



UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ - UNIOESTE

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

Colegiado de Ciência da Computação

Curso de Bacharelado em Ciência da Computação

Implantação de Sistema Virtualizado para Manutenção de Laboratórios

Trabalho de Conclusão de Curso

Guilherme de Oliveira Felipe



Cascavel-PR

2021

UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ - UNIOESTE

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

Colegiado de Ciência da Computação

Curso de Bacharelado em Ciência da Computação

Guilherme de Oliveira Felipe

Implantação de Sistema Virtualizado para Manutenção de Laboratórios

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel.

Orientador(a): Guilherme Galante

Cascavel-PR

2021

GUILHERME DE OLIVEIRA FELIPE

**IMPLANTAÇÃO DE SISTEMA VIRTUALIZADO PARA MANUTENÇÃO
DE LABORATÓRIOS**

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos professores:

Prof. Guilherme Galante (Orientador)
Colegiado de Ciência da Computação, UNIOESTE

Prof. Edmar André Bellorini
Colegiado de Ciência da Computação, UNIOESTE

Prof. Marcio Seiji Oyamada
Colegiado de Ciência da Computação, UNIOESTE

Cascavel, 26 de Julho de 2022

Agradecimentos

Em primeiro lugar a Deus, que fez com que meus objetivos fossem alcançados durante todos os meus anos de estudos. Aos amigos, familiares e professores, por todo o apoio e contribuição para a realização deste trabalho. Ao professor Guilherme Galante, por ter sido meu orientador e ter desempenhado tal função com dedicação e amizade. À Universidade Estadual do Oeste do Paraná, essencial no meu processo de formação profissional, e por tudo o que aprendi ao longo dos anos do curso.

Resumo

Nos dias de hoje, é esperado que universidades e faculdades ofereçam recursos computacionais adequados para seus estudantes pois os laboratórios de informática são uma parte essencial das universidades que oferecem a tecnologia para ensino e é a principal maneira pelas quais a tecnologia é colocada na mão dos alunos. No entanto, os laboratórios em geral são consideravelmente caros para as instituições de ensino, e suas operações de manutenção e administração são complexas. Diante disso, a virtualização tornou-se uma das mais citadas tecnologias para solucionar alguns desses problemas. Outro fator que dificulta mais o dia a dia do laboratorista, alunos e professores é a falta de monitoramento do ambiente e a tarefa manual de identificar se o hardware está estável. Para solucionar estes problemas, foi implantado nos laboratórios do Curso de Ciência da Computação da Universidade Estadual do Oeste do Paraná do campus de Cascavel o uso da virtualização, *scripts* e outras ferramentas para reduzir o tempo de trabalho do laboratorista com o gerenciamento, manutenção e configuração dos ambientes. O método anterior de formatação utilizado pelo laboratorista consistia no uso da clonagem de HD o qual leva aproximadamente 1 hora por máquina. O método utilizado neste trabalho consiste em criar uma Live personalizada do Ubuntu 20.04 LTS, que por sua vez leva em torno de 15 minutos para realizar a formatação acrescido de 1 hora para sincronização das VMs. Ao replicar o mesmo procedimento em 20 máquinas o tempo reduzido chega a 70% pois o tempo de formatação entre máquinas leva apenas 15 minutos.

Palavras-chave: Tecnologia no ensino, virtualização nas Universidades, gerenciamento e manutenção de laboratórios.

Lista de figuras

Figura 1 – Tipos de Hypervisor.	16
Figura 2 – Tela de opções do Systemback.	28
Figura 3 – Configuração dos ambientes.	29
Figura 4 – Atualização das imagens.	30
Figura 5 – Monitoramento com o Nagios.	37
Figura 6 – Terminal do Linux.	45
Figura 7 – Pesquisando pelo VirtualBox instalado.	48
Figura 8 – Tela de boas vindas do VirtualBox.	48
Figura 9 – Tela de opções do Systemback.	54
Figura 10 – Tela de Criar Sistema Live.	55
Figura 11 – Criando Sistema Live.	55
Figura 12 – Fim da criação do Sistema Live.	55
Figura 13 – Criando flash USB.	56

Lista de códigos

Código 1 – <i>Script</i> rsyncPush.sh primeira parte.	32
Código 2 – <i>Script</i> rsyncPush.sh segunda parte.	33
Código 3 – <i>Script</i> rsyncPull.sh primeira parte.	34
Código 4 – <i>Script</i> rsyncPull.sh segunda parte.	35
Código 5 – Exemplo de <i>host</i> do Nagios	53

Lista de abreviaturas e siglas

BCC	Bacharelado em Ciência da Computação
VDI	<i>Virtual Disk Image</i>
VM	<i>Virtual Machine</i>
VMM	<i>Virtual Machine Monitor</i>
IaaS	<i>Infrastructure as a Service</i>
LTS	<i>Long term support</i>
LCAD	Laboratório de Computação de Alto Desempenho
NRPE	<i>Nagios Remote Plugin Executor</i>
OVA	<i>Open Virtual Appliance</i>
PXE	<i>Preboot Execution Environment</i>
SSH	<i>Secure Shell</i>
SO	Sistema Operacional
TI	Tecnologia da Informação
Unioeste	Universidade Estadual do Oeste do Paraná
WSL	<i>Windows Subsystem for Linux</i>

Sumário

1	Introdução	10
1.1	Motivação	12
2	Conceitos Gerais Sobre a Virtualização	14
2.1	Virtualização	14
2.1.1	Caracterizando a Virtualização	15
2.1.2	Virtualização nos Laboratórios de Informática	17
2.1.3	A Virtualização de Desktop	18
2.2	Oportunidades e benefício das tecnologias virtuais	19
2.2.1	Máquinas Virtuais mais Populares em 2021/2022	20
2.3	Arquivos de imagem de disco do VirtualBox	22
2.3.1	Exportar e Importar Máquinas Virtuais	23
2.4	Considerações finais	23
3	O caso do curso de Ciência da Computação, Unioeste - Cascavel	24
3.1	O cenário dos laboratórios	24
3.1.1	Dificuldades Encontradas	25
3.1.2	Requisitos desejáveis para melhorar o ambiente	25
3.2	Determinando um Sistema Operacional com segurança, eficiência e desempenho	26
3.3	Software de virtualização e suas VMs	26
3.4	Centralização das VMs	27
3.5	Disponibilização das VMs	27
3.6	Monitoramento de aplicação: Nagios	27
3.7	Ferramenta de backup: Systemback	28
3.8	Configuração e Atualização	28
3.9	Considerações finais	30
4	Experimentação	31
4.1	Configuração e disponibilização das VMs	31
4.1.1	Atualização das imagens no servidor	32
4.1.2	Atualização das imagens nos Laboratórios	34
4.2	Monitoramento com Nagios	36
4.3	Criação da Live e formatação dos computadores	37
4.4	Considerações finais	37
5	Conclusão	39

Referências	41
-------------------	----

Apêndices	44
------------------	-----------

APÊNDICE A Configurando o ambiente	45
A.1 Rsync e OpenSSH	45
A.1.1 Criando as Chaves Pública-Privada	46
A.2 Requisitos de hardware para o VirtualBox	46
A.2.1 Instalando o VirtualBox	47
A.3 Requisitos de hardware para o Nagios	49
A.3.1 Instalando o Nagios	49
A.3.2 Instalando o Nagios Plugins	51
A.3.3 Verifique a Configuração do Nagios	51
A.3.4 Acesse a interface da Web do Nagios	52
A.3.5 Adicione <i>hosts</i> remotos	52
A.4 Instalando o Systemback	53
A.4.1 Como utilizar o Systemback	54
A.5 Configuração da BIOS RTC	57

1

Introdução

O uso dos computadores nas universidades iniciou-se na metade do século XX. Durante os estágios iniciais, os computadores eram grandes, caros e possuíam disponibilidade limitada. Para os campi que tinham computadores, seus recursos eram normalmente dedicados a apurar números derivados de complexas pesquisas matemáticas e científica. Nos dias de hoje, é esperado que universidades e faculdades ofereçam recursos computacionais adequados para seus estudantes pois os laboratórios de informática são uma parte essencial das universidades que oferecem a tecnologia para ensino e é a principal maneira pelas quais a tecnologia é colocada na mão dos alunos. Apesar da importância dos laboratórios de informática, poucas pessoas fora do gerenciamento do laboratório dedicam seu tempo tentando entender realmente os laboratórios e seus usuários ([HOWELL, 2007](#)).

Os laboratórios das universidades são amplamente utilizados por estudantes e professores porém professores e laboratoristas enfrentam vários desafios com as tecnologias atuais, por exemplo: em caso de falha da máquina, é necessário configurar uma nova máquina que levará a um aumento de tempo e custo; antes de cada avaliação, as máquinas de laboratório precisam ser reinstaladas, o que é uma tarefa muito tediosa e demorada; os laboratórios são configurados de acordo com os requisitos da matéria e não existe laboratório separado por disciplinas individuais; toda máquina é instalada com várias aplicações pesadas (aplicações que demandam muitos recursos como de processamento e memória), todo esse processo é uma preocupação para assistência em laboratório. Portanto, o gerenciamento de um número elevado de computadores pode ser uma atividade cara e propensa a erros. Ademais, o laboratório tradicional de computadores físicos exige muitos recursos limitados, vincula um espaço valioso e limita a maneira como os alunos optam por trabalhar ([THAKRAL, 2014](#); [BURD](#); [SEAZZU](#); [CONWAY, 2009](#)).

Os laboratórios de informática da Universidade Estadual do Oeste do Paraná (Unioeste) do campus de Cascavel fornecem muitos benefícios ao estudante inclusive evitando-o da compra de hardware e software e também da administração e manutenção. No entanto, os laboratórios em

geral são consideravelmente caros para as instituições de ensino e suas operações de manutenção e administração são complexas (HAWKINS; OBLINGER, 2007). De acordo com Caruso (2007), Oberdick (2006) os notebooks tornaram-se comuns entre os estudantes, mas mesmo com seu aumento significativo não retirou da administração do campus a necessidade de possuir laboratórios de informática.

Diante disso, segundo Kizza (2012), a virtualização tornou-se uma das mais citadas tecnologias recentemente para solucionar alguns dos problemas mencionados nos parágrafos anteriores. A virtualização é um processo no qual o software carrega máquinas virtuais, incluindo o monitor da máquina virtual chamado de *hypervisor*, que aloca recursos de hardware de forma dinâmica e transparente para que vários sistemas operacionais, chamados de “sistemas operacionais convidados”, possam ser executados simultaneamente em um único computador físico. Pode ser visto como abstração e criação de vários sistemas lógicos em uma única plataforma física. É chamado de virtualização porque cria uma versão virtual de recursos de computação em vez de reais. A virtualização permite a maximização do hardware por meio do compartilhamento de recursos.

A utilização de tecnologias de virtualização para conduzir exercícios em aulas práticas em cursos associados a Ciência da Computação não é novidade. (STOCKMAN, 2003) descreveu um dos mais simples exemplos da utilização de máquina virtual (do inglês VM), para executar no laboratório exercícios de rede de computadores; Collins (2006) apresentou as vantagens do uso da virtualização nas disciplinas para exercícios de segurança e de sistema operacional (SO), e sugeriu que laboratoristas e professores considerassem o uso da virtualização no ensino; Stackpole et al. (2008) realizou uma boa revisão a respeito dos problemas enfrentados na utilização do sistema operacional real para exercícios de laboratório e descreveu as vantagens da utilização de um laboratório utilizando ambientes de virtualização para aumentar a produtividade e utilização do laboratório pelos acadêmicos. De acordo com Bulbrook (2006) e Bullers, Burd e Seazzu (2006) essas são as vantagens da utilização de tecnologia de virtualização para os estudantes realizarem os exercícios nos quais são relevantes para nosso cenário: (i) facilitar a configuração e implantação, a qual consiste basicamente em criar uma cópia do arquivo da VM, (ii) prevenir conflitos de configuração entre diferentes encontros nos laboratórios pois cada VM pode ser preparada especialmente para um exercício específico, (iii) possibilitar conceder os privilégios de administrador para o aluno sem comprometer a segurança, (iv) possibilitar a padronização dos laboratórios utilizados para as várias disciplinas em vez da utilização de um único exclusivamente, e (v) a possibilidade de deixar os acadêmicos usufruírem do ambiente sem correr “perigos” como por exemplo uma instalação ou configuração mal sucedida durante as atividades.

1.1 Motivação

As instituições de ensino superior encontram-se muitas vezes com problemas em providenciar um serviço de bom desempenho e soluções de baixo custo para seus laboratórios de informática. De acordo com Nunes (2015), não é uma tarefa simples para as instituições de ensino conseguirem uma infraestrutura que busque oferecer um alto nível de segurança, escalabilidade e desempenho, visando um custo benefício para a instituição. Tendo em vista que a instituição terá de capacitar sua equipe de suporte, manter máquinas e licença de softwares atualizados, aplicando manutenção e substituição de cada máquina quando necessário. Segundo Maia (2017) e Falvo (2014), é função da equipe de suporte dos laboratórios fazer uma manutenção preventiva de todas as máquinas que se encontram nos laboratórios de informática da instituição, tanto no hardware para evitar falha dos componentes quanto nos softwares que os professores e alunos farão uso ao ministrarem suas aulas nos laboratórios.

Tradicionalmente, professores distribuem o software requisitado para executar os exercícios de aula prática, ou ainda os alunos são instruídos para carregarem um website e realizar o download. Assim, os alunos podem instalar e utilizar as ferramentas necessárias para a aula em seus computadores pessoais. Opcionalmente, para exercícios que requerem instalações e configurações mais complexas, professores podem preparar uma aplicação virtual contendo sistema operacional e as aplicações relevantes para distribuí-las. Os alunos então podem executar a máquina virtual nos seus computadores pessoais e completar os exercícios através da VM. As duas abordagens possuem suas desvantagens. A distribuição dos instaladores requer que os acadêmicos utilizem tempo útil com a instalação, e ainda podem acarretar conflitos com os softwares pré-existentes na máquina do aluno. A solução de problema de instalações corrompidas, conflito de software, problemas de firewall e configuração do SO demanda muito tempo e distrai o acadêmico do objetivo principal do exercício. Esta abordagem não é viável para instaladores que requerem sistemas operacionais específicos não comumente utilizado pelo acadêmico. A segunda abordagem é a distribuição de softwares preparados pela virtualização, o qual elimina a maioria dos problemas citados anteriormente, mas requer suficiente espaço em disco e maior quantidade de memória RAM (MOK; LEE; TAN, 2012). Como vimos, muitos problemas ocorrem em laboratórios de informática devido à configuração tradicional. Assim, com a ajuda da virtualização, podemos reduzir os desafios acima mencionados.

As dificuldades enfrentadas motivaram os esforços aqui apresentados de encontrar uma tecnologia de virtualização que atenda os requisitos visando a usabilidade e ainda facilitar o gerenciamento dos laboratórios. Para atender estes requisitos, foi realizado um estudo das diferentes tecnologias de virtualização para os laboratórios do curso de Ciência da Computação da Unioeste que permite a homogeneidade e disponibilidade dos laboratórios dentro dos requisitos de hardware e software disponíveis. O gerenciamento de atualização das máquinas virtuais foi obtido através de *scripts* que funcionam em conjunto com uma ferramenta de monitoramento e uma ferramenta de backup. Os resultados obtidos com esta solução foram satisfatórios uma

vez que facilitam e promovem um gerenciamento mais eficaz, além de reduzir o tempo para a manutenção e formatação das máquinas de 20 horas para 6 horas.

Para compreensão do tema, este trabalho será dividido em 5 Capítulos: o Capítulo 1 traz a presente introdução e é alusiva ao conteúdo deste estudo; o Capítulo 2 contempla a apresentação dos temas que servirão de base teórica para os assuntos-chave abordados no entendimento do problema; o Capítulo 3 aborda o caso dos Laboratórios de Informática do Curso de Bacharelado em Ciência da Computação da Universidade Estadual do Oeste do Paraná onde é explicado a infraestrutura de hardware disponível e também reúne as informações do projeto, como a elicitação de requisitos e detalhamento do desenvolvimento e a solução para alcançar os requisitos desejados para melhorar o ambiente; o Capítulo 4 se encarrega da análise do experimento; o Capítulo 5 apresentará a conclusão e trabalhos futuros.

2

Conceitos Gerais Sobre a Virtualização

O objetivo deste capítulo é apresentar um embasamento teórico dos assuntos e tecnologias explorados neste trabalho. Aqui é apresentado o resultado da consulta bibliográfica que auxiliou a entender os conceitos que proporcionaram melhor compreensão dos assuntos pesquisados.

2.1 Virtualização

A ideia de virtualização surgiu em meados de 1960, quando os grandes e caros computadores da época atingiram uma capacidade maior de processamento. No entanto, como o gerenciamento dos processos era feito manualmente pelo operador, ocorria desperdício de tempo e processamento no cálculo das operações. Percebeu-se que para otimizar as tarefas computacionais seria necessário efetuar diversos processos paralelamente, ou seja, a ideia de tempo compartilhado que culminou na virtualização (BOSING; KAUFMANN, 2012). A virtualização tem um potencial extremamente grande, conta com inúmeras vantagens que chega a constituir um novo campo da informática, permitindo a simulação de aplicativos, ferramentas e demais recursos. Facilita a transformação de ambientes físicos complexos em ambientes simplificados e fáceis de gerenciar (SABINO; CARVALHO, 2017; BOSING; KAUFMANN, 2012).

Segundo [Bandeira \(2017\)](#) a tarefa da virtualização é estender ou substituir um recurso com o intuito de reproduzir o comportamento do alvo virtualizado. Para isto, é usado uma camada de software como intermediária, responsável por transformar ações de um sistema A, virtualizado, em ações equivalentes a um sistema B, não virtualizado. Os princípios básicos que a virtualização visa atender são: (i) compatibilidade, seguindo a filosofia Java de “*write once, run anywhere*”; (ii) isolamento, que garante que algo que esteja executando em uma máquina virtual não interfira nem conheça as outras máquinas virtuais e, por último; (iii) encapsulamento, permitindo a qualquer momento, capturar o estado de uma máquina em sua execução e posteriormente retomar o seu estado anterior, conforme [Toscani, Carissimi e Oliveira \(2010\)](#).

Com o uso de uma infraestrutura virtualizada dentro de um laboratório de informática em uma instituição de ensino, tem-se uma série de vantagens e desvantagens. [Mattos \(2008\)](#) aponta algumas vantagens em se fazer uso da virtualização:

- **Segurança a falhas:** Fazendo uso de VMs e definindo uma máquina para serviço, sua vulnerabilidade ou uma eventual falha não prejudicará os outros serviços. Com o uso de VMs ainda pode-se definir qual o SO mais adequado em determinado ambiente para executar cada serviço, com requerimentos de segurança diferentes;
- **Disponibilidade:** A falha em uma VM não irá parar o fornecimento dos demais serviços oferecidos;
- **Custo:** Com um baixo custo de manutenção e consumo de energia reduzido, um menor custo geral pode ser alcançado;
- **Adaptação às diferentes cargas de trabalho:** Com o uso de ferramentas autônomas pode-se alocar recursos de uma VM para outra e assim tratar as diferentes cargas de trabalho;
- **Balanceamento de carga:** Como a VM está encapsulada pelo *Virtual Machine Monitor* (VMM) pode-se trocar a máquina virtual de plataforma com o intuito de aumentar seu desempenho;
- **Suporte a aplicações legadas:** Com a virtualização as aplicações processadas em hardware legadas deixam de ser processadas com alto custo de manutenção e sujeito a falhas, e passam a ter a possibilidade do seu processamento ser executado em hardwares mais novos com uma maior confiabilidade e um menor custo de manutenção.

Por outro lado [Mattos \(2008\)](#) também destaca algumas desvantagens do uso da virtualização:

- **Segurança do hospedeiro:** Se o SO hospedeiro tiver alguma vulnerabilidade todas as VMs que estiverem hospedada na máquina física também passam a estar vulneráveis;
- **Gerenciamento:** Maior investimento na implantação, pois ambientes virtuais precisam ser instalados, monitorados e configurados;

2.1.1 Caracterizando a Virtualização

Segundo [Rosenblum \(2004\)](#), [Toscani, Carissimi e Oliveira \(2010\)](#) o processo de virtualização pode se dar em diferentes locais dentro das camadas da arquitetura computacional. Os softwares de virtualização conseguem atuar em vários locais dessa arquitetura, criando assim três camadas de virtualização, sendo elas: a virtualização a nível de hardware, a virtualização a

nível de sistema operacional e por fim a virtualização a nível de linguagem de programação mas esta última não será abordada neste trabalho.

Na virtualização no nível de hardware, a camada responsável por virtualizar fica no topo do hardware, disponibilizando às camadas superiores um hardware similar ao original, portanto a máquina virtual é semelhante ao original (ROSENBLUM, 2004). As máquinas virtuais a nível de hardware são implementadas por uma camada de software chamada *Virtual Machine Monitor* ou *Hypervisor*. Nesse nível de virtualização denomina-se de Hypervisor (ou VMM) tipo I¹. O VMM tipo I executa sobre o hardware da máquina real, fazendo que a máquina virtual se pareça com o hardware. Esse nível de virtualização é bem difundido no mercado e exemplos de tecnologias que implementam são Xen, VMWare ESX e Hyper-V, de acordo com Toscani, Carissimi e Oliveira (2010).

A virtualização no nível de sistema operacional se caracteriza por ficar entre o sistema operacional, programas e aplicativos que são executados sobre ele. Sua função envolve a criação de partições lógicas sob o sistema operacional hospedeiro capaz de virtualizar sistemas diferentes do hospedeiro e é conhecido como VMM ou *hypervisor* do tipo II². Ferramentas conhecidas que implementam esse conceito de virtualização são o VirtualBox, Virtual PC, VMware Workstations, segundo Toscani, Carissimi e Oliveira (2010). A Figura 1 apresenta a comparação entre *hypervisor* tipo I e tipo II.

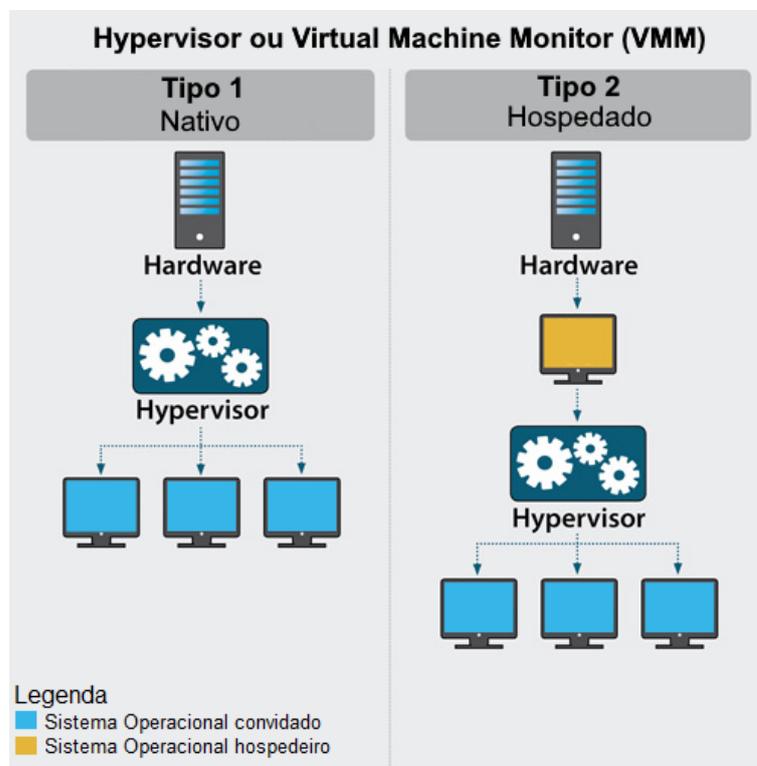


Figura 1 – Tipos de Hypervisor. Adaptado de Taylor (2020).

¹ VMM tipo I é também conhecido como nativo ou *native* ou ainda *bare metal* do inglês.

² VMM tipo II também é conhecido como hospedado ou *hosted* do inglês.

2.1.2 Virtualização nos Laboratórios de Informática

A virtualização pode ter um grande impacto na tecnologia educacional. Segundo [Mok, Lee e Tan \(2012\)](#), a virtualização para as faculdades ou universidades é uma maneira eficiente de fornecer dados e aplicativos aos alunos a qualquer hora, em qualquer lugar, sem exceder a equipe atual e os recursos financeiros. As soluções de virtualização otimizam o laboratório de informática e criam um laboratório virtual que atinge qualquer aluno, em qualquer lugar e a qualquer hora. O ambiente virtual permite que os laboratórios progridam de maneira segura e portátil ([THAKRAL, 2014](#)). Vários tipos de soluções de virtualização podem ajudar a gerenciar o laboratório de informática de maneira eficiente, tais como:

- **Virtualização de servidor:** A virtualização de servidor é a partição de um servidor físico em servidores virtuais menores para ajudar a maximizar os recursos do servidor. Na virtualização de servidor, os recursos do próprio servidor são ocultados dos usuários e o software é usado para dividir o servidor físico em vários ambientes virtuais chamados de servidores virtuais ou privados. A virtualização de servidor pode ser usada para eliminar a proliferação de servidores físicos, para fazer um uso mais eficiente dos recursos do servidor, para melhorar a disponibilidade do servidor, para ajudar na recuperação de desastres, teste e desenvolvimento e para centralizar a administração do servidor.
- **Virtualização de desktop:** A virtualização de desktop é usada para separar um ambiente de desktop do computador físico e é considerada um tipo de modelo de computação cliente-servidor porque o desktop “virtualizado” é armazenado em um servidor centralizado. Uma única imagem de desktop, incluindo instalação e manutenção de software, atualizações e *patches*, é mantida no nível do servidor e, em seguida, entregue com segurança ao dispositivo final do aluno pela rede. Com ferramentas de virtualização de desktop, a equipe de Tecnologia da Informação (TI) pode habilitar o provisionamento central de computadores e aplicativos para melhor segurança, confiabilidade e padronização, além de ajudar a economizar tempo e trabalho, reduzir os custos de manutenção, aumentar a segurança, facilitar a implantação, reduzir os custos de energia e obter um gerenciamento centralizado.
- **Virtualização de armazenamento:** A virtualização de armazenamento é o processo de agrupar o armazenamento físico de vários dispositivos de armazenamento de rede para que pareça um único dispositivo de armazenamento. A virtualização de armazenamento ajuda o administrador de armazenamento a executar as tarefas de backup, arquivamento e recuperação com mais facilidade e em menos tempo. Ele permite a redução de dados duplicados e tenta maximizar a eficiência.
- **Virtualização de rede:** Na virtualização de rede, várias redes podem ser combinadas em uma única rede, ou uma única rede pode ser separada logicamente em várias partes.

A virtualização de rede trata todos os servidores e serviços da rede como um único *pool* de recursos que pode ser acessado sem levar em conta seus componentes físicos. A virtualização de rede tem como objetivo otimizar a velocidade, confiabilidade, flexibilidade, escalabilidade e segurança da rede.

2.1.3 A Virtualização de Desktop

A virtualização de desktop é uma tecnologia que permite aos usuários executar o carregamento de uma estação de trabalho para acessar o desktop a partir de um dispositivo conectado. Esse modelo tem como base a ideia de isolar o sistema operacional e suas aplicações do dispositivo físico que o acessa. Os produtos de virtualização de desktop não realizam completamente a virtualização de desktop, em outras palavras, este software executa a instância da máquina virtual no computador local ou remoto, o que é possível ao fornecer e emular o conjunto de hardware para que possa ser transferido para outra máquina por meio de mídia removível ou rede. A virtualização de desktop permite ao usuário ter seu próprio desktop onde pode trabalhar e salvar ou modificar dados sem entrar em conflito com outros usuários possíveis que podem ser executados em cima de uma instância de sistema operacional separada ou compartilhada e que pode ser acessada através de qualquer dispositivo que têm conectividade com a rede. Dessa maneira, será utilizado como parte da solução proposta, a virtualização de desktop.

De acordo com [Patil e Shekar \(2012\)](#) a implementação da virtualização de desktop pode seguir as etapas fornecidas a seguir.

1. Na primeira etapa, o administrador do sistema ou o fornecedor deve coletar informações sobre que tipo de virtualização e sistemas operacionais subjacentes os usuários usarão e o número de usuários que usarão aquela instância particular virtualizada de desktop. É importante porque decide que tipo de implantação toda a configuração precisará.
2. A segunda etapa consiste na implantação da tecnologia de virtualização. Ela constitui no isolamento de um sistema operacional individual fornecendo um conjunto emulado de dispositivos de hardware. Também permite que o computador desktop virtual ou instância virtualizada em particular, migre e reutilize ou copie livremente para que possa ser escalado e usado sob demanda sempre que houver necessidade, sem reiniciá-lo do zero.
3. A terceira etapa consiste na análise e implementação da rede e gerenciamento centralizado para melhor controle do número de instâncias em execução de virtualização de desktops. A implementação de uma rede adequada permite que o usuário acesse perfeitamente sua área de trabalho na rede fornecida. O gerenciamento centralizado permite facilidade de controle em várias instâncias virtualizadas em execução em uma ou mais máquinas dentro da rede.

4. A quarta etapa, consiste em gerenciar a disponibilidade, escalabilidade e segurança. Quando o isolamento e a rede da instância virtualizada de desktop são implantados, cada usuário deve obter seu desktop sem problemas e deve suportar escalabilidade para fornecer desktop personalizado sob demanda. A disponibilidade de desktops virtualizados pode ser alcançada por redundância e armazenamento dessas instâncias no armazenamento de rede onde pode ser acessado facilmente.

Os softwares de virtualização de desktop atuais seguem apenas as duas primeiras etapas e pouco ou nenhuma concentração nas duas etapas restantes.

2.2 Oportunidades e benefício das tecnologias virtuais

A virtualização oferece oportunidades e benefícios potenciais em um ambiente educacional, incluindo a criação de uma área restrita virtual para atividades arriscadas, fornecendo instrução em vários aplicativos ou sistemas operacionais, usando dispositivos virtuais, usando equipes virtuais para aplicativos cliente/servidor e permitindo educação a distância via infraestrutura de virtualização de área de trabalho ou desktop. De acordo com [Lunsford \(2009\)](#) as tecnologias de virtualização são ideais para fornecer uma área restrita para a execução de código arriscado ou envolvimento em experimentos.

Conforme já discutido, os alunos podem realizar ações na máquina virtual que não deveriam realizar em uma máquina física, pois essas ações representam uma ameaça à disponibilidade e estabilidade da máquina física. Os alunos podem se envolver em atividades em uma máquina virtual que os educadores e administradores de rede não gostariam que eles fizessem em uma máquina *host*³ ou rede ativa. Por exemplo, os alunos podem decifrar as senhas em suas máquinas virtuais. Como a máquina virtual limitou o escopo dessa atividade, as informações reais do usuário na máquina *host* e na rede permaneceram seguras. Os alunos também podem experimentar vários aplicativos, como firewalls, antivírus e aplicativos de criptografia que têm o potencial de enfraquecer a segurança se não forem implementados adequadamente. Mais uma vez, a máquina virtual isola a potencial vulnerabilidade da máquina real.

A virtualização fornece uma ferramenta útil para a introdução de sistemas operacionais. Enquanto uma configuração de inicialização dupla, ou ainda, *dual boot* permite a introdução de vários sistemas operacionais, a virtualização oferece várias vantagens, de acordo com [Lunsford \(2009\)](#) algumas dessas vantagens são:

- Primeiro, a virtualização pode ser mais fácil de implementar e gerenciar do que as configurações de inicialização dupla.

³ *Host* é qualquer computador ou máquina conectado a uma rede, que conta com número de IP e nome definidos. Essas máquinas oferecem recursos, informações e serviços aos usuários ou clientes.

- Em segundo lugar, como a máquina virtual e o sistema operacional *host* coexistem e são acessíveis, o aluno pode acessar mais facilmente os dois recursos em sala de aula.
- Terceiro, os alunos podem instalar e configurar seus próprios sistemas operacionais, ampliando assim a gama de tópicos possíveis.
- Quarto, o educador pode distribuir máquinas virtuais aos alunos se não houver necessidade de eles instalarem seus próprios sistemas operacionais.
- Finalmente, pode-se introduzir vários sistemas operacionais diferentes usando uma máquina virtual para cada sistema operacional (LUNSFORD, 2009).

2.2.1 Máquinas Virtuais mais Populares em 2021/2022

Uma máquina virtual é um ambiente de software que emula um sistema de computador. Eles imitam arquiteturas de computador e oferecem a mesma funcionalidade de um computador físico como dito anteriormente. A seguir consta uma lista dos principais softwares de máquinas virtuais e subsistemas para Windows, Mac e Linux de acordo com Plaza (2019), Pickavance e Turner (2020). A lista contém software de código aberto (gratuito) e comercial (pago).

1. **VMware Workstation Player:** O VMware Workstation Player permite instalar e executar vários sistemas operacionais em um ambiente seguro, isolado do sistema principal. O VMware permite a execução de mais de 200 sistemas nas versões de 32 e 64 bits, como Windows 10/8/7/ XP, Ubuntu, Red Hat, Debian, Mint, Fedora. Ainda, possui a opção de alocar até oito núcleos e até 32 GB de RAM para uma máquina virtual individual, desde que os componentes do computador permitam. O programa suporta a tecnologia de arrastar e soltar onde basta arrastar o arquivo do sistema para a janela da máquina virtual e ele será copiado. Sendo um dos poucos *hosts* que suportam o DirectX 10 e o OpenGL 3.3, permite trabalhar no modo de virtualização o CAD e outros aplicativos acelerados pela GPU. Outro ponto forte é o modo de suspensão, que permite desligar a máquina virtual em alguns segundos e reiniciá-la de onde foi desligada. O valor unitário varia entre US\$ 149,00 com sua versão básica a US\$ 241,00 para sua versão mais completa.
2. **VirtualBox:** O VirtualBox é um dos programas de virtualização mais conhecidos além de ser de licença livre. Ele permite a criação e gerenciamento de várias máquinas virtuais podendo executar diversos sistemas operacionais diferentes. Suporta portas USB e muitos tipos diferentes de equipamentos conectados a um computador ou laptop. Pode-se ainda, criar uma emulação de disco virtual em uma unidade USB portátil ou pendrive, para que sempre tenha-se o sistema virtual à mão. O VirtualBox também permite clonar máquinas e compartilhar pastas entre a máquina virtual e o *host*. Caso seja necessário a utilização de variados sistemas operacionais como máquina virtual o VirtualBox possui o respaldo necessário em relação ao múltiplo suporte de sistemas como o Windows NT e todas

as versões a partir do XP, bem como qualquer versão Linux 2.4 ou posterior, Solaris, OpenSolaris e até OpenBSD Unix. A criação de máquinas virtuais no VirtualBox é feita usando um assistente passo a passo. Compreender o seu funcionamento é possível para qualquer usuário de computador mais ou menos experiente.

3. **Xen Project:** O Xen Project é um VMM de código aberto que serve como um *hypervisor* tipo I para muitos sistemas operacionais. Originalmente desenvolvido pela Universidade de Cambridge, o pessoal que o criou transformou-o em uma empresa que mais tarde foi adquirida pela Citrix. O projeto Xen agora está caminhando com a Linux Foundation para a promoção de aplicativos de código aberto. É especialmente usado para virtualização avançada, especialmente em servidores, em ambientes comerciais e de código aberto. Isso inclui, entre outros, infraestrutura como aplicativos de serviço (do inglês *Infrastructure as a Service* - IaaS), virtualização de desktop e virtualização de segurança. O software Xen Project é usado até em sistemas automotivos e de aviação. Este serviço é especialmente útil para nuvens de grande escala e pode ser facilmente usado com AWS, Azure, Rackspace, IBM Softlayer e Oracle. O foco principal é a segurança, usando a menor base de código possível, tornando-a não apenas segura, mas particularmente flexível.
4. **Hyper-V:** Originalmente Viridian, depois Windows Server Virtualization, foi renomeado para Hyper-V Server quando foi lançado no final de 2008. O Hyper-V é uma das maneiras mais rápidas e fáceis para criar uma máquina virtual, pois o software que já está integrado ao Windows 8.1 e Windows 10. Ele pode ser usado em casa e no ambiente corporativo, pois pode gerenciar todas as máquinas virtuais criadas em um servidor equipado com o Windows 10. A virtualização é extremamente eficiente, principalmente com hardware capaz de suportar instruções específicas do Hyper-V, presente em processadores modernos. O suporte ao sistema convidado inclui Windows Server, Windows XP SP3 ou mais recente, Linux com kernel 3.4 ou mais recente e FreeBSD. Porém, o suporte ao *driver* para Linux não é excelente. Por se tratar de uma ferramenta integrada, é um software indicado para usuários relativamente inexperientes nesse universo de virtualização. O Hyper-V tem 5 edições de preços com valores entre US\$ 24,95 a US\$ 199,00.
5. **Parallels Desktop:** Parallels Desktop para Mac é um software de virtualização de emulação de hardware, que utiliza tecnologia de hipervisor que funciona mapeando os recursos de hardware do computador *host* diretamente para os recursos da máquina virtual. Cada máquina virtual portanto, opera de forma idêntica a um computador autônomo, com praticamente todos os recursos de um computador físico. Como todas as máquinas virtuais convidadas usam os mesmos drivers de hardware, independentemente do hardware real no computador *host*, as instâncias de máquinas virtuais são altamente portáteis entre computadores. Sua licença custa US\$ 79,99 para estudantes ou US\$ 99,99 ao ano para desenvolvedores, testadores e usuários avançados.

6. **WSL:** O subsistema Windows para Linux (do inglês *Windows Subsystem for Linux*) e abreviado como WSL é um módulo do sistemas operacionais Windows 10, 11 e Server 2019 que visa a disponibilizar um ambiente Linux compatível no sistema da Microsoft, de forma que se possam executar programas nativos dos sistemas GNU/Linux dentro do próprio Windows sem a necessidade de emuladores ou do uso de máquinas virtuais [Microsoft \(2022\)](#).

Dentre as tecnologias apresentadas, o software de virtualização utilizado foi a virtualização de desktop com o software VirtualBox.

2.3 Arquivos de imagem de disco do VirtualBox

Os arquivos de imagem de disco residem no sistema *host* e são vistos pelos sistemas convidados como discos rígidos de uma determinada geometria. Quando um sistema operacional convidado lê ou grava em um disco rígido, o VirtualBox redireciona a solicitação para o arquivo de imagem.

Assim como um disco físico, um disco virtual tem um tamanho ou capacidade que deve ser especificado quando o arquivo de imagem é criado. No entanto, ao contrário de um disco físico, o VirtualBox permite expandir um arquivo de imagem após a criação, mesmo que já tenha dados ([VIRTUALBOX, 2022](#)).

O VirtualBox suporta os seguintes tipos de arquivos de imagem de disco:

- **VDI:** Normalmente, o VirtualBox usa seu próprio formato de contêiner para discos rígidos convidados. Isso é chamado de arquivo de imagem de disco virtual (VDI). Esse formato é usado quando é criado uma nova máquina virtual com um novo disco.
- **VMDK:** O VirtualBox também oferece suporte total ao popular e aberto formato de contêiner VMDK que é usado por muitos outros produtos de virtualização, como VMware.
- **VHD:** O VirtualBox também oferece suporte total ao formato VHD usado pela Microsoft.
- **HDD:** Arquivos de imagem do Parallels versão 2 (formato HDD) também são suportados. Devido à falta de documentação do formato, as versões mais recentes, como 3 e 4, não são suportadas. No entanto, é possível converter esses arquivos de imagem para o formato da versão 2 usando as ferramentas fornecidas pela Parallels.

Independentemente da capacidade e formato do disco, existem duas opções para criar uma imagem de disco: tamanho fixo ou alocado dinamicamente.

- **Tamanho fixo:** Ao criar uma imagem de tamanho fixo, um arquivo de imagem será criado no sistema *host* que tem aproximadamente o mesmo tamanho que a capacidade do disco

virtual. A criação de uma imagem de tamanho fixo pode levar muito tempo, dependendo do tamanho da imagem e do desempenho de gravação do disco rígido.

- **Alocado dinamicamente** A imagem alocada dinamicamente é utilizada para um gerenciamento de armazenamento mais flexível. Esta imagem será inicialmente muito pequena e não ocupará nenhum espaço para setores de disco virtual não utilizados, mas aumentará toda vez que um setor de disco for gravado pela primeira vez, até que a unidade atinja a capacidade máxima escolhida quando a unidade foi criada. Embora esse formato ocupe menos espaço inicialmente, o fato de o VirtualBox precisar expandir o arquivo de imagem consome recursos de computação adicionais; portanto, até que o tamanho do arquivo em disco se estabilize, as operações de gravação podem ser mais lentas do que com discos de tamanho fixo. No entanto, depois de um tempo, a taxa de crescimento diminuirá e a penalidade média para operações de gravação será insignificante.

2.3.1 Exportar e Importar Máquinas Virtuais

Ao criar uma máquina virtual, toda a configuração está contida em um único arquivo. Em vez de espalhar arquivos em todos os lugares como instalações tradicionais de SO, o VirtualBox os encapsula em um arquivo independente para manter tudo organizado e em um só lugar, garantindo que a VM tenha tudo o que precisa para funcionar. Se esse arquivo estiver em conformidade com o *Open Virtualization Format* (OVF ou OVA), ele poderá ser importado ou exportado pelo VirtualBox.

2.4 Considerações finais

Neste capítulo foi exposto o referencial bibliográfico com os assuntos relevantes para o entendimento da virtualização e suas possibilidades.

3

O caso do curso de Ciência da Computação, Unioeste - Cascavel

O objetivo principal deste capítulo é contextualizar o cenário dos laboratórios de informática do curso de Bacharelado em Ciência da Computação (BCC) da Unioeste campus de Cascavel e apresentar a infraestrutura disponível, mostrando o que existe atualmente e apontando os principais problemas.

3.1 O cenário dos laboratórios

O curso de Bacharelado em Ciência da Computação é uma importante unidade de ensino da Unioeste, que recebe em torno de 100 alunos por dia, comportando-os em salas de aula, auditórios e laboratórios. Para atender às diversas disciplinas do curso BCC, bem como outros cursos temporários como Programas de Extensão ou outros cursos do campus, o campus dispõe de 3 laboratórios de informática designados ao curso BCC que são de livre acesso pelos alunos quando não há aula ministrada.

As aulas que acontecem nesses laboratórios fazem uso de variados tipos de softwares, indo desde programação básica até disciplinas avançadas. Pode-se separar a Grade Curricular que utiliza os laboratórios para fins práticos em:

- **1º Ano:** Algoritmos e Introdução à Ciência da Computação.
- **2º Ano:** Estruturas de Dados, Cálculo Numérico Computacional, Probabilidade e Estatística, Linguagens de Montagem, Processo de Engenharia de Software I, Teoria da Computação e Sistemas Digitais.
- **3º Ano:** Sistemas Operacionais, Organização e Arquitetura de Computadores, Processo de Engenharia de Software II, Computação Gráfica, Tecnologias para Desenvolvimento de Sistemas, Inteligência Artificial, Banco de Dados e Linguagens de Programação.

- **4º Ano:** Projeto e Análise de Algoritmos, Redes de Computadores, Compiladores, Processamento de Imagens Digitais, optativas como Sistemas Embarcados, Aprendizagem de Máquina e Programação Paralela.

Cada laboratório suporta até 25 alunos e um professor. Dois laboratórios (Lab 1 e Lab 2) são equipados com 20 máquinas desktops completas e o Lab 3 equipado com 14 máquinas desktops completas. Em todos laboratórios os computadores possuem *dual boot* para Windows 10 e Linux Ubuntu.

3.1.1 Dificuldades Encontradas

De acordo com uma entrevista realizada com o laboratorista do curso BCC, uma das principais dificuldades enfrentadas é no início do ano letivo pois leva-se em torno de 7 a 10 dias de trabalho para formatar todos os computadores e realizar a instalação de SO e programas. A formatação dos computadores do laboratório é feita através da clonagem de HD com o Clonezilla, no qual clona-se a imagem em um HD alvo previamente configurado e então é realizada a cópia para os HDs das máquinas do laboratório. O tempo de formatação de cada máquina dura em torno de 1 hora, entretanto é comum ocorrer a quebra da ISO¹ durante o processo tendo então que refazer a formatação.

Outro problema destacado ocorre quando há a necessidade da instalação de um novo software ou uma atualização para determinada disciplina, o que pode levar boa parte da aula até que todos os alunos consigam realizar a instalação com sucesso. Ademais, há casos onde o tempo de instalação e configuração apropriada da ferramenta é muito grande e esse tempo gasto em contornar os problemas identificados pode atrapalhar o andamento de uma avaliação, prejudicando todos os alunos e professor.

Além disso, outro fator que dificulta mais o dia a dia do laboratorista, alunos e professores é a falta de monitoramento do ambiente e a tarefa manual de identificar se o hardware está estável em cada máquina. Pois após cada substituição de hardware como um HD com defeito é necessário realizar a formatação e instalação dos programas novamente.

3.1.2 Requisitos desejáveis para melhorar o ambiente

Baseado no cenário encontrado, foram identificados os seguintes requisitos para melhorar o funcionamento dos laboratórios:

1. Determinar um sistema operacional que tenha segurança, eficiência e desempenho.
2. Determinar um software de virtualização.
3. Criar uma imagem para cada série com o Linux Ubuntu 20.04 LTS.

¹ Um arquivo ISO é uma cópia idêntica (imagem) de arquivos gravados num CD/DVD/USB.

4. Criar uma única imagem para o Windows 10.
5. Centralizar em um servidor da Unioeste as imagens criadas.
6. Disponibilizar aos alunos, professores e laboratorista o acesso às imagens.
7. Implantar ferramentas de monitoramento e backup de sistemas.

3.2 Determinando um Sistema Operacional com segurança, eficiência e desempenho

A segurança de um sistema operacional refere-se à garantia de algumas propriedades fundamentais associado ao próprio sistema. De acordo com [Maziero \(2019\)](#) pode-se classificar a palavra segurança de um sistema operacional em três qualidades. A primeira qualidade busca entendê-la como um sistema que possui proteção contra ameaças intencionais como intrusão, ataque ou roubo de informação. Já a segunda qualidade se relaciona a problemas que possam ser acusados pelo sistema aos seus usuários ou ao ambiente bem como acidentes provocados por erros de programação. Por fim, a terceira qualidade pode ser entendida como a confiabilidade do sistema onde um sistema confiável é construído para tolerar erros de software, hardware ou dos usuários. Apesar dos sistemas operacionais serem projetados para serem seguros com a atualização contínua é possível direcionar seus recursos afim de obter maior segurança tornando-o menos vulnerável a ataques.

Um sistema operacional eficiente e com desempenho é aquele que utiliza bem seus recursos computacionais, ou seja são aqueles que executam programas de forma fácil e conveniente ao uso. Portanto, o sistema operacional escolhido para os laboratórios foi o Linux Ubuntu (versão 20.04 LTS²) pois atende aos requisitos desejados.

3.3 Software de virtualização e suas VMs

Conforme discutido no Capítulo 2 o tipo de virtualização escolhida para este experimento foi a virtualização de desktop, executando a instância da máquina virtual nos computadores do laboratório e o software de virtualização utilizado foi o VirtualBox pois trata-se de um software gratuito que possui uma interface de fácil compreensão e é ideal para testar, desenvolver, demonstrar e implementar soluções em diferentes plataformas a partir de uma única máquina ([ORACLE, 2021](#)).

² LTS - do inglês *long term support* (suporte de longo prazo). Significa que durante toda a vida de uma versão há um compromisso de atualizar, corrigir e manter o software.

3.4 Centralização das VMs

O laboratorista cedeu uma máquina no Laboratório de Computação de Alto Desempenho (LCAD) para a centralização das VMs. Esta máquina desempenhou o papel de servidor do experimento. Assim como as máquinas *host*, o servidor foi configurado com o Linux Ubuntu 20.04 LTS e possui o VirtualBox para modificar as VMs. O diretório definido para salvar os arquivos *Open Virtual Appliance* foi “/home/aluno/Downloads/VirtualBox/ova”.

3.5 Disponibilização das VMs

A ferramenta utilizada para realizar a sincronização remota para disponibilizar as VMs foi o *rsync*. O *rsync* é utilitário nativo dos sistemas do tipo Unix, é ideal para a criação de *scripts* já que todos seus comandos podem ser utilizados pelo terminal e também utiliza o *Secure Shell* (SSH) para realizar a transferência dos arquivos pela rede, tornando assim a transferência mais segura com um protocolo de rede criptográfico. Além disso, o *rsync* por padrão realiza a sincronização incremental, ou seja, transfere apenas a diferença dos arquivos da origem e destino. Portanto, a primeira sincronização levará um tempo maior e as subsequentes serão mais rápidas uma vez que o arquivo modificado já exista no diretório de destino. Nenhuma instalação adicional foi necessária para realizar a sincronização já que o Linux Ubuntu 20.04 LTS possui nativamente o *rsync*.

A utilização do *rsync* será restrita para a atualização das VMs no servidor e nos *hosts* para manter a segurança e integridade dos arquivos pois é necessário a senha de usuário e IP do servidor para a sincronização. Portanto, o *script* de atualização não será compartilhado aos alunos e a atualização das VMs nos computadores pessoais poderá ser feita com a cópia do arquivo no diretório destino das máquinas *host* nos laboratórios.

3.6 Monitoramento de aplicação: Nagios

O Nagios é uma ferramenta de código aberto para monitorar aplicativos de rede, serviços e outros dispositivos de rede essenciais. Ele detecta falhas por meio de monitoramento ativo e envia alertas de notificação quando uma atividade suspeita é detectada. Da mesma forma, ele envia notificações quando os dispositivos ou serviços voltam ao seu estado funcional.

O Nagios funciona em uma arquitetura cliente/servidor onde o servidor que hospeda o Nagios usa plugins para se comunicar com *hosts* remotos através de agentes como o *Nagios Remote Plugin Executor* (NRPE). Os relatórios finais gerados a partir dos logs são representados visualmente por meio da interface do usuário.

O motivo pelo qual foi determinado o uso do Nagios foi que além de possuir uma versão gratuita e monitoração remota, é possível personalizar os plugins desejados para a monitoração.

3.7 Ferramenta de backup: Systemback

O backup é utilizado para auxiliar na prevenção e perda de dados acidentalmente seja por falha física ou humana. O Systemback realiza backup e restauração de sistemas com Ubuntu, ainda possui recursos extras como a cópia do sistema de uma partição, instalação do sistema, criação de Live em CD/DVD/USB do sistema atual com ou sem os dados do usuário e até reparação do Grub. Um sistema Live CD é um disco compacto inicializável que contém seu próprio sistema operacional. A inicialização a partir de um Live CD permite executar o sistema operacional sem fazer alterações no sistema operacional, discos rígidos ou arquivos existentes no computador. O conceito de Live CD pode ser estendido a qualquer outro dispositivo de armazenamento.

A criação da Live personalizada foi determinante para a escolha do Systemback. A partir da configuração de uma única máquina foi criada uma ISO, ou imagem de disco, para formatar as demais máquinas do laboratório. A Figura 9 mostra a execução do Systemback e suas opções.

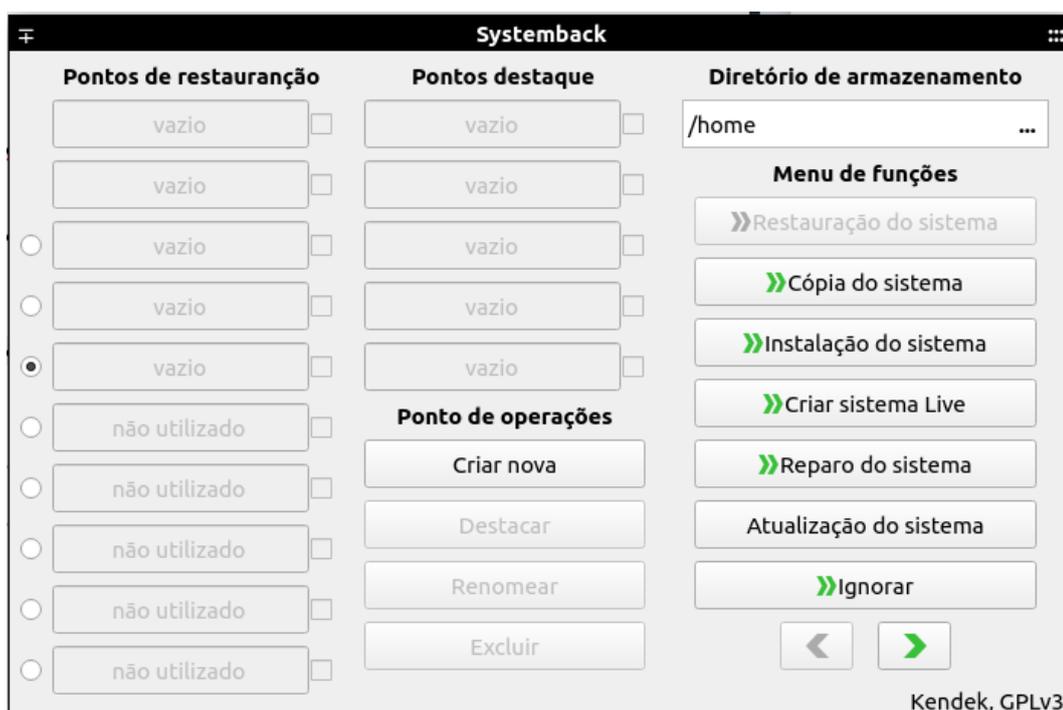


Figura 2 – Tela de opções do Systemback.

3.8 Configuração e Atualização

Esta seção se encarrega de listar a sequência de passos para a configuração e atualização dos ambientes com as ferramentas citadas anteriormente.

Para realizar a configuração do ambiente deverá ser seguido os seguintes passos como mostra a Figura 3:

1. Instalar e configurar o VirtualBox e Nagios no servidor. O VirtualBox será utilizado para atualizar as VMs e o Nagios para monitorar os computadores dos laboratórios.
2. Realizar a formatação de um computador *host* com o Sistema Operacional Ubuntu 20.04 LTS (instalação mínima).
3. Instalar e configurar o VirtualBox, Nagios e *script* de atualização pelo *rsync* no computador *host* do passo anterior. Este passo irá determinar o tamanho da Live personalizada e é desejável a instalação exclusivamente das ferramentas essenciais.
4. Criar a Live personalizada utilizando o Systemback. Esta Live poderá ser convertida para ISO se seu tamanho não ultrapassar 4 GB. O Systemback permite copiar para um pendrive a Live criada.
5. Formatar os demais computadores com a Live personalizada. Desta maneira será dispensada a instalação e configuração das ferramentas novamente.

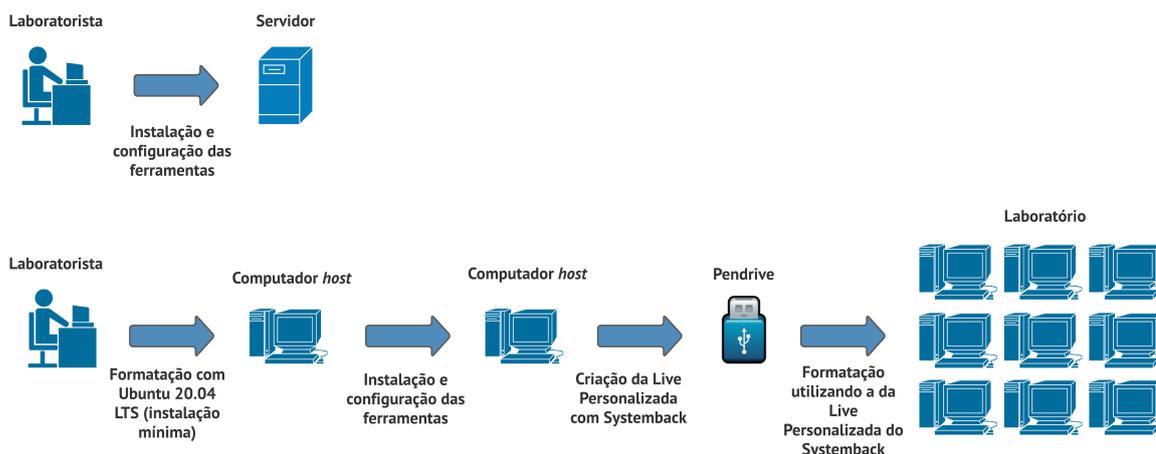


Figura 3 – Configuração dos ambientes.

Durante a formatação das máquinas do laboratório não foi realizado a configuração de domínio das máquinas. Portanto, não há a centralização dos nomes de contas e os nomes de computadores em um banco de dados compartilhado. O mapeamento de IP das máquinas foi feita manualmente com a ferramenta *net-tools* do Linux.

Para realizar a atualização do ambiente, deverá ser seguido os seguintes passos como mostra a Figura 4:

1. Quando necessário, o laboratorista poderá atualizar uma ou mais VMs no servidor. Esta atualização da VM pode ser realizada diretamente no servidor ou remotamente utilizando um *script*.

2. A atualização das máquinas *hosts* irá ocorrer de forma automática através de *script* que será executado após a inicialização automática programada. Também é possível executar o *script* de forma manual pelo terminal.
3. Os alunos poderão copiar para sua máquina pessoal a VM atualizada na máquina *host* acessando a cópia da VM na pasta destino `/home/aluno/Downloads/VirtualBox/ova/`.

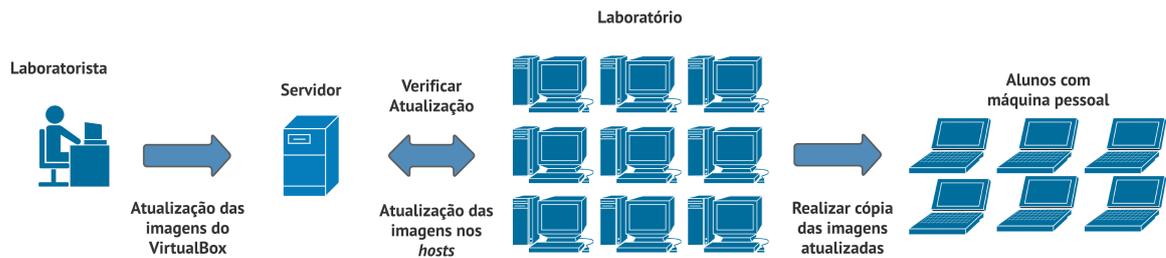


Figura 4 – Atualização das imagens.

3.9 Considerações finais

Neste capítulo foi detalhado o cenário atual dos laboratórios do curso de BCC, quais são as dificuldades encontradas e quais requisitos desejáveis para melhor a manutenção, gerenciamento e monitoramento dos laboratórios. Também foi apresentado as ferramentas e processos para alcançar os requisitos desejados. O apêndice deste trabalho apresenta o passo a passo de como configurar um sistema com o Ubuntu com as ferramentas utilizadas no experimento.

4

Experimentação

No Capítulo 2 vimos que o tipo de virtualização determinado foi a virtualização de desktop e o software escolhido para executar a VM foi o VirtualBox. Em sequência, no Capítulo 3 foi evidenciado o cenário dos laboratórios do curso de Bacharelado em Ciência da Computação, suas dificuldades e quais são os requisitos desejados para melhorar o ambiente. Ainda, foi ilustrado uma série de ferramentas que em conjunto podem melhorar o ambiente. Este Capítulo mostra a experimentação com as ferramentas mencionadas no Capítulo anterior e a sequência de passos para reproduzir o experimento considerando o emprego do sistema operacional Linux Ubuntu 20.04 LTS.

4.1 Configuração e disponibilização das VMs

A primeira versão da VM do Ubuntu foi obtida de [OSboxes \(2022\)](#) no formato VDI contendo 2,58 GB. Após isto, foi incluído na VM o pacote de idiomas com a versão português e criado o usuário “aluno”. Na sequência, utilizando o VirtualBox foi realizada a exportação desta imagem no formato OVA, considerando uma imagem para cada turma para que sejam instalados separadamente os programas utilizado em sala de aula. Cada imagem obteve o tamanho de 6,9 GB. O mesmo procedimento foi realizado para uma única imagem do Windows 10 disponível em [Windows \(2022\)](#) e seu tamanho em disco no formato OVA foi de 5,8 GB.

Com o objetivo de facilitar o uso do rsync foram criados 2 *scripts*. O primeiro realiza a atualização dos arquivos no servidor a partir de outra máquina qualquer dentro da rede da Unioeste. O segundo *script* realiza a atualização das VMs contidas no servidor para os demais computadores *hosts* dos laboratórios. Estes computadores foram programados para ligar a partir da meia noite com o intuito de verificar por atualizações nas VMs do servidor e desligar automaticamente após o término do *script*. Essa inicialização automática foi configurada pela BIOS RTC que é responsável pela inicialização em horário agendado através da BIOS.

4.1.1 Atualização das imagens no servidor

O primeiro *script*¹ utilizado se encarrega de atualizar as imagens no servidor ou localmente com o auxílio de caixa de diálogo. Este *script* é opcional, caso o laboratorista utilize o próprio servidor para alterar as imagens não há a necessidade de sua execução. O código a seguir ilustra o *script* denominado `rsyncPush.sh`.

Código 1 – *Script* `rsyncPush.sh` primeira parte.

```

1  #!/usr/bin/env bash
2  # VARIÁVEIS
3  LOG="$(date +%m%Y)"
4  ARQUIVO_LOG="bkp-$LOG.log"
5  MENSAGEM_LOG="#$(date +%A, %d %B %Y)#"
6  # TESTES
7  [ ! -x "$(which dialog)" ] && sudo apt install dialog -y 1> /dev/null
8  2>&1 # dialog instalado?
9  [ ! -x "$(which rsync)" ] && sudo apt install rsync -y # rsync
10 instalado?
11 # FUNCOES
12 Backup_local () {
13     dialog --infobox 'Iniciando Backup...' 3 25; sleep 1
14     echo "$MENSAGEM_LOG" >> "$ARQUIVO_LOG"
15     rsync -avhP "$dir_origem" "$dir_destino" --log-file="$ARQUIVO_LOG" \
16     | perl -lane 'BEGIN { $/ = "\r"; $|++ } $F[1] =~ /(\d+)%/ && print
17     $1' \
18     | dialog --gauge 'Aguarde... Copiando Arquivos' 8 70 0
19     dialog --msgbox 'Backup concluído com sucesso!' 6 35
20     dialog --title 'Log de Backup' --textbox "$ARQUIVO_LOG" 0 0
21     clear
22     exit
23 }
24 Backup_remoto () {
25     clear
26     echo "$MENSAGEM_LOG" >> "$ARQUIVO_LOG"
27     sudo rsync -avhP -e ssh "$dir_origem" "$dir_destino" --log-file="
28     $ARQUIVO_LOG" \
29     | perl -lane 'BEGIN { $/ = "\r"; $|++ } $F[1] =~ /(\d+)%/ && print
30     $1'
31     dialog --msgbox 'Backup concluído com sucesso!' 6 35
32     dialog --title 'Log de Backup' --textbox "$ARQUIVO_LOG" 0 0
33     clear
34     exit
35 }

```

¹ Esse *script* é uma adaptação do *script* criado por Flávio Varejão e seu conteúdo original pode ser encontrado em <https://github.com/Flavio-Varejao/Backup-Rsync>.

Código 2 – Script rsyncPush.sh segunda parte.

```
32 # EXECUCAO DO RSYNC
33 dialog --title 'Backup com rsync' \
34 --yesno 'Deseja fazer um Backup?' 6 28
35
36 [ $? -eq 0 ] && {
37     menu=$(dialog --title 'Backup com rsync' \
38     --menu 'Selecione o tipo de Backup' 10 32 2 \
39     1 'Backup local' \
40     2 'Backup remoto' --stdout \
41     )
42     case "$menu" in
43     1)
44         dir_origem=$(dialog --title 'Selecione o arquivo ou diretorio' \
45         --fselect "$HOME"/ 14 70 --stdout)
46         [ $? -eq 0 ] && {
47             dir_destino=$(dialog --title 'Selecione o diretorio em que sera
48             salvo' \
49             --dselect "$HOME"/ 14 50 --stdout)
50             [ $? -eq 0 ] && {
51                 Backup_local
52             }
53             clear && exit 0
54         }
55         clear && exit 0
56     ;;
57     2)
58         dir_origem=$(dialog --title 'Selecione o arquivo ou diretorio' \
59         --fselect "$HOME"/ 14 70 --stdout)
60         [ $? -eq 0 ] && {
61             dir_destino=$(dialog --title 'Selecione o diretorio em que sera
62             salvo' \
63             --dselect "$HOME"/ 14 50 --stdout)
64             [ $? -eq 0 ] && {
65                 Backup_remoto
66             }
67             clear && exit 0
68         }
69         clear && exit 0
70     ;;
71     esac
72 }
73 clear && exit
```

Os Códigos 1 e 2 ilustram o backup determinando o diretório de origem e o diretório de destino com o auxílio da caixa de diálogo. Com a utilização deste *script* para realizar a carga das imagens para o servidor o tempo estimado é de 10 minutos na rede cabeada para transferir 4 arquivos OVA do Ubuntu com 6.9 GB e um arquivo OVA do Windows 10 com 5,8 GB.

4.1.2 Atualização das imagens nos Laboratórios

O segundo *script* se encarrega de atualizar os demais computadores ou *hosts* com o conteúdo das imagens disposto no servidor. Este *script* verifica no servidor se houve alteração de alguma VM e realiza o download para o *host*, remove VM antiga e inclui a VM nova. Este processo pode ser um pouco demorado e sua execução pode ser realizada de dois modos. O primeiro modo é executar o *script* manualmente pelo terminal e caso uma VM foi atualizada mas está em execução pelo VirtualBox será reportado um erro no terminal. O segundo modo é executado automaticamente no login do usuário “aluno” pelo *script* `.bashrc` (*script* utilizado para salvar e carregar as preferências de terminal e variáveis de ambiente) caso o login ocorra entre às 00:00 e 06:00. Em seguida, após término da atualização ocorre o desligamento automático da máquina se o fim da execução do *script* estiver dentro do intervalo definido anteriormente. Por este motivo na inicialização da máquina não é requerido a senha do usuário “aluno”. O tempo estimado para a primeira execução deste *script* é em torno de 1 hora já que é realizado o download de todas as VMs e a importação das VMs no VirtualBox. Seu tempo pode variar de acordo com o tamanho da VM modificada.

Código 3 – Script `rsyncPull.sh` primeira parte.

```
1 #!/usr/bin/env bash
2
3 echo "Verificando Atualizacoes..."
4 #Define caminhos de Origem e Destino
5 dir_origem="usuario@IP:~/Downloads/VirtualBox/ova/"
6 dir_destino="/home/$USER/Downloads/VirtualBox/ova/"
7
8 #Declaracao de Variaveis
9 listaArquivos=()
10 arquivosNovos="$(rsync -nia -e ssh $dir_origem $dir_destino --delete |
11     egrep -v "^(\.|<)"|\|"deleting")"
12 echo $arquivosNovos > vmsList.txt
13 echo $(sed 's/>f.st..... //g;s/>f+++++++ //g;s/cd+++++++ //g;s/\.\. //
14     g; s/*deleting //g' vmsList.txt) > vmsList.txt
15 #Atualiza a variavel listaArquivos com o nome dos arquivos que serao
16     atualizados
17 for arquivo in `cat vmsList.txt`; do
18     listaArquivos+=("$arquivo")
19 done
20 rm vmsList.txt
```

Código 4 – Script rsyncPull.sh segunda parte.

```

19 #Percorre lista de novos arquivos removendo as VMs que serao
    atualizadas
20 if [ $#listaArquivos[@] -eq 0 ]; then
21     echo "Nenhuma VM para remover."
22 else
23     echo "Atencao! Aguarde... Executando o RSYNC."
24     for i in "${listaArquivos[@]}"
25     do
26         removedVM=${i::-4}
27         VBoxManage unregistervm --delete "$removedVM"
28     done
29
30 #Verifica existencia do diretorio
31 if [ -d "$dir_destino" ]; then
32     cd $dir_destino
33 else
34     mkdir ${dir_destino::-4}
35     mkdir $dir_destino
36     cd $dir_destino
37 fi
38
39 #Realiza download dos arquivos novos.
40 rsync -avhP -e ssh $dir_origem $dir_destino --delete
41
42 #Percorre lista de novos arquivos criando as VMs atualizadas
43 echo "Importando VMs..."
44 for i in "${listaArquivos[@]}"
45 do
46     newVM=${i::-4}
47     VBoxManage import $i --vsys 0 --vmname $newVM
48 done
49 fi
50 #Verifica horario para desligar o computador.
51 t=$(date +%H)
52 if [[ $t -lt '6' ]]; then
53     echo "Desligando o computador..."
54     shutdown -h
55 else
56     echo "VirtualBox atualizado."
57 fi

```

Como mostra o Código 3 na linha 5, é preciso saber o nome de usuário e o IP da máquina servidor para o rsync ter sucesso na atualização. Também foi utilizado o VBoxManage que é a interface de linha de comando para o VirtualBox, por ele é possível remover a VM antiga e importar a nova através do nome da VM. Ex: Caso as VMs atualizadas possuam os nomes de turma1.ova e turma2.ova, o *script* irá buscar no VirtualBox pelas VMs com o nome “turma1” e “turma2” e removê-las para em seguida importar as novas dentro da pasta “/home/\$USER/Downloads/VirtualBox/ova”.

Os comandos do rsync utilizados neste trabalho estão listados na Tabela 1 com suas respectivas descrições.

Tabela 1 – Comandos do rsync.

Comando	Descrição
-a	Conservar arquivos e diretório durante a sincronização.
-e	Especificar o shell remoto a ser usado.
-h	Exibir os números de saída em um formato legível por humanos.
-i	Gerar um resumo de alterações para todas as atualizações.
-n	Executar um teste sem a sincronização.
-v	Modo detalhado para exibir informações durante a transferência.
-delete	Excluir arquivos estranhos dos diretórios de destino.
-progress	Mostrar o progresso da sincronização durante a transferência.
-partial	Manter arquivos parcialmente transferidos.
-P	O mesmo que -partial -progress.

Fonte: [Computer Hope \(2021\)](#)

Afim de obter maior segurança durante a transferência de dados na execução dos *scripts*, foi utilizado o protocolo de rede criptográfico SSH. O SSH usa criptografia de chave pública para autenticar o computador remoto. Ainda, para não haver a necessidade de informar a senha para autenticação do computador remoto toda vez que executar os *scripts* foi utilizado um par de chaves pública-privada geradas manualmente. O utilitário responsável por criar as chaves é chamado de ssh-keygen e vem incluso com o conjunto padrão de ferramentas do OpenSSH. O algoritmo de autenticação utilizado para criar as chaves público-privada foi o RSA.

A coleta de dados sobre o tempo de execução e sucesso/falha dos *scripts* foi obtida com testes em paralelo levando em consideração as 5 máquinas *host* disponibilizadas. Além dos testes manuais também foi levado em conta o cenário com a inicialização automática no horário predefinido para *boot*. Em ambos cenários os *scripts* executaram com sucesso.

4.2 Monitoramento com Nagios

O Nagios foi configurado para monitorar o servidor e as máquinas *host* dos laboratórios. Os plugins utilizados monitoram a utilização de CPU, o uso de disco, o total de processos, a quantidade de usuários e quantidade de processos zumbis (processos que concluíram sua execução, mas suas entradas não são removidas da tabela de processos). Com estes e demais plugins, o laboratorista poderá identificar com maior facilidade problemas e falhas da máquinas. A Figura 5 mostra os serviços sendo executado onde cada “client-IP” refere-se a uma máquina do laboratório.

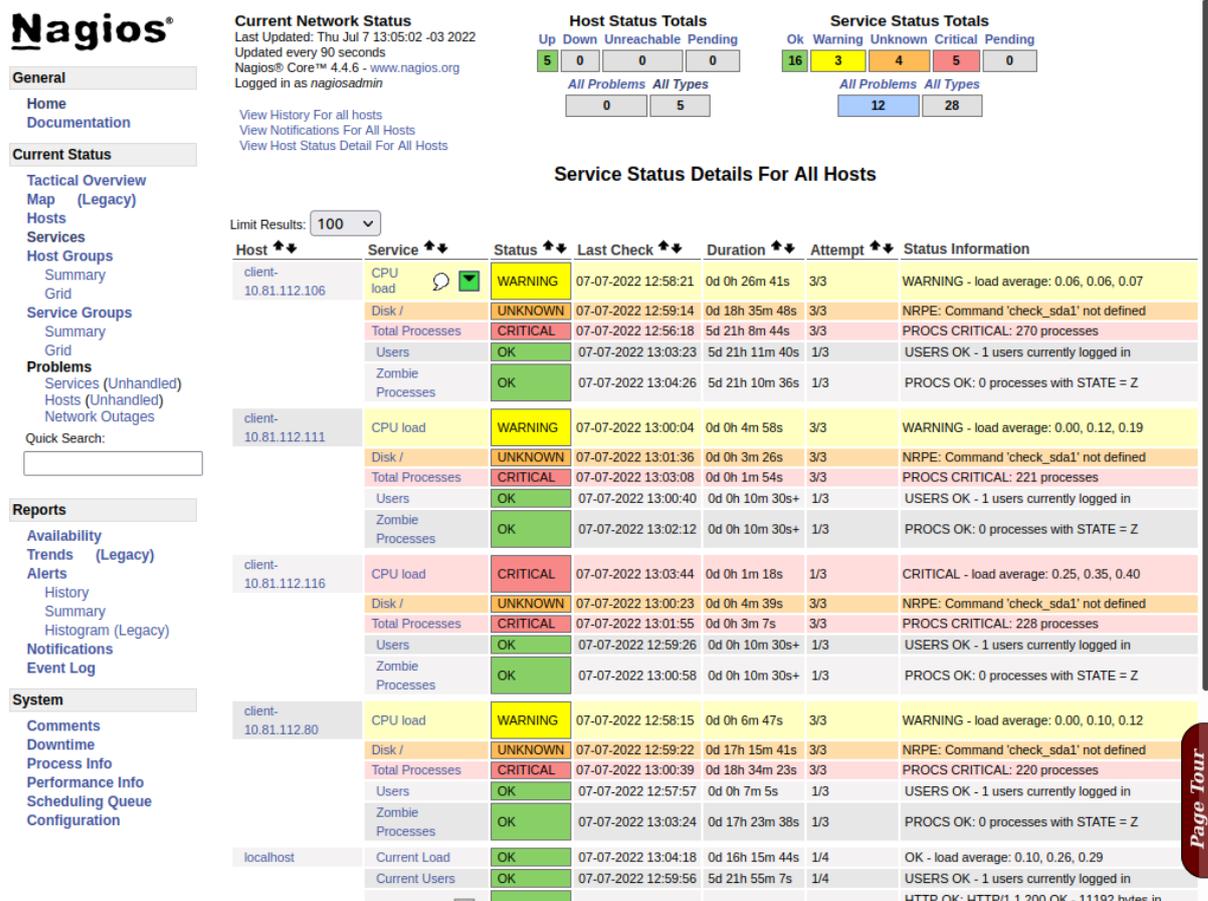


Figura 5 – Nagios: Monitoramento das máquinas *hosts*.

4.3 Criação da Live e formatação dos computadores

Com o intuito de obter a homogeneidade de software dos computadores dos laboratórios, foi necessário a criação de uma Live personalizada do Ubuntu 20.04 LTS pelo Systemback, pronto para executar os *scripts* de inicialização e atualização para as VMs e demais ferramentas como VirtualBox, Nagios e o próprio Systemback. Esta ISO criada pelo Systemback obteve o total de 3,95 GB e o tempo de formatação de cada computador no laboratório ficou em torno de 15 minutos. Foram disponibilizadas 5 máquinas com um HD alternativo para realizar o experimento. A partir dos testes realizados, constatou-se que ao utilizar 1 pendrive bootável em um laboratório completo com 20 máquinas, o tempo necessário para formatar todas as máquinas fica em torno de 5 horas acrescido de 1 hora para a sincronização e importação das VMs.

4.4 Considerações finais

Os resultados obtidos mostram como o uso da virtualização, *scripts* e outras ferramentas podem reduzir o tempo de trabalho do laboratorista com o gerenciamento, manutenção e configuração dos ambientes dos laboratórios. Levando em consideração o cenário ideal, onde não ocorre falhas para a formatação de um laboratório completo com 20 máquinas, foi possível

notar a redução do tempo estimado da formatação de 20 horas com o método anterior para 6 horas com o método do experimento, quando utilizado 1 dispositivo de mídia para realizar a formatação, obtendo então uma redução de tempo em 70%. Além disso, também foi evidenciado como realizar um monitoramento ativo da saúde dos computadores.

5

Conclusão

O gerenciamento e manutenção dos laboratórios na universidades não é uma tarefa fácil e é indispensável a busca por técnicas e ferramentas que auxiliem os laboratoristas a realizar estas atividades uma vez que demandam muito tempo. No caso do curso de Ciência da Computação da Unioeste Cascavel foi necessário identificar quais eram os problemas que demandam tempo e dificultam a manutenção nos laboratórios e foi visto que a formatação no início de todo ano letivo, instalação dos programas novos ou atualização e ainda a falta de monitoramento da saúde dos computadores são os maiores problemas enfrentados pelo laboratorista, alunos e professores.

Com base no cenário identificado, foram expostos os seguintes requisitos para melhorar o funcionamento dos laboratórios: determinar um sistema operacional que tenha segurança, eficiência e desempenho; determinar um software de virtualização; criar imagens para o VirtualBox; centralizar as imagens criadas em um servidor para disponibilização aos alunos, professores e laboratorista; por fim a implantação de ferramentas de monitoramento e backup de sistemas. Com base nesses requisitos foi utilizado as ferramentas VirtualBox para realizar a criação e manutenção das imagens ou VMs, o rsync associado aos *scripts* para realizar a sincronização e manter o servidor e *hosts* atualizados, também foi utilizado Nagios para monitoramento dos *hosts* com plugins do NRPE e por fim o Systemback para criar backups e a conversão da Live personalizada para formatar as demais máquinas com uma configuração pré-definida.

Os resultados obtidos do experimento mostrou que foi possível reduzir o tempo total de formatação e atualização dos laboratórios em 70% quando comparado ao método anterior que tinha como base a clonagem de um HD completo para formatar as máquinas. Isto ocorre porque a Live personalizada possui um tamanho muito menor e o tempo de formatação da máquina é consideravelmente mais rápida reduzindo de 1 hora com o método anterior para 15 minutos com o novo método. Para a primeira atualização com os *scripts* o tempo estimado foi de 1 hora portanto o tempo de formatação acrescido do tempo de atualização ficou aproximadamente 1

hora e 15 minutos por máquina. Ainda, como o `rsync` realiza a sincronização incremental as atualizações futuras deverão consumir um tempo menor.

Para os trabalhos futuros foi identificado que pode-se melhorar o ambiente utilizando uma versão mais compacta do Ubuntu como as versões Lite ou uma versão mais antiga como o Ubuntu 18.04 LTS, já que quanto menor for a Live personalizada gerada mais rápida será a formatação. Ainda, para dispensar a formatação utilizando pendrives (pois esse modelo dependem das unidades de pendrive disponível), é possível realizar a configuração de um Ambiente de Pré-execução (PXE¹). O PXE é um ambiente para inicializar computadores usando a interface da placa de rede sem a dependência de dispositivos de armazenamento, ou seja, o sistema operacional do equipamento é carregado pela interface de rede. Por fim, verificar como escalonar a execução dos *scripts* no momento da inicialização automática pela BIOS RTC, para não inundar a rede com todos os computadores *hosts* dos laboratórios atualizando ao mesmo tempo.

¹ PXE ou 'pixie', ambos do inglês: Preboot eXecution Environment.

Referências

BANDEIRA, G. R. de O. *Auto-Gerenciamento de Recursos em Infraestruturas baseada em Contêineres para Desktop-as-a-Service: Um Estudo de Caso nos Laboratórios de Informática da ECT/UFRN*. Dissertação (Dissertação de Mestrado) — Universidade Federal do Rio Grande do Norte, Natal, Rio Grande do Norte, Junho 2017. Citado na página 14.

BOSING, A.; KAUFMANN, E. R. Virtualização de servidores e desktops. *Unoesc & Ciência - ACET*, v. 3, n. 1, p. 47–64, 2012. Citado na página 14.

BULBROOK, H. Using virtual machines to provide a secure teaching lab environment. *Durham Technical Community College*, Durham, 2006. Disponível em: <http://www.infosecwriters.com/text_resources/pdf/Virtual_Machines_HBulbrook.pdf>. Acesso em 12 de Julho de 2020. Citado na página 11.

BULLERS, W. I.; BURD, S.; SEAZZU, A. F. Virtual machines - an idea whose time has returned: application to network, security, and database courses. In: *SIGCSE06: Technical Symposium on Computer Science Education*. Houston Texas USA: [s.n.], 2006. p. 102–106. Disponível em: <<https://dl.acm.org/doi/10.1145/1121341.1121375>>. Acesso em 12 de Julho de 2020. Citado na página 11.

BURD, S. D.; SEAZZU, A. F.; CONWAY, C. Virtual computing laboratories: A case study with comparisons to physical computing laboratories. In: *Journal of Information Technology Education: Innovations in Practice*. [S.l.: s.n.], 2009. v. 8, p. 55–78. Disponível em: <<http://www.jite.org/documents/Vol8/JITEv8IIP055-078Burd693.pdf>>. Acesso em 26 de Julho de 2020. Citado na página 10.

CARUSO, J. B. *The ECAR Study of Undergraduated Students and Information Technology*. Colorado: EDUCAUSE, 2007. Citado na página 11.

COLLINS, D. Using vmware and live cd's to configure a secure, flexible, easy to manage computer lab environment. *Journal of Computing Sciences in Colleges*, April 2006. Disponível em: <<https://dl.acm.org/doi/10.5555/1127389.1127439>>. Acesso em 12 de Julho de 2020. Citado na página 11.

COMPUTER HOPE. *Linux rsync command*. [S.l.], 2021. Disponível em: <<https://www.computerhope.com/unix/rsync.htm>>. Acesso em 16 de Julho de 2022. Citado na página 36.

FALVO, M. R. *Uma avaliação experimental do uso de desktops virtuais*. Dissertação (Dissertação de Mestrado) — UFSCar - Universidade Federal de São Carlos, São Paulo, Abril 2014. Disponível em: <<https://repositorio.ufscar.br/handle/ufscar/574?locale-attribute=es>>. Acesso em 02 de Novembro de 2020. Citado na página 12.

HAWKINS, B. L.; OBLINGER, D. The myth about managing it. *EDUCAUSE*, v. 42, n. 2, p. 10–11, March/April 2007. Disponível em: <<https://er.educause.edu/articles/2007/1/the-myth-about-managing-it>>. Acesso em 26 de Julho de 2020. Citado na página 11.

HOWELL, T. M. *DETERMINING COMPUTER LAB USAGE FOR ACADEMIC, SOCIAL AND PERSONAL PURPOSES*. Dissertação (Dissertação de Mestrado) — University of North Carolina, Chapel Hill, North Carolina, Abril 2007. Citado na página 10.

KIZZA, J. M. Africa can greatly benefit from virtualization technology – part 1. In: *International Journal of Computing and ICT Research*. [S.l.: s.n.], 2012. v. 6, p. 6–10. Disponível em: <http://nci2tm.sinhgad.edu/NCIT2M2014_P/index.html>. Acesso em 26 de Julho de 2020. Citado na página 11.

LUNSFORD, D. L. Virtualization technologies in information systems education. *Journal of Information Systems Education*, The University of Southern Mississippi, v. 20, n. 3, 2009. Citado 2 vezes nas páginas 19 e 20.

MAIA, B. C. Virtualização de desktops em ambiente educacional. *Revista Eletrônica Eng Tech Science*, v. 4, n. 1, p. 102–124, 2017. Citado na página 12.

MATTOS, D. M. F. *Virtualização: Vmware e xen*. [S.l.]: Grupo de Teleinformática e Automação da UFRJ, 2008. Citado na página 15.

MAZIERO, C. A. Sistemas operacionais: Conceitos e mecanismos. In: _____. 1. ed. Curitiba: [s.n.], 2019. cap. Conceitos básicos de segurança, p. 336–346. Citado na página 26.

MICROSOFT. *Microsoft: O que é o Subsistema do Windows para Linux?* [S.l.], 2022. Disponível em: <<https://docs.microsoft.com/pt-br/windows/wsl/about>>. Acesso em 17 de Julho de 2022. Citado na página 22.

MOK, H. N.; LEE, Y. L.; TAN, W. K. Setting up a low-cost lab management system for a multipurpose computing laboratory using virtualisation technology. *Australasian Journal of Educational Technology*, v. 28, n. 2, April 2012. Citado 2 vezes nas páginas 12 e 17.

NUNES, L. S. A virtualização: solução de baixo custo para falta de recursos de tecnologia da informação no ensino superior. *Revista Humanas Et Al.*, Paço do Lumiar, MA, v. 3, p. 178–188, Julho 2015. Citado na página 12.

OBERDICK, J. *2006 FACAC student survey reveals trends in student technology use of interest to faculty*. 2006. Citado na página 11.

ORACLE. *Visão geral do Oracle VM VirtualBox*. [S.l.], 2021. Disponível em: <https://www.oracle.com/br/a/ocom/docs/oracle-vm-virtualbox-technical-paper-pt-br.pdf?source=:ad:pas:go:dg:a_lad:71700000086180912-58700007521112083-p67711490795:RC_WWMK220429P00062:PORT&gclid=Cj0KCQjw8amWBhCYARIsADqZJoXTb0_Wvb9UIEsb2msbhmjPB_Mum-O-rKT3sdRr6seG5QgbXIaAF0aAtFdEALw_wcB&gclid=aw.ds>. Acesso em 10 de Julho de 2022. Citado 2 vezes nas páginas 26 e 46.

OSBOXES. *OSboxes*. 2022. Disponível em: <<https://www.osboxes.org/ubuntu/>>. Acesso em 10 de Julho de 2022. Citado na página 31.

PATIL, P.; SHEKAR, S. Desktop virtualization technologies and implementation. *IOSR Journal of Engineering*, v. 2, p. 310–314, February 2012. Citado na página 18.

PICKAVANCE, M.; TURNER, B. *Best virtual machine software of 2020: virtualization for different OS*. 2020. Disponível em: <<https://www.techradar.com/best/best-virtual-machine-software>>. Acesso em 25 de Setembro de 2020. Citado na página 20.

- PLAZA, W. R. *Top 5: melhores softwares para virtualização*. 2019. Disponível em:<<https://www.hardware.com.br/dicas/top-5-melhores-softwares-para-virtualizacao.html>>. Acesso em 25 de Setembro de 2020. Citado na página 20.
- ROSENBLUM, M. The reincarnation of virtual machines. v. 2, p. 34, July 2004. Citado 2 vezes nas páginas 15 e 16.
- SABINO, L. B. L.; CARVALHO, M. *Análise da implantação de Virtual Desktop Infrastructures em laboratórios de informática de uma universidade*. 2017. Citado na página 14.
- STACKPOLE, B. et al. Decentralized virtualization in systems administration education. In: *SIGITE '08: Proceedings of the 9th ACM SIGITE conference on Information technology education*. [S.l.: s.n.], 2008. p. 249–254. Disponível em:<<https://dl.acm.org/doi/10.1145/1414558.1414619>>. Acesso em 12 de Julho de 2020. Citado na página 11.
- STOCKMAN, M. Creating remotely accessible "virtual networks" on a single pc to teach computer networking and operating systems. In: *CITC4 '03: Proceedings of the 4th Conference on information Technology Curriculum*. Lafayette Indiana USA: [s.n.], 2003. p. 67–71. Citado na página 11.
- TAYLOR, C. *Hypervisor Server Review*. 2020. Disponível em:<<https://www.serverwatch.com/virtualization/hypervisor-server/>>. Acesso em 22 de Novembro de 2020. Citado na página 16.
- THAKRAL, A. Effective use of virtualization in computer laboratory management. In: *National Conference on Innovations in IT and Management*. [S.l.: s.n.], 2014. p. 19–22. Disponível em:<http://nci2tm.sinhgad.edu/NCIT2M2014_P/index.html>. Acesso em 26 de Julho de 2020. Citado 2 vezes nas páginas 10 e 17.
- TOSCANI, S. S.; CARISSIMI, A. da S.; OLIVEIRA, R. S. de. *Sistemas Operacionais*. 4. ed. [S.l.]: Bookman, 2010. 7-22 p. Citado 3 vezes nas páginas 14, 15 e 16.
- VIRTUALBOX. *Manual do VirtualBox. Capítulo 5: Armazenamento virtual*. [S.l.], 2022. Disponível em:<<https://www.virtualbox.org/manual/ch05.html>>. Acesso em 13 de Julho de 2022. Citado na página 22.
- WINDOWS, M. *Windows 10*. 2022. Disponível em:<<https://www.microsoft.com/pt-br/software-download/windows10>>. Acesso em 10 de Julho de 2022. Citado na página 31.

Apêndices

APÊNDICE A – Configurando o ambiente

Neste apêndice será ilustrado os requisitos, ferramentas necessárias e como realizar as suas instalações para simular o exposto neste trabalho. Todos os comando abaixo são executados em um terminal (que pode ser aberto com Ctrl+Shit+T) e é necessário a senha do administrador para realizá-los.

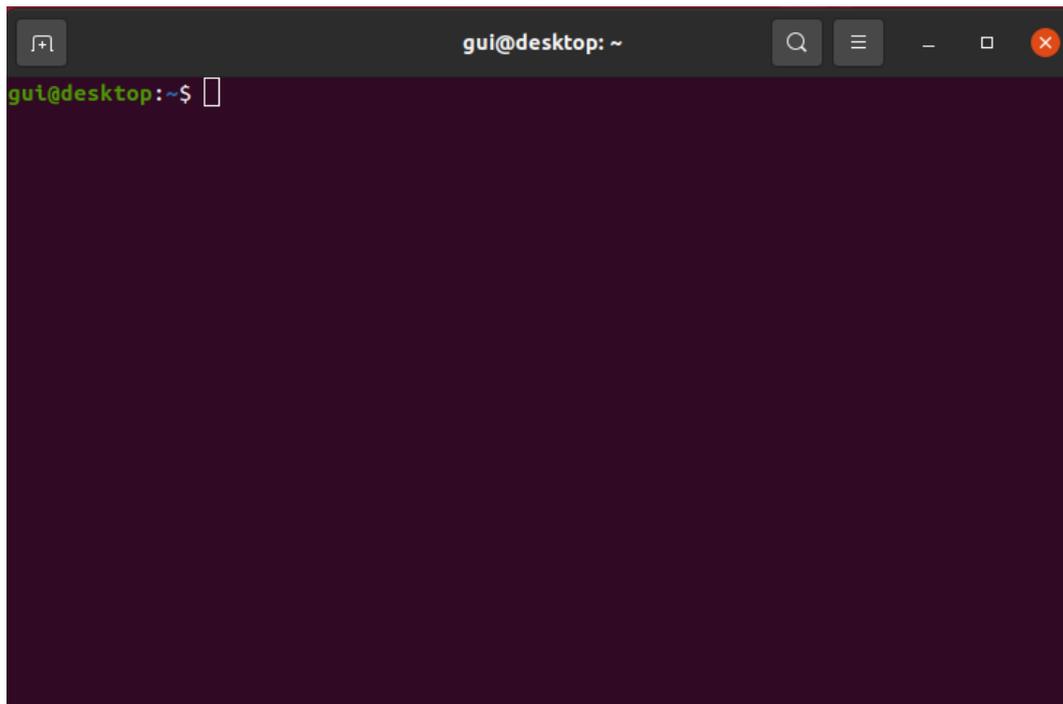


Figura 6 – Terminal do Linux.

A.1 Rsync e OpenSSH

A versão do Ubuntu 20.04 já contém os pacotes do rsync instalado mesmo quando realizada a instalação mínima. Para verificar sua versão entre com o comando:

```
$ sudo rsync -version
```

Se o pacote por alguma razão não estiver instalado então instale com o comando:

```
$ sudo apt install rsync
```

O OpenSSH será necessário para prover maior segurança com a criptografia dos dados. Será necessário a instalação no servidor e no cliente (*host*). Para instalar no servidor entre com o comando:

```
$ sudo apt install openssh-server
```

Para instalar no cliente entre com o comando:

```
$ sudo apt install openssh-client
```

Em seguida, tanto no servidor quanto no cliente entre com comandos para habilitar e iniciar os serviços do ssh:

```
$ sudo systemctl enable ssh
```

```
$ sudo systemctl start ssh
```

Para testar a conectividade digite o comando substituindo o nomeUsuario e IP do servidor:

```
$ ssh nomeUsuario@IP-servidor
```

A.1.1 Criando as Chaves Pública-Privada

Para executar os *scripts* sem a requisição de senha do usuário será necessário a geração das chaves pública-privada para autenticação. No cliente e servidor entre com o seguinte comando:

```
$ ssh-keygen
```

Será solicitado a digitar uma frase secreta para a chave. Esta é uma frase secreta opcional que pode ser usada para criptografar o arquivo de chave privada no disco. Após criar as chaves será necessário importar a chave pública do servidor para o cliente com o comando:

```
$ ssh-copy-id -i ~/.ssh/id_rsa.pub nomeUsuario@IP-servidor
```

A.2 Requisitos de hardware para o VirtualBox

A documentação do Oracle VirtualBox informa a necessidade de no mínimo 512 MB de memória RAM e aproximadamente 30 MB de espaço em disco. As necessidades de memória e processamento crescem conforme o número de sistemas operacionais rodando em conjunto. O Oracle VM VirtualBox Enterprise é a única solução de virtualização de desktop disponível para sistemas operacionais x86, como Microsoft Windows, Linux, Apple MAC OS X e Solaris x86, que oferece a mesma solução em todas as plataformas ([ORACLE, 2021](#)).

A.2.1 Instalando o VirtualBox

Neste método, o VirtualBox será instalado utilizando o repositório de pacotes APT do Ubuntu. O benefício de instalar o VirtualBox usando esse método é que esse método torna a instalação muito fácil e rápida. O lado negativo desse método é que ele não é atualizado quando uma nova versão é lançada.

Para instalar o VirtualBox através do repositório de pacotes APT, primeiro atualize o repositório de pacotes do sistema com o seguinte comando.

```
$ sudo apt update
```

Depois de atualizar o repositório de cache APT do sistema, instale o VirtualBox usando o comando abaixo.

```
$ sudo apt install virtualbox -y
```

Depois de concluir a instalação do VirtualBox, pode-se instalar o pacote de extensão do VirtualBox se precisar dele para suportar dispositivos USB, conectar a webcam *host*, controlar uma máquina virtual remotamente e mais recursos como este. Para instalar o pacote de extensão, digite o comando a seguir.

```
$ sudo apt install virtualbox-ext-pack -y
```

Após este passo, será aberta uma caixa de diálogo para a Configuração do VirtualBox Extension Pack e é necessário concordar com os termos para concluir a instalação. Uma vez que o pacote de extensão tenha sido instalado, você pode começar a usar o VirtualBox procurando por “VirtualBox” no menu “Aplicativos” e clicando no ícone do VirtualBox.

Agora, a versão estável mais recente do VirtualBox deve estar instalada no sistema Ubuntu 20.04 LTS.

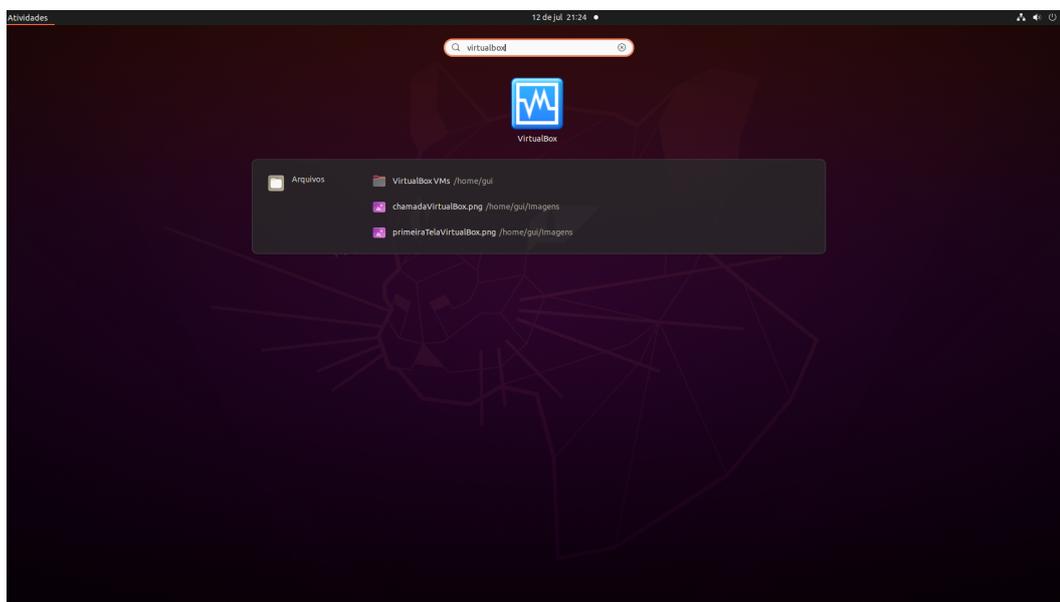


Figura 7 – Pesquisando pelo VirtualBox instalado.



Figura 8 – Tela de boas vindas do VirtualBox.

A.3 Requisitos de hardware para o Nagios

O único requisito para executar o Nagios (core) é uma máquina executando Linux (ou variante UNIX) que tenha acesso à rede e um compilador C instalado (se estiver instalando a partir do código-fonte). Existem requisitos mínimos de hardware adicionais dependendo dos nós/serviços monitorados.

- **Requisitos mínimos:** Processador com pelo menos 1 GHz, 2 GB de memória RAM e 8 GB de HD.
- **Requisitos recomendados:** Processador com 2 GHz ou mais, 2 GB de memória RAM e 40GB de HD.

A.3.1 Instalando o Nagios

1. Instale todos os pacotes necessários.

```
$ sudo apt install wget unzip curl openssl build-essential  
libgd-dev libssl-dev libapache2-mod-php php-gd php apache2 -y
```

2. Carregar arquivos de configuração do Nagios. Para baixar a versão mais recente, visite o site oficial de lançamentos <https://github.com/NagiosEnterprises/nagioscore/releases>.

```
$ wget  
https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
```

3. Extraia os arquivos baixados.

```
$ sudo tar -zxvf nagios-4.4.6.tar.gz
```

4. Navegue até o diretório de configuração.

```
$ cd nagios-4.4.6
```

5. Execute o script de configuração do Nagios.

```
$ sudo ./configure
```

6. Compile o programa principal e os CGIs.

```
$ sudo make all
```

7. Crie e instale o grupo e o usuário.

```
$ sudo make install-groups-users
```

8. Adicione o usuário de diretórios www-data ao grupo nagios.

```
$ sudo usermod -a -G nagios www-data
```

9. Instale o Nagios.

```
$ sudo make install
```

10. Inicialize todos os scripts de configuração de instalação.

```
$ sudo make install-init
```

11. Instale e configure as permissões no diretório de configuração.

```
$ sudo make install-commandmode
```

12. Instale os arquivos de configuração de amostra.

```
$ sudo make install-config
```

13. Instale os arquivos apache.

```
$ sudo make install-webconf
```

14. Ative o modo de reescrita do apache.

```
$ sudo a2enmod rewrite
```

15. Ative a configuração CGI.

```
$ sudo a2enmod cgi
```

16. Reinicie o serviço Apache.

```
$ sudo systemctl restart apache2
```

17. Crie um usuário e defina a senha quando solicitado.

```
$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

A.3.2 Instalando o Nagios Plugins

1. Baixe o plug-in do Nagios. Para baixar os plugins mais recentes, visite a página de download de plugins <https://nagios-plugins.org/download/>.

```
$ cd / && wget  
https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
```

2. Extraia o plugin baixado.

```
$ sudo tar -zxvf nagios-plugins-2.3.3.tar.gz
```

3. Navegue até o diretório dos plugins.

```
$ cd nagios-plugins-2.3.3/
```

4. Execute o script de configuração do plug-in.

```
$ sudo ./configure --with-nagios-user=nagiosadmin  
--with-nagios-group=nagios
```

5. Compile os plugins do Nagios.

```
$ sudo make
```

6. Instale os plugins.

```
$ sudo make install
```

A.3.3 Verifique a Configuração do Nagios

1. Verifique a configuração do Nagios.

```
$ sudo /usr/local/nagios/bin/nagios -v  
/usr/local/nagios/etc/nagios.cfg
```

2. Inicie o serviço Nagios.

```
$ sudo systemctl start nagios
```

3. Habilite o serviço Nagios para ser executado na inicialização do sistema.

```
$ sudo systemctl enable nagios
```

A.3.4 Acesse a interface da Web do Nagios

Abra seu navegador web e acesse a interface web do Nagios através da URL `http://ServerIP/nagios`. Por exemplo:

```
http://192.0.2.10/nagios
```

Para fazer login, use `nagiosadmin` como seu nome de usuário e a senha definida durante a etapa de criação da conta de usuário como senha. Agora é possível acessar o painel e começar a configurar o Nagios.

A.3.5 Adicione *hosts* remotos

Para monitorar *hosts*, precisamos adicioná-los ao Nagios. Por padrão, o Nagios monitora apenas o `localhost` (o servidor em que está sendo executado). Vamos adicionar *hosts* que fazem parte da nossa rede para ganhar ainda mais controle. Você precisará usar as instruções a seguir em todos os *hosts* que deseja monitorar.

1. Primeiro, instale `nagios-plugins` e `nagios-nrpe-server`

```
$ sudo apt-get install nagios-plugins nagios-nrpe-server
```

2. Configure NRPE: Em seguida, abra o arquivo `/etc/nagios/nrpe.cfg`. Substitua o valor de `allowed_hosts` por `127.0.0.1,ServerIP` substituindo o segundo IP pelo endereço IP do servidor Nagios.
3. Agora vamos abrir o arquivo `/etc/nagios/nrpe.cfg` e substituir alguns valores.
 - Substitua o valor de `server_address` pelo endereço IP privado do *hosts*.
 - Defina `allowed_hosts` para o endereço IP privado do seu servidor Nagios.
4. Reinicie o NRPE

```
$ service nagios-nrpe-server restart
```

5. Adicione o *hosts* ao Nagios: Agora que configuramos o *hosts* que vamos monitorar, precisamos voltar ao nosso servidor Nagios e adicionar o *hosts* a ele. Abra o seguinte arquivo com seu editor favorito:

```
$ nano /usr/local/nagios/etc/servers/host.cfg
```

6. Use o bloco a seguir como modelo. Substitua `nome-host` por um nome apropriado para seu *host* remoto e atualize os valores de `host_name`, `alias` e endereço IP de acordo.

Código 5 – Exemplo de *host* do Nagios

```
1  define host{
2      use                linux-server
3      host_name          nome-host
4      alias              nagios-client
5      address            IP-Host
6  }
7  define service {
8      use                generic-service
9      host_name          nome-host
10     service_description Users
11     check_command      check_nrpe!check_users
12 }
13 define service {
14     use                generic-service
15     host_name          nome-host
16     service_description CPU load
17     check_command      check_nrpe!check_load
18 }
19 define service {
20     use                generic-service
21     host_name          nome-host
22     service_description Disk /
23     check_command      check_nrpe!check_sda1
24 }
25 define service {
26     use                generic-service
27     host_name          nome-host
28     service_description Zombie Processes
29     check_command      check_nrpe!
30     check_zombie_procs
31 }
32 define service {
33     use                generic-service
34     host_name          nome-host
35     service_description Total Processes
36     check_command      check_nrpe!
37     check_total_procs
38 }
```

7. Isso permitirá que você simplesmente monitore se o servidor está ativo ou inativo. Agora recarregue o Nagios.

```
$ service nagios reload
```

A.4 Instalando o Systemback

Execute o seguinte comando para importar a chave de assinatura PGP deste PPA. Isso verifica a assinatura do gerenciador de pacotes.

```
$ sudo apt-key adv -keyserver keyserver.ubuntu.com -recv-keys  
382003C2C8B7B4AB813E915B14E4942973C62A1B
```

Adicione o PPA com o comando.

```
$ sudo add-apt-repository "deb  
http://ppa.launchpad.net/nemh/systemback/ubuntu xenial main"
```

Por fim, atualize e instale Systemback no Ubuntu 20.04 LTS.

```
$ sudo apt update
```

```
$ sudo apt install systemback
```

A.4.1 Como utilizar o Systemback

Abra o Systemback e digite a senha de usuário do sistema. Para criar um CD/DVD/USB Live clique em “Criar sistema Live” mostrado na Figura 9. Em seguida defina o diretório de trabalho e o nome do novo sistema Live, marque a opção “Incluir arquivos de dados do usuário” para manter todos os dados do usuário. Por fim clique em “Criar nova” para iniciar o processo de criação da Live como mostra a Figura 10 e aguarde o término da operação como mostra a Figura 11.

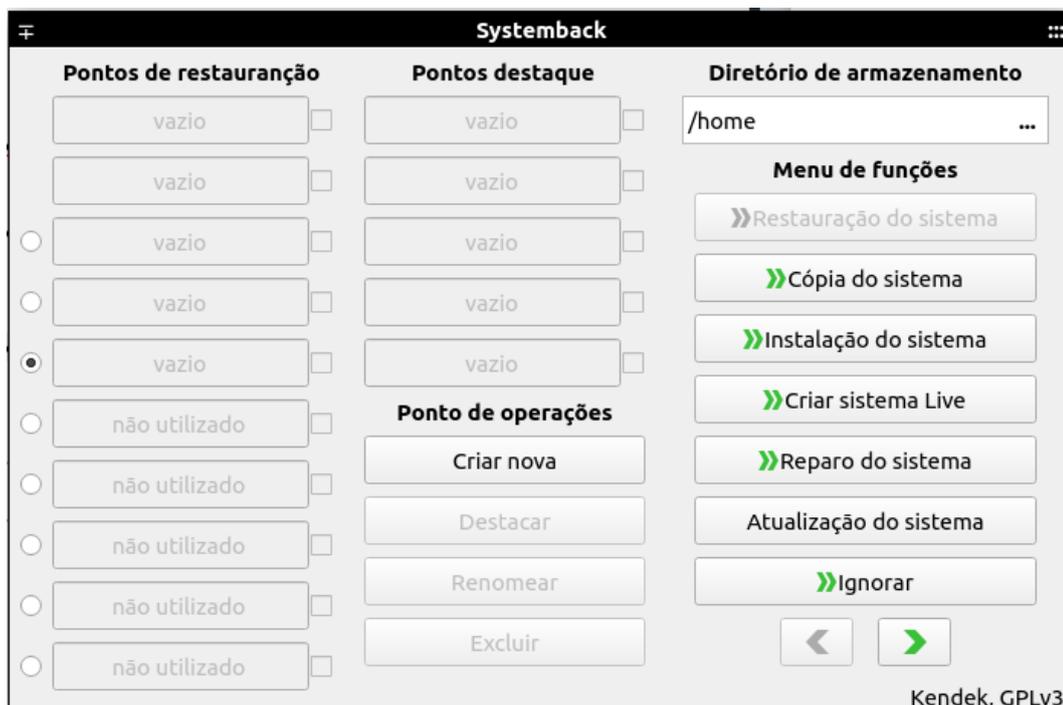


Figura 9 – Tela de opções do Systemback.

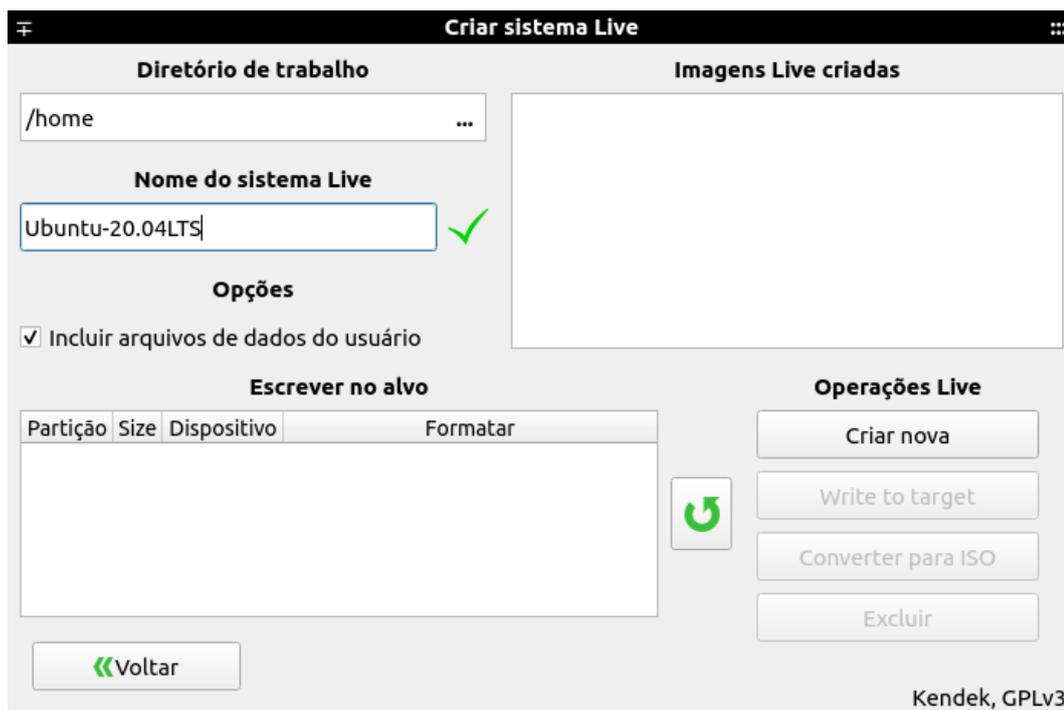


Figura 10 – Tela de Criar Sistema Live.

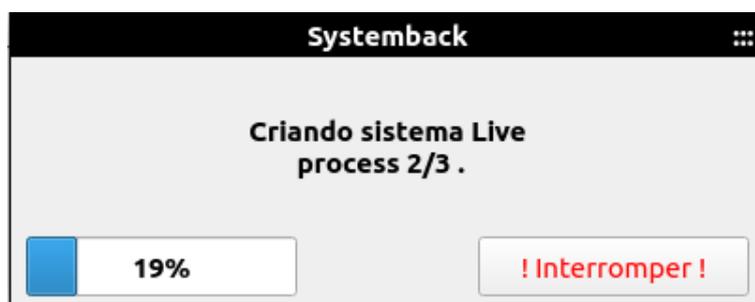


Figura 11 – Criando Sistema Live.

Após o término do processo a imagem será criada como `.sblive` conforme mostrado na Figura 12.

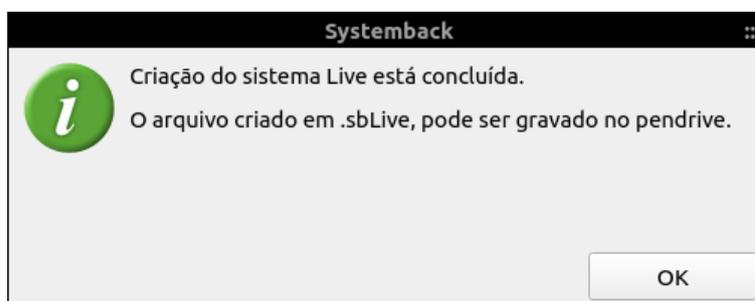


Figura 12 – Fim da criação do Sistema Live.

Depois que um arquivo `.sblive` é criado, o Systemback permite gravar o sistema diretamente

em uma unidade flash USB. Para isso, insira um pendrive e clique no botão “Green Refresh”. A caixa “Write target” deve exibir o pendrive. No exemplo a seguir é "SanDisk Cruzer Blade" como mostra a Figura 13.

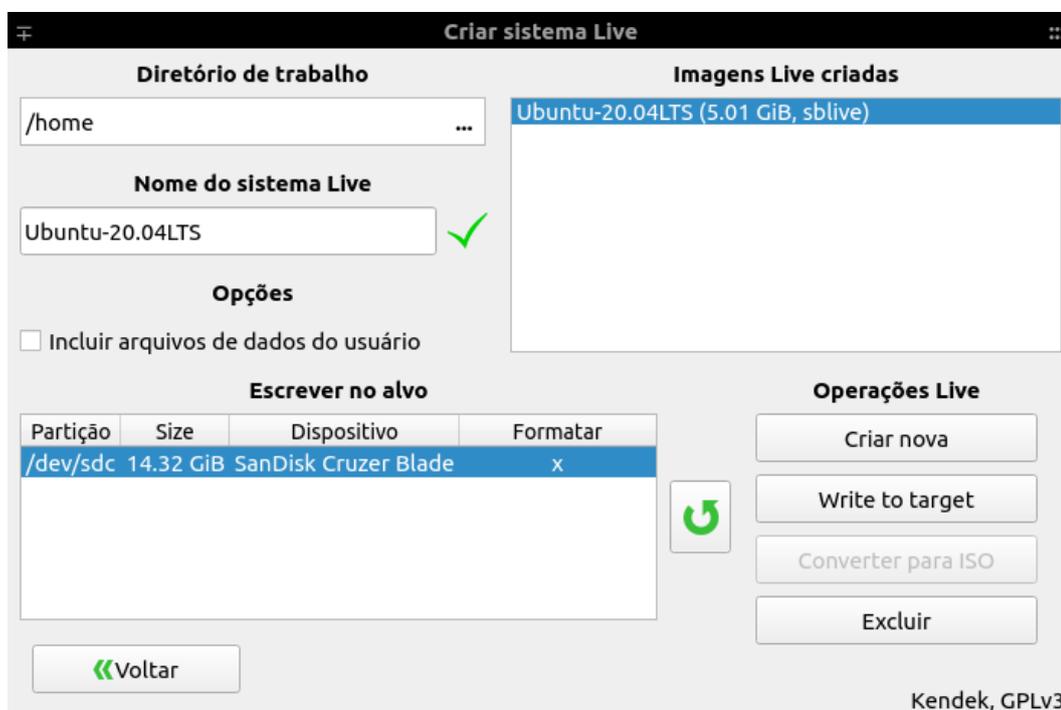


Figura 13 – Criando flash USB.

Haverá diferentes “Operações Live” disponíveis para aplicar a esta imagem.

- Clique em “Escrever no destino”. Você verá a caixa de diálogo de configuração, clique em Iniciar. Esta operação grava a nova imagem em sua unidade flash e uma barra de progresso é exibida.
- Clique em “Converter para ISO” se quiser criar um arquivo iso inicializável a partir da imagem .sblive criada. Porém a imagem não pode ser convertida em iso se o arquivo .sblive for maior que 4 GB.
- Você também pode “Excluir” a imagem .sblive e recriar a nova imagem clicando em “Criar nova”.

O processo de instalação da Live personalizada criada pelo Systemback é similar à instalação padrão do Ubuntu, apesar de sua interface ser pouco diferente ela segue os mesmos preceitos.

A.5 Configuração da BIOS RTC

Para programar o computador para iniciar através de um horário definido pela BIOS deverá ser seguido os seguintes passos:

1. Acessar a BIOS (tecla DELETE ou F2 ao ligar a máquina).
2. Entrar no “Modo Avançado”.
3. Alternar para a aba “Avançado”.
4. Configurações APM.
5. Habilitar “Ligar por RTC” e definir o horário.
6. Salvar e sair.

Estes passos podem variar de acordo com a placa-mãe da máquina.