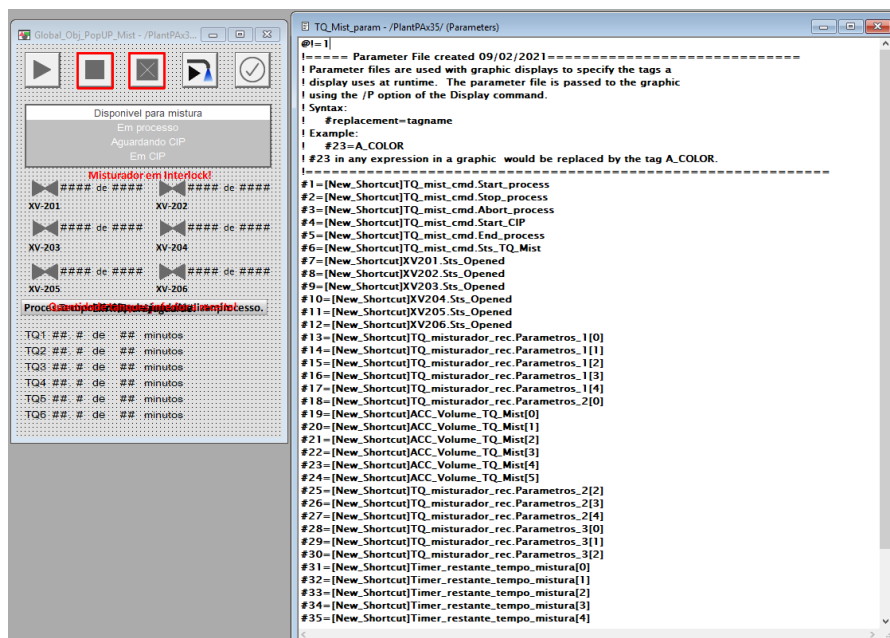
Figura 12 - Desenvolvimento de telas do sistema supervisório



Fonte: O autor

Para facilitar na elaboração das telas, foram feitas listas de parâmetros, os quais são atribuídos ao índice configurado no objeto global, conforme ilustra a Figura 13. Da mesma maneira foram replicadas as mesmas tags das pop-ups dos demais tanques de processo. Quando pressionado o botão de configuração, serão carregados os parâmetros atribuídos na ação do botão e a tela aparecerá com as variáveis carregadas.

Figura 13 - Desenvolvimento de pop-ups e parâmetros



Fonte: O autor

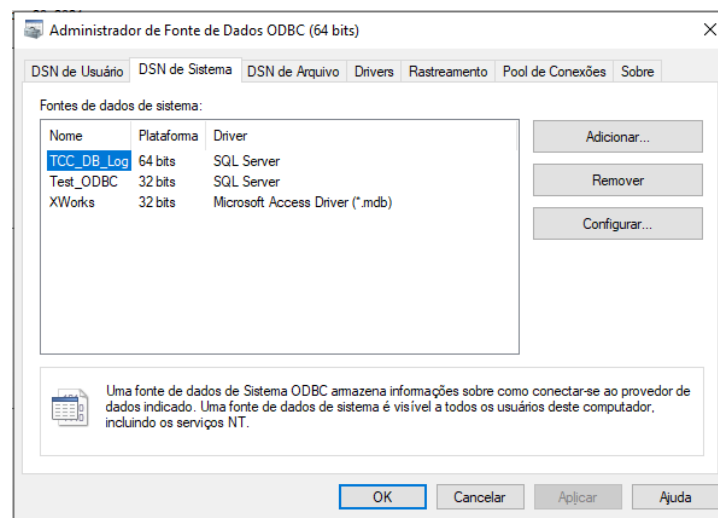
2.5 Configuração do historiador local e gráficos de tendência

É de suma importância que as variáveis críticas de processo sejam historiadas e armazenadas em um servidor, tanto quanto se diz respeito à produção em si ou quanto à sua segurança. Cita-se o exemplo de processos de produção de energia térmica, onde, há diversos riscos de acidentes operacionais, e, geralmente, variáveis como pressão, temperatura e até mesmo rotação de motores que controlam o fluxo de ar com o intuito de controlar a combustão são armazenados em um servidor. Nesse projeto, utiliza-se o protocolo ODBC (Open Database Connectivity) para se comunicar com o sistema supervisório.

Primeiramente, foi criado uma nova fonte de dados para alimentar o servidor SQL, conforme ilustra

Figura 14. Posteriormente, foi criado um novo log de eventos no FactoryTalk, já atribuindo as variáveis que serão armazenadas no banco de dados, que no caso refere-se ao nível do tanque de processo e tanque de mistura, conforme mostra a Figura 15.

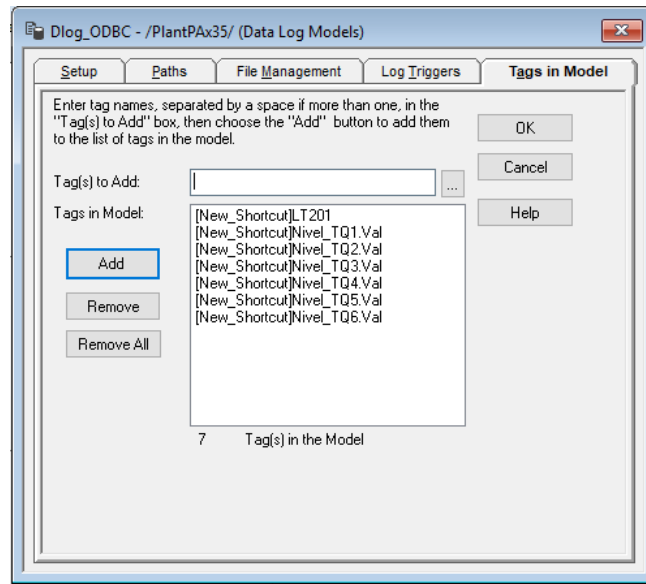
Figura 14 - Criação da fonte de dados ODBC



Fonte: O autor

A vantagem de se armazenar dados em um servidor SQL é que outras plataformas podem acessar esse banco e “consumir” essas informações, tais como: sistemas ERP e MES.

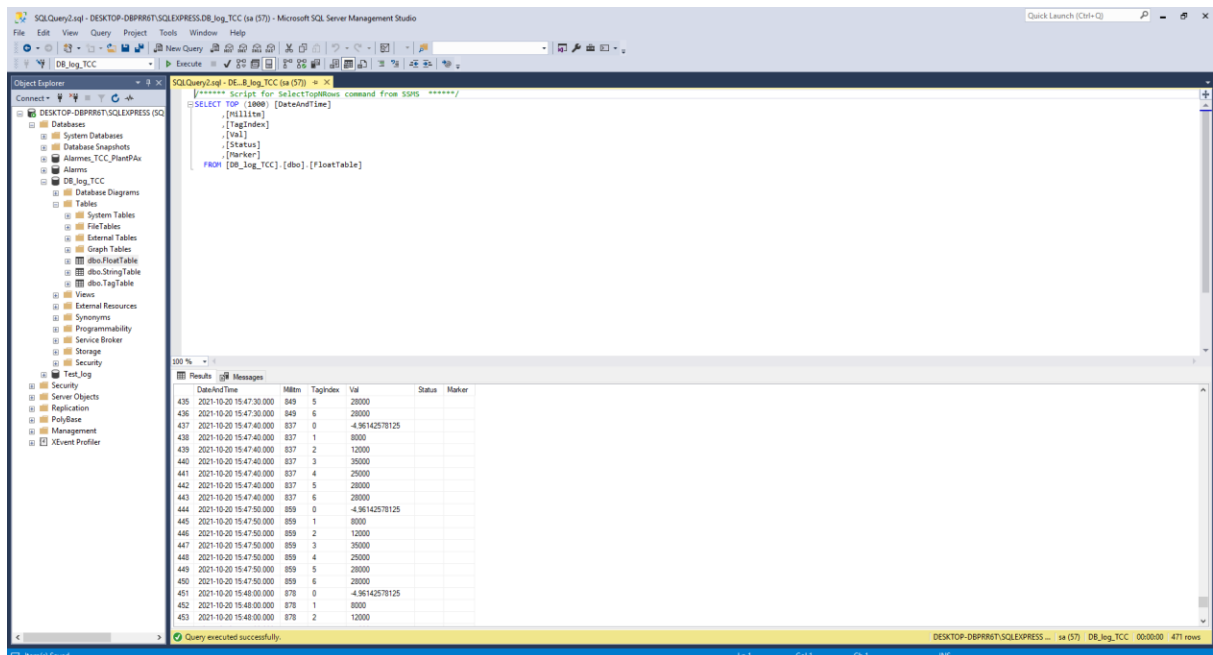
Figura 15 - Atribuição de tags para o historiador local



Fonte: O autor

Ao fazer tais configurações, pode-se checar no gerenciador do banco de dados, as variáveis e seus respectivos valores, salvos automaticamente no decorrer do tempo. No caso, o gerenciador utilizado foi o SQL Management Studio da Microsoft e pode-se verificar na Figura 16 o respectivo banco de dados.

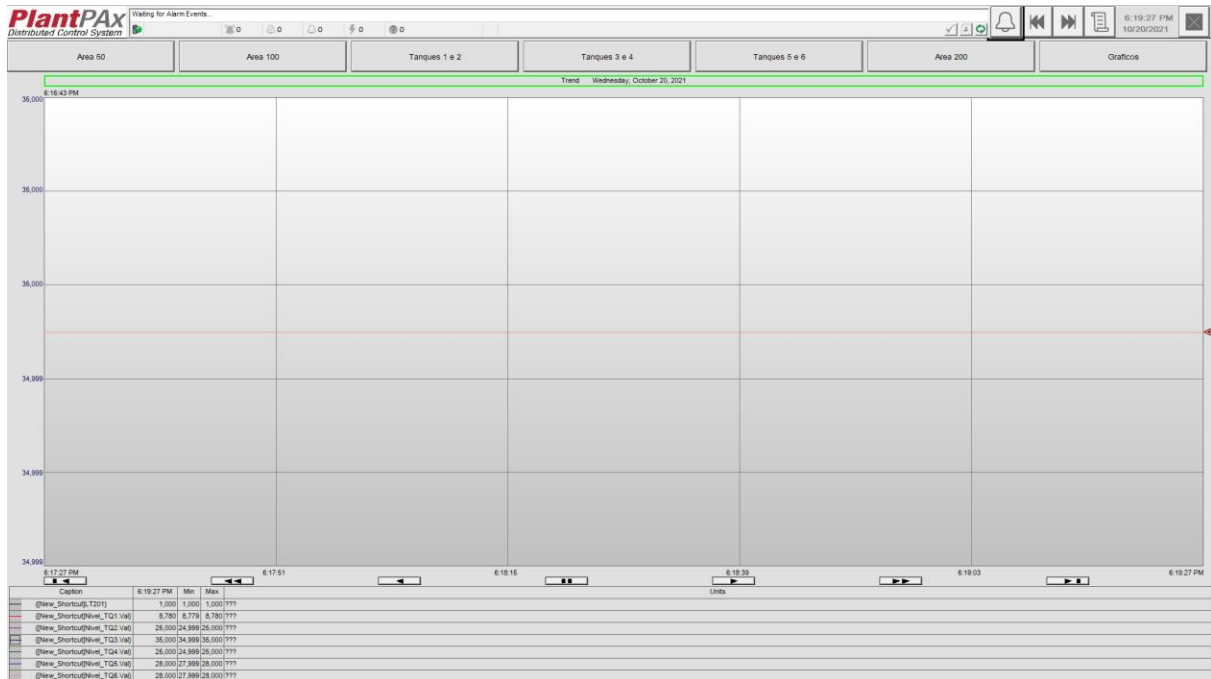
Figura 16 - Gerenciador do banco de dados



Fonte: O autor

Tendo a base de dados configurada, cria-se uma tela para gráficos de tendências no arquivo do sistema supervisorio, conforme ilustra Figura 17.

Figura 17 - Gráfico de tendências

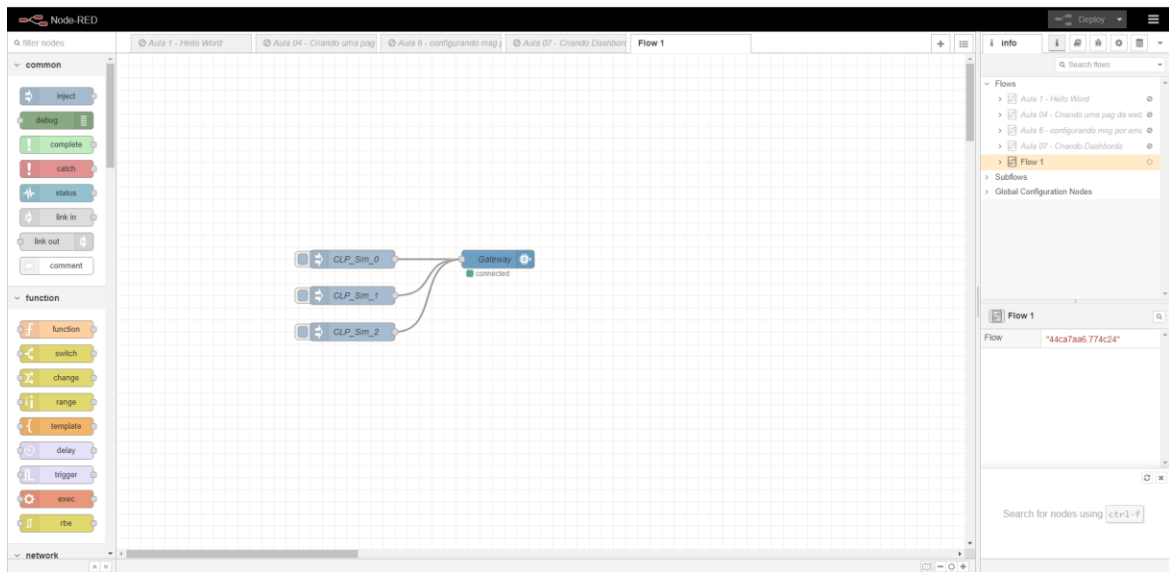


Fonte: O autor

2.6 Interface Node-RED e comunicação com o servidor na nuvem e arquivo client

O Node-RED, é uma plataforma de programação visual e código livre, que, foi desenvolvida com o intuito de implementar e conectar dispositivo IoT. Compreende-se em um editor que é executado no próprio browser do computador, onde as programações e configurações dos fluxos de informações, ações, entradas, saídas e *dashboards* são feitos através de nós pré-configurados. A Figura 18 mostra a configuração usada para simular os dados enviados. Caso usado um CLP real, o envio de dados não seria simulado, pois, as variáveis reais do processo estariam em comunicação direta com a nuvem, mas, como a utilização é através de um CLP simulado, não é possível comunicar devido à ausência de um endereço IP.

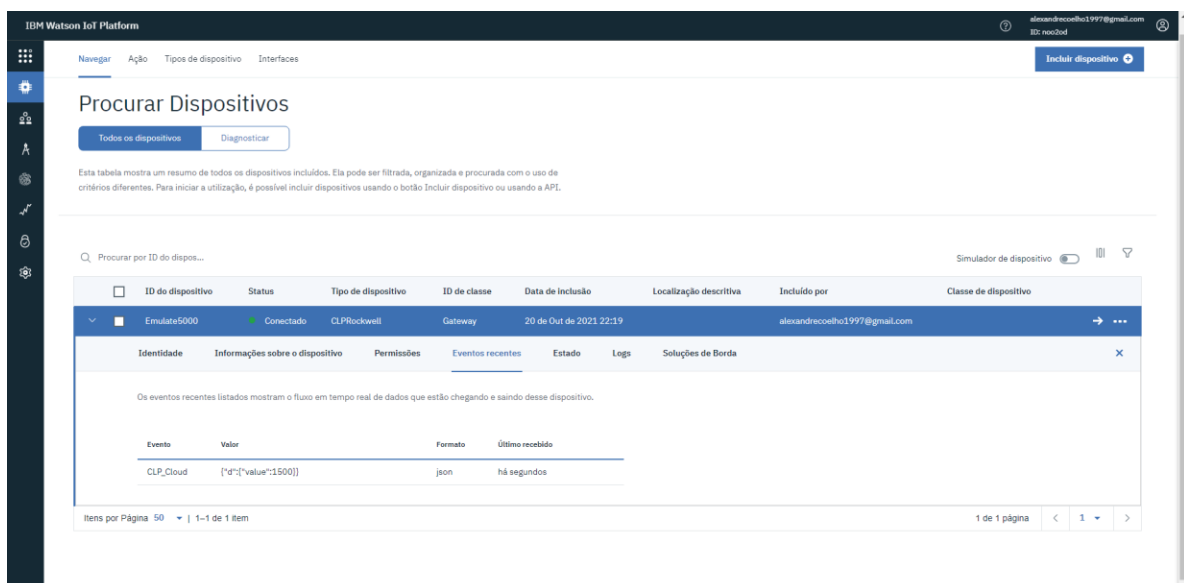
Figura 18 - Configuração do Node-RED



Fonte: O autor

Com a interface do Node-RED configurada, tem-se que criar um novo aplicativo na plataforma IoT da IBM, a IBM Watson e autenticar o nó com o app criado através de um domínio e um token gerado. Ao realizar esse passo, foi testada a comunicação, injetando valores no nó e checando o valor recebido, juntamente com o status do aplicativo, conforme mostra a Figura 19 Com esses valores, é possível criar gráficos, visualizar valores instantâneos do processo, de qualquer local, a qualquer momento com qualquer dispositivo que consiga rodar um browser.

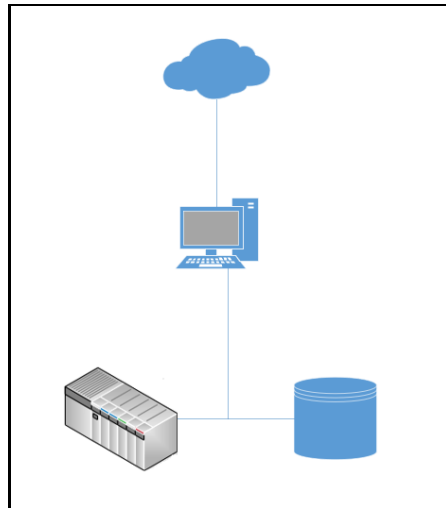
Figura 19 - Configuração da comunicação IBM Watson e Node-RED



Fonte: O autor

Levando em conta os estudos, configurações, criações de lógicas de programações e telas apresentados, foi possível gerar um sistema de controle do processo, trabalhando em conjunto com um sistema supervisorio armazenando dados em um servidor local e podendo ser, através da mesma estação de operação, fazer a interligação com a plataforma IoT e Node-RED, obtendo a arquitetura de rede mostrada na Figura 20.

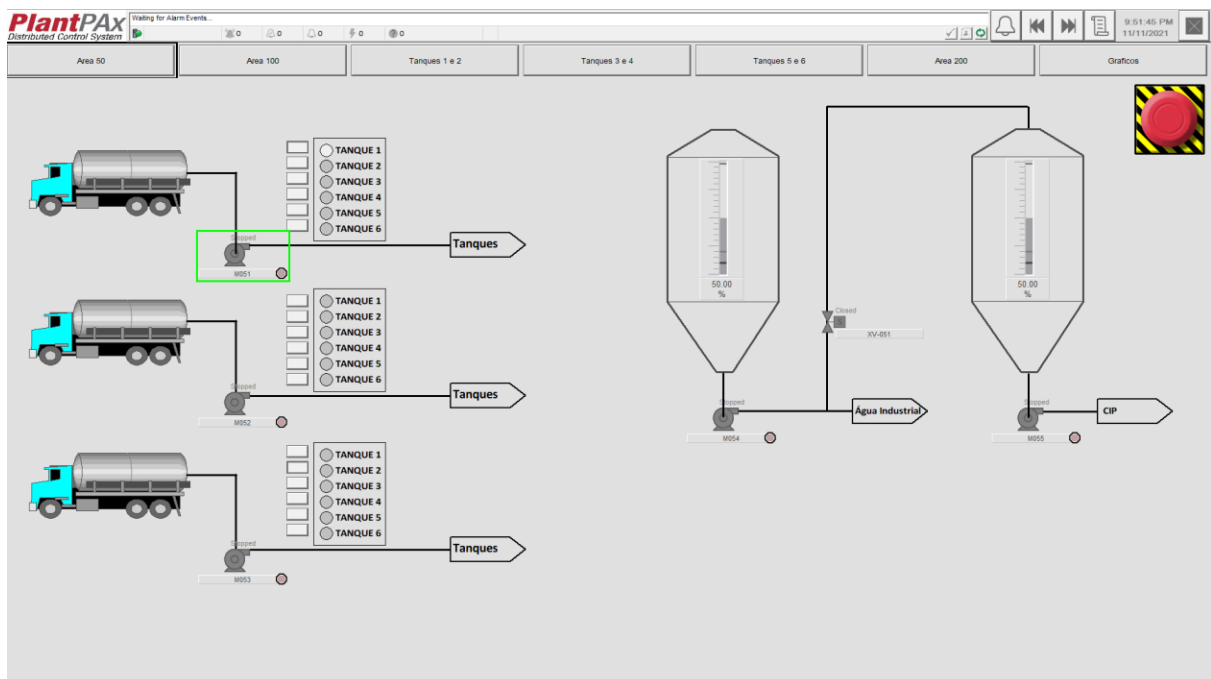
Figura 20 - Arquitetura da rede da aplicação



Fonte: O autor

A Figura 21 ilustra a tela de processos da área 50 no arquivo Client.

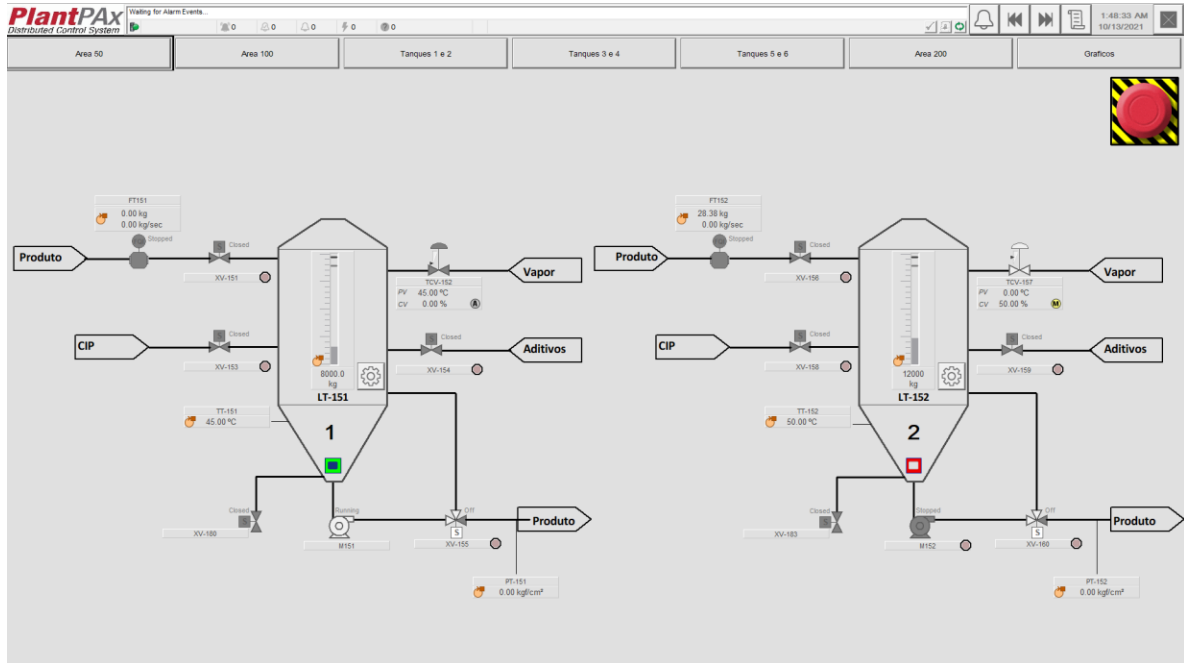
Figura 21 - Tela do arquivo *Client* para área 50



Fonte: O autor

A Figura 22 ilustra o processo dos tanques de preparação de produto.

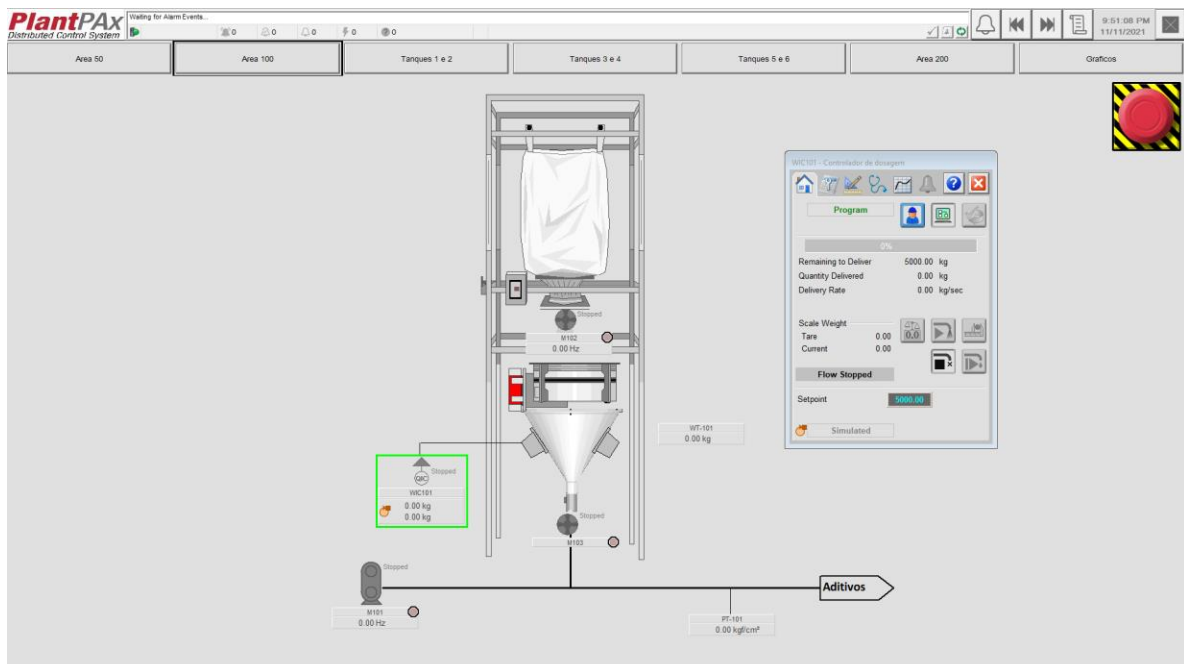
Figura 22 - Tela do arquivo Client para área 150



Fonte: O autor

A Figura 23 mostra o processo de dosagem de aditivos

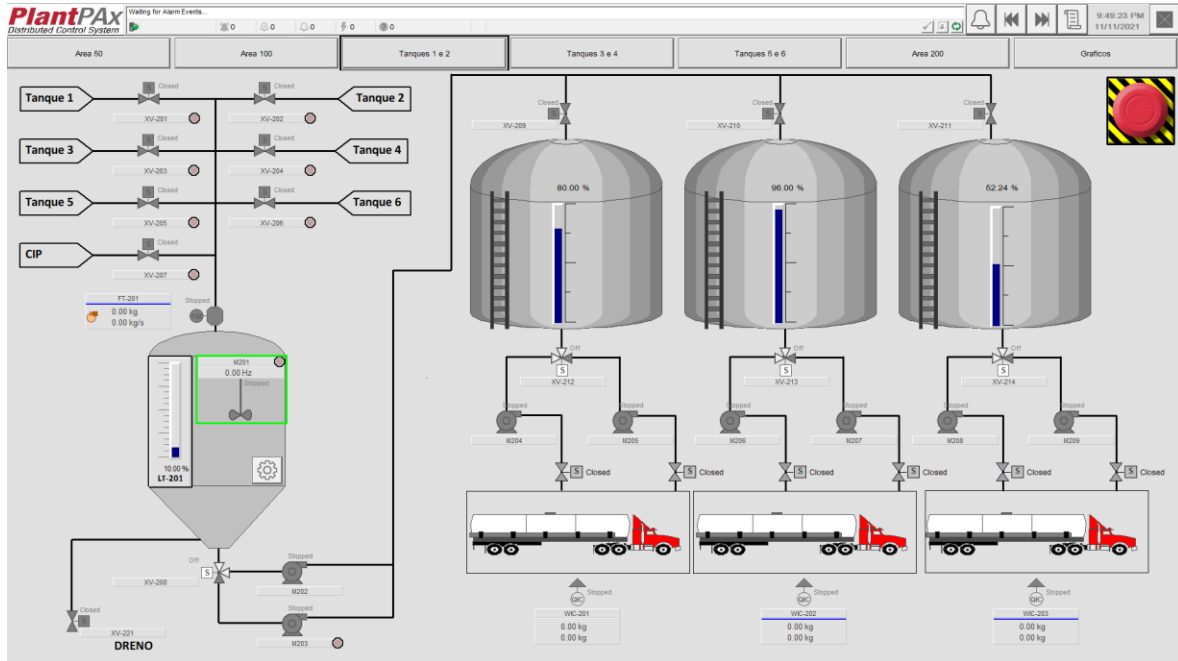
Figura 23 - Tela do arquivo Client para área 100



Fonte: O autor

A Figura 24 ilustra o processo de mistura, armazenagem e carregamento de produto acabado.

Figura 24 - Tela do arquivo Client para área 200



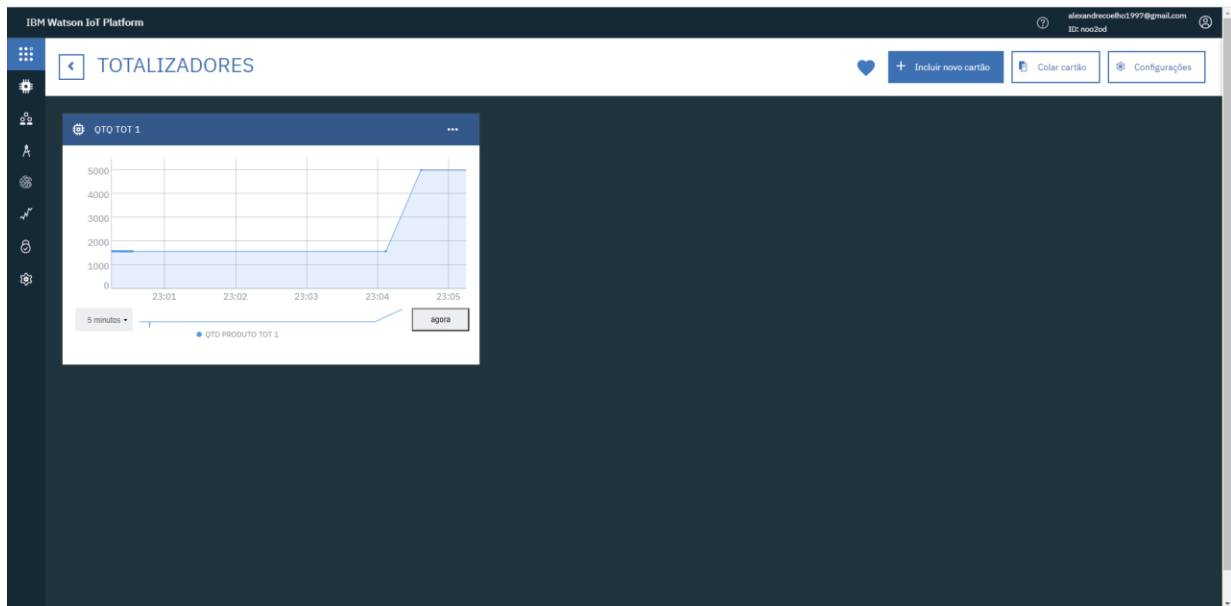
Fonte: O autor

Obteve-se o resultado da visualização dos dados de processo, ao configurar o aplicativo da IBM ilustrado pela Figura 25. Neste exemplo, usou-se somente um gráfico indicando o nível de produto em um dos tanques de processo. Porém qualquer informação de interesse gerencial, que esteja dentro da rede de dados do CLP, pode ser migrada à essa plataforma, como: quantia de matéria prima utilizada, quantia de produto acabado, taxa instantânea de produção, entre outros.

É exatamente isso que une os conceitos de sistemas ciber-físicos e internet das coisas. Uma comunicação entre dispositivos fabris e compartilhamento de informações para outros dispositivos de maneira autônoma, se trata de sistemas ciber-físicos.

A migração para uma rede e o uso da internet para o compartilhamento dessas informações para uma plataforma, se diz à respeito da IoT.

Com essa tecnologia, busca-se um melhor apoio gerencial, pois, as informações de interesse industriais podem ser facilmente visualizadas de qualquer lugar, bastando, somente o acesso à uma rede internet e um navegador.

Figura 25 - Gráficos de processo na nuvem

Fonte: O autor

3 CONCLUSÃO

Este trabalho teve como objetivo simular um processo industrial, caracterizado por produção em batelada. Foi possível atingi-lo ao utilizar softwares de programação e ferramentas de simulação de CLPs industriais.

Foram utilizados como base para simulação, processos encontrados em empresas que trabalham com aditivos para rações animais, porém, podem ser replicados à outros processos similares.

Para proposta de avanço desse projeto, pode-se propor a implementação prática em um CLP real para controlar o processo. Também, pode-se incrementar conceitos de segurança cibernética, pois, há tráfego de dados externamente à rede local.

REFERÊNCIAS

- Drath, R., & Horch, A. (2014). Industrie 4.0: Hit or Hype? *IEEE Industrial Electronics Magazine*, 56-58.
- Firjan. (2016). *Indústria 4.0*. Rio de Janeiro: Federação das Indústrias do Estado do Rio de Janeiro.
- Heiner, L., Hans-Georg, K., Fettke, P., Feld, T., & Hoffmann, M. (2014). Industry 4.0. *Business & Information Systems Engineering*, 239-242.
- Pereira, A., & Simonetto, E. d. (2018). INDÚSTRIA 4.0: CONCEITOS E PERSPECTIVAS PARA O BRASIL. *Revista da Universidade Vale do Rio Verde*, 1-9.
- Rockwell Automation, Inc. (2017). *Logix 5000 Controllers Sequential Function Charts*.
- Rockwell Automation, Inc. (2017). *Rockwell Automation Library of Process Objects Version 3.5*.
- Tabaa, M., Chouri, B., Saadaoui, S., & Alami, K. (2018). Industrial Communication based on Modbus and Node-RED. *Procedia Computer Science* 130, 583-588.