

Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais  
Campus Formiga

**Desenvolvimento de um sistema PIMS com comunicação OPC e  
acesso por Navegador *Web***

Silas Martins Sousa

Formiga MG

2014

Silas Martins Sousa

**Desenvolvimento de um sistema PIMS com comunicação OPC e  
acesso por Navegador *Web***

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais, Campus Formiga, como requisito para obtenção do título de Bacharel em Engenharia Elétrica.

Área de Concentração: Automação Industrial.

Orientadora: Profa. Msc. Michelle Mendes Santos.

Formiga MG

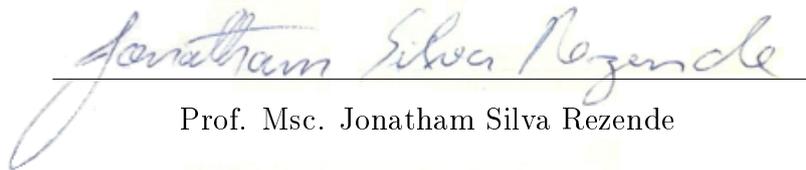
2014

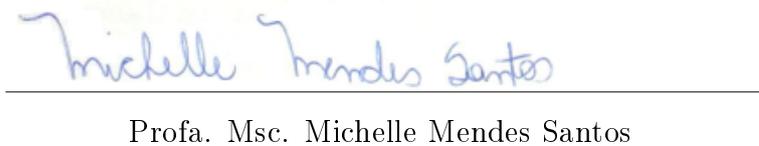
Silas Martins Sousa

**Desenvolvimento de um sistema PIMS com comunicação OPC e  
acesso por Navegador Web**

Aprovado em: 18 / 08 / 2014

Resultado: 96,0 pontos

  
Prof. Msc. Jonatham Silva Rezende

  
Profa. Msc. Michelle Mendes Santos

  
Prof. Msc. Pedro Marinho Sizenando Silva

Formiga MG

2014

# Dedicatória

À minha família e à minha namorada Isabel.

# Agradecimentos

Agradeço primeiramente a Deus pelo dom da vida. Agradeço à minha família por ter ajudado em toda a caminhada acadêmica. À minha namorada Isabel, que sempre me incentivou e motivou. À minha orientadora Michelle por ter contribuído de forma direta na realização deste trabalho. Ao Instituto Federal de Educação, Ciência e Tecnologia, Campus Formiga, por ter cedido o laboratório de Automação onde foram realizados os testes. A todos os funcionários do campus e professores da Engenharia Elétrica por ter contribuído para minha formação. Em especial aos professores da área de Automação, Jonathan e Fábio que me influenciaram na escolha da carreira profissional. Aos professores Everthon e Natália que auxiliaram no desenvolvimento do trabalho.

Agradeço ao Mauro Arruda por ajudar com ensinamentos da Automação durante o período de estágio. Aos professores Niltom e Carlos Renato pela orientação em projetos de pesquisa.

Agradeço também a todos os amigos que encontrei no IFMG Campus Formiga, que durante todo o período da faculdade me ajudou e apoiou.

**"Tornamo-nos deuses na tecnologia, mas permanecemos macacos na vida."**

*Arnold Toynbee*

## Resumo

Os sistemas de Automação como CLPs (Controlador Lógico Programável) e SCADAs (*Supervisory Control and Data Acquisition*) detêm uma gama de dados inerentes a um processo industrial. Essas informações podem ajudar os setores corporativos da indústria, principalmente os setores administrativos, na tomada de decisões estratégicas. Para que as informações de campo cheguem aos setores administrativos é necessário obter e filtrar os dados provenientes do chão de fábrica. O presente trabalho trata do desenvolvimento de uma ferramenta de obtenção de dados dos sistemas de automação, conhecido como PIMS (*Process Information Management System*). O *software* desenvolvido neste trabalho utiliza o Padrão OPC (*OLE for Process Control*) para se comunicar com Servidores OPC, presente nos principais fabricantes de CLPs e SCADAs do mundo. Essas informações são armazenadas em um banco de dados SQL Server. Por meio de um Web Site, também desenvolvido no trabalho, diversos setores da indústria podem visualizar os dados em gráficos e tabelas, de qualquer lugar do mundo, por meio de computadores, *tablets* e *smartphones*. Outra funcionalidade do sistema é gerar relatórios de dados e exportá-los nos formatos PDF, DOC e XLS. O sistema foi testado em uma planta didática de controle de temperatura e nível, presente no Laboratório de Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais, Campus Formiga.

Palavras chave: Sistema PIMS; Padrão OPC; Banco de Dados; SQL Server.

# Abstract

Automation systems such as PLCs (Programmable Logic Controller) and SCADAs (Supervisory Control and Data Acquisition) hold many data inherent to industry process. This information can help the corporate sectors of industry, mainly the administrative sectors in strategic decision-making. For field information from reaching administrative sectors is necessary to obtain and filter the data from the shop floor. This work presents the development of a tool for getting data from automation systems, known as PIMS (Process Information Management System). The software developed in this work uses the OPC Standard (OLE for Process Control) to communicate with OPC Servers, present in the main manufacturers of PLCs and SCADAs in the world. This information is stored in a SQL Server database. Through a Web site, also developed at work, many industry sectors can view data in graphs and tables, from anywhere in the world through computers, tablets and smartphones. Another feature of the system is to generate data reports and export them to PDF, DOC and XLS formats. The system was tested in a didactic plant of temperature and level control, present in the laboratory of Automation of the Federal Institute of Education, Science and Technology of Minas Gerais, Formiga Campus.

Keywords: PIMS System; OPC Standard; Database; SQL Server.

## Lista de Figuras

Figura 1.1	Quantidade e concentração dos dados de uma empresa. . . . .	14
Figura 2.2	Exclusão de pontos nos algoritmos de compressão de dados do PIMS. . .	21
Figura 2.3	Processo de compressão de dados realizado por algoritmos. . . . .	22
Figura 2.4	Integração dos setores da empresa com ERP. . . . .	23
Figura 2.5	Exemplo de uso das chaves Primária e Estrangeira. . . . .	27
Figura 2.6	Exemplo de relacionamentos: a) 1:1, b) 1:n e c) n:n. . . . .	28
Figura 2.7	Exemplo de tabela do Sistema PIMS. . . . .	29
Figura 2.8	Visualização dos componentes <i>div</i> na tela Substituir <i>Login</i> . . . . .	31
Figura 3.9	Tela de Configuração de Conexão ao Banco de Dados. . . . .	38
Figura 3.10	Tela de Gerenciamento dos Usuários do Sistema PIMS. . . . .	39
Figura 3.11	Tela de Gerenciamento dos Servidores OPC. . . . .	40
Figura 3.12	Tela de Cadastro dos Itens OPC. . . . .	41
Figura 3.13	Tela de Conexão OPC. . . . .	42
Figura 3.14	Fluxograma do processo de Conexão com o Servidor OPC. . . . .	43
Figura 3.15	Tela de Tutorial do Executável Local. . . . .	43
Figura 3.16	Organização de uma <i>Master Page</i> em relação às páginas Web. . . . .	44
Figura 3.17	<i>Master Page</i> do Sistema PIMS. . . . .	45
Figura 3.18	Tela de Construção de Gráficos. . . . .	46
Figura 3.19	Tela de Visualização de Gráficos e Tabelas. . . . .	47
Figura 3.20	Tela de Autenticação do Usuário. . . . .	48
Figura 3.21	Tela de Substituição do <i>login</i> e senha do Usuário. . . . .	49
Figura 3.22	Tela de Informações sobre sistemas PIMS, MES e ERP. . . . .	50
Figura 3.23	Tela de Informações sobre Banco de Dados. . . . .	50
Figura 3.24	Fluxograma do funcionamento completo do sistema PIMS. . . . .	51
Figura 4.25	Planta didática de Controle de Temperatura e Nível de Tanques. . . . .	52
Figura 4.26	Configuração do Servidor OPC: a) Adicionando Conexão OPC e b) Associando o CLP ao programa do Logix5000. . . . .	53
Figura 4.27	<i>Software</i> RSI - OPC Test Client (Cliente OPC). . . . .	53
Figura 4.28	Sistema de teste e configuração da rede. . . . .	54

Figura 4.29	Conexão do Cliente OPC com o Servidor OPC. . . . .	55
Figura 4.30	Conexão do Cliente OPC com o Servidor OPC não realizada. . . . .	56
Figura 4.31	SQL Server Management Studio e os valores armazenados. . . . .	56
Figura 4.32	Comunicação OPC: a) ICONICS OPC Simulator e b) <i>Web Site</i> . . . . .	57
Figura 4.33	Autenticação do Usuário para entrar no <i>Web Site</i> do sistema PIMS. . .	58
Figura 4.34	Visualização do <i>Web Site</i> do Sistema PIMS. . . . .	58
Figura 4.35	Exemplo de: a) montagem e b) visualização do gráfico. . . . .	59
Figura 4.36	Relatório de dados nos formatos: a) PDF, b) DOC e c) XLS. . . . .	60

## Lista de Tabelas

Tabela 1.1	Tabela de preços e funções dos Historiadores PIMS do mercado. . . . .	17
Tabela 1.2	Cronograma de Atividades. . . . .	18
Tabela 3.3	Tabela de Usuários. . . . .	36
Tabela 3.4	Tabela de Servidores OPC. . . . .	36
Tabela 3.5	Tabela de Itens OPC. . . . .	36
Tabela 3.6	Tabela de Valores das variáveis do sistema PIMS. . . . .	37

## Lista de Siglas

ASP	<i>Active Server Pages</i> - Ativação de páginas do Lado do Servidor
CEP	Controle Estatístico do Processo
CLP	Controlador Lógico Programável
DA	<i>Data Access</i>
DCOM	<i>Distributed Component Object Model</i> - Modelo de Objetos Distribuídos
DDL	<i>Data Definition Language</i> - Linguagem de Definição de Dados
DLL	<i>Dynamic Link Library</i> - Biblioteca de Acesso Dinâmico
DML	<i>Data Manipulation Language</i> - Linguagem de Manipulação de Dados
ERP	<i>Enterprise Resource Planning</i> - Planejamento de Recursos Empresariais
GB	GigaByte
HTML	<i>HyperText Markup Language</i> - Linguagem de Marcação de Hipertexto
IDE	<i>Integrated Development Environment</i> - Ambiente de Desenvolvimento Integrado
MEI	Microempreendedor Individual
MES	<i>Manufacturing Execution System</i> - Sistema de Execução da Manufatura
OEE	<i>Overall Equipment Effectiveness</i> - Eficiência Global do Equipamento
OLE	<i>Object Linking and Embedding</i> - Objetos Vinculados e Incorporados
OPC	<i>OLE for Process Control</i> - OLE para Controle de Processo
PIMS	<i>Process Information Management System</i> - Sistema de Gerenciamento das Informações do Processo
SCADA	<i>Supervisory Control and Data Acquisition</i> - Supervisão e Aquisição de Dados
SDCD	Sistema Digital de Controle Distribuído
SGBD	Sistemas de Gerenciamento de Banco de Dados
SQL	<i>Structured Query Language</i> - Linguagem de Consulta Estruturada
TA	Tecnologia da Automação
TI	Tecnologia da Informação

# Sumário

<b>1</b>	<b>Introdução</b>	<b>14</b>
1.1	Objetivos	16
1.2	Justificativa	16
1.3	Metodologia	17
1.4	Organização do Texto	18
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>19</b>
2.1	PIMS	19
2.1.1	Fonte de dados	19
2.1.2	Compressão de dados	20
2.2	MES	21
2.3	ERP	23
2.4	Padrão OPC	24
2.5	Banco de Dados	26
2.6	Visual Studio	30
2.6.1	Estruturação de um <i>Web Site</i>	31
<b>3</b>	<b>Desenvolvimento do Sistema PIMS</b>	<b>33</b>
3.1	Executável Local: Configuração do Sistema PIMS	33
3.1.1	Cliente OPC	33
3.1.2	Armazenamento das variáveis no banco de dados SQL	35
3.1.3	Telas do Executável local	38
3.2	<i>Web Site</i> de visualização dos dados	44
3.2.1	<i>Master Page</i>	44
3.2.2	Tela de Construção de Gráficos	45
3.2.3	Tela de Visualização do Gráfico	47
3.2.4	Tela de Autenticação do Usuário	48
3.2.5	Tela de Alteração do <i>Login</i> e Senha do Usuário	49
3.2.6	Telas de Informações sobre sistemas PIMS, MES, ERP e Banco de Dados	49

<b>4 Resultados e Discussões</b>	<b>52</b>
4.1 Plataforma de <i>hardware</i> utilizada nos testes . . . . .	52
4.1.1 Configuração da Rede do Sistema de Teste . . . . .	54
4.2 Testes do Cliente OPC . . . . .	55
4.2.1 Teste de conectividade com outro Servidor OPC . . . . .	57
4.3 Teste do <i>Web Site</i> do Sistema PIMS . . . . .	57
4.4 Análise Econômica do Sistema PIMS . . . . .	59
<b>5 Conclusões</b>	<b>62</b>
5.1 Sugestão de trabalhos futuros . . . . .	63
<b>6 Referências Bibliográficas</b>	<b>64</b>

# 1. Introdução

A informação é a ferramenta mais importante para gerenciar os ativos de uma planta industrial, sejam eles os custos com máquinas, equipamentos e matéria prima ou homem/hora de produção. Segundo Stair (2004) "O valor da informação está diretamente ligado à maneira como ela ajuda os tomadores de decisões a atingirem as metas da organização". De posse das informações do processo é possível identificar quais os elementos que não estão atingindo os resultados esperados. Assim, a indústria consegue aumentar a qualidade e diminuir os custos de produção, tornando-se competitiva no mercado.

A massa de dados tem de ser filtrada e direcionada a diferentes setores de uma indústria pois cada setor tem um interesse específico. Para o gestor não lhe interessa saber se um instrumento ou uma máquina estão descalibrados, porém a informação da produção atual e de meses anteriores são de extrema importância. Para os engenheiros de processo, por outro lado, informações pontuais e específicas são a base para seu trabalho.

A integração das informações do chão de fábrica com as áreas corporativas da indústria só foi possível através de um meio físico de acesso seguro aos dados, que são as redes de comunicação, ou redes industriais.



Figura 1.1: Quantidade e concentração dos dados de uma empresa.

Fonte: (SEIXAS, 2014).

Um sistema de automação industrial pode ser classificado em níveis hierárquicos. Esses níveis necessitam comunicar entre si com o objetivo de trocar informações. A Figura 1.1a, também conhecida como pirâmide da automação, mostra três níveis diferentes de

informações. O primeiro é o nível de controle e supervisão. Nesse nível estão presentes os equipamentos que interagem diretamente com o processo, como instrumentação, CLP e SCADA. O segundo nível é representado pelas ferramentas de gerenciamento da produção, PIMS (*Process Information Management System*) e MES (*Manufacturing Execution System*). PIMS é um sistema para armazenamento de informações do processo com ferramentas para análise de desempenho. O MES é um sistema que auxilia no planejamento da produção. O último nível é representada pelo ERP (*Enterprise Resource Planning*), que engloba os diversos setores da empresa como estoque, financeiro, logística, produção e administração, auxiliando no gerenciamento de negócios.

A Figura 1.1b apresenta o fluxo de informações através dos níveis da pirâmide de automação. Quanto mais elevado é o nível na pirâmide mais filtrados e informativos ficam os dados, conseqüentemente o volume de dados diminui. Nos níveis gerenciais chega apenas um resumo da coleção de dados que são obtidos no chão de fábrica.

A Manufatura Inteligente é a utilização dos dados do processo em benefício dos setores corporativos de uma empresa. Suas ferramentas, PIMS, MES e ERP, alimentam o gestor com informações em tempo real do chão de fábrica, com a possibilidade de confrontar os dados atuais com dados passados.

Para analisar o desempenho de um processo produtivo, além das informações, bons critérios se fazem necessários. O OEE (*Overall Equipment Effectiveness*) é um índice que mede a eficiência da produção levando em conta três critérios (KITE, 2014):

- Disponibilidade;
- Desempenho;
- Qualidade.

Uma prática comum para visualização dos índices OEE é a utilização de telas colocadas no chão de fábrica. Assim, os próprios funcionários podem observar seu desempenho e qualidade em tempo real. Esse procedimento é conhecido como gestão *online*. Os sistemas de manufatura inteligente podem utilizar esse índice para avaliar a produção, até mesmo para estabelecer metas a serem seguidas.

## 1.1 Objetivos

Este trabalho tem como objetivo geral o desenvolvimento de um sistema PIMS de baixo custo que armazena os dados da produção em um servidor e os disponibilizam para os gestores e engenheiros da empresa por meio de gráficos, tabelas ou relatório, de acordo com a demanda do usuário. Esse sistema é composto de duas partes. A primeira é um *software* executável cuja função é realizar a comunicação com os sistemas de Automação por meio do padrão OPC e obter os dados do processo. Esse *software* também deve armazenar os dados em um banco de dados. A segunda parte do sistema é um *Web Site*, utilizado para visualização dos gráficos e tabelas. Esse *Web Site* deve ser acessível remotamente por meio da rede e deve conter mecanismos de controle de acesso por *login* e senha.

Para atingir o objetivo de desenvolver um sistema PIMS com as características citadas, os seguintes objetivos específicos devem ser alcançados:

- Levantamento das características e restrições do padrão OPC;
- Criação de um *software* Cliente OPC capaz de extrair os dados de campo obtidos por meio de um CLP (Servidor OPC);
- Criação de uma interface para configuração e gerenciamento da comunicação OPC;
- Interfaceamento do *software* com o banco de dados SQL;
- Criação de um *Web Site* que acesse o banco de dados SQL;
- Desenvolvimento de ferramentas para montagem e visualização de gráficos e tabelas no *Web Site*;
- Disponibilizar no *Web Site* opções de exportar os gráficos e tabelas nos formatos PDF, DOC e XLS.

## 1.2 Justificativa

Muitas indústrias de pequeno e médio porte ainda possuem sua solução de automação voltada apenas a CLP e SCADA. Porém, cada vez mais, essas indústrias têm se mostrado a favor da manufatura inteligente para conseguirem melhores resultados. A Tabela 1.1 mostra algumas características e preço dos historiadores PIMS existentes no

Tabela 1.1: Tabela de preços e funções dos Historiadores PIMS do mercado.

<b>Produto</b>	<b>Empresa</b>	<b>Características (além de Historiador, Gráfico e OPC)</b>	<b>Preço (R\$)</b>
Infoplus 21	Aspentech	Relatórios (Web e Excel)	R\$120.000/ano (c/ suporte)
PI System	OSI Software	Relatórios (Web e Excel)	R\$600.000(1ºano) + R\$120.000/ano (c/ suporte)
FactoryTalk Historian	Rockwell	Relatórios Excel, (Menor pacote de tags)	R\$9.945( <b>só licença</b> )
Wondeware Historian	Wondeware	Relatórios	R\$20.995( <b>só licença</b> )

Fonte: (CONTROL, 2014; TIERNEY, 2014).

mercado. Os principais *softwares* que desempenham essa função, InfoPlus da AspenTech e PI System da OSI Soft, não são acessíveis às médias e pequenas indústrias. Surge, então, uma lacuna no mercado com a necessidade de um sistema PIMS de baixo custo, que oferte à essas indústrias o suporte necessário para obter, armazenar e mostrar os dados do processo.

### 1.3 Metodologia

O período de desenvolvimento do sistema PIMS foi da segunda semana de Abril (2014) à terceira semana de Julho (2014). A metodologia de trabalho seguiu diversas etapas: Revisão Bibliográfica, desenvolvimento do *Software* Executável local e do *Web Site*, testes práticos e parte escrita. Essas etapas foram divididas em atividades, conforme pode ser observado na Tabela 1.2.

Tabela 1.2: Cronograma de Atividades.

Atividades Realizadas	Abril			Maio			Junho				Julho			
	2	3	4	1	2	3	4	1	2	3	4	1	2	3
Revisão Bibliográfica sobre PIMS, MES e ERP.	X													
Revisão Bibliográfica sobre o padrão OPC.		X												
Criação de um Cliente OPC.			X	X										
Criação de banco de dados e Tabelas SQL.					X									
Integração do Clientes PIMS com o banco de dados.						X								
Criação do <i>software</i> executável para configuração do sistema PIMS.							X	X						
Desenvolvimento do <i>layout</i> do <i>Web Site</i> em ASP.NET (Visual Studio for Web).									X	X				
Construção da parte de visualização dos gráficos e tabelas no <i>Web Site</i> .									X	X				
Criação de relatório de gráficos e tabelas exportados nos formatos PDF, DOC e XLS.											X			
Testes do <i>software</i> local e do <i>Web Site</i> .												X	X	
Desenvolvimento da parte escrita.		X	X	X	X	X	X	X	X	X	X	X	X	X

## 1.4 Organização do Texto

Este texto está organizado em seis capítulos que abordam os seguintes assuntos: o Capítulo 2 contempla as informações necessárias para o devido entendimento deste trabalho, como PIMS, MES, ERP, Padrão OPC, Banco de Dados e Visual Studio. O Capítulo 3 descreve os detalhes e especificações do desenvolvimento do Sistema PIMS. No Capítulo 4 apresenta-se o sistema de teste e os resultados obtidos. As conclusões são apresentadas no Capítulo 5 e as referências bibliográficas no Capítulo 6.

## 2. Revisão Bibliográfica

Este capítulo irá apresentar definições sobre as ferramentas de manufatura inteligente PIMS, MES e ERP, necessárias para a utilização dos dados da produção em benefício da indústria. Para melhor entendimento do desenvolvimento deste trabalho também serão abordados temas como arquitetura de banco de dados, protocolo de comunicação OPC e o ambiente de programação Visual Studio.

### 2.1 PIMS

O sistema PIMS é uma ferramenta de gerenciamento das informações do processo que obtém os dados, os armazenam e os disponibilizam através de gráficos, tabelas e relatórios. Sua principal função é reunir a massa de dados em um banco de dados relacional centralizado, filtrá-los, transformando-os em informação e, com essa informação, gerar conhecimento, auxiliando no processo de tomada de decisões estratégicas.

#### 2.1.1 Fonte de dados

Os dados do processo podem ser obtidos de um sistema supervisório (SCADA) ou diretamente de CLPs e SDCDs (Sistema Digital de Controle Distribuído). A vantagem de se utilizar CLP é que ele é mais confiável, se comparado ao SCADA. Ele também apresenta menor atraso na obtenção dos dados. Por outro lado, no SCADA as variáveis já estão no formato de engenharia, assim, para utilizá-lo como fonte de dados não é necessário transformar as variáveis. Além disso, algumas variáveis derivadas da manipulação dos sistemas supervisórios só existem no SCADA (SEIXAS, 2014).

As informações apresentadas no PIMS podem ser de tempo real ou históricas. Os dados de tempo real mostram o valor instantâneo da variável, sendo portanto, mais utilizados para monitoramento do processo. A consulta a esses dados pode ser feita através de uma linguagem de consulta a banco de dados como, por exemplo, SQL Server, Access e MySQL.

O PIMS é uma ferramenta de grande importância para um engenheiro de processo.

Ele precisa se embasar em informações para tirar conclusões sobre planta e assim ter o diagnóstico da produção. Quando se consegue um ótimo resultado em um processo, todas as configurações (*set-ups*) são armazenadas a fim de buscar essa combinação ótima. Com base nesse padrão de produção o sistema PIMS pode mostrar ao engenheiro qual parâmetro precisa melhorar para maximizar a eficiência.

O sistema PIMS unifica as bases de dados e, com essa centralização, qualquer computador da empresa consegue apresentar dados de diversas áreas do processo. Isso facilita o acesso à informação.

Os principais constituintes de um sistema PIMS são (CARVALHO et al., 2003):

- Historiador de dados;
- Interface gráfica de consulta e visualização de dados.

O historiador faz a aquisição de dados a partir do CLP ou SCADA. A interface mais utilizada para fazer isso é o padrão OPC que facilita a integração entre sistemas de diversos fabricantes. Os dados também podem ser solicitados ciclicamente pelo PIMS, onde o historiador irá gravá-los em um banco de dados relacional.

A interface de visualização dos dados irá realizar uma consulta ao banco de dados e apresentar na tela o resultado através de gráficos ou outras representações. Esses dados também podem ser visualizados por meio do navegador *Web Browser* (CARVALHO et al., 2003).

### 2.1.2 Compressão de dados

Uma característica importante de um sistema PIMS é a grande capacidade de armazenamento de dados, guardando-os por cerca de 15 anos. Para isso, esses sistemas lançam mão de algoritmos de compressão de dados que conseguem reduzir entre 10 a 20 vezes seu tamanho original, ou seja, taxa de compressão de 10:1 a 20:1 (SEIXAS, 2014).

Os algoritmos que realizam a compressão devem atender a critérios como velocidade de compressão e de descompressão. Caso essas premissas não sejam atendidas podem ocorrer erros de solicitação, por exemplo, quando um gráfico necessita rapidamente de uma gama de informações históricas, sua velocidade de descompressão não pode ser menor que a taxa de requisição dos dados.

Diferentemente dos algoritmos de compressão de texto, os quais procuram por repetições de palavras, a compressão do PIMS registra o dado apenas quando ele varia significativamente (além de um valor limiar pré estabelecido). Por exemplo, dois pontos ligados formam-se uma reta, caso existam pontos complementares entre eles e próximos dessa reta, então pode-se desconsiderar os pontos complementares.

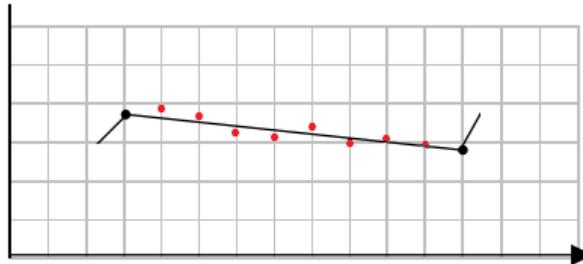


Figura 2.2: Exclusão de pontos nos algoritmos de compressão de dados do PIMS.

Na Figura 2.2 é possível observar que os pontos da extremidade, ligados, formam uma reta. Assim, os pontos complementares (em vermelho) são desconsiderados e apenas os pontos da extremidade (em preto) são armazenados.

Dois tipos de algoritmos são largamente utilizados em historiadores PIMS (MELO, 2012):

- Swinging Doors - Utilizado pelo PIMS PI System da empresa OSI (OSISOFT, 2014);
- Boxcar/Backslope - Utilizado pelo PIMS Infoplus 21 da Apenstech (ASPENTECH, 2014).

Ambos aplicam o método de descarte de pontos fora de uma faixa de variação, as etapas dessa técnica estão mostradas nos gráficos da Figura 2.3.

## 2.2 MES

MES (*Manufacturing Execution System* - Sistema de Execução de Manufatura) faz o gerenciamento e controle da produção, envolvendo todas as etapas de um processo como planejamento, execução até a expedição de um produto. São atribuições de sistemas MES (ALMEIDA, 2004):

- Controle Estatístico do Processo (CEP);

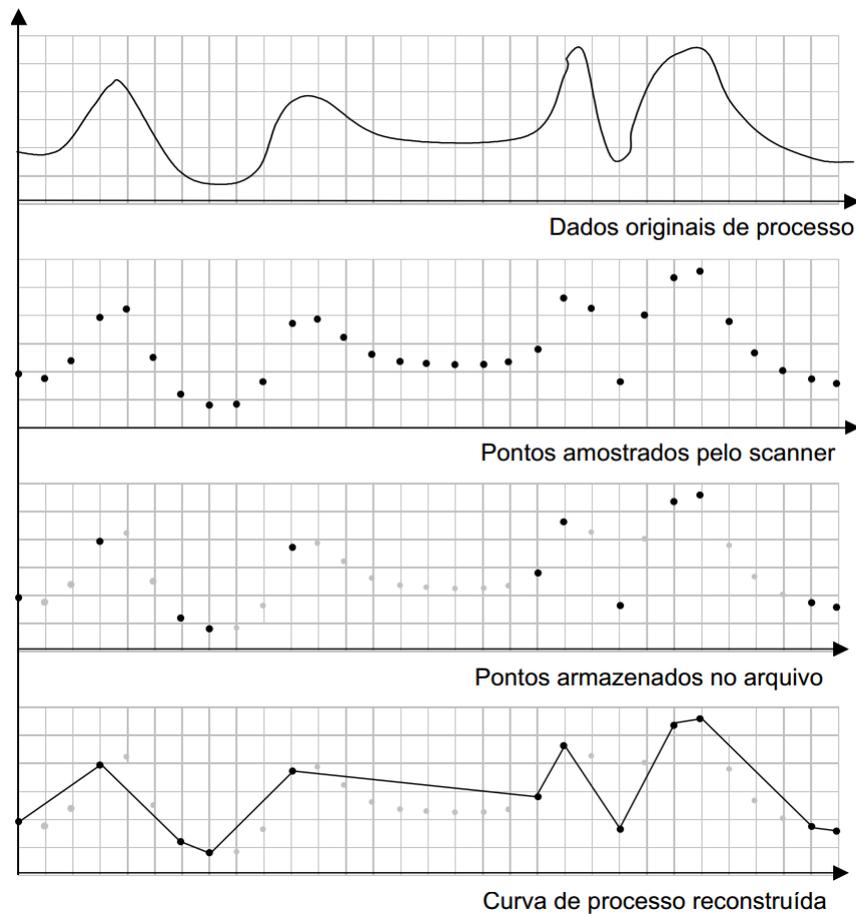


Figura 2.3: Processo de compressão de dados realizado por algoritmos.

Fonte: (SEIXAS, 2014).

- Rastreabilidade dos produtos;
- Controle de Eficiência e Gerenciamento de Paradas (OEE, *Downtime*, Perdas e Ganhos);
- Gerenciamento de Processos (Expedição, Recebimento e *Logging*);
- Gerenciamento de Manutenção (Preditiva e Preventiva);
- Integração com sistemas ERP (Sistemas de Gestão Empresarial).

O MES se aplica mais a indústrias de processos discretos, por batelada, onde uma produção é baseada em receitas com começo, meio e fim (NOGUEIRA, 2009). Os sistemas MES têm uma função mais “ativa” que os PIMS. Isto no sentido de que eles obtêm as várias informações associadas ao plano de produção gerado pelos sistemas ERP e, com base nisso, atuam em conjunto com os sistemas supervisórios no controle da execução

das atividades, fluxo de materiais, emissão de ordens de armazenamento, entre outras inúmeras ações.

## 2.3 ERP

Com o desenvolvimento da Tecnologia da Informação (TI) e sua utilização na Tecnologia da Automação (TA), juntamente com a padronização da rede *ethernet*, surge uma tendência natural da interligação da rede industrial (de chão de fábrica) com a rede corporativa de nível gerencial (ALMEIDA, 2004).

O ERP (*Enterprise Resource Planning* - Planejamento de Recursos Empresariais) tem objetivo de conectar todas as áreas de uma empresa para evitar ilhas de informações. A centralização dos dados pode ajudar as áreas corporativas de uma organização no gerenciamento da planta, como é o caso do estoque, logística, recursos humanos e financeiro.

Essa filosofia de Gestão Integrada que o ERP apresenta possui diversas vantagens, como:

- Maior controle sobre a organização;
- Aumento da capacidade de tratamento das informações;
- Rapidez na obtenção dos dados.

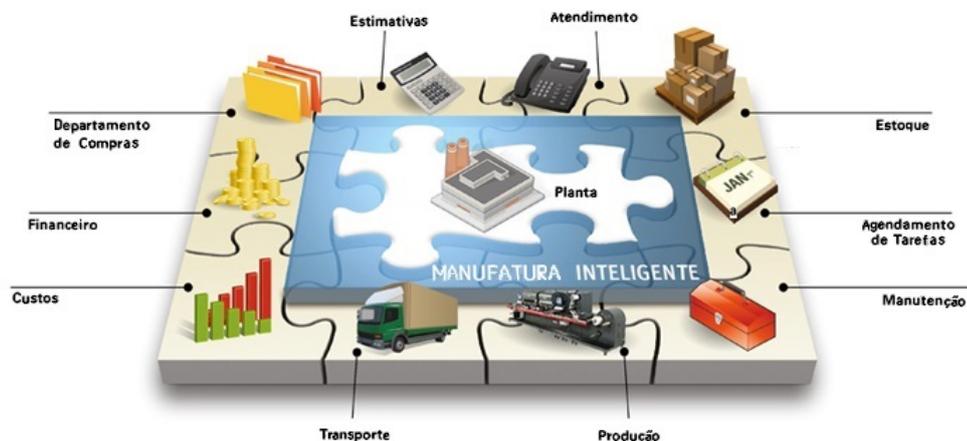


Figura 2.4: Integração dos setores da empresa com ERP.

Fonte: adaptado de (MTHOMAZINI, 2014).

A Figura 2.4 mostra a integração dos diversos setores de uma organização feita pelo ERP, evitando, assim, problemas como indisponibilidade de um dado para algum setor da empresa ou também a inconveniência de ter que inserir o mesmo dado duas vezes. O ERP também auxilia no *marketing* empresarial e na logística de vendas. Quando uma venda é realizada os dados do estoque e transporte são requisitados, assim, é possível informar ao cliente a data de entrega do seu produto.

## 2.4 Padrão OPC

Antes da criação do padrão OPC (*OLE for Process Control*), os clientes eram dependentes de *drivers* proprietários, os quais eram difíceis de serem atualizados. Para cada aplicação era necessário uma versão do *driver*. Em 1995 empresas que fabricavam equipamentos de automação se reuniram com a Microsoft com o objetivo de desenvolver um padrão para acesso a dados de tempo real. O OPC foi baseado nas tecnologias OLE e DCOM, ambas da Microsoft e realizam comunicação entre plataformas Windows. Após a reunião, os técnicos e especialistas fundaram a OPC Foundation. O objetivo desse grupo é atender às necessidades da indústria e promover a interoperabilidade, onde equipamentos de diferentes fabricantes conseguem comunicar entre si.

O padrão OPC estabelece regras bem definidas para que sejam desenvolvidos sistemas aplicativos. Esses sistemas realizam a comunicação entre os dispositivos de campo (CLPs e instrumentos) e os sistemas de monitoramento e supervisão (SCADA, PIMS, MES e ERP). Esse conjunto de regras e especificações são divulgados gratuitamente no *site* da OPC Foundation (OPC FOUNDATION, 2014). Com elas os programadores e desenvolvedores obtêm todas as especificações para criarem programas com esse padrão de comunicação.

Existem dois tipos de interfaces que podem ser feitas para realizar a comunicação OPC. A primeira, do tipo *custom*, utiliza linguagens de programação que suportam chamada de funções por ponteiros, como por exemplo C, C++ e Delphi. A segunda é a *automation* onde os métodos são chamados pelo nome e não por ponteiros. Este tipo de interface pode ser desenvolvida por meio da linguagem Visual Basic.

Para realizar uma comunicação OPC é necessário um Servidor e um ou mais Clientes OPC. O servidor é a fonte de dados, é ele que fornece os dados para um cliente. O

cliente OPC recebe esses dados. Normalmente, os sistemas de monitoramento são clientes OPC (sistemas supervisórios e IHM) os quais obtêm as informações do processo por meio dos CLPs (sendo estes Servidores OPC).

O protocolo de comunicação OPC possui diversas vantagens sobre os *drivers* proprietários, são elas:

- Facilita a integração de Sistemas;
- Interoperabilidade entre sistemas de diversos fabricantes;
- Integração com sistemas MES, ERP e aplicações Windows (Excel);
- Redução dos custos e tempo para desenvolvimento de interfaces.

Outra grande vantagem do padrão OPC é a capacidade de um cliente obter dados de vários servidores ao mesmo tempo e um servidor conseguir entregar informações para diversos clientes. Um exemplo de Servidor OPC é o RSLinx da Rockwell Automation. O servidor OPC foi incorporado a esse programa a partir da versão 2.10 (MECATRÔNICA ATUAL, 2013). Seu próprio sistema SCADA (*FactoryTalk*) o utiliza para comunicar com o CLP. Existem também servidores de terceiros (aqueles que não são desenvolvidos por fabricantes de equipamentos de automação), por exemplo, Matrikon e Kepware.

O OPC é dividido em três camadas:

- Servidor;
- Grupo;
- Item.

O item é a informação de uma variável. Além do valor, ele carrega outras informações importantes dessa variável como o *Time stamp* e a qualidade do dado. O *Time stamp* informa a data e a hora em que o dado foi transmitido e a qualidade informa se o dado é bom ou ruim. Dado ruim pode significar erro de transmissão ou perda de informação. Isso se faz necessário porque alguns sistemas de gerenciamento da produção (como o MES) tomam decisões baseadas nos resultados das variáveis. O grupo OPC consiste num agrupamento de itens. As funcionalidades de um grupo são pouco utilizadas pelos programadores. Por exemplo, quando as informações de um setor da indústria são agrupadas e eventualmente esse setor esteja em manutenção, uma simples desativação desse grupo (no cliente OPC) faria com que o sistema melhorasse seu desempenho. Vários grupos podem estar contidos no servidor OPC, que é a fonte de dados.

A leitura de dados realizada por um cliente OPC pode ser síncrona ou assíncrona. A leitura síncrona depende de confirmação antes de uma nova leitura, o que gera um desempenho inferior, porém com confiabilidade levemente superior do que a assíncrona, na qual o cliente espera pelos dados.

As informações podem ser atualizadas ciclicamente ou por mudança de estado. Na atualização cíclica o cliente OPC faz requisições periódicas ao servidor. Já no caso de mudança de estado, a cada alteração da variável (maior que a banda morta) o servidor envia essa alteração para o cliente. A banda morta é a base para os algoritmos de compressão de dados já citados na seção 2.1.2. Isso permite que pequenas variações (dentro da banda morta) não gerem um novo valor para o cliente OPC.

Existem vários grupos de especificação OPC, como por exemplo, OPC-DA (*Data Access*) e OPC-XML. O OPC-XML é utilizado para buscar itens OPC diretamente da *Web*, já o OPC-DA é utilizado em aplicações locais.

## 2.5 Banco de Dados

Atualmente há empresas que investem muito dinheiro para terem acesso às informações, sejam elas informações do mercado ou de sua própria produção (CAIÇARA, 2012). Com esse valor que a informação tem, faz-se necessário boas práticas para utilização dos bancos de dados, tornando-os confiáveis e organizados.

Banco de dados é uma coleção de dados inter-relacionados que carregam informações de uma aplicação ou do mundo real, organizados de forma que possam ser buscados facilmente por meio de consultas. Eles proporcionam confiabilidade, organização e facilidade de alteração, o que se torna muito difícil caso os dados estivessem armazenados, por exemplo, em arquivos de texto (WELLING, THOMSON, 2005).

Os Sistemas de Gerenciamento de Banco de Dados (SGBD) são *softwares* com recursos específicos para facilitar a manipulação dos dados e o desenvolvimento de programas aplicativos. Um exemplo é o SQL Server Management Studio, um SGBD do banco de dados SQL Server. Os SGBDs podem oferecer controle de acesso aos dados e gerenciador de *backups* (GAUDÊNCIO, 2014).

Os bancos de dados são compostos de tabelas. Também conhecidas como entidades, as tabelas representam um conjunto de dados que de alguma forma se relacionam.

Elas possuem colunas (campos) que carregam um determinado tipo de informação como nome, cidade, telefone e idade. Cada linha dessa tabela é um registro, ou seja, um conjunto de informações. A chave primária (*primary key*) é utilizada quando se deseja identificar um registro, para isso ele deve ser único (unívoco), assim como o número de CPF identifica uma pessoa. As chaves estrangeiras (*foreign key*) são utilizadas quando se deseja representar um dado de outra tabela sem que esse dado seja inserido novamente. A Figura 2.5 mostra a utilização da coluna ClienteID (Chave primária da tabela Cliente) como chave estrangeira na tabela Compras.

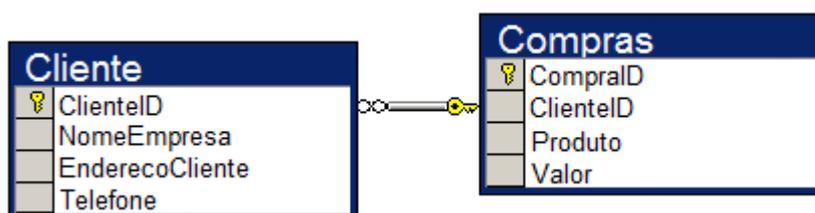


Figura 2.5: Exemplo de uso das chaves Primária e Estrangeira.

Alguns critérios devem ser levados em conta para se obter um banco de dados eficiente, são eles: redundância, inconsistência e integração. A redundância é a repetição de um dado, ou seja, um mesmo dado inserido duas vezes. Isso é inadequado e por tanto deve ser evitado, pois além de aumentar os custos de armazenamento existe a possibilidade desse dado ser escrito de formas diferentes. Como os dados passam por alterações ao longo do tempo, caso apenas um desses dados repetidos seja atualizado ocorrerá problemas de inconsistência de dados (MEDEIROS, 2007).

Para satisfazer aos três critérios apresentados são utilizados os Relacionamentos. Eles descrevem como as tabelas se relacionam entre si. Primeiramente é necessário identificá-los. Os relacionamentos são representados por um par de elementos, chamado de cardinalidade. Um exemplo é o relacionamento um-para-muitos (1:n), onde o número 1 representa que um dado da tabela A pode estar relacionado a qualquer (n) valor da tabela B. A seguir são apresentados os tipos de relacionamentos e como eles devem ser devidamente tratados:

- **(1:1)**: O relacionamento de um-para-um acontece quando um valor em A está relacionado com apenas um valor em B e um valor em B está associado com apenas um valor em A. Então para implementar essa relação basta registrar os valores em uma só tabela, visto que os dados dependentes podem ficar armazenados nessa

tabela.

- **(1:n)**: O relacionamento de um-para-muitos é quando um valor em A está relacionado a qualquer valor em B, mas um valor em B está associado a apenas um valor em A. Para este caso deve-se associar a informação de B à chave estrangeira em A.
- **(n:n)**: O relacionamento de muitos-para-muitos acontece quando um valor em A está relacionado a qualquer valor em B e um valor em B está relacionado a qualquer valor em A. Para este relacionamento deve ser criada uma nova tabela que conterá os dados do relacionamento entre A e B. Esta nova tabela possuirá uma chave composta que consiste na junção das chaves primárias de A e B, tornando-se, assim, dois relacionamentos de um-para-muitos. As três relações, acima citadas, podem ser observadas na Figura 2.6.

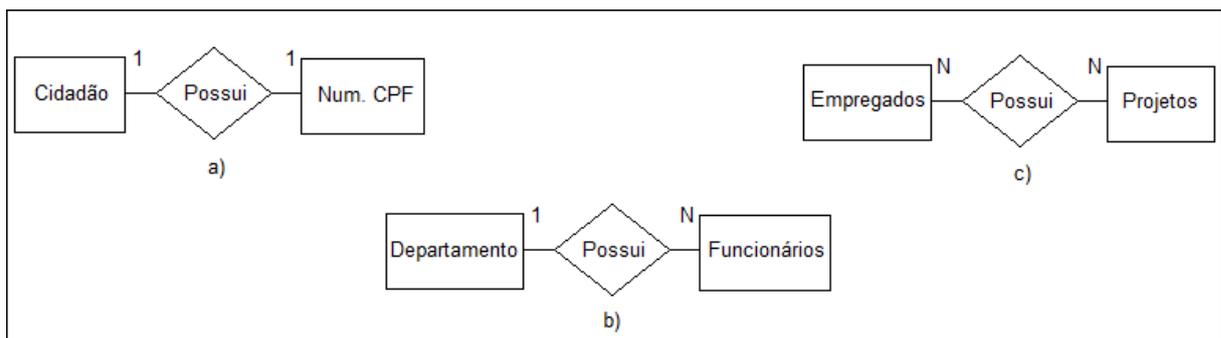


Figura 2.6: Exemplo de relacionamentos: a) 1:1, b) 1:n e c) n:n.

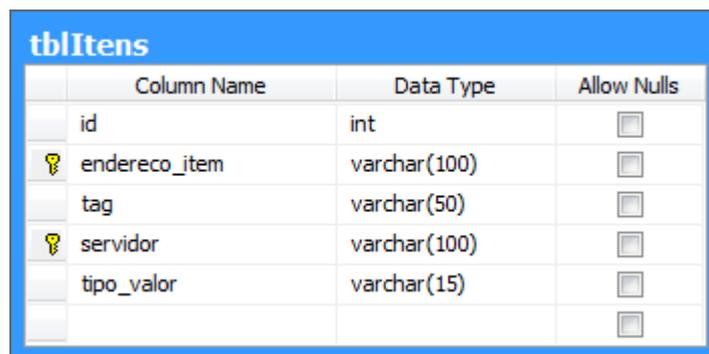
As Linguagens de Definição de Dados (DDL) são linguagens que definem esquemas de banco de dados. Elas são utilizadas para criar (*create*), apagar (*drop*) e editar (*alter*) tabelas. As Linguagens de Manipulação de Dados (DML) são os métodos utilizados para tratar as informações que estão contidas nas tabelas. Com as DMLs é possível selecionar (*select*), inserir (*insert*), apagar (*delete*) e editar (*update*) registros de uma tabela (FRANQUINI, 2014).

Existe uma normalização que descreve regras para a construção de modelos de banco de dados mais robustos, com menos dependências entre seus elementos e menos redundância de informações. Através dessa normalização pode-se verificar se o esquema lógico do banco de dados está eficiente. Ela é dividida em três definições: 1ª, 2ª e 3ª Forma Normal:

- **1FN (1ª Forma Normal)**: Toda relação deve ter uma chave primária e deve-se

garantir que todo atributo seja atômico (não composto). A atomicidade de um dado é importante para realizar consultas eficientes. Atributos compostos devem ser separados, por exemplo, o endereço deve ser dividido em logradouro, número, bairro, cidade e CEP. Caso contrário torna-se difícil distinguir um usuário que mora na cidade do Rio de Janeiro de outro que mora na Rua Rio de Janeiro na cidade de Belo Horizonte.

- **2FN (2ª Forma Normal)**: Deve obedecer a 1FN e todo atributo deve ser totalmente dependente da chave primária. São especialmente problemas para aquelas relações que possuem chave primária composta.
- **3FN (3ª Forma Normal)**: Deve obedecer a 2FN e todo atributo que não for chave primária ou estrangeira, deve ser mutuamente independente. Caso esse requisito não seja atendido, modificações em um desses dados levaria à modificação de uma série de outras informações desnecessariamente. Deve-se, portanto, desmembrar em outra tabela os atributos mutuamente dependentes, assim esses dados apareceriam na tabela antiga apenas referenciado (chave estrangeira) da tabela criada. Esta forma normal também ajuda a diminuir redundâncias e aumentar a independência das relações.



tblItens			
	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
🔑	endereco_item	varchar(100)	<input type="checkbox"/>
	tag	varchar(50)	<input type="checkbox"/>
🔑	servidor	varchar(100)	<input type="checkbox"/>
	tipo_valor	varchar(15)	<input type="checkbox"/>
			<input type="checkbox"/>

Figura 2.7: Exemplo de tabela do Sistema PIMS.

As formas normais e os relacionamentos são utilizados nas tabelas do sistema PIMS. A Figura 2.7 mostra a tabela de Itens OPC, ela foi criada para armazenar as informações das variáveis do processo. Nesta tabela os dados de todas as colunas são atômicos e dizem a respeito exclusivamente ao Item OPC (com exceção do *servidor*, o qual é uma chave primária).

## 2.6 Visual Studio

O Visual Studio é uma IDE (ambiente de desenvolvimento) que possui, além do ambiente de programação, a parte visual. A proposta deste *software* é oferecer uma ferramenta de desenvolvimento que seja amigável, de rápida programação, proporcionando *layouts* agradáveis para o usuário final.

O Visual Studio possui versões gratuitas que podem ser obtidas pelo *site* da própria empresa (VISUAL STUDIO, 2014). Para continuar funcionando, essas versões tem que ser registradas no próprio *site* (também gratuitamente).

A IDE do Visual Studio pode ser utilizada para programar em diversas linguagens como: C++, C# e Visual Basic. As mais utilizadas são Visual Basic e C#. A quantidade de linguagens suportadas depende da versão do *software*.

Os arquivos executáveis gerados pelo Visual Studio necessitam da versão compatível do .Net Framework para serem executados. Os computadores que têm as últimas versões do Visual Studio (2008, 2010, 2012 e 2013) já possuem esse *software* instalado.

Além de fazer programas executáveis, o Visual Studio pode ser utilizado no desenvolvimento de *Web Sites* e aplicativos para *smartphones* (plataforma Windows Phone). No programa local, o formulário é conhecido como *Windows Form* e no caso das páginas Web o formulário é chamado de *Web Form*. Para construção de *sites* o Visual Studio utiliza a tecnologia ASP.NET para compilar códigos no lado do Servidor. As páginas Web construídas possuem a extensão “aspx”.

A IDE para criação de *Sites* conta com três ambientes de programação (integrados):

- Código;
- Visual;
- HTML.

O primeiro ambiente corresponde ao Código (*Code*), exatamente igual ao desenvolvimento dos programas locais, utilizando C#, C++ e VB. O segundo é a ferramenta de *Designer*, parte visual utilizada para detalhar melhor o ambiente gráfico do *Site*. O terceiro faz a integração com o ambiente de navegação *Web*, que é a linguagem nativa HTML. Qualquer alteração na parte visual automaticamente modifica o *script* HTML (GRAÇA, 2014).

Para realizar a troca de informações entre telas de um *site*, o ASP.NET utiliza o método *Session*. Este método armazena as variáveis (que se deseja passar para outra tela)

e quando a nova tela é chamada essas variáveis são passadas para essa nova tela. Para recuperá-los, basta ler a variável *session* correspondente (na nova tela).

Para o site ser publicado é necessário gerar os arquivos para divulgá-lo na Internet, isso é feito com a ferramenta "Publicar *Web Site*" do Visual Studio.

O Visual Studio, a partir da versão 2012, oferece benefícios na utilização de banco de dados SQL. Sua própria IDE possui um Sistema de Gerenciamento de Banco de Dados, que diminui o tempo de integração entre a aplicação e o banco de dados.

### 2.6.1 Estruturação de um *Web Site*

Para realizar a estruturação do ambiente gráfico do *Web Site* são necessárias adequações das dimensões e do posicionamento dos objetos. Para fazer o posicionamento dos objetos foi utilizado o componente HTML *div*. Nos formulários dos *softwares* executáveis os objetos gráficos são arrastados do *Toolbox* para o formulário e permanecem no lugar desejado, pois guardam uma propriedade de posição *x* e *y* na tela de acordo com a construção. Isso não acontece nos formulários *Web* e por isso utiliza-se o *div*. Esse componente é uma espécie de tabela que divide a tela em retângulos utilizados para dar o espaçamento e abrigar outros objetos do *site* como, por exemplo, botões, caixas de textos, imagens e *menu*.

A imagem mostra uma interface web para troca de login e senha. O título é "Trocar Login ou Senha do Usuário.". Abaixo, há dois campos de entrada para "Login antigo:" e "Senha antiga:". Abaixo disso, há dois campos de entrada para "Novo Login:" e "Nova Senha:". No final, há um botão azul com o texto "Substituir". Todos os elementos do formulário (campos de texto e o botão) estão sobrepostos a uma grade de retângulos brancos com bordas pontilhadas, que representam os componentes *div* utilizados para a estruturação da página.

Figura 2.8: Visualização dos componentes *div* na tela Substituir *Login*.

Na Figura 2.8 são mostrados os componentes *div* (retângulos brancos pontilhados).

Essa figura foi retirada do ambiente de programação do Visual Studio. Os componentes *div* podem ser vistos no *site* ou ficar invisíveis ao usuário. Neste último caso, o *div* fica visível somente na IDE de programação, servindo apenas para a estruturação do *site*.

Para adequar o *site* às diferentes resoluções de telas, nas quais ele será visualizado, é necessário utilizar dimensões em porcentagem. Essa configuração é conhecida como *Layout* Fluido ou *Layout* Responsivo (BARROS, 2014). Assim, as subpartes do *site* ocuparão uma área relativa ao tamanho da janela do navegador. Distâncias fixas, em *pixels*, podem comprometer o *layout* do *site* e por isso devem ser evitadas. Ao utilizar as larguras em porcentagem o navegador recalcula todas as dimensões baseando-se na resolução da tela utilizada pelo usuário. É possível delimitar larguras mínima e máxima para garantir que dimensões importantes não sejam afetadas com o redimensionamento.

O Visual Studio foi utilizado no trabalho para desenvolver um *software* e um *Web Site*. Para isso foram utilizadas essas ferramentas descritas anteriormente, como formulários (*windows form* e *web form*), a parte visual e de código, *layout* fluido e a estruturação com componentes *div*.

## 3. Desenvolvimento do Sistema PIMS

Este capítulo mostra as etapas de desenvolvimento do Sistema PIMS. O sistema foi dividido em duas partes: um *software* executável local e um *Web Site*. O executável local é um gerenciador das configurações do sistema. Ele se conecta a um Servidor OPC, obtém os dados e os armazenam em um banco de dados SQL. Nele é possível gerenciar o *login* de usuários, os servidores OPC e cadastrar as variáveis do processo (Itens OPC). A outra parte do Sistema PIMS, o *Web Site*, lê as informações do banco de dados e as apresenta ao usuário. No *site* é possível escolher a variável, o tipo de gráfico e o período de visualização. Ambas as partes foram feitas no Visual Studio, o executável local foi feito na versão Microsoft Visual C# 2010 e o *Web Site* no Visual Studio Express 2013 for Web. O banco de dados é o único meio de comunicação entre o executável local e o *Web Site*.

O objetivo de dividir o Sistema PIMS em duas partes é de melhorar o desempenho das consultas ao banco de dados e também aumentar a segurança da informação. Como os usuários do sistema possuem acesso apenas ao *site*, algumas funções de configuração ficam asseguradas do lado do desenvolvedor (executável local).

A seguir é apresentado o detalhamento das duas partes do sistema PIMS desenvolvido neste projeto, o executável local e o *Web Site*.

### 3.1 Executável Local: Configuração do Sistema PIMS

O *software* executável local tem como objetivo estabelecer a comunicação com Servidores OPC, presentes em CLPs e SCADAs e armazenar as informações obtidas em um banco de dados. Para realizar essa tarefa, um cliente OPC foi implementado nesse *software*, conforme descrito a seguir.

#### 3.1.1 Cliente OPC

Para desenvolver sistemas aplicativos com comunicação OPC é necessário ter profundo conhecimento das regras do padrão, contidas nas cartilhas da OPC Foundation

(OPC FOUNDATION, 2014). Porém, desenvolvedores traduziram esse conjunto de normas e especificações em bibliotecas. Essas bibliotecas são disponibilizadas no formato DLL e assim podem ser utilizadas mais facilmente, diminuindo o tempo de desenvolvimento da aplicação. O padrão utilizado é o OPC-DA (*Data Access*) e as bibliotecas utilizadas neste trabalho para interpretar esse padrão são:

- OpcNetApi.dll;
- OpcNetApi.Com.dll;
- OpcNetApi.Xml.dll.

Essas bibliotecas estão disponibilizadas no *site* da Mesta Automation (MESTA, 2014). Elas foram desenvolvidas para a linguagem C# e para seu correto funcionamento devem ser declaradas no preâmbulo do código: `using Opc.Da.`

O Cliente OPC primeiramente se conecta a um Servidor OPC. Para isso os parâmetros de endereço e nome do servidor devem ser informados. A função utilizada para realizar a conexão é `server.Connect()`. Durante a conexão, o Cliente OPC cria um grupo de leitura de itens. Cada item deve ser adicionado ao grupo OPC para que se faça, periodicamente, a leitura desse item. Além do endereço do item é necessário indicar o tipo da variável, como por exemplo, inteiro, binário, decimal ou texto.

A cada período de atualização do Cliente OPC a função de tratamento dos dados é chamada. Essa função é responsável por dar algum destino aos dados obtidos. No caso do executável local, a cada atualização os dados são armazenados no banco de dados SQL.

Assim como o Cliente OPC lê as variáveis do Servidor, ele também pode escrever dados no CLP e SCADA. Esta função permite a um sistema PIMS executar tarefas características de outros sistemas, como por exemplo o MES. Além de visualizar os dados, o sistema PIMS pode influenciar no processo. Por exemplo, o sistema PIMS avalia o histórico da temperatura de um tanque e com base nesses dados liga um aquecedor. Esse tipo de função não é o foco principal de um sistema PIMS. Como o sistema PIMS criado neste trabalho será acessado por diferentes departamentos da empresa, não é interessante que todas essas pessoas possam interferir diretamente no processo.

Para realizar varreduras periódicas dos valores dos itens OPC, é utilizado um temporizador (*timer*). Ele especifica a taxa de atualização dos dados. Cada aplicação pode ter uma necessidade diferente quanto a taxa de atualização. É importante salientar que se essa taxa for muito alta o desempenho do sistema pode ficar comprometido, bem como

a capacidade do armazenador de dados. A seção a seguir mostra como as informações são armazenadas no banco de dados SQL.

### 3.1.2 Armazenamento das variáveis no banco de dados SQL

Para o correto funcionamento do executável local, o *software* Microsoft SQL Server deve estar instalado no computador destinado a banco de dados. As variáveis do processo, obtidas por meio do Servidor OPC, são armazenadas nesse banco de dados. Ao adicionar um item OPC é criada uma tabela exclusiva para ele. Esse tipo de tabela possui apenas duas colunas, o *time stamp* e o valor. O objetivo de criar uma tabela para cada item é melhorar o desempenho das consultas SQL e também evitar repetição de dados (redundância).

Para realizar a conexão com o banco de dados é necessário que o usuário informe o nome do servidor SQL. Caso ele tenha optado por autenticação SQL (no momento de instalação do SQL Server), ele também tem que informar o *login* e a senha. O executável local realiza um teste para informar ao usuário se a conexão foi bem sucedida. Ao longo do programa são utilizadas inúmeras conexões com o banco de dados. O comando SQL `SqlConnection con= new SqlConnection("Data Source=ServidorSQL; User ID=Login; Password=Senha")` é utilizado para realizar a conexão com o banco de dados. No código acima *con* é a variável de conexão e o texto entre parênteses é a *string* de conexão. É nessa *string* que são passados os parâmetros de Nome do Servidor SQL, Login e Senha do Usuário.

Se a conexão for bem sucedida, o programa pode criar o banco de dados e as tabelas, utilizando os comandos SQL: `create database` e `create table`. Para criação de tabela é necessário informar os parâmetros de cada coluna. Como essas tabelas são criadas pelo próprio executável local, não é necessário que o usuário tenha conhecimento de banco de dados e também se evita parâmetros errados nas tabelas.

Para entender melhor como foram armazenados os dados do programa, são apresentadas as tabelas do banco de dados. A Tabela 3.3 mostra os dados referentes aos usuários do *Site*. O gerenciamento dos usuários é feito apenas no programa local, pois assim o sistema fica menos vulnerável a entrada de pessoas não autorizadas. O usuário pode apenas alterar o *login* e a senha no *Site*. A coluna *login* informa o nome de acesso do usuário. Esse campo é também a Chave Primária (CP) da Tabela Usuários. Isso é

necessário para que não haja dois *logins* com o mesmo nome. A coluna Data de Criação informa há quanto tempo esse *login* foi criado. A coluna Data do Último *Login* informa a última vez que o usuário utilizou o sistema. A coluna cargo representa o Cargo Profissional ocupado pelo usuário dentro da empresa. Essa informação pode servir de filtro para salvaguardar dados sigilosos que algum setor da empresa não deve ter acesso.

Tabela 3.3: Tabela de Usuários.

Nome da Coluna	Tipo do Dado	Permitir Nulo	Detalhe
login	varchar(50)	Não permite	CP
senha	varchar(25)	Não permite	
data_criacao	date	Não permite	(getdate())
data_ultimo_login	date	Permite	
cargo	varchar(50)	Permite	

Tabela 3.4: Tabela de Servidores OPC.

Nome da Coluna	Tipo do Dado	Permitir Nulo	Detalhe
nome_servidor	varchar(100)	Não permite	CP
status_conexao	int	Permite	

A Tabela 3.4 contém os dados dos Servidores OPC cadastrados. O sistema PIMS só pode se conectar a servidores cadastrados. A coluna Nome do Servidor informa o endereço de conexão do Servidor. Esta coluna é Chave Primária, isso evita que um mesmo servidor seja adicionado duas vezes (redundância). Outra informação importante é o status da conexão que indica se o servidor está conectado ou desconectado.

Tabela 3.5: Tabela de Itens OPC.

Nome da Coluna	Tipo do Dado	Permitir Nulo	Detalhe
id	int	Não permite	Auto Inc.(1,1)
endereco_item	varchar(100)	Não permite	CP
tag	varchar(50)	Não permite	
servidor	varchar(100)	Não permite	CP
tipo_valor	varchar(15)	Não permite	

A Tabela 3.5 contém as variáveis do processo. Apenas podem ser visualizadas no *Site* as variáveis cadastradas no executável local. A coluna *id* identifica cada valor de item e o Auto Incremento (1,1) significa que os valores dessa coluna são preenchidos automaticamente, começando em 1 e sendo incrementado em uma unidade a cada registro. A coluna *Endereço do Item* informa o nome que foi utilizado no Servidor OPC para se referir à variável do processo. A *Tag* é uma informação que pode auxiliar na identificação de uma variável. Muitas vezes a empresa que realiza o projeto de Automação padroniza as *tags* por setores da indústria. A coluna *Servidor* indica o Servidor OPC ao qual pertence aquele item. A coluna *Tipo do Valor* identifica o tipo da variável, por exemplo, binária, inteira, ponto flutuante ou decimal.

Um fator importante na Tabela 3.5 é que ela tem a Chave Primária formada por duas colunas (*Endereço do Item* e *Servidor*), caracterizando uma chave primária composta. Isto é necessário pois não é permitido haver dois itens com o mesmo endereço (no mesmo servidor), porém pode haver dois itens iguais em servidores diferentes. O relacionamento entre o *Item* e o *Servidor* é de um-para-muitos (1:n) pois em um servidor pode haver vários itens, mas cada item está relacionado a um só servidor.

Tabela 3.6: Tabela de Valores das variáveis do sistema PIMS.

Nome da Coluna	Tipo do Dado	Permitir Nulo	Detalhe
data_time	datetime	Não permite	(getdate()), PK
valor	int,bool,decimal(18,2)	Não permite	

A Tabela 3.6 representa um modelo de tabela que é utilizado para cada Item OPC. Quando um item é adicionado, sua correspondente tabela também é criada. O nome dessa tabela é a junção de *tblValor* com o seu *Id* (na tabela de Itens OPC), por exemplo, *tblValor5*. Ao alterar o *tag* de cada item o nome da tabela também é modificado, o mesmo acontece quando o item é excluído.

A Tabela 3.6 carrega duas informações importantes de um item OPC, o seu valor e o *date/time* (ou *time stamp*). A coluna *Data Time* obtém a data e a hora do sistema, no momento em que o valor é armazenado. Isso é realizado através da função `sql getdate()`. O tipo de dado da coluna *Valor* depende do tipo da variável que foi declarado na tabela de Itens OPC, podendo ser binário, inteiro, decimal, ponto flutuante ou texto.

### 3.1.3 Telas do Executável local

O executável local é dividido em algumas telas de navegação. A seguir é mostrado o funcionamento de cada uma delas.

#### 3.1.3.1 Tela de Configuração do Banco de Dados

The screenshot displays the 'Configuração Banco de Dados' interface. At the top, a blue navigation bar includes tabs for 'Conexão', 'Itens OPC', 'Servidores OPC', 'Configuração Banco de Dados', 'Administrador de Usuários', and 'Tutorial do Programa'. The main area is split into two panels. The upper panel, 'Conexão', has input fields for 'Nome do Servidor SQL' (192.168.68.83\SQLEXPRESS), 'Login SQL' (sa), and 'Senha SQL' (masked). A 'Testar Conexão' button and a 'Status da Conexão' field are also present. The lower panel, 'Criação do Banco de Dados e das Tabelas SQL', contains a 'Criar banco de dados e Tabelas' button, a 'Nome do Banco de Dados Criado' field (dbPims), and a 'Status da Conexão' field.

Figura 3.9: Tela de Configuração de Conexão ao Banco de Dados.

A Figura 3.9 mostra a Tela de Configuração da conexão com o banco de dados SQL Server. Esta tela deve ser a primeira a ser configurada, pois sem a correta conexão com o banco de dados as outras também não irão funcionar corretamente. Essas configurações requisitadas na Tela de Configurações do Banco de Dados são as mesmas exigidas ao abrir qualquer Sistema de Gerenciamento de Banco de Dados, os SGBDs, por exemplo o Microsoft SQL Management Express. O usuário do programa pode testar a conexão. Se a conexão não for bem sucedida, ele deve buscar os dados corretos. Caso o resultado

da conexão seja satisfatório, com um *click* no botão Criar Banco de Dados e Tabelas o usuário gera todas as tabelas necessárias para o funcionamento do programa.

### 3.1.3.2 Tela de Gerenciamento dos Usuários do *Web Site*

	Login	Senha	Data Criação	ÚltimoLogin
▶	bernardo	123	21/06/2014	
	carlos	abc123	21/06/2014	
	joão	jo123	21/06/2014	
	Júlio	jul321	21/06/2014	
	maria	senha 123	21/06/2014	

Adicionar Usuários

Login:  Senha:  **Adicionar Usuário**

Remover Usuários

Login:  Senha:  **Remover Usuário**

Substituir Usuário

Login:  Senha:  **Substituir Usuário**

Status:

Figura 3.10: Tela de Gerenciamento dos Usuários do Sistema PIMS.

A Figura 3.10 mostra a tela de gerenciamento dos usuários que podem utilizar o *Site* do sistema PIMS. Nessa tela existe uma tabela com todos os usuários cadastrados e é possível adicionar, remover e alterar as informações dos usuários. Não é possível cadastrar dois *logins* iguais (pois esta coluna é Chave Primária). Para remover um usuário o programa lista as opções de usuários cadastrados e, ao clicar em uma das opções, o campo senha é preenchido automaticamente. Isso evita erros no processo de remoção de usuários e evita também remover um usuário o qual não se queria remover. Também é possível substituir o *login* e a senha do usuário.

### 3.1.3.3 Tela de Gerenciamento dos Servidores OPC

Lista de Servidores OPC:

	Nome do Servidor	Status da Conexão
▶	ICONICS.SimulatorOPCDA	
	RSLinx OPC Server	
*		

Adicionar Servidor OPC

**Adicionar Servidor**

Remover Servidor OPC

**Remover Servidor**

Substituir Servidor OPC

**Substituir Servidor**

Status:

Figura 3.11: Tela de Gerenciamento dos Servidores OPC.

A Figura 3.11 mostra a tela de gerenciamento dos servidores OPC aos quais o Sistema PIMS poderá se conectar. Pode ser feito o cadastro, alteração ou remoção dos Servidores OPC. Uma tabela dos servidores cadastrados é mostrada nesta tela. A tabela é atualizada a cada vez que o usuário entra na tela ou quando realiza as ações de cadastro, remoção ou substituição de um servidor.

### 3.1.3.4 Tela de Cadastro dos Itens OPC

A Figura 3.12 mostra a Tela de Cadastro dos Itens OPC. Primeiramente é necessário escolher o Servidor OPC, assim, a tabela é preenchida com os itens existentes nesse servidor. Isso evita o cadastro de itens em servidores errados. Para cadastrar os itens é necessário preencher os seguintes campos: endereço OPC do item, *tag* de identificação e o tipo de dado da variável. Ao adicionar um Item, a tabela de valores desse item é criada.

Servidor OPC: Escolha o Servidor para preencher a tabela de itens! ▼

Lista de Itens do Servidor  
Servidor: **Nome do Servidor**

	Endereço do Item	Tag	Servidor	Tipo do Dado
*				

Adicionar Item OPC

ENDEREÇO:  TAG:  TIPO:  **Adicionar Item**

Remover Item OPC

ENDEREÇO:  TAG:  TIPO:  **Remover Item**

Substituir Item OPC

ENDEREÇO:  TAG:  TIPO:  **Substituir Item**

Status:

Figura 3.12: Tela de Cadastro dos Itens OPC.

Caso o item seja removido, sua tabela também é removida e os dados são apagados do banco de dados.

### 3.1.3.5 Tela de Conexão com o Servidor OPC

A Figura 3.13 mostra a Tela de Conexão. Ela pode ser considerada a tela principal pois, se os parâmetros das outras telas já estiverem configurados, pouco se irá mexer nelas. Primeiramente o usuário deve verificar a conexão com o banco de dados. Para proceder com a conexão o usuário deve escolher o Servidor OPC no qual deseja conectar. O período de atualização dos dados também deve ser preenchido, ou então manter o período de atualização padrão, igual a 1000 milisegundos (1 segundo). Com isso basta clicar no botão Conectar.

A estrutura *try-catch* está sendo utilizada no executável local. Esta estrutura realiza a tentativa (*try*) de conexão. Caso não seja bem sucedida, ao invés de gerar um

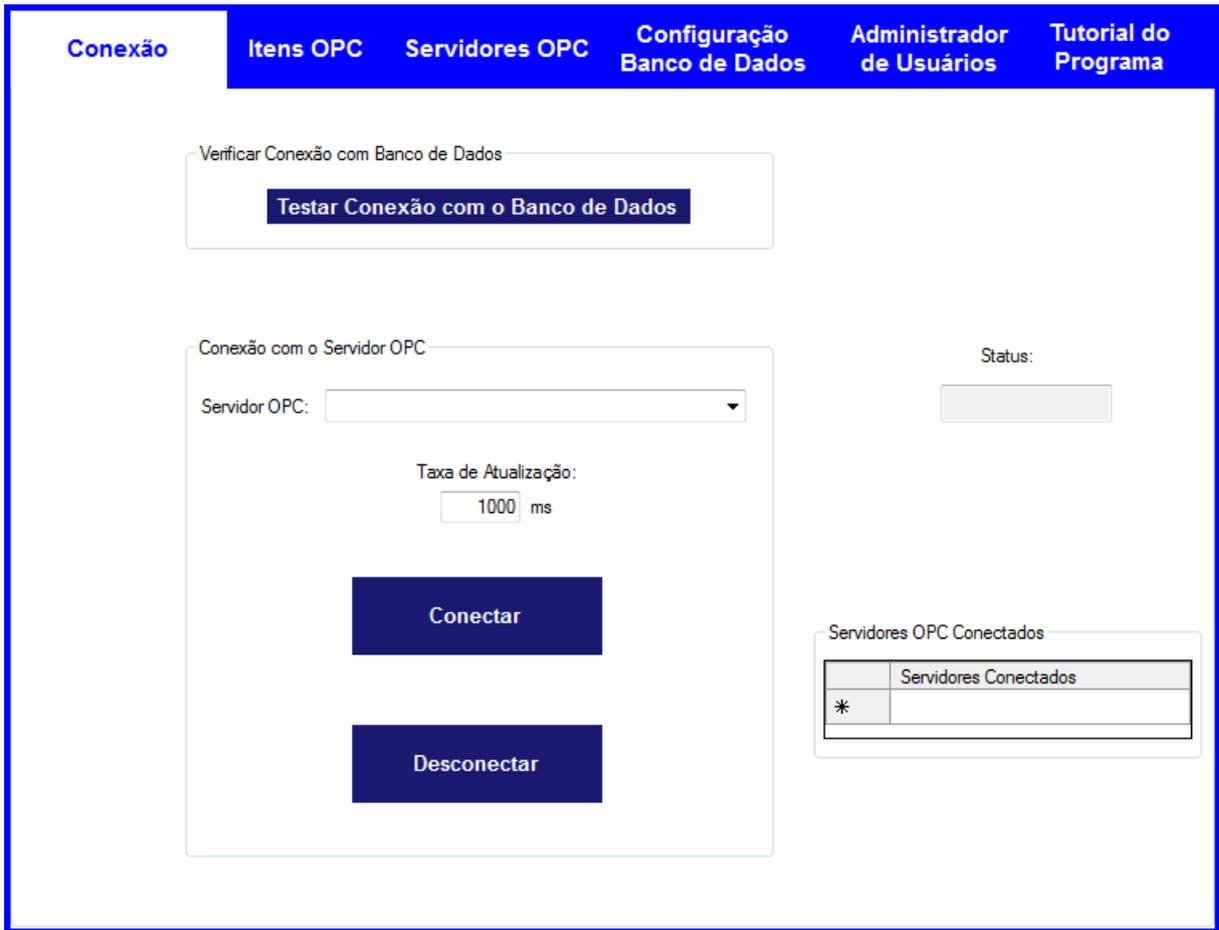


Figura 3.13: Tela de Conexão OPC.

erro a rotina do programa segue para a função auxiliar (*catch*). O programa mostra uma caixa de mensagem indicando se conexão foi bem sucedida. Para desconectar basta clicar no botão Desconectar. A Figura 3.14 mostra o Fluxograma da Conexão com o Servidor OPC. Os passos descritos anteriormente podem ser visualizados nesse fluxograma, para maior compreensão.

### 3.1.3.6 Tela de Tutorial do Executável local

A Figura 3.15 mostra a tela do tutorial, criada para guiar o usuário na utilização do executável local. Mesmo o *software* tendo a proposta de ser intuitivo, os passos podem ajudar o usuário. Os passos descritos nessa tela devem ser executados em sequência.

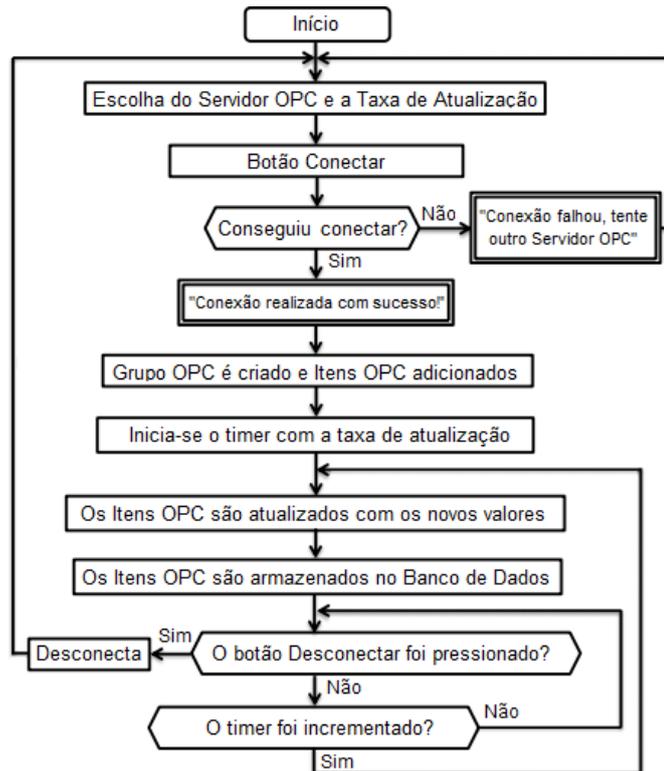


Figura 3.14: Fluxograma do processo de Conexão com o Servidor OPC.

Conexão	Itens OPC	Servidores OPC	Configuração Banco de Dados	Administrador de Usuários	Tutorial do Programa
<p>Aqui são apresentadas as instruções necessárias para realizar a Conexão OPC. Vá até as telas indicadas e realize os passos descritos aqui. Se der certo, prossiga para o próximo passo.</p>					
<p><b>Tela "Configuração Banco de Dados"</b></p> <ul style="list-style-type: none"> <li>✓ 1 - Vá até essa tela e preencha os campos Nome do Servidor SQL e, se tiver, login e senha. Caso dê algum erro, certifique que o Sql Server está Instalado ou Instale o SQL Management Express e utilize as informações de conexão que aparecem ao abri-lo.</li> <li>✓ 2 - Ao estabelecer a conexão clique no botão "Criar Banco de Dados e Tabelas". Se isso foi feito corretamente será escrito na caixa de texto a mensagem de verificação ou de erro.</li> </ul>					
<p><b>Tela "Servidores OPC"</b></p> <ul style="list-style-type: none"> <li>✓ 3 - Clique nessa tela, clique na opção "Cadastrar Servidor OPC" e cadastre um servidor. Se isso foi realizado corretamente, o Servidor cadastrado aparecerá na tabela.</li> </ul>					
<p><b>Tela "Itens OPC"</b></p> <ul style="list-style-type: none"> <li>✓ 4 - Vá até essa tela e escolha o Servidor OPC no qual irá cadastrar os itens. Para cadastrar um item digite seu Endereço OPC e o Tag de identificação. Escolha o tipo da variável. Caso não saiba as informações dos Itens, baixe o software ICONICS: <a href="http://www.iconics.com/">www.iconics.com/</a> Se isso foi feito corretamente o item aparece na Tabela.</li> </ul>					
<p><b>Tela "Conexão"</b></p> <ul style="list-style-type: none"> <li>✓ 5 - Vá até essa tela e verifique novamente a conexão com o banco de dados.</li> <li>✓ 6 - Na parte "Conexão" escolha o Servidor OPC a se conectar.</li> </ul> <p>Pronto. Agora basta clicar no botão "CONNECTAR" e verificar a mensagem que irá aparecer na tela. Se estiver tudo OK, a tabela ao lado direito mostrará o Servidor OPC conectado. Mas, caso dê algum erro, as causas mais prováveis são: servidor não encontrado ou não disponível, ou algum item não está disponível nesse Servidor OPC.</p>					

Figura 3.15: Tela de Tutorial do Executável Local.

## 3.2 *Web Site* de visualização dos dados

O *Web Site* do sistema PIMS tem o objetivo de acessar a base de dados SQL, onde foram armazenados os dados obtidos no Servidor OPC, e apresentá-los ao usuário por meio de gráficos e tabelas. Para isso o usuário irá escolher as variáveis e também o período de visualização.

### 3.2.1 *Master Page*

Para criação do *Site* foi utilizado o conceito de *Master Page*, que usa um cabeçalho comum para várias páginas, alternando apenas o corpo do texto (conteúdo de cada página). O recurso *Master Page* permite criar páginas padrão com conteúdo que pode ser replicado às demais páginas do *site*, isso é chamado de herança visual.

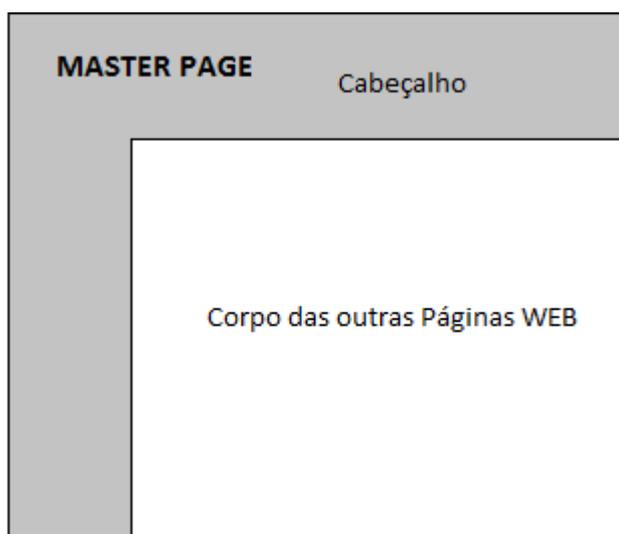


Figura 3.16: Organização de uma *Master Page* em relação às páginas Web.

A Figura 3.16 mostra um exemplo de posicionamento de uma *Master Page* em relação às telas *Web*. A parte em cinza é composta pela *Master Page* e a parte restante pelas outras telas. A Figura 3.16 também mostra o cabeçalho na parte superior.

A *Master Page* é um componente diferente de um *Web Form*. O primeiro é utilizado como ferramenta na construção de páginas *Web*, por isso uma *Master Page* nunca pode funcionar sozinha (sem uma página *Web*).

Ao inserir uma *Master Page* são adicionados trechos de código HTML: o cabeçalho (*head*), o corpo do texto (*body*) e a parte onde fica o conteúdo de cada página

(*ContentPlaceholder*).

No *Web Site* do sistema PIMS a *Master Page* possui um *banner* e um *menu* com quatro botões. Estes componentes podem ser visualizados na Figura 3.17.



Figura 3.17: *Master Page* do Sistema PIMS.

O *banner* contém a logomarca do Sistema PIMS e o *menu* possui botões que são utilizados para navegar entre as páginas. A seguir é mostrado o conteúdo das páginas do *Web Site* do sistema PIMS.

### 3.2.2 Tela de Construção de Gráficos

O objetivo desta tela é realizar as configurações necessárias para escolha da variável e também de seu intervalo de visualização.

A Figura 3.18 mostra a tela de configuração para construção de gráficos. Primeiramente o usuário deve escolher o Servidor OPC. Ao escolher o servidor, irão aparecer seus respectivos itens em um quadro. Para escolher um desses itens basta selecioná-lo e clicar no botão de adicionar. O item é retirado da lista de itens disponíveis e passado

Siga os passos  1  2  3 para construir o gráfico!

Escolha os itens para o gráfico

Servidor: ICONICS.SimulatorOPCDA

1

Logical.Motor  
Logical.SilasOnOff

Itens disponíveis

>>

<<

Numeric.Temperatura

Itens Selecionados

Escolha o modelo de gráfico

- Escolha data e hora, início e fim
- Últimos dias
- Últimos meses
- 2
- Média do dia
- Média do mês
- Porcentagem ligado
- Histórico (Tabela)

Escolha o período de visualização dos dados

3

17/06/2014 00:00:00 à 26/06/2014 00:00:00

dd/mm/aaaa hh mm ss dd/mm/aaaa hh mm ss

< junho de 2014 >

D	S	T	Q	Q	S	S
25	26	27	28	29	30	31
1	2	3	4	5	6	7

1  2  3 GERAR GRÁFICO

Figura 3.18: Tela de Construção de Gráficos.

para a lista de itens selecionados.

Para ser mais intuitivo, o *site* mostra a sequência dos passos necessários para construir um gráfico e também indica quando o passo foi realizado. Uma frase de aviso foi colocada no alto da tela: "Siga os passos 1, 2 e 3 para construir o gráfico". A cada passo realizado um *checkbox* é marcado, indicando que aquele passo foi concretizado. Por exemplo, se um item foi adicionado, o quadrado correspondente ao passo 1 é marcado, avisando o usuário que ele pode passar para o segundo passo.

O segundo passo para construir um gráfico é a escolha do modelo de gráfico. Os tipos de gráfico disponíveis são: Gráfico de Linha, Coluna e de Setores (conhecido como Gráfico de Pizza). Cada representação tem sua melhor aplicação dependendo da variável (analógica ou digital).

O terceiro passo é a escolha do intervalo de visualização dos dados. O usuário deve escolher valores de data e hora (início e fim). Para facilitar o usuário, o *site* possui um calendário para a escolha da data. Para filtrar melhor a busca, os campos de hora, minuto e segundo também podem ser preenchidos. Abaixo das caixas de texto, de preenchimento da data e hora, há indicações do formato que o usuário deve utilizar nos campos. Para a data deve ser usado "dd/mm/aaaa". Isso significa que o usuário deve preencher dois

dígitos para o dia e para o mês, e quatro dígitos referentes ao ano (com uma barra separando-os).

Ao final desses passos, os *checkbox* 1, 2 e 3 estarão preenchido e basta clicar no botão Gerar Gráfico.

### 3.2.3 Tela de Visualização do Gráfico

Ao pressionar o botão Gerar Gráfico será exibida a Tela de Visualização do Gráfico, que pode ser vista na Figura 3.19.

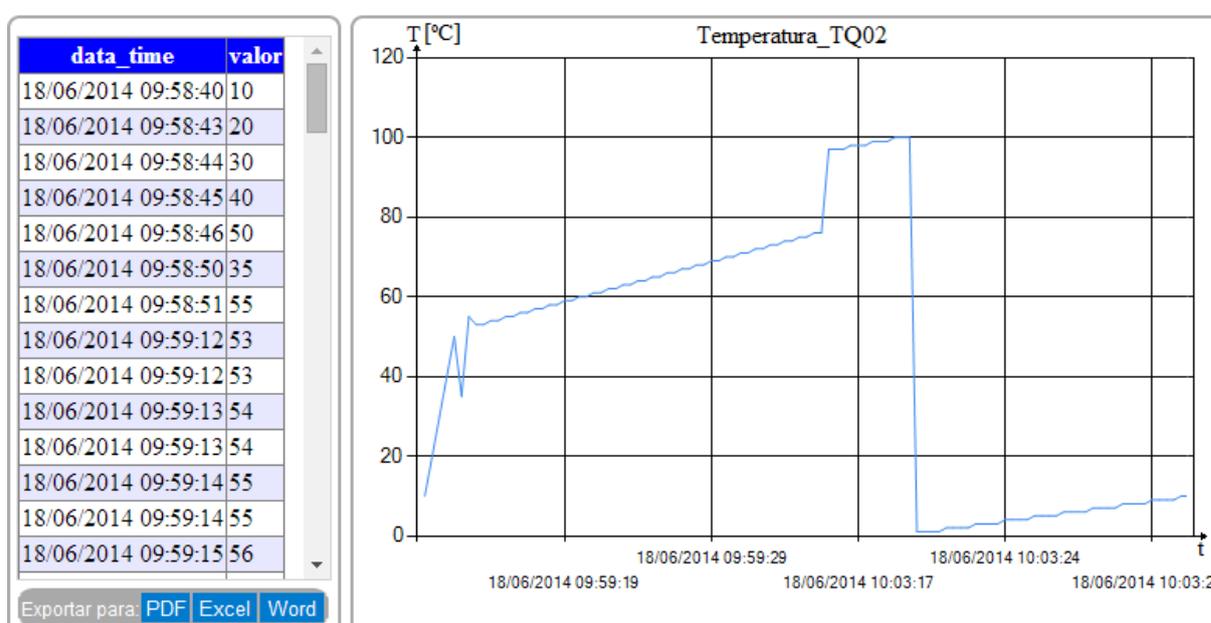


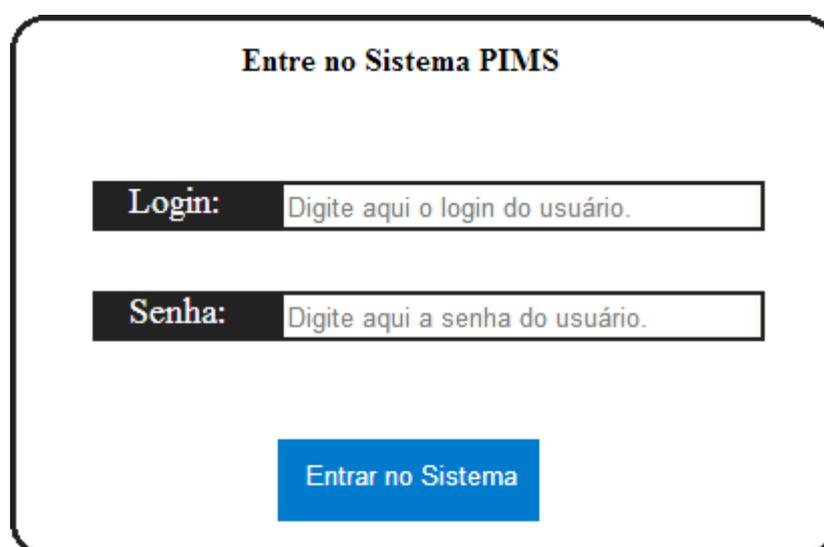
Figura 3.19: Tela de Visualização de Gráficos e Tabelas.

Esta tela possui o gráfico, a tabela com os valores obtidos e também as opções de exportar essas representações. O Sistema PIMS primeiro constrói a tabela, realizando uma *query* SQL, ou seja, consulta ao banco de dados. Os parâmetros do gráfico são passados da tela anterior por meio do método *session*. Os dados da tabela são utilizados para gerar o gráfico, isso evita uma nova consulta ao banco de dados. No gráfico, o eixo X (horizontal) representa os valores de tempo (*time stamp*) e o eixo Y (vertical) os valores das variáveis. Em um mesmo gráfico podem ser previstas várias séries. Cada série pode ser configurada com valores e configurações diferentes, como a cor da linha e o tipo de gráfico (linha, coluna ou pizza). Para gerar o gráfico, antes, o sistema verifica o modelo de gráfico escolhido pelo usuário e aponta a série que deve ser utilizada.

Na tela de visualização do gráfico foram implementadas opções de exportar as representações (gráfico e tabela) nos formatos PDF, XLS e DOC. Para exportar o arquivo no formato PDF foi utilizada a biblioteca *iTextSharp.dll*. Para exportar o arquivo em DOC e XLS utilizou-se a biblioteca System.IO (nativa da IDE do Visual Studio).

### 3.2.4 Tela de Autenticação do Usuário

Ao entrar no *Web Site* do sistema PIMS, a primeira tela visualizada é a de autenticação do usuário. Isso é necessário para restringir a utilização do sistema a apenas usuários cadastrados.



A imagem mostra a interface de autenticação do usuário. No topo, o título "Entre no Sistema PIMS" está centralizado. Abaixo dele, há dois campos de entrada de texto. O primeiro campo é rotulado "Login:" e contém o texto "Digite aqui o login do usuário.". O segundo campo é rotulado "Senha:" e contém o texto "Digite aqui a senha do usuário.". Abaixo dos campos, há um botão azul com o texto "Entrar no Sistema" em branco.

Figura 3.20: Tela de Autenticação do Usuário.

A Figura 3.20 mostra a tela de autenticação do usuário. Para identificar se o usuário tem cadastro no sistema é feita uma consulta SQL ao banco de dados. Essa consulta retorna a quantidade de registros encontrados com o *login* e senha especificados. Caso esse valor seja 1, significa que existe o cadastro desse *login* (e sua senha correspondente), se for igual 0 o acesso é negado ao usuário, pois não existe essa combinação no banco de dados.

Se o usuário possui cadastro no sistema, ao clicar no botão Entrar no Sistema o *site* o direciona à tela principal do sistema. Ao fazer isso, o *login* do usuário é passado para essa nova tela através do método *session*.

### 3.2.5 Tela de Alteração do *Login* e Senha do Usuário

O cadastro de *login* dos usuários que utilizam o *site* é feito apenas no executável local. Apenas a opção de substituir *login* e senha estão disponíveis no *site*. Como a tabela de usuário tem a data do último acesso, o usuário pode conferir se a data do último *login* corresponde à última vez que ele entrou no sistema. Isso pode indicar que uma pessoa utilizou o cadastro de outra, assim, surge a necessidade da substituição da senha.

**Trocar Login ou Senha do Usuário**

<b>Login antigo:</b> Usuário 1	<b>Senha antiga:</b> ***
<b>Novo Login:</b> Usuário 1 modificado	<b>Nova Senha:</b> ****

**Substituir**

Usuário substituído com sucesso!

Figura 3.21: Tela de Substituição do *login* e senha do Usuário.

A Figura 3.21 mostra a tela de substituição do *login* e/ou senha do usuário. Para realizar esta ação corretamente, os dados digitados de *login* e senha antigos devem estar previamente cadastrados no banco de dados.

### 3.2.6 Telas de Informações sobre sistemas PIMS, MES, ERP e Banco de Dados

A Figura 3.22 mostra a tela do *site* dedicada a informar o usuário conceitos das ferramentas de Manufatura Inteligente (PIMS, MES e ERP) e a Figura 3.23 sobre Banco de Dados. O conteúdo dessas páginas é o mesmo contido no Capítulo 2 deste trabalho (Revisão Bibliográfica).

### PIMS

O sistema PIMS é uma ferramenta que obtém os dados do processo produtivo, os armazena e os disponibiliza através de gráficos, tabelas e relatórios. Sua principal função é reunir a massa de dados em um banco de dados relacional centralizado, filtrá-los, transformando-os em informação e, com essa informação, gerar conhecimento, auxiliando no processo de tomada de decisões estratégicas.

**Fonte de dados**

Os dados do processo podem ser obtidos de um sistema supervisorio (SCADA) ou diretamente de CLPs e SCDCs (Sistemas de Controle Distribuído). A vantagem de se utilizar CLP é que ele é mais confiável, se comparado ao SCADA. Ele também apresenta menor atraso na obtenção dos dados. Por outro lado, no SCADA as variáveis já estão no formato de engenharia, assim, para utilizá-lo como fonte de dados não é necessário transformar as variáveis. Além disso, algumas variáveis derivadas da manipulação de variáveis do processo só existem no SCADA.

As informações apresentadas no PIMS podem ser de tempo real ou históricas. Os dados de tempo real mostram o valor instantâneo da variável, sendo portanto, mais utilizados para monitoramento do processo. Os dados históricos são as informações que foram armazenadas em bancos de dados relacionais. A consulta a esses dados pode ser feita através de uma linguagem de consulta a banco de dados como, por exemplo, SQL, Oracle, Access e MySQL.

O PIMS é uma ferramenta de grande importância para um engenheiro de processo. Ele precisa se embasar em informações para tirar conclusões sobre planta e assim ter o diagnóstico da produção. Quando se consegue um ótimo resultado em um processo, todas as configurações (set-ups) são armazenadas a fim de buscar essa combinação ótima. Com base nesse padrão de produção o sistema PIMS pode mostrar ao engenheiro qual parâmetro precisa melhorar para maximizar a eficiência.

O sistema PIMS unifica as bases de dados e, com essa centralização, qualquer computador da empresa consegue apresentar dados de diversas áreas do processo. Isso facilita o acesso à informação.

Figura 3.22: Tela de Informações sobre sistemas PIMS, MES e ERP.

### Banco de Dados

Atualmente há empresas que investem muito dinheiro para terem acesso às informações, sejam elas informações do mercado ou de sua própria produção. Com esse valor que a informação tem, faz-se necessário boas práticas para utilização dos bancos de dados, tornando-os confiáveis e organizados.

Banco de dados é uma coleção de dados inter-relacionados que carregam informações de uma aplicação ou do mundo real, organizados de forma que possam ser buscados facilmente por meio de consultas. Eles proporcionam confiabilidade, organização e facilidade de alteração, o que se torna muito difícil caso os dados estivessem armazenados, por exemplo, em arquivos de texto.

Os Sistemas de Gerenciamento de Banco de Dados (SGBD) são softwares com recursos específicos para facilitar a manipulação dos dados e o desenvolvimento de programas aplicativos. São exemplos de SGBD: SQL Server, Oracle Database, IBM DB2, MySQL e PostgreSQL. Além disso os SGBD podem oferecer Controle de Acesso aos dados e Gerenciador de Backups. No presente trabalho foi utilizado o SQL Server que é um SGBD gratuito da empresa Microsoft.

Os bancos de dados são compostos de tabelas. Também conhecidas como entidades, as tabelas representam um conjunto de dados que de alguma forma se relacionam. Elas possuem colunas (campos) que carregam um determinado tipo de informação como nome, cidade, telefone e idade. Cada linha dessa tabela é um registro, ou seja, um conjunto de informações. A chave primária primary key é utilizada quando se deseja identificar um registro, para isso ele deve ser único, unívoco, assim como o número de CPF identifica uma pessoa. As chaves estrangeiras (foreign key) são utilizadas quando se deseja representar um dado de outra tabela sem que esse dado seja inserido novamente.

A Figura 1 mostra a utilização da coluna ClienteID (Chave primária da tabela Clientes) como chave estrangeira na tabela Compras.

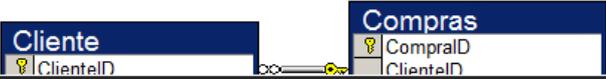


Figura 3.23: Tela de Informações sobre Banco de Dados.

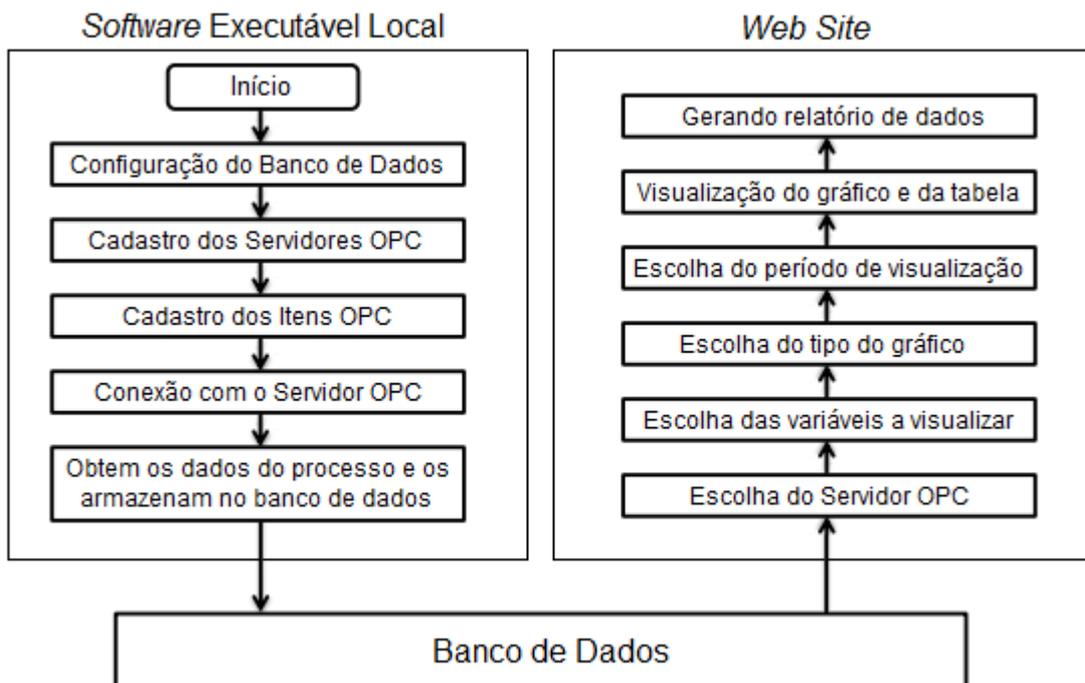


Figura 3.24: Fluxograma do funcionamento completo do sistema PIMS.

Nesta seção foi mostrado o desenvolvimento das duas partes do sistema PIMS separadamente. O fluxograma da Figura 3.24 mostra como as duas partes se interagem. Pode-se observar que a única ligação entre o *software* executável local e o *Web Site* é pelo banco de dados.

## 4. Resultados e Discussões

Este capítulo apresenta a descrição dos testes realizados e do funcionamento do sistema PIMS. São apresentados o sistema sob teste, sua configuração de rede e os resultados encontrados do *software* executável local e do *Web Site*.

### 4.1 Plataforma de *hardware* utilizada nos testes

Para fornecer dados ao sistema PIMS foi utilizada uma planta didática de controle de temperatura e nível de tanques, presente no Laboratório de Automação do IFMG campus Formiga. Ela foi desenvolvida pelos professores da área de automação<sup>1</sup> e alunos dos cursos de graduação em Engenharia Elétrica e técnico em Eletrotécnica. A Figura 4.25a mostra a planta utilizada para o teste.



Figura 4.25: Planta didática de Controle de Temperatura e Nível de Tanques.

Na planta didática é utilizado o CLP CompactLogix 5370 L2 - Allen Bradley (da fabricante Rockwell Automation), o qual pode ser observado na Figura 4.25b. O CLP foi programado com linguagem *ladder* pelos desenvolvedores da planta didática.

<sup>1</sup>Professores: Michelle Mendes Santos, Fábio Lúcio Corrêa Junior e Jonatham Silva Rezende.

Com a lógica de programação funcionando, configura-se o *software* RSLinx como Servidor OPC. Para realizar essa ação, uma conexão OPC é adicionada (Figura 4.26a). Na Figura 4.26b é mostrada a associação entre o programa *ladder* do Logix5000 com o CLP utilizado.

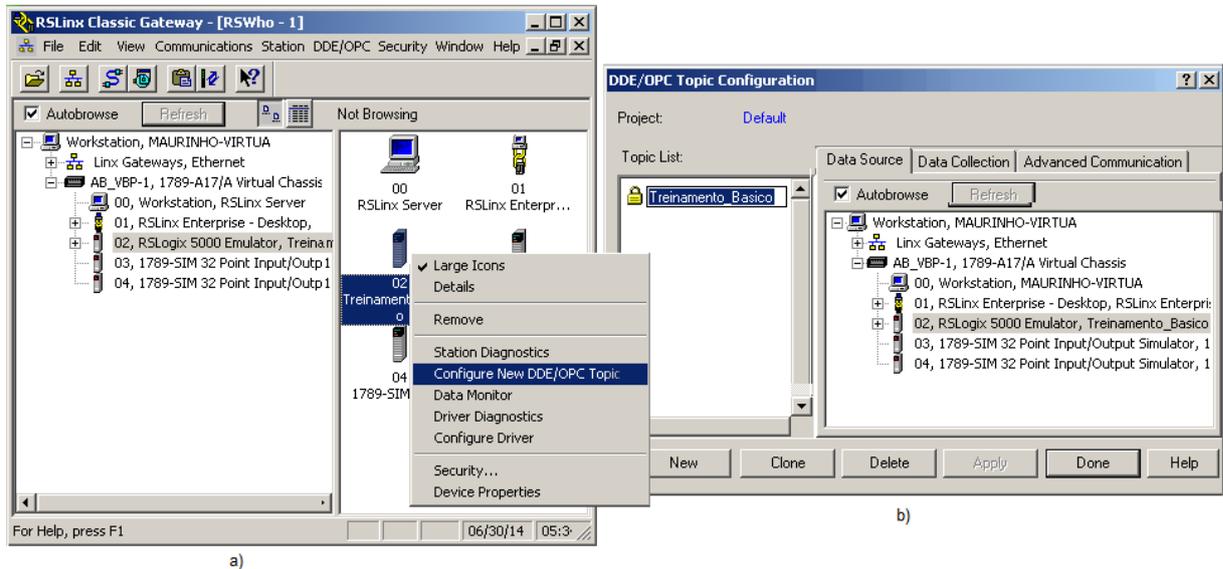


Figura 4.26: Configuração do Servidor OPC: a) Adicionando Conexão OPC e b) Associando o CLP ao programa do Logix5000.

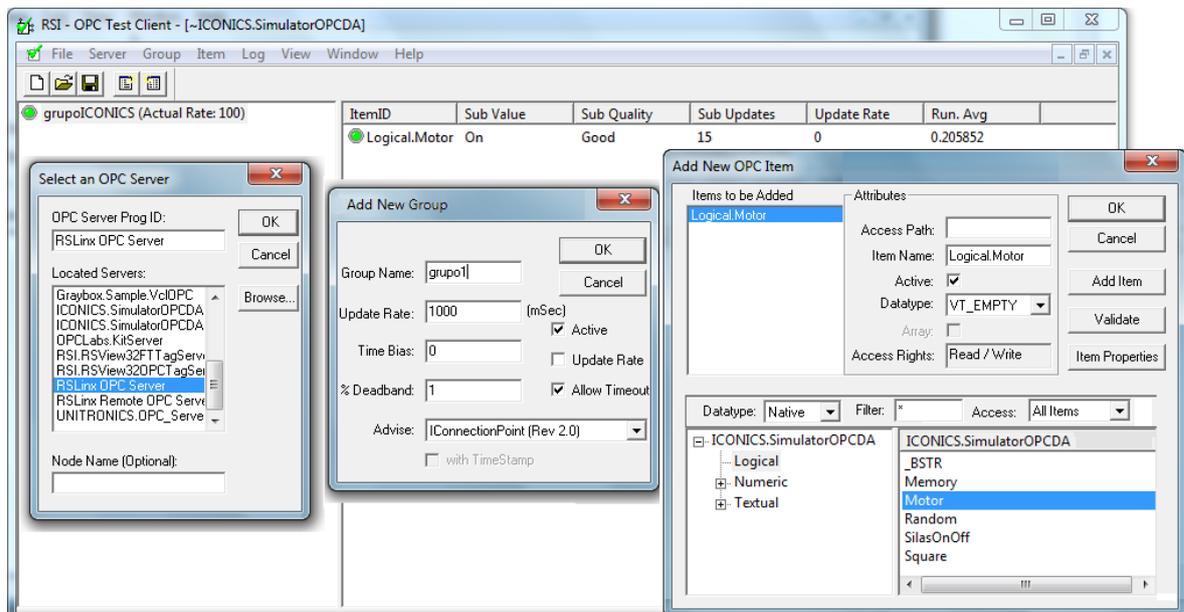


Figura 4.27: *Software* RSI - OPC Test Client (Cliente OPC).

Realizando essas etapas, o Servidor OPC é configurado como fonte de dados para o sistema PIMS. Para validar a conexão e assegurar que ela esteja funcionando, foi utili-

zado o *software* RSI - OPC Test Client (Cliente OPC utilizado para testar conexão com Servidores OPC). Esse cliente OPC é distribuído juntamente com o RSLinx. A Figura 4.27 mostra as configurações de escolha do Servidor, Grupo e Itens OPC.

#### 4.1.1 Configuração da Rede do Sistema de Teste

Para entender como o teste foi realizado, a Figura 4.28 ilustra a configuração de rede que integra o CLP, o Computador *Desktop* (PC) e o *Notebook* para os testes realizados neste trabalho.

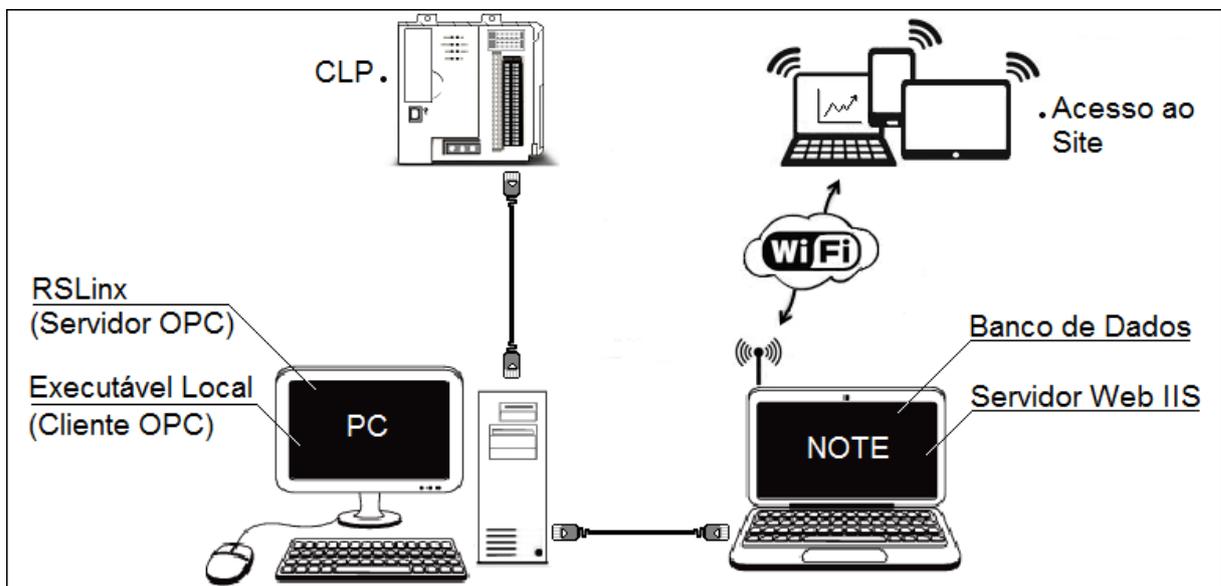


Figura 4.28: Sistema de teste e configuração da rede.

A integração entre o PC e o CLP é realizada por um cabo de rede e possibilita a obtenção dos dados do CLP por meio do *software* RSLinx configurado como Servidor OPC. Nesse PC também é executado o *software* Cliente OPC desenvolvido neste trabalho e detalhado na seção 3.1. Os dados que o Cliente OPC recebe são armazenados em um outro computador (neste caso, um *notebook*), que funciona como um banco de dados distribuído. A conexão entre PC e *notebook* é feita por meio de um cabo de rede. Para que o Cliente OPC possa armazenar dados no banco de dados no *notebook*, o IP deste deve ser informado.

O *notebook* é hospedeiro do *Web Site* do sistema PIMS. Para isso, ele está configurado como Servidor Web IIS (*Internet Information Services*), nativo do Windows. Assim, computadores que também estão na mesma rede podem acessar a página do sistema

PIMS.

No caso de uma aplicação industrial a arquitetura de rede pode ser modificada. Ao invés da utilização de cabos de rede pode ser utilizada conexão sem fios (*wireless*), tanto para comunicação CLP-PC quanto para PC-*Notebook*. Não é necessária a utilização de dois computadores, todas as subpartes do sistema PIMS (Cliente OPC, banco de dados e Servidor Web) podem ser executados no mesmo computador utilizado para o Servidor OPC. A hospedagem do *Web Site* não precisa ser, necessariamente, em um computador da empresa, ela pode contratar um serviço de hospedagem e acessar o *Web Site* pela internet.

## 4.2 Testes do Cliente OPC

Com o executável local iniciado, é feita a escolha do Servidor OPC *RSLinx OPC Server*. Ao pressionar o botão *Conectar*, a rotina de comunicação com o Servidor OPC se inicia. Uma caixa de texto informa o *status* da comunicação. O teste de conexão com o Servidor OPC *RSLinx* foi realizado com sucesso. Na Figura 4.29 é possível observar o nome do Servidor OPC escolhido, a mensagem da conexão e também o valor instantâneo de algumas variáveis. Para testar se a função estava respondendo corretamente foi realizado um teste com o nome de Servidor OPC inválido. Para esse caso, o programa retornou uma mensagem informando ao usuário da impossibilidade de conectar com o servidor especificado (Figura 4.30).

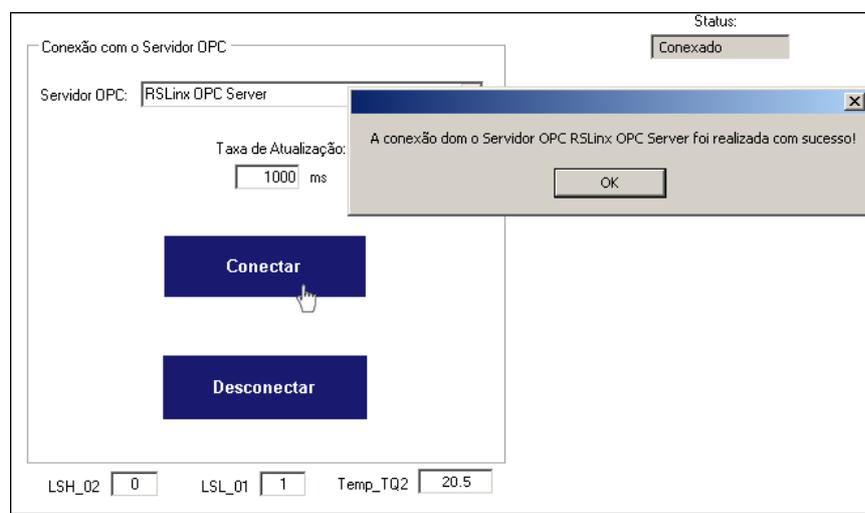


Figura 4.29: Conexão do Cliente OPC com o Servidor OPC.

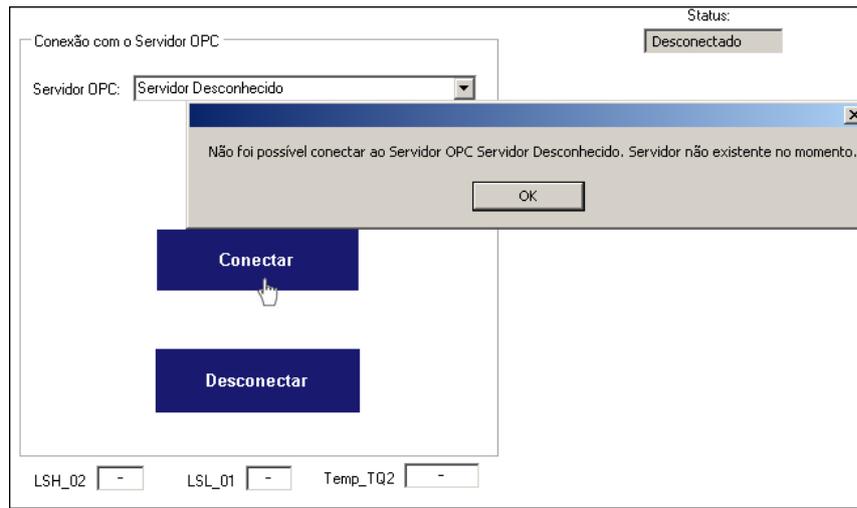


Figura 4.30: Conexão do Cliente OPC com o Servidor OPC não realizada.

O segundo teste foi feito para verificar se o Cliente OPC consegue armazenar os Itens OPC no banco de dados hospedado em outra máquina (*notebook*). Primeiro foi realizado o teste de comunicação entre as máquinas PC-*notebook*, executando o comando *ping* no *prompt* de comando de ambas as máquinas. Para saber se os dados foram gravados corretamente foi utilizado o SQL Server Management Studio. A Figura 4.31 mostra os valores da variável LSH\_02 armazenados no banco de dados SQL Server.

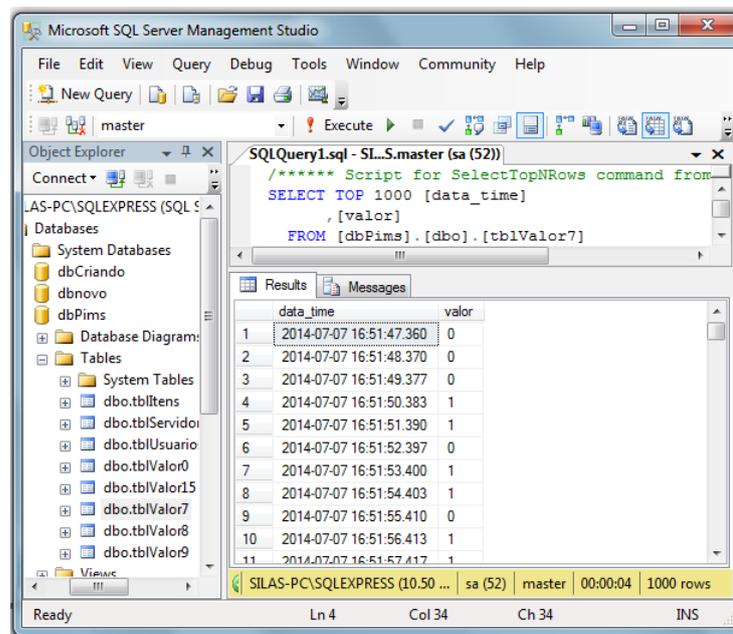


Figura 4.31: SQL Server Management Studio e os valores armazenados.

### 4.2.1 Teste de conectividade com outro Servidor OPC

É possível utilizar o Cliente OPC do sistema PIMS com qualquer Servidor OPC (não só com o RSLinx), pois a comunicação segue um padrão bem definido. Para demonstrar que isso é possível foi utilizado um outro Servidor OPC, o ICONICS OPC Simulator. Ele é um simulador de servidor OPC, gratuito, que fornece dados simulados no padrão OPC.

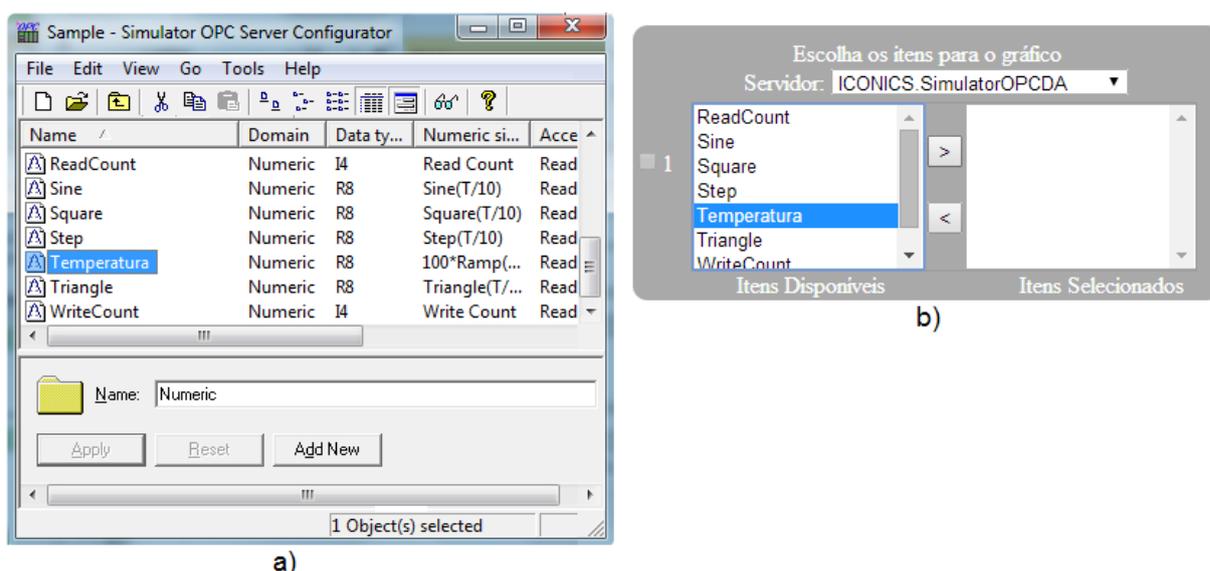


Figura 4.32: Comunicação OPC: a) ICONICS OPC Simulator e b) *Web Site*.

A Figura 4.32a mostra a tela de configuração do ICONICS OPC Simulator e na Figura 4.32b a visualização desse servidor OPC e seus respectivos Itens no *Web Site* do sistema PIMS.

## 4.3 Teste do *Web Site* do Sistema PIMS

No *notebook*, o servidor Web IIS foi configurado para hospedar o *site* do sistema PIMS. O *site* foi acessado por um outro computador dentro da mesma rede. Na primeira tela (Tela de Autenticação do Usuário) foi testada a conexão do *site* com o banco de dados. Para isso foi observado se o *login* e senha (previamente cadastrados) eram compatíveis. Como o banco de dados e a hospedagem estão na mesma máquina basta indicar o próprio nome do Servidor SQL local na *string* de conexão. A Figura 4.33 mostra a verificação do *login* e senha para entrar no *Web Site* do sistema PIMS.

Figura 4.33: Autenticação do Usuário para entrar no *Web Site* do sistema PIMS.

Também foi testada a fluidez do *layout* do *site*, ou seja, a adaptação da parte gráfica para diferentes resoluções de tela. A Figura 4.34 mostra a visualização do *site* em diferentes tamanhos de tela. Nos computadores *desktop*, *notebooks* e *tablets* o *layout* manteve a configuração. No *smartphone* o *layout* modificou para se adaptar ao tamanho reduzido da tela e o *banner* ficou levemente descaracterizado. Isso acontece por causa da existência da imagem, a qual não se adapta completamente ao *layout* fluido.



Figura 4.34: Visualização do *Web Site* do Sistema PIMS.

Foram verificadas a configuração e montagem de gráficos no *site* do sistema PIMS. A Figura 4.35a mostra a configuração do gráfico de porcentagem de uma variável digital (LSL\_01 - Chave de nível baixo do Tanque 1). Os *checkboxes* auxiliaram na identificação dos passos para se montar o gráfico. A Figura 4.35b mostra o resultado obtido com a configuração realizada. A tabela com os valores e o gráfico foram plotados. Esse tipo de gráfico indica qual a porcentagem de tempo que uma variável digital fica ligada e desligada. Em uma aplicação industrial esse gráfico pode monitorar a quantidade de

tempo que uma máquina permanece em funcionamento e também indicar o tempo de parada. Com essas informações pode-se conhecer a eficiência da máquina.

Escolha os itens para o gráfico  
Servidor: RSLinx OPC Server

Itens Disponíveis: LSH\_02, Temperatura\_TQ02, tag1  
Itens Selecionados: LSL\_01

Escolha o modelo de gráfico

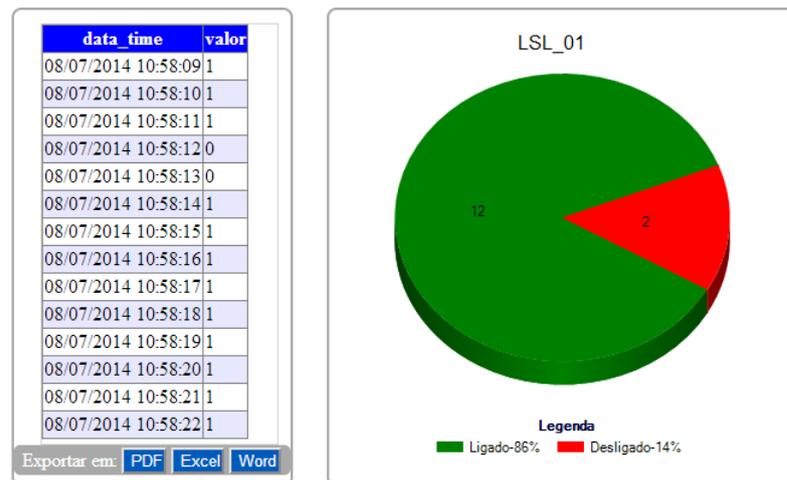
- Escolha data e hora, início e fim
- Últimos dias
- Últimos meses
- Média do dia
- Multi Variáveis
- Porcentagem ligado
- Histórico (Tabela)

Escolha o período de visualização dos dados

08/07/2014 10:58:09 a 08/07/2014 10:58:23  
dd/mm/aaaa hh mm ss dd/mm/aaaa hh mm ss

GERAR GRÁFICO

a)



b)

Figura 4.35: Exemplo de: a) montagem e b) visualização do gráfico.

Realizado a plotagem do gráfico e da tabela foi possível testar as opções de relatórios nos formatos PDF, DOC e XLS. A Figura 4.36 mostra como ficam os relatórios nesses formatos. No relatório em PDF foram inseridos o gráfico e a tabela, nos outros formatos (DOC e XLS) apenas a tabela foi inserida.

## 4.4 Análise Econômica do Sistema PIMS

Comparar historiadores de processo não é uma tarefa fácil, pois muitos fatores devem ser levados em conta como, por exemplo, as funcionalidades básicas de um historiador na coleta, compressão, armazenamento e recuperação de dados, as ferramentas

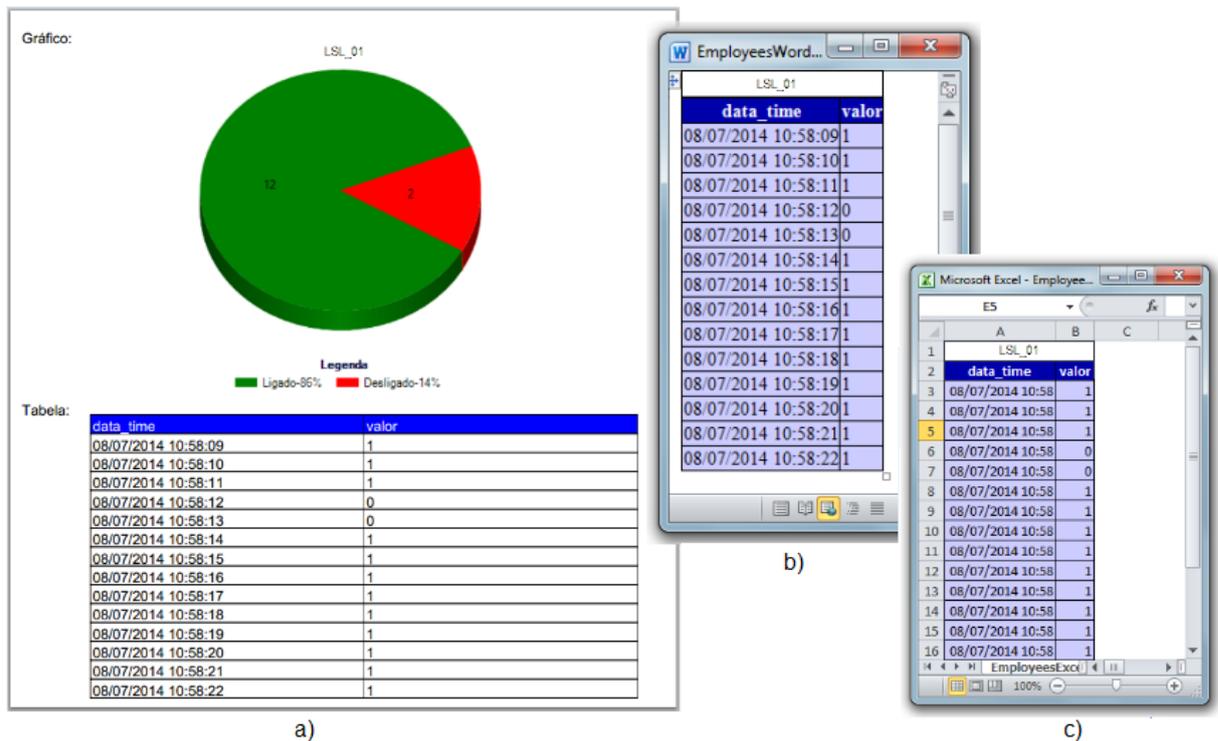


Figura 4.36: Relatório de dados nos formatos: a) PDF, b) DOC e c) XLS.

funcionais incorporadas ao historiador e também a utilização dos padrões da indústria como a comunicação OPC. A Tabela 1.2 (na Seção 1.3) mostra algumas características e preços dos historiadores PIMS do mercado.

O sistema PIMS desenvolvido possui características compatíveis com os grandes historiadores do mercado, oferecendo soluções como, por exemplo, obtenção de dados por padrão OPC, armazenamento de dados, sem limitação de *tags*, visualização de gráficos e tabelas, relatórios nos formatos PDF, XLS e DOC e também a visualização dos dados pelo navegador *Web*. Essas características mostram que o sistema PIMS desenvolvido é tecnicamente competitivo com os seus principais concorrentes.

Além das funcionalidades, um fator de extrema relevância para as empresas que adquirem esse tipo de produto é o preço a se pagar por ele. O custo inicial para produzir o sistema PIMS consiste no tempo de engenharia gasto no seu desenvolvimento, a aquisição de um computador e despesas contábeis como, por exemplo, INSS de uma Micro Empresa. O piso salarial de um Engenheiro é cerca de R\$ 6.154,00 por mês, atualmente (MUTUA, 2014). O período gasto no desenvolvimento foi de três meses e quinze dias. Para aquisição do computador o valor investido é de R\$ 1.500,00 e para pagamento

do INSS de um Microempreendedor Individual (MEI) R\$ 42,00 por mês (PORTAL DO EMPREENDEDOR, 2014). O custo inicial de desenvolvimento do sistema PIMS é de R\$23.186,00 como mostra a Equação 4.1.

$$\text{Custo inicial} = (3,5 * 6154) + 1500 + (3,5 * 42) = 23186 \quad (4.1)$$

Para cada sistema PIMS vendido será necessário instalar e configurar (*startup*) o sistema na empresa cliente. O tempo de instalação varia com a quantidade de variáveis a serem armazenadas. O tempo médio de instalação e configuração é de uma semana. Cada sistema terá um custo operacional de R\$ 2.000,00. Isso inclui o valor de R\$ 1538,50 para o desenvolvedor, despesas com transporte para deslocamento até o cliente e alimentação durante uma semana. Os valores que um investidor desse sistema irá receber está de acordo com a Equação 4.2.

$$\text{Valor} = n * (\text{PS} - \text{CO}) - \text{CI}. \quad (4.2)$$

Onde:  $n$  é o número de sistemas PIMS vendidos, PS é o preço cobrado por cada sistema, CO é o custo operacional por cada sistema vendido e CI é o custo inicial investido no desenvolvimento.

Com toda a solução do sistema PIMS sendo oferecida por R\$ 8.000,00 o retorno do investimento, por parte do investidor, virá na venda do quarto sistema PIMS (Equação 4.3). Após essa venda o investidor consegue um lucro de 300% por cada sistema vendido.

$$\text{Valor} = 4 * (8000 - 2000) - 23186 = +\text{R\$ } 814,00. \quad (4.3)$$

O FactoryTalk Historian (versão com limitações de *tags*) é o historiador com menor preço dentre os listados na Tabela 1.2. A solução completa do sistema PIMS (licença, instalação e suporte) pode ser adquirido por um preço inferior ao *software* apresentado. Além disso, o preço do FactoryTalk Historian se refere apenas a licença de utilização. Para configurá-lo é necessária uma empresa de Automação especializada em sistemas PIMS.

A indústria cliente pode contratar o serviço de hospedagem ou também optar por hospedar o *Web Site* em um computador da indústria. O serviço de hospedagem (armazenamento do site e do banco de dados) pode ser adquirido por R\$ 20,00 a R\$ 80,00 mensais, dependendo da capacidade de armazenamento do banco de dados requerido (UOL HOST, 2014). Para este caso também deve ser adquirido um domínio, por exemplo, *.com.br* que custa à indústria o valor de R\$ 30,00 por ano (REGISTRO.BR, 2014).

## 5. Conclusões

Este trabalho mostrou a importância da informação de chão de fábrica nos setores corporativos de uma indústria. A manufatura inteligente evita ilhas de informação e auxilia no controle e gerenciamento do processo produtivo.

A comunicação OPC é o padrão mais utilizado nos sistemas de dados da automação, assim, com a utilização desse padrão no sistema PIMS desenvolvido torna-o compatível com grandes historiadores do mercado. Os testes mostraram que o cliente OPC do sistema PIMS pode se conectar a diferentes Servidores OPC, devido à padronização da comunicação OPC. Essa característica o torna interoperável em sistemas de automação de diferentes fornecedores.

Por meio do *software* executável local, detalhado na seção 3.1, pôde-se configurar os Servidores e Itens OPC e também cadastrar os usuários que podem acessar o *Web Site*.

Ao se fazer os testes, pôde-se observar uma característica de flexibilidade em relação à arquitetura de rede, onde as partes não precisam estar funcionando, necessariamente, em um só computador. Isso permite que o sistema adapte melhor a configuração de rede da indústria que utilizar o sistema PIMS.

Para a montagem dos gráficos, a indicação das etapas a serem seguidas, auxiliaram os usuários do *Web Site* do sistema PIMS. É possível exibir gráficos de linha, coluna e de pizza, inclusive saber a porcentagem de tempo de parada de uma máquina ou equipamento. Foi plotada, também, a tabela com a descrição de cada valor e seu o respectivo *time stamp*. As opções de gerar relatórios facilitam a elaboração de documentos técnicos embasados nas informações do processo.

A análise de custos e funcionalidades, em comparação com os principais concorrentes, mostrou que o sistema PIMS possui as principais características dos melhores historiadores comercializados e pode ser vendido a um preço inferior, além da possibilidade de adequar o sistema às necessidades de cada cliente.

Além dos ganhos quantitativos houve também ganhos qualitativos como, por exemplo, melhor organização dos dados, maior controle do processo, rapidez no acesso aos dados e também alta disponibilidade da informação de chão de fábrica.

## 5.1 Sugestão de trabalhos futuros

Algumas ferramentas podem ser adicionadas ao sistema PIMS para melhorar seu desempenho e funcionalidade. Uma sugestão é desenvolver uma ferramenta que procura os itens OPC existentes. Essa ferramenta, também chamada de *Item Browser*, obtém todos os itens OPC existentes de um determinado Servidor OPC.

Como o sistema PIMS pode ser visualizado pela internet, outra sugestão é a utilização do padrão OPC-XML. Ele permite a criação de clientes OPC diretamente em aplicações *Web*.

A versão gratuita do SQL Server utilizado (Express), possui uma restrição de espaço de armazenamento, que é de até 10 GB (MICROSOFT, 2014). Por isso pode ser utilizado o banco de dados MySQL, além de ser gratuito é o mais utilizado em aplicações Web (WELLING, THOMSON, 2005).

O trabalho realizado foi testado apenas em uma intranet (rede interna do IFMG Campus Formiga), então, pôde-se disponibilizar o *site* na internet, para isso é necessário adquirir um domínio e uma hospedagem, sendo que a hospedagem também pode ser feita em um computador do próprio cliente.

## 6. Referências Bibliográficas

ALMEIDA, D. C.; **A importância da Manufatura Inteligente na Engenharia de Produção**. Artigo selecionado pelo Crea Digital. Pouso Alegre MG, Março 2004.

ASPENTECH; **Software PIMS InfoPlus 21**. Disponível em: <www.aspentech.com>. Acesso em: 29 de junho de 2014.

BARROS, B.; **Layout Fluido ou Líquido e técnicas de Análise de renderização dos navegadores Web**. Disponível em: <www.designadaptavel.com.br>. Acesso em: 30 de junho de 2014.

CARVALHO, F. B.; TORRES, B. S.; FONSECA, M. O.; SEIXAS FILHO, C.; **Sistemas PIMS: conceituação, usos e benefícios**. Anais do Seminário de Automação de Processos, Santos, 2003.

CAIÇARA, J. C.; **Sistemas Integrados de Gestão - ERP: uma abordagem gerencial**. Editora InterSaberes, p. 53, Curitiba, 2012.

CONTROL; **OSI PI Versus Aspen Info Plus 21**. Global Online Community of Automation professionals. Disponível em: <www.control.com>. Acesso em: 28 de junho de 2014.

FRANQUINI, F.; **Capítulo 3 - Manipulating Data SQL**. Engenheiro de Computação da empresa Softplan. Disponível em: <www.certificacaobd.com>. Acesso em: 14 de julho de 2014.

GAUDÊNCIO, E.; **Conceituando Banco de Dados e SGBD**. Disponível em: <certificacaobd.com.br>. Acesso em: 14 de junho de 2014.

KITE; **O que é OEE e quais as vantagens de medi-lo?** Empresa de soluções em Automação Industrial. Disponível em: <www.kitemes.com.br>. Acesso em: 2 de Maio de 2014.

MECATRÔNICA ATUAL; **O que é OPC?** Revista Mecatrônica Atual nº 42, Editora Saber, Agosto, 2013.

MEDEIROS, L. F.; **Banco de Dados: princípios e prática**, Editora IBPEX, pp.

14-16, Curitiba, 2007.

MELO, O. V.; **Gerenciamento das Informações Industriais**. Monografia apresentada ao curso de Controle e Automação na Universidade de Ouro Preto, 2012.

MESTA, Automation; **Client OPC With C#**. Biblioteca de interpretação do Protocolo OPC. Disponível em: <[www.mesta-automation.com](http://www.mesta-automation.com)>. Acesso em: 13 de maio de 2014.

MICROSOFT; **SQL Server Express Edition**. Disponível em: <[www.microsoft.com](http://www.microsoft.com)>. Acesso em: 13 de Julho de 2014.

MTHOMAZINI; **Consultoria especializada em ERP**. Empresa de consultoria M. Thomazini. Disponível em: <[www.mthomazini.com.br](http://www.mthomazini.com.br)>. Acesso em: 26 de junho de 2014.

MUTUA; **Salário Mínimo Profissional é de R\$ 6.154 em 2014**. Mútua - Caixa de Assistência dos Profissionais dos Creas. Disponível em: <[www.mutua.com.br](http://www.mutua.com.br)>. Acesso em: 11 de Julho de 2014.

NOGUEIRA, T. A.; **Redes de Comunicação para Sistemas de Automação Industrial**. Monografia apresentada ao curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto. Agosto 2009.

OPC FOUNDATION; **The Interoperability Standard for Industrial Automation**. Fundação de divulgação das características do padrão OPC. Disponível em: <[opcfoundation.org](http://opcfoundation.org)>. Acesso em: 09 de junho de 2014.

OSISOFT; **PI System - the industry standard in enterprise infrastructure for management of real-time data and events**. Empresa fabricante de historiadores de dados. Disponível em: <[www.osisoft.com](http://www.osisoft.com)>. Acesso em: 29 de junho de 2014.

PORTAL DO EMPREENDEDOR; **Definição do Microempreendedor Individual - MEI**. Disponível em: <[www.portaldoempreendedor.gov.br](http://www.portaldoempreendedor.gov.br)>. Acesso em: 22 de julho de 2014.

REGISTRO.BR; **Registro de Domínio .com.br**. Disponível em: <[registro.br](http://registro.br)>. Acesso em: 29 de junho de 2014.

SEIXAS, F. C.; **PIMS – Process Information Management System**. Notas de Aula, Capítulo 6, Disponível em: <[www.cpdee.ufmg.br/~seixas/](http://www.cpdee.ufmg.br/~seixas/)>. Acesso em: 11 de

Abril de 2014.

STAIR, M. R; **Princípios de sistemas de informação: uma abordagem gerencial**, 4ª Edição, Editora LTC, Rio de Janeiro, 2004.

TIERNEY, S.; **FactoryTalk Historian SE**. Data Logging Options For CompactLogix and ControlLogix. Disponível em: <[www.insightsinautomation.com](http://www.insightsinautomation.com)>. Acesso em: 28 de junho de 2014.

UOL HOST; **Preços e planos de hospedagem de Sites**. Disponível em: <[www.uolhost.com.br](http://www.uolhost.com.br)>. Acesso em: 14 de Julho de 2014.

VISUAL STUDIO; **Site oficial do *software* Visual Studio**, empresa Microsoft, versões 2013 e 2010. Disponível em: <[www.visualstudio.com](http://www.visualstudio.com)>. Acesso em: 16 de junho de 2014.

WELLING, L.; THOMSON, L.; **PHP e MySQL: Desenvolvimento Web**. Editora Pearson, 3ª Edição, 2005.