UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS DE TELECOMUNICAÇÕES

ALEXANDRE DE OLIVEIRA BRIXEL
FILIPE TIAGO CORREIA MUNIZ DE RESENDE

VIRTUALIZAÇÃO DE SERVIDOR COM BASE EM PEQUENA EMPRESA

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2018

ALEXANDRE DE OLIVEIRA BRIXEL
FILIPE TIAGO CORREIA MUNIZ DE RESENDE

VIRTUALIZAÇÃO DE SERVIDOR COM BASE EM PEQUENA EMPRESA

Trabalho de Conclusão de Curso de Graduação, apresentado ao Curso Superior de Tecnologia em Sistemas de Telecomunicações, do Departamento Acadêmico de Eletrônica – DAELN, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Kleber Kendy Horikawa Nabas

CURITIBA
2018

TERMO DE APROVAÇÃO

ALEXANDRE DE OLIVEIRA BRIXEL
FILIPE TIAGO CORREIA MUNIZ DE RESENDE

VIRTUALIZAÇÃO DE SERVIDOR COM BASE EM PEQUENA EMPRESA

Este trabalho de conclusão de curso foi apresentado no dia 13 de dezembro de 2018, como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Telecomunicações, outorgado pela Universidade Tecnológica Federal do Paraná. Os alunos foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

_____
Prof. Dra. Tânia Lúcia Monteiro
Coordenadora de Curso
Departamento Acadêmico de Eletrônica

_____
Prof. M.Sc. Sérgio Moribe
Responsável pela Atividade de Trabalho de Conclusão de Curso
Departamento Acadêmico de Eletrônica

BANCA EXAMINADORA

_____
Prof. Omero Francisco Bertol
UTFPR

_____
Prof. Dr. Joilson Alves Junior
UTFPR

_____
Prof. Dr. Kleber Kendy Horikawa Nabas
Orientador - UTFPR

"A Folha de Aprovação assinada encontra-se na Coordenação do Curso"

AGRADECIMENTOS

Agradecemos aos nossos pais, namorada, noiva e amigos por nos dar o apoio necessário, por acreditar na gente e por terem paciência enquanto estivemos focados com a realização deste projeto.

À Universidade Tecnológica Federal do Paraná, Campus Curitiba, ao Departamento Acadêmico de Eletrônica desta Universidade e a todos os nossos Professores.

À Prof. Dra. Tânia Lúcia Monteiro que fez o máximo possível para que pudéssemos entregar esse projeto, ao Prof. Dr. Kleber Kendy Horikawa Nabas que mesmo após tantos anos continuou nos prestando ajuda, nos auxiliando e, principalmente, não nos deixando desistir de completar mais esta etapa importantíssima em nossas vidas, que até que enfim, se realizou.

E também aos Professores da Banca que acompanharam todo nosso projeto e nos deram a honra de poder finalizar. A todos, o nosso muito obrigado.

RESUMO

BRIXEL, Alexandre de Oliveira; RESENDE, Filipe Tiago Correia Muniz de. **Virtualização de servidor com base em pequena empresa**. 2018. 38 p. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Sistemas de Telecomunicações), Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

O objetivo deste trabalho é otimizar a rede interna da Produtora de Áudio CANJA Produções Musicais de forma rápida, funcional, segura, de fácil manutenção e com baixo custo. Entre os procedimentos a serem aplicados, estão a virtualização dos servidores utilizando o sistema operacional Linux, o *software* Citrix XenServer e configuração dos serviços, de forma que otimize espaço físico e *hardware* de baixo custo. Os serviços que serão instalados são: DHCP, FTP, *Proxy*, DNS e *Firewall*.

Palavras chave: Máquina Virtual. Servidor. Rede. Segurança.

ABSTRACT

BRIXEL, Alexandre de Oliveira; RESENDE, Filipe Tiago Correia Muniz de. **Server virtualization for small business**. 2018. 38 p. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Sistemas de Telecomunicações), Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

The objective of this work is to optimize the internal network of the Music Producer Company CANJA Produções Musicais in a fast way, functional, safe, easy maintenance and with low cost. Among the procedures to be applied are server virtualization using the Linux operating system, Citrix XenServer software, and service configuration, to optimize physical space and low-cost hardware. The services that will be installed are: DHCP, FTP, Proxy, DNS and Firewall.

Keywords: Virtual Machine. Server. Network. Safe.

LISTA DE FIGURAS

LISTA DE SIGLAS

DAELN	Departamento Acadêmico de Eletrônica
DHCP	*Dynamic Host Configuration Protocol*
DNS	*Domain Name System*
FTP	*File Transfer Protocol*
IP	*Internet Protocol*
NTP	*Network Time Protocol*
SMB	*Server Message Block*
SR	*Storage Repository*
SSH	*Secure Shell*
UFW	*Uncomplicated Firewall*
UTFPR	Universidade Tecnológica Federal do Paraná
VLSM	*Variable Length Subnet Mask*
VM	*Virtual Machine*
VMM	*Virtual Machine Monitor*

SUMÁRIO

1 INTRODUÇÃO

Hoje em dia, cerca de 97% das empresas no Brasil estão informatizadas, isto se deve a grande facilidade que a tecnologia traz para otimizar quase todos os processos de uma empresa. No ato da instalação de uma rede conectada a internet, há necessidade de criação de um bom sistema de segurança, para que todas as informações da empresa estejam em segurança. Da mesma forma que a comunicação de dados facilita, pessoas mal-intencionadas utilizam de mecanismos para furtar informações e dados importantes. Existem várias ferramentas que podem ser utilizadas para proteger a rede, tais como: Servidores *Firewall*, *Proxy* e DNS. Uma empresa também necessita de um servidor de e-mail e uma centralização dos dados.

Neste trabalho, irá ser projetado a instalação de alguns serviços de informatização e proteção em um único servidor com *software* livre utilizando máquina virtual. A ideia de utilizar uma máquina virtual vem de otimizar os custos da empresa com equipamentos, manutenção e energia.

Dentre os serviços que serão instalados estão: DHCP, FTP, *proxy*, DNS, *firewall*.

1.1 PROBLEMA

Para uma pequena empresa que acaba de se informatizar, o simples acesso a internet não basta, tem que existir uma rede interna onde todos os computadores estejam devidamente funcionando dentro dela. Todos os arquivos necessários para o trabalho precisam estar centralizados para que todas as estações de trabalho tenham acesso aos arquivos. Por exemplo: Em uma empresa que trabalha com produção de áudio, todas as pessoas que estão trabalhando em uma produção precisam acessar os mesmos arquivos e projetos para o *"workflow"* funcionar, porém as pessoas que trabalham com produção não devem ter acesso aos dados e arquivos do departamento financeiro da empresa. Ao mesmo tempo, todos os arquivos devem estar protegidos de invasões para não serem furtados, apagados e informações confidenciais da empresa vazarem.

Diante dos problemas expostos, é de grande importância que um sistema altamente seguro seja implementado à informatização da empresa. Pode-se também

questionar: Quais sistemas de segurança poderão ser aplicados? Visto que estamos tratando de uma empresa de pequeno porte, qual a forma mais econômica de implantar um sistema seguro? Para otimizar e economizar energia e equipamentos, quais serviços podemos incluir na centralização da informatização da empresa?

Para expor de forma mais concreta, será exposto um estudo sobre soluções de virtualização de um servidor. Neste servidor incluiremos DHCP, FTP, *proxy*, DNS, *firewall*.

1.2 JUSTIFICATIVA

Baseado nos problemas descritos, será proposta uma solução de virtualização de servidores aconselhado para empresas com pouco espaço físico e/ou que desejam centralizar o gerenciamento de seus serviços. Essa solução possui muitos recursos e tem interface e configuração simples, podendo de forma fácil configurar vários recursos e servidores.

Este projeto visa à implantação de um servidor virtual utilizando Citrix XenServer, onde será virtualizado um servidor Ubuntu, que contará com os seguintes servidores de serviço:

- Protocolo DHCP para configuração dinâmica de endereço IP nos terminais que irão se conectar ao servidor;
- Protocolo DNS para fazer o gerenciamento e tradução de nomes a partir de um endereço IP;
- Protocolo de transferência de arquivos FTP para controle dos arquivos entre terminais e servidor;
- Servidor *Proxy* para controle e distribuição da internet na rede interna;
- Servidor *Firewall* para controle do fluxo de dados que são transmitidos dentro da rede interna assim como os que são recebidos ou transmitidos de fontes externas.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Através de resultados de estudos das soluções técnicas, será apresentada uma solução que garanta que todas as estações de trabalho terão acesso aos devidos dados e arquivos de forma segura, utilizando um servidor instalado em uma máquina virtual, assim tendo um custo de implantação extremamente baixo e confiável.

1.3.2 Objetivos Específicos

- Apresentar a real necessidade de a empresa ter uma rede segura.
- Expor vulnerabilidades que a tecnologia pode trazer.
- Descrever características, conceitos e a importância de um servidor em empresas informatizadas.
- Abordar a economia que um servidor virtual trará para o projeto.
- Configurar o servidor com os serviços DHCP, FTP, *Proxy*, DNS e *Firewall*.
- Simular um ambiente de rede funcional com o servidor em total funcionamento.

1.4 PROCEDIMENTOS METODOLÓGICOS

A implantação do projeto será realizada por meio de guias, exemplos práticos, manuais, normas e bibliografias que abordam o tema.

O projeto será desenvolvido em quatro etapas. Na primeira etapa, serão abordadas as reais necessidades da implementação do tema escolhido. Em seguida, serão apresentadas pesquisas dos serviços que serão utilizados no servidor e o porquê de ele ser um servidor virtual.

Na segunda etapa será apresentado a fundo o que é cada serviço que será instalado no servidor e qual a função e como ele pode beneficiar o ambiente de rede.

A terceira etapa será representada pela configuração do servidor virtual, os serviços DHCP, FTP, *proxy*, DNS e *firewall*, assim como, serão configuradas as particularidades necessárias. Após as instalações e configurações, os processos realizados serão testados no servidor. Será criado um ambiente de rede de pequeno

porte para simular e testar os funcionamentos do servidor virtual e seus serviços em estações de trabalho diferentes.

A quarta e última etapa visa vincular o conhecimento obtido nas simulações e testes ao conhecimento teórico, para demonstrar os reais benefícios e uma possível aplicação do projeto.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 MÁQUINAS VIRTUAIS

O conceito de máquinas virtuais surgiu pela década de 1960 pela IBM, com a proposta de aproveitamento de toda capacidade dos *mainframes*, assim se tornando uma alternativa à compra de equipamentos e redução de custos.

Algum tempo depois, com a redução dos preços, a virtualização foi se tornando menos utilizada por ser mais vantajoso usar o modelo de computação distribuída, onde existem vários computadores clientes interligados a um servidor.

Após isso, houve um crescente nas aquisições de serviço de virtualização pela redução de custos em aquisição de servidores e a economia de recursos em aquisição de equipamentos de infraestrutura.

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 (SIQUEIRA, 2008, p. 91).

Com as aplicações rodando em máquinas virtuais, é possível em caso de falha de algum ambiente que outro seja utilizado como recurso de contingência. Mediante de *softwares* apropriados, pode-se mover estações virtuais para outro hardware sem perda de produtividade (MARAN, 2008).

Apesar de rodarem sobre o mesmo hospedeiro, as máquinas virtuais são independentes umas das outras, comportando-se da mesma forma que um computador real. A quantidade de máquinas virtuais que uma máquina física pode suportar depende, além do limite estabelecido pelo *software* de virtualização, de sua estrutura de *hardware* e de sua capacidade de processamento (SARDINHA, 2009).

Os sistemas computacionais tradicionais, de uma forma básica, são projetados com três componentes: o *hardware*, o sistema operacional e as aplicações, o *hardware* executa as operações solicitadas pelas aplicações. O sistema operacional recebe as solicitações das operações por meio das chamadas de sistemas e controla o *hardware*.

Utilizando Máquinas Virtuais, permite que que diferentes aplicações de diferentes plataformas, antes incompatíveis entre si, executem ao mesmo tempo em um mesmo *hardware*, o responsável por criar e gerenciar os ambientes é o monitor de máquinas virtuais (*Virtual Machine Monitor* - VMM) que implementa uma camada de virtualização, interpretando e emulando o conjunto de instruções entre as máquinas virtuais e a máquina real.

2.2 VIRTUAL MACHINE MONITOR (VMM)

O *hypervisor* (VMM) aloca recursos para os hosts hóspedes, isolando-os uns dos outros, além de preparar as máquinas virtuais para rodar de forma eficiente. Por realizar a gestão dos recursos o *hypervisor* trabalha em alto nível de prioridade (BUENO, 2009).

De acordo com Silva (2007), definir o ambiente de máquinas virtuais, alterar o modo de execução do sistema operacional entre privilegiado e não privilegiado, escalonar o uso da CPU para as máquinas virtuais, intermediar as chamadas de sistema e controlar o acesso a dispositivos, são funções básicas de um monitor de máquinas virtuais.

Segundo Popek e Goldberg (1974), é importante salientar que na implementação de um VMM deve-se levar em conta características como compatibilidade, desempenho e simplicidade. A compatibilidade é importante para a execução do legado de *software*. O desempenho é de extrema importância para a execução do sistema operacional e aplicações na máquina virtual. A simplicidade é buscada para evitar falhas no monitor, que causaria problemas para todas as máquinas virtuais em execução. As principais características de um monitor de máquinas virtuais devem ser:

- Eficiência: é extremamente importante que um grande número de instruções do processador virtual seja executado diretamente pelo processador real, sem que haja intervenção do monitor. As instruções que não puderem ser tratadas pelo processador real precisam ser tratadas pelo monitor.
- Integridade: todas as requisições aos recursos de *hardware* devem ser alocadas explicitamente pelo monitor (memória, processamento etc).

- Equivalência: o monitor deve prover um comportamento de execução semelhante ao da máquina real para o qual ele oferece suporte de virtualização, salvo haja a necessidade de se fazer alterações na disponibilidade de recursos da máquina.

2.3 TÉCNICAS DE VIRTUALIZAÇÃO

Podemos utilizar duas técnicas de implementação de virtualização, a Virtualização total e a para-virtualização:

- Virtualização total oferece uma camada de abstração de todos os componentes físicos e lógicos de baixo nível, criando dessa forma um ambiente virtual em que o sistema operacional visitante é executado. Não é necessária nenhuma alteração no sistema operacional hóspede, pois ele percebe a máquina virtual como se fosse uma máquina física;
- A para-virtualização é uma alternativa à virtualização total. Nesse modelo de virtualização, o sistema operacional é modificado para chamar o VMM sempre que executar uma instrução que possa alterar o estado do sistema, uma instrução sensível. Isso acaba com a necessidade de o VMM testar instrução por instrução, o que representa um ganho significativo de desempenho. Outro ponto positivo da para-virtualização é que os dispositivos de *hardware* são acessados por *drivers* da própria máquina virtual, não necessitando mais do uso de *drivers* genéricos que inibiam o uso da capacidade total do dispositivo.

Os inconvenientes de cada técnica é que na total virtualização é difícil implementar uma máquina virtual que imite o comportamento exato de cada dispositivo e que devido ao sistema operacional hóspede não ser modificado, suas instruções devem ser testadas pelo monitor de máquinas virtuais, o que representa um custo de processamento, enquanto no para-virtualização, a desvantagem é a redução da portabilidade, devido ao fato de exigir que o sistema a ser virtualizado seja modificado.

Portanto, tendo em vista as técnicas de virtualização, a decisão de qual melhor a técnica de virtualização para um dado ambiente está intimamente ligada à qual o processador da máquina física que vai hospedar as virtuais, bem como se o

processador possui ou não uma extensão no seu conjunto de instruções que suporte a virtualização.

3 CONFIGURAÇÕES NECESSÁRIAS

Para configurar o projeto, instalou-se os *softwares* a seguir: Citrix XenServer, responsável pelo gerenciamento das máquinas virtuais, o XenCenter que é a ferramenta necessária para gerenciar as máquinas virtuais dentro do XenServer, e o sistema operacional Ubuntu, o qual foi usado para instalar os serviços DHCP, DNS, FTP, *Firewall* e *Proxy*. A seguir iremos descrever cada ferramenta e serviço, assim como os passos para instalação e configuração.

3.1 CITRIX XENSERVER

O Citrix XenServer é o sistema operacional responsável por gerenciar as máquinas virtuais, é recomendado uma máquina dedicada para o XenServer para garantir o funcionamento dos servidores, com o XenServer é possível instalar e configurar várias máquinas virtuais dentro do mesmo *hardware*, economizando espaço físico, energia e facilitando o gerenciamento.

Escolheu-se o Citrix pois é o muito utilizado em empresas, com muitos profissionais com experiência na ferramenta e convenientemente a licença educacional é gratuita.

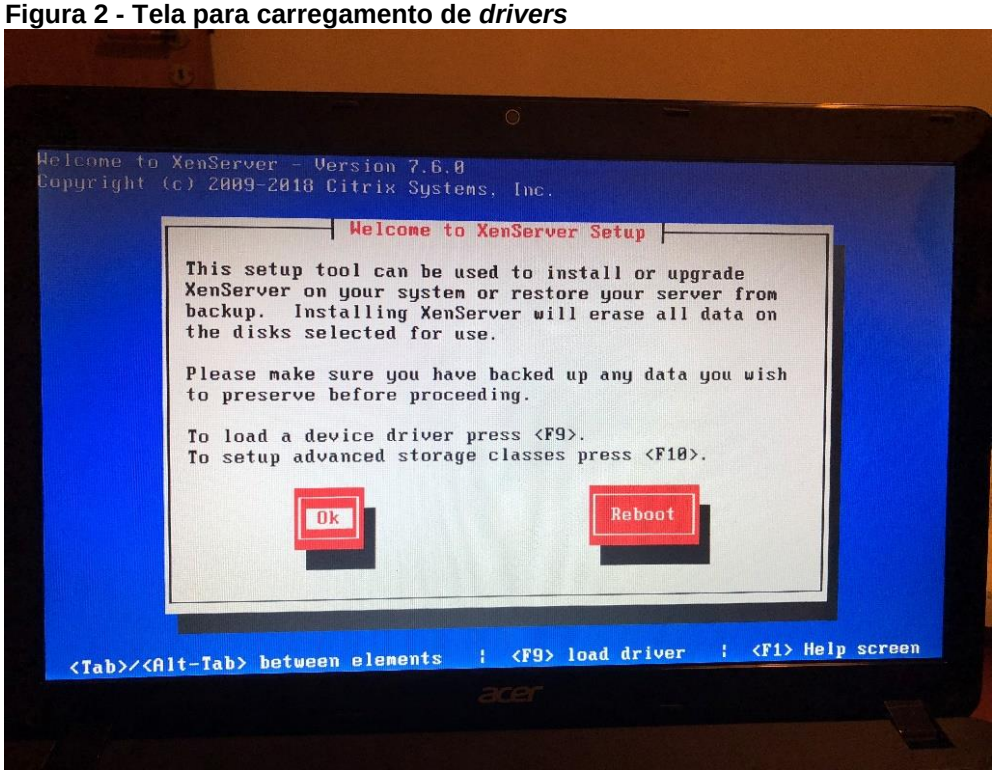
Após preparar o computador e gravar o instalador do XenServer num *pendrive*, encontrou-se pronto para configurá-lo. Na Figura 1, tem-se a tela de início para a instalação.

Figura 1 - Tela de início da instalação do XenServer

Fonte: Autoria própria.

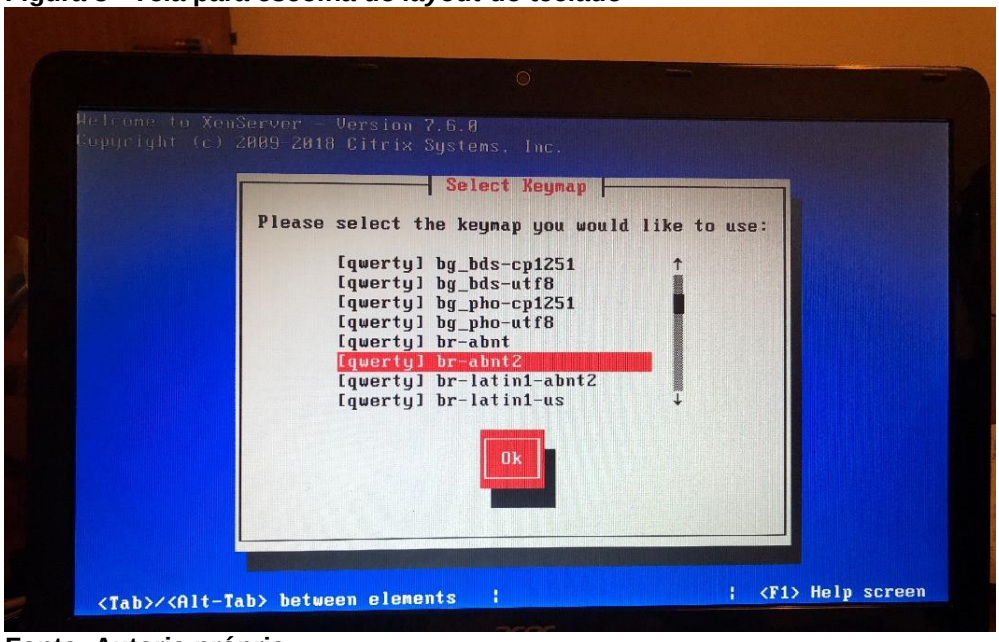
Na Figura 2: Tela para carregamento de *drivers*; Figura 3: Tela para escolha do layout do teclado; Figura 4: Tela para escolha do disco para instalação do XenServer; e Figura 5: Escolha de onde os arquivos de instalação serão instalados; mostram-se os principais passos para instalar o XenServer com as configurações básicas.

Figura 2 - Tela para carregamento de *drivers*



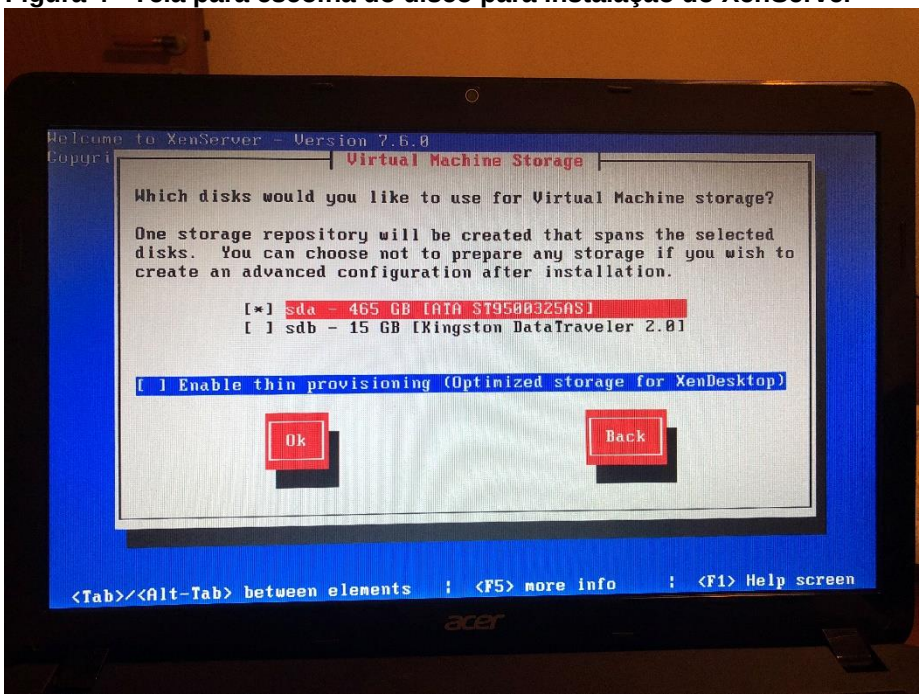
Fonte: Autoria própria.

Figura 3 - Tela para escolha do *layout* do teclado



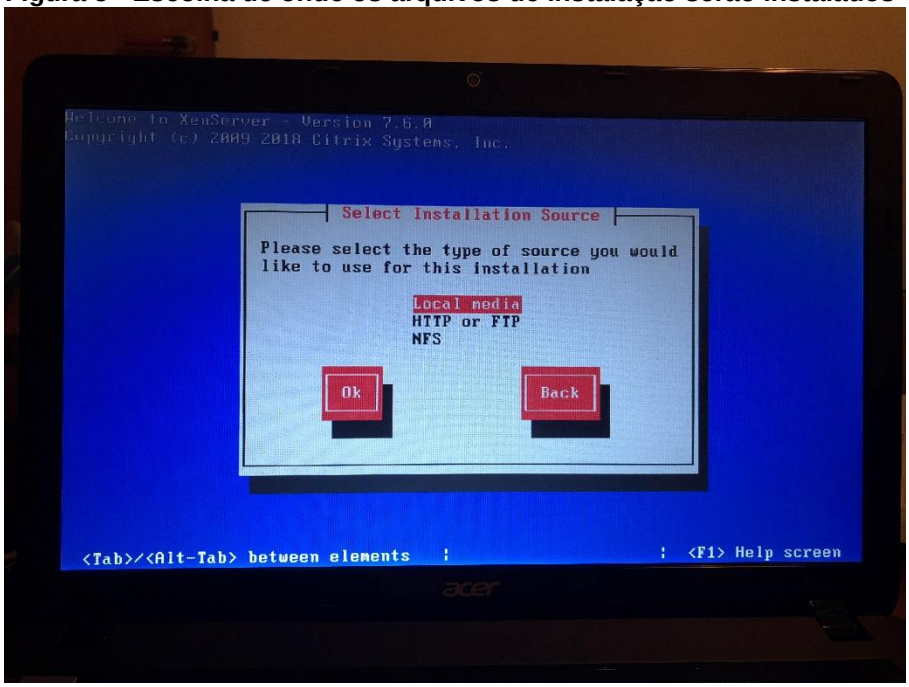
Fonte: Autoria própria.

Figura 4 - Tela para escolha do disco para instalação do XenServer



Fonte: Autoria própria.

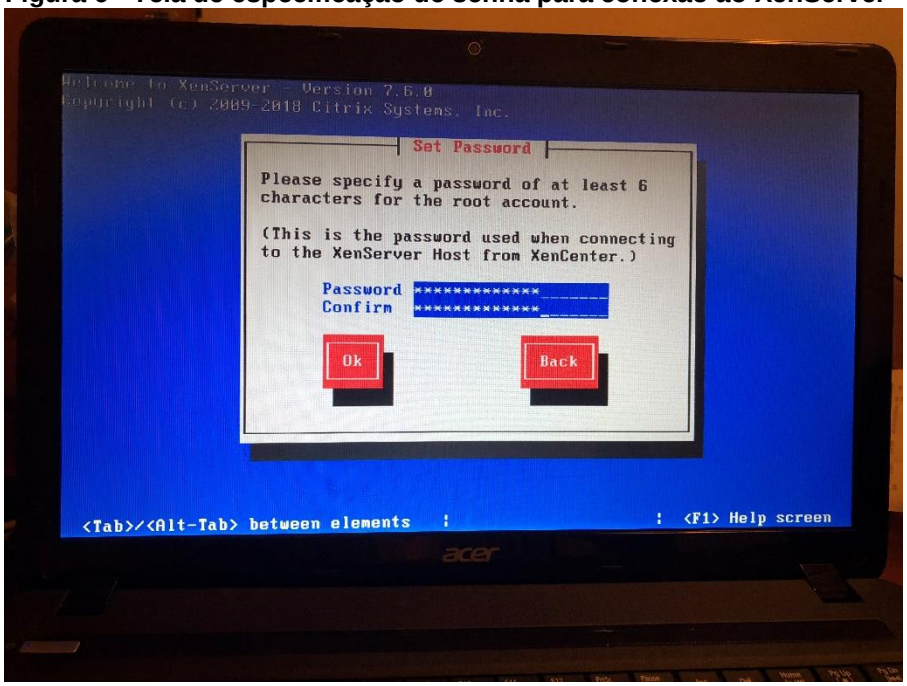
Figura 5 - Escolha de onde os arquivos de instalação serão instalados



Fonte: Autoria própria.

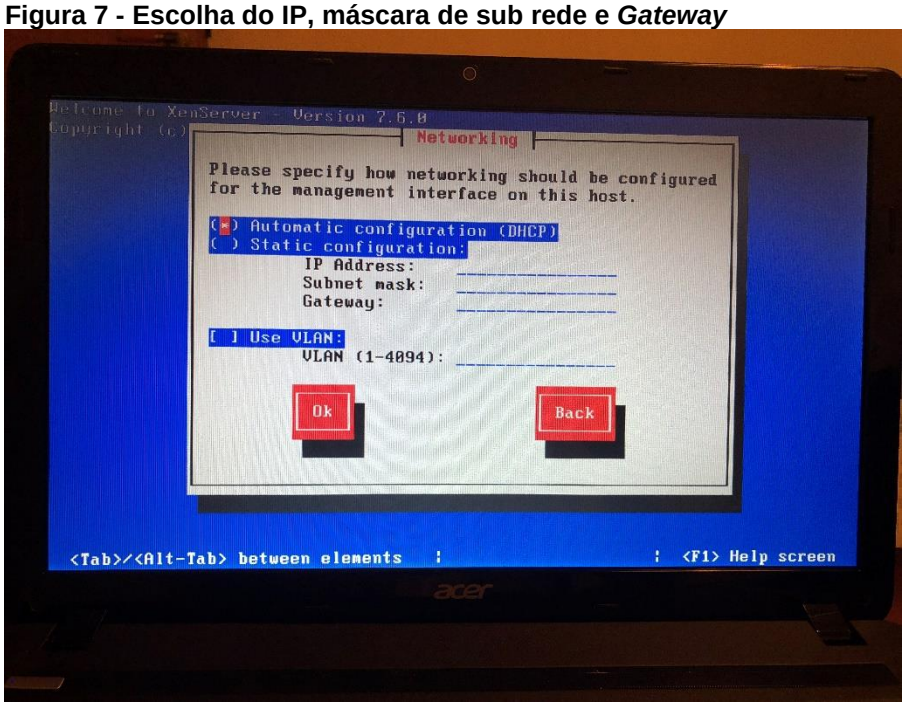
Nos passos a seguir, configurou-se a senha e a interface de rede para a comunicação com o XenCenter. Na Figura 6, tem-se a tela para determinar a senha que utilizou-se para o XenCenter acessar o XenServer, enquanto na Figura 7, tem-se a tela de configuração de IP. Deve ser utilizado um IP dentro da sub rede do XenCenter para que os dois computadores se enxerguem.

Figura 6 - Tela de especificação de senha para conexão ao XenServer



Fonte: Autoria própria.

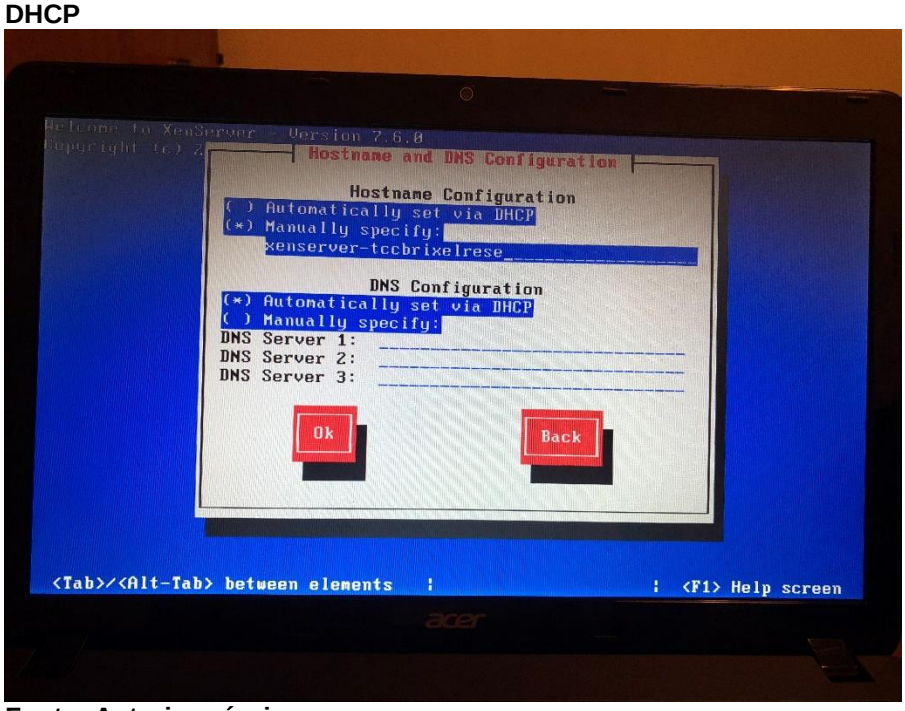
Figura 7 - Escolha do IP, máscara de sub rede e *Gateway*



Fonte: Autoria própria.

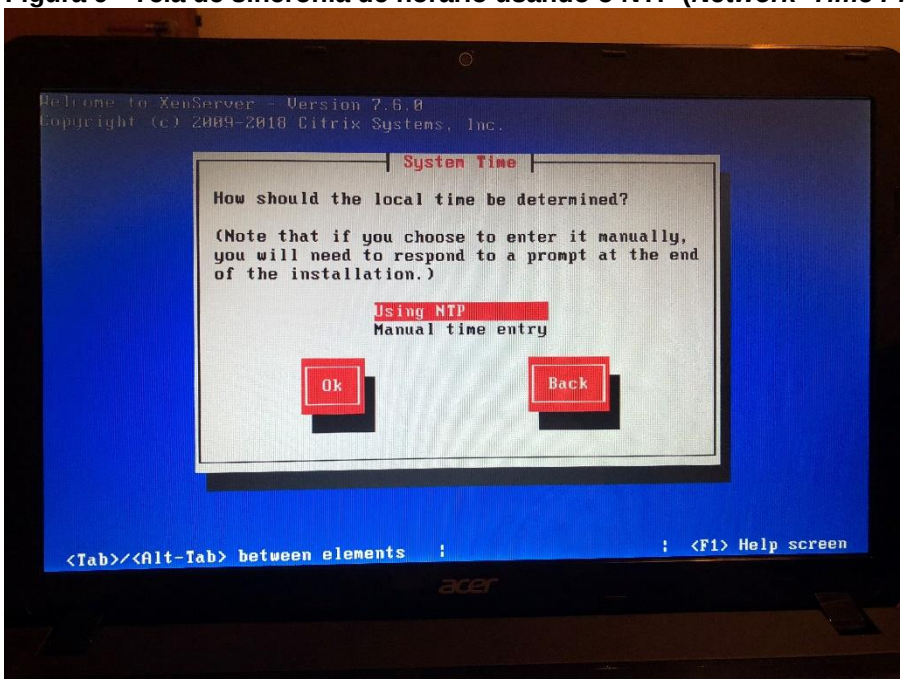
A partir desse ponto, configurou-se o *hostname*, onde foi atribuído como “xenserver-tccbrixelrese” e o DNS automático, conforme indicado na Figura 8. Na Figura 9, foi feita a escolha da região de sincronia de horário.

Figura 8 - Tela de especificação do *hostname*, o DNS será configurado automaticamente via DHCP



Fonte: Autoria própria.

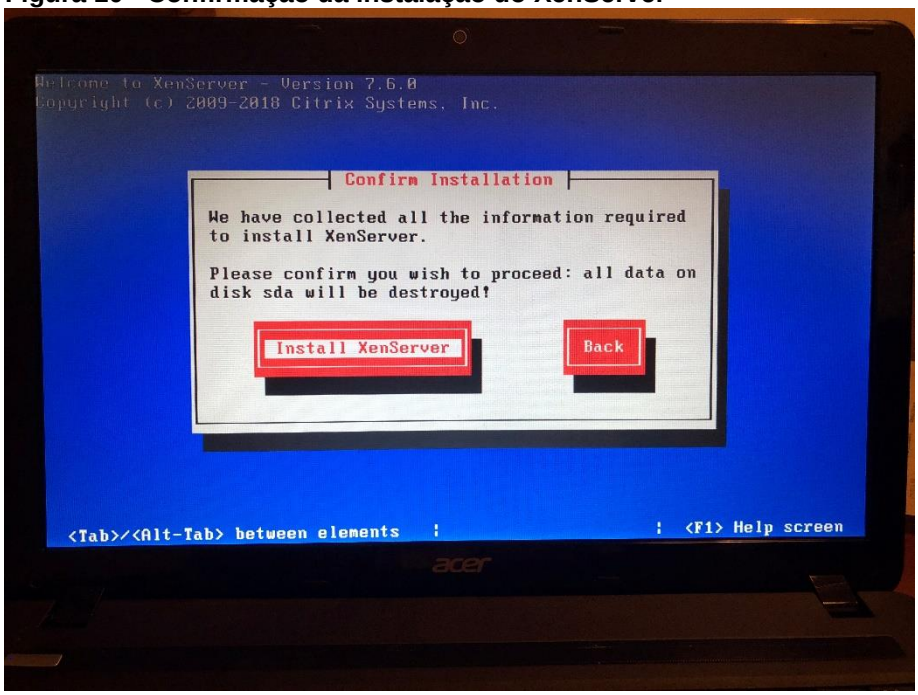
Figura 9 - Tela de sincronia de horário usando o NTP (*Network Time Protocol*)



Fonte: Autoria própria.

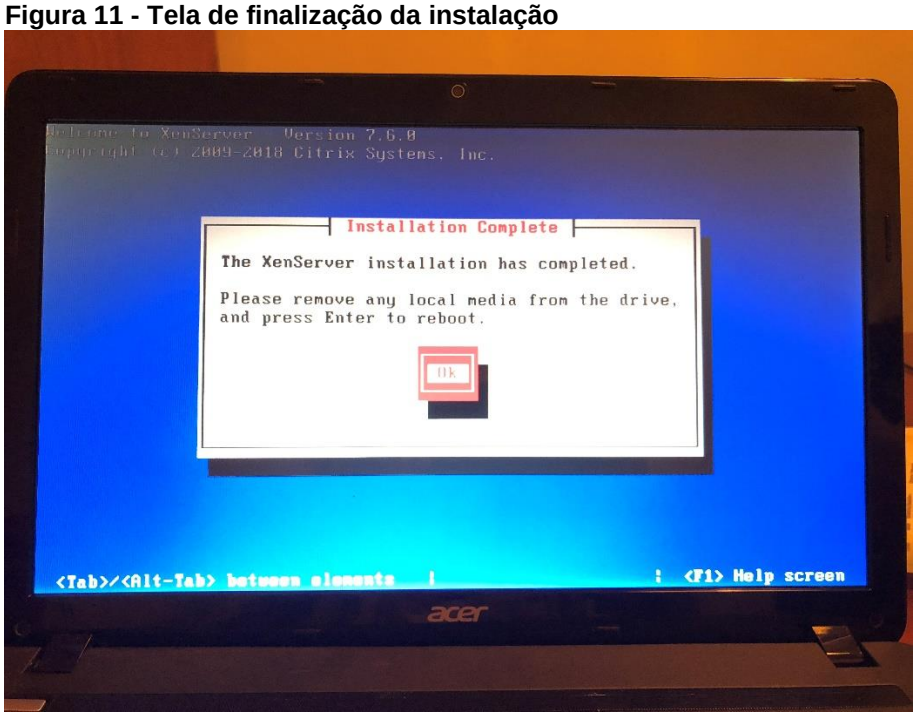
Para confirmar a instalação, deve-se apertar em "*Install XenServer*" e aguardar o final da instalação, conforme mostrado na Figura 10. Ao finalizar a instalação, deve-se remover a mídia de instalação e reiniciar o computador, como mostra a Figura 11.

Figura 10 - Confirmação da instalação do XenServer



Fonte: Autoria própria.

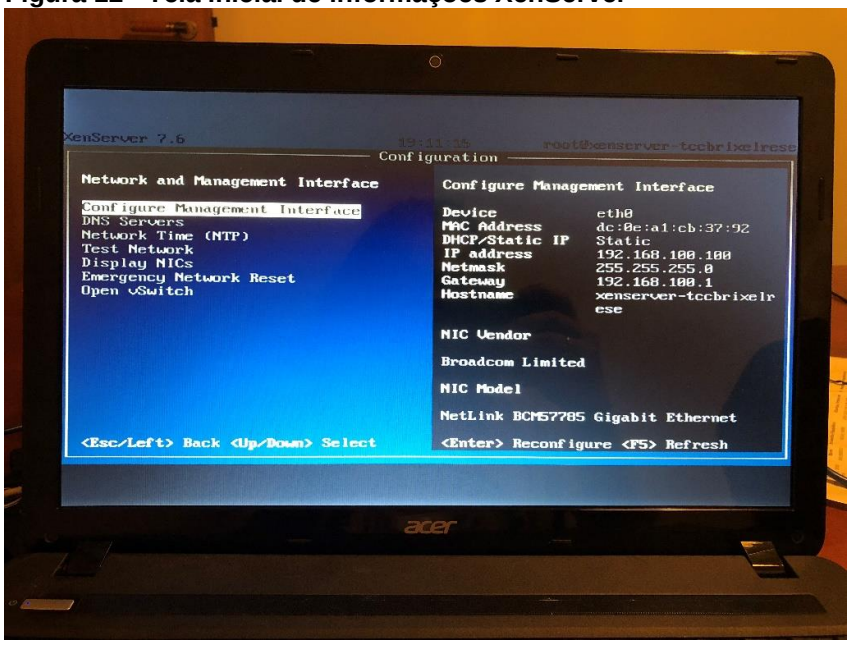
Figura 11 - Tela de finalização da instalação



Fonte: Autoria própria.

Na Figura 12, apresenta-se a tela do XenServer, onde se encontram informações como versão, configurações de rede e de maquinas virtuais. A princípio foi utilizado a opção para atribuir um endereço IP manualmente para que os computadores do XenServer e XenCenter fiquem na mesma sub-rede e se enxerguem corretamente.

Figura 12 - Tela inicial de informações XenServer



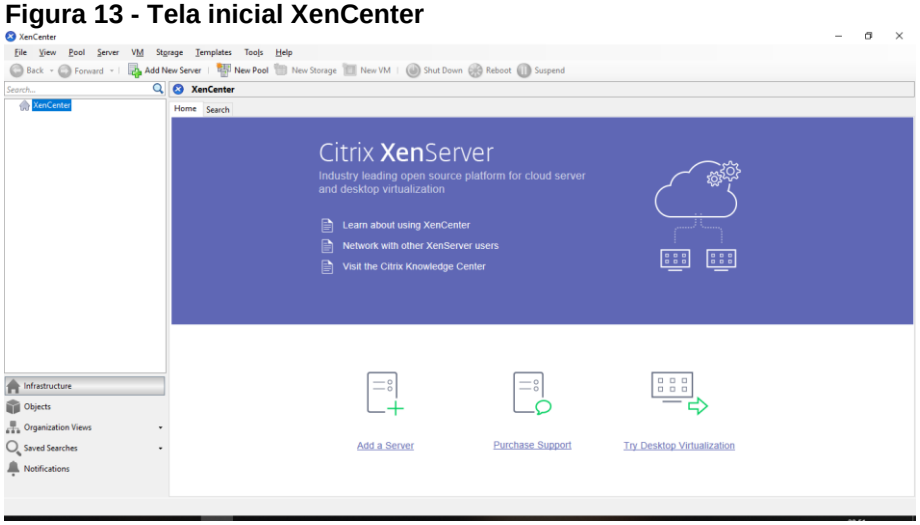
Fonte: Autoria própria.

3.2 CITRIX XENCENTER

O Citrix XenCenter é a ferramenta para gerenciar e monitorar as VMs no XenServer, ele deve ser instalado em um computador onde o Sistema Operacional é Windows. A instalação é simples e não necessita atenção especial. Nos passos a seguir, é mostrado os detalhes para configuração do XenCenter e do IP do Windows.

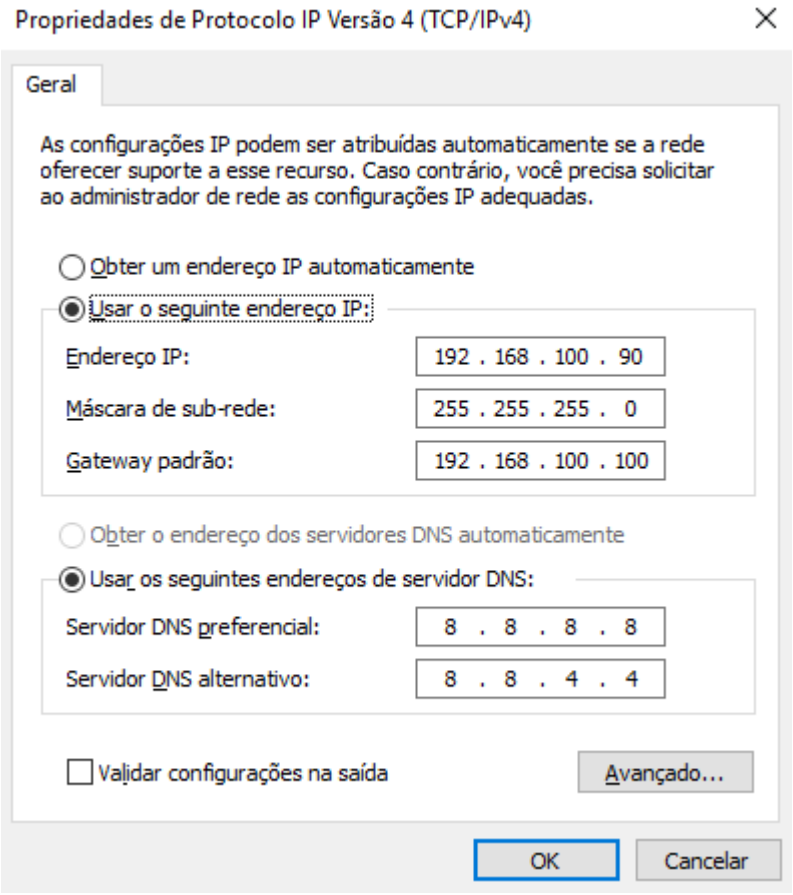
Após instalado, conforme a Figura 13, o XenCenter irá mostrar a lista de máquinas virtuais, que por enquanto está vazia, e algumas configurações do software. Necessita-se configurar o endereço IP do computador para que possa se comunicar com o servidor onde o XenServer está instalado. Coloca-se um conjunto de IPs temporários para configurarmos os servidores, futuramente esses endereços serão preenchidos automaticamente pelo serviço DHCP. No XenServer especificamos o IP 192.168.100.100 /24, e na estação com o XenCenter, é colocado a interface de rede estática para o IP 192.168.100.90/24 com o *gateway* atribuído para o IP do XenServer (192.168.100.100), conforme mostrado na Figura 14.

Figura 13 - Tela inicial XenCenter



Fonte: Autoria própria.

Figura 14 - Configuração da interface de rede do Windows onde o XenCenter se encontra instalado

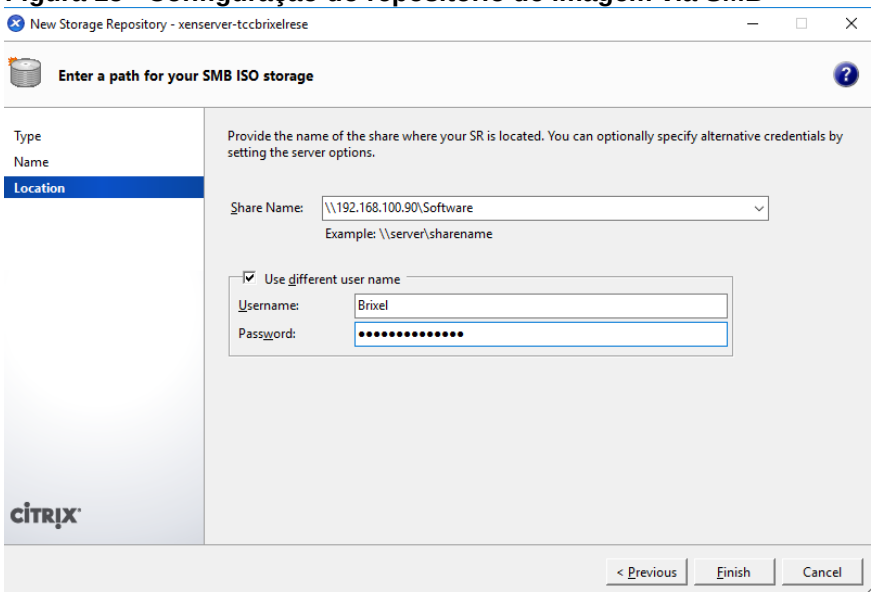


Fonte: Autoria própria.

Após os computadores se comunicarem, será necessário conectar ao XenServer pelo XenCenter, apertando em *Add New Server* na tela inicial, inserindo o *hostname* ou endereço ip do Servidor (nosso caso 192.168.100.100) e as credenciais de *login* configuradas anteriormente.

Como o XenCenter/XenServer não provê o *download* automático para a instalação das máquinas virtuais, é necessário a mídia de instalação de cada sistema operacional desejado, também há a opção de criar um repositório de imagens, onde os instaladores dos sistemas operacionais se encontram. No nosso caso, foi criado um compartilhamento no computador Windows com o arquivo de imagem do Ubuntu, e foi conectado via XenCenter pelo menu "*New SR (Storage Repository)*" e depois em "*Windows File Sharing (SMB/CIFS)*" conforme mostra-se na Figura 15.

Figura 15 - Configuração do repositório de imagem via SMB

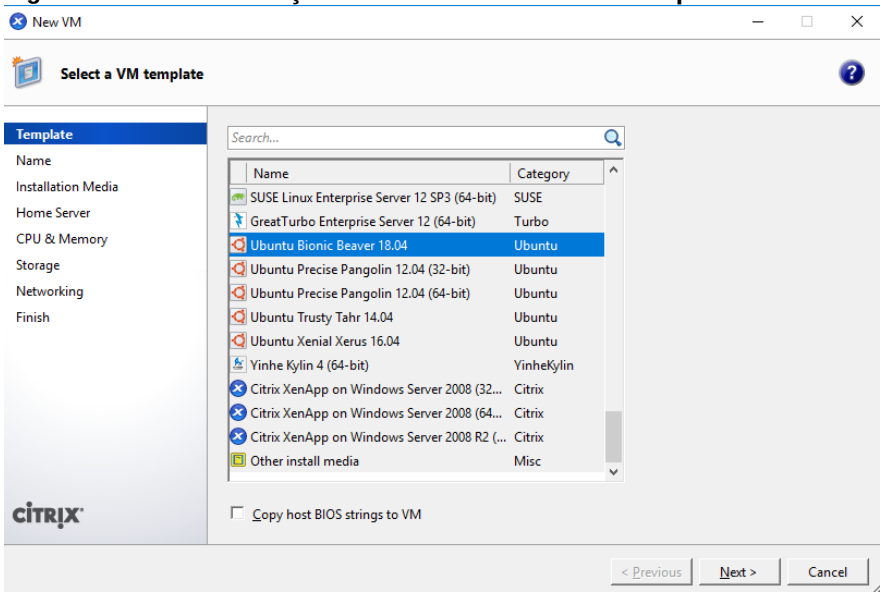


Fonte: Autoria própria.

Para iniciar a instalação das máquinas virtuais, deve-se apertar no menu "VM" e depois em "New VM", ou pelo comando de atalho "Ctrl+N".

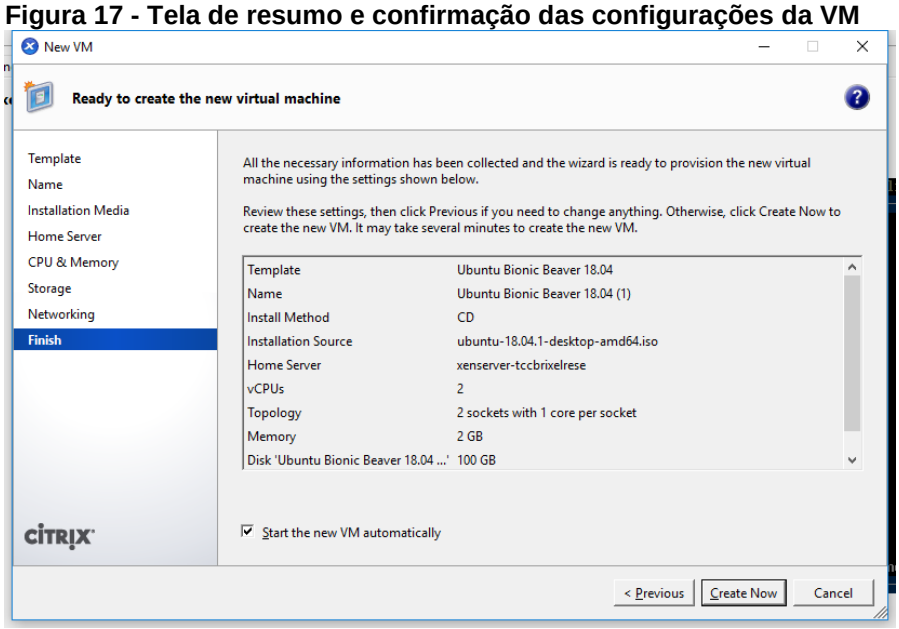
Com isso, é mostrado uma lista com vários modelos de instalação de sistemas operacionais, os menus são customizáveis, então não há necessidade de selecionar algum específico, caso queira um mais genérico, recomenda-se selecionar "*Other Install Media*". Como mostra a Figura 16, foi selecionado a opção "*Ubuntu Bionic Beaver 18.04*", na tela "*Installation Media"* foi selecionado a imagem do sistema operacional no nosso *Storage Repository* que foi configurado previamente. Na tela "*CPU & Memory"* foi modifcado o número de processadores virtuais (vCPUs) para 2 e a memória para 2048MB, na tela seguinte "*storage*" foi modificado a capacidade do disco rígido virtual para 100GB. Na última tela o XenCenter é mostrado as configurações selecionadas, após confirmadas, deve-se apertar no botão "*Create Now",* assim como mostrado na Figura 17.

Figura 16 - Tela de Seleção de modelo dos sistemas operacionais



Fonte: Autoria própria.

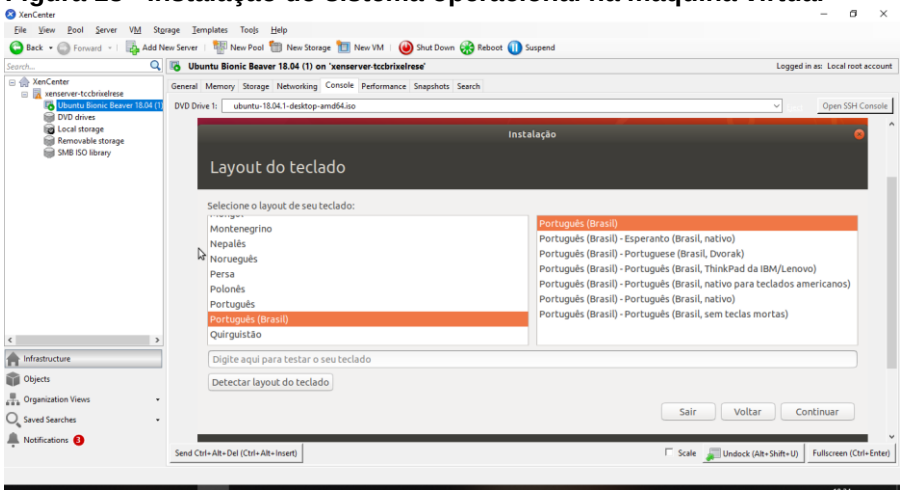
Figura 17 - Tela de resumo e confirmação das configurações da VM



Fonte: Autoria própria.

A partir desse momento, no menu “console”, conforme mostrado na Figura 18, irá ser instalado o sistema operacional dento da máquina virtual, essa instalação irá avançar idêntico à uma instalação em máquina física, sendo apenas necessário acompanhar para fazer as configurações básicas.

Figura 18 - Instalação do sistema operacional na máquina virtual



Fonte: Autoria própria.

Ao completar a instalação, o sistema operacional estará pronto para que possamos usá-lo. Como iremos separar os serviços em diferentes máquinas virtuais, teremos que criar novas VMs, onde, ou poderíamos repetir os passos, ou simplesmente clonar a máquina recém-criada utilizando o menu “VM” e depois “*Copy VM...*”. Foi optado por cloná-la pois é uma operação bem rápida e que nos poupou alguns passos adicionais.

A seguir, iremos descrever e configurar os serviços de rede que instalaremos nas máquinas virtuais.

4 SERVIÇOS

4.1 DYNAMIC HOST CONFIGURATION PROTOCOL (DHCP)

O *Dynamic Host Configuration Protocol* (DHCP) *é* o serviço que permite que os dispositivos solicitem e recebam um endereço de IP, *gateway* e DNS automaticamente dentro de uma faixa pré-definida pelo administrador.

Como foi configurado uma rede pequena, foi fixado a faixa em 254 endereços IPs disponíveis. Nossa configuração de DHCP ficou com o IP mínimo em 192.168.0.1, o máximo em 192.168.0.254 e a sub-rede 255.255.255.0. Assim definido, partimos para a instalação e configuração do serviço.

Como nosso servidor não possuía o DHCP instalado, teve-se que baixar o pacote e instalá-lo conforme comando abaixo:

 # apt-get install isc-dhcp-server

Finalizada a instalação, precisou-se ir até o arquivo /etc/default/isc-server-dhcp e definir a interface de rede que o servidor irá usar para servir os pedidos DHCP. No nosso caso ficou na interface "*eth0*", onde foi inserido após o "*INTERFACESv4=*", resultando na linha abaixo:

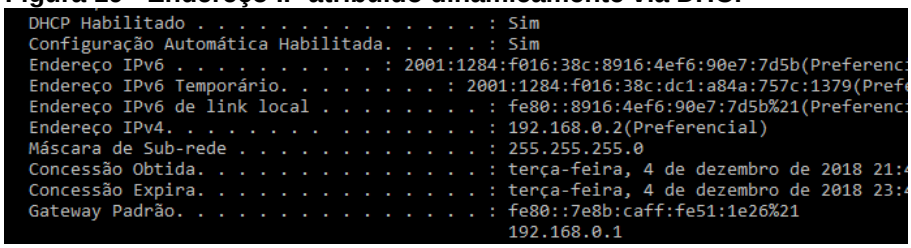
INTERFACESv4="eth0"

Após a inserção desse comando, pode-se avançar para a configuração do serviço, que é feita através do arquivo *"dhcpd.conf*" que fica localizado na pasta "*/etc/dhcp*". Dentro desse arquivo, foi alterado os parâmetros responsáveis por atribuir os endereços nos dispositivos, para isso, foram inseridas as seguintes linhas abaixo:

```
subnet 192.168.0.0 subnet 255.255.255.0 {
   range 192.168.0.1 192.168.0.254
}
```

Ao reiniciar o serviço DHCP, todos os dispositivos que fizeram a solicitação de receberam o endereço na faixa atribuída acima, conforme mostrado na Figura 19.

Figura 19 - Endereço IP atribuído dinamicamente via DHCP



Fonte: Autoria própria.

4.2 DOMAIN NAME SYSTEM (DNS)

O *Domain Name System* (DNS) é o serviço que transforma o nome do domínio que estamos tentando acessar em um endereço de IP. Cada domínio é acessado via seu endereço IP, o DNS permite que possamos acessar esses IPs por um método mais prático. Em uma rede interna, possuir um DNS interno é importante para resolver nomes de outras máquinas, servidores e sistemas da rede. Para instalarmos o nosso servidor DNS, usamos o comando abaixo:

 # apt-get install bind9 bind9utils bind9-doc

Antes de iniciar a configuração do *BIND*, foi necessário atribuir um domínio para nosso servidor, que foi definido como “tccubuntu.brixel.rese”, apontando para o nosso próprio IP estático. Também foi configurado os “*forwarders*”, para que possa ser usado um DNS externo caso nosso servidor não consiga resolver algum nome. Para isso inseriu-se o IP de nosso *gateway* e o DNS do *Google* dentro do arquivo “*named.conf.options*”, encontrado no diretório “*/etc/bind”*, conforme mostrado na Figura 20.

Figura 20 - *Gateway* e DNS do *Google* configurado no *forwarders*

Fonte: Autoria própria.

Então seguiu-se para o arquivo /etc/bind/named.conf.local para definir os arquivos de configurações das zonas, nelas foram configuradas como o sistema irá resolver os nomes e os IPs acessados. Para a pesquisa direta foi criado e apontado para o arquivo /etc/bind/db.direto e para a pesquisa reversa o arquivo /etc/bind/db.192, utilizou-se as linhas de configuração abaixo:

```
zone “tccubuntu.brixel.rese” {
        type master;
        file “/etc/bind/db.direto”;
}

zone “100.168.192.in-addr.arpa” {
        type master;
        file “/etc/bind/db.192”;
}
```

Foram feitas algumas configurações simples nos arquivos, apenas em caráter de teste, onde alterou-se o arquivo db.direto, que é responsável por resolver os nomes em IP, nisso foi configurado o servidor e o modem roteador conforme mostrado na Figura 21. No arquivo db.192, que é o arquivo que recebe o IP e transforma em endereço, foram feitas as configurações do sentido inverso do arquivo de zona direta, essas configurações estão demostradas na Figura 22.

Figura 21 - Zona direta configurada para o domínio e o roteador

```
root@tccubuntu:/etc/bind# cat db.direto
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     tccubuntu.brixel.rese. root.brixel.rese. (
                              5         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
                              IN      NS      tccubuntu.brixel.rese.

tccubuntu.brixel.rese.        IN      A       192.168.100.42
tccubuntu.brixel.rese.modem.  IN      A       192.168.100.1
```

Fonte: Autoria própria.

Figura 22 - Zona reversa configurada para o domínio e o roteador

```
root@tccubuntu:/etc/bind# cat db.192
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     tccubuntu.brixel.rese. root.brixel.rese. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
; Servers
        IN      NS      tccubuntu.brixel.rese.

; PTR
42      IN      PTR     tccubuntu.brixel.rese.
1       IN      PTR     tccubuntu.brixel.rese.modem.
```

Fonte: Autoria própria.

Ao reiniciar o serviço do DNS e apontar nas configurações da interface, o servidor DNS já começou a resolver os IPs e nomes via comando *nslookup*. Depois testou-se o redirecionamento do endereço tccubuntu.brixel.rese.modem para o IP do nosso roteador 192.168.100.1. Na Figura 23 mostra a imagem do nosso navegador redirecionado para a página do modem.

Figura 23 - Navegador com o nome redirecionado para o IP

Fonte: Autoria própria.

4.3 SQUID

O *Squid* foi escolhido para servidor *proxy*, ele é responsável por manter o cache dos sites mais visitados pelos usuários aumentando a velocidade de navegação e diminuir o uso de banda. Ele é usado também como um filtro para controlar o acesso aos endereços, evitando que usuários acessem sites e objetos maliciosos. Além disso, o *Squid* armazena um registro de cada URL que a rede acessa. Assim como os outros serviços, foi necessário instalá-lo, onde usamos o seguinte comando:
 # apt-get install squid

O principal arquivo de configuração foi o /etc/squid/squid.conf, esse é um arquivo bem extenso com muitas opções de configurações, neste caso, apenas alterou-se as configurações básicas e algumas regras de bloqueio.

O primeiro passo foi definir a rede local, que no caso é a 192.168.100.0 máscara sub rede 255.255.255.0, para isso procurou-se dentro do arquivo as linhas onde a entrada “acl localnet src” se encontravam e foi adicionado a seguinte linha: acl localnet src 192.168.100.0/24.

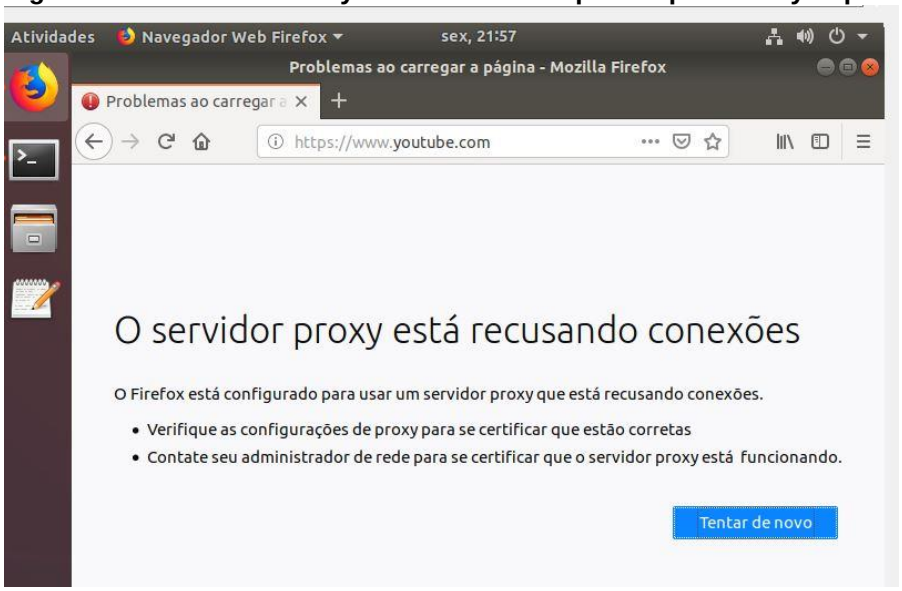
Foi necessário, também, configurar o squid para permitir o tráfego por HTTP, onde procurou-se pelo termo http_access e retirou-se o símbolo de comentário “#”, assim a linha ficou: http_acess allow localnet.

Foi adicionada, também, a linha http_port 8080 para mudar o número da porta do padrão 3128 para 8080. Após isso, foram configurados os sites que o proxy deverá bloquear. Para isso, foi criado o arquivo /etc/squid/acl/banned e, dentro dele, foi inserida a linha:.youtube.com, que significa que qualquer site dentro do domínio “youtube.com” será bloqueado. Para que esse arquivo fosse lido pelo squid, foi necessário adicionar as seguintes linhas de comando no arquivo de configuração squid.conf:

acl sites-bloqueados url_regex –i “/etc/squid/acl/banned”
http_access deny sites-bloqueados.

Após o reinicio do serviço via comando /etc/init.d/squid restart, foi apontado no navegador para o proxy: 192.168.100.42:8080, com isso ele já estava funcional com nossas regras que configuramos anteriormente, na Figura 24 foi demostrado o *proxy* bloqueando o endereço www.youtube.com.

Figura 24 - Website “www.youtube.com” bloqueado pelo Proxy implementado



Fonte: Autoria própria.

4.4 FIREWALL

O *firewall* é o responsável por monitorar e controlar todo o trafego de dados, aplicando uma política de segurança e barrando todo acesso não desejado. Para o servidor, foi usado o UFW: *Uncomplicated Firewall* (*Firewall* descomplicado), que basicamente é *front-end* do *firewall* padrão do Linux, o *iptables*. O UFW é intuitivo e suporta as tarefas mais comuns esperadas de um *firewall*, como bloqueio de portas, evitando redes não confiáveis conectarem à nossa rede local.

Normalmente o UFW já vem instalado por padrão na maioria das distribuições Linux e é bem suportado pela comunidade, para verificar se ele já está em execução deve-se executar o comando: sudo ufw status, caso não esteja instalado deve se usar o comando apt-get install ufw e ele será instalado. Se ele estiver ativo, ele irá mostrar uma listagem com as regras ativas no momento. Se o caso dele for estar instalado e inativo, deve se usar o comando ufw enable para ativá-lo. Na Figura 25 mostra-se o comando para ativar o UFW e o quadro de resposta de quando está ativo.

Figura 25 - Comando para ativar o ufw e a tela de status

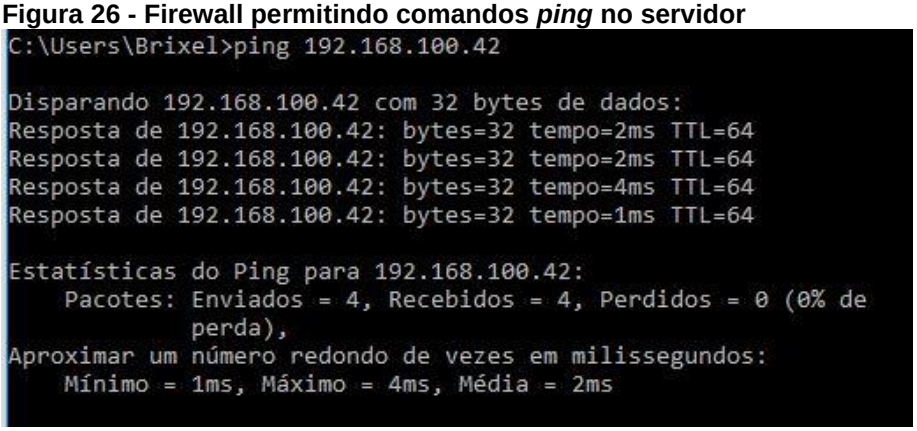
Fonte: Autoria própria.

Para testar nosso *firewall*, foi bloqueado o comando *ping* em nosso servidor, para isso alteramos o arquivo /etc/ufw/before.rules, localizado nas linhas abaixo:

 # ok icmp codes
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-input -p icmp --icmp-type source-quench -j ACCEPT
-A ufw-before-input -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-input -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-input -p icmp --icmp-type echo-request -j ACCEPT

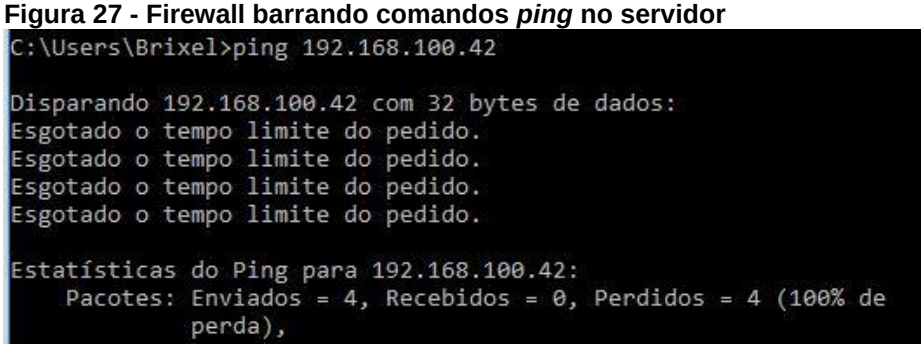
Então, foram alterados os atributos "*ACCEPT*" para "*DROP*", assim, bloqueando toda requisição de *ping* de qualquer máquina externa. Na Figura 26, mostra-se o retorno do comando *ping* do computador com Windows para o servidor enquanto os atributos do servidor estão em *ACCEPT*, e na Figura 27 quando foi modificado para *DROP*, negando que o comando fosse respondido.

Figura 26 - Firewall permitindo comandos *ping* no servidor



Fonte: Autoria própria.

Figura 27 - Firewall barrando comandos *ping* no servidor



Fonte: Autoria própria.

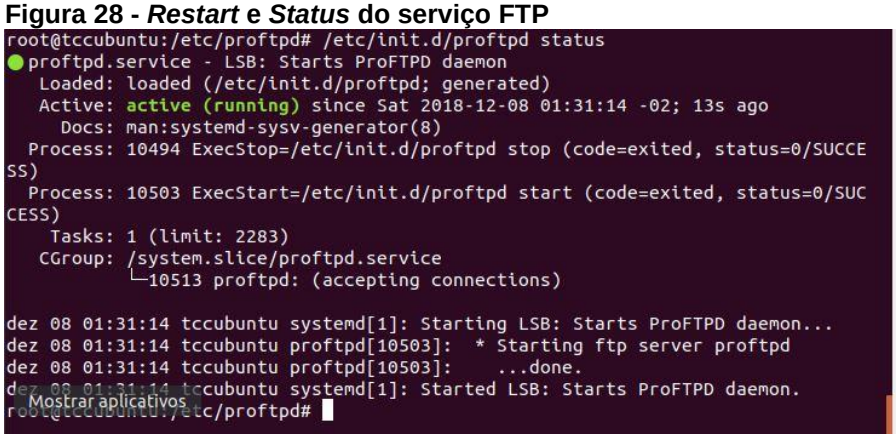
4.5 FILE TRANSFER PROTOCOL (FTP)

O *File Transfer Protocol* (FTP, ou Protocolo de Transferência de Arquivo) é o protocolo responsável pela transferência de arquivos e acessos a sistemas de arquivos em máquinas e servidores remotos. Foi utilizado o *software* proftp para gerenciar o protocolo. Para a instalação, foi utilizado o comando apt-get install proftpd.

Após a instalação, basta acessar o diretório /etc/proftpd, dentro do arquivo proftpd.conf estarão os atributos de configuração. Primeiro localizou-se a linha ServerName e foi trocado o atributo “Debian” para o nome do nosso servidor: tccbrixelrese. Após isso, procurou-se o item User e Group e alterou-se ambos para ftp. Após isso, foi atribuída uma senha para o usuário FTP no Linux através do seguinte comando passwd ftp e criado o grupo com o comando addgroup ftp. Ao criar a senha e o grupo FTP, adicionou-se o usuário FTP ao grupo FTP. Para isso, foi usado o comando addgroup ftp ftp. Por fim, foi alterada a linha do FTP dentro do arquivo /etc/passw, substituindo a palavra “srv para home, foi criada a pasta para o usuário FTP e concedeu-se acesso total para essa pasta. Para isso foram utilizados os seguintes comandos: mkdir /home/ftp e chmod 777 /home/ftp.

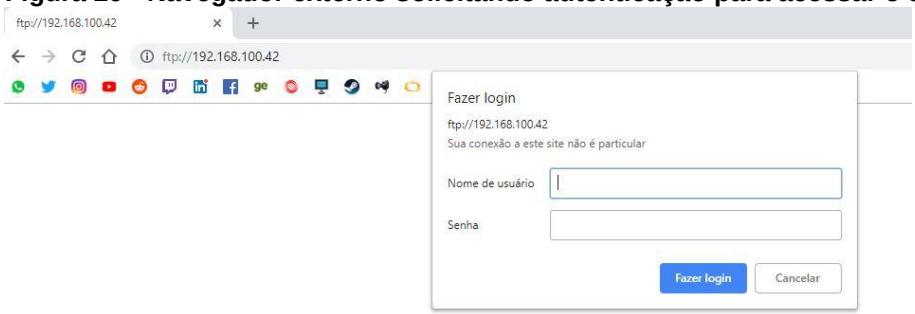
Assim, o servidor FTP encontrou-se configurado. Utilizou-se o comando /etc/init.d/proftpd restart para reiniciar o serviço e o comando /etc/init.d/proftpd status para checar seu *status*, como demonstrado na Figura 28. Após isso, o servidor está pronto para conexões externas. Na Figura 29, mostra-se o servidor solicitando autenticação ao receber o endereço ftp://192.168.100.42.

Figura 28 - *Restart* e *Status* do serviço FTP



Fonte: Autoria própria.

Figura 29 - Navegador externo solicitando autenticação para acessar o servidor FTP



Fonte: Autoria própria.

5 CONSIDERAÇÔES FINAIS

Sabemos que hoje em dia é essencial ter uma rede de computadores nas empresas, e para empresas de pequeno porte não é diferente. A informatização facilita muito os processos internos e, consequentemente, a relação das empresas com seus clientes.

Com a facilidade na transmissão de dados, existe uma volatilidade e um risco muito grande. Por esse motivo, esse trabalho visa aumentar a segurança dos dados de empresas pequenas com um custo muito baixo.

No projeto, foi proposto a instalação do XenServer em uma máquina para servir como servidor. Utilizou-se o XenCenter em uma estação de gerenciamento para instalar máquinas virtuais com o sistema operacional Linux Ubuntu no servidor.

Houve uma grande facilidade de configuração e gerenciamento das máquinas virtuais pelo XenCenter, onde há opções de clonar a VM em poucos segundos, criando uma cópia idêntica com todas as informações e configurações do sistema clonado, pode-se também criar *snapshots*, onde a máquina virtual cria um ponto de restauro, podendo retornar a esse ponto, facilitando o backup de configurações.

Nas máquinas virtuais, foram configurados os seguintes serviços: a) o DHCP foi configurado para atribuir um endereço físico em todas as estações de trabalho da rede automaticamente; b) para a segurança de tráfego e controle de acesso, foi configurado o *firewall* e *proxy*; c) para facilitar o acesso das máquinas na mesma rede e acessar sites externos, foi configurado um serviço de DNS; e d) para ter um melhor controle dos dados da rede, um servidor FTP.

Considerou-se uma rede para uma empresa de pequeno porte, mas se fosse utilizar os mesmos serviços em uma empresa com mais estações, é importante realizar um *upgrade* no *hardware* do servidor. Principalmente no processamento, na memória e no espaço de armazenamento para o servidor FTP.

Neste projeto, obteve-se sucesso em todos os serviços que foram propostos a configurar, deixando em segurança os dados da empresa. Em projetos futuros, pode-se considerar a integração do servidor FTP com um sistema de backup físico e um backup em nuvem, assim tem-se uma redundância ainda maior e mais segurança dos arquivos. Também é possível estudar novas tecnologias de segurança de tráfego de dados e acesso remoto.

REFERÊNCIAS

BUENO, Henrique. **Virtualização: Um pouco de história.** Wordpress, 29 abr. 2009. Disponível em: <http://hbueno.wordpress.com/2009/04/29/virtualizacao-um-pouco-de- historia/>. Acesso em: 15 set. 2018.

MARAN, Fabio. **Virtualização de Sistemas.** Viva o Linux, 17 jan. 2008. Disponível em: <https://www.vivaolinux.com.br/artigo/Virtualizacao-de-sistemas>. Acesso em: 15 set. 2018.

POPEK, G. J.; GOLDBERG, R. P. **Formal requirements for virtualizable third generation architectures**. ACM 17, 7 ,1974.

SARDINHA, Gelson. **Virtualização de Servidores.** Voolivrelinux, 30 set. 2009. Disponível em: <http://voolivrelinux.blogspot.com/2009/09/virtualizacao-de-servidores.html>. Acesso em: 15 set. 2018.

SILVA, Rodrigo Ferreira da Silva. **Virtualização de sistemas operacionais.** 2007. 26 f. Monografia (Graduação em tecnologia da Informação e Comunicação). Instituto Superior de Tecnologia em Ciências da Computação, Petrópolis, 2007.

SIQUEIRA, Ethevaldo. **Para compreender o mundo digital.** São Paulo: Editora Globo. 2008.