

**UNIVERSIDADE DO ESTADO DE MINAS GERAIS
INSTITUTO SUPERIOR DE ENSINO E PESQUISA DE ITUIUTABA
CURSO DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO**

DANILO HUMBERTO DIAS SANTOS

**UTILIZANDO O SOFTWARE FACTORY TALK PARA
SIMULAR UM PROCESSO INDUSTRIAL**

**ITUIUTABA
2012**

DANILO HUMBERTO DIAS SANTOS

**UTILIZANDO O SOFTWARE FACTORY TALK PARA
SIMULAR UM PROCESSO INDUSTRIAL**

Trabalho de Conclusão de Curso apresentado à Fundação Educacional de Ituiutaba unidade associada à Universidade do Estado de Minas Gerais como pré-requisito para obtenção parcial de créditos em Bacharel em Engenharia de Computação.

Orientador: Prof. Me. Walteno Martins Parreira Júnior

**ITUIUTABA
2012**

DANILO HUMBERTO DIAS SANTOS

**UTILIZANDO O SOFTWARE FACTORY TALK PARA
SIMULAR UM PROCESSO INDUSTRIAL**

Trabalho de conclusão de curso apresentado à
Fundação Educacional de Ituiutaba unidade
associada à Universidade do Estado de Minas
Gerais como pré-requisito para obtenção parcial de
créditos em Projeto de Conclusão sob orientação
do Prof. Me. Walteno Martins Parreira Júnior.

Ituiutaba, ?? de Novembro de 2012

Banca Examinadora

:

Prof. Me. Walteno Martins Parreira Júnior (Orientador)

Prof.

Prof.

**ITUIUTABA
2012**

AGRADECIMENTOS

A Deus.

Aos meus pais, José Humberto e Francisca Iraudete, que sempre acreditaram e investiram para que fosse possível concluir minha jornada na universidade.

Aos meus amigos e companheiros de sala de aula pelos momentos compartilhados durante esses anos em que estivemos estudando juntos e buscando um mesmo interesse de crescer profissionalmente.

A minha namorada Liliane, pela motivação e por me apoiar a concluir o trabalho neste segundo semestre.

Aos meus professores pelos ensinamentos e conselhos passados em sala de aula, colaborando para minha formação.

Ao meu orientador Walteno pela disposição e paciência e demais pessoas que contribuíram de alguma forma para o desenvolvimento deste projeto.

“Não ergas alto um edifício sem fortes alicerces, se o fizeres viverás com medo!”

Sabedoria Persa

RESUMO

O avanço de automação está ligado, em grande parte, ao avanço da microeletrônica que se deu nos últimos anos. Os CLP's surgiram na década de 60 e substituíram os painéis de cabina, os quais possuíam centenas de reles e ocupavam muito espaço físico e no momento que se fosse fazer uma alteração na programação lógica do processo que era realizado por interconexões elétricas com lógica fixa. A evolução dos CLP's foi dividida em 5 gerações cada uma com um notável avanço ou funcionalidade específica. As indústrias utilizam este recurso a muito tempo, e sistemas cada vez mais modernos estão tomando conta dos campos industriais, sistemas SCADA de supervisão e monitoramento de processos são necessários para um bom funcionamento industrial. Com o avanço da informática, softwares cada vez mais robustos estão surgindo no mercado, é o caso do *FactoryTalk* que permite fazer uma simulação de algum processo industrial com monitoramento, controle e implementação de regras existentes nas indústrias reais. É com base neste software que este projeto foi montado, sendo possível visualizar um processo industrial na sua máquina e realizar o supervisório deste.

Palavras-chave: *Factory Talk*; Simulação; Indústria; Sistemas Supervisórios.

ABSTRACT

The advancement of automation is part of the advancement of electronics has happened in recent years. Programmable Controllers emerged in the 60's and replaced the large control panels that used hundreds electronics occupying so much space and when they were carrying out any change in logic programming process generated much difficulty. The evolution of CLPs was divided into five generations each with a notable advance or specific functionality. The industries use this feature a long time, and modern systems increasingly are taking care of the fields industrial SCADA supervisory and monitoring processes are necessary for a functioning industrial. With the advancement of computers, software are increasingly emerging best, is the case of FactoryTalk that allows a simulation of some industrial process monitoring, control and implementation of existing rules in real industries. It is based on the FactoryTalk software that this project was developed, where you can view an industrial process perfectly.

Key words: Factory Talk; Simulation, Industry, Supervisory Systems.

LISTA DE FIGURAS

Figura 1 – Sistema de supervisão e controle.....	26
Figura 2-Comunicação CLP / PC	32
Figura 3 - Driver de comunicação 1784-PCIC(S) for ControlNet. Fonte: Programa RSLinx	33
Figura 4- Programa Ladder que pode ser feito Upload ou Download para o CLP	33
Figura 5-Rack contendo os programas Ladder. Fonte: Programa RSLogix Chassis Monitor.....	34
Figura 6-Posição PROG do processador. Fonte: Programa RSLogix 5000.....	34
Figura 7- Posição REMOTE do processador. Fonte: Programa RSLogix 5000	35
Figura 8-Posição RUN do processador. Fonte: Programa RSLogix 5000	35
Figura 9-UPLOAD do código Ladder para o controlador.....	35
Figura 10-Aplicação em rede do Factory Talk View SE. Fonte: RSTECHEd (2010).....	37
Figura 11-Arquitetura de um sistema em rede Factory Talk View SE. Fonte: RSTECHEd.....	38
Figura 12 – Botões disponíveis na biblioteca <i>FactoryTalk</i>	41
Figura 13 - Sensores disponíveis na biblioteca <i>Factory Talk</i>	41
Figura 14 - Tela de alarmes e eventos disponível na biblioteca <i>Factory Talk</i>	41
Figura 15 - Esteiras disponíveis na biblioteca <i>Factory Talk</i>	42
Figura 16 - Exemplo de <i>Tag</i> de referência direta. Fonte: RSTECHEd (2010)	42
Figura 17 - Exemplo de <i>Tag</i> de IHM. Fonte: RSTECHEd (2010).....	43
Figura 18 - Barra de navegação do Communication Setup Factory Talk	46
Figura 19 - Propriedades do atalho onde estará armazenada as configurações de comunicação Factory Talk / RSLogix 5000.....	46
Figura 20 - Caminho para o programa principal	46
Figura 21 - Slots ocupados no SoftLogix 5000.....	47
Figura 22- Processo de enchimento das garrafas. Factory Talk.....	48
Figura 23 - Animação das garrafas	48
Figura 24- Janela de configuração de animações. Factory Talk.	49

Figura 25- TAG "Motion" presente no controlador. Factory Talk.....	49
Figura 26- Colocando rótulos nas garrafas. Factory Talk.	50
Figura 27-Animação dos rótulos. Factory Talk.....	51
Figura 28- TAGS de animação presentes no controlador. RSLogix 5000.	51
Figura 29-Esteira de empacotamento. Factory Talk.	52
Figura 30-janela de configuração da animação. Factory Talk.....	52
Figura 31- TAGS presente no controlador. RSLogix5000.	53

LISTA DE TABELAS

Tabela 1 - Comparação entre Tags de IHM e referência direta. Fonte RSTECHEd (2010) 45

LISTA DE ABREVIATURAS E SIGLAS

CLP - *Controlador lógico Programável*

HVAC - *Heating, Ventilation and Air Conditioning*

IHM – *Inferface Homem Maquina*

PLC - *programmable logic controller*

SCADA - *Supervisory Control And Data Acquisition*

SUMÁRIO

1	INTRODUÇÃO.....	16
1.1	Cenário atual da simulação de processos	17
1.2	Identificação do problema	17
1.3	Objetivos	18
1.3.1	Objetivo geral	18
1.3.2	Objetivos específicos	18
1.4	Justificativa	18
1.5	Organização do trabalho	19
2	ESTADO DA ARTE	20
2.1	Origem da Automação Industrial.....	20
2.2	Surgimento do CLP (Controlador lógico Programável)	21
2.3	Modelagem e simulação	23
2.3.1	Simulação física	23
2.3.2	Simulação analógica	24
2.3.3	Simulação digital	24
2.3.4	Vantagens e desvantagens da simulação.....	24
2.4	Sistemas Supervisórios.....	25
2.5	Linguagem Ladder.....	27
2.6	Conclusão	27
3	REFERENCIAL TEÓRICO	28
3.1	Software Factory Talk	28
3.1.1	<i>Factory Talk View SE</i>	28
3.1.2	<i>Factory Talk View Studio</i>	29
3.1.3	<i>Factory Talk View SE Client</i>	29
3.1.4	<i>Factory Talk View Administration Console</i>	30
3.2	Software RSLogix 5000.....	30
3.3	VMware Workstation	31
3.4	Conclusão	31

4	MODELO DO PROCESSO	32
4.1	Softwares RSLogix 5000 e Factory Talk View SE	32
4.1.1	RSLink Classic	32
4.1.2	Configurando o RSLink	33
4.1.3	RSLogix 5000	33
4.1.4	RSLogix Chassis Monitor	34
4.1.5	Chaves do processador	34
4.1.6	Inserindo o programa Ladder no Controlador	35
4.1.7	Factory Talk – Aplicações em rede	36
4.1.8	Factory Talk – Áreas	36
4.1.9	Factory Talk – Servidores de IHM	38
4.1.10	Factory Talk – Servidores de dados	38
4.1.11	Factory Talk – Servidores de alarmes e eventos	39
4.1.12	Bibliotecas	40
4.1.13	Botões	40
4.1.14	Sensores	41
4.1.15	Alarmes e eventos	41
4.1.16	Esteiras	42
4.1.17	Tags	42
4.1.18	Configurando a Conexão do Factory Talk com o RSLogix 5000	45
4.2	Testes do processo de esteiras	47
4.2.1	Processo de enchimento das garrafas	47
4.2.2	Animação dos elementos	48
4.2.3	Processo de rotulação das garrafas	50
4.2.4	Processo de empacotamento	51
4.3	Conclusão	53
5	CONCLUSÃO	54
5.1	Trabalhos futuros	55
	REFERÊNCIAS	56

1 Introdução

O termo SCADA (*Supervisory Control And Data Acquisition*) na automação refere-se a sistemas de supervisão, controle e aquisição de dados composto por um ou mais computadores monitorando e controlando um processo. O processo pode ser industrial, infra-estrutura ou facilidade conforme descrito a seguir (SILVA, 2009):

Processos industriais incluem manufatura, geração de energia, refino de petróleo e muitos outros. Podem ser executados de forma contínua ou batelada. Os sinais tratados podem ser tanto analógicos quanto digitais (SILVA, 2009).

Processos de facilidade ocorrem em instalações públicas e privado, incluindo edifícios, aeroportos, navios, plataformas *offshore* e estações espaciais. Esses sistemas monitoram e controlam HVAC (*Heating, Ventilation and Air Conditioning*) e consumo de energia. 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. A utilização de sistemas SCADA permite uma série de vantagens se comparados com os painéis convencionais (SILVA, 2009):

Grandes revoluções na humanidade acontecem sem que o homem se de conta de sua importância. A maioria das ideias que revolucionaram épocas está ligada a mudança de rotinas e quebra de métodos de trabalho. Com a revolução da informação que segundo STAIR (1998) ocorreu no final da década de 90 provocou um crescente interesse pelas questões da informação, do conhecimento e das tecnologias da informação como uma via eficaz para resolução de muitos problemas operacionais cotidianos, e também como um instrumento estratégico de grande potencial.

Se for possível juntar essas operações industriais como simulações, irá proporcionar aos funcionários destas empresas um aprendizado melhor, antes de começarem as práticas "externas". Modelagem, de forma geral, é a elaboração de um modelo para a representação de alguma coisa. Modelo é a representação de um sistema real ou imaginário usando uma linguagem, um meio, e segundo um posto de vista.

1.1 Cenário atual da simulação de processos

Atualmente várias empresas estão utilizando as chamadas simulações, que tem como principal função simular tomadas de decisões, prevendo manobras e despesas a serem gastos na empresa. Segundo (STAIR 1998), para se adaptarem a um novo modelo organizacional, muitas empresas estão mudando as suas estratégias de negócios, tentando estruturar-se de tal forma a interagir, permanentemente, com este novo modelo.

As simulações não se limitam a um modismo, mas sim uma tendência secular que se acentua nos dias atuais como o avanço tecnológico (VICENTE, 2001) sendo uma alternativa didática para levar o estudante a desempenhar, em representações de situações reais que podem acontecer na indústria, vários papéis comparáveis aos do sistema real, privilegiando o aprender fazendo num ambiente simulado (RAMOS, 2001).

1.2 Identificação do problema

Segundo (COOPER E SCHINDLER, 2003), a definição do problema é uma das fases mais importantes em uma pesquisa e muitas vezes pode tornar-se seu ponto crítico: sua definição incorreta ou inadequada é uma das maiores causas dos fracassos de projetos.

Com isso, a finalidade é de capacitar uma pessoa a interagir com a área de supervisão de automação de uma empresa.

Sendo assim, o trabalho objetiva articular esta ferramenta de simulação com uma temática específica em automação.

1.3 Objetivos

1.3.1 Objetivo geral

Proporcionar o desenvolvimento de um cenário industrial para simulação utilizando o software *Factory talk* como ferramenta de apoio no processo de ensino identificando os conteúdos relevantes à Automação industrial e sistemas supervisórios. Permitir o aprimoramento do conhecimento no processo de sistemas supervisórios no ambiente de uma empresa.

1.3.2 Objetivos específicos

a) Analisar vantagens e desvantagens da aplicação de um software de simulação em uma área de automação de uma empresa.

b) Buscar melhor equilíbrio nas operações de monitoramento, através da simulação, sem afetar de forma real os negócios na parte de Supervisão, abrangendo diversas áreas como Softwares, Hardwares e comunicação.

1.4 Justificativa

A aplicação de um software para simulação de algum processo facilita a identificação de erros por parte daqueles que estão tomando decisões evitando assim, que os mesmos erros se repitam em uma situação real de um cenário acadêmico ou industrial.

A simulação pode desenvolver habilidades na análise e interpretação de decisões a serem tomadas quando de fato, acontecer um problema que precisa ser resolvido imediatamente no ambiente de supervisão de uma indústria.

1.5 Organização do trabalho

O projeto está organizado em cinco capítulos, a saber:

O capítulo 1 apresenta a introdução referente ao assunto, a identificação do problema, os objetivos e a justificativa do projeto.

O capítulo 2 compreende a fundamentação teórica utilizada para o desenvolvimento da pesquisa e é apoiada no referencial bibliográfico, onde são abordados os assuntos relacionados ao tema.

O capítulo 3 compreende as ferramentas que foram utilizadas para o desenvolvimento do projeto.

O capítulo 4 consiste nas telas gráficas, desenvolvimento e testes.

O capítulo 5 apresenta as considerações finais e a conclusão do projeto.

2 Estado da Arte

Neste capítulo são abordados os conceitos mais importantes sobre a evolução e o histórico da automação industrial, modelagem e simulação de sistemas supervisórios, como e onde surgiram.

2.1 Origem da Automação Industrial

A automação existe desde a pré-história, quando os humanos que habitavam a terra inventaram a roda para transportar materiais pesados, reduzindo assim, o esforço humano. Já no início da era industrial que realmente começou a despertar interesse, a história mostra que em 1784 o norte-americano Oliver Evans construiu um moinho de trigo automático, na Filadélfia nos Estados Unidos. Alguns anos mais tarde, em 1801, o francês Joseph Marie Jacquard construiu um tear que é um aparelho mecânico ou eletromecânico empregado para fins de tecelagem controlados por uma espécie de sistema de cartões perfurados (BOWEN, 1969).

Ainda segundo (BOWEN, 1969) esse sistema se tornou tão popular que a palavra “jacquard” passou a designar, na França, determinados padrões que podia ser tecidos ou tricotados com essa nova máquina. Cerca de mil novos teares foram vendidos somente na terra de Napoleão. O sistema revolucionário de cartões perfurados caiu no esquecimento até após a segunda Grande Guerra, quando se percebeu que podiam ser utilizados para fornecer informações aos grandes computadores eletrônicos. Provavelmente a máquina a vapor, inventada por James Watt, foi um dos dispositivos mais antigos de controle automático que utilizava a força centrífuga para regular a quantidade de vapor que penetrava no cilindro da máquina.

Podemos citar muitos outros exemplos dos primórdios da automação como: o descaroçador de algodão de Eli Whitney, a ceifadeira mecânica de Cyrus McCormick e o rifle de repetição fabricado e tornando-se famoso pela Companhia Winchester de Armas de repetição, de Massachusetts.

De acordo com (FERNANDINO, 1995) a pessoa que maior influência exerceu sobre a aplicação da automação a um processo industrial completo foi Henry Ford,

revolucionando completamente a indústria automobilística com a introdução da técnica de linha de montagem. Notou-se então, a necessidade de fazer algo para que fosse aumentada a produtividade das indústrias, iniciando-se assim o desenvolvimento de máquinas para executar as tarefas com maior precisão, rapidez e qualidade.

2.2 Surgimento do CLP (Controlador lógico Programável)

O avanço de automação está ligado, em grande parte, ao avanço da microeletrônica que se deu nos últimos anos. Os CLP's surgiram na década de 60 e substituíram os painéis de cabina, os quais possuíam centenas de reles e ocupavam muito espaço físico e no momento que se fosse fazer uma alteração na programação lógica do processo que era realizado por interconexões elétricas com lógica fixa, exigiam interrupções no processo produtivo a fim de se reconectar os elementos de controle do painel o que levava muito tempo ocasionando uma enorme perda de produção, além do alto consumo de energia (BOWEN, 1969).

Em 1968 na divisão hidráulica da empresa “*General Motors Corporation*”, realizou-se uma experiência de controle de lógica que permitisse a programação por recursos de software aliado ao uso de dispositivos periféricos, capazes de realizar operações de entrada e saída, um microcomputador com capacidade de programação o que reduziu os custos de automação da época (HENRIQUE, 2008).

Ainda segundo (HENRIQUE, 2008) com a criação de microprocessadores durante os anos 70, não precisou se utilizar mais os computadores de grande porte assim a PLC tornou-se uma unidade isolada, recebendo recursos como: Interface de programação e operação facilitadas ao usuário, instruções de aritmética e de manipulação de dados, recursos de comunicação por meio de redes PLC, possibilidades de configuração específica a cada finalidade por meio de módulos intercambiáveis.

Antonelli (1998) dividiu os CLP's historicamente de acordo com o sistema de programação por ele utilizado:

Primeira Geração: Os CLPs de primeira geração se caracterizam pela programação intimamente ligada ao hardware do equipamento. A linguagem utilizada era a *Assembly* que variava de acordo com o processador utilizado no projeto do CLP,

ou seja, para poder programar era necessário conhecer a eletrônica do projeto do CLP. Assim a tarefa de programação era desenvolvida por uma equipe técnica altamente qualificada, gravando-se o programa em memória EPROM, sendo realizada normalmente no laboratório junto com a construção do CLP.

Segunda Geração: Aparecem as primeiras “Linguagens de Programação” não tão dependentes do hardware do equipamento, possíveis pela inclusão de um “Programa Monitor” no CLP, o qual converte as instruções do programa, verifica o estado das entradas, compara com as instruções do programa do usuário e altera os estados das saídas. Os Terminais de Programação (ou Maletas, como eram conhecidas) eram na verdade Programadores de Memória EPROM. As memórias depois de programadas eram colocadas no CLP para que o programa do usuário fosse executado.

Terceira Geração: Os CLPs passam a ter uma Entrada de Programação, onde um Teclado ou Programador Portátil é conectado, podendo alterar, apagar, gravar o programa do usuário, além de realizar testes no equipamento e no programa. A estrutura física também sofre alterações sendo a tendência para os Sistemas Modulares com Bastidores.

Quarta Geração: Com a popularização e a diminuição dos preços dos microcomputadores, os CLPs passaram a incluir uma entrada para a comunicação serial. Com o auxílio dos microcomputadores, a tarefa de programação passou a ser realizada nestes. As vantagens eram a utilização de várias representações das linguagens, possibilidade de simulações e testes, treinamento e ajuda por parte do software de programação, possibilidade de armazenamento de vários programas no micro, etc.

Quinta Geração: Atualmente existe uma preocupação em padronizar protocolos de comunicação para os CLP's, de modo a proporcionar que o equipamento de um fabricante se comunique com o equipamento outro fabricante, não só CLP's, como Controladores de Processos, Sistemas Supervisórios, Redes Internas de comunicação e etc., proporcionando uma integração a fim de facilitar a automação, gerenciamento e desenvolvimento de plantas industriais mais flexíveis e normalizadas, fruto da chamada Globalização. Existe uma fundação Mundial para o estabelecimento de normas e protocolos de comunicação.

2.3 Modelagem e simulação

Modelagem, de forma geral, é a elaboração de um modelo para a representação de alguma coisa. Modelo é a representação de um sistema real ou imaginário usando uma linguagem, um meio, e segundo um posto de vista. O aspecto mais importante de um modelo é a relação simplicidade versus fidelidade. Um modelo é a representação do conhecimento e a principal ferramenta para o estudo do comportamento de sistemas complexos. Modelar é o primeiro passo para a análise de um sistema de qualquer natureza e sob qualquer aspecto. Quando o modelo é uma representação válida de um sistema, informações significativas podem ser retiradas sobre sua dinâmica ou seu desempenho (ZEIGLER, 2000).

Segundo Dahmann (2000) modelagem é um processo complexo e, em vários campos, envolve a capacidade de dedução e inferência. A modelagem de um sistema será mais fácil se: a) conhecermos as leis (físicas, etc.) pertinentes ao sistema, b) se for fácil obter uma representação gráfica ou simbólica do sistema, e, c) se as incertezas entre as entradas, parâmetros e saídas puderem ser quantificadas.

Johansson (1993) destaca o crescimento exponencial da capacidade de processamento dos computadores digitais e sua penetração em todas as áreas da vida humana permitiu uma verdadeira explosão no uso de modelos e da simulação. Sem modelagem e identificação é praticamente impossível implementarmos simulações.

Uma simulação, seguindo o ambiente em que os modelos são implementados a simulação pode ser classificada de três formas descritas nos próximos itens (FUJIMOTO, 2000).

2.3.1 Simulação física

É a construção de modelos físicos, escalonados ou não, para a representação de sistemas reais. Pode-se implementar todo sistema ou somente partes que representam os aspectos de interesse.

2.3.2 Simulação analógica

Os modelos analógicos guardam relação de semelhança com partes ou com o todo do sistema modelado, mas são implementados em áreas distintas deste. A simulação analógica é a implementação destes modelos em meios analógicos.

Essa categoria de modelos é utilizada desde os primórdios da ciência. Com o advento dos computadores analógicos hidráulicos, pneumáticos e principalmente elétricos a implementação destes modelos nesses computadores passou a ser de importância fundamental para o avanço da engenharia.

Pode-se observar extraordinário avanço na área de Engenharia de Controle, os computadores analógicos são dispositivos pneumáticos, hidráulicos e eletrônicos compostos de blocos que programam as funções básicas matemáticas. Um modelo matemático pode ser desmembrado nestas funções básicas e implementado nestes dispositivos.

2.3.3 Simulação digital

São os modelos matemáticos ou lógicos implementados em computador digital através de funções (relações) de discretização no tempo. Também estão incluídos os modelos utilizando tabelas, curvas, banco de dados, etc.

A capacidade de implementação de modelos discretizados, incluindo tabelas e curvas, em computadores digitais ampliou a capacidade de modelagem de sistemas, principalmente os não lineares e variantes no tempo.

2.3.4 Vantagens e desvantagens da simulação

Seguindo os conceitos e definições sobre simulação do último item, pode-se destacar que a simulação representa um cenário de alguma empresa, simplificando situações para melhor visualização e análise, objetivando o desenvolvimento de soluções para problemas que podem acontecer na empresa, de forma que, gera

benefícios para o aprendiz e empresa. Alguns autores destacam que, as simulações possuem uma série de benefícios, assim como desvantagens na sua utilização.

Law e Kelton (1991) citam algumas vantagens da simulação:

- a) A possibilidade de reaplicação precisa dos experimentos, o que permite o teste de várias alternativas diferentes para o mesmo sistema. A manipulação das condições experimentais, o que não seria possível no sistema real;
- b) A avaliação de longos períodos em um espaço curto de tempo;
- c) A economicidade se comparada a experiências no sistema real, pois estas, quando realizadas, podem acarretar consequências danosas ou irreparáveis.

De acordo com Niveiros, Lopes e Kopittke (1999), as pessoas aprendem melhor por intermédio da própria experiência. Aprender fazendo só traz sucesso quando o *feedback* das ações for rápido e objetivo. Os jogos de empresa proporcionam meios para que os participantes possam, efetivamente, aprender, na prática, a lidar com seus mais importantes problemas sugeridos no dia-a-dia.

As principais desvantagens segundo Cornélio Filho (1998) e que merecem destaque são:

- a) Necessidade de exatidão na seleção dos dados de entrada do sistema, visto que, dados incorretos ocasionarão informações incorretas;
- b) Possíveis dificuldades na interpretação dos resultados, principalmente, para as pessoas não versadas na linguagem utilizada no modelo;
- c) Exigência de sólidos conhecimentos de informática e do objetivo da simulação para construção dos modelos;
- d) Lentidão no processo de desenvolvimento, em virtude do tempo necessário para modelagem e experimentação do sistema.

2.4 Sistemas Supervisórios

O primeiro registro da utilização de um sistema supervisório data do início da década de 1980, estes supervisórios eram dotados de pouca tecnologia visto que os

computadores da época não tinham muitos recursos e ocupavam enormes espaços. Havia alguns supervisórios que utilizavam microcomputadores, mas por se tratar de nova tecnologia, seu custo era extremamente alto, então somente plantas de energia e petróleo utilizavam-se destas tecnologias, pois a necessidade de mais recursos viabilizava o alto custo (VAX, 2008).

Com a crescente demanda dos processos e as novas práticas de gestão, que com o passar do tempo exigiam informações do processo em intervalos de tempo cada vez mais curtos, aliado a isto a grande evolução computacional com a descoberta dos transistores, começa-se a desenvolver sistemas supervisórios cada vez mais robustos, até que em meados da década de 1980 surgem os sistemas SCADA (*Supervisory Control And Data Acquisition*), os quais não só supervisionam o processo, mas também o controlam. Surgia deste ponto um novo mercado emergente, o mercado de sistemas supervisórios (SILVA, 2009).

Atualmente, os sistemas de automação industrial utilizam tecnologias de computação e comunicação para automatizar a monitoração e controle dos processos industriais, efetuando coleta de dados em ambientes complexos, eventualmente dispersos geograficamente, e a respectiva apresentação de modo amigável para o operador, com recursos gráficos elaborados e conteúdo multimídia.

Para permitir isso, os sistemas SCADA identificam 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. Essas variáveis são em tempo real como vazão, pressão, temperatura, nível, etc.. Uma vez coletada essas variáveis, os valores são apresentados ao usuário do sistema. A figura 1 representa o sistema de supervisão e controle:

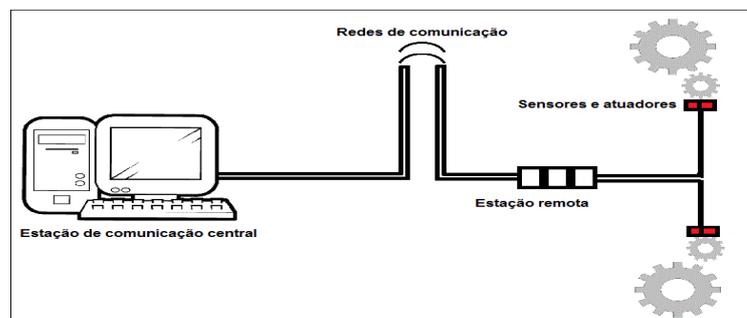


Figura 1 – Sistema de supervisão e controle.

2.5 Linguagem Ladder

A linguagem LADDER foi a primeira e ainda é a mais utilizada para programar Controladores Lógicos Programáveis, CLP's. A programação Ladder é um paradigma de programação diferente das linguagens de programação comuns. Ela foi inspirada na descrição de painéis de contatos e relés elétricos através de um diagrama em escada. A linguagem Ladder é uma linguagem gráfica que essencialmente representa redes de conexões de relés, temporizadores, contadores, comutadores, sinais de relógio, linhas de comunicação, operações matemáticas sobre os valores de variáveis. A linguagem ladder é uma das linguagens previstas no padrão IEC 61131-3 (GEORGINE, 2000).

A denominação “Ladder” deve-se à representação da linguagem parecer com uma escada onde entre duas barras verticais paralelas é apresentada a lógica de controle de acionamento da bobina de relés.

2.6 Conclusão

Com base nas teorias citadas de diferentes autores, podemos analisar que a automação industrial tem um importante papel de facilitar os processos industriais melhorando assim, a produtividade, tempo e qualidade dos produtos. Importante ressaltar que atualmente as indústrias estão dependentes desta tecnologia visto que, sem os recursos da automação é impossível competir com indústrias automatizadas.

A partir daí, pode-se concluir que estes sistemas se tornaram uma peça fundamental para o bom funcionamento dos processos industriais e também para a pesquisa, aprimoramento, e controle de qualidade e precisão da produção.

Pode-se dizer ainda que com o aumento de confiabilidade e flexibilidade destes sistemas, empresas de diversas áreas de produção demandam cada vez mais este tipo de solução.

Sobre a simulação, é importante dizer que, quando possível realizar uma simulação de algum evento no caso, sistemas supervisórios, se torna uma ferramenta poderosa no processo de treinamento de pessoas para operar em um cenário real futuramente.

3 Referencial Teórico

Este capítulo contempla uma revisão da literatura sobre a utilização dos softwares necessários para simulação de sistemas supervisórios abordando conceitos e características fundamentais para o desenvolvimento de um modelo.

3.1 *Software Factory Talk*

O *Software FactoryTalk* foi desenvolvido pela *Rockwell Automation* e contém um pacote integrado para desenvolvimento e operação de aplicação de interface homem-máquina (IHM) local, em rede, único e múltiplos usuários. O *Software* foi desenvolvido para processos automatizados ou monitoração da máquina e controle supervisório (RSTECHEd, 2010).

3.1.1 *Factory Talk View SE*

O *FactoryTalk View Site Edition* (também chamado de *FactoryTalk View SE*) é um pacote de *software* integrado para desenvolvimento e operação das aplicações de interface homem-máquina (IHM) que envolve diversos servidores e usuários distribuídos em uma rede. Um membro da família de produtos *FactoryTalk*, o *FactoryTalk View Site Edition* fornece todas as ferramentas que você precisa para criar aplicações de controle supervisório e monitoração do processo poderosas e confiáveis (RSTECHEd, 2010).

O *software FactoryTalk View SE* foi desenvolvido para ser usado com os sistemas operacionais *Microsoft Windows Server 2003*, *Windows XP*, e *Windows 2000*. O *FactoryTalk View Site Edition* consiste em várias partes de um *software* que você pode usar para construir aplicações de automação (RSTECHEd, 2010).

3.1.2 *Factory Talk View Studio*

FactoryTalk View Studio é o *software* de configuração para desenvolver e testar aplicações do *FactoryTalk View SE*. O *FactoryTalk View Studio* contém editores para criar aplicações completas e inclui *software* cliente e servidor para testar as aplicações criadas. Os editores servem para criar aplicações que tenham a simplicidade ou a complexidade que você precisa. Podemos usar o *FactoryTalk View Studio* para desenvolver aplicações do *FactoryTalk View Site Edition* e do *FactoryTalk View Machine Edition (ME)* (RSTECHEd, 2010).

O *FactoryTalk View* vem com bibliotecas gráficas e frontais de processos que podem ser usados nas aplicações sendo que, muitos dos objetos da biblioteca gráfica são pré-configurados com animação. No final do desenvolvimento de uma aplicação, é necessário visualizar e interagir com ela, para isso existe o *FactoryTalk View SE Client* (RSTECHEd, 2010).

3.1.3 *Factory Talk View SE Client*

O *FactoryTalk View SE Client* é um ambiente de operação completo para visualizar e interagir com as aplicações do *FactoryTalk View SE* locais e em rede.

De acordo com (RSTECHEd, 2010) o *FactoryTalk View SE Client*, você pode:

- a) Carregar, visualizar e interagir com diversas telas gráficas ao mesmo tempo a partir de múltiplos servidores;
- b) Executar a gestão dos alarmes;
- c) Visualizar as tendências históricas e em tempo real;
- d) Ajustar valores de referência;
- e) Ligar e parar componentes em qualquer servidor;
- f) Fornecer um ambiente seguro ao operador.

3.1.4 *Factory Talk View Administration Console*

O *FactoryTalk View Administration Console* administra as aplicações do *FactoryTalk View* depois que elas forem distribuídas. O *FactoryTalk View Administration Console* contém um subconjunto dos editores do *FactoryTalk View Studio* para que você possa fazer pequenas alterações em uma aplicação sem precisar instalar o *FactoryTalk View Studio* (RSTECHEd, 2010).

3.2 *Software RSLogix 5000*

O *software* RSLogix 5000 suporta a arquitetura ControlLogix, e, especificamente, é o *software* de programação do controlador Logix 5550. Este controlador representa a nova geração de produtos de controle da Allen-Bradley que é a marca de uma linha de equipamentos de automação de fábrica fabricado pela *Rockwell Automation*. A empresa, com faturamento de cerca de EUA \$ 4,5 bilhões em 2006, fabrica controladores programáveis para automação (PAC, PLC), interfaces homem-máquina, sensores, componentes de segurança e sistemas, *software*, unidades e sistemas de acionamento, contadores, motor centros de controle, e os sistemas feitos de produtos destes e outros (ROCKWELL *softwares*, 1999).

Algumas funcionalidades do *software* (ROCKWELL *softwares*, 1999):

- a) Facilidade de configuração, incluindo um organizador do controlador gráfico, com I / O (entrada e saída) de configuração, uma ferramenta de configuração de movimento e métodos de ponto e clique de configuração.
- b) Manipulação de dados, usando as duas matrizes e estruturas definidas pelo usuário, para proporcionar a flexibilidade necessária para a aplicação, ajusta uma estrutura de memória particular, tal estrutura definido pelo controlador de memória da tabela de dados.
- c) Fácil de usar métodos I/O.
- d) Um editor de linguagem *Ladder*, que permitem modificar vários degraus da lógica ao mesmo tempo, bem como inserir a lógica de programação a partir

da interface ponto e clique ou ainda, através de uma entrada ASCII *prompt*.

- e) Organização lógica de aplicações, usando estruturas de rotina.

3.3 VMware Workstation

VMware é um *software*/máquina virtual que permite a instalação e utilização de um sistema operacional dentro de outro dando suporte real a *software* de outros sistemas operativos (WIKIPEDIA, 2012).

Usando *software* de virtualização como o VMware é possível executar um ou mais sistemas operacionais simultaneamente num ambiente isolado, criando computadores completos (virtuais) a executar dentro de um computador físico que pode rodar um sistema operacional totalmente distinto. Do ponto de vista do utilizador e do *software* nem sequer se nota a diferença entre a máquina real e a virtual. É muito usado em centros de dados, pois permite criar redundância e segurança adicional sem recorrer a tantas máquinas físicas e distribuindo e aproveitando melhor os recursos das máquinas hospedeiras (WIKIPEDIA, 2012).

O *software FactoryTalk View SE* foi desenvolvido para ser usado com os sistemas operacionais *Microsoft Windows Server 2003, Windows XP, e Windows 2000* portanto, para haver uma compatibilidade com o *software*, é necessário que a máquina virtual VMware seja instalada em um computador para emular o Windows XP por exemplo, que será o sistema utilizado neste projeto.

3.4 Conclusão

O *Software FactoryTalk* da empresa *Rockwell* é perfeito para desenvolver um cenário simulado de uma indústria. Apesar de ser um sistema que tem compatibilidade apenas com *Microsoft Windows Server 2003, Windows XP, e Windows 2000*, que são relativamente Sistemas Operacionais antigos, a utilização deste *software* como processo de aprendizagem é de grande valia.

4 Modelo do Processo

O objetivo desenvolvido neste projeto foi simular o funcionamento de uma esteira industrial. Neste capítulo serão demonstrados como funcionam as aplicações em rede e local do *Factory Talk*, *Software RSLogix 5000*, códigos em linguagem Ladder, alguns objetos mais utilizados pela biblioteca de imagens do *Factory Talk* e finalmente os testes e resultados obtidos.

4.1 Softwares RSLogix 5000 e Factory Talk View SE

Neste índice será apresentado o desenvolvimento do projeto, configuração do RSLogix, interface homem máquina (IHM) apresentando os diagramas de funcionamento do projeto e o desenvolvimento e testes dos softwares.

4.1.1 RSLinx Classic

Programa responsável por promover a comunicação entre PC e CLP através de drivers / protocolos de comunicação.

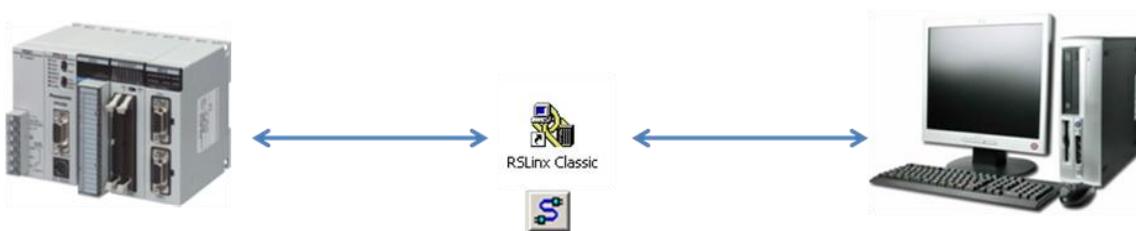


Figura 2-Comunicação CLP / PC

O *RSLinx* é solicitado quando:

- a) Quando for preciso fazer *Upload* ou *Download* de programas.
- b) Configuração das portas para programação *Ladder*.

O driver utilizado pra comunicação foi o 1784 – PCIC(S) do tipo ControlNet.

4.1.2 Configurando o RSLinx

Para ser possível realizar uma conexão entre CLP e PC virtuais é necessário configurar o RSLinx. Primeiramente selecionamos o driver de comunicação.

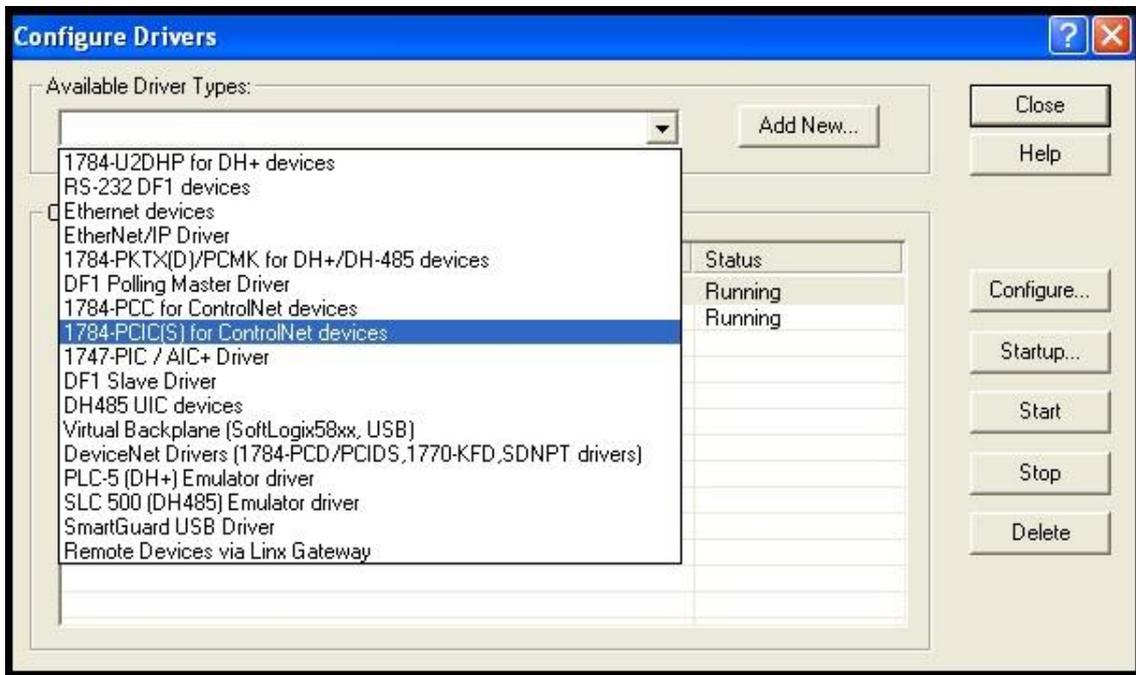


Figura 3 - Driver de comunicação 1784-PCIC(S) for ControlNet. Fonte: Programa RSLinx

Logo após, é necessário selecionar a porta de comunicação como “auto-configure”. Para finalizar é indispensável que o RSLogix 5000 esteja também configurado com esse mesmo driver de comunicação.

4.1.3 RSLogix 5000

Programa usado para desenvolvimento de aplicações em LADDER para a família de PLC's da Rockwell, bem como download, upload e monitoração de programas no PLC.

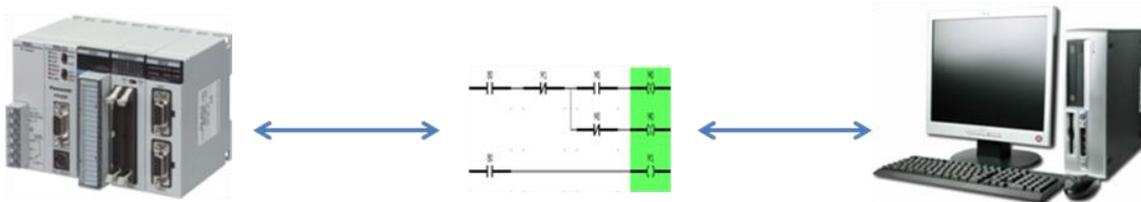


Figura 4- Programa Ladder que pode ser feito Upload ou Download para o CLP

Importante ser configurado com o mesmo Driver utilizado para comunicação: 1784 – PCIC(S) do tipo ControlNet.

4.1.4 *RSLogix Chassis Monitor*

RSLogix chassis monitor é o Rack de processamento onde concentra os módulos que executa as instruções previamente programadas, controla os dispositivos de entrada e saída e controla os canais de comunicação.

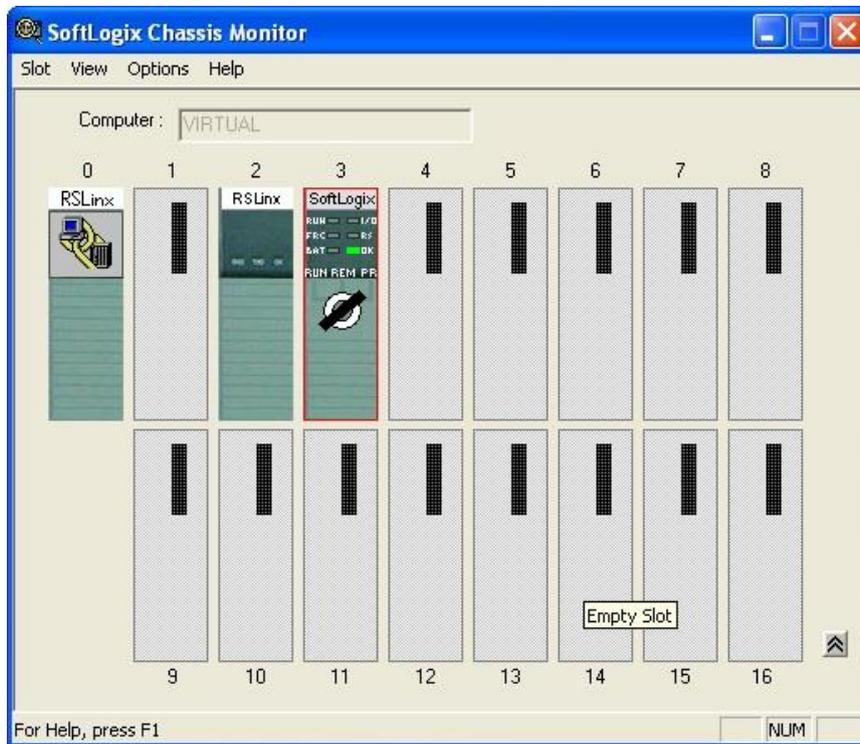


Figura 5-Rack contendo os programas Ladder. Fonte: Programa RSLogix Chassis Monitor

4.1.5 *Chaves do processador*

- Posição PROG: habilita o processador a ser programado, para qualquer execução e impede que passe para RUN remotamente;



Figura 6-Posição PROG do processador. Fonte: Programa RSLogix 5000

- b) Posição REMOTE permite que o processador seja programado e que passe para RUN remotamente (LED RUN apagado);



Figura 7- Posição REMOTE do processador. Fonte: Programa RSLogix 5000

- c) Posição RUN: inicia a execução do código na memória, impede que o processador seja programado remotamente e não permite download;



Figura 8-Posição RUN do processador. Fonte: Programa RSLogix 5000

4.1.6 *Inserindo o programa Ladder no Controlador*

Para inserir o código Ladder no controlador do terceiro *Slot* é necessário virar a chave para a posição PROG e assim realizar o *Upload* utilizando o programa *RSLogix 5000* assim como ilustrado na figura a seguir.



Figura 9-UPLOAD do código Ladder para o controlador.

4.1.7 *Factory Talk* – Aplicações em rede

Uma aplicação em rede pode conter vários servidores, executando em diversos computadores em uma rede com diversos usuários clientes que se conectam a aplicação simultaneamente de qualquer ponto da rede. Por exemplo, você pode usar servidores separados para diferentes áreas funcionais dentro da empresa e permitir que os clientes façam a interface a qualquer um dos servidores. As aplicações em rede têm uma ou mais áreas, um servidor de IHM por área e um ou mais servidores de dados. Uma área pode conter outra área dentro dela (RSTECHEd, 2010).

Uma vez que você criou as aplicações e um servidor de IHM, você pode usar os editores do *FactoryTalk View Studio* no projeto do servidor de IHM para criar os componentes da aplicação como telas gráficas, objetos globais e modelos de armazenamento de dados (RSTECHEd, 2010).

4.1.8 *Factory Talk* – Áreas

Uma área é uma divisão lógica dentro de sua aplicação. Você pode pensar em áreas como partições de seu disco rígido. As partições estão todas no mesmo disco principal (ou aplicação, nesta analogia), mas elas se dividem logicamente e mantêm as informações independentemente umas das outras. Uma área também pode ser usada para organizar a aplicação de maneira que faça sentido para o processo que ela está controlando (RSTECHEd, 2010).

Por exemplo, uma área pode representar uma parte de um processo ou uma região dentro da instalação do processo. Uma fábrica automotiva pode ser dividida em áreas chamadas Prensa e fabricação, Armação, Pintura, Motor e Transmissão; uma confeitaria pode ser dividida em áreas chamadas ingredientes, misturas cozimento e empacotamento (RSTECHEd, 2010).

Alternativamente, uma fábrica com linhas de produção idênticas pode ser dividida em áreas chamadas Linha 1, Linha 2, Linha 3 e assim por diante. Isso permitiria que você adicionasse novas linhas de produção idênticas à aplicação

copiando os projetos do servidor de IHM nas áreas novas (RSTECHEd, 2010).

A seguir um exemplo de uma aplicação em rede utilizando o *FactoryTalk View SE*.

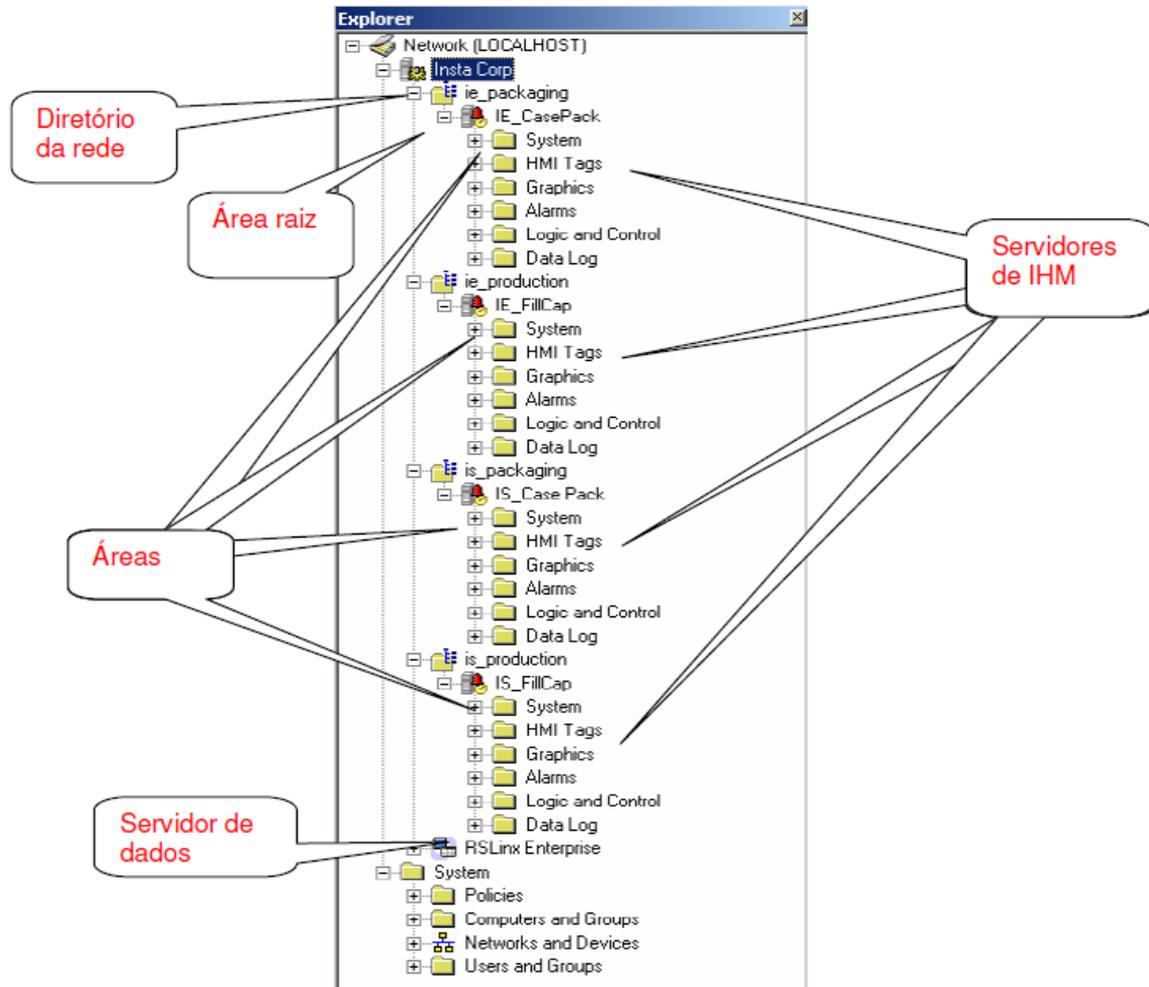


Figura 10-Aplicação em rede do Factory Talk View SE. Fonte: RSTECHEd (2010)

O diagrama a seguir mostra um exemplo de arquitetura de sistema que usa uma aplicação em rede com parte de um sistema FactoryTalk distribuído contando com: Centro de operação, centro de desenvolvimento, Servidores, Banco de dados, Servidores de rede o RSLogix 5000 e o processador para executar o código.

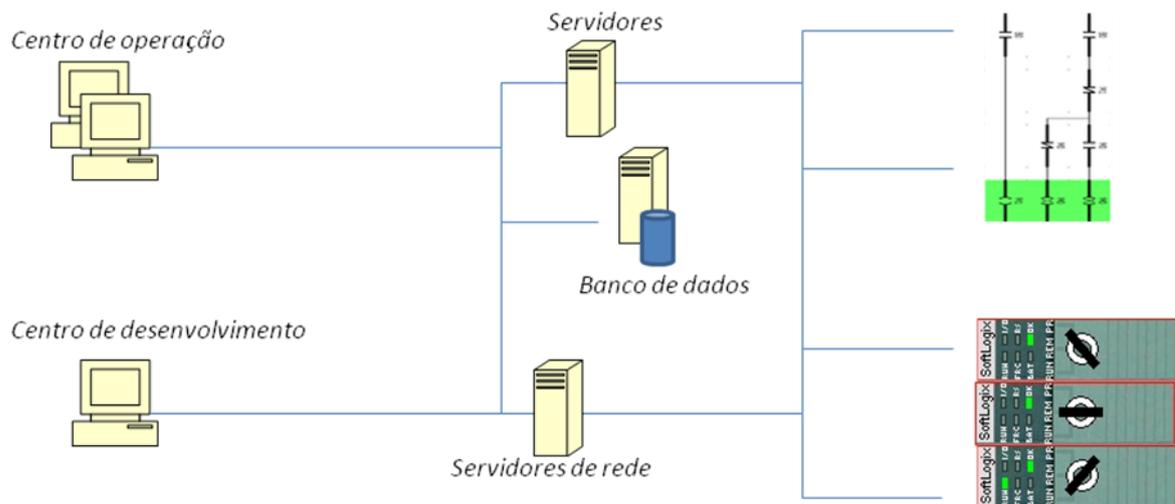


Figura 11-Arquitetura de um sistema em rede Factory Talk View SE. Fonte: RSTECHEd (2010)

4.1.9 Factory Talk – Servidores de IHM

Os servidores de IHM são programas de aplicação que fornecem informações aos clientes sempre que solicitam. Um servidor de IHM armazena os componentes do projeto de IHM (como telas gráficas) e disponibiliza esses componentes aos clientes. Um servidor de IHM também administra um banco de dados de TAGS, detecta os alarmes de TAGS da IHM e armazena dados históricos (RSTECHEd, 2010).

TAGS são mensagens digitais que levam informações como o endereço dentro do CLP, para o caso de retorno da informação, e o tipo de TAG (NATALE, 2000). Há vários tipos de TAGS, que servirão a propósitos distintos. TAGS podem ser do tipo:

- *Device*: significa que os dados se originam no CLP
- *DDE*: os dados se originam num servidor;
- *Memory*: os dados existem localmente no sistema supervisorio.

4.1.10 Factory Talk – Servidores de dados

Um servidor de dados fornece uma rota para os dispositivos físicos na rede, permitindo que as aplicações monitorem e controlem os valores nesses dispositivos.

Um servidor de dados pode ser um servidor de dispositivos *Rockwell Automation (RSLinx Enterprise)* ou um servidor de dados OPC que fornecem os valores dos TAGS. Uma vez que um servidor de dados está configurado, você pode ajustá-lo para apontar para um controlador específico como, por exemplo, um controlador ControlLogix. A configuração correta de um servidor de dados permite que você acesse diretamente um TAG (RSTECHEd, 2010).

Servidores de dispositivos *Rockwell Automation (RSLinx Enterprise)* fornecem melhor desempenho ao se comunicar com os controladores Logix5000 ou com muitos clientes. Você também pode usar o servidor *RSLinx Enterprise* para publicar os alarmes e eventos baseados em dispositivos (RSTECHEd, 2010).

Os servidores de dados OPC (incluindo *RSLinx Classic*) suportam todos os servidores de dados que estejam em conformidade com a norma OPC-DA 2.0.OPC, significa “*OLE for Process Control*”, um protocolo que permite que o *FactoryTalk View* recupere os valores dos TAGS de (RSTECHEd, 2010):

- a) Controladores programáveis e dispositivos *Rockwell Automation* usando o *RSLinx Classic* ou o *RSLinx Gateway* como um servidor OPC .
- b) Dispositivos de controladores de terceiros, como Siemens ou Modicon, que usem servidores OPC de terceiros.

4.1.11 *Factory Talk* – Servidores de alarmes e eventos

O *FactoryTalk Alarms and Events* permite que vários produtos *FactoryTalk* compartilhem uma visualização comum e consistente de alarmes e eventos por todo o sistema *FactoryTalk*. O *FactoryTalk Alarms and Events* suporta dois tipos de monitoração de alarmes (RSTECHEd, 2010):

- a) Monitoração de alarme baseada em dispositivo.
- b) Monitoração de alarme baseada em TAG.

Um servidor de alarmes e eventos pode ser um servidor de dispositivos *Rockwell Automation (RSLinx Enterprise)* que seja habilitado para monitorar os alarmes baseados em dispositivos ou um servidor de alarmes e eventos para monitorar

os alarmes baseados em TAGS (RSTECHEd, 2010).

Antes de configurar alarmes, é necessário fazer proposições e estabelecer definições, com a aprovação dos responsáveis técnicos do processo. Essas definições podem referir-se a:

- a) Condições de acionamento dos alarmes;
- b) Escolha e notificação de operadores;
- c) Envio de mensagens;
- d) Providência de ações.

Em tese, os alarmes teriam a função de:

- a) Chamar a atenção do operador para uma modificação do estado do processo;
- b) Sinalizar um objeto atingido;
- c) Fornecer indicação global sobre o estado do processo.

4.1.12 Bibliotecas

A Biblioteca gráfica vem com uma variedade de telas gráficas prontas contendo objetos que podem ser usados em outras telas.

Há muitos objetos que podem ser utilizados na aplicação. Neste projeto foram utilizados:

4.1.13 Botões

Existe uma variedade de botões na biblioteca FactoryTalk assim como disposto na figura a seguir.

4.1.16 Esteiras

Na figura 15, a ilustração de algumas esteiras as quais, foram utilizadas neste projeto.

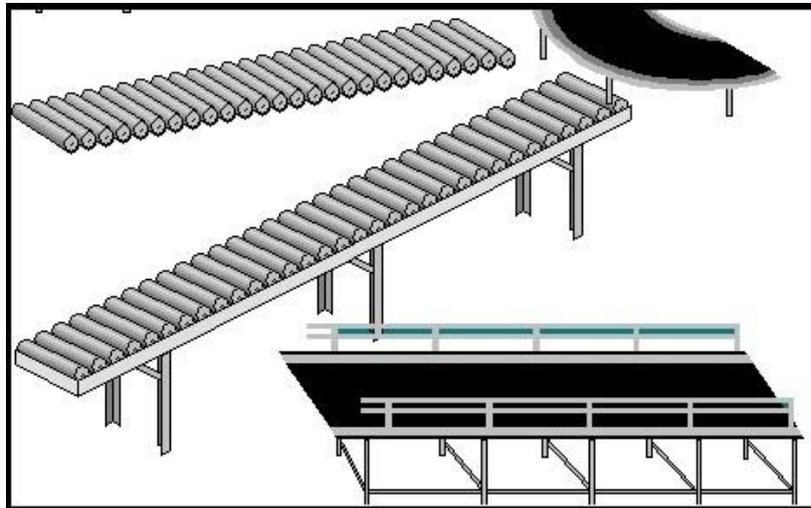


Figura 15 - Esteiras disponíveis na biblioteca *Factory Talk*

4.1.17 Tags

Um *tag* é um nome lógico que representa uma variável na memória de um dispositivo de rede ou local. O *FactoryTalk View SE* possui dois tipos de tags:

- a) *Tags* do servidor de dados (também chamados de *tags* de referência direta) fornecem acesso direto aos *tags* do controlador através dos servidores de dados que você adiciona a uma aplicação, assim como exemplificado na figura 9 abaixo.

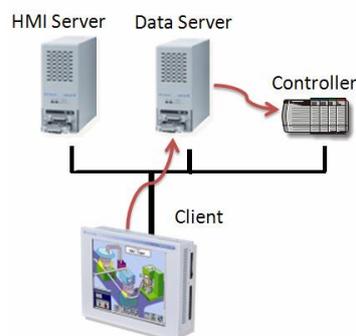


Figura 16 - Exemplo de *Tag* de referência direta. Fonte: RSTECHEd (2010)

- b) Tags de IHM fornece propriedades adicionais para a segurança do runtime e manipulação de dados. Os *tags* de IHM são criados no editor *Tags* e armazenados em um banco de dados de *tags* do servidor de IHM assim como exemplificado na figura 10 a seguir.

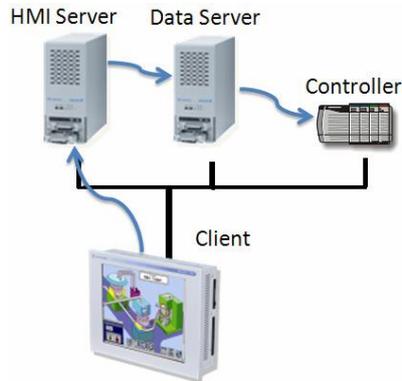


Figura 17 - Exemplo de Tag de IHM. Fonte: RSTECHEd (2010)

Para alguns fins em uma aplicação, usar os *tags* de referência direta oferece vantagens em relação ao uso dos *tags* de IHM e vice-versa. A tabela a seguir contém uma lista de vantagens tanto para os *tags* de IHM quanto para os *tags* de referência direta. Dependendo da aplicação, você pode escolher apenas um deles ou combinar os dois como de acordo com a tabela 1 (RSTECHEd, 2010).

Tags de IHM	Tags de referência direta
<p>Uso de recursos estendidos de <i>tags</i> de IHM</p> <p>Para aproveitar as vantagens dos recursos estendidos, como segurança no <i>runtime</i>, conversão de escala ou <i>offseting</i> dos valores de <i>tags</i>, ou endereçamento mais flexível, crie <i>tags</i> de IHM no <i>FactoryTalk View Studio</i>, no editor <i>Tags</i>.</p>	<p>Uso de referência direta para eliminar Duplicações</p> <p>Usar uma referência direto minimiza erros, pois é necessário criar o banco de dados de <i>tags</i> somente uma vez (no controlador) e não terá que duplicá-lo novamente na IHM. Isto se aplica a todos os objetos incluindo o <i>FactoryTalk Alarms and Events</i> baseado em dispositivos. Além disso, não é necessário criar um banco de dados de <i>tags</i> enquanto desenvolve a IHM, o que economiza tempo. Use também os <i>tags</i> do servidor de</p>

	<p>dados para adicionar, modificar ou excluir <i>tags</i> em um dispositivo sem ter que duplicar as alterações no banco de dados de <i>tags</i> do servidor de IHM.</p>
<p>Organização de <i>tags</i> em pastas</p> <p>Os <i>tags</i> podem ser organizados em pastas e subpastas lógicas. Por exemplo, crie uma pasta para o processo CIP de uma fábrica de soda. Em uma aplicação grande, você pode encontrar <i>tags</i> facilmente se organizá-las em pastas.</p>	<p>Permissão de acesso a tipos de dados complexos.</p> <p>Alguns dispositivos (controladores Logix5000, por exemplo) suportam tipos de dados como vetores e estruturas (UDT) que podem conter centenas de elementos como membros. Use os <i>tags</i> do servidor de dados para referenciar diretamente os valores dos <i>tags</i> e eliminar a necessidade de criar um <i>tag</i> de IHM para cada elemento membro. Você pode maximizar a otimização colocando os dados em vetores e estruturas.</p>
<p>Armazenamento de valores na memória</p> <p>Crie <i>tags</i> de memória da IHM para armazenar valores sem precisar de um dispositivo conectado ou acessível. Por exemplo, você pode usar os <i>tags</i> de memória para: Armazenar o resultado de um cálculo; Agir como um contador ou índice temporário; manter informações sobre o estado atual do sistema, por exemplo, qual foi a última tela gráfica exibida.</p>	<p>Reutilização de telas e outros componentes entre as aplicações.</p> <p>Por exemplo, exporte telas de uma aplicação para outra que use os mesmos <i>tags</i> do controlador. Não há necessidade de mover o banco de dados de <i>tags</i> de IHM. Todos os <i>tags</i> de referência direta são transferidos com a tela.</p>
<p>Reutilização dos nomes do <i>tag</i>.</p> <p>Os <i>tags</i> de IHM não precisam de endereços físicos codificados nem de nomes de variáveis específicas do</p>	<p>Velocidade.</p> <p>As referências diretas são um link direto com o servidor de dados e o controlador que ajuda a acelerar a taxa de atualização</p>

dispositivo. Isto significa que você pode reutilizar uma aplicação com outros dispositivos alterando os endereços físicos para os quais os nomes dos <i>tags</i> de IHM estão mapeados.	da aplicação.
Nomes descritivos Alguns controladores e servidores OPC não permitem nomes de <i>tags</i> descritivos, por exemplo, SLC. Criar um <i>tag</i> de IHM permite que você atribua nomes significativos aos <i>tags</i> ao invés de locais de memória como N7: 0.	

Tabela 1 - Comparação entre Tags de IHM e referência direta. Fonte RSTECHEd (2010)

4.1.18 Configurando a Conexão do *Factory Talk* com o *RSLogix 5000*

Um atalho de dispositivo permite que você crie um "indicador" para um dispositivo que você pode consultar através da aplicação enquanto desenvolve as telas. Isto permite que o usuário mude o local de um controlador ou outro dispositivo em um lugar, que se propaga para todo o resto do projeto sem precisar alterar todas as referências de *tags* para aquele controlador. Um atalho de dispositivo é semelhante a um atalho do *Windows* na área de trabalho de seu computador que fornece acesso fácil a uma aplicação (RSTECHEd, 2010).

A navegação *off-line* das *tags* não precisa mais que você associe o atalho do dispositivo a um controlador real. Nas versões anteriores, era solicitado que você associasse um atalho a um controlador e um arquivo de *tag off-line* para poder navegar pelos *tags* no arquivo. Agora, você pode criar um atalho que esteja associado somente ao arquivo de *tag off-line* (RSTECHEd, 2010).

.A figura a seguir indica o local na barra de navegação para configurar a comunicação com o *RSLogix 5000*.



Figura 18 - Barra de navegação do Communication Setup Factory Talk

Na próxima imagem podemos ver 2 Slots que foram configura disponíveis no RSLogix 5000. Para este projeto foi utilizado o Slot 1 onde contém todas as configurações necessárias para o funcionamento da esteira (1, 1789-L60/A Softlogix5860 Controller). Logo a esquerda da imagem está o atalho (*Shortcut*) que irá armazenar as informações configuradas no *Communication Setup*.

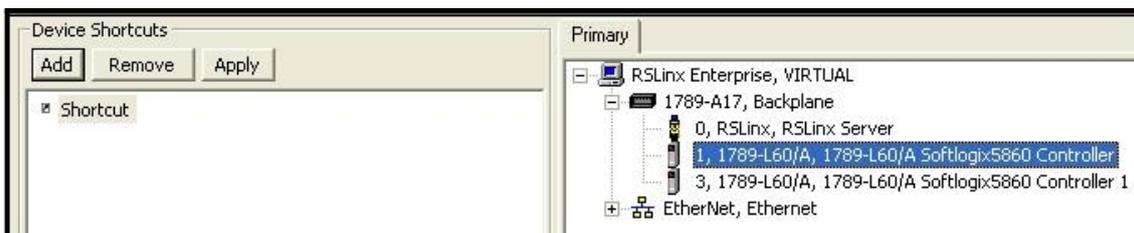


Figura 19 - Propriedades do atalho onde estará armazenada as configurações de comunicação Factory Talk / RSLogix 5000

Para registrar o programa (Main Program) no communication setup é necessário mostrar o endereço do programa, assim como exibido na figura 17.

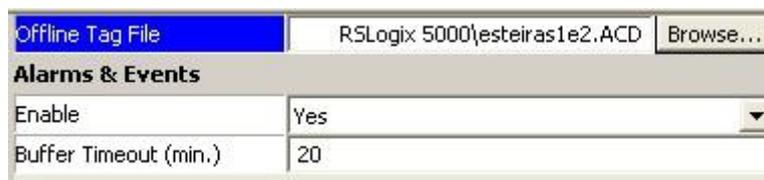


Figura 20 - Caminho para o programa principal

A figura 18 mostra os Slots ocupados no controlador Logix 5000 no computador “VIRTUAL” que é a maquina Local.

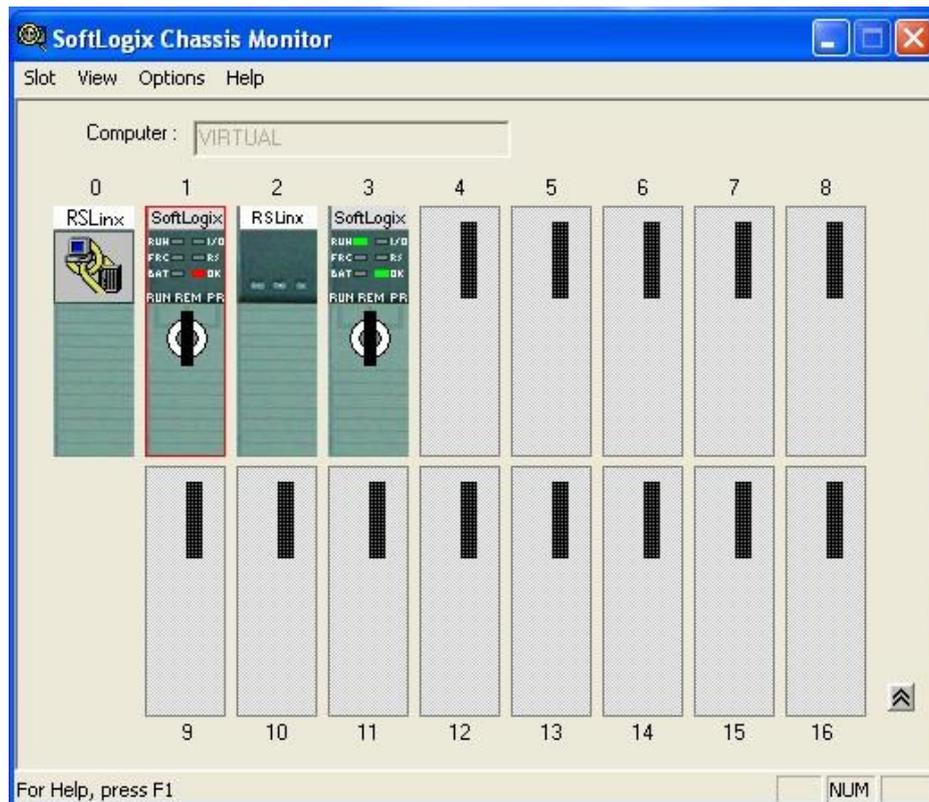


Figura 21 - Slots ocupados no SoftLogix 5000

4.2 Testes do processo de esteiras

O cenário utilizado para simulação foi a utilização de esteiras em uma fábrica de sucos assim, a esteira participa do processo de enchimento das garrafas, rotulação e empacotamento. Nos tópicos seguintes será demonstrado como é possível fazer essa simulação.

4.2.1 Processo de enchimento das garrafas

Como em toda indústria, é necessário ter um controle sobre a linha de produção para saberá quantidade exata de garrafas produzidas durante um determinado período de tempo, quantas garrafas foram cheias com sucesso para depois ir para o próximo processo de rotulação. A imagem a seguir ilustra a parte de enchimento das garrafas assim como o painel de supervisão e controle necessário para iniciar o processo,

escolher o tipo de produto a ser colocado nas garrafas bem como analisar a linha de produção.



Figura 22- Processo de enchimento das garrafas. Factory Talk.

4.2.2 Animação dos elementos

A animação é a capacidade de adicionar lógica a um objeto gráfico de forma que algumas características do objeto mudem quando um valor de TAG é alterado.

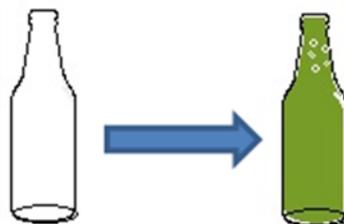


Figura 23 - Animação das garrafas

Por exemplo, um objeto pode ser feito para ser preenchido (acima, abaixo, à esquerda, à direita) ou para mudar de cor dependendo do valor de TAG. A caixa de diálogo de “Animation” mostra que expressões são usadas para animar objetos. Há uma

guia para cada tipo de animação, se houver um marcador de seleção em frente ao tipo de animação, isto significa que o objeto selecionado está usando tal animação assim como ilustrado na figura a seguir.

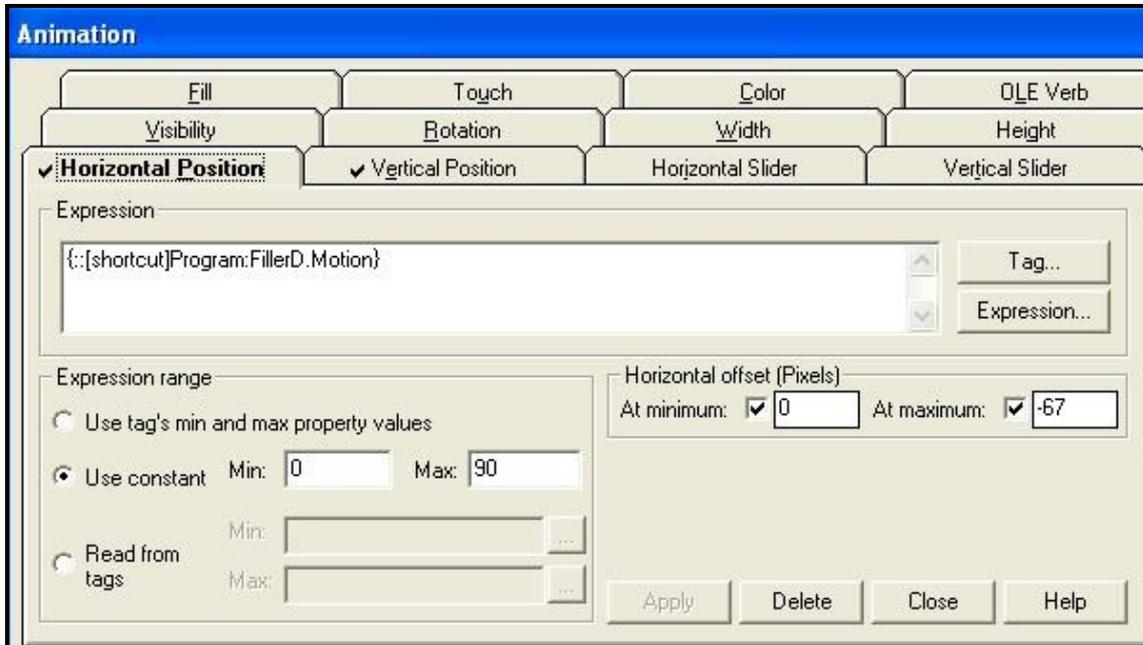


Figura 24- Janela de configuração de animações. Factory Talk.

Acessando esta janela é possível modificar o valor da TAG definindo as constantes “Min” e “Max” e assim gerar uma animação. O atalho “Shortcut” faz referência direta a TAG no programa do controlador.

Name	Value	Force Mask	Style	Data Type	Description
+ Block_Delay	{...}	{...}		TIMER	
- Motion_Count	{...}	{...}		COUNTER	
+ Motion_Count.PRE	300		Decimal	DINT	
+ Motion_Count.ACC	61		Decimal	DINT	
- Motion_Count.CU	0		Decimal	BOOL	
- Motion_Count.CD	0		Decimal	BOOL	
- Motion_Count.DN	0		Decimal	BOOL	
- Motion_Count.OV	0		Decimal	BOOL	
- Motion_Count.UN	0		Decimal	BOOL	
+ MotionCount	{...}	{...}		TIMER	
+ MotionCount1	{...}	{...}		TIMER	
+ Starve_Delay	{...}	{...}		TIMER	

Figura 25- TAG "Motion" presente no controlador. Factory Talk.

4.2.3 Processo de rotulação das garrafas

A próxima etapa é colocar rótulos nas garrafas de acordo com o tipo de produto, para isso, é necessário criar uma animação apenas para a garrafa chegar até determinado ponto e depois substituir a garrafa sem rótulo por uma com rótulo.



Figura 26- Colocando rótulos nas garrafas. Factory Talk.

Da mesma maneira feita no processo de enchimento das garrafas, no rótulo também é possível acompanhar a quantidade de garrafas por minuto, garrafas por dia, garrafas rejeitadas e garrafas boas e prontas para empacotamento, também é possível iniciar e parar o processo a qualquer momento através de uma botoeira de emergência e selecionar o tipo de rótulo a ser gravado.

A animação feita para gravação dos rótulos é feita seguindo a linha de raciocínio do enchimento de garrafas, altera-se o valor da TAG sendo que, para cada valor seleciona um tipo diferente de rótulo assim como pode ser visto na figura a seguir.

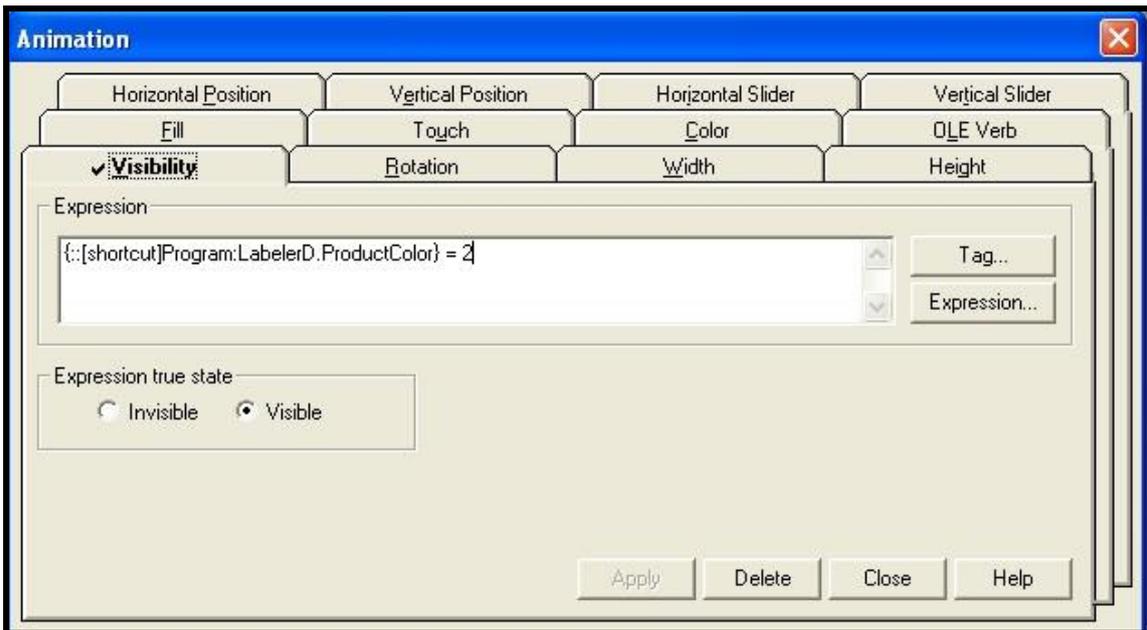


Figura 27-Animação dos rótulos. Factory Talk.

O atalho “Shortcut” faz referência direta as TAGS presentes no controlador, que como default contém o valor zero da cor do produto, sendo alterada na janela de configuração da animação `{:[shortcut]Program:LabelerD.ProductColor} = 2`.

Name	0	1	2	3	4	5	6	7
Motion	0							
Motion_Counter	0							
Motion_Speed	9							
onsbits	0							
ProductColor	0							

Figura 28- TAGS de animação presentes no controlador. RSLogix 5000.

4.2.4 Processo de empacotamento

Realizada a rotulação das garrafas a próxima etapa é o empacotamento e com um detalhe, apenas as garrafas em bom estado serão empacotadas. São duas esteiras com velocidades sincronizadas e um elevador para alinhar e deslocar as garrafas para a segunda esteira. É importante que o elevador também esteja sincronizado com a velocidade das esteiras para que a animação aconteça satisfatoriamente.

A animação da esteira é feita da mesma forma e como em todas as outras etapas, os objetos estão na posição horizontal e deslocados em uma determinada velocidade.



Figura 29-Esteira de empacotamento. Factory Talk.

A imagem a seguir representa a TAG de animação de objetos na vertical.

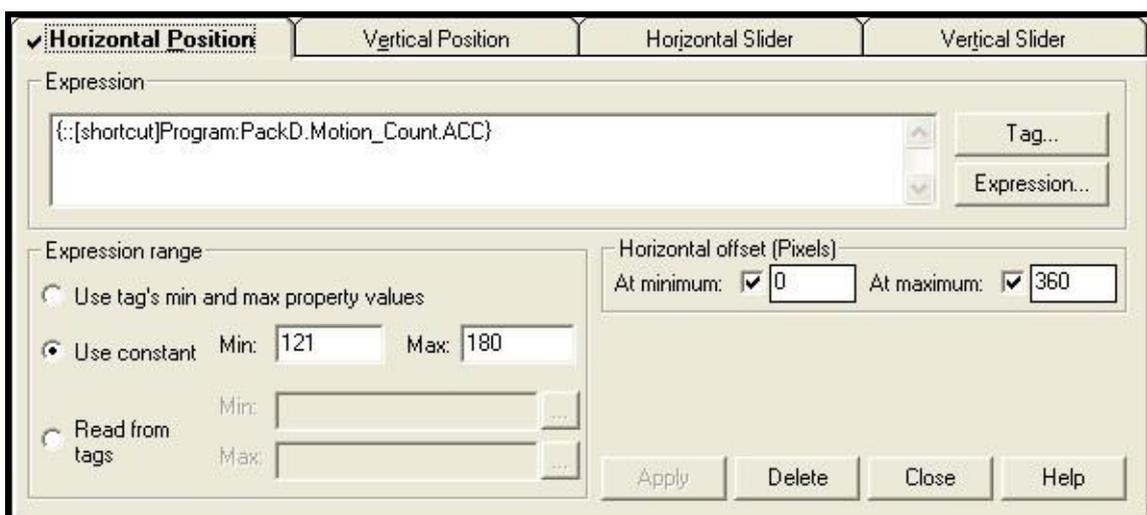
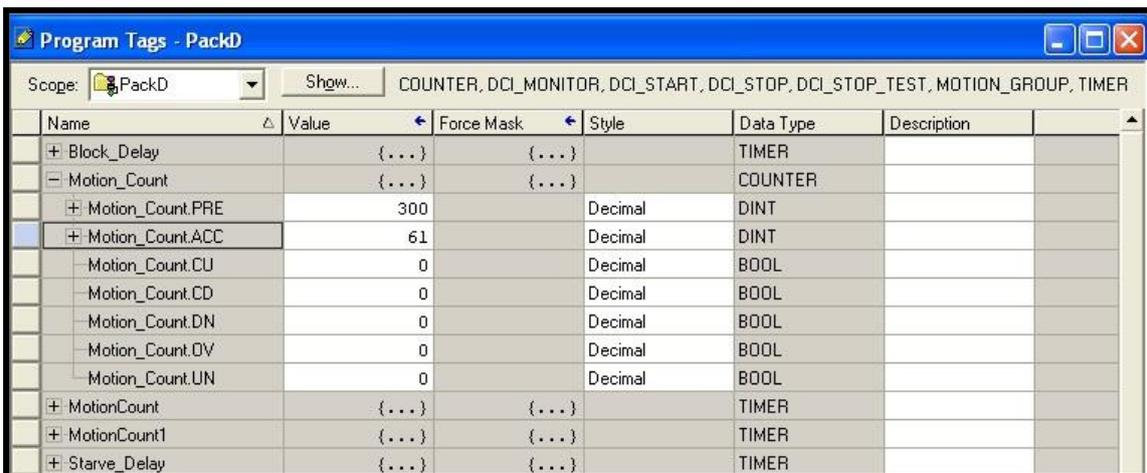


Figura 30-janela de configuração da animação. Factory Talk.

O atalho “Shortcut” faz referência direta as TAGS do controlador.



Name	Value	Force Mask	Style	Data Type	Description
+ Block_Delay	{...}	{...}		TIMER	
- Motion_Count	{...}	{...}		COUNTER	
+ Motion_Count.PRE	300		Decimal	DINT	
+ Motion_Count.ACC	61		Decimal	DINT	
- Motion_Count.CU	0		Decimal	BOOL	
- Motion_Count.CD	0		Decimal	BOOL	
- Motion_Count.DN	0		Decimal	BOOL	
- Motion_Count.OV	0		Decimal	BOOL	
- Motion_Count.UN	0		Decimal	BOOL	
+ MotionCount	{...}	{...}		TIMER	
+ MotionCount1	{...}	{...}		TIMER	
+ Starve_Delay	{...}	{...}		TIMER	

Figura 31- TAGS presente no controlador. RSLogix5000.

4.3 Conclusão

O Factory Talk foi o software escolhido para desenvolver este projeto, pois, ele oferece o suporte necessário para uma implementação de um cenário industrial.

Devido a complexidade do software *Factory Talk* me deparei com diversas dificuldades em representar um cenário fiel a uma planta industrial real, por isso, foram utilizados valores aleatórios nas variáveis porém, com fundamentos lógicos adaptados ao cenário.

O desenvolvimento da planta da esteira rolante foi perfeito, utilizando os recursos e desenhos da biblioteca do *Factory Talk* foi possível visualizar o processo simulado através do *Factory Talk Client*.

5 Considerações Finais

Após o término do desenvolvimento deste trabalho, chegou-se a conclusão de que é possível desenvolver uma simulação de um cenário industrial utilizando o Software Factory Talk.

Ao se buscar reproduzir no modelo a simulação da realidade cotidiana das indústrias, enfatizando-se o monitoramento de esteiras, objetivava-se criar condições para que os usuários do sistema vivenciassem as situações do ambiente industrial e, fundamentados nos conceitos de Automação Industrial e Sistemas SCADA, pudessem tomar decisões e testar seus resultados no ambiente simulado, capacitando-se, portanto, à reprodução destas estratégias nas indústrias reais.

Em relação ao primeiro objetivo específico “Analisar vantagens e desvantagens da aplicação de um software de simulação em uma área de automação de uma empresa.”, ele foi realizado a partir da revisão da literatura.

O segundo objetivo específico “Buscar melhor equilíbrio nas operações de monitoramento, através da simulação, sem afetar de forma real o andamento da indústria na parte de Supervisão, abrangendo diversas áreas como Softwares, Hardwares e comunicação.” Não foi totalmente atingido por estar utilizando variáveis aleatórias no trabalho.

O objetivo geral deste projeto que é “proporcionar o desenvolvimento de um cenário industrial para simulação utilizando o software Factory talk como ferramenta de apoio no processo de ensino identificando os conteúdos relevantes à Automação industrial e sistemas supervisórios e permitir o aprimoramento do conhecimento no processo de sistemas supervisórios no ambiente de uma empresa.” Foi parcialmente atingido, com um cenário de simulação montado utilizando o software Factory Talk, porém, com algumas divergências com relação à realidade.

5.1 Trabalhos futuros

Após o desenvolvimento e testes da simulação, alguns aprimoramentos podem ser feitos para melhorar sua funcionalidade. Dentre os quais, pode-se destacar:

- a) Possibilidade de acesso em rede, que o software *Factory Talk* oferece suporte.
- b) Aperfeiçoar a simulação com valores mais próximos aos reais para obter maior sucesso e fidelidade nos resultados.
- c) Desenhar plantas de outras partes da indústria.

Referências

ANTONELLI, P.L. **CLP Básico**, 1998

BOWEN, H.R.; MANGUM, G.L. **Automação e Progresso Econômico**. Tradução: Pinheiro, A. ., Rio de Janeiro, Zahar Editores, 1969

COOPER, D.R.; SCHINDLER, P.S. **Métodos de Pesquisa em Administração**. Porto Alegre: Bookman, 2003.

CORNÉLIO FILHO, P. **O modelo de simulação do GPCP-1: Jogo do planejamento e controle da produção**. Florianópolis, 1998. 95p. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-graduação em Engenharia de Produção, UFSC, 1998.

FERNANDINO. **Evolução da automação nos processos industriais**, 1995

FUJIMOTO, R. M. **Parallel and Distributed Simulation Systems**. New York: John Wiley & Sons, 2000.

GEORGINE M. **Automação Aplicada**. Editora Érica, 2000.

GIL, A. C. **Métodos e técnicas de pesquisa social**. 5. ed. São Paulo: Atlas, 1999.

HENRIQUE, M. **Automação industrial**. 2008

JOHANSSON, R. **System Modeling and Identification**. Englewood Cliffs: Prentice Hall, 1993.

KUHL, F. K.; WEATHERLY, R. and DAHMANN, J. **Creating Computer Simulation Systems**. Englewood Cliffs: Prentice Hall, 2000.

RSTECHEd. **Rockwell Automation FactoryTalk View SE 01**, 2010 (Apostila).

SILVA, B. S. **Processos industriais e controladores lógicos programáveis**, 2009.

VAX, M. **Automação aplicada a sistemas de monitoramento**, 1. Ed. São paulo, 2008.

ZEIGLER, B. P. PRAEHOFER, H. and KIM, T. G.; **Theory of Modeling and Simulation**. Second Edition. New York: Academic Press, 2000.