



**FACULDADE DE TECNOLOGIA SENAI CIMATEC
ESPECIALIZAÇÃO EM CONTROLE, AUTOMAÇÃO E
ROBÓTICA**

DANIEL SANTOS DE QUADROS CORREIA

**SISTEMA DE TRATAMENTO DE DADOS DE UM
PROTÓTIPO DE UNIDADE DE BOMBEIO
MECÂNICO UTILIZANDO A PLATAFORMA
PIMS**

Salvador
Setembro, 2014

DANIEL SANTOS DE QUADROS CORREIA

**SISTEMA DE TRATAMENTO DE DADOS DE UM PROTÓTIPO DE
UNIDADE DE BOMBEIO MECÂNICO UTILIZANDO A PLATAFORMA
PIMS**

Trabalho de conclusão de curso apresentado ao Curso de Especialização em Automação e Controle da Faculdade de Tecnologia SENAI-CIMATEC como parte dos requisitos para a obtenção do título de Especialista em Controle, Automação e Robótica.

Orientador: Prof. Me. Milton Bastos de Souza

Salvador
Setembro, 2014

DANIEL SANTOS DE QUADROS CORREIA
SISTEMA DE TRATAMENTO DE DADOS DE UM PROTÓTIPO DE
UNIDADE DE BOMBEIO MECÂNICO UTILIZANDO A PLATAFORMA PIMS/
DANIEL SANTOS DE QUADROS CORREIA. – Salvador, Setembro, 2014-
71 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Me. Milton Bastos de Souza

Trabalho de Conclusão de Curso – Faculdade de Tecnologia SENAI CIMATEC,
Setembro, 2014.

1. PIMS. 2. PI. I. Prof. Me. Milton Bastos de Souza. II. Faculdade de
Tecnologia SENAI CIMATEC. IV. SISTEMA DE TRATAMENTO DE DADOS
DE UM PROTÓTIPO DE UNIDADE DE BOMBEIO MECÂNICO UTILIZANDO
A PLATAFORMA PIMS

CDU 02:141:005.7

DANIEL SANTOS DE QUADROS CORREIA

SISTEMA DE TRATAMENTO DE DADOS DE UM PROTÓTIPO DE UNIDADE DE
BOMBEIO MECÂNICO UTILIZANDO A PLATAFORMA PIMS

Trabalho de conclusão de curso apresentado ao Curso de Especialização em Automação e Controle da Faculdade de Tecnologia SENAI-CIMATEC como parte dos requisitos para a obtenção do título de Especialista em Controle, Automação e Robótica.

Aprovado em 18 de setembro de 2014.

Banca Examinadora

Milton Bastos de Souza - Orientador _____
Mestre em Engenharia Elétrica pela Universidade Federal da Bahia
Faculdade de Tecnologia SENAI CIMATEC

Oberdan Rocha Pinheiro _____
Mestre em Modelagem Computacional e Tecnologia Industrial pela Faculdade de Tecnologia SENAI CIMATEC
Faculdade de Tecnologia SENAI CIMATEC

Setembro, 2014

AGRADECIMENTOS

A todos os colaboradores do SENAI CIMATEC que, direta ou indiretamente, contribuíram para o desenvolvimento deste projeto. Em especial a Fábio Correia, pelo fundamental apoio, e a todos os professores do Curso de Especialização em Controle, Automação e Robótica.

Ao Professor Milton Bastos, por sua orientação e suporte em diversos momentos.

Aos colegas Rafael Neri, Tiano Dórea e Tiago Torres sem os quais não seria possível o desenvolvimento do protótipo da Unidade de Bombeio.

E aos meus amigos e familiares.

RESUMO

A revolução industrial, ocorrida em meados do século XVIII, trouxe novas perspectivas e abriu um amplo horizonte para inovações relacionadas à mecanização de tarefas manuais. À medida que a tecnologia evoluía, tornava-se cada vez mais importante obter maior controle e conhecimento sobre o processo produtivo. Estes foram os primeiros passos para que a automação alcançasse o estágio de desenvolvimento atual, envolvendo cada vez mais processos e integrando o "chão de fábrica" com as decisões gerenciais estratégicas. No final do século XX a Indústria encontrava-se num ambiente cada vez mais competitivo e dinâmico. Isto estimulou a realização de grandes investimentos em automação e em sistema de informação como forma de responder à procura por produtos com menores custos, de alta qualidade e com curtos períodos de produção. Com esta demanda surgiram diversos *softwares* capazes de integrar as diferentes camadas de uma organização. Atualmente esta integração pode ser alcançada através do uso de sistemas como o *Manufacturing Execution Systems* (MES), o *Process Information Management System* (PIMS) e o *Enterprise Resource Planning* (ERP). A utilização de diferentes sistemas e fornecedores, entretanto trouxe como consequência problemas de interoperabilidade. A solução encontrada foi o desenvolvimento de um padrão aberto de protocolos de comunicação como alternativa aos *drivers* proprietários, resultando no padrão OPC (*OLE for Process Control*). O PIMS, objeto deste trabalho, abrange uma categoria de *softwares* capazes de gerenciar a aquisição, armazenamento e exibição de dados de distintas fontes de um sistema produtivo e é amplamente utilizado em diversos tipos de processo industrial. Um dos aspectos mais importantes nos sistemas PIMS é a sua capacidade de tratar um grande número de dados e transformá-los em informação e conhecimento, agregando valor aos dados brutos. Este estudo de caso apresentará conceitos sobre automação e controle industrial incluindo sistemas supervisórios, Controladores Lógicos Programáveis (CLPs), padrão OPC e a implementação de um ambiente PIMS utilizando o PI, da *OsiSoft*. Serão descritas as etapas e recursos necessários para a implementação do PI em um protótipo de Unidade de Bombeio Mecânico, simulando uma aplicação industrial real.

Palavras-chaves: *Process Information Management System* (PIMS). *Plant Information* PI. Automação.

ABSTRACT

The industrial revolution during the mid-eighteenth century, brought new perspectives and opened a wide horizon for innovations related to the mechanization of manual tasks. As technology was evolving, it became increasingly important to gain greater control and knowledge about the manufacturing process. These were the first steps for automation to reach the current stage of development, increasingly involving and integrating processes "shop floor" with the strategic management decisions. In the late twentieth century the Industry was in a more and more competitive and dynamic scenario. This stimulated major investments in automation and information system in order to respond the request for products with lower cost, high quality and short production periods. With this demand several software able to integrate the different layers of an organization were developed. Currently this integration can be achieved through the use of systems such as Manufacturing Execution Systems (MES), Process Information Management System (PIMS) and Enterprise Resource Planning (ERP). The use of different systems and suppliers, however brought interoperability problems. The solution was the development of an open standard for communication protocols as alternative for proprietary drivers, resulting in the OPC standard (OLE for Process Control). The PIMS, object of this work, covers a group of softwares able to manage the acquisition, storage and display of data from different sources in a productive system and is widely used in various types of industrial process. One of the most important aspects on the PIMS system is their ability of handling a large number of data and transform them into information and knowledge, adding value to the raw data. This case study will present concepts of automation and industrial control including supervisory systems, Programmable Logic Controllers (PLCs), OPC standard and the implementation of a PIMS environment using the PI, OSIsoft. Will be described the steps and resources needed to implement the PI on a Sucker Rod Pump prototype, simulating a real industrial application.

Key-words: Process Information Management System (PIMS). Plant Information PI. Automation.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de blocos de um sistema de automação	17
Figura 2 – Exemplo de controle	17
Figura 3 – Pirâmide de Automação	18
Figura 4 – Diagrama de blocos de um CLP	20
Figura 5 – Sistema Supervisório	22
Figura 6 – Sistemas com e sem OPC	25
Figura 7 – Arquitetura OPC	26
Figura 8 – Integração do chão de fábrica com o ERP com o PIMS	27
Figura 9 – Dados X Informação X Conhecimento	28
Figura 10 – Arquitetura PIMS	31
Figura 11 – Diagrama de funcionamento PI-API x PI-UDS	33
Figura 12 – Exemplo de funcionamento do algoritmo de exceção	34
Figura 13 – Reconstrução dos dados originais de processo	35
Figura 14 – Fluxo de dados no PI	35
Figura 15 – Tela inicial do PI Process Book	36
Figura 16 – PI Connection Manager	37
Figura 17 – About PI-SDK	37
Figura 18 – Criação de Tags através do System Management Tools	38
Figura 19 – Criação de Tags através do Excel	39
Figura 20 – Tag Search	39
Figura 21 – Novo ProcessBook	42
Figura 22 – Objetos do ProcessBook	42
Figura 23 – Exibir valor no ProcessBook	43
Figura 24 – Barra Time Range and Playback	43
Figura 25 – Principais métodos de elevação artificial	44
Figura 26 – Principais componente de uma UBM	45
Figura 27 – Estrutura mecânica completa	46
Figura 28 – Tela inicial do sistema supervisório	47
Figura 29 – Diagrama de comunicação do CLP com o PI Server	48
Figura 30 – Aba Service no PI-ICU	49
Figura 31 – Seleção de TAGs no PI OPC Client	49
Figura 32 – Exportação das TAGs através do Excel	50
Figura 33 – Tela principal do ProcessBook	50

LISTA DE ABREVIATURAS E SIGLAS

CAD	<i>Computer Aided Design</i>
CLP	Controlador Lógico Programável
CN	Comando Numérico
ERP	<i>Enterprise Resource Planning</i>
IHM	Interface Homem-Máquina
ISA	<i>International Society of Automation</i>
NEMA	<i>National Electrical Manufacturers Association</i>
MES	<i>Manufacturing Execution Systems</i>
ODBC	<i>Open Database Connectivity</i>
OLE	<i>Object Linking Embedding</i>
OPC	<i>OLE for Process Control</i>
OPC A&E	<i>OPC Alarms & Events</i>
OPC DA	<i>OPC Data Access</i>
OPC HDA	<i>OPC Historical Data Access</i>
PI	<i>Plant Information - Software de PIMS da OsiSoft</i>
PIMS	<i>Plant Information Management System</i>
PLC	<i>Programmable Logic Controller</i>
PI-ICU	<i>PI - Interface Configuration Utility</i>
PI-SDK	<i>PI - Software Development Kit</i>
PI-UDS	<i>PI - Universal Data Server</i>
SCADA	<i>Supervisory Control and Data Acquisition Systems</i>
SDCD	Sistema Digital de Controle Distribuído
SGBD	Sistema de Gerenciamento de Banco de Dados

TI	Tecnologia da Informação
UB	Unidade de Bombeio
UBM	Unidade de Bombeio Mecânico
VBA	<i>Visual Basic for Applications</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1	OBJETIVOS	13
1.2	JUSTIFICATIVA	14
1.3	METODOLOGIA	14
2	AUTOMAÇÃO INDUSTRIAL	16
2.1	ARQUITETURA DA AUTOMAÇÃO INDUSTRIAL	17
2.2	CONTROLADORES LÓGICOS PROGRAMÁVEIS - CLPs	19
2.2.1	PRINCÍPIO DE FUNCIONAMENTO	20
3	SISTEMAS SUPERVISÓRIOS	21
3.1	PLANEJAMENTO DO SISTEMA SUPERVISÓRIO	22
4	OPC - OLE FOR PROCESS CONTROL	24
5	PIMS	27
5.1	INFRAESTRUTURA	28
5.2	FUNCIONALIDADES	29
5.3	PI SYSTEM	31
5.3.1	PI-UDS	32
5.3.2	NÓS DE COLETAS - PI-API	32
5.3.3	ALGORITMOS DE COMPRESSÃO E EXCEÇÃO	33
5.3.4	Tags	34
5.3.5	Fluxo de dados	35
6	PROCESSBOOK	36
6.1	Conexão com o PI Server	36
6.2	TAGS	37
6.2.1	Performance Equations TAGs	38
6.2.2	PI Time	40
6.3	Telas gráficas	40
6.3.1	Time Range and Playback	43
7	RESULTADOS	44
7.1	UNIDADE DE BOMBEIO MECÂNICO	44
7.1.1	Protótipo	45
7.1.1.1	Estrutura mecânica	45

7.1.1.2	Sensores e Atuadores	45
7.2	CLP E IHM	46
7.3	Configuração da Interface OPC	46
7.3.1	Configuração OPC no PI	48
7.4	PI ProcessBook	50
8	CONSIDERAÇÕES FINAIS	51
	Referências	52
	ANEXOS	54
	ANEXO A – SUBSISTEMAS DO PI-UDS	55
	ANEXO B – PRINCIPAIS ATRIBUTOS DE UMA TAG	56
	ANEXO C – CÓDIGOS DO CLP	58

1 INTRODUÇÃO

Toda inovação tecnológica desenvolvida pelo homem tem o potencial de gerar uma nova perspectiva de vida, com novos desafios, oportunidades e necessidades, além de consequências importantes na economia e sociedade. Diversos inventos foram criados motivados pela necessidade de se reduzir o esforço braçal, permitindo aos beneficiados por estes dedicarem maior atenção a atividades intelectuais ou simplesmente liberando-os de tarefas inseguras ou desagradáveis.

A revolução industrial, iniciada em meados do século XVIII, na Inglaterra, foi responsável por alterar profundamente as estruturas de uma sociedade predominantemente agrícola e com meio de produção de bens de consumo exclusivamente artesanal em uma sociedade, do ponto de vista industrial, muito mais eficiente, produtiva e lucrativa. Para a indústria, este evento caracterizou-se pela substituição de diversas atividades braçais e repetitivas realizadas pelo homem por máquinas simples e especializadas em cada tarefa. Em 1788, pouco depois da criação da máquina a vapor de James Watt em 1781, já haviam sido desenvolvimentos meios primários de medição e controle de variáveis de processo com a criação de reguladores hidráulicos e pneumáticos, pelo próprio Watt. Já havia neste período, mesmo que de forma primária, o controle em malha fechada¹ em algumas máquinas (SILVEIRA; SANTOS, 2004).

Em 1950 surgiu o comando numérico (CN), permitindo a fabricação, de forma automática, de peças com geometria complexa e em grande quantidade. Estas máquinas eram comandadas por circuitos transistorizados e, já na década seguinte, por computadores. Com a consolidação desta nova tecnologia, pode-se afirmar que se iniciava uma nova era para a automação industrial. Neste período também foram desenvolvidos os primeiros Controladores Lógicos Programáveis (CLPs), uma nova ferramenta para apoiar o controle e acompanhamento de processos industriais. Este equipamento proporcionou uma evolução na automação industrial, trazendo maior flexibilidade aos sistemas automatizados, controlados anteriormente por relés eletromecânicos, temporizadores e contadores. Nos anos 70, os robôs, uma evolução das máquinas CN, já eram realidade na indústria e em diversas outras áreas (SILVEIRA; SANTOS, 2004).

Com a introdução da instrumentação eletrônica na indústria de processos, surgiram as primeiras grandes salas de controle, afastadas do chão de fábrica e que centralizavam em si todo o controle da planta industrial. A evolução das redes industriais possibilitou o desenvolvimento dos Sistemas Digitais de Controle Distribuído (SDCD), que consistem em

¹ Modo de controle caracterizado por uma modificação nos parâmetros do processo em função dos valores obtidos nas variáveis de saída.

diversos pontos de controle dispersos geograficamente pela planta e conectados a uma sala central de supervisão. Neste modelo há um monitoramento e controle globais do processo e diversos pontos de responsabilidade local, integrando processos em um sistema mais complexo (SILVEIRA; SANTOS, 2004).

No início da década de 90, surgiram as primeiras Interfaces Homem-Máquina (IHM), permitindo a visualização, através de telas configuráveis via *software*, diversas informações a respeito do processo de forma natural e prática. Neste período também foram desenvolvidos programas de computador para supervisão, que além da monitoração, permitiam a manipulação de variáveis do processo. A evolução destes *softwares* acompanhou o desenvolvimento dos sistemas operacionais, como o *Windows*, resultando em um salto de qualidade, tornando as IHMs mais completas e ainda mais amigáveis (GAIDZINSKI, 2003).

No final do século XX, num cenário cada vez mais competitivo e dinâmico, as empresas perceberam a necessidade de realizar investimentos mais significativos em automação e nos sistemas de informação para serem capazes de responder à demanda por produtos com custos cada vez menores, de alta qualidade e em curtos períodos de produção. Neste contexto surgiram *softwares* capazes de integrar as diversas camadas de uma organização, como compras, vendas e produção (GAIDZINSKI, 2003).

A velocidade de obtenção e a precisão das informações, além da integração entre as áreas, tornaram-se fatores cada vez mais críticos para a sobrevivência e crescimento das organizações. Para suprir estas demandas surgiram programas altamente especializados, classificados como *Supervisory Control and Data Acquisition Systems* (SCADA) -uma evolução dos *softwares* de supervisão- os *Process Information Management Systems* (PIMS), os *Manufacturing Execution Systems* (MES), entre outros (GAIDZINSKI, 2003).

O PIMS abrange uma categoria de *softwares* que teve como função inicial gerar bancos de dados em tempo real sobre variáveis de processo. Entretanto, o PIMS evoluiu e atualmente passou a incorporar outras características sendo capaz de gerenciar a aquisição, armazenamento e exibição de dados de diversas fontes de um sistema produtivo qualquer. Surgiu na indústria química e petroquímica, de processos contínuos, mas hoje é amplamente utilizado nos diversos tipos de processo industrial. Com o uso destes *softwares* é possível acompanhar parâmetros do processo de forma estruturada e clara através de tabelas, gráficos, curvas de tendências entre outras ferramentas visuais. Estes aplicativos são de amplo uso na indústria.

1.1 OBJETIVOS

Este trabalho tem como objetivo geral a realização um estudo de caso a respeito da implantação, configuração e utilização de um sistema PIMS para o tratamento de

dados em um protótipo de uma Unidade de Bombeio Mecânico (UBM) (maiores detalhes do protótipo podem ser vistos na Seção 7.1). Os conceitos e procedimentos utilizados neste estudo podem ser extrapolados para aplicações industriais reais. Adicionalmente, apresentar conceitos e funcionalidades de um sistema PIMS e a infraestrutura básica necessária para a sua implementação.

Como produto final deste trabalho, espera-se obter um ambiente PIMS totalmente configurado para o acompanhamento do funcionamento de uma UBM. Para tanto será utilizado o *software Plant Information* (PI) da *OSIsoft Inc.*. Além disso, espera-se como resultado o desenvolvimento de especialistas nesta plataforma de historiamento e tratamento de dados.

1.2 JUSTIFICATIVA

Sempre foi importante ter acesso e controle dos dados de um processo industrial, mas é cada vez maior a necessidade das empresas de integrar as informações dos diversos setores como forma de se manter competitiva em um mercado global, altamente dinâmico. Sob esta perspectiva, o PIMS se mostra uma poderosa ferramenta, capaz de coletar e armazenar dados de diversos pontos de uma planta industrial e disponibilizá-los, inclusive de forma remota, com segurança, para tantos quantos necessários níveis organizacionais da empresa.

Entre outros fatores que demonstram a importância do PIMS, podem ser destacados a possibilidade de acesso remoto, reduzindo as barreiras geográficas; capacidade de historiamento de dados, permitindo a rastreabilidade da produção e a flexibilidade de aplicação em diversos ramos da indústria.

Este trabalho justifica-se pois se propõe a planejar e implementar um sistema PIMS, utilizando ferramentas modernas e totalmente aplicáveis a situações industriais reais. O *software* utilizado será o *Process Information* (PI) da *OsiSoft*, amplamente disseminado entre grandes empresas, desenvolvendo um ambiente para supervisão e historiamento de dados, que tem o potencial de otimizar o processo de análise e tomada de decisões sobre os processos de interesse.

1.3 METODOLOGIA

A metodologia aplicada neste trabalho pode ser descrita através das seguintes etapas:

1. Estudo dos princípios de funcionamento das Unidade de Bombeio Mecânico reais;
2. Levantamento das variáveis relevantes para o acompanhamento do processo;

3. Desenvolvimento de um protótipo de uma UBM;
4. Estudo do pacote de programas *PI* da *OsiSoft*.
5. Obtenção dos *softwares* necessários para a a implantação de um sistema PIMS;
6. Desenvolvimento dos códigos para a programação do CLP;
7. Desenvolvimento de uma IHM para operação do protótipo;
8. Criação das telas de acompanhamento no *PI ProcessBook*;
9. Configuração do servidor PIMS;
10. Configuração da comunicação e implementação do conjunto CLP - IHM - PIMS.

2 AUTOMAÇÃO INDUSTRIAL

A palavra "*automation*" (traduzida como automação) foi criada por uma equipe de marketing da indústria de equipamentos nos anos 60. Neste período relacionava-se este termo ao controle automático de processos industriais, com a utilização de computadores, sistemas supervisórios e controladores ligados em rede.

De forma geral, utilizam-se duas plataformas de suporte para a automação industrial (MORAES; CASTRUCCI, 2001):

- Pequenos equipamentos de controle dedicados, os CLPs, que permitem o controle lógico¹ e dinâmico². Estes elementos de automação são de simples programação e instalação, com robustez adequada para operar no ambiente industrial.
- Os computadores de processo são ferramentas mais complexas e completas que os CLPs. Com a aquisição de dados relevantes sobre o processo, são capazes de simular condições, implementar leis de controle mais sofisticadas e permitem o desenvolvimento de interfaces com supervisores e gerentes.

Grande parte dos processos industriais automatizados está baseado em sistemas de malhas de controle fechadas, onde há retroalimentação dos valores das variáveis de processo para ação do controle. Este modelo está sustentado por três elementos básicos classificados como sensores, atuadores e controladores, como pode ser visto na Figura 1 (SILVEIRA; SANTOS, 2004).

Os sensores são os elementos sensíveis aos fenômenos físicos como a pressão ou a temperatura de um fluido, mensurando variáveis manipuladas e controladas. Atuadores são os mecanismos capazes de interagir no processo, alterando o valor das variáveis manipuláveis. Os controladores, por sua vez, são os dispositivos que, através da aquisição dos dados transmitidos pelos sensores e em função da sua programação e *setpoint* pré-estabelecidos, geram comandos para os atuadores (SILVEIRA; SANTOS, 2004).

Variáveis manipuláveis são aquelas que são modificadas pelos controladores para obter os valores esperados (*setpoints*) das variáveis controladas, geralmente na saída do processo. Um exemplo destes conceitos pode ser observado na Figura 2. Neste caso, o vapor do produto (saída deste sistema) é a variável controlada. A sua temperatura é medida e transmitida ao controlador pelo termoelemento, representado por TT. O controlador

¹ Esta forma de controle realiza-se através de circuitos onde as variáveis são binárias (valores 0 ou 1).

² Quando o ajuste dos parâmetros de processo se dá através da realimentação dos valores de saída das variáveis de interesse.

TC, em função do *setpoint* armazenado na sua memória, controla a válvula de vapor que alimenta o trocador. A vazão de vapor, neste exemplo, é a variável manipulada.

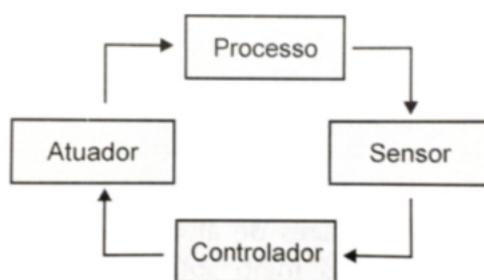


Figura 1 – Diagrama de blocos de um sistema de automação (SILVEIRA; SANTOS, 2004)

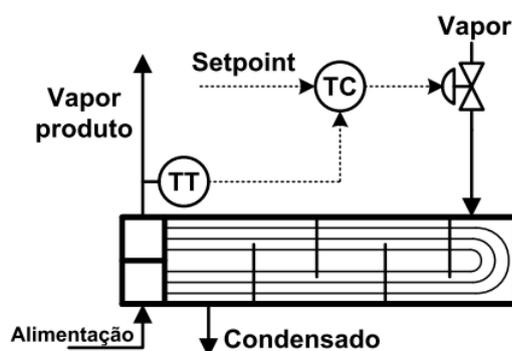


Figura 2 – Exemplo de controle (BOJORGE, 2012)

2.1 ARQUITETURA DA AUTOMAÇÃO INDUSTRIAL

A Figura 3 é conhecida como a pirâmide de automação. Nela estão representados as diferentes camadas de automação que podem ser encontradas numa planta industrial, partindo do chão de fábrica, na base da pirâmide, aos níveis gerenciais, no seu topo. Resumidamente:

1. No primeiro nível estão as máquinas, dispositivos e outros componentes localizados no chão de fábrica, como os atuadores, sensores e transmissores. Nesta camada é realizada a aquisição dos dados do processo e o controle é realizado, geralmente, por CLPs.
2. No segundo nível estão os elementos que realizam o controle das atividades da planta, em função das informações coletadas no chão de fábrica. Estão presentes neste nível os relés, CLPs e SDCDs.

3. A terceira camada permite o controle do processo da planta, há a presença de banco de dados com informações de mais alto nível, com indicadores sobre a qualidade da produção, produtividade, controle estatístico entre outros.
4. Na quarta camada estão as informações relacionadas à programação e planejamento da produção, geralmente envolvendo os setores de compras e logística.
5. O último nível da pirâmide inclui *softwares* que apoiam as decisões mais estratégicas para a empresa, como gestão de vendas e financeira. As aplicações desta camada apoiam a tomada de decisões que afetam a empresa de forma geral.

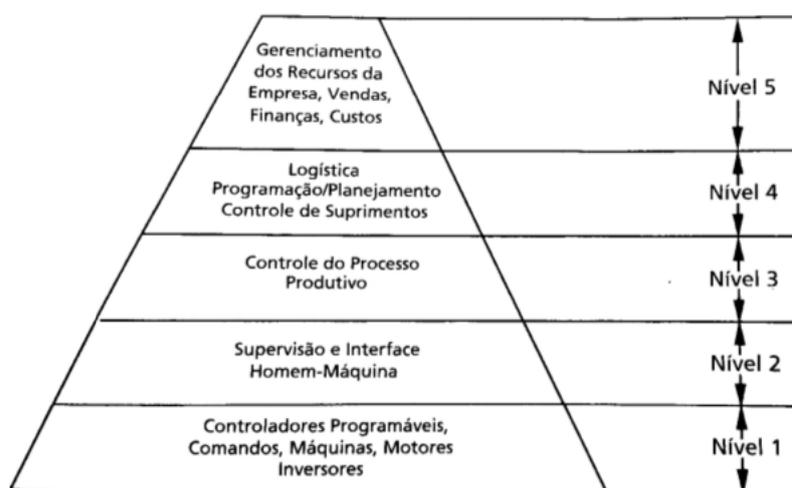


Figura 3 – Pirâmide de Automação (MORAES; CASTRUCCI, 2001)

Segundo Moraes e Castrucci (2001), os sistemas de automação também podem ser classificados de acordo com seu grau de complexidade:

- Automação Especializada: É aquela que utiliza sistemas microprocessados instalados no interior de equipamentos como TVs e telefones celulares. A programação é realizada, geralmente, utilizando-se linguagem de baixo nível, como o *Assembly*.
- Automação Industrial: Esta tem nível de complexidade médio e possui emprego, majoritariamente, local. É a automação presente em células de fabricação, processos químicos, térmicos, entre outros. O principal elemento de controle é o CLP.
- Grandes Sistemas de Automação: São os sistemas que estendem-se por grandes áreas, utilizando uma rede de computadores de diversos tipos e portes. É a automação empregada em sistemas de controle de vôo nos aeroportos, de defesa militar e nos bancos.

A automação trouxe aos processos industriais maior qualidade, produtividade, flexibilidade e confiabilidade ao mesmo tempo que reduziu os custos associados à mão de

obra. Além disso, facilitou a aquisição de dados para a tomada de decisões estratégicas e permitiu a realização de atividades perigosas ou inadequadas para o homem.

O PIMS está localizado na terceira camada da pirâmide, criando banco de dados temporais, com informações provenientes de diversas fontes e as disponibilizando em tempo real. Ainda é capaz de atuar como agente integrador, entre os dados obtidos por sensores no chão de fábrica na primeira camada da pirâmide com informações úteis esperadas pelos setores gerencias do quinto nível (BARRETO, 2006 apud MENDES, 2010).

2.2 CONTROLADORES LÓGICOS PROGRAMÁVEIS - CLPS

Os CLPs³ são equipamentos digitais capazes de controlar máquinas e processos, tornando possível a implementação de uma lógica de programação de forma dinâmica e flexível. É capaz, através de funções simples, realizar energização/desenergização, temporização, contagem, sequenciamento, operações matemáticas e manipulações de dados (MORAES; CASTRUCCI, 2001).

É definido pela norma NEMA (*National Electrical Manufacturers Association*) como "suporte eletrônico-digital para armazenar instruções de funções específicas, como de lógica, sequenciação, contagem e aritméticas, todas dedicadas ao controle de máquinas e processos".

Antes da consolidação dos CLPs o controle lógico e o intertravamento das linhas de produção era realizado por meio de elementos eletromecânicos, basicamente relés. Este método de controle exigia, em indústrias mais sofisticadas como a automobilística, painéis com centenas de relés. Em função das suas características físicas era necessário instalar painéis de grande dimensão, que garantissem a proteção contra umidade, temperatura, gases inflamáveis, poeira, entre outras condições comumente encontradas no ambiente industrial (SILVEIRA; SANTOS, 2004).

Os controladores lógicos programáveis surgiram no final da década de 60 como resposta a uma demanda da *General Motors* que necessitava de longos períodos (dias ou semanas) para alterar o seu sistema de controle baseado em relés. Era necessário realizar grandes mudanças todas as vezes que se iniciava a produção de um novo veículo ou quando havia alterações significativas na linha de montagem. Todo este esforço resultava em grande consumo de tempo e dinheiro. Os controladores lógicos programáveis, por sua vez, obtinham os mesmos resultados apenas com mudanças na configuração do *software*, gerando agilidade e economia (MORAES; CASTRUCCI, 2001).

Ao longo do tempo foram desenvolvidas diversas linguagens de programação. Estas, como em qualquer linguagem, constituem-se de sintaxe, semântica, comandos, blocos,

³ Também conhecidos como PLC, abreviatura do termo em inglês *Programmable Logic Controller*

entre outras características próprias de cada uma delas. Dentre as várias possibilidades, a LADDER é a linguagem mais utilizada para a programação dos CLPs. Nesta, o código é construído através de blocos funcionais que representam contatos e bobinas, de forma similar aos esquemas elétricos, sendo uma linguagem mais amigável aos engenheiros e técnicos da área de automação (MENDES, 2010).

Os CLPs foram responsáveis por tornar a automação uma realidade presente em praticamente todos os ramos industriais. Em relação aos antigos sistemas eletromecânicos, eles apresentavam uma significativa redução nos custos relacionados a materiais, ajustes de novos sistemas, mão-de-obra e instalação. Além disso, possuem características interessantes para a indústria como maior facilidade na detecção e correção de falhas, dimensões reduzidas e alta flexibilidade e confiabilidade. Levando em conta estes fatores, os CLPs foram rapidamente aceitos após a instalação das primeiras unidades em 1969 (MORAES; CASTRUCCI, 2001).

2.2.1 PRINCÍPIO DE FUNCIONAMENTO

A Figura 4 representa o diagrama de blocos de um CLP genérico. Este equipamento é constituído por três elementos principais: unidade de processamento (CPU) e interfaces de entrada e saída (SILVEIRA; SANTOS, 2004).

As variáveis de entrada são os sinais que chegam ao CLP, podendo ser provenientes tanto do processo industrial quanto de sinais gerados pelo operador. As variáveis de saída são respostas produzidas pelo controlador, podendo ser utilizadas para acionamento de equipamentos ou para sinalização de estados nos painéis. A CPU tem a função de processar a programação gravada na memória realizando ciclicamente a leitura das entradas, execução do programa e atualização das saídas (SILVEIRA; SANTOS, 2004).

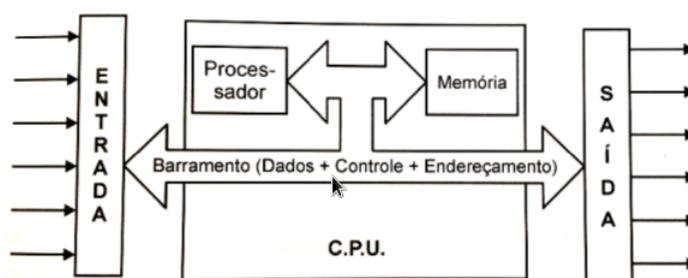


Figura 4 – Diagrama de blocos de um CLP (SILVEIRA; SANTOS, 2004)

3 SISTEMAS SUPERVISÓRIOS

Em função da complexidade e do grau de automação da planta industrial pode ser necessário utilizar algum tipo de interface que apoie a visualização e operação do processo. Assim desenvolveram-se as Interfaces Homem-Máquina (IHM) que permitem, de forma mais intuitiva e prática, a visualização de valores de variáveis de processo. De forma geral, as IHMs ainda possibilitam o controle de alguns elementos, como válvulas e atuadores, e a atualização de novos valores de *setpoints*. As IHMs, na maioria das vezes, são encontradas em um *hardware* específico desenvolvido pelo próprio fabricante do equipamento ou como um *software* de controle e supervisão, normalmente executado em computador comum.

Quando a interface está na forma de *hardware* num CLP, por exemplo, apresenta a vantagem de ser altamente robusta, sendo bem adaptada ao ambiente industrial. Por outro lado, esta solução pode ser aplicada somente aos equipamentos disponibilizados pelo fabricante em questão. Os Sistemas Supervisórios (também chamados de *Supervisory Control and Data Acquisition - SCADA*), por sua vez, permitem a comunicação com um grande número de fabricantes disponíveis no mercado. Além disso, podem ser executados em computadores não industriais, que embora mais frágeis, são de custo muito inferior (SILVEIRA; SANTOS, 2004).

Para que as informações sejam corretamente transmitidas e identificadas no sistema supervisório e nos bancos de dados, são associadas a cada dado uma *Tag* específica. Assim, cada variável de processo dentro de uma IHM está representada por uma *Tag* única. De forma geral, o sistema supervisório pode operar em dois modos (MORAES; CASTRUCCI, 2001):

- Modo de Desenvolvimento: Permite a criação e edição das telas gráficas utilizadas para supervisionar e controlar o processo.
- Modo Run Time: Utiliza as telas criadas no ambiente de desenvolvimento, como a da Figura 5. É o modo que interage diretamente com o CLP e que, efetivamente, supervisiona a planta.

Um funcionalidade importante disponível na maioria dos sistemas supervisórios é a capacidade de armazenar *setpoints* para diferentes configurações de operação dos CLPs. Desta maneira, é possível alterar rapidamente os parâmetros do processo de fabricação de um produto para um outro. O operador, neste caso, necessita apenas realizar a atualização destes *setpoints*, com comandos simples (SILVEIRA; SANTOS, 2004).

A interface gráfica torna a atividade de acompanhamento do processo muito mais prática e natural para o operador, substituindo os painéis com lâmpadas e rótulos. Esta solução utiliza informações por cores, textos, imagens e elementos animados graficamente.

As características de comunicação de uma IHM, como o SCADA, possuem vantagens adicionais como a operação remota e a integração com bancos de dados. A primeira permite que se controle processos a distância, aumentando a segurança do operador e viabilizando a operação de equipamentos e processos em locais remotos. A segunda vantagem possibilita uma troca dinâmica de dados entre diferentes níveis hierárquicos de controle, conectando o sistema a redes corporativas, à internet, entre outras.

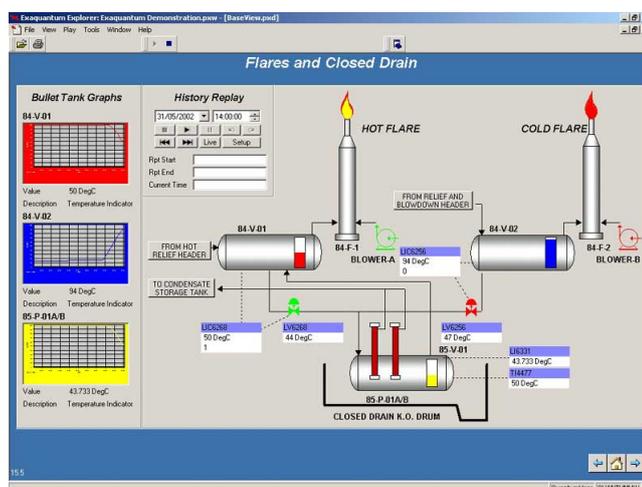


Figura 5 – Sistema Supervisório (YOKOGAWA, 2014)

3.1 PLANEJAMENTO DO SISTEMA SUPERVISÓRIO

Segundo Moraes e Castrucci (2001), são recomendadas as seguintes etapas para o planejamento e desenvolvimentos de um sistema supervisório:

1. **Entendimento do processo:** Consiste num estudo detalhado do processo a ser automatizado. Nesta fase devem ser levantados quais dados deverão ser monitorados e historiados.
2. **Tomada de dados:** Nesta etapa são escolhidos quais dados estarão presentes no sistema supervisório, de forma que este se torne conciso, facilitando a visualização e operação do sistema.
3. **Planejamento do banco de dados:** Em plantas de maior porte e complexidade é necessário utilizar um banco de dados para armazenar e gerenciar as *tags*.
4. **Planejamento dos alarmes:** Esta etapa requer o envolvimento dos responsáveis técnicos pelo processo para definições das condições de acionamento dos alarmes, o

modo de notificação e as ações que devem ser tomadas. É importante considerar que um grande número de alarmes possivelmente sobrecarregará o operador do processo, podendo tornar confusa a operação em certas situações. Por isso é necessário que se faça uma análise para se definir quais alarmes são realmente relevantes.

5. **Planejamento da navegação entre telas:** A organização das telas do processo em grupos com elementos relacionados (de uma mesma área ou processo, por exemplo) e condizentes com a realidade da planta facilita o uso do sistema, reduzindo o tempo necessário para o aprendizado.
6. **Desenho das telas:** Da mesma forma que para o planejamento da navegação, o desenho das telas deve visar simplificar a sua utilização. Além disso, deve-se buscar manter um padrão de cores, de nomenclatura e de distribuição dos botões nas telas. Os elementos gráficos do padrão ISA (*International Society of Automation*) são recomendados, pois o seu uso é comum na indústria, assim como utilizar cores com significados facilmente reconhecidos, como o verde e o vermelho.
7. **Gráfico de tendências:** Estes elementos demonstram o comportamento das variáveis analisadas ao longo do tempo. Podem ser utilizados, por exemplo, para observar tendências, monitorar a qualidade do processo e estudar a relação entre variáveis distintas.
8. **Planejamento do sistema de segurança:** No desenvolvimento de um sistema supervisório é possível restringir o nível de acesso e capacidade de interação dos usuários através de senhas.
9. **Padrão Industrial de Desenvolvimento:** Deve-se buscar a capacidade de integração do sistema supervisório com outras ferramentas normalmente utilizadas no ambiente industrial, como os editores de planilhas ou outros sistemas de informação.

4 OPC - OLE FOR PROCESS CONTROL

Os fornecedores de *softwares* geralmente concentram seus esforços no desenvolvimento das funcionalidades dos seus programas e resolvem as questões de comunicação com outros dispositivos e sistemas através do uso de *drivers*. Estes podem ser criados tanto pelos fabricantes de equipamentos quanto pelos desenvolvedores de *softwares*. Segundo Cândido (2004), a falta de padronização implica em:

- Existência de diversos *drivers* para um mesmo equipamento;
- Possibilidade de divergência de informações, em decorrência de distintas formas de uso das funções nos diferentes *drivers* existentes;
- Mudanças no *hardware* devem ser acompanhadas por atualizações nos *drivers*;
- Incompatibilidade entre aplicações devido às diferenças nos seus respectivos *drivers*.

Considerando estes inconvenientes, muitas empresas têm optado por adotar um padrão aberto de protocolos de comunicação como uma alternativa aos *drivers* proprietários. A principal solução adotada atualmente é o OPC (*OLE for Process Control*). O uso de um padrão aberto e comum permite a redução de custos e possibilita que sejam obtidos dados de qualquer fonte, equipamento ou aplicação, resultando em maior interoperabilidade entre as soluções de automação. A redução do ciclo de projeto é outra vantagem obtida com o uso deste padrão.

O primeiro OPC desenvolvido (hoje conhecido como *OPC Data Access Specification* - OPC DA) foi o resultado de um trabalho em conjunto de algumas das maiores empresas fornecedoras de dispositivos para automação atuando em cooperação com a Microsoft, utilizando como base a sua plataforma proprietária OLE¹ COM (*component object model*) e DCOM (*distributed component object model*). Este sistema é utilizado para viabilizar a comunicação entre processos e a criação de objetos em qualquer linguagem de programação que utilize essa tecnologia. Desta maneira foram definidos padrões para objetos, interfaces e métodos a serem utilizados no controle de processos e no desenvolvimento de aplicativos para automação (OPC FOUNDATION, 2014).

Uma analogia comum em relação ao surgimento do *OPC DA* é o desenvolvimento dos *drivers* de impressoras para o Windows. No ambiente DOS cada desenvolvedor era responsável por produzir os *drivers* específicos para cada impressora de cada marca que

¹ *Object Linking and Embedding*: Criação e manipulação de objetos que podem ser utilizados em diferentes pacotes de programas.

fosse considerada interessante suportar. Neste caso, uma empresa fornecedora de um *software* de CAD², por exemplo, além do desenvolvimento do seu próprio programa, era responsável por produzir e dar suporte a um grande número de *drivers* distintos. A solução encontrada pela Microsoft para aumentar a compatibilidade entre dispositivos foi incorporar o suporte no próprio sistema operacional, assim os *drivers* das impressoras (agora desenvolvidos pelos fornecedores destes equipamentos) seriam comuns a qualquer aplicação no Windows.

O padrão OPC possui alta qualidade e confiabilidade, já que um único protocolo padrão é reutilizado diversas vezes, facilitando o suporte e propiciando melhorias contínuas. Observando a Figura 6 pode-se perceber como a adoção do padrão OPC simplifica o processo de comunicação entre diversos dispositivos e sistemas. Neste caso, os fornecedores desenvolvem apenas interfaces OPC (clientes ou servidores OPC) que permitem a comunicação com qualquer outro elemento que também siga o padrão (OPC FOUNDATION, 2014).

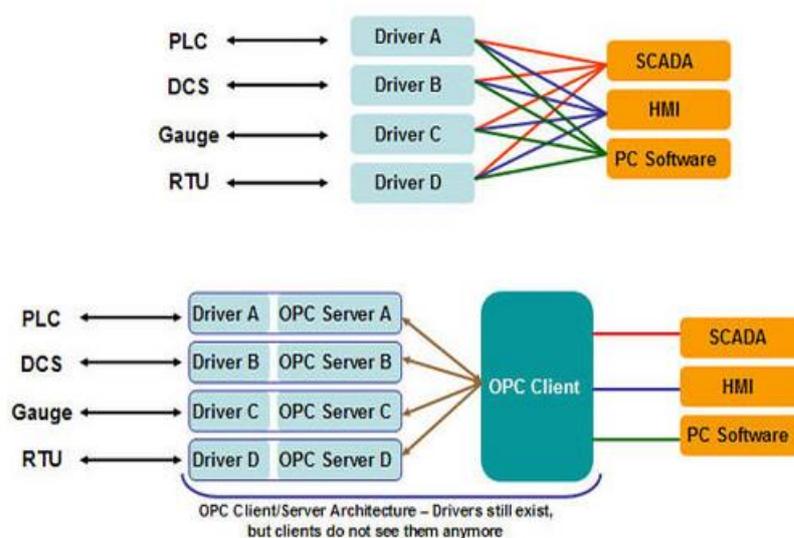


Figura 6 – Sistemas com e sem OPC (MOXA, 2014)

Com o uso do padrão OPC os fornecedores ainda foram capazes de criar servidores de alta performance com desempenho igual ou superior ao obtido com os *drivers* proprietários. Adicionalmente, os clientes ganharam maior flexibilidade e liberdade na escolha dos seus fornecedores de *software* uma vez que, com o uso do padrão, não há o problema de indisponibilidade de *driver* para aquele dispositivo específico.

Os benefícios obtidos com o OPC DA para a aquisição de dados de processo impulsionou a criação de padrões para outros tipos de informações como alarmes, eventos, dados históricos e dados em lote (batelada). Atualmente, os principais padrões OPC são:

² *Computer Aided Design*: Categoria de *softwares* de apoio ao desenvolvimento de projetos e desenhos técnicos

- *OPC Data Access* (DA): Permite o acesso de dados em tempo real e a obtenção de *Timestapms*³. Aplicado, por exemplo, na comunicação entre CLPs e IHMs.
- *OPC Historical Data Access* (HDA): Utilizado para recuperar e analisar dados históricos em um banco de dados.
- *OPC Alarms & Events* (A&E): Usado para o gerenciamento das notificações de alarmes e eventos. Este sistema de monitoramento ocorre somente sob demanda, em contraste ao OPC DA, que fornece um fluxo contínuo de dados.

Há ainda outros padrões como o *OPC Batch*, específico para processos em batelada, o *OPC Data eXchange*, utilizado para comunicação entre servidores, o *OPC Security*, que tem como objetivo controlar o acesso de clientes aos servidores de forma segura, entre outros. A manutenção e o desenvolvimento do padrão OPC são regulados pela *OPC Foundation*⁴, uma entidade composta por mais de 400 empresas. A Figura 7 demonstra um modelo de aplicação seguindo a arquitetura dos padrões OPC.

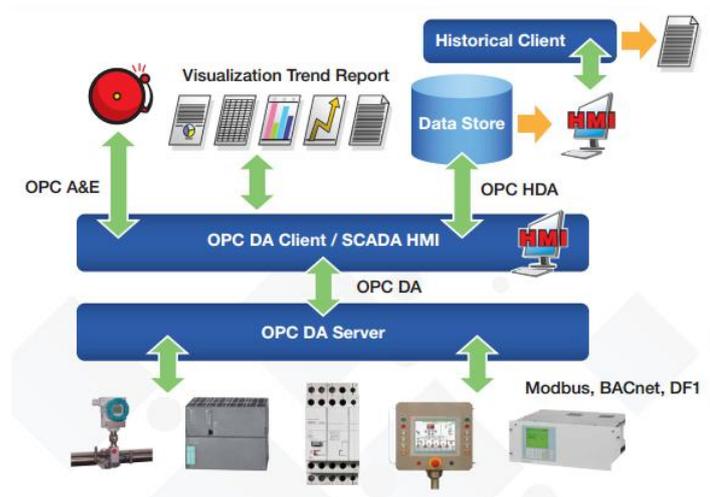


Figura 7 – Arquitetura OPC (OPC FOUNDATION, 2014)

³ Informação relativa ao momento em que certo evento ocorreu.

⁴ <<http://www.opcfoundation.org/>>

5 PIMS

O modelo ERP (*Enterprise Resource Planning*), utilizado para o planejamento de recursos de uma empresa, busca a integração do gerenciamento da informação de diversos sistemas. O ERP localiza-se no topo da pirâmide da automação, como pode ser visto nas Figuras 3 e 8 e, por isso, opera com dados de valor mais estratégico para a companhia. O PIMS é uma das ferramentas que dá suporte ao modelo ERP (JESUS; RAMOS; GUEDES, 2012).

Segundo Seixas Filho (2005), os PIMS (*Process Information Management Systems*) "são sistemas que adquirem dados de processo de diversas fontes, os armazenam num banco de dados históricos e os disponibilizam através de diversas formas de representação". Surgiram como uma demanda da indústria de processos contínuos para eliminar a fragmentação de dados e fornecer uma visão global da produção. Com o uso dessa ferramenta o engenheiro de processo pode acompanhar os valores das variáveis em tempo real, assim como acessar informações históricas. Além disso, o PIMS pode ser visto como agente facilitador para a implementação de outros *softwares*, possibilitando a integração do ERP com os sistemas de chão de fábrica, como visto na Figura 8.



Figura 8 – Integração do chão de fábrica com o ERP com o PIMS (SEIXAS FILHO, 2005)

O PIMS é uma ferramenta muito abrangente e, por isso, muitos confundem a sua aplicabilidade com a dos sistemas MES e SCADA. Embora o PIMS tenha incorporado muitas das funcionalidades destes sistemas e tenha deixado de ser meramente um historiador de dados, é importante ressaltar que este não é o seu uso ideal, não possuindo algumas das suas principais funções. Assim, geralmente, não é possível utilizar o PIMS como um substituto completo destes sistemas.

Um dos aspectos mais importantes nos sistemas PIMS é a sua capacidade de tratar um grande número de dados e transformá-los em informação e conhecimento, agregando valor aos dados brutos à medida que são tratados, como pode ser visto na Figura 9. O engenheiro se beneficia da possibilidade enxergar o processo de forma global e poder

comparar os parâmetros, atuais e passados, como a quantidade de reagentes utilizados, perfil de temperatura numa torre, velocidade do agitador, entre outros. Assim, é possível armazenar valores ótimos de *set-up* utilizando-os como *benchmarking* para uma contínua melhoria dos processo.



Figura 9 – Dados X Informação X Conhecimento (SEIXAS FILHO, 2005)

De acordo com [Carvalho et al. \(2005\)](#), os principais benefícios gerados pelo uso de uma ferramentas PIMS são:

- **Centralização de dados de processo:** A utilização de uma base de dados única permite maior facilidade de análise sobre estes dados.
- **Democratização da informação:** Múltiplos usuários podem ter acesso aos dados de processo instantaneamente, inclusive de forma remota.
- **Visualização do processo produtivo em tempo real:** É possível acompanhar o processo em forma de gráficos de tendência, relatórios dinâmicos, aplicações Web, etc.
- **Maior interatividade com os dados do processo:** O PIMS é capaz de realizar cálculos, análises estatísticas e lógicas, entre outros.
- **Histórico de dados:** Graças à utilização de algoritmos de compressão, é possível armazenar uma grande quantidade de dados de processo.
- **Receita de processo:** Através do armazenamento dos valores de *set-up* referentes aos melhores resultados obtidos é possível utilizar essa informação para referências futuras.

5.1 INFRAESTRUTURA

A infraestrutura necessária para a implementação de um sistema PIMS pode ser descrita, basicamente, como sendo ([CARVALHO et al., 2005](#)):

- **Servidor Principal:** É neste equipamento que os dados proveniente dos servidores de comunicação se concentram e são armazenados e é o elemento que possibilita a disponibilização destes para as demais aplicações.
- **Servidores de Comunicação:** São os elementos intermediários (e opcionais) entre os sistemas de supervisão e controle da planta com o servidor principal. É possível adquirir dados diretamente do chão de fábrica utilizando o protocolo OPC, configurando o PIMS como cliente OPC e o supervisor como servidor OPC.
- **Estações Clientes:** São computadores que possuem alguma aplicação cliente do PIMS (ver Seção 5.3). Utilizadas para visualização de telas gráficas e tratamento dos dados do servidor principal.
- **Banco de Dados Relacional:** O *software* PIMS pode fazer uso de um banco de dados relacional externo.
- **Infraestrutura de Rede:** É necessário utilizar *switchs* para possibilitar a comunicação entre o servidor principal e às redes corporativa e de controle, servidores de comunicação, estações clientes e o banco de dados.

5.2 FUNCIONALIDADES

A arquitetura do PIMS pode ser dividida em três camadas: *Coleta, Armazenamento e Exibição*, como visto na Figura 10. Na primeira, o PIMS recebe os dados fornecidos por CLPs, sistemas SCADA e SDCs. Cada variável possui uma *tag* única associada. As *tags* são as principais unidades da camada de armazenamento. Ela centraliza o historiamento dos dados e os disponibiliza para a camada seguinte. Para o armazenamento, são incorporados à *tag* o valor da variável e uma *timestamp*, relativa ao momento de coleta daquele dado. No último nível são disponibilizados ao usuário os dados armazenados na camada anterior (JESUS; RAMOS; GUEDES, 2012).

Segundo Carvalho et al. (2005), basicamente, o PIMS possui três grupos de função:

1. **Historiador de processos:** O PIMS é capaz de coletar dados provenientes de diversas fontes como CLPs, sistemas supervisórios e SDCDs. Geralmente utiliza o padrão OPC para a comunicação apesar de possuir interface para a maioria dos sistemas comerciais. O método de leitura dos dados pode ser cíclico ou sob demanda (*unsolicited messages*). Normalmente o armazenamento destas informações é realizado num banco de dados temporal. Neste ponto pode-se destacar uma das principais características do PIMS: uma grande capacidade de armazenar dados históricos. Isso é possível graças à utilização de algoritmos de compressão.

2. **Interface gráfica:** Utilizada para a extração de dados, consulta e visualização. O usuário pode solicitar dados históricos, criar gráficos de tendência, exportar dados para outros programas, ver telas com animações gráficas sobre o processo (tempo real ou histórico), exportar e importar dados para um banco de dados relacional, entre outros.

3. Aplicações complementares:

- Banco de dados relacionais: É possível estabelecer uma interface entre o sistema PIMS e bancos de dados relacionais. Isto porque para algumas aplicações, quando é necessário tratar informações relacionais, recomenda-se que estas sejam gerenciadas por um banco de dados relacional externo.
- ERP: O PIMS pode ser utilizado como uma camada intermediária (*middleware*) entre os sistemas de chão de fábrica e o ERP, no topo da pirâmide da automação. Muitas vezes esta é a justificativa utilizada para a implantação de um sistema PIMS.
- Gestão de bateladas: Para suprir demandas das indústrias que operam com algum processo em batelada, foram desenvolvidos módulos para monitorar a produção neste modelo.
- Reconciliação de dados: Alguns dados são obtidos através da combinação da leitura de distintos instrumentos. Caso algum destes equipamentos esteja descalibrado, a margem de erro do resultado será ampliada. Entretanto, existem ferramentas complementares que possuem a capacidade de distribuir os erros de medição e calcular balanços de massa e energia de forma a melhorar a precisão dos dados finais.
- Controle estatístico de processos: O PIMS possui módulos que são capazes de analisar o processo através do controle estatístico.
- Genealogia: É possível relacionar, com o uso deste módulo, o produto final com os seus insumos. Desta forma é possível rastrear a que lote pertence cada um dos componentes daquele produto acabado. Além disso, é possível recuperar o resultado de testes aplicados, o responsável pela montagem, o momento de introdução no processo, etc.
- Outros aplicativos: É possível configurar interfaces entre o sistema PIMS e outras aplicações, como as utilizadas para o gerenciamento da cadeia de suprimento (*supply chain*), por exemplo.

Alguns autores discutem qual a fonte ideal para a aquisição de dados para o PIMS, se esta deve ser o sistema supervisório ou os CLPs. Segundo [Seixas Filho \(2005\)](#), são vantagens de cada opção:

- CLPs:
 - Busca de eventos com menor atraso. Neste caso, a *timestamp* será mais precisa.
 - Caso todos os CLPs sejam do mesmo fabricante, é possível coletar os dados em um ponto único, desde que haja uma rede.
 - CLPs são menos suscetíveis a falhas, em comparação ao sistema SCADA.
 - Os CLPs são mais estáveis, uma vez que raramente sofrem *updates*.
- SCADA:
 - Os dados no sistema SCADA já estão em unidades de engenharia, enquanto que em CLPs antigos os dados estão brutos (0 - 4095).
 - É possível que algumas variáveis estejam definidas apenas no SCADA (por exemplo: um parâmetro de entrada manual no SCADA).

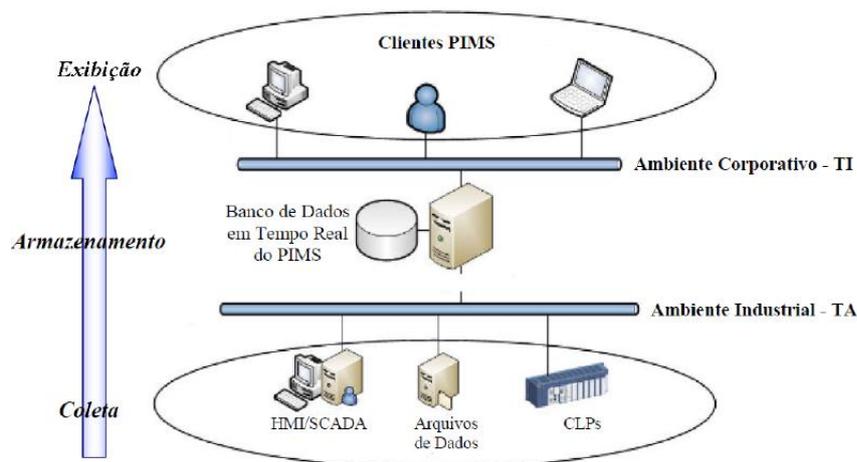


Figura 10 – Arquitetura PIMS (URBANO, 2009 apud JESUS; RAMOS; GUEDES, 2012)

5.3 PI SYSTEM

O *PI System* (*Plant Information System*), desenvolvido pela *OSIsoft*¹, é um conjunto de *softwares* que formam um sistema para o monitoramento, análise e historiamento de informações provenientes de uma planta industrial, sendo assim classificado como uma plataforma PIMS. Esta é uma ferramenta utilizada em diversas plantas industriais.

¹ <<http://www.osisoft.com/>>

5.3.1 PI-UDS

A base do sistema PI é o PI-UDS (*PI - Universal Data Server*), que opera como um servidor de dados, sendo capaz de guardar e recuperar informações armazenadas no repositório de dados. Além disso, o UDS é capaz de se comunicar com outros sistemas através dos *softwares* conhecidos como *COM Connectors*. O PI-UDS é composto por diversos subsistemas, descritos no anexo A (MENDES, 2010; SCHEUER, 2004).

5.3.2 NÓS DE COLETAS - PI-API

Segundo Scheuer (2004), o PI-API é uma biblioteca de funções que podem ser utilizadas em algoritmos escritos em C, C++, Visual Basic, Delphi, entre outros. As funções possibilitam a leitura e escrita de valores no servidor PI e o acesso às configurações das variáveis acompanhadas. As aplicações clientes do PI são desenvolvidas com o suporte do PI-API, PI-SDK (*Software Development Kit*) ou uma combinação de ambos.

Os nós de coleta são os elementos responsáveis por receber os dados provenientes de uma fonte de dados e enviá-los ao PI-UDS. É nos nós que são executados os algoritmos de exceção (descritos na Seção 5.3.3) que filtram os dados que deverão ser enviados ao PI-UDS. As aplicações clientes, por sua vez, extraem os dados armazenados neste último servidor. Um esquema do funcionamento deste sistema pode ser visto na Figura 11.

Através das aplicações clientes, os usuários têm acesso aos dados armazenados no servidor PI. Os principais clientes são (SCHEUER, 2004; CYBERTÉCNICA, 2014):

- PI-ProcessBook: Exibe os dados armazenados no banco de dados. Com ele é possível elaborar gráficos, tabelas e telas para acompanhamento da planta em tempo real. Com o *Visual Basic for Applications* (VBA) pode-se criar *scripts* para automatizar rotinas. O *Processbook* é integrado com diversos programas de *desktop*, sendo possível, por exemplo, enviar visualizações para outras pessoas através de e-mail. É totalmente compatível com o *Microsoft Office*, permitindo que o usuário copie e cole itens entre o *ProcessBook* e aplicações do *Office*. Maiores detalhes sobre este cliente podem ser encontrados na Seção 6.
- PI-DataLink: É o *Add-in* para o *Microsoft Excel* que permite comunicar bidirecionalmente dados entre este *software* e o PI-UDS, possibilitando a criação de planilhas e relatórios. É uma importante ferramenta para reunir e analisar os dados do PI.
- PI-ActiveView: Programa para visualização de dados históricos e em tempo real. Com o uso em conjunto do *ProcessBook*, é possível desenvolver aplicações para divulgação de dados na internet e em rede corporativa.
- PI-ProfileView: Esta aplicação é tipicamente utilizada em indústrias que produzem materiais bidimensionais como papel, mas também é usado para monitorar dados

razoavelmente constantes como a temperatura de uma caldeira. O *ProfileView* é capaz de verificar até 4000 pontos no plano para detectar inconsistências.

- **PI-Control Monitor:** Monitora o sistema de controle da planta, garantindo a sua precisão, além de armazenar informações sobre este. O *Control Monitor* observa os dados dos *loops* de controle e os envia para o *ProcessBook*, possibilitando que o operador possa checar o estado atual do sistema de controle.
- **PI-Manual Logger:** Este módulo é usado para permitir a inserção de dados manualmente no PI, através de dispositivos portáteis, computadores, *scanners* e outros meios de entrada. Desta maneira é possível se beneficiar das ferramentas do PI com dados que não estão sob controle automático.
- **PI-AlarmView:** Esta aplicação é responsável por notificar a ocorrência de um evento fora dos parâmetros esperados. Ela resume as informações do servidor *PI Alarm* e as exibe em forma de árvore hierárquica em aplicações clientes.

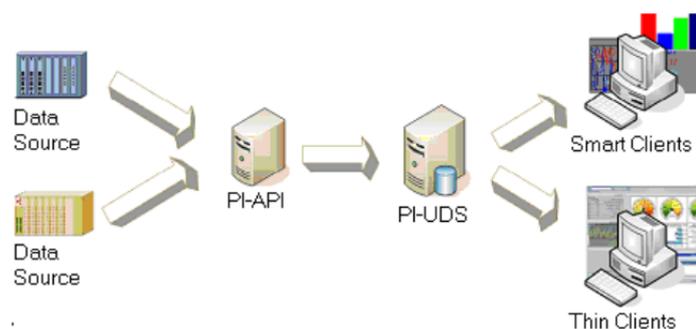


Figura 11 – Diagrama de funcionamento PI-API x PI-UDS (OSISOFT, 2009 apud MENDES, 2010)

5.3.3 ALGORITMOS DE COMPRESSÃO E EXCEÇÃO

A grande capacidade de armazenamento do PI se deve ao uso de algoritmos de compressão e exceção. A lógica utilizada nestes programas resulta em alta taxa de compressão, simples reconstrução dos dados e alta velocidade na execução da compressão/reconstrução. Os valores são armazenados apenas quando variam significativamente, assim é possível acumular uma reduzida quantidade de dados, sem prejuízo na qualidade da informação. Basicamente, a diferença entre estes algoritmos é que o de exceção determina quais dados serão enviados ao PI-UDS enquanto o de compressão define quais eventos serão armazenados.

A lógica utilizada no algoritmo de exceção executa os seguintes passos para verificar se o novo valor deve enviado ao servidor: (SCHEUER, 2004)

- Diferença entre o novo e o último valor é superior à banda morta (*ExcDev - Exception Deviation Specification*)? Além disso, a diferença de tempo entre eles é superior ao mínimo especificado (*ExcMin - Minimum Exception Time*)?²
- A diferença entre o novo e o último valor excede o tempo máximo definido (*ExcMax - Maximum Exception Time*)?

Caso pelo menos uma destas condições seja atendida, o novo valor será enviado ao PI-UDS, desta forma descartam-se dados sem relevância, como pode ser visto na Figura 12.

O *Swinging Doors Compression* (algoritmo de compressão do PI) é executado pelo próprio servidor do PI. O seu objetivo é reduzir ainda mais a quantidade de informação suficiente para reproduzir o cenário original com fidelidade. Como pode ser observado na Figura 13, uma quantidade significativamente menor de informação (em relação aos dados amostrados) é necessária para reconstruir de forma aproximada os dados originais de processo (MENDES, 2010).

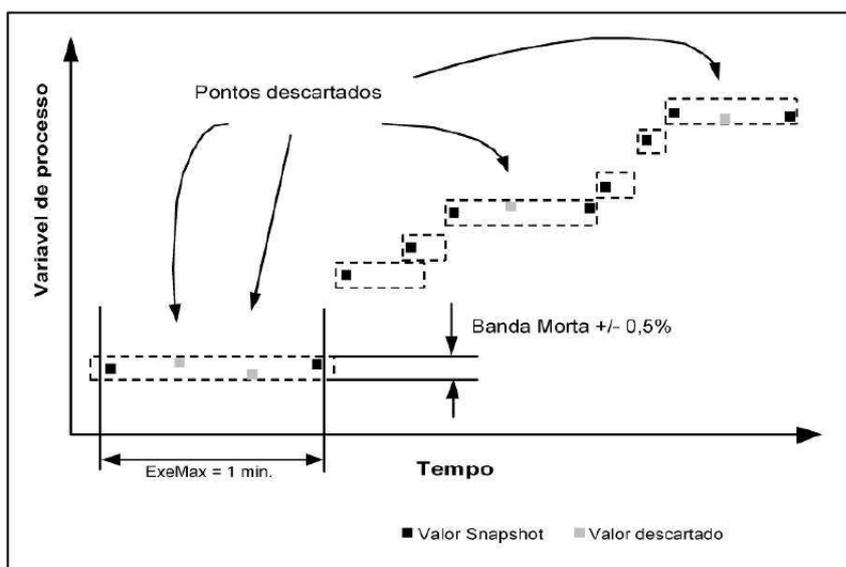


Figura 12 – Exemplo de funcionamento do algoritmo de exceção (BARRETO, 2009 apud MENDES, 2010)

5.3.4 Tags

A *tag* é um rótulo utilizado para associar uma variável de processo no PI, representando o seu valor medido ou calculado. Cada uma destas *tags* é única no sistema e deve ser escolhida de forma a representar fluxos individuais de dados, como a vazão em uma tubulação ou o nível de uma torre. Cada *tag* possui uma série de atributos que caracterizam aquele dado para o sistema, definindo, por exemplo, como ele será exibido,

² Este segundo critério é utilizado para eliminar sinais ruidosos.

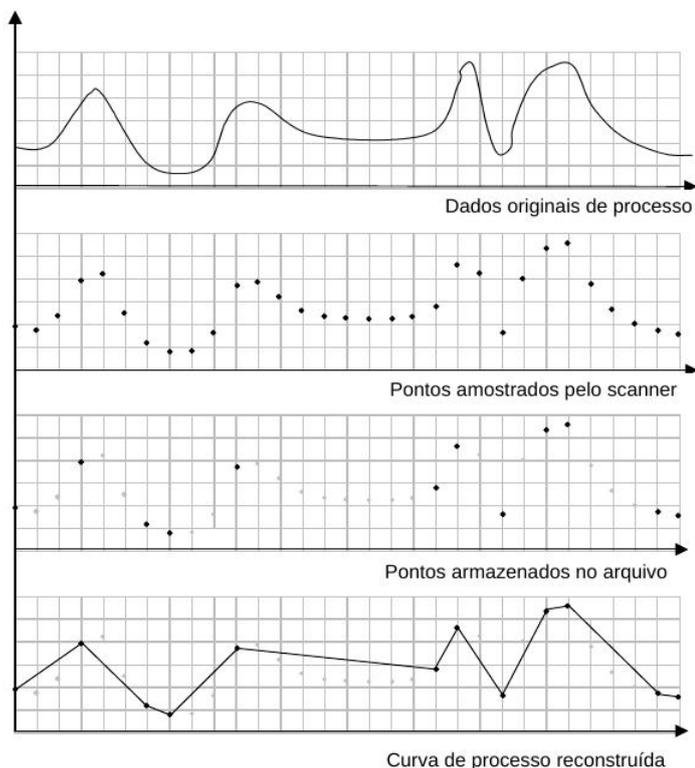


Figura 13 – Reconstrução dos dados originais de processo (SEIXAS FILHO, 2005)

manipulado, acessado e armazenado. Um resumo dos principais atributos pode ser visto no Anexo B.

5.3.5 Fluxo de dados

O fluxo de dados no PI pode ser visto na Figura 14. Os valores recebidos pelos nós de coleta, de acordo com os atributos das suas *tags*, passam pelo teste de exceção. Os que passarem por esta etapa se tornam um valor de *snapshot* e passam, desta vez, pelo teste de compressão (ainda de acordo com os atributos previamente estabelecidos). Apenas após passar por estes dois teste a informação é armazenada no *PI-Archive* (SCHEUER, 2004).

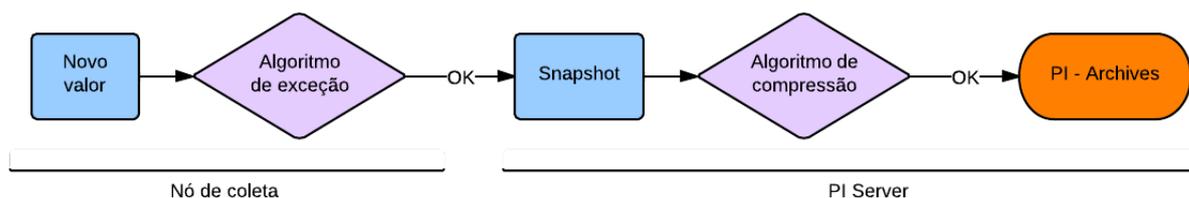


Figura 14 – Fluxo de dados no PI

6 PROCESSBOOK

O *PI ProcessBook* é um *software* que permite a criação de telas gráficas dinâmicas e interativas com os dados fornecidos pelo *PI Server* em tempo real. Através dele valores primários, coletados pelos sensores, são transformados em informações úteis, capazes de facilitar a compreensão e análise do que se passa na planta. Com o *Visual Basic for Applications Microsoft* (VBA) podem ser escritos *scripts* para automação de tarefas como a visualização ou avisos de alarme. O *ProcessBook* fornece acesso a banco de dados ODBC e é totalmente compatível com o pacote *Office* da *Microsoft* (CYBERTÉCNICA, 2014). A tela inicial do *ProcessBook* pode ser vista na Figura 15.

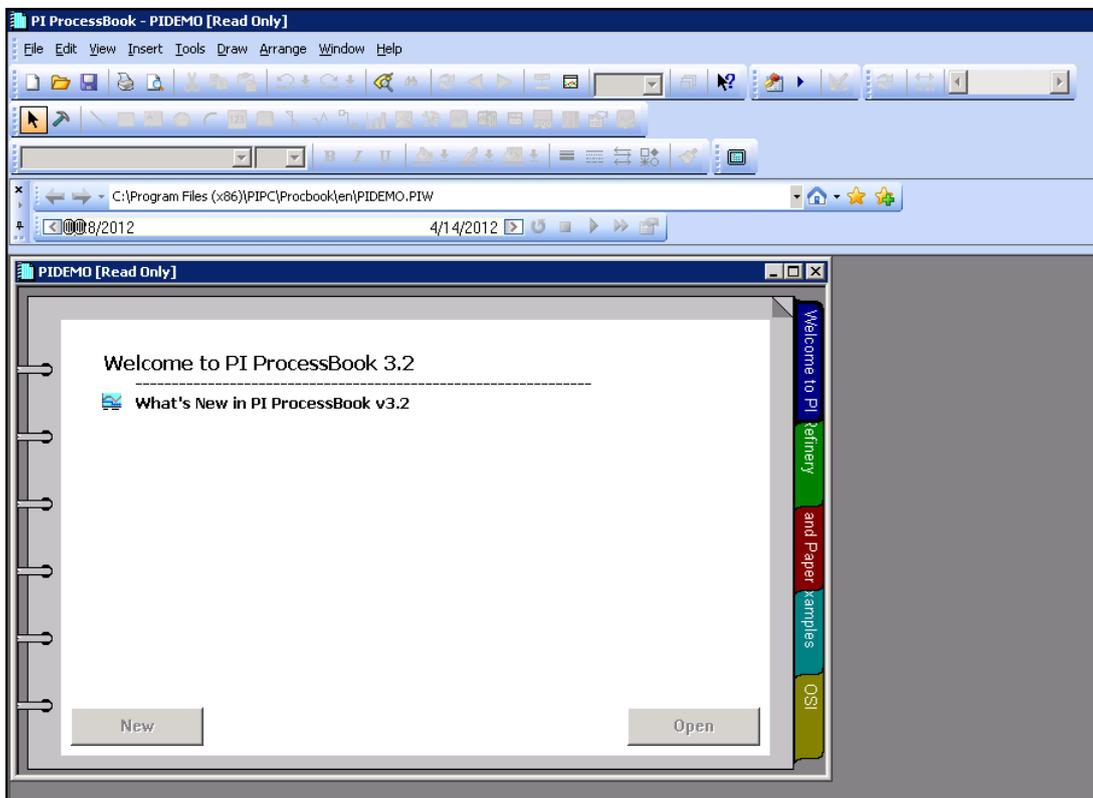


Figura 15 – Tela inicial do Process Book

6.1 CONEXÃO COM O PI SERVER

Através do *PI Connection Manager* (Figura 16) é possível conectar o *ProcessBook* aos servidores PI presentes naquela rede. Esta aplicação é utilizada para habilitar a comunicação da maioria dos clientes. Para tal, é necessário apenas conhecer o endereço IP (ou o *Host Name*) e nome e senha do usuário. No *ProcessBook*, o *Connection Manager* pode ser encontrado em *File > Connections*.

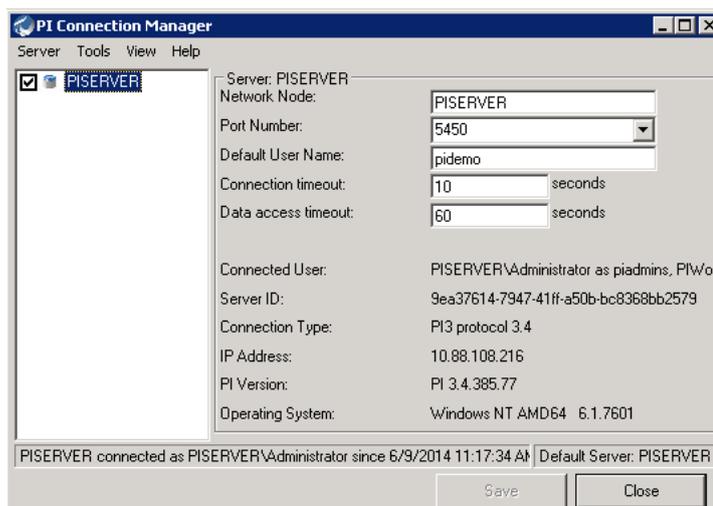


Figura 16 – PI Connection Manager

Todos as conexões que, em algum momento, foram realizadas naquele computador cliente são armazenadas no *Known Servers Table* (KST). Muitas vezes este registro contém entradas duplicadas ou incorretas. Para corrigir estes dados utiliza-se o *About PI-SDK*. Através de *Tools > Cleanup Connections* é possível resolver estas incoerências, como pode ser visto na Figura 17.

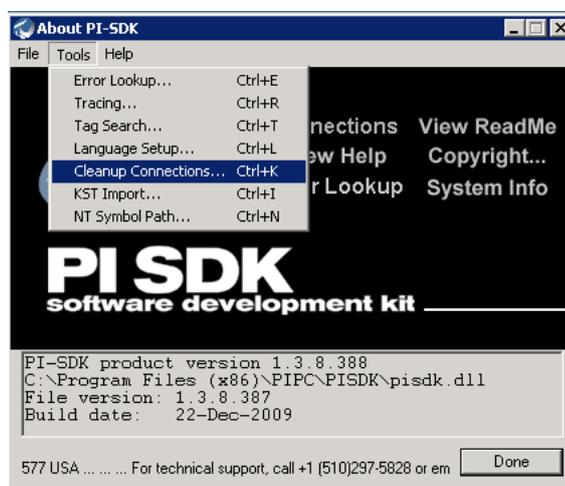


Figura 17 – About PI-SDK

6.2 TAGS

Os principais meios de criação de TAGs são através do *PI Tag Configurator for Excel* (uma extensão para o *Microsoft Excel*) ou o *PI SMT Console*, acessado pelo *PI System Management Tools* (Points > Point Builder), como pode ser visto na Figura 18. Caso opte-se pela criação a partir do *Excel*, deve-se criar uma planilha onde a primeira

linha será preenchida com os nomes dos atributos¹ que se deseja editar, aqueles que não forem incluídos ou estiverem sem nenhum valor, permanecerão na configuração padrão daquele atributo (ver Anexo B). A partir daí, através do *PI-SMT > Export Tags*, as TAGs podem ser exportadas para o *PI Server*, como pode ser visto na Figura 19. De forma análoga, estas podem ser importadas para o Excel para edição em *PI-SMT > Import Tags* (OSISOFT, 2014).

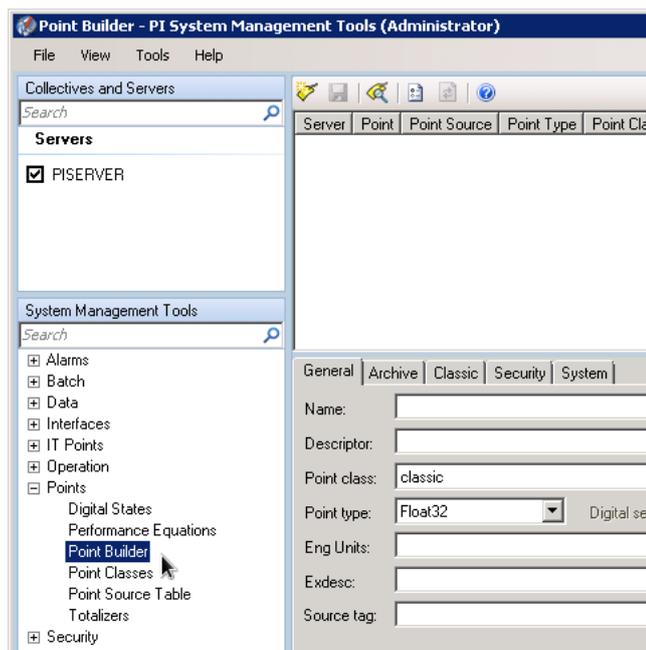


Figura 18 – Criação de Tags através do System Management Tools

A busca de TAGs é uma das funções mais utilizadas no *ProcessBook*. É possível encontrar o *Tag Search* em *Tools > Tag Search*. A busca geralmente é realizada utilizando-se o nome da TAG, a sua descrição ou pelo nó de coleta. Entretanto é possível refinar a pesquisa com outros critérios e caracteres especiais (como "*" e "?", por exemplo) como pode ser visto na Figura 20. O aplicativo *Tag Search* pode ser acessado de outros modos, como quando se deseja inserir um valor numa tela. Uma vez encontrada a TAG desejada é possível obter-se todos os seus atributos clicando no botão *Pt. Attr.*

6.2.1 Performance Equations TAGs

Performance Equations é um tipo especial de TAG no *PI Server* que armazena valores calculados a partir de dados fornecidos por outras TAGs. As equações são desenvolvidas utilizando a seguinte convenção:

- *Números e operadores*: São escritos utilizando a forma convencional, como: 10, 12.5, +, -, *, / e ^ ;

¹ A primeira coluna deve ser reservada para que se possa marcar com um "x" quais as TAGs devem ser exportadas.

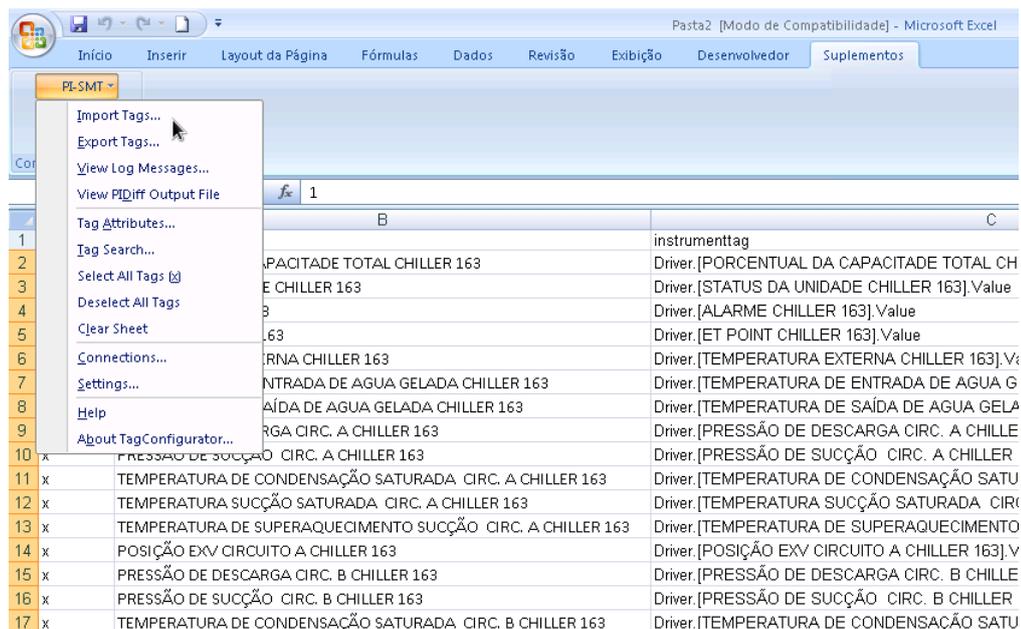


Figura 19 – Criação de Tags através do Excel

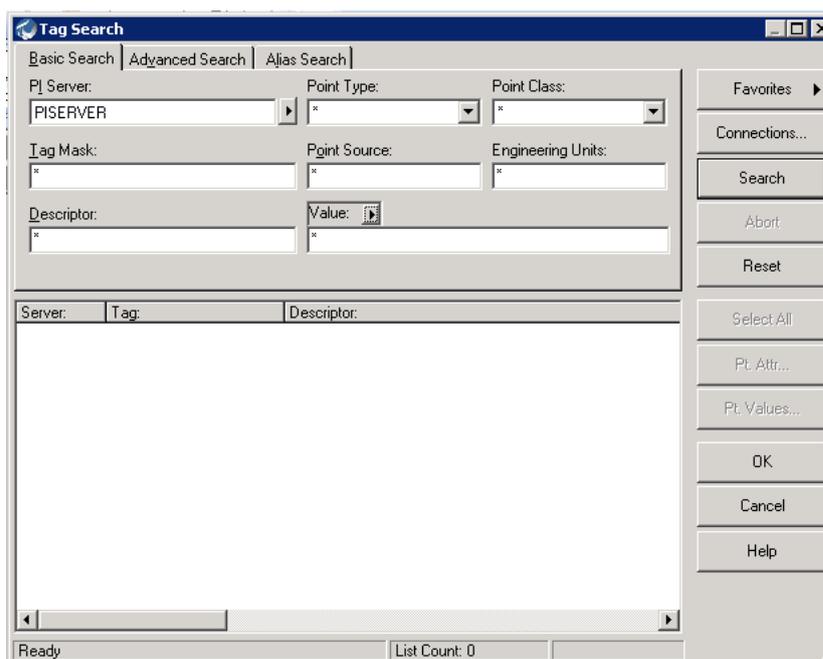


Figura 20 – Tag Search

- *TAGs e Timestamps* (ver Seção 6.2.2): Estes valores, isolados ou combinados, são escritos entre aspas simples, como em: 'tanqueA', 'TI012', '31-Ago-2013' e 'T+6h';
- *Strings*: São escritas entre aspas.

Além disso as *Performance Equations TAGs* possuem uma biblioteca de funções matemáticas que podem ser utilizadas. Há funções básicas como seno, cosseno, logaritmo,

etc. Há métodos próprios do PI aplicáveis especificamente a este tipo de dado e expressões do tipo *IF-THEN-ELSE*, assim como operadores lógicos. São exemplos destes métodos:

- *PctGood()*: Percentual do tempo que aquela TAG permaneceu em valores aceitáveis;
- *Range()*: Diferença entre o valor máximo e mínimo;
- *StDev()*: Desvio padrão baseado no tempo;
- *TagAvg()*: Média baseada no tempo;
- *TagMean()*: Média baseada na quantidade de eventos;
- *TagMax()*: Valor máximo;
- *TagMin()*: Valor mínimo;
- *TagTot()*: Soma de todos os valores.

6.2.2 PI Time

As abreviações descritas na Tabela 1 permitem especificar momentos no tempo através de expressões aplicáveis para a exibição de valores de variáveis em objetos dinâmicos. Dessa forma é possível montar expressões como "t - 5d", representando o período de tempo da zero hora do dia atual até cinco dias atrás. Os tipos de expressão estão categorizados em três grupos:

- *Tempo relativo*: Representa um tempo relativo a uma referência, como "-3m"(regressando três minutos) ou "+1d"(acrescentando um dia);
- *Tempo absoluto*: Faz referência a um momento específico, como "31-Dec-13"ou "t";
- *Tempo combinado*: Utiliza ambos os tipos, como em "* - 8h"(momento atual até oito horas no passado) ou "31-Dec-13 + 4d"(zero hora do dia trinta e um de dezembro de dois mil e treze mais quatro dias).

Também é possível utilizar *strings* como nos exemplos da Tabela 2.

6.3 TELAS GRÁFICAS

Primeiramente é necessário distinguir os modos de operação do *ProcessBook*. O primeiro destes é o *Run Mode*, que é o modo padrão e permite a navegação no modo de operação, utilizando os elementos disponíveis na tela. Também é possível navegar no *Build Mode*, quando deseja-se alterar os elementos (como gráficos, imagens, valores, entre outros)

NOME	ABREVIACÃO
Segundo	s
Minuto	m
Hora	h
Dia	d
Mês	mo
Ano	y
Semana	w
Dia da semana	wd
Dia do ano	yd
Momento atual	*
Hoje (0h)	t

Tabela 1 – Resumo das abreviações do PI Time

STRING	SIGNIFICADO
10	Zero hora do dia dez do mês atual
10:	Dez horas do dia atual
10 10:	Dez horas do dia dez do mês atual
sun, mon, tue, wed, thu, fri, sat	Zero hora de um dia da semana (domingo a segunda)
1.5h	Uma hora e meia

Tabela 2 – Resumo das Strings do PI Time

e suas propriedades. A transição entre os dois modos pode ser realizada através de *Tools* > *Run* ou *Build*, respectivamente.

Para uma nova tela há três opções em *File* > *New ProcessBook File*, *ProcessBook Entry* e *ProcessBook Display File*. A primeira corresponde à criação de um novo *Book*, uma espécie de sumário, onde várias telas podem ser organizadas, como pode ser visto na Figura 21. A segunda refere-se às entradas no *Book*, como "*PI Calculated Data Sets*" ou "*My display*" no exemplo da Figura 21 e suas respectivas telas. A última opção cria apenas uma tela, sem associação com nenhum *Book*.

Os elementos podem ser adicionados à tela através dos atalhos, na barra de tarefa, ou no *menu Draw*. Para adicionar objetos como gráficos ou imagens é necessário arrastar o *mouse* na área onde deseja-se adicionar o item, além de habilitar o *Build Mode*. As opções podem ser vistas na Figura 22. É possível incluir formas, imagens, valores numéricos (de uma TAG, por exemplo), gráficos, textos, tabelas e outros objetos.

Os valores (*Value*) serão associados às variáveis de processo através suas TAGs encontradas no *Tag Search*. Uma vez selecionada a opção *Value*, a janela *Define Value* surgirá e será possível acessar o *Tag Search*, como pode ser visto na Figura 23. No modo *Run*, ao pousar o *mouse* sobre um valor, o *ProcessBook* exibirá algumas estatísticas sobre aquela variável, como máximo, mínimo e média. A definição dos valores que devem ser exibidos pode ser acessada através de *Tools* > *ToolTip Statistics* Também é possível ver

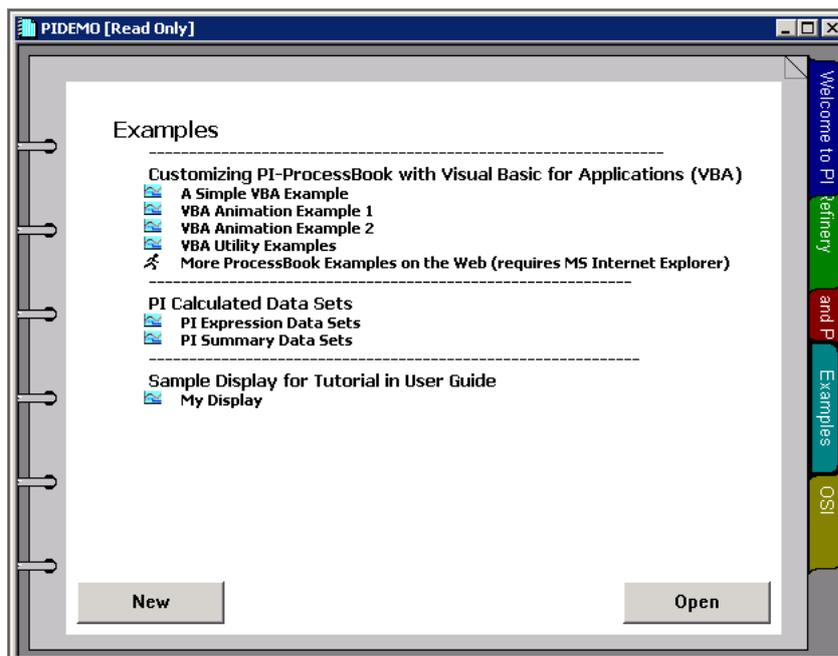


Figura 21 – Novo ProcessBook

e alterar outros detalhes de um elemento em *View > Details*. Desta forma um usuário pode adicionar uma observação a um valor específico e esta será armazenada no *PI Server*, estando disponível a outros usuários. Estes detalhes podem ser exportados para um arquivo *.csv* ainda nesta aba *Details*.

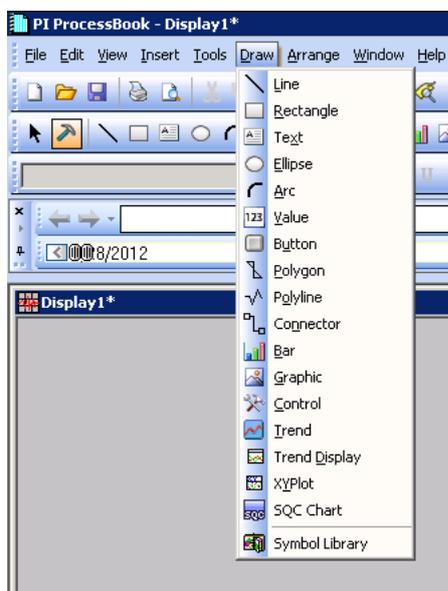


Figura 22 – Objetos do ProcessBook

Os objetos dinâmicos são aqueles que estão conectados a uma variável e que são atualizados em tempo real. Podem ser utilizados no *ProcessBook* os seguintes objetos dinâmicos:

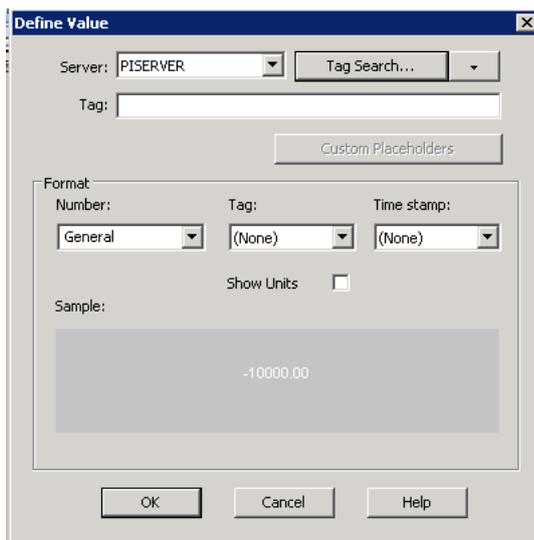


Figura 23 – Exibir valor no ProcessBook

- *Trend*: Gráfico de tendência, mostrando o comportamento de uma variável ao longo do tempo;
- *XY Plots*: Demonstra a relação entre duas variáveis, indicando o seu coeficiente de correlação;
- *Bar Graphs*: Cria um gráfico de barra delimitado por valores de máximo e mínimo definidos pelo usuário;
- *Values*: Valores obtidos a partir da leitura de TAGs no banco de dados;
- *Button*: Cria um botão associado à execução de uma macro ou arquivo do sistema;
- *Multi-State Objects*: São símbolos ou imagens que mudam alguma característica em função do valor de uma TAG associada.

6.3.1 Time Range and Playback

A barra *Time Range and Playback* (Figura 24) permite que o usuário inicie, pare, defina um *loop* ou navegue manualmente num determinado período de tempo em qualquer elemento dinâmico. Esta ferramenta é útil, principalmente, para exibir das condições de processos relacionadas a um período ou evento específico, facilitando a análise.



Figura 24 – Barra Time Range and Playback

7 RESULTADOS

Este estudo de caso propôs-se a implementar um sistema de publicação de dados utilizando uma plataforma PIMS, neste caso, o PI da *OsiSoft*. Para suportar o desenvolvimento deste trabalho e criar um ambiente capaz de simular uma aplicação industrial, foi necessário construir um protótipo, de pequena escala, de uma Unidade de Bombeio Mecânico (UBM), ver Figura 27. Neste capítulo serão descritas as etapas necessárias para alcançar os objetivos deste estudo.

7.1 UNIDADE DE BOMBEIO MECÂNICO

O Bombeio Mecânico foi o primeiro método de elevação artificial a ser desenvolvido pela indústria do petróleo e, ainda hoje, é o mais utilizado no mundo, como pode ser visto na Figura 25. As principais vantagens deste método são a sua simplicidade de operação, manutenção e instalação, além da vantajosa relação custo/produção durante o seu tempo de operação (NASCIMENTO, 2005).

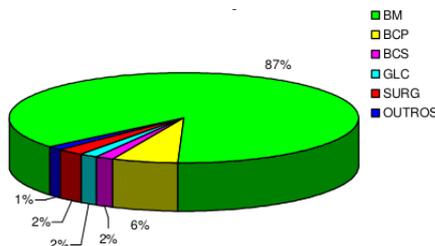


Figura 25 – Principais métodos de elevação artificial (NASCIMENTO, 2005)

Os componentes de uma UBM podem ser divididos em superfície e fundo (NASCIMENTO, 2005):

- **Superfície:** Motor, caixa de redução, unidade de bombeio e haste polida;
- **Fundo:** Coluna de hastes e bomba de fundo.

O motor e caixa de redução, através da unidade de bombeio, movimentam o conjunto de hastes. Na sua porção inferior localiza-se a bomba de fundo, composta por duas válvulas, denominadas *Válvula de Passeio* e *Válvula de Pé*. O acionamento destas válvulas provoca a elevação do fluido. Um esquemático com os principais componentes pode ser visto na Figura 26.

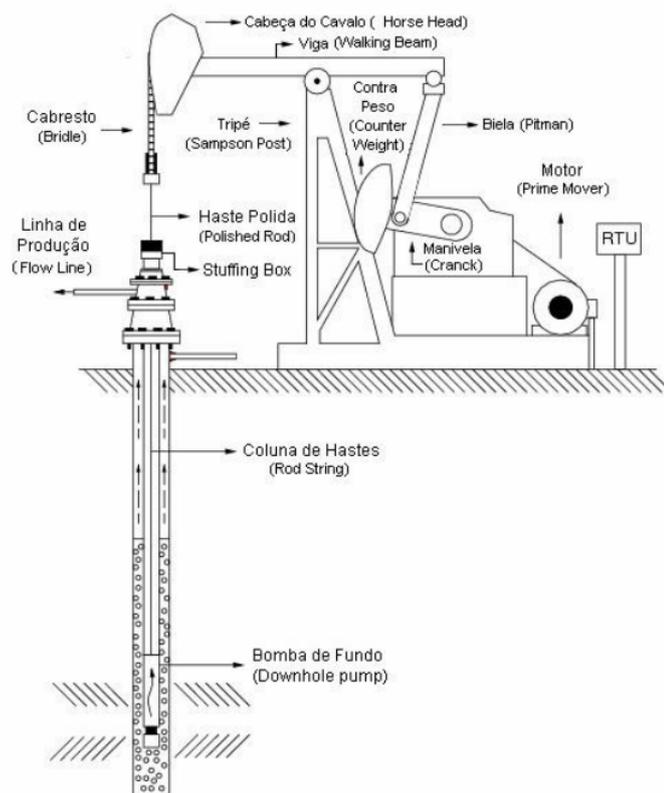


Figura 26 – Principais componente de uma UBM (NASCIMENTO, 2005)

7.1.1 Protótipo

7.1.1.1 Estrutura mecânica

A estrutura mecânica desenvolvida pode ser vista na Figura 27.

7.1.1.2 Sensores e Atuadores

Após a definição de quais variáveis deveriam ser acompanhadas (nível do fluido, posição da unidade, velocidade do bombeio, produção e carga na haste), foi determinada a instrumentação necessária, desta forma o protótipo conta com os seguintes sensores e atuadores:

- **Sensores:**

1. *Sensor de nível:* Através de um sensor de pressão diferencial, obtém-se o nível dinâmico de fluido dentro da poço;
2. *Potenciômetro:* Anexado ao balancim, o potenciômetro tem a função de determinar a posição atual da unidade de bombeio (UB);
3. *Célula de carga:* Este sensor é utilizado para determinar a carga a qual a haste está sujeita. Em conjunto com a posição atual, esta variável é utilizada para analisar a condição de operação da UB.



Figura 27 – Estrutura mecânica completa

- **Atuadores:**

1. *Motor elétrico*: Fornece potência à unidade;
2. *Bomba*: Responsável por alimentar o poço.

7.2 CLP E IHM

O controle da unidade foi feito utilizando-se um CLP da *Allen Bradley*, da *Rockwell Automation*, da série CompactLogix 1769-L32E. Os códigos utilizados podem ser encontrados no Anexo C.

Para a operação e realização de testes da UBM desenvolveu-se um sistema supervi-sório com o *software FactoryTalk*¹, também da *Rockwell*. A tela principal pode ser vista na Figura 28.

7.3 CONFIGURAÇÃO DA INTERFACE OPC

O primeiro passo para a implementação de controle no protótipo foi a programação do CLP. Nesta etapa foram criadas, no controlador, todas as TAGs necessárias para a

¹ <<http://www.rockwellautomation.com/pt/rockwellsoftware/factorytalk/overview.page>>

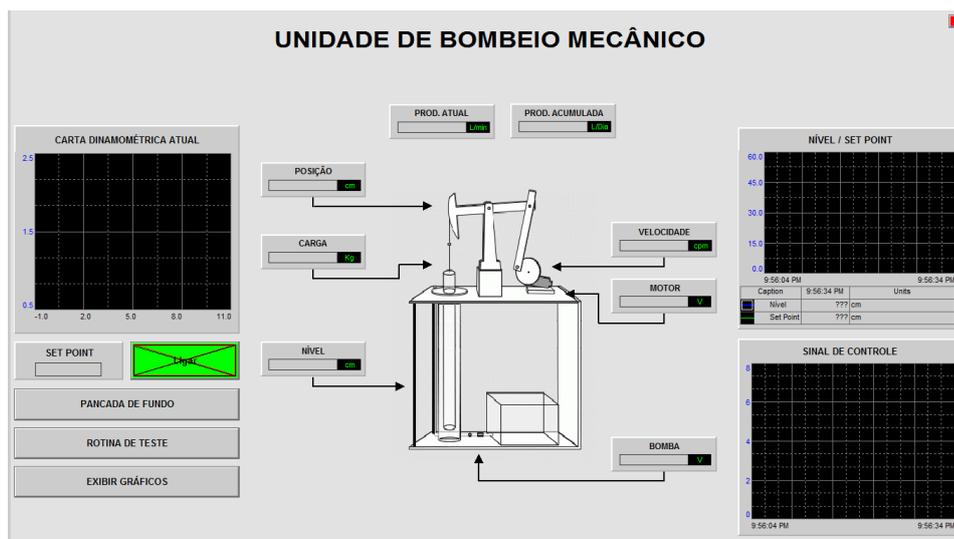


Figura 28 – Tela inicial do sistema supervisório

ITEM	DESCRIÇÃO
Servidor	Servidor para o PI Server
CLP	CLP Allen Bradley CompactLogix 1769-L32E
RSLinx Classic	Programa para estabelecer comunicação com o CLP
RSLogix 5000	Software para programação do CLP
FactoryTalk View Studio	Programa para desenvolvimento do supervisório
PI-ICU	Software para configuração da interface de comunicação
PI OPC Client	Programa importação das TAGs utilizando OPC
PI ProcessBook	Programa para a criação das telas do PIMS
PI-SDK	Programa acesso ao PI
PI DataLink	Fornece, entre outras coisas, o <i>add-in</i> para Excel
PI-SMT	Utilizado para administrar o PI

Tabela 3 – Software e Hardware utilizados

operação da planta. Posteriormente foi desenvolvido o sistema supervisório para operação e realização de testes através de uma rotina específica. Neste processo foram criadas, desta vez no *FactoryTalk*, as TAGs de interesse que também seriam utilizadas no PIMS. Por conveniência, optou-se pela importação das TAGs do *FactoryTalk* para o servidor PI, evitando o retrabalho de criá-las novamente no PIMS. Uma vez que o servidor PI tenha sido configurado e as TAGs importadas, elas tornam-se disponíveis para todas as aplicações clientes do PI, como o *ProcessBook*.

Para que esta comunicação fosse possível foram utilizadas duas interfaces OPC distintas entre os diferentes sistemas do protótipo. Os dados colhidos pelos sensores são transmitidos ao CLP através das suas portas analógicas e deste para o sistema supervisório através do protocolo OPC. Neste caso o CLP comporta-se como *Servidor OPC* e o *FactoryTalk* como *Cliente OPC*. A segunda interface opera entre o supervisório e o PIMS, os valores das TAGs do *FactoryTalk* são enviados ao servidor PI e este disponibiliza estas

informações às aplicações clientes. Desta vez o *FactoryTalk* é o servidor e o PI é o cliente OPC.

Resume-se este esquema na Figura 29.

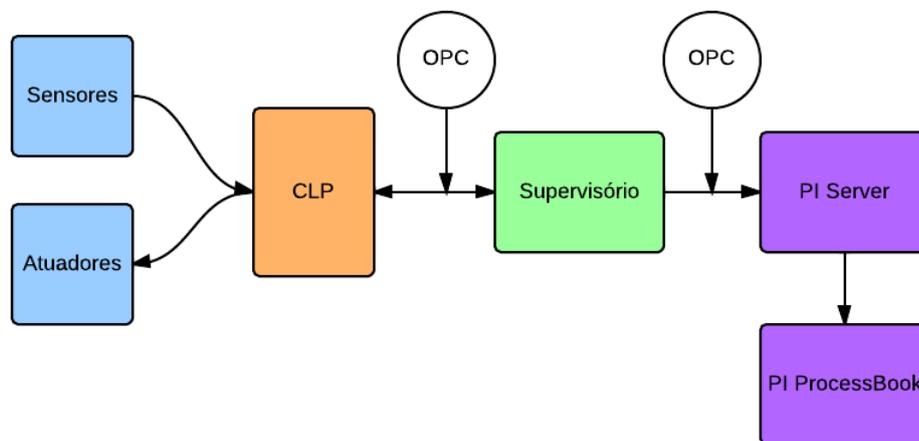


Figura 29 – Diagrama de comunicação do CLP com o PI Server

7.3.1 Configuração OPC no PI

A configuração para a criação e utilização de uma interface OPC para o PI seguiu, basicamente, os seguintes passos em cada uma das aplicações destacadas:

1. PI-ICU (Interface Configuration Utility)

- a) Cria-se uma nova interface OPC em *Interface > New Windows Interface Instance from BAT file... > opcint1.bat*;
- b) Define-se o nome do ponto de coleta (*Point Source*);
- c) Define-se o servidor PI a ser utilizado em *Server/Collective*;
- d) Em *Service*, adicionam-se as dependências necessárias, neste caso *tcpip* e *PI-Bufss*;
- e) Cria-se (*Create*) e inicia-se (*Play*) um novo serviço (Figura 30).

2. PI OPC Client

Como todas as TAGs já foram criadas no *FactoryTalk*, não há a necessidade de criá-las novamente no PI, bastando importá-las. Neste caso podem ser utilizados, em conjunto, o *PI OPC Client* e o *PI DataLink*, através do seu *add-on* no Excel.

- a) Listam-se as interfaces OPC em *Connect to Node* (primeiro ícone da primeira linha);

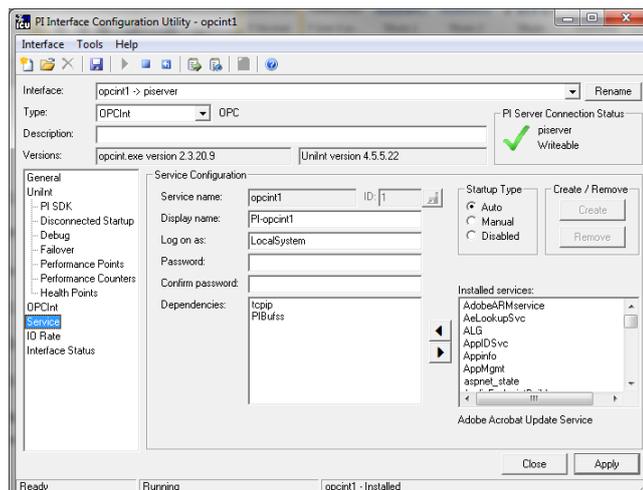


Figura 30 – Aba Service no PI-ICU

- b) Conecta-se (*Connect to OPC*, segundo ícone da primeira linha) à interface desejada, neste caso *FactoryTalk Gateway*;
- c) Cria-se um grupo em *Add Group* (sexto ícone da primeira linha);
- d) Listam-se as TAGs disponíveis naquele servidor em *Browse OPC Server*, *Add Tag(s)* (primeiro ícone da segunda linha);
- e) Selecionam-se as TAGs desejadas (*Select All > Add Selected*), ver Figura 31;
- f) Exporta as TAGs (Save As) no formato *.csv*.

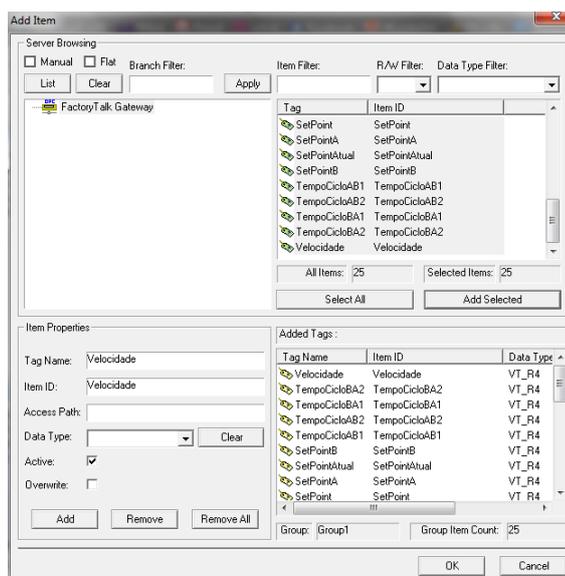


Figura 31 – Seleção de TAGs no PI OPC Client

3. Microsoft Excel - PI DataLink

- a) Abre-se, no Excel, o *.csv* criado na etapa anterior;

- b) Conecta-se ao servidor (*Connections*), ver Figura 32;
- c) Exportam-se as TAGs (*Export TAGs...*).

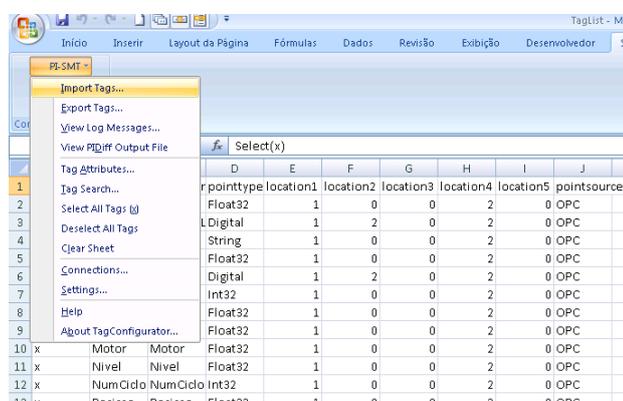


Figura 32 – Exportação das TAGs através do Excel

7.4 PI PROCESSBOOK

Uma vez exportadas as TAGs e estabelecida a comunicação OPC entre o PI e o *FactoryTalk*, foi possível desenvolver as telas gráficas desejadas. A tela principal pode ser vista na Figura 33.

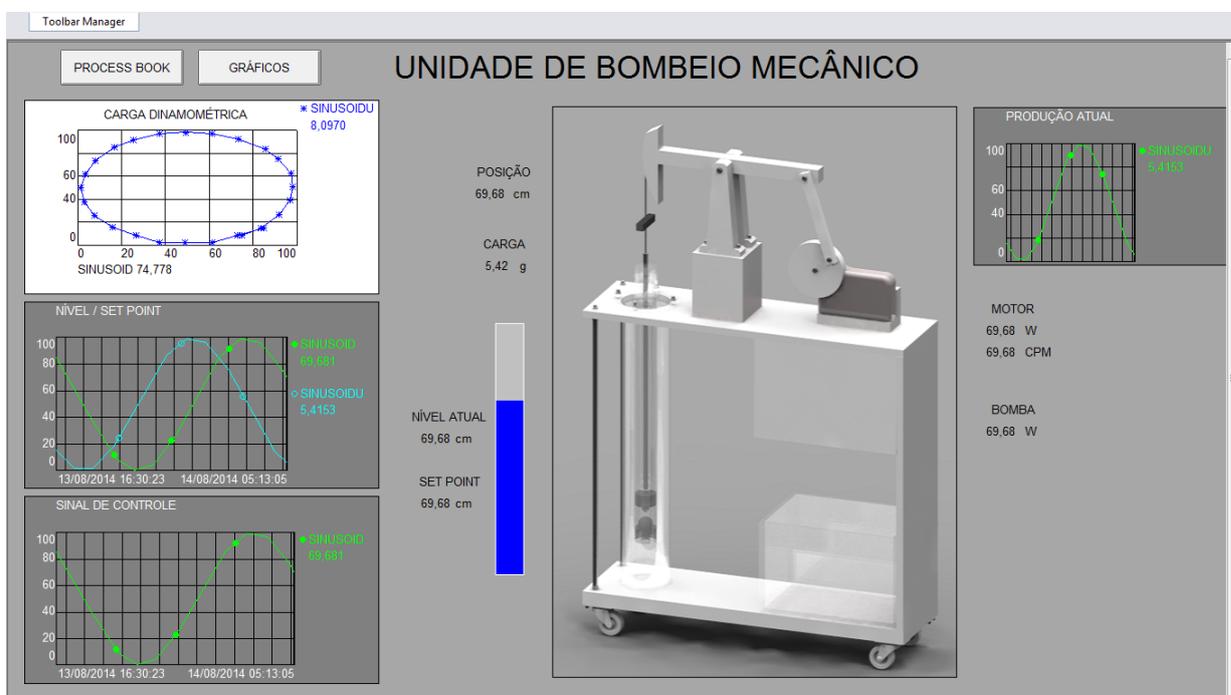


Figura 33 – Tela principal do ProcessBook

8 CONSIDERAÇÕES FINAIS

Neste trabalho foi possível analisar aspectos e benefícios da utilização de uma plataforma PIMS, como a economia de espaço em memória para o armazenamento de informação, segurança de acesso, interoperabilidade e a disponibilização de dados em tempo real. A revisão feita sobre os conceitos de automação industrial nos capítulos iniciais permitiu aprofundar conhecimentos nesta área e na implementação e uso de controle em plantas industriais com CLPs e sistemas supervisórios.

Apesar da simplicidade do protótipo, pôde-se observar questões de interoperabilidade através da utilização do OPC e as vantagens associadas ao seu uso. Este padrão permite simples comunicação entre diferentes sistemas, sendo muito importante em ambientes industriais complexos, onde coabitam diversos sistemas de fornecedores distintos. A aplicação deste padrão se mostrou muito simples e eficaz, tornando ágil e clara a configuração de Servidores e Clientes OPC. O estudo revelou que, mesmo com as claras vantagens apresentadas, a interface OPC não está disponível em todos os principais fornecedores, sendo uma forma de impor a aquisição de soluções completas e fechadas.

Não estava no escopo deste projeto implementar condições de acesso seguro, entretanto estas restrições poderiam ser facilmente adotadas no PI, através do PI-SMT. Neste mesmo *software* também é possível criar diferentes níveis de privilégio para que usuários tenham permissão distintas para intervenções no PI. O mesmo pode ser aplicado no *FactoryTalk* para a criação de restrições de acesso e interação em telas do supervisório.

A disponibilidade de informações em tempo real, agregando valor aos dados brutos do chão de fábrica, é cada vez mais importante para gestão de plantas industriais e para tomada de decisões estratégicas, impactando diretamente na competitividade do negócio. Com esta percepção, este trabalho buscou demonstrar fundamentos e instruções para o uso de uma das principais ferramentas de PIMS disponíveis no mercado.

Como possibilidade de estudos futuros, podem ser expandidos os resultados deste trabalho para plantas reais, de maior complexidade sem maiores modificações. Além disso, podem ser implementados acesso em tempo real aos dados via *web* com o *WebParts* e a criação de um banco de dados relacional, explorando ainda mais o potencial do sistema PIMS.

REFERÊNCIAS

- BARRETO, S. S. *Automação Integrada na Petrobras: Disponibilização de Informações de processo industrial em arquitetura aberta*. — Faculdade Ruy Barbosa, 2006. Citado na página 19.
- BARRETO, S. S. *Desenvolvimento de Metodologia para Atualização em Tempo Real de Modelos Matemáticos de Processos Decisórios*. — Universidade Federal da Bahia, 2009. Citado na página 34.
- BOJORGE, N. Notas de aula - Controle de Processos, *Introdução ao controle de processos*. 2012. Disponível em: <http://www.professores.uff.br/controldeprocessos-eq/images/stories/Control_Aula01_2sem2012.pdf>. Acesso em: Mar. 2014. Citado na página 17.
- CÂNDIDO, R. V. B. Padrão opc: Uma alternativa de substituição dos drivers proprietários para acessar dados de plcs. *Universidade FUMEC. Belo Horizonte*, 2004. Citado na página 24.
- CARVALHO, F. B. de et al. Sistemas pims - conceituação, usos e benefícios. *Tecnologia em Metalurgia e Materiais*, 2005. Citado nas páginas 28 e 29.
- CYBERTÉCNICA. *PI Server Applications (PI-ServerApps)*. 2014. Disponível em: <http://www.cybertecnica.com/article64_pt.htm>. Acesso em: Mar. 2014. Citado nas páginas 32 e 36.
- GAIDZINSKI, V. H. *A tecnologia da informação no chão de fábrica: As novas ferramentas e a gestão integrada da informação*. Dissertação de Mestrado — Universidade Federal de Santa Catarina, 2003. Citado na página 13.
- JESUS, T. C.; RAMOS, B. C.; GUEDES, G. P. Desenvolvimento de um sistema de aquisição, armazenamento e publicação de dados baseado em sistemas pims. *Congresso Brasileiro de Automática*, 2012. Citado nas páginas 27, 29 e 31.
- MENDES, A. da S. *Historiamento e Publicação de Dados em Tempo Real, Utilizando Arquitetura PIMS*. Monografia de Pós Graduação — Faculdade de Tecnologia SENAI CIMATEC, 2010. Citado nas páginas 19, 20, 32, 33, 34, 56 e 57.
- MORAES, C. C. D.; CASTRUCCI, P. B. de L. *Engenharia de automação industrial*. [S.l.]: Livros Tecnicos e Cientificos, 2001. Citado nas páginas 16, 18, 19, 20, 21 e 22.
- MOXA. *OPC Server in Industrial Automation*. 2014. Disponível em: <http://www.moxa.com/newsletter/connection/2009/07/OPC_Server_in_Industrial_Automation.htm>. Acesso em: 23 Jan. 2014. Citado na página 25.
- NASCIMENTO, J. M. A. d. *Simulador computacional para poços de petróleo com método de elevação artificial por bombeio mecânico*. Tese de Doutorado — Universidade Federal do Rio Grande do Norte, 2005. Citado nas páginas 44 e 45.
- OPC FOUNDATION. *OPC Foundation*. 2014. Disponível em: <<http://www.opcfoundation.org/>>. Acesso em: 22 Jan. 2014. Citado nas páginas 24, 25 e 26.

- OSISOFT. *Introduction to PI Server System Management*. 2009. Disponível em: <http://techsupport.osisoft.com/TechSupport/NonTemplates/download%20center/downloadcenter.aspx?download_content=User%20Manuals>. Acesso em: Jan. 2010. Citado na página 33.
- OSISOFT. *OSIssoftLearning*. 2014. Disponível em: <<https://www.youtube.com/user/OSIssoftLearning>>. Acesso em: 18 Jun. 2014. Citado na página 38.
- SCHEUER, A. Instalação e administração do sistema pi na unidade multipropósito de fcc. *Universidade Federal de Santa Catarina*, 2004. Citado nas páginas 32, 33, 35 e 55.
- SEIXAS FILHO, C. Pims - process information management system: Uma introdução. *Universidade Federal de Minas Gerais*, 2005. Citado nas páginas 27, 28, 30 e 35.
- SILVEIRA, P. R. da; SANTOS, W. E. dos. *Automação e controle discreto*. [S.l.]: Érica, 2004. Citado nas páginas 12, 13, 16, 17, 19, 20 e 21.
- URBANO, L. R. H. Gerenciamento de ativos de automação: Uma solução baseada em pims. *4o Seminário Nacional de Sistemas Industriais e Automação*, 2009. Citado na página 31.
- YOKOGAWA. *Exaquantum PIMS*. 2014. Disponível em: <http://www.yokogawa.com/eu/pims/Exaquantum/eu-exaquantum_pims.htm>. Acesso em: 10 Jan. 2014. Citado na página 22.

Anexos

ANEXO A – SUBSISTEMAS DO PI-UDS

SUBSISTEMA	DESCRIÇÃO
<i>Base Subsystem</i>	Cada variável de interesse possui um ponto (<i>tag</i>) definido no PI. Cada um destes pode possuir até 50 atributos que definem o modo da coleta e historiamento daquela informação. Este subsistema é responsável pelo armazenamento dos atributos e pela manutenção da tabela de estados digitais e configurações de segurança para usuários e grupos.
<i>Snapshot Subsystem</i>	Este subsistema recebe o último valor (<i>Snapshot</i>) de cada <i>tag</i> , após o algoritmo de exceção, e então aplica o algoritmo de compressão, que decidirá se este valor será armazenado no <i>Archive Subsystem</i> . Além disso, torna os dados disponíveis ao usuário quando solicitado.
<i>Archive Subsystem</i>	Programa responsável por armazenar os valores medidos em cada <i>tag</i> , formando um banco de dados temporal.
<i>Update Manager Subsystem</i>	Informa mudanças em valores ou atributos de uma <i>tag</i> para outras aplicações clientes.
<i>Message Subsystem</i>	Gerencia um arquivo de registros de controle, com mensagens de <i>status</i> e erros do PI-UDS.
<i>Conector COM Redirector</i>	É o <i>software</i> que realiza a interface entre o PI-UDS e outros sistemas.
<i>PI Network Manager</i>	É o subsistema responsável pela conexão entre os subsistemas do UDS e entre o servidor PI e aplicações clientes. É responsável também pela segurança do PI, com a validação de clientes na rede.
<i>SQL Subsystem</i>	Responsável por processar consultas SQL a bancos de dados relacionais.

Tabela 4 – Subsistemas do PI-UDS (SCHEUER, 2004)

ANEXO B – PRINCIPAIS ATRIBUTOS DE UMA TAG

ATRIBUTO	DESCRIÇÃO
<i>Tag</i>	Nome que a variável de processo receberá para ser armazenada no sistema PI.
<i>Compdev</i>	É um parâmetro utilizado para filtrar dados transmitidos através do <i>snapshot</i> , armazenando apenas mudanças significativas de valor.
<i>Compdevpercent</i>	É um parâmetro similar ao CompDev, porém o desvio é especificado em porcentagem.
<i>Compmax</i>	Neste atributo é especificado o tempo máximo para um novo evento ser armazenado no PI-UDS.
<i>Compmin</i>	Qualquer evento que venha antes do parâmetro Compmin será descartado do armazenamento ao PI-UDS.
<i>Compressing</i>	Este parâmetro é utilizado para ativar ou não a compressão dos dados no sistema, utiliza-se (1) para ativar e (0) para desativar.
<i>Descriptor</i>	Descrição do nome do <i>tag</i> . Geralmente útil para refinar os critérios de busca.
<i>DigitalSet</i>	Este atributo é utilizado para criar estados digitais ao <i>tag</i> . Ex.: "Ligado/Desligado" e "Aberto/Fechado".
<i>Engunits</i>	Unidade de engenharia do <i>tag</i> .
<i>Excdev</i>	O atributo Excdev especifica, em unidades de engenharia, quanto um valor pode ser diferente do valor anterior antes que ele seja considerado um valor significativo.
<i>Excdevpercent</i>	Este atributo é semelhante ao Excdev, porém sua especificação é feita em percentual e não em unidades de engenharia.
<i>Excmax</i>	O máximo período de tempo permitido para o envio de valores ao servidor PI-UDS.
<i>Excmin</i>	O mínimo período de tempo permitido para o envio de valores ao servidor PI-UDS.
<i>Instrumenttag</i>	Este campo é utilizado para criar uma associação da <i>tag</i> criada no PI, com a variável de processo cadastrada no controlador do sistema supervisor.

Tabela 5 – Principais atributos de uma *tag* (MENDES, 2010)

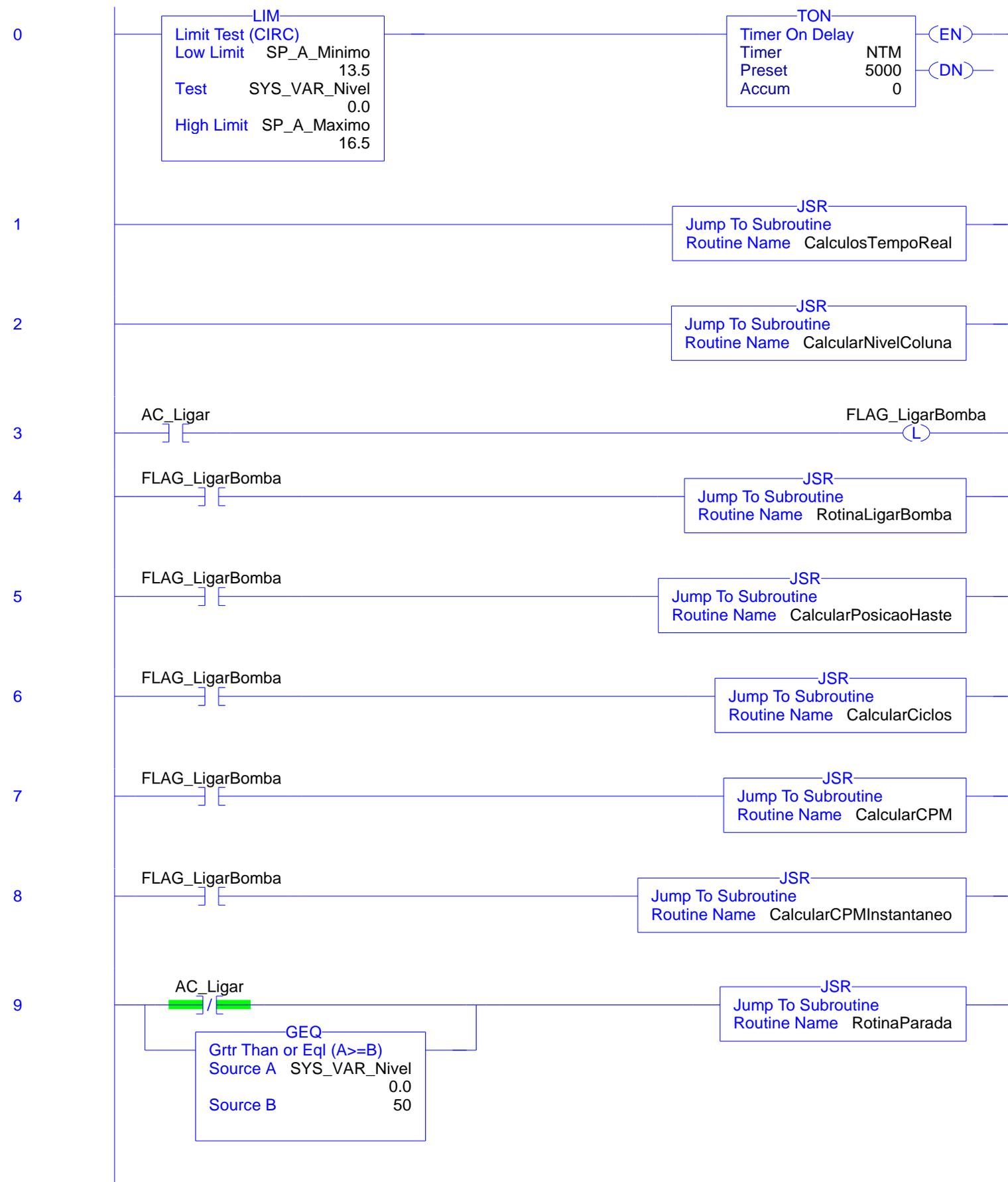
<i>Location1</i>	Este parâmetro indica a instância que o ponto pertence. O valor deste atributo deve corresponder ao "id" da tela principal de configuração da interface OPC Client, que poderá ser verificado através do PI-ICU (<i>PI-Interface Configuration Utility</i>).
<i>Location2</i>	Este parâmetro poderá receber os valores 0 (zero) ou 1 (um), onde 0 (zero) representa processamento normal, ou seja, um valor analógico do sistema, e 1 (um) representando um valor digital, e poderá receber tratamento dependendo do valor coletado.
<i>Location3</i>	Este campo é utilizado para indicar o tipo de coleta realizado: 0 - Evento; 1 - <i>Advise</i> ¹ ; 2 - <i>Output</i> ² .
<i>Location4</i>	Este parâmetro significa o tempo de varredura que o PI terá para armazenar o próximo <i>tag</i> , e está vinculado à classe de <i>scan</i> configurada na interface <i>OPC Client</i> .
<i>Point Type</i>	Tipo de dado do ponto. Poderá ser do tipo <i>digital</i> , <i>int16</i> , <i>int32</i> , <i>float16</i> , <i>float32</i> , <i>float64</i> , <i>string</i> e <i>blob</i> .
<i>Point Source</i>	É utilizado para identificar o ponto do PI em um nó de coleta de dados.

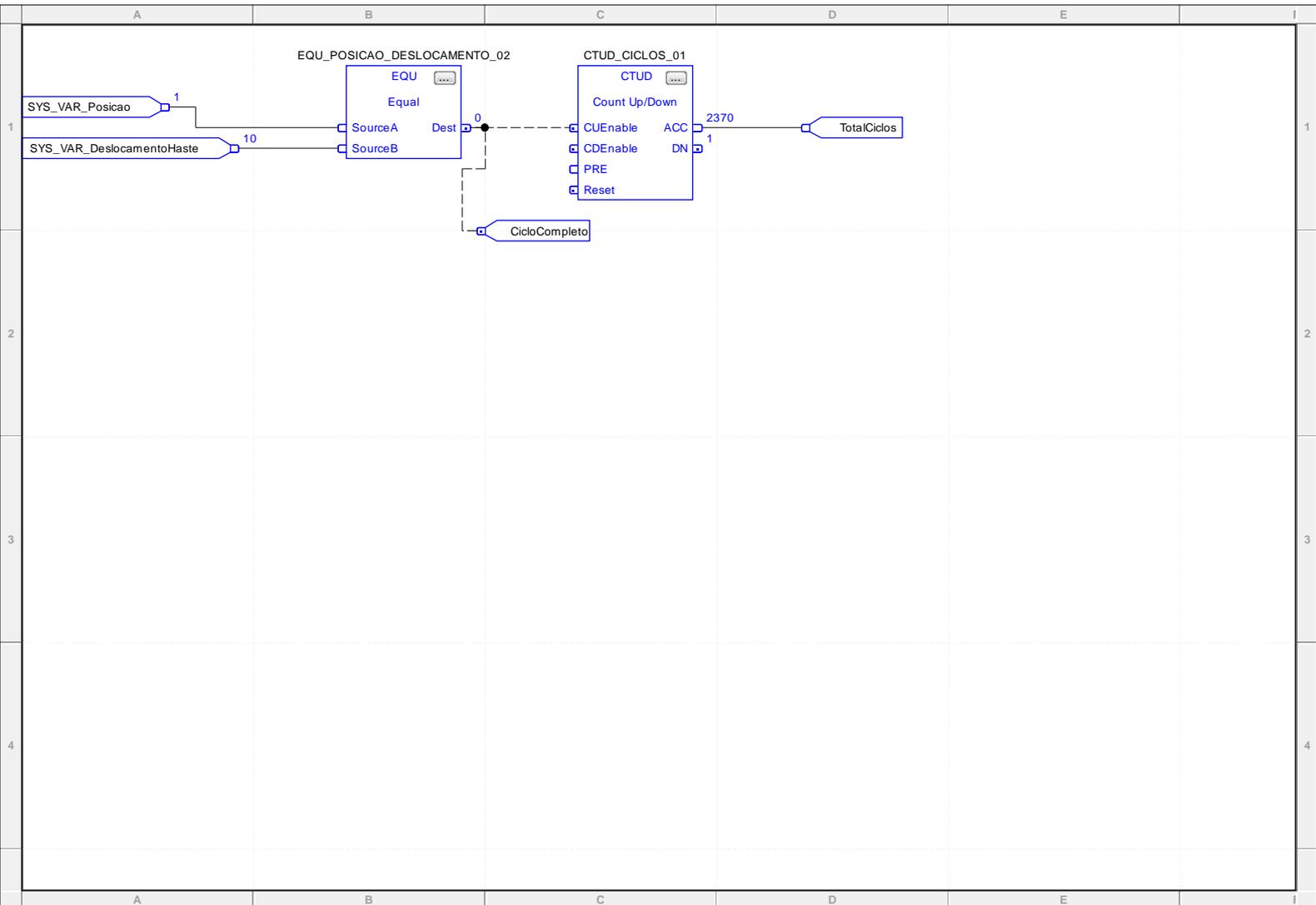
Tabela 6 – Continuação - Principais atributos de uma *tag* (MENDES, 2010)

¹ Para *tags* do tipo *Advise*, a aplicação cliente solicita ao servidor OPC Server que envie os dados automaticamente quando for gerado um novo evento no supervisório ou sistema de controle.

² São denominados *tags* de *output*, as *tags* que permitem ser utilizadas para escrita em um ambiente supervisório ou de controle, através de uma configuração realizada na interface do nó de aquisição.

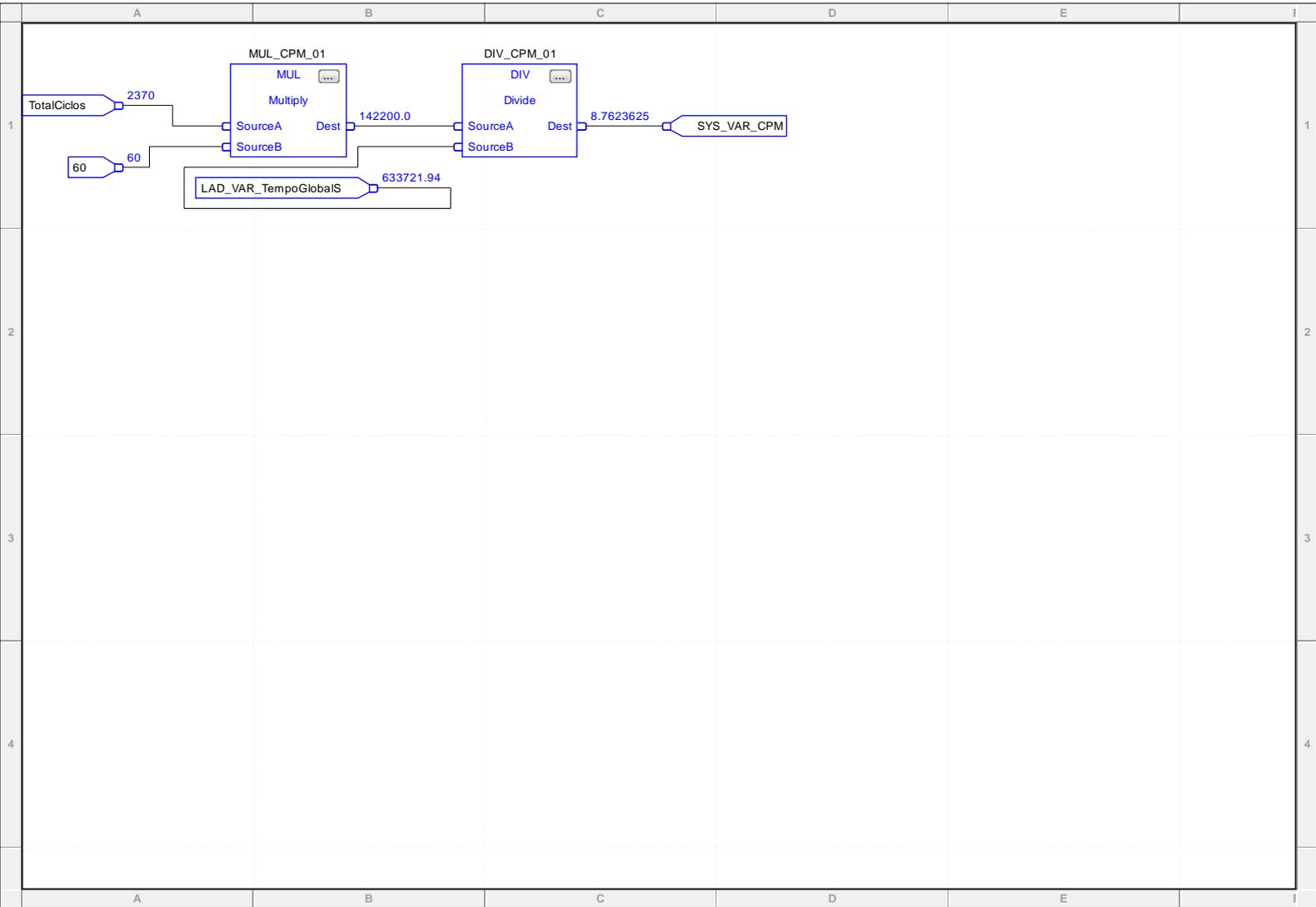
ANEXO C – CÓDIGOS DO CLP

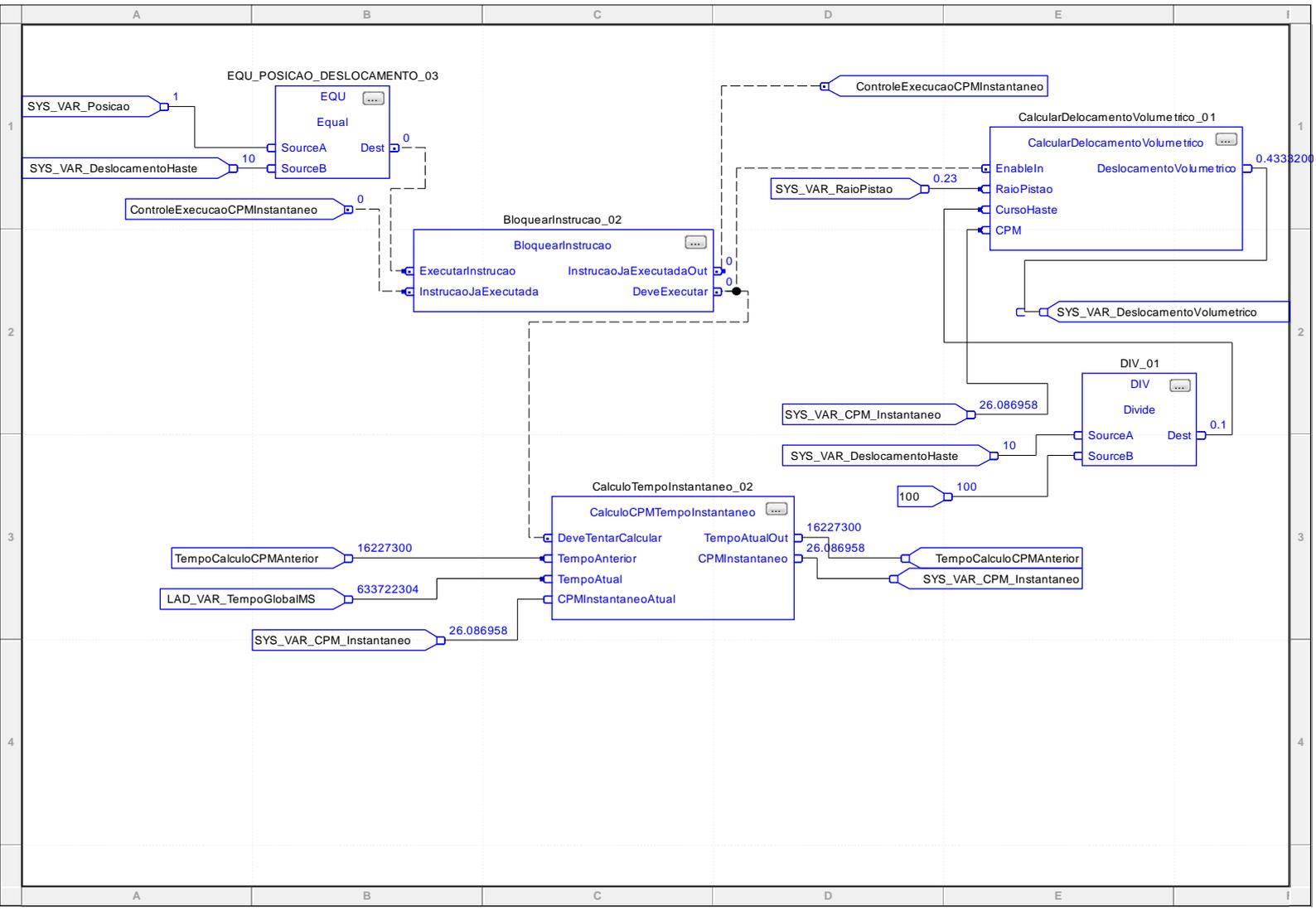


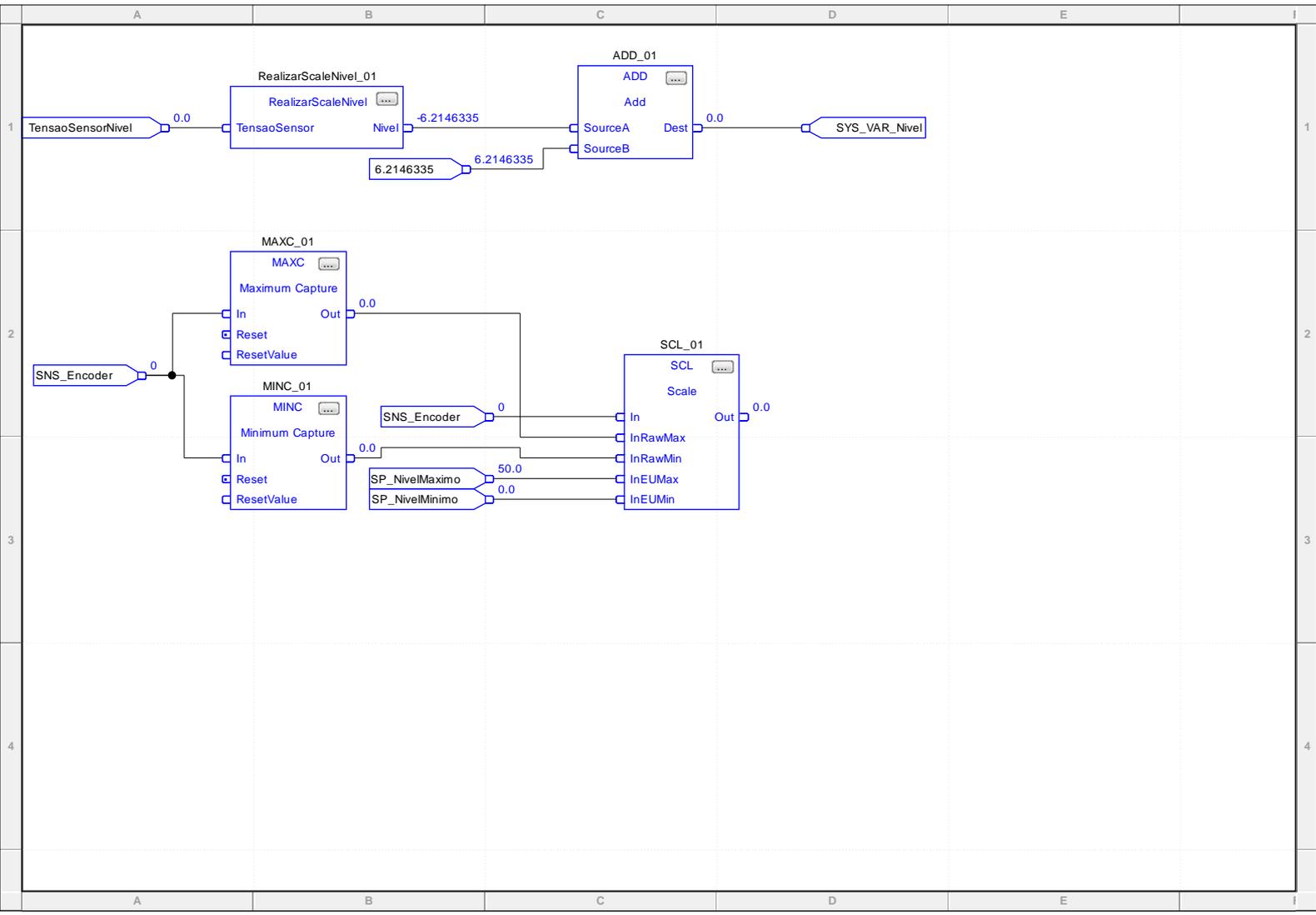


CalcularCPM - Function Block Diagram

SBM:MainTask:MainProgram
1 of 1 total sheets in routine





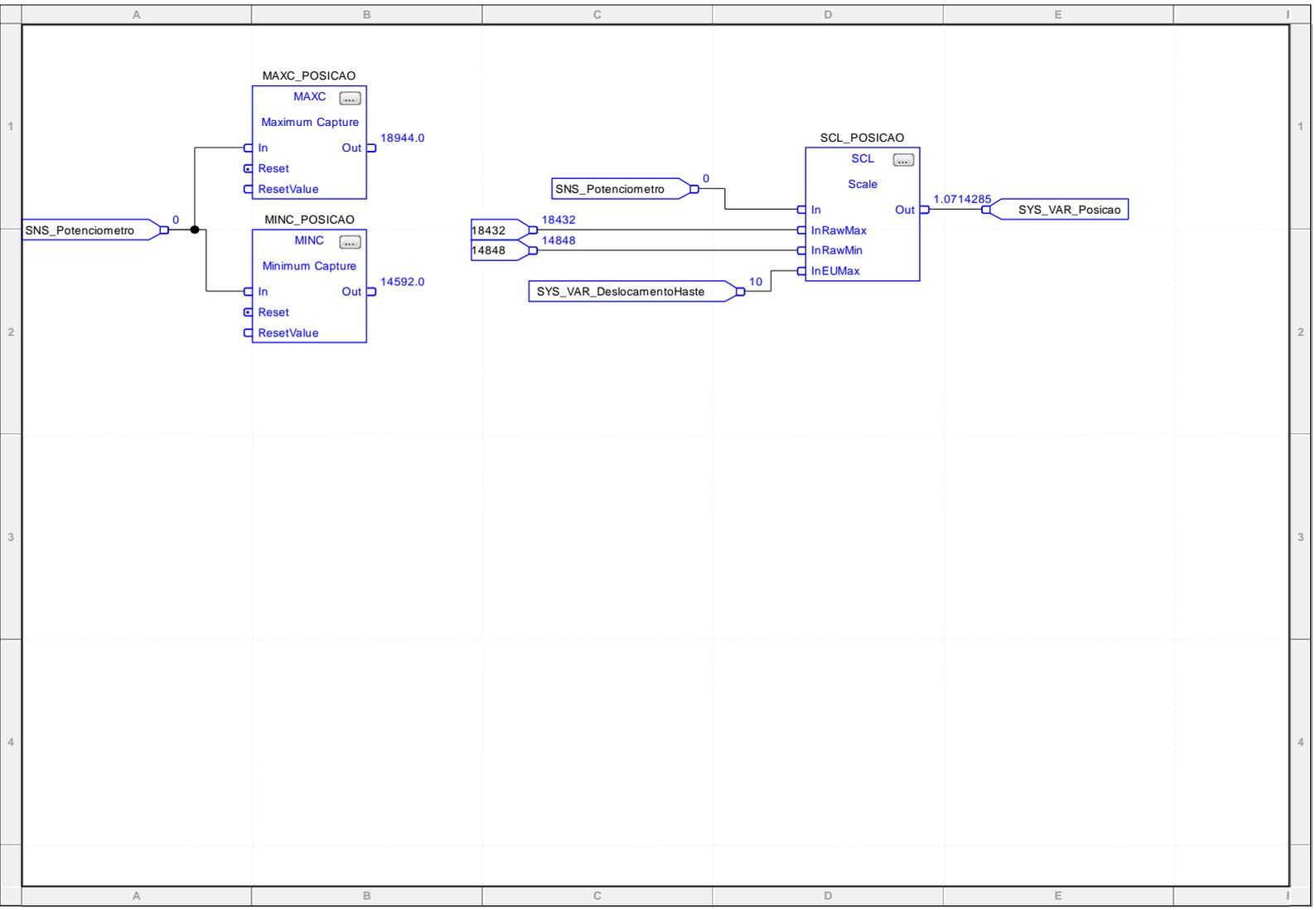


CalcularPosicaoHaste - Function Block Diagram

SBM:MainTask:MainProgram
1 of 1 total sheets in routine

22/08/2014 22:59:11

C:\Users\daniel.correia\Dropbox\TCC\Codigos CLP\SBM.ACD



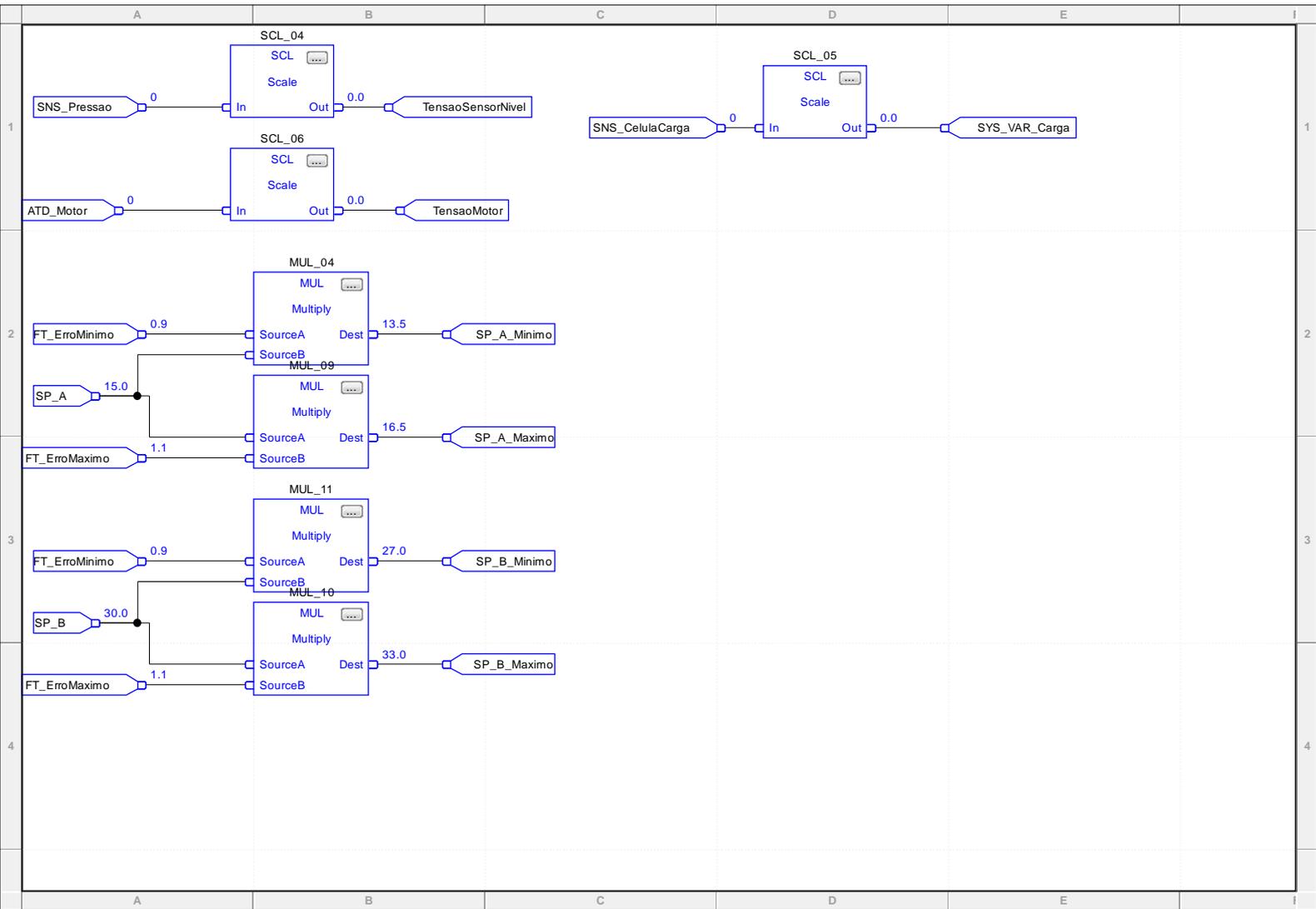
CalculosTempoReal - Function Block Diagram

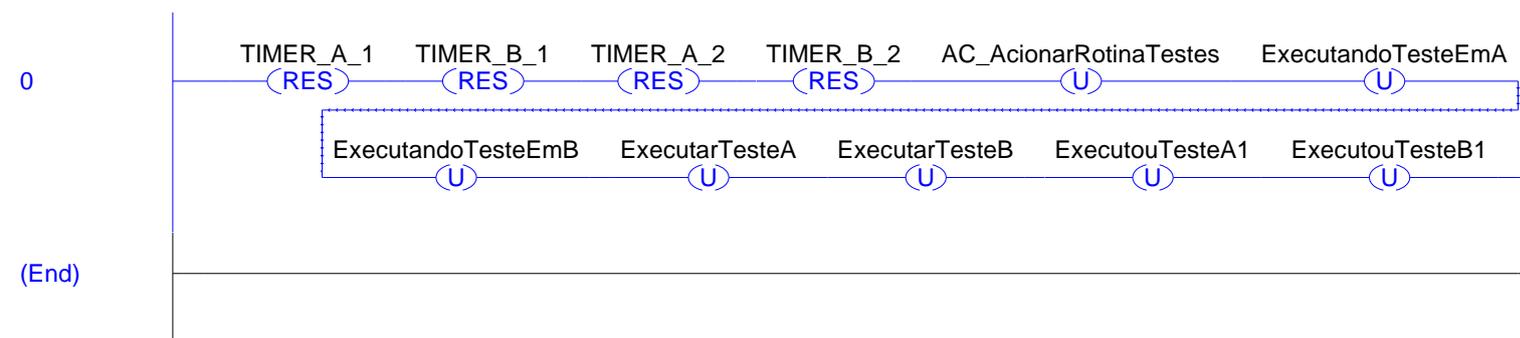
SBM:MainTask:MainProgram

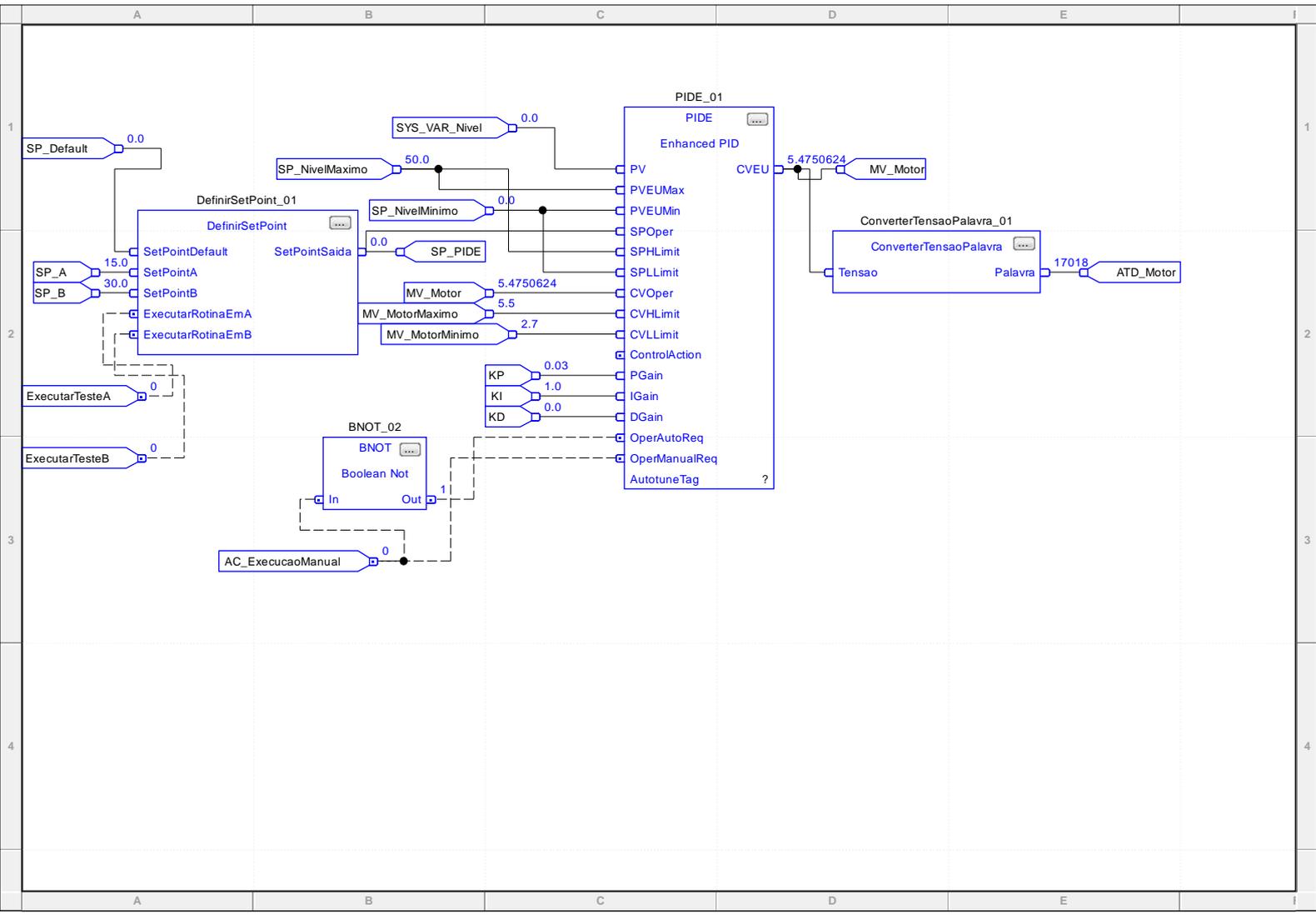
1 of 1 total sheets in routine

22/08/2014 22:59:36

C:\Users\daniel.correia\Dropbox\TCC\Codigos CLP\SBM.ACD







RotinaLigarBomba - Function Block Diagram

SBM:MainTask:MainProgram
1 of 1 total sheets in routine

