



**UNIVERSIDADE ESTADUAL DA PARAÍBA  
CAMPUS VII – GOVERNADOR ANTÔNIO MARIZ  
CENTRO DE CIÊNCIAS EXATAS E SÓCIAS APLICADAS**

**VINÍCIUS REUTEMAN FEITOZA ALVES DE ANDRADE**

**TI VERDE: EXPERIMENTO NO USO DO LINUX UBUNTU COM LTSP NA  
UEPB.**

PATOS  
2014

**VINÍCIUS REUTEMAN FEITOZA ALVES DE ANDRADE**

**TI VERDE: EXPERIMENTO NO USO DO LINUX UBUNTU COM LTSP NA  
UEPB.**

Trabalho de Conclusão de Curso apresentado ao curso de Licenciatura Plena em Computação da Universidade Estadual da Paraíba – UEPB, em cumprimento à exigência para TTC II e obtenção do grau de Licenciado em Computação.

**Orientador:**

**Prof. Dr. Wellington Candeia de Araújo**

**Co-Orientador:**

**Francisco Anderson Mariano da Silva**

PATOS  
2014

UEPB - SIB - Setorial - Campus VII

A554t Andrade, Vinícius Reuteman Feitoza Alves de  
Ti Verde: experimento no uso do Linux Ubuntu com LTSP na  
UEPB [manuscrito] / Vinicius Reuteman Feitoza Alves Andrade. –  
2014.  
54 p. : il. color.

Digitado.  
Trabalho de Conclusão de Curso (Graduação em  
Computação) – Centro de Ciências Exatas e Sociais Aplicadas,  
Universidade Estadual da Paraíba, 2014.  
“Orientação: Prof. Dr. Wellington Candeia de Araújo, CCEA”.

1. Linux Terminal Server Project. 2. TI Verde. 3. Terminal  
Cliente. 4. Inclusão Digital. I. Título.

21. ed. CDD 005.268

Vinicius Reuteman Feitoza Alves Andrade

**TI VERDE: EXPERIMENTO NO USO DO LINUX UBUNTU COM  
LTSP NA UEPB.**

Trabalho de Conclusão de Curso apresentado ao  
Curso de Licenciatura em Computação da  
Universidade Estadual da Paraíba, em cumprimento  
à exigência para obtenção do grau de Licenciado em  
Computação

Aprovado em 15 de julho de 2014

BANCA EXAMINADORA

Wellington Candeia de Araújo

Wellington Candeia de Araújo  
(Orientador)

Elder Eldervitch Carneiro de Oliveira

Elder Eldervitch Carneiro de Oliveira  
(Examinadora)

Janine Vicente Dias

Janine Vicente Dias  
(Examinador)



Dedico esse trabalho primeiramente a Deus, aos meus pais, irmãos e irmãs, esposa e especialmente ao meu filho, sempre lembrando o quanto eles acreditaram no meu potencial e me ajudaram a concluir mais esse sonho, com amor, apoio, compreensão e confiança.

Dedico!

DEDICO!

## AGRADECIMENTO

A **Deus**, por ter me dado força, sabedoria, saúde e maturidade, e por ter iluminado meu caminho permitindo a realização desse trabalho.

A minha mãe **Rita Feitoza Alves de Andrade**, que me deu a vida e me ensinou a vivê-la com dignidade, que se doou para que eu conseguisse mais essa conquista, que me deu amor, respeito, apoio, que não me deixou desistir nas inúmeras vezes que me deixei abater, que sempre esteve presente nos momentos mais difíceis mesmo hoje sem vossa presença.

Ao meu pai **Francisco Alves de Andrade (Zomim)**, que participou e que herdei parte da vida, participando também em meu ensinamento por meio das conversas, dialogo, carinho e amor.

Aos meus irmãos **Victor Rannyo Feitoza A. Andrade, Victoremborg Feitoza A. Andrade, Vicderlannia Feitoza A. Andrade** e demais irmãos, que eu sei que sempre acreditaram na minha capacidade e que vibram comigo essa vitória.

A minha **família**, vocês que sempre fizeram parte da minha história, que de alguma forma também são responsáveis por eu estar finalizando esse curso, e que ainda teremos muitas outras histórias pra viver, muito obrigado.

A minha esposa **Liliane Silva Batista**, por estar sempre por perto, me orientar, me ajudar em todos os momentos, obrigado pela dedicação, você me ajudou a alcançar tudo isso. Amo para sempre e toda a vida.

Ao meu filho **Jonathas Reuteman B. F. Andrade**, o qual se firmou como um pilar para todos os meus desejos e sonhos, além de ser uma luz para minha vida. Te amo, meu filho!

Ao **Prof. Rodrigo Alves Costa** pelo qual me auxiliou inicialmente nas pesquisas realizadas na UEPB.

Ao Técnico de Informática **Giancarlos da Silva Almeida** por auxiliar e proporcionar a execução do LTSP na Universidade Estadual da Paraíba, mesmo com todas as dificuldades encontradas manteve firme nos objetivos traçados. Muito obrigado.

Ao **Prof. Dr. Wellington Candeia de Araújo** que foi muito mais de que um orientador foi um amigo, que me ajudou da melhor forma que existe nessa trajetória, com humildade e dedicação. Muito obrigado.

Agradecer ao **SESI**, em especial ao **CAT DMA** por proporcionar conhecimento prático durante a permanência do estágio como professor de Informática e Técnico do mesmo.

Considero oportuno agradecer a todos que me ajudaram e contribuíram para que chegasse ao final deste curso, porém seria necessário enumerar uma lista muito extensa de todos que me ajudaram durante esse período.

A todos os professores da Universidade Estadual da Paraíba pelo profissionalismo, paciência e incentivo.

*Seja a mudança que você deseja ver no mundo.  
(Mahatma Gandhi)*

# TI VERDE: EXPERIMENTO NO USO DO LINUX UBUNTU COM LTSP NA UEPB.

## RESUMO

O presente trabalho tem como objetivo analisar o uso do Linux Terminal Server Project (LTSP), implantados no Laboratório de Informática da UEPB, seguindo o princípio da TI Verde, propondo o uso dos computadores antigos já em desuso como Terminais Clientes utilizando a união concisa e ordenada de algumas ferramentas e protocolos de comunicação, como DHCP (Dynamic Host Configuration Protocol), NFS (Network File System), TFTP (Trivial File Transfer Protocol) e XDMCP (X Display Manager Control Protocol), disponíveis em uma distribuição GNU/Linux Ubuntu. O trabalho apresenta: um breve histórico sobre a Tecnologia de Informação Verde, com suas tendências e práticas; uma discussão sobre Sistema Operacional Linux em especial o Ubuntu; a descrição do LTSP e sua utilização; os protocolos usados para a comunicação no LTSP; o funcionamento do LTSP mostrando as principais etapas; as vantagens e desvantagens do LTSP; os aplicativos usados para análise do desempenho, Monitoramento de Sistemas e Ferramentas de Rede. A metodologia utilizada foi a da pesquisa bibliográfica, que consiste na pesquisa de várias fontes a respeito do LTSP e Fat Client juntamente com análise e comparações das informações encontradas provenientes do estudo de caso, com a participação da análise e implantação do LSTP na instituição referida, bem como a descrição de todo este processo. A partir dos dados colhidos, pode ser observado que os Terminais Clientes possuem desempenho similar ao Servidor LTSP, obtendo uma diferença considerável quanto ao tempo para inicialização do Fat Client e dos seus aplicativos testados. Foi levada em consideração a média do tempo para inicialização dos aplicativos no Servidor LTSP, no Terminal Cliente e no Fat Client.

**Palavras-chave:** *Linux Terminal Server Project (LTSP); TI Verde; Terminal Cliente; Inclusão Digital.*

## ABSTRACT

This study aims to analyze the use of the Linux Terminal Server Project (LTSP), deployed in the Computer Laboratory of UEPB, following the principle of Green IT, proposing of the use of old computers without use, as terminals Clients using a concise and ordered union of some tools and communication protocols such as DHCP (Dynamic Host Configuration Protocol), NFS (Network File System), TFTP (Trivial File Transfer Protocol) and XDMCP (X Display Manager Control Protocol), available in a GNU / Linux distribution Ubuntu. The study presents: a brief history about the Green Information Technology with its trends and practices; a discussion of Linux OS Ubuntu in particular; LTSP and the description of its use; the protocols used for communication on the LTSP; LTSP showing the operation of its main stages; the advantages and disadvantages of LTSP; applications used to analyze the performance, Monitoring Systems and Network Tools. The methodology used was the literature research, which consists of research from various sources about the LTSP and Fat Client along with analysis and comparison of information found from the case study, involving the analysis and implementation of the LSTP in that institution, and a description of this process. From the data collected, it can be seen that the Clients Terminals feature similar to LTSP server performance, achieving a considerable difference in the time to boot Fat Client and their applications tested. Was taken into account the average time to startup applications in LTSP Server, Terminal Client and Fat Client.

**Keywords:** *Linux Terminal Server Project (LTSP); Green IT; Client Terminal; Digital inclusion.*

## LISTA DE FIGURAS

<b>Figura 1</b>	Logomarca do LTSP.....	09
<b>Figura 2</b>	Estrutura da rede usando Thin Client ou Fat Client LTSP.....	12
<b>Figura 3</b>	Diagrama temporal do arranque remoto do LTSP.....	23
<b>Figura 4</b>	Fat Client usado para o Laboratório de Informática.....	26
<b>Figura 5</b>	<i>Servidor LTSP do Laboratório de Informática.....</i>	28
<b>Figura 6</b>	Laboratório de Informática com LTSP.....	30
<b>Figura 7</b>	Switch de 16 portas, usando no Laboratório de Informática.....	31
<b>Figura 8</b>	Inicialização do Fat Client usando LTSP.....	32
<b>Figura 9</b>	Utilização dos processadores na inicialização dos Fat Clients.....	33
<b>Figura 10</b>	Utilização da memória RAM e dos processadores no uso do aplicativo Chromium	33
<b>Figura 11</b>	Pico da memória RAM e dos processadores com uso do adobe flashplugin.....	34
<b>Figura 12</b>	Aplicativo de gerenciamento dos Fat Clients.....	34

## LISTA DE GRÁFICOS

<b>Gráfico 1</b>	Distribuições que mais usam o LTSP.....	09
<b>Gráfico 2</b>	Países que usam o LTSP.....	11
<b>Gráfico 3</b>	Tempo de inicialização do grub e do Login.....	35
<b>Gráfico 4</b>	Tempo de inicialização do Writer.....	36
<b>Gráfico 5</b>	<i>Tempo de inicialização do Calc.....</i>	36
<b>Gráfico 6</b>	Tempo de inicialização do Impress.....	37

## LISTA DE TABELA

<b>Tabela 1</b>	Tempo dos aplicativos no Sistema Operacional Ubuntu 10.04 do Fat Client.....	28
<b>Tabela 2</b>	Tempo dos aplicativos no Sistema Operacional Ubuntu 10.04 do Servidor.....	29
<b>Tabela 3</b>	Tempo dos aplicativos do Servidor LTSP com uso do Fat Client.....	30

## LISTA DE ABREVIATURAS E SIGLAS

ARPANet	Advanced Research Projects Agency Network
CIO	Chief Information
CPU	Central Processing Unit
CRT	Cathodic Ray Tube
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
FTP	File Transfer Protocol
GNU	Acronico "GNU's Not Unix"
GPL	GNU Public License
HD	Hard Disk
IP	Internet Protocol
LCD	Liquid Cristal Display
LTS	Long Term Support
LTSP	Projeto Servidor Terminal Linux
NFS	Network File System
OLED	Organic Light-Emitting Diode
PC's	Computadores
PXE	Preboot Execution Environment
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
RPC	Remote Procedure Call
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TI	Tecnologia da Informação
TIC	Tecnologia de Informação e Comunicação
UDP	User Datagram Protocol
UDP	User Datagram Protocol
UEPB	Universidade Estadual da Paraíba
WEEE	Waste Electrical and Electronic Equipment
XDMCP	X Display Manager Control Protocol
XDMCP	X Display Manager Control Protocol
XDR	External Data Representation



# SUMÁRIO

1	INTRODUÇÃO .....	1
1.1	METODOLOGIA .....	3
2	FUNDAMENTAÇÃO TEÓRICA .....	6
2.1	TECNOLOGIA DE INFORMAÇÃO VERDE .....	6
2.2	SISTEMA OPERACIONAL .....	7
2.2.1	GNU/LINUX .....	7
2.2.2	UBUNTU .....	8
2.3	O LTSP .....	8
2.4	COMPUTAÇÃO BASEADA EM THIN CLIENT .....	12
2.5	OS BENEFÍCIOS DO LTSP.....	13
2.6	AS DESVANTAGENS DO LTSP.....	14
2.7	SERVIDORES LTSP .....	15
2.8	SEGURANÇA .....	15
3	PROTOCOLOS DE COMUNICAÇÃO DO LTSP .....	17
3.1	O PROTOCOLO TCP/IP .....	17
3.2	O PROTOCOLO DHCP.....	18
3.3	O PROTOCOLO TFTP .....	19
3.4	O PROTOCOLO NFS .....	29
3.5	O PROTOCOLO XDMCP .....	20
4	FUNCIONAMENTO DO LTSP .....	21
4.1	FUNCIONAMENTO DO FAT CLIENT NO AMBIENTE LTSP.....	21
5	APLICATIVOS USADOS .....	24
5.1	HTOP .....	24
5.2	NTOP .....	24
5.3	MONITOR DE SISTEMA .....	24
5.4	FERRAMENTAS DE REDE .....	25
6	COLETA DE DADOS .....	26
6.1	CONFIGURAÇÃO DO FAT CLIENT .....	26
6.2	DESEMPENHO DO FAT CLIENT .....	27
6.3	CONFIGURAÇÃO DO SERVIDOR .....	28
6.4	DESEMPENHO DO SERVIDOR .....	28
6.5	DESEMPENHO DO TERMINAL CLIENTE.....	29
6.6	LTSP NO LABORATÓRIO DA UEPB .....	30
6.6.1	DESEMPENHO DO LTSP NO LABORATÓRIO DA UEPB .....	32
7	ANALISE DOS DADOS .....	35
8	CONCLUSÃO .....	39
9	REFERÊNCIAS .....	40

# 1 INTRODUÇÃO

O computador alguns anos atrás era considerado um artigo de luxo, contudo esses equipamentos se tornaram um eletrodoméstico comum e indispensável em nosso dia a dia.

Conforme CROFFI (2012) o número de computadores no Brasil quase triplicou desde 2005, quando havia cerca de 30 milhões de aparelhos no país, que em 2011 passou para 85 milhões de computadores pessoais. A expectativa é que o número chegue a 140 milhões até final de 2014. Desta forma um problema que a cada dia vem ganhando espaço segundo conforme Environment (2012) para discussões é sobre o lixo eletrônico ou e-lixo e que também é conhecido como *Waste Electrical and Electronic Equipment* (WEEE) ou ainda Resíduos de Equipamentos Elétricos e Eletrônicos, introduzido na legislação do Reino Unido em Janeiro de 2007. Os WEEE contêm materiais recicláveis e substâncias perigosos (metais pesados). Ambos, sem a devida disposição final, causam danos ambientais e problemas de saúde. Os eletrônicos normalmente incluem “metais como lumíneo, antimônio, arsênio, bário, berílio, cádmio, cromo, cobalto, cobre, gálio, ouro, ferro, manganês, chumbo, mercúrio, paládio, platina, selênio, prata e zinco”. (Prado, 2012).

O Brasil possui resolução do Conselho Nacional do Meio Ambiente, órgão ligado ao Ministério do Meio Ambiente. Como relata Conama (2012), o principal foco o descarte de pilhas e baterias, indicando um avanço para uma legislação de responsabilidade ambiental. À medida que novas tecnologias são disponibilizadas no mercado e os aparelhos velhos são substituídos com frequência cada vez maior, por conseguinte o volume de WEEE cresce.

Muitas vezes com uso da *Tecnologia da Informação* (TI), pode ser reutilizada por total ou por parte do WEEE para novos projetos. O sistema com servidor *Linux Terminal Server Project* (LTSP), que possibilita o uso dos computadores antigos já em desuso como terminais de acesso, é um exemplo de TI.

No campus VII, da UEPB possuem computadores o qual estão em desuso e que podem por meio da TI serem reutilizados, sendo alguns destes: monitores *Cathodic Ray Tube* (CRT), *mouses ball*, teclados, gabinetes (contendo processador, placa-mãe, memória RAM e fonte de alimentação). Os computadores podem ser usados, por exemplo: nos laboratórios de informática, como terminais para secretárias, terminais de acesso à internet, para projetores e computadores em salas de aulas. Prolongando desta forma a vida útil desses equipamentos, evitando o descarte e armazenamento em locais impróprios.

Dessa forma o presente TCC busca neste contexto analisar o benefício e desempenho no uso do *LTSP* para Terminais Clientes, que foi implantando na UEPB, cidade de Patos, seguindo os princípios do uso de *TI Verde*, em que o “conjunto de ações efetivas que podem ser colocadas em prática com o objetivo de maximizar o uso dos recursos de informática, sejam estes ativos de hardware, software ou humanos”. (SANCHEZ, 2012).

Os objetivos específicos para o desenvolvimento deste trabalho consiste em:

- Analisar o desempenho de tempo do computador servidor;
- Analisar o desempenho de tempo do computador cliente;
- Reutilizar os computadores antigos;
- Promover a inclusão digital;
- Reduzir custos na criação de laboratórios de informática; e
- Ampliar os recursos tecnológicos e didáticos da UEPB de forma econômica e eficiente.

Devido à necessidade de crescimento e desenvolvimento tecnológica, muitas instituições buscam melhorias em sua *Tecnologia de Informação e Comunicação* (TIC), onde a substituição dos seus equipamentos eletrônicos se torna necessário, provocando conseqüentemente o desuso e até mesmo o descarte dos mesmos. Desta forma, muitos equipamentos eletrônicos, como exemplos os computadores em que poderiam ser reutilizados como terminais de acesso são descartados, e que poderiam ser reutilizados usando recursos como *LTSP* conforme Santos (2013) que “permite a reutilização de computadores defasados e suporte para terminais magros (*Thin Clients*) ao *Linux*”. Tal recurso tem obtido aceitação e resultados excelentes por permitir a reutilização de computadores considerados antigos de maneira eficiente, segura e eficaz.

A reutilização planejada de hardwares, que venha a suprir as necessidades específicas e possa dar subsídio para uso dos *aplicativos* (softwares) mais atuais é chamada de *Tecnologia da Informação Verde* (TI Verde), ou seja, o reaproveitamento de equipamentos eletrônicos de forma planejada e sustentável. Desta forma, foi proposta a utilização do software livre chamado *Linux Terminal Server Project* (LTSP), tendo a aplicação cliente sendo rodada no servidor, usando desta forma a *Central Processing Unit* (CPU) e a memória *Random Access Memory* (RAM) do servidor, enquanto é mostrada a saída de tela na estação usando o teclado e mouse da estação, conforme Mcquillan (2000). Desta forma, todas as informações dos usuários e das estações de trabalho estarão centralizadas. Os computadores

antigos serão utilizados apenas como um dispositivo de entrada e saída, pois tudo será executado e controlado diretamente pelo computador servidor.

Neste contexto, a escolha se deu em visto da UEPB possui diversos computadores antigos em desuso, parecendo ser um meio extremamente adequado para a implantação da TI Verde na forma de Terminal de Acesso ou da criação de Laboratórios para acesso à internet. Pretendemos com isso, minimizar os atuais problemas do *e-lixo*.

## 1.1 METODOLOGIA

O delineamento da pesquisa está relacionado ao tema proposto no TCC e tem como objetivo identificar o método a ser utilizado nesta pesquisa. De acordo com Cervo e Bervian (2002) “o método é a ordem que se deve impor aos diferentes processos necessários para atingir certo fim ou um resultado desejado. Nas ciências, entende-se por método o conjunto de processos empregados na investigação da verdade”.

Para realização do estudo a instituição *Universidade Estadual da Paraíba* (UEPB), sendo uma instituição de nível superior de ensino, pesquisa e extensão. Possui vários campi em toda Paraíba. O objeto de estudo se passa no Campus VII, localizado na cidade de Patos, Estado da Paraíba.

Conforme o Estatuto (2007) publicado no Diário Oficial no ano de 2007:

“UNIVERSIDADE ESTADUAL DA PARAÍBA – UEPB, entidade autárquica estadual, criada pela Lei nº 4.977, de 11 de outubro de 1987, regulamentada pelo Decreto nº 12.404, de 18 de março de 1988, modificado pelo Decreto nº 14.830, de 16 de outubro de 1992, substituta da Universidade Regional do Nordeste, instituída pela Lei Municipal nº 23, de 15 de março de 1966, é uma instituição de nível superior de ensino, pesquisa e extensão, vinculada à Secretaria de Educação e Cultura do Estado da Paraíba e tem sede e foro na cidade de Campina Grande – PB, com atuação em todo o Estado da Paraíba.”

A UEPB possui oito campi, distribuídos em toda Paraíba da seguinte forma:

- Campus I – Campina Grande;
- Campus II – Lagoa Seca;
- Campus III – Guarabira;
- Campus IV – Catolé do Rocha;
- Campus V – João Pessoa;
- Campus VI – Monteiro;

- Campus VII – Patos;
- Campus VIII – Araruna.

A instituição possui 46 (quarenta e seis) cursos de graduação e 02 (dois) de nível técnico. Pertencentes aos cursos de graduação, 25 (vinte e cinco) são no Campus I, 01(um) no Campus II, 05 (cinco) no Campus III, 02 (dois) no Campus IV, 03 (três) no Campus V, 03 (três) no Campus VI, 05 (cinco) no Campus VII e 03 (três) no Campus VIII. Dos cursos técnicos, 01 (um) no Campus II e 01 (um) no Campus IV. Os cursos de graduação e técnicos da UEPB são de responsabilidade da Pró-Reitoria de Ensino de Graduação, que supervisiona e coordena suas atividades, dinamizadas, sobretudo, através do Projeto Pedagógico que centraliza a questão curricular.

O Campus VII da UEPB está situado na cidade de Patos, Paraíba, na Rua Alfredo Lustosa Cabral, CEP 58.706-560, Bairro Salgadinho.

Os cursos pertencentes ao Campus VII são de responsabilidade do Centro de Ciências Exatas e Aplicadas – CCEA, sendo eles:

- Administração – Bacharelado (Diurno e noturno);
- Computação<sup>1</sup> – Bacharelado (Diurno e noturno);
- Física – Licenciatura (Diurno e noturno);
- Matemática – Licenciatura (Diurno e noturno);
- Ciências Exatas – Licenciatura (Diurno e noturno).

Para Oliveira o método como um conjunto de processos que se torna possível conhecerem uma determinada realidade, produzir determinado objeto ou desenvolver certos procedimentos ou comportamentos. Tornado possível identificar como alcançar determinado objetivo.

O método indutivo que na definição de Oliveira (2002) “possibilita o desenvolvimento de enunciados gerais sobre as observações acumuladas de casos específicos ou proposições que possam ter validades universais”.

O estudo de caso é um estudo empírico que investiga um fenômeno atual dentro do seu contexto de realidade, quando as fronteiras entre o fenômeno e o contexto não são claramente definidas e no qual são utilizadas várias fontes de evidencias. Sendo, utilizado

---

1 Em processo de transição do Curso de Licenciatura em Computação para o Bacharel em Computação no ano de 2014.

com frequência cada vez maior por ter diferentes propósitos, tais como:

- Explorar situações da vida real;
- Descrever a situação do contexto em que está sendo feita determinada investigação;
- Explicar as variáveis causais de determinado fenômeno em situações muitas complexas que não possibilitam a utilização de levantamentos e experimentos.

Este TCC apresenta uma pesquisa exploratória e foi desenvolvida com uma metodologia baseada nas etapas apresentadas a seguir: Inicialmente foi feita uma pesquisa bibliográfica sobre os temas TI Verde e LTSP. Em termos metodológicos, este trabalho se enquadra dentro da tipologia de estudo de caso e pesquisa-ação onde, a partir de diversas intencionalidades, passando a compor um vasto mosaico de abordagens teórico-metodológicas, instigando-nos a refletir sobre sua essencialidade epistemológica, bem como sobre suas possibilidades como práxis investigativa. Desta maneira, partindo da observação de teste realizando em *Computadores Antigos (Fat Client)*, no Servidor LTSP e no Terminal Cliente existentes na instituição, procurou responder aos objetivos traçados.

Para o recolhimento dos dados realizou-se testes tomando como base o tempo gasto para inicializar o Sistema Operacional (*Grub* e *Login*) e dos aplicativos (*Writer*, *Calc* e *Impress*) em estudo. Os teste eram realizados num período de tempo máximo de 20'00"00 (vinte minutos) e tomando como notação de tempo *minuto, segundos e milésimo de segundos* (00'00"00). Os testes eram realizados 03 (três) vezes consecutivas, sendo efetuado a média da soma dos 03 (três) tempos obtidos. Para os testes no Sistema Operacional (S.O.) foram executados de três formas: medição do tempo para iniciar o Grub; medição da inicialização do Login com adição do usuário e senha; e medição do tempo do Grub + Login para o Terminal Cliente, já que o mesmo inicia o Login do usuário automaticamente. Nos aplicativos (*Writer*, *Calc* e *Impress*), os testes seguiram o mesmo padrão de notação de tempo, sendo que para o 1º (primeiro) teste o mesmo era incia lizado pela primeira vez e os outros 02 (dois) testes realizados consecutivamente. A média era obtido pela soma dos 03 (três) tempos obtidos.

Este trabalho foi organizado em 06 (seis) partes, estando os iniciais voltados à fundamentação teórica necessária, seguindo para os protocolos de comunicação do LTSP, o funcionamento do LTSP, passando para os aplicativos usados para análise de desempenho, por penúltimo a coleta de dados e finalizando na análise de dados.

## 2 FUNDAMENTAÇÃO TEÓRICA

Na sequência serão apresentados alguns temas relevantes para a consecução do objetivo proposto neste TCC, destacando o conceito: sobre a Tecnologia de Informação Verde, com suas tendências e práticas; sobre Sistema Operacional Linux em especial o Ubuntu; o *LTSP* e sua utilização; os protocolos usando para a comunicação no *LTSP*; o funcionamento do *LTSP* mostrando as principais etapas; as vantagens e desvantagens do *LTSP*; e os aplicativos usados para análise do desempenho, o Monitoramento de Sistemas e Ferramentas de Rede.

### 2.1 TECNOLOGIA DE INFORMAÇÃO VERDE

O significado de TI conforme Takahashi (2009) “é um acrônimo de Tecnologia da Informação, que pode ser definida como o conjunto de todas as atividades e soluções providas por recursos de computação”. As grandes corporações e o mundo passaram a se preocupar com o impacto dos recursos tecnológicos no meio ambiente. Desta forma os conceitos de sustentabilidade e desenvolvimento econômico são recorrentes e se inserem em todos os segmentos da sociedade. Todos participam direta ou indiretamente das ações que podem ser nomeada como TI Verde ou *Green IT*, ou ainda, Tecnologia da Informação Verde que é uma tendência mundial voltada para o impacto dos recursos tecnológicos no meio ambiente. A preocupação dessa tendência está desde a utilização mais eficiente de energia, dos recursos e insumos na produção de tecnologia, assim como uso de matéria prima e substâncias menos tóxicas na fabricação, abrange recursos tecnológicos que consumam menos energia, que não agridam o meio ambiente na sua utilização e por fim não proporcione ou minimize impactos no seu descarte, permitindo reciclagem e sua reutilização.

Com a adoção das práticas de TI Verde existe a possibilidade de reduzir a degradação do meio ambiente, procurando reaproveitar hardwares ou destinando para locais adequados que realizem separação seletiva de seus componentes. A substituição dos monitores comuns de tubo *CRT* por *Liquid Cristal Display (LCD)* ou de *Organic Light-Emitting Diode (OLED)* que ajudam a reduzir a energia elétrica, a consolidação ou virtualização dos servidores auxiliam na melhoria de processamento. A economia de energia pode chegar até a 50% (cinquenta por cento), só com a utilização mais eficiente do equipamento, a readequação dos sistemas de controle do ambiente com ar-condicionado, promover a reciclagem e o descarte controlado do lixo eletrônico.

A TI Verde é uma forma de mostrar que neste mundo globalizado e com a redução de recursos naturais a TI pode ajudar as instituições e empresas. Podendo fornecer caminhos para as instituições resolverem seus problemas cumprindo com a sua missão e protegendo o meio ambiente, reutilizando equipamentos considerados obsoletos, projetando formas para economizar energia elétrica através da mudança de processos e práticas. Com tudo isso a TI deve englobar em suas funções a de preservar o meio ambiente.

## 2.2 SISTEMA OPERACIONAL

### 2.2.1 GNU/LINUX

O Sistema Linux é foi desenvolvido por Linus Torvalds, com suporte à rede nativo, em formato 32 *bits* e 64 *bits*, baseado no formato UNIX de Sistema Operacional, com código fonte aberto e otimizado para internet, com uso crescente mundialmente, tanto em servidores quanto *usuários domésticos* (desktops), segundo Silva (2006) sendo “um sistema operacional multitarefa e multiusuário que tem a vantagem de rodar em uma grande variedade de computadores”. Desta forma, pode ser executado em diferentes *hardwares*, que vai desde processadores de família Intel x86 até RISC, passando por PALM e arquitetura SPARC.

O sistema Linux é dividido em duas partes, onde a primeira seria o *kernel* (núcleo), que é o sistema responsável pela comunicação com o *hardware* e a segunda parte são os programas e serviços que dependem do *kernel* para sua interação.

Uma das características mais conhecidas no Linux é que de ser um software livre, de acordo com o GNU.org (2012):

“devemos entender aquele software que respeita a liberdade e senso de comunidade dos usuários. (...) os usuários possuem a liberdade de executar, copiar, distribuir, estudar, mudar e melhorar o software. Com essas liberdades, os usuários (tanto individualmente quanto coletivamente) controlam o programa e o que ele faz por eles.”

Podendo ser obtido gratuitamente, baseado na licença *GNU Public License* (GPL) e da licença criada por Richard Stallman, fundador da *Free Software Foundation*. O usuário pode “usar, copiar, modificar e redistribuir os códigos-fonte e binários livremente”. (Tanenbaum, 2009).

A expressão software livre, de acordo com a GNU.org (2012) é uma questão de liberdade, mas não de ser cobrado um preço pelo *software*. Desta forma, para se caracterizar como *software* livre, os usuários teriam que desfrutar de quatro liberdades essenciais:



- a liberdade de executar o programa, para qualquer propósito;
- a liberdade de estudar como o programa funciona, e adaptá-lo às suas necessidades;
- a liberdade de redistribuir cópias de modo que você possa ajudar ao próximo; e
- a liberdade de distribuir cópias de suas versões modificadas a outros.

## 2.2.2 UBUNTU

O Ubuntu segundo a Ubuntu-br.org (2014) é um sistema operacional de código aberto, construído a partir do núcleo *Linux*, baseado no Debian, diferenciando-se por ser lançado semestralmente, por disponibilizar suporte técnico nos 18 (dezoito) meses seguintes ao lançamento de cada versão, possuindo as versões *Long Term Support* (LTS) para desktop tendo um período de 3 (três) anos de suporte, e para servidor recebem 5 (cinco) anos. O Ubuntu propõe oferecer um sistema que qualquer pessoa possa utilizar sem dificuldades, independentemente de nacionalidade, nível de conhecimento ou limitações físicas. O sistema deve ser constituído principalmente por *Software* livre. Deve também ser isento de qualquer taxa.

Em 2005 Mark Shuttleworth e a Canonical Ltd. anunciaram a criação da Fundação Ubuntu, dirigido por Jane Silber, tendo como finalidade garantir apoio e desenvolvimento a todas as versões posteriores à 5.10. O nome Ubuntu “deriva do conceito sul-africano de mesmo nome, diretamente traduzido como ‘humanidade com os outros’ ou ‘sou o que sou pelo que nós somos’” (Ubuntu-br.org, 2014).

## 2.3 O LTSP

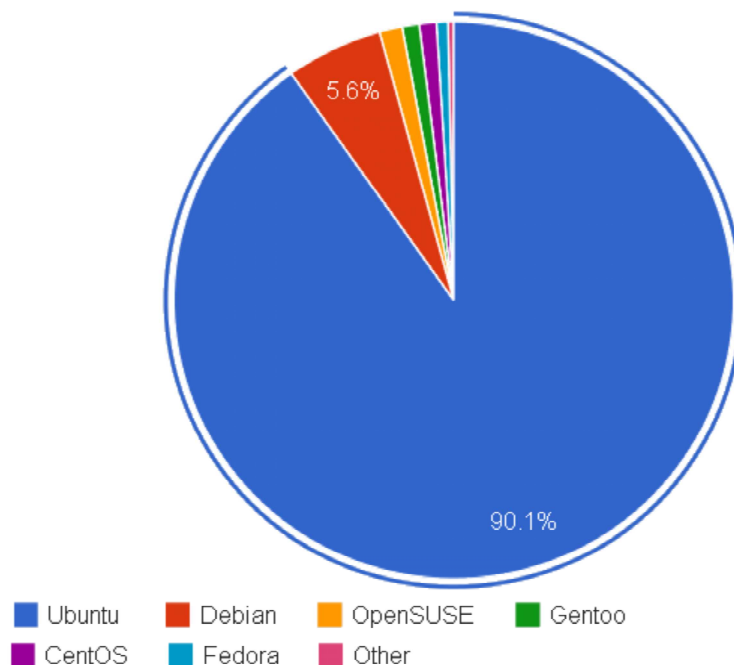
O Projeto de Servidor de Terminais Linux ou *Linux Terminal Server Project* (LTSP) foi criado por James Mcquillan, em 1999, nos Estados Unidos, buscando soluções para um cliente em Detroit, Michigan USA, Binson's Home Health Care Centers. O cliente necessitava de um método barato e funcional que permitisse adicionar 35 novos usuários à sua rede e acessassem uma aplicação baseada em servidores *Unix* em um servidor IBM AS/400. (MCQUILLAN, 2000).



**Figura 1-** Logomarca do LTSP

**Fonte:** [www.ltsp.org](http://www.ltsp.org) (2013).

Atualmente o *LTSP* (logomarca na *Figura 1*) conta com a contribuição de vários desenvolvedores em todo o mundo, tendo o Projeto LTSP-BR como site oficial de suporte no Brasil. O *LTSP* conforme Mcquillan (2000) vem a fornecer um meio simples de utilizar estações de baixo custo como terminais gráficos ou caracteres em um servidor *GNU/Linux* de código aberto licenciado pela *GNU* (Acrônimo para GNU is not Unix GPL), comumente chamado de *GNU GPL*, sendo formado por um conjunto de ferramentas que juntas proporcionam um ambiente de terminais gráficos de baixo custo, conectados a um servidor que centraliza e processa as aplicações da rede, tendo no *Gráfico 1* as principais distribuições que usam o *LTSP*.



**Gráfico 1 –** Distribuições que mais usam o LTSP

**Fonte:** [www.ltsp.org](http://www.ltsp.org) (2013).

Acessando o endereço eletrônico do *LTSP*<sup>2</sup> é possível encontrar a última versão deste conjunto de ferramentas que são baseadas em software livre. Com isto o *LTSP* torna-se uma solução atrativa para projetos de rede que propõem a diminuição de gastos com software, dispensando o custo com licenças proprietárias. Além disso, o fato de todos os aplicativos serem rodadas no próprio servidor faz com que as outras máquinas clientes não precisem de grande poder de processamento economizando, desta forma, o custo com hardware.

O *LTSP* é um software utilizado para promover um ambiente de trabalho em rede a baixo custo que após um longo período de teste e utilização passou por inúmeras modificações e atualizações. Hoje ele se encontra na 5.3.7 (5a) versão estável, sendo incluído no repositório de algumas distribuições *Linux* e na facilidade na instalação e configuração.

Segundo Morimoto (2006) o *LTSP* “é uma solução mais usada para a criação de terminais leves”. Os arquivos de configuração e os protocolos necessários para a inicialização destes terminais ficam armazenados em um servidor e são transferidos assim que solicitados através da rede.

Partindo da ideia de Takahashi (2009) da reutilização de hardwares ou do e-lixo, podendo muitas vezes reaproveitar os *computadores antigos (Fat Clients)* que não estão mais sendo utilizados ou que não dão suporte a softwares mais modernos, para usá-los em uma rede que utilize a tecnologia *LTSP*. Mesmo no caso de aquisição de novos computadores para a rede, não há a preocupação de atualizar o hardware delas constantemente, podendo passar muitos anos em perfeito funcionamento e com desempenho satisfatório. O computador que precisará ser atualizada com o tempo ou com o acréscimo de máquinas é o servidor, pois o mesmo é responsável por todas as aplicações da rede, e sempre que necessário, terão que ser feitas nele, mudanças de hardware e *software* visando à melhoria do seu desempenho.

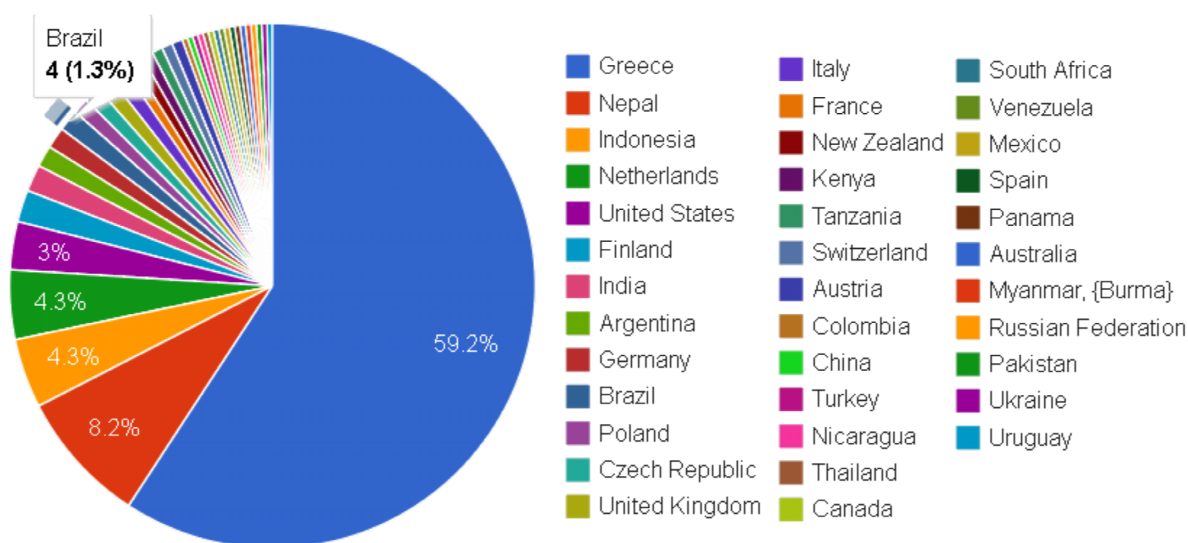
Os *Fat Clients* podem ser reaproveitados e serem utilizados da mesma forma que os *Thin Clients* onde existe a possibilidade de serem usados para navegar na *Web*, enviar *e-mail*, criar documentos, e executar outras aplicações *desktop*. O *LTSP* por si só proporciona várias vantagens, pois sua adoção praticamente obriga a utilizar-se de boas práticas de gerência em infraestrutura. Quando o *LTSP* é implantado com *Thin Clients* seus benefícios são ainda mais potencializados, permitindo atingir níveis altíssimos de economia em vários aspectos diretamente e indiretamente influenciados pela adoção da solução. Permite um retorno do investimento, em curto espaço de tempo, como também permite um baixo custo de propriedade. Mas, o mais importante é agrega valor sobre soluções de computação

---

2 [www.ltsp.org](http://www.ltsp.org)

tradicionais, como por exemplo, reutilizar os *desktops* antigos.

Por ser um projeto de código aberto mantido pela comunidade *Linux* no mundo como mostra no *Gráfico 2*, possui a contribuição de inúmeros programadores e estudiosos, tendo grande aceitação pelas instituições públicas de ensino, nas quais já existem vários laboratórios de informática para uso dos alunos, servindo de auxílio às atividades escolares e extracurriculares.



**Gráfico 2** – Países que usam o LTSP.  
**Fonte:** [www.ltsp.org](http://www.ltsp.org) (2013).

Utilizando o *LTSP*, o computador cliente não necessita do *disco rígido* (HD), disquete e de CD-ROM, sendo que o computador cliente deverá possuir uma placa de rede com suporte a *boot* por rede. Durante a fase de boot, a estação sem disco obtém suas informações IP e o *kernel* do servidor, e então monta o sistema de arquivos raiz via *Network File System* (NFS).

A estação de trabalho pode ser configurada em um dos três modos:

- Interface Gráfica X *Window System*: usando o X *Windows*, a estação de trabalho pode ser usada para acessar qualquer aplicação no servidor, ou em outros servidores na rede;
- Interface Caractere Sessões *Telnet*: as estações de trabalho podem obter múltiplas sessões *telnet* no servidor. Cada sessão *telnet* estará em uma janela virtual separada; e
- Aviso *Shell*: a estação de trabalho pode ser configurada para cair diretamente em um

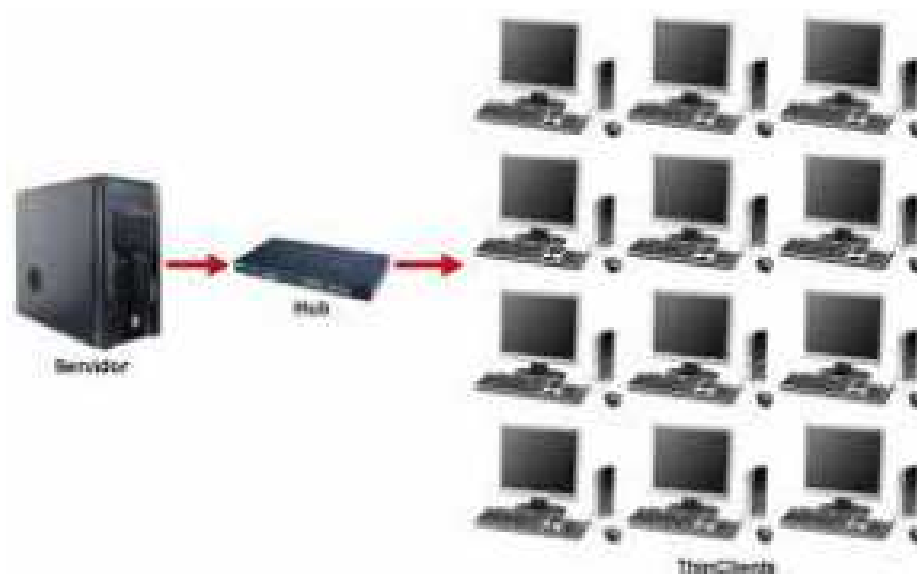
*shellbash* no console. Isto é muito útil quando estiver depurando problemas com o *X Windows* ou *NFS*.

O desempenho depende do quanto robusto<sup>3</sup> é o servidor, através do *LTSP* os terminais poderão executar programas que exigem alto desempenho ou somente programas de escritório comum. Isto só é possível porque o poder de processamento do servidor será compartilhado com os terminais, o que é imperceptível para o usuário, que irá desfrutar todos os recursos computacionais que o servidor dispôr. Sendo que quanto mais usuários conectados ao servidor *LTSP* mais robusto deve ser o servidor, conseqüentemente, mais será gasto no servidor.

O *LTSP* não deve ser visto apenas como um utilizador de *Fat Clients*, pois reutilizar o hardware antigo dispênde-se de muita mão de obra, pode-se gastar mais horas do que se imagina e está mais propenso a defeitos. Além disso, seu hardware não é otimizado para economia de energia, diferente dos *Thin Clients*, mas é um conceito interessante sobre e-lixo.

## 2.4 COMPUTAÇÃO BASEADA EM THIN CLIENT

Conforme Balneaves (2009) o termo *Thin Client Computing* ou *Computação para Terminais Leves* está em constante utilização, um dos motivos é a facilidade no gerenciamento dos terminais, economia no suporte, hardware e custos de upgrade. Mesmo com a crescente utilização, o conceito tem-se mantido o mesmo, com arquitetura cliente-servidor, ao invés de executar aplicações localmente. As aplicações são executadas do servidor e processadas no mesmo, distribuindo somente atualizações de *teclado, vídeo e mouse* (KVM), e do áudio.



**Figura 2:** Estrutura da rede usando thin client ou fat client LTSP.  
3 **Fonte:** [www.ltsp.org](http://www.ltsp.org) (2013).

A *Figura 1* mostra um exemplo de uma arquitetura cliente-servidor para que se possa ter um melhor entendimento de como é estruturada o *Thin Client* e *Fat Clients*.

David (2002) demonstra que a computação baseada em *Thin Clients* é uma opção importante a considerar na estratégia de negócios de qualquer organização. Quando implantada corretamente, ela pode reduzir significativamente seus custos operacionais, aumentar a qualidade de seu suporte e reduzir o tempo em que os usuários ficam sem trabalhar devido a problemas técnicos.

## 2.5 BENEFÍCIOS DO LTSP

Uma das vantagens mais atrativas do uso de *Thin Clients* ou *Fat Clients* seria o corte de custos. De qualquer maneira, a computação baseada em servidores oferece muitos outros benefícios além de simplesmente reduzir custos:

- Acesso direto a arquivos de trabalho e aplicações empresariais: a computação baseada em *Thin Clients* aumentou a produtividade por permitir que os usuários acessem o seu ambiente de trabalho de qualquer lugar no mundo.
- Redução da administração e suporte ao usuário final: os *Fat Clients* são altamente simples de gerenciar e tendo um único ponto de administração reduz, sobretudo, os custos com a administração.
- Gerenciamento e Manutenção: menores custos de manutenção são obtidos, pois as atualizações de aplicativos, antivírus e *patches* podem ser executados apenas no servidor.
- Maior segurança: menor risco de *vírus*: o uso de *Fat Clients* permite que os administradores possam restringir o acesso a dispositivos de armazenamento local como *flash cards*, *floppies*, cartões de memória *USB* e *CD ROMS*. É possível também prevenir os usuários que carreguem aplicações estrangeiras aos dispositivos, isto incrementa o nível de segurança e virtualmente elimina *vírus* e *malware*.
- Disco rígido e Segurança da informação: por não necessitar de um disco rígido, permitindo uma maior capacidade de armazenamento seguro de dados e aplicativos do servidor.

- Recuperação de desastres: é mais seguro e fácil de fazer backup dos dados, se um terminal falhar, nenhum dado será perdido, simplesmente será armazenado no servidor.
- Fácil gerenciamento de licenças e conformidade com os requerimentos legais: próprio para a centralização, licenciamento de software torna-se mais fácil para monitorar e gerenciar.
- Reduz o dispêndio de capital em hardware computacional: possível estender o tempo de vida de um computador convertendo-o para *Fat Client*.

Os *Thin Clients* e os *Fat Clients* podem ser configurados em poucos minutos e permitem fácil implantação de novos usuários em locais remotos, também, podem ser configurados e gerenciados remotamente. (BALNEAVES, 2009).

## 2.6 AS DESVANTAGENS DO LTSP

Para implantar um *LTSP*, deve-se fazer um estudo de aplicabilidade para um projeto completo, pois é possível utilizar normalmente *Fat Clients* em conjunto com servidores virtualizados, proporcionando todos os benefícios associados à virtualização. Conforme Davis (2008) as desvantagens em usar os *Thin Clients* não funcionam bem para rodar aplicativos complexos e também não servem à mobilidade.

Quanto mais usuários se conectarem ao Servidor *LTSP* mais recursos computacionais o servidor terá que dispor, necessitando muitas vezes um investimento massivo em servidores.

A centralização no servidor facilita a manutenção, mas revela ser um ponto negativo por concentrar um ponto de falha, tornando-se quase indispensável um plano de contingência, conseqüentemente haverá um maior gasto pertinente à prevenção de falhas. Desta forma, como todas as informações estão em um único lugar deve-se tomar muito cuidado com a disponibilidade, pois haverá um único ponto de falha.

As empresas que pretendem realizar migração encontrarão resistência dos usuários, geralmente “habitados a um ambiente livre de gerenciamento, onde os desktops são configurados de acordo com suas preferências pessoais, costumam ver o Thin Client como perda de controle pessoal”. (DAVIS, 2008).

## 2.7 SERVIDORES LTSP

O servidor é responsável por executar todas as aplicações e contém todos os dados de todos os usuários. As manutenções regulares desde as atualizações de software e administração são realizadas no servidor. O número de *Fat Clients* que um servidor pode suportar é proporcional a sua capacidade de processamento.

Em um ambiente de computação baseada em *Thin Clients*, a estabilidade do servidor é fundamental para garantir um bom funcionamento. É importante ter a certeza de que o servidor tenha uma boa capacidade de suportar aplicações robustas, e dependendo do quanto é necessário a disponibilidade, fontes de alimentação redundantes podem ser utilizadas, ou até mesmo servidores e *switchs* redundantes ou outras soluções de prevenção de falhas.

Para aplicações em microempresas, escolas, laboratórios, empreendimentos de baixo risco e baixa disponibilidade, certamente não será necessário todas as soluções de alta disponibilidade, por exemplo. No nosso caso, como se destina para laboratório de acesso (acessar *internet*, editar documentos texto e programas de baixo desempenho) não teria tanto problema para implantação. Serão descritos mais adiante as configurações do servidor e dos computadores usando como *Fat Clients*.

## 2.8 SEGURANÇA

Muitas organizações, empresas e instituições de ensino não possuem pessoas especializadas em TI ou de tempo para uma melhoria na infraestrutura, em sua segurança e manutenção preventiva de computadores. No projeto *LTSP* a segurança tornou-se um desafio fundamental, que está sempre em constante melhoria. Por se tratar de um ambiente *Unix-like*<sup>4</sup> possuem uma das vantagens do *Linux* que é a segurança. Como seu código é aberto, pessoas no mundo todo podem fazer correções e melhorias no sistema. Isto se traduz em maior qualidade de código, onde este ambiente é pouquíssimo afetado por *spywares*<sup>5</sup> e *vírus*<sup>6</sup> (BALNEAVES, 2009).

---

4 Sistemas operacionais baseados em Unix.

5 Programa mal-intencionado que atua como um espião.

6 É um software malicioso que é desenvolvido por programadores que infecta o sistema, faz cópias de si mesmo e tenta se espalhar para outros computadores.



No *LTSP* ainda conta com uma rigorosa política de segurança pró-ativa, o que significa que muitos problemas comuns, tais como portas abertas ou *softwares* mal configurados, não ocorrerão na versão liberada do *LTSP*. Os sistemas baseados em *LTSP* são sistemas operacionais multiusuário, tornando fácil a permissão para que os usuários executem suas tarefas de modo prático, sem ter um nível de acesso que possa comprometer o sistema.

Como o *LTSP* baseado em *Thin Clients* conta com benefícios de segurança de *diskless*<sup>7</sup>. Caso um ladrão leve um Terminal Cliente, ele não terá acesso a dados críticos, pois os mesmos ficam gravados remotamente por exemplo.

---

7 Conceito de computadores interligados em rede, numa arquitetura cliente/servidor, que não usa disco rígido.

### 3 PROTOCOLOS DE COMUNICAÇÃO DO LTSP

Para ser iniciada uma estação *LTSP* sem o HD são envolvidos alguns passos. Onde, conhecendo esses passos torna mais fácil a solução de problemas e a correção dos mesmos.

Há 5 (cinco) protocolos de comunicação que são necessários para iniciar uma estação *LTSP*. Cada um dos protocolos relacionados abaixo pode ser oferecido pelo mesmo servidor, ou de diferentes servidores, sendo eles:

- TCP/IP;
- DHCP;
- TFTP;
- NFS; e
- XDMCP.

#### 3.1 O PROTOCOLO TCP/IP

Entre 1969 a 1972 foi criada a *ARPANet (Advanced Research Projects Agency Network)*, que deu origem a atual Internet, onde o sucesso foi tão grande que em 1973 já estavam interligando 30 instituições, incluindo algumas grandes empresas. Desta forma foi criado o padrão Ethernet, que foi usado a priori para criar redes locais, e os protocolos de comunicação “TCP/IP (Transmission Control Protocol/Internet Protocol), além de muitas outras melhorias que usamos ainda hoje como o e-mail, o telnet e o FTP (File Transfer Protocol)”. (MORIMOTO, 2005).

O protocolo TCP/IP propôs interligar vários sistemas de arquiteturas diferentes como se fosse um só sistema. Com o tempo, devido a vários aspectos, ganhou popularidade sendo hoje o protocolo mais utilizado em redes locais.

Os Terminais Burros, ou Terminais Leves, ou *Thin Clients* já vem sendo discutidos a mais de 40 (quarenta) anos atrás, sendo a tecnologia mais disseminada com os antigos mainframes da IBM, e vem acompanhando a evolução no decorrer dos anos. Hoje ressurgiu com força total com o nome de *Thin Clients*, movimentado as indústrias e as grandes corporações, que estão repensando seus conceitos sobre TI. (CARMONA, 2005).

Como desempenho e implantação dos *Thin Clients* estão diretamente relacionados as redes de computadores e esta última aos protocolos de comunicação, faz-se necessário para melhor compreender a estrutura lógica das redes uma breve descrição do modelo de camadas,

em especial o modelo TCP/IP.

A popularização do protocolo TCP/IP se deu por dois motivos: por ser um protocolo leve e possuía um baixo custo de utilização. Devido a esses fatores teve seu sucesso garantido com a popularização da Internet e a agregação do protocolo pelos fabricantes de tecnologias de rede. O TCP/IP foi tão difundido que se tornou um padrão mundial, sendo composto de 04 (quatro) camadas:

- Camada de Aplicação: Segundo Torres (2001), esta camada é formada pelos protocolos utilizados pelas diversas aplicações do modelo TCP/IP. A referida camada não possui um padrão comum. O padrão é estabelecido por cada aplicação. Isto é, o *File Transfer Protocol* (FTP) possui seu próprio protocolo, assim como o *TELNET*, *Simplex Mail Transfer Protocol* (SMTP), *Post Office Protocol* (POP3), *Domain Name System* (DNS) e outros.
- Camada de Transporte: De acordo com Torres (2001), a camada de transporte do TCP/IP é responsável por pegar os dados enviados pela camada de aplicação e transformá-los em pacotes, a serem repassados para a camada de *Internet*. É uma camada fim-a-fim, isto é, uma entidade desta camada só se comunica com a sua entidade par do *host* destinatário. É nesta camada que se faz o controle da conversação entre as aplicações intercomunicadas da rede. Dois protocolos aqui são usados: o TCP e o UDP. O TCP é orientado à conexão e o UDP não. O acesso das aplicações à camada de transporte é feito através de portas que recebem um número inteiro para cada tipo de aplicação.
- Camada IP: Essa camada é responsável pelo endereçamento, roteamento e controle de envio e recepção. Ela não é orientada à conexão, comunica-se através de datagramas.
- Camada de Rede: Camada de abstração de *hardware* tem como principal função a interface do modelo TCP/IP com os diversos tipos de redes. É responsável por enviar a datagrama recebido pela camada de Internet em forma de um quadro, através da rede.

### **3.2 O PROTOCOLO DHCP**

O *Dynamic Host Configuration Protocol* (DHCP) é um protocolo que atua na camada de aplicação da pilha de protocolos do TCP/IP conforme Kurose (2006) “permite que um hospedeiro obtenha um endereço IP automaticamente, bem como adquira informações adicionais, tais como sua máscara de sub-rede, o endereço do seu roteador do primeiro salto

(frequentemente denominado default gateway) e o endereço do seu servidor DNS local”. Desta forma se torna um protocolo bastante útil, pois não há necessidade de configuração manual de endereçamento na rede e todas as configurações são atribuídas aos clientes por um servidor DHCP.

O serviço DHCP permite vários tipos de configuração tais como atribuição de endereços somente para uma determinada faixa de IP, bem como uma configuração estática para algumas estações da rede, dentre outros.

O DHCP é o protocolo para distribuição dinâmica de endereços de rede. É instalado em um computador servidor e configurado para que cada computador cliente receba as configurações de rede automaticamente.

O servidor DHCP é o primeiro a ser acessado pela estação. Logo na inicialização do *boot* é o DHCP que responde entregando as configurações da rede, informando qual *Kernel* ou cliente PXE a estação deve carregar e em qual compartilhamento de rede (no servidor) está o sistema a ser carregado. (MORIMOTO, 2005).

### **3.3 O PROTOCOLO TFTP**

O *Trivial File Transfer Protocol* (TFTP) é um protocolo de transferência de arquivos bem similar ao FTP. O TFTP é utilizado para a transmissão de pequenos arquivos entre os computadores de uma rede, tais como terminais remotos ou clientes que queiram iniciar uma sessão com um servidor. É um protocolo bem simples, tende a se comunicar rápido por utilizar portas UDP que não necessitam fazer verificação de erros na transmissão dos dados. Os dados são simplesmente transmitidos da forma mais rápida e simples possível. (MORIMOTO, 2011).

O TFTP é um protocolo que necessita de um pequeno *software* que seja armazenado em cada cliente, podendo estar nos chips das placas de redes ou em alguma mídia externa (CD-ROM, disquete) o que proporciona ao terminal carregar uma imagem de *boot* do servidor LTSP com o drive da placa de rede. (MORIMOTO, 2005).

### **3.4 O PROTOCOLO NFS**

O *Network File System* (NFS) provê o acesso remoto, de forma transparente, a arquivos compartilhados em redes de computadores. O NFS foi projetado para ser portátil

entre diferentes plataformas de hardware, sistemas operacionais, arquiteturas de redes e protocolos de nível de transporte. Esta portabilidade é possível graças ao uso de dos protocolos RPC e XDR. O protocolo NFS têm a intenção de ser o mais "stateless"<sup>8</sup> quando possível, liberando o servidor de manter quaisquer informação de estado de protocolo dos clientes. (GARCIA, 2012).

Através do NFS, usuários e programas da rede podem acessar remotamente pastas e arquivos no servidor, como se estivessem fazendo isso em sua estrutura local de diretórios. Além de pastas e arquivos, o protocolo NFS também tem a capacidade de compartilhar, em uma rede, dispositivos, como CD-ROM, disquetes e outros periféricos.

Com o NFS funcionando, a estação avança mais um pouco no *boot*. Agora ela consegue montar o diretório raiz e começar o carregamento do sistema, mas para no ponto em que procura pelo arquivo *lts.conf*, que é justamente o principal arquivo de configuração do LTSP, carregado pelas estações no início do *boot*. (MORIMOTO, 2006).

### 3.5 O PROTOCOLO XDMCP

O *X Display Manager Control Protocol* (XDMCP) atua como um protocolo de compartilhamento de *desktop* utilizado pelo X, onde foi desenvolvido em seu início para ser usado em mainframes rodando *Unix*. Sendo um protocolo bastante sólido e rápido de comunicação via rede. O terminal encaminha o que é recebido pelo servidor através da rede, direto para a saída de vídeo. Isso faz com que o servidor receba toda a carga, possibilitando um mínimo de processamento dos clientes. Com um mínimo de configuração de hardware já é possível ter um terminal X funcional. (MORIMOTO, 2005).

O XDMCP transmite as imagens e gráficos na forma de comandos que consomem relativamente pouca banda da rede e são rapidamente processados pelo destinatário, fazendo com que, apesar de rodar a distância, o usuário não perceba demora na atualização das imagens.

---

8 Descreve um sistema ou processo que não monitorar todos os detalhes de seu estado.

## 4 FUNCIONAMENTO DO LTSP

No funcionamento do *LTSP* cada Terminal Cliente exibe uma sessão de usuário do servidor. Para o Servidor *LTSP* cada Terminal Cliente ligado a ele representa apenas mais uma sessão aberta, enquanto para cada usuário este seria um computador que funciona como uma máquina nova e completa, herdando a configuração do servidor. Embora a sua única função seja exibir uma tela do que se passa em outro computador de melhor desempenho conforme Morimoto (2006):

“Ele utiliza uma combinação de DHCP, TFTP, NFS e XDMCP para permitir que as estações não apenas rodem aplicativos instalados no servidor, mas realmente deem boot via rede, baixando todos os softwares de que precisam diretamente do servidor. Não é preciso ter HD nem CD-ROM nas estações, apenas um disquete (ou CD) de boot ou ainda um chip de boot espetado na placa de rede.”

O *LTSP* é um conjunto de serviços que transforma um *GNU/Linux* normal em um servidor de Terminal Cliente, seja ele *Fat Client* ou *Thin Client*. Isto permite que computadores antigos de baixo poder computacional sejam usados como terminais leves. Como o *LTSP*, não necessita de software no lado do cliente, exigindo apenas uma interface de rede com *Pre-Execution Environment* (PXE) habilitado, o qual muitos *Thin Clients* e *Fat Clients* já têm embutido em seu *firmware* ou BIOS. O PXE é um protocolo de *boot* remoto desenvolvido pela Intel, que consiste em um pequeno software, gravado na ROM da placa de rede ou na placa-mãe, que permite que o computador realize *boot* através da rede. O protocolo PXE é não precisa absolutamente de nenhum suporte para dispositivos físicos de armazenamento local (seja disco rígido ou flash compacto), para inicializar um *Thin Clients* ou *Fat Clients* pelo *LTSP*. Além de reduzir significativamente o gerenciamento necessário para manter a sua rede funcionando, pois todo gerenciamento se encontra centralizado no servidor.

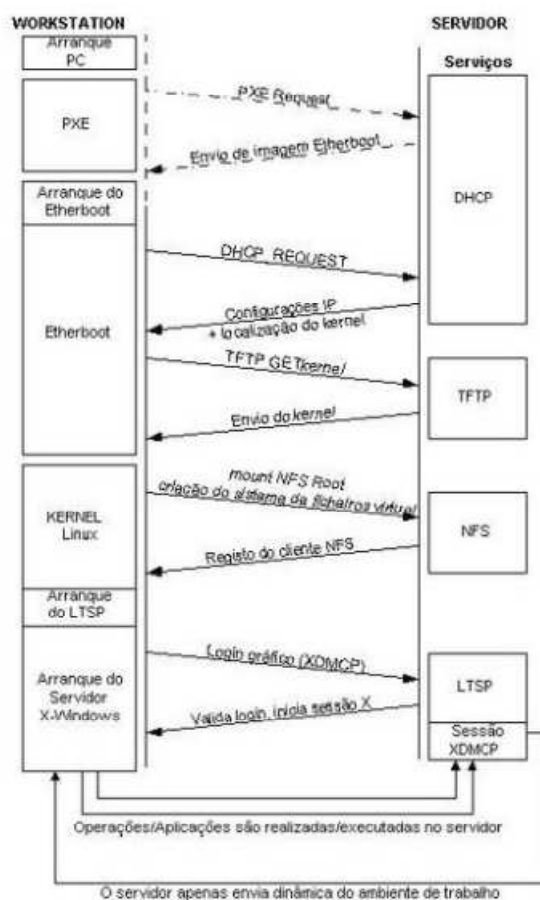
### 4.1 FUNCIONAMENTO DO TERMINAL CLIENTE NO AMBIENTE LTSP

Segundo Morimoto (2006), o *LTSP* seria uma pequena distribuição *Linux*, que através da rede é carregada na memória RAM dos Terminais Clientes. Um conjunto de pacotes que compõe o *LTSP* cria um sistema de arquivos em uma pasta chamada “/opt/ltsp/<arquitetura específica>” com a versão da sua arquitetura no servidor, que é compartilhada com a rede e acessada via NFS pelas máquinas clientes como se fosse uma partição local. Neste diretório,

existe um sistema simplificado, que detecta o *hardware* do computador cliente e permite que ele abra uma sessão em modo gráfico.

Logo após o termino do *boot*, o cliente recebe a tela de *Login* que é fornecida pelo servidor XDMCP. Após esse processo, o Terminal Cliente apenas mostra na tela as imagens geradas, sendo o servidor responsável por rodar todos os aplicativos.

O *LTSP* usa uma série de serviços, para que possa ser carregado nos Terminais Clientes. Primeiro o Terminal Cliente dá o *boot*, através de uma imagem de *boot*, que pode ser gravada em um *chip* de *boot* da própria placa de rede, disquete ou CD-ROM. Dentro dessa imagem, há um *software* pequeno, que ativa a placa de rede e envia um pacote de *broadcast* e pede a configuração da rede. O servidor DHCP, que é instalado no servidor *LTSP*, responde esse chamado e manda a configuração da rede para o cliente, enviando também informações de *Kernel* que o cliente carrega via TFTP e a pasta do servidor que vai ser acessada via protocolo NFS. Quando o *Kernel* estiver carregado, via TFTP, começa realmente o *boot* da estação. Depois disso, o TFTP é substituído por um cliente NFS que é usado para montar a pasta “/opt/ltsp/<arquitetura específica>” do servidor como diretório raiz e finalmente o cliente carrega o sistema *LTSP* que detecta o hardware da estação e abre o servidor gráfico. Conforme ilustrado na *Figura 3* a seguir.



**Figura 3** – Diagrama temporal do arranque remoto do LTSP.  
**Fonte:** [www.ltsp.org](http://www.ltsp.org) (2013).

O Servidor *LTSP* roda todos os aplicativos que são usados pelos Terminais Clientes, garantindo o compartilhamento de recursos. Analisando um Servidor *LTSP*, mesmo com muitas máquinas ligadas a ele, utilizando seus recursos, essa percentagem de utilização continua muito baixa, não chegando nem a 15% do processamento total do servidor. Assim, quando um usuário tenta executar um programa a partir de um cliente, quase sempre encontra o processador livre, como se ele tivesse sozinho no servidor.

O uso da memória RAM é compartilhado de forma que os aplicativos abertos são carregados na memória do servidor apenas uma vez, independente do número de usuários que o utilizem simultaneamente. O sistema carrega o aplicativo uma vez e depois passa a abrir diferentes seções do mesmo programa, tornando o carregamento mais rápido e o uso da memória otimizado.



## 5 APLICATIVOS USADOS

Para verificar e efetuar comparativos quanto ao desempenho do computador servidor, serão usados alguns aplicativos que mostram em forma de números ou gráficos a sua utilização. Desta forma serão usados 2 (dois) aplicativos para estes fins, sendo o HTOP e NTOP. Já o monitoramento do uso da rede, usaremos mais 2 (dois) aplicativos, o Monitor de Rede e Monitor de Sistema, que serão descritos nos tópicos a seguir.

### 5.1 HTOP

O Htop é um avançado sistema interativo visualizador de processos. Escrito para *Linux*, o Htop mostra uma lista frequentemente atualizada de processos que rodam no computador, e utiliza-se de cores para facilitar a leitura de informações sobre o processador, *swap*, status da memória RAM entre outros.

### 5.2 NTOP

O Ntop consiste em uma ferramenta para monitorar e gerenciar redes, além de ter muitos recursos que demonstram tudo através de gráficos e informações detalhadas que permitem com que haja interação entre usuários e tem suporte para rodar em vários Sistemas Operacionais (S.O.), monitora, gera relatórios sobre o tráfego e suporte dos *hosts* por estes protocolos: TCP/UDP/ICMP, (R)ARP, IPX, DLC, *Decnet*, *AppleTalk*, *Netbios*, TCP/UDP.

Ntop é uma ferramenta de monitoramento e gerência de rede para sistemas *UNIX/Linux* e *Win32*, e teve seu desenvolvimento e estudo na "University of Pisa", na Itália. (DERI, 2008).

Seus recursos permitem ao usuário visualizar através de gráficos e de forma bem detalhada, todas as requisições a internet. Qual a banda consumida? Quais endereços das máquinas? *Reports* (avisos) de falhas, monitorando passivamente uma rede, coletando dados sobre os protocolos e sobre os *hosts* da rede.

### 5.3 MONITOR DE SISTEMA

O aplicativo Monitor de Sistema foi desenvolvido por Baptiste Mille-Mathias e hoje conta com vários desenvolvedores. O Monitor de Sistema permite a exibição de informações

básicas do S.O. e processos de sistema de monitoramento, o uso de recursos do sistema, e sistemas de arquivos. Podendo ser usado também para modificar o comportamento do seu sistema. (MONITOR, 2011).

## **5.4 FERRAMENTAS DE REDE**

A Ferramenta de Rede GNOME conforme Network (2013) contém um conjunto de ferramentas para rede orientadas para clientes com informação da rede, um *shell* clientes remotos e *desktop*, um servidor *web* pessoal e uma ferramenta de compartilhamento de *desktop*. O Dispositivo de Ferramenta de Rede é um *Software* Livre distribuído sob os termos da GNU GPL.

## 6 COLETA DE DADOS

### 6.1 CONFIGURAÇÃO DO FAT CLIENT

Os computadores a serem usados são provenientes do Laboratório de Informática, que são formados por lotes diferentes, possuindo configurações diferentes, sendo:

- 3 (três) computadores com processadores AMD Sempron, com frequência de 2.2 Ghz, com 256 MB de Memória RAM, placa de rede onboard, *mouse* PS2, teclado PS2, monitor CRT (*Cathodic Ray Tube*) de 15" e Leitor/Gravador de DVD/CD;
- 12 (doze) computadores com processadores Intel Celeron, com frequência de 2.13 GHz, com 256 MB de Memória RAM, placa de rede *onboard*, mouse PS2, teclado PS2, monitor CRT de 15" e Leitor/Gravador de DVD/CD.

O computador que será usado para uso nos testes de inicialização do S.O. e dos aplicativos, possui processador Intel Celeron, frequência de 2.13 GHz, com 512 MB de Memória RAM, placa de rede *onboard*, *mouse* PS2 Óptico, teclado PS2, monitor CRT de 15", Leitor/Gravador de DVD/CD e disco rígido (HD) de 80 GB. Foram acrescentados mais 256 MB de memória RAM, concretizando 512 MB, para a configuração mínima exigida para suporte ao Ubuntu 10.04. A *Figura 4* mostra o computador usado para teste.



**Figura 4** – Fat client usado para o Laboratório de Informática.  
**Fonte:** Pesquisador do estudo (2011).

## 6.2 DESEMPENHO DO FAT CLIENT

Para observar o desempenho foram realizados 3 (três) testes consecutivos, onde todos serão cronometrados desde seu estado inicial de processo e finalizados quando aparecer toda a estrutura visual do aplicativo. Após os tempos serem cronometrados será feita a média dos 3 (três) tempos. Para realização dos testes o computador permanecerá no máximo 20 (vinte) minutos ligados, para não ocorrer desvio quanto ao carregamento dos processos do S.O..

Quanto ao formato de marcação do tempo, será usada a tipografia (Minutos'Segundos'Milésimos) correspondente a (00'00"00).

O primeiro teste refere-se à inicialização do S.O., sendo dividida em 2 (duas) partes: a primeira seria após a inicialização do *Grub*, marcando como início do cronometro, finalizando quando carregar a tela de *Login*, onde seria escolhido o usuário para efetuar *Login*; na segunda parte, inicia-se no *Login* do usuário, finalizando o cronometro quando todos os ícones e barras de ferramentas são carregadas.

Os tempos dos testes dos aplicativos estão na *Tabela 1* a seguir, todos com os 3 três tempos e sua média. O primeiro teste corresponde a do *Grub* até a tela de *Login*, obtendo um menor tempo no 3º (terceiro) teste, tendo uma média de cinquenta e cinco segundos e dezessete milésimos de segundos (00'55"17). No teste do *Login* o menor tempo corresponde ao 3º (terceiro) teste, com uma média de seis segundos e cinquenta e cinco milésimos de segundos (00'06"55).

Foram efetuados testes para verificar quanto tempo gasta para iniciar os aplicativos de escritório: *Writer*, *Calc* e *Impress*. Os aplicativos serão acionados 3 (três) vezes consecutivas para obtenção da média do seu tempo. Como já citado anteriormente, a primeira vez que é carregado o aplicativo na memória RAM, a segunda vez tende a ser mais rápido.

Para o aplicativo *Writer*, teve como menor tempo o 2º (segundo) teste e média de quarenta e nove segundos e quarenta milésimos de segundos (00'49"40). No aplicativo *Calc* o menor tempo gasto foi o 1º (primeiro) teste, com média de quarenta e nove segundos e vinte milésimos de segundos (00'49"20). Já para o aplicativo *Impress*, o menor tempo foi o 3º (terceiro) teste, tendo uma média de um minuto, trinta segundos e vinte milésimos de segundos (01'30"20).

**Tabela 1** – Tempo dos aplicativos no Sistema Operacional Ubuntu 10.04 do Fat Client.

Aplicativo	1º teste	2º teste	3º teste	Média
Grub	01'08"13	00'49"13	00'48"25	00'55"17
Login	00'06"43	00'07"30	00'06"34	00'06"55
Writer	00'55"00	00'44"00	00'50"00	00'49"40
Calc	01'46"00	01'26"00	01'19"00	01'30"20

**Fonte:** Pesquisador do estudo (2013).

### 6.3 CONFIGURAÇÃO DO SERVIDOR



**Figura 5** – Servidor LTSP do Laboratório de Informática.

**Fonte:** Pesquisador do estudo (2013).

O Servidor utilizado para o LTSP é um computador *desktop* modelo UDP ZMAX MERCHANT 3500, processador Intel(R) Core (TM) 2 Duo E7400, com frequência de 2.80 GHz, com 2 (duas) memória RAM de 2.048 GB, barramento DDR2 de 800 MHz, com HD (Hard Disk) de 512 GB, placa *Onboard* de rede 10/100 MB e um Leitor e gravador de CD/DVD.

### 6.4 DESEMPENHO DO SERVIDOR

Os testes realizados no servidor correspondem a mesma forma do *Fat Client*, descrito anteriormente, estando ativados apenas 1 (um) usuário, o administrador.

Nos testes de inicialização temos a *Tabela 2* com o tempo dos 3 (três) testes e sua correspondente média. Iniciamos com o tempo da inicialização do *Grub*, onde o menor tempo obtido foi do 1º (primeiro) teste com média de dezoito segundo e vinte nove milésimos de segundos (00'18"29). No *Login* o 3º (terceiro) teste foi o de menor tempo, a média obtida é de quatro segundos e cinquenta milésimos de segundos (00'04"50).

Para os aplicativos de escritório o *Writer* obteve o maior tempo para carregar no 1º (primeiro) teste, tendo a média de cinco segundos e quarenta e quatro milésimos de segundos (00'05"44). Para o *Calc*, o menor tempo foi o 2º (segundo) teste com média de dois segundos e três milésimos de segundos (00'02"03). E por último o *Impress*, que teve o menor tempo no 3º (terceiro) teste obteve média de dois segundos, vinte e três milésimos de segundos (00'02"23).

**Tabela 2 – Tempo dos aplicativos no Sistema Operacional Ubuntu 10.04 do Servidor.**

<b>Aplicação</b>	<b>1º teste</b>	<b>2º teste</b>	<b>3º teste</b>	<b>Média</b>
<b>Grub</b>	00'18"19	00'18"31	00'18"38	00'18"29
<b>Login</b>	00'05"30	00'04"39	00'04"20	00'04"50
<b>Writer</b>	00'13"34	00'01"42	00'01"58	00'05"44
<b>Calc</b>	00'02"34	00'01"45	00'01"51	00'02"03
<b>Impress</b>	00'02"39	00'02"24	00'02"08	00'02"23

**Fonte:** Pesquisador do estudo (2013).

## 6.5 DESEMPENHO DO TERMINAL CLIENTE

Os testes realizados com o Terminal Cliente terá o mesmo método usado no *Fat Client* e no Servidor descrito anteriormente. Diferenciando quanto ao teste do *Grub* e do *Login*, onde os Terminais Clientes são autenticados automaticamente. Desta forma, serão efetuada a soma dos tempos do *Grub* mais o *Login* ( $Grub + Login = Inicialização$ ) para comparação da inicialização do Terminal Cliente.

No teste de inicialização temos a *Tabela 3* que mostra o 1º (primeiro) teste como o mais rápido, pois apresenta média de vinte e sete segundo e trinta e cinco milésimos de segundos (00'27"35) para iniciar o Terminal Cliente.

Ainda na *Tabela 3*, temos os aplicativos de escritório *Writer* que obteve um tempo mais longo o 1º (primeiro) teste e a média de sete segundos e vinte e sete milésimos de segundos (00'07"27). Para o aplicativo *Calc* o de menor tempo sendo o 3º (terceiro) teste e sua

respectiva média de cinco segundos e dezesseis milésimos de segundos (00'05"16). E para o *Impress* o menor tempo marcado foi do 2º (segundo) teste obtendo a média de sete segundos e cinquenta e dois milésimos de segundos (00'07"52).

**Tabela 3** – Tempo dos aplicativos do Servidor LTSP com uso do Fat Client.

Aplicação	1º teste	2º teste	3º teste	Média
<b>Inicialização</b>	00'25"36	00'29"45	00'27"24	00'27"35
<b>Writer</b>	00'14"16	00'05"32	00'02"33	00'07"27
<b>Calc</b>	00'09"23	00'03"35	00'02"50"	00'05"16
<b>Impress</b>	00'14"09	00'04"21	00'05"08	00'07"52

*Fonte: Pesquisador do estudo (2013).*

## 6.6 LTSP NO LABORATÓRIO DA UEPB

Na UEPB, campus VII havia 15 (quinze) computadores os quais estavam em desuso. Sendo que no período 2011.2 a instituição, possuía 01 (um) Laboratório com toda estrutura física, faltando apenas os computadores novos para funcionar. Desta forma, os Técnicos de Informática *Giancarlos da Silva Almeida*, *Vinícius Reuteman Feitoza Alves de Andrade* junto com o coordenador do curso de Licenciatura em Computação Professor *José Wilker de Lima*



**Figura 6** – Laboratório de Informática com LTSP.

*Fonte: Pesquisador do estudo (2011).*



*Silva* buscaram utilizar os recursos do *LTSP* para uso do TI Verde. Aproveitando assim os 15 (quinze) *Computadores Antigos (Fat Client)* os quais se encontravam parados no almoxarifado da instituição, conforme *Figura 6*.



**Figura 7** – Switch de 16 portas, usando no Laboratório de Informática.  
**Fonte:** Pesquisador do estudo (2011).

Sendo efetuados a configuração do servidor *LTSP* no *Centro de Processamento de Dados (CPD)* da *UEPB*, tendo a participação de *Victoremberg Feitoza Alves de Andrade* como *TI* que auxiliou na configuração do servidor. Após configuração foi efetuado a triagem e limpeza dos computadores, averiguando quais encontravam em bom estado.

O Laboratório contava com 15 (quinze) *Fat Client* (configuração descrita anteriormente), conexão por par trançado *B100*, interligado no laboratório por um switch de 16 portas, como mostra *Figura 7*. Tendo o servidor localizado no *CPD*, com uma distância de 60 (sessenta) metros dos *Fat Clients*. Para composição do servido, foi usado um dos computadores do Laboratório I, sendo acrescentado de 1 (uma) memória *RAM* de 2 GB, conforme descrito no capítulo anterior.

Estavam funcionando 13 (treze) Terminais Clientes, devido problemas com teclado e fonte de energia, os quais não tinha para reposição dos mesmos.



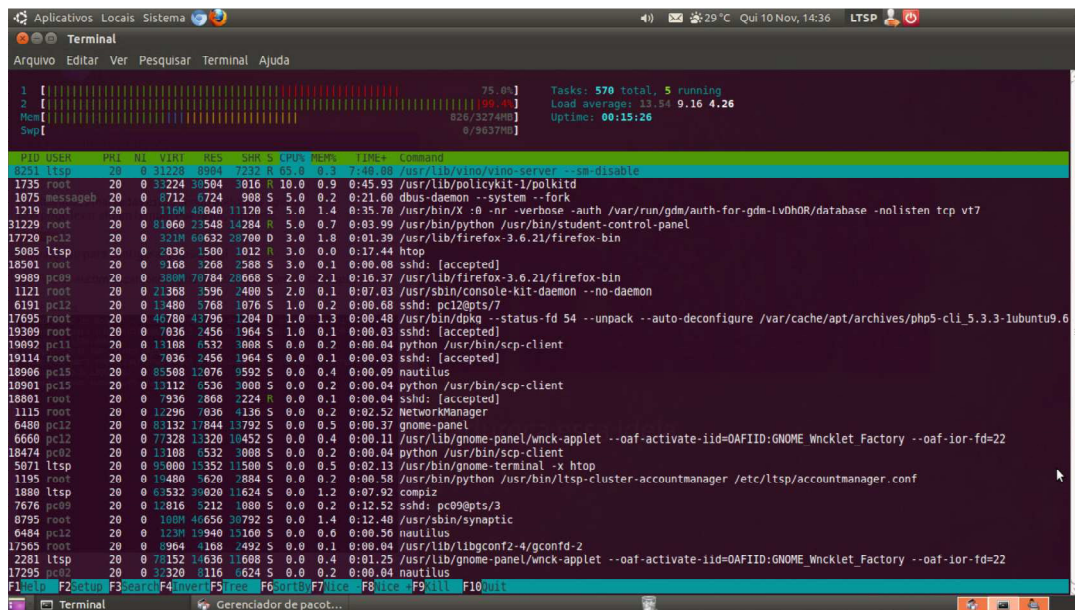
### 6.6.1 DESEMPENHO DO LTSP NO LABORATÓRIO DA UEPB

O uso do *LTSP* no Laboratório II foi liberado para os alunos terem acesso à internet e usarem para uso acadêmico, aulas e Minicursos (ocorrido no II ELIC, em 2011). Os aplicativos disponíveis eram os mesmos contidos no Laboratório I.

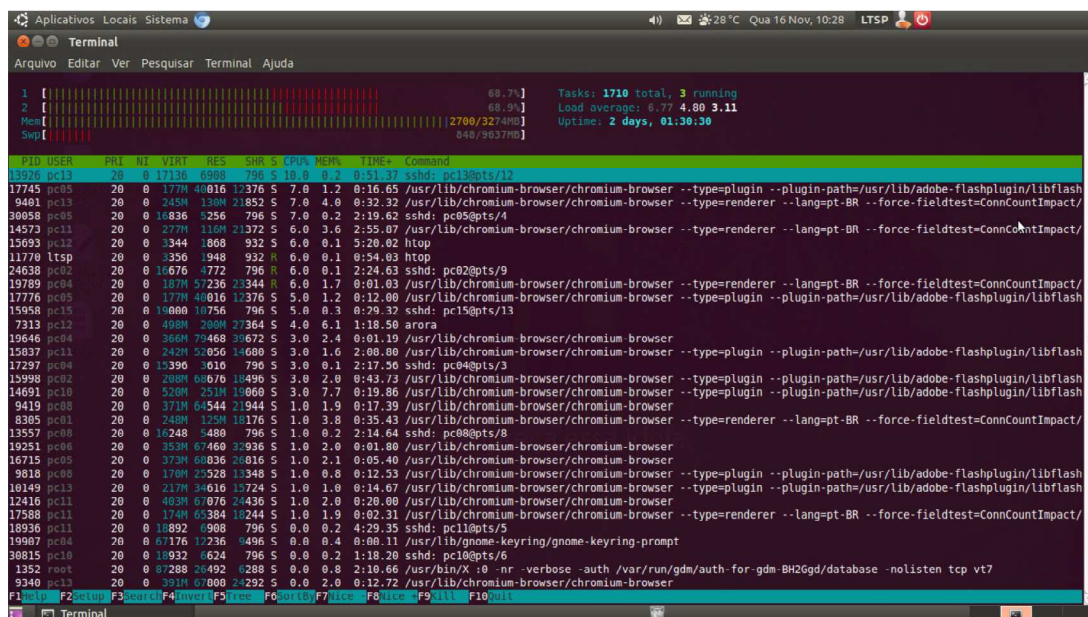


**Figura 8** – Inicialização do *fat client* usando *LTSP*.  
**Fonte:** Pesquisador do estudo (2011).

Os períodos de maior pico de processo do processador no servidor foram identificados quando estão iniciando os *Fat Clients* (*Figura 9*), onde a carga de processo e envio de requisitos estão elevados. Como também, no uso de alguns aplicativos, que estavam sendo usados em quase todos os Terminais Clientes.

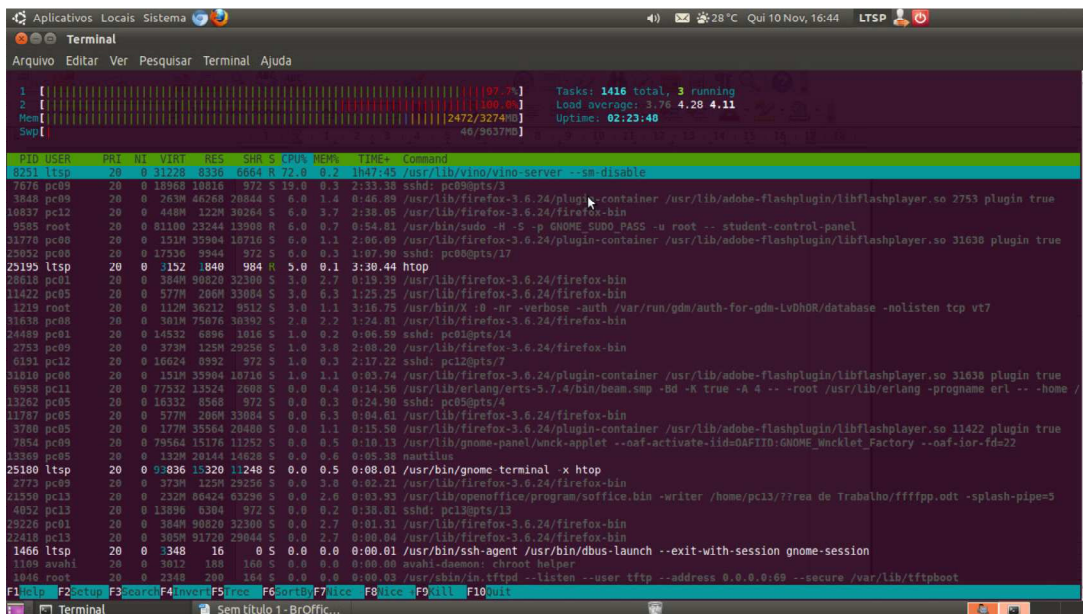


**Figura 9** – Utilização dos processadores na inicialização dos fat clients.  
**Fonte:** Pesquisador do estudo (2011).



**Figura 10** – Utilização da memória RAM e dos processadores no uso do aplicativo Chromium.  
**Fonte:** Pesquisador do estudo (2011).

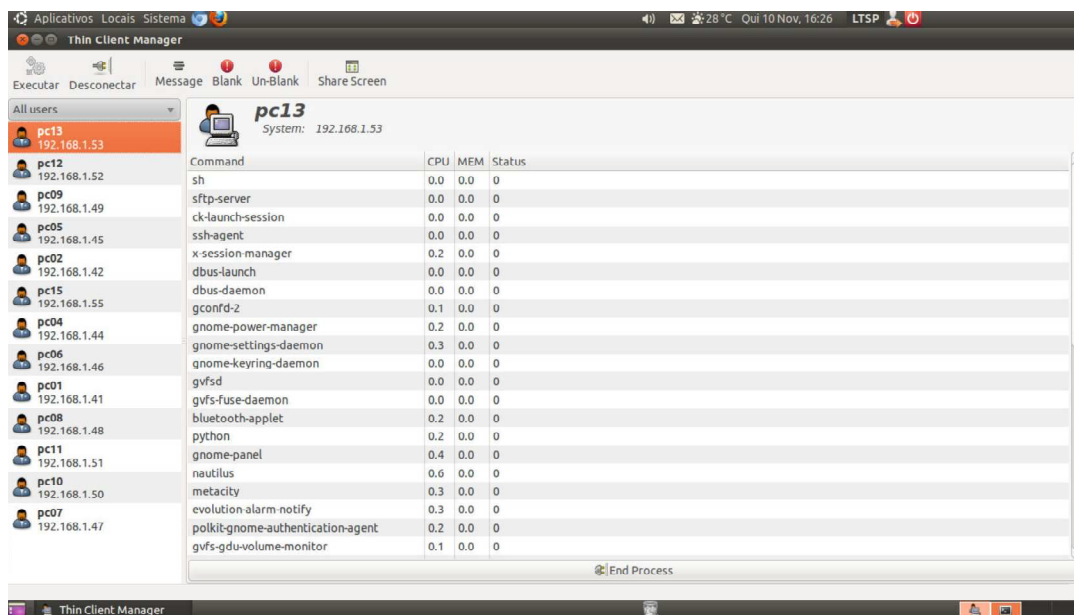
O aplicativo *Chromium* foi removido do servidor devido ao elevado requisito de processos pai ao processador (Figura 10). Causando assim atrasos nos outros Terminais Clientes, chegando até a trava dos mesmos quando os usuários usavam-no em quase todos os *Fat Clients* ao mesmo tempo.



**Figura 11** – Pico da memória RAM e dos processadores com uso do adobe flashplugin.  
**Fonte:** Pesquisador do estudo (2011).

Outras observações durante o uso do *LTSP* no Laboratório II, quanto ao uso do navegador para acessar sites com feitos/animações/vídeos utilizando o *adobe flash plugin*. Chegando ao uso de 100% (cem por cento) dos dois processadores e quase a totalidade da memória RAM, como mostra na (Figura 11).

A Figura 12 mostra o aplicativo *Thin Client Manager* pode ser observado os 13 (treze) Terminais Clientes funcionando no Laboratório II.

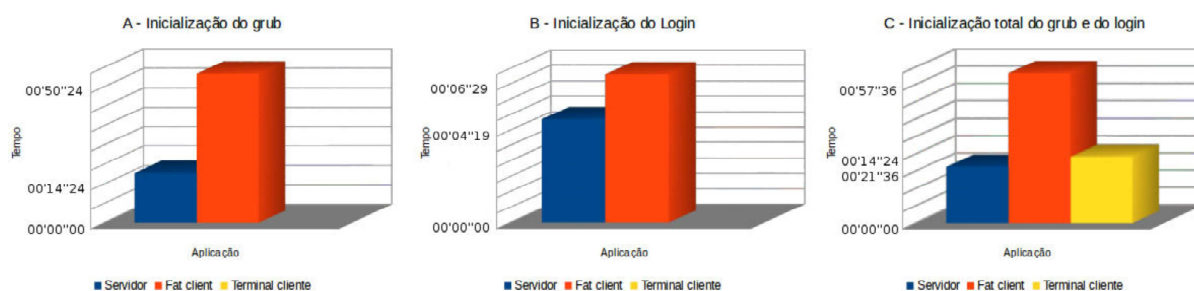


**Figura 12** – Aplicativo de gerenciamento dos fat clients.  
**Fonte:** Pesquisador do estudo (2011).

## 7 ANÁLISE DOS DADOS

Existem diversas formas para efetuar métrica uso comparativo de desempenho, sendo escolhido o tempo para inicialização do S.O. e dos aplicativos. Ou seja, o comparativo do mesmo aplicativo executado nos 3 (três) ambientes distintos Servidor, *Fat Client* e no Terminal Cliente. O primeiro corresponde o Servidor que é caracterizado por ser mais robusto, conforme descrito anteriormente. Já o segundo, o *Fat Client*, corresponde ao computador antigo que se encontrava em desuso. E por último, o Terminal Cliente que usa o Servidor.

O *Gráfico 3* nos dá o resultado de qual aplicação obtém o menor tempo para sua inicialização. Desta forma pode ser observado que o Servidor obteve o menor tempo para inicializar o S.O. em todos os testes, o tempo de inicialização do Terminal Cliente obteve resultado quase igual ao do Servidor, como mostrado no *Gráfico 3* item C. Também podemos observar que a inicialização do *Fat Client* gasta mais do triplo do tempo do que o Terminal Cliente.

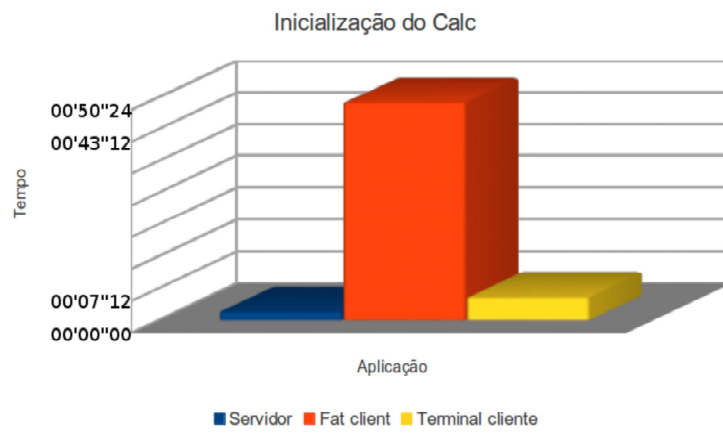


**Gráfico 3** – Tempo de inicialização do Grub e do Login.

**Fonte:** Pesquisador do estudo (2013).

Na inicialização do *Writer* o *Gráfico 4* mostra que o S.O. instalado no *Fat Client* demora mais para carregar do que usando o Terminal Cliente. A média do tempo gasto para iniciar o *Writer* no Servidor é quase igual o do Terminal Cliente.





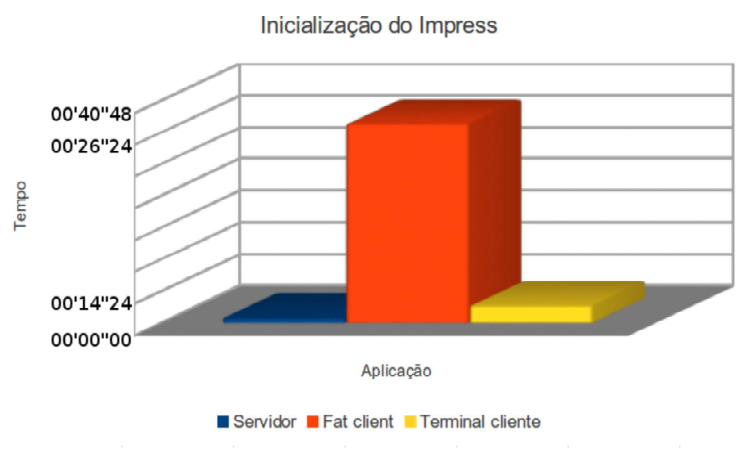
**Gráfico 4** – Tempo de inicialização do Calc.  
**Fonte:** Pesquisador do estudo (2013).

O aplicativo de escritório *Calc* demonstra não ser diferente do *Writer*, obtendo um menor tempo para inicializar no Servidor, em seguida no uso do Terminal Cliente e por último com quase 1 (um) minuto no *Fat Client*, conforme o *Gráfico 5*.



**Gráfico 5** – Tempo de inicialização do Writer.  
**Fonte:** Pesquisador do estudo (2013).

Para o aplicativo de escritório *Impress* obteve um menor tempo para inicializar no Servidor, em seguida no uso do Terminal Cliente e por último no *Fat Client*, conforme o *Gráfico 6*.



**Gráfico 6** – Tempo de inicialização do Impress.  
**Fonte:** Pesquisador do estudo (2013).

Pode-se observar que a qualidade do serviço e a quantidade de Terminais Clientes dependem dos seguintes itens de *hardware* do servidor: placa de rede, processador e memória RAM. O primeiro é responsável pelo tráfego de informações entre os terminais e o servidor. O processador por sua vez será responsável por todo processamento realizado pelo servidor e pelos terminais e a memória RAM é a responsável pela alocação dos programas, tanto no servidor quanto nos terminais. Poderíamos aplicar também o mesmo para os computadores *desktop*, que quanto mais memória, processador mais moderno, melhor seria a qualidade do serviço.

Nos gráficos descritos anteriormente, obtemos melhor desempenho ao carregar os aplicativos no Servidor, pois o mesmo conta com uma configuração mais robusta em comparação ao *Fat Client*. Ambos foram instalados com S.O. Ubuntu 10.04 LTS, contendo os mesmos aplicativos. Os testes foram realizados com duração máxima de 20 (vinte) minutos e efetuado 3 (três) testes consecutivos com os mesmos aplicativos.

Para instalação e teste no *Fat Client*, foi necessário acrescentar 256 MB de memória RAM, obtendo um total de 512 MB. Mesmo com acréscimo da memória RAM, o *Fat Client* demonstra desvantagem em comparação ao uso do Terminal Cliente. Os testes foram realizados rodando apenas 1 (um) Terminal cliente, sendo que quanto mais Terminal Cliente, mais processos serão requisitados.

Conforme visto, a inicialização do S.O. no Fat Client demora o dobro do tempo do Terminal Cliente, podendo ser desvantagem caso tenha muitos Terminais Clientes para inicializar. Neste caso, também dependerá da rede de comunicação, sendo preferencial a de

*1000 Mbps (Gigabit Ethernet)*. Já os aplicativos, o Terminal Cliente obteve um desempenho similar ao do Servidor. Comparando o Terminal Cliente ao *Fat Client*, por exemplo o *Writer* cheta a ser 7 (sete) vezes mais rápido para carregar no Terminal Cliente, do que no *Fat Client*. Mesmo tendo vários Terminais Clientes, ao usarem o aplicativo já executado anteriormente, o tempo para carregá-lo se torna quase o mesmo devido os processos serem compartilhando na memória *RAM* ou na *SWAP*.

## 8 CONCLUSÃO

Com o uso das TI apresentadas, pode-se criar laboratórios de pesquisas com uso dos computadores obsoletos sendo “reciclados” de forma a atender à demanda crescente pela democratização do acesso às tecnologias de informação, através do uso da tecnologia de Terminais Clientes.

LTSP é uma tecnologia em que os computadores de uma rede são inicializados por meio de um servidor. Os computadores desta rede baseados na tecnologia *LTSP* não necessitam possuir disco rígido (HD), CD-ROM, ou quaisquer dispositivos que possam ser usados para carregar um sistema operacional. O servidor fornece aos Terminais Clientes, as telas dos aplicativos nele instalado, possibilitando ao usuário acesso aos aplicativos gráficos como navegador na Internet (*browser*), *e-mail*, editores de texto entre outros aplicativos gráficos.

Desta forma, é possível maximizar a infraestrutura existente, utilizando *Fat Clients* como Terminais Clientes ou, ainda, possibilitando a compra de *Thin Clients* a custos mais acessíveis, conseguindo assim menor consumo de energia e mão de obra em manutenção mais barata. O servidor e os Terminais Clientes possibilitam uma administração centralizada de informações, como o *backup*. A manutenção de aplicativos também é facilitada, já que alterações no servidor são repercutidas em todas as máquinas. Basta atualizar o servidor de terminais para que todas as máquinas tenham acesso imediato à nova versão do aplicativo ou a um novo aplicativo. O *LTSP* diminui o custo relacionado com a ativação de novos computadores, aquisição de *softwares* proprietários, bem como a manutenção da rede. Portanto, a Tecnologia do *LTSP* oferece uma solução profissional para Terminais Clientes, utilizando o S.O. *Linux* e aplicativos de produtividade gratuitos. Isto representa uma economia considerável com a infraestrutura de recursos tecnológicos.

O *LTSP* pode ser usado ainda como Terminais Clientes nas salas de aulas, junto aos *Data Shows*. Não tendo a necessidade da compra de novos computadores para este fim.

O uso de *LTSP* se torna favorável para as instituições governamentais. Não sendo necessário o pagamento de Licença dos aplicativos. Logo, o dinheiro poderia ser investido em expansão em novas máquinas, ou em *Thin Clients* e em treinamentos para os usuários.



## 9 REFERÊNCIAS

BALNEAVES, Scott, et al. **Linux Terminal Server Project Administrator's Reference: A Guide to LTSP Networks**. 2009. Disponível em:

<<http://ufpr.dl.sourceforge.net/project/ltsp/Docs-Admin-Guide/LTSPManual.pdf>>

Acesso em: 10 jun. 2014.

CARMONA, Tadeu. **Universidade LINUX**. São Paulo: Digerati Books, 2005. ISBN 8589535-93-2.

CERVO, Amado L; BERVIAN, Pedro A. **Metodologia Científica**. 5ª Edição, São Paulo: ed. Pearson Prentice Hall, 2002.

CONAMA. **Conselho Nacional do Meio Ambiente. Resolução nº 257**. 1999. Disponível em: <<http://www.mma.gov.br/port/conama/legiabre.cfm?codlegi=257>> Acesso em: 20 jun. 2014.

CROFFI, Flávio. **Brasil terá 98 milhões de computadores em domicílios até 2012**. 2011. Disponível em: < <http://www.techguru.com.br/brasil-tera-98-milhoes-de-computadores-em-domicilios-ate-2012/>> Acesso em: 17 jun. 2013.

DAVID, Barrie. **Thin Client Benefits**. Newburn Consulting. 2002.. Disponível em: <[http://www.thinclient.net/pdf/Thin\\_Client\\_Benefits\\_Paper.pdf](http://www.thinclient.net/pdf/Thin_Client_Benefits_Paper.pdf)> Acesso em: 11 maio 2014.

DAVIS, Euan. et all. **Put Thin-Client Computing Back On The Desktop Hardware Agenda**. Forrester, 2008. Disponível em:

<<http://www.forrester.com/Green+Benefits+Put+ThinClient+Computing+Back+On+The+Desktop+Hardware+Agenda/fulltext/-/E-RES43638>> Acesso em: 04 ago. 2013.

DERI, Luca. Carbone, Rocco. **NTop: Configurações gerais**. 2008. Disponível em: <<http://www.vivaolinux.com.br/artigo/NTop-Configuracoes-gerais>>. Acessada em: 09 nov. 2013.

ENVIRONMENT, Agency. **Waste electrical and electronic equipment (WEEE)**. Disponível em: <<http://www.environment-agency.gov.uk/business/topics/waste/32084.aspx>> Acesso em: 17 jun. 2013.

ESTATUTO. **Universidade Estadual da Paraíba Estatuto**. Diário Oficial, 2007. Disponível em: <[http://transparencia.uepb.edu.br/wp-content/files/Estatuto\\_Dirio\\_Oficial.\\_03\\_a\\_09.pdf](http://transparencia.uepb.edu.br/wp-content/files/Estatuto_Dirio_Oficial._03_a_09.pdf)> Acesso em: 17 jun. 2013.

GARCIA, Luís Fernando Fortes. **RPC: Remote Procedure Call. Disponível em:** <<http://penta2.ufrgs.br/rc952/trab1/rpc.html>> Acesso em: 20 out. 2013.

GNU.ORG. **O que é o software livre?**. 2012. Disponível em: <<http://www.gnu.org/philosophy/free-sw.html>> Acesso em: 20 jun. 2014.

KUROSE, James F. **Redes de computadores e a Internet**: uma abordagem top-down. 3. ed. - São Paulo: Pearson Addison Wesley, 2006.

MCQUILLAN, James. **Rolling out your own Linux thin clients**, 2000. Disponível em: <[http://www.developer.com/tech/article.php/10923\\_628661\\_2/Rolling-out-your-own-Linux-thin-clients.htm](http://www.developer.com/tech/article.php/10923_628661_2/Rolling-out-your-own-Linux-thin-clients.htm) > Acesso em: 03 jun. 2014.

MONITOR, Manual Sistema. **System Monitor Manual**. V2.2. 2011.

MORIMOTO, Carlos E. **Terminais leves como LTSP**. 2006. Disponível em: <<http://www.hardware.com.br/livros/linux-redes/capitulo-terminais-leves-com-ltsp.html>> Acesso em: 03 jun. 2014.

\_\_\_\_\_. **Redes e Servidores Linux**: Guia Prático. Porto Alegre: Sul Editores, 2005. ISBN 85-205-0404-3

\_\_\_\_\_. **TFTP: Trivial File Transfer Protocol**. 2005. Disponível em: <<http://www.guiadohardware.net/termos/tftp>>. Acesso em: 09 nov. 2013

\_\_\_\_\_. **Redes e Servidores Linux**. 2 ed. Editora GDH Press e Sul Editores. 2006. Disponível em: <<http://www.hardware.com.br/livros/linux-redes/nfs.html>>. Acesso em: 09 nov. 2013.

\_\_\_\_\_. **XDMCP**. 2005. Disponível em: <<http://www.hardware.com.br/termos/xdmcp>>. Acesso em: 15 nov. 2013.

\_\_\_\_\_. **Terminais leves como LTSP**. 2006. Disponível em: <<http://www.hardware.com.br/livros/linux-redes/capitulo-terminais-leves-com-ltsp.html>> Acesso em: 13 nov. 2013.

NETWORK, GNOME. **Ferramentas de rede GNOME**. Disponível em: <<http://projects.gnome.org/gnome-network/index.shtml>>. Acessada em: 15 nov. 2013.

OLIVEIRA, Silvio Luiz de. **Tratado de Metodologia Científica**: projetos de pesquisas, Tgi, tcc, monografias, dissertações e teses. São Paulo: Pioneira Thomson Learning, 2002.

PRADO, Mônica Igreja. **Eletrônicos: do lixo ao lucro**: a escassez de matéria prima para a contínua comercialização de produtos eletrônicos e o peso para a reciclagem pós-consumo. *Universitas Gestão e TI*, v. 2, n. 1, p. 27-33. 2012

SANCHEZ, A. P., et al. **TI Verde**. 2011. Disponível em: <[rabci.org/rabci/sites/default/files/TI%20verde\\_FINAL.pdf](http://rabci.org/rabci/sites/default/files/TI%20verde_FINAL.pdf)> Acesso em: 28 dez. 2013.

SANTOS, R. H. S. dos, et al. **Performance Tests with LTSP**. *IEEE Latin America Transactions*, Vol. 10, Nº. 1, Jan. de 2013.

SILVA, Jefferson Estanislaua. **História do GNU/Linux**: 1965 assim tudo começou! 2006. Disponível em: <<http://www.vivaolinux.com.br/artigo/Historia-do-GNU-Linux-1965-assim-tudo-comecou/>> Acesso em: 12 jan. 2014.

TANENBAUM, Andrew S. **Sistemas Operacionais Modernos**. 3ª Edição, São Paulo: Editora Pearson Prentice Hall, Brasil. 2009.

TAKAHASHI, Arthur Garcia. **TI Verde**: conceitos e práticas. 2009. Disponível em: <<http://www.hardware.com.br/artigos/ti-verde/>> Acesso em: 20 jun 2014.

TORRES, Gabriel. **Redes de Computadores Curso Completo**. Axcel Books, 2001.

UBUNTU-BR.ORG. **Filosofia**: Ubuntu. Disponível em: <<http://wiki.ubuntu-br.org/Filosofia>> Acesso em: 12 jan. 2014.<http://ubuntu-br.org/ubuntu>

UBUNTU-BR.ORG. **Ubuntu**. Disponível em: <<http://ubuntu-br.org/ubuntu>> Acesso em: 12 jan. 2014.

VERGARA, Sílvia C. **Projeto e relatório de pesquisa em Administração**. 2 ed. São Paulo: Atlas, 1998.