

UNIVERSIDADE ESTADUAL PAULISTA
JÚLIO DE MESQUITA FILHO
Câmpus de Presidente Prudente

MATHEUS ALEXANDRE FERNANDES POLETTO

WSPControl: aplicação distribuída baseada em Web Services para monitoramento remoto de computadores

2010

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de Presidente Prudente

MATHEUS ALEXANDRE FERNANDES POLETTO

WSPControl: aplicação distribuída baseada em Web Services para monitoramento remoto de computadores

Monografia para Conclusão de Curso

Orientador: Prof. Dr. Ronaldo Celso Messias

2010

MATHEUS ALEXANDRE FERNANDES POLETTO

WSPControl: aplicação distribuída baseada em Web Services para monitoramento remoto de computadores

Monografia apresentada na Faculdade de Ciências e Tecnologia da Universidade Estadual Paulista JÚLIO DE MESQUITA FILHO - UNESP - Câmpus de Presidente Prudente, como requisito obrigatório para a conclusão do Curso de Bacharelado em Ciência da Computação.

Orientador: Prof. Dr. Ronaldo Celso Messias

Presidente Prudente

2010

Termo de Aprovação

MATHEUS ALEXANDRE FERNANDES POLETTO

WSPControl: aplicação distribuída baseada em Web Services para monitoramento remoto de computadores

Monografia sob o título **WSPControl: aplicação distribuída baseada em Web Services para monitoramento remoto de computadores**, defendida por MATHEUS ALEXANDRE FERNANDES POLETTO e aprovada em 14 de Dezembro de 2010, em Presidente Prudente, Estado de São Paulo, pela banca examinadora constituída pelos doutores:

Prof. Dr. Ronaldo Celso Messias
Universidade Estadual Paulista

Prof. Dr. Rogério Eduardo Garcia
Universidade Estadual Paulista

Prof. Dr. Celso Olivete Júnior

Presidente Prudente, 14 de Dezembro de 2010

Agradecimentos

Agradeço principalmente à Deus que meu deu a força necessária para superar todos os obstáculos que encontrei durante os anos de faculdade.

A minha família a qual sempre me ajudou a vencer qualquer obstáculo da vida e sempre manteve a paciência. E ao orientador Prof. Dr. Ronaldo Celso Messias Correia, pela paciência e confiança que depositou em mim durante toda a elaboração deste trabalho.

Resumo

A ferramenta proposta neste trabalho, denominada como WSPControl, possibilita o monitoramento remoto de computadores através da internet utilizando aplicações distribuídas. Através de uma arquitetura de Web Services é possível a comunicação entre estas alpicações distribuídas em plataformas heretogêneas, também elimina a necessidade de configurações adicionais em redes de computadores, como liberação de portas ou proxy. A ferramenta é dividia em três módulos, sendo eles:

- **Interface de Cliente:** desenvolvido em C Sharp, é responsável por captar os dados em execução do computador monitorado, também conecta-se com o Web Services para informar estes dados.
- **Interface de Web Services:** desenvolvido em PHP utilizando a biblioteca PHP Soap, é responsável por intermediar a comunicação entre as aplicações de internet e cliente.
- **Interface de Internet:** desenvolvido em PHP, é responsável pela leitura e interpretação das informações capturadas disponibilizando estas na Internet.

palavras-chave: Web Services, monitoramento remoto, aplicação distribuída.

Abstract

The tool proposed, known as WSPControl, enables remote monitoring of computers across the Internet using distributed applications. Through a Web Services architecture is possible the communication between these distributed applications across heterogeneous platforms, also eliminates the need for additional settings in computer networks, such as release of ports or proxy. The tool is divided into three modules, namely:

- **Client Interface:** developed in C Sharp, is responsible for capturing data on performance of the monitored computer also connects to the Web Services to report this data.
- **Web Services Interface:** developed in PHP using the PHP SOAP library, is responsible for facilitating the communication between internet applications and client.
- **Internet Interface:** developed in PHP, is responsible for reading and interpreting the information captured these available on the Internet.

keywords: Web Services, remote monitoring, distributed applications.

Sumário

1 Introdução p. 10

1.1 Contextualização . . . . . p. 10

1.2 Objetivos . . . . . p. 11

1.3 Organização do trabalho . . . . . p. 11

2 Revisão Bibliográfica p. 12

2.1 Web Services . . . . . p. 12

2.1.1 Definição e Padronização . . . . . p. 13

2.1.2 Ciclo de Vida . . . . . p. 14

2.1.3 Camadas . . . . . p. 14

2.1.4 *Universal Description, Discovery and Integration* (UDDI) . . . . p. 15

2.1.5 Web Services Description Language (WSDL) . . . . . p. 16

2.2 *Extensible Markup Language* (XML) . . . . . p. 17

2.2.1 Estrutura da linguagem . . . . . p. 18

2.3 *Simple Object Access Protocol* (SOAP) . . . . . p. 19

2.4 Monitoramento remoto de computadores . . . . . p. 20

2.4.1 Dificuldades para o monitoramento remoto . . . . . p. 21

2.5 Tecnologias empregadas para o desenvolvimento da ferramenta WSP-Control . . . . . p. 22

2.5.1 PHP 5 . . . . . p. 22
2.5.2 PHP SOAP . . . . . p. 23
2.5.3 C Sharp . . . . . p. 23
2.5.4 C Sharp - System.Diagnostics . . . . . p. 24
2.5.5 Tecnologias auxiliares . . . . . p. 24
2.5.6 Considerações sobre as tecnologias . . . . . p. 25
2.6 Trabalhos correlatos . . . . . p. 25

3 A ferramenta WSPControl p. 28
3.1 Visão Geral . . . . . p. 29
3.2 Contextualização da Arquitetura . . . . . p. 30
3.3 Interface de Cliente . . . . . p. 31
3.4 Interface de Web Services . . . . . p. 34
3.5 Interface de Internet . . . . . p. 36
3.5.1 Segurança . . . . . p. 37

4 Estudo de Caso p. 38

5 Conclusão p. 46
5.1 Trabalhos Futuros . . . . . p. 46

Lista de Figuras

1 Exemplo de invocação de um Web Services . . . . . p. 13

2 Ciclo de vida de um Web Services . . . . . p. 14

3 Pilha de serviços de Web Services . . . . . p. 15

4 Elementos de um documento WSDL . . . . . p. 17

5 Exemplo de um documento XML . . . . . p. 19

6 Estrutura da mensagem SOAP . . . . . p. 20

7 Comunicação entre as aplicações cliente, administrador e Web Services p. 29

8 Estrutura do banco de dados . . . . . p. 30

9 Arquitetura das interfaces do projeto . . . . . p. 31

10 Contextualização da arquitetura de redes de computadores . . . . . p. 32

11 Execução do Monitor de Processos em *system-tray* . . . . . p. 32

12 Exemplo da sintaxe de comunicação, formato XML . . . . . p. 33

13 Trecho do código fonte para utilização de Proxy . . . . . p. 34

14 Esquema de invocação do Web Services . . . . . p. 35

15 Trecho do código para alterar o status de um computador monitorado . p. 36

16 Estrutura do ambiente do estudo de caso . . . . . p. 39

17 Interface de Internet: Cadastro de administrador . . . . . p. 40

18 Interface de Internet: Cadastro de computadores . . . . . p. 40

19 Interface de Internet: Computadores cadastrados . . . . . p. 41

20 Chave de identificação de um computador . . . . . p. 41
21 Interface de Cliente: Área de configuração . . . . . p. 42
22 Interface de Cliente: Histórico de atividades em execução . . . . . p. 43
23 Interface de Internet: Visualizar mais informações de um computador . p. 44
24 Histórico de atividades de um computador . . . . . p. 44
25 Monitoramento do status de um computador . . . . . p. 45
26 Alteração do status de um computador monitorado . . . . . p. 45

Capítulo 1

Introdução

Neste capítulo será apresentado uma breve introdução sobre o trabalho desenvolvido, assim como demonstrar seus objetivos e sua organização estrutural.

1.1 Contextualização

Com a expansão demográfica de empresas, indústrias, equipes de trabalhos, entre outros, há a necessidade de aplicações que se beneficiam do acesso remoto a equipamentos localizados em lugares geograficamente distintos.

Podemos mencionar também, a enorme facilidade de obter conteúdo impróprio por crianças e adolescentes através da Internet. Sendo assim, é importante que pais instruam e também monitorem seus filhos na utilização de computadores.

Com a rápida evolução da internet, a qual atinge mais de 150 países e reunindo cerca de 300 milhões de computadores (DIZARD, 2000, p. 24), consequentemente a grande variedade de provedores de acesso, há um aumento significativo no intervalo de IPs (Protocolo de Internet) e DNS (Sistema de Nomes de Domínios) em âmbito mundial.

A arquitetura de componentes de Web Services tem mudado o modo como a internet é vista e utilizada, segundo Hansen e Pinto (2003) os Web Services permitem a comunicação entre aplicações com linguagens heterogêneas, fornecendo funções e serviços e comunicando-se através de uma linguagem padrão. Desta forma, estas funções podem ser utilizadas livremente por outros programas através de chamadas remotas de procedimentos.

Visando o monitoramento remoto da utilização de computadores e incorporando a

evolução da internet, faz o proveito, suprindo a necessidade, o desenvolvimento de uma aplicação de monitoramento remoto de computadores utilizando-se estes meios.

A incorporação da arquitetura de Web Services, visa a comunicação entre aplicações distribuídas em plataformas heterogêneas, para que haja a objetividade e praticidade no monitoramento remoto de computadores conectados a Internet, sem que necessite de uma configuração adicional de DNS, *Firewall*, NAT (Tradução de Endereço de Rede) ou preocupações com bloqueios por Proxy ou de portas.

1.2 Objetivos

A aplicação idealizada tem por objetivo monitorar computadores, independente da sua posição geográfica ou estrutura de rede, listando as atividades em execução, permitir bloquear e desligar estes computadores e possibilitar a leitura destas informações de qualquer lugar conectado à internet.

1.3 Organização do trabalho

O capítulo seguinte apresenta toda a fundamentação teórica estudada para o desenvolvimento das ferramentas necessárias para o monitoramento e leitura dos dados através da internet.

Já no capítulo 3 é apresentado o desenvolvimento da ferramenta WSPControl, o qual foi dividido em três etapas: Interface de Cliente, Interface de Web Services e Interface de Internet.

Para validar o estudo proposto é apresentado no capítulo 4 um estudo de caso.

Por último, no capítulo 5 é apresentada a conclusão do projeto, também é apresentada ideias para trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

Neste capítulo é apresentada toda a fundamentação teórica pesquisada para o desenvolvimento e a implementação deste trabalho.

2.1 Web Services

Os Web Services são padrões tecnológicos focado na utilização da internet para a integração de sistemas heterogêneos de forma flexível e independente da plataforma utilizada.

Segundo Hansen e Pinto (2003) os Web Services permitem a comunicação entre aplicações com linguagens heterogêneas, fornecendo funções e serviços e comunicando-se através de uma linguagem padrão, o XML (*Extensible Markup Language*).

De acordo com o Grupo de Trabalho de Arquitetura de Web Services do W3C (W3C *Web Services Architecture Working Group*): Um Web Services é um sistema de software desenvolvido para dar suporte a interação de máquinas através de uma rede. Possui uma descrição de interface passível de processamento escrita em WSDL. Outros sistemas podem interagir com o Web Services da maneira descrita pela sua interface usando mensagens SOAP, tipicamente transportadas usando HTTP e serialização XML, em conjunto com outros padrões relacionados a Web.

No ramo empresarial é comum que o ambiente de tecnologia seja bastante heterogêneo, com sistemas distintos para cada aplicação. Esta tecnologia possibilita a interação de novas aplicações com aplicações já existentes, independente da plataforma desenvolvida.

A essência de um Web Services é fazer com que os recursos da aplicação estejam dis-

poníveis sobre a internet de uma forma padronizada. Utilizando esta tecnologia, aplicações clientes independentes de plataformas ou linguagens de programação, poderão operar e extrair recursos fornecidos entre si. Esta comunicação é feita através da internet, utilizando o protocolo HTTP (*Hypertext Transfer Protocol*).

Exemplificando um Web Services seria como um repositório de dados que é acessado remotamente por computadores clientes, em localizações remotas, com sistemas operacionais possivelmente diferentes e arquiteturas distintas. Os clientes efetuam uma requisição de serviço, o servidor pode efetuar alguma computação sobre os dados e retornar um resultado. Na Figura 1 é possível visualizar o processo de invocação de um Web Services.

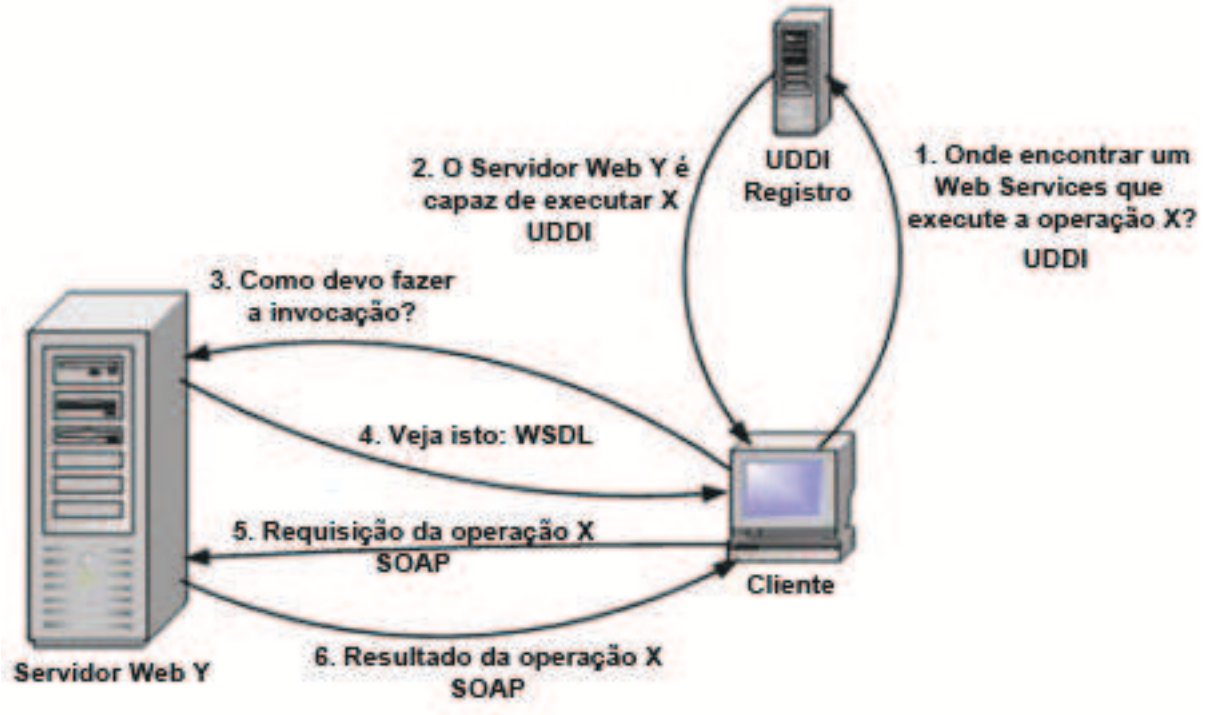


Figura 1: Exemplo de invocação de um Web Services

2.1.1 Definição e Padronização

O W3C (*World Wide Web Consortium*) é a instituição responsável pela padronização dos Web Services. Em setembro de 2000, foi criado um grupo de trabalho visando desenvolver uma arquitetura que permitisse a comunicação entre aplicações e sistemas heterogêneos. Neste grupo, haviam representantes das principais empresas software do mundo, tais como Microsoft, IBM, Oracle e Sun Microsystems, surgindo assim uma nova arquitetura computacional chamada de Web Services.

2.1.2 Ciclo de Vida

O ciclo de vida de um Web Services compreende quatro estados distintos: Publicação, Descoberta, Descrição e Invocação. Cada ciclo poderá ser visto na Figura 2:

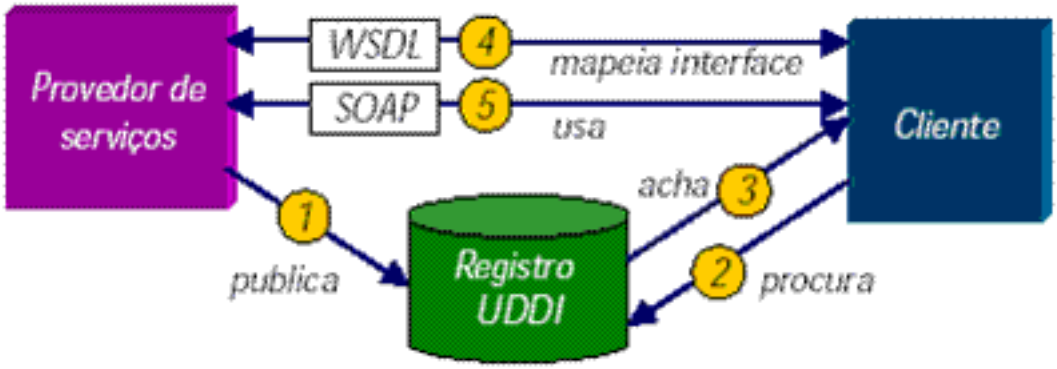


Figura 2: Ciclo de vida de um Web Services

- **Publicação:** é um processo opcional, no qual o fornecedor do Web Services poderá divulgar o seu serviço, efetuando o registro do mesmo no repositório de Web Services UDDI (*Universal Description Discovery and Integration*).
- **Descoberta:** é um processo opcional, no qual uma aplicação cliente toma conhecimento da existência de um Web Services pretendido, pesquisando no repositório de Web Services UDDI.
- **Descrição:** é o processo no qual o Web Services expõe a sua API (documento WSDL - *Web Services Description Language*). Desta maneira a aplicação cliente tem acesso a toda interface e descrição do Web Services, contendo as suas funcionalidades, assim como os tipos de mensagens que permitem processar as rotinas.
- **Invocação:** é o processo no qual o cliente e servidor interagem, através do envio de mensagens de entrada e/ou saída do Web Services.

2.1.3 Camadas

Os Web Services utilizam uma pilha de serviços dividida em camadas, sendo elas dívidas conforme a Figura 3:

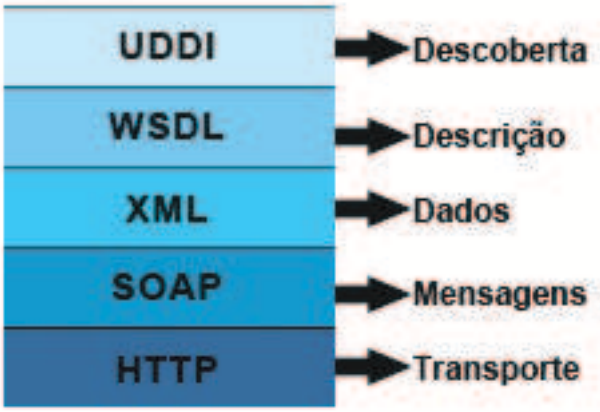


Figura 3: Pilha de serviços de Web Services

- ***Universal Description, Discovery and Integration* (UDDI):** atua na camada de descoberta, fornecendo a descrição do objetivo do Web Services - de que forma ele pode ser utilizado, proprietário, etc.
- ***Web Services Description Language* (WSDL):** é responsável por fornecer as informações (documento) que contém a interface do Web Services para que possa ser implementada. É situado na camada de descrição.
- ***Extensible Markup Language* (XML):** responsável pelo formato das mensagens, atua na camada de dados.
- **Simple Object Access Protocol (SOAP):** utilizado na camada de mensagens, é o protocolo padrão para os Web Services.
- ***Hypertext Transfer Protocol* (HTTP):** encapsula as mensagens SOAP em requisições do tipo POST. O HTTP está situado na camada de transporte.

2.1.4 *Universal Description, Discovery and Integration* (UDDI)

A UDDI é definida como uma maneira de publicar e descobrir informações relativas aos Web Services, com a finalidade de facilitar a sua localização, permitindo a busca e a publicação destes serviços em um banco de dados centralizado.

Os repositórios UDDI permitem a interação através de uma interface com os usuários oferecida pelos repositórios de dados ou então através de uma API - Interface de Programação de Aplicativos.

O funcionamento da UDDI contempla os seguintes passos: havendo uma necessidade de serviço, há o processo de localização (*Discovery*), que consiste em acessar os repositórios (*Registries*), o qual contém os documentos relativos a um Web Services.

A informação que faz parte da base de serviço é classificada dentro de três componentes:

- **Páginas Brancas:** contêm o endereço, pessoa de contato e outros identificadores relativos ao provedor de serviços.
- **Páginas Amarelas:** contêm listagens comerciais baseadas nos tipos de negócios, organizada por categoria específica ou regiões demográficas.
- **Páginas Verdes:** contêm informações específicas sobre serviços disponibilizados pelo provedor.

2.1.5 Web Services Description Language (WSDL)

O WSDL trata-se de um padrão de documento escrito no formato XML que define a interface de um Web Service. Neste documento contém os métodos expostos, nomes, tipos de dados, ordem dos parâmetros e os tipos de dados de retorno do Web Services.

Desta forma, o utilizador do Web Services pode implementar chamadas para os serviços disponível em XML, independente de linguagens ou ambientes de execução utilizados, uma vez que os WSDL são especificados em formatos comuns a fim de eliminar estes problemas. Da mesma forma, o WSDL especifica de que forma o serviço poderá ser acessado, por exemplo HTTP.

O documento WSDL possuí uma estrutura bem definida, sendo subdividido em três partes:

- **Definição de tipos de dados:** determinam a estrutura e o conteúdo das mensagens.
- **Operações:** determinam as operações disponíveis para o serviço.
- **Protocolo de Ligações:** determina a forma de transmissão das mensagens pela rede.

Segundo Mello et al (2006) os componentes do documento WSDL são definidos conforme a Figura 4, onde:

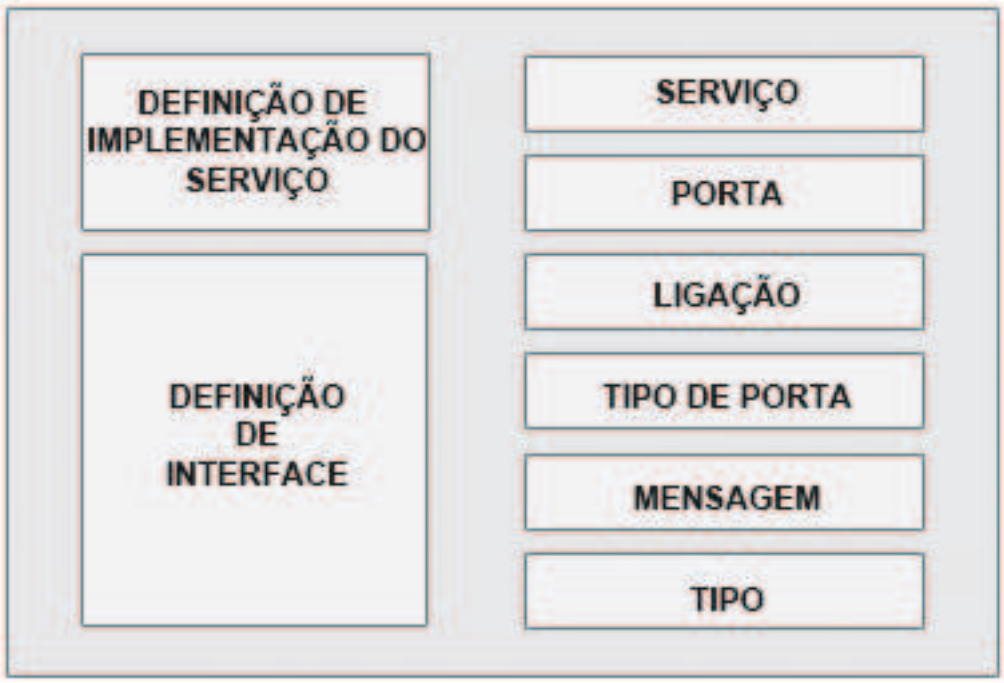


Figura 4: Elementos de um documento WSDL

- ***Types*:** são os tipos de dados utilizados no serviço.
- ***Messages*:** este elemento define, de uma forma abstrata, as mensagens que serão trocadas.
- ***Operations*:** este elemento define, de uma forma abstrata, a estrutura de uma operação chamada a um método de retorno.
- ***Porttype*:** descreve um conjunto abstrato de operações mapeadas para um ou mais serviços, os quais são descritos como pontos finais de rede ou portas.
- **Bindings:** define o formato das mensagens e mapeamento dos elementos Operation para cada Porttype.
- ***Port*:** nomeia o endereço físico (Protocolo de Internet - IP e Porta) que será utilizado para comunicação do Web Services.
- ***Service*:** define onde encontrar um serviço usando sua porta.

2.2 *Extensible Markup Language* (XML)

Por muitos anos a troca de informações entre sistemas heterogêneos foi baseada em arquivos textos. O grande desafio para a comunicação destes sistemas é forma de integrar

os dados, já que este modelo de integração utilizado exige um ao custo de processamento e também a falta de mecanismos de validação da consistência dos dados.

Visando a solução deste problema e a otimização da comunicação entre estes sistemas a W3C (*World Wide Web Consortium*) propôs uma tecnologia aberta para a troca de dados (Bento, 2005, p. 19).

Segundo Turtschi et al (2004, p. 290), o XML é uma linguagem de descrição de dados genérica e independente de plataforma e, como tal, ganhou grande popularidade na área da computação, sendo adotada por grandes empresas do setor computacional, tais como Sun, Microsoft e IBM.

O uso do XML substitui a utilização de arquivos textos e simplifica a tarefa de criar documentos para a troca de dados entre sistemas heterogêneos. A sua especificação possui um conjunto de regras e sintaxe que determina o formato que as marcações devem seguir. O XML tem como uma forte característica o seu mecanismo de extensão de marcações, onde o próprio usuário define suas marcações.

Basicamente, a estrutura do XML é divida em Documento XML, *Parser* XML, *Document Type Definition* (DTD), Esquema XML e *Namespaces*, onde:

- **Documento XML:** é um arquivo que obedece às normas XML, neste compreende toda a estrutura de informações necessárias.
- **Parser XML:** é um programa que analisa o documento XML e reproduz os dados de uma maneira legível para o sistema.
- ***Document Type Definition* (DTD):** é uma especificação que descreve as marcações que podem ser utilizadas e seus relacionamentos com as demais marcações. Desde 2001, este modelo vem sendo substituído pelo modelo de esquema (Esquema XML).
- **Esquema XML:** Evolução do DTD, mantendo as características da forma anterior, garantindo a validação das marcações de acordo com as regras definidas no esquema. Evita a reprogramação, uma vez que a validação é feita de forma externa ao sistema através do Parser XML.
- ***Namespaces*:** utilizado para evitar conflitos de nomes idênticos entre as marcações.

2.2.1 Estrutura da linguagem

Na linguagem XML, dois são os seus componentes principais:

- **Elementos:** são estruturas que permitem atribuição de dados simples, tais como sequências de números e letras.
- **Atributos:** provem informações que não fazem parte dos dados, porém são importantes para a manipulação dos elementos.

Através da Figura 5 podemos ver exemplo de um documento XML:

```
01 <?xml version="1.0" encoding="ISO-8859-1"?>
02 <note>
03 <to>Tove</to>
04 <from>Jani</from>
05 <heading>Reminder</heading>
06 <body>Don't forget me this weekend!</body>
07 </note>
```

Figura 5: Exemplo de um documento XML

Onde temos:

- A primeira linha do exemplo define a versão do XML e o tipo de codificação dos caracteres usados.
- A segunda linha descreve o elemento raiz do documento, como se fosse um título.
- Das linhas 03 a 06 descrevem quatro elementos da raiz (to, from, heading, body).
- E por fim, na última linha aparece o fechamento do elemento raiz. É importante notar que todos elementos possuem uma tag de abertura, por exemplo ¡modelo¿ e também uma tag de fechamento ¡/modelo¿, o não segmento deste padrão é considerado incorreto.

2.3 *Simple Object Access Protocol* (SOAP)

Segundo Santanna (2002), SOAP é um protocolo elaborado para facilitar a chamada remota de funções via internet, permitindo que dois programas se comuniquem de uma maneira tecnicamente muito semelhante à invocação de páginas de internet.

Este protocolo invoca aplicações remotas através de troca de mensagens em um ambiente independente de plataforma e linguagem de programação. Desta forma, garante a

interoperabilidade entre sistemas em ambientes heterogêneos. Sua implementação padrão utiliza-se da linguagem XML e trabalha sobre o protocolo HTTP, permitindo que as mensagens evitem o bloqueio de firewalls.

Breitman (2006), define SOAP como o padrão mais utilizado no desenvolvimento de Web Services, sendo composto dos seguintes elementos, Figura 6:

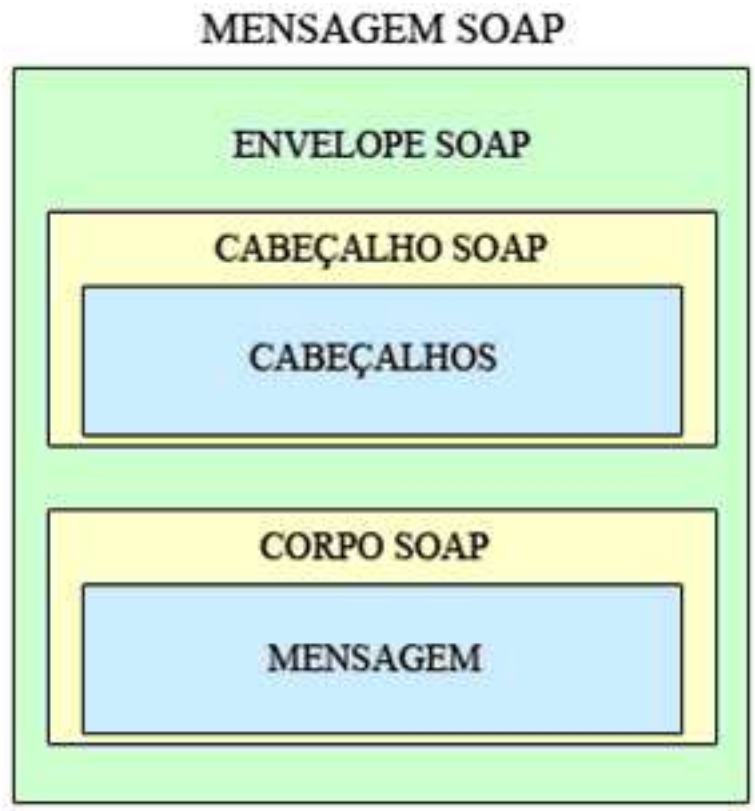


Figura 6: Estrutura da mensagem SOAP

- **Envelope:** elemento que envolve a mensagem.
- **Cabeçalho:** indica o modo que os dados são codificados.
- **Corpo:** contém a mensagem específica.

2.4 Monitoramento remoto de computadores

O modo proposto para realizar o monitoramento remoto de computadores através da internet é a utilização de uma aplicação sendo executada no computador, que dedica-se a comunicar periodicamente com um servidor (Web Services), o qual recebe, processa e

armazena os dados. Desse modo, diversos computadores podem ser monitorados ao mesmo tempo e diversos usuários (administradores) podem monitorar estes computadores.

O Web Services, localizado no servidor, torna a comunicação indireta entre a aplicação executada em computadores e o banco de dados. Pelos padrões utilizados neste serviço mantém a possibilidade de plataformas heterogêneas e garante a integridade dos dados.

Após o processamento dos dados, eles são disponibilizados para os usuários (administradores) que estão executando o monitoramento, para a realização de consultas posteriores (arquivamento dos dados no banco de dados) ou instantâneas. Através de navegadores de internet os usuários acessam estes dados e conseguem interagir com o computador monitorado.

2.4.1 Dificuldades para o monitoramento remoto

O monitoramente remoto de computadores através da internet encontra algumas barreiras as quais dificultam a praticidade e até mesmo podem inviabilizar este processo. Alguns serviços de redes de computadores podem bloquear o acesso aos programas de monitaramento, ou então, é preciso que seja adicionado configurações nestas redes para que se tenha acesso direto aos computadores pertencentes a ela. Algumas destas barreiras são:

- ***Firewall*:** é um dispositivo de rede, o qual seu objetivo é aplicar uma política de segurança a um determinado ponto de controle da rede. Em sua grande maioria, as redes corporativas permitem o acesso à internet através do protocolo HTTP. Este dispositivo pode inviabilizar o processo de monitoramento remoto de computadores via internet, uma vez que este monitoramento não é feito através do protocolo padrão.

- **NAT (*Network Address Translation*):** é uma técnica que consiste em reescrever os endereços IP de origem de pacotes que passam por um Firewall ou Router, para viabilizar o acesso ao endereço de destino. Em uma rede privada com Internet compartilhada, é necessário aplicar esta técnica para viabilizar o monitoramento remoto dos computadores, uma vez que os IP internos da rede não são vistos através da Internet.

- **IP (*Internet Protocol*):** é um protocolo de comunicação usado entre duas ou mais máquinas em rede para encaminhamento de pacotes. Este protocolo sendo

dinâmico (IP Dinâmico) dificulta o monitoramento remoto via internet, uma vez que a cada conexão do cliente, será atribuído um novo valor de IP.

- **Servidor Proxy:** atende requisições de dados de clientes e passa para a sua frente. Por exemplo, um usuário (cliente) conecta-se a um servidor proxy, requisitando algum serviço, conexão, website ou outro recurso disponível em outro servidor. Em geral, computadores em uma rede interna de computadores não possuem endereços válidos na internet e, portanto, não tem uma conexão direta com a mesma.

2.5 Tecnologias empregadas para o desenvolvimento da ferramenta WSPControl

2.5.1 PHP 5

PHP (*Hypertext Preprocessor*) é uma linguagem de programação de computadores interpretada, livre e muito utilizada para gerar conteúdo dinâmico na internet. Por ser uma linguagem modularizada, a torna ideal para instalação e uso em servidores de internet, sendo criado diversos módulos no repositório de extensões PECL (PHP *Extension Community Library*). Sua sintaxe, funções e tidos de dados são parecidas com a linguagem C e C++.

Converse (2001, p. 67) destaca o PHP (*Hypertext PreProcessor*) pela sua capacidade, confiabilidade e facilidade de uso, além de oferecer o melhor tipo de conectividades para todos os servidores *back-end*.

Segundo o PHP Group (2004), na nova versão desta linguagem (PHP 5) destaca-se inúmeras vantagens em relação a versão anterior, entre elas, é citada uma nova modelagem do gerenciamento de memória e também um completo sistema de suporte á orientação a objetos.

A popularidade do PHP se deve à sua facilidade para criar aplicações dinâmicas para a internet, com suporte à maioria dos gerenciadores de bancos de dados existentes, além de interoperabilidade provida por um grande conjunto de funções e extensões (Dall'Oglio).

A liguagem de programação PHP é focada no desenvolvimento para internet, embora tenha variantes em outras áreas (como o PHP-GTK). Seu propósito principal é de implementar soluções velozes, simples e eficientes. Suas características são:

- Velocidade e robustez

- Estruturado e orientação a objetos
- Portabilidade
- Tipagem fraca
- Sintaxe similar a C/C++ e o Perl

2.5.2 PHP SOAP

O SOAP é uma classe nativa do PHP 5 para desenvolvimento de Web Services, a qual utiliza o formato de dados em XML para a sua comunicação. Esta classe comunica-se utilizando o protocolo HTTP, permitindo criar aplicações distribuídas.

Através da utilização de SOAP em PHP é possível escrever um código em PHP que envia informações encapsulada em XML pelo SOAP para uma aplicação de banco de dados.

O funcionamento inicia-se quando a aplicação cliente realiza a chamada remota de uma função enviando um pacote SOAP contendo a descrição do método a ser invocado via HTTP. O servidor recebe o pacote SOAP, interpreta, executa a função correspondente e retorna a resposta da execução também encapsulada via SOAP.

2.5.3 C Sharp

C Sharp é uma linguagem de programação orientada a objetos desenvolvida pela Microsoft, fortemente tipada, baseada no C++ e influenciada por outras linguagens, como Java e Pascal. A sua sintaxe orientada a objetos foi baseada no C++, mas inclui muitas influências de outras linguagens de programação, como Object Pascal e Java.

Em Dezembro de 2001, a pedido da Microsoft, a ECMA (*European Computer Manufacturers Association*) associação internacional dedicada à padronização de sistemas de informação, emitiu a especificação ECMA-334 - Especificação da Linguagem C Sharp. Tal especificação tornou-se um padrão ISO (ISO/IEC 23270). Dentre as características essências do C Sharp, podem ser destacadas:

- **Simplicidade:** uma linguagem simples para se utilizar, porém tão poderosa quanto o C++.

- **Orientada a Objeto:** totalmente orientada a objeto, até mesmo, qualquer variável tem que pertencer a uma classe.
- **Objeto:** tudo é um objeto, System.Object é a classe base de todo o sistema de tipos.
- **Fortemente tipada:** focando evitar erros por manipulação imprópria de tipos, atribuições incorretas, etc.
- **Controle de versões:** para cada código Assembly gerado, é atribuído informação sobre a versão do código, permitindo a coexistência de dois assemblies homônimos, mas de versões diferentes no mesmo ambiente.
- **Suporte a código não legado:** a linguagem pode interagir com código legado de objetos COM e DLL escritas em uma linguagem não-gerenciada.
- **Linguagem gerenciada:** o gerenciamento de memória é feito pelo Runtime via GarbageCollector, e não diretamente pelo programador, reduzindo as chances de cometer erros de gerenciamento de memória.

2.5.4 C Sharp - System.Diagnostics

O *namespace* System.Diagnostics contém classes que permitem interagir com os processos do sistema, logs de eventos e contadores de desempenho do sistema operacional Microsoft Windows.

Através da classe Process é possível monitorar os processos ativos do sistema, iniciar novos processos e interromper os processos em execução. Também é possível recuperar listas de processos em execução ou exibir informações sobre o processo que está em andamento com informações detalhadas de *threads* de processo e módulos.

A classe ProcessStartInfo permite especificar uma variedade de elementos, por exemplo iniciar um novo processo, fluxos de erro, diretórios de trabalho e os verbos de linha de comando e argumentos. Outras classes relacionadas permitem especificar estilos de janela, processo e as prioridades de *thread* e interagir com conjuntos de módulos existentes.

2.5.5 Tecnologias auxiliares

- **Microsoft .NET:** é uma plataforma única para o desenvolvimento e execução de sistemas e aplicações. Esta é executada sobre um ambiente de execução indepedente

de linguagem, iteragindo com um *framework*, isto é, um conjunto de bibliotecas unificadas. Atualmente na versão 3.5, esta plataforma é capaz de executar mais de 20 linguagens de programação diferentes, havendo a interação entre si, como se fossem uma única linguagem.

- **MySQL:** trata-se de um sistema de gerenciamento de banco de dados, sob a licença livre GPL (*General Public License*), que utiliza a linguagem SQL (Linguagem de Consulta Estruturada) para comunicação. O MySQL é otimizado para proporcionar processamento rápido dos dados e tempo curto de resposta sem exigir muito do hardware. Possui características como portabilidade (atualmente suporta diversas plataformas), suporta controle transacional, *Triggers*, cursores, procedimentos e funções.

2.5.6 Considerações sobre as tecnologias

O critério para escolha das tecnologias utilizadas para o desenvolvimento da ferramenta WSPControl baseia-se no enquadramento ao módulo que cada uma constituiu. O PHP é utilizado para o desenvolvimento da aplicação para internet e Web Services por possuir características focadas nesta área, por exemplo, o PHP SOAP, além de ser uma linguagem de programação livre, interpretada e rápida. Já o C Sharp possui bibliotecas nativas para a interação com o Microsoft Windows, posssibilitando um maior controle sobre a aplicação.

2.6 Trabalhos correlatos

Dentre os trabalhos relacionados a ferramenta WSPControl, é possível equiparar aplicações comerciais para acesso remoto à computadores e também mencionar ferramentas de pesquisa acadêmicas de monitoramento remoto utilizando um Web Services.

Aplicações comerciais: o protocolo *Virtual Network Computing* - VNC, segundo a empresa Real VNC Ltda, desenvolvedora deste padrão, permite o acesso remoto a computadores utilizando a internet ou rede de computadores como seu meio de comunicação. Quando conectado, o programa exibe a tela do outro computador em tempo real e permite o controle da máquina remota como se estivesse presente no lugar, podendo assumir o controle total do mouse e teclado.

O programa é divido em dois módulos (cliente e servidor) que fazem a conexão entre os

computadores. Sua segurança é estabelecida com a utilização de senha, onde a máquina com o módulo servidor define uma senha que será utilizada para conexão da máquina cliente e tem total controle sobre quem pode se conectar e quem está conectado naquele instante, oferecendo a opção de retirar as conexões ativas.

Existem diversas implementações deste conceito, utilizando o mesmo protocolo, o que permite a interoperação de clientes e servidores de versões distintas. São exemplos de implementações do VNC e sua principal característica as seguintes aplicações:

- **Real VNC:** programa inicial da utilização do padrão VNC.
- **Ultra VNC:** possui funções extras, como transferência de arquivos e chat.
- **Tight VNC:** otimizado para utilização em conexões lentas.
- **Win2 VNC:** projeção de uma área de trabalho virtual entre duas estações distintas.
- **Tridia VNC:** focado para gerenciamento administrativo da máquina.

Trabalhos acadêmicos: o trabalho entitulado como "Utilização de dispositivos móveis, Web Services e sotfware livre no monitoramento remoto de pacientes", desenvolvido por Alencar Machado, Edson Luiz Padoin, Fabiano Salvadori, Liane Righi, Maurício de Campos, Paulo Sérgio Sausen e Sergio Luiz Dill apresenta uma arquitetura de um sistema para monitoramento remoto de pacientes através da aplicação de dispositivos móveis utilizando um Web Services. No contexto deste trabalho a implementação da aplicação utilizando o Web Services permitiu resolver o problema de interoperabilidade e a interligação com aplicações já existentes.

Já para o trabalho entitulado como "Utilização de Web Service no desenvolvimento de um sistema para monitoramento e diagnose de dispositivos de potência através de telefonia móvel", desenvolvido por Edson L. Padoin, Maurício de Campos, Fabiano Salvadori, Paulo S. Sausen, Robinson F. de Camargo, Gideon Villar Leandro e Tiago A. Idalencio propõe o desenvolvimento de um Web Services que possibilita monitorar o comportamento de dispositivos eletrônicos, e em caso de alguma falha, informar o responsável através de mensagem texto em um equipamento móvel. Neste projeto conclui-se que a utilização do Web Services possibilita um gerenciamento em tempo real do sistema aprimorando as características de mobilidade, robustez, versatilidade e confiabilidade.

A ferramenta WSPControl une duas características importantes destas aplicações (monitoramento remoto e Web Services) e foca o monitoramente remoto de computadores. Com esta união de características obtém a vantagem do armazenamento do histórico

de dados coletados, tornando possível a visualização das informações de monitoramento a qualquer momento e não somente instantaneamente durante a utilização da ferramenta.

Capítulo 3

A ferramenta WSPControl

A aplicação proposta neste trabalho, denominada por WSPControl, auxilia o monitoramento remoto de computadores através da internet, com a utilização de um Web Services é eliminada a necessidade de configurações adicionais em redes de computadores para a liberação de portas específicas ou redirecionamento interno de endereços dinâmicos.

O cenário ilustrado na Figura 7 mostra a comunicação entre uma aplicação cliente sendo executada em computadores a serem monitorados e a aplicação de controle, a qual está hospedada em um servidor na internet, utilizando-se um Web Services como intermediário, também hospedado em um servidor na internet.

A aplicação cliente é responsável por monitorar o computador, conectando-se ao Web Services e enviando estas informações de monitoramento. Já a aplicação de controle, situada na Internet, é responsável pela leitura das informações de monitoramento e também para a definição de novas ações do computador monitorado, como bloqueá-lo ou desligá-lo.

Os dados obtidos pela aplicação são armazenados em um banco de dados para que seja feita a sua leitura posteriormente. Para incorporar a ferramenta WSPControl, o sistema gerenciador escolhido para a base de dados foi o MySQL, o qual está sob a licença livre GPL (General Public License) e que têm evoluído constantemente nos últimos anos. A estrutura do banco de dados para o armazenamento é composto por 4 tabelas e 2 relacionamentos entre elas, Figura 8.

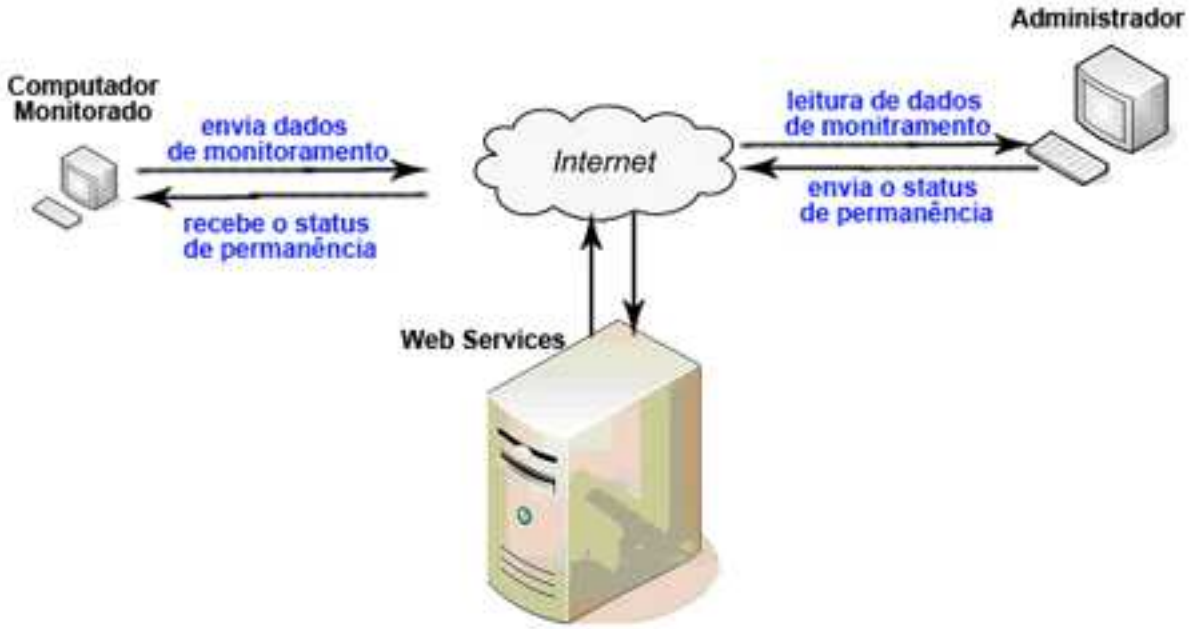


Figura 7: Comunicação entre as aplicações cliente, administrador e Web Services

3.1 Visão Geral

A ferramenta WSPControl é dividida em três módulos, os quais foram denominados como: Interface de Cliente, Interface de Web Services e Interface de Internet, é possível visualizar através da Figura 9 a arquitetura proposta para estes módulos.

A Interface de Cliente é responsável por monitorar o computador, gerar a sintaxe XML, conectar-se ao Web Services e informar os dados captados. A Interface de Web Services é responsável por intermediar a comunicação entre as duas outras aplicações, sendo que da primeira, ele recebe as informações de monitoramente, faz as verificações necessárias e conecta-se com o banco de dados para a gravação das informações. Também faz a leitura e retorna para a Interface de Cliente as possíveis ações do computador monitorado. Já a terceira, Interface de Internet, é responsável pela leitura das informações capturadas, esta por estar hospedada juntamente com o Web Services, conecta-se diretamente com o banco de dados para tal leitura. Através desta interface é possível também controlar a relação de computadores a serem monitorados.

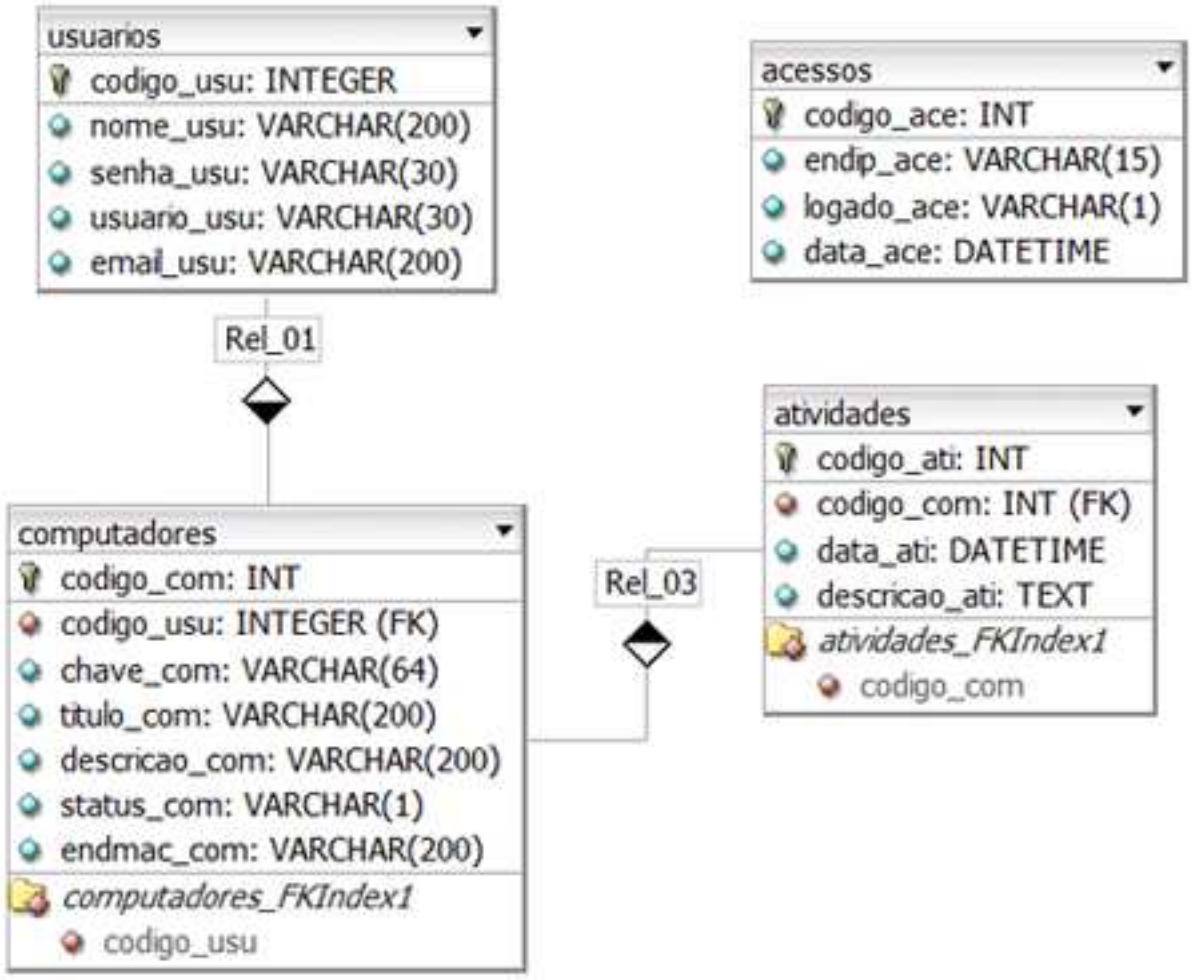


Figura 8: Estrutura do banco de dados

3.2 Contextualização da Arquitetura

O monitoramente remoto de computadores através da Internet encontra algumas barreiras as quais dificultam a praticidade e até mesmo podem inviabilizar este processo, é possível visualizar esta estrutura através da Figura 10.

Podemos observar que a rede corporativa composta por diversos computadores, não encontra-se conectada diretamente com a internet. Para a rede interna é atribuído valores de IP não conhecidos na internet. E também, para a conexão externa à rede, existe um *firewall*, o qual mantém acesso liberado apenas para portas convencionais (por exemplo HTTP na porta 80).

Já para usuários domésticos a dificuldade no monitoramento remoto é grande variação de IP, uma vez que a cada conexão é atribuído um novo valor.



Figura 9: Arquitetura das interfaces do projeto

3.3 Interface de Cliente

Este módulo consiste em uma aplicação para computadores com sistemas operacionais Microsoft Windows, sendo desenvolvido com a linguagem de programação C Sharp e o ambiente de programação Microsoft Visual Studio, possuindo as seguintes funcionalidades:

- Ler as aplicações em execução no computador.
- Conectar com o Web Services, informar as aplicações em execução e ler o status que deverá manter o computador.
- Forçar o computador a possíveis *status*

Para o funcionamento desta interface, ao iniciar a sua execução no computador a ser monitorado é preciso configurá-la, para isto é necessário preencher os campos na aba de Configurações. Deverá ser informado o endereço do Web Services, a chave de identificação e o tempo de comunicação. Caso a conexão com a internet não seja direta e possua Proxy, também deverá ser informado o endereço do servidor e o usuário de autenticação.

Durante o seu funcionamento a aplicação é executa no modo *system-tray* (Figura 11), traduzido por área de notificação, local nos quais são mostrados elementos de controle e visualização de programas em execução no Microsoft Windows. Ela se comunica de tempo em tempo com o Web Services informando as atividades em execução e requisita o status de permanência do computador.

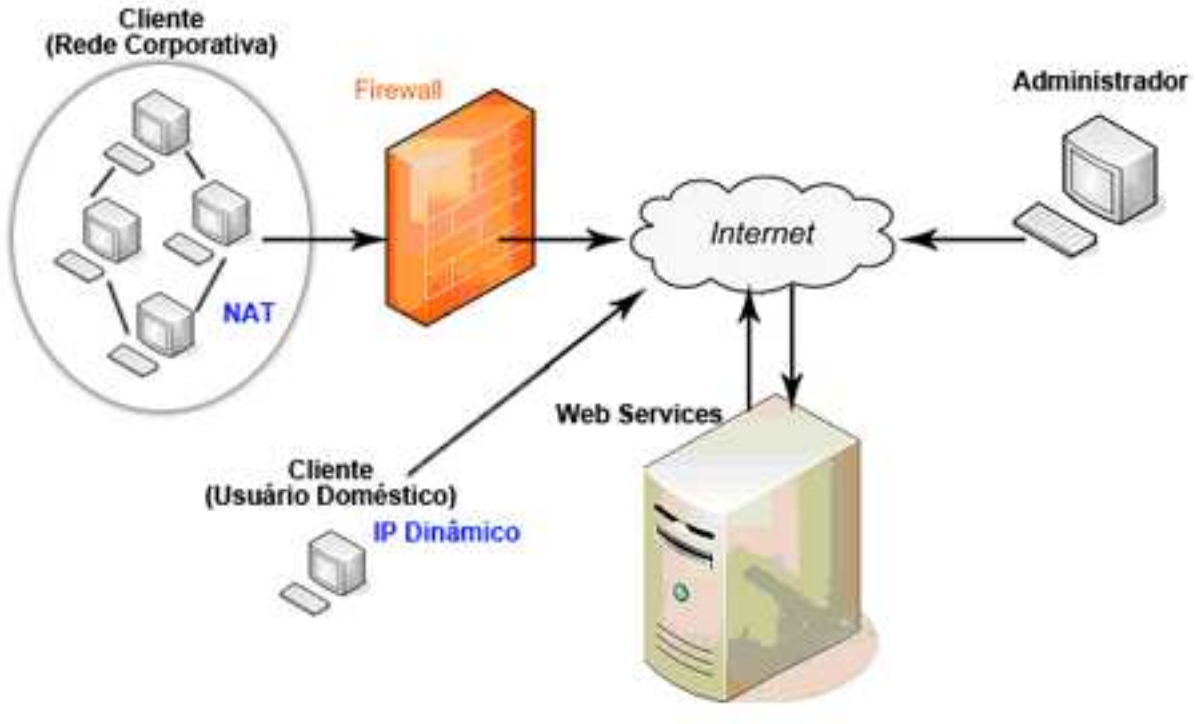


Figura 10: Contextualização da arquitetura de redes de computadores



Figura 11: Execução do Monitor de Processos em *system-tray*

Na aplicação há também a aba Histórico, a qual imprime nesta tela o resultado referente ao monitoramento do computador. Se o resultado é satisfatório, implica que a comunicação e a autenticação com o Web Services foram efetuadas com sucesso, então a aplicação imprime os valores (data/hora e atividades em execução) enviados ao Web Services. Caso contrário, é impresso nesta tela o erro retornado pelo Web Services.

A comunicação entre a aplicação cliente e o Web Services é realizada via RMI (Invocação de Métodos Remotos), através desta comunicação é enviada para o Web Services as informações lidas pela aplicação.

Através de um formato comum, padrão XML, estas informações são enviadas à aplicaçao Web Services. A validação deste formato é composta pela seguinte estrutura:

- **chave:** cadeia de 64 caracteres alfanúmericos.
- **atividades:** texto livre, sem formatação.
- **data:** data e horário, seguindo o padrão DD/MM/AAAA HH:MM:SS (dia, mês,

ano, hora, minuto e segundo).

Na Figura 12 é possível visualizar um exemplo desta sintaxe de comunicação.

```
<comunicacao>
  <chave>
    GP4uKJ3P0munvEW8efRu0wyX3gpxu0ctfYqXXEZ6vfpfZgVnJDfxIPq9csXD5mBF
  </chave>
  <atividades>
    explorer spoolsv svchost httpd postgres pg_ctl svchost searchindexer
    mysqld jqs firefox wuauclt sqlwriter smax4pnp smss winlogon rundll32
    httpd AvastUI httpd httpd svchost svchost MonitorProcessos ctfmon
    nvsvc32 billy AvastSvc oldmcdonald ibserver csrss tsnpstd3 SMax4
    searchprotocolhost svchost lsass GoogleUpdate postgres svchost
    postgres services postgres alg reader_sl sqlservr ibguard svchost
    System postgres postgres Idle
  </atividades>
  <data>
    07/11/2010 21:02:02
  </data>
</comunicacao>
```

Figura 12: Exemplo da sintaxe de comunicação, formato XML

Na classe denominada ChamadorWebservice há os principais métodos desta aplicação, vale ressaltar:

- **SalvarProcessos():** através deste método a aplicação lê as atividades em execução no computador, gera a sintaxe XML e envia para o Web Services.
- **VerificarStatus():** este método é faz a leitura, através do Web Services, do status que o computador deverá permanecer e forçá-lo para tal status de permanência (bloquear, desligar ou desbloquear).
- **FazRequisicaoWeb():** é responsável pela conexão com o Web Services, ele recebe as informações referente ao tipo de conexão, se há Proxy, faz a autenticação na rede.

Por se tratarem dos métodos que se conectam com o Web Services através da internet, ambos possuem a forma de conexão direta, via Proxy e via Proxy autenticado, na Figura 13 é observado como é feito esta autenticação.

Todas as informações registradas após a configuração da interface são fixadas no objeto criado, denominado como Chamador().

```
HttpWebRequest httpRequest = (HttpWebRequest)HttpWebRequest.Create(urlCompleta);

if (usarProxy)
{
        String strAdress = urlProxy + ":" + portaProxy;
        httpRequest.Proxy = new WebProxy(strAdress, true);
}

if (autenticacao)
{
        httpRequest.Proxy.Credentials = new NetworkCredential(usuario, senha);
}
```

Figura 13: Trecho do código fonte para utilização de Proxy

Para o bloquear o computador o método VerificarStatus() cria um novo objeto, a qual a sua classe foi denominada como FormBloqueado, onde há uma tela cheia (*full screen*) que desativa as atividades do teclado e *mouse* do computador. A classe nativa KeyboardHook é responsável pelo bloqueio destas atividades. Já para o ação de desligar o computador é utilizado a classe nativa ShutDown, a qual força encerrar as atividades do computador.

3.4 Interface de Web Services

O módulo Interface de Web Services consiste em uma aplicação desenvolvida em PHP, utilizando da biblioteca nativa PHP SOAP e contém as seguintes funcionalidades:

- Receber as informações de atividades vindas da Interface de Cliente, tratá-las e gravá-las no banco de dados.
- Ler e retornar para a Interface de Cliente o status de permanência do computador.

O Web Services aguarda pela invocação de seus métodos, inicialmente a ferramenta WSPControl, módulo Interface de Cliente, solicita o método atividades, logo o método status. Ao efetuar a primeira invocação é enviado as informações do monitoramento, seguindo o padrão da sintaxe XML, Figura 12. Após ser instanciado, este método recebe as informações, faz a leitura dos dados e os valida, registra o acesso ao Web Services e grava estas informações no banco de dados. Logo, a Interface de Cliente invoca o método de status, que também é instanciado, lê a sintaxe XML, valida os dados e por último retorna o status de permanência do computador. Em ambos os casos de invocações,

caso haja algum erro durante o processamento dos dados, o Web Services retorna uma mensagem informando o problema. A Figura 14 mostra o esquema de invocação do Web Services.

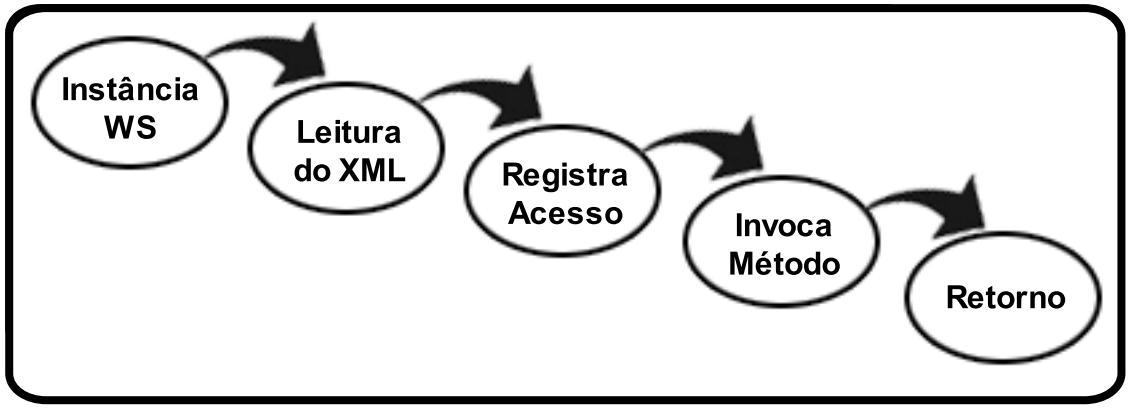


Figura 14: Esquema de invocação do Web Services

A extensão SimpleXML, nativa da liguagem PHP, facilita a manipulação do arquivo XML, convertendo-o em objetos, com esta separação dos dados de monitoramento é possível validá-los.

O registro de acesso ao Web Services é feito através da função ValidaAcesso(), a qual conecta-se com o banco de dados e grava o endereço IP do requisitante e se a tentativa de acesso foi satisfatório.

A descrição e formato dos métodos do Web Services é identificado pelo documento WSDL, o qual é obtido com a ferramenta Visual Studio durante a programação da Interface de Cliente.

Para a execução no servidor, o Web Services deverá ser configurado manualmente. Nas primeiras linhas de programação contém o endereço do Web Services, também deverá ser informado os dados de conexão com o banco de dados.

Os principais métodos do Web Services são: getAtividade() e StatusMicro(). É possível resumidar que o primeiro recebe as informações de monitoramento enviada pela Interface de Cliente, trata estes dados e grava no banco de dados. O segundo, recebe os dados de solicitação, valida estes, e retorna o status.

3.5 Interface de Internet

Para este módulo foi desenvolvido uma aplicação com a linguagem de programação PHP, esta interface tem como funcionalidades:

- Cadastrar responsáveis pelo monitoramento.
- Cadastrar computadores a serem monitorados.
- Interpretar os dados de monitoramento.
- Controlar o status de um computador monitorado.

Este módulo poderá ser hospedado juntamente com o servidor que contém o Web Services, uma vez que ambos são desenvolvidos com a tecnologia PHP. O acesso a ele será via internet, devendo o administrador identificar-se para listar ou monitorar os computadores cadastrados em seu nome. Para cada computador adicionado, esta interface gera uma chave de identificação, que deverá ser configurada na Interface de Cliente.

Através desta interface é possível alterar o status de um computador monitorado, forçando ele a ser bloqueado, desbloqueado ou desligado. Na Figura 15 é possível visualizar trecho do código desta ação.

```
if ($acao == "nstatus")
{
     $nstatus = validaStatus(post("nstatus"));

     mysql_query("UPDATE `computadores` SET `status_com`='$nstatus'
                  WHERE (`codigo_com`=$cod) AND (`codigo_usu`=$codigo_usu)
                  LIMIT 1");

     $nstatus = setStatus($nstatus);

     echo '<script language="JavaScript">';
     echo 'alert("Status do Computador Alterado \n\nStatus: '. $nstatus .'");';
     echo 'window.location="computadores_detalhes.php?cod='. $cod .'";';
     echo '</script>';
     die();
}
```

Figura 15: Trecho do código para alterar o status de um computador monitorado

A utilização deste módulo inicia-se na tela de identificação, onde é possível efetuar o cadastro de um novo usuário (administrador) ou então preencher os dados (usuário e senha) de um já existente.

Após a identificação, a aplicação valida os dados informados. Se a identificação for satisfatória, é criado uma sessão entre o cliente e o servidor utilizando o código de cadastro. Esta visa garantir a segurança das informações que serão mostradas para o usuário. A sintaxe em PHP utilizada é $_SESSION["sessaoCODIGO"].

A validação da sessão é feita através da função Ver_Sessao(), que verifica a existência desta e valida o valor que existe nela.

Através do menu lateral 'Computadores: Incluir' é possível adicionar computadores para serem monitorados, após preencher os campos do formulário e confirmar o cadastro, o sistema grava no banco de dados estas informações. A função chave_aleatoria() gera uma chave de identificação de 64 caracteres alfanúmericos para o computador cadastrado, que deverá ser configurada na Interface de Cliente. Para que não haja duplicidade, antes da confirmação desta chave, a aplicação verifica se ela é única.

É possível listar os computadores cadastrados vinculados ao usuário através da opção 'Computadores: Listar', selecionando o botão detalhes de um computador a aplicação trará informações referente ao monitoramente deste. Nesta tela, é possível visualizar o histórico de atividades enviado pela Interface de Cliente, também poderá alterar o status de permanência do computador monitorado. Ao forçar o novo status, a aplicação irá gravar no banco de dados esta requisição, onde a Intercade de Cliente deverá efetuar a leitura no próximo instante programado.

3.5.1 Segurança

Esta interface é desenvolvida para funcionar através da internet, sendo assim é necessário que haja uma segurança nesta aplicação. Para isso foi utilizado o sistema de Sessions da liguagem de programação PHP. Este sistema estabelece uma sessão única entre o cliente (navegador) e o servidor, esta sessão se expirá após 20 minutos de inatividade dentro do sistema.

A sessão é criada após a identificação correta do usuário administrador, sendo validado através das informações enviadas pelo formulário inicial. Durante toda a utilização do sistema, ao iniciar uma página, é verificado a existência da sessão para o usuário, em caso negativo o usuário é direcionado para a página inicial da aplicação.

Outra característica de segurança tratada nesta aplicação é inibição de ataques do tipo SQL Injection, na qual, todas as informações obtidas por formulários são validadas para que não haja caracteres especiais nocivos.

Capítulo 4

Estudo de Caso

Para explicação mais concreta da utilização do projeto em questão, é proposto o seguinte estudo de caso:

"Em uma escola existe um laboratório de informática com oito computadores que precisam ser monitorados. Este laboratório está conectado em uma rede de computadores e possui conexão com a internet através de um servidor, o qual utiliza um Proxy sem autenticação para navegação. A escola não contém nenhum funcionário em tempo integral para controlar o uso destes computadores, entretanto existe esta necessidade por seus usuários serem, em sua maioria, crianças e adolescentes.". O ambiente proposto para o estudo de caso pode ser visualizado através da Figura 16.

A partir desta situação deseja-se efetuar o cadastro dos computadores a serem monitorados e também efetuar o acompanhamento da utilização dos mesmos.

Para iniciar, o responsável pelo monitoramento deverá efetuar o seu cadastro na Interface de Internet, criando um usuário e uma senha de identificaçao, Figura 17.

Após o seu cadastro e a sua identificação nesta interface, o responsável deverá adicionar os computadores a serem monitorados. Para isto, deverá utilizar o item 'Computadores: Incluir' de sua área administrativa, Figura 18.

Na Figura 19 podemos observar a relação de todos os computadores que foram cadastrados. Para cada computador adicionado, uma função específica gera uma chave aleatória de identificação, Figura 20. A chave deverá ser configurada na Interface de Cliente para cada computador cadastrado, sendo esta que irá identificar o computador monitorado, Figura 21. Também nesta tela de configuração da Interface de Cliente deverá ser informado o endereço do Web Services, tempo de atualização das informações e, se ne-

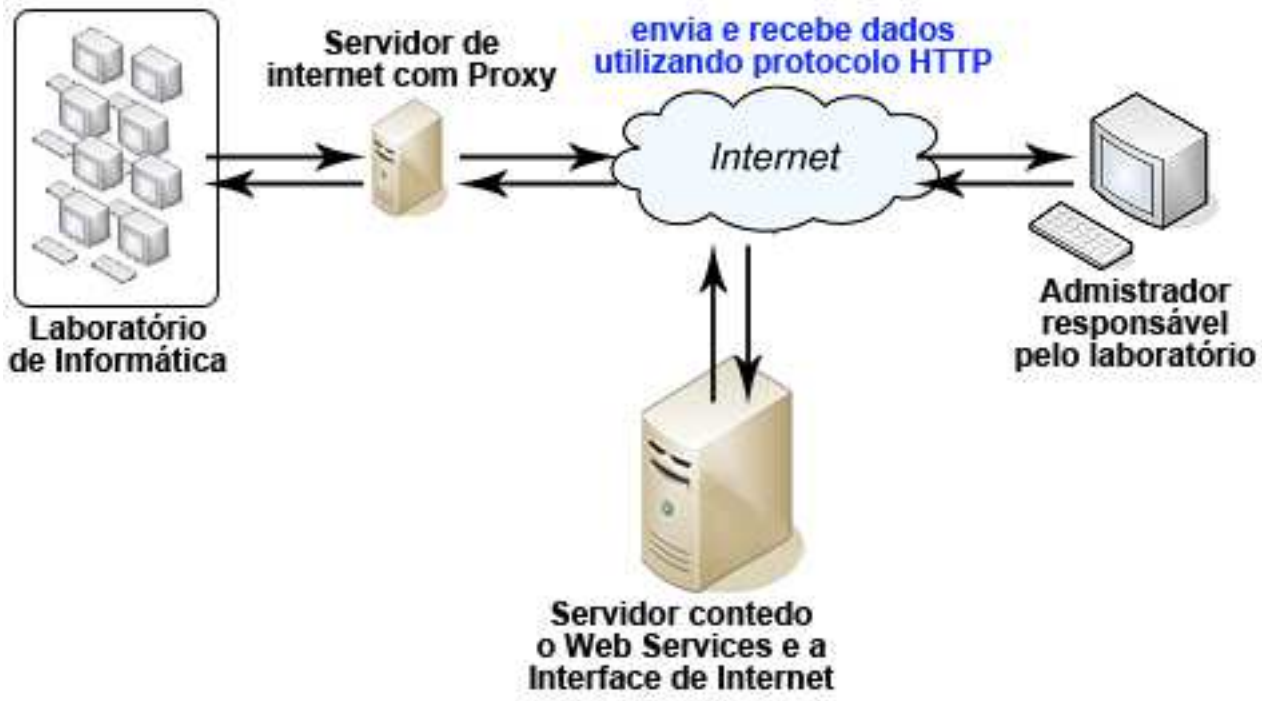


Figura 16: Estrutura do ambiente do estudo de caso

cessário, o Proxy para conexão com a internet. Após a confirmação deste processo inicial de configuração, com os dados preenchidos corretamente, a Interface de Cliente iniciará a comunicação com o Web Services e enviará as informações captadas no computador, Figura 22.

Após dar início ao monitoramento dos computadores, é possível visualizar os dados coletados pelo sistema através da Interface de Cliente. Ao selecionar o botão 'Visualizar' de um determinado computador cadastrado (Interface de Internet), aparecerá às informações de monitoramento referente a este, Figura 23.

O administrador responsável pelo laboratório poderá analisar estes dados obtidos através da Interface de Internet, na Figura 24 podemos visualizar o histórico de atividades de um computador em específico, onde são mostrados todos os processos em execução que foram captados pelo sistema de monitoramento. Podemos notar que alguns destes processos são nativos do Microsoft Windows, como: svchost, nvsvc32, taskmgr, spoolsv, explorer, smss, winlogon, RTHDCPL, services, System Idle e outros. Já outros processos, são atividades sendo executadas durante a utilização do computador, como: msnmsgr (MSN Messenger - bate-papo), WINWORD (Microsoft Word - processador de texto), iexplore (Internet Explorer - navegador de páginas na internet).

Através desta mesma tela de leitura de dados, o administrador poderá também veri-

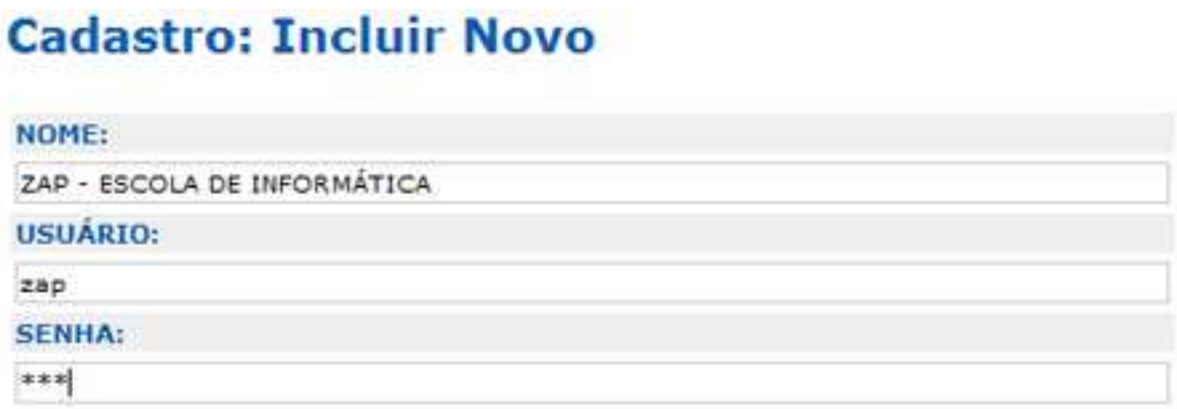


Figura 17: Interface de Internet: Cadastro de administrador



Figura 18: Interface de Internet: Cadastro de computadores

ficar e definir o status do computador, Figura 25. É possível alterar este item utilizando a opção 'Forçar status para'. Ao definir o status para BLOQUEADO, o computador monitorado desabilitará as suas funções de teclado e mouse através da Interface de Cliente e abrirá um aviso, Figura 26. Neste mesmo item também é possível desligar o computador monitorado, o qual finalizará todas as suas atividades e encerrará o seu uso.



Figura 19: Interface de Internet: Computadores cadastrados



Figura 20: Chave de identificação de um computador

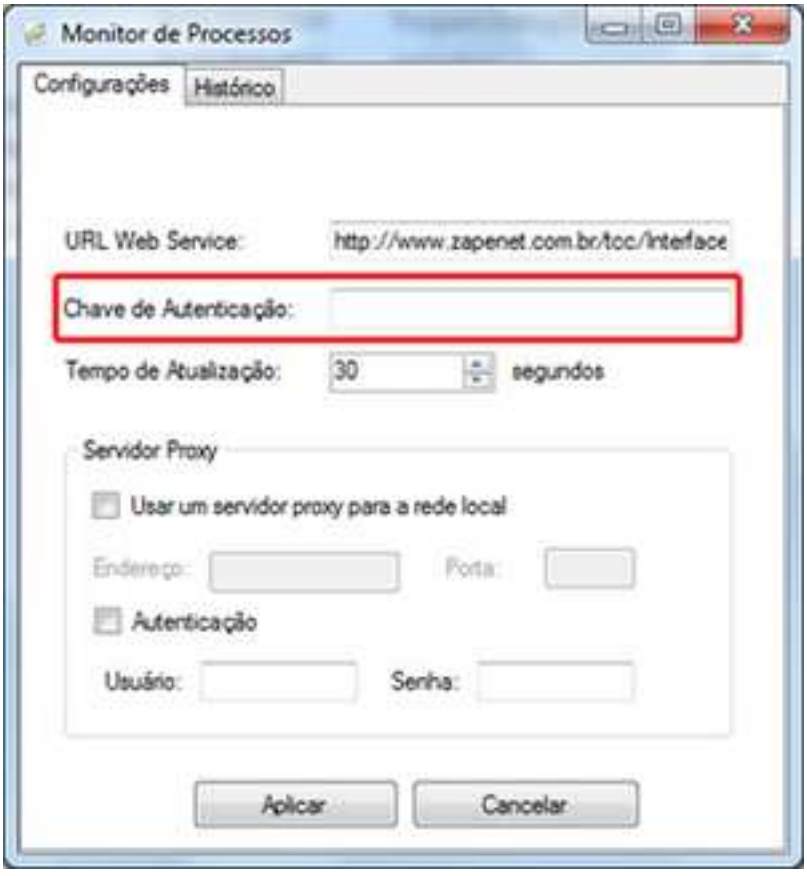


Figura 21: Interface de Cliente: Área de configuração

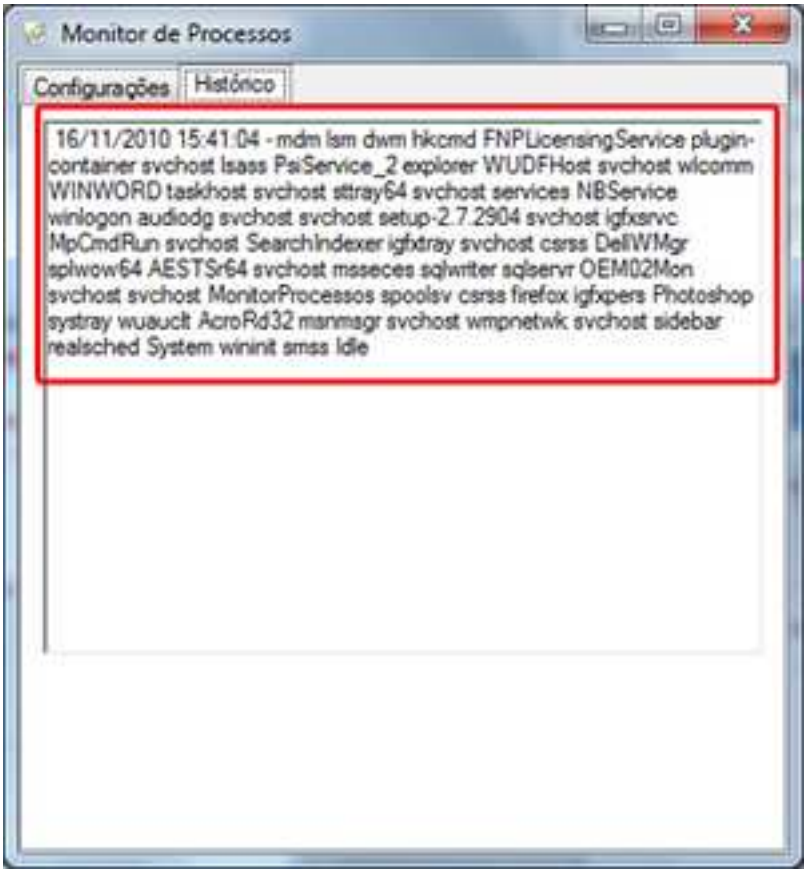


Figura 22: Interface de Cliente: Histórico de atividades em execução



Figura 23: Interface de Internet: Visualizar mais informações de um computador



Figura 24: Histórico de atividades de um computador



Figura 25: Monitoramento do status de um computador

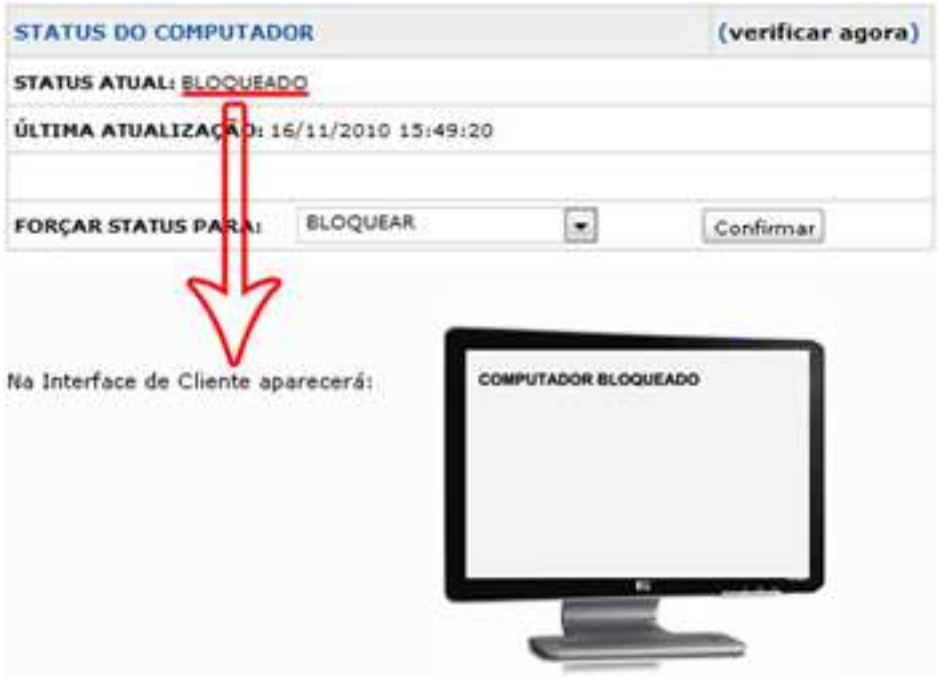


Figura 26: Alteração do status de um computador monitorado

Capítulo 5

Conclusão

Através da ferramenta WSPControl é possível obter informações da utilização de computadores, bloqueá-los e desligá-los em tempo ágil. A sua principal vantagem para o monitoramento é permanência do histórico de dados coletados, sendo assim, é possível analisar estas informações a qualquer momento e não somente instantaneamente durante a utilização do computador pelo usuário.

A Interface de Internet possibilitou o monitoramente remoto dos computadores através da internet, independentemente do local em que o responsável pelo monitoramento se encontra, bastando apenas conexão com a internet. O que torna possível o monitoramento, até mesmo, através do celular, uma vez que o crescimento da tecnologia 3G em celulares é evidente.

Com a utilização de um Web Services para a ferramenta é eliminada a preocupação com configurações adicionais em redes de computadores, não sendo necessário adicionar regras ou permissões no servidor da internet nestas redes, uma vez que o acesso da aplicação é via internet utilizando o protocolo comum HTTP.

Por isto é satisfatório o resultado do projeto proposto de monitoramento remoto através da internet utilizando aplicações distribuídas.

5.1 Trabalhos Futuros

Para continuação de pesquisa para a ferramenta WSPControl poderão ser inseridos novas funcionalidades para os módulos, na Interface de Cliente: realizar pesquisa para o desenvolvimento de uma ferramenta de segurança para evitar que o programa seja, real-

mente, finalizado. Outra função adicional para este módulo é a incorporação de funções além da leitura dos dados em execução, como por exemplo o bloqueio de programas específicos definidos pelo administrador antes que eles sejam iniciados.

Já na Interface de Internet é possível incrementar a aplicação com a interpretação automática dos principais programas em execução (por exemplo Microsoft Word, Internet Explorer, Live Messenger, entre outros) e também gráficos com estatísticas de uso dos programas e horários de monitoramento.

Referências Bibliográficas

BREITMAN, Karin Koogan. Web Semântica a Internet do Futuro. Editora LTC, Rio de Janeiro, 2005.

CONVERSE, Tim; PARK, Joyce. PHP a bíblia. Tradução Edson Furmankiewics. Rio de Janeiro, Campus, 2001.

Dall'Oglio, P. PHP-GTK, Criando Aplicações Gráficas com PHP. Editora Novatec, 2004.

DIZARD Jr., Wilson. A nova mídia: a comunicação de massa na era da informação. Rio de Janeiro, Jorge Zahar Ed., 2000.

HANSEN, R.; PINTO S.; Construindo Ambientes de Educação na Web Através de Web Services Educacionais. Programa Interdisciplinar de PósGraduação em Computação Aplicada.

MACHADO, ALENCAR; PADOIN, EDSON L.; SALVADORI , FABIANO; RIGHI , LIANE; CAMPOS, MAURÍCIO; SAUSEN, PAULO SÉRGIO; DILL, SERGIO LUIZ; Utilização de dispositivos móveis, Web Services e sotfware livre no monitoramento remoto de pacientes. Universidade Regional do Noroeste do Estado do Rio Grande do Sul

Mello, E. R., Wangham, M. S., da Silva Fraga, J., e Camargo, E. (2006). Segurança em Serviços Web, chapter 1, pages 1-48. Minicursos do SBSeg 2006. Sociedade Brasileira

de Computação.

PADOIN, EDSON; CAMPOS, MAURÍCIO; SALVADORI, FABIANO; SAUSEN, PAULO S.; CAMARGO, ROBINSON F.; LEANDRO, GIDEON VILLAR; IDALENCIO, TIAGO A.; Utilização de dispositivos móveis, Web Services e sotfware livre no monitoramento remoto de pacientes. Universidade Regional do Noroeste do Estado do Rio Grande do Sul

PHP Group. PHP: hypertext preprocessor. [S.1.], 2004. Disponível em: http://www.php.net. Acessado em: 30 abril 2010.

SANTANA, A.; Lopes, L.; Bittencourt, P.; Ohnish, Y.; WEB SERVICE .NET. Disciplina Tecnologias Web, Professor: Antônio Benedito Coimbra Sampaio Jr, UNAMA - Universidade da Amazônia.

TURTSCHI, Adrian et al. C Sharp .NET: guia do desenvolvedor Web, curso completo. 2. ed. Rio de Janeiro, Alta Books, 2004. 517 p.

WHAT IS VNC. RealVNC Ltd. Disponível em: http://www.realvnc.com/what.html Acessado em: 08 outubro 2010.