

Protótipo de ferramenta para
monitoração de computadores
utilizando o padrão de gerência
WMI da Microsoft e a plataforma de
desenvolvimento .NET

Acadêmico: Rodrigo Jacobowski
Orientador: Prof. Francisco Adell
Pélicas



Roteiro

- ❑ Introdução
- ❑ Objetivos
- ❑ Fundamentação teórica
- ❑ Requisitos de software
- ❑ Especificação
- ❑ Implementação
- ❑ Resultados e discussão
- ❑ Conclusão e extensões



Introdução

- ❑ O padrão de gerência WMI possibilitará automaticamente manipular as solicitações de clientes, para poder controlar o que acontece com os computadores na rede
- ❑ Algumas dessas informações já podem ser encontradas em outras ferramentas, mas não as funcionalidades desenvolvidas no protótipo



Objetivos

- ❑ Visualizar informações de inventário de um computador
- ❑ Efetuar comparativo entre arquivos de inventário
- ❑ Enviar mensagem de e-mail para o administrador da rede notificando o mesmo de eventuais alterações de inventário



Fundamentação teórica: Gerência Corporativa

- ❑ Como um conjunto de aplicações associadas a softwares e hardwares
- ❑ Desafio de controlar diversas plataformas de hardware de softwares
- ❑ Arquitetura utiliza a mesma estrutura básica e conjunto de relações
- ❑ O SNMP como o protocolo padrão



Áreas funcionais

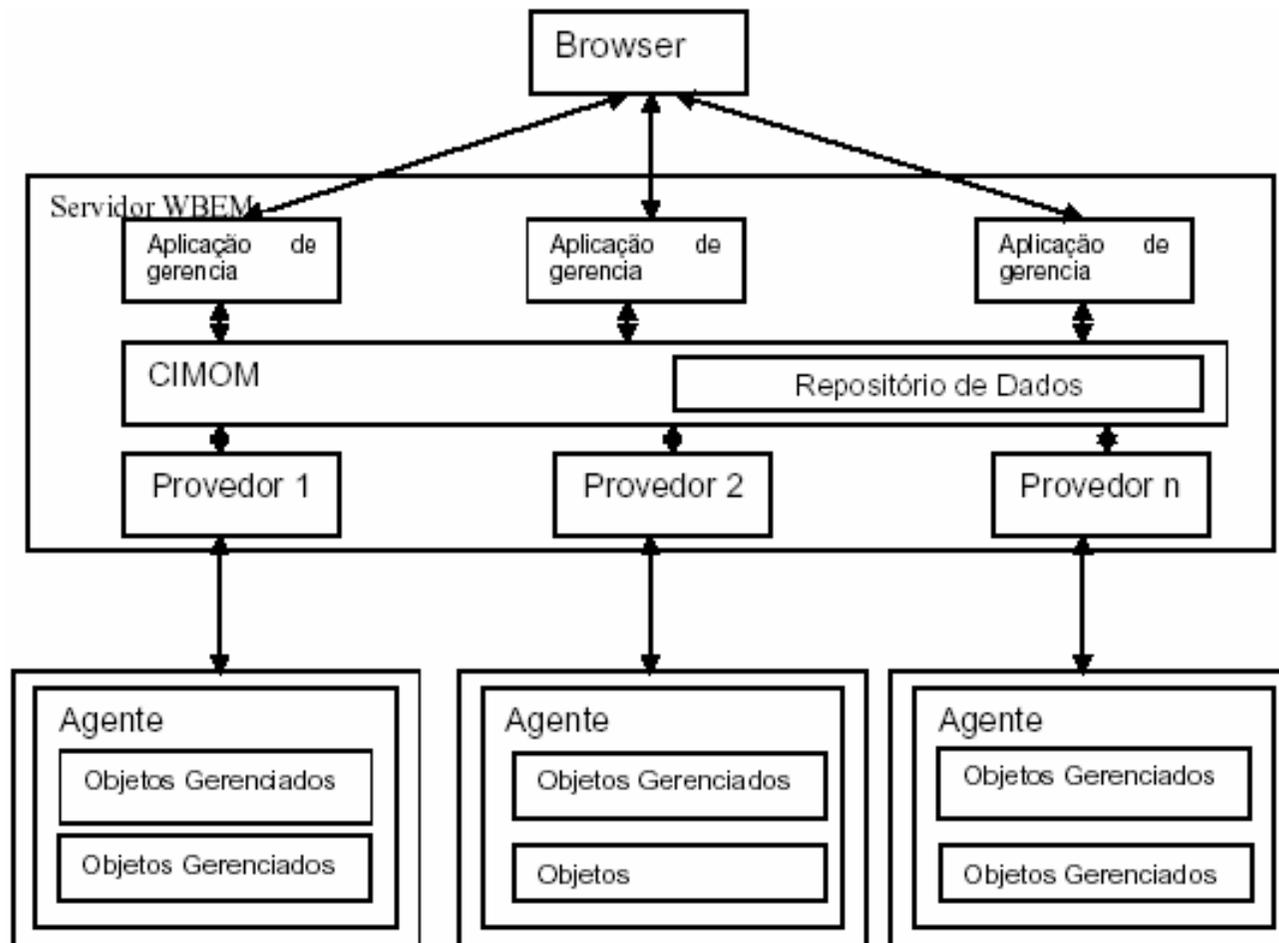
- ❑ Gerência de falhas
- ❑ Gerência de desempenho
- ❑ Gerência de configuração
- ❑ Gerência de contabilização
- ❑ Gerência de segurança



WBEM

- ❑ Resultado de uma iniciativa da indústria da informática
- ❑ Totalmente orientado a objetos
- ❑ Acesso a informações sobre objetos gerenciáveis
- ❑ Independência de padrão proposto

Modelo WBEM





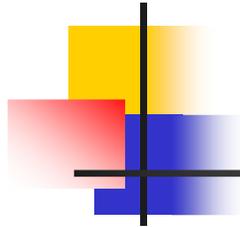
WMI

- Implementação da Microsoft para o WBEM
- Arquitetura:
 - Aplicações de gerenciamento
 - Infra-estrutura do WMI
 - Provedores
 - Objetos gerenciados
- Implementado seguindo os conceitos de *namespaces*



WMI Query Language (WQL)

- ❑ Linguagem de consulta do WMI
- ❑ Suporta três tipos de consultas:
 - ❑ Consulta de dados
 - ❑ Consultas de eventos
 - ❑ Consulta de esquemas



Plataforma .NET

- ❑ Aplicativos para desktop (Windows)
- ❑ Aplicativos para aparelhos móveis (celulares e palms-tops)
- ❑ Multi-plataforma
- ❑ Multi-linguagem
- ❑ Orientado a objetos



.NET Framework

- ❑ Equivalente a máquina virtual java
- ❑ *Assemblies* do .NET
- ❑ Linguagem intermediária (IL)
- ❑ Biblioteca de classes



.NET Framework - Linguagem

- ❑ Arquitetura aberta
- ❑ Java, Delphi, Perl, Cobol, Python, Smalltalk, RPG, APL, Fortran, C#, VisualBasic, ASP
- ❑ Muitas empresas estão apostando nesta plataforma



Trabalhos Correlatos

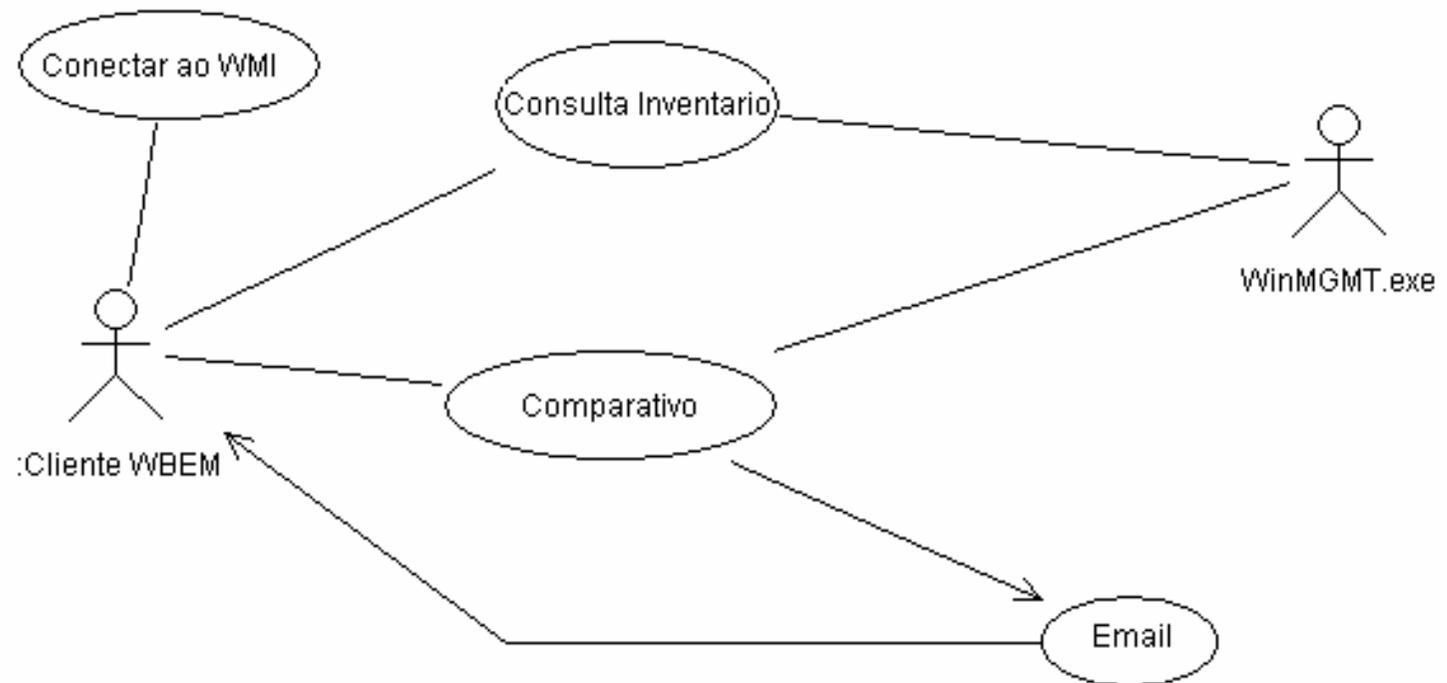
- ❑ Complemento do trabalho de Braz (2003)
- ❑ Implementar sugestões de extensão
- ❑ Protótipo de software de Braz (2003)
- ❑ Objetivos:
 - ❑ Visualizar informação de inventário
 - ❑ Visualizar processos
 - ❑ Receber notificações

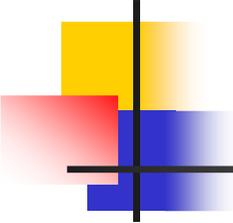


Desenvolvimento do protótipo: Requisitos de Software

- ❑ Operar no Sistema Operacional Windows NT, 2000 ou XP
- ❑ Coletar informações de inventário
- ❑ Realizar inventário
- ❑ Gerar relatório de comparativos
- ❑ Enviar e-mail de notificação de alteração

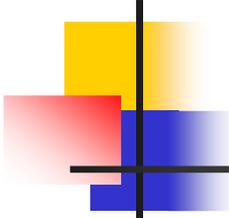
Especificação: Diagrama de casos de uso





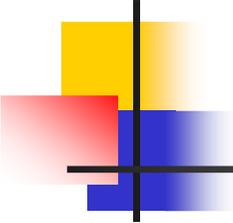
Implementação: Técnicas e Ferramentas Utilizadas

- ❑ WMI presente no Windows XP e 2000
- ❑ *Namespaces* System.Management
- ❑ Borland Delphi 8 .NET Architect
- ❑ Framework .NET
- ❑ Sistema Operacional Windows XP e 2000
- ❑ Rede Local



Implementação: Envio de E-Mail

```
procedure envioemail;
var
  Mensagem : System.Web.Mail.MailMessage;
  ArqErro : TextFile;
  linhaErro : String;
  LinhasMensagem : String;
  Arq : TextFile;
  linha : string;
  config : array[0..2] of string;
  i : integer;
  dir : string;
begin
  assignfile(ArqErro, dir + 'ArquivoLog\ERRO.TXT');
  reset(ArqErro);
  readln(ArqErro, linhaErro);
  dir := ExtractFilePath(Application.ExeName);
  assignfile(Arq, dir + '\Config.TXT');
  reset(Arq);
  i := 0;
  while not eof(Arq) do
  begin
    readln(Arq, linha);
    if i < 3 then
      config[i] := linha;
      i := i + 1;
    end;
  Mensagem := System.Web.Mail.MailMessage.Create();
  Mensagem.From := config[2]; // 'rodrigo@brwnet.com.br';
  Mensagem.Set_To(config[1]); // 'msbinformatica@netuno.com.br'
  Mensagem.Subject := linhaErro;
  //Incluir dados no corpo da mensagem de texto
  LinhasMensagem := '';
  while not eof(ArqErro) do
  begin
    readln(ArqErro, linhaErro);
    LinhasMensagem := LinhasMensagem + linhaErro + chr(13);
  end;
  closefile(ArqErro);
  Mensagem.Body := LinhasMensagem;
  System.Web.Mail.SmtpMail.SmtpServer := config[0]; // 'smtp.indusul.com.br';
  System.Web.Mail.SmtpMail.Send(Mensagem);
  Mensagem.Free;
end;
```



Implementação: Código de Consulta de Softwares Instalados

```
//Informações referentes aos Softwares Instalados
procedure WMISoftwaresInfo(Scope: ManagementScope);
var
  QueryCollection: ManagementObjectCollection;
  Enumerator: ManagementObjectCollection.ManagementObjectEnumerator;
  Item: ManagementObject;
begin
  //Passa para a classe de consulta o nome do computador e classe consultada
  QueryCollection := ConsultaClasse(Scope, 'win32_Product'); // ou CIM_Product
  //Atribui o valor resultante da consulta a uma variável que permite visualizar os
  //itens consultados
  Enumerator := QueryCollection.GetEnumerator;

  frmprincipal.dados.lines.add('');
  frmprincipal.dados.lines.add('Informacao sobre Programas Instalados');
  frmprincipal.dados.lines.add('-----');
  //Percorre os itens consultados e coloca na variavel item o valor das propriedades
  //que estão sendo executadas no momento. Ex. GetPropertyValue('Caption');
  while Enumerator.MoveNext do
  begin
    Item := Enumerator.Current as ManagementObject;
    with Item, Console do
    begin
      frmprincipal.dados.lines.add('    Programa : ' + GetPropertyValue('Caption').ToString);
    end;
  end;
  frmprincipal.dados.lines.add('-----');
end;
```

Operacionalidade da implementação

Trabalho de Conclusão de Curso 2004/2 - FURB

Configurações Geral Relatórios Sair

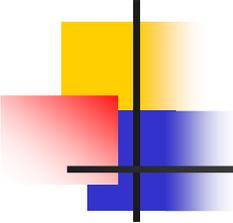
Informação sobre Programas Instalados

- Programa : Wise Owl Demeanor for .NET, Personal Edition
- Programa : WMI Tools
- Programa : ComponentOne Studio Enterprise For Delphi .Net
- Programa : Microsoft Visual J# .NET Redistributable Package 1.1
- Programa : Borland Delphi 8.0 Architect Edition
- Programa : Norton AntiVirus 2002
- Programa : Symantec Network Drivers Update
- Programa : Microsoft .NET Framework SDK (English) 1.1
- Programa : Microsoft Office 2000 Professional
- Programa : WebFldrs XP
- Programa : VBA (2627.7)
- Programa : Borland Delphi 6
- Programa : Java 2 Runtime Environment, SE v1.4.2_04
- Programa : Corel Graphics Suite 11
- Programa : MSN Messenger 6.2
- Programa : Microsoft .NET Framework 1.1

Dorinha
Rodrigo

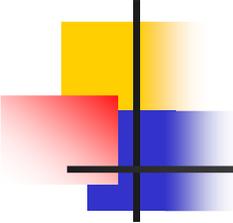
Geração de Inventário Comparativo Listar Computadores Sair

Computador Conectado: Rodrigo



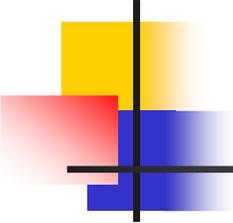
Resultados e Discussão

- Somente possível efetuar a consulta onde o protótipo estiver instalado
- É possível consultar todos o computadores da rede local



Conclusão

- ❑ WMI possibilita operações de gerenciamento
- ❑ Limita-se apenas a consultar dados de estações que estão rodando o WMI
- ❑ Framework .NET esteja instalado
- ❑ Borland Delphi 8 .NET ambiente de desenvolvimento .NET que ainda precisa de amadurecimento



Extensões

- ❑ Módulo que faça busca de nomes dos computadores da rede utilizando o .NET
- ❑ Módulo para visualizar a tela do computador consultado
- ❑ Módulo de consulta de todos os softwares instalados