

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

**SOFTWARE DE GERAÇÃO DE FILTROS WMI NA CRIAÇÃO
DE REGRAS *GROUP POLICY OBJECTS***

VALDIR STEINBACH JÚNIOR

BLUMENAU
2007

2007/1-21

VALDIR STEIBACH JÚNIOR

**SOFTWARE DE GERAÇÃO DE FILTROS WMI NA CRIAÇÃO
DE REGRAS *GROUP POLICY OBJECTS***

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho
de Conclusão de Curso II do curso de Sistemas
de Informação - Bacharelado.

Prof. Francisco Adell Péricas, Mestre

**BLUMENAU
2007**

2007/1-21

**SOFTWARE DE CRIAÇÃO DE FILTROS WMI NA CRIAÇÃO
DE REGRAS *GROUP POLICY OBJECTS***

Por

VALDIR STEINBACH JÚNIOR

Trabalho aprovado para obtenção dos créditos
na disciplina de Trabalho de Conclusão de
Curso II, pela banca examinadora formada
por:

Presidente: _____
Prof. Francisco Adell Péricas, Mestre – Orientador, FURB

Membro: _____
Prof. Antonio Carlos Tavares, Especialista – FURB

Membro: _____
Prof. Sérgio Stringari, Mestre – FURB

Blumenau, 2 de julho de 2007

Dedico este trabalho a todos os amigos, especialmente aqueles que me ajudaram diretamente na realização deste. Ao meu orientador pelo apoio dado desde o início e minha família e namorada pela paciência e compreensão nos momentos difíceis da minha jornada.

AGRADECIMENTOS

À Deus, pelo seu imenso amor e graça.

À minha família, que mesmo longe, sempre esteve presente.

Aos meus amigos, pelos empurrões e cobranças.

A empresa pela ajuda muitas vezes em horas difíceis.

Aos meus pais por terem acreditado em mim e na minha capacidade.

A minha namorada por tantos momentos de compreensão.

Ao meu orientador, Francisco Péricas, por ter acreditado na conclusão deste trabalho.

Os bons livros fazem “sacar” para fora o que a
pessoa tem de melhor dentro dela.

Lina Sotis Francesco Moratti

RESUMO

Com o acentuado crescimento das opções de software e hardware, a tarefa de administração de rede tem se tornado muito complexa. A Microsoft inovou e trouxe para o mercado o *Active directory* e com ele vieram inúmeras possibilidades de gerenciamento e administração de políticas de segurança. As *Group policy objects* tiveram seu crescimento com o surgimento do *Active Directory*, mas com o número imenso de possibilidades e o crescimento do número de máquinas encontradas nas empresa surgiu o *Windows Management Instrumentation*, WMI, com o qual é possível fazer a leitura de informações do hardware e software da máquina do cliente, possibilitando a criação de políticas mais específicas para determinados software e hardware. Este trabalho tem como finalidade apresentar uma software que auxilie no processo de criação de *Group policy objects* vinculadas a regras WMI.

Palavras-chave: *Group policy objects*. *Active directory*. *WMI*.

ABSTRACT

With the accentuated growth of the options of software and the hardware, the task of network administration has become very complex. Microsoft innovated and brought for the market Active Directory and with it they had come innumerable possibilities of management and administration of security policies. The Group policy objects its growth with the sprouting of Active Directory, but with the immense number of possibilities and the growth of the numbers of machines in the company become the Windows Management Instrumentation appeared, WMI, with which possible to make the reading of information of the hardware and software machine of the customer, making possible the creation of more specific policies for determined software and hardware. This work has a purpose to present a tool that assists in the process of creation of Group policy objects tied rules WMI.

Key-words: *Group policy objects. Active directory. WMI.*

LISTA DE ILUSTRAÇÕES

Figura 1: Estrutura de armazenamento das políticas.....	18
Figura 2: Seqüência de aplicação da GPO	23
Figura 3: Console de administração das GPOs	25
Figura 4: Console de edição de uma determinada GPO.....	26
Figura 5: Diagrama da arquitetura de funcionamento do WMI e CIM.....	27
Figura 6: Protocolos e interfaces em uma rede de uma empresa.....	31
Figura 7: Arquitetura padronizada do WBEM.	32
Figura 8: GPO com filtro WMI aplicado.....	33
Figura 9: Diagrama de casos de uso.	37
Figura 10: Diagrama de pacotes.	37
Figura 11: Diagrama de atividades.....	38
Figura 12: Diagrama de classes.	39
Figura 13: Diagrama de <i>grid</i> do XML.	39
Figura 14: Diagrama do XML.	40
Figura 15: Arquivo do CSS usado pelo software.	41
Figura 16: Arquivo do CSS usado pelo software (continuação)	42
Figura 17: Arquivo do CSS usado pelo software (continuação)	43
Figura 18: Arquivo do CSS usado pelo software (continuação)	44
Figura 19: Arquivo responsável pela criação dos objetos do AJAX utilizado pelo software. .	45
Figura 20: Arquivo do AJAX usado pelo software.	45
Figura 21: Estrutura dos arquivos.....	46
Figura 22: <i>Function</i> com o carregamento e o cancelar.	47
Figura 23: <i>Function</i> com a validação dos dados.	48
Figura 24: <i>Function</i> com a adição de filtros ao software.	48
Figura 25: <i>Function</i> responsável por gravar os dados.....	49
Figura 26: Tela inicial do software.....	50
Figura 27: Importação de filtros.	51
Figura 28: Alteração ou visualização do filtro.	51
Figura 29: Cadastro de um filtro.	52
Figura 30: Visualização dos filtros depois de cadastrados.....	52
Figura 31: Exclusão de filtro.	53

Figura 32: Exportação de um filtro.	53
Figura 33: Filtro na console GPMC.	54
Figura 34: <i>include</i> responsável pela configuração do software.....	62
Figura 35: <i>include</i> responsável pela configuração do software.....	63
Figura 36: <i>include</i> responsável pela importação do software.....	64
Figura 37: <i>include</i> responsável pela adição de filtros.....	65
Figura 38: <i>include</i> responsável pela adição de filtros.....	65
Figura 39: <i>include</i> responsável por funções no software.	66
Figura 40: <i>include</i> responsável por funções no software (continuação)	67
Figura 41: <i>include</i> responsável por colocá-los em ordem alfabética.....	67
Figura 42: <i>include</i> responsável por carregar os filtros para a memória.....	68
Figura 43: <i>include</i> responsável por alterar os filtros que já existem na memória.	69
Figura 44: <i>include</i> responsável por excluir que estão no software.....	69
Figura 45: <i>include</i> responsável por carregar as <i>queries</i> para o software.....	70
Figura 46: <i>include</i> responsável por carregar o <i>where</i> para as <i>queries</i>	70

LISTA DE QUADROS

Quadro 1: Perguntas freqüentes enfrentadas pelo administrador	17
Quadro 2: Configurações dos usuários.....	21
Quadro 3: Configurações dos computadores.....	22
Quadro 4: Exemplos de filtros WMI.....	33
Quadro 5: Requisitos funcionais.....	35
Quadro 6: Requisitos não funcionais.....	36
Quadro 7: dispositivos de refrigeração.....	71
Quadro 8: classes dos dispositivos de entrada.....	72
Quadro 9: classes dos dispositivos de armazenamento	72
Quadro 10: representação das classes da placa mãe e outros componentes.....	74
Quadro 11: classe dos dispositivos de rede	74
Quadro 12: classes da subcategoria energia	75
Quadro 13: apresentado as classes da impressora	76
Quadro 14: classe de vídeo e monitores	76
Quadro 15: classes da área de trabalho.....	77
Quadro 16: classes do sistema de arquivos	78
Quadro 17: classes da categoria rede e domínio.....	79
Quadro 18: classes responsáveis pelas configurações do sistema operacional	80
Quadro 19: classes do serviço e suas bases	81
Quadro 20: classes WMI da subcategoria do compartilhamento	82
Quadro 21: classes do armazenamento.....	83
Quadro 22: classes da categoria usuário.....	84

LISTA DE SIGLAS

AD – *Active Directory*

AJAX – *Asynchronous Javascript and XML*

CASE – *Computer Aided Software Engineering*

CIM – *Common Information Model*

CIMOM – *repositório do objeto CIM*

DMFT – *Distributed Management Task Force*

GPMC – *Group Policy Management Console*

GPO – *Group Policy Objects*

MOF – *Managed Object Format*

OU – *Organization Unit*

PHP – *Hypertext Preprocessor*

TCO – *Total Cost of Ownership*

UML – *Unified Modeling Language*

XML - *eXtensible Markup Language*

WBEM – *Web-based Enterprise Management*

WDM – *Windows Driver Model*

WMI – *Windows Management Instrumentation*

WQL – *WMI Query Language*

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 OBJETIVOS DO TRABALHO	15
1.2 ESTRUTURA DO TRABALHO	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 GROUP POLICY OBJECTS	16
2.1.1 Armazenamento das políticas	18
2.1.2 Processamento das políticas.....	19
2.1.3 Tipos de GPOs	19
2.1.4 Políticas de usuários.....	20
2.1.5 Políticas de computadores.....	21
2.1.6 Associações, links e herança na GPO	22
2.1.7 Console de administração das políticas.....	24
2.2 FILTROS WMI.....	26
2.2.1 Arquitetura do WMI.....	29
2.2.2 Visão geral sobre o WBEM	30
2.3 UM FILTRO WMI A UMA GPO.....	32
2.4 TRABALHOS CORRELATOS.....	33
3 DESENVOLVIMENTO DO TRABALHO.....	35
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	35
3.2 ESPECIFICAÇÃO	36
3.2.1 Diagrama de caso de uso.....	36
3.2.2 Diagrama de atividades	37
3.2.3 Diagrama de classes	38
3.2.4 Diagrama do XML	39
3.3 IMPLEMENTAÇÃO	40
3.3.1 Técnicas e ferramentas utilizadas.....	40
3.3.2 Operacionalidade da implementação	49
3.4 RESULTADOS E DISCUSSÃO	54
4 CONCLUSÕES.....	56
4.1 EXTENSÕES	57
REFERÊNCIAS BIBLIOGRÁFICAS	58

APÊNDICE A – Descrição dos cenários.....	60
ANEXO A – Código-fonte dos <i>includes</i> do software	62
ANEXO B – Classes WMI relevantes	71
Classes dos dispositivos de refrigeração	71
Classes dos dispositivos de entrada	71
Classes de dispositivos de armazenamento	72
Classes da Placa Mãe, portas e controladores	72
Classes dos dispositivos de rede	74
Classe da energia	74
Classes das impressoras	75
Classes do Video e Monitor	76
Área de trabalho.....	76
Sistema de arquivos	77
Rede e domínio	78
Configurações do Sistema operacional.....	79
Serviços	80
Compartilhamentos.....	81
Armazenamento.....	82
Usuários	83

1 INTRODUÇÃO

Atualmente o administrador de rede tem um árduo trabalho para conseguir configurar todas as estações existentes no ambiente corporativo, pela grande diversificação de máquinas, usuários e sistemas operacionais. De acordo com Battisti (2003), os grandes desafios enfrentados hoje estão em definir configurações de uma maneira centralizada e organizada, com padrões pré-estabelecidos. Outro problema encontrado é como impedir que os usuários possam alterar as configurações do Windows, que muitas vezes acaba causando problemas no sistema operacional, o que faz necessário um chamado à equipe de suporte.

“O recurso de *Group Policy Objects* (GPO) é de enorme utilidade para o Administrador. Com o uso de GPO o administrador pode definir as configurações de vários elementos da estação de trabalho do usuário. As configurações feitas via GPO são aplicadas para usuários, computadores, *member servers* e DC's, mas somente para computadores executando Windows 2000 (Server ou Professional), Windows XP ou Windows 2003 Server” (BATTISTI, 2003, p.1006).

As regras de políticas de grupo (GPO) trouxeram uma grande revolução ao mercado de servidores corporativos com o lançamento desse recurso no Windows 2000 Server. O administrador tem um recurso muito eficiente.

Group Policy está no *Active Directory* – mecanismo responsável pelo controle de usuários e computadores em domínios. De acordo com Evenson (2003), políticas de grupo são aplicadas e forçadas em múltiplos níveis.

Mesmo com a GPO, outros problemas apareceram com o seu surgimento. Um deles é a diferenciação dos ambientes em que essas regras existem. Em ambientes muito heterogêneos, onde se encontram diversos tipos de máquinas, processadores, e sistemas operacionais, torna-se um modelo complexo de se administrar, trazendo muitos custos operacionais e por consequência requerendo uma equipe especializada responsável pelo funcionamento dessa estrutura.

Desta forma, este trabalho propõe a utilização de filtros *Windows Management Instrumentation* (WMI) para ajudar na elaboração das GPO existentes no ambiente Windows. Com os filtros do WMI é possível fazer um melhor acerto ao se aplicar uma política em um grupo de máquinas, com determinado sistema operacional ou processador, por exemplo.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho foi desenvolver um software destinado à geração de filtros *Windows Managment Instrumentation* para uma melhor utilização das regras de política de grupo existente no *Active Directory* no *Windows 2003 Server*.

Os objetivos específicos do trabalho são:

- a) possibilitar a criação de filtros através das classes apresentadas e especificadas no *Windows Managment Instrumentation*;
- b) a importação dos filtros deve ser automática para o ambiente atual do *Group Policy Managment Console*;
- c) utilizar a linguagem *eXtensible Markup Language* (XML) para armazenamento dos filtros criados e a exportação para o console de administração das GPOs (*Group Policy Managment Console*);
- d) auxiliar o administrador de redes no processo de aplicação das regras de política de grupo.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está organizado em quatro capítulos. A estrutura do trabalho foi definida seguindo os padrões propostos pela coordenação. No primeiro capítulo tem-se a introdução, a justificativa do trabalho, o objetivo geral e objetivos específicos. No segundo capítulo é exposta a fundamentação teórica, onde são abordados temas relevantes como GPO e Filtros WMI, sua arquitetura bem como componentes relevantes na sua estrutura. Além também de trabalhos correlatos. No terceiro é abordado o desenvolvimento do trabalho. Para um melhor entendimento esta seção foi dividida em requisitos principais, especificação, implementação, técnicas e ferramentas utilizadas e resultados. E o quarto capítulo é exposto a conclusão final e as extensões, este último com sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são abordados alguns assuntos relevantes sobre o tema deste trabalho. Na primeira seção são relatados os conceitos, as definições sobre GPO, políticas ligadas ao usuário, políticas ligadas ao computador e sobre os filtros WMI e sua arquitetura. Ao final será exposto sobre filtros e as diretivas de grupo.

2.1 GROUP POLICY OBJECTS

O conceito de GPO tem sido cada vez mais explorado pelos administradores de redes e gerentes de rede. Muito desse interesse se deu pelo seu surgimento no Windows 2000 e com suas novas tecnologias como a GPO (EVENSON, 2003). A GPO surgiu com o *Active Directory* e trouxe muitos benefícios aos administradores pela sua centralização, padronização e simplicidade de uso.

“As GPOs são objetos mais importantes do ponto de vista de segurança lógica no Windows 2000, pois permitem definir todos os aspectos no nível de permissões que cada utilizador pode ter.” (FERREIRA, 2003).

Segundo Battisti (2003), o recurso de GPO é algo grandioso. E de enorme utilidade para o administrador de redes.

Utilizando-se das GPOs é possível fazer ajustes e/ou definições sobre usuários e computadores presentes no domínio do *Active Directory*. Em outras palavras, pode-se ter um controle de tudo o que o usuário do computador tem acesso, onde irão salvar arquivos, permissões de registro e configurações de internet (DUNALEY et al., 2004, tradução nossa).

Com o uso de GPO o administrador pode definir configurações para diversos elementos da estação do usuário e também políticas para o próprio usuário. As regras vão desde ter acesso ao menu executar, ou quais programas serão acessíveis para aquele usuário. São milhares de configurações disponíveis para o administrador (BATTISTI, 2003)

As configurações da GPO têm suas limitações. Conforme Battisti (2003), somente computadores executando Windows 2000, Windows XP Professional e Windows 2003 têm possibilidade de serem gerenciado através dessas políticas.

Atualmente os custos com a área de administração são elevados para manter a configuração de milhares de estações rodando diversos tipos de sistema operacionais, softwares e hardware. Muitas questões são levantadas pelos administradores e diretores.

No quadro 1 são mostradas algumas perguntas que são enfrentadas pelos administradores:

Como definir configurações de maneira centralizada, para que seja possível padronizar as configurações das estações de trabalho?
Como impedir que os usuários possam alterar as configurações do Windows (diversas versões), muitas vezes inclusive causando problemas no Windows, o que faz com que seja necessário um chamado à equipe de suporte, para colocar a estação de trabalho novamente em funcionamento?
Como aplicar configurações de segurança e bloquear opções que não devam estar disponíveis para os usuários de uma maneira centralizada, sem ter que fazer estas configurações em cada estação de trabalho. Quando houver alterações, como fazê-las em um único local e ter estas alterações aplicadas em toda a rede ou em partes específicas da rede?
Como fazer a instalação e distribuição de software de uma maneira centralizada, sem ter que fazer a instalação em cada estação de trabalho da rede?
Como definir um conjunto de aplicações diferente, para diferentes grupos de usuários, de acordo com as necessidades específicas de cada grupo?
Como aplicar diferentes configurações aos computadores de diferentes grupos de usuários, de acordo com as necessidades específicas de cada grupo?

Fonte: adaptado de Battisti (2004, p. 581)

Quadro 1: Perguntas freqüentes enfrentadas pelo administrador

Conforme Battisti (2003) com a introdução das políticas de grupo, o administrador acabou ganhando um poderoso recurso para gerenciamento e administração da sua rede. As questões levantadas anteriormente podem ser todas elas resolvidas através da GPO.

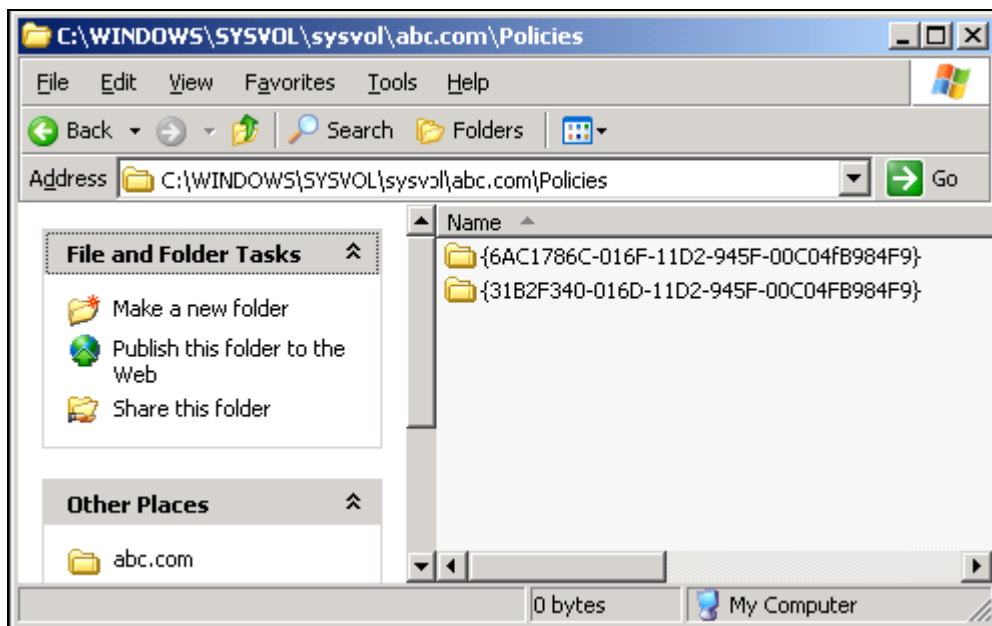
Seguem algumas características que as políticas de grupo podem fazer, conforme Battisti (2003):

- a) gerenciamento centralizado, conforme definições no *registry* do Windows e com base em *templates* de administração;
- b) atribuição de scripts na hora que o usuário faz o *login* ou *logoff*;
- c) redirecionamento de pastas comuns do usuário, como meus documentos;
- d) instalação de softwares;
- e) configurações de segurança.

2.1.1 Armazenamento das políticas

As configurações de GPO são armazenadas em uma estrutura de pastas e arquivos existente nos controladores de domínio da rede. Estas informações são gravadas na pasta SYSVOL e são utilizados por todos os membros existentes do domínio (BATTISTI, 2004).

Na figura 1, visualizamos a estrutura onde ficam gravadas as informações sobre as políticas no domínio abc.com.



Fonte: adaptado de BATTISTI (2003, p. 586)

Figura 1: Estrutura de armazenamento das políticas

Cada pasta representa uma GPO individualmente. Dentro desse local de armazenamento ainda constam os arquivos de *templates* administrativos, *scripts* utilizados, a pasta *machine* onde são armazenados dados que se aplicam ao computador, a pasta *user* onde são armazenadas as informações que se aplicam ao usuário e o arquivo *gpt.ini* que contém informação sobre a versão da GPO (BATTISTI, 2004).

As informações gravadas dentro da pasta SYSVOL são consideradas os *group policy templates* (GPT). A partir do *template* é que são definidas quais as opções que estarão disponibilizadas para serem configuradas via GPO. Conforme Battisti (2003), ao criar uma nova GPO, o Windows 2003 cria uma nova base de templates na pasta SYSVOL. Depois de criada é feito a sua configuração no console *Group Policy Management Console* (GPMC) e armazenada na estrutura de pastas da sua GPO.

2.1.2 Processamento das políticas

Segundo Battisti (2003), a aplicação da GPO é processada na seguinte seqüência. Primeiramente a GPO local, sendo que cada computador possui uma GPO local, onde é aplicada antes das demais GPOs. Por segundo plano são aplicadas as GPOs relacionadas ao site (domínio global - onde o mesmo pode possuir mais de uma sub-rede, a identificação do computador ocorre de acordo com a identificação da máquina). Em terceiro está à aplicação de GPOs associadas a um determinado domínio. Dentro do domínio podem existir diversas GPOs e as mesmas também possuem uma identificação para definição da ordem do processamento. Por último são aplicadas as GPOs associadas a todas as *Organization Units* (OUs). Se existirem mais de uma GPO associada à mesma OU, as GPOs são aplicadas conforme ordem definida pelo administrador.

Conforme Battisti (2003), as políticas aplicadas por último têm procedência sobre as políticas aplicadas anteriormente. Se houver um conflito entre uma GPO aplicada ao domínio e a uma GPO aplicada a uma unidade organizacional (OU), esta última irá prevalecer.

Essa característica é padrão, mas pode ser mudada fazendo com que uma regra inicial se prevaleça perante as outras que virão em subseqüência.

2.1.3 Tipos de GPOs

Atualmente existem dois tipos de *Group Policy Objects*: as relacionadas aos usuários e àquelas destinadas às configurações da máquina.

As GPOs incluem configurações que são aplicadas no nível de usuários e de computador. As políticas de usuários, isto é, políticas associadas à conta do usuário, são as configurações presentes na opção *User Configuration* do console de administração de GPOs e são aplicadas quando o usuário faz o *logon* independente do computador que o mesmo esteja usando. Já as políticas de computador são configuradas através da opção *Computer Configuration* e são aplicadas quando o computador é inicializado.

Segundo Battisti (2003), ao iniciar o computador com Windows 2000, XP Professional ou 2003, pertencente ao domínio da empresa, é feita uma verificação se existem novas políticas ou alterações nas GPOs existentes. Após esse processo é feita a aplicação das

políticas relacionadas ao computador. Já quando um usuário faz o *logon* em algum computador do domínio compatível com o sistema de GPO é feita novamente uma verificação das políticas novas e/ou alterações presentes nas políticas que já existiam. Com isso são processadas as políticas de usuário.

2.1.4 Políticas de usuários

“As configurações relacionadas são questões do sistema operacional, configurações do *desktop*, de segurança, instalação de softwares públicos ou publicados, *scripts* de *login* dos usuários, redirecionamentos de pastas dos usuários.” (RICE, 2001, pg. 79, tradução nossa).

A política de usuário é uma política muito importante para configurações comuns de um determinado grupo de usuários ou de um setor determinado da empresa. Segundo Rice (2001), as políticas de usuários ocorrem ao usuário fazer o *login* no sistema operacional.

No quadro 2 constam as principais configurações possíveis na política de usuários.

POLÍTICAS	DESCRIÇÃO
Instalação de software	É possível fazer a instalação de um software relacionado a um específico usuário fazendo com que o aplicativo seja instalado ao fazer o processo de entrada (<i>login</i>) na máquina.
Configurações do Windows	É possível nessa área fazer a configuração de diversos programas relacionados ao Windows como, Windows Explorer, NetMeeting, Internet Explorer, dentre outros. Além de também ajustes relacionados à área de trabalho, e segurança do seu usuário.
Configurações de segurança	Relacionado às restrições da senha do usuário e aos softwares utilizados.
Scripts	Possibilidade de configurar um <i>script</i> ao fazer o <i>login</i> no sistema e na saída do sistema (<i>logoff</i>). Utilizado para configurações de unidades virtuais do usuário, por exemplo.
Redirecionamento de pastas	Responsável pelo redirecionamento das pastas comuns dos usuários (meus documentos, por exemplo) para pastas pré-determinadas pelo administrador. Normalmente armazenadas num servidor central.

Fonte: adaptado de Rice (2001, p. 73-79)

Quadro 2: Configurações dos usuários

Com essa solução adotada nas políticas de usuários é possível se fazer num tempo mínimo a configuração de diversos usuários, através da normalização e centralização, trazendo segurança ao processo de aplicação de políticas de grupo (BATTISTI, 2003).

2.1.5 Políticas de computadores

“Configurações de computador permite o controle do sistema operacional, do desktop, configurações de segurança, *scripts* de inicialização e desligamento, distribuição de softwares e configurações de aplicativos.” (HANEY, 2001, pg. 14, tradução nossa).

De acordo com Rice (2001), a política de computador é aplicada quando o sistema operacional inicia e durante um período de atualização. A política de computador acaba sendo fundamental na política de distribuição de softwares e configurações das máquinas de determinado setor da empresa.

No quadro 3 são apresentadas as principais configurações possíveis na política de computadores

POLÍTICAS	DESCRIÇÃO
Instalação de software	É possível fazer a instalação de um software relacionado a um grupo de máquinas.
<i>Scripts</i>	Possibilidade de configurar um <i>script</i> ao fazer a inicialização do sistema operacional e na saída do sistema também.
Configurações de Segurança	Nessas configurações estão presentes políticas de segurança dos usuários locais da máquina, políticas de bloqueio relacionadas aos usuários locais, configurações do <i>kerberos</i> , ajustes na auditoria dos objetos (<i>logon, management, directory, privilege</i>), direitos do usuário, ajustes no <i>event log</i> , grupos responsáveis pela administração, serviços do sistema e sua respectiva segurança, atribuições, configurações e segurança do registro e sistema de arquivos do computador e suas restrições e liberações e ajustes sobre a política de chaves públicas.
Configurações do Windows	É possível nessa área fazer a configuração de diversos programas relacionados ao Windows como, Windows Explorer, NetMeeting, Internet Explorer, Agendador de tarefas e Windows Installer.
Configurações de impressora	Configurações na área de impressoras. Permitindo ou não a publicação de impressoras e prioridades de impressão.
Configurações de Rede	Ajustes sobre a parte de redes, presente nos arquivos <i>offline</i> , conexões de rede e <i>dial-up</i> .
Configurações do sistema	Possibilidade de configurações do sistema operacional, desabilitação de botões de desligamento do sistema e de <i>logoff</i> , desligamento de recursos, configurações de <i>scripts</i> , cotas de disco e do cliente <i>DNS</i> além de ajustes para aplicação das políticas de grupo

Fonte: adaptado de Rice (2001, p. 73-79)

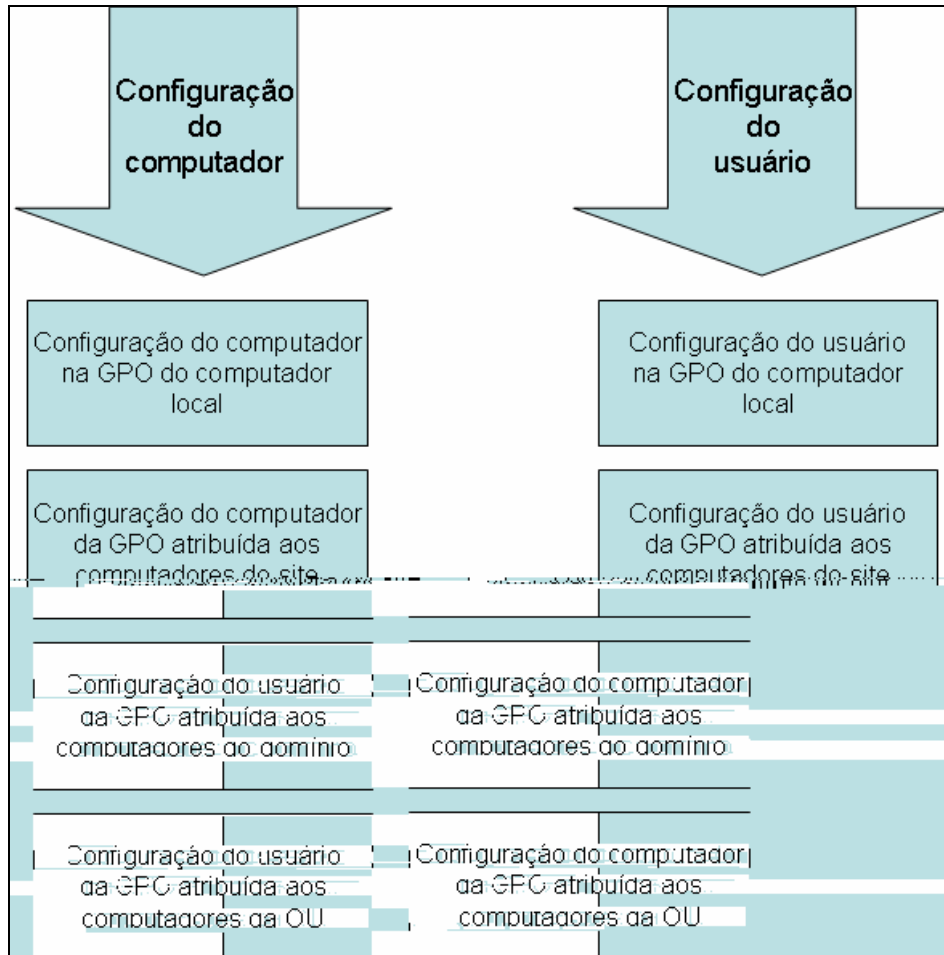
Quadro 3: Configurações dos computadores

2.1.6 Associações, links e herança na GPO

Segundo McIntosh et al (2004), cada computador tem a sua GPO local. É possível que sejam atribuídas diversas GPOs a múltiplos computadores e usuários, em vários níveis da organização do *Active Directory*. As GPOs podem ser atribuídas ao *AD Container*, como nas

unidades organizacionais e também sites inteiros dependendo da estrutura existente no AD.

Múltiplas GPOs podem ser aplicadas a um computador ou usuário, e algumas vezes com isso podem ocorrer conflitos nas configurações. Quando várias GPOs têm a mesma configuração existe certa seqüência a ser seguida, conforme apresentado na figura 2.



Fonte: adaptado de MCINTOSH (2004, p. 10)

Figura 2: Seqüência de aplicação da GPO

De acordo com Mcintosh et al (2004), GPO local do computador fica armazenada em cada computador individualmente. Quando o computador aplica essa é a primeira regra a ser aplicada, independente se é uma política de usuário ou de computador.

Já as GPOs atribuídas a site (site é objeto do AD que representam layout físico de rede), seriam aplicadas na seqüência, normalmente para definir um parâmetro de rede (MCINTOSH et al, 2004).

Quando se utiliza a GPO aplicada a um domínio, essa é atribuída a todos os usuários e/ou computadores do domínio escolhido. Segundo Mcintosh (2004), não é recomendável a utilização desse tipo de GPO em domínios filhos, normalmente presente em grandes corporações.

Por último, as políticas aplicadas aos usuários e computadores de uma OU específica. Se caso houver várias OUs dentro da OU, a GPO será aplicada a todas as unidades organizacionais filhas. As políticas são aplicadas das maiores organizações até os menores (MCINTOSH et al, 2004).

Segundo Battisti (2003), as políticas são aplicadas a partir do objeto pai (raiz do domínio), passando pelas diversas topologias existentes no Active Directory. Se você associar uma determinada GPO a um elemento do AD, as configurações serão aplicadas a todos os objetos que estiverem contidos no objeto filho.

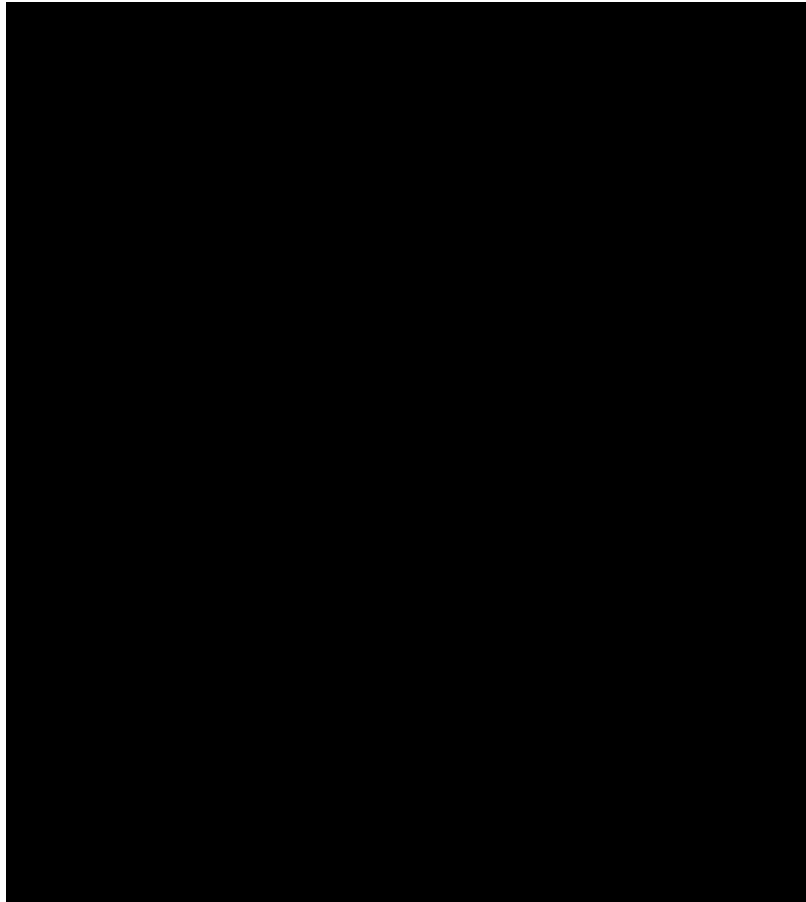
Segundo Mcintosh et al (2004), ao se aplicar uma GPO em um domínio, é aplicada a todos os usuários e máquinas daquele domínio, através de herança, independente da estrutura presente no AD. Mas essas heranças podem ser lidadas através do comando Block Policy Inheritance existente no Group Policy Management Console (GPMC). O bloqueio pode ser definido em nível de domínio ou até uma unidade organizacional. Também é possível que uma regra seja aplicada de qualquer maneira independente da sua herança ou de conflitos com outras regras, a GPO será passada para todas as máquinas abaixo, através do comando Enforced presente no console de administração, fazendo com que a regra seja aplicada mesmo que existam outras GPO em conflito. E por fim, criando a desabilitação de uma GPO através do GPMC pelo comando Disabled, facilitando assim a desativação de regras.

As GPOs consistem de duas partes, conforme já apontado, a de usuários e a de computadores. É possível através do console de administração configurar a desabilitação de usuário ou de computador numa determinada GPO, fazendo que somente seja processado aquilo que estiver habilitado. Esse tipo de configuração é bem útil em ambientes com OUs separadas de usuários e computadores. Desabilitando metade da GPO que não faz com que o processamento das GPOs seja mais eficiente e mais rápido, segundo Battisti (2003).

2.1.7 Console de administração das políticas

Segundo Battisti (2004), existe uma console específica para criação, configuração e administração das políticas de grupo. Esse console é acessado pelo administrador. O console é acessado pelas ferramentas administrativas presentes no painel de controle do Windows 2003 Server. A ferramenta tem o nome de Usuários e Computadores do Active Directory. Após seleção de uma unidade organizacional ou domínio é possível acessar o

console de administração das políticas através das propriedades daquele objeto. Na guia diretiva de grupo serão apresentados quais são as GPOs existentes para aquele objeto, conforme figura 3.



Fonte: adaptado de BATTISTI (2004, p. 593)

Figura 3: Console de administração das GPOs

Conforme Battisti (2004), após selecionar a GPO, você pode estar fazendo a configuração da mesma, utilizando o editor de objeto diretiva de grupo. Com ele é possível fazer a configuração das diretivas para os usuários e computadores, conforme na figura 4.



Fonte: adaptado de BATTISTI (2004, p. 594)

Figura 4: Console de edição de uma determinada GPO

Após a sua configuração, a GPO já estará sendo aplicada para o objeto ao qual foi associado.

2.2 FILTROS WMI

Segundo Wikipédia (2006), a instrumentação da geração Windows (WMI) é um jogo de extensões que o Windows fornece para estabelecer uma comunicação com o sistema, através de componentes instrumentados para fornecer informações e notificações. WMI é instalado como padrão nos Windows Vista, 2000, 2003, ME. No Windows 98, 95, NT é possível a instalação do WMI através de software de terceiros. O filtro possibilita linguagens de script como VBScript que podem gerenciar computadores pessoais e servidores em sistemas operacionais compatíveis, podendo ainda ser executado localmente ou remotamente.

WMI é uma implementação da Microsoft da iniciativa Common Information Model (CIM) desenvolvida pela Distributed Management Task Force (DMTF). O DMTF é uma associação de vários padrões das companhias de hardware e software (por exemplo, Novell, Microsoft, Cisco, etc.) que desenvolveram padrões para a gestão da empresa (tradução nossa, WIKIPEDIA, 2006).

Conforme Cooperstein (2000), os objetivos desta iniciativa são de desenvolver a tecnologia padrão para acesso a informação na gestão de um ambiente da empresa, e permitir o desenvolvimento de ferramentas e técnicas que reduzam a complexidade e

custos da gerência da empresa. Através desse é possível baixar o Total Cost of Ownership (TCO). O TCO consulta os custos administrativos associados a aquisição de software de computadores, distribuição e configuração de ferramentas e softwares, treinamento, manutenção e sustentação técnica.

Conforme Wikipédia (2006), as grandes empresas têm muitos computadores com muitos ambientes de software, e controlar esses sistemas diversos pode ser um desafio real. A fim de unificar as técnicas de gerência para a aplicação, o DMTF definiu o CIM para representar entidades que podem ser gerenciadas de maneira unificada. O modelo do objeto CIM é usado dos dados do objeto e um modelo lógico que é único para todos os construtores e colaboradores de software. Esse modelo é executado em uma base de dados chamada repositório do CIM. A figura 5 apresenta como funciona a arquitetura do WMI, acessando o CIM e trazendo o resultado para os clientes que administram as máquinas.

Fonte: Cooperstein (2000)

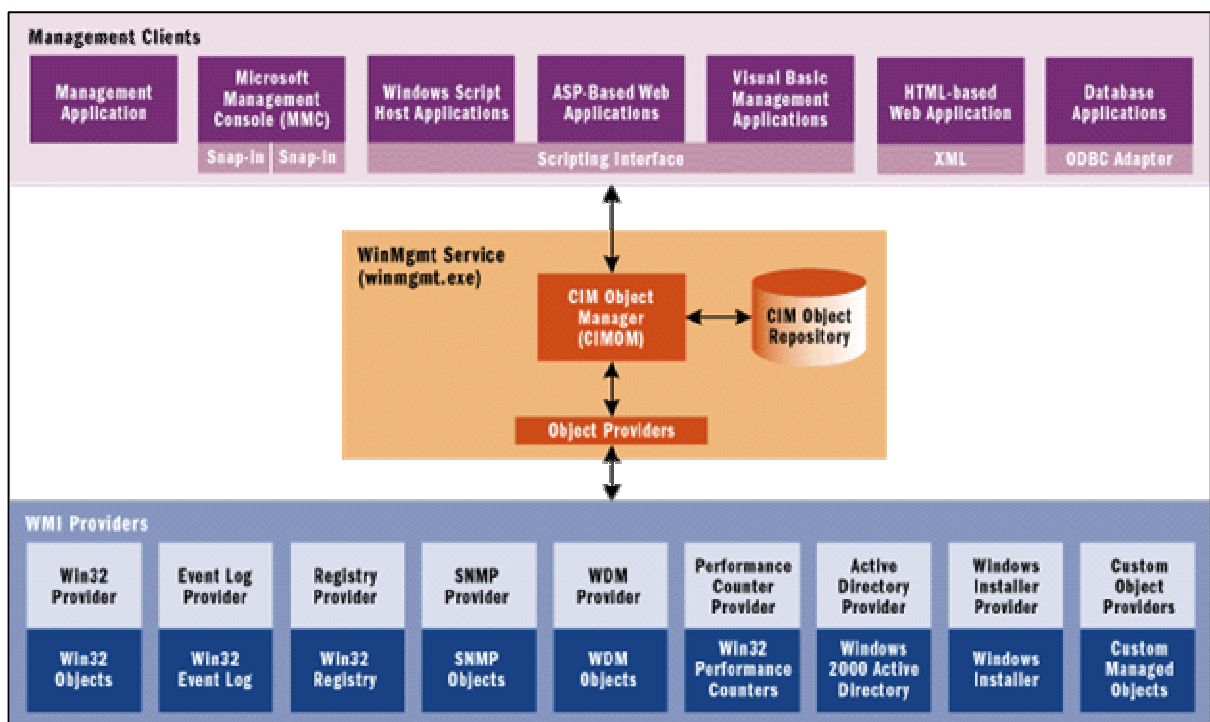


Figura 5: Diagrama da arquitetura de funcionamento do WMI e CIM

Segundo Wikipédia (2006), baseado no modelo CIM, o WMI inclui os componentes gerenciáveis disponíveis dos padrões DMTF com algumas extensões específicas que representam os vários componentes do Windows. Além disso, o WMI expõe uma gama de objetos COM que permitem que várias aplicações façam as informações para o gerenciamento da rede. WMI é a execução da Microsoft do CIM.

Conforme Cooperstein (2000), o WMI, é baseado no CIM, se utiliza da Managed Object Format (MOF) e de uma interface de programação comum. O MOF é usado para definir a estrutura e índices do esquema do CIM.

Segundo Cooperstein (2000), a tecnologia Windows Management Instrumentation consegue fornecer:

- a) acesso ao monitor, comando, controle em toda a rede com um comum, independente do mecanismo da instrumentação. O WMI mecanismo de acesso;
- b) um modelo de operação do Windows, uma configuração, status logicamente organizados, consistentes;
- c) uma relação de programação de aplicação de COM (para a informação fornecida e o acesso informação para a gerência;
- d) operacionalidade com outros serviços de gerência Windows. Esta aproximação possibilita e simplifica esforços dos colaboradores para criar uma solução de gerência integrada e bem arquitetada;
- e) uma arquitetura muito flexível. Isso acaba permitindo que os vendedores estendam o modelo de informação para cobrir objetos como dispositivos e aplicações;
- f) extensões para o modelo Windows Driver Mode (WDM) para capturar dados de instrumentação e eventos dos drivers dos dispositivos e dos componentes do kernel-side;
- g) um poderoso mecanismo que permite que os eventos de gerência sejam afirmados e associados a outras informações da gerência. Estas informações podem ser enviadas as aplicações de gerência local ou remotas;
- h) uma rica linguagem que permite perguntas detalhadas no modelo da informação.

Como parte de um processo da instalação e do Deployment dos softwares de hoje disponível pela Microsoft (SQL Server 2005, Internet Explorer 7.0, Office 2003/2007) estendem o modelo padrão do objeto do CIM para a representação de seus objetos no repositório CIM. Esta representação é denominada classe, e expõe a informação através das propriedades e permite a execução de algumas através de métodos.

Os filtros WMI consistem na WMI Query Language (WQL), onde existem diversas classes no repositório do CIM que retornam informações sobre o processador, memória virtual e física, espaço de disco, softwares instalados, service packs, aplicações instaladas, processos que estão rodando e a lista segue ainda com outras informações (ALLEN, 2003, p. 295).

De acordo com Melber (2004), WMI é algo novo e começou a se tornar muito popular por controlarem muitos aspectos do ambiente Active Directory. Quando um filtro WMI é associado a uma GPO, será feita uma melhor adequação, quais objetos receberão as configurações de uma determinada GPO através de um determinado. É possível a utilização de vários critérios num filtro único.

O filtro funciona através do WQL. Se o valor de um atributo é válido conforme a informação requisitada pelo filtro, a GPO será aplicada. Se o valor não for válido ou igual ao proposto pelo filtro, a GPO não será aplicada.

Conforme Melber (2004), filtros WMI são excelentes para aplicação de software, configurações de Internet Explorer, e outras configurações ligadas diretamente ao sistema operacional, requerimentos de hardware ou software e outras variáveis de ambiente.

Segundo Melber (2004), os filtros WMI são arquivados na parte das GPO e são referenciados para determinada GPO através do Group Policy Management Console (GPMC).

Uma restrição importante é que somente pode ser criado um filtro por GPO.

2.2.1 Arquitetura do WMI

Conforme Cooperstein (2000), a infra-estrutura de gerenciamento consiste no gerente de objetos CIM e no repositório do objeto CIMOM. As aplicações dependem do gerente de objetos para segurar a relação aplicações de gerência e fornecedores dos dados. O gerente do objeto do CIM facilita estas comunicações fornecendo uma relação de programação comum, usando a COM, para o WMI. Esta API COM fornece a definição e a pergunta do evento que processam serviços, e está disponível em diversas linguagens de programação. O WMI também suporta automatização com scripts em VBScript, em JScript e em Perl, assim como em outras linguagens de scripts suportadas pelo Windows.

O repositório do objeto do CIMOM prende os esquemas CIM e da extensão e a informação dos dados ou detalhes da origem dos dados. O gerente de objetos do CIM usa os dados do repositório ao prestar serviços de manutenção pedidos das aplicações da gerência para objetos controlados, conforme Cooperstein (2000).

Segundo Cooperstein (2000), os objetos controlados são componentes físicos ou lógicos que são modelos usando o CIM e o DMTF. Por exemplo, um objeto controlado pode ser um hardware tal como um cabo ou alguma parte do sistema, o software tal como um

aplicação de banco de dados, ou outras entidades tais como os sistemas computacionais, os arquivos e os dispositivos. Aplicações de gerência podem alcançar objetos controlados através do CIM.

As aplicações da gerência são aplicações ou servidores que processam ou indicam dados dos objetos controlados ou mudam dados dos objetos controlados (COOPERSTEIN, 2000).

Conforme Cooperstein (2000), os fornecedores de WMI funcionam como intermediários entre o gerente do objeto do CIM e os objetos controlados. Usando as APIs WMI, os fornecedores fornecem ao gerente do CIM os dados dos objetos controlados, seguram pedidos em nome da aplicação de gerência, e geram notificações dos eventos, conforme ilustrado na figura 5.

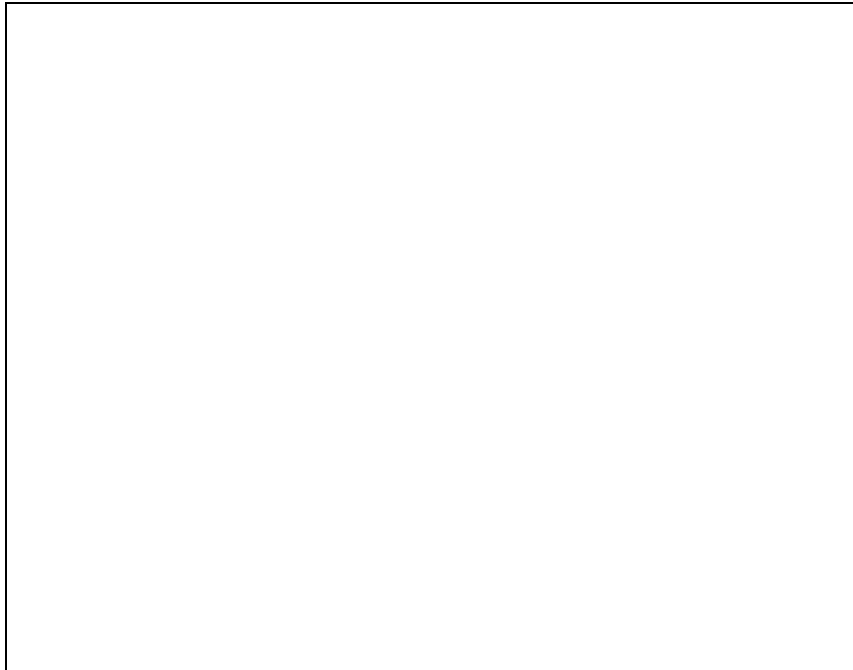
Segundo Cooperstein (2000), as aplicações alcançam diretamente as relações com a COM para interagir com o gerente do objeto CIM e fazer os pedidos para a gerência quando usarem métodos de acesso adicionais como o Open Database Connectivity (ODBC), Active Directory Service Interface (ADSI), ou o script WMI API, previamente chamado de WBEM, para fazer seus pedidos. A parte mais inferior da figura 5 mostra exemplos de objetos controlados e de fornecedores associados (tais como o registro do Windows e o seu fornecedor). Uma comunicação entre o gerente do objeto CIM em componentes locais e remotos usa o modelo DCOM.

2.2.2 Visão geral sobre o WBEM

Segundo Microsoft (2003), Web-based Enterprise Management (WBEM) foi uma iniciativa da BMC Software Inc, Cisco Systems, Compaq Computer Corporation, Intel Corporation e pela Microsoft Corporation.

De acordo com Microsoft (2000), a WBEM é uma iniciativa da indústria para desenvolver um padrão para compartilhar as informações da gerência em uma rede de uma empresa. O WBEM resultará na tecnologia que permite a coleta, associação e agregação de dados de gerência de diversas fontes, para assumir uma fonte rica e com informações exatas para os seus ambientes. A iniciativa da WBEM pretende resolver o problema de coleta de dados de diagnósticos nas redes da empresa que incluem hardware de múltiplas marcas, numerosos protocolos e sistemas operacionais, e a integração de aplicações

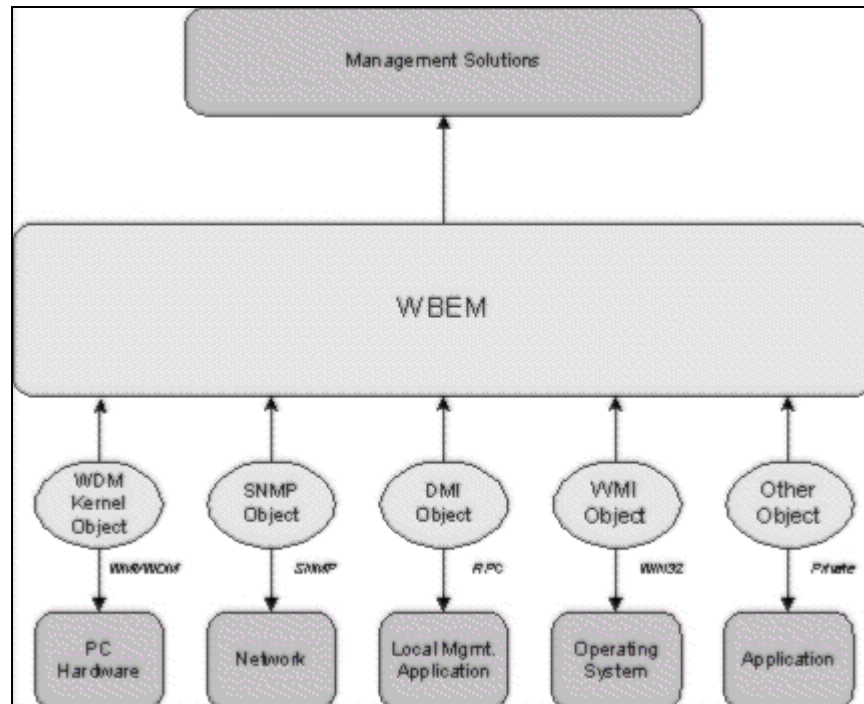
distribuídas, como ilustrado na figura 6.



Fonte: Microsoft (2000)

Figura 6: Protocolos e interfaces em uma rede de uma empresa.

Conforme Microsoft (2000), a gerência da rede da empresa foi amarrada aos protocolos e às relações diferentes para tipos diferentes como, por exemplo, o *Simple Network Management Protocol* (SNMP) é usado para gerenciar redes e a *Desktop Management Interface* (DMI) que é usada para a gerência de sistemas *desktop*. O WBEM acredita que a gerência de rede da empresa requer que as ferramentas trabalhem juntas para fornecer um único modelo compartilhado para a coleção de informação. O WBEM fornece esta origem de dados comuns ao modelo e pode ser estendido ao trabalho com componentes, ferramentas e protocolos existentes da rede. A figura 7 demonstra a estrutura padronizada do WBEM para a gerência de rede de uma empresa.



Fonte: Microsoft (2000)

Figura 7: Arquitetura padronizada do WBEM.

Em resumo, o WBEM não é ferramenta de interface para o usuário, um repositório de dados, um protocolo de gerência de redes, um modelo de componente, ou um registro, um diretório ou uma recolocação do sistema de arquivos. WBEM é uma iniciativa para controlar a rede da empresa, como mostra a figura 7.

2.3 UM FILTRO WMI A UMA GPO

Segundo Computer Performace (2007), os filtros de WMI permitem que se selecione somente os computadores que se encontram com seus critérios escolhidos. Naturalmente, sua política de grupo é aplicada somente aos objetos que combinam com seu filtro. Conforme Microsoft (2005), ao colocar uma política GPO aplicada a um filtro WMI e ela é processada pelo computador, se o filtro WMI retorna como falso, a política não é aplicada. Já se o retorno do filtro WMI é verdadeiro, a GPO é aplicada.

Após a criação da GPO é possível atribuir um filtro WMI a uma determinada GPO. Essa função é feita através do console de administração das diretivas de grupos apresentado anteriormente, o GPMC. Conforme Battisti (2004) é possível selecionar o filtro WMI e após sua seleção o filtro WMI será vinculado a uma determinada GPO.

A figura 8 apresenta uma GPO com um filtro atribuído.



Fonte: Computer Performance (2007)

Figura 8: GPO com filtro WMI aplicado.

No quadro 4 são mostrados alguns exemplos de filtros WMI que podem ser atribuídos a diretiva de grupo.

Intenção do filtro	Filtro WMI
Aplicar política nos computadores que estão localizados na costa leste dos Estados Unidos	Root\cimv2 ; select * from win32_timezone where bias = -300
Aplicar política nos computadores que contém um <i>hotfix</i> específico	Root\cimv2 ; select * from win32_quickfixengineering where HotfixID = 'XX'
Aplicar diretiva específica para computadores com Windows XP Professional	Root\cimv2 ; select * from win32_operatingsystem where caption = "Microsoft Windows XP Professional"
Aplicação de GPO em micros com fabricante específico	Root\cimv2 ; select * from win32_computersystem where manufacturer = "XPTO"

Fonte: adaptado de Microsoft (2005)

Quadro 4: Exemplos de filtros WMI

2.4 TRABALHOS CORRELATOS

A *Group Policy Objects* em ambientes com Windows Server vem sendo bastante difundida, mas devido aos poucos que conhecem o tema profundamente, são poucos os trabalhos que tratam sobre GPOs. Porém, existem algumas pesquisas que exploram os princípios básicos do assunto.

Evenson (2003) elaborou um artigo que explica os impactos das políticas de grupo na área de segurança, trazendo a fundamentação sobre o assunto e uma abordagem bem profunda sobre a área de segurança.

E Ferreira (2003) demonstrou em seu artigo a grande importância do trabalho de segurança em cima do Windows Server através da utilização de *Group Policy Objects*.

Todos os trabalhos citados têm relação com o aqui proposto, pois se utilizam das políticas de grupo e conceitos apresentados e desenvolvidos no software de geração de filtros WMI. Mas nenhum deles explora a capacidade e potencial do WMI em junção com as GPOs.

3 DESENVOLVIMENTO DO TRABALHO

Esta seção aborda a metodologia de desenvolvimento do trabalho. O tópico inicial descreve os requisitos atendidos pelo software. O próximo tópico refere-se à especificação do software, onde são apresentados alguns diagramas UML (*Unified Modeling Language*) visando uma melhor compreensão do software. O tópico seguinte refere-se à implementação do software, onde está descrito seu funcionamento, tecnologias utilizadas, operacionalidade e resultados obtidos.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

No quadro 5 são apresentados os requisitos funcionais previstos para o sistema e sua rastreabilidade, ou seja, vinculação com o(s) caso(s) de uso associado(s).

Requisitos Funcionais	Caso de Uso
RF01: O software deverá permitir ao usuário o cadastramento dos filtros WMI.	UC01
RF02: O software deverá permitir ao usuário escolher quais filtros serão exportados para o GPMC.	UC02
RF03: O software deverá permitir ao usuário a remoção do filtro WMI selecionado.	UC03
RF04: O software possibilitará o salvamento desses filtros em arquivo XML.	UC04
RF05: O software permitirá ao usuário a importação de filtros WMI salvos no computador em formato XML.	UC05

Quadro 5: Requisitos funcionais

No quadro 6 são listados os requisitos não funcionais previstos para o software.

Requisitos Não Funcionais
RNF01: O software se utilizará de linguagem de programação PHP.
RNF02: O software deverá interagir com o <i>Group Policy Management Console</i> , já implantado

no <i>Active Directory</i> no Windows.
RNF03: O software deverá utilizar a linguagem de exportação e/ou importação como padrão o XML.
RNF04: O software deverá possuir uma interface simples e com uma boa navegabilidade.
RNF05: O software deverá ser totalmente via navegador, para facilitar ainda mais a administração dos filtros WMI.

Quadro 6: Requisitos não funcionais

3.2 ESPECIFICAÇÃO

Esta seção descreve os modelos e diagramas desenvolvidos durante o trabalho. Os primeiros tópicos tratam, respectivamente, os diagramas UML de caso de uso e de atividades. Ambos foram desenvolvidos utilizando o *Enterprise Architect 6.07*.

3.2.1 Diagrama de caso de uso

A figura 09 representa o diagrama de casos de uso. Este diagrama faz referência aos requisitos funcionais do software. Cada pacote representa as funcionalidades exclusivas de um ator. A descrição dos cenários encontra-se na seção Apêndice A.

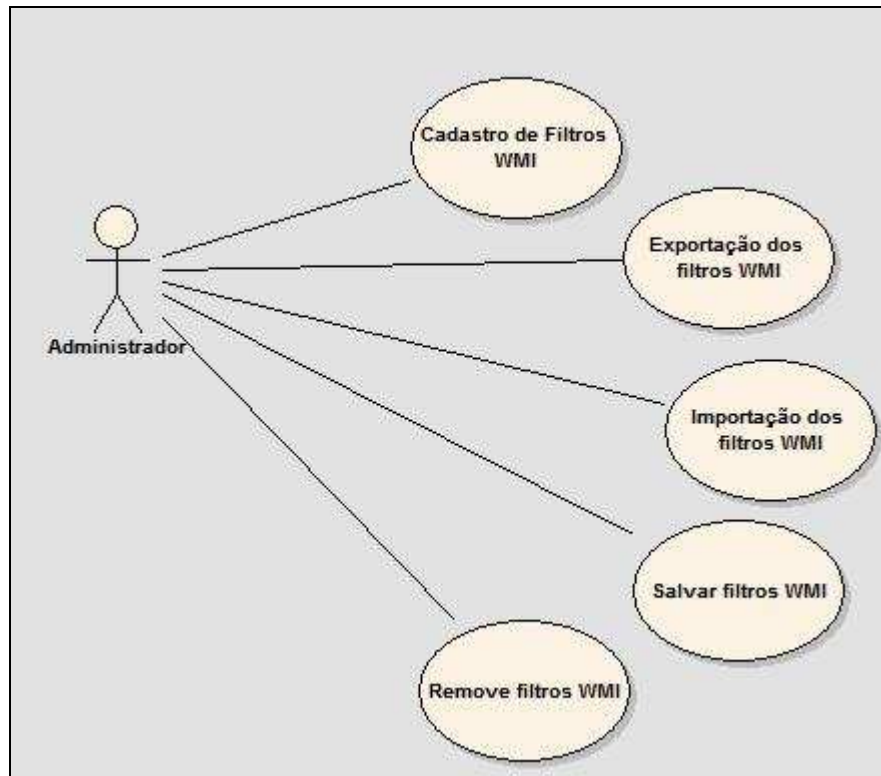


Figura 9: Diagrama de casos de uso.

A figura 10 representa o diagrama de pacotes. Esse diagrama faz referência aos requisitos funcionais do software. Cada pacote representa as funcionalidades de cada ator, exceto o pacote geral que contém funcionalidades comuns realizadas por mais de um ator.



Figura 10: Diagrama de pacotes.

3.2.2 Diagrama de atividades

A figura 11 representa o diagrama de atividades. Este diagrama faz referência às ações tomadas pelo usuário para chegar ao cenário final, sendo o processo de adicionar um filtro ao sistema e sua exportação para a console do GPMC e sua referida importação com

sucesso na console da Microsoft.

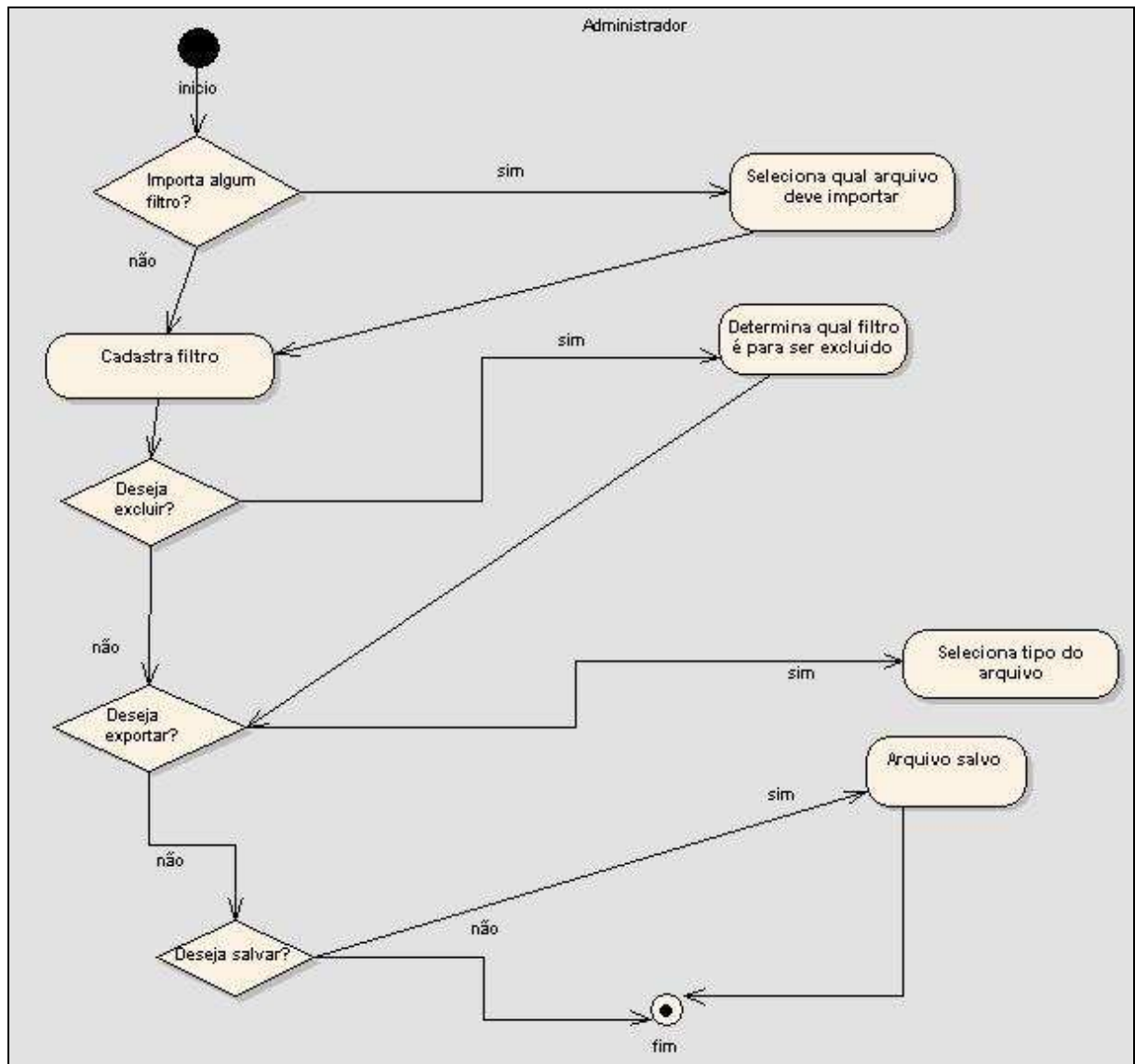


Figura 11: Diagrama de atividades.

3.2.3 Diagrama de classes

A figura 12 apresenta as principais classes do software desenvolvido, objetivando uma visão do domínio do problema.

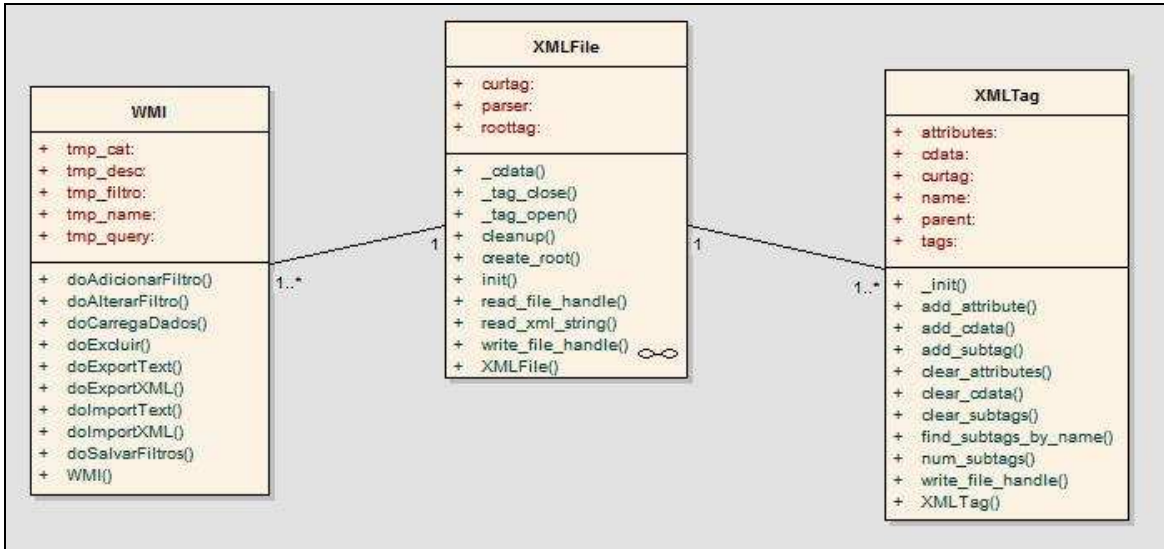


Figura 12: Diagrama de classes.

3.2.4 Diagrama do XML

A figura 13 apresenta o diagrama de *grid* do XML, que é responsável pela configuração das classes que serão apresentadas ao usuário final no sistema. Esse diagrama foi gerado pela ferramenta *XMLSPY*.

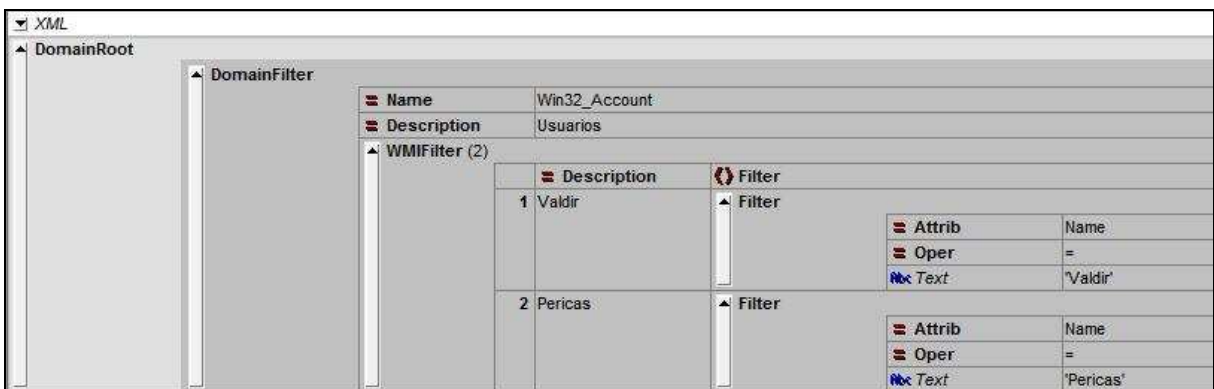


Figura 13: Diagrama de *grid* do XML.

Na figura 14 é demonstrado o diagrama do XML de forma textual utilizando o mesmo padrão do diagrama anterior.

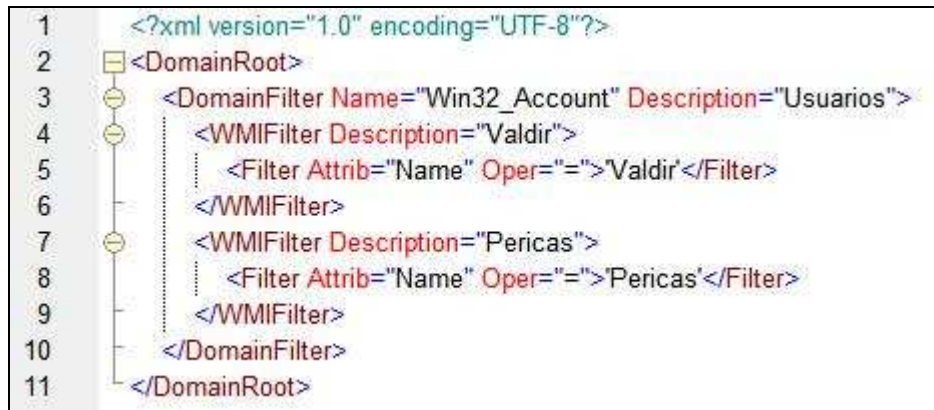


Figura 14: Diagrama do XML.

3.3 IMPLEMENTAÇÃO

Esta seção contém o detalhamento sobre a implementação do software. O tópico inicial identifica as técnicas e ferramentas utilizadas. O tópico seguinte apresenta um estudo de caso do ponto de vista do usuário, destacando a funcionalidade ou operacionalidade do software. O último tópico descreve os resultados obtidos.

3.3.1 Técnicas e ferramentas utilizadas

O software implementado faz uso das tecnologias atuais para desenvolvimento de sistemas *web*, como o PHP 5.0 para a codificação e acesso a informações gravadas no XML e outros recursos como *Hypertext Markup Language*, o HTML, *Javascript* e CSS para a interface e navegação do sistema. Além, também, de técnicas utilizando AJAX. Nota-se que todas as tecnologias utilizadas no desenvolvimento deste software são do tipo *Freeware* (Software Livre) ou *Open Source* (Código Aberto).

PHP é uma linguagem de *script* que pode ser acoplada ao HTML, permitindo a construção de páginas *web* dinamicamente. Os scripts PHP são interpretados no servidor da aplicação *web*. A versão 5 do PHP utiliza uma nova *engine* denominada *Zend Engine II*. O

manuseio de objetos nesta nova *engine* foi totalmente reescrito, possibilitando o uso de novos recursos em OO (Orientação a Objeto) (ZEND, 2004).

Cascading Style Sheets (CSS) é uma linguagem desenvolvida pela W3C (*World Wide Web Consortium*) que oferece um controle visual nas apresentações de páginas *web*. O CSS pode ser acoplado ao HTML, permitindo a inclusão de efeitos visuais baseado em eventos. São vários os navegadores e programas que suportam a linguagem. O CSS pode ser utilizado também em dispositivos móveis como telefones, PDAs (*Personal Digital Assistants*) e televisores à pilha (W3C, 2004).

As figuras 15, 16, 17 e 18 apresentam o CSS utilizado pelo software.

```
1
2 #dhtmltooltip{
3 position: absolute;
4 width: 150px;
5 border: 1px solid black;
6 padding: 2px;
7 background-color: lightyellow;
8 visibility: hidden;
9 z-index: 100;
10 font-family: Arial, Helvetica, sans-serif;
11 font-size: 11px;
12
13 /*Remove below line to remove shadow. Below line should always appear last within this CSS*/
14 filter: progid:DXImageTransform.Microsoft.Shadow(color=gray,direction=135);
15 }
```

Figura 15: Arquivo do CSS usado pelo software.

```
1 #divLoading, #divImportExport {
2     left: 0px;
3     width: 100%;
4     top: 0px;
5     height: 100%;
6     background-color: #FFFFFF;
7     position: absolute;
8     z-index: 2;
9     filter: alpha(opacity=80);
10    opacity: .8;
11 }
12
13 #divDadosImportExport {
14     position: absolute;
15     width: 326px;
16     height: 185px;
17     z-index: 3;
18     left: 50%;
19     margin-left: -163px;
20     top: 180px;
21     background-image: url(../images/bg/bg_imp_exp.gif);
22     background-repeat: no-repeat;
23 }
24
25 .center {
26     height: 32px;
27     width: 32px;
28     left: 50%;
29     margin-left: -16px;
30     top: 200px;
31     position: relative;
32 }
33
34 /* Botao */
35 .btn {
36     font-family: Arial, Helvetica, sans-serif;
37     font-size: 11px;
38     font-weight: bold;
39     text-decoration: none;
40     border-bottom: 1px solid #D7D7D7;
```

Figura 16: Arquivo do CSS usado pelo software (continuação)

```
34 /* Botao */
35 .btn {
36     font-family: Arial, Helvetica, sans-serif;
37     font-size: 11px;
38     font-weight: bold;
39     text-decoration: none;
40     border-bottom: 1px solid #D7D7D7;
41     border-right: 1px solid #C7C7C7;
42     border-top: 1px solid #DBDBDE;
43     border-left: 1px solid #DBDBDE;
44     background-color: #FFFFFF;
45     height: 23px;
46     background-image: url(../images/btn/fundo_btn.jpg);
47     cursor: pointer;
48 }
49
50 /* Form */
51 .frmCampo {
52     font-family: Arial, Helvetica, sans-serif;
53     font-size: 11px;
54     color: #000000;
55     border-right: #d4d0c8 1px solid;
56     border-top: #656462 1px solid;
57     border-left: #656462 1px solid;
58     border-bottom: #d4d0c8 1px solid;
59 }
60
61 /* Textos */
62 .titNormal {
63     font-family: Arial, Helvetica, sans-serif;
64     font-size: 12px;
65     font-weight: bold;
66     color: #000000;
67 }
68
69 .titWhite {
70     font-family: Arial, Helvetica, sans-serif;
71     font-size: 13px;
72     font-weight: bold;
73     color: #FFFFFF;
```

Figura 17: Arquivo do CSS usado pelo software (continuação)

```
61 /* Textos */
62 .titNormal {
63     font-family: Arial, Helvetica, sans-serif;
64     font-size: 12px;
65     font-weight: bold;
66     color: #000000;
67 }
68
69 .titWhite {
70     font-family: Arial, Helvetica, sans-serif;
71     font-size: 13px;
72     font-weight: bold;
73     color: #FFFFFF;
74 }
75
76 .txtNormal {
77     font-family: Arial, Helvetica, sans-serif;
78     font-size: 12px;
79     color: #000000;
80 }
81
82 /* Links */
83 .linkNormal, a.linkNormal:link, a.linkNormal:visited, a.linkNormal:active {
84     font-family: Arial, Helvetica, sans-serif;
85     font-size: 12px;
86     color: #000000;
87     text-decoration: none;
88 }
89
90 a.linkNormal:hover {
91     text-decoration: underline;
92 }
```

Figura 18: Arquivo do CSS usado pelo software (continuação)

O AJAX é o uso sistemático de *Javascript* e XML (e derivados) para tornar o navegador mais interativo com o usuário, melhorando ainda mais a navegabilidade do software. O AJAX torna a construção de aplicações web mais dinâmicas e criativas. (WIKIPEDIA, 2006). Nas figuras 19 e 20 é apresentada a estrutura do AJAX utilizado pelo software.

```

1  function openAjax() {
2
3      if (window.XMLHttpRequest) { // Mozilla, Safari,...
4          ajax = new XMLHttpRequest();
5          if (ajax.overrideMimeType) {
6              ajax.overrideMimeType('text/xml');
7          }
8      }
9      else if (window.ActiveXObject) { // IE
10
11         try {
12             ajax = new ActiveXObject("Msxml2.XMLHTTP");
13         }
14         catch (e) {
15             try {
16                 ajax = new ActiveXObject("Microsoft.XMLHTTP");
17             }
18             catch (e) {}
19         }
20     }
21
22     return ajax;
23 }
24
25 function doAjaxReplace(responseTextAjax) {
26
27     responseTextAjax = unescape(responseTextAjax.replace(/\\+/g, " "));
28     return responseTextAjax;
29 }
30 }

```

Figura 19: Arquivo responsável pela criação dos objetos do AJAX utilizado pelo software.

```

105 // Busca dados para serem alterados.
106 function doAlterar(id) {
107
108     doCarregando(true);
109
110     var obAjax = openAjax();
111
112     obAjax.open("GET", "inc/inc_carregar_filtro.php?ID=" + id, true);
113     obAjax.send(null);
114
115     obAjax.onreadystatechange = function () {
116         if (obAjax.readyState == 4) {
117             if (obAjax.status == 200) {
118
119                 // Valida retorno
120                 eval(obAjax.responseText);
121
122                 // Esconde o loading
123                 doCarregando(false);

```

Figura 20: Arquivo do AJAX usado pelo software.

Antes de explicar as técnicas utilizadas na implementação do software, é importante descrever a sua estrutura de arquivos. A estrutura está representada na figura 21.

Nome	Modificado em	Tipo	Tamanho
class	01/01/2004 00:05	Pasta de Arquivos	
css	01/01/2004 00:05	Pasta de Arquivos	
images	01/01/2004 00:05	Pasta de Arquivos	
inc	01/01/2004 00:05	Pasta de Arquivos	
js	01/01/2004 00:05	Pasta de Arquivos	
tmp	01/01/2004 00:05	Pasta de Arquivos	
xml	01/01/2004 00:05	Pasta de Arquivos	
index	09/05/2007 13:14	PHP Script	22 KB

Figura 21: Estrutura dos arquivos.

Nota-se a existência de somente um arquivo na pasta principal. O arquivo do tipo PHP, refere-se à página principal do software. A estrutura de pastas do software ficou dividida em sete subpastas. Cada subpasta armazena os arquivos correspondentes a seu nome, exceto a pasta “tmp” que contém arquivos para finalidade de manutenção do software. O arquivo `index.php` é onde consta o software em si.

O arquivo `index.php` possui diversas *functions* que serão demonstrados nas figuras abaixo, mostrando como a implementação ocorreu.

```

28 // Mostra barra de carregando.
29 function doCarregando(flag) {
30
31     var exibir = '';
32     if (flag == false) {
33         exibir = 'none';
34     }
35
36     var listObj = document.getElementsByTagName('input');
37     |
38     // Ativa/Desativa todos elementos INPUT do documento
39     for (cnt = 0; cnt < listObj.length; cnt++) {
40         listObj[cnt].disabled = flag;
41     }
42     // Exibe/Oculta o loading
43     getObj('divLoading').style.display = exibir;
44
45 }
46
47 // Limpa campos de cadastro
48 function doCancelar() {
49     getObj('DESCRIPTION').value = '';
50     getObj('NAME').value = '';
51     getObj('ID_ALT').value = '';
52     getObj("lblIndWrite").style.visibility = 'visible';
53     getObj("Alterar").style.display = 'none';
54     getObj("Acicionar").style.display = '';
55     getObj('WMIFilter').value = "";
56     getObj('IND_WRITE').checked = false;
57
58     obFilter      = getObj("WMIFilterWhere");
59     obFilter.length=0;
60
61     var newElem    = document.createElement("OPTION");
62     newElem.value  = '';
63     newElem.text   = '';

```

Figura 22: *Function* com o carregamento e o cancelar.

Na próxima figura a *function* é utilizada para fazer a validação dos dados no software, para que posteriormente possa ser inserido num arquivo ou exportado para a console do GPMC. Na figura 23 é demonstrado o código-fonte responsável por essa função estar em funcionamento.


```

84 // Validação dos dados
85 function doAdicionar(action) {
86
87     obName     = getObj('NAME');
88     obDesc     = getObj('DESCRIPTION');
89     obCat      = getObj('WMIFilter');
90     obFiltro   = getObj('WMIFilterWhere');
91     obWrite    = getObj('IND_WRITE');
92
93     if (doValidInput(obName, 'Informe a Descrição')) {
94         if (doValidInput(obDesc, 'Informe o Grupo')) {
95             if (doValidInput(obCat, 'Informe a categoria')) {
96                 if (doValidInput(obFiltro, 'Informe o filtro')) {
97                     doAjaxAdicionar(action, obName, obDesc, obCat, obFiltro, obWrite);
98                 }
99             }
100         }
101     }
102 }
103 }

```

Figura 23: *Function* com a validação dos dados.

A *function* “doAjaxAdicionar” é responsável por cadastrar as novas *queries* no software (chamando outras rotinas que fazem o inserção no software), limpar os dados do formulário e carregá-los na lista de filtros cadastrados no software. Na figura 24 é demonstrado o código-fonte dessa função.

```

131 // Cadastra/Altera a nova Query
132 function doAjaxAdicionar(action, obName, obDesc, obCat, obFiltro, obWrite) {
133
134     var obAjax = openAjax();
135
136     ind_checked = 0;
137     if (obWrite.checked == true) {
138         ind_checked = 1;
139     }
140
141     if (action == "CADASTRAR") {
142         obAjax.open("GET", "inc/inc_adicionar.php?NAME=" + obName.value + "&DESCRIPTION=" + obDesc.value + "&CAT=" + obCat.value +
143             "&FILTRO=" + obFiltro.value + '&IND_WRITE=' + ind_checked, true);
144     }
145     else {
146         var id = getObj('ID_ALT').value;
147         obAjax.open("GET", "inc/inc_alterar.php?NAME=" + obName.value + "&DESCRIPTION=" + obDesc.value + "&CAT=" + obCat.value +
148             "&FILTRO=" + obFiltro.value + '&ID=' + id, true);
149     }
150
151     obAjax.send(null);
152
153     doCarregando(true);
154
155     obAjax.onreadystatechange = function () {
156         if (obAjax.readyState == 4) {
157             if (obAjax.status == 200) {
158
159                 // Limpa dados do formulário
160                 doCancelar();
161                 // Valida dados da resposta do AJAX
162                 eval(obAjax.responseText);
163                 // Exibe a listagem das classes
164                 doCarregarFiltros();
165             }
166         }
167     };
168 }

```

Figura 24: *Function* com a adição de filtros ao software.

Na figura 25 é mostrado parte do código da *function* “doSalvarDados” responsável por salvar os dados do software num arquivo físico.

```

193 // Grava dados fisicamente
194 function doSalvarDados() {
195
196     var obAjax = openAjax();
197
198     doCarregando(true);
199
200     obAjax.open("GET", "inc/inc_salvar.php", true);
201     obAjax.send(null);
202
203     obAjax.onreadystatechange = function () {
204         if (obAjax.readyState == 4) {
205             if (obAjax.status == 200) {
206
207                 doCancelar();
208
209                 // Esconde o loading
210                 doCarregando(false);
211
212             }
213         }
214     }
215 }
216 }

```

Figura 25: *Function* responsável por gravar os dados.

Observa-se que o software tem diversas outras funções internas que não foram demonstradas. Outro ponto importante foi à utilização do XML como banco de dados do software. O XML acabou se tornando muito mais útil pois era o mesmo formato utilizado pela console GPMC.

3.3.2 Operacionalidade da implementação

Esta seção apresenta um estudo de caso, do ponto de vista do usuário, objetivando mostrar a funcionalidade e operacionalidade do software. O estudo de caso abrange o Diagrama de Atividades, apresentado na seção 3.2.2., e descreve o processo da exportação para a console.

Uma empresa do Vale do Itajaí em Santa Catarina que já utiliza há algum tempo a estrutura do Windows 2003, o *Active Directory*, onde se possuem 500 computadores utilizando o Windows XP, além de 12 servidores utilizando o Windows 2003 e um servidor Linux. A empresa sempre se utilizou de políticas de grupo (GPO) para um melhor gerenciamento de seus colaboradores. Hoje a empresa se expandiu fazendo com que tenha uma estrutura de mais de 500 computadores na sua matriz, além de filiais espalhadas pelo

Brasil, que se utilizam da estrutura do AD da empresa mãe, mas também possuem sua estrutura de *Active Directory* que se comunica com a matriz através de acessos dedicados. Nesse cenário as GPOs acabam tendo uma degradação do seu desempenho pelo fato de ser ter muitas vezes diversas políticas que devem ser processadas mas que muitas vezes acabam nem sendo utilizadas pela estação ou pelo usuário. Com esse intuito a Microsoft desenvolveu os filtros WMI dentro do seu console do GPMC.

A tendência da Microsoft é de expandir ainda mais a compatibilidade com esses filtros. Mas hoje a console do filtros acabou se tornando algo muito complicado. Os *queries* são todas montadas na mão complicando o trabalho dos administradores. Além de que os mesmos não podem facilmente estar transportando esses filtros para as filiais que são também controladas pela matriz. Com esse nicho de mercado surgiu o software que auxilia a criação desses filtros WMI e que são exportados em subseqüência para a console da Microsoft.

Primeiramente na figura 26 é demonstrada a tela inicial do software.

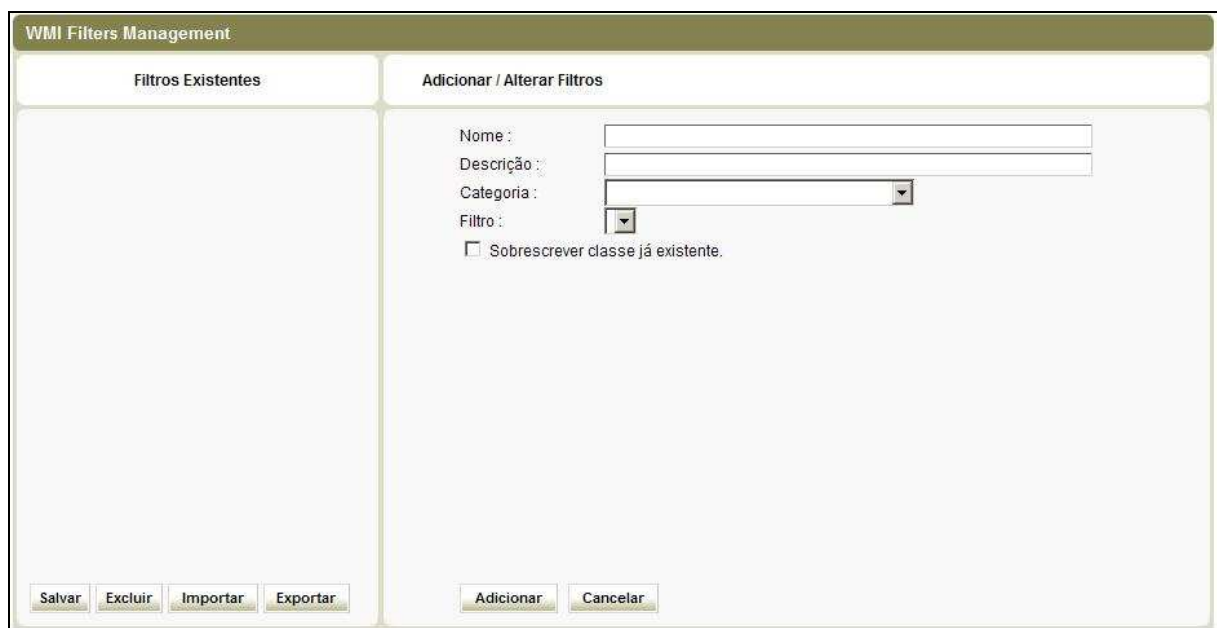


Figura 26: Tela inicial do software.

O administrador agora irá importar os filtros que ele já criou anteriormente no software, conforme a figura 27. Para chegar nessa tela o administrador deverá apertar no botão de importar. Ele tem a opção de escolher o arquivo que ele deseja importar e sobrescrever os filtros que ele já tiver com nome igual, sendo bem útil essa função em caso de atualização nas filiais.

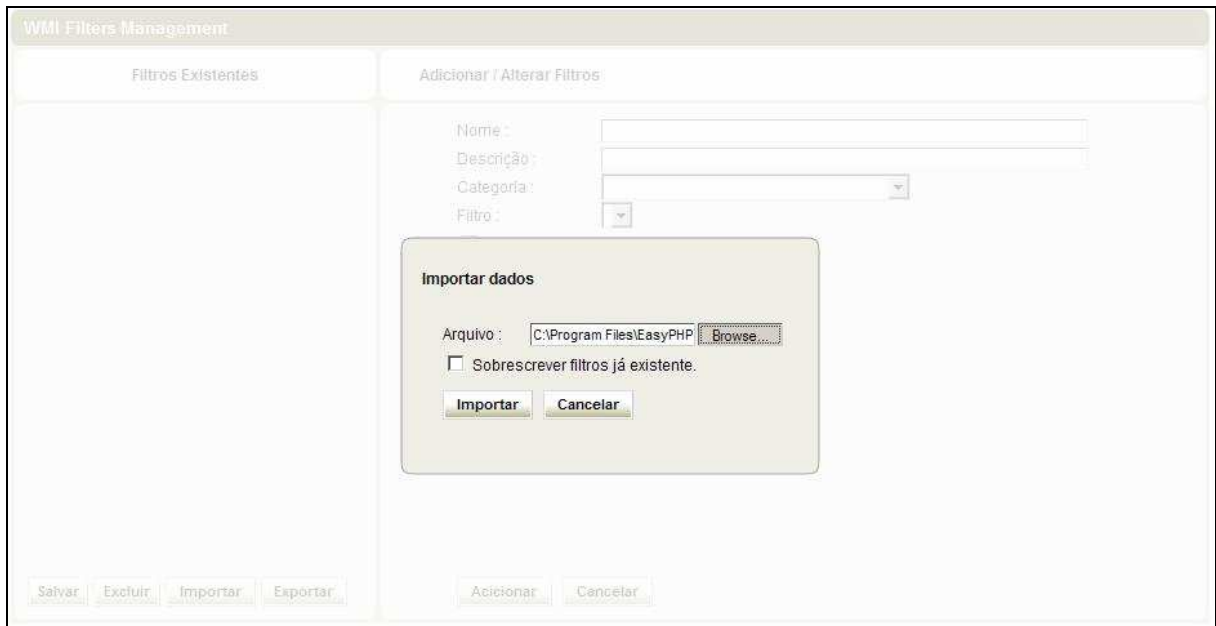


Figura 27: Importação de filtros.

Após a sua importação no software é possível alterar e/ou visualizar os dados do filtro para manutenção e respectiva utilização no console do GPMC. Na figura 28 é demonstrado um filtro já importado e podendo ser alterado pelo administrador.

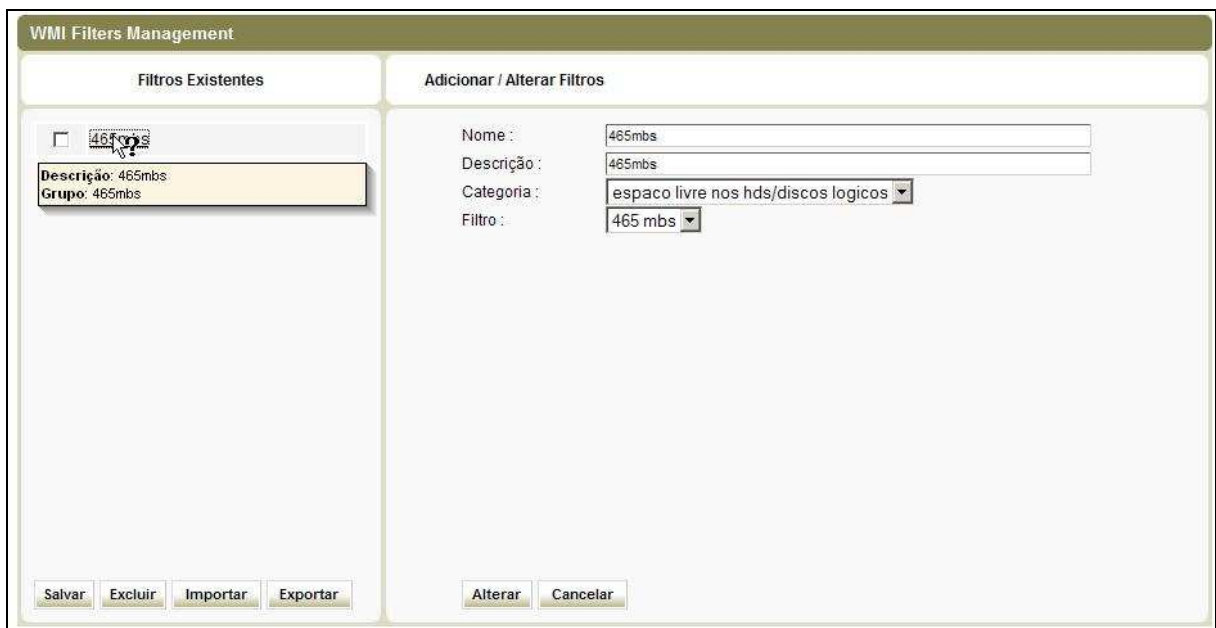


Figura 28: Alteração ou visualização do filtro.

No software também é possível cadastrar um filtro conforme a necessidade do cliente. Os filtros são cadastrados conforme categorias que já são pré-cadastradas no software. Após o usuário cadastrar o seu filtro conforme a sua necessidade ele irá adicioná-lo no software. Na figura 29 é mostrado o cadastro de um filtro pelo usuário.

WMI Filters Management

Filtros Existentes

Adicionar / Alterar Filtros

Nome : Proxy para SQUID

Descrição : Filtro para aplicar configuracoes para os computador com proxy SQUID.

Categoria : proxy

Filtro : Squid

Sobrescrever classe já existente.

Salvar Excluir Importar Exportar

Acicionar Cancelar

Figura 29: Cadastro de um filtro.

Após o administrador adicionar o filtro ao software ele será visualizado nos filtros existentes no software, conforme é mostrado na figura 30.

WMI Filters Management

Filtros Existentes

Adicionar / Alterar Filtros

Nome :

Descrição :

Categoria :

Filtro :

Sobrescrever classe já existente.

Salvar Excluir Importar Exportar

Acicionar Cancelar

Figura 30: Visualização dos filtros depois de cadastrados.

O software também traz a possibilidade de exclusão de determinado filtro. É necessário escolher qual filtro deseja excluir e apertar o botão excluir, conforme demonstrado na figura 31. Após a confirmação o filtro será excluído com sucesso. Se caso o usuário não selecionar nenhum filtro para excluir o software irá mostrar uma mensagem de erro pedindo que o administrador escolha um filtro para excluir.

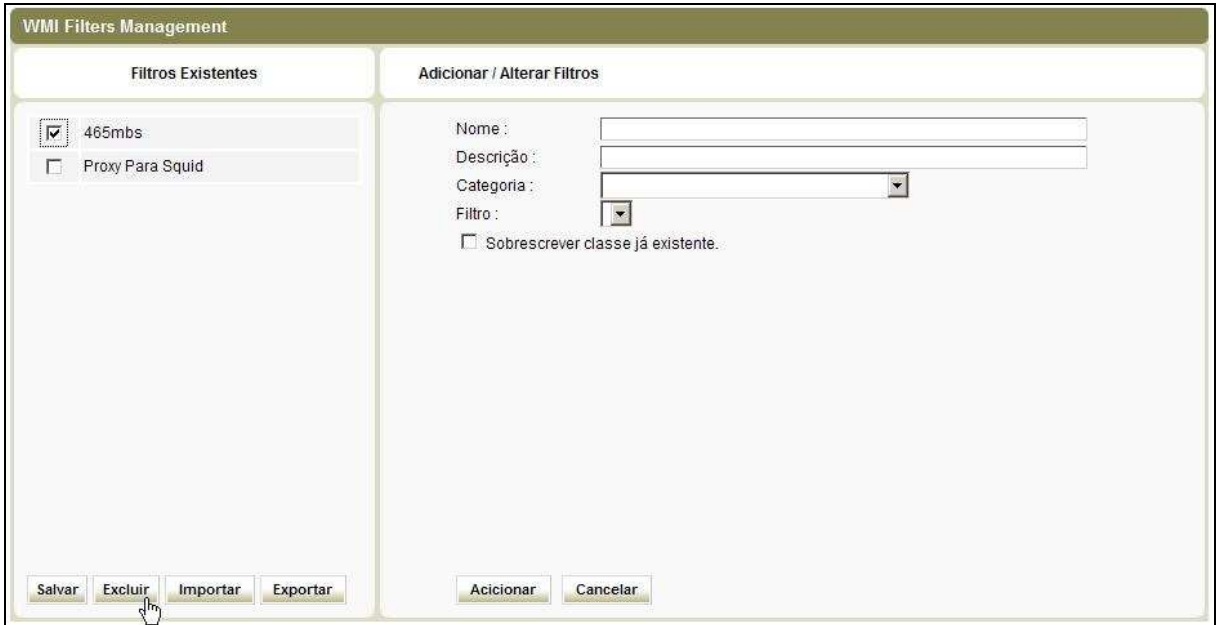


Figura 31: Exclusão de filtro.

Após o cadastramento de todos os filtros que o administrador deseja para sua rede ele poderá exportá-lo para a console da Microsoft, além de exportá-lo em arquivo texto para posterior utilização em outra filial ou na matriz. Deverão ser selecionados quais filtros serão exportados. Se caso o usuário não selecionar nenhum filtro será apresentada uma mensagem de erro. Na figura 32 é demonstrada a tela de exportação.

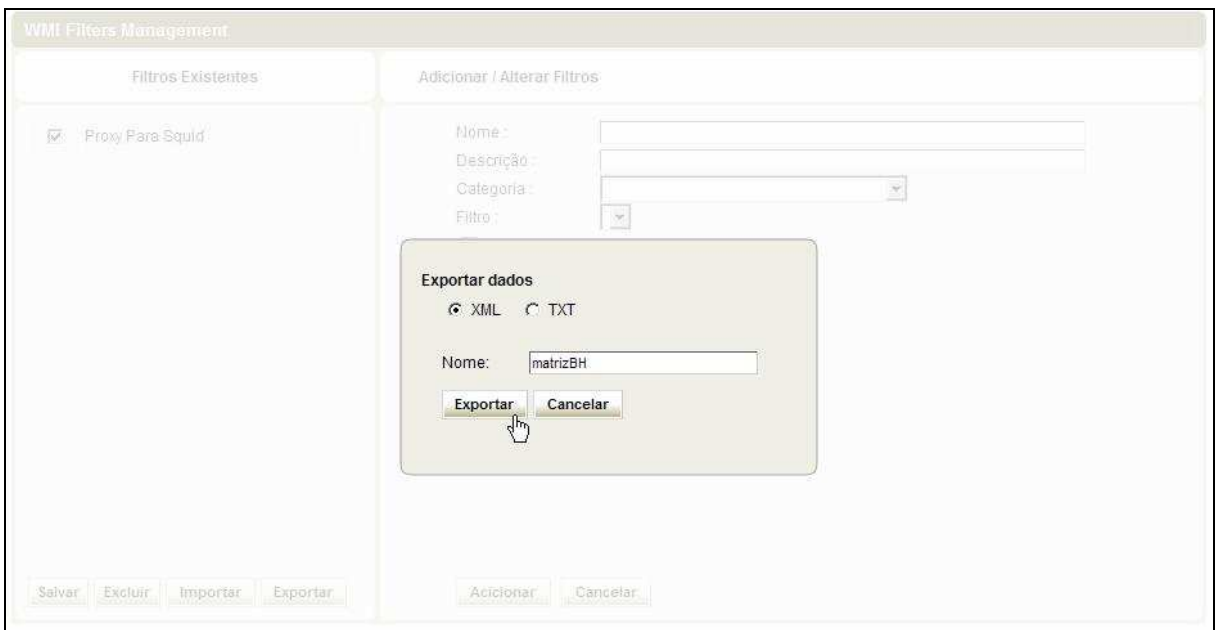


Figura 32: Exportação de um filtro.

A exportação se realizando com sucesso será possível visualizar o filtro cadastro no software no GPMC, conforme visualizado na figura 33.

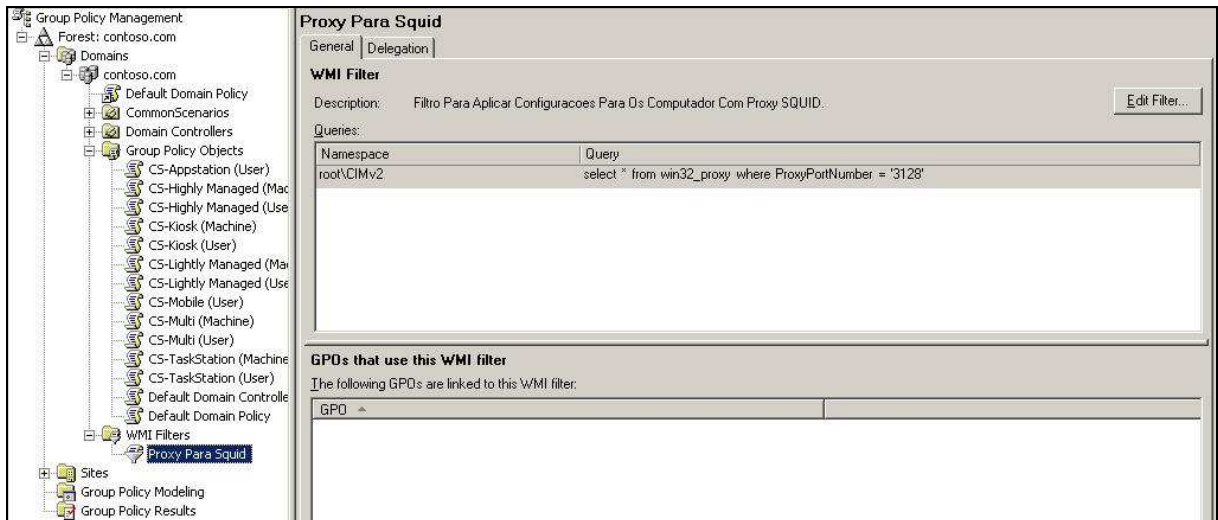


Figura 33: Filtro na console GPMC.

Finalizada a exportação para a console o administrador já pode estar utilizando os filtros WMI para melhoria no processamento das políticas de grupo da empresa.

3.4 RESULTADOS E DISCUSSÃO

Os resultados obtidos através do trabalho foram muito satisfatórios. Não foi encontrado nenhum trabalho de pesquisa com esta profundidade sobre o tema WMI.

O gerenciamento de rede está crescendo conforme a necessidade das empresas. O WMI trouxe como diferencial a sua filtragem sobre os objetos que existiam nas política de segurança.

Os resultados obtidos conseguiram complementar todos os objetivos iniciais do trabalho, que seriam possibilitar a criação de filtros através das classes apresentados no anexo B, a importação e exportação dos filtros de um ambiente para outro, a utilização do XML para armazenamento dos filtros.

Um dos resultados obtidos através do trabalho foi que o software facilita ao administrador, pois ele não tem a necessidade de cadastrar os filtros de forma manual (*query*) e sim somente com o *mouse* para escolher qual filtro ele deseja criar.

Outra facilidade implantada e que trouxe muitos benefícios ao usuário final é a possibilidade de exportação dos filtros para a console e também importação de arquivos vindos de outros ambientes. E também a possibilidade de estar transportando esses filtros para outros domínios existentes em outras filiais.

Foram feitos diversos testes em ambientes diferentes para consolidar a aplicação. Esses testes foram compostos por:

- importação de filtros vindos de outros domínios;
- alterações de filtros e inclusão de filtros diferentes;
- exportação para a console da Microsoft;
- salvar o arquivo para ser importado em outro ambiente.

Esses testes foram realizados por administradores de duas empresas que possuem mais de 200 máquinas. Ambos se mostram muito satisfeitos com o software e com a sua facilidade e usabilidade para seu uso no dia-a-dia.

4 CONCLUSÕES

As atuais discussões sobre um melhor gerenciamento e um menor custo operacional da área do suporte da empresa vêm gerando ótimos aplicativos para os administradores de redes no ambiente Windows.

O *Active Directory* presente desde o Windows 2000 para o gerenciamento de redes vem crescendo e ganhando cada vez mais adeptos. Um grande percentual dos servidores hoje são Windows e grande parte é gerenciada pelo AD. A união do AD com as políticas de grupos acabaram se tornando um marco.

As GPOs se tornaram algo que atribui um ótimo benefício a um baixo custo operacional, trazendo muitas possibilidades aos gerenciadores de uma rede de computadores. A sua implantação é rápida e acabou se tornando cada vez mais simplificada. As políticas de grupo também trouxeram para o ambiente corporativo os filtros WMI, que são responsáveis por uma melhora nas regras, possibilitando que GPOs sejam aplicadas a determinadas máquinas com hardware e/ou software que o administrador especificou, dentre outras inúmeras possibilidades.

O presente estudo pretende colaborar justamente com o conceito abordado acima, construindo um software responsável pela geração de filtros WMI sendo facilmente importados e adaptados para as *Group Policy Objects*. A sua atribuição a uma regra ainda será feita através do GPMC.

O trabalho desenvolvido pretende colaborar ainda mais com esse mercado que está em constante expansão. Atualmente o mercado tem a necessidade de melhoria no desempenho de políticas em ambiente complexo (filiais e matrizes) e também redes com mais de 500 computadores.

O trabalho foi concluído com sucesso, atingindo todos os seus objetivos iniciais. A sua implantação em uma rede corporativa acabou acontecendo numa rede de aproximadamente 500 máquinas, além de ambientes de testes onde ocorreram melhoras no tempo de processamento das políticas com a utilização dos filtros WMI gerados pelo software.

4.1 EXTENSÕES

Devidos as atualizações que irão ocorrer com a chegada do novo software operacional da Microsoft para servidores, o trabalho terá como extensão possíveis adaptações ao novo padrão se caso a Microsoft estabeleça, fazendo com que o software funcione no ambiente antigo e novo.

Outra sugestão para possível continuidade seria a melhor integração entre a console e o programa fazendo com que o programa consulte os filtros WMI exportados para a console anteriormente, melhorar a integração das GPOs com os respectivos filtros e após a exportação, escolher o filtro e atribuí-lo a GPO necessária.

Outra extensão que está ganhando força com a vinda do novo software operacional é a linha de comando chamada *Powershell*, que trará um grande gama de aplicações possíveis no ambiente corporativo. Através de linha de comando é possível atribuir endereços IPs para máquinas, renomear placas de rede, fazer instalações de software sem nenhuma interface para o usuário final, dentre outras possibilidades. Essa extensão deverá ser o carro chefe do novo software operacional.

REFERÊNCIAS BIBLIOGRÁFICAS

ALLEN, Robbie. **Active Directory Cookbook For Windows Server 2003 & 2000**. Sepastopol: O`Reilly, 2003. 622 p.

BATTISTI, Júlio. **Windows Server 2003 Curso Completo**. Rio de Janeiro: Axcel Books, 2003. 1537 p.

BATTISTI, Júlio. **Guia de estudos para a prova MCSE 70-290**. Rio de Janeiro: Axcel Books, 2004. 1000 p.

COMPUTER PERFORMANCE. **WMI Filters for Group Policies**, 2007. Disponível em <http://www.computerperformance.co.uk/w2k3/gp/group_WMI_filters.htm>. Acesso em: 21 mar. 2007.

COOPERSTEIN, Jeffrey. Windows Management Instrumentation: Administering Windows and Applications across Your Enterprise. **MSDN Magazine**, v. 15, n. 5, 2000. Disponível em <<http://msdn.microsoft.com/msdnmag/issues/0500/wmiover/>>. Acesso em: 15 mar. 2007.

DUNALEY et al. **A guide to group policy**. New York: Windows & .net Magazine, 2004. 77 p.

EVENSON, Jeff. The impact of security group policies - a case study. Disponível em: <http://www.giac.org/certified_professionals/practicals/GSEC/2374.php>. Acesso em: 02 set. 2006.

FERREIRA, Alexandre Marinho. **Windows 2000 e Linux: um estudo comparativo no contexto de segurança informática lógica**. 2003. 85 p. Monografia (2) - Universidade Fernando Pessoa, Porto, 2003.

HANEY, Julie M.. **A guide to securing Microsoft Windows 2000 Group Policy**. 1.1 Washington: Nsa, 2001. 77 p.

MCINTOSH, Robert et al. **A guide to group policy**. New York: Windows & .net Magazine, 2004. 35 p.

MELBER, Derek. **Advanced Group Policy Objects**. Irvine: Mcpmag, 2004. 137 p.

MICROSOFT CORPORATION. **Windows Management Instrumentation and the Common Information Model**, 1998. Disponível em <<http://msdn2.microsoft.com/en-us/library/ms811552.aspx>>. Acesso em: 15 mar. 2007.

MICROSOFT CORPORATION. **Windows Management Instrumentation: Background and Overview**, 2000. Disponível em <<http://msdn2.microsoft.com/en-us/library/ms811553.aspx>>. Acesso em: 14 mar. 2007.

MICROSOFT CORPORATION. **WMI: Introduction to Windows Management Instrumentation**, 2003. Disponível em <<http://www.microsoft.com/whdc/system/pnppwr/wmi/WMI-intro.msp>>. Acesso em: 18 mar. 2007.

MICROSOFT CORPORATION. **WMI filtering using GPMC**, 2005. Disponível em <<http://technet2.microsoft.com/WindowsServer/en/library/6237b9b2-4a21-425e-8976-2065d28b31471033.msp?mfr=true>>. Acesso em: 12 mar. 2007.

MICROSOFT CORPORATION. **Win32 Classes**. 2007. Disponível em <<http://msdn2.microsoft.com/en-us/library/aa394084.aspx>>. Acesso em: 14 maio 2007.

RICE, David. **Group Policy reference**. 1.0.8 Washington: Nsa, 2001. 223 p.

W3C – WORLD WIDE WEB CONSORTIUM – **W3C**: style active statement, Massachusetts, 2004. Disponível em <<http://www.w3.org/Style/Activity>>. Acesso em: 21 mar. 2007.

WINDOWS MANAGEMENT INSTRUMENTATION. In: WIKIPEDIA, a enciclopédia livre. Wikipédia Foundation, 2006. Disponível em: <http://en.wikipedia.org/wiki/Windows_Management_Instrumentation>. Acesso em: 20 mar. 2007.

ZEND – THE PHP COMPANY – **ZEND**: changes in PHP 5/Zend Engine II, Cupertino, 2004. Disponível em <<http://www.zend.com/php5/articles/engine2-php5-changes.php>>. Acesso em: 25 mar. 2007.

APÊNDICE A – Descrição dos cenários

UC 01.01. Cadastro de filtros WMI

Requisitos

- *RF 01.* O software deverá permitir o cadastramento de filtros WMI.

Cenários

Cadastro de filtros WMI {Principal}.

- 1) O administrador coloca qual o nome do filtro que deseja cadastrar.
- 2) O administrador coloca uma descrição para o filtro que deseja estar criando.
- 3) O administrador escolhe dentro de uma das opções de filtros existentes no software.
- 4) O administrador cadastra o filtro no software.

UC 01.02. Exportação dos filtros WMI

Requisitos

- *RF 02.* O software deverá permitir quais filtros serão exportados para o GPMC.

Restrições

- *Pré-condição* . Ter instalado o GPMC no servidor de domínio.
- *Pré-condição*. Ter cadastrado algum filtro (UC 01.01).

Cenários

Exportação dos filtros WMI {Principal}.

- 1) Administrador seleciona quais filtros ele irá exportar.
- 2) O administrador seleciona qual o formato ele irá importa.
- 3) A exportação é realizada com sucesso.

Exportação para XML {Alternativo}.

- 1) No passo 2 ao selecionar o formato XML, o software irá fazer a importação para o console do GPMC.

Exportação para TXT {Alternativo}.

- 1) No passo 2 ao selecionar o formato TXT, o software irá fazer a exportação utilizando a linguagem XML para o arquivo TXT.

UC 01.03. Importação dos filtros WMI

Requisitos

- *RF 05.* O software deverá permitir importar filtros WMI.

Restrições

- *Pré-condição* . Ter instalado o GPMC no servidor de domínio.

Cenários

Exportação dos filtros WMI {Principal}.

- 1) O administrador seleciona a opção de importação no software.
- 2) O administrador seleciona qual o arquivo deseja importar, sendo ele somente um XML ou um TXT.
- 3) O administrador importa seus filtros com sucesso.

Importação com opção de subscrever { Alternativo }.

- 1) No passo 2 ao selecionar o arquivo XML, tem a opção de subscrever os filtros que já tiverem iguais no software.

UC 01.04. Salvar os filtros WMI

Requisitos

- *RF 04.* O software deverá possibilitar o salvamento dos filtros em arquivos XML.

Restrições

- *Pré-condição.* Ter cadastrado algum filtro (UC 01.01).

Cenários

Salvar os filtros WMI {Principal}.

- 1) O administrador seleciona quais filtros deseja salvar.
- 2) O administrador seleciona a opção salvar no software.
- 3) O arquivo será gerado no local onde está configurado pelo arquivo de configuração do software.

UC 01.05. Remove filtros WMI

Requisitos

- *RF 03.* O software deverá permitir o usuário remover algum filtro WMI.

Restrições

- *Pré-condição.* Ter cadastrado algum filtro (UC 01.01).

Cenários

Remoção de filtro WMI {Principal}.

- 1) O administrador seleciona qual filtro quer excluir.
- 2) O administrador seleciona a opção excluir.

ANEXO A – Código-fonte dos *includes* do software

Na figura 34 é demonstrado o código fonte do *include* responsável pela configuração do software.

```
1 <?
2
3 // Titulo.
4 $path_titulo      = "WMI Filters Management ";
5
6 // Caminho XML.
7 $path_xml         = "xml/";
8
9 // Caminho TMP.
10 $path_tmp        = "tmp/";
11
12 // Caminho Absoluto TMP.
13 $path_abs_tmp    = "C:\\Program Files\\EasyPHP1-8\\www\\tmp\\";
14
15 // Caminho Script externo.
16 $path_script     = "%programfiles%\\gpmc\\scripts\\CreateEnvironmentFromXML.wsf";
17
18 // Arquivo onde salva os dados.
19 $path_nome_xml_dbc = $path_xml."wmi_filtering_config.xml";
20
21 // Arquivo Dos Filtros.
22 $path_nome_xml_filter = $path_xml."wmi_filter.xml";
23
24 ?>
```

Figura 34: *include* responsável pela configuração do software.

Na figura 35 é demonstrado parte do código fonte do *include* responsável pela exportação do software.

```

37 $ext = strtolower($ext);
38
39 // Trunca o arquivo (se existir) e grava os dados guardados na sessão
40 $nom_file = $nom_file.".".$ext;
41 $dir = "../".$path_tmp.$nom_file;
42 $file = fopen($dir, "w+");
43 fwrite($file, $out_val);
44 fclose($file);
45
46 // Exportação
47 if (strtoupper($ext) == "XML") {
48
49     // Comando que executa importação na ferramenta externa.
50     $comando = "cscript \"\" . $path_script . "\" /XML:\"\" . $path_abs_tmp . $nom_file . "\" $*";
51     // Executa script de importação.
52     system($comando, $retorno);
53     // Exibe o retorno na janela.
54     echo $retorno;
55
56 }
57 else {
58     // Redireciona janela para efetuar download do arquivo
59     ?><script>window.location = 'inc_download.php?nom_file=?= $nom_file; ?>';</script><?
60 }
61
62 }
63 else {
64     ?><script>
65     alert('É necessário escolher algum filtro para exportar.');
```

Figura 35: *include* responsável pela configuração do software.

Na figura 36 é demonstrado parte do código fonte do *include* responsável pela importação dos filtros no software.


```

18 // Exclui o arquivo se existir
19 if (file_exists($caminho)) {
20     unlink($caminho);
21 }
22
23 // extensão do arquivo
24 $ext = strtoupper(substr($nom_file, -3));
25
26 // Cópia para a pasta dos arquivos
27 copy($_FILES["NOM_FILE_IMPORT"]["tmp_name"], $caminho);
28
29 if (file_exists($caminho)) {
30
31     switch($ext) {
32
33         case "XML" :
34             $obWMI->doImportXML($caminho, $ind_write);
35             break;
36
37         case "TXT" :
38             $obWMI->doImportText($caminho, $ind_write);
39             break;
40
41     }
42
43     // Exclui o arquivo ao final do processo de importação
44     @unlink($caminho);
45
46 }
47
48 ?>
49 <script type="text/javascript" src="../js/funcao.js"></script>
50 <script type="text/javascript" src="../js/ajax.js"></script>
51 <script type="text/javascript">
52 alert('Importação Concluída');

```

Figura 36: *include* responsável pela importação do software.

Na figura 37 é demonstrado o código fonte do *include* responsável pela adição de novos filtros no software.

```

1  <?
2
3  session_start();
4
5  header("Content-type: text/html; charset=ISO-8859-1");
6
7  require_once("../class/wmi.php");
8  require_once("inc_funcao.php");
9
10 // Objeto WMI
11 $obWMI = new WMI();
12
13 $cat    = $_GET["CAT"];
14 $filtro = $_GET["FILTRO"];
15
16 $query = "select * from ".$cat." where ".$filtro;
17
18 $obWMI->setImpName(strtolower(trim($_GET["NAME"])));
19 $obWMI->setImpDesc(trim($_GET["DESCRIPTION"]));
20 $obWMI->setImpQuery($query);
21 $obWMI->setImpCat($cat);
22 $obWMI->setImpFiltro($filtro);
23
24 $msg = $obWMI->doAdicionarFiltro($_GET["IND_WRITE"]);
25
26 echo "alert('" . $msg . "')";
27
28 ?>

```

Figura 37: *include* responsável pela adição de filtros.

Na figura 38 é demonstrado o código fonte do *include* responsável por salvar os filtros no arquivo.

```

1  <?
2
3  session_start();
4
5  header("Content-type: text/html; charset=ISO-8859-1");
6
7  require_once("config.php");
8  require_once("../class/wmi.php");
9  require_once("inc_funcao.php");
10
11 // Objeto WMI
12 $obWMI = new WMI();
13
14 $xml_out = $obWMI->doSalvarFiltros();
15
16 // Trunca o arquivo (se existir) e grava os dados guardados em sessão
17 $file = fopen("../".$path_nome_xml_dbc, "w+");
18 fwrite($file, $xml_out);
19 fclose($file);
20
21 ?>

```

Figura 38: *include* responsável pela adição de filtros.

Na figura 39 e 40 será demonstrado parte do código fonte do *include* responsável por

funções dentro do software, como por exemplo, retirada de acentuação nos filtros.

```

3 // tira acentuação
4 function tiraAcento($dados, $espaco=true, $lower=true) {
5
6     $dados = ereg_replace("[áâãäå]","a", $dados);
7     $dados = ereg_replace("[ÁÂÃÄÅ]","A", $dados);
8     $dados = ereg_replace("[éêëè]","e", $dados);
9     $dados = ereg_replace("[ÉÊËÈ]","E", $dados);
10    $dados = ereg_replace("[íîï]","i", $dados);
11    $dados = ereg_replace("[ÍÎÏ]","I", $dados);
12    $dados = ereg_replace("[óôõö]","o", $dados);
13    $dados = ereg_replace("[ÓÔÕÖ]","O", $dados);
14    $dados = ereg_replace("[úûü]","u", $dados);
15    $dados = ereg_replace("[ÚÛÜ]","U", $dados);
16    $dados = str_replace("ç", "c", $dados);
17    $dados = str_replace("Ç", "C", $dados);
18
19    if ($espaco) {
20        $dados = str_replace(" ", "_", $dados);
21    }
22    if ($lower) {
23        $dados = strtolower($dados);
24    }
25
26    return $dados;
27
28 }
29
30 function doLimpaListagem($text) {
31
32     $text = str_replace("'", "", $text);
33     //$text = str_replace('"', "", $text);
34     $text = str_replace("<", "&lt;", $text);
35     $text = str_replace(">", "&gt;", $text);
36     $text = str_replace("\\", "\\\\", $text);

```

Figura 39: *include* responsável por funções no software.

```

64 function doDownloadFile($nom_arquivo) {
65
66     global $path_abs_tmp;
67
68     // extensão do arquivo
69     $ext = substr($arquivo,-3 );
70
71     // Caminho absoluto do arquivo a baixar.
72     $arquivo = $path_abs_tmp . $nom_arquivo;
73
74     // Aparece ja com o tipo para o usuario salvar
75     switch($ext){
76
77         case "xls":
78             $ctype="application/vnd.ms-excel";
79             break;
80
81         case "csv":
82             $ctype="application/vnd.ms-excel";
83             break;
84
85         case "txt":
86             $ctype="text/plain";
87             break;
88
89         default:
90             $ctype="application/force-download";
91
92     }

```

Figura 40: *include* responsável por funções no software (continuação)

Na figura 41 é demonstrado parte do código fonte do *include* responsável por ordená-los por ordem alfabética.

```

5 header("Content-type: text/html; charset=ISO-8859-1");
6
7 require_once("inc_funcao.php");
8
9 ?><table width="100%" border="0" cellpadding="2" cellspacing="2" bgcolor="#FFFFFF">
10 <?
11
12 $arr = $_SESSION["CLASS"];
13
14 // Ordena o array em ordem alfabética
15 if (is_array($arr["NAME"])) {
16     array_multisort($arr["NAME"], SORT_ASC, SORT_STRING, $arr["CATEGORIA"], SORT_ASC, SORT_STRING, $arr["FILITRO"], SORT_ASC, SORT_STRING,
17     $arr["ID"], SORT_ASC, SORT_STRING, $arr["DESCRIPTION"], SORT_ASC, SORT_STRING, $arr["QUERY"], SORT_ASC, SORT_STRING);
18 }
19
20 // Vars o array para listagem das classes
21 for ($cont = 0; $cont < count($arr["NAME"]); $cont++) {
22
23     $NAME = doLimpaListagem(ucwords($arr["NAME"][$cont]));
24     $DESCRIPTION = doLimpaListagem(ucwords($arr["DESCRIPTION"][$cont]));
25
26     $camada = '<b>Descrição</b>: '.$NAME.'<br>';
27     $camada .= '<b>Grupo</b>: '.$DESCRIPTION;
28
29 ?><tr>
30     <td width="15%" style="height: 18px;" align="center" bgcolor="#F4F4F4"><input name="FILTER[]" id="FILTER" type="checkbox" value="<?=$arr["ID"][$cont]; ?>" /></td>
31     <td width="85%" bgcolor="#F4F4F4" class="txtNormal" align="left"><a href="#void(0)" onclick="doAlterar('<?=$arr["ID"][$cont]; ?>');"
32     onMouseover="ddrivetip('<?=$camada; ?>', '#FBF5E1', 250);" style="cursor: help;" onMouseout="hidddrivetip();" class="linkNormal"><?=$arr["NAME"][$cont], 25); ?></a></td>
33 </tr><?
34 }
35 ?></table>

```

Figura 41: *include* responsável por colocá-los em ordem alfabética.

Na figura 42 é demonstrado parte do código fonte do *include* responsável por carregar

os filtros para a memória.

```

11 // Objeto WMI
12 $obWMI = new WMI();
13
14 $id = $_GET["ID"];
15
16 if ($id <> "") {
17
18     // Verifica se já existe um array na memória
19     if(is_array($_SESSION["CLASS"]["NAME"])) {
20
21         reset($_SESSION["CLASS"]["ID"]);
22         reset($_SESSION["CLASS"]["NAME"]);
23         reset($_SESSION["CLASS"]["DESCRIPTION"]);
24         reset($_SESSION["CLASS"]["QUERY"]);
25
26         // Verifica se o conteúdo já existe em memória
27         if (in_array($id, $_SESSION["CLASS"]["ID"])) {
28             // Busca a posição da classe
29
30             $num_pos = $obWMI->getNumPosFiltro("ID", $id);
31
32             ?>
33             getObj("Alterar").style.display = '';
34             getObj("Acionar").style.display = 'none';
35             getObj("lblIndWrite").style.visibility = 'hidden';
36             getObj("ID_ALT").value = '<?=' . $id . '?>';
37             getObj("NAME").value = '<?=' . ucwords($_SESSION["CLASS"]["NAME"][$num_pos]) . '?>';
38             getObj("DESCRIPTION").value = '<?=' . $_SESSION["CLASS"]["DESCRIPTION"][$num_pos] . '?>';
39             getObj("WMIFilter").value = '<?=' . $_SESSION["CLASS"]["CATEGORIA"][$num_pos] . '?>';
40             doLoadWMIFilterWhere(getObj("WMIFilter"), "<?=" . $_SESSION["CLASS"]["FILTRO"][$num_pos] . "?>");
41             <?

```

Figura 42: *include* responsável por carregar os filtros para a memória.

Na figura 43 é demonstrado o código fonte do *include* responsável por alterar os filtros que já existem na memória do software.

```

1  k?
2
3  session_start();
4
5  header("Content-type: text/html; charset=ISO-8859-1");
6
7  require_once("../class/wmi.php");
8  require_once("inc_funcao.php");
9
10 // Objeto WMI
11 $obWMI = new WMI();
12
13 $cat    = $_GET["CAT"];
14 $filtro = $_GET["FILTRO"];
15
16 $query = "select * from ".$cat." where ".$filtro;
17
18 $obWMI->setImpName(strtolower(trim($_GET["NAME"])));
19 $obWMI->setImpDesc(trim($_GET["DESCRIPTION"]));
20 $obWMI->setImpQuery($query);
21 $obWMI->setImpCat($cat);
22 $obWMI->setImpFiltro($filtro);
23
24 $obWMI->doAlterarFiltro($_GET["ID"]);
25
26 echo "alert('Filtro Atualizado');";
27
28 ?>

```

Figura 43: *include* responsável por alterar os filtros que já existem na memória.

Na figura 44 é demonstrado o código fonte do *include* responsável por excluir os filtros do software.

```

1  k?
2
3  session_start();
4
5  header("Content-type: text/html; charset=ISO-8859-1");
6
7  require_once("config.php");
8  require_once("../class/wmi.php");
9  require_once("inc_funcao.php");
10
11 // Objeto WMI
12 $obWMI = new WMI();
13
14 $msg = $obWMI->doExcluir(strtolower($_GET["COD_EXCLUIR"]));
15
16 echo "alert('" . $msg . "')";
17
18 ?>

```

Figura 44: *include* responsável por excluir que estão no software.

Nas figuras 45 e 46 são demonstradas partes do código fonte do *include* responsável por carregar as *queries* dos filtros WMI.

```

1  <?
2
3  session_start();
4
5  header("Content-type: text/html; charset=ISO-8859-1");
6
7  ?>
8
9  obFilter    = getObj("WMIFilter");
10 obFilter.length=0;
11
12 var newElem = document.createElement("OPTION");
13 newElem.value = '';
14 newElem.text = '';
15 obFilter.options.add(newElem);
16
17 <?
18
19     for ($i = 0; $i < count($_SESSION["FILTER"]["NAME"]); $i++ ) {
20
21     ?>
22         newElem = document.createElement("OPTION");
23         newElem.value = '<?=$_SESSION["FILTER"]["NAME"][$i]; ?>';
24         newElem.text = '<?=$_SESSION["FILTER"]["DESCRIPTION"][$i]; ?>';
25         obFilter.options.add(newElem);
26     <?

```

Figura 45: *include* responsável por carregar as *queries* para o software.

```

3  session_start();
4
5  header("Content-type: text/html; charset=ISO-8859-1");
6
7  $name = strtoupper($_GET["NAME"]);
8  $where = strtoupper(trim($_GET["WHERE"]));
9
10 ?>
11
12 obFilter    = getObj("WMIFilterWhere");
13 obFilter.length=0;
14
15 var newElem = document.createElement("OPTION");
16 newElem.value = '';
17 newElem.text = '';
18 obFilter.options.add(newElem);
19
20 <?
21
22     for ($i = 0; $i < count($_SESSION["FILTER"]["NAME"]); $i++) {
23         if (strtoupper($_SESSION["FILTER"]["NAME"][$i]) == $name) {
24             for ($x = 0; $x < count($_SESSION["FILTER"]["WMIFILTER"][$i]["DESCRIPTION"]); $x++) {
25
26             ?>
27                 newElem = document.createElement("OPTION");
28                 newElem.value = "<?=$_SESSION["FILTER"]["WMIFILTER"][$i]["WHERE"][$x]; ?>";
29                 newElem.text = "<?=$_SESSION["FILTER"]["WMIFILTER"][$i]["DESCRIPTION"][$x]; ?>";
30             <?
31
32             if (stripslashes(strtoupper($_SESSION["FILTER"]["WMIFILTER"][$i]["WHERE"][$x])) == stripslashes($where)) {
33
34             ?>
35                 newElem.selected = true;
36             <?

```

Figura 46: *include* responsável por carregar o *where* para as *queries*.

ANEXO B – Classes WMI relevantes

As classes a seguir são algumas classes que poderiam estar sendo importadas para o software. Essas classes têm cada uma delas a sua particularidade, assim como configurações que podem estar ocorrendo através de seus parâmetros. Para a implementação dessas classes no software é necessário seu cadastramento no arquivo “wmi_filter.xml” localizado dentro da pasta “xml”.

Classes dos dispositivos de refrigeração

Os dispositivos de refrigeração têm uma subcategoria no grupo de classes do WMI. Essa classe representa ventiladores, locais onde se tiram temperaturas e instrumentos de refrigeração. No quadro 7 são apresentados os dispositivos de refrigeração.

Classes	Descrição
Win32_Fan	Representa as propriedades do ventilador que está dentro do computador.
Win32_HeatPipe	Representa as propriedades do <i>heat pipe</i> do cooler.
Win32_Refrigeration	Representa as propriedades do dispositivo de refrigeração.
Win32_TemperatureProbe	Representa as propriedades do sensor de temperatura.

Fonte: adaptado de Microsoft (2007)

Quadro 7: dispositivos de refrigeração

Classes dos dispositivos de entrada

A subcategoria das classes de dispositivos de entrada representaria os teclados e mouses. No quadro 8 é demonstrado as classes da entrada.

Classes	Descrição
Win32_Keyboard	Representa o teclado instalado no sistema operacional.
Win32_PointingDevice	Representa o dispositivo utilizado como mouse.

Fonte: adaptado de Microsoft (2007)

Quadro 8: classes dos dispositivos de entrada

Classes de dispositivos de armazenamento

No quadro 9 é apresentado a subcategoria de dispositivos de armazenamento representa os dispositivos conhecidos como *hard disk*, CD-ROM e unidades de fita.

Classes	Descrição
Win32_AutochkSetting	Representa as propriedades da operação de checagem automática do disco.
Win32_CDROMDrive	Representa a unidade de CD-ROM instalada no sistema operacional.
Win32_DiskDrive	Representa o disco físico assim como ele é enxergado pelo sistema operacional.
Win32_FloppyDrive	Gerencia as capacidades de uma unidade de disquete.
Win32_PhysicalMedia	Representa qualquer tipo de documentação ou armazenamento mediano.
Win32_TapeDrive	Representa uma unidade de fita no computador.

Fonte: adaptado de Microsoft (2007)

Quadro 9: classes dos dispositivos de armazenamento

Classes da Placa Mãe, portas e controladores

A placa mãe, portas e controladores é uma subcategoria das classes que representam os dispositivos do sistema. Exemplos desses dispositivos seria memória do sistema, memória cache e seus controladores. No quadro 10 são apresentados as classes da placa mãe e outros componentes.

Classes	Descrição
Win32_1394Controller	Representa as capacidades e gerenciamento do controlador 1394.
Win32_1394ControllerDevice	Relacionado ao controlador da IEEE1394 (Firewire).
Win32_AllocatedResource	Relacionado um dispositivo lógico alocado para o sistema.
Win32_AssociatedProcessorMemory	Relacionado ao processador e sua memória <i>cache</i> .
Win32_BaseBoard	Representa a placa mãe.
Win32_BIOS	Representa os atributos do sistema de entrada e saída do computador.
Win32_Bus	Representa a entrada física como o sistema operacional enxerga.
Win32_CacheMemory	Representa a memória <i>cache</i> (interna e externa) no computador.
Win32_FloppyController	Representa a capacidade de gerenciamento do controlador do disquete.
Win32_IDEController	Representam a capacidade de controlador os dispositivos ligados a IDE.
Win32_InfraredDevice	Representa a capacidade e gerenciamento do dispositivo de infravermelho.
Win32_MemoryDevice	Representa as propriedades do dispositivo da memória do sistema com os seus respectivos endereços.
Win32_MotherboardDevice	Representa o dispositivo que contém os componentes centrais do sistema operacional.
Win32_OnBoardDevice	Representa algum dispositivo que está adaptado na placa mãe.
Win32_ParallelPort	Representa as propriedades da porta paralela do sistema operacional.
Win32_PCMCIAController	Gerencia as capacidades do dispositivo PCMCIA.
Win32_PhysicalMemory	Representa a memória física disponível alocada para o sistema operacional.
Win32_Processor	Representa o dispositivo capaz de interpretar a seqüência de instruções para o computador com o sistema operacional Windows.
Win32_SCSIController	Representa o dispositivo SCSI no sistema.
Win32_SoundDevice	Representa as propriedades disponíveis no dispositivo de som do Windows.
Win32_SystemBIOS	Relaciona o sistema operacional e o sistema da BIOS e

	suas propriedades gerenciáveis.
Win32_USBController	Gerencia as capacidades da USB.

Fonte: adaptado de Microsoft (2007)

Quadro 10: representação das classes da placa mãe e outros componentes

Classes dos dispositivos de rede

No quadro 11 são apresentadas as classes dos dispositivos de rede representa a interface do controlador de rede, suas configurações e suas opções.

Classes	Descrição
Win32_NetworkAdapter	Representa o adaptador de rede no sistema operacional.
Win32_NetworkAdapterConfiguration	Representa os atributos do adaptador de rede.
Win32_NetworkAdapterSetting	Relaciona o adaptador de rede e suas configurações do sistema.

Fonte: adaptado de Microsoft (2007)

Quadro 11: classe dos dispositivos de rede

Classe da energia

A subcategoria de energia é a classe que representa os suprimentos de energia, baterias e eventos relacionados a esse tipo de dispositivos. No quadro 12 segue as respectivas classes.

Classes	Descrição
Win32_Battery	Representa a bateria conectada ao computador.
Win32_CurrentProbe	Representa as propriedades do sensor do monitor da temperatura corrente.
Win32_PortableBattery	Representa as propriedades de uma bateria portátil, muito utilizada em <i>notebooks</i> .
Win32_UninterruptiblePowerSupply	Representa as capacidades e gerenciamento da fonte de energia ininterrupta.
Win32_VoltageProbe	Representa as propriedades do sensor de voltagem eletrônico.

Fonte: adaptado de Microsoft (2007)

Quadro 12: classes da subcategoria energia

Classes das impressoras

A subcategoria das impressoras representa as impressoras, suas configurações e os trabalhos de impressão. No quadro 13 são demonstrados as classes da impressora.

Classes	Descrição
Win32_Printer	Representa o dispositivo conectado a um sistema Windows e que tem a capacidade de reproduzir imagens visuais numa mídia.
Win32_PrinterConfiguration	Define as configurações do dispositivo de impressão.
Win32_PrinterController	Relaciona a impressora e seu dispositivo local conectado.
Win32_PrinterSetting	Relaciona a impressora e suas respectivas configurações.
Win32_PrintJob	Representa o trabalho de impressão criado pelo sistema operacional.
Win32_TCPIPPrinterPort	Representa o serviço de TCP/IP acessando a impressora.

Fonte: adaptado de Microsoft (2007)

Quadro 13: apresentado as classes da impressora

Classes do Video e Monitor

No quadro 14 as classes da subcategoria de vídeo e monitores são apresentadas que seriam os monitores, placas de vídeo e suas configurações associadas.

Classes	Descrição
Win32_DesktopMonitor	Representa o tipo de monitor ou tela conectada ao sistema operacional.
Win32_DisplayConfiguration	Representa as informações das configurações do dispositivo de visualização.
Win32_VideoController	Representa as capacidades e gerenciamento do controlador do vídeo do sistema operacional.
Win32_VideoSettings	Relaciona o controlador de vídeo e suas respectivas configurações aplicadas no dispositivo.

Fonte: adaptado de Microsoft (2007)

Quadro 14: classe de vídeo e monitores

Área de trabalho

A subcategoria da área de trabalho no grupo de classes representa os objetos da configuração da área de trabalho. No quadro 15 é demonstrado as classes da área de trabalho.

Classes	Descrição
Win32_Desktop	Classe instanciada Representa as características comuns do usuário da área de trabalho.
Win32_Environment	Classe instanciada Representa o ambiente ou as configurações do sistema para o ambiente do sistema operacional.
Win32_TimeZone	Classe instanciada Representa a zona de tempo configurada no sistema operacional.
Win32_UserDesktop	Classe associada Relaciona a conta do usuário e as configurações da área de trabalho específica.

Fonte: adaptado de Microsoft (2007)

Quadro 15: classes da área de trabalho

Sistema de arquivos

A subcategoria do sistema de arquivos representa as classes diretamente ligadas ao disco do computador. Isso inclui o tipo de arquivo usado pelo sistema, a estrutura de diretórios e o jeito que o disco está particionado. No quadro 16 são apresentados as classes da subcategoria em questão.

Classes	Descrição
Win32_Directory	Classe instanciada Representa a entrada do diretório do sistema operacional.
Win32_DiskPartition	Classe instanciada Representa a capacidade e gerenciamento da área particionado no disco físico para o sistema operacional Windows.
Win32_DiskQuota	Classe associada Espaço utilizado em sistemas de volume NTFS.
Win32_QuotaSetting	Classe instanciada Contém as informações da quota daquele volume do disco.
Win32_SubDirectory	Classe associada Relaciona o diretório como uma ou demais subdiretórios.
Win32_SystemPartitions	Classe associada Relaciona o sistema e a partição do disco com o sistema operacional.

Fonte: adaptado de Microsoft (2007)

Quadro 16: classes do sistema de arquivos

Rede e domínio

A categoria de rede de computadores representa as conexões de rede, clientes de rede e as configurações das redes existentes em cada dispositivo. No quadro 17 são apresentados as classes existentes nessa categoria.

Classes	Descrição
Win32_NetworkClient	Classe instanciada Representa a classes de rede do Windows.
Win32_NetworkConnection	Classe instanciada Representa a conexão de rede ativa com o sistema operacional.
Win32_NetworkProtocol	Classe instanciada Representa o protocolo e as características da rede com o sistema operacional Windows.
Win32_NTDomain	Classe instanciada Representa um domínio do Windows NT.

Fonte: adaptado de Microsoft (2007)

Quadro 17: classes da categoria rede e domínio

Configurações do Sistema operacional

A categoria de configurações do sistema operacional representa as configurações e o sistema operacional, está representada no quadro 18 que segue abaixo.

Classes	Descrição
Win32_BootConfiguration	Classe instanciada Representa a configuração da inicialização do sistema operacional.
Win32_ComputerSystem	Classe instanciada Representa o sistema operacional do computador.
Win32_ComputerSystemProcessor	Classe associada Relaciona o sistema do computador com o processador que está rodando o sistema.
Win32_ComputerSystemProduct	Classe instanciada Representa o produto do sistema.
Win32_OperatingSystem	Classe instanciada Representa o sistema operacional instalado no computador.
Win32_SystemBootConfiguration	Classe associada Relaciona o sistema com a configuração de inicialização.
Win32_SystemDesktop	Classe associada Relaciona o sistema operacional com a configuração da área de trabalho.
Win32_SystemDevices	Classe associada Relaciona o sistema do computador com o os seus dispositivos lógicos instalados no sistema operacional.
Win32_SystemNetworkConnections	Classe associada Relaciona a conexão de rede com o sistema operacional onde a conexão reside.
Win32_SystemSetting	Classe associada Relacionam o sistema operacional as configurações gerais do próprio sistema.
Win32_SystemTimeZone	Classe associada Relaciona o computador a sua zona de horário.
Win32_SystemUsers	Classe associada Relaciona o computador as contas cadastradas no própria sistema.

Fonte: adaptado de Microsoft (2007)

Quadro 18: classes responsáveis pelas configurações do sistema operacional

Serviços

No quadro 19 são apresentados as classes da subcategoria dos serviços e suas bases.

Classes	Descrição
Win32_BaseService	Classe instanciada Representa os objetos executáveis que são instalados no banco de dados do registro.
Win32_Service	Classe instanciada Representa o serviço no computador com sistema operacional Windows.

Fonte: adaptado de Microsoft (2007)

Quadro 19: classes do serviço e suas bases

Compartilhamentos

A subcategoria dos compartilhamentos representa os detalhes que cada compartilhamento como suas pastas ou impressoras. No quadro 20 são apresentados as classes da subcategoria compartilhamento.

Classes	Descrição
Win32_DFSNode	Classe associada Representa a pasta mãe do nó do sistema de arquivos distribuído.
Win32_ServerConnection	Classe instanciada Representa as conexões feitas de um computador remoto para o compartilhamento que está sendo compartilhado no computador local.
Win32_ConnectionShare	Classe associada Relaciona o recurso compartilhado no computador com uma conexão feita externa para o recurso compartilhado.
Win32_PrinterShare	Classe associada Relaciona a impressora local e seu compartilhamento para que seja visualizada pela rede.
Win32_ShareToDirectory	Classe associada Relaciona o recurso compartilhado no computador e o diretório onde o mesmo está mapeado.
Win32_Share	Classe instanciada Representa o compartilhamento do recurso no sistema operacional.

Fonte: adaptado de Microsoft (2007)

Quadro 20: classes WMI da subcategoria do compartilhamento

Armazenamento

No quadro 21 são representadas as informações sobre as classes do armazenamento.

Classes	Descrição
Win32_ShadowBy	Classe associada Representa uma associação entre a cópia de sombra e o dispositivo que está criando a cópia de sombra.
Win32_ShadowContext	Classe associada Especifica como o cópia de sombra irá ser criada, deletada e selecionada.
Win32_ShadowCopy	Classe instanciada Representa a duplicidade de um arquivo original num tempo anterior.
Win32_ShadowOn	Classe associada Representa a associação entre a cópia de sombra e os dados diferenciais existentes.
Win32_Volume	Classe instanciada Representa a área onde a cópia irá ser armazenada.
Win32_VolumeUserQuota	Classe associada Representa o volume que cada usuário terá disponível na sua cota.

Fonte: adaptado de Microsoft (2007)

Quadro 21: classes do armazenamento

Usuários

Essa classe representa as informações sobre o usuário como seu grupo, informações sobre a sua conta, dentre outros. No quadro 22 seguem as classes do respectiva categoria.

Classes	Descrição
Win32_Account	Classe instanciada Representa a informação sobre o usuário e seu grupo.
Win32_Group	Classe instanciada Representa os dados sobre um determinado grupo.
Win32_LogonSession	Classe instanciada Descreve a sessão associada ao usuário que fez o <i>login</i> no Windows.
Win32_NetworkLoginProfile	Classe instanciada Representa a informação da rede de cada usuário no sistema do Windows.
Win32_SystemAccount	Classe instanciada Representa uma conta do sistema.
Win32_UserAccount	Classe instanciada Representa uma informação sobre um determinado usuário do sistema operacional.
Win32_UserInDomain	Classe associada Relaciona uma conta de usuário a um domínio.

Fonte: adaptado de Microsoft (2007)

Quadro 22: classes da categoria usuário