

# **AUDITCO – AUDITORIA E COMPLIANCE DE SOFTWARES DE TI EM REDES MICROSOFT**

Aluno: Diego Antonio Pamplona

Orientador: Roberto Heinzle

# Roteiro

1. Introdução
2. Objetivos
3. Fundamentação teórica
4. Trabalhos Correlatos
5. Princ. Req. Funcionais/Não Funcionais
6. Especificação do Software
7. Implementação do Software
8. Operacionalidade
9. Resultados/Discussões
10. Conclusões/Sugestões

# Introdução

- Auditorias na empresa
  - Tipos de auditoria
  - Ativos envolvidos (pessoas, processos)
  - Gestão da qualidade
  - Compliance
- Auditoria de Software?
  - Por que?
  - Qual motivo? Quem Auditam?
  - Cenário de Exemplo

# Introdução

- Exemplos: <https://expresso-noticia.jusbrasil.com.br/noticias/141513/empresa-pagara-r-150-mil-por-usar-software-pirata>
- <https://www.baguete.com.br/noticias/software/25/02/2011/software-pirata-multa-e-10x-preco-do-original>
- <https://www.baguete.com.br/noticias/software/18/11/2010/software-piratas-prejudicam-69-das-pmes>
- <https://www.baguete.com.br/noticias/software/18/06/2010/pirataria-gera-indenizacao-de-r-1-bilhao>

# Objetivos

- Software de levantamento de informações e controle dos softwares instalados nos computadores de uma empresa
- Agrupar as informações levantadas de forma automatizada
- Manter de forma controlada as versões e licenças dos softwares instalados;

# Fundamentação Teórica

- **Pirataria de Software:** Reproduzir ilegalmente um programa de Computador sem autorização expressa do titular da obra e por consequência sem devida CDU(Cessão Direito de Uso).

Fonte: ABES - Associação Brasileira das Empresas de

# Fundamentação Teórica

- **Auditoria e Compliance Software:** A Auditoria é a única forma de se determinar tamanho da violação de direito. Como benefício reforça o controle da aplicação, versão e segurança dos dados. Previne desastres e litígio judicial.

Fonte: Sartore (2014)

# Fundamentação Teórica

- **.NET Framework:** Biblioteca ou Conjunto de Bibliotecas usadas para construir uma base onde a aplicação será construída. O NET é a biblioteca que permite construir diversas aplicações Desktop, WEB, IOT, Jogos e Cloud-Based.

Fonte: Microsoft (2018)



# Trabalhos Correlatos

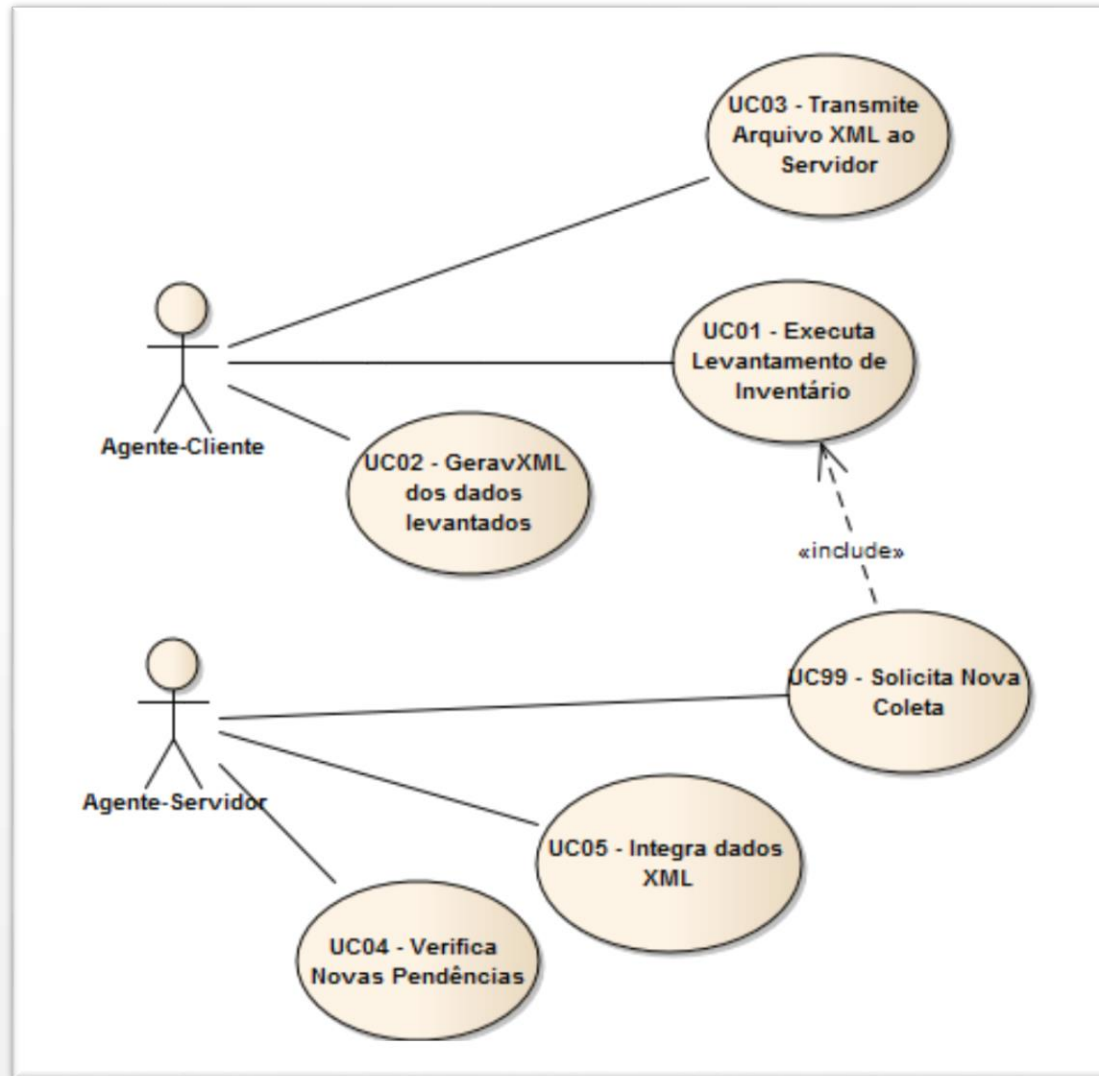
| Característica                         | Kreuch (2006) | Bambinetti (2008) | Heiden (2013) |
|--|---------------|-------------------|---------------|
| Alertas                                | Não           | Não               | Sim           |
| Controle Licença                       | Não           | Não               | Sim           |
| Levantamento Computadores              | Sim           | Não               | Sim           |
| Relatório Gera XML                     | Sim           | Não               | Sim           |
| Levantamento Banco de Dados Relacional | MySQL         | PostgreSQL        | SQL Server    |
| Linguagem Programação                  | Delphi        | Python            | Delphi        |
| Protocolo Transmissão                  | SMB           | Webservice        | TCP/IP e POP  |
| Módulo de Troca de Versão              | Não           | Não               | Não           |

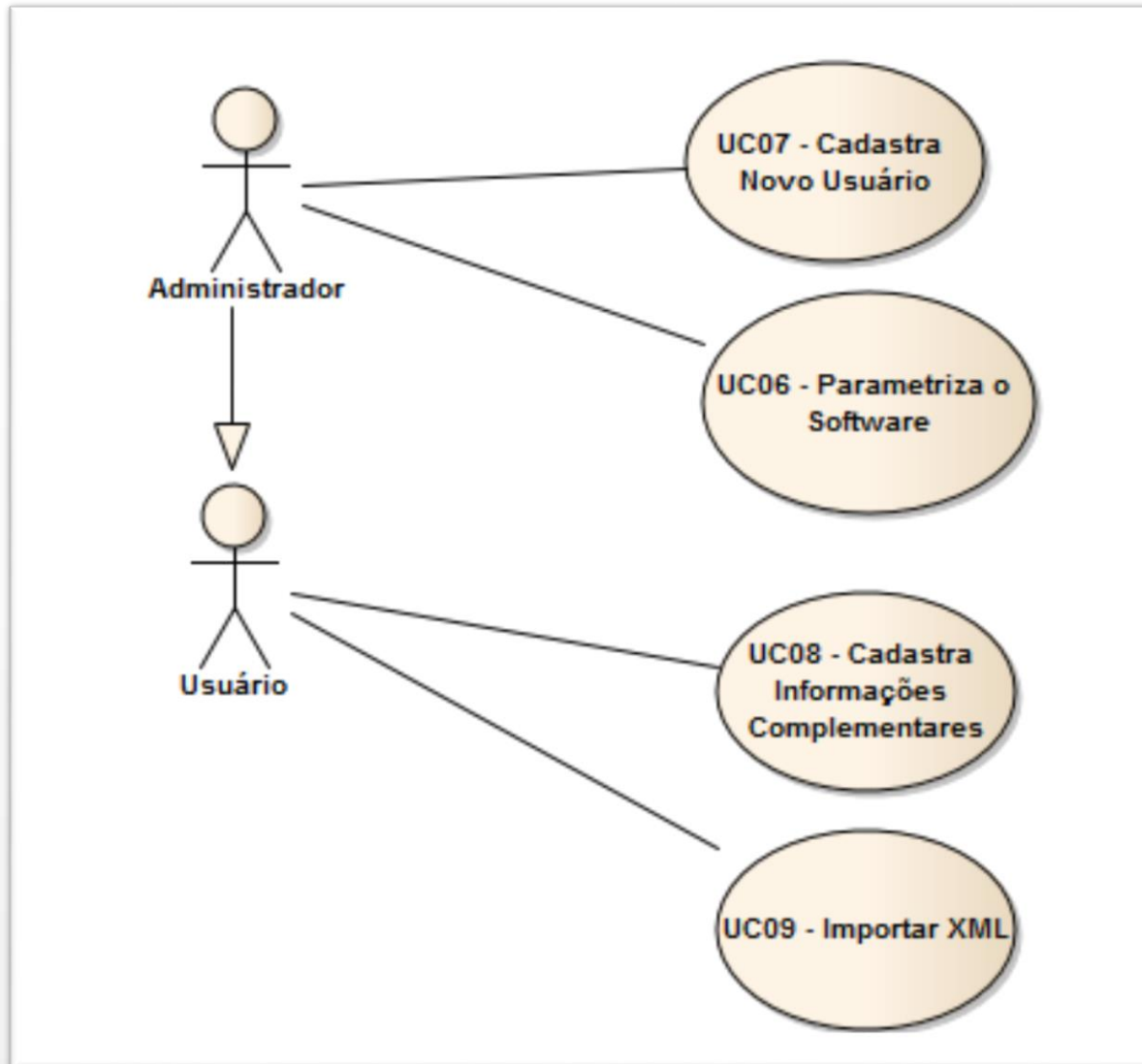
# Requisitos (RF e RNF)

- Funcionais (Principais)
  - Levantamento licenças Windows e Office principalmente
  - Levantamento informações hardware (CPU, Memória...etc)
  - Levantamento informações softwares Instalados
- Não Funcionais (Principais)
  - Visual basic .net
  - Plataforma Windows

# Especificação

- Casos de Uso do Agente-Cliente e Agente-Servidor

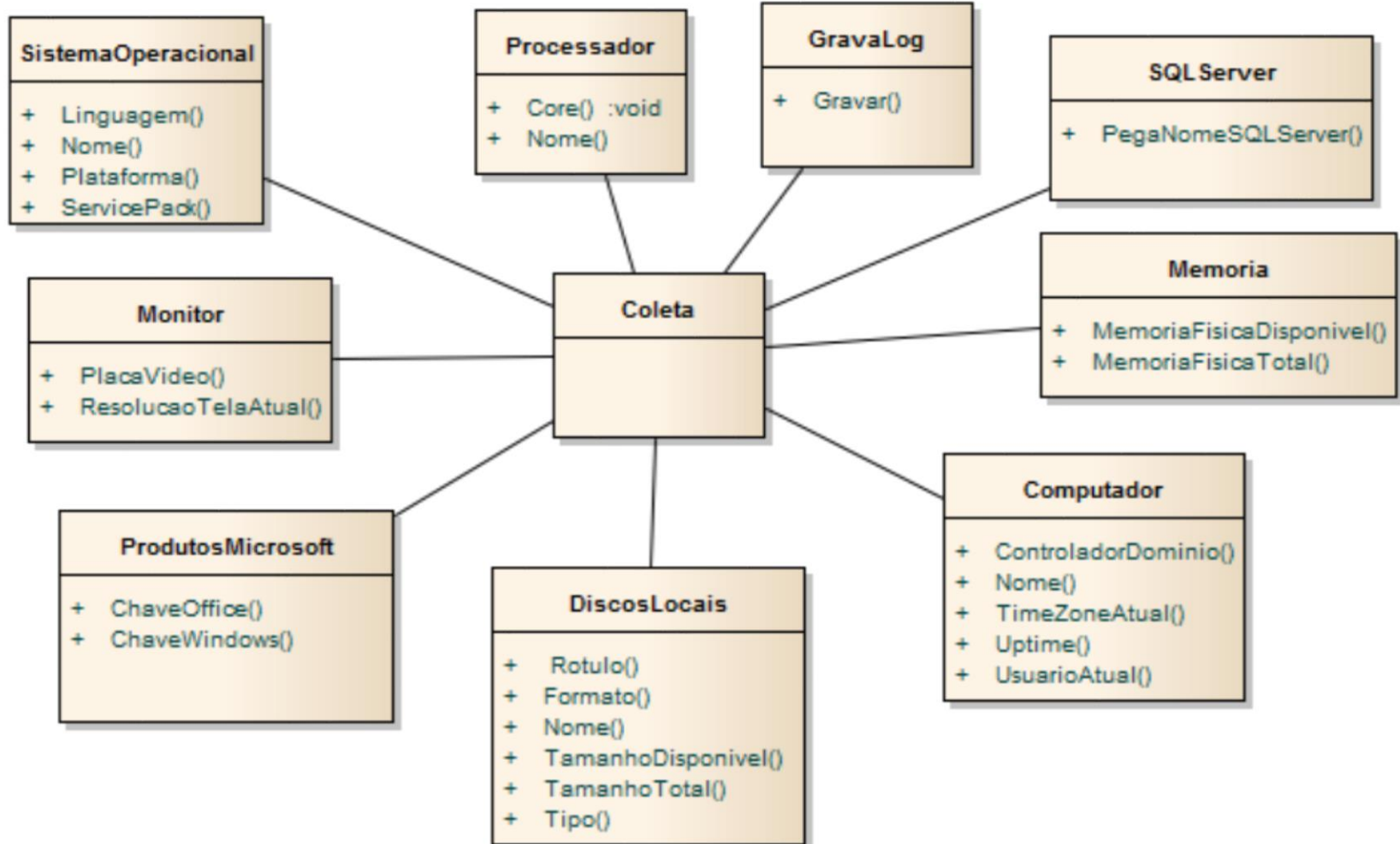




# Especificação

- Diagrama de Classes do AuditCO

## class Diagrama de Classes



# Implementação

- Visual Studio 2015



Aplicação\_Teste - Microsoft Visual Studio

File Edit View Project Build Debug Team Format Tools Architecture Test Analyze Window Help

Quick Launch (Ctrl+Q)

Diego Antônio Pamplona DP

Form1.vb [Design]\*

Form1

Properties

Form1 System.Windows.Forms.Form

|      |       |
|------|-------|
| Tag  |       |
| Text | Form1 |

Text

The text associated with the control.

Solution Explorer

Search Solution Explorer (Ctrl+ç)

Solution 'Aplicação\_Teste' (1 project)

- Aplicação\_Teste
  - My Project
  - References
  - Form1.vb

Ready 15, 15 478 x 398 Publish

## Parte da função de coleta das informações do *guest*

```
Public Function ExecutaColeta()  
    StringComandoColeta = Nothing  
    MontaComando(DataHoje, ",")  
    MontaComando(HoraAtual, ",")  
    MontaComando(Computador.Nome, ",")  
    MontaComando(Computador.UsuarioAtual, ",")  
    MontaComando(Computador.ControladorDominio, ",")  
    MontaComando(Computador.Uptime, ",")  
    MontaComando(Computador.TimeZoneAtual, ",")  
    MontaComando(Processador.Nome, ",")  
    MontaComando(Processador.Core, ",")  
    MontaComando(SistemaOperacional.Nome, ",")  
    MontaComando(SistemaOperacional.Plataforma, ",")  
    MontaComando(SistemaOperacional.Linguagem, ",")  
    MontaComando(Memoria.MemoriaFisicaTotal, ",")  
    MontaComando(Memoria.MemoriaFisicaTotal, ",")  
    For Each drive As IO.DriveInfo In My.Computer.FileSystem.Drives  
        If drive.IsReady = True And drive.DriveType <> IO.DriveType.CDRom Then  
            MontaComando(DiscosLocais.Nome(drive.ToString), ",")  
            MontaComando(DiscosLocais.Formato(drive.ToString), ",")  
            MontaComando(DiscosLocais.TamanhoTotal(drive.ToString), ",")  
            MontaComando(DiscosLocais.TamanhoDisponivel(drive.ToString), ",")  
        End If  
    Next drive
```

## Parte da função de coleta das informações do *guest*

```
MontaComando(Monitor.PlacaVideo, ",")
MontaComando(Monitor.ResolucaoTelaAtual, ",")
For Each pkInstalledPrinters In PrinterSettings.InstalledPrinters
    MontaComando(pkInstalledPrinters.ToString, ",")
Next pkInstalledPrinters
MontaComando(ProdutosMicrosoft.ChaveWindows, ",")
MontaComando(ProdutosMicrosoft.ChaveOffice, ",")
MontaComando(SQLServer.PegaNomeSQLServer().Replace("\", ""), ",")
Dim SubKey As RegistryKey
'Abre a chave que consta os programas que possuem desinstalador
Dim Key As RegistryKey = Registry.LocalMachine.OpenSubKey("SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall", False)
'Levanta Todos os Programas
Dim SubKeyNames() As String = Key.GetSubKeyNames()
'Lista todos os Programas

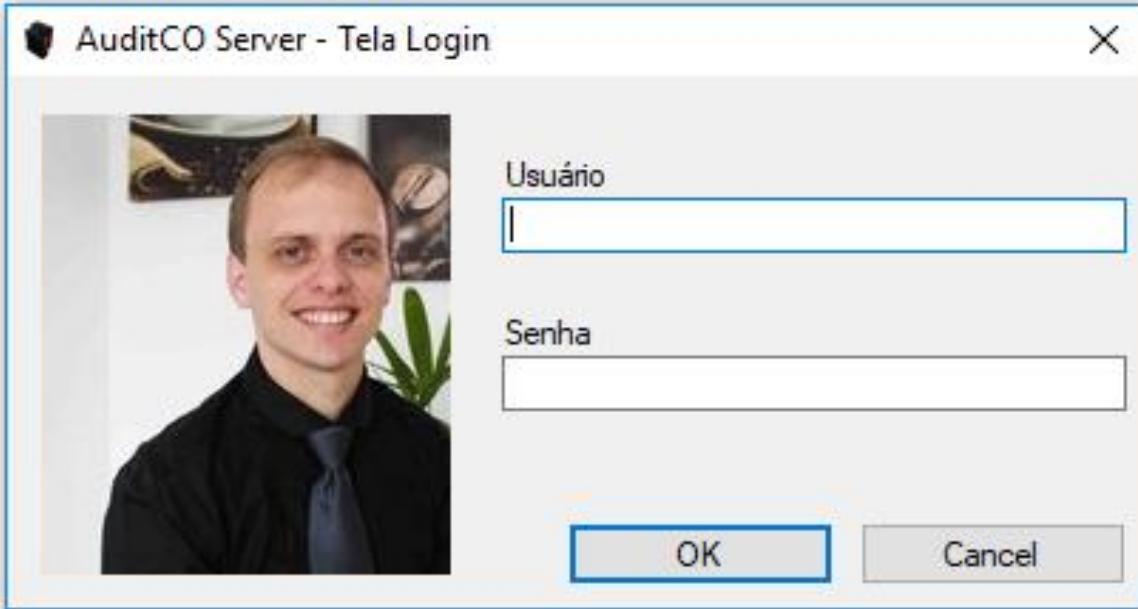
For Index = 0 To Key.SubKeyCount - 1
    SubKey = Registry.LocalMachine.OpenSubKey("SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall" & "\" & SubKeyNames
    'Valida se o Programa tem um Nome Valido para Exibicao
    If Not SubKey.GetValue("DisplayName", "") Is "" Then
        GravaLog((CType(SubKey.GetValue("DisplayName", ""), String)))
        MontaComando((CType(SubKey.GetValue("DisplayName", ""), String)), ",")
    End If
Next
```

## Parte da função de upload/download de arquivos de versões

```
Public Function DownloadUploadVersao()  
    Dim EndFTP As String = "ftp.cliente.com.br"  
    Dim UsuarioFTP As String = "cliente123"  
    Dim SenhaFTP As String = "3mLfth54s#"  
    Dim DiretorioDownload As String = "/downloads/AuditCO_Cliente"  
    Dim DiretorioUpload As String = "/uploads/AuditCO_Cliente"  
    Dim DiretorioLocalUpload As String = My.Application.Info.DirectoryPath & "\Upload"  
    Dim DiretorioLocalDownload As String = My.Application.Info.DirectoryPath & "\NovaVersao"  
    Dim ClienteFTP As FluentFTP.FtpClient = New FluentFTP.FtpClient(EndFTP)  
    ClienteFTP.Credentials = New Net.NetworkCredential(UsuarioFTP, SenhaFTP)  
  
    Try  
        ClienteFTP.Connect()  
        GravaLog("Data/Hora Conexão " & Date.Now & vbCrLf)  
        If Not ClienteFTP.DirectoryExists(DiretorioDownload) Then  
            ClienteFTP.CreateDirectory(DiretorioDownload)  
        End If  
        If Not ClienteFTP.DirectoryExists(DiretorioUpload) Then  
            ClienteFTP.CreateDirectory(DiretorioUpload)  
        End If  
        GravaParametro("VERSAO", "VERSAOATUAL", Date.Now.ToString)  
        'Verifica de o Arquivo de Versão Indica Nova Versão disponível  
        GravaLog("Verificando o Arquivo de Versão...")  
        For Each ItemFTP As FluentFTP.FtpListItem In ClienteFTP.GetListing(DiretorioDownload)  
  
            If ItemFTP.Name = "versao.txt" Then  
                Dim DataModificacao As DateTime = ClienteFTP.GetModifiedTime(ItemFTP.FullName)  
                GravaLog("Horario Atual: " & DateTime.Now)  
                GravaLog("Arquivo: " & ItemFTP.Name & " DataModificação: " & DataModificacao)  
            End If  
        Next  
        GravaLog("-----")  
    End Try  
End Function
```

# Operacionalidade da Implementação

- Tela Login



AuditCO Server - Tela Login

Usuário

Senha

OK Cancel

# Operacionalidade da Implementação

- Server – Tela Principal



# Operacionalidade da Implementação

- Usuários – Cadastro de Usuários




AuditCO - Server. - Cadastro de Usuár... — □ ×

Nome Usuário:

Tipo Usuário:

Foto usuario



# Resultados e Discussões

- Resultados
  - Funciona adequadamente para Maquinas Windows XP, Vista, 7, 8 e 10.
  - Levantamento dos Dados
- Discussões
  - O Office 365 (nova Forma de Licenciamento)
  - Programas com licenciamento Online(Adobe, Senior, ...etc).



# Resultados e Discussões

- Comparativo entre Correlatos e AuditCO
  - Flexibilidade: Os dados geração são de fácil extração e uso, podendo ser importado complementando outras soluções de compliance.
  - Desenvolvimento em linguagem estruturada com funções de fácil melhoramento e reuso.
  - Auditoria de Chave do Windows e Office, que facilita e agiliza verificação de uso principalmente em licenciamentos por volume

# Conclusões e Sugestões

- Conclusão
  - O Software final atendeu todos os objetivos principais, sendo o maior o de levantar e consolidar os dados acerca dos *guests* auditados.
  - Todos os dados gerados estão disponíveis em Banco de Dados e também em XML para importação em qualquer software desejado pelo cliente.
- Sugestões
  - API para Consumo dos Dados do Banco
  - Aplicativo para Consulta Mobile
  - Cliente para Linux