



# **AUTOMAÇÃO DE INVENTARIADO DE HARDWARE E SOFTWARE**

Aluno: Henrique Silva Moura – RA: 2011488/0

Orientador: MSc. Roberto Schaefer

Brasília – DF, Dezembro de 2008

# AUTOMAÇÃO DE INVENTARIADO DE HARDWARE E SOFTWARE

por

Henrique Silva Moura

Trabalho apresentado à Banca examinadora do curso de Engenharia da Computação da FATECS - Faculdade de Tecnologia e Ciências Sociais Aplicadas do UniCEUB – Centro Universitário de Brasília Como requisito parcial para obtenção do título de Engenheiro da Computação

Brasília – DF, Dezembro de 2008  
**Banca Examinadora**

---

Prof. MSc. Roberto Schaefer  
Orientador

---

Prof. MSc. Antônio Barbosa Junior  
Examinador

---

Prof. MSc. Marco Antônio de Oliveira Araújo  
Examinador

---

Prof. MSc. Roberto Ávila Paldes  
Examinador

## Agradecimentos

---

A Deus por minha vida e saúde.

Ao meu pai Afonso pelo exemplo e apoio incondicional. Meu ídolo.

A minha mãe Cristina pelo carinho, serenidade e confiança sempre. Minha fortaleza.

Aos meus irmãos, Isabela e Daniel, pelo companheirismo e cumplicidade em todos os momentos.

Aos meus avós, Maria Helena, Nelson; Ângela e Afonso, pelo acolhimento e amor sempre a mim dedicados.

A minha namorada, Luísa, pela compreensão e ajuda nessa reta final de curso.

Ao meu orientador Professor MSc. Roberto Schaefer, por me guiar no desenvolvimento desse projeto.

Aos "*Pingüins*" pela amizade, alegria e camaradagem.

Aos amigos David Mascarenhas Watkins e Rafael Gomes da Silva por estarem sempre dispostos a me ajudar no desenvolvimento desse projeto.

Aos professores que me deram uma base sólida de conhecimento nesse tempo.

A todos que participaram de forma direta ou indireta no desenvolvimento desse projeto.

# Sumário

---

|   |             |
|---|-------------|
| <b>LISTA DE FIGURAS</b>                                   | <b>V</b>    |
| <b>ÍNDICE DE SIGLAS E ABREVIATURAS</b>                    | <b>VI</b>   |
| <b>RESUMO</b>   | <b>VII</b>  |
| <b>ABSTRACT</b>   | <b>VIII</b> |
| <b>1. CAPÍTULO 01 – INTRODUÇÃO</b>                        | <b>09</b>   |
| 1.1. MOTIVAÇÃO  | 09          |
| 1.2. OBJETIVOS  | 10          |
| 1.3. METODOLOGIA  | 11          |
| 1.4. ESTRUTURA DA MONOGRAFIA                              | 12          |
| <b>2. CAPÍTULO 02 – CONCEITOS GERAIS</b>                  | <b>13</b>   |
| 2.1. INVENTÁRIO   | 13          |
| 2.2. LINGUAGEM DE SCRIPT                                  | 14          |
| 2.2.1. <i>C Sharp</i>                                     | 15          |
| 2.2.2. <i>Shell Script</i>                                | 15          |
| 2.3. GERENCIAMENTO DE BANCO DE DADOS COMPUTACIONAL        | 15          |
| 2.3.1. <i>My SQL</i>                                      | 16          |
| 2.4. INTERFACE GRÁFICA DE USUÁRIO                         | 17          |
| 2.4.1. <i>PHP</i>   | 17          |
| <b>3. CAPÍTULO 03 – DESENVOLVIMENTO DO APLICATIVO</b>     | <b>18</b>   |
| 3.1. INTRODUÇÃO   | 18          |
| 3.2. RAZÕES DO USO DA TECNOLOGIA ESCOLHIDA                | 19          |
| 3.2.1. <i>Requisitos para execução do aplicativo SIHS</i> | 20          |
| 3.3. FLUXO DE FUNCIONAMENTO                               | 21          |
| 3.4. SCRIPT C #   | 23          |
| 3.5. SCRIPT SHELL   | 28          |
| 3.6. BANCO DE DADOS MySQL                                 | 30          |
| 3.7. INTERFACE GRÁFICA PHP                                | 33          |
| 3.8. APRESENTAÇÃO DO APLICATIVO                           | 35          |
| <b>4. CAPÍTULO 04 – EXPERIMENTOS E RESULTADOS</b>         | <b>42</b>   |
| 4.1. INTRODUÇÃO   | 42          |
| 4.2. IDENTIFICAÇÃO DO PLANO DE TESTE                      | 42          |
| 4.3. CONFIGURAÇÃO DO AMBIENTE DE TESTE                    | 43          |
| 4.4. TESTES REALIZADOS                                    | 45          |
| 4.5. RESULTADOS OBTIDOS E PROBLEMAS ENCONTRADOS           | 44          |
| <b>5. CAPÍTULO 05 – CONCLUSÃO</b>                         | <b>48</b>   |
| 5.1. SUGESTÕES PARA TRABALHOS FUTUROS                     | 49          |
| <b>6. REFERÊNCIAS BIBLIOGRÁFICAS</b>                      | <b>51</b>   |
| <b>7. APÊNDICE A – CÓDIGOS DE PROGRAMAÇÃO</b>             | <b>53</b>   |

## LISTA DE FIGURAS

---

|   |    |
|---|----|
| Figura 3.1 – Fluxo do aplicativo desenvolvido -----   | 20 |
| Figura 3.2 – Criando novo projeto no <i>Visual Studio</i> -----                                 | 22 |
| Figura 3.3 – Adição da referência para uso de WMI e do MySQL-----                               | 23 |
| Figura 3.4 – Variáveis utilizadas no script -----   | 24 |
| Figura 3.5 – Métodos utilizados no script-----  | 24 |
| Figura 3.6 – Código do método <i>escreveBanco</i> -----   | 25 |
| Figura 3.7 – Código do método <i>softwares_instalado</i> -----                                  | 26 |
| Figura 3.8 – Criação de um novo banco de dados-----   | 28 |
| Figura 3.9 – Tabela <i>tb_hardware</i> -----  | 29 |
| Figura 3.10 – Tabela <i>tb_software</i> -----   | 29 |
| Figura 3.11 – Tabela <i>tb_usuarios</i> -----   | 30 |
| Figura 3.12 – <i>index.php</i> -----  | 32 |
| Figura 3.13 – <i>login.php</i> -----  | 32 |
| Figura 3.14 – <i>conexão.php</i> -----  | 33 |
| Figura 3.15 – Tela de <i>login</i> do Sistema de Inventário de Hardware e Software – SIHS ----- | 34 |
| Figura 3.16 – Tela principal do SIHS-----   | 35 |
| Figura 3.17 – Tela <i>Software</i> -----  | 36 |
| Figura 3.18 – Tela <i>Relatórios</i> -----  | 37 |
| Figura 3.19 – Tela <i>Gerenciar Usuários</i> -----  | 38 |
| Figura 3.20 – Tela <i>Gerenciar Usuários</i> -----  | 39 |
| Figura 4.1 – <b>Tela de mensagem de <i>logout</i> do sistema</b> -----                          | 42 |

# ÍNDICE DE SIGLAS E ABREVIATURAS

---

|      |  |
|------|--|
| AR   | Administrador de rede                        |
| C#   | C Sharp                                      |
| DBMS | Database Management Systems                  |
| GUI  | Graphical User Interface                     |
| HTML | Hypertext Markup Language                    |
| PHP  | PHP: Hypertext Preprocessor                  |
| RAM  | Ram Access Memory                            |
| SIHS | Sistema de Inventário de Hardware e Software |
| SO   | Sistema Operacional                          |
| SQL  | Structured Query Language                    |
| VC#  | Visual C Sharp                               |
| VS   | Visual Studio                                |
| WMI  | Windows Management Instrumentation           |
| XML  | eXtensible Markup Language                   |

## RESUMO

---

Este trabalho apresenta um estudo que possibilite criar uma ferramenta que gere um inventário de hardwares e softwares de uma dada rede contendo diferentes sistemas operacionais (Windows e Linux). Por meio de uma interface gráfica, o administrador de redes tem acesso rápido e fácil a todas as informações captadas.

A ferramenta acima citada foi desenvolvida na linguagem de programação C Sharp para a versão Windows e em Shell Script para versão Linux, utilizando PHP para o desenvolvimento da interface gráfica de usuário e *MySQL* para a criação do banco de dados responsável pelo armazenamento de dados coletados.

Palavras-chave: gerência de rede; automatização de tarefas; inventário de *hardware* e *software*.

## **ABSTRACT**

---

This work presents a study which makes possible to create a tool that manages an inventory of hardwares and softwares of a given network containing different operating systems (Windows and Linux). Through a graphic interface, the network administrators has quick and easy access to all the caught informations, as well as to be warned of critical problems found in the net.

The tool above quoted was developed in the programming language C Sharp for the version Windows and in Shell Script for version Linux, using PHP for the development of the graphic interface of user and MySQL for the creation of the database responsible for the storage of collected data.

Key words: network managing; automation of tasks; inventory of hardware and software.

# CAPÍTULO 1 - INTRODUÇÃO

---

## 1.1 MOTIVAÇÃO

A cada dia, os avanços tecnológicos na área da computação fazem com que os recursos computacionais se tornem mais acessíveis a todos. Novas redes de computadores surgem em todos os lugares. Quanto mais redes, mais administradores de redes (AR) são necessários para o gerenciamento desses parques computacionais.

O trabalho de um AR consiste em tarefas que demandam esforços e muitas vezes a perda de tempo desnecessário, como levantar o inventário da rede de forma manual. Visando essa melhoria de desempenho, surgiu a necessidade de uma ferramenta que otimize todo o trabalho, além de gerar dados com mais recursos, para que o AR possa ter um melhor gerenciamento da rede.

Também numa crescente vem o conceito de software livre, não só em usuários comuns, mas grandes órgãos e empresas vêm aderindo ao uso. Um exemplo é o Governo Federal do Brasil. Uma das diretrizes dessa implantação do software livre é priorizar soluções, programas e serviços baseados em softwares livres que promovam a otimização de recursos e investimentos em tecnologia da informação [GOVERNO FEDERAL, 2008].

Tomando conhecimento do trabalho de conclusão de curso de Engenharia de Computação no UniCEUB do graduado David Mascarenhas Watkins com o título de

“Inventário de hardware e registro de uso”, aliado aos avanços tecnológicos dos recursos computacionais e a crescente dos softwares livres, surgiu a necessidade de uma melhoria na ferramenta desenvolvida.

Antes elaborada sob uma rede Windows e analisando apenas o hardware, a nova ferramenta a ser disponibilizada visa aliar os dois lados: software livre e software proprietário, além de uma maior varredura por cada máquina do parque computacional, incluindo os hardwares e softwares existentes, criando assim uma ferramenta mais completa e de maior utilidade para o AR.

## **1.2OBJETIVOS**

O trabalho desenvolvido tem como objetivo melhorar o sistema desenvolvido pelo aluno David Mascarenhas Watkins, disponibilizando informações de hardware e de software em um banco de dados, independente do sistema operacional da rede, para suporte aos administradores de rede.

Os objetivos específicos do trabalho são:

a) Obter informações de hardware e de software de computadores pessoais de uma dada rede assim que ele for ligado, gerando o inventário daquela determinada máquina automaticamente e guardando essas informações no banco de dados.

b) Desenvolver uma interface gráfica de usuário, para que o AR possa ter um rápido acesso às informações geradas e armazenadas pela ferramenta.

### **1.3METODOLOGIA**

O desenvolvimento do projeto começou com a definição da pesquisa bibliográfica, que ajudaram a definir quais os tipos de tecnologia seriam utilizados para a implementação do projeto. Após essa pesquisa, o aplicativo começou a ser elaborado com o desenvolvimento dos *scripts* responsáveis pela coleta de dados em rede Windows e Linux. O banco de dados foi elaborado, criando as tabelas para o armazenamento das informações coletadas. Paralelamente foi desenvolvida a interface gráfica de usuário para que o administrador de redes possa ter um rápido e fácil acesso aos dados obtidos.

Terminada a fase de criação e desenvolvimento do aplicativo, deu-se início a fase de testes. Uma rede foi montada, utilizando Máquinas Virtuais, com o objetivo de verificar o bom funcionamento do aplicativo, verificando as alterações necessárias para um melhor desempenho.

Com os testes e experimentos do aplicativo concluídos, foi possível finalizar a parte escrita da monografia.

## 1.4 ESTRUTURA DA MONOGRAFIA

Este projeto é dividido em cinco capítulos principais, incluindo a introdução. Eles são subdivididos em tópicos conforme a necessidade de descrição mais detalhada. Segue uma breve descrição do que será exposto nos próximos capítulos:

- Capítulo 2: Apresenta informações sobre produtos de inventários disponíveis no mercado e toda a parte de conceitos gerais sobre as tecnologias necessárias para o desenvolvimento do projeto.
- Capítulo 3: Mostra todo o desenvolvimento do aplicativo, especificando as tecnologias escolhidas para a criação do mesmo.
- Capítulo 4: Descreve os testes realizados e resultados obtidos.
- Capítulo 5: Traz a conclusão, além de mostrar sugestões para projetos futuros.

## Capítulo 2 - Conceitos gerais

---

### 2.1 INVENTÁRIO

A principal atribuição do AR é o gerenciamento da rede local, bem como dos recursos computacionais relacionados direta ou indiretamente [WIKIPÉDIA, 2008a]. Manter dados de inventário sempre atualizado tornou-se quase impossível devido à dificuldade quando o AR têm de realizá-lo manualmente.

Para ajudar aos responsáveis pela infra estrutura de redes existem uma série de utilitários que auxiliam esta tarefa nada simples [SUDRÉ , 2008].

- **CACIC - Configurador Automático e Coleta de Informações Computacionais:** Primeiro Software Público do Governo Federal desenvolvido pela Unidade Regional da DATAPREV no Espírito Santo. O CACIC é um sistema capaz de fornecer um diagnóstico preciso do parque computacional, disponibilizando informações centralizadas como o número de equipamentos e sua distribuição nos mais diversos órgãos, configurações de hardware, atualizações de segurança, entre outras. Também pode fornecer informações patrimoniais e a localização física dos equipamentos, ampliando o controle do parque computacional e a segurança na rede [DATAPREV, 2008].

- **Empirum PRO:** Permite o gerenciamento de aplicativos instalados assim como o inventário de hardware de cada estação. O aplicativo ainda oferece o recurso de instalação remota de novas versões do sistema operacional e aplicativos incluindo o backup de todas as informações pessoais dos usuários guardadas no disco local [MATRIX42, 2008].
- **IBM Tivoli Configuration Manager:** Permite que os usuários obtenham controle total sobre software e hardware corporativo. Seu módulo de distribuição de software fornece a capacidade de implementar de forma rápida e eficiente aplicativos complexos críticos para a missão a partir de um ponto central. Após a implementação dos sistemas, o módulo de inventário permite que os usuários varram e coletem automaticamente informações de configuração e hardware e software de sistemas de computadores em sua empresa. [IBM, 2008].

## 2.2 LINGUAGEM DE SCRIPT

Linguagem de *script* são linguagens de programação executadas direto no código fonte dos programas ou até mesmo dentro de outras linguagens de programação. São utilizadas para estender a funcionalidade de um programa e até mesmo controlá-lo [WIKIPÉDIA, 2008b].

### 2.2.1 C SHARP

O *C Sharp* (C#) é uma linguagem de programação desenvolvida pela Microsoft como parte da plataforma .Net. É baseada nas linguagens C++ e inclui aspectos e influência de outras linguagens como Java e Delphi [MSDN, 2008a]. Um dos principais objetivos é promover suporte aos princípios de engenharia de software [ECMA, 2006].

### 2.2.2 SHELL SCRIPT

O *Shell Script* é uma poderosa ferramenta de automação de instruções.

*Script* pode ser definido como arquivos executáveis, com instruções definidas, que são executadas por um interpretador. *Shell* pode ser definido como interpretador de instruções e comandos. [VIVA O LINUX, 2007].

## 2.3 GERENCIADOR DE BANCO DE DADOS COMPUTACIONAL

Bancos de dados computacionais são coleções de dados estruturados que são armazenados em um computador. Esses dados são utilizados por programas ou por

usuários através de linguagens de *query* e são gerenciados por sistemas chamados de *Database Management Systems (DBMS)*.

Um DBMS é composto por quatro itens-chave. Uma linguagem de modelagem é responsável por definir o esquema em que os bancos de dados serão armazenados de acordo com o modelo de dados do DBMS. Ele também deve ter estruturas de dados, composto por campos, arquivos e objetos, otimizado para lidar com grandes quantidades de informações. Existe também uma linguagem de *query* que permite que o administrador de banco de dados obtenha dados armazenados, faça análise deles e atualize campos de forma interativa. Finalmente, existe também um mecanismo de transação responsável por manter a integridade de dados [GEORGE, DELANO. 2006].

### **2.3.1 MySQL**

O MySQL é um servidor e gerenciador de bando de dados (SGBD) relacional que utiliza a linguagem SQL como interface. Possui licença dupla, sendo uma delas de software livre [MILANI, 2006].

Tornou-se o banco de dados open source mais utilizado no mundo, com mais de seis milhões de instalações entre Websites, datawarehouse, aplicações comerciais e outras mais. Alguns de seus usuários são o Yahoo! Finance, Motorola, Nasa e Wikipédia [MYSQL, 2008].

## 2.4 INTERFACE GRÁFICA DE USUÁRIO

Interfaces gráficas de usuário, também conhecidas como *Graphical User Interface* (GUI), são tipos de interface de usuário responsáveis por fornecer meios para que usuários possam interagir com computadores e dispositivos controlados por computadores. Elementos gráficos com ícones, indicadores visuais ou elementos gráficos especiais são utilizados para facilitar a interação usuário/máquina.

### 2.4.1 PHP

PHP, que significa "PHP: Hypertext Preprocessor", é uma linguagem de programação de ampla utilização, interpretada, que é especialmente interessante para desenvolvimento para a Web e pode ser mesclada dentro do código HTML. A sintaxe da linguagem lembra C, Java e Perl, e é fácil de aprender. O objetivo principal da linguagem é permitir a desenvolvedores escreverem páginas que serão geradas dinamicamente [NIEDERAUER, 2008].

Uma das grandes vantagens do PHP é que ele é gratuito e com código aberto. Tanto o arquivo de instalação quanto o código-fonte podem ser baixados diretamente do site <http://www.php.net> [PHP, 2008].

## Capítulo 3 - Desenvolvimento do Aplicativo

---

### 3.1 INTRODUÇÃO

O aplicativo desenvolvido, chamado de Sistema de Inventário de Hardware e Software (SIHS), tem o objetivo de auxiliar o AR na otimização de seu trabalho, pois, dependendo do tamanho da rede sob sua responsabilidade, exigirá várias horas de trabalho.

Através do script que será rodado em cada máquina da rede assim que o usuário fizer o login, do banco de dados para salvar os dados obtidos durante a execução do script e da interface gráfica de usuário que será capaz de mostrar todos os dados obtidos de forma clara e objetiva, o AR terá em suas mãos importantes informações necessárias para o gerenciamento de sua rede.

Sem fins comerciais, o projeto em questão visa apenas facilitar o trabalho de um AR, além de demonstrar que as tecnologias escolhidas para o desenvolvimento do mesmo possibilitam a criação de tal solução e que os objetivos propostos podem ser atingidos através dessas tecnologias.

Este capítulo aborda os seguintes itens:

- O porquê determinadas tecnologias usadas foram escolhidas;
- Quais são os requisitos computacionais para o desenvolvimento do aplicativo;

- Demonstração do fluxo de funcionamento do aplicativo;
- Desenvolvimento do *script* utilizando *C#* e *Shell Script*;
- Desenvolvimento do banco de dados utilizando *MySQL*;
- Desenvolvimento da interface gráfica de usuário utilizando *PHP*;
- Apresentação do aplicativo;

### 3.2 RAZÃO DO USO DA TECNOLOGIA ESCOLHIDA

Dando continuidade ao projeto do graduado David Mascarenhas Watkins, o *script* previamente gerado por ele em *C#*, passou por algumas mudanças para adequação à nova proposta e melhoria do mesmo, mantendo a mesma linguagem de programação. Já para o *script* com comandos Linux, o *Shell Script* foi utilizado para seu desenvolvimento.

Por ter a premissa de rodar em ambientes diversos com diferentes SO, incluindo softwares livres, o graduado optou pela mudança do DBMS *SQL Server 2005* para o *MySQL*, aproveitando a experiência prévia adquirida em cadeira do curso e trabalhos profissionais.

Finalmente, aproveitando a facilidade da linguagem aliado à experiência adquirida, a linguagem *PHP* foi escolhida para o desenvolvimento da porção GUI do aplicativo aqui proposto.

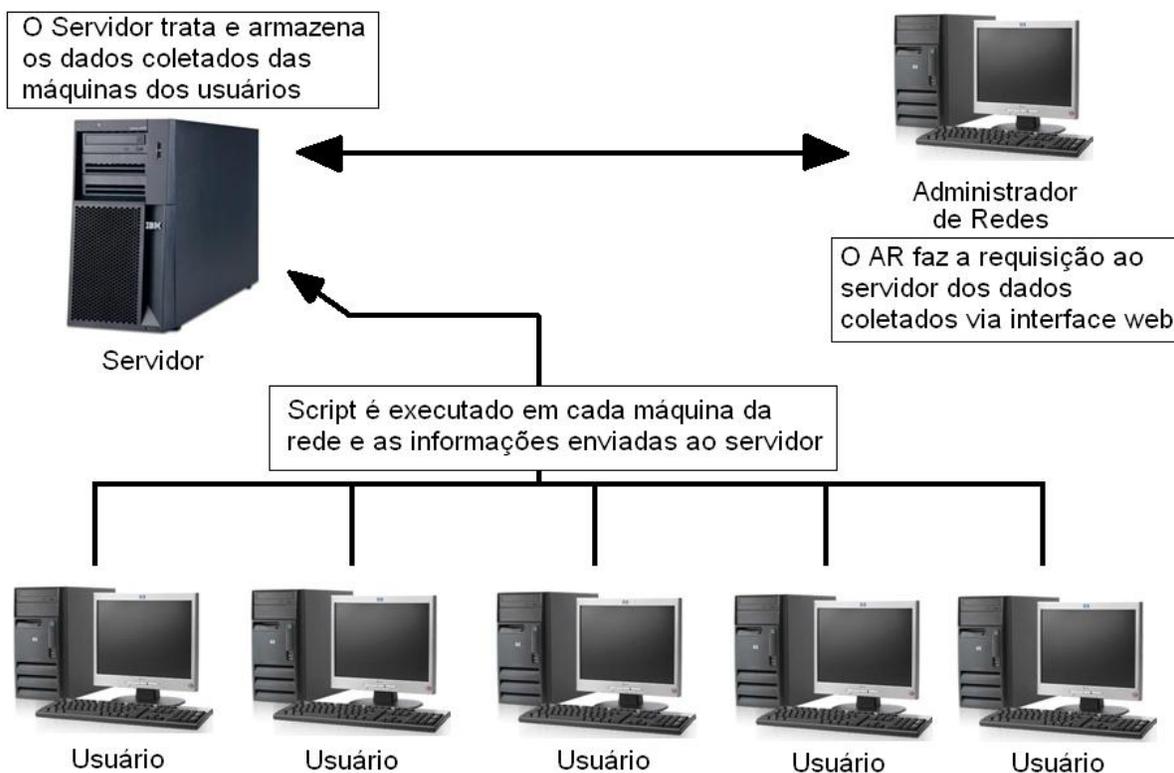
### 3.2.1 Requisitos para execução do aplicativo SIHS

Para o correto funcionamento do aplicativo SIHS, alguns requisitos devem ser observados e seguidos.

- Para o *Visual Studio 2005* [MICROSOFT, 2007a]
  - Processador de 600 MHz ou mais veloz; 192 MB de memória RAM ou mais; 1 GB de disco rígido disponível ou mais; Sistema operacional *Windows 2000 SP4, Windows XP SP2, Windows Server 2003 ou Windows Vista*; Drive de CD ou DVD; Vídeo de 800x600, 256 cores ou melhor; *Mouse*.
- Para o *MySQL*:
  - Por ser um aplicativo leve e de fácil instalação, o *MySQL* pode rodar em praticamente todos os computadores existentes no mercado.
- Para o computador atuando como servidor:
  - Atender requisitos do *MySQL*; Estar conectado à mesma rede dos computadores cliente.
- Para os computadores atuando como cliente:
  - Clientes *Windows*: *.NET Framework 2.0*; Estarem conectados à mesma rede do computador servidor; Nível de permissão de usuário alto para acesso a dados requisitados.
  - Clientes *Linux*: Instalar o *Ishw*. Estarem conectados à mesma rede do computador servidor; Nível de permissão de usuário alto para acesso a dados requisitados.

### 3.3 FLUXO DE FUNCIONAMENTO

Na figura 3.1 é apresentada uma visão geral do fluxo de funcionamento do aplicativo proposto.



**Figura 3.1 – Fluxo do aplicativo desenvolvido.**

O *Script* desenvolvido é responsável pela coleta de dados dos computadores de cada usuário da rede proposta. Esses dados são coletados assim que o usuário realiza o login e enviados diretamente para o servidor, que faz o tratamento apropriado e armazena-os em suas respectivas tabelas. Por último, o AR faz a solicitação direta ao servidor dos dados coletados anteriormente. Essa interação se dá por meio da interface web.

Os passos enumerados abaixo se referem ao processo de funcionamento do aplicativo em um ambiente onde os computadores dos usuários já estão configurados com o *script* e o banco de dados está configurado e ativado em um computador atuando como servidor.

**1º passo** – O usuário liga o computador. Quando o sistema operacional termina de carregar e o usuário realiza o login, o *script* é executado coletando os dados requisitados e gerando uma mensagem para ser enviada ao servidor.

**2º passo** – A mensagem é enviada do computador cliente ao servidor, pois *script* contém dados sobre o número IP do servidor.

**3º passo** – Banco de dados faz o correto tratamento de mensagem recebida de computadores cliente e os armazena em tabelas pré-estruturadas.

**4º passo** – Por meio de um de *web browser*, o administrador de rede se conecta ao servidor, através de página gerada em PHP, que solicita seu *login* e senha.

**5º passo** – *Login* e senha de administrador são enviados ao servidor e verificados em tabela do banco de dados.

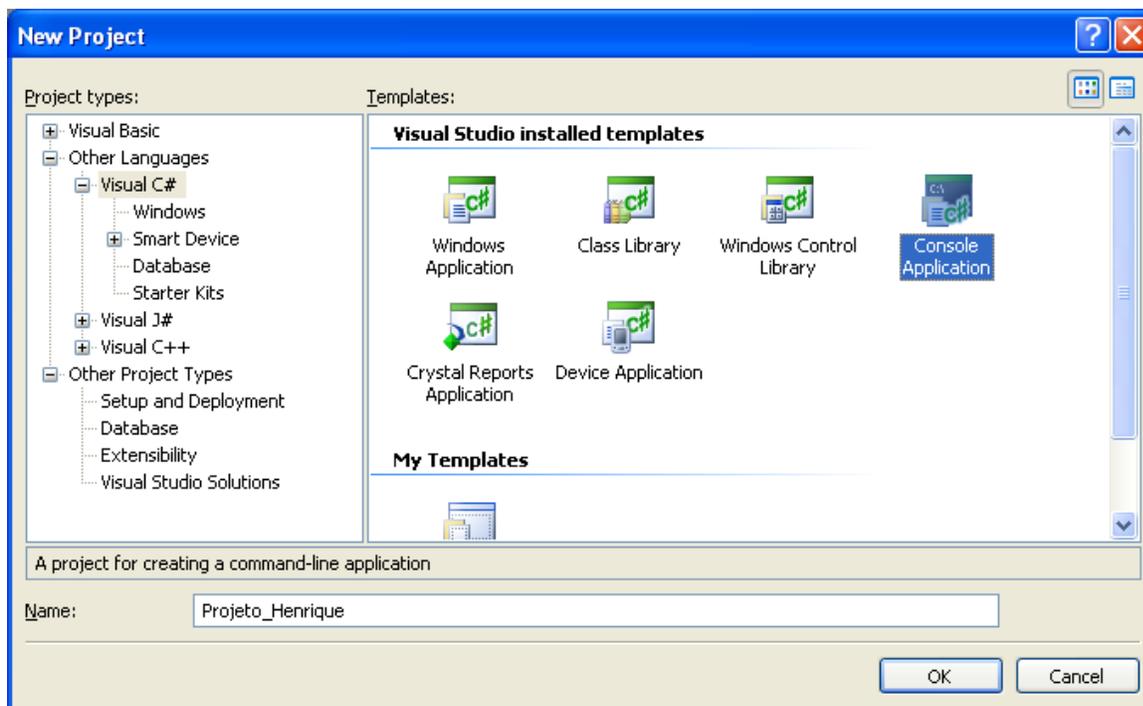
**6º passo** – Se os dados fornecidos pelo administrador de rede conferem com tabela em banco de dados, o *MySQL* disponibiliza acesso a dados armazenados nele.

**7º passo** – Com dados disponibilizados pelo DBMS, a página *web* PHP é composta com dados armazenados no banco de dados.

### 3.4 SCRIPT C#

Os passos enumerados abaixo constituem o processo de desenvolvimento da primeira porção do aplicativo proposto por esse projeto, o *script* em C# através da ferramenta *Visual Studio 2005*.

**1º passo** – Como demonstrado na figura 3.2 abaixo, foi criado um novo projeto no *Visual Studio 2005* com o template *Console Application* denominado *Projeto\_Henrique*.



Figura

#### 3.2 – Criando novo projeto no *Visual Studio*.

**2º passo** – É adicionada ao projeto *Projeto\_Henrique* a referência *.NET System.Management*, que possibilita o uso de chamadas WMI no *script*, além da

referências *MySql.Data*, que possibilitam a conexão com o banco de dados MySQL, como ilustrado na figura 3.3.

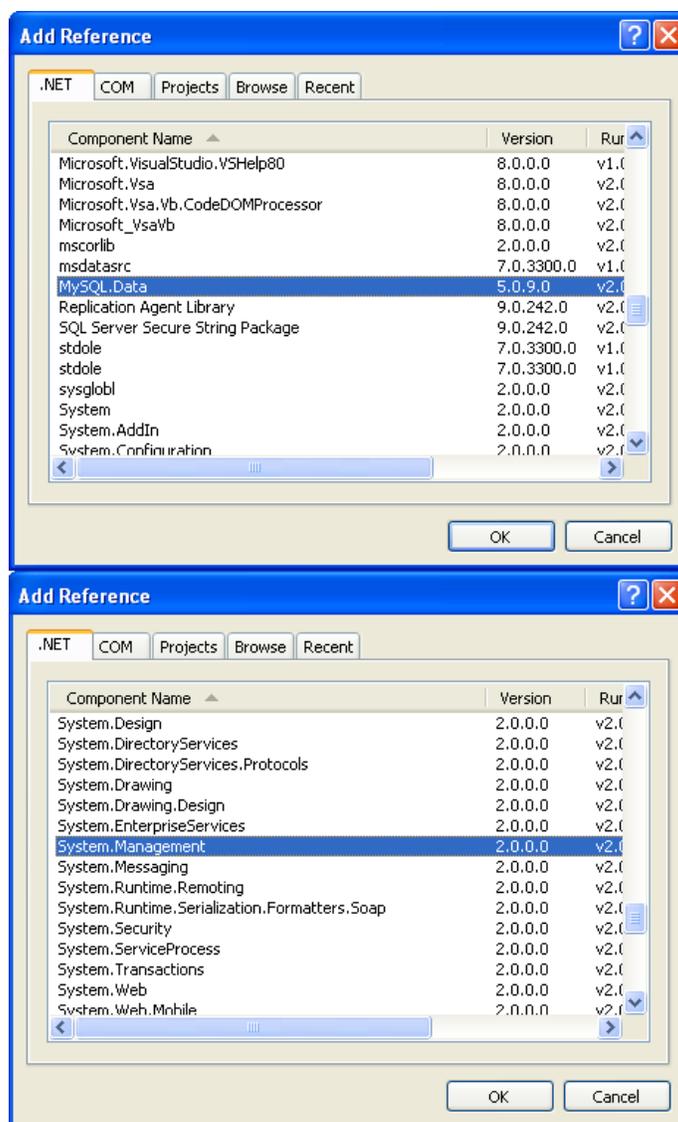


Figura 3.3 – Adição da referência para uso de WMI e do MySQL

**3º passo** – Criada a classe *Projeto\_Final\_Henrique*

**4º passo** – Definir as variáveis utilizadas no *script* de modo global (conforme figura 3.4) já que as coletas de dados foram divididas em métodos diferentes.

```

//variaveis globais devido a coleta de dados ser dividida em métodos diferentes
public int ram_MB = (1024);
public int ram_Pentes = 0;
public Int64 ram_Capacidade = 0;
public long ram_Total = 0;

public String hd_Modelo = "";
public int hd_MB = (1024 * 1024 * 1024);
public int hd_Quantidade = 0;
public Int64 hd_Capacidade = 0;
public long hd_Total = 0;

public String mouse_Fabricante = "";
public String teclado_Tipo = "";
public String monitor_Tipo = "";
public String monitor_Identificacao = "";

public String rede_MAC = "";
public String rede_Nome = "";

public String cpu_Fabricante = "";
public String cpu_ID = "";
public Int64 cpu_Velocidade = 0;
public Int64 cpu_Familia = 0;

public String pm_Fabricante = "";
public String pm_Modelo = "";
public String pm_Serial = "";

public String idatual = "";

public String sw_Nome = "";
public String sw_Data_Instalacao = "";
public String sw_Versao = "";

public DateTime tempoliga = DateTime.Now;

```

Figura 3.4 – Variáveis utilizadas no script

**5º passo** – Os métodos utilizados pelo script são nomeados, conforme a figura 3.5. É necessário instanciar a própria classe *Projeto\_Final\_Henrique* e fazer referência a seus métodos secundários no método *Main*. A não-invocação de métodos estáticos diretamente é um requisito da orientação a objetos em C# [MARSHALL, 2006].

```

static void Main(string[] args)
{
    IntPtr winHandle = System.Diagnostics.Process.GetCurrentProcess().MainWindowHandle;
    ShowWindow(winHandle, SW_MINIMIZE);

    Projeto_Final_Henrique projeto = new Projeto_Final_Henrique();
    projeto.memoria_Ram();
    projeto.hd();
    projeto.mouse();
    projeto.teclado();
    projeto.monitor();
    projeto.rede();
    projeto.cpu();
    projeto.placa_Mae();
    projeto.escreveBanco("Database=ceub;Server=localhost;User Id=root;Password=");
    projeto.softwares_instalados("Database=ceub;Server=localhost;User Id=root;Password=");
    Thread t = new Thread(new ThreadStart(projeto.escreveDesliga));
    t.Start();
}

```

Figura 3.5 – Métodos utilizados no script

**6º passo** – Os métodos *statsram*, *statshd*, *statspointingdevice*, *statskeyboard*, *statsmonitor*, *statsnetworkadapter*, *statscpu* e *statsmotherboard* foram desenvolvidos no projeto anterior e foram reaproveitados, apenas mudando o nome para

*memoria\_Ram, hd, mouse, teclado, monitor, rede, cpu e placa\_Mae* respectivamente.

Já o método `escreveBanco` foi alterado devido conexão com o DBMS *MySQL*, vide figura 3.6.

```
private void escreveBanco(string myConnectionString)
{
    myConnectionString = "Database=ceub;Data Source=localhost;User Id=root;Password=";
    MySqlConnection myConnection = new MySqlConnection(myConnectionString);
    string myInsertQuery = "INSERT INTO tb_hardware values (" + totalram + "," + rammb + "," + rampentes + "," + hdcc";
    MySqlCommand myCommand = new MySqlCommand(myInsertQuery);
    myCommand.Connection = myConnection;
    try
    {
        myConnection.Open();
    }
    catch (Exception e)
    {
        Console.WriteLine(e.ToString());
    }
    try {
        myCommand.ExecuteNonQuery();
    }
    catch (Exception e)
    {
        Console.WriteLine(e.ToString());
    }
    try{
        MySqlDataReader leitor = null;
        MySqlCommand comando = new MySqlCommand("select id from tb_hardware order by id desc", myConnection);
        leitor = comando.ExecuteReader();
        while (leitor.Read()){
            idatual = leitor["id"].ToString();
        }
    }
    catch (Exception e){
        Console.WriteLine(e.ToString());
    }
    myCommand.Connection.Close();
}
```

**Figura 3.6 – Código do método `escreveBanco`**

**7º passo** – Seguindo o princípio de cada método e utilizando a classe *WMI Win32\_Product* correspondente aos softwares, foi desenvolvido o método `softwares_instalados`, recuperando dados sobre o nome, a data da instalação e a versão. A figura 3.7 demonstra esse método.

```

private void softwares_instalados()
{
    ManagementObjectSearcher busca = new ManagementObjectSearcher("SELECT * FROM Win32_Product");
    foreach (ManagementObject result in busca.Get())
    {
        name = Convert.ToString(result.GetProperty("Name"));
        installdate = Convert.ToString(result.GetProperty("InstallDate"));
        version = Convert.ToString(result.GetProperty("Version"));
        Console.WriteLine("Nome do Software: {0}", name);
        Console.WriteLine("Data de instalacao: {0}", installdate);
        Console.WriteLine("Versao: {0}", version);
        Console.WriteLine("");
    }
}

```

**Figura 3.7 – Código do método *softwares\_instalados***

**8º passo** – A *string* de conexão com o banco de dados é desenvolvida. São inseridas informações sobre a localização do banco de dados, nome de usuário e senha para que a mensagem gerada pela *string* possa ser inserida no servidor.

**9º passo** – A *string* de dados recolhida é estruturada para que ela possa ser inserida nos campos corretos do banco de dados. Para sua construção, são concatenadas as estruturas de inserção de dados (cláusula *insert*) SQL com os valores das variáveis recolhidas.

**10º passo** – Finalmente, gera-se o executável do *script* para que ele possa ser instalado nos computadores cliente.

### 3.5 Script *Shell*

Os passos enumerados abaixo constituem o processo de desenvolvimento da segunda porção do aplicativo proposto por este projeto, o *script* em *Shell Script*.

**1º passo** – Utilizando diretamente a linha de comando do Linux, o script foi desenvolvido.

**2º passo** – Foi instalada a ferramenta *lshw*, que é capaz de fornecer informações detalhadas sobre os hardwares instalados na máquina. Após a instalação, salvou-se as informações obtidas num arquivo temporário para que os dados necessários ao aplicativo pudessem ser facilmente recuperados.

**3º passo** – A seguir, cada parte do script começou a ser desenvolvida. Cada coleta de dados necessários para o inventário é dividida em diferentes comandos do Linux: como *fdisk* para coletar dados de disco (capacidade, quantidade), *ifconfig* para dados de rede (nome do dispositivo e endereço mac) entre outros.

**4º passo** – Definido o script com todas as informações coletadas, foi criada a estrutura em formato XML. A figura 3.8 abaixo mostra a estrutura do XML gerado.

```

- <hardware>
- <memoria>
- <itens>
  <memoria_total>126212</memoria_total>
  <qtde_pentes>1</qtde_pentes>
</itens>
</memoria>
- <hd>
- <itens>
  <hd_modelo>VMware Virtual S Rev: 1.0</hd_modelo>
  <hd_qtde>1</hd_qtde>
  <hd_capacidade>21.4 GB,</hd_capacidade>
</itens>
</hd>
- <mouse>
- <itens>
  <mouse_fabricante>"ImPS/2 Generic Wheel Mouse"</mouse_fabricante>
</itens>
</mouse>
- <teclado>
- <itens>
  <teclado_fabricante>"AT Translated Set 2 keyboard"</teclado_fabricante>
</itens>
</teclado>
- <monitor>
- <itens>
  <monitor_tipo>"SVGA"</monitor_tipo>
  <monitor_identificacao>"Virtual Monitor"</monitor_identificacao>
</itens>
</monitor>
- <rede>
- <itens>
  <rede_nome>eth0</rede_nome>
  <rede_mac>00:0C:29:A0:71:E2</rede_mac>

```

**Figura 3.8 – XML gerado com dados de hardware do Linux**

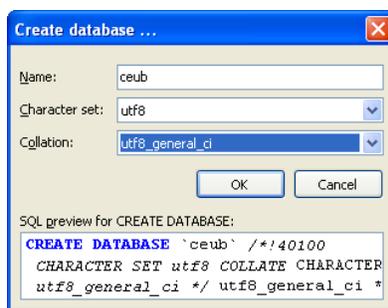
**5º passo** – Habilita-se o FTP para que o arquivo gerado possa ser enviado para o banco de dados no servidor. Foi utilizado o software livre *FileZilla* como cliente FTP. As configurações foram o *host* apontando para o servidor, *usuário* e *senha* para ter acesso ao servidor, além dos arquivos gerados pelo script que serão transferidos para o banco de dados.

**6º passo** – Finalmente, gera-se o executável do *script* para que ele possa ser instalado nos computadores cliente.

### 3.6 BANCO DE DADOS MySQL

Utilizando o mesmo nome das variáveis criadas no script, a criação das tabelas do banco de dados seguiu os passos enumerados abaixo, utilizando o DBMS *MySQL*.

**1º passo** - É criado um novo banco de dados denominado *ceub*, conforme figura 3.9



**Figura 3.9 – Criação de um novo banco de dados**

**2º passo** – A tabela *tb\_hardware* é criada. O campo *id* é um identificador com auto-incrementação e é utilizado como a chave principal. Os demais campos são os mesmo criados no passo 4 da geração do script em *C#*, excluindo apenas os que fazer referência aos softwares. Conforme figura 3.10, os campos seguem o tipo definido anteriormente.

| Name                  | Type         | Null | Extra          |
|-----------------------|--------------|------|----------------|
| id                    | int(11)      | No   | auto_increment |
| ram_Total             | bigint(20)   | Yes  |                |
| ram_Pentes            | int(11)      | Yes  |                |
| hd_Total              | bigint(20)   | Yes  |                |
| hd_Quantidade         | int(11)      | Yes  |                |
| hd_Modelo             | varchar(255) | Yes  |                |
| mouse_Fabricante      | varchar(255) | Yes  |                |
| teclado_Fabricante    | varchar(255) | Yes  |                |
| monitor_Tipo          | varchar(255) | Yes  |                |
| monitor_Identificacao | varchar(255) | Yes  |                |
| rede_MAC              | varchar(255) | Yes  |                |
| rede_Nome             | varchar(255) | Yes  |                |
| cpu_Fabricante        | varchar(255) | Yes  |                |
| cpu_ID                | varchar(255) | Yes  |                |
| cpu_Velocidade        | bigint(20)   | Yes  |                |
| cpu_Familia           | bigint(20)   | Yes  |                |
| pm_Fabricante         | varchar(255) | Yes  |                |
| pm_Modelo             | varchar(255) | Yes  |                |
| pm_Serial             | varchar(255) | Yes  |                |
| tempoliga             | varchar(50)  | Yes  |                |
| tempodesliga          | varchar(50)  | Yes  |                |

Figura 3.10 – Tabela *tbHardware*

**3º passo** – A tabela *tbSoftware* é criada com o campo *idSoftware* sendo a chave principal com auto-incremento. Ela se relaciona com a *tbHardware* através do campo *id*, existente também na tabela criada, conforme figura 3.11.

| Name               | Type         | Null | Extra          |
|--------------------|--------------|------|----------------|
| idSoftware         | int(11)      | No   | auto_increment |
| id                 | int(11)      | Yes  |                |
| sw_Nome            | varchar(255) | Yes  |                |
| sw_Data_Instalacao | varchar(255) | Yes  |                |
| sw_Versao          | varchar(255) | Yes  |                |

Figura 3.11 – Tabela *tbSoftware*

**4º passo** - Como o Linux não informa quais softwares foram instalados e sim os pacotes de cada um, ficaria uma lista muito extensa de softwares para ser inserida no banco. Para facilitar, criou-se a tabela *tbSwLinux* que irá armazenar arquivos texto contendo todos os pacotes instalados na máquina Linux. Os campos da tabela são *idSwLinux* como chave principal e auto-incremento, o campo *txtLinux*, do tipo

*longtext*, que irá armazenar o arquivo com a extensão *.txt* gerado e o campo *rede\_MAC* do tipo *varchar* conforme figura abaixo.

**Table-Properties for ceub: tb\_sw\_linux**

| Name        | Type         | Null | Extra          |
|-------------|--------------|------|----------------|
| id_sw_linux | int(11)      | No   | auto_increment |
| txt_linux   | longtext     | No   |                |
| rede_MAC    | varchar(255) | Yes  |                |

**Figura 3.12 – Tabela *tb\_sw\_linux***

**6º passo** - Uma tabela *tb\_usuario* foi criada com o intuito de guardar os dados do usuário que terá acesso aos dados via interface web. Contém o campo *id\_usuario* como chave principal e auto-incremento, além dos campos *login\_usuario*, *senha\_usuario*, *nome\_usuario* e *email\_usuario*, visualizados na figura 3.13

**Table-Properties for ceub: tb\_usuarios**

| Name          | Type        | Null | Extra          |
|---------------|-------------|------|----------------|
| id_usuario    | int(11)     | No   | auto_increment |
| login_usuario | varchar(50) | No   |                |
| senha_usuario | varchar(50) | No   |                |
| nome_usuario  | varchar(50) | No   |                |
| email_usuario | varchar(50) | No   |                |

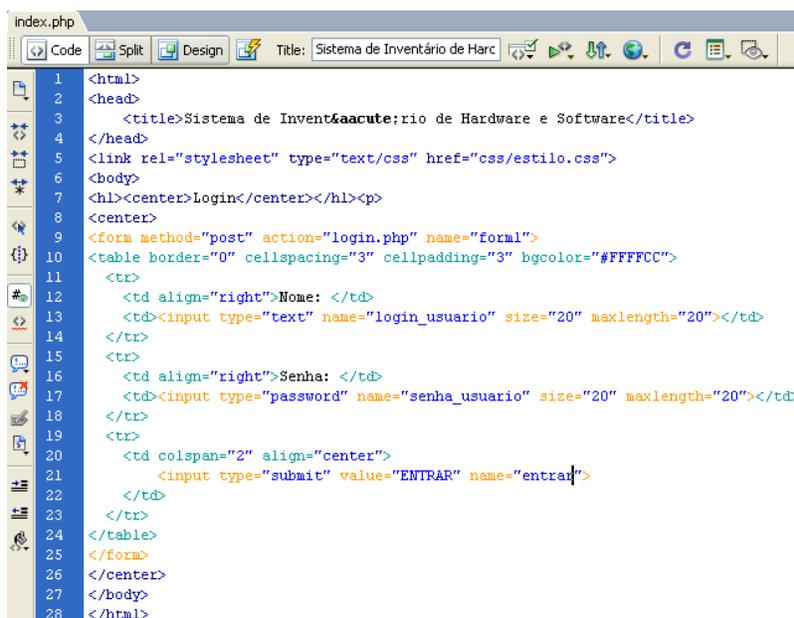
**Figura 3.13 – Tabela *tb\_usuarios***

**7º passo** – Por último o script é executado para verificar a correta inserção dos dados de hardware e software coletados.

### 3.7 INTERFACE GRÁFICA PHP

Por último foi desenvolvida uma interface gráfica de usuário (GUI) para *web* do aplicativo utilizando a linguagem PHP combinada com HTML (*Hypertext Markup Language*). A primeira parte é a autenticação do usuário no banco de dados, dando a permissão para que ele possa visualizar os dados coletados pelo script. A segunda mostra o sistema por completo, com suas opções de menus. Abaixo estão descritos os passos necessários para seu desenvolvimento.

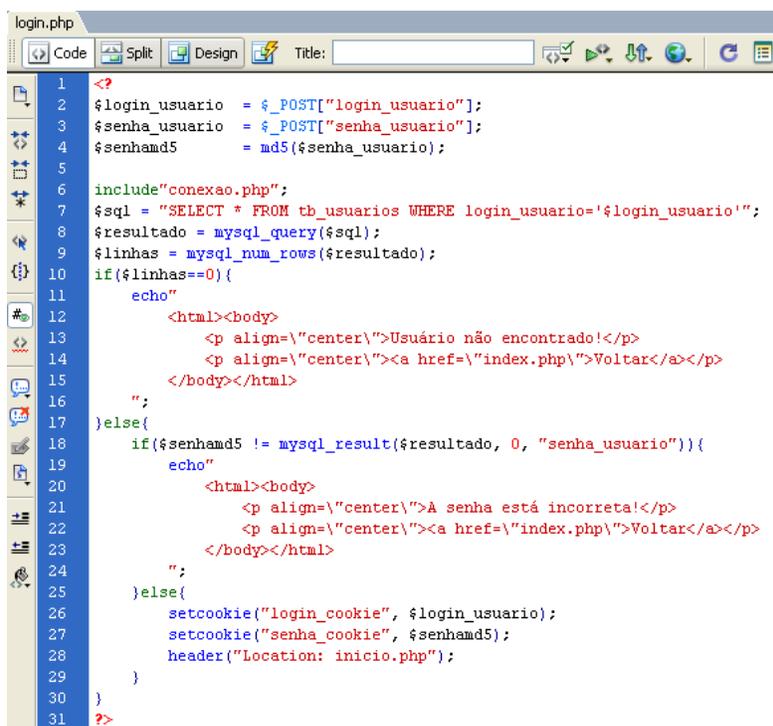
**1º passo** – Foi criado o documento *index.php*, gerado no padrão HTML contém os campos para que o usuário possa informar seu *login* e *senha* para entrar no sistema. A figura 3.14 mostra o código da página.



```
1 <html>
2 <head>
3   <title>Sistema de Inventário de Hardware e Software</title>
4 </head>
5 <link rel="stylesheet" type="text/css" href="css/estilo.css">
6 <body>
7 <h1><center>Login</center></h1><p>
8 <center>
9 <form method="post" action="login.php" name="form1">
10 <table border="0" cellspacing="3" cellpadding="3" bgcolor="#FFFFCC">
11   <tr>
12     <td align="right">Nome: </td>
13     <td><input type="text" name="login_usuario" size="20" maxlength="20"></td>
14   </tr>
15   <tr>
16     <td align="right">Senha: </td>
17     <td><input type="password" name="senha_usuario" size="20" maxlength="20"></td>
18   </tr>
19   <tr>
20     <td colspan="2" align="center">
21       <input type="submit" value="ENTRAR" name="entrar">
22     </td>
23   </tr>
24 </table>
25 </form>
26 </center>
27 </body>
28 </html>
```

Figura 3.14 – *index.php*

**2º passo** – A validação do usuário é feita na página *login.php*, onde os valores informados nos campos *login* e *senha* são checados diretamente no banco de dados. A figura 3.15 mostra essa verificação.



```

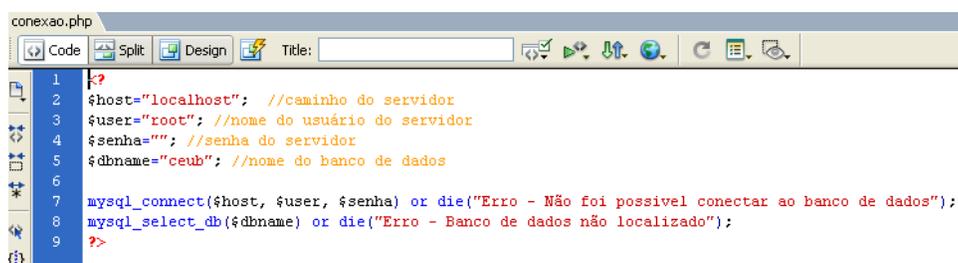
1  <?
2  $login_usuario = $_POST["login_usuario"];
3  $senha_usuario = $_POST["senha_usuario"];
4  $senhamd5     = md5($senha_usuario);
5
6  include"conexao.php";
7  $sql = "SELECT * FROM tb_usuarios WHERE login_usuario='".$login_usuario'";
8  $resultado = mysql_query($sql);
9  $linhas = mysql_num_rows($resultado);
10 if($linhas==0){
11     echo"
12     <html><body>
13         <p align=\"center\">Usuário não encontrado!</p>
14         <p align=\"center\"><a href=\"index.php\">Voltar</a></p>
15     </body></html>
16 ";
17 }else{
18     if($senhamd5 != mysql_result($resultado, 0, "senha_usuario")){
19         echo"
20         <html><body>
21             <p align=\"center\">A senha está incorreta!</p>
22             <p align=\"center\"><a href=\"index.php\">Voltar</a></p>
23         </body></html>
24 ";
25     }else{
26         setcookie("login_cookie", $login_usuario);
27         setcookie("senha_cookie", $senhamd5);
28         header("Location: inicio.php");
29     }
30 }
31 ?>

```

**Figura 3.15 – login.php**

**3º passo** – Definido o acesso de usuários, inicia-se a página inicial. Utilizando *frames*, a tela inicial é dividida em quatro partes: *superior.php*, *principal.php*, *menu.php* e *rodape.php*. Na página *superior.php* é informado apenas o nome do sistema (Sistema de Inventário de Hardware e Software). Na *principal.php*, a tela de boas vindas é exibida junto com o nome do usuário logado no momento. Na *menu.php* é criado os links para as páginas principais do aplicativo: *Software*, *Hardware*, *Relatórios*, *Usuários* e *Logout*. Já na *rodape.php* é informado a data atual e as informações relativas sobre o autor que desenvolveu o aplicativo.

**4º passo** – A página *Software* é desenvolvida a seguir, mostrando as opções de *Busca por Máquina Windows*, *Busca por Máquina Linux* e *Possíveis Alterações*. Para que o sistema possa realizar as devidas buscas, é necessário criar a forma de acesso ao banco de dados, criando o arquivo *conexao.php* (vide figura 3.16) onde são passados os seguintes parâmetros para conectar ao banco de dados: caminho do servidor, nome do usuário do servidor, senha do servidor e nome do banco de dados. Após essa conexão, é possível realizar a busca por quaisquer informações contida no banco através de comandos *SQL*. Com os dados recuperados no banco, é mostrado através de tabelas criadas diretamente no *HTML* e populadas via *PHP*.



```
conexao.php
1  <?
2  $host="localhost"; //caminho do servidor
3  $user="root"; //nome do usuário do servidor
4  $senha=""; //senha do servidor
5  $dbname="ceub"; //nome do banco de dados
6
7  mysql_connect($host, $user, $senha) or die("Erro - Não foi possível conectar ao banco de dados");
8  mysql_select_db($dbname) or die("Erro - Banco de dados não localizado");
9  ?>
```

**Figura 3.16 – conexao.php**

**5º passo** – Para cada página do *SIHS* é repetido o 4º passo, diferenciando apenas nos dados solicitados, nas *queries* implementadas e formas de visualização em tabelas. Termina-se assim o desenvolvimento da *GUI*.

### 3.8 APRESENTAÇÃO DO APLICATIVO

As telas de funcionamento do aplicativo Sistema de Inventário de Hardware e Software (SIHS) serão apresentadas a seguir, especificando suas descrições e funcionalidades.

Ao acessar o aplicativo, a primeira interação com o usuário é a tela de *login*, conforme figura 3.17.



**Figura 3.17 – Tela de *login* do Sistema de Inventário de Hardware e Software - SIHS**

Após o AR ter informado seus dados, o sistema faz uma verificação direta no banco de dados, autorizando ou não o acesso ao aplicativo.

Caso seja liberado, o AR é remetido à página principal, apresentada na figura 3.18 abaixo.



Figura 3.18 – Tela principal do SIHS

Utilizando o menu lateral, o AR tem acesso às informações das seguintes maneiras:

- *Software*
  - Busca por Máquina Windows: Após informar o endereço MAC de que se deseja consultar, o sistema mostra os resultados em forma de tabela conforme busca solicitada.
  - Busca por Máquina Linux: Segue o mesmo padrão da Busca por Máquina Windows.

- Possíveis Alterações: O AR informa a data e o sistema gera todos os registros coletados de *software* naquele período.



Figura 3.19 – Tela *Software*

- *Hardware*
  - Segue o mesmo princípio do *Software*, existindo a Busca por Máquina Windows, Linux e as Possíveis Alterações agora trazendo os dados de *hardwares* existentes.

- Relatórios
  - Relatório por Máquina Windows: Seleciona-se o endereço MAC e o sistema gera um relatório contendo todos os dados de *hardware* e *software* do computador especificado.
  - Relatório por Máquina Linux: Segue o mesmo padrão da Busca por Máquina Windows.
  - Relatório Geral: Mostra todos os dados coletados de todos os computadores existentes na rede.



Figura 3.20 – Tela *Relatórios*

- Usuários
  - São listadas três opções: *Incluir Usuário*, *Editar/Excluir Usuário* e *Listar Todos Usuários*. Nessas opções o AR tem total controle das pessoas que tem acesso aos dados coletados anteriormente.

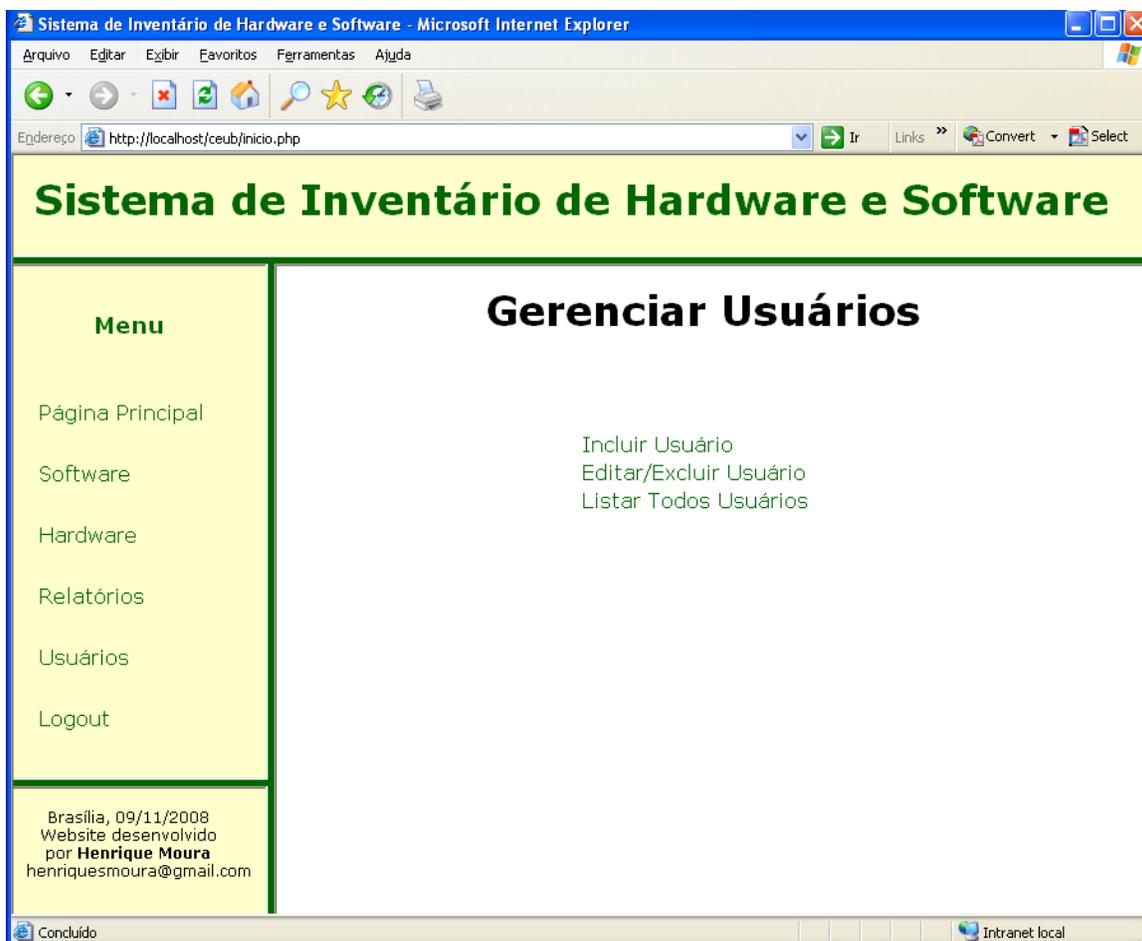


Figura 3.21 – Tela *Gerenciar Usuários*

- *Logout*
  - Ao clicar em *logout*, é perguntando ao AR se ele deseja realmente sair do sistema. Caso afirmativo, a janela é fechada e remetida à tela de *login*. Caso negativo continua-se na mesma página em que o AR estiver analisando.

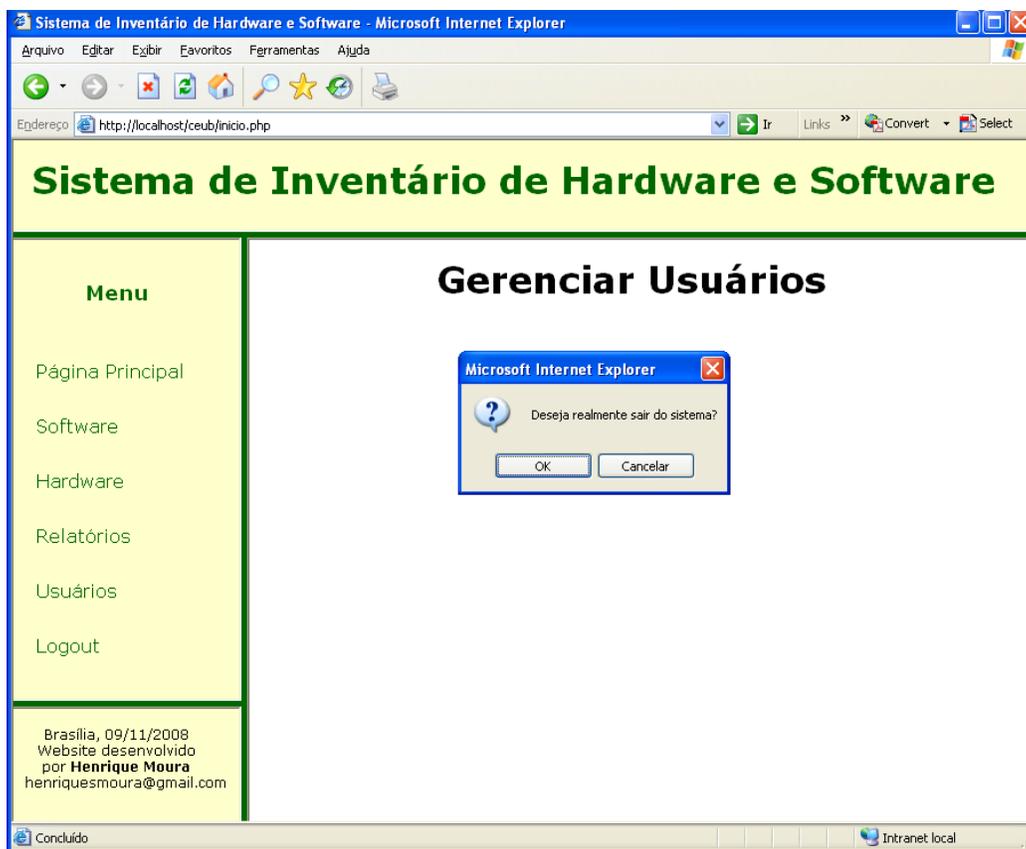


Figura 3.22 – Tela de mensagem de *logout* do sistema

## Capítulo 4 - Experimentos e Resultados

---

### 4.1 INTRODUÇÃO

Após toda a implementação e desenvolvimento do aplicativo apresentado nos capítulos anteriores, este capítulo tem o intuito de demonstrar os resultados obtidos e testar a funcionalidade do aplicativo SIHS.

### 4.2 IDENTIFICAÇÃO DO PLANO DE TESTE

Uma rede privada foi configurada com três computadores utilizando um roteador *DI-604 4-Port Broadband Router* da marca *D-Link* para a devida comunicação. Destes três computadores um foi configurado como servidor enquanto os outros dois como computadores usuários, sendo um com o SO *Linux* e o outro com *Windows*.

Os seguintes itens de *hardware* foram coletados:

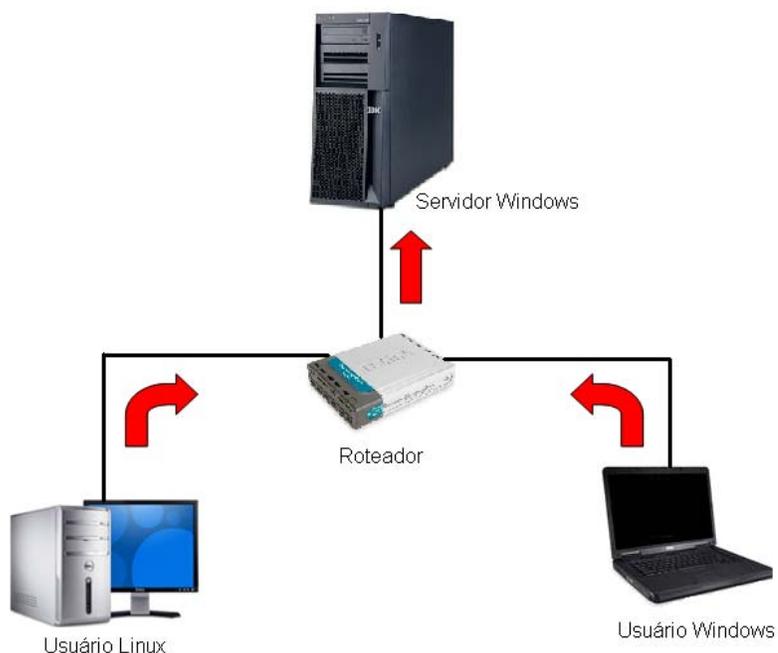
- Memória RAM:
  - Quantidade de pentes; Capacidade total;
- Disco Rígido:
  - Capacidade total; Quantidade; Modelo;
- Mouse:

- Fabricante;
- Teclado:
  - Fabricante;
- Monitor:
  - Tipo; Identificação;
- Placa de rede:
  - Nome; Endereço MAC;
- Processador:
  - Identificação; Velocidade; Família;
- Placa mãe:
  - Fabricante; Modelo; Número de série;

Já para os itens de *software*, em cada sistema operacional foi realizada uma maneira diferente de coleta. Para o SO *Windows*, apenas os *softwares* que foram instalados via *Windows Installer* foram coletados. Já para o SO *Linux*, uma lista de pacotes de todos os *softwares* foi coletada.

### 4.3 CONFIGURAÇÃO DO AMBIENTE DE TESTE

O seguinte cenário de teste foi constituído, conforme figura abaixo.



**Figura 4.1 – Ambiente de Experimento**

As configurações básicas de hardware de cada computador são as seguintes:

- Computador Servidor:
  - Desktop fabricante HP dc 5750 Microtower; Processador AMD Athlon 64 X2 Dual Core Processor 4000+ 2.09GHz; 2GB de memória RAM; 150GB de disco rígido.
- Computador Usuário Linux:
  - Desktop fabricante Dell Optiflex GX620; Processador Pentium 4 3.20GHz; 2GB de memória RAM; 80GB de disco rígido.
- Computador Usuário Windows:
  - Notebook fabricante Dell Vostro 1000; Processador AMD Turion 64x Mobile Technology TL-56 792MHz; 2GB de memória RAM; 80GB de disco rígido.

Após a configuração da rede, cada computador foi configurado conforme sua necessidade. O Servidor, atendendo os requisitos de *hardware* e *software* do sistema explicados no item 3.2.1, foi configurado com o banco de dados para que os computadores usuários pudessem enviar as informações coletadas em cada máquina.

Os outros dois computadores também atendiam aos requisitos do item 3.2.1 e foram configurados, exigindo a instalação do *.NET Framework 2.0* na máquina Windows e do *software lshw* na Linux. Além de instalados cada *script* diferente para o respectivo SO.

No computador Windows, o *script* pode ser instalado diretamente através do arquivo executável gerado pelo *Visual Studio 2005*. Após sua instalação, configurou-o para ser inicializado juntamente com o SO.

Já na máquina Linux, o *script* foi gerado diretamente na linha de comando. Também foi configurado para ser inicializado assim que o usuário fizer o *login*.

#### **4.4 TESTES REALIZADOS**

Configurado todo o ambiente de rede e instalados os respectivos *scripts* de coletas em cada SO, deu-se inicio a fase de testes. Algumas ações foram tomadas para sua correta execução.

Primeiramente o computador teve seu banco de dados iniciado. Por meio do comando *“truncate nome\_da\_tabela”*, todas as tabelas foram reiniciadas, não mostrando nenhum dado inserido anteriormente.

Uma rotina de ligar e desligar cada computador, efetuando o *login* e *logout* respectivamente, foi efetuada diversas vezes para a execução do script e a inserção de dados nas tabelas do banco de dados, verificando assim a correta comunicação de cada computador usuário com o computador servidor.

Simulando várias sessões de usuários de durações variadas, instalação e remoção de diferentes softwares, pode-se verificar a correta inserção de todos os dados coletados no banco de dados através de *query*, com o comando *“select \* from nome\_da\_tabela”*. Comando este que retorna todas as entradas e valores inseridos na tabela. Com essa etapa foi confirmado o funcionamento do aplicativo.

A inserção de dados também pode ser observada através do uso da GUI desenvolvida.

#### **4.5 RESULTADOS OBTIDOS E PROBLEMAS ENCONTRADOS**

Durante todo o período de testes de aproximadamente uma hora foram computados um total de 15 registros na tabela *tb\_hardware* dos dois computadores ligados na rede desenvolvida.

O computador Usuário Windows teve 9 registros na tabela *tb\_hardware* e 423 registros na *tb\_software*. O Usuário Linux teve 6 registros na tabela *tb\_hardware* e 6 registros na tabela *tb\_sw\_linux*. A diferença de registros para as tabelas de *software* são porque na *tb\_software*, cada *software* coletado corresponde a um único registro da tabela. Já na *tb\_sw\_linux*, é armazenado um arquivo *txt* contendo todos os pacotes de *softwares* instalados.

As durações do tempo de sessão variaram de cinco a quinze minutos. Essa variação se justifica pelos testes de inclusão e remoção de softwares instalados, além de gerar mais de um registro para cada máquina da rede.

Alguns problemas foram detectados no decorrer dos testes:

- O *script* rodado na máquina *Windows* retorna apenas os *softwares* instalados via *Windows Installer*.
- Foi necessária a inclusão de um botão para salvar os arquivos XML e TXT gerados pelo *script Linux* para dados de *hardware* e *software* respectivamente. Não precisou ser feito para o *script Windows*, pois este já salva os dados coletados diretamente no banco de dados.

## Capítulo 5 - Conclusão

---

Para a elaboração do aplicativo, tendo por base o trabalho do aluno David Mascarenhas Watkins, foi necessário um estudo e análise dos resultados anteriormente obtidos. Após essa etapa, verificou-se necessidade de uma melhoria do aplicativo, já que um inventário apenas dos *hardwares* não seria bem aproveitado pelo AR. Era necessário a inclusão dos softwares para que o inventário ficasse mais completo. Após essa etapa, deu-se início ao novo projeto.

Ao final de todo o desenvolvimento e testes realizados, pode-se constatar que o SIHS teve seus objetivos perfeitamente atendidos. Fornece um inventário de *hardware* e *software* que é atualizado a cada início de sessão do computador usuário e, finalmente, disponibiliza para o AR uma interface gráfica de usuário onde os dados coletados são disponibilizados com rapidez e facilidade.

O desenvolvimento do aplicativo foi dividido em três etapas. A primeira delas foi a criação do *script* para coletar os dados de *hardware* e *software* de cada máquina existente no parque computacional. A próxima etapa remete à criação das tabelas do banco de dados, responsáveis por armazenar todos os dados coletados, que para serem visualizados, foi necessária a criação de uma GUI, sendo essa o terceiro e último passo do desenvolvimento.

Concluindo a etapa de desenvolvimento, deu-se início à fase de teste do projeto. Foi possível nessa etapa a realização de um levantamento dos resultados adquiridos, permitindo uma análise da sua funcionalidade.

Com a fase de testes efetuada com sucesso, pode-se observar que todas as tecnologias utilizadas foram apropriadas à criação do aplicativo.

Executados remotamente e sempre que é ligada cada máquina, o SIHS facilitou o trabalho do AR. De qualquer lugar conectado à rede em análise, o profissional de redes pode ter acesso aos dados computados, evitando o deslocamento até o local e ganhando em rapidez e agilidade.

## **5.1 SUGESTÕES PARA TRABALHOS FUTUROS**

Durante o desenvolvimento, surgiram algumas possibilidades de melhoria e expansão, mas que não puderam ser aplicadas devido ao curto período de tempo de desenvolvimento. Algumas dessas sugestões seguem abaixo:

- O desenvolvimento de uma GUI que possa tratar os dados obtidos e gerar gráficos, utilizando conhecimentos de estatística, além de gerar relatórios para impressão.
- O aplicativo desenvolvido é visível para o usuário. Portanto, pode-se modificá-lo para que seja executado como um serviço do sistema operacional, de forma oculta.

- Foram desenvolvidos dois scripts, um para Linux e outro para Windows. Juntar ambos em um único script torna-o mais fácil para futura manutenção e/ou desenvolvimento.
- A inclusão do *IP* do usuário na tabela *tb\_hardware* facilitaria a busca. Antes buscado pelo *endereço MAC*, o aplicativo buscaria pelo *IP* de cada máquina, já que o dado é mais fácil de ser guardado e digitado para a possível busca.
- O *script* rodado na máquina *Windows* retorna apenas os *softwares* instalados via *Windows Installer*. Fazer um *script* que colete todos os outros *softwares* tornaria mais rico o inventário.
- Já o *script* rodado na máquina *Linux* retorna o pacote de *softwares*. A utilização de um filtro, excluindo os pacotes da instalação do *Linux*, deixaria os dados mais *clean* e de melhor visualização e entendimento para o aplicativo.

## Referências Bibliográficas

---

DATAPREV. Dataprev – Empresa de Tecnologia e Informações da Previdência Social. **CACIC - configurador automático e coleta de informações computacionais**. Disponível em : <<http://www.dataprev.gov.br/produtos/cacic.htm>>. Acesso em 05 dez 2008.

ECMA. European Computer Manufacturer Association. **C# Language Specification**. Jun. 2006. Disponível em: <<http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-334.pdf>>. Acessado em 08 nov. 2008.

GEORGE, Rajesh; DELANO, Lance. **Wrox's SQL Server 2005 Express Edition Starter Kit**. Indianápolis, Indiana, EUA; Wiley Publishing, Inc., 2006.

GOVERNO FEDERAL. Software Livre no Governo do Brasil. **Planejamento Estratégico para Implementação de Software Livre**. Disponível em: < <http://www.softwarelivre.gov.br/objetivos-do-portal> >. Acessado em 27 ago. 2008.

IBM. International Business Machines Corp. **Tivoli Configuration Manager**. Disponível em: <[http://www142.ibm.com/software/dre/ecatalog/detail.wss?locale=pt\\_BR&synkey=M106021O58324P96](http://www142.ibm.com/software/dre/ecatalog/detail.wss?locale=pt_BR&synkey=M106021O58324P96)>. Acesso em 05 dez 2008.

MARSHALL, Donis. **Programming Microsoft Visual C# 2005 – The Language**. Washington, EUA; Microsoft Press, 2006.

MICROSOFT. Microsoft Corporation. **Visual Studio Hardware Requirements**. 2008a. Disponível em: < [http://msdn.microsoft.com/en-us/library/4c26cc39\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/4c26cc39(VS.80).aspx)>. Acesso em: 04 out. 2008.

MATRIX42. Matrix42 AG. **Empirum PRO**. Disponível em: <<http://www.matrix42.com/en/products/empirum-pro>> Acesso em 05 dez 2008.

MILANI, André. **MySQL : Guia do programador**. São Paulo, Brasil; Novatec Editora, 2006.

MSDN. Microsoft Developer Network. **Visual C#**. 2008a. Disponível em: <<http://www.microsoft.com/brasil/msdn/csharp/default.aspx>>. Acessado em 08 nov. 2008.

MYSQL. MySQL AB. **O banco de dados open-source mais popular do mundo**. Disponível em: <<http://www.mysqlbrasil.com.br/>>. Acessado em 03 out. 2008.

NIEDERAUER, Juliano. **Desenvolvendo Wbsites com PHP**. São Paulo, Brasil; Novatec Editora, 2008.

PHP. PHP Hypertext Preprocessor. **What is PHP?** Disponível em: <<http://www.php.net>>. Acessado em 03 out. 2008.

RANDAL, Michael K. **Dominando Unix Shell Scripting** tradução Daniel Vieira – Rio de Janeiro: Elsevier, 2003.

SUDRÉ, Gilberto. **Inventário das configurações dos computadores**. Disponível em: <[http://imasters.uol.com.br/artigo/8442/redes/inventario\\_das\\_configuracoes\\_dos\\_computadores](http://imasters.uol.com.br/artigo/8442/redes/inventario_das_configuracoes_dos_computadores)>. Acesso em: 05 dez 2008.

VIVA O LINUX. **O que é shell script? [Artigo]**. Mar. 2007. Disponível em : < <http://www.vivao-linux.com.br/artigo/O-que-e-Shell-Script/>>. Acessado em 23 nov. 2008.

WIKIPÉDIA. Wikipédia, a enciclopédia livre. **Administrador de redes**. 2008<sup>a</sup>. Disponível em: <[http://pt.wikipedia.org/wiki/Administrador\\_de\\_redes](http://pt.wikipedia.org/wiki/Administrador_de_redes)>. Acesso em: 05 dez 2008.

WIKIPÉDIA. Wikipédia, a enciclopédia livre. **Linguagem de script**. 2008<sup>b</sup>. Disponível em: <[http://pt.wikipedia.org/wiki/Linguagem\\_de\\_script](http://pt.wikipedia.org/wiki/Linguagem_de_script)>. Acesso em: 05 dez 2008.

## APÊNDICE A – CÓDIGO DE PROGRAMAÇÃO

---

### Script Windows de Coleta de dados

```

using System;
using System.IO;
using System.Management;
using System.Threading;
using MySql.Data.MySqlClient;
using MySql.Data.Types;
namespace Projeto_Final{
    class Projeto_Final_Henrique{
        //variaveis globais devido a coleta de dados ser dividida em métodos
diferentes
        public int ram_MB = (1024);
        public int ram_Pentes = 0;
        public Int64 ram_Capacidade = 0;
        public long ram_Total = 0;
        public String hd_Modelo = "";
        public int hd_MB = (1024 * 1024 * 1024);
        public int hd_Quantidade = 0;
        public Int64 hd_Capacidade = 0;
        public long hd_Total = 0;
        public String mouse_Fabricante = "";
        public String teclado_Fabricante = "";
        public String monitor_Tipo = "";
        public String monitor_Identificacao = "";
        public String rede_MAC = "";
        public String rede_Nome = "";
        public String cpu_Fabricante = "";
        public String cpu_ID = "";
        public Int64 cpu_Velocidade = 0;
        public Int64 cpu_Familia = 0;
        public String pm_Fabricante = "";
        public String pm_Modelo = "";
        public String pm_Serial = "";
        public String idatual = "";
        public String sw_Nome = "";
        public String sw_Data_Instalacao = "";
        public String sw_Versao = "";
        public DateTime tempoliga = DateTime.Now;
        //serve pra minimizar o aplicativo assim que ele for iniciado
        private const int SW_MINIMIZE = 6;
        [System.Runtime.InteropServices.DllImport("user32.dll")]
        private static extern bool ShowWindow(IntPtr hWnd, int nCmdShow);
        [STAThread]
        static void Main(string[] args){
            IntPtr winHandle =
System.Diagnostics.Process.GetCurrentProcess().MainWindowHandle;
            ShowWindow(winHandle, SW_MINIMIZE);
            Projeto_Final_Henrique projeto = new Projeto_Final_Henrique();
            projeto.memoria_Ram();
            projeto.hd();

```

```

projeto.mouse();
projeto.teclado();
projeto.monitor();
projeto.rede();
projeto.cpu();
projeto.placa_Mae();
projeto.escreveBanco("Database=ceub;Server=localhost;User
Id=root;Password=");
projeto.softwares_instalados("Database=ceub;Server=localhost;User
Id=root;Password=");
Thread t = new Thread(new ThreadStart(projeto.escreveDesliga));
t.Start();
}
//coleta de informações sobre memória RAM
private void memoria_Ram(){
ManagementObjectSearcher busca = new
ManagementObjectSearcher("Select * from Win32_PhysicalMemory");
foreach (ManagementObject result in busca.Get()){
ram_Capacidade +=
Convert.ToInt64(result.GetProperty("Capacity"));
ram_Pentes++;
ram_Total = ram_Capacidade / ram_MB; //retorna o valor em kb.
}
Console.WriteLine("RAM Total: {0}", ram_Total);
Console.WriteLine("Qtd de Pentes: {0}", ram_Pentes);
Console.WriteLine("");
}
//coleta de informações sobre discos rígidos
private void hd(){
//Dados HD - nao pega USB
ManagementObjectSearcher busca = new
ManagementObjectSearcher("SELECT * FROM Win32_DiskDrive WHERE
InterfaceType!='USB'");
//comando InterfaceType!='USB' faz com que não seja coletadas dados
sobre drives USB
foreach (ManagementObject result in busca.Get()){
hd_Capacidade +=
Convert.ToInt64(result.GetProperty("Size"));
hd_Quantidade++;
hd_Total = hd_Capacidade / hd_MB; //retorna o valor em gb.
hd_Modelo =
Convert.ToString(result.GetProperty("Model"));
Console.WriteLine("Nome do Disco Rigido: {0}", hd_Modelo);
}
Console.WriteLine("Capacidade de Armazenamento de Disco Rigido:
{0}", hd_Total);
Console.WriteLine("Qtd de HDs: {0}", hd_Quantidade);
Console.WriteLine("");
}
//coleta de informações sobre Mouse
private void mouse(){
ManagementObjectSearcher busca = new
ManagementObjectSearcher("SELECT * FROM Win32_PointingDevice");
foreach (ManagementObject result in busca.Get()){
mouse_Fabricante =
Convert.ToString(result.GetProperty("Manufacturer"));
}
}

```

```

        Console.WriteLine("Fabricante Mouse: {0}", mouse_Fabricante);
        Console.WriteLine("");
    }
    //coleta de informações sobre teclado
    private void teclado(){
        ManagementObjectSearcher busca = new
ManagementObjectSearcher("SELECT * FROM Win32_Keyboard");
        foreach (ManagementObject result in busca.Get()){
            teclado_Fabricante =
Convert.ToString(result.GetPropertyValue("Name"));
        }
        Console.WriteLine("Fabricante de Teclado: {0}",
teclado_Fabricante);
        Console.WriteLine("");
    }
    //coleta de informações sobre monitor
    private void monitor(){
        ManagementObjectSearcher busca = new
ManagementObjectSearcher("SELECT * FROM Win32_DesktopMonitor");
        foreach (ManagementObject result in busca.Get()){
            monitor_Tipo =
Convert.ToString(result.GetPropertyValue("MonitorManufacturer"));
            monitor_Identificacao =
Convert.ToString(result.GetPropertyValue("MonitorType"));
        }
        Console.WriteLine("Tipo de Monitor: {0}", monitor_Tipo);
        Console.WriteLine("Identificacao: {0}", monitor_Identificacao);
        Console.WriteLine("");
    }
    //coleta de informações sobre adaptador de rede
    private void rede(){
        ManagementObjectSearcher busca = new
ManagementObjectSearcher("SELECT * FROM Win32_NetworkAdapter WHERE
NetConnectionStatus=2");
        //NetConnectionStatus=2 faz com que informações só sejam
coletadas de adaptadores de rede com conexão ativa
        foreach (ManagementObject result in busca.Get()){
            rede_MAC =
Convert.ToString(result.GetPropertyValue("MACAddress"));
            rede_Nome =
Convert.ToString(result.GetPropertyValue("Name"));
        }
        Console.WriteLine("Endereco MAC: {0}", rede_MAC);
        Console.WriteLine("Nome: {0}", rede_Nome);
        Console.WriteLine("");
    }
    //coleta de informações sobre processador
    private void cpu(){
        ManagementObjectSearcher busca = new
ManagementObjectSearcher("Select * from Win32_Processor");
        foreach (ManagementObject result in busca.Get()){
            cpu_Velocidade =
Convert.ToInt64(result.GetPropertyValue("CurrentClockSpeed"));
            cpu_Familia =
Convert.ToInt64(result.GetPropertyValue("Family"));
            cpu_ID =
Convert.ToString(result.GetPropertyValue("DeviceId"));

```



```

        catch (Exception e){
            Console.WriteLine(e.ToString());
        }
        try{
            MySqlDataReader leitor = null;
            MySqlCommand comando = new MySqlCommand("select id from
tb_hardware order by id", myConnection);
            leitor = comando.ExecuteReader();
            while (leitor.Read()){
                idatual = leitor["id"].ToString();
            }
        }
        catch (Exception e){
            Console.WriteLine(e.ToString());
        }
        myCommand.Connection.Close();
    }
    //coleta de informações sobre softwares (nome, data de instalação e
versão)
    private void softwares_instalados(string myConnectionString){
        if (myConnectionString == ""){
            myConnectionString = "Database=ceub;Server=localhost;User
Id=root;Password=";
        }
        MySqlConnection myConnection = new
MySqlConnection(myConnectionString);
        ManagementObjectSearcher busca = new
ManagementObjectSearcher("SELECT * FROM Win32_Product");
        foreach (ManagementObject result in busca.Get()){
            sw_Nome = Convert.ToString(result.GetProperty("Name"));
            sw_Data_Instalacao =
Convert.ToString(result.GetProperty("InstallDate"));
            sw_Versao =
Convert.ToString(result.GetProperty("Version"));
            string myInsertQuery = "INSERT INTO tb_software values ('', "
+ idatual + ", ' + sw_Nome + ', ' + sw_Data_Instalacao + ', ' + sw_Versao +
'')";

            MySqlCommand myCommand = new MySqlCommand(myInsertQuery);
            myCommand.Connection = myConnection;
            try{
                myConnection.Open();
            }
            catch (Exception e){
                Console.WriteLine(e.ToString());
            }
            try{
                myCommand.ExecuteNonQuery();
            }
            catch (Exception e){
                Console.WriteLine(e.ToString());
            }
            Console.WriteLine("Nome do Software: {0}", sw_Nome);
            Console.WriteLine("Data de instalacao: {0}",
sw_Data_Instalacao);
            Console.WriteLine("Versao: {0}", sw_Versao);
            Console.WriteLine("");

```



```

quantidade_disco=`fdisk -l | grep Disk | wc -l` #faz a captura de quantos discos estao instalados
atraves do comando fdisk
capacidade_disco=`fdisk -l | grep Disk | cut -f3,4 -d ' '` # faz a captura da capacidade do disco
instalado atraves do comando fdisk
versao_disco=`cat /proc/scsi/scsi | grep Vendor | cut -f7,8,9,10,11 -d ' '` # faz a captura do
fabricante do disco atraves do arquivo scsi localizado no diretorio /pro/scsi/
echo "<hd>" >> /tmp/inventario_teste.xml
echo "<itens>" >> /tmp/inventario_teste.xml
echo "<hd_modelo>" >> /tmp/inventario_teste.xml
echo "$versao_disco" >> /tmp/inventario_teste.xml #saida para a tela
echo "</hd_modelo>" >> /tmp/inventario_teste.xml
echo "<hd_qtde>" >> /tmp/inventario_teste.xml
echo "$quantidade_disco" >> /tmp/inventario_teste.xml #saida para a tela
echo "</hd_qtde>" >> /tmp/inventario_teste.xml
echo "<hd_capacidade>" >> /tmp/inventario_teste.xml
echo "$capacidade_disco" >> /tmp/inventario_teste.xml #saida para a tela
echo "</hd_capacidade>" >> /tmp/inventario_teste.xml
echo "</itens>" >> /tmp/inventario_teste.xml
echo "</hd>" >> /tmp/inventario_teste.xml
# MOUSE
fabricante_mouse=`cat /proc/bus/input/devices | tail -8 | grep N: | cut -f2 -d '='` # detecta o
fabricante do mouse atraves do arquivo devices
echo "<mouse>" >> /tmp/inventario_teste.xml
echo "<itens>" >> /tmp/inventario_teste.xml
echo "<mouse_fabricante>" >> /tmp/inventario_teste.xml
echo "$fabricante_mouse" >> /tmp/inventario_teste.xml #saida para a tela
echo "</mouse_fabricante>" >> /tmp/inventario_teste.xml
echo "</itens>" >> /tmp/inventario_teste.xml
echo "</mouse>" >> /tmp/inventario_teste.xml
# TECLADO
fabricante_teclado=`cat /proc/bus/input/devices | head -3 | grep N | cut -f2 -d '='` # detecta o
fabricante do teclado atraves do arquivo devices
echo "<teclado>" >> /tmp/inventario_teste.xml
echo "<itens>" >> /tmp/inventario_teste.xml
echo "<teclado_fabricante>" >> /tmp/inventario_teste.xml
echo "$fabricante_teclado" >> /tmp/inventario_teste.xml #saida para a tela
echo "</teclado_fabricante>" >> /tmp/inventario_teste.xml
echo "</itens>" >> /tmp/inventario_teste.xml
echo "</teclado>" >> /tmp/inventario_teste.xml
# MONITOR
tipo_monitor=`cat /usr/lib/vmware-tools/configurator/XFree86-3/XF86Config | grep Identifier |
sed -e '1d' | cut -f6 -d ' '` # detecta o tipo de monitor atraves do arquivo XF86Config
identificacao_monitor=`cat /usr/lib/vmware-tools/configurator/XFree86-3/XF86Config | grep
Model | cut -f6,7 -d ' '` # Faz a identificacao do monitor atraves do arquivo XF86Config
echo "<monitor>" >> /tmp/inventario_teste.xml
echo "<itens>" >> /tmp/inventario_teste.xml
echo "<monitor_tipo>" >> /tmp/inventario_teste.xml
echo "$tipo_monitor" >> /tmp/inventario_teste.xml #saida para a tela
echo "</monitor_tipo>" >> /tmp/inventario_teste.xml
echo "<monitor_identificacao>" >> /tmp/inventario_teste.xml
echo "$identificacao_monitor" >> /tmp/inventario_teste.xml #saida para a tela

```

```

echo "</monitor_identificacao>" >> /tmp/inventario_teste.xml
echo "</itens>" >> /tmp/inventario_teste.xml
echo "</monitor>" >> /tmp/inventario_teste.xml
# REDE
nome_placarede=`ifconfig | cut -f1 -d ' ' | sed '/^$/d' | sed '/^lo$/d` # Detecta o nome da interface
atraves do comando ifconfig
mac_placarede=`ifconfig | cut -f11 -d ' ' | head -1` # detecta o mac da placa de rede atraves do
comando ifconfig
echo "<rede>" >> /tmp/inventario_teste.xml
echo "<itens>" >> /tmp/inventario_teste.xml
echo "<rede_nome>" >> /tmp/inventario_teste.xml
echo "$nome_placarede" >> /tmp/inventario_teste.xml #saida para a tela
echo "</rede_nome>" >> /tmp/inventario_teste.xml
echo "<rede_mac>" >> /tmp/inventario_teste.xml
echo "$mac_placarede" >> /tmp/inventario_teste.xml #saida para a tela
echo "</rede_mac>" >> /tmp/inventario_teste.xml
echo "</itens>" >> /tmp/inventario_teste.xml
echo "</rede>" >> /tmp/inventario_teste.xml
# PROCESSADOR
fabricante_processador=`cat /proc/cpuinfo | grep vendor_id | cut -f2 -d ' '` # detecta o fabricante
do processador atraves do arquivo cpuinfo
familia_processador=`cat /proc/cpuinfo | grep "cpu family" | cut -f3 -d ' '` # detecta a familia do
processador atraves do arquivo cpuinfo
identificacao_processador=`cat /proc/cpuinfo | grep processor | cut -f1,2 -d ' '` # detecta a
identificacao do processdor do arquivo cpuinfo
velocidade_processador=`cat /proc/cpuinfo | grep MHz | cut -f3 -d ' '` # detecta a identificacao
da velocidade do processador atraves do arquivo cpuinfo
echo "<processador>" >> /tmp/inventario_teste.xml
echo "<itens>" >> /tmp/inventario_teste.xml
echo "<processador_fabricante>" >> /tmp/inventario_teste.xml
echo "$fabricante_processador" >> /tmp/inventario_teste.xml #saida para a tela
echo "</processador_fabricante>" >> /tmp/inventario_teste.xml
echo "<processador_familia>" >> /tmp/inventario_teste.xml
echo "$familia_processador" >> /tmp/inventario_teste.xml #saida para a tela
echo "</processador_familia>" >> /tmp/inventario_teste.xml
echo "<processador_identificacao>" >> /tmp/inventario_teste.xml
echo "$identificacao_processador" >> /tmp/inventario_teste.xml #saida para a tela
echo "</processador_identificacao>" >> /tmp/inventario_teste.xml
echo "<processador_velocidade>" >> /tmp/inventario_teste.xml
echo "$velocidade_processador" >> /tmp/inventario_teste.xml #saida para a tela
echo "</processador_velocidade>" >> /tmp/inventario_teste.xml
echo "</itens>" >> /tmp/inventario_teste.xml
echo "</processador>" >> /tmp/inventario_teste.xml
# PLACA MAE
fabricante_placamae=`cat /tmp/hardware | grep "vendor: Intel Corporation" | cut -f9,10 -d ' '` #
detecta o fabricante da placa mae atraves do arquivo gerado pelo programa lshw, localizado no
diretorio /tmp - o nome do arquivo eh hardware
modelo_placamae=`cat /tmp/hardware | grep "product: 440BX" | awk "NR < 2" | cut -
f9,10,11,12 -d ' '` # mesma situacao do arquivo acima
numeroserie_placamae=`cat /tmp/hardware | grep "serial: None" | cut -f9 -d ' '` # mesma
situacao do arquivo acima

```

```

echo "<placa_mae>" >> /tmp/inventario_teste.xml
echo "<itens>" >> /tmp/inventario_teste.xml
echo "<pm_fabricante>" >> /tmp/inventario_teste.xml
echo "$fabricante_placamae" >> /tmp/inventario_teste.xml #saida para a tela
echo "</pm_fabricante>" >> /tmp/inventario_teste.xml
echo "<pm_modelo>" >> /tmp/inventario_teste.xml
echo "$modelo_placamae" >> /tmp/inventario_teste.xml #saida para a tela
echo "</pm_modelo>" >> /tmp/inventario_teste.xml
echo "<pm_serie>" >> /tmp/inventario_teste.xml
echo "$numerodeserie_placamae" >> /tmp/inventario_teste.xml #saida para a tela
echo "</pm_serie>" >> /tmp/inventario_teste.xml
echo "</itens>" >> /tmp/inventario_teste.xml
echo "</placa_mae>" >> /tmp/inventario_teste.xml
echo "</hardware>" >> /tmp/inventario_teste.xml
# GERA A LISTA DOS PACOTES INSTALADOS
`rpm -qa > /tmp/lista_pacotes.txt` # Lista todos os pacotes instalados atraves do comando rpm
gravando no arquivo lista_pacotes.txt

```

```

cd /tmp
host='10.0.1.1'
user='gomera'
passwd='teste'
arquivo='inventario_teste.xml'
file_txt='lista_softwares.txt'

```

```

ftp -u $host <<END_SCRIPT
quote user $user
quote pass $passwd
prompt off
put *.xml $file
put *.txt $file_txt
bye
END_SCRIPT

```

```

rm -rf /tmp/inventario_teste.xml
exit 0

```

## Interface gráfica de usuário

### index.php – Responsável pelo *login* do usuário

```

<html>
<head>
    <title>Sistema de Invent&acut;rio de Hardware e Software</title>
</head>
<link rel="stylesheet" type="text/css" href="css/estilo.css">

```

```

<body>
<h1><center>Login</center></h1><p>
<center>
<form method="post" action="login.php" name="form1">
<table border="0" cellspacing="3" cellpadding="3" bgcolor="#FFFFCC">
  <tr>
    <td align="right">Nome: </td>
    <td><input type="text" name="login_usuario" size="20" maxlength="20"></td>
  </tr>
  <tr>
    <td align="right">Senha: </td>
    <td><input type="password" name="senha_usuario" size="20" maxlength="20"></td>
  </tr>
  <tr>
    <td colspan="2" align="center">
      <input type="submit" value="ENTRAR" name="entrar">
    </td>
  </tr>
</table>
</form>
</center>
</body>
</html>

```

### login.php – Validação do login

```

<?
$login_usuario= $_POST["login_usuario"];
$senha_usuario      = $_POST["senha_usuario"];
$senhamd5           = md5($senha_usuario);
include"conexao.php";
$sql = "SELECT * FROM tb_usuarios WHERE login_usuario='$login_usuario'";
$resultado = mysql_query($sql);
$linhas = mysql_num_rows($resultado);
if($linhas==0){

```

```

    echo"
        <html><body>
            <p align=\"center\">Usuário não encontrado!</p>
            <p align=\"center\"><a href=\"index.php\">Voltar</a></p>
        </body></html>
    ";
}else{
    if($senhamd5 != mysql_result($resultado, 0, "senha_usuario")){
        echo"
            <html><body>
                <p align=\"center\">A senha está incorreta!</p>
                <p align=\"center\"><a href=\"index.php\">Voltar</a></p>
            </body></html>
        ";
    }else{
        setcookie("login_cookie", $login_usuario);
        setcookie("senha_cookie", $senhamd5);
        header("Location: inicio.php");
    }
}
?>

```

### **conexao.php - Conexão com o banco de dados**

```

<?
$host="localhost"; //caminho do servidor
$user="root"; //nome do usuário do servidor
$senha=""; //senha do servidor
$dbname="ceub"; //nome do banco de dados
mysql_connect($host, $user, $senha) or die("Erro - Não foi possível conectar ao banco de
dados");
mysql_select_db($dbname) or die("Erro - Banco de dados não localizado");
?>

```

### **valida\_cookie.php – Verifica se o usuário efetuou o *login***

&lt;?

```

if(isset($_COOKIE["login_cookie"])){
    $login_cookie = $_COOKIE["login_cookie"];
}
if(isset($_COOKIE["senha_cookie"])){
    $senha_cookie = $_COOKIE["senha_cookie"];
}
if(!(empty($login_cookie) or empty($senha_cookie))){
    include "conexao.php";
    $sql = "SELECT * FROM tb_usuarios WHERE login_usuario='$login_cookie'";
    $resultado = mysql_query($sql);
    if(mysql_num_rows($resultado)==1){
        if($senha_cookie != mysql_result($resultado, 0, "senha_usuario")){
            setcookie("login_cookie");
            setcookie("senha_cookie");
            echo "1- Você não efetuou o LOGIN!";
            exit;
        }
    }else{
        setcookie("login_cookie");
        setcookie("senha_cookie");
        echo "2- Você não efetuou o LOGIN!";
        exit;
    }
}
}

```

?&gt;

**inicio.php – Página principal**

&lt;?

include "valida\_cookie.php";

?&gt;

&lt;html&gt;

```

<head>
    <title>Sistema de Inventário de Hardware e Software</title>
</head>
<link rel="stylesheet" type="text/css" href="css/estilo.css">
<frameset rows="80,*" cols="*" border="5" bordercolor="#006600">
    <frame src="superior.php" name="superior" noresize/>
    <frameset rows="*" cols="200,*" border="5" bordercolor="#006600">
        <frameset rows="*,100" cols="*" border="5" bordercolor="#006600">
            <frame src="menu.php" name="menu" noresize/>
            <frame src="rodape.php" name="rodape" noresize/>
        </frameset>
    <frame src="principal.php" name="principal"/>
</frameset>
<noframes>
<body>
Seu browser não suporta frames.
</body>
</noframes></html>

```

### superior.php – Página do título

```

<?
include "valida_cookie.php";
?>
<html>
<head>
    <title>Superior</title>
</head>
<link rel="stylesheet" type="text/css" href="css/estilo.css">
<body bgcolor="#FFFFCC">
<font color="#006600">
    </font><center>
        <h1><font color="#006600"><b>Sistema de Inventário de Hardware e
Software</b></font></h1>
    </center>

```

```
</body>
</html>
```

### menu.php – Menu lateral

```
<?
include "valida_cookie.php";
?>
<script language="javascript">
    function confirma_logout(){
        if(confirm("Deseja realmente sair do sistema?")){
            location.href("logout.php");
            parent.window.close();
            window.open("index.php");
        }
    }
</script>
<html>
<head>
    <title>Menu</title>
</head>
<link rel="stylesheet" type="text/css" href="css/estilo.css">
<body bgcolor="#FFFFCC" leftmargin="0">

<table width="100%" border="0" cellspacing="10" cellpadding="10">
    <tr>
        <td align="center"><font color="#006600"><h3>Menu</h3></font></td>
    </tr>
    <tr>
        <td align="left"><a href="principal.php" target="principal">P&aaacute;gina Principal</a></td>
    </tr>
    <tr>
        <td align="left"><a href="principal/software.php" target="principal">Software</a></td>
    </tr>
    <tr>
```

```

    <td align="left"><a href="principal/hardware.php" target="principal">Hardware</a></td>
</tr>
<tr>
    <td align="left"><a href="principal/relatorio.php"
target="principal">Relat&oacute;rios</a></td>
</tr>
<tr>
    <td align="left"><a href="principal/usuario.php" target="principal">Usu&aacute;rios</a></td>
</tr>
<tr>
    <td align="left"><a href="#" onClick="confirma_logout()">Logout</a></td>
</tr>
</table>
</body>
</html>

```

### **usuario.php – Lista opções de inserção, edição, exclusão e visualização de usuários.**

```

<?
include "../valida_cookie.php";
?>
<html>
<head>
    <title>Gerenciar Usu&aacute;rios</title>
</head>
<link rel="stylesheet" type="text/css" href="../css/estilo.css">
<body>
<center>
<h1>Gerenciar Usu&aacute;rios</h1><br><br><br><br>
<table border="0">
    <tr>
        <td>&nbsp;</td>
        <td width="200"><a href="cadastro_usuario.php" target="principal">Incluir
Usu&aacute;rio</a></td>
    </tr>

```

```

<tr>
  <td>&nbsp;</td>
  <td width="200"><a href="alterar_usuario.php" target="principal">Editar/Excluir
Usu&aacute;rio</a></td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td width="200"><a href="listar_usuario.php" target="principal">Listar Todos
Usu&aacute;rios</a></td>
</tr>
</table>
</center>
</body>
</html>

```

#### **alterar\_usuario.php – Altera os dados do usuário.**

```

<html>
<head>
  <title>Usuário</title>
</head>
<link rel="stylesheet" type="text/css" href="../css/estilo.css">
<body>
<?
include "../valida_cookie.php";
$sql= "SELECT id_usuario, login_usuario, senha_usuario, nome_usuario, email_usuario
      FROM tb_usuarios";
if($_GET['ordenacao'] != ""){
  $sql.=" ORDER BY ".$_GET['ordenacao'];
}
$resultado= mysql_query($sql);
if (mysql_num_rows($resultado)==0){
  echo "<strong><center>Nenhum usuário cadastrado no momento.</strong></center>";
  return false;
}else{

```

```

?>
<script language="javascript">
    function confirma_exclusao(id){
        if(confirm("Deseja excluir o usuário?")){
            location.href("gravar_usuario.php?id="+id+"&excluir=1");
        }
    }
</script>
<h1><center>Lista de Usuários:</center></h1><p>
<table border="1" align="center" cellpadding="3" cellspacing="1" bgcolor="#000000">
  <tr style="color:#FFFFFF; font:bold ; text-align:center">
    <td ><strong><a href="alterar_usuario.php?ordenacao=id_usuario" style="text-decoration:none; color:#FFFFFF;">ID Usuário</a></strong></td>
    <td><strong><a href="alterar_usuario.php?ordenacao=nome_usuario" style="text-decoration:none; color:#FFFFFF;">Nome do Usuário</a></strong></td>
    <td><strong><a href="alterar_usuario.php?ordenacao=email_usuario" style="text-decoration:none; color:#FFFFFF;">Login</strong></td>
    <td><strong>Email</strong></td>
    <td><strong>Editar</strong></td>
    <td><strong>Excluir</strong></td>
  </tr>
<?
    $cor="CCCCCC";
    while ($lista_usuario = mysql_fetch_array($resultado)){
        if ($cor == "CCCCCC"){
            $cor = "FFFFFF";
        }else{
            $cor = "CCCCCC";
        }
    }
?>
<tr bgcolor="<?=$cor ?>">
  <td align="center"><?=$lista_usuario['id_usuario'] ?></td>
  <td align="center"><?=$lista_usuario['nome_usuario'] ?></td>
  <td align="center"><?=$lista_usuario['login_usuario'] ?></td>

```

```

        <td align="center"><?=$lista_usuario['email_usuario'] ?></td>
        <td align="center"><a
href="cadastro_usuario.php?id=<?=$lista_usuario['id_usuario']?>">Editar</a> </td>
        <td align="center"><a href="#"
onClick="confirma_exclusao(<?=$lista_usuario['id_usuario']?>)">Excluir</a> </td>
    </tr>
    <?
        } //fecha while
} //fecha else
    ?>
<tr>
    <td colspan="7" align="center">
        <input type="button" onClick="javascript:location.href('usuario.php')"
value="VOLTAR" />
    </td>
</tr>
</table>
</body>
</html>

```

### **cadastro\_usuario.php – Cadastro de novos usuários**

```

<?
include "../valida_cookie.php";
$id_usuario      = "";
$login_usuario= "";
$senha_usuario  = "";
$nome_usuario   = "";
$email_usuario   = "";
$destino         ="gravar_usuario.php";
if($_GET["id"]!=""){
    $sql= " SELECT id_usuario, login_usuario, senha_usuario, nome_usuario,
email_usuario
        FROM tb_usuarios
        WHERE id_usuario=".$_GET["id"];

```

```

$resultado=mysql_query($sql);
while($lista_usuario=mysql_fetch_array($resultado)){
    $id_usuario      = $lista_usuario['id_usuario'];
    $login_usuario= $lista_usuario['login_usuario'];
    $senha_usuario   = $lista_usuario['senha_usuario'];
    $nome_usuario    = $lista_usuario['nome_usuario'];
    $email_usuario   = $lista_usuario['email_usuario'];
}
$destino = "gravar_usuario.php?id=".$$_GET["id"];
}
?>
<script language="javascript">
function verifica_campos() {
    if (form1.login_usuario.value == "") {
        alert('É necessário informar um Login.');
```

```
<table width="500" border="0" cellspacing="3" cellpadding="3" align="center">
  <tr>
    <td align="right">Login:</td>
    <td><input type="text" name="login_usuario" maxlength="50" size="30"
value="<?=$login_usuario?>"></td>
  </tr>
  <tr>
    <td align="right">Senha:</td>
    <td><input type="password" name="senha_usuario" maxlength="50" size="30"
value="<?=$senha_usuario?>"></td>
  </tr>
  <tr>
    <td align="right">Nome:</td>
    <td><input type="text" name="nome_usuario" maxlength="50" size="30"
value="<?=$nome_usuario?>"></td>
  </tr>
  <tr>
    <td align="right">Email:</td>
    <td><input type="text" name="email_usuario" maxlength="50" size="30"
value="<?=$email_usuario?>"></td>
  </tr>
  <tr>
    <td colspan="2" align="center">
      <input type="button" value="GRAVAR" onClick="verifica_campos()">
      <input type="reset" value="LIMPAR">
      <input type="button" value="VOLTAR"
onClick="javascript:location.href('usuario.php')">
    </td>
  </tr>
</table>
</form>
</body>
</html>
```

**gravar\_usuario.php – Grava o usuário no banco**

```

<?
include "../valida_cookie.php";
$login_novo          = $_POST['login_usuario'];
$senha_novo          = $_POST['senha_usuario'];
$senha_md5           = md5($senha_novo);
$nome_novo           = $_POST['nome_usuario'];
$email_novo          = $_POST['email_usuario'];
if($_GET['id']!=""){
    if($_GET['excluir']){
        //DELETE
        $sql="DELETE FROM tb_usuarios WHERE
id_usuario=".$_GET['id'];
        $msg="Usuário excluído com sucesso!";
        $location="alterar_usuario.php";
    }else{
        //UPDATE
        $sql="
UPDATE tb_usuarios
SET login_usuario='$login_novo', senha_usuario =
'$senha_md5',
                                nome_usuario='$nome_novo',
email_usuario = '$email_novo'
                                WHERE id_usuario=".$_GET['id'];
        $msg="Usuário alterado com sucesso!";
        $location="alterar_usuario.php";
    }
}else{
//INSERT
        $sql =" INSERT INTO tb_usuarios
                                (login_usuario, senha_usuario,
nome_usuario, email_usuario)
                                VALUES

```

```

                                ('$login_novo', '$senha_md5',
'$nome_novo', '$email_novo');"

                                $msg="Usuário cadastrado com sucesso!";
                                $location="listar_usuario.php";
}
mysql_query($sql) ;
?>
<script language="javascript">
                                alert("<?=$msg?>");
                                location="<?=$location?>";
</script>

```

### **lista\_usuario.php – Lista todos os usuário existentes.**

```

<html>
<head>
                                <title>Usuário</title>
</head>
<link rel="stylesheet" type="text/css" href=" ../css/estilo.css">
<body>
<?
include "../valida_cookie.php";

$sql= "SELECT id_usuario, login_usuario, senha_usuario, nome_usuario, email_usuario
                                FROM tb_usuarios";
if($_GET['ordenacao'] != ""){
                                $sql.=" ORDER BY ".$_GET['ordenacao'];
}
$resultado= mysql_query($sql);
if (mysql_num_rows($resultado)==0){
                                echo "<strong><center>Nenhum usuário disponível no
momento.</strong></center>";
                                return false;
}else{

```

```

?>
<h1><center>Lista de Usuários:</center></h1><p>
<table border="1" align="center" cellpadding="3" cellspacing="1" bgcolor="#000000">
  <tr style="color:#FFFFFF; font:bold ; text-align:center">
    <td ><strong><a href="listar_usuario.php?ordenacao=id_usuario" style="text-
decoration:none; color:#FFFFFF;">ID Usuário</a></strong></td>
    <td><strong><a href="listar_usuario.php?ordenacao=nome_usuario" style="text-
decoration:none; color:#FFFFFF;">Nome do Usuário</a></strong></td>
    <td><strong><a href="listar_usuario.php?ordenacao=email_usuario" style="text-
decoration:none; color:#FFFFFF;">Login</strong></td>
    <td><strong>Email</strong></td>
  </tr>
<?
        $cor="CCCCCC";
        while ($lista_usuario = mysql_fetch_array($resultado)){
          if ($cor == "CCCCCC"){
            $cor = "FFFFFF";
          }else{
            $cor = "CCCCCC";
          }
        }
?>
  <tr bgcolor="<?=$cor ?>">
    <td align="center"><?=$lista_usuario['id_usuario'] ?></td>
    <td align="center"><?=$lista_usuario['nome_usuario'] ?></td>
    <td align="center"><?=$lista_usuario['login_usuario'] ?></td>
    <td align="center"><?=$lista_usuario['email_usuario'] ?></td>
  </tr>
  <?
        } //fecha while
  } //fecha else
  <?>

<tr>
  <td colspan="5" align="center">

```

```

                <input type="button"
onClick="javascript:location.href('usuario.php')" value="VOLTAR" />
            </td>
        </tr>
    </table>
</body>
</html>

```

### software.php – Lista opções de busca por software

```

<?
include "../valida_cookie.php";
?>
<html>
<head>
    <title>Gerenciar Softwares</title>
</head>
<link rel="stylesheet" type="text/css" href="../css/estilo.css">
<body>
<center>
<h1>Gerenciar Softwares</h1><br><br><br><br>
<table border="0">
    <tr>
        <td>&nbsp;</td>
        <td><a href="busca_software_win.php" target="principal">Buscar por M&acute;quina
Windows</a></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td><a href="busca_software_linux.php" target="principal">Buscar por M&acute;quina
Linux</a></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td><a href="alteracao_software.php" target="principal">Poss&iacute;veis
Altera&ccedil;&otilde;es</a></td>

```

```

</tr>
</table>
</center>
</body>
</html>

```

### **busca\_software\_win.php – Busca os softwares em computadores Windows.**

```

<?
include "../valida_cookie.php";
?>
<html>
<head>
    <title>Busca por Software - Windows</title>
</head>
<link rel="stylesheet" type="text/css" href="../css/estilo.css">
<body>
<center>
<h1>Busca por M&aacute;quina Windows</h1>
<br><br><br><br>
<form method="post" action="lista_software_win.php" name="form1">
<table border="0">
    <tr>
        <td align="right">Endere&cedil;o MAC: </td>
        <td><input type="text" name="rede_MAC" size="20" maxlength="20"></td>
    </tr>
    <tr>
        <td colspan="2" align="center">
            <input type="submit" value="BUSCAR" name="buscar">
        </td>
    </tr>
</table>
</form>
</center>
</body>

```

```
</html>
```

### lista\_software\_win.php – Lista o resultado da busca anterior.

```
<?
include "../valida_cookie.php";
$rede_MAC = $_POST["rede_MAC"];
echo $sql= " SELECT sw.id_software, sw.id, sw.sw_Nome, sw.sw_Data_Instalacao,
sw.sw_Versao
          FROM tb_software sw,
          tb_hardware hw
          WHERE hw.rede_MAC = ".$rede_MAC."
          AND sw.id = hw.id";

$resultado=mysql_query($sql);
if (mysql_num_rows($resultado)==0){
    echo "<strong><center>Nenhum software foi localizado.</strong></center>";
    return false;
}else{
?>
<h1><center>Lista de Softwares:</center></h1><p>
<h2><center>(Endereço MAC: <?=$rede_MAC?>)</center></h2><p>
<table border="1" align="center" cellpadding="3" cellspacing="1" bgcolor="#000000">
  <tr style="color:#FFFFFF; font:bold ; text-align:center">
    <td ><strong>Nome do Software</strong></td>
    <td><strong>Data de Instalação</strong></td>
    <td><strong>Versão</strong></td>
  </tr>
?>
    $cor="CCCCCC";
    while ($lista_sw = mysql_fetch_array($resultado)){
    if ($cor == "CCCCCC"){
        $cor = "FFFFFF";
    }else{
        $cor = "CCCCCC";
    }

```

```

        }
?>
<tr bgcolor="<?=$cor ?>">
  <td align="center"><?=$lista_sw['sw_Nome'] ?></td>
  <td align="center"><?=$lista_sw['sw_Data_Instalacao'] ?></td>
  <td align="center"><?=$lista_sw['sw_Versao'] ?></td>
</tr>
<?
        } //fecha while
} //fecha else
        ?>

<tr>
  <td colspan="5" align="center">
    <input type="button" onClick="javascript:location.href('busca_software_win.php')"
value="VOLTAR" />
  </td>
</tr>
</table>
</body>
</html>

```

### **busca\_software\_linux.php – Busca os softwares em computadores Linux**

```

<?
include "../valida_cookie.php";
?>
<html>
<head>
        <title>Busca por Software - Linux</title>
</head>
<link rel="stylesheet" type="text/css" href="../css/estilo.css">
<body>
<center>
<h1>Busca por M&aaacute;quina Linux</h1>
<br><br><br><br>

```

```

<form method="post" action="lista_software_linux.php" name="form1">
<table border="0">
  <tr>
    <td align="right">Endereço MAC: </td>
    <td><input type="text" name="rede_MAC" size="20" maxlength="20"></td>
  </tr>
  <tr>
    <td colspan="2" align="center">
      <input type="submit" value="BUSCAR" name="buscar">
    </td>
  </tr>
</table>
</form>
</center>
</body>
</html>

```

### **lista\_software\_linux.php – Lista o resultado da busca anterior.**

```

<?
include "../valida_cookie.php";
$rede_MAC = $_POST["rede_MAC"];
echo $sql= " SELECT id_sw_linux, txt_linux
           FROM tb_sw_linux sw
           WHERE txt_linux = ".$rede_MAC.".txt";
$resultado=mysql_query($sql);
if (mysql_num_rows($resultado)==0){
    echo "<strong><center>Nenhum software foi localizado.</strong></center>";
    return false;
}else{
?>
<h1><center>Lista de Softwares:</center></h1><p>
<h2><center>(Endereço MAC: <?=$rede_MAC?></center></h2><p>
<table border="1" align="center" cellpadding="3" cellspacing="1" bgcolor="#000000">
  <tr style="color:#FFFFFF; font:bold ; text-align:center">

```

```

        <td ><strong>Nome do Software</strong></td>
        <td><strong>Data de Instalação</strong></td>
        <td><strong>Versão</strong></td>
    </tr>
<?
        $cor="CCCCCC";
        while ($lista_sw = mysql_fetch_array($resultado)){
?>
    <tr>
        <td align="right">Pacotes de Softwares:</td>
        <td><textarea name="txt_linux" cols="50" rows="10"><?=$lista_sw["txt_linux"]
?></textarea></td>
            <td>&nbsp;</td>
    </tr>
<?
            } //fecha while
} //fecha else
        ?>

    <tr>
        <td colspan="5" align="center">
            <input type="button"
onClick="javascript:location.href('busca_software_linux.php')" value="VOLTAR" />
        </td>
    </tr>
</table>
</body>
</html>

```

### hardware.php – Lista opções de busca por hardware

```

<?
include "../valida_cookie.php";
?>
<html>
<head>

```

```

        <title>Gerenciar Hardwares</title>
</head>
<link rel="stylesheet" type="text/css" href="../css/estilo.css">
<body>
<center>
<h1>Gerenciar Hardwares</h1><br><br><br><br>
<table border="0">
  <tr>
    <td>&nbsp;</td>
    <td><a href="busca_hardware_win.php" target="principal">Buscar por M&aacute;quina
Windows</a></td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td><a href="busca_hardware_linux.php" target="principal">Buscar por M&aacute;quina
Linux</a></td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td><a href="alteracao_hardware.php" target="principal">Poss&iacute;veis
Altera&ccedil;&otilde;es</a></td>
  </tr>
</table>
</center>
</body>
</html>

```

### **busca\_hardware\_win.php – Busca os hardwares em computadores Windows**

```

<?
include "../valida_cookie.php";
?>
<html>
<head>
    <title>Busca por Hardware</title>

```

```

</head>
<link rel="stylesheet" type="text/css" href="../css/estilo.css">
<body>
<center>
<h1>Busca por M&aacute;quina Windows</h1>
<br><br><br><br>
<form method="post" action="lista_hardware_win.php" name="form1">
<table border="0">
  <tr>
    <td align="right">Endere&cedil;o MAC: </td>
    <td><input type="text" name="rede_MAC" size="20" maxlength="20"></td>
  </tr>
  <tr>
    <td colspan="2" align="center">
      <input type="submit" value="BUSCAR" name="buscar">
    </td>
  </tr>
</table>
</form>
</center>
</body>
</html>

```

### **lista\_hardware\_win.php – Lista o resultado da busca anterior.**

```

<?
include "../valida_cookie.php";
$rede_MAC      = $_POST["rede_MAC"];
$sql= " SELECT *
        FROM tb_hardware
        WHERE rede_MAC = ".$rede_MAC."";
$resultado=mysql_query($sql);
if (mysql_num_rows($resultado)==0){
    echo "<strong><center>Nenhum hardware foi
localizado.</strong></center>";

```

```

        return false;
    }else{
        ?>
        <h1><center>Lista de Hardware:</center></h1><p>
        <h2><center>(Endereço MAC: <?=$rede_MAC?></center></h2><p>
        <table border="1" align="center" cellpadding="3" cellspacing="1" bgcolor="#000000">
        <tr style="color:#FFFFFF; font:bold ; text-align:center">
            <td ><strong>Memória Ram Total</strong></td>
            <td><strong>Qtde. Pentes</strong></td>
            <td><strong>HD Total</strong></td>
            <td ><strong>HD Qtde.</strong></td>
            <td><strong>HD Modelo</strong></td>
            <td><strong>Mouse</strong></td>
            <td ><strong>Teclado</strong></td>
            <td><strong>Tipo Monitor</strong></td>
            <td><strong>Id. Monitor</strong></td>
            <td ><strong>Endereço MAC</strong></td>
            <td><strong>Nome da rede</strong></td>
            <td><strong>Fabricante CPU</strong></td>
            <td ><strong>Id. CPU</strong></td>
            <td><strong>Velocidade CPU</strong></td>
            <td ><strong>Família CPU</strong></td>
            <td><strong>Fabricante Placa-Mãe</strong></td>
            <td><strong>Modelo Placa-Mãe</strong></td>
            <td ><strong>Serial Placa-Mãe</strong></td>
        </tr>
        <?
            $cor="CCCCCC";
            while ($lista_hw = mysql_fetch_array($resultado)){
            if ($cor == "CCCCCC"){
                $cor = "FFFFFF";
            }else{
                $cor = "CCCCCC";
            }
    }

```

```

?>
<tr bgcolor="<?=$cor ?>">
  <td align="center"><?=$lista_hw['ram_Total'] ?></td>
  <td align="center"><?=$lista_hw['ram_Pentes'] ?></td>
  <td align="center"><?=$lista_hw['hd_Total'] ?></td>
  <td align="center"><?=$lista_hw['hd_Quantidade'] ?></td>
  <td align="center"><?=$lista_hw['hd_Modelo'] ?></td>
  <td align="center"><?=$lista_hw['mouse_Fabricante'] ?></td>
  <td align="center"><?=$lista_hw['teclado_Fabricante'] ?></td>
  <td align="center"><?=$lista_hw['monitor_Tipo'] ?></td>
  <td align="center"><?=$lista_hw['monitor_Identificacao'] ?></td>
  <td align="center"><?=$lista_hw['rede_MAC'] ?></td>
  <td align="center"><?=$lista_hw['rede_Nome'] ?></td>
  <td align="center"><?=$lista_hw['cpu_Fabricante'] ?></td>
  <td align="center"><?=$lista_hw['cpu_ID'] ?></td>
  <td align="center"><?=$lista_hw['cpu_Velocidade'] ?></td>
  <td align="center"><?=$lista_hw['cpu_Familia'] ?></td>
  <td align="center"><?=$lista_hw['pm_Fabricante'] ?></td>
  <td align="center"><?=$lista_hw['pm_Modelo'] ?></td>
  <td align="center"><?=$lista_hw['pm_Serial'] ?></td>
</tr>
<?
      } //fecha while
} //fecha else
      ?>

<tr>
  <td colspan="18" align="center">
    <input type="button"
onClick="javascript:location.href('busca_hardware_win.php')" value="VOLTAR" />
  </td>
</tr>
</table>
</body>
</html>

```

**busca\_hardware\_linux.php – Busca os hardwares em computadores Linux**

```

<?
include "../valida_cookie.php";
?>
<html>
<head>
    <title>Busca por Hardware</title>
</head>
<link rel="stylesheet" type="text/css" href="../css/estilo.css">
<body>
<center>
<h1>Busca por Hardware Linux</h1>
<br><br><br><br>
<form method="post" action="lista_hardware_linux.php" name="form1">
<table border="0">
    <tr>
        <td align="right">Endereço MAC: </td>
        <td><input type="text" name="rede_MAC" size="20" maxlength="20"></td>
    </tr>
    <tr>
        <td colspan="2" align="center">
            <input type="submit" value="BUSCAR" name="buscar">
        </td>
    </tr>
</table>
</form>
</center>
</body>
</html>

```

**lista\_hardware\_linux.php – Lista o resultado da busca anterior.**

```

<?

```

```

include "../valida_cookie.php";
$rede_MAC      = $_POST["rede_MAC"];
echo $sql= " SELECT id_hw_linux, xml_linux
            FROM tb_hw_linux sw
            WHERE xml_linux = ".$rede_MAC.".xml";
$resultado=mysql_query($sql);
if (mysql_num_rows($resultado)==0){
    echo "<strong><center>Nenhum hardware foi
localizado.</strong></center>";
    return false;
}else{
?>
<h1><center>Lista de Hardware:</center></h1><p>
<h2><center>(Endereço MAC: <?=$rede_MAC?>)</center></h2><p>
<table border="1" align="center" cellpadding="3" cellspacing="1" bgcolor="#000000">
<tr style="color:#FFFFFF; font:bold ; text-align:center">
<td ><strong>Memória Ram Total</strong></td>
<td><strong>Qtde. Pentes</strong></td>
<td><strong>HD Total</strong></td>
<td ><strong>HD Qtde.</strong></td>
<td><strong>HD Modelo</strong></td>
<td><strong>Mouse</strong></td>
<td ><strong>Teclado</strong></td>
<td><strong>Tipo Monitor</strong></td>
<td><strong>Id. Monitor</strong></td>
<td ><strong>Endereço MAC</strong></td>
<td><strong>Nome da rede</strong></td>
<td><strong>Fabricante CPU</strong></td>
<td ><strong>Id. CPU</strong></td>
<td><strong>Velocidade CPU</strong></td>
<td ><strong>Família CPU</strong></td>
<td><strong>Fabricante Placa-Mãe</strong></td>
<td><strong>Modelo Placa-Mãe</strong></td>
<td ><strong>Serial Placa-Mãe</strong></td>

```

```

</tr>
<?
    $cor="CCCCCC";
    while ($lista_hw = mysql_fetch_array($resultado)){
    if ($cor == "CCCCCC"){
        $cor = "FFFFFF";
    }else{
        $cor = "CCCCCC";
    }
?>
<tr bgcolor="<?=$cor ?>">
    <td align="center"><?=$lista_hw['ram_Total'] ?></td>
    <td align="center"><?=$lista_hw['ram_Pentes'] ?></td>
    <td align="center"><?=$lista_hw['hd_Total'] ?></td>
    <td align="center"><?=$lista_hw['hd_Quantidade'] ?></td>
    <td align="center"><?=$lista_hw['hd_Modelo'] ?></td>
    <td align="center"><?=$lista_hw['mouse_Fabricante'] ?></td>
    <td align="center"><?=$lista_hw['teclado_Fabricante'] ?></td>
    <td align="center"><?=$lista_hw['monitor_Tipo'] ?></td>
    <td align="center"><?=$lista_hw['monitor_Identificacao'] ?></td>
    <td align="center"><?=$lista_hw['rede_MAC'] ?></td>
    <td align="center"><?=$lista_hw['rede_Nome'] ?></td>
    <td align="center"><?=$lista_hw['cpu_Fabricante'] ?></td>
    <td align="center"><?=$lista_hw['cpu_ID'] ?></td>
    <td align="center"><?=$lista_hw['cpu_Velocidade'] ?></td>
    <td align="center"><?=$lista_hw['cpu_Familia'] ?></td>
    <td align="center"><?=$lista_hw['pm_Fabricante'] ?></td>
    <td align="center"><?=$lista_hw['pm_Modelo'] ?></td>
    <td align="center"><?=$lista_hw['pm_Serial'] ?></td>
</tr>
<?
        } //fecha while
} //fecha else
?>

```

```

<tr>
  <td colspan="18" align="center">
    <input type="button"
onClick="javascript:location.href('busca_hardware_win.php')" value="VOLTAR" />
  </td>
</tr>
</table>
</body>
</html>

```

### relatorio.php – Lista opções de relatório geral

```

<?
include "../valida_cookie.php";
?>
<html>
<head>
  <title>Relat&ocute;rios</title>
</head>
<link rel="stylesheet" type="text/css" href="../css/estilo.css">
<body>
<center>
<h1>Relat&ocute;rios</h1><br><br><br><br>
<table border="0">
  <tr>
    <td>&nbsp;</td>
    <td><a href="relatorio_maquina_win.php"
target="principal">Relat&ocute;rio por Máquina Windows</a></td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td><a href="relatorio_maquina_linux.php"
target="principal">Relat&ocute;rio por Máquina Linux</a></td>
  </tr>
</table>

```

```

        <td>&nbsp;</td>
                <td><a href="relatorio_completo.php" target="principal">Relat&ocirc;rio
Completo</a></td>
    </tr>
</table>
</center>
</body>
</html>

```

### busca\_relatorio\_maquina\_win.php – Busca o relatório geral de uma Máquina Windows

```

<?
include "../valida_cookie.php";
?>
<html>
<head>
        <title>Relatório por Máquina Windows</title>
</head>
<link rel="stylesheet" type="text/css" href="../css/estilo.css">
<body>
<center>
<h1>Relatório por Máquina Windows</h1>
<br><br><br><br>
<form method="post" action="relatorio_maquina_win.php" name="form1">
<table border="0">
    <tr>
        <td align="right">Endere&ccedil;o MAC: </td>
        <td><input type="text" name="rede_MAC" size="20" maxlength="20"></td>
    </tr>
    <tr>
        <td colspan="2" align="center">
                <input type="submit" value="BUSCAR" name="buscar">
        </td>
    </tr>
</table>

```

```

</form>
</center>
</body>
</html>

```

### relatorio\_maquina\_win.php – Lista todos os hardwares e softwares existentes

```

<?
include "../valida_cookie.php";
$rede_MAC = $_POST["rede_MAC"];
$sql= " SELECT *
        FROM tb_hardware hw, tb_software sw
        WHERE rede_MAC = ".$rede_MAC."
        AND hw.id = sw.id";

$resultado=mysql_query($sql);
if (mysql_num_rows($resultado)==0){
    echo "<strong><center>Nenhum hardware ou software foi
localizado.</strong></center>";
    return false;
}else{
?>
<h1><center>Lista de Hardware e Software:</center></h1><p>
<h2><center>(Endereço MAC: <?=$rede_MAC?>)</center></h2><p>
<table border="1" align="center" cellpadding="3" cellspacing="1" bgcolor="#000000">
<tr style="color:#FFFFFF; font:bold ; text-align:center">
<td ><strong>Memória Ram Total</strong></td>
<td><strong>Qtde. Pentes</strong></td>
<td><strong>HD Total</strong></td>
<td ><strong>HD Qtde.</strong></td>
<td><strong>HD Modelo</strong></td>
<td><strong>Mouse</strong></td>
<td ><strong>Teclado</strong></td>
<td><strong>Tipo Monitor</strong></td>
<td><strong>Id. Monitor</strong></td>

```

```

<td ><strong>Endereço MAC</strong></td>
<td><strong>Nome da rede</strong></td>
<td><strong>Fabricante CPU</strong></td>
<td ><strong>Id. CPU</strong></td>
<td><strong>Velocidade CPU</strong></td>
<td ><strong>Família CPU</strong></td>
<td><strong>Fabricante Placa-Mãe</strong></td>
<td><strong>Modelo Placa-Mãe</strong></td>
<td ><strong>Serial Placa-Mãe</strong></td>
</tr>
<?
    $cor="CCCCCC";
    while ($lista_hw = mysql_fetch_array($resultado)){
        if ($cor == "CCCCCC"){
            $cor = "FFFFFF";
        }else{
            $cor = "CCCCCC";
        }
    }
?>
<tr bgcolor="<?=$cor ?>">
    <td align="center"><?=$lista_hw['ram_Total'] ?></td>
    <td align="center"><?=$lista_hw['ram_Pentes'] ?></td>
    <td align="center"><?=$lista_hw['hd_Total'] ?></td>
    <td align="center"><?=$lista_hw['hd_Quantidade'] ?></td>
    <td align="center"><?=$lista_hw['hd_Modelo'] ?></td>
    <td align="center"><?=$lista_hw['mouse_Fabricante'] ?></td>
    <td align="center"><?=$lista_hw['teclado_Fabricante'] ?></td>
    <td align="center"><?=$lista_hw['monitor_Tipo'] ?></td>
    <td align="center"><?=$lista_hw['monitor_Identificacao'] ?></td>
    <td align="center"><?=$lista_hw['rede_MAC'] ?></td>
    <td align="center"><?=$lista_hw['rede_Nome'] ?></td>
    <td align="center"><?=$lista_hw['cpu_Fabricante'] ?></td>
    <td align="center"><?=$lista_hw['cpu_ID'] ?></td>
    <td align="center"><?=$lista_hw['cpu_Velocidade'] ?></td>

```

```

<td align="center"><?=$lista_hw['cpu_Familia'] ?></td>
<td align="center"><?=$lista_hw['pm_Fabricante'] ?></td>
<td align="center"><?=$lista_hw['pm_Modelo'] ?></td>
<td align="center"><?=$lista_hw['pm_Serial'] ?></td>
</tr>
<?
    } //fecha while
    ?>
</table>
<br>
<table border="1" align="center" cellpadding="3" cellspacing="1" bgcolor="#000000">
<tr style="color:#FFFFFF; font:bold ; text-align:center">
<td ><strong>Nome do Software</strong></td>
<td><strong>Data de Instalação</strong></td>
<td><strong>Versão</strong></td>
</tr>
<?
    $cor="CCCCCC";
    while ($lista_sw = mysql_fetch_array($resultado)){
    if ($cor == "CCCCCC"){
        $cor = "FFFFFF";
    }else{
        $cor = "CCCCCC";
    }
    ?>
<tr bgcolor="<?=$cor ?>">
<td align="center"><?=$lista_sw['sw_Nome'] ?></td>
<td align="center"><?=$lista_sw['sw_Data_Instalacao'] ?></td>
<td align="center"><?=$lista_sw['sw_Versao'] ?></td>
</tr>
<?
    } //fecha while
} //fecha else
    ?>

```

```

<tr>
  <td colspan="5" align="center">
    <input type="button" onClick="javascript:location.href('busca_software_win.php')"
value="VOLTAR" />
  </td>
</tr>
</table>
</body>
</html>

```

### relatorio\_maquina\_linux.php - Busca o relatório geral de uma Máquina Linux

```

<?
include "../valida_cookie.php";
?>
<html>
<head>
  <title>Relatório por Máquina Linux</title>
</head>
<link rel="stylesheet" type="text/css" href="../css/estilo.css">
<body>
<center>
<h1>Relatório por Máquina Linux</h1>
<br><br><br><br>
<form method="post" action="relatorio_maquina_linux.php" name="form1">
<table border="0">
  <tr>
    <td align="right">Endereço MAC: </td>
    <td><input type="text" name="rede_MAC" size="20" maxlength="20"></td>
  </tr>
  <tr>
    <td colspan="2" align="center">
      <input type="submit" value="BUSCAR" name="buscar">
    </td>
  </tr>

```

```

</table>
</form>
</center>
</body>
</html>

```

### relatorio\_maquina\_linux.php – Lista todos os hardwares e softwares existentes

```

<?
include "../valida_cookie.php";
$rede_MAC = $_POST["rede_MAC"];
$sql= " SELECT *
        FROM tb_hw_linux hw, tb_sw_linux sw
        WHERE hw.xml_linux = ".$rede_MAC.".xml'
        AND sw.txt_linux = ".$rede_MAC.".txt";
$resultado=mysql_query($sql);
if (mysql_num_rows($resultado)==0){
    echo "<strong><center>Nenhum hardware ou software foi
localizado.</strong></center>";
    return false;
}else{
?>
<h1><center>Lista de Hardware e Software:</center></h1><p>
<h2><center>(Endereço MAC: <?=$rede_MAC?>)</center></h2><p>
<table border="1" align="center" cellpadding="3" cellspacing="1" bgcolor="#000000">
<tr style="color:#FFFFFF; font:bold ; text-align:center">
<td ><strong>Memória Ram Total</strong></td>
<td><strong>Qtde. Pentes</strong></td>
<td><strong>HD Total</strong></td>
<td ><strong>HD Qtde.</strong></td>
<td><strong>HD Modelo</strong></td>
<td><strong>Mouse</strong></td>
<td ><strong>Teclado</strong></td>
<td><strong>Tipo Monitor</strong></td>
<td><strong>Id. Monitor</strong></td>

```

```

<td ><strong>Endereço MAC</strong></td>
<td><strong>Nome da rede</strong></td>
<td><strong>Fabricante CPU</strong></td>
<td ><strong>Id. CPU</strong></td>
<td><strong>Velocidade CPU</strong></td>
<td ><strong>Família CPU</strong></td>
<td><strong>Fabricante Placa-Mãe</strong></td>
<td><strong>Modelo Placa-Mãe</strong></td>
<td ><strong>Serial Placa-Mãe</strong></td>
</tr>
<?
    $cor="CCCCCC";
    while ($lista_hw = mysql_fetch_array($resultado)){
        if ($cor == "CCCCCC"){
            $cor = "FFFFFF";
        }else{
            $cor = "CCCCCC";
        }
    }
?>
<tr bgcolor="<?=$cor ?>">
    <td align="center"><?=$lista_hw['ram_Total'] ?></td>
    <td align="center"><?=$lista_hw['ram_Pentes'] ?></td>
    <td align="center"><?=$lista_hw['hd_Total'] ?></td>
    <td align="center"><?=$lista_hw['hd_Quantidade'] ?></td>
    <td align="center"><?=$lista_hw['hd_Modelo'] ?></td>
    <td align="center"><?=$lista_hw['mouse_Fabricante'] ?></td>
    <td align="center"><?=$lista_hw['teclado_Fabricante'] ?></td>
    <td align="center"><?=$lista_hw['monitor_Tipo'] ?></td>
    <td align="center"><?=$lista_hw['monitor_Identificacao'] ?></td>
    <td align="center"><?=$lista_hw['rede_MAC'] ?></td>
    <td align="center"><?=$lista_hw['rede_Nome'] ?></td>
    <td align="center"><?=$lista_hw['cpu_Fabricante'] ?></td>
    <td align="center"><?=$lista_hw['cpu_ID'] ?></td>
    <td align="center"><?=$lista_hw['cpu_Velocidade'] ?></td>

```

```

<td align="center"><?=$lista_hw['cpu_Familia'] ?></td>
<td align="center"><?=$lista_hw['pm_Fabricante'] ?></td>
<td align="center"><?=$lista_hw['pm_Modelo'] ?></td>
<td align="center"><?=$lista_hw['pm_Serial'] ?></td>
</tr>
<?
    } //fecha while
    ?>
</table>
<br>
<table border="1" align="center" cellpadding="3" cellspacing="1" bgcolor="#000000">
<tr style="color:#FFFFFF; font:bold ; text-align:center">
<td ><strong>Nome do Software</strong></td>
<td><strong>Data de Instalação</strong></td>
<td><strong>Versão</strong></td>
</tr>
<?
    $cor="CCCCCC";
    while ($lista_sw = mysql_fetch_array($resultado)){
    if ($cor == "CCCCCC"){
        $cor = "FFFFFF";
    }else{
        $cor = "CCCCCC";
    }
    ?>
<tr bgcolor="<?=$cor ?>">
<td align="center"><?=$lista_sw['sw_Nome'] ?></td>
<td align="center"><?=$lista_sw['sw_Data_Instalacao'] ?></td>
<td align="center"><?=$lista_sw['sw_Versao'] ?></td>
</tr>
<?
    } //fecha while
} //fecha else
?>

```

```

<tr>
  <td colspan="5" align="center">
    <input type="button" onClick="javascript:location.href('busca_software_win.php')"
value="VOLTAR" />
  </td>
</tr>
</table>
</body>
</html>

```

### relatorio\_geral.php – Traz todos os dados de todos as máquinas ligadas na rede

```

<?
include "../valida_cookie.php";
$rede_MAC = $_POST["rede_MAC"];
$sql= " SELECT *
        FROM tb_hardware hw, tb_software sw
        WHERE hw.id = sw.id";
$resultado=mysql_query($sql);
if (mysql_num_rows($resultado)==0){
    echo "<strong><center>Nenhum hardware ou software foi
localizado.</strong></center>";
    return false;
}else{
?>
<html>
<body>
<h1><center>Lista de Hardware e Software de PC Windows:</center></h1><p>
<h2><center>(Endereço MAC: <?=$rede_MAC?>)</center></h2><p>
<table border="1" align="center" cellpadding="3" cellspacing="1" bgcolor="#000000">
  <tr style="color:#FFFFFF; font:bold ; text-align:center">
    <td ><strong>Memória Ram Total</strong></td>
    <td><strong>Qtde. Pentes</strong></td>
    <td><strong>HD Total</strong></td>
    <td ><strong>HD Qtde.</strong></td>

```

```

<td><strong>HD Modelo</strong></td>
<td><strong>Mouse</strong></td>
<td ><strong>Teclado</strong></td>
<td><strong>Tipo Monitor</strong></td>
<td><strong>Id. Monitor</strong></td>
<td ><strong>Endereço MAC</strong></td>
<td><strong>Nome da rede</strong></td>
<td><strong>Fabricante CPU</strong></td>
<td ><strong>Id. CPU</strong></td>
<td><strong>Velocidade CPU</strong></td>
<td ><strong>Família CPU</strong></td>
<td><strong>Fabricante Placa-Mãe</strong></td>
<td><strong>Modelo Placa-Mãe</strong></td>
<td ><strong>Serial Placa-Mãe</strong></td>
</tr>
<?
        $cor="CCCCCC";
        while ($lista_hw = mysql_fetch_array($resultado)){
            if ($cor == "CCCCCC"){
                $cor = "FFFFFF";
            }else{
                $cor = "CCCCCC";
            }
        }
?>
<tr bgcolor="<?=$cor ?>">
    <td align="center"><?=$lista_hw['ram_Total'] ?></td>
    <td align="center"><?=$lista_hw['ram_Pentes'] ?></td>
    <td align="center"><?=$lista_hw['hd_Total'] ?></td>
    <td align="center"><?=$lista_hw['hd_Quantidade'] ?></td>
    <td align="center"><?=$lista_hw['hd_Modelo'] ?></td>
    <td align="center"><?=$lista_hw['mouse_Fabricante'] ?></td>
    <td align="center"><?=$lista_hw['teclado_Fabricante'] ?></td>
    <td align="center"><?=$lista_hw['monitor_Tipo'] ?></td>
    <td align="center"><?=$lista_hw['monitor_Identificacao'] ?></td>

```

```

<td align="center"><?=$lista_hw['rede_MAC'] ?></td>
<td align="center"><?=$lista_hw['rede_Nome'] ?></td>
<td align="center"><?=$lista_hw['cpu_Fabricante'] ?></td>
<td align="center"><?=$lista_hw['cpu_ID'] ?></td>
<td align="center"><?=$lista_hw['cpu_Velocidade'] ?></td>
<td align="center"><?=$lista_hw['cpu_Familia'] ?></td>
<td align="center"><?=$lista_hw['pm_Fabricante'] ?></td>
<td align="center"><?=$lista_hw['pm_Modelo'] ?></td>
<td align="center"><?=$lista_hw['pm_Serial'] ?></td>
</tr>
<?
        } //fecha while
    ?>
</table>
<br>
<table border="1" align="center" cellpadding="3" cellspacing="1" bgcolor="#000000">
<tr style="color:#FFFFFF; font:bold ; text-align:center">
<td ><strong>Nome do Software</strong></td>
<td><strong>Data de Instalação</strong></td>
<td><strong>Versão</strong></td>
</tr>
<?
        $cor="CCCCCC";
        while ($lista_sw = mysql_fetch_array($resultado)){
        if ($cor == "CCCCCC"){
            $cor = "FFFFFF";
        }else{
            $cor = "CCCCCC";
        }
    ?>
<tr bgcolor="<?=$cor ?>">
<td align="center"><?=$lista_sw['sw_Nome'] ?></td>
<td align="center"><?=$lista_sw['sw_Data_Instalacao'] ?></td>
<td align="center"><?=$lista_sw['sw_Versao'] ?></td>

```

```

</tr>
<?
        } //fecha while
} //fecha else
        ?>

<tr>
    <td colspan="5" align="center">
        <input type="button" onClick="javascript:location.href('busca_software_win.php')"
value="VOLTAR" />
    </td>
</tr>
</table>

```

```

<?
include "../valida_cookie.php";
$rede_MAC = $_POST["rede_MAC"];
$sql= " SELECT *
        FROM tb_hw_linux hw, tb_sw_linux sw
        WHERE hw.xml_linux = ".$rede_MAC.".xml'
        AND sw.txt_linux = ".$rede_MAC.".txt";
$resultado=mysql_query($sql);
if (mysql_num_rows($resultado)==0){
    echo "<strong><center>Nenhum hardware ou software foi
localizado.</strong></center>";
    return false;
}else{
?>
<h1><center>Lista de Hardware e Software de PC Linux:</center></h1><p>
<h2><center>(Endereço MAC: <?=$rede_MAC?>)</center></h2><p>
<table border="1" align="center" cellpadding="3" cellspacing="1" bgcolor="#000000">
    <tr style="color:#FFFFFF; font:bold ; text-align:center">
        <td ><strong>Memória Ram Total</strong></td>
        <td><strong>Qtde. Pentes</strong></td>

```

```

<td><strong>HD Total</strong></td>
<td ><strong>HD Qtde.</strong></td>
<td><strong>HD Modelo</strong></td>
<td><strong>Mouse</strong></td>
<td ><strong>Teclado</strong></td>
<td><strong>Tipo Monitor</strong></td>
<td><strong>Id. Monitor</strong></td>
<td ><strong>Endereço MAC</strong></td>
<td><strong>Nome da rede</strong></td>
<td><strong>Fabricante CPU</strong></td>
<td ><strong>Id. CPU</strong></td>
<td><strong>Velocidade CPU</strong></td>
<td ><strong>Família CPU</strong></td>
<td><strong>Fabricante Placa-Mãe</strong></td>
<td><strong>Modelo Placa-Mãe</strong></td>
<td ><strong>Serial Placa-Mãe</strong></td>
</tr>
<?
    $cor="CCCCCC";
    while ($lista_hw = mysql_fetch_array($resultado)){
    if ($cor == "CCCCCC"){
        $cor = "FFFFFF";
    }else{
        $cor = "CCCCCC";
    }
?>
<tr bgcolor="<?=$cor ?>">
<td align="center"><?=$lista_hw['ram_Total'] ?></td>
<td align="center"><?=$lista_hw['ram_Pentes'] ?></td>
<td align="center"><?=$lista_hw['hd_Total'] ?></td>
<td align="center"><?=$lista_hw['hd_Quantidade'] ?></td>
<td align="center"><?=$lista_hw['hd_Modelo'] ?></td>
<td align="center"><?=$lista_hw['mouse_Fabricante'] ?></td>
<td align="center"><?=$lista_hw['teclado_Fabricante'] ?></td>

```

```

<td align="center"><?=$lista_hw['monitor_Tipo'] ?></td>
<td align="center"><?=$lista_hw['monitor_Identificacao'] ?></td>
<td align="center"><?=$lista_hw['rede_MAC'] ?></td>
<td align="center"><?=$lista_hw['rede_Nome'] ?></td>
<td align="center"><?=$lista_hw['cpu_Fabricante'] ?></td>
<td align="center"><?=$lista_hw['cpu_ID'] ?></td>
<td align="center"><?=$lista_hw['cpu_Velocidade'] ?></td>
<td align="center"><?=$lista_hw['cpu_Familia'] ?></td>
<td align="center"><?=$lista_hw['pm_Fabricante'] ?></td>
<td align="center"><?=$lista_hw['pm_Modelo'] ?></td>
<td align="center"><?=$lista_hw['pm_Serial'] ?></td>
</tr>
<?
        } //fecha while
    ?>
</table>
<br>
<table border="1" align="center" cellpadding="3" cellspacing="1" bgcolor="#000000">
<tr style="color:#FFFFFF; font:bold ; text-align:center">
<td ><strong>Nome do Software</strong></td>
<td><strong>Data de Instalação</strong></td>
<td><strong>Versão</strong></td>
</tr>
<?
        $cor="CCCCCC";
        while ($lista_sw = mysql_fetch_array($resultado)){
        if ($cor == "CCCCCC"){
            $cor = "FFFFFF";
        }else{
            $cor = "CCCCCC";
        }
    ?>
<tr bgcolor="<?=$cor ?>">
<td align="center"><?=$lista_sw['sw_Nome'] ?></td>

```

```
<td align="center"><?=$lista_sw['sw_Data_Instalacao'] ?></td>
<td align="center"><?=$lista_sw['sw_Versao'] ?></td>
</tr>
<?
        } //fecha while
} //fecha else
        ?>

<tr>
<td colspan="5" align="center">
        <input type="button" onClick="javascript:location.href('busca_software_win.php')"
value="VOLTAR" />
</td>
</tr></table></body></html>
```