

Laboratório Virtual de Controle e Automação com Arquitetura Baseada em Serviço

André Luís Silva Barbosa

Dissertação de Mestrado submetida à Coordenadoria do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Instrumentação e Controle

Péricles Rezende Barros, Ph.D.

Orientador

Campina Grande, Paraíba, Brasil

©André Luís Silva Barbosa, Agosto de 2010

Laboratório Virtual de Controle e Automação com Arquitetura Baseada em Serviço

André Luís Silva Barbosa

Dissertação de Mestrado apresentada em Agosto de 2010

Péricles Rezende Barros, Ph.D.
Orientador

Maria de Fátima Queiroz Vieira, Ph.D, UFCG
José Sérgio da Rocha Neto, D.Sc, UFCG

Campina Grande, Paraíba, Brasil, Agosto de 2010

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

B2381 Barbosa, André Luís Silva

Laboratório virtual de controle e automação com arquitetura baseada em serviço / André Luís Silva Barbosa. — Campina Grande, 2010.

91 f. : il.

Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.

Referências.

Orientador: Prof. Dr. Péricles Rezende Barros.

1. Laboratório Virtual. 2. Educação a Distância. 3. Ambiente Virtual de Aprendizagem. 4. Controle PID. 5. WCF. 6. OPC. I. Título.

CDU – 681.5(043)

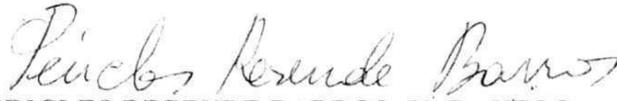
681.515.8(043)

UFCG - BIBLIOTECA - CAMPUS I	
2577	05-05-011

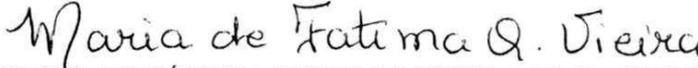
**LABORATÓRIO VIRTUAL DE CONTROLE E AUTOMAÇÃO COM
ARQUITETURA BASEADA EM SERVIÇO**

ANDRÉ LUIS SILVA BARBOSA

Dissertação Aprovada em 22.09.2010


PÉRICLES REZENDE BARROS, Ph.D., UFCG
Orientador


JOSÉ SÉRGIO DA ROCHA NETO, D.Sc., UFCG
Componente da Banca


MARIA DE FÁTIMA QUEIROZ VIEIRA, Ph.D., UFCG
Componente da Banca

CAMPINA GRANDE - PB
SETEMBRO - 2010

Dedicatória

Dedico esta conquista a meus pais, Antônio Eduardo Barbosa e Doralice Francisca do Espírito Santo Barbosa, exemplos a serem seguidos.

Agradecimentos

A DEUS pela proteção de cada dia.

Aos meus pais, Antônio Eduardo Barbosa e Doralice Francisca do Espírito Santo Barbosa, que não mediram esforços para me oferecer uma educação de qualidade, me incentivaram e sempre me apoiaram em todos os momentos da minha vida.

Aos meus irmãos, Paulo Eduardo, Maria Clara Stela e Pedro Yóssis, que cresceram comigo e sempre estiveram torcendo por mim.

A minha namorada, Maria Auriane, que conviveu comigo durante o processo de realização deste trabalho, me apoiando, incentivando e sabendo administrar minhas ausências.

A todos os meus amigos que me acompanharam e participaram comigo de momentos de angústia e lazer. Em especial a Raffael Carvalho, Bruno Almeida e Francisco Sérgio Clementino, amigos para todas as horas.

Ao professor Péricles Rezende Barros pela orientação e contribuição na minha formação acadêmica desde os primeiros trabalhos de iniciação científica.

Aos professores membros da banca examinadora, José Sérgio e Fátima, pela atenção e sugestões.

Aos amigos, colegas, professores e funcionários do Laboratório de Instrumentação Eletrônica e Controle da UFCG que colaboraram para a realização deste trabalho.

Ao CNPq que viabilizou a realização desta pesquisa com apoio financeiro.

Resumo

Os processos em escala de laboratório de centros de pesquisas e universidades são utilizados para testes, treinamento, desenvolvimento e demonstrações, auxiliando estudantes e pesquisadores na atuação e observação de vários fenômenos que podem acontecer no meio industrial. Nesse contexto, são apresentadas as atividades relacionadas ao desenvolvimento de uma ferramenta de Laboratório Virtual com arquitetura baseada em serviços WCF e OPC para possibilitar o acesso remoto, por meio da Internet ou intranet, a simuladores e processos industriais reais. A ferramenta disponibiliza inicialmente a observação e atuação, com componente de controle PID, em três processos (dois simuladores e um processo real). São apresentados ainda, descrição das tecnologias utilizadas, modelagem e estudos de casos com análise dos resultados para cada um dos três processos disponibilizados na ferramenta.

Palavras chave: Laboratório Virtual; Educação a Distância; Ambiente Virtual de Aprendizagem; Controle PID; WCF; OPC.

Abstract

The lab scale processes in research centers and universities laboratories are used for testing, training, developing and demonstrating, supporting students and researchers in acting and observing several phenomena that can occur in the industrial environment. In this context, we present the activities related to the development of a tool for architectures WCF and OPC based-service Virtual Laboratory to enable remote access via Internet or private network, to simulators and real industrial processes. Initially, the tool provides observation and acting, with a PID Control component for three cases (two simulators and a real process). We also present the description of the technologies, modeling and case studies with analysis of the results for each of three available processes in the tool.

Keywords: Virtual Laboratory; Distance Education; Virtual Learning Environment; PID Control; WCF; OPC.

Índice

1	Introdução	4
1.1	Motivação	5
1.2	Objetivos	6
1.3	Estrutura da dissertação	7
2	<i>E-Learning</i> e Ambientes Virtuais de Aprendizado	9
2.1	<i>e-Learning</i>	10
2.2	Laboratórios Virtuais de Controle e Automação	13
2.3	Análise comparativa	14
2.4	Conclusão	15
3	Tecnologias de Comunicação Utilizadas	17
3.1	<i>Windows Communication Foundation</i> - WCF	17
3.1.1	Primeiros mecanismos e protocolos de comunicação entre computadores e demais dispositivos	18
3.1.2	Arquitetura Orientada a Serviço - SOA (<i>Service-Oriented Architecture</i>)	20
3.1.3	Protocolo de Acesso Simples a Objeto - SOAP (<i>Simple Object Access Protocol</i>)	21
3.1.4	<i>Web Services</i>	23
3.1.5	<i>Windows Communication Foundation</i> (WCF)	26
3.1.6	Arquitetura WCF	28
3.2	<i>OLE for Process Control</i> - OPC	30

3.3	Conclusão	33
4	Laboratório Virtual para Estudos de Controle	35
4.1	Análise de Requisitos do <i>Software</i> e Especificações	35
4.2	Arquitetura do Sistema	37
4.3	Codificação e Descrição do Sistema	38
4.3.1	Serviço OPC	39
4.3.2	Serviço WCF	41
4.3.3	Banco de Dados	42
4.3.4	<i>Software</i>	45
4.4	Conclusão	52
5	Descrição dos Processos	54
5.1	Reator de Bateladas Sequenciais - RBS	54
5.2	Tanques de Agitação Contínua (<i>CSTR - Continuous Stirred-Tank Reactor</i>)	61
5.3	Módulo Peltier	64
5.4	Conclusão	67
6	Estudo de caso	69
6.1	Simulação de CSTR	69
6.2	Simulação RBS	70
6.3	Módulo Peltier	76
6.4	Conclusão	78
7	Considerações Finais	79
7.1	Trabalhos Futuros	80
	Referências Bibliográficas	82
A	Especificações OPC	85
B	Simulador CSTR no MatLab	88

Lista de Símbolos e Abreviaturas

ΔH	Reação de Calor	[kJ/kmol]
ρ	Densidade do Componente dentro do Reator	[kg/m ³]
A	Área de secção transversal do tanque	[m ²]
C_p	Calor Específico	[kJ/kgK]
J_b	Fluxo Ascendente de Sólidos Suspensos Decorrente da Alimentação do Decantador	
J_g	Fluxo Descendente de Sólidos Suspensos Devido à Força Gravitacional	
Q	Fluxo de saída do reator	[m ³ /s]
Q_i	Fluxo de entrada do reator	[m ³ /s]
S_i	Matéria Orgânica Solúvel Inerte	[mgDQO/l]
S_O	Oxigênio Dissolvido	[−mgDQO/l]
S_s	Substrato Rapidamente Biodegradável	[mgDQO/l]
S_{ND}	Nitrogênio Orgânico Solúvel Biodegradável	[mgN/l]
S_{NH}	Amônia	[mgN/l]
S_{NO}	Nitrato e Nitrito	[mgN/l]
T_c	Temperatura do Refrigerante	[K]
T_f	Temperatura de Alimentação	[K]
U	Coefficiente de Transferência de Calor	[J/min/K/m ²]
V	Volume do Reator	[l]

X_i	Matéria Orgânica Particulada Inerte	[mgDQO/l]
X_p	Produtos Particulados Resultantes do Decaimento da Biomassa	[mgDQO/l]
X_s	Substrato Lentamente Biodegradável	[mgDQO/l]
X_{ND}	Nitrogênio Orgânico Particulado Biodegradável	[mgN/l]
ASM1	<i>Activated Sludge Model N° 1</i>	
CLP	Controlador Lógico Programável	
CMC	Comunicação Mediada por Computador	
COM	<i>Component Object Model</i>	
CSTR	Tanques com Agitação Contínua (<i>Continuous Stirred-Tank Reactor</i>)	
DCOM	<i>Distributed Component Object Model</i>	
DDE	<i>Dynamic Data Exchange</i>	
HTTP	<i>Hypertext Transport Protocol</i>	
IEEE	<i>Institute of Electrical and Electronics Engineers</i>	
IIS	<i>Internet Information Service</i>	
LIEC	Laboratório de Instrumentação Eletrônica e Controle	
MV	Variável Manipulada (<i>Manipulated Variable</i>)	
OLE	<i>Object Linking and Embedding</i>	
OPC	<i>OLE for Process Control</i>	
PV	Variável do Processo (<i>Process Variable</i>)	
RAD	<i>Rapid Application Development</i>	
RBS	Reator de Bateladas Sequenciais	
RFiD	<i>Radio-Frequency IDentification</i>	
RMI	<i>Remote Method Invocation</i>	
SIS	Sistemas Instrumentados de Segurança	
SOA	Arquitetura Orientada a Serviço (<i>Service-Oriented Architecture</i>)	

SOAP	Protocolo de Acesso Simples a Objeto (<i>Simple Object Access Protocol</i>)
SP	Ponto de Operação (<i>Set Point</i>)
TIC	Tecnologia da Informação e Comunicação
UDDI	<i>Universal Description, Discovery and Integration</i>
UML	<i>Universal Modelling Language</i>
WAS	<i>Windows Activation Service</i>
WCF	<i>Windows Communication Foundation</i>
WSDL	<i>Web Services Definition Language</i>
XML	<i>Extensible Markup Language</i>
XSD	<i>XML Schema Definition</i>
XSL	<i>XML Schema Language</i>

Lista de Tabelas

3.1	Linguagens de programação aceitas pelo <i>.NET Framework</i> (CSharp-Online.NET,)	20
3.2	Comparação entre tecnologias (IWANITZ; LANGE, 2002)	32
5.1	Parâmetros de um CSTR exotérmico	64
6.1	Resultados obtidos com a configuração 1 do simulador de RBS para remoção de poluentes	73
6.2	Resultados obtidos com a configuração 1 do simulador de RBS para remoção de poluentes utilizando o controlador PID aplicado remotamente com a ferra- menta desenvolvida	73

Lista de Figuras

2.1	Representação de uma arquitetura geral de laboratório virtual com acesso remoto	15
2.2	Representação de uma arquitetura geral de laboratório virtual baseado em serviço remoto	15
3.1	Caminho simples para a mensagem SOAP	22
3.2	Estrutura de uma mensagem SOAP	22
3.3	Estrutura XML de uma mensagem SOAP (DEITEL et al., 2003)	24
3.4	Diagramas de uma requisição do cliente a um <i>Web Service</i>	25
3.5	Pilha de protocolos básicos para Web Services	25
3.6	Metas do WCF	26
3.7	Unificação de Tecnologias Distribuídas	27
3.8	Arquitetura WCF	29
4.1	Diagrama esquemático do sistema proposto.	37
4.2	Representação da tela de configuração de um novo projeto para serviços OPC no Visual Studio 2008 com o <i>plug-in</i> DANSrv OPC DA .NET.	39
4.3	Representação da tela do <i>software</i> OPCDA.NET Test Client para gerencia dos serviços OPC.	40
4.4	Representação da tela de configuração de um novo projeto no Visual Studio 2008 para um serviço WCF.	41
4.5	Diagrama de classes do serviço WCF proposto.	42
4.6	Diagrama de blocos da comunicação entre os serviços, banco de dados e processos.	42

4.7	Modelo E-R para o sistema proposto.	44
4.8	Diagrama do banco de dados do sistema.	45
4.9	Fluxograma de ações no programa.	46
4.10	Representação da tela de <i>login</i> do Laboratório Virtual.	47
4.11	Representação da tela de cadastro de usuário do Laboratório Virtual.	47
4.12	Representação da tela principal do programa.	48
4.13	Representação da tela de visualização dos dados recebidos do processo remoto.	49
4.14	Representação da tela do componente para implementação do Controle PID.	50
4.15	Representação da tela com visualização dos dados recebidos em tabelas.	50
4.16	Representação da tela de exportação de dados noutros formatos de arquivos.	51
4.17	Componente para inserção de um novo processo.	52
5.1	Representação de um reator de mistura perfeita	56
5.2	Representação da tela de escolha de configuração do simulador de RBS	59
5.3	Representação da tela principal do simulador de RBS	59
5.4	Diagrama de componentes do simulador	60
5.5	Representação ideal de CSTR	62
5.6	Representação do funcionamento do sistema peltier	64
5.7	Foto do processo com Módulo Peltier	67
6.1	Representação da tela da ferramenta com os dados recebidos do simulador de CSTR (temperatura [K]).	70
6.2	Representação da tela da ferramenta com os dados recebidos do simulador de CSTR (concentração do componente no interior do tanque [$kmol/m^3$]).	71
6.3	Resultados obtidos a partir da simulação de CSTR com o MatLab (a) concentração (b) temperatura	72
6.4	Representação da tela da ferramenta com os dados recebidos do simulador RBS com a estratégia de controle implementada.	74

6.5	Representação da tela da ferramenta com os dados recebidos do simulador RBS sem estratégia de controle.	75
6.6	Representação da tela do simulador RBS com a estratégia de controle implementada.	75
6.7	Representação da tela do simulador RBS sem estratégia de controle. . .	76
6.8	Resultados obtidos com o Módulo Peltier	77

Capítulo 1

Introdução

Os recursos de tecnologia atuais permitem uma maior flexibilidade na união de aspectos considerados fundamentais na educação profissional de um cidadão, a aproximação entre a teoria e prática. Os laboratórios universitários e dos centros de pesquisas são os locais onde os alunos e pesquisadores tem os primeiros contatos com a parte prática das disciplinas.

Quando os engenheiros e demais profissionais, ao final de seus cursos de graduação e/ou profissionalizantes, partem para estágios ou iniciam suas vidas profissionais em empresas e na indústria, é normal escutarem de profissionais mais experientes que “na teoria tudo funciona, porém na prática a coisa é diferente”. Na prática é onde todo o conhecimento teórico adquirido em livros e na sala de aula é aplicado e os experimentos em laboratórios ajudam a entender um pouco desta diferença. Por parte do aluno que escolhe uma determinada área de interesse, existe uma motivação muito grande quando parte para ver os fenômenos físicos, químicos, biológicos, etc., acontecendo.

Com a disseminação da Internet e da utilização do computador como meio educacional, foi permitida uma forma de educação massificada, já que há muita informação disponível e de fácil acesso. Para alunos de cursos de graduação, o computador tornou-se, nos últimos anos, uma ferramenta importante na sua formação profissional. São utilizados na elaboração de trabalhos, coleta e armazenamento de informações importantes das disciplinas do curso e em muitas outras situações.

Para certos tipos de experimentos no ramo da engenharia, torna-se inviável disponi-

bilizar tempo e dinheiro para a montagem de laboratórios com processos que atinjam um nível industrial. Há associado um custo muito elevado de equipamentos como Controladores Lógicos Programáveis (CLPs), bombas, sensores, atuadores, além de vários outros equipamentos e dispositivos que possibilitem a construção de plantas e módulos experimentais que se aproximem de um ambiente industrial real.

Os laboratórios virtuais são ferramentas que permitem a simulação ou emulação de sistemas por meio do processamento de dados, sons e imagens em ambientes locais ou pela *Web*. Promovem a integração de seus recursos computacionais e laboratoriais pela interoperabilidade de dados e aplicativos (MENEGHEL, 2003).

Existem hoje várias ferramentas de laboratórios virtuais. Vários sites na Internet que promovem experimentos virtuais são expostos no *cyber* espaço que, com o avanço desta tecnologia, viabilidade e facilidade utilização, tem se tornado, a cada dia, um dos meios mais importantes de comunicação. Além disso, as aplicações de laboratórios virtuais não estão dispostos somente na Internet, há ferramentas locais que possibilitam a criação de sistemas que simulem situações adversas tais como efeito de temperatura sobre determinado processo, processos químicos, construções civis, simuladores de falhas em plantas e processos industriais, análise e probabilidade de erros em projetos de engenharia, etc.

Os pesquisadores, alunos e interessados pelo assunto, podem se utilizar destes recursos para efetuar testes de situações que, feitos em ambientes reais, poderiam causar perdas de bens materiais (perda de produção, danos de máquinas e equipamentos, etc.) e até mesmo de vidas humanas. Tornar o ensino acessível significa permitir uma melhor capacitação profissional, principalmente para os que não estão próximos dos grandes centros de formação (MIGUEL, 2008).

1.1 Motivação

Os processos em escala de laboratório de centros de pesquisas e universidades são utilizados, normalmente, para testes, treinamento, desenvolvimento e demonstrações, auxiliando estudantes e pesquisadores na atuação e observação de vários fenômenos que podem acontecer no meio industrial. Para fins de estudos de controle, os processos labo-

ratoriais podem trazer ainda incentivos para o desenvolvimento de novas técnicas, bem como implementação, treinamento e aperfeiçoamento de técnicas existentes.

O surgimento de várias tecnologias e ambientes ergonômicos de programação, despertou o interesse no desenvolvimento de novas ferramentas que auxiliem na educação em todos os níveis. A idéia é fazer com que os alunos e pesquisadores se motivem ao ver o funcionamento e comportamento de sistemas, podendo atuar não somente em simuladores, mas em processos reais situados nos laboratórios, deixando o presente trabalho, uma solução que visa tornar o ensino mais gratificante possibilitando o acesso à informação.

1.2 Objetivos

Este trabalho teve como objetivo desenvolver uma arquitetura flexível baseada na utilização de serviços remotos para laboratório virtual de controle e automação.

O desenvolvimento de uma ferramenta que possibilite o acesso remoto a processos laboratoriais via Internet ou intranet, o monitoramento e atuação em tempo real, a distância, nos processos em escala de laboratório e simuladores executados também numa máquina remota, bem como um componente de controle que possibilite o usuário testar técnicas de sintonia PID, também é parte integrante do trabalho. A idéia é disponibilizar na ferramenta um componente de controle a parte para que haja a possibilidade de sua modificação em etapas futuras, existindo, assim, a possibilidade de estudos com várias técnicas controle.

Na fase atual, foram acompanhados dois simuladores (um para o Reator de Batedas Sequenciais para Sistemas Biológicos de Tratamento de Esgotos - RBS e um para o Sistema de Tanques com Agitação Contínua - CSTR) e um processo real (que utiliza um módulo com efeito Peltier) a fim de disponibilizar a ferramenta já com processos prontos para sua utilização. Os simuladores e processos citados também foram desenvolvidos para o presente trabalho (para o caso dos simuladores) ou modificados e aperfeiçoados (para o caso do processo).

A intenção é que a ferramenta de laboratório virtual obtenha dados já de alguns processos simulados ou executados remotamente, e que possibilite a agregação de outros

processos e simuladores no futuro.

Para haver uma padronização nos dados coletados foi utilizado o padrão de comunicação *OLE for Process Control* (OPC), que se tornou o padrão para se resolver problemas de incompatibilidade entre dispositivos e interfaces dos vários fornecedores de ferramentas (de *hardware* e *software*) no ramo industrial.

O *Windows Communication Foundation* (WCF) é a tecnologia utilizada para o serviço remoto que fará comunicação direta com o serviço OPC e disponibilizará todos os dados obtidos na rede (Internet ou intranet).

As principais características dessas tecnologias são descritas ao longo do documento. Também são colocadas vantagens e desvantagens e justificativas da sua utilização no presente trabalho.

1.3 Estrutura da dissertação

Neste primeiro capítulo, apresenta-se alguns aspectos sobre a importância dos experimentos em escala de laboratório para a formação de profissionais, além dos objetivos e motivação do trabalho.

No segundo capítulo, demonstra-se ferramentas de ensino a distância e *e-Learning*, que é o ensino a distância com auxílio da tecnologia. São colocadas algumas ferramentas semelhantes a que é proposta neste trabalho além das formas que os recursos de tecnologia vêm sendo utilizados nos dias atuais.

No terceiro capítulo, são apresentadas as tecnologias utilizadas na comunicação entre o servidor, clientes, dispositivos do laboratório virtual e simuladores, bem como um histórico da evolução dessas tecnologias, vantagens e desvantagens com relação às outras existentes no mercado e as justificativas de utilização neste projeto.

No quarto capítulo, discute-se a metodologia de desenvolvimento da ferramenta de Laboratório Virtual baseada em serviços. É realizado o levantamento dos requisitos do sistema, demonstrada sua arquitetura, diagramas de classes e demais diagramas necessários para documentação, bem como a implementação do seu banco de dados de descrição das telas da aplicação.

No quinto capítulo, faz-se uma descrição dos processos e simuladores que foram

utilizados e implementados (três disponibilizados). Discute-se ainda as questões de modelagem dos simuladores, expondo como e quais tecnologias foram utilizadas para desenvolvimento.

Apresenta-se no capítulo seis, um estudo de caso demonstrando a utilização da ferramenta para cada um dos três processos disponibilizados. Realiza-se ainda uma análise e comparação dos resultados com outras ferramentas e, em um dos casos, comparados os resultados com implementação de uma técnica de controle PID.

As considerações finais e propostas de trabalhos futuros são apresentadas no capítulo sete.

Capítulo 2

E-Learning e Ambientes Virtuais de Aprendizado

Os Laboratórios Virtuais são ambientes que vêm sendo bastante difundidos no meio acadêmico (como ferramentas de fácil manuseio, baixo custo para alguns casos, e de grande utilidade no auxílio de trabalhos experimentais) e industrial (por projetistas e no treinamento de operadores). Surgiram com a necessidade de diversificar a criação de material didático, em especial, com o uso das TICs (Tecnologia da Informação e Comunicação) e *software* especiais, proporcionando o acesso a experiências que facilitem a consolidação de conceitos (MIGUEL, 2008).

O termo empregado para referenciar a educação e treinamento à distância com o auxílio da tecnologia é o *e-Learning* e vem sendo utilizado não somente nas salas de aula, mas é uma exigência das grandes companhias que visam o aperfeiçoamento e reciclagem dos profissionais. Nestas ferramentas, existe a possibilidade do desenvolvimento da aptidão para realização de alguma tarefa específica ou procedimentos que leve a uma situação próxima da real, fazendo com que o usuário (seja ele pesquisador, operador, projetista, etc):

1. tenha consciência da sua tomada de decisão;
2. amplie a visão de trabalho em grupo e, até mesmo,
3. desenvolva soluções de baixo custo, baseadas nos resultados das simulações e

experimentos.

Neste capítulo, são apresentadas as características de *e-Learning* e discutidos os impactos que sua implantação vem causando na sociedade. É feita ainda uma análise das principais ferramentas existentes que os laboratórios de engenharia elétrica e companhias utilizam para treinamento, bem como o que há em pesquisa no meio acadêmico. Vantagens e desvantagens deste sistema de aprendizagem são colocados bem como uma comparação do que é atualmente pesquisado e desenvolvido nesta área com a ferramenta desenvolvida no presente trabalho ainda é realizada no final do capítulo.

2.1 *e-Learning*

O *e-Learning* engloba as formas de apoio ao ensino e aprendizagem por meio eletrônico, de natureza processual, com o objetivo de efetuar a construção do conhecimento com referência à experiência individual, da prática e do conhecimento do aluno. Sistemas de informação e comunicação, na internet ou não, são meios específicos para implementação do processo de aprendizagem (TAVANGARIAN et al., 2004).

Para as empresas o *e-Learning* é um aliado. Os programas de treinamento de funcionários que utilizam essa ferramenta atingem maior público e tem custos menores. Um treinamento à distância costuma ser 40 a 60% mais barato que o presencial (*e-Learning Brasil*, 2005).

A Comunicação Mediada por Computador (CMC) se refere ao uso de redes de computadores para facilitar a interação entre alunos geograficamente separados (JONASSEN et al., 1995). Com a evolução da Internet, pesquisas voltadas para uma utilização deste recurso para “reduzir o espaço geográfico” entre professores, estudantes e pesquisadores, vêm sendo realizadas. Várias formas de “encurtamento” desta distância estão disponíveis e são largamente utilizadas nos dias atuais, são as chamadas ferramentas de CMC. Com os avanços destas ferramentas, torna-se fácil ministrar uma aula para vários interessados pelo mundo por meio de vídeo e áudio conferências, onde tudo que se passa é transmitido em tempo real através da rede mundial de computadores, bate-papos e fóruns de discussão. Mais fácil ainda, pode-se verificar aulas gravadas por meio de páginas na *Web* que as disponibilizem para todos que tenham interesse.

O processo de aprendizado com *e-Learning* envolve um conjunto de conteúdos disponibilizados no computador, permitindo que o conhecimento esteja disponível a qualquer hora e a qualquer lugar. Sua difusão aumentou a possibilidade do conhecimento e acesso à informação e democratizou a educação para todas as camadas da população.

Os laboratórios virtuais vêm para estender ainda mais os conceitos de *e-Learning*. São recursos computacionais oferecidos em versões na Internet ou para instalação local, com o intuito de oferecer os diversos recursos laboratoriais, de determinadas áreas específicas, a que se destinam. Com eles, pode-se realizar treinamentos, experimentos, simulações e estudos diversos, possibilitando uma difusão da informação e o compartilhamento do conhecimento entre os seus utilizadores;

Existem várias vantagens e desvantagens associadas aos laboratórios virtuais. Dentre elas, podem-se destacar aquelas listadas a seguir:

Vantagens:

- Não há custo associado à compra de equipamentos, porém pode haver custo associado à implementação das aplicações;
- Diversificação na oferta de informação, onde vários conteúdos podem ser abordados utilizando uma mesma ferramenta;
- Redução de tempo na execução de experimentos quando utilizados para esses fins, já que num laboratório convencional pode haver perda tempo na identificação de problemas ocasionados por equipamentos antigos ou danificados;
- Possibilidade de treinamentos e simulações na indústria para verificação do comportamento dos sistemas envolvidos, em diversas situações que, nas quais a ocorrência em sistemas reais, poderia ocasionar prejuízos materiais e humanos;
- Atualização rápida, distribuição e disponibilidade permanente de conteúdos;
- Possibilidade de acesso em tempo real a sistemas reais, visto que o usuário pode utilizar aplicações remotas, havendo então uma mesclagem entre laboratórios virtuais e reais.

- Facilidade no registro das informações, quando há a possibilidade do armazenamento dos dados recebidos serem armazenados em banco de dados (locais ou remotos), além de prover o registro e acompanhamento detalhado da participação do utilizador com geração de estatísticas, relatórios, etc;
- Ambientes com recursos de visualização diversificados (tabelas, animações, imagens, textos, etc);
- Ritmo de aprendizagem e horários podem ser definidos pelo utilizador.

Desvantagens:

- Custo de desenvolvimento pode ser elevado;
- Necessidade de um esforço elevado para motivar e obter a atenção de alunos e aprendizes, visto que na Internet existem diversas formas de desviar a atenção dos mesmos;
- Prejuízo no desenvolvimento da socialização do aluno, já que não há contato físico com professores e demais alunos;
- Exigência de uma maior disciplina e organização nos horários;
- Quebra na comunicação física com o serviço pode ocasionar na interrupção de utilização da ferramenta, ocasionando atrasos de cursos e treinamentos;
- Dependência de velocidade de conexão com a Internet;
- Não permite o conhecimento/contato com equipamentos reais, podendo ocasionar no empobrecimento da parte prática dos estudos;
- Problemas com incompatibilidade com sistemas operacionais e/ou com programas tidos como pré-requisitos;
- Atualização constante dos componentes tecnológicos;

2.2 Laboratórios Virtuais de Controle e Automação

Atualmente, para a formação de um engenheiro, é necessário, durante o curso de graduação, o contato com ferramentas que fazem o papel de laboratórios no computador. A utilização de aplicações como o *MatLAB* e *Simulink* (MATLAB/SIMULINK,), *Mathematica* (MATHEMATICA,), *CircuitMaker*, entre outros, possibilitam a observação das teorias de fenômenos que são estudados nas disciplinas da graduação. Para um engenheiro de Controle e Automação, o *MatLAB* e *Simulink*, possibilitam a implementação e simulação de diversas técnicas de controle para processos definidos pelo usuário por meio de diagramas de blocos (para o *Simulink* que já contém blocos que desempenham funções específicas, tais como implementação de função de transferência, somador, osciloscópio, experimento com relé, etc) ou por meio de código específico do *MatLAB* para implementação dos modelos matemáticos que o usuário desenvolve. Essas ferramentas podem ser utilizadas para estudos locais e remotos, porém, estudos mais intensivos são requeridos não só por parte do desenvolvedor, mas também por quem vai utilizar as aplicações.

Outro caminho para disponibilização dos laboratórios virtuais é o navegador de Internet (*browsers*). Recursos das linguagens de programação, principalmente aplicações utilizando as linguagens de programação *Java* e *Javascript*, são utilizados para prover animações gráficas e realizar a lógica matemática. Diversas instituições disponibilizam este tipo de ferramenta, porém, grande parte destas, são simuladores locais que apenas demonstram situações ideais de determinados processos. Há disponível na Internet em *sites* de Universidades e Laboratórios de pesquisa, uma infinidade de laboratórios virtuais para realização de experimentos de física (experimento de pêndulo, eletricidade, magnetismo, etc.).

Universidades, centros de pesquisas e indústrias de todos os portes, desenvolvem projetos específicos para atender suas necessidades de estudos e treinamento de pessoal para áreas específicas. No Brasil, temos como exemplo empresas como a Petrobras que desenvolve pesquisas em parceria com universidades com o intuito de ter como produto ferramentas para treinamento dos operadores e projetistas.

No mundo, há uma busca constante não só das grandes empresas, mas de instituições de ensino que buscam o aperfeiçoamento do ensino de controle e automação.

Conferências de *e-Learning* organizadas por associações como *Institute of Electrical and Electronics Engineers* (IEEE) são a oportunidade para estudantes e pesquisadores demonstrarem seus trabalhos e compartilharem informações com o mundo. Como é o caso dos trabalhos publicados em (DORMIDO et al., 2007), sobre o laboratório virtual com finalidade de apoiar estudos de controle e automação com atuação em tempo real em um sistema de três tanques; e (KO et al., 2001), na qual, com a ferramenta, é possível também a atuação e utilização de técnicas de controle em um sistema de tanques acoplados, utilizando também recursos de áudio e vídeo.

O Laboratório de Instrumentação Eletrônica e Controle (LIEC) do Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande (UFCG), onde o presente trabalho foi realizado, é responsável pelo desenvolvimento da ferramenta *BRSIL* que avalia a probabilidade de falha dos Sistemas Instrumentados de Segurança (SIS) para o ambiente industrial da empresa Petrobras e realiza testes de configurações de projeto identificando o comportamento e a existência de estados indesejáveis. O ambiente virtual da ferramenta deve ainda auxiliar o desenvolvimento de configurações seguras que atendam às especificações de segurança (BARROSO, 2009). O LIEC é responsável ainda por vários outros projetos de ambientes virtuais de aprendizado tais como projetos com *Radio-Frequency IDentification* (RFiD), Sintonia de Controladores PID Industriais, Avaliação da Confiabilidade em Malhas de Controle, entre outros.

2.3 Análise comparativa

Os processos com tanques são os que encontram mais ferramentas desenvolvidas para estudo e treinamento. Na seção anterior, foram citadas referências a processos deste tipo, devido ao seu processo de desenvolvimento e finalidade serem semelhantes ao que foi desenvolvido no presente trabalho, porém, tais laboratórios são bem específicos para determinados tipos de estudo (controle automação em reatores, como citado) conforme arquitetura genérica representada na Figura 2.1.

A diferença entre os trabalhos citados e o apresentado aqui está na arquitetura genérica do laboratório baseado em serviços *Web*, visto que outros processos, tais como de temperatura, da área de robótica, projetos de unidades produtivas, entre outros; bem

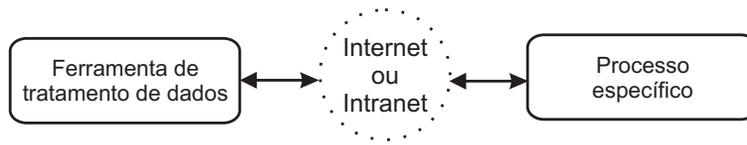


Figura 2.1: Representação de uma arquitetura geral de laboratório virtual com acesso remoto

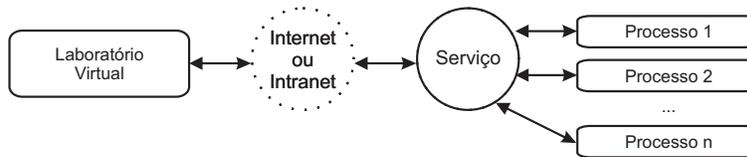


Figura 2.2: Representação de uma arquitetura geral de laboratório virtual baseado em serviço remoto

como diversos tipos de simuladores de trocador e condução de calor, sistemas com respostas inversas, sistemas oscilatórios, de primeira e segunda ordem; além de módulos para implementação de estratégias de controle; podem ser agregados ao laboratório virtual, proporcionando um ambiente com diversidades de recursos educacionais e flexibilidade de estudo. O diagrama de blocos representado na Figura 2.2 demonstra a intenção de verificar a funcionalidade da ferramenta de laboratório virtual para vários processos baseando-se em serviços.

2.4 Conclusão

Foram apresentadas neste capítulo algumas das principais vantagens e desvantagens do sistema de educação por meio do computador. Pôde-se observar que é uma tendência das grandes companhias a adoção do sistema já que há um baixo custo associado para o treinamento de uma quantidade muito grande de pessoal, se comparado ao treinamento presencial.

Pensando neste crescimento do *e-Learning*, há um empenho de universidades no Brasil e no mundo que já desenvolvem produtos visando o “encurtamento” da distância entre o aluno e a fonte da informação. Alunos e pesquisadores pelo mundo já desenvolvem soluções que visam o aperfeiçoamento do estudo, buscando uma didática me-

lhor e aproximação da sala de aula com o ambiente industrial real. A Internet é um meio de comunicação que pode ser cada vez mais “descoberta” e explorada para este fim.

As ferramentas de laboratórios virtuais permitem o compartilhamento de recursos em diferentes localidades geográficas, além de permitir o acesso à uma quantidade de material pedagógico bastante elevada, permitindo um aumento da produtividade, diminuição no tempo de aprendizado e redução de custos que as situações presenciais impõem. É óbvio que os ambientes virtuais não vêm para substituir os ambientes reais, mas sim incrementá-los e possibilitar ao aprendiz, simular e analisar situações adversas de um processo ou experimento.

Capítulo 3

Tecnologias de Comunicação Utilizadas

A ferramenta de Laboratório Virtual proposta neste trabalho necessita de recursos voltados para Internet e intranet. Neste capítulo, são apresentadas todas as tecnologias que foram utilizadas para comunicação entre computadores e dispositivos de aquisição de dados. Inicialmente, um breve histórico a respeito do surgimento dos principais protocolos de comunicação é apresentado, visando o entendimento aprofundado dos conceitos que envolvem a tecnologia de comunicação utilizada neste trabalho, o *Windows Communication Foundation* (WCF) e *OLE for Process Control* (OPC). Comparações entre tecnologias são apresentadas ao longo do texto e, por fim, são colocados pontos importantes justificando a escolha destas tecnologias.

3.1 *Windows Communication Foundation - WCF*

A tecnologia de comunicação baseada em serviço mudou a forma de desenvolvimento de *software*. Aplicações que interagem com serviços vem se tornando uma norma para o desenvolvedores de *Windows* e essa mudança foi possível com o *Windows Communication Foundation* (WCF). Lançado em 2006, passou pela versão do *.NET Framework* 3.5 e, atualmente, a versão mais recente da tecnologia está inclusa no *.NET Framework* 4. WCF vem se tornando o fundamento certo para quem procura o desenvolvimento de aplicações na plataforma *.NET* (CHAPPELL, 2009).

Antes de apresentar a tecnologia WCF, uma pesquisa breve sobre a história dos

principais protocolos de comunicação já lançados foi realizada e apresentadas a seguir.

3.1.1 Primeiros mecanismos e protocolos de comunicação entre computadores e demais dispositivos

Quando os primeiros computadores pessoais surgiram, os trabalhos das empresas eram feitos em aplicativos que, geralmente, eram constituídos por um editor de texto e se mostravam ao usuário na forma de planilhas eletrônicas. Além disso, o formato dos dados nessas planilhas eletrônicas tinha que ser específico e não era qualquer editor de texto que poderia abri-las. Ainda não havia uma forma de compartilhamento de arquivos e somente um usuário podia ter acesso à leitura ou escrita.

Com o passar dos anos, os computadores foram se tornando bem mais baratos e passaram a se consolidar cada vez mais como uma ferramenta de negócios. Começou a crescer, assim, a necessidade das grandes empresas de trocar informações quase em tempo real em diferentes locais do estabelecimento, o que motivou as grandes indústrias de *software* e *hardware* a desenvolverem soluções para redes de computadores, permitindo os vários departamentos de uma instituição interconectar seus computadores e compartilhar os arquivos e recursos dos mesmos.

As soluções em rede passaram a se expandir a partir daí. Outros computadores, impressoras, scanners e muitos outros dispositivos podem trocar informações e as aplicações tiveram de se adaptar a enviar e receber dados, além de coordenar e sincronizar as ações com outras aplicações em funcionamento ao mesmo tempo noutros computadores.

Padrões e mecanismos de comunicação foram então desenvolvidos para, além de garantir a integridade e segurança da informação, transmiti-las na forma mais rápida possível. Os *sockets* são um elo bidirecional de comunicação utilizado para enviar e receber dados a partir de dois programas que comunicam numa rede. Os *pipes* são um outro mecanismo de comunicação, no qual os processos podem se conectar quando necessitam realizar uma troca de informação com um outro processo. Estes dois meios de comunicação podem trazer alguns problemas na rede quando são acessados ao mesmo tempo. Um algoritmo para abrir e fechar a conexão a cada pedido de envio ou aquisição

de dados deve ser muito bem trabalhado para evitar travamento quando mais de um processo tenta ler o mesmo *socket* e coordenar o tráfego das informações.

Lançado em 1993, o *Component Object Model* (COM) é usado para permitir a comunicação entre os processos em um mesmo computador e criar objetos dinamicamente em qualquer linguagem de programação. Apesar de lançado em 1993, a Microsoft só começou a divulgá-la amplamente a partir de 1997 (BERGER,). Apesar de ser desenvolvida pela Microsoft, o COM pode ser utilizado em sistemas operacionais como as várias versões do Linux, já que o COM independe da linguagem de programação para implementar objetos para que estes possam ser utilizados em diversos ambientes, inclusive máquinas com diferentes arquiteturas. Dessa forma, foi dado início às tecnologias de reutilização de *software*, onde os desenvolvedores podem usar COM para criar componentes de *software* e interligar componentes para construir aplicações.

Com a necessidade de comunicação entre processos entre computadores diferentes, o *Distributed Component Object Model* (DCOM) foi criado (também pela Microsoft) para suprir esta necessidade. Após o DCOM foi criado o COM+ onde foi adicionado um gerenciamento de recursos automático de conexão (como o fechamento automático de uma conexão com um banco de dados ou de um arquivo quando o aplicativo deixa de utilizar o componente de conexão) e o chamado *garbage collection* (que foi inserido no DCOM e aperfeiçoado no COM+ garantindo de que as referências realizadas por clientes em aplicativos sejam liberadas quando, por exemplo, o processo do cliente caiu, ou a conexão da rede foi perdida). O COM+ também provê as operações assíncronas úteis nos processos de engenharia, já que um dispositivo numa rede pode demorar um pouco para responder, com isso, as aplicações podem continuar sendo executadas e o dispositivo solicitado enviará uma notificação quando a operação estiver concluída.

Seguindo a ordem cronológica, a Microsoft lançou o *.NET Framework* que visou padronizar como plataforma única de desenvolvimento e execução de sistemas e aplicações. Esta nova tecnologia incorporou muitas outras para a construção de componentes em rede. O *Remoting*, por exemplo, a permitiu um cliente acessar objetos remotos através de uma rede, como se ambos os objetos estivessem executando no mesmo local. Na Tabela 3.1 estão listadas as linguagens aceitas pela atual versão desta tecnologia (*.NET Framework 4*).

Abstract State Machine Language	Ada	APL
ASP	Basic	Beta
BF.NET	Boo	C
C#	C# variants	C++
Caml	COBOL	CULE
Eiffel	Forth	Fortran
G#	Haskell	IL - MSIL - Intermediate Language
Java	Javascript	Lexico
LISP	LOGO	Lua
Mercury	Mixal Assembly Language	Modula-2
Mondrian	Nemerle	Oberon
Pan	Perl	Pascal
Pascal variants	PHP	Processing
Prolog	Python	Python variants
Ruby	Scala	Scheme
Smalltalk	SML (Standard Meta Language)	Spec Explorer
TCL - TK	Visual series from Microsoft	xBase
Zonnon		

Tabela 3.1: Linguagens de programação aceitas pelo *.NET Framework* (CSharp-Online.NET,)

3.1.2 Arquitetura Orientada a Serviço - SOA (*Service-Oriented Architecture*)

A Arquitetura Orientada a Serviço (SOA) baseia-se na idéia de um conjunto de que aplicações pode ter suas funcionalidades implementadas e disponibilizadas por meio de serviços. O que quer dizer que estes serviços podem ser modificados independentemente da aplicação que esteja utilizando (a aplicação deverá se ajustar a ela, mas isto só será um problema se o tipo do dado recebido for modificado).

Um serviço, do ponto de vista de SOA, consiste em funcionalidades que um sistema deixa disponível para outro sistema que possui interface bem definida e deve funcionar independentemente de outros serviços rodando no sistema (no lado do servidor e cliente). Os detalhes são transparentes ao consumidor, o que quer dizer que a forma de comunicação funciona como uma caixa preta, os dados complexos são encapsulados e a informação que interessa é acessada através de mensagens, nas quais contratos definem as operações que os serviços podem oferecer.

Os conceitos de SOA se aproximam da programação orientada a componente, pois sua implementação compreende um ou mais conjuntos de aplicações que apresentam interoperabilidade, são francamente acoplados e funcionam, de uma forma geral, da seguinte forma:

1. O sistema cliente faz uma requisição dos serviços ao servidor;
2. O sistema servidor recebe essa requisição, processa as informações necessárias e manda uma resposta para o cliente;
3. O sistema cliente recebe a resposta processada pelo lado servidor e utiliza, da forma que lhe convém, os dados recebidos.

Com esse mecanismo, torna-se fácil o compartilhamento de informação via *Web* agilizando os processos de negócios e permitindo a sua utilização com diferentes linguagens e arquiteturas.

3.1.3 Protocolo de Acesso Simples a Objeto - SOAP (*Simple Object Access Protocol*)

SOAP é um protocolo que independe de plataforma, o qual utiliza o *Extensible Markup Language* (XML) para fazer requisições em HTTP. As respostas das requisições são empacotadas numa mensagem em XML que contém toda a informação considerada relevante para o processo. O protocolo HTTP foi escolhido para transmitir mensagens de SOAP por ser o padrão de envio de informação através da Internet.

As principais características deste protocolo são listadas abaixo:

- Extensível, pois suporta vários tipos de objetos desde os mais básicos (*int, double, strings, etc.*) aos mais complexos (*DataSets, DateTime, Arrays, etc.*);
- Independe de linguagem, plataforma ou modelo de programação;
- Contém mecanismos para lidar com erros;

Como já foi mencionado, a mensagem SOAP é um arquivo XML. O caminho que esta percorre é aquele que pode ser visualizado na Figura 3.1. A mensagem deve conter ainda as informações dispostas na Figura 3.2. Onde,

- **Envelope:** É o elemento no topo do documento XML que representa a mensagem que deve ser transmitida;

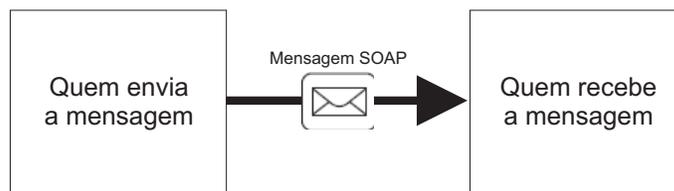


Figura 3.1: Caminho simples para a mensagem SOAP

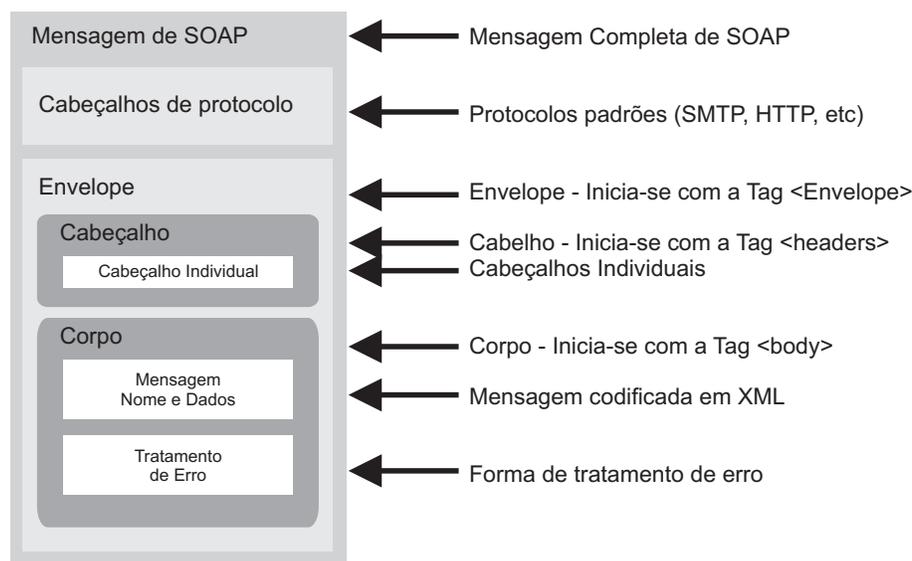


Figura 3.2: Estrutura de uma mensagem SOAP

- **Cabeçalho:** É um mecanismo genérico para adicionar, de forma descentralizada, recursos a uma mensagem de SOAP, onde não há um acordo prévio entre as partes comunicantes. Alguns atributos devem ser definidos para indicar quem deve lidar com este recurso e se é facultativa ou obrigatória;
- **Corpo:** É um recipiente para a informação obrigatória prevista para o destinatário final da mensagem.

As regras gramaticais da mensagem SOAP, de acordo com (BOX et al., 2008) são as seguintes:

1. Envelope

- O nome do elemento é "Envelope";

- O elemento deve estar presente em uma mensagem SOAP;
- O elemento pode conter declarações de *namespace*, bem como atributos adicionais. Se estiverem presentes, os atributos adicionais devem ter *namespace* qualificados. Da mesma forma, o elemento pode conter sub-elementos adicionais. Se apresentar estes elementos, devem ter *namespace* qualificados e seguir o elemento no corpo da mensagem.

2. Cabeçalho

- O nome do elemento é "Cabeçalho";
- O elemento pode estar presente na mensagem SOAP. Se estiver, este deve ser o primeiro elemento filho imediato ao elemento Envelope.
- O elemento pode conter um conjunto de entradas cabeçalho e ser um filho imediato de cada elemento de cabeçalho do SOAP. Todos os elementos filhos imediatos de um cabeçalho SOAP devem ter *namespace* qualificados.

3. Corpo

- O nome do elemento é "Corpo";
- O elemento deve estar presente na mensagem de SOAP e deve ser um elemento filho imediato de um elemento Envelope de SOAP. Ele deve seguir diretamente o elemento Cabeçalho, caso contrário, deve ser o primeiro elemento filho imediato do elemento Envelope;
- O elemento pode conter um conjunto de entradas do corpo e cada um ser um elemento filho imediato do elemento de Corpo SOAP. SOAP define *SOAP Fault*, que é usado para indicar mensagens de erro.

A Figura 3.3 representa a estrutura do arquivo XML de uma mensagem SOAP.

3.1.4 *Web Services*

Com o surgimento da Internet, surgiram também várias aplicações *Web* que são executadas, remotamente à máquina na qual o aplicativo está hospedado, através dos *browsers*.

```
1 POST /HugeIntegerWebService/HugeInteger.asmx HTTP/1.1
2 Host: localhost
3 Content-Type: text/xml; charset=utf-8
4 Content-Length: length
5 SOAPAction: "http://www.deitel.com/csphtml/ch21/Bigger"
6
7 <?xml version="1.0" encoding="utf-8"?>
8
9 <soap:Envelope
10   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
11   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
12   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
13
14   <soap:Body>
15     <Bigger xmlns="http://www.deitel.com/csphtml/ch21/">
16       <first>string</first>
17       <second>string</second>
18     </Bigger>
19   </soap:Body>
20
21 </soap:Envelope>
```

Figura 3.3: Estrutura XML de uma mensagem SOAP (DEITEL et al., 2003)

Os *Web Services* surgiram para flexibilizar, aos aplicativos locais, a utilização de funcionalidades que outros aplicativos já disponibilizam na Internet e prover, assim, o *software* distribuído. Funciona da mesma forma que um site da *Web*, onde uma aplicação armazenada numa máquina pode ser acessada por outra, através da Internet.

Os sistemas distribuídos mais modernos foram desenvolvidos utilizando os princípios de SOA, a aceitação global dos *Web Services* mudou a forma como os componentes de *software* são definidos e construídos.

Promovendo também a interoperabilidade, os *Web Services* podem ser incorporados aos aplicativos independentemente destes, bastando que o serviço se encontre ativo, dessa forma, diferentes aplicações podem ser criadas a qualquer momento e em qualquer lugar consumindo o mesmo *Web Service*.

A *Web Services Definition Language* (WSDL) é a linguagem de definição dos *Web Services* e apresenta interfaces públicas indicando ao usuário e cliente que pretende se utilizar daquele serviço como o mesmo deve ser consumido. Além disso, o *Universal Description, Discovery and Integration* (UDDI) é o protocolo universal responsável por pesquisar e publicar os *Web Services*, descrevendo bem os detalhes do serviço por

WSDL e se comunicando via SOAP. Já o *XML Schema Definition (XSD)* contém todas as definições e regras de validação do arquivo XML da mensagem, descrevendo a estrutura do documento. Por fim, uma outra característica importante é o WS-i, que significa integração entre *Web Service*, que garante que eles podem “conversar entre si”.

As funções em um *Web Service* são chamadas remotamente usando um procedimento de chamada remota, no qual o atributo [WebMethod] é colocado nas funções que estão disponíveis para utilização por outros usuários via uma conexão remota. A forma de acesso de um aplicativo cliente pode ser observada na Figura 3.4, na qual mostra uma comunicação simples entre as partes.

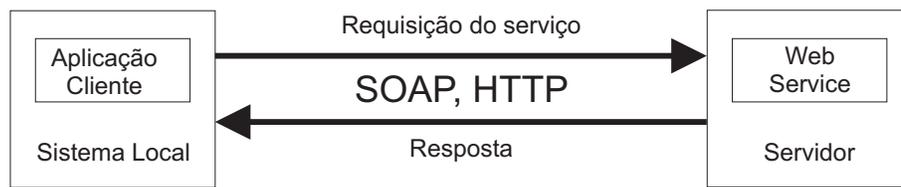


Figura 3.4: Diagramas de uma requisição do cliente a um *Web Service*

Em resumo, na Figura 3.5 é apresentada a pilha de protocolos básicos para *Web Services*. A interação com o serviço vai seguir normalmente a ordem de cima para baixo da pilha invocando as funções do serviço.

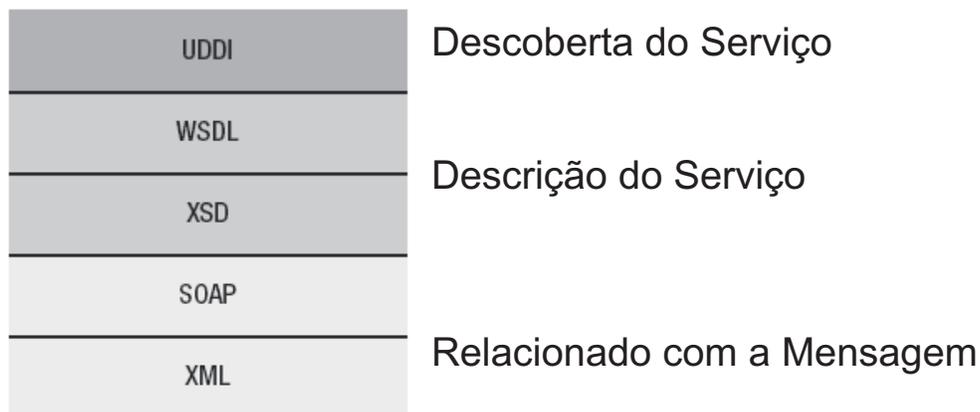


Figura 3.5: Pilha de protocolos básicos para *Web Services*

3.1.5 *Windows Communication Foundation (WCF)*

As principais tecnologias citadas são integradas em um único framework. O WCF visa a promoção de maior segurança e confiabilidade no tráfego de informação além de facilitar o desenvolvimento de aplicações distribuídas. Para (PEIRIS; MULDER, 2007), WCF veio para atingir as metas ilustradas na Figura 3.6, ou seja:

- Unificação das tecnologias existentes;
- Interoperabilidade entre plataformas;
- Desenvolvimento visando o SOA.

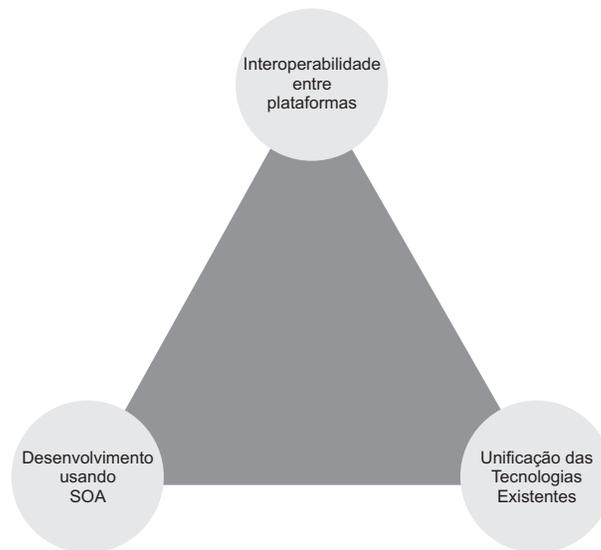


Figura 3.6: Metas do WCF

Unificação das tecnologias existentes

As várias tecnologias existentes de *software* distribuído representam um problema para os desenvolvedores quando estes têm que instalar suas soluções com as várias tecnologias de comunicação entre dispositivos existentes. Então, surge um dilema: “A tecnologia **T1** é compatível com esta **T2**? Com **T2** vou poder me conectar a um computador ou dispositivo com tecnologia **T3**? ”.

Um resumo das principais tecnologias de *software* que WCF engloba e como são melhor empregadas nas aplicações é apresentado na Figura 3.7.

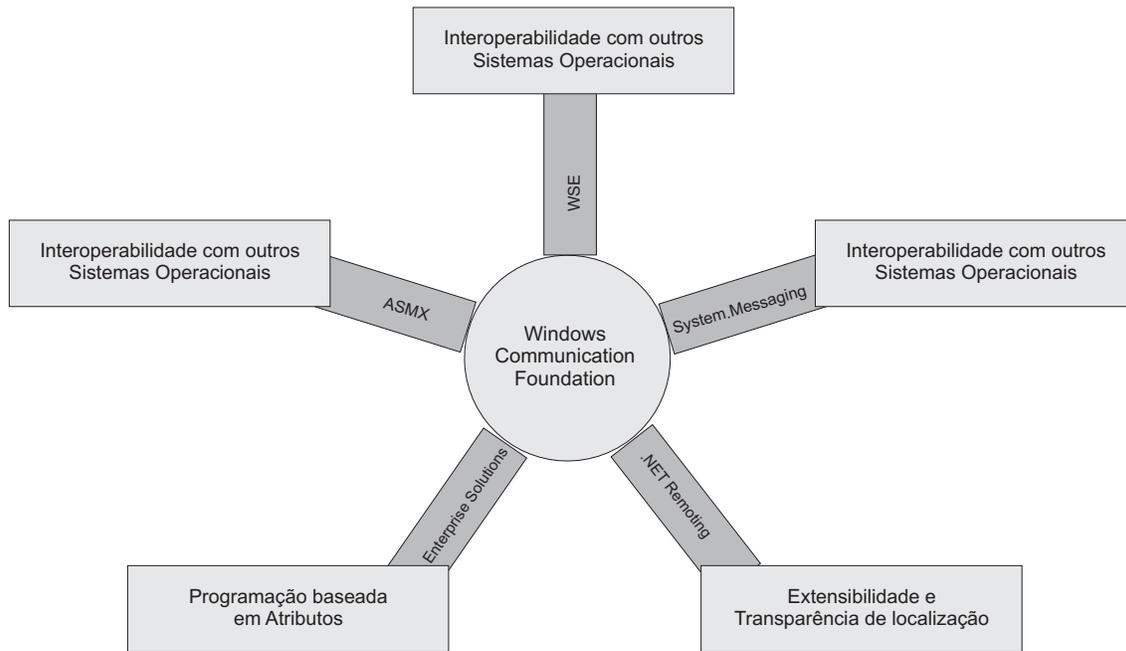


Figura 3.7: Unificação de Tecnologias Distribuídas

Interoperabilidade entre plataformas

Os mais diversos sistemas operacionais desenvolvidos pelas grandes empresas de *software* têm características próprias, o que traz o problema de nem todas as aplicações terem compatibilidade quando são executadas em plataformas diferentes. O protocolo nativo de mensagens WCF é o SOAP, o que provê a oportunidade aos serviços WCF de se comunicar com as demais plataformas que não sejam da família *Windows*, já que é um protocolo padrão baseado em XML, como já foi mencionado.

Desenvolvimento visando o SOA

Os serviços WCF baseiam-se nos princípios de SOA. O compartilhamento de classes e objetos na prática de programação orientada a objeto, são utilizadas também no WCF. Porém, ao invés de objetos, são utilizados esquemas, que são compartilhados utilizando

o *XML Schema Language* (XSL) e os contratos são definidos com WSDL. Dessa forma, como o desenvolvedor pode apenas se utilizar dos serviços WCF, não há a necessidade de conhecer detalhadamente as especificidades de cada tecnologia de comunicação, mas, mesmo assim, têm acesso a todos os recursos que estas tecnologias anteriores proviam. Não é necessário preocupação com formas de acesso e protocolos, bastando realizar algumas configurações no serviço.

3.1.6 Arquitetura WCF

Na Figura 3.8 há a representação gráfica das camadas que compõem a arquitetura WCF. As descrições e funcionalidades de cada camada são detalhadas a seguir (MSDN,).

Contratos e Descrições

Define os vários aspectos do sistema de mensagens. **Contrato de Dados** descreve todos os parâmetros que compõem todas as mensagens que um serviço pode criar ou consumir. Todos esses parâmetros são definidos por um documento XSD autorizando algum sistema que entende XML a processar a mensagem. A **Contrato da Mensagem** define a mensagem específica utilizando o protocolo SOAP e permite o controle detalhado das partes que compõem a mensagem. O **Contrato do Serviço** o atual método de assinatura do serviço e é distribuído como uma interface das linguagens de programação suportadas. **Política de Vínculos** estipula as condições necessárias para se comunicar com o serviço, inclui os requisitos de segurança e outros aspectos necessários para realização da comunicação com o serviço.

Tempo de Execução do Serviço

A camada de tempo de execução do serviço contém todos os comportamentos que devem ocorrer quando o serviço estiver sendo executado. Controles de otimização para o melhor processamento da mensagem, bem como o seu comportamento quando um erro e todos os blocos contidos nesta camada ocorre. Além disso, a forma como a mensagem é transmitida para o cliente, visando sempre a segurança da informação. A extensibilidade permite ainda a personalização da mensagem, pois é possível inspecionar a mensagem em tempo de execução e os aspectos de filtragem podem ocorrer a partir de ações predefinidas com base em filtros agindo em cabeçalhos das mensagens.

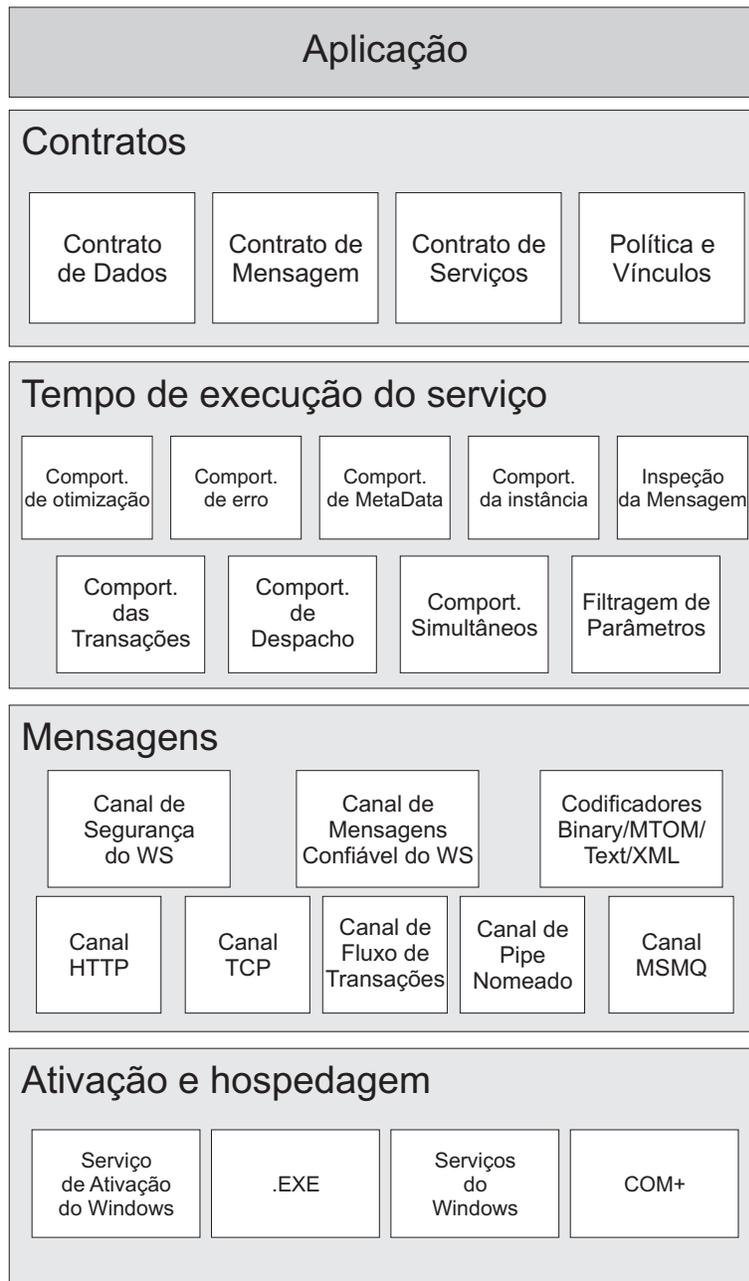


Figura 3.8: Arquitetura WCF

Mensagens

Esta camada é composta por canais que processam as mensagens. A camada de mensagens ilustra os possíveis formatos e padrões de troca de dados. Nela, é imple-

mentado o protocolo de segurança *WS-Security* para obter segurança na mensagem. O *WS-Reliable* também é implementado, permitindo que haja garantia que o usuário de destino recebeu o pacote. Os codificadores apresentam uma variedade de codificações que visam atender as necessidades das mensagens. Dessa forma, o canal HTTP especifica que o protocolo *Hypertext Transport Protocol* é o protocolo utilizado para entrega de mensagens. Já o canal TCP, da mesma forma, especifica o protocolo TCP. O canal de transações regula a forma de transação e os padrões da mensagem. O *pipe* nomeado permite a comunicação entre processos e o canal MSMQ a interoperação com aplicações MSMQ.

Ativação e Hospedagem

Nesta camada, o serviço é tratado como um programa. Os serviços são executados por meio de arquivos .EXE e podem ser hospedados ou executados em um arquivo executável gerenciado por um agente externo, como o *Internet Information Service (IIS)*, *Windows Activation Service (WAS)*. Um serviço pode ser executado manualmente como arquivos executáveis ou como um serviço do *Windows*.

3.2 OLE for Process Control - OPC

OLE é a abreviação de *Object Linking and Embedding*, criado em meados da década de 90 para descrever os sistemas orientados a objeto da Microsoft. OPC surgiu nesta época para especificar a comunicação entre dispositivos de controle de diferentes fabricantes. Nos dias de hoje, o OPC é utilizado para realizar uma comunicação eficiente entre os dispositivos de campo no ramo industrial e o hardware de controle.

Baseada noutra tecnologia da Microsoft, a DCOM, OPC implementa todas as definições da tecnologia, desde interfaces, métodos e objetos.

Vantagens e Comparações

A seguir, são apresentadas as principais vantagens do uso do OPC. Todos os envolvidos na cadeia do produto são considerados, desde o desenvolvimento até a aplicação da tecnologia OPC (IWANITZ; LANGE, 2002).

Para o fabricante de hardware

O produto pode ser utilizado por todos os sistemas no mercado, compatíveis com OPC e não é limitado a um sistema individual no qual uma solução correspondente (com drivers específicos) deve ser desenvolvida. Devido à existência da padronização de interfaces e da interoperabilidade relatada por eles, há somente a necessidade da familiarização com os requerimentos específicos dos outros sistemas. O tempo de colocação do produto no mercado para geração de novos dispositivos é reduzido e somente um servidor de OPC é atualizado. O tempo de suporte também é reduzido visto que somente um servidor terá de ser assistido.

Para o fabricante de *software*

O produto pode ser usado com todos os dispositivos e protocolos de comunicação que tem a interface OPC habilitada. O fabricante não precisa mais desenvolver soluções específicas. Devido à existência da padronização de interfaces e da interoperabilidade relatada por eles, há somente a necessidade da familiarização das especificações dos outros dispositivos e protocolos de comunicação. Haverá uma redução considerável na prestação de suporte ao produto.

Para quem integra sistemas

A flexibilidade na escolha de produtos é aumentada consideravelmente, fazendo com que aumente o número de seus projetos. O tempo de integração e treinamento é reduzido consideravelmente, já que OPC padroniza a interface para todos os produtos.

Para o usuário final

OPC provê uma flexibilidade adicional (distribuição de componentes, uso de novas tecnologias, escolha entre produtos, etc) na qual, durante o projeto de um sistema, produtos de vários fabricantes podem ser combinados. O aumento na variedade de produtos disponíveis no mercado, deverá, no futuro, conduzir a uma redução sensível dos custos, com o aumento da qualidade e conforto do usuário.

A OPC Foundation é dedicada a garantir a interoperabilidade na automação, pela criação e manutenção de especificações abertas que padronizam a comunicação de aquisição de dados dos processos, alarmes e registros de eventos, dados históricos, dados e sistemas corporativos e entre os dispositivos de produção. Dispositivos de produção incluem sensores, instrumentos, CLPs, RTUs, DCSs, IHMs, historiadores, os subsistemas de alarme, e mais utilizado na indústria de processos, manufatura e na aquisição e transporte de petróleo, gás e minerais (The OPC Foundation, 2010). Criada em 1996 como uma organização sem fins lucrativos, a OPC Foundation tem o objetivo de desenvolver e apoiar o padrão OPC. As especificações OPC podem ser encontradas no apêndice A.

	<i>Drivers</i> específicos de um produto	<i>Dynamic Data Exchange</i> (DDE)	OPC
Padronização	Não	Não	Sim
Interoperabilidade	Não	Limitado para níveis avançados	Sim
Desempenho	Muito rápido pois requer aplicações específicas	Lenta	Muito rápido
Acesso remoto	Não	Sim (NetDDE)	Sim (COM Distribuído)
Interações	Normalmente não disponível	Não	Sim. Vários métodos podem ser usados para troca de informação entre servidor e cliente
Complexidade	Muitos fabricantes disponibilizam ferramentas que drivers podem ser criados rapidamente para sistemas específicos. Um novo sistema requer tempo adicional de aprendizado.	Ferramentas são disponibilizadas pelos fabricantes. Desenvolvimento de DDE sem essas ferramentas é de uma dificuldade avançada	Requer aprendizado de programar em DCOM. Existem diversas ferramentas de ajuda para minimizar o tempo necessário para esse aprendizado.
Participação no mercado	Reduzido com o aumento de velocidade	Servidores DDE deverão desaparecer no futuro	Soluções OPC devem dominar o mercado
Disponível noutras plataformas	Poucos estão disponíveis	Disponível apenas para o sistema operacional Windows	Disponível para os sistemas que possuem DCOM disponível. O novo OPC e especificação XML deve possibilitar componentes que vão operar em sistemas sem o DCOM.

Tabela 3.2: Comparação entre tecnologias (IWANITZ; LANGE, 2002)

Na tabela 3.2 são colocadas as avaliações e comparações entre tecnologias com o mesmo objetivo de comunicação entre dispositivos.

A tabela 3.2 compara OPC com o DDE, que é outro mecanismo que descreve a interação entre dados de aplicações Windows, e os drivers específicos de um produto

levando em consideração os seguintes critérios:

- **Padronização:** Possuem definições aprovadas internacionalmente para utilização em ambiente de automação industrial?
- **Interoperabilidade:** Soluções de fabricantes diferentes podem trabalhar umas com as outras?
- **Desempenho:** Eficiência no que diz respeito ao tempo necessário para transferência de dados.
- **Acesso remoto:** É possível criar sistemas distribuídos?
- **Interações:** Suas funcionalidades vão além da troca de dados do processo? Trata-se, por exemplo, consulta de informações estruturais em tempo de execução.
- **Complexidade:** Qual o mínimo de conhecimento necessário para criar o produto correspondente?
- **Participação no mercado:** Qual a participação atual no mercado e previsão para o futuro?
- **Disponível noutras plataformas:** É possível utilizar a tecnologia em plataformas não-PC e sistemas operacionais não-Microsoft?

3.3 Conclusão

A questão de incompatibilidade das interfaces de comunicação entre equipamentos e dispositivos de fabricantes diferentes é um problema a ser superado no ambiente de automação industrial. Para o sistema de produção e ambiente de negócios trabalharem juntos, esta interface deve ser bem definida e confiável. Neste capítulo, foram feitas comparações entre as tecnologias de padronização na comunicação entre equipamentos de fabricantes diferentes. OPC vem se tornando a tecnologia mais utilizada no ambiente industrial pois proporciona a interoperabilidade entre equipamentos diferentes fabricantes, fornecendo uma padronização na troca de informação entre dispositivos em

tempo real. O *Ice middleware* e *Java Remote Method Invocation* (RMI) são exemplos de tecnologias bastante utilizadas, bem como o WCF, nos dias atuais, que proporcionam acesso remoto e, conseqüentemente, fornecem a possibilidade de desenvolvimento de aplicações distribuídas. Porém, a vantagem do WCF é ser a tecnologia para a nova especificação OPC (OPC Xi) na qual mapeia suas primeiras especificações. O acesso à informação para o WCF é bem mais vasto se comparado com as demais tecnologias, tanto nas questões de documentação existente como na quantidade de desenvolvedores que, para WCF, é muito superior às demais. Além disso, o ambiente de desenvolvimento agradável e intuitivo da plataforma .NET é um ponto positivo para implementação de serviços WCF.

A escolha da tecnologia se deu ainda ao fato do Laboratório Virtual dever se aproximar ao máximo do ambiente industrial real. Uma tendência que é observada é o crescimento no desenvolvimento de aplicações para a indústria com comunicação via OPC, pelas vantagens oferecidas apresentadas neste capítulo.

Capítulo 4

Laboratório Virtual para Estudos de Controle

Neste capítulo, o intuito é apresentar a metodologia de desenvolvimento da ferramenta de Laboratório Virtual que possibilitará o acesso remoto aos simuladores e processos laboratoriais além de suas funcionalidades. A idéia do presente trabalho é deixar a ferramenta disponível com dois simuladores (um simulador para o processo de Reator de Bateladas Sequenciais - RBS e um de Tanques de Agitação Contínua - CSTR) e um processo de temperatura (que se utiliza de um módulo com efeito Peltier), todos executando num ambiente remoto. Porém, esta ferramenta ficará flexível o suficiente para utilização noutros processos laboratoriais. Toda a modelagem dos processos e os procedimentos realizados para o desenvolvimento dos simuladores serão discutidos no capítulo seguinte.

4.1 Análise de Requisitos do *Software* e Especificações

O *software* deverá conter um controle de acesso para usuário com um login e senha a ser registrada na abertura de sessão no servidor, o qual armazenará, num banco de dados contido no mesmo, os dados referentes ao usuário. O cadastro de usuários deverá ser feito a partir de um painel, onde deverá ser registrada uma senha que também será registrada no banco de dados do servidor. Para maior segurança, mesmo com a

segurança que as tecnologias utilizadas oferecem, os dados serão codificados de uma forma segura, com uma codificação MD5 por exemplo, que é um algoritmo de *hash* irreversível bastante utilizado neste tipo situação, para dificultar o acesso de usuários mal-intencionados que visem utilizar a ferramenta a fim de se apoderar de dados de outros usuários.

O *software* deve ser ainda eficiente na aquisição e atualização dos dados, além de demonstrar uma interface ergonômica para visualizar os dados recebidos. Atualizações em tempo real com tabelas demonstrando tanto o histórico como os dados mais recentes, bem como ambientes visualizações gráficas do comportamento dos simuladores e processos remotos são uma boa forma de desenvolver um ambiente agradável de aprendizagem. Deverá ter a capacidade de manter seu desempenho quando utilizado em situações adversas (conexão com a Internet lenta, computadores com pouca memória ou pouca capacidade de processamento, etc.) , sendo tolerante a falhas e mantendo características de recuperabilidade.

A atuação em sistemas remotos deverá ser possível. Assim, um módulo no qual o usuário poderá implementar algumas técnicas de controle PID deverá ser incorporado ao sistema.

O *software* deverá ser compatível, inicialmente, apenas com as versões do *Windows*. No futuro, devido à flexibilidade da tecnologia de comunicação, aplicações para outros sistemas operacionais poderão ser desenvolvidas.

A seguir estão listadas as principais tarefas que o *software* deverá realizar na sua execução:

- Cadastro de usuário;
- Efetuar login no sistema;
- Selecionar um processo disponível;
- Ajustar parâmetros específicos do processo;
- Ajustar parâmetros de controle;
- Ler estado atual do processo;

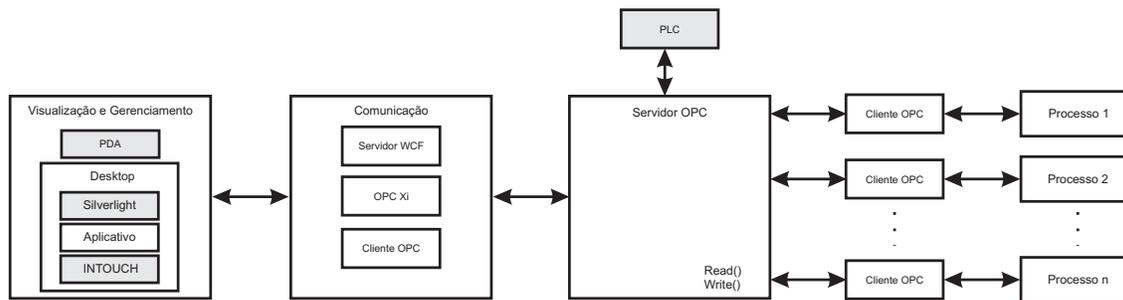


Figura 4.1: Diagrama esquemático do sistema proposto.

- Apresentar os resultados graficamente na tela do programa;
- Apresentar resultados na forma de tabelas;
- Salvar os dados obtidos localmente para que possam ser analisados noutra momento.

4.2 Arquitetura do Sistema

Os modelos em diagramas de blocos devem ser criados para ajudar a entender o que se está querendo desenvolver. Além disso, ajudam a manter a precisão no foco, consistência e um roteiro de tarefas pode ser criado, sendo possível a divisão e o particionamento das tarefas. A representação pode ainda ser mapeada no contexto da implementação.

Um recurso bastante utilizado por programadores para começar a projetar a arquitetura dos sistemas por meio de diagramas de blocos é a *Universal Modelling Language* (UML). A Figura 4.1 apresenta um diagrama de blocos dos componentes do sistema proposto. Vários diagramas de UML tais como de **Classes**, **Casos de Usos** e de **Sequência** podem ser utilizados para descrição de um sistema.

Analisando a Figura 4.1 da direita para a esquerda, pode-se observar vários processos interligados a um servidor via OPC. OPC é a sigla para *OLE (Object Linking and Embedding) for Process Control*, que, como mencionado no capítulo 3, é um padrão que especifica a comunicação entre processos reais e dispositivos de controle de diferentes fabricantes. Este servidor, por sua vez, envia as informações recebidas dos processos por meio de uma camada de comunicação que implementa serviços em WCF. O WCF

torna possível uma computação descentralizada e a troca de informações entre computadores por meio de uma rede.

Por fim, as informações chegam ao computador ou dispositivo remoto e são expostas em telas criadas exclusivamente para o monitoramento e gerenciamento dos processos, através de aplicativos *desktop*, aplicativos *Web* utilizando a tecnologia Silverlight, supervisórios utilizando o INTOUCH e dispositivos móveis. Os blocos na cor cinza do diagrama da Figura 4.1 não serão implementados neste trabalho, mas ficarão como sugestão para trabalhos futuros.

Todos os dados de registro da sessão aberta pelo usuário, bem como os cadastros e os dados dos experimentos realizados pelos mesmos, devem ficar armazenados no banco de dados contido no servidor.

4.3 Codificação e Descrição do Sistema

A ferramenta de Laboratório Virtual para estudos de Controle foi desenvolvida utilizando a linguagem de programação C# na plataforma de desenvolvimento do Microsoft Visual Studio 2008 que oferece todas as tecnologias mencionadas neste documento. Alguns arquivos de configuração do usuário serão armazenados localmente em XML.

O Visual Studio 2008 tem uma interface de ferramenta *Rapid Application Development* (RAD) onde predomina o uso do “seleciona, arrasta e configura” as propriedades. Para a realização das tarefas e eventos, há a necessidade de inserir códigos de programação. Neste ambiente, pode-se desenvolver aplicações utilizando várias linguagens, dentre elas o C# e o Basic .NET.

A interoperabilidade de linguagens é um outro benefício desta plataforma .NET que, com isso, oferece muitas vantagens para o desenvolvimento de *software*. Dessa forma, os recursos compartilhados na rede podem ser utilizados na ferramenta. O pacote gráfico por exemplo, a Microsoft disponibiliza um *kit* de desenvolvimento gráfico, o *MS Chart*, que pode ser integrado ao Visual Studio, que será utilizado para visualização dos dados recebidos dos processos e simuladores remotos.

Para o banco de dados, o SQL será utilizado com o *SQL Server Express Edition*,

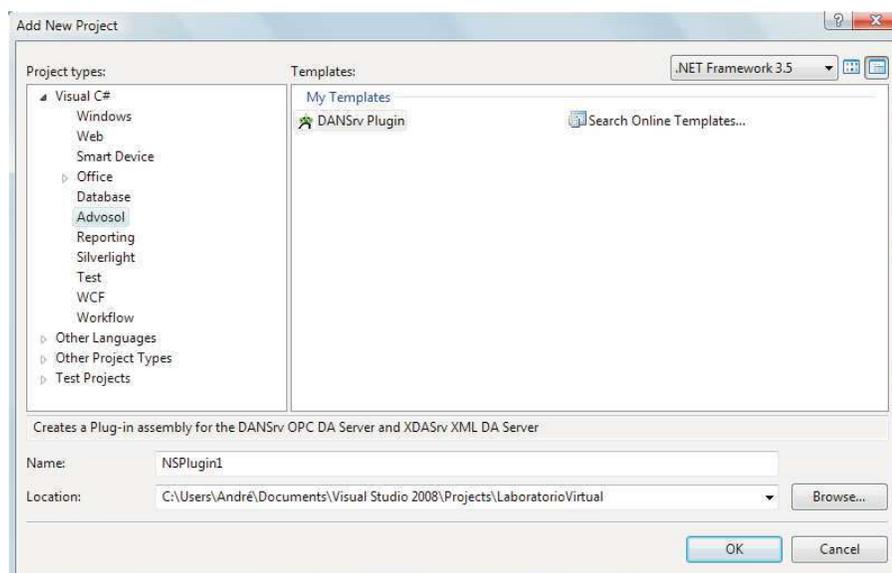


Figura 4.2: Representação da tela de configuração de um novo projeto para serviços OPC no Visual Studio 2008 com o *plug-in* DANSrv OPC DA .NET.

para armazenamento das informações mais relevantes, tais como os dados referentes aos usuários. Os dados experimentais serão registrados em arquivos do *SQL Server Compact Edition*, para dar opção de mobilidade dos dados, podendo serem utilizados para outros fins educacionais.

4.3.1 Serviço OPC

Para criação do serviço OPC foi utilizado o serviço do DANSrv OPC DA .NET que implementa o serviço OPC em um executável com recurso de personalização via *plug-in* como um *assembly* .NET.

A personalização dos *plug-in* é desenvolvida como uma biblioteca de classes em C#, criando, por meio das ferramentas assistentes do Visual Studio, um servidor simples e funcional de acordo com o que o usuário deseja. Após instalado o *plug-in*, a configuração de um novo projeto é muito simples. Os itens necessários para sua configuração já se encontram na tela de adição de novo projeto do Visual Studio (Figura 4.2). Após criado, um serviço pré-configurado já pode ser visualizado e disponível para alterações necessárias.

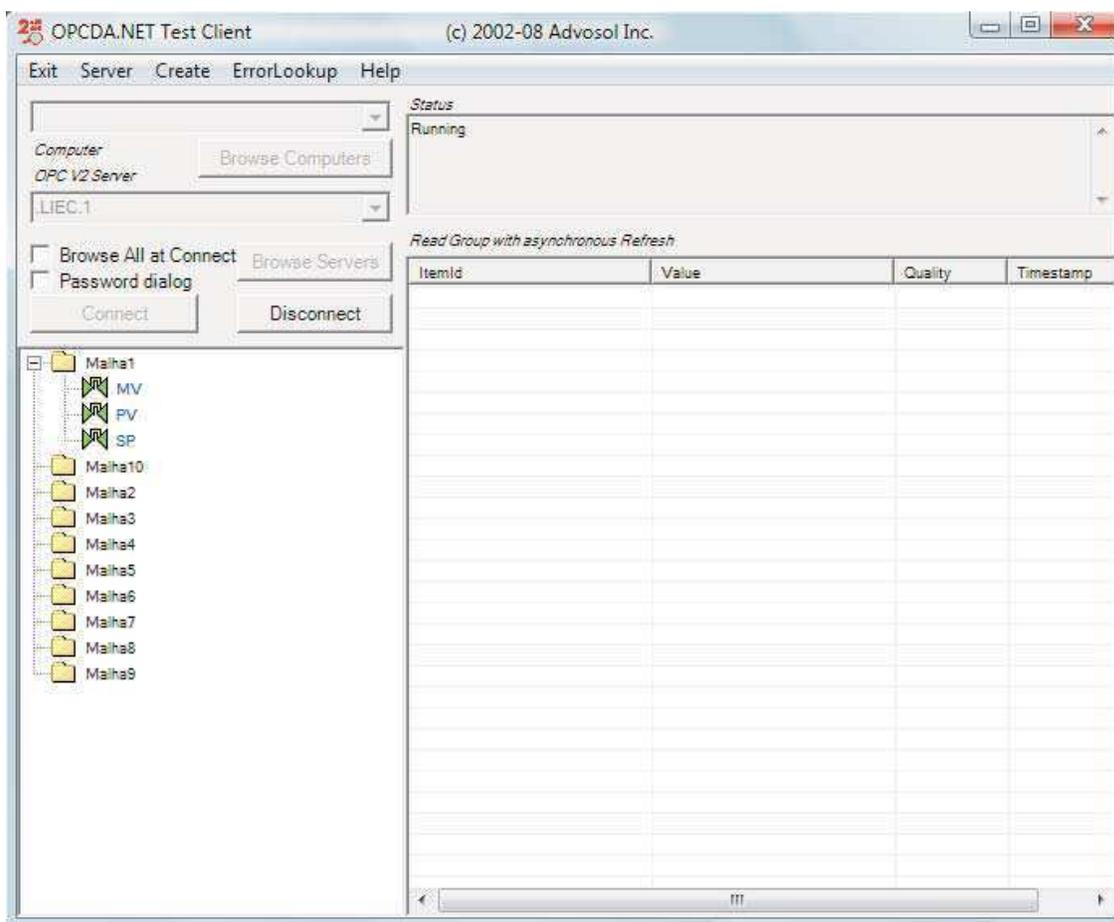


Figura 4.3: Representação da tela do *software* OPCDA.NET Test Client para gerencia dos serviços OPC.

Para iniciar, foi configurado um serviço que poderá receber até dez conjuntos de sinais PV, MV, SP (*Process Variable*, *Manipulated Variable* e *Set Point* respectivamente), o que significa que o sistema está configurado neste momento para aceitar até dez processos conectados no serviço ou dez variáveis de um mesmo processo. Havendo necessidade da utilização de mais processos, o *plug-in* pode ser modificado para a quantidade desejada. Por convenção, cada conjunto de sinais é chamado de Malha-X, onde X representa o número da Malha que varia de 1 a 10. Cada um dos três sinais de cada malha é do tipo double e inicializado com o valor zero.

A partir do *software* OPCDA.NET Test Client fornecido gratuitamente pela Advosol

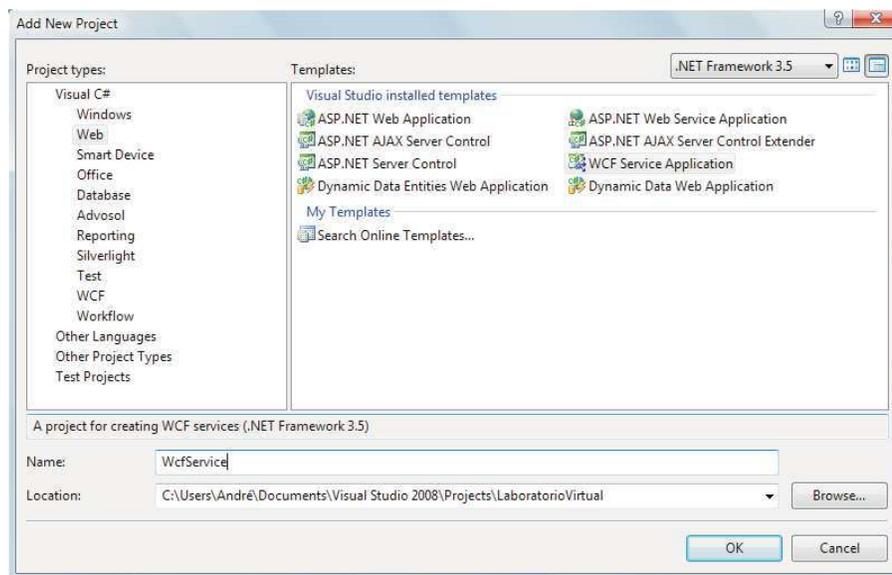


Figura 4.4: Representação da tela de configuração de um novo projeto no Visual Studio 2008 para um serviço WCF.

Inc., cuja tela de apresentação está representada na Figura 4.3 pode-se gerenciar os serviços OPC via uma interface prática e simples e analisar se os serviços estão funcionando corretamente.

4.3.2 Serviço WCF

O serviço WCF é o responsável por coletar os dados do serviço OPC e disponibilizá-los na rede (Internet ou Intranet) para serem acessados remotamente. Para sua criação, um novo projeto do tipo *WCF Service Application* (Figura 4.4) foi criado e configurado para duas classes e uma interface, conforme o diagrama de classes do serviço representado na Figura 4.5.

De acordo com a Figura 4.5, a classe *ServiceWCF* disponibiliza os métodos *Read()*, que lê os dados do servidor OPC disponibilizando-os na rede (variáveis do tipo double), e *Write()* que possibilita a atuação no sistema conectado remotamente. O método *GetData()* serve somente para obter o último valor de entrada no serviço OPC. Existe ainda o método privado *OPCServerConnect()* que realiza a devida conexão com o serviço OPC local utilizando o *plug-in* do *DANSrv OPC DA .NET*. Através do método *Info-*

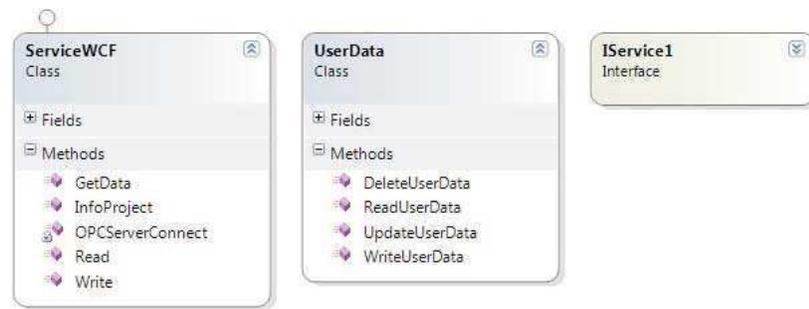


Figura 4.5: Diagrama de classes do serviço WCF proposto.

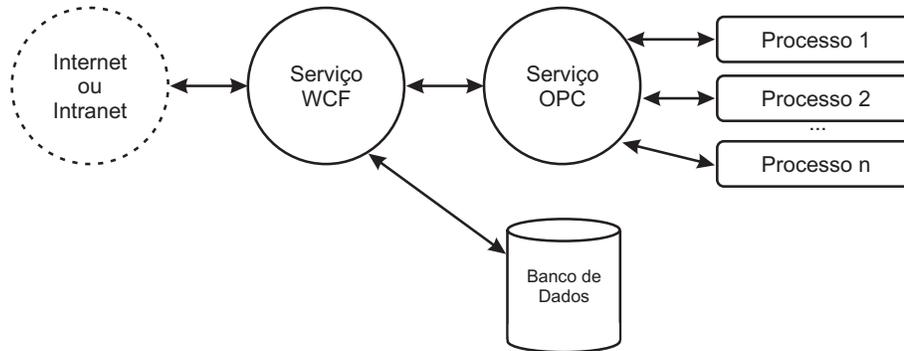


Figura 4.6: Diagrama de blocos da comunicação entre os serviços, banco de dados e processos.

Project() são passadas informações do processo atual em observação (nome do projeto, no qual se refere ao nome do processo, usuário que iniciou, data e hora).

Ainda de acordo com a Figura 4.5 a classe *UserData* disponibiliza métodos para inserção, atualização, leitura de dados e remoção de usuário do banco de dados geral, contendo todas as informações do usuário. O fluxo de informação de leitura e acesso aos dados do usuário e do processo, ficam de acordo com a Figura 4.6.

4.3.3 Banco de Dados

A modelagem do banco de dados foi realizada baseando-se na idéia de que seria possível o armazenamento de algumas informações locais. Os principais objetivos desta modelagem são os seguintes:

- Documentação do *software*;

- Facilitar o entendimento do projeto de usuários e desenvolvedores;
- Representar o ambiente observado;
- Validar e verificar os relacionamentos entre os objetos;
- Manter uma visão geral do projeto;
- Verificar a consistência dos dados.

O Modelo Entidade-Relacionamento (E-R) proposto em (CHEN, 1976) foi desenvolvido para apoiar a elaboração de projetos de banco de dados. Em sua notação original, os diagramas são compostos por:

- **Entidades** - Representados por retângulos, são colocados referências a objetos do mundo real;
- **Relacionamentos** - Representados por losangos, indicam a relação entre as entidades;
- **Atributos** - Representados por círculos, indicam as características das entidades;
- **Linhas** - Unem os atributos às entidades e entidades aos seus relacionamentos.

A Figura 4.7 representa o diagrama de Modelo E-R para o sistema proposto. As entidades e relacionamentos são descritas logo a seguir:

Entidades:

- **Usuário** - Contém as informações necessárias para o usuário entrar no sistema;
- **Dados do Usuário** - Contém informações relevantes dos usuários mas não necessárias para entrada no sistema;
- **Sessão** - Contém data de abertura de sessão dos usuários no sistema;
- **Projeto** - Contém informações relevantes dos projetos criados pelos usuários;
- **Malha** - Contém as informações relevantes do processo estudado;

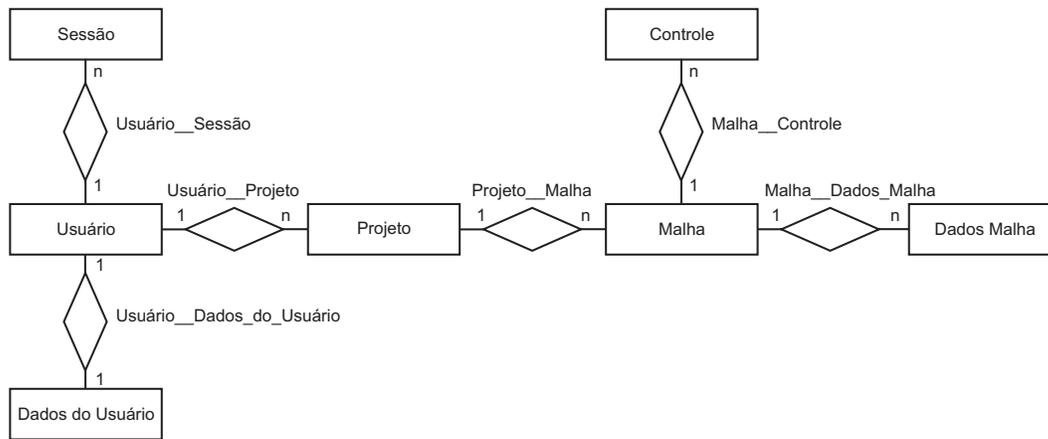


Figura 4.7: Modelo E-R para o sistema proposto.

- **Dados Malha** - Contém os dados coletados dos processos;
- **Controle** - Contém os parâmetros dos controladores PID utilizados.

Baseado no Modelo de Entidade-Relacionamento da Figura 4.7, o banco de dados foi criado utilizando o Microsoft Sql Server 2008 que é a tecnologia de banco de dados que mais se adequa às tecnologias utilizadas neste projeto. O diagrama da Figura 4.8 pôde ser gerado contendo todas as informações mais relevantes do banco, desde os atributos de cada entidade, como seus tipos, chaves (primárias e estrangeiras) e relacionamentos.

O banco de dados local para o usuário é um espelho deste banco geral descrito no diagrama da Figura 4.8. As informações contidas nele não poderão ser alteradas pelo usuário, mas somente observadas por ele. Localmente ao usuário, o banco utilizado é o *Sql Server Compact Edition* na sua versão 3.5, onde todos os dados podem ser armazenados em um único arquivo com extensão sdf. Para a utilização do *Sql Server Compact Edition* foram tomadas alguma precauções porque esta versão do Sql Server é limitada para algumas funções. O banco de dados foi criado de uma forma simples para não haver comprometimento da funcionalidade do banco de dados devido às restrições desta versão.

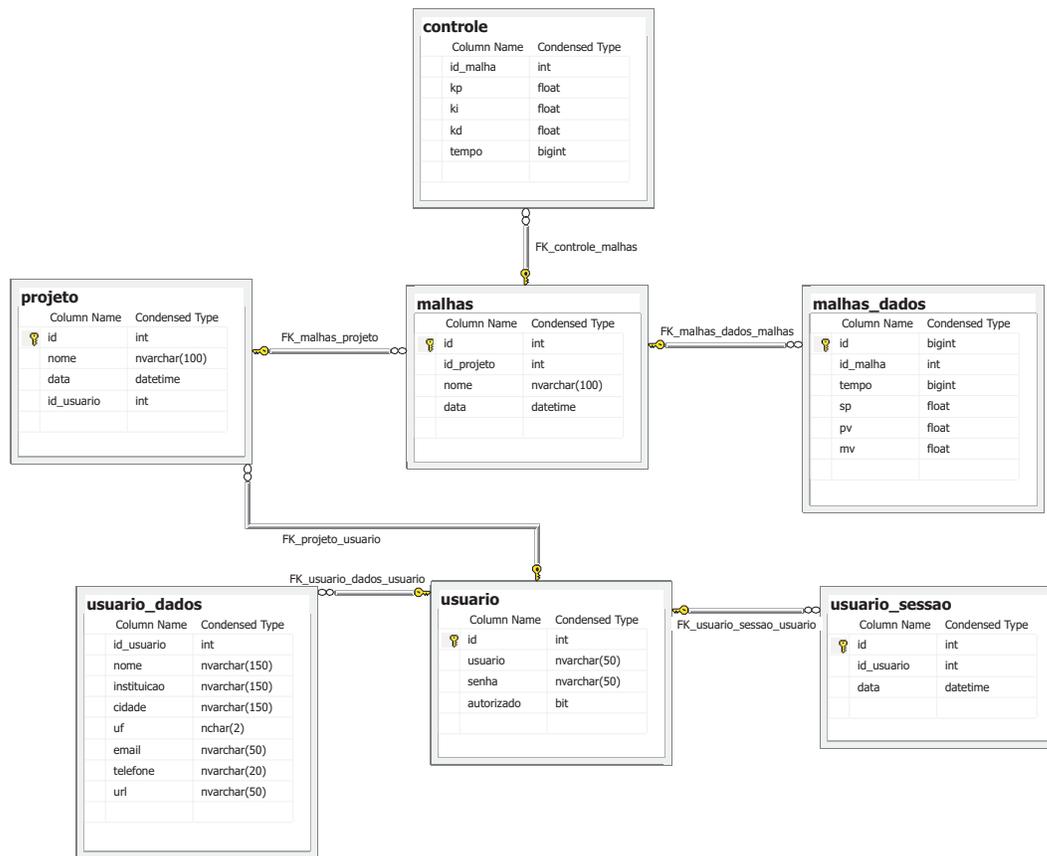


Figura 4.8: Diagrama do banco de dados do sistema.

4.3.4 Software

Laboratório Virtual - Aplicação Remota ao Servidor

Com a ferramenta de Laboratório Virtual desenvolvida neste trabalho, é possível ter o acesso a experimentos laboratoriais e processos industriais remotamente. Isto é possível devido aos serviços WCF (para o envio das informações recebidas via Internet ou intranet) e OPC (para a padronização dos dados recebidos pelos processos). O *software* que acessa estes serviços online foi desenvolvido utilizando a linguagem de programação C# com o Visual Studio 2008 como ferramenta de desenvolvimento.

A idéia das funcionalidades do Laboratório Virtual está representada no fluxograma apresentado na Figura 4.9. Neste diagrama, cada bloco pôde ser considerado uma tela do programa para execução da ação de nome no bloco, ou ponto de acesso para uma

tela.

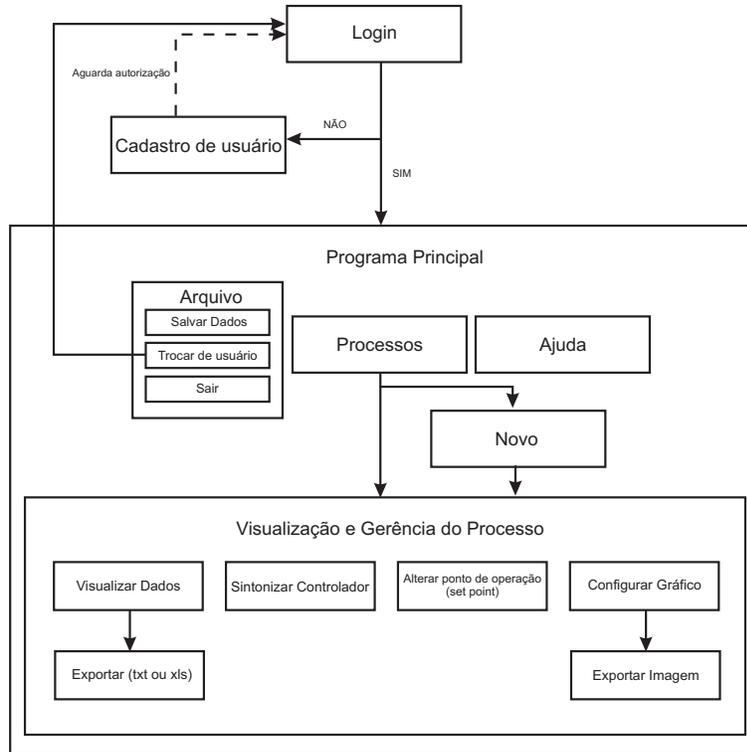
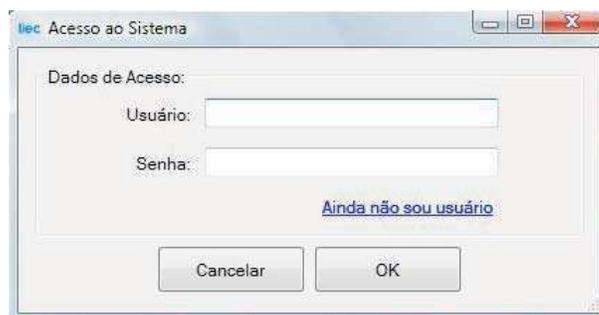


Figura 4.9: Fluxograma de ações no programa.

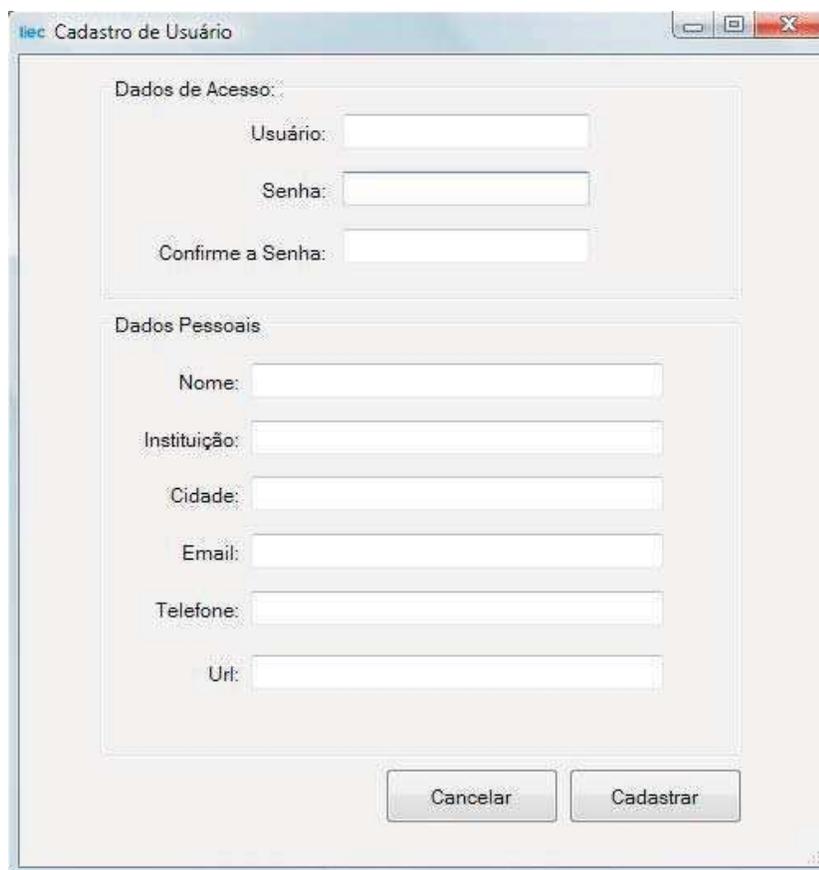
Para iniciar uma sessão, é necessário efetuar um *login* no sistema. Na Figura 4.10 é mostrada a representação da tela de *login* do programa a qual solicita os dados de acesso. Existe a possibilidade, através desta tela, de acessar diretamente a tela de cadastro de usuário. Após cinco tentativas frustradas de *login*, o usuário também é redirecionado para a tela de cadastro, onde poderá fazer um novo cadastro ou recuperar sua senha. Para recuperação de senha, é necessário o preenchimento de alguns dos dados de contato (telefone e email do cadastrado).

A tela de cadastro está representada na Figura 4.11, onde são solicitadas algumas informações. A intenção não é fazer com que o usuário preencha vários formulários para ter acesso ao sistema, visando facilitar ao máximo para não deixar o seu acesso enfadonho. Logo após o cadastro, o usuário não está autorizado imediatamente a atuar nos processos remotos. Um *software* é executado no servidor (descrito na sessão seguinte) autorizando e gerenciando os processos do usuário.



A screenshot of a Windows-style dialog box titled "Acesso ao Sistema". It contains a section labeled "Dados de Acesso:" with two input fields: "Usuário:" and "Senha:". Below these fields is a blue hyperlink that reads "Ainda não sou usuário". At the bottom of the dialog are two buttons: "Cancelar" and "OK".

Figura 4.10: Representação da tela de *login* do Laboratório Virtual.



A screenshot of a Windows-style dialog box titled "Cadastro de Usuário". It is divided into two sections. The top section, "Dados de Acesso:", contains three input fields: "Usuário:", "Senha:", and "Confirme a Senha:". The bottom section, "Dados Pessoais", contains six input fields: "Nome:", "Instituição:", "Cidade:", "Email:", "Telefone:", and "Url:". At the bottom of the dialog are two buttons: "Cancelar" and "Cadastrar".

Figura 4.11: Representação da tela de cadastro de usuário do Laboratório Virtual.

Tendo efetuado o *login* com sucesso no sistema, o usuário tem acesso à tela principal do Laboratório Virtual. A Figura 4.12 representa esta tela. Para começar a acompanhar os processos, o usuário deve dirigir-se ao menu situado na barra superior da

tela principal do programa, selecionar o item “Processos” e escolher dentre os que tem disponíveis. Conforme vem sendo mencionado ao longo deste trabalho, o Laboratório Virtual está disponível para três processos iniciais: O processo com Reator de Bateladas Sequenciais (RBS), Tanques com Agitação Contínua (CSTR) e Módulo Peltier. Escolhendo uma destas três opções, o usuário terá acesso imediatamente à situação atual do processo.

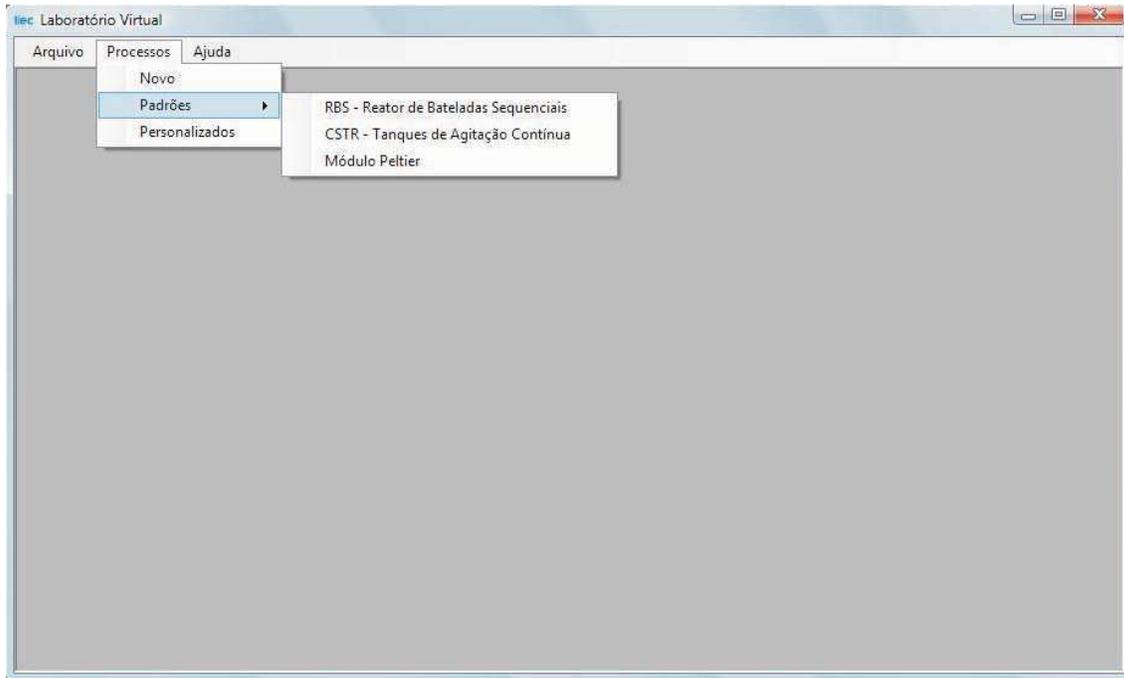


Figura 4.12: Representação da tela principal do programa.

A tela que demonstra a situação atual do processo está representada na Figura 4.13. Em seu gráfico, pode-se observar:

- *SP (Set Point)*: Visualizado graficamente através da curva (1) é a referência para o ponto de operação do processo;
- *PV (Process Variable)*: Visualizado graficamente através da curva (2) refere-se às informações recebidas do processo;
- *MV (Manipulated Variable)*: Visualizado graficamente através da curva (3) refere-se ao sinal de controle de entrada no processo;

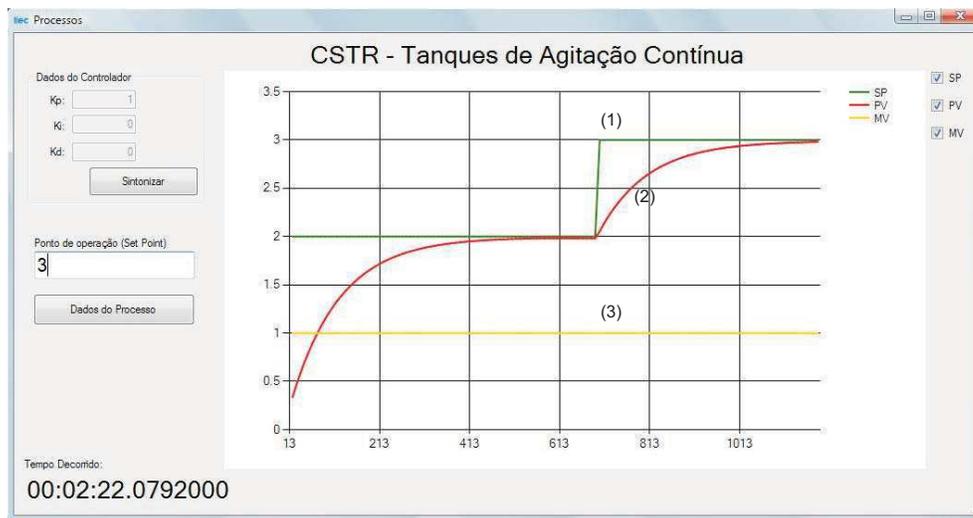


Figura 4.13: Representação da tela de visualização dos dados recebidos do processo remoto.

A partir dos *checkboxes* no canto superior direito da tela (situação atual do processo) pode-se filtrar as curvas que o usuário deseja visualizar e, posteriormente, salvá-la como imagem.

O componente gráfico utilizado no projeto foi o MS Chart que é um componente adicional ao Visual Studio o qual disponibiliza os *assemblies* do *ASP.NET* e *Windows Forms Chart Controls*. O controle MS Chart está associado a uma grade de dados (objeto *DataGrid*). Esta grade de dados é uma tabela que contém os dados que estão sendo mapeados e que pode incluir também as legendas utilizadas para identificar as séries e categorias no gráfico. Quem projeta seu aplicativo gráfico preenche a grade com as informações, inserindo ou importando dados de uma planilha ou array. Mais informações a respeito deste componente podem ser encontradas em (MSDN-MSChart,).

Os dados do controlador podem ser alterados a partir do campo “Dados do Controlador” na parte superior da tela representada na Figura 4.13. Os parâmetros do controlador PID podem ser ajustados a partir do botão “Sintonizar”. A tela representada na Figura 4.14 deve aparecer com as informações dos ganhos de cada componente (Proporcional Integral e Derivativa) do controlador PID. Esses parâmetros, ajustados com as técnicas de controle disponibilizadas na literatura e aprendidas em sala de aula, podem ser utilizados para efetuar o controle do processo, testar e aperfeiçoar novas técnicas de

sintonia PID.

Através da caixa de texto rotulada “Ponto de operação (*Set Point*)” na tela demonstrada na Figura 4.13, o usuário pode ainda alterar o ponto de operação do processo e verificar a eficácia da estratégia de controle utilizada para cada situação.

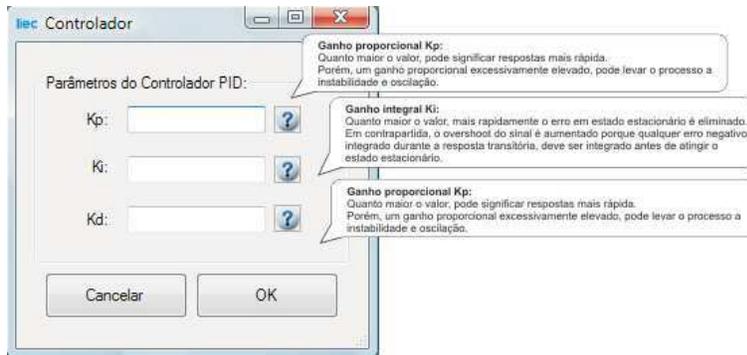


Figura 4.14: Representação da tela do componente para implementação do Controle PID.

Data Aquisição	time	SP	PV	MV
8/18/2010 6:59:...	246	2	1.790153321014...	1
8/18/2010 6:59:...	253	2	1.802888215610...	1
8/18/2010 6:59:...	262	2	1.818096774996...	1
8/18/2010 6:59:...	271	2	1.832091818007...	1
8/18/2010 6:59:...	280	2	1.8449701725513	1
8/18/2010 6:59:...	289	2	1.856820940596...	1
8/18/2010 6:59:...	299	2	1.868882768011...	1
8/18/2010 6:59:...	308	2	1.878825506776...	1
8/18/2010 6:59:...	317	2	1.8879748958936	1
8/18/2010 6:59:...	327	2	1.897287232387...	1
8/18/2010 6:59:...	335	2	1.904141784510...	1
8/18/2010 6:59:...	342	2	1.909737672227...	1
8/18/2010 6:59:...	349	2	1.914983093120...	1
8/18/2010 6:59:...	356	2	1.91989996626...	1
8/18/2010 6:59:...	363	2	1.92450895752	1
8/18/2010 6:59:...	372	2	1.930013175012...	1

Figura 4.15: Representação da tela com visualização dos dados recebidos em tabelas.

Todos os dados informados graficamente são expostos também, de forma textual,

numa tabela que é atualizada em tempo real, a medida que os dados chegam dos processos remotos. Para ter acesso a estes dados, o usuário deve pressionar o botão rotulado “Dados do Processo” que fará surgir a tela com todas as variáveis que estão sendo utilizadas no processo (SP, PV, MV e time) com a respectiva data e hora de aquisição. Na Figura 4.15 está representada a tela que permite esta visualização.

Pode-se verificar ainda na Figura 4.15, que existe a possibilidade de exportar os dados para arquivos no formato de planilhas do Excel (arquivo com terminação .xls) ou para o formato de texto (txt) permitindo que o usuário utilize os dados coletados para análise noutros programas. A tela de exportação dos dados nestes formatos está representada na Figura 4.16.



Figura 4.16: Representação da tela de exportação de dados noutros formatos de arquivos.

Existe a possibilidade do usuário incorporar novos processos à ferramenta. Para isso, não há a necessidade de entender alguma linguagem de programação ou as tecnologias envolvidas, basta conectar o processo (ou simulador) em alguma das malhas do servidor OPC criado e, por meio da tela de configuração da ferramenta do Laboratório Virtual, colocar um nome apropriado para o processo e indicar a malha utilizada que recebe os dados do processo (SP, PV e MV). Para realizar este procedimento, basta acessar no menu da tela principal do *software* o item “Novo” para visualizar a tela apresentada na Figura 4.17. Caso a ferramenta não consiga realizar a conexão com o novo processo com as configurações realizadas, uma mensagem de erro surgirá na tela e o processo ficará inativo no menu principal até que seja feita a primeira conexão.



Figura 4.17: Componente para inserção de um novo processo.

Aplicação Local ao Servidor

Há a necessidade de um aplicativo executando no servidor que realiza a gerência dos usuários cadastrados. Ao se cadastrar, o usuário não é autorizado automaticamente pelo sistema. Um *software* executando no servidor pode ser configurado para autorizar os usuários cadastrados por meio de um administrador (que autoriza manualmente os usuários). Ideal para autorizar somente os usuários de uma turma específica de uma disciplina por exemplo. Ou ainda, automaticamente a cada hora (por meio de um evento de *timer* que realiza a autorização dos usuários na última hora registrada), podendo o administrador optar por qualquer uma dessas duas opções, dependendo da situação.

Este *software* também monitora o controle de acesso dos usuários, registrando as sessões dos usuários que entram no sistema e começam a realizar seus experimentos no banco de dados local ao servidor.

4.4 Conclusão

O sistema foi desenvolvido com base na arquitetura e especificações demonstradas neste capítulo. A modelagem por meio dos diagramas e análises demonstradas neste capítulo, não só do banco de dados mas dos serviços (OPC e WCF) e da ferramenta final, foi de extrema importância para manter documentado todos os passos de desenvolvimento bem como servir de orientação para os desenvolvedores, que desejem implementar outros tipos de processos na ferramenta, e usuários que utilizarão a ferramenta como ponto de base para o entendimento mais abrangente dos processos e simuladores.

A ferramenta permite a inserção de até dez processos (podendo esta quantidade ser alterada) de forma rápida e sem a necessidade do conhecimento de programação por parte do usuário. Os serviços e a ferramenta podem ser aperfeiçoados de tal forma que possam atuar de vários outros meios nos processos. Para simuladores, por exemplo, pode-se criar funções no serviço de WCF que façam trafegar informações a respeito da planta utilizada no processo e diversas variáveis (tais como alteração no volume do reator, temperatura, concentração inicial, etc. e diversas outras variáveis que influenciam na resposta e fazem parte de um processo).

Capítulo 5

Descrição dos Processos

Neste capítulo, são apresentados os principais conceitos e características dos processos implementados em simuladores e dos processos reais que são analisados e implementados na ferramenta de Laboratório Virtual. O objetivo é familiarizar o leitor com os conceitos que envolvem o funcionamento de cada um dos processos, destacando suas importâncias e como e para quê estes sistemas são utilizados no ambiente industrial.

5.1 Reator de Bateladas Sequenciais - RBS

Um sistema com Reator de Bateladas Sequencias (RBS) é responsável por efetuar o tratamento de esgotos no regime de bateladas, ou seja, introduz-se no sistema um determinado volume de esgoto bruto a ser tratado e somente após os processos de higienização desse material terem sido realizados, o efluente tratado é descartado e uma nova batelada é introduzida no sistema (BARBOSA, 2009).

Um sistema aeróbio é aquele que utiliza oxigênio como oxidante da matéria orgânica que está presente no esgoto. O oxigênio é injetado no sistema por meio de aeradores que realizam a transferência de oxigênio da atmosfera para dentro do reator. Os sistemas de lodo ativado são os mais difundidos dentre os sistemas que utilizam mecanismo aeróbio. O sistema, conhecido hoje como Lodo Ativado, foi proposto pelos pesquisadores Arden e Lockett em 1914 e trata de um lodo composto por bactérias, fungos, algas e protozoários responsáveis pela utilização do substrato e nutrientes presentes naquele meio

para seu crescimento e fonte de energia.

Define-se ainda o lodo ativado como um floco produzido num esgoto bruto ou decantado pelo crescimento de bactérias zooglêias ou outros organismos, na presença de oxigênio dissolvido e acumulado em concentração suficiente graças ao retorno de outros flocos previamente formados (JORDÃO; PESSOA, 1995).

Em (BARBOSA, 2009) e (LIRA, 2001) foi realizada a modelagem de um sistema de RBS, que se encontra na categoria de lodo ativado de fluxo intermitente. A modelagem do sistema RBS foi realizada baseada no modelo *Activated Sludge Model N^o 1* (ASM1), que descreve o comportamento de sistemas de lodo ativado utilizados na remoção de matéria orgânica e nitrogênio de efluentes. O modelo ASM1 é apresentado em (HENZE et al., 1987), onde são colocadas treze variáveis de estado, oito processos e dezenove parâmetros cinéticos e estequiométricos. Em (GUIMARÃES, 2003) podem ser encontradas as principais vantagens e desvantagens que estes sistemas têm em relação aos processos de fluxo contínuo. Em (BARBOSA, 2009) as etapas do processo utilizando RBS são detalhadas da seguinte forma:

- Enchimento - o sistema recebe o esgoto a ser tratado;
- Hidrólise - a biomassa é mantida em suspensão por meio de agitação mecânica, ocorrendo, assim, reações químicas que fazem com que haja uma ação microbiana mais eficiente;
- Aeração - é feita a transferência de oxigênio para o licor misto, promovendo oxidação de matéria orgânica pela ação das bactérias, bem como a redução de amônia para nitrato;
- Sedimentação - promove a clarificação do efluente tratado e o adensamento do lodo ativado;
- Descarte de efluente - consiste na retirada do efluente tratado.

Os reatores de mistura perfeita são utilizados, normalmente, para se iniciar a modelagem de processos permitindo uma abordagem simplificada, pois não são consideradas

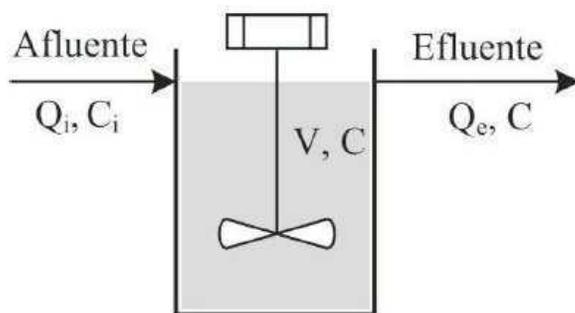


Figura 5.1: Representação de um reator de mistura perfeita

diversas variáveis que em condições reais normalmente influenciam, tais como temperatura ambiente, composições dos materiais envolvidos, etc. Um reator de mistura perfeita é apresentado na Figura 5.1. Dessa forma, sua modelagem por meio de equação de balanço de massa da concentração C de um determinado componente é descrita pela equação (5.1)

$$\frac{d(VC)}{dt} = Q_i - QC + V\phi \quad (5.1)$$

onde Q_i é o fluxo de entrada do líquido, C_i a concentração de entrada do componente analisado, Q representa o fluxo de saída líquido e C é a concentração de saída do componente. V representa o volume do reator e ϕ a taxa de reação do processo.

Assumindo que V é constante, encontra-se, na equação (5.1) que $Q = Q_i$, que também serão constantes. Dessa forma, a equação de mistura perfeita pode então ser reescrita da forma da equação (5.2). Com Q e Q_i constantes, pode-se ainda encontrar a equação (5.3) já que, como característica do processo, para as etapas de hidrólise e aeração no RBS, não existe fluxo hidráulico entre as etapas do processo.

$$\frac{dC}{dt} = \frac{Q_i}{V}C_i - \frac{Q}{V}C + \phi \quad (5.2)$$

$$\frac{dC}{dt} = \phi \quad (5.3)$$

No modelo ASM1 tem como base a equação (5.4), pois a variação da concentração de um determinado componente por unidade de tempo, mediante a ocorrência dos pro-

cessos descritos no modelo, é igual às taxas de reação R_i obtidas a partir da matriz de reações do Modelo ASM1 (BARBOSA, 2009) e (LIRA, 2001).

$$R_i = \sum_j v_{ij} p_i \quad (5.4)$$

onde R_i representa as taxas de reações das treze variáveis de estado do modelo, v_{ij} os elementos da matriz de reações e p_i os oito processos descritos no modelo ASM1.

Um outro ponto importante na modelagem do RBS é o modelo do decantador. O balanço de massa em cada camada deve ser realizada. O transporte das partículas é função do fluxo de alimentação do decantador (J_b) e da força gravitacional (J_g), que, por sua vez, atua apenas no transporte das partículas das camadas superiores para as camadas inferiores.

O fluxo devido à gravidade é dado pelo produto da concentração dos sólidos X pela velocidade de sedimentação v_s expresso pela equação 5.5.

$$J_g = v_s X \quad (5.5)$$

O fluxo de sólidos em cada camada devido à gravidade é limitado pelo fluxo de sólidos que pode ser assegurado pela camada logo abaixo, portanto, para uma determinada camada i pode-se encontrar a equação 5.6.

$$J_{g,i} = \min(J_{g,i}, J_{g,i+1}) \quad (5.6)$$

Para as camadas acima do ponto de alimentação, deve-se aplicar uma outra condição para o fluxo, devido à gravidade, exposta na equação 5.7. Para as camadas de baixo, a condição da equação 5.8 deve também ser aplicada.

$$v_{b_{up}} = \frac{Q_e}{A} \quad (5.7)$$

$$v_{b_{dn}} = \frac{Q_u}{A} \quad (5.8)$$

onde Q_e e Q_u são os fluxos no topo e no fundo do decantador e A representa a área da seção reta do decantador.

Na camada de alimentação existe tanto fluxo ascendente como descendente. Assim, o fluxo devido ao movimento da massa líquida é encontrado de acordo com a equação 5.9.

$$J_{b,i} = v_{b,i}X_i \quad (5.9)$$

Desta maneira, a dinâmica da concentração de sólidos para cada uma das camadas modeladas no decantador é descrita por (5.10), onde ΔJ representa a diferença de fluxo em cada camada, h representa a altura de cada camada e X a concentração de sólidos suspensos presentes no decantador. Mais informação para este modelo pode ser encontrado também em (LIRA, 2001) e (BARBOSA, 2009).

$$\frac{dX}{dt} = \frac{1}{h}\Delta J \quad (5.10)$$

O Simulador de RBS

Uma ferramenta de simulação do Reator de Bateladas Sequencias (RBS) foi desenvolvida para ser utilizada por operadores e projetistas das estações de tratamento de esgotos que poderão realizar treinamentos e simulações, obtendo informações suficientes para levar a frente seus projetos, otimizando-os, prevendo o comportamento de determinadas variáveis, podendo atuar nestes sistemas com técnicas cada vez mais eficientes. Nas Figuras 5.2 e 5.3 são demonstradas algumas telas da versão local deste simulador. Na Figura 5.4 é mostrado um diagrama de blocos com os componentes do simulador.

No processo de RBS, o início da etapa de Aeração, há uma significativa atividade bacteriana e o consumo de oxigênio no sistema é bastante elevado. Praticamente todo o oxigênio inserido no sistema é absorvido, deixando a concentração deste componente no interior do reator muito baixa. Quando a matéria orgânica existente no esgoto é praticamente consumida pelas bactérias, a concentração de oxigênio tende a subir rapidamente, significando que todo o oxigênio injetado no sistema não é mais utilizado pelas bactérias, já que as mesmas não estão mais atuando sobre o esgoto. O que acontece agora é o desperdício do oxigênio injetado, sendo assim, liberado para o ambiente.

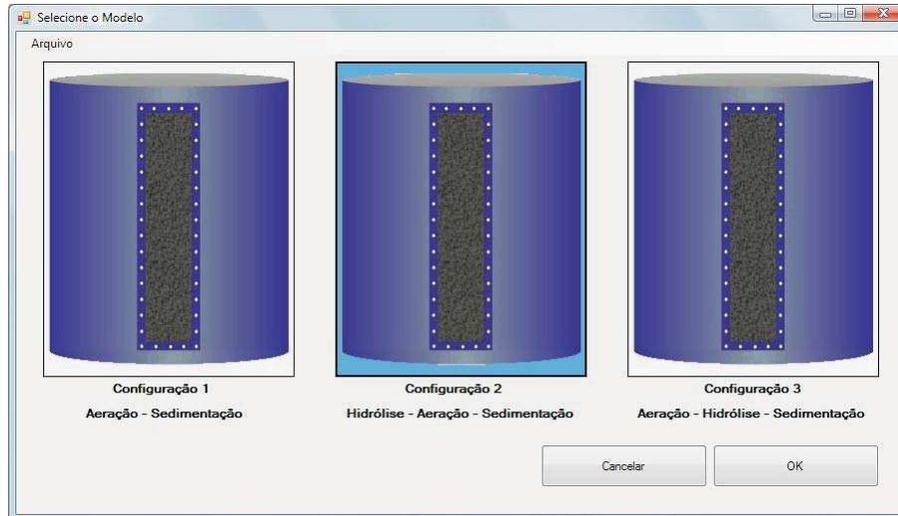


Figura 5.2: Representação da tela de escolha de configuração do simulador de RBS

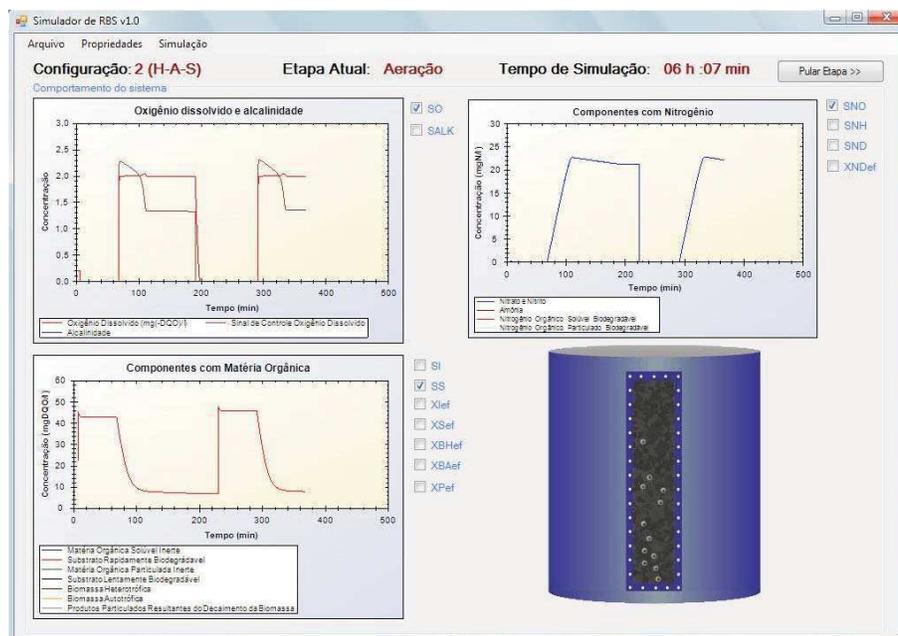


Figura 5.3: Representação da tela principal do simulador de RBS

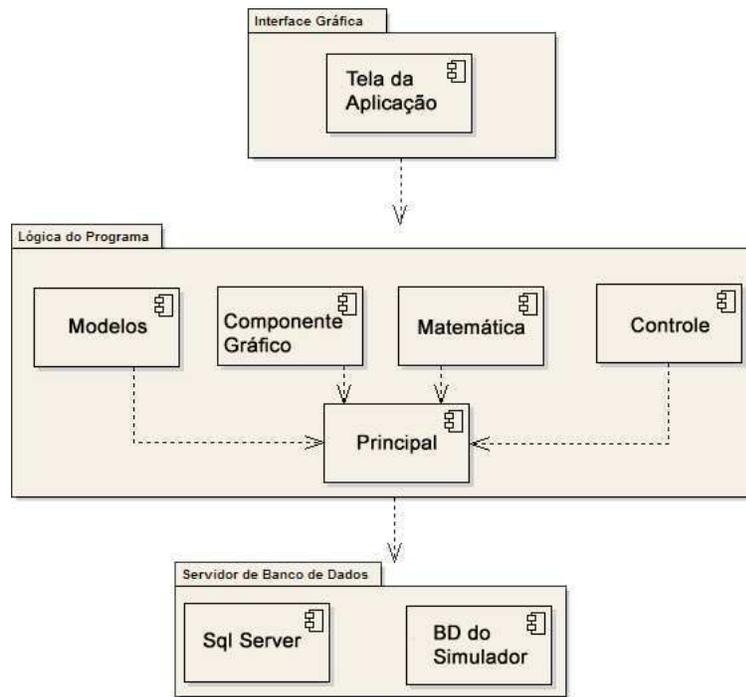


Figura 5.4: Diagrama de componentes do simulador

Em (BARBOSA, 2009) foi apresentada esta ferramenta de forma preliminar para simulação de RBS, desenvolvida utilizando a linguagem de programação C Sharp (C#). Esta ferramenta foi aperfeiçoada com o módulo de comunicação OPC que disponibiliza todos os dados num servidor OPC, além de que houve o aperfeiçoamento na parte de modelagem do sistema (ajustes das equações de balanço de massa e energia) bem como melhoramento na funções que implementam a integração das equações diferenciais (melhoramento no algoritmo da função que implementa o integrador de Runge Kutta de quarta ordem). Foi implementado ainda um componente de controle PID, cuja idéia é controlar a concentração de oxigênio dissolvido na etapa de aeração, num patamar que leve as bactérias de dentro do reator ao seu máximo rendimento, ou seja, consumir toda a matéria orgânica sem desperdiçar oxigênio injetado. Foi escolhido somente o controle deste componente porque estudos comprovam que o consumo de energia dos aeradores representa, normalmente, mais de 50% dos custos operacionais de um sistema como este, o que significa que implementar um controle bem feito desta etapa do processo poderá representar uma diminuição significativa nestes custos.

Com a realização de estudos com este simulador, pode-se definir estratégias cada vez mais eficientes de controle e identificar, sem prejuízo de capital ou operacional, possíveis erros de projeto e/ou operação da planta e, a partir daí, estabelecer critérios específicos a cada caso analisado.

Os sistemas de Aeração poderão ser dimensionados de tal forma que a necessidade de oxigênio seja menor, pois, para os sistemas de RBS, a transferência de oxigênio no período de aeração é algo em torno de 3 a 4 vezes maior que no sistema convencional (BARBOSA, 2009). A solução deixaria o sistema com um custo operacional bem mais baixo e viável que os processos RBS sem um sistema de controle.

O simulador de RBS pode ser utilizado em treinamentos e simulações para verificação do comportamento do sistema para diversos tipos de controle. O algoritmo do módulo de controle PID pode ser modificado e reutilizado para testes com outras técnicas de controle.

5.2 Tanques de Agitação Contínua (*CSTR - Continuous Stirred-Tank Reactor*)

Em um processo de CSTR, reagentes fluidos são introduzidos em um tanque equipado com um rotor para prover a agitação e garantir uma mistura adequada. Além disso, o tanque possui um sistema de aflente para entrada dos componentes e eflente para remoção dos mesmos. A partir de parâmetros como o volume do tanque, taxa do fluxo de entrada e saída dos reagentes, tempo médio que os componentes permanecem no tanque (tempo médio de residência), entre outros, é possível realizar a modelagem e observar o comportamento do sistema.

O modelo de CSTR baseia-se na proposição de que uma reação exotérmica que acontece no interior dos tanques e o calor é removido por um refrigerante.

O sistema pode ser modelado por meio de equações de balanço de massa e energia, cujo comportamento pode ser descrito da seguinte forma:

$$[\textit{acumulado}] = [\textit{o que entra}] + [\textit{gerado}] - [\textit{o que sai}]$$

O que significa que os cálculos são feitos na proposta de que acontece uma mis-

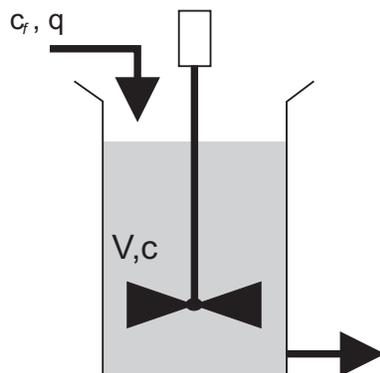


Figura 5.5: Representação ideal de CSTR

tura perfeita no interior do tanque. Uma representação gráfica de um CSTR pode ser verificada na Figura 5.5.

Para uma reação exotérmica $X \rightarrow Y$ a equação de balanço de massa e energia é (ASTROM; HAGGLUND, 2006):

$$\frac{dc}{dt} = \frac{q}{V}(c_f - c) - k(T)c \quad (5.11)$$

onde c é a concentração do componente X dado em $kmol/m^3$, c_f a concentração de X na alimentação, q é a taxa do fluxo que entra dado em m^3/s , V o volume do reator (m^3) e $k(T)$ a taxa que é uma função da temperatura descrita por:

$$k(T) = k_0 e^{-E/RT} \quad (5.12)$$

O primeiro termo do lado direito de (5.11) refere-se ao fluxo de massa que está entrando no tanque, já o segundo à taxa de remoção de X por conta da reação.

Analisando agora a energia do sistema, a equação (5.13) descreve o balanço de energia:

$$\frac{dT}{dt} = \frac{q}{V}(T_f - T) + k(T) \frac{-\Delta H}{\rho C_p} c + \frac{UA}{\rho V C_p} (T_c - T) \quad (5.13)$$

onde ΔH é a reação de calor ($kJ/kmol$), ρ (kg/m^3) é a densidade do componente X , C_p o calor específico (kJ/kgK), U é o coeficiente de transferência de calor ($J/min/K/m^2$), A é a área de secção transversal do tanque (m^2), T_c a temperatura do refrigerante (K), e T_f

a temperatura de alimentação (K). Observa-se que o primeiro termo do lado direito da equação de balanço de energia (5.13) refere-se à taxa de energia que está sendo injetada no sistema, o segundo termo representa a potência gerada pela reação, e o último termo a energia removida devido à refrigeração.

Uma primeira análise deste sistema pode ser feita observando a solução para o estado estacionário. Neste estado, a taxa de fluxo de massa que entra no reator deve ser igual à taxa de fluxo que sai, caso contrário, o reator pode esvaziar (para a situação do fluxo que sai seja maior que o fluxo que entra) ou extrapolar (para a situação contrária). Considerando a equação de balanço de massa descrita em (5.11), a mesma pode ser reorganizada para encontrar:

$$c = \frac{1}{1 + Vk(T)/q} c_f \quad (5.14)$$

Substituindo a equação (5.14) na equação (5.13), tem-se $c = c_f$ para o estado estacionário. Dessa forma, observa-se que a potência gerada pela reação é dada por:

$$P_g = \frac{k(T)}{1 + Vk(T)/q} (-\Delta H)c \quad (5.15)$$

e a taxa de remoção de energia é:

$$P_g = \frac{k(T)}{1 + Vk(T)/q} (-\Delta H)c \quad (5.16)$$

Simulador de CSTR

O simulador de CSTR será implementado com base nos modelos de balanço de massa e energia descritos.

Quando posto em funcionamento, os dados referentes à concentração c e temperatura T do fluido, deverão ser observados e enviados para um servidor OPC. O princípio é o mesmo do simulador RBS, só que neste caso o processo em questão é bem mais simples pois somente estas duas variáveis podem ser observadas e controladas.

Os parâmetros que serão utilizados na reação exotérmica são aqueles demonstrados na tabela 5.1 de acordo com (ASTROM; HAGGLUND, 2006).

$q = 0.002m^3/s$	$T_f, T_c = 300K$
$V = 0.1m^3$	$E/R = 8750K$
$\rho = 1000kg/m^3$	$c_f = 2kmol/m^3$
$C_p = 4kJ/kgK$	$UA = 50W/K$
$k_0 = 3 * 10^8 s^{-1}$	$\Delta H = -5 * 10^5 kJ/kmol$

Tabela 5.1: Parâmetros de um CSTR exotérmico

5.3 Módulo Peltier

Os dispositivos de efeito Peltier produzem um gradiente de temperatura quando submetidos a uma diferença de potencial. O módulo Peltier do processo estudado é composto por materiais diferentes, placas condutoras e semicondutoras. No efeito Peltier, um gradiente de temperatura é alcançado quando uma corrente passa diretamente em um ou mais pares termelétricos de tipo n e p . Ocorre um aquecimento do material quando a corrente é aplicada no sentido $p \rightarrow n$ do material semicondutor e no sentido contrário, $n \rightarrow p$, ocorre o resfriamento do material. Seu funcionamento está representado na Figura 5.6. Um dispositivo de refrigeração, um ventilador e um trocador de calor, é conectado ao sistema para realizar a transferência do calor entre as faces do módulo Peltier.

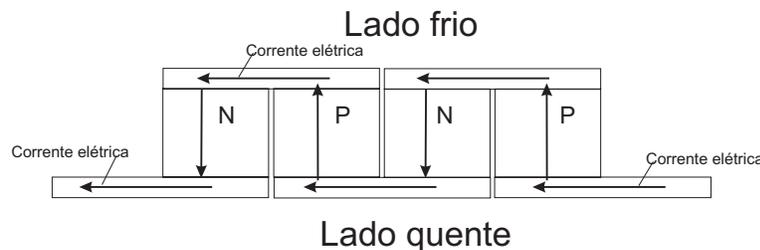


Figura 5.6: Representação do funcionamento do sistema peltier

O efeito Peliter foi observado pela primeira vez em 1834 por Jean Charles Athanase Peltier. O efeito Seebeck é o efeito contrário ao de Peltier, no qual uma corrente elétrica circula entre duas junções de condutores ou semicondutores de materiais diferentes é produzida quando essas junções são submetidas a uma diferença de térmica. Este efeito foi observado pela primeira vez pelo físico Thomas Johann Seebeck, em 1821, antes do efeito Peltier.

No sistema do laboratório, que a ferramenta de Laboratório Virtual deve prover o gerenciamento e implementação de técnicas de controle, há um sensor de temperatura do tipo LM35 e um microcontrolador do tipo Aduc812 programado para enviar e receber os dados de temperatura fornecidos pelo LM35 e enviar, via interface RS-232, as informações lidas.

O comportamento dinâmico deste processo de obtenção de energia térmica a partir de elétrica é não-linear e um pouco complexo. As propriedades físicas dos materiais semicondutores utilizados são as principais causas desta complexidade. Uma modelagem para este processo foi apresentada em (BARROS; JÚNIOR; SANTOS, 2007) e pode ser expresso pela equação de balanço de energia conforme a equação (5.17).

$$(M_L C_L + M_C C_C) \frac{dT_L}{dt} = Q_L - Q_k - I \alpha_{pn} T_L \quad (5.17)$$

onde M_L e C_L é a massa e a capacidade calorífica do metal; M_C e C_C é a massa e capacidade calorífica da parte fria da placa; T_L , Q_L , Q_k , I e α_{pn} , a temperatura da parte fria da placa, o calor absorvido pelo metal e conduzido até a parte quente, a condução de calor na parte fria no limite do peltier, a corrente aplicada e o coeficiente de Seebeck do material termelétrico. O coeficiente de Seebeck é obtido a partir da tensão e diferença de temperatura obtida.

Do balanço de energia aplicado ao material termelétrico obtém-se a equação (5.18).

$$C \gamma \frac{\partial T(x,t)}{\partial t} = k \frac{\partial^2 T(x,t)}{\partial x^2} - \frac{\tau}{A} I \frac{\partial T(x,t)}{\partial x} + \frac{\rho}{A^2} I^2 \quad (5.18)$$

com C e γ sendo a capacidade calorífica e densidade do material termelétrico; $T(x,t)$ é a temperatura de distribuição do peltier; k é a condutividade térmica do material $p-n$; τ é o coeficiente de Thompson definido por $T d\alpha_{pn}/dT$; A é a área total de secção transversal do material termelétrico; ρ é a resistividade elétrica do material termelétrico.

O balanço de energia também pode ser aplicado ao trocador de calor, a modelagem da parte quente da placa pode então ser expressa pela equação (5.19). Esta equação fecha o conjunto das três equações que torna possível realizar estudos sobre o referido processo.

$$(M_F C_F + M_H C_H) \frac{dT_H}{dt} = I \alpha_{pn} T_H + Q_o - h A_F (T_H - T_a) \quad (5.19)$$

onde M_F e C_F é a massa e a capacidade calorífica do trocador de calor; M_H e C_H da extremidade quente da placa; Q_o é a condução de calor no limite do peltier; h e A_F é o coeficiente de transferência de calor convertido e o calor total na superfície do trocador de calor; T_a é a temperatura ambiente.

A forma de aquisição de dados é feita por um sensor LM35 que envia informações para um microcontrolador AduC812 e este, por sua vez, conecta-se a um computador via RS-232 e envia a informação da temperatura do sensor para um computador. Esta temperatura varia entre 10 e 60 °C quando a temperatura ambiente está em torno dos 23 °C. A potência é aplicada utilizando atuadores PWM enquanto o sensor de temperatura realiza suas medições.

Um *software* é encarregado de ler e escrever os dados na porta serial. O microcontrolador está programado para receber comandos que comecem com o caractere **L** seguidos do número do endereço do dispositivo (já que este módulo está preparado para receber mais de um processo com o microcontrolador mencionado). O próximo algoritmo define a função a ser usada, se será analógica ou digital e, por fim, se quer acionar a função de leitura ou escrita. Para explicar melhor, estes comando estão listados abaixo. A listagem abaixo é genérica para uma rede de dispositivos, onde o caractere **N** indica o número de endereço do mesmo. Para todos os efeitos, como está sendo considerado apenas um dispositivo, o do módulo peltier, este está endereçado como **2**, ou seja, em todos os comandos abaixo pode-se substituir o **N** pelo **2**. A Figura 5.7 representa a imagem real do processo na bancada do laboratório.

- **LNAR0**: Lê o canal analógico 0.
- **LNAW00000**: Escreve no canal analógico 0 a palavra 0000. Esta palavra pode variar de 0000 a 4095, pois o conversor de digital do microcontrolador é de 12 bits.
- **LNDR0**: Lê a porta digital 0.
- **LNDW0002**: Escreve na porta digital 0 a palavra 002. Cada porta possui 8 bits,

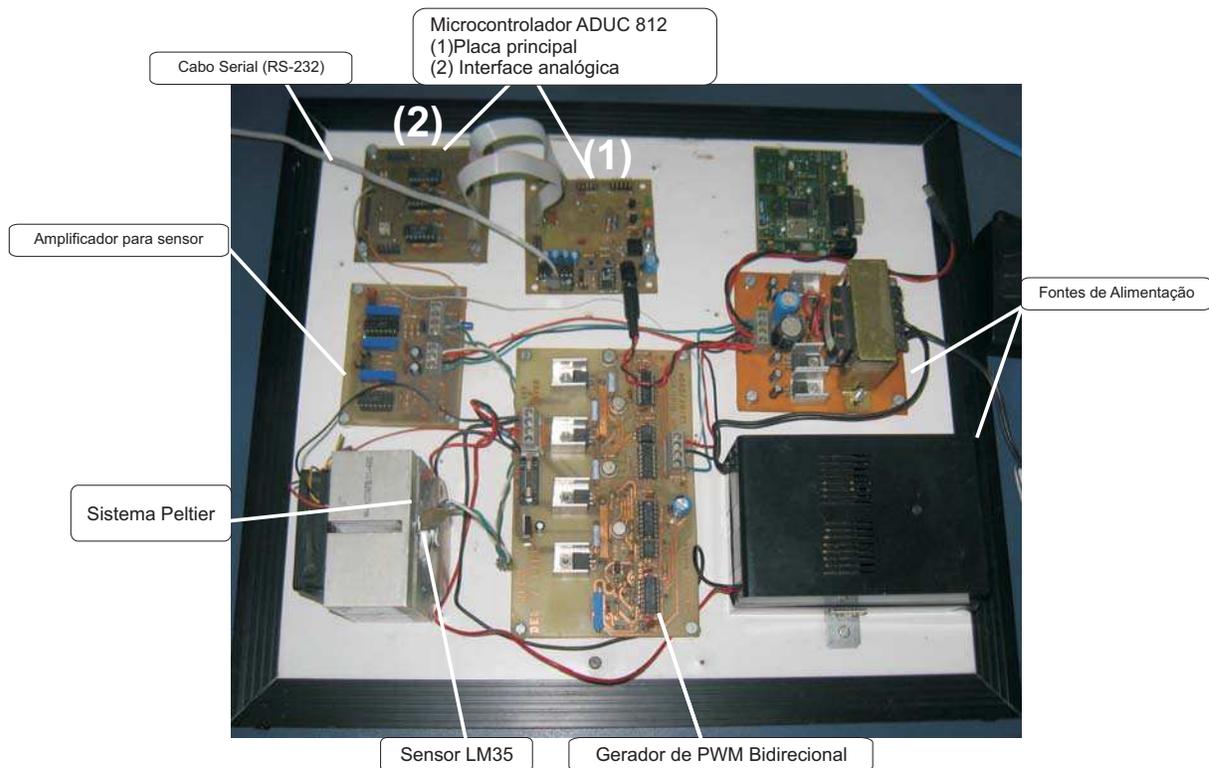


Figura 5.7: Foto do processo com Módulo Peltier

podendo ser escrito de 0 a 256.

5.4 Conclusão

Neste capítulo, foram apresentados os três processos iniciais tratados na ferramenta de Laboratório Virtual baseada em serviço. O entendimento e modelagem de cada processo foi de extrema importância pois, a partir daí, pôde-se implementar, com mais facilidade, rotinas de programação cujo resultados se aproximassem dos resultados reais e já simulados noutras ferramentas disponíveis de simulação (MatLab, Simulink, entre outros).

Pode-se ainda realizar testes nas ferramentas e entender o funcionamento da mesma, observando fenômenos próximos ao comportamento real dos sistemas envolvidos, o que permite estudos mais profundos e uma melhor especificação para possíveis projetos e treinamento de operadores.

Vale salientar que a estratégia envolvida de comunicação da ferramenta com os processos e simuladores, não se limita somente às propostas neste capítulo. Dois processos simulados (RBS e CSTR) e um processo real (Módulo com efeito Peltier) são utilizados para comprovar sua eficiência para ambas as situações. A arquitetura fica flexível e poderão ser agregados novos processos a critério do pesquisador, estudante e quem desejar utilizar a ferramenta para analisar e atuar em qualquer outro tipo de processo, lógico que adaptado às limitações que serão expostas no decorrer deste trabalho.

Capítulo 6

Estudo de caso

Neste capítulo, são apresentados três estudos com a ferramenta desenvolvida para os três processos apresentados ao longo do texto. São colocados ainda no capítulo, os procedimentos necessários para realização de cada experimento, os resultados obtidos, comparações com resultados conseguidos com outros *software* e visualizações do estado dos processos nos simuladores. Além disso, são colocados alguns pontos importantes justificando o comportamento das variáveis do sistema para as situações que foram realizados os experimentos.

6.1 Simulação de CSTR

Para processo de CSTR, como o simulador não possui um componente gráfico para expor os resultados da simulação, foi verificada a resposta ao degrau do sistema. Uma pequena tela com os resultados obtidos nas equações de balanço de massa e energia discutidas no capítulo 5.

Inicialmente o simulador foi conectado ao servidor OPC para enviar os resultados da simulação referentes à concentração do componente de dentro do reator. A concentração, inicialmente 0 ($kmol/m^3$), deve ser alterada para 2 ($kmol/m^3$). Os resultados da simulação podem ser encontrados na Figura 6.2. O mesmo experimento foi realizado, agora observando a temperatura do componente no interior do reator. Conectado às variáveis SP, PV e MV do servidor OPC estava os resultados referentes à temperatura

que está demonstrada na tela do *software* representada na Figura 6.2. No experimento, a variação de temperatura se deu de 0 a 300K.

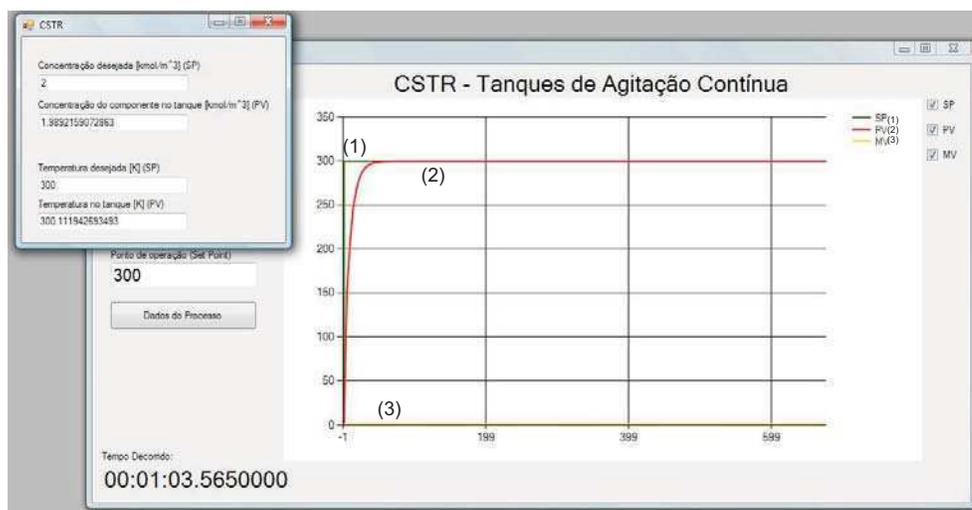


Figura 6.1: Representação da tela da ferramenta com os dados recebidos do simulador de CSTR (temperatura [K]).

O mesmo experimento foi realizado utilizando o *software* MatLab da Mathworks e as respostas do sistema podem ser visualizadas na Figura 6.3 com resultados semelhantes aos que foram obtidos com a ferramenta. O código MatLab para realização deste experimento pode ser encontrado no apêndice B.

O resultados deste experimento foram bem proveitosos. Verificou-se que a temperatura e concentração do componente no interior do reator atingiram o ponto de operação desejado e se aproximaram bem do resultado obtido com o experimento em MatLab. Estes valores de concentração e temperatura no interior CSTR podem ser obtidos a partir da manipulação destas variáveis na alimentação. As equações utilizadas no simulador foram aquelas demonstradas no Capítulo 5, bem como os valores das variáveis envolvidas no processo.

6.2 Simulação RBS

O simulador de RBS foi apresentado em (BARBOSA, 2009) e demonstrou sua eficácia na simulação para diferentes configurações de operação de RBS. Três casos foram apre-

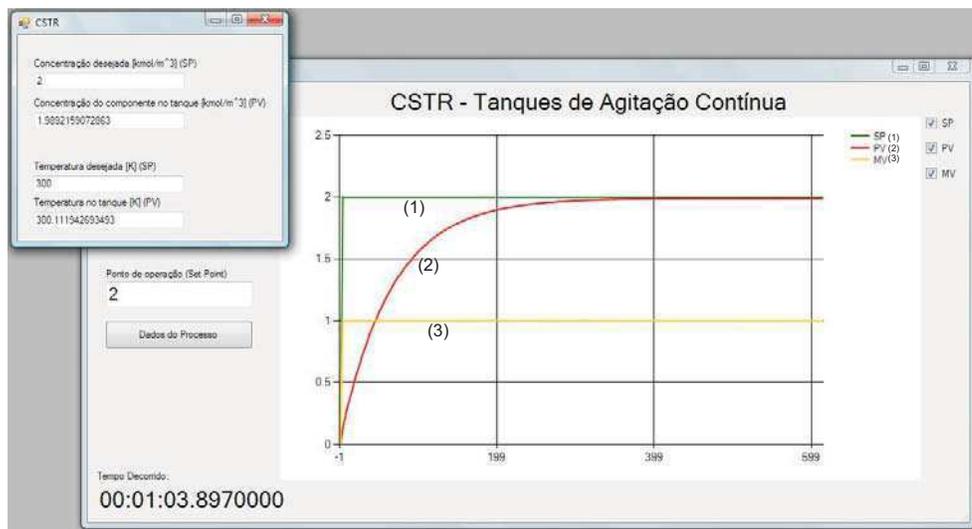


Figura 6.2: Representação da tela da ferramenta com os dados recebidos do simulador de CSTR (concentração do componente no interior do tanque $[kmol/m^3]$).

sentados:

- **Configuração 1:** três horas de Aeração e uma hora de Sedimentação;
- **Configuração 2:** uma hora de Hidrólise, seguido de duas horas de Aeração e uma de Sedimentação;
- **Configuração 3:** duas horas de Aeração, seguido de uma hora de Hidrólise e uma de Sedimentação.

O estudo de caso no presente trabalho é realizar efetuar o controle remotamente com a ferramenta desenvolvida.

Como mencionado no capítulo 5, a etapa de Aeração tem uma significativa atividade bacteriana e o consumo de oxigênio do sistema é bastante elevado. A idéia de controlar a concentração de oxigênio dissolvido nesta etapa, pode levar as bactérias envolvidas no sistema a ter um rendimento máximo, ou seja, consumir toda a matéria orgânica sem desperdiçar o oxigênio injetado. Os custos operacionais poderão ser reduzidos já que o projeto poderá ser direcionado para dimensionar melhor os sistemas de Aeração de acordo com a estratégia de controle adotada.

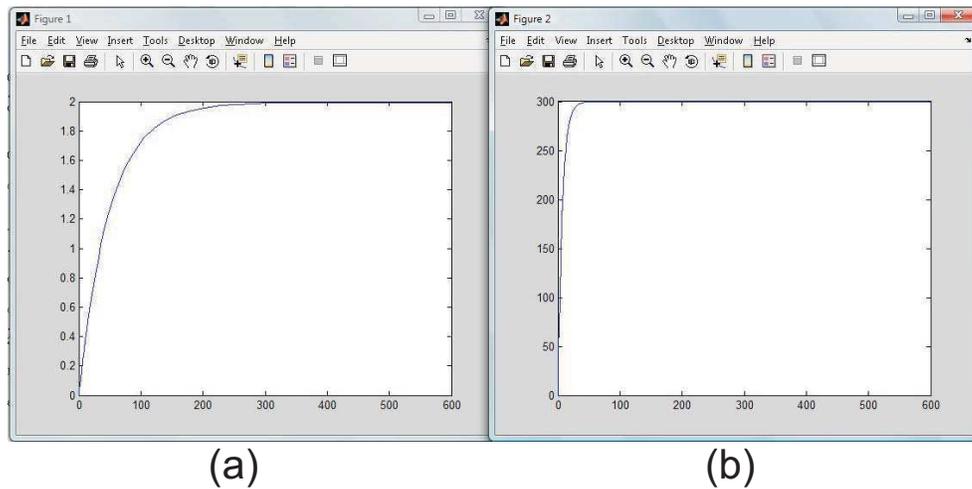


Figura 6.3: Resultados obtidos a partir da simulação de CSTR com o MatLab (a) concentração (b) temperatura

No experimento com a ferramenta de Laboratório Virtual pretende-se atuar no simulador remotamente com a **Configuração 1**. Para visualização, foi conectado às variáveis OPC as variáveis referentes ao oxigênio dissolvido pelo sistema. O simulador foi configurado para a atuação remota aconteça somente na etapa de Aeração, após esta, o controle era desligado e somente poderia ser visualizado na tela os resultados das outras etapas.

A técnica de sintonia do controlador remoto utilizada foi o método clássico de sintonia de Ziegler-Nichols, para conseguir valores que conseguisse a estabilidade do sistema e, depois, os parâmetros foram configurados de forma empírica para obtenção de um melhor resultado.

Utilizando a técnica de controle PID na etapa de aeração para controlar o oxigênio dissolvido pelo sistema, os resultados expostos na tabela 6.2 foram obtidos. Comparando com os resultados do simulador para remoção de poluentes do RBS sem controle da tabela 6.1, verificou-se que o sistema efetuou uma boa remoção de matéria orgânica e material nitrogenado. O processo de desnitrificação, que é o responsável pela conversão do nitrato em nitrogênio molecular (S_{NO}), é incompleto, daí o porquê de um resultado não satisfatório neste item. Como também não há nenhum outro tipo de ação biológica na matéria solúvel inerte (S_i), a mesma deixa o sistema com concentração

Componente	Afluente	Configuração 1 A-S
S_i (mgDQO/l)	59.4	38.51
S_s (mgDQO/l)	70.2	4.86
X_i (mgDQO/l)	75.6	2.79
X_s (mgDQO/l)	334.8	0.38
X_p (mgDQO/l)	0	1.93
S_{NO} (mgN/l)	0.5	21.43
S_{NH} (mgN/l)	40	0
S_{ND} (mgN/l)	5	1.08
X_{ND} (mgN/l)	5	0.02

Tabela 6.1: Resultados obtidos com a configuração 1 do simulador de RBS para remoção de poluentes

Componente	Afluente	Configuração 1 A-S
S_i (mgDQO/l)	59.4	38.51
S_s (mgDQO/l)	70.2	5.02
X_i (mgDQO/l)	75.6	2.79
X_s (mgDQO/l)	334.8	0.46
X_p (mgDQO/l)	0	1.93
S_{NO} (mgN/l)	0.5	19.98
S_{NH} (mgN/l)	40	0
S_{ND} (mgN/l)	5	1.03
X_{ND} (mgN/l)	5	0.02

Tabela 6.2: Resultados obtidos com a configuração 1 do simulador de RBS para remoção de poluentes utilizando o controlador PID aplicado remotamente com a ferramenta desenvolvida

ainda muito alta, sofrendo apenas uma diluição/mistura com materiais dentro do reator. Os demais componentes expostos nas tabelas 6.2 e 6.1 referem-se ao substrato rapidamente biodegradável (S_s), matéria orgânica particulada inerte (X_i), substrato lentamente biodegradável (X_s), produtos particulados resultantes do decaimento da biomassa (X_p), amônia (S_{NH}), nitrogênio orgânico solúvel biodegradável (S_{ND}) e nitrogênio orgânico particulado biodegradável (X_{ND}).

Analisando a tabela 6.2, na qual foi utilizada uma estratégia de controle PID na etapa de aeração, e comparando os valores com os da tabela 6.1, verifica-se valores finais muito próximos, o que significa que o controle não melhorou o sistema no que

diz respeito à remoção de matéria orgânica e material nitrogenado, mas também não piorou. A modelagem apresentada em (BARBOSA, 2009) para sistema RBS já apresenta resultados interessantes e a mesma foi implementada no simulador. Os parâmetros do controlador podem ser melhorados de acordo com a estratégia de controle implementada e também do modo de operação da planta.

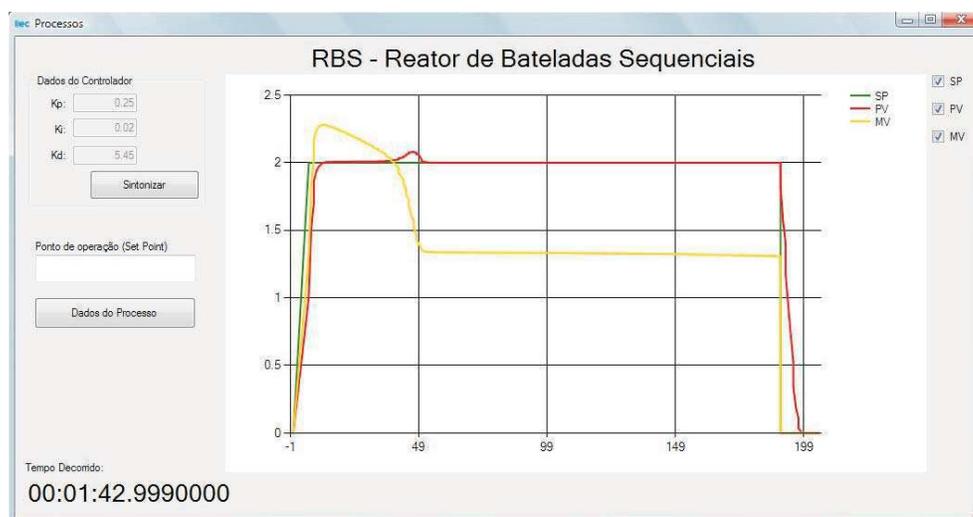


Figura 6.4: Representação da tela da ferramenta com os dados recebidos do simulador RBS com a estratégia de controle implementada.

As Figuras 6.4 e 6.5 representam graficamente os resultados obtidos com e sem a estratégia de controle implementada, respectivamente, a partir do simulador sendo executado remotamente, na tela do Laboratório Virtual. Já as Figuras 6.6 e 6.7 representam as telas visualizadas no simulador para o oxigênio dissolvido com e sem controle PID, respectivamente.

Pode-se observar nos gráficos gerados na ferramenta, que o oxigênio dissolvido no sistema estabilizou-se nos $2 \text{ mg}(-DQO)/l$ com a estratégia de controle utilizada. Isto significa que as bactérias estão mais eficientes no consumo de oxigênio em toda a etapa de aeração, diferentemente da visualização do processo sem o controle PID, onde o gráfico sobe bruscamente próximo dos 50 minutos de simulação. Isto significa que a partir dali, as bactérias do sistema já não conseguem mais consumir todo o oxigênio injetado no sistema, acarretando no desperdício de oxigênio que é liberado para o ambiente. A curva satura próximo do valor de $7 \text{ mg}(-DQO)/l$.

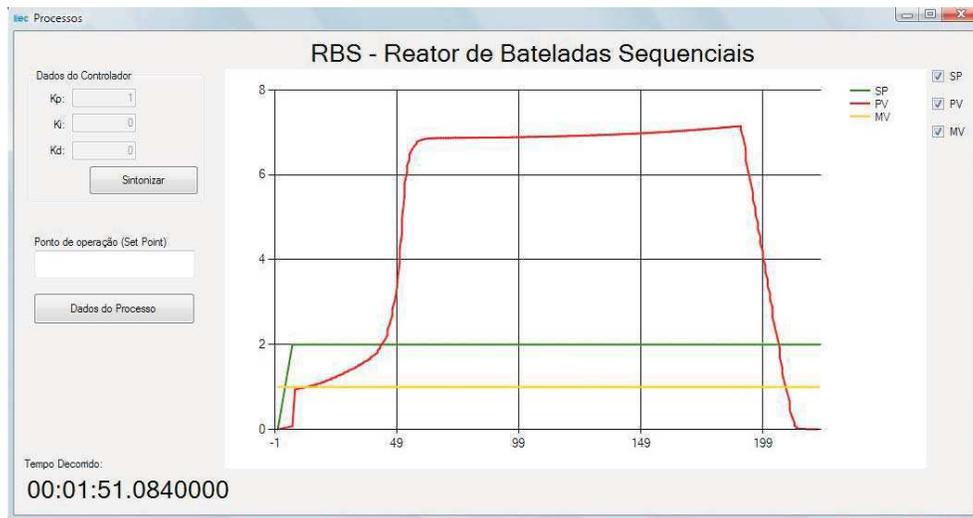


Figura 6.5: Representação da tela da ferramenta com os dados recebidos do simulador RBS sem estratégia de controle.

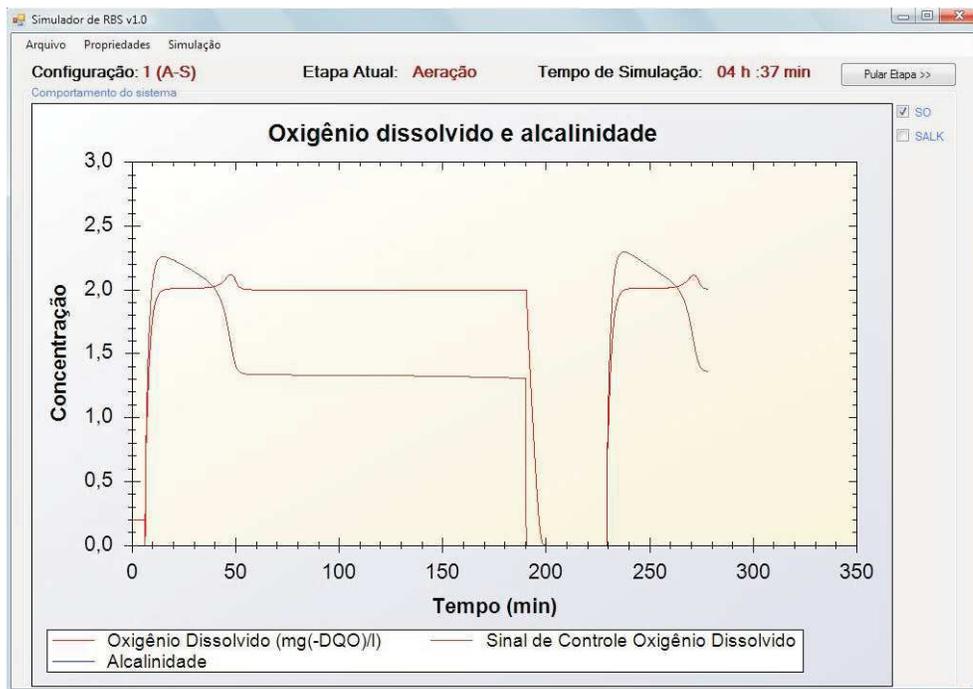


Figura 6.6: Representação da tela do simulador RBS com a estratégia de controle implementada.

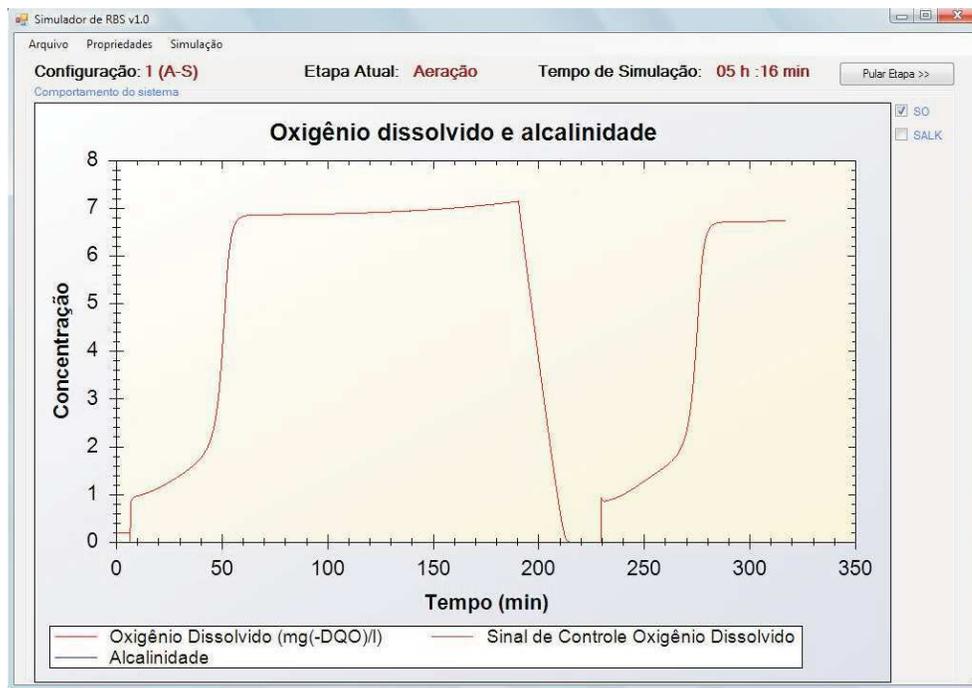


Figura 6.7: Representação da tela do simulador RBS sem estratégia de controle.

Como já mencionado, para análise final dos resultados da simulação, a etapa de aeração tem grande impacto nos custos operacionais devido ao consumo excessivo de energia por parte dos aeradores. A simulação e atuação no sistema com o controle PID demonstra bons resultados neste sentido, pois o oxigênio injetado garantiu que a matéria orgânica foi bem consumida e boa parte do nitrogênio foi convertida em amônia e, na sequência, em nitrogênio molecular, resultando em baixas concentrações de nitrogênio no efluente o que significa que o oxigênio injetado deverá ser melhor aproveitado pelas bactérias do sistema, não havendo mais quantidades excessivas liberadas para o meio ambiente, significando menos energia consumida.

6.3 Módulo Peltier

No capítulo 5 foi apresentada uma modelagem para o módulo Peltier do processo disponibilizado neste trabalho. No processo, há um sensor de temperatura do tipo LM35 e um microcontrolador do tipo Aduc812 programado para enviar e receber os dados de tem-

peratura fornecidos pelo LM35 e enviar, via interface RS-232, as informações lidas. Os comandos de envio e leitura de dados do microcontrolador referentes à temperatura no LM35 são aqueles demonstrados no capítulo 5.

O *plug-in* do serviço OPC do DANSrv OPC DA .NET descrito no capítulo 4 foi configurado para ler os dados recebidos na porta serial e enviá-los para a variável SP do serviço OPC. A variável MV recebe o valor recebido diretamente do componente de controle PID da ferramenta de Laboratório Virtual e envia para o processo por meio do método *Write()*. O serviço, por sua vez, recebe esse valor MV e envia para a porta serial.

A Figura 6.8 representa a saída do processo para a mudança de *Set Point* quando é utilizado o controle PID sintonizado utilizando a técnica de sintonia de Ziegler-Nichols (ZIEGLER; NICHOLS, 1942).

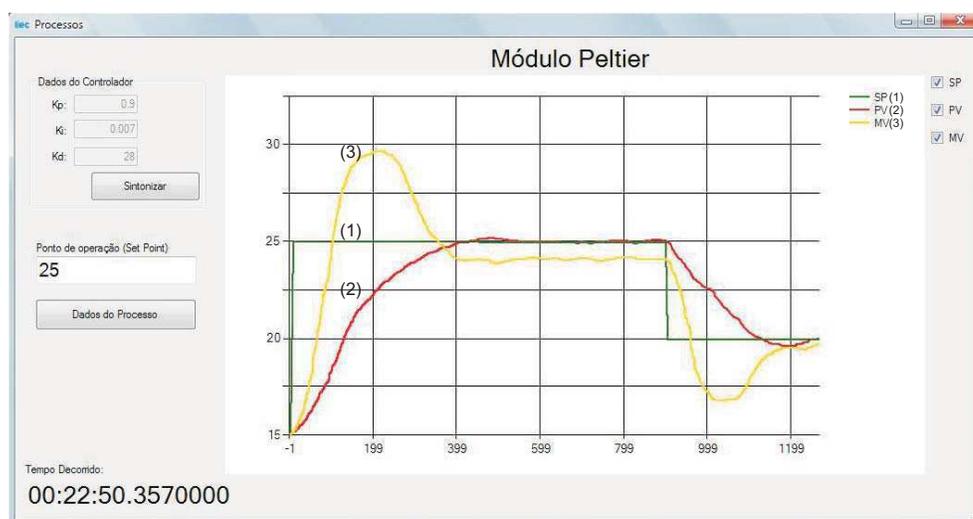


Figura 6.8: Resultados obtidos com o Módulo Peltier

A resposta deste sistema é muito sensível a perturbações externas e depende de alguns fatores tais como temperatura ambiente e posição do sensor LM35 na coleta da informação de temperatura para obtenção de resultados coerentes. A utilização de uma pasta térmica envolvendo o sensor também foi necessária na tentativa de minimização dos fatores externos de temperatura.

O processo também está limitado para algumas situações devido às condições das

placas e demais materiais envolvidos. No momento dos experimentos para este trabalho, a temperatura só podia variar entre 20 e 30 graus Celsius para obtenção de resultados relativamente precisos. Ocorreu saturação na saída do processo quando essas condições não foram respeitadas.

6.4 Conclusão

A ferramenta foi utilizada para as três situações disponibilizadas com resultados proveitosos.

Para os simuladores, foram realizados experimentos de mudança de *Set Point* com o CSTR e comparados com os resultados obtidos em simulação idêntica com o MatLab. Com o simulador de RBS foi realizado um experimento com implementação de técnica de controle PID e pode ser utilizado por projetistas, estudantes e operadores para simular situações adversas do processo e estudar técnicas para redução de custos operacionais, além de novas estratégias de controle.

Por fim, os resultados de um processo real foi apresentado no capítulo, demonstrando a funcionalidade da ferramenta não só para simuladores, mas para processos reais.

Vale salientar que para o processo real, foi necessário o reajuste do serviço OPC, sendo o *plug-in* de comunicação incrementado com uma função de comunicação com a serial que envia os comandos necessários para obtenção da resposta do processo. Isso não é o padrão para a ferramenta, foi somente necessário devido às condições de comunicação do processo (somente via porta serial) e porque o servidor OPC está hospedado num computador. Para comunicação com qualquer outro processo, pode ser necessário uma configuração parecida no *plug-in*, ou que este processo envie diretamente as variáveis de resposta para as variáveis no serviço OPC, por meio de algum outro tipo de rede.

Capítulo 7

Considerações Finais

Procurou-se colocar neste trabalho a importância da educação com o auxílio da tecnologia que se caracterizam pela rapidez e abrangência no processo de transmissão da informação. Ferramentas para auxílio de pesquisadores, professores e para ajudar no treinamento de operadores no ramo industrial vêm sendo utilizadas cada vez mais nos dias atuais. No ambiente acadêmico, o aluno pode interagir com o ambiente virtual, inserindo seus valores e observando os resultados.

O tema central deste trabalho foi o desenvolvimento de uma ambiente de laboratório virtual com uma arquitetura baseada em serviços para permitir que alunos, pesquisadores e profissionais do meio industrial, atuem em processos industriais remotos, ligados a um servidor OPC que, por sua vez, se conecta a um servidor WCF para envio das informações pela rede. A ferramenta final ficou disponibilizada com três processos laboratoriais, sendo dois simuladores e um processo real. Estudos de caso durante o desenvolvimento da ferramenta foram feitos e apresentados neste trabalho como forma de comprovação da eficiência de aquisição de dados por meio dos serviços.

Como mostrado, a ferramenta com os processos e simuladores já disponibilizados, podem ser utilizados para treinamentos e simulações do comportamento dos sistemas envolvidos para diversas situações. Técnicas de sintonia de controlador PID podem ser estudadas, utilizadas e aperfeiçoadas.

No trabalho, foi colocado ainda um pouco das principais tecnologias envolvidas no desenvolvimento da ferramenta, sua história e o porque da utilização no presente

trabalho.

Foi feita uma modelagem dos processos envolvidos e disponibilizados na ferramenta como forma de manter documentado e facilitar o entendimento do utilizador da ferramenta, bem como foi necessário para o desenvolvimento dos simuladores. O comportamento das variáveis dos processos podiam ser esperados com o conhecimento aprofundado das equações que envolviam os fenômenos que ocorriam.

7.1 Trabalhos Futuros

Conforme mencionado no capítulo 4, a ferramenta fica disponível inicialmente para os três processos discutidos neste trabalho, porém, há a possibilidade de sua extensão para até dez processos sem a necessidade do usuário ter conhecimento da linguagem e ferramenta de desenvolvimento utilizada neste projeto. Para poder atuar um

Alguns trabalhos futuros podem ser destacados. A implementação de mais simuladores para outros tipos de processos poderá fazer com que a ferramenta seja utilizada numa vasta área voltada para pesquisa e treinamento de operadores. Sistemas multivariáveis, trocadores de calor, sistemas com tanques em cascata são apenas alguns exemplos dentre os vários processos que podem ser simulados.

Não necessariamente somente aplicativos com a linguagem de programação utilizada no desenvolvimento dos simuladores deste trabalho, mas sim todo e qualquer dispositivo ou aplicativo que possa ter conexão com OPC. Aplicativos do MatLab, IN-TOUCH, entre outros, já possibilitam esta conexão e podem ser anexados ao Laboratório Virtual sem quaisquer problemas. Dessa forma, a arquitetura proposta no capítulo 4 poderia ser completada com um aplicativo Web utilizando a tecnologia do Silverlight para expor resultados na tela. A conexão com CLP (Controlador Lógico Programável) tanto simulando como obtendo dados de processos também poderiam incrementar o sistema.

Pode-se pensar ainda em algumas formas de aperfeiçoamento da ferramenta no que diz respeito ao banco de dados e funcionalidades do serviço WCF:

- Ajuste do banco de dados para o registro de informações mais detalhadas dos usuários e dos processos.

- Passagem dos parâmetros via serviço WCF que influenciam nos processos tais como volumes, concentrações, temperaturas, etc.
- Interface gráfica para visualizar etapas dos processos.

Com relação às teorias de controle, outros algoritmos que implemente as teorias de controle preditivo, adaptativo, inteligente entre outras, podem ser inseridas e utilizadas no componente de controle do projeto atual. Desta forma, o Laboratório Virtual proporcionará uma variedade muito grande de processos e a pesquisa e estudo com a ferramenta poderá ser rica na quantidade de informação repassada para os usuários.

Referências Bibliográficas

ASTROM, K. J.; HAGGLUND, T. *Advanced PID Control*. [S.l.]: ISA - The Instrumentation Systems and Automation Society, 2006.

BARBOSA, A. de O. *Modelagem, Simulação e Automação de Sistemas Biológicos de Tratamento de Esgotos*. Tese (Dissertação de Mestrado) — Universidade Federal de Campina Grande, Março 2009.

BARROS, P. R.; JÚNIOR, G. A.; SANTOS, J. B. M. dos. Two-input two-output laboratory-scale temperature system based on peltier modules. *IFAC World Congress*, Setembro 2007.

BARROSO, H. C. *Software para Avaliação de Confiabilidade de Sistemas Instrumentados de Segurança*. Tese (Dissertação de Mestrado) — Universidade Federal de Campina Grande, Março 2009.

BERGER, D. *COM: A Brief Introduction*. Disponível em: <<http://www.cs.ucr.edu/~dberger/Documents/Presentations/com-intro.ppt>>. Acesso em: 19 de Setembro de 2010.

BOX, D. et al. *Simple Object Access Protocol (SOAP) 1.1*. Maio 2008. Disponível em: <<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>>. Acesso em: 15 de Setembro de 2010.

CHAPPELL, D. *Introducing Windows Communication Foundation in .NET Framework 4*. Abril 2009. Disponível em: <[http://msdn.microsoft.com/en-us/library/ee958158\(v=MSDN.10\).aspx](http://msdn.microsoft.com/en-us/library/ee958158(v=MSDN.10).aspx)>. Acesso em: 19 de Outubro de 2010.

- CHEN, P. The entity-relationship model-toward a unified view of data. *ACM Transactions on Database Systems*, v. 1, n. 1, March 1976.
- CSharp-Online.NET. *.NET Languages*. Disponível em: <<http://en.csharp-online.net/>>. Acesso em: 15 de Setembro de 2010.
- DEITEL, H. M. et al. *C# Como Programar*. [S.l.]: Pearson Education, 2003.
- DORMIDO, R. et al. Development of a web-based laboratory for automation technicians: The three-tank system. *IEEE Transactions on Education*, 2007.
- e-Learning Brasil. *e-Learning ganha cadeira cativa nas empresas brasileiras*. 2005. Disponível em: <<http://www.elearningbrasil.com.br/imprensa/clipping.asp?id=1791>>. Acesso em: 19 de Outubro de 2010.
- GUIMARÃES, P. Modelagem e otimização de um sistema anaeróbio-aeróbio de tratamento de esgotos utilizando o reator uasb e o reator de bateladas sequenciais. tese de doutorado. Universidade Federal do Rio Grande do Norte. 2003.
- HENZE, M. et al. *Activated sludge model No. 1*. [S.l.]: IAWQ, 1987. London, England.
- IWANITZ, F.; LANGE, J. *OPC Fundamentals, Implementation, and Application*. 2 rev.. ed. [S.l.]: Huthig Verlag Heidelberg, 2002.
- JONASSEN, D. et al. Constructivism and computer-mediated communication in distance education. *Amer. J. Distance Educ.* 1995.
- JORDÃO, E. P.; PESSOA, A. C. *Tratamento de Esgotos Domésticos*. 3. ed. [S.l.]: ABES - Associação Brasileira de Engenharia Sanitária e Ambiental, 1995. Rio de Janeiro, RJ.
- KO, C. C. et al. Development of a web-based laboratory for control experiments on a coupled tank apparatus. *IEEE Transactions on Education*, v. 44, n. 1, Fevereiro 2001.
- LIRA, V. V. *Automação de um Sistema de Tratamento de Esgotos Baseado no Processo de Lodo Ativado*. Tese (Dissertação de Mestrado) — Universidade Federal da Paraíba, Julho 2001.

MATHEMATICA. Wolfram Research, Soluções em software matemático, técnico e científico. Disponível em: <<http://www.wolfram.com/>>. Acesso em: 19 de Outubro de 2010.

MATLAB/SIMULINK. The MathWorks, Inc., Natick, Massachusetts. Disponível em: <<http://www.mathworks.com/>>. Acesso em: 19 de Outubro de 2010.

MENEGHEL, L. *Desenvolvimento de Laboratórios Virtuais para o Ensino Fundamental e o Ensino Superior*. Tese (Dissertação de Mestrado) — Universidade Estadual de Campinas, Maio 2003.

MIGUEL, P. V. de O. *Laboratório Virtual Aplicado à Educação em Engenharia Elétrica*. Tese (Dissertação de Mestrado) — Universidade Estadual de Campinas, Outubro 2008.

MSDN. *Microsoft Developer Network*. Disponível em: <<http://msdn.microsoft.com/pt-br/default.aspx>>. Acesso em: 15 de Setembro de 2010.

MSDN-MSChart. *MSChart Control*. Disponível em: <[http://msdn.microsoft.com/en-us/library/3ks53324\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/3ks53324(VS.71).aspx)>. Acesso em: 19 de Setembro de 2010.

OPC Training Institute. *OPC Specifications Define the Type of Data Transfer*. 2010. Disponível em: <<http://www.opcti.com/OPC-Specifications.aspx>>. Acesso em: 18 de Outubro de 2010.

PEIRIS, C.; MULDER, D. *Pro WCF - Practical Microsoft SOA Implementation*. [S.l.]: apress, 2007.

TAVANGARIAN, D. et al. Is e-learning the solution for individual learning? *Journal of e-learning*, 2004.

The OPC Foundation. *Dedicated to Interoperability in Automation and Beyond*. 2010. Disponível em: <<http://www.opcfoundation.org/>>. Acesso em: 18 de Outubro de 2010.

ZIEGLER, J. G.; NICHOLS, N. B. Optimum settings for automatic controllers. *Transactions of the ASME*, p. 759–768, 1942.

Apêndice A

Especificações OPC

OPC é composto por várias especificações individuais. Cada uma dessas especificações define o tipo de dados para transferência entre as aplicações. Por exemplo, a especificação do OPC DA (Data Access), que é, atualmente, a especificação mais utilizada do OPC e define como as aplicações devem transferir dados em tempo real. Abaixo estão cada uma das especificações OPC (OPC Training Institute, 2010):

- **OPC Xi (Express Interface)** - É a mais nova especificação OPC e é baseada em WCF para uma comunicação rápida em intranet e segura. Modelo bastante simples, Xi fornece um conjunto integrado de métodos para acessar tanto em tempo de execução como dados históricos, eventos e alarmes. A interface é projetada para permitir um mapeamento simples para o clássico OPC DA, HDA e interfaces AE.
- **OPC AE (Alarms and Events)** - É utilizado para o intercâmbio de eventos e alarmes dos processos. Pode ser utilizado para obter sequência de eventos, notificar alarmes, realização de auditorias estratégicas de alarmes e otimizar seu funcionamento. A OPC Foundation é responsável pela definição e manutenção da especificação OPC AE.
- **OPC ADI (Analyzer Devices Integration)** - É um modelo de informação para permitir a interoperabilidade entre os dispositivos de vários fornecedores.

- **OPC Batch** - Especificação utilizada quando há a necessidade de verificação de um lote de processos. Oferece interfaces para a troca de capacidade de equipamentos nas atuais condições operacionais.
- **OPC Commands** - Define um conjunto de interfaces que permitem aos clientes e servidores OPC a identificar, enviar e acompanhar os comandos de controle que são executados em um dispositivo.
- **OPC Complex Data** - Especificação para OPC DA e OPC XML-DA que habilita os servidores a expor e descrever tipos de dados mais complexos, como estruturas binárias e documentos XML.
- **OPC DA (Data Access)** - Provê o acesso a dados de automação em tempo real. Com OPC DA, aplicações de *software* podem recuperar dados em tempo real para permitir acompanhar um determinado processo. Pode ser utilizado ainda para ativar um aplicativo para gravar dados em um sistema de controle, desta forma, OPC DA pode ser utilizado para lidar com supervisório. OPC DA é utilizado por quase todas aplicações de controle de processos como otimizadores, *software* de análise, manutenção, ERP (Enterprise Resource Planning), etc.
- **OPC DX (Data eXchange)** - Especificação que define como servidores OPC trocam dados com outros servidores OPC. Esta tecnologia não é muito utilizada pelos fabricantes, utilizam, normalmente, o OPC bridging para transferir dados entre servidores OPC.
- **OPC HDA (Historical Data Access)** - Utilizado para recuperar e analisar dados dos processos com a proposta de análise, otimização, controle de estoque, etc.
- **OPC Security** - Habilita aplicações OPC a aplicar as especificações de segurança que é maior que as disponíveis no DCOM. O DCOM permite o administrador do servidor habilitar o acesso de usuários a determinados itens e negar a outros. Isso permite a proteção de informações importantes e evita alterações não autorizadas dos parâmetros do sistema.

- **OPC UA (Unified Architecture)** - É uma das mais recentes especificações OPC que unifica todas as especificações já citadas. OPC UA provê compatibilidade com o OPC clássico (baseados em DCOM). Permite ainda os desenvolvedores a selecionar vários métodos de transporte de dados baseados em protocolos como HTTP e TCP. As mensagens podem ser codificadas por SOAP o que lhes permitem fluir pela internet entre aplicativos que não sejam da Microsoft.
- **OPC XML Data Access** - Encapsula os dados de controle de processos tornando-os disponíveis em todos os sistemas operacionais por meio de XMLs. Essa especificação está sendo abandonada em favor de OPC DA e OPC UA.

Apêndice B

Simulador CSTR no MatLab

Arquivo cstr.m

```
% CSTR model from
```

```
% Michael A. Henson and Dale E. Seborg. Nonlinear Process Control.  
% Prentice Hall PTR, Upper Saddle River, New Jersey, 1997.
```

```
% Description: % Continuously Stirred Tank Reactor with energy balance  
and reaction  $A \rightarrow B$ .
```

```
% The temperature of the cooling jacket is the control.
```

```
function  $\dot{x}$  = cstr1(t,x)
```

```
global u
```

```
% Input (1):
```

```
% Temperature of cooling jacket ( $K$ )
```

```
 $T_c = u;$ 
```

```
% States (2):
% Concentration of A in CSTR ( $mol/m^3$ )
Ca = x(1,1);
% Temperature in CSTR (K)
T = x(2,1);

% Parameters:
% Volumetric Flowrate ( $m^3/sec$ )
q = 0.002;
% Volume of CSTR ( $m^3$ )
V = 0.1;
% Density of A-B Mixture ( $kg/m^3$ )
rho = 1000;
% Heat capacity of A-B Mixture ( $J/kg-K$ )
Cp = 4;
% Heat of reaction for A->B ( $J/mol$ )
mdelH = -5e5;
% E - Activation energy in the Arrhenius Equation ( $J/mol$ )
% R - Universal Gas Constant = 8.31451  $J/mol-K$ 
EoverR = 8750;
% Pre-exponential factor (1/sec)
k0 = 3e8;
% U - Overall Heat Transfer Coefficient ( $W/m^2-K$ )
% A - Area - this value is specific for the U calculation ( $m^2$ )
UA = 50;
% Feed Concentration ( $mol/m^3$ )
Caf = 2;
% Feed Temperature (K)
Tf = 300;

% Compute xdot:
```

```

$$\dot{x}(1,1) = (q/V * (Caf - Ca) - k0 * \exp(-EoverR/T) * Ca);$$

$$\dot{x}(2,1) = (q/V * (Tf - T) + mdelH / (rho * Cp) * k0 * \exp(-EoverR/T) * Ca + UA/V / rho / Cp * (Tc - T));$$

```

Arquivo step.m

```
% Step test for Model 1 - CSTR
```

```
% Created by John Hedengren
```

```
global u
```

```
% Steady State Initial Conditions for the States
```

```
 $C_{a,s} = 0;$ 
```

```
 $T_s = 0;$ 
```

```
 $x_s = [C_{a,s}; T_s];$ 
```

```
% Steady State Initial Condition for the Control
```

```
 $u_s = 300;$ 
```

```
% Open Loop Step Change
```

```
 $u = 300;$ 
```

```
% Final Time (sec)
```

```
 $tf = 600;$ 
```

```
 $[t, x] = \text{ode45}('cstr1', [0, tf], x_s);$ 
```

```
% Parse out the state values
```

```
 $Ca = x(:, 1);$ 
```

```
 $T = x(:, 2);$ 
```

```
% Plot the results
```

```
figure
```

```
plot(t,Ca);
```

```
figure
```

```
plot(t,T);
```