

Senai – Santos Dumont

Técnico de Redes de Computadores

Alexandre Victor Viena de Melo  
Anna Júlia de Morais Guimarães  
Daniel Ferreira dos Santos Filho  
Jonas do Nascimento Lemes

Virtualização na Nuvem com KVM

Professor(a) Orientador(a): Kleber Gelli e Wellington Carlos Joffre

São José dos Campos-SP

2018

Alexandre Victor Viena de Melo  
Anna Júlia de Morais Guimarães  
Daniel Ferreira dos Santos Filho  
Jonas do Nascimento Lemes

**Virtualização na Nuvem com KVM**

Trabalho de Conclusão de Curso apresentado a escola SENAI Santos Dumont de São José dos Campos, como parte dos requisitos necessários para a conclusão do curso Técnico de Redes de Computadores.

Orientadores: **Kleber Gelli**

**Wellington Carlos Joffre**

São José dos Campos-SP

2018

Gostaria de primeira de agradecer aos professores que durante o período do curso, passaram seus conhecimentos e ensinamentos, gostaria também de agradecer aos meus colegas por ajudar e realizar os devidos trabalhos.

**Alexandre Victor Vieira de Melo**

Gostaria de agradecer aos professores que nos ajudaram bastante com o projeto, nos dando dicas e informações.

**Anna Júlia de Morais Guimarães**

Quero agradecer, primeiramente a Deus, por me dar a oportunidade de realizar, com paciência e determinação este projeto, do qual me orgulho muito.

À minha família, por todo carinho e apoio em todas as escolhas que fiz. Mãe e pai, o apoio de vocês foi de total importância para o meu desenvolvimento como cidadão.

**Daniel Ferreira dos Santos Filho**

Gostaria de agradecer aos professores por esses 2 anos juntos, e pelo conhecimento passado durante esse tempo.

**Jonas do Nascimento Lemes**

AGRADECIMENTOS

A Deus, por iluminar meus passos para que eu possa trilhar meus caminhos nesta nova jornada.

Ao Diretor do SENAI – Santos Dumont, por ter proporcionado o Curso técnico de Redes de Computadores.

Ao Coordenador do Curso de Redes de Computadores, José Rogério Chavier, pela ética e pela destreza.

Ao professor Kleber Gelli, Wellington Carlos Joffre, Josemar Monteiro, Airton César Zombarti e Paulo Eduardo Galvão pela orientação deste estudo.

Aqueles, que direta ou indiretamente, colaboraram para o desenvolvimento desta pesquisa.

O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo. Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas admiráveis.

**José de Alencar**

**RESUMO**

Com a expansão da tecnologia a virtualização ganhou muita força no mercado e vem sendo utilizada em diversas empresas. Com essa tecnologia a virtualização permite executar diversos sistemas operacionais simultaneamente no mesmo hardware, o que são chamados de maquinas virtuais. Vemos diversos problemas com maquinas virtuais, como problemas de perda, identificação de maquinas, compartilhamento e compatibilidade de maquinas. O projeto tem por fim ajudar as empresas e instituições que utilizam maquinas virtuais para que tenha menos problemas com perda. As maquinas virtuais seriam um software de ambiente computacional em que um sistema operacional ou programa pode ser instalado e executado, e nas maquinas virtuais também é muito utilizado *hypervisor*.

*Hypervisor* é uma camada de software entre o hardware e o sistema operacional, sendo responsável por fornecer ao sistema operacional visitante ao funcionamento da máquina virtual. Além de controla o acesso dos sistemas operacionais visitantes aos dispositivos de hardware, atualmente no mercado existem diversos *hypervisors* e entre eles a ferramenta *Kernel-based Virtual Machine* (KVM) que foi utilizada no projeto da VirtuaCloud a ferramenta *Kernel-based Virtual Machine* (KVM) que é uma das ferramentas mais importante para virtualização de computadores. O *Kernel-based Virtual Machine* (KVM) é uma infraestrutura de virtualização integrada ao sistema operacional Linux. O *Kernel-based Virtual Machine* (KVM) converte o Linux em um *hypervisor*, para executar as maquinas virtuais (VM – Virtual Machine), todos os *hypervisores* precisam de alguns componentes em nível de sistema operacional, como gerenciador de memória, agenda de processos e entre outros, com isso muitas das vezes é utilizada linguagens de *scripts* para melhorar a configuração de hypervisors como o os *shells* de sistemas operacionais (OS), a linguagem *shell* *script* seria uma linguagem script usada em muitos sistemas operacionais como o Linux. O Shell Script facilita consideravelmente trabalho do administrador de sistemas e de qualquer outro usuário com automatização de tarefas que é refletida em aumento de velocidade e facilidade. Além de todos as linguagens *scripts* usados em projetos de virtualizações em maquinas é muita utilizada a computação na nuvem (*cloud computing*) que da possibilidade de acessar arquivos e executar diferentes tarefas pela internet, sem a necessidade de instalar aplicativos no computador, o armazenamento de dados é feito em serviços on-line, em uma rede.

Com isso o projeto VirtuaCloud fez uma pesquisa para pessoas relacionadas a área da informática porque seria importante o salvamento de maquinas virtuais no armazenamento na nuvem e se os usuários tinham costume de salvar suas vm’s, com isso poderia ter uma ideia como esse projeto iriam ajudar os usuários que usam maquinas virtuais. Conclua que virtualização vem ganhando muita força no mercado e com isso projetos para melhoria da virtualização vem crescendo e com isso vemos um projeto como VirtuaCloud para ajudar as empresas e instituições com problemas com virtualização.

**Keywords:** KVM. Virtual machine. Hypervisor.

***ABSTRACT***

With the expansion of technology virtualization has gained a lot of strength in the market and has been used in several companies. With this technology virtualization allows to run several operating systems simultaneously on the same hardware, which are called virtual machines. We see several problems with virtual machines, such as problems of loss, identification of machines, sharing and compatibility of machines. The project aims to help companies and institutions that use virtual machines to have fewer problems with loss. The virtual machines would be a computer environment software in which an operating system or program can be installed and executed, and in virtual machines it is also much used hypervisor.

Hypervisor is a layer of software between the hardware and the operating system, being responsible for providing the operating system visitor with the operation of the virtual machine. In addition to controlling the access of the operating systems visitors to the hardware devices, there are currently several hypervisors in the market and among them the Kernel-based Virtual Machine (KVM) tool that was used in the VirtuaCloud project the Kernel-based Virtual Machine (KVM) which is one of the most important tools for computer virtualization. The Kernel-based Virtual Machine (KVM) is a virtualization infrastructure integrated with the Linux operating system. KVM converts Linux into a hypervisor, to run virtual machines (VMs), all hypervisors need some operating system, such as memory manager, process scheduler and so on, with this is often used scripting languages ​​to improve the configuration of hypervisors such as OS (operating system (OS) shells, shell language would be a scripting language used on many operating systems such as Linux. Shell Script greatly facilitates the work of your system administrator and any other user with task automation that is reflected in increased speed and ease. In addition to all scripting languages ​​used in machine virtualization projects, cloud computing is often used to access files and perform different tasks over the Internet without the need to install applications on the computer, data storage is done in online services on a network.

With this the VirtuaCloud project did a research for people related to computer science because it would be important to save virtual machines in the cloud storage and if users had a habit of saving their vm's, this could have an idea how this project would help the users using virtual machines. Conclude that virtualization has been gaining a lot of force in the market and with this projects for the improvement of virtualization is growing and with this we see a project like VirtuaCloud to help companies and institutions with problems with virtualization.

**Keywords:**KVM*.* Cloud. Virtual machines.

**LISTA DE FIGURAS**

[Figura 1 - Representação de uma rede WAN. 21](#_Toc531854139)

[Figura 2 - Tipo *bare-metal* 23](#_Toc531854140)

[Figura 3 - Representação de Acesso Remoto 25](#_Toc531854141)

[Figura 4 - Tela inicial do menu de usuário 36](#_Toc531854142)

[Figura 5 - Adicionando um novo usuário 37](#_Toc531854143)

[Figura 6 - Listando os usuários existentes no servidor 37](#_Toc531854144)

[Figura 7 - Deletando um usuário criado no sistema 37](#_Toc531854145)

[Figura 8 - Alterando a senha de um usuário já existente 38](#_Toc531854146)

[Figura 9 - Tela inicial do menu de gerenciamento de VMs 38](#_Toc531854147)

[Figura 10 - Listando as máquinas virtuais criada pelo usuário 38](#_Toc531854148)

[Figura 11 - Iniciando a VM criada pelo usuário 38](#_Toc531854149)

[Figura 12 - Deletando uma VM criada pelo usuário 39](#_Toc531854150)

[Figura 13 - Escolhendo o tipo de VM que deseja criar 39](#_Toc531854151)

[Figura 14 - Menu de importação das VMs Windows 39](#_Toc531854152)

[Figura 15 - Menu de importação de VMs Debian 39](#_Toc531854153)

[Figura 16 - Informações pedidas para a criação de uma nova VM 39](#_Toc531854154)

**LISTA DE GRÁFICOS**

[Gráfico 1 - Nível de conhecimento sobre Cloud Computer. 32](#_Toc531854165)

[Gráfico 2 - Utilização de armazenamento na nuvem. 33](#_Toc531854166)

[Gráfico 3 - Nível de conhecimento sobre máquina virtual. 33](#_Toc531854167)

[Gráfico 4 - Utilização de máquina virtual. 34](#_Toc531854168)

[Gráfico 5 - Consumo do processador. 34](#_Toc531854169)

[Gráfico 6 - Costume de exportar e salvar a máquina. 35](#_Toc531854170)

**LISTA DE TABELAS**

[Tabela 1 – Cronograma 26](#_Toc531854171)

[Tabela 2 - Servidor Storage Silix X1200H25 V6 27](#_Toc531854172)

[Tabela 3 - Servidor Silix E5-2600dr V4 27](#_Toc531854173)

[Tabela 4 - Servidor Montado pela VirtuaCloud 28](#_Toc531854174)

[Tabela 5 - Valor da mão de obra 30](#_Toc531854175)

[Tabela 6 - Trabalho durante o turno 30](#_Toc531854176)

[Tabela 7 - Trabalho fora do turno 31](#_Toc531854177)

[Tabela 8 - Custo final da mão de obra 31](#_Toc531854178)

**LISTA DE SIGLAS E ABREVIATURAS**

HD *Hard Disk*

KVM *Kernel-based Virtual Machine*

LAN *Local Area Network*

WAN *Wide Area Network*

MAN *Metropolitan Area Network*

PAN *Personal Area Network*

QUANT Quantidade

SSD S*olid-state Drive*

SSH *Secure Shell*

TI Tecnologia da informação

UN Unidade

VPS *Virtual Private Server*

NAT *Network Address Translation*

VM *Virtual Machine*

SUMÁRIO

[1. INTRODUÇÃO 18](#_Toc531854384)

[1.1 OBJETIVO 19](#_Toc531854385)

[1.1.1 OBJETIVOS GERAIS 19](#_Toc531854386)

[1.1.2 OBJETIVOS ESPECÍFICOS 19](#_Toc531854387)

[1.2 JUSTIFICATIVA 19](#_Toc531854388)

[2. FUNDAMENTAÇÃO TEÓRICA 20](#_Toc531854389)

[2.1 REDES DE COMPUTADORES 20](#_Toc531854390)

[2.2 SISTEMA OPERACIONAL 21](#_Toc531854391)

[2.3 KVM 22](#_Toc531854392)

[2.4 MÁQUINA VIRTUAL 22](#_Toc531854393)

[2.5 ACESSO REMOTO 23](#_Toc531854394)

[3. MÉTODO E PROCEDIMENTOS 26](#_Toc531854395)

[3.1 CRONOGRAMA 26](#_Toc531854396)

[3.2 CUSTOS 27](#_Toc531854397)

[3.2.1 HARDWARE 27](#_Toc531854398)

[3.2.2 CUSTOS GERAIS 29](#_Toc531854399)

[3.2.3 CUSTO MÃO DE OBRA ESPECIALIZADA 30](#_Toc531854400)

[3.3 TIPO DE PESQUISA 31](#_Toc531854401)

[3.4 CARÁTER DA PESQUISA 31](#_Toc531854402)

[3.5 MÉTODO DE ABORDAGEM 32](#_Toc531854403)

[3.6 DADOS DA PESQUISA DE CAMPO 32](#_Toc531854404)

[4. DESENVOLVIMENTO 36](#_Toc531854405)

[5. RESULTADOS E DISCUSSÃO 40](#_Toc531854406)

[6. CONCLUSÃO 41](#_Toc531854407)

[7. RECOMENDAÇÕES 42](#_Toc531854408)

[8. REFERÊNCIAS 43](#_Toc531854409)

[9. APÊNDICE A – INSTRUMENTO DE COLETA DE DADOS 46](#_Toc531854410)

# 1. INTRODUÇÃO

Com o a chegada da tecnologia, é possível reduzir o número de servidores e com isso, há uma maior economia em custos de gerenciamento, energia e cabeamento, gastos esses que estão cada vez mais elevados, que resulta em gastos não só para a empresa, mas também para o meio ambiente, com mais matéria prima retirada da natureza para criar esses produtos e também com o consumo de energia que impacta diretamente no meio ambiente. A virtualização é uma tecnologia voltada para o uso de máquinas virtuais. Seu objetivo é fazer com que uma única máquina possa simular várias maquinas, ou seja, a partir dessa máquina podem ser instalados servidores e realizado um controle centralizado.

Hoje em dia empresas que trabalham com servidores e maquinas, acabam tendo problemas com as mesmas. E com a utilização das maquinas virtuais não precisa de troca de *hardware* caso tenham um problema. A vários tipos de virtualizadores disponíveis no mercado que oferecem esse suporte, porém algumas delas não são 100% confiáveis, e também se der algum problema em sua máquina física, corre o risco de perder todas as suas maquinas.

Por conta disso o projeto da *VirtuaCloud* é criar um *software* de maquinas virtuais, porém acessível e prático, com sistema de *login*, poderá utilizar as suas maquinas virtuais em qualquer computador sem precisar exportar e importar. Utilizaremos o *cloud* *computing* para as maquinas virtuais ficarem salvas no servidor da empresa, para isso a empresa terá de manter constantemente conectados os computadores e servidores na mesma rede.

Estamos desenvolvendo o projeto para não só as empresas como um estabelecimento qualquer que possua um servidor e algumas maquinas reais, para não gastarem muito para poder ter credibilidade sem riscos para usufruir de maquinas virtuais, porque querem aumentar o número de maquinas sem ter de adquirir novos *hardwares*.

Isso é ótimo para empresas desenvolvedoras, para testarem seus programas em diferentes sistemas operacionais, ou mesmo para evitar e já ter conhecimento sobre erros,*bugs*,problemas,melhoramentos etc. Com isso a empresa já irá está preparada caso ocorra algo inesperado, que envolva seus servidores e maquinas reais, eles poderão se aprofundar testando seus conhecimentos nas maquinas virtuais sem correr riscos no *hardware*.

## OBJETIVO

### 1.1.1 OBJETIVOS GERAIS

Criar uma nuvem privada para o gerenciamento de máquinas virtuais usando a ferramenta *Kernel-based Virtual Machine* para ajudar na virtualização.

### 1.1.2 OBJETIVOS ESPECÍFICOS

* Refazer a infraestrutura de rede, para que os servidores e as maquinas estejam na mesma rede.
* Desenvolver um sistema de *login* de usuários na interface do *software*, para controle das maquinas virtuais
* Desenvolver um script para criação e gerenciamento das maquinas virtuais.
* Aplicar um servidor montado para a empresa
* Desenvolver na nuvem um ambiente para guardar as maquinas virtuais.

## 1.2 JUSTIFICATIVA

Justifica-se que com a tecnologia da virtualização utilizada cada vez mais nas empresas e instituições, precisa de um cuidado melhor na monitoração de máquina virtuais utilizando ferramentas de um bom desenvolvimento e uma alta tecnologia de monitoramento de máquinas virtuais, com esse intuito nosso projeto tem a finalidade de oferecer mais segurança e um melhor gerenciamento utilizando a ferramenta KVM.

Com o uso de uma ferramenta de alta tecnologia como oferece o KVM, será possível colocar várias maquinas e servidores virtuais em único espaço físico, onde terá um gerenciamento centralizado, criando-se uma redução de vários computadores sendo usados e com isso utilizando menos cabos e tomadas.

Além disso o VirtuaCloud utiliza o Cloud Computing (computação na nuvem), que é uma ferramenta muito utilizada no mercado da virtualização para salvar as máquinas virtuais e também pela segurança oferecida no VirtuaCloud junto com a tecnologia KVM que oferece uma restrição de usuário e senha e também com a conexão das máquinas em uma só rede.

# 2. FUNDAMENTAÇÃO TEÓRICA

## 2.1 REDES DE COMPUTADORES

Redes de Computadores é um conjunto formado por máquinas eletrônicas com processadores capazes de trocar informações e compartilhar recursos, interligados por um subsistema de comunicação segundo o livro Redes de Computadores e a Internet de Kurose Ross, ou seja, é quando se tem pelo menos dois ou mais computadores, e outros dispositivos interligados entre si de modo a poderem compartilhar recursos físicos e lógicos, estes podem ser do tipo: dados, impressoras, mensagens (e-mails), entre outros. Dentro deste conjunto de computadores, existem tipos de conexões que fornecem o estilo de rede para os dispositivos interligados nele, estas são: Lan, Wan, Pan, Man, etc.

LAN é *Local Área Network*. Este termo geralmente se refere a redes de computadores restritas a um local físico definido como uma casa, escritório ou empresa em um mesmo prédio. Uma rede sem fio de uma empresa também faz parte da LAN. O que realmente limita a rede LAN é uma faixa de IP restrita, com uma máscara de rede comum.

WAN é *Wide Área Network*. Significa uma rede que cobre uma área física maior, como o campus de uma universidade, uma cidade, um estado ou mesmo um país. É usados frequentemente nas configurações dos roteadores para se referir à rede externa à empresa, que não é considerada parte da LAN.

WAN também é usado para se referir à rede da internet em geral, apesar desta ser uma designação genérica demais. As redes WAN se tornaram necessárias pois grandes empresas com milhares de computadores precisavam trafegar grande quantidade de informações entre filiais em diferentes localidades geográficas.

Esta nova demanda não podia ser satisfeita dentro das capacidades de uma rede LAN e novos protocolos para atender às exigências de velocidade e qualidade das redes WAN foram criados.

O termo WAN é normalmente encontrado em configurações de equipamentos de rede, como modens, *access points* e roteadores, para se referir à rede externa da empresa. Essa sigla vem do inglês *Wide Area Network*, onde são necessárias para que grandes empresas trafeguem grandes quantidades de informações entre filiais em diferentes locais, como mostra a figura 1.

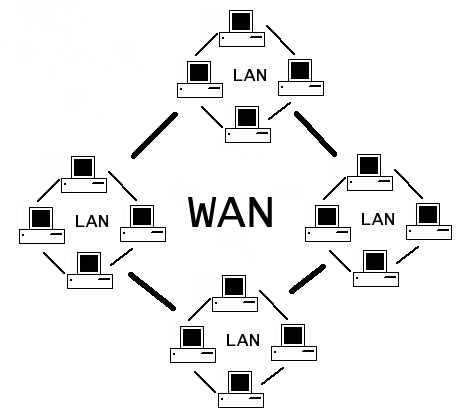


Figura 1 - Representação de uma rede WAN.

**Fonte:** https://www.palpitedigital.com/i/876/rede-lan-rede-wan-467.webp

## 2.2 SISTEMA OPERACIONAL

Um sistema operacional consiste em um conjunto de programas cuja a função é gerenciar os recursos do sistema fornecendo uma interface entre o computador e o usuário. Estes programas podem oferecer serviços de compartilhamento de arquivos, impressoras, entre outros dispositivos através da rede, onde são chamados de servidores e os programas que permitem utilizar esses serviços são os clientes.

O sistema que faz comunicação entre o hardware e os demais softwares. O Sistema Operacional cria uma plataforma comum a todos os programas utilizados. Exemplos: Dos, Unix, Linux, Mac OS, OS-2, Windows NT.

Considerado um dos sistemas operacionais mais utilizados em todo o mundo, o Windows começou a ser projetado em 1981, mas foi apenas em 1993 que sua primeira versão foi lançada.

Inicialmente, o Windows oferecia apenas uma interface gráfica, funcionando unicamente em programas em modo gráfico. Desde então, este sistema operacional tem evoluído, tanto em termos de funcionalidades, design e eficiência. Até o momento, o Microsoft já conta com dezenas de atualizações, sendo as mais conhecidas: Windows 95, Windows 98, Windows 2000, Windows XP, Windows Vista e, mais recentemente, o Windows 10.

O Linux é um sistema operacional, assim como o Windows da Microsoft e o Mac OS da Apple. Ele foi criado pelo finlandês Linus Torvalds, e o nome é a mistura do nome do criador com Unix, um antigo sistema operacional da empresa de mesmo nome.

O Linux não é totalmente um sistema operacional. Ele é o que se chama de kernel (o núcleo do sistema operacional). O núcleo do sistema é o responsável pela integração dos dispositivos de hardware do computador com os programas (softwares). Como o Linux está disponível de graça e com código-fonte aberto, qualquer pessoa ou organização pode juntá-lo a um conjunto de softwares para criar um sistema operacional customizado. Cada uma destas customizações é chamada de distribuição ou distros.

## 2.3 KVM

O KVM (*Kernel-based Virtual Machine*) é uma solução de virtualização completa para Linux em hardware x86 que contém extensões de virtualização (Intel VT ou AMD-V). Ele consiste em um módulo do kernel carregável, kvm.ko, que fornece a infraestrutura de virtualização central e um módulo específico do processador, kvm-intel.ko ou kvm-amd.ko.

Usando o KVM, é possível executar várias máquinas virtuais executando imagens Linux ou Windows não modificadas. Cada máquina virtual possui hardware virtualizado particular: uma placa de rede, disco, adaptador gráfico, etc.

KVM é um software de código aberto. O componente do kernel do KVM está incluído no Linux mainline, a partir do 2.6.20. O componente *userspace* do KVM está incluído no QEMU principal, a partir de 1.3.

## 2.4 MÁQUINA VIRTUAL

Uma máquina virtual é um software de ambiente computacional em que um sistema operacional ou programa pode ser instalado e executado. De maneira mais simplificada, podemos dizer que a máquina virtual funciona como um “computador dentro do computador”.

Máquinas virtuais são extremamente úteis no dia a dia, pois permitem ao usuário rodar outros sistemas operacionais dentro de uma janela, tendo acesso a todos os softwares que precisa. Elas são usadas em diversos casos, como no lançamento de programas e SOs ainda em estágio de desenvolvimento. Dessa forma, você não se torna refém de aplicativos inacabados que podem apresentar diversos bugs.

A máquina virtual irá alocar, durante a execução de sistemas operacionais, uma quantidade definida de memória RAM. Ela normalmente emula um ambiente de computação física, mas requisições de CPU, memória, disco rígido, rede e outros recursos de hardware serão todos geridos por uma “camada de virtualização” que traduz essas solicitações para o hardware presente na máquina.

A camada de virtualização utilizada nas virtualizações de maquinas é a *hypervisor*, da qual é uma camada de software entre o hardware e o sistema operacional, responsável por simular todo o hardware da máquina física fazendo com que as máquinas virtuais executem de forma isolada.

O hypervisor possui dois tipos, sendo o bare-metal e o hosted. Hypervisor do tipo bare-metal interage diretamente com o hardware da máquina física, como mostra a Figura 2. Sendo completamente independente do sistema operacional do host. Já no tipo hosted, o hypervisor roda sobre o sistema operacional do host, sendo isto possível em qualquer tipo de SO.

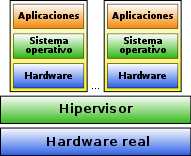


Figura 2 - Tipo *bare-metal*

## 2.5 ACESSO REMOTO

É uma tecnologia que permite que haja uma conexão entre computadores ou outros artigos eletrônicos, sem a necessidade de uma ligação física entre os aparelhos. A organização tem um servidor, e por meio desses equipamentos é possível acessar o sistema corporativo.

Um elemento importante para entender esse recurso é a conexão à distância, sem a necessidade de algo que conecte os aparelhos fisicamente. Além disso, outra questão fundamental é o uso de uma rede que interconecta os aparelhos e cria um canal de comunicação entre diferentes estações de trabalho.

Outro fator marcante desse recurso para as organizações é que ele permite a comunicação com um sistema privado. Assim, a corporação pode abrir plataformas e locais no meio virtual para que os colaboradores possam acessar os dados do empreendimento e realizar suas tarefas de onde estiverem.

Logo quando surgiu a rede de computadores, não demorou muito para aparecer métodos que permitiam o acesso a máquinas e a troca de arquivos entre computadores a distância. Dessa forma, o administrador conseguia, do seu terminal, controlar todas as outras máquinas da rede remotamente, não precisando estar presente no local. Nesse sentido, o protocolo mais comum era o Telnet, que possibilitava uma comunicação interativa entre a máquina cliente e o servidor. No entanto, como essa comunicação ocorria por meio de texto puro, ou seja, sem criptografia, qualquer usuário mal-intencionado que interceptasse a conexão tinha acesso a todas as informações, inclusive as mais sensíveis.

Diante desse problema, o engenheiro e pesquisador finlandês Tatu Ylönen desenvolveu, em 1995, um protocolo muito mais seguro: o SSH, sigla para *Secure Shell* ou, em bom português, terminal seguro. Basicamente, ele realiza as mesmas funções que o Telnet e muito mais, com uma (boa) diferença: toda transferência de dados, do servidor à máquina cliente e vice-versa, é criptografada.

SSH é o acrônimo de Secure Shell, que traduzido literalmente seria algo como “cápsula segura”. O protocolo SSH permite se conectar a um servidor remoto pela internet, de maneira segura. A conexão exige autenticação dos dois lados (servidor e computador) e é criptografada. Assim, mesmo se alguém interceptar o pacote de dados que está sendo transmitido, não será possível visualizar o conteúdo da mensagem, pois apenas os computadores que estão conectados entre si possuem a chave para descriptografá-la.



Figura 3 - Representação de Acesso Remoto

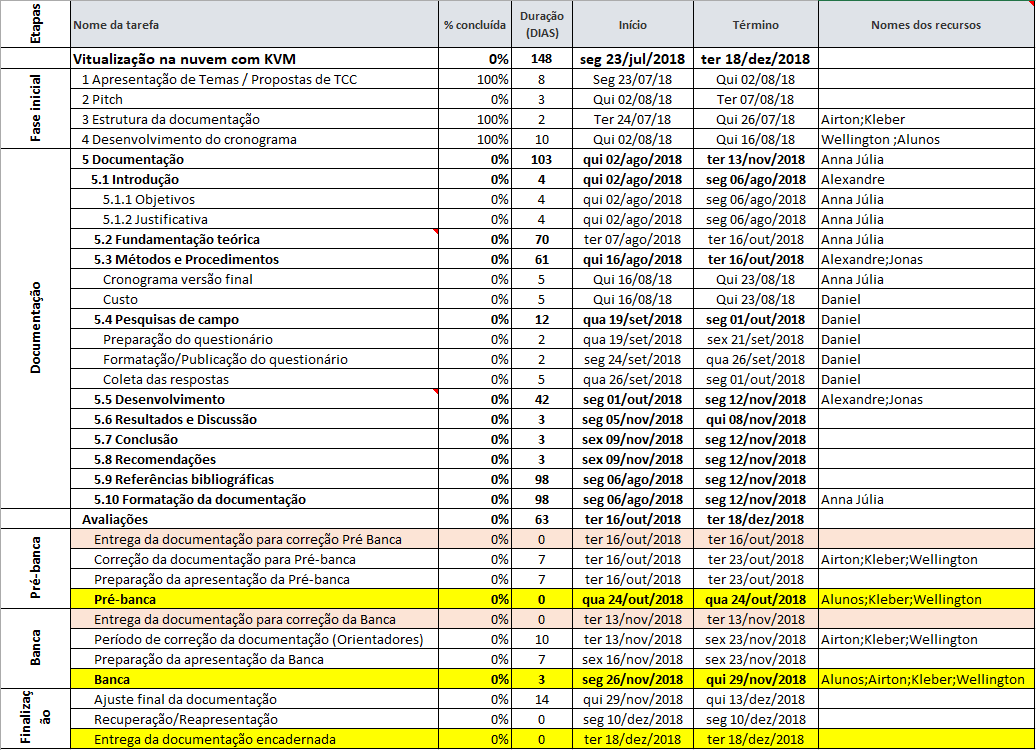
Fonte:http://s.glbimg.com/po/tt/f/original/2013/10/18/acesso-remoto-da-acesso-seguro-a-servidores-de-empresas-e-possibilita-controlar-outras-maquinas.jpg

# 3. MÉTODO E PROCEDIMENTOS

## 3.1 CRONOGRAMA

As atividades à serem desenvolvidas seguirão as seguintes etapas, conforme o cronograma do projeto

Tabela 1 – Cronograma



## 3.2 CUSTOS

### 3.2.1 HARDWARE

De acordo com pesquisas realizadas pelo grupo ao decorrer do projeto, em sites e lojas, foi selecionado três servidores, sendo eles um montado pela empresa *VirtuaCloud* e dois já montado por outras distribuidoras de servidores, levando em conta que era necessário um servidor com uma quantidade alta de HD (Hard Disk) e de Memória RAM.

Com isso foi selecionado um servidor, Servidor-Storage Silix® X1200H24 V6 Intel Xeon 3.5 Ghz / 8GB / SSD / 48TB / RAID / Hot-Swap / Win 10, da qual o sistema operacional seria mudado para adequação do projeto, na tabela 2, são mostrados os componentes desse servidor, sendo seu preço de R$ 31.399,00.

Tabela 2 - Servidor Storage Silix X1200H25 V6

|  |  |
| --- | --- |
| **Equipamento** | **Modelo** |
| Processador | Intel Xeon E3-1230 V6 Kaby Lake |
| Placa mãe | Intel Server – Silver Pass |
| Memória | Memória Kingston 3 ECC DDR4 64GB |
| Fonte | Fonte Fixa 700W |
| Unidade de armazenamento (HD) | 24x 2TB SATA3 6.0Gb/s |
| Unidade de armazenamento (SSD) | 1x 120GB 2.5' SATA3 6.0 Gb/s |
| Sistema operacional | MICROSOFT WINDOWS 10 PRO |

**Fonte:** https://www.walmart.com.br/item/4590826/sk

O Segundo servidor levantado foi o Servidor Silix® E5-2600dr V4 Intel Xeon E5-2660 V4 32gb Ddr4 Ecc / 1tb / Dvd / Rack 2u / Dual Xeon, não possui sistema operacional, da qual seria instalado e configurado de acordo com o projeto e adequabilidade ao cliente, sendo seu preço de R$ 30.900,00, na tabela 3 mostra seus componentes.

Tabela 3 - Servidor Silix E5-2600dr V4

|  |  |
| --- | --- |
| **Equipamento** | **Modelo** |
| Processador | Intel® Xeon® E5-2660 V4 Broadwell |
| Placa mãe | Intel Server – Cottonwood Pass |
| Memória | Memória Kingston 14 ECC DDR4 256GB |
| Fonte | Fonte Fixa 650W |
| Unidade de armazenamento (HD) | 1x 1TB SATA3 6.0Gb/s |
| Unidade de armazenamento (SSD) | NÃO INCLUSO |
| Sistema operacional | NÃO INCLUSO |

**Fonte:** https://www.americanas.com.br/produto/17438964/servidor-silix-e5-2600dr-v4-intel-xeon-e5-2660-v4-32gb-ddr4-ecc-1tb-dvd-rack-2u-dual-xeon

O terceiro servidor levantado foi um servidor montado pela VirtuaCloud, da qual na tabela abaixo mostra as peças utilizada pela empresa, além de do preço de cada componente e a quantidade utilizada, como mostra a tabela 4, sendo assim o servidor ficou com o preço de R$ 36.582,10.

Tabela 4 - Servidor Montado pela VirtuaCloud

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Equipamento** | **Modelo** | **Preço** | **Quant** | **Total** |
| Processador | Processador Intel XEON E5 BX80660E52660V4 | R$ 9.407,65 | 1 Un. | R$ 9.407,65 |
| Placa mãe | Intel Dbs2600cw2r Dual Xeon E5-2600v3/V4 16x Ddr4 2xrede Gbe Lga2011-3 | R$ 4.637,67 | 1 Un. | R$4.637,67 |
| Memória | Memoria dell kingston ktd-pe424/32g 32gb 2400mhz ddr4 reg ecc dimm | R$ 3.346,46 | 4 Un. | R$ 13.385,84 |
| Fonte | Fonte Delta Modelo Tdps-600ab A | R$ 350,00 | 2 Un. | R$ 700,00 |
| Unidade de armazenamento (HD) | HD 10 TB SATA 3 256 MB 7200 RPM SEAGATE | R$ 3.144,52 | 2 Un. | R$ 6.289,04 |
| Unidade de armazenamento (SSD) | Ssd Kingston 480GB 2.5' SATA3 6.0 Gb/s | R$ 2.161,90 | 1 Un. | R$ 2.161,90 |
| Sistema operacional | DEBIAN LINUX 9 | - | 1 Un. | - |
| Custo final: |  |  |  | R$ 36.582,10 |

**Fonte:** Autoria Própria

Com isso foi calculado a média do preço dos servidores escolhido, do qual o primeiro valor obtido no servidor foi de R$ 31.399,00, esse valor foi somado ao valor obtido no segundo servidor que foi de R$ 36.582,10, juntamente com o obtido no terceiro que foi de R$ 36.582,10.

Após a soma foi obtido o valor de R$ 98.881,10, esse valor foi divido por três, pois foi a quantidade de orçamento feito, da qual a média obtida foi de R$ 32.960,37 por servidor.

### 3.2.2 CUSTOS GERAIS

Realizamos diversas pesquisas sobre o salário de um profissional de TI, e o salário de um técnico de redes de computadores no Brasil. O salário de um técnico em rede pode variar de R$ 1.961,38 a R$ 3.087,57 com média de R$ 2.524,47/mensal para 42 horas trabalhada, no período de 03/2018 a 10/2018, segundo o *site* salarios.com.br. Já o salário de um profissional de TI, pode variar de acordo com o tipo de serviço e sua formação. Foram pesquisados valores em três fontes distintas, onde os valores encontrados foram pela mão de obra técnica, sendo eles R$00,00, R$00,00 e R$00,00. Esses valores foram obtidos em sites de TI e um arquivo de texto.

Tabela 5 - Valor da mão de obra

|  |  |
| --- | --- |
| **Empresas para mão de obra** | |
| **Empresa** | **Valor por hora** |
|  | R$ |
|  | R$ |
|  | R$ |
| **Custo médio por hora** | R$ |

Fonte: Autoria própria

A empresa X, com a proposta de serviços na área de tecnologia, cobra R$ 00,00 a hora técnica, conforme a Tabela X.

A empresa Y conforme visto na Tabela X acima, publicou um artigo em seu site no ano de X, qual o valor a ser cobrado em hora técnica sendo ele R$00,00, e como foi calculado esse valor.

Por fim, foi encontrado um arquivo de texto na internet, apresentando o valor por hora da empresa X, conforme a Tabela X, preço de R$00,00 a hora. Para obtermos o valor da mão de obra por hora do projeto *VirtuaCloud,* foi retirado a média entre os três valores, obtendo o valor final de R$ 00,00.

### 3.2.3 CUSTO MÃO DE OBRA ESPECIALIZADA

Para se obter o valor da mão de obra, é preciso multiplicar o valor cobrado por hora, R$ 00,00, com o total de horas trabalhadas e o número de integrantes para o desenvolvimento do projeto.

Tabela 6 - Trabalho durante o turno

|  |  |  |
| --- | --- | --- |
| **Trabalho durante o turno** | | |
| **Dias** | **Horas** | **Total de Horas** |
|  |  |  |

Fonte: Autoria própria

Conforme a Tabela x, durante o turno de trabalho, foram X dias de projeto, nesses X dias, trabalhamos X horas por dia, totalizando X horas de trabalho durante o turno.

Tabela 7 - Trabalho fora do turno

|  |  |  |
| --- | --- | --- |
| **Trabalho fora do turno** | | |
| **Dias** | **Horas** | **Total de Horas** |
|  |  |  |

Fonte: Autoria própria

Como mostrado na Tabela X, foi realizado X dias de trabalho fora do turno, (final de semana), a cada um dia do final de semana foram trabalhados X horas, totalizando X horas trabalhadas fora do turno.

Tabela 8 - Custo final da mão de obra

|  |  |  |  |
| --- | --- | --- | --- |
| **Custo por hora** | **Horas Trabalhadas** | **Integrantes** | **Total** |
|  |  | 4 | R$ |

Fonte: Autoria própria

Por fim, obtemos o valor total da mão de obra, somando a quantidade de horas trabalhadas durante e fora do turno, X horas, multiplicamos com o salário por hora médio de um técnico, R$ X, e multiplicamos pelo total de integrantes para a realização do trabalho, X integrantes, totalizando o valor de R$ X como diz a Tabela X. Para cada integrante do grupo de trabalho, será o valor de R$ X.

## 3.3 TIPO DE PESQUISA

Em caráter cartesiano,foi constatado a viabilidade do projeto através as pesquisas realizadas com pessoas de diferentes áreas de atuação.

Através da pesquisa de campo, será possível analisar a necessidade da empresa para adquirir um servidor para realizar a virtualização das máquinas virtuais em uma nuvem privada.

## 3.4 CARÁTER DA PESQUISA

Ao utilizar o método descritivo**,** os dados obtidos através da pesquisa irão demonstrar o motivo pelo qual, empresas precisam de um serviço de virtualização numa nuvem privada, afim de obter melhorias.

Buscar a compreensão, quanto a dificuldade de se armazenar os dados dentro do servidor da virtualização, tendo em mente os custos os apoios que irão beneficiar as empresas que solicitarão os nossos serviços.

## 3.5 MÉTODO DE ABORDAGEM

A coleta de dados será realizada com pessoas de diversas áreas de conhecimento, desde profissionais de TI, estudantes, engenheiros até profissionais da área de elétrica, eletroeletrônica e arquitetura.

## 3.6 DADOS DA PESQUISA DE CAMPO

Após o levantamento de dados obtidos a partir da pesquisa de campo, foi analisada a viabilidade do projeto de virtualização em uma nuvem privada.

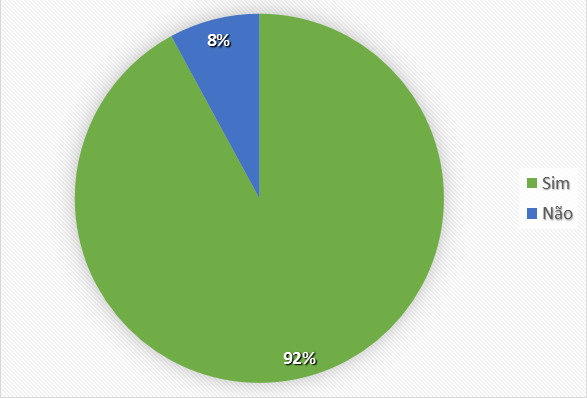


Gráfico 1 - Nível de conhecimento sobre Cloud Computer.

De acordo com o Gráfico 1, é possível notar que 92% sabem o que é Cloud Computer, e os outros 8% não sabem seu significado.

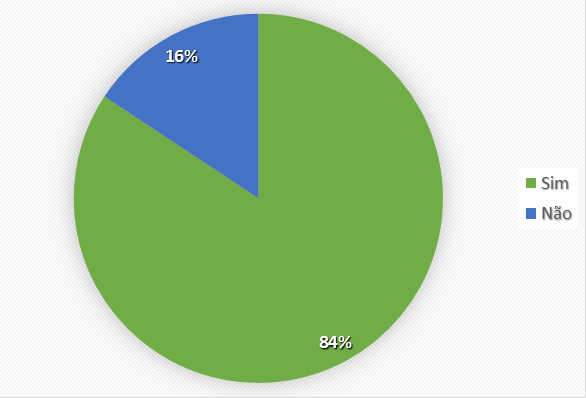


Gráfico 2 - Utilização de armazenamento na nuvem.

De acordo com os dados do Gráfico 2, a maior parte dos entrevistados utilizam armazenamento na nuvem, e alguns não utilizam.

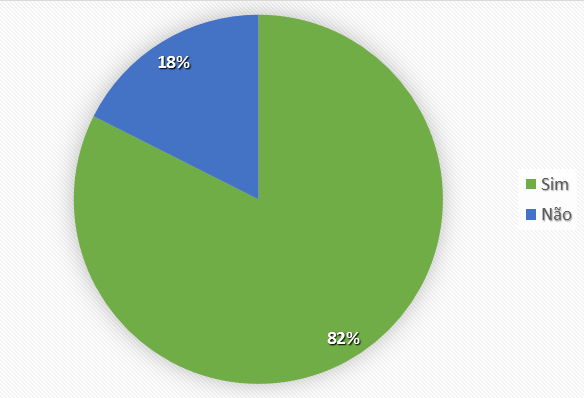


Gráfico 3 - Nível de conhecimento sobre máquina virtual.

De acordo com o Gráfico 3, 82% das pessoas entrevistadas tem conhecimento do que é uma máquina virtual, e 18% não conhecem seu significado.

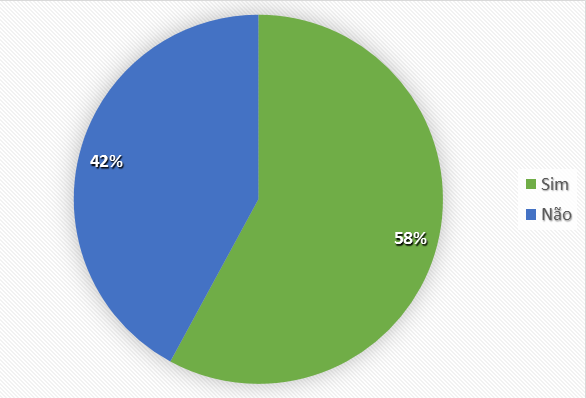


Gráfico 4 - Utilização de máquina virtual.

De acordo com os dados do Gráfico 4, 58% dos entrevistados utilizam máquina virtual, e 42% nunca utilizaram.

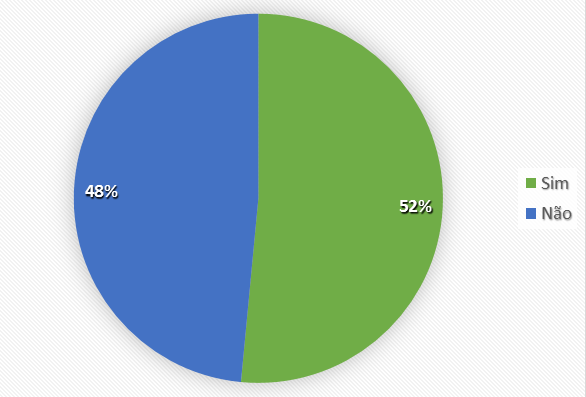


Gráfico 5 - Consumo do processador.

De acordo com o Gráfico 5, 52% acham que as máquinas virtuais consomem bastante do processador da máquina real, e 48% acham que não consomem.

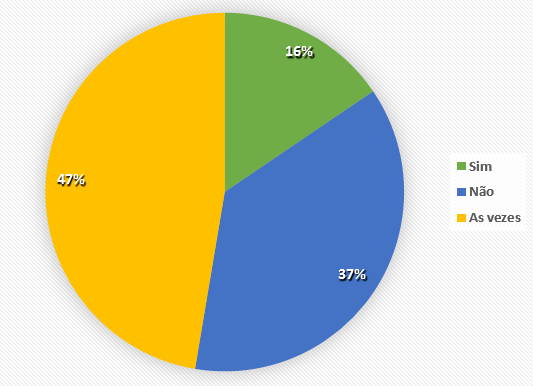


Gráfico 6 - Costume de exportar e salvar a máquina.

De acordo com os dados do Gráfico 6, 47% das pessoas entrevistadas costumam as vezes exportar e salvar a máquina virtual, 37% não fazem nenhuma das duas opções e apenas 16% as fazem.

# 4. DESENVOLVIMENTO

Para darmos início ao desenvolvimento do projeto é necessário a implementação dos serviços. Do qual são responsáveis pelo gerenciamento das maquinas, como criação de uma nova máquina, inicialização das maquinas já existentes, apagar uma máquina, entre outras funções.

O comando utilizado para instalação dos serviços necessário utilizado foi o apt-get, pois é um comando já existente na partição do Debian 9, caso o administrador não esteja como *root* é necessário a utilização do sudo.

|  |
| --- |
| # sudo apt-get install qemu-kvm libvirt-clients libvirt-daemon-system  # sudo apt-get install openssh-server |

Através do menu de usuários, o administrador de sistema utilizara ele para configurar um usuário, podendo fazer alteração no mesmo, como senhas, adicionar ou deletar um perfil. Na figura 4, mostra a primeira tela do programa da qual onde o administrador escolhera o número que deseja, para assim ser redirecionado para a tela seguinte. Caso o administrador escolha o número 1, será redirecionado para a tela de cadastro, como mostra a figura 5, assim deverá ser informado o nome do novo usuário, e em seguida a senha do mesmo, assim o novo usuário já será colocado nos grupos necessários, como o *libvirt-qemu* e o *libvirt*.

Caso o administrador escolha o número 2, será redirecionada para a tela aonde listara os usuário existentes na partição do Linux, como mostra a figura 6. Sendo assim caso escolha o número 3 – Remover, será solicitado ao usuário que ele informe o nome de usuário que deseja deletar, como mostra na figura 7. Na figura 8, é solicitado ao administrador que informe o usuário que deseja alterar a senha, caso o administrador escolha a opção 4, caso tenha encerrado todas as configurações o administrador poderá escolher a opção 9 – *exit*, da qual o administrador sairá do menu.

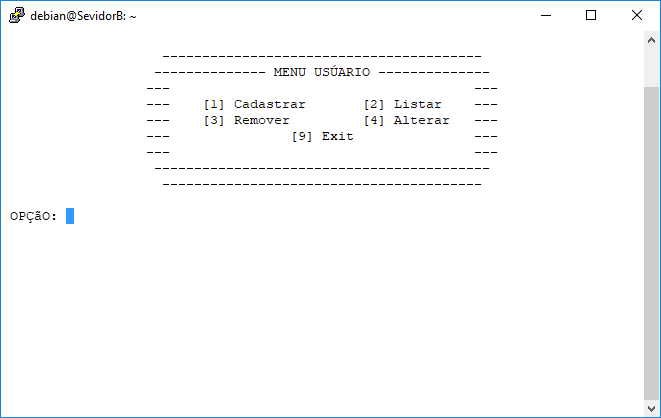


Figura 4 - Tela inicial do menu de usuário

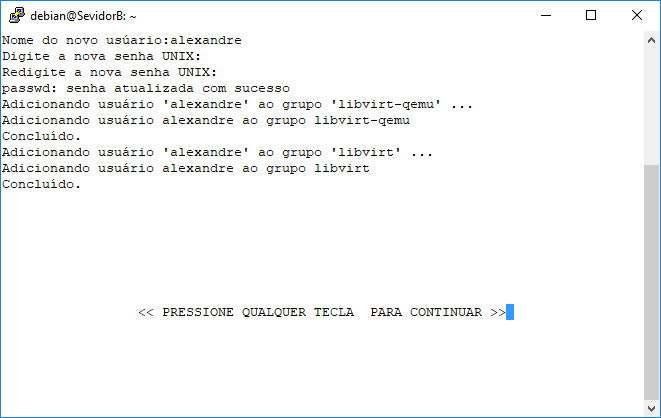


Figura 5 - Adicionando um novo usuário

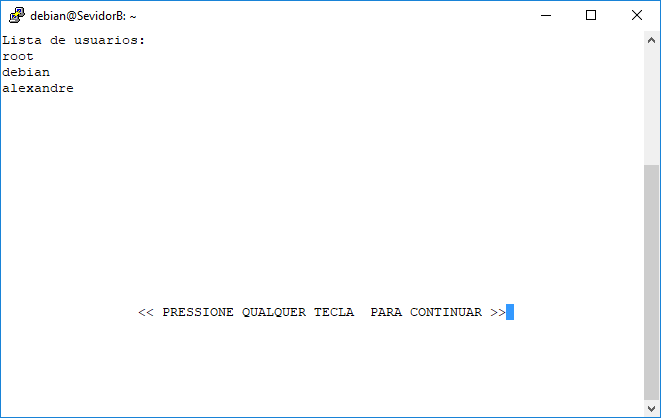


Figura 6 - Listando os usuários existentes no servidor

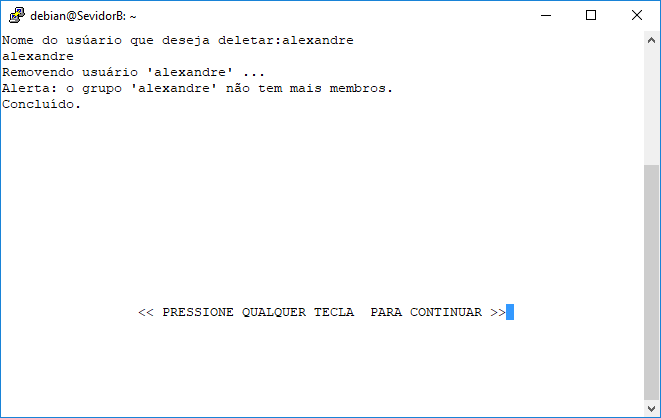


Figura 7 - Deletando um usuário criado no sistema

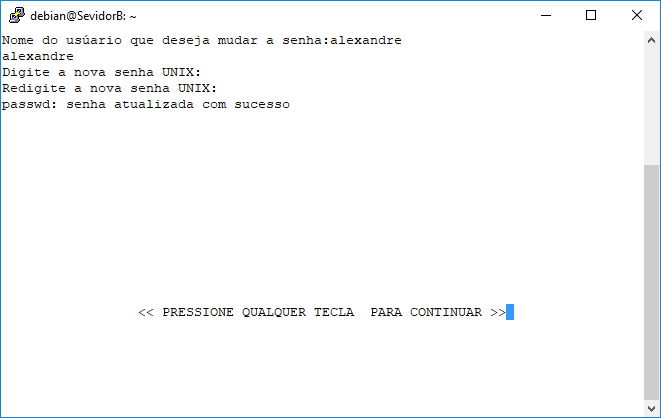


Figura 8 - Alterando a senha de um usuário já existente

Através do menu de maquinas virtuais, o usuário ou administrador utilizara para gerenciar as VM’s, como configurar uma nova máquina virtual, além de conseguir inicializar e apagar uma VM antiga. A primeira dela do programa tem as seguintes opções para o usuário sendo 1 – criar, 2 – listar, 3 – iniciar, 4 – deletar, 5 – pausar, 9 – *exit*. Como mostra na figura 9.

Caso o usuário digite 2 será solicitado a lista das maquinas já criada antes pelo usuário, que não terá acesso a maquinas criada por outro usuário. Como mostra a figura 10.

Na figura 11, temos o usuário solicitando ao programa a inicialização de uma máquina virtual já existente, para isso o usuário deverá informar o nome da máquina criada, para que a mesma seja iniciada. Assim o usuário deva selecionar a opção número 3.

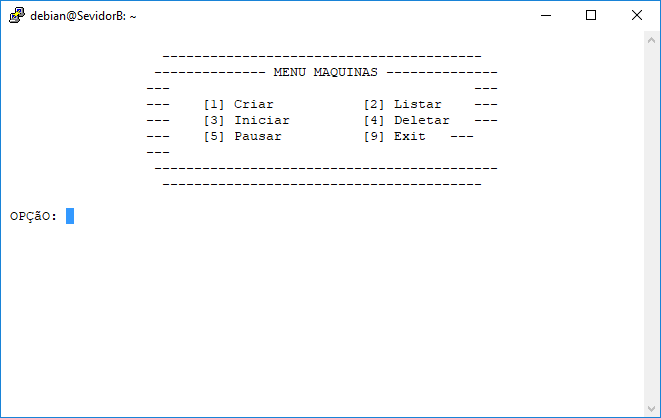


Figura 9 - Tela inicial do menu de gerenciamento de VMs

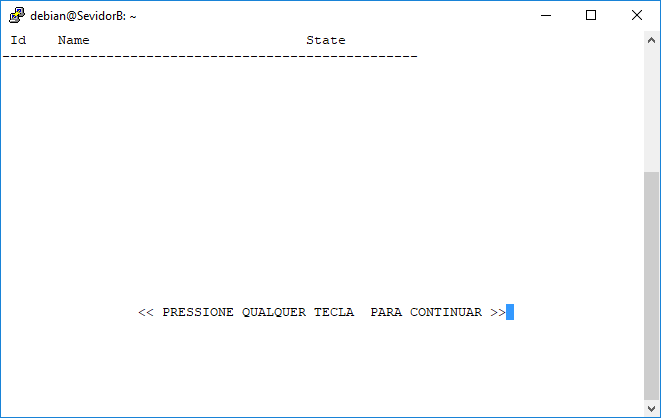


Figura 10 - Listando as máquinas virtuais criada pelo usuário

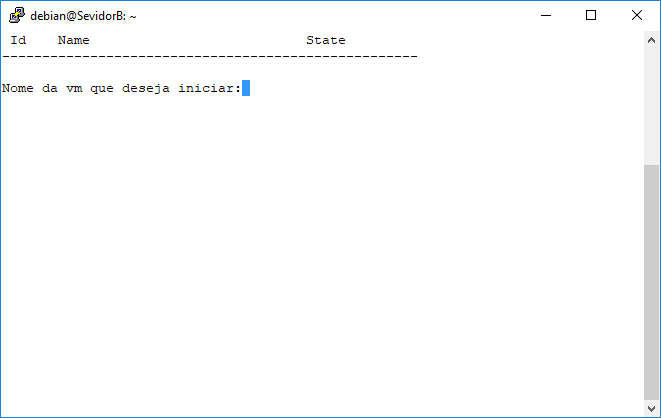


Figura 11 - Iniciando a VM criada pelo usuário

Caso o usuário deseja deletar uma máquina antiga ter que selecionar a opção número 4, assim será solicitado o nome da VM que deseja deletar, para que a mesma seja apagada e libere espaço na máquina. Como mostra a figura 12.

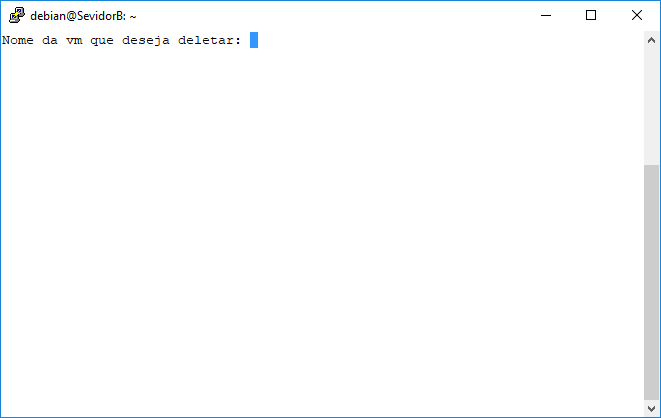


Figura 12 - Deletando uma VM criada pelo usuário

Para a criação de uma nova máquina virtual o usuário deverá ter selecionado a opção número 1, da qual redirecionar a para a tela onde o usuário, devera selecionar o tipo de máquina que deseja importar, como mostra na figura 13.

Com Isso caso o usuário selecione a opção 1 – Debian, será enviado para a tela aonde devera selecionar a versão do Debian em que deseja, ou seja, a partição existente no momento e mais utilizada no momento são a Debian 7, Debian 8 e Debian9. Como é mostrado na figura 15.

Caso o usuário selecione a opção Windows, número 2, será redireciona para a tela onde contém as versões do Windows existente, como a Windows XP, Windows 7, Windows 8, Windows 10, além dos Windows server, como o Windows server 2008 e 2012. Como mostra a figura 14.

Quando o usuário selecionar a opção 3 – Kali, ou qualquer outra máquina como Windows 7 ou Debian 7, o programa solicitara ao usuário informações da máquina virtuais como nome, tamanho de HD tendo um máximo de 50 Gb. Quantidade de memória RAM, tenho um máximo de 3 GB, além da quantidade de CPU sendo um máximo de 2 cpus por máquina virtuais. Como mostra na figura 16.

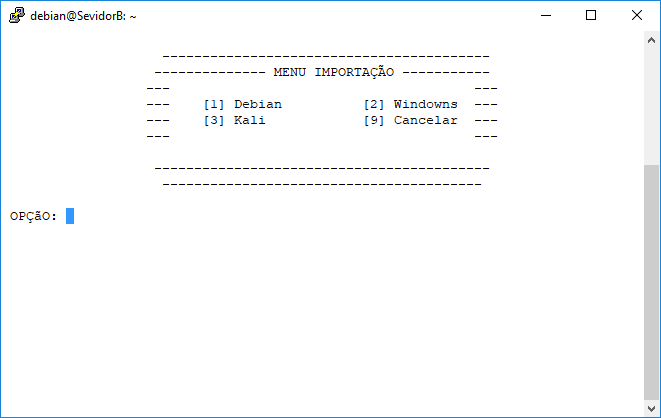


Figura 13 - Escolhendo o tipo de VM que deseja criar

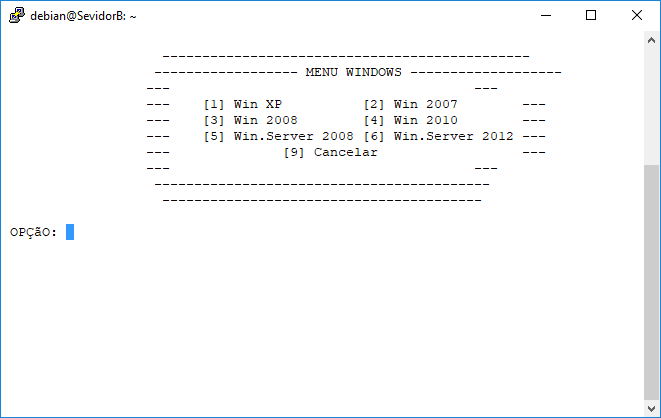


Figura 14 - Menu de importação das VMs Windows

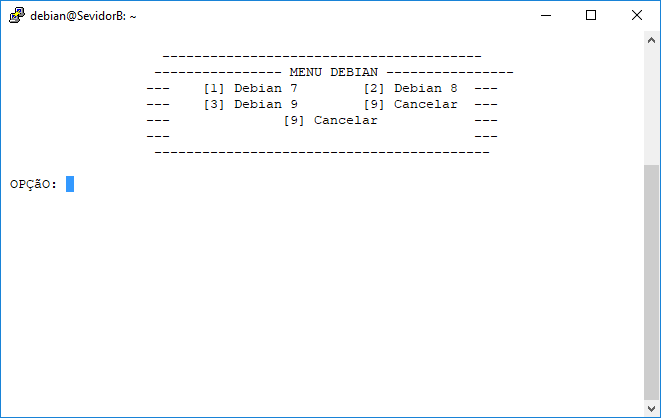


Figura 15 - Menu de importação de VMs Debian

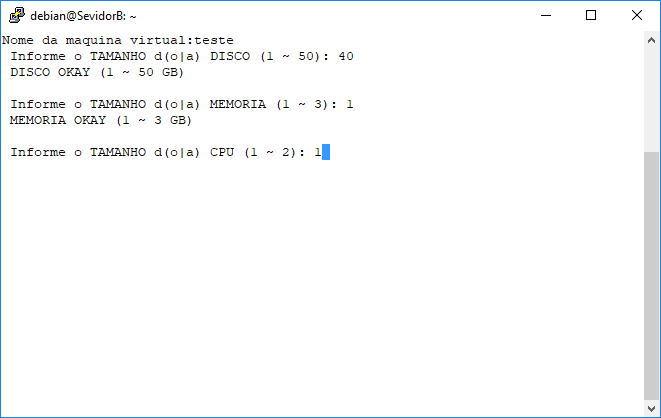


Figura 16 - Informações pedidas para a criação de uma nova VM

# 5. RESULTADOS E DISCUSSÃO

No início do projeto, foi necessário muitas pesquisas e embasamento teórico, a fim de adquirir mais conhecimento sobre o assunto. Foi pesquisado cada item da fundamentação teórica, tanto em páginas virtuais quanto em livros na biblioteca, além de tutoriais que ajudassem na parte prática. Nessas pesquisas, foi feita a escolha da ferramenta KVM para a implementação da virtualização na nuvem. O ponto principal para a escolha dessa ferramenta foi a facilidade de instalação e de criação de máquinas virtuais. Por outro lado, mesmo a vinda do KVM em 2011 tivemos algumas dificuldades para poder compreender as pesquisas porquê quase a maioria estava em outro idioma, por conta disto levou um pouco mais de tempo para poder entendê-las.

Outra dificuldade foi a busca de vídeos explicativos que atendessem a mesma distribuição Linux que foi utilizada, na qual foi preciso adaptar formas diferentes para implementação, além dos comandos utilizados. Muitos dos arquivos de configuração não foram encontrados, por estarem em uma distribuição diferente da de muitos vídeos e arquivos explicativos. Com isso, foi preciso criar diversos arquivos de configuração, a fim do funcionamento do projeto.

Com a finalização do projeto, foi concluído que todos os objetivos foram alcançados com sucesso, como a implementação da virtualização com o KVM, a ligação do protocolo SSH para a conexão dos usuários ao servidor local onde se encontra as máquinas virtuais.

Foi notado as vantagens que a virtualização proporciona às empresas, uma vez que estabelece segurança e armazenamento dos arquivos que contem nas máquinas virtuais, identificando o acesso dos usuários com o sistema de login em que se oferece no momento de se fazer a implementação.

Em conformidade com a pesquisa de campo aplicada com pessoas de áreas diferentes, foi concluído que a implementação do KVM é de interesse para a maioria dos entrevistados, visto que cerca de 90% atuam na área de TI e conhecem para que serve tal ferramenta e 10% desconhece este tipo de assunto.00

# 6. CONCLUSÃO

Conclui-se que nosso projeto teve o principal motivo de criar uma infraestrutura de virtualização local com custo acessível para as empresas e instituições que trabalham com desenvolvimento de software ou mesmo possibilitando uma possível descoberta erros de sistemas operacionais.

O desenvolvimento do VirtuaCloud possibilitou uma análise de como um software de virtualização de maquinas pode ser gerenciado em uma rede local mantendo uma segurança, utilizando um sistema de login.

Além disso, também permitiu uma pesquisa de campo para obter dados mais consistentes sobre as etapas do processo, com pessoas que possuem um certo grau de conhecimento em informática e também professores, para entender se será acessível a empresas que o utilizarão nosso projeto. Com essa pesquisa a podemos considerar que as pessoas têm problemas de salvar maquina virtuais, e também deslocá-las em outros computadores.

Com os resultados da pesquisa de campo, começamos a desenvolver nosso software utilizando ferramentas como o KVM (Kernel-based Virtual Machine) que foi essencial para o desenvolvimento do projeto pois oferecia uma interface leve e uma configuração simplificada. Além disso utilizamos o cloud computing (computação na nuvem) que utilizamos para salvar as maquinas virtuais em um servidor que está conectado em uma rede local, onde as maquinas virtuais estão sendo utilizadas nos computadores que estão conectados na mesma rede do servidor.

Apesar de nosso projeto só utilizar ferramentas computacionais gratuitas, para usufruir dos serviços de nosso software precisara de um servidor. Com isso fizemos um orçamento de três servidores, que podem ser utilizados caso a empresa não possua.

Com isso conclui-se que este trabalho trouxe uma grande contribuição por encontrar uma solução para problemas críticos da virtualização, compatível com a disponibilidade financeira de diversas empresas e instituições e que, de certa forma, se adapta à atual forma da virtualização utilizada das empresas.

# 7. **RECOMENDAÇÕES**

Para caso alguém queria prosseguir com o nosso projeto com melhorias, seria interessante fazer uma interface web, para assim ficar com um visual agradável para o usuário, e também alocar um gerenciamento de usuários via web, pois assim seria mais seguro, e irá possuir um design web.

Recomendamos também, tentar fixar no projeto um disco diferencial e disco dinamicamente alocado, para assim ter menos consumo de memória no HD.

# REFERÊNCIAS

ALECRIM, Emerson**. Virtualização. 2013.** Disponível em: <https://www.infowester.com/virtualizacao.php>. Acesso em: 01 ago. 2018.

BENATO, Gabriel. **Diferenças de kvm e openvz para virtualização. 2017.** Disponível em: <https://www.brasilwebhost.com.br/blog/diferencas-de-kvm-e-openvz-para-virtualizacao/>. Acesso em: 01 ago. 2018.

**REDE MAN.** Disponível em: <https://cddiogo.wordpress.com/man-rede-metropolitana/>. Acesso em: 01 ago. 2018.

MESQUITA, Juciely; RONNER, Rodrigo; NEIVA, Jessica**. Virtualização e seus Benefícios para empresas. 2011.** Disponível em: <https://www.aedb.br/seget/arquivos/artigos11/55714687.pdf>. Acesso em: 01 ago. 2018.

FORZZA AMARAL, Allan Francisco. **Definição de Redes de Computadores. 2012.** Disponível em: <http://redeetec.mec.gov.br/images/stories/pdf/eixo\_infor\_comun/tec\_inf/081112\_rede\_comp.pdf>. Acesso em: 01 ago. 2018.

MAIA BRITO FILHO, Francisco Xavier. **Definição de Redes de Computadores. 2010.** Disponível em: <http://www.teleco.com.br/tutoriais/tutorialwifimanaus1/pagina\_2.asp>. Acesso em: 01 ago. 2018.

PINHEIRO, Ricardo. **Definição de Redes de Computadores. 2011.** Disponível em: <http://cooperati.com.br/2011/06/28/virtualizacao-com-kvm-2/>. Acesso em: 01 ago. 2018.

CAMARGO CARLOS, Henrique. **Configuração de máquinas virtuais com KVM. 2015.** Disponível em: <http://henriquecc.info/linux/kvm.html>. Acesso em: 01 ago. 2018

LEITÃO, Breno; SCARAPICCHIA JUNIOR, Pedro; RODRIGUES CRUZ, Erlon; ROY CHOWDHURY, Pratip**. Virtualização com KVM. 2014.** Disponível em: <https://www.ibm.com/developerworks/br/local/linux/instalandokvmnolinux/index.html>. Acesso em: 01 ago. 2018.

D´AVILLA FERREIRA, Luciano**. Virtualização com KVM. 2014**. Disponível em: <http://montanha2004.blogspot.com/p/virtualizacao.html>. Acesso em: 01 ago. 2018.

**O MANUAL do Administrador Debian: 12.2. Virtualização.** Disponível em: <https://debian-handbook.info/browse/pt-BR/stable/sect.virtualization.html>. Acesso em: 02 ago. 2018.

BESSA MONQUEIRO, Julio Cesar. **Debian em desktops: Usando o KVM (Qemu). 2010.** Disponível em: <https://www.hardware.com.br/guias/debian-desktops/kvm-qemu.html>. Acesso em: 02 ago. 2018.

SAKHAVALA, Pradip**. Como instalar e configurar o servidor VNC no Debian 9 Stretch. 2017.** Disponível em: <https://linoxide.com/debian/configure-vnc-server-debian-9-stretch/>. Acesso em: 20 ago. 2018.

NEGROMONTE, EMANUEL. **Como corrigir o erro não foi possível obter trava /var/lib/apt/lists/lock. 2017.** Disponível em: <https://sempreupdate.com.br/corrigir-erro-nao-foi-possivel-obter-trava-var-lib-apt-lists-lock/>. Acesso em: 03 ago. 2018.

**QUAL TIPO DE DISCO ESCOLHER EM MÁQUINAS VIRTUAIS NO HYPER-V?. 2016.** Disponível em: <https://blog.infomach.com.br/qual-tipo-de-disco-escolher-em-maquinas-virtuais-no-hyper-v/>. Acesso em: 03 ago. 2018.

**COMO acessar Virtual Machine (feita em qemu-kvm) [fechada]. 2016.** Disponível em: <https://pt.stackoverflow.com/questions/106791/como-acessar-virtual-machine-feita-em-qemu-kvm>. Acesso em: 02 ago. 2018.

**GUIA de acesso remoto: SSH, FreeNX e VNC. 2016.** Disponível em: <https://www.hardware.com.br/comunidade/guia-ssh/292641/>. Acesso em: 02 ago. 2018.

**COMO instalar VNC no linux remoto?. 2006.** Disponível em: <https://www.hardware.com.br/comunidade/instalar-linux/1316362/>. Acesso em: 02 ago. 2018.

**APOSTILA DE SHELL SCRIPT PARA INICIANTES. 2012.** Disponível em: <https://www.vivaolinux.com.br/dica/Apostila-de-Shell-Script-para-iniciantes>. Acesso em: 02 ago. 2018.

**FREENX. 2010.** Disponível em: <https://www.vivaolinux.com.br/topico/FreeNX/FreeNX-3/>. Acesso em: 02 ago. 2018.

RICHARDSON, Tristan. **Vncviewer. 2018.** Disponível em: <https://tigervnc.org/doc/vncviewer.html>. Acesso em: 02 ago. 2018.

RASMUSSEN, Bruna**. LAN, WLAN, MAN, WAN, PAN: conheça os principais tipos de redes.** Disponível em: <https://canaltech.com.br/infra/lan-wlan-man-wan-pan-conheca-os-principais-tipos-de-redes/>. Acesso em: 02 ago. 2018.

BENATO, Gabriel**. Tecnologias promissoras: PAN - rede sem fio de área pessoal. 2011.** Disponível em: <https://www.tecmundo.com.br/bluetooth/11029-tecnologias-promissoras-pan-rede-sem-fio-de-area-pessoal.htm>. Acesso em: 02 ago. 2018.

**O QUE é um sistema operacional ?.** Disponível em: <https://edu.gcfglobal.org/pt/informatica-basica/o-que-e-um-sistema-operacional/1/>. Acesso em: 20 ago. 2018.

KURTZ, João**. O que é um sistema operacional?. 2015.** Disponível em: <https://www.techtudo.com.br/noticias/noticia/2015/03/linux-tudo-o-que-voce-precisa-saber-antes-de-comecar-usar.html>. Acesso em: 20 ago. 2018.

KURTZ, João. **Noções de Sistema Operacional: Windows e Linux. 2015.** Disponível em: <http://www.okconcursos.com.br/apostilas/apostila-gratis/130-informatica-para-concursos/1633-nocoes-de-sistema-operacional-windows-e-linux#.W-AwjJNKiUl>. Acesso em: 20 ago. 2018.

ALVES, Paulo. **O que é acesso remoto? Entenda tudo sobre conexão à distância. 2013.** Disponível em: <https://www.techtudo.com.br/noticias/noticia/2013/10/o-que-e-acesso-remoto-entenda-tudo-sobre-conexao-distancia.html>. Acesso em: 20 ago. 2018.

**O QUE é e como usar SSH?.** Disponível em: <https://www.cybernetfx.com/clientes/knowledgebase.php?action=displayarticle&id=71>. Acesso em: 20 ago. 2018.

**O QUE é uma rede LAN e uma rede WAN?. 201**. Disponível em: <https://www.palpitedigital.com/o-que-e-uma-rede-lan-e-uma-rede-wan/>. Acesso em: 20 ago. 2018.

MELO BARBALHO, José Valney. **Artigo Utilizando JCombobox/Jlist e JRadioButton. 2006.** Disponível em: <https://www.devmedia.com.br/hypervisor-seguranca-em-ambientes-virtualizados/3099>. Acesso em: 30 ago. 2018.

# APÊNDICE A – INSTRUMENTO DE COLETA DE DADOS

|  |
| --- |
| #!/bin/bash  PCM(){ PROMPT ;clear; PMENU;}    PROMPT()  { printf "${CE}\e[18;$(( ($(tput cols)-46 )/2+1))f<< PRESSIONE QUALQUER TECLA PARA CONTINUAR >>${FC}";  read -e -n1 PRMPT ;  local PRMPT ;  }    PMENU()  {  #GERACOR;  COLS;  }    MENU() {  tput clear  printf "${CE}\e[2;${POS2}f----------------------------------------"  printf "\e[3;${POS1}f-------------- MENU USÚARIO --------------"  printf "\e[4;${POS}f---\e[4;${POS3}f---"  printf "\e[5;${POS}f---${CSM} [1] Cadastrar [2] Listar ${CE}---"  printf "\e[6;${POS}f---${CSM} [3] Remover [4] Alterar ${CE}---"  # printf "\e[7;${POS}f---${CSM} [5] Opção 5 [6] Opção 6 ${CE}---"  printf "\e[7;${POS}f---${CSM} [9] Exit ${CE}---"  printf "\e[8;${POS}f---\e[8;${POS3}f---"  printf "${CE}\e[9;${POS1}f------------------------------------------"  printf "\e[10;${POS2}f----------------------------------------\n\n"$FC""  read -p " OPÇãO: " -n1 OPC    case "$OPC" in  1) ADICIONAR ;;  2) LISTAR ;;  3) DELETAR ;;  4) SENHA ;;  9) echo -e "\n\n Saindo...\n";sleep 0.5; exit ;;  \*) MSGM ; PMENU ;;  esac  }    COLS() {    COLS=$(tput cols)  POS=$(((COLS-44)/2+1))  POS1=$((POS+1))  POS2=$((POS+2))  POS3=$((POS2+39))  (($COLS>=44)) && MENU || { echo -e "$CVA AJUSTE A TELA!\n No MÍNIMO 44 Colunas $FC" ; PCM ; }  }    GERACOR() {  local cnt=$((1+RANDOM%31));local atb=$((RANDOM%2));  CSM="\e["$atb";38;5;"$cnt"m"  }    MSGM() { echo -e "$CVA\n\n \" OPÇãO INVáLIDA! \" $FC" ; sleep 1 ; }    ADICIONAR() {  tput clear;    read -p "Nome do novo usúario:" addusuario  test $addusuario    if [ $? != "1" ];  then  useradd $addusuario    if [ $? != "1" ];  then  passwd $addusuario  usermod --shell /bin/bash $addusuario    adduser $addusuario libvirt-qemu  adduser $addusuario libvirt  fi  fi    sleep 0.5  PCM    }    LISTAR() {  tput clear ;    echo "Lista de usuarios:" ;    #cat /etc/passwd | tr ':' '\t' |grep "/bin/bash" |cut -f1  #volta o usuario que possuem terminal /bin/bash    cat /etc/passwd | tr ':' '\t' |grep "/bin/bash" |cut -f1    sleep 0.5  PCM  }    DELETAR() {  tput clear ;    read -p "Nome do usúario que deseja deletar:" delusuario    cat /etc/passwd | tr ':' '\t' |cut -f1 |grep "${delusuario}" ;    if [ $? != "1" ];  then  deluser $delusuario ;  else  echo "Usuário não existe"  fi    sleep 0.5  PCM  }  SENHA() {  tput clear ;    read -p "Nome do usúario que deseja mudar a senha:" senha\_usuario    cat /etc/passwd | tr ':' '\t' |cut -f1 |grep "${senha\_usuario}" ;    if [ $? != "1" ];  then  passwd $senha\_usuario ;  else  echo -e "Usuario $senha\_usuario não existe."  fi    sleep 0.5;  PCM  }  PMENU |

|  |
| --- |
| #!/bin/bash  # Pasta definida na configuração.    DIRETORIO=/var/lib/libvirt/images    # Nome das imagens  KALI\_IMG=  DEBIAN7\_IMG=  DEBIAN8\_IMG=debian-8.0.0-openstack-amd64.qcow2 #https://cdimage.debian.org/cdimage/openstack/archive/8.0.0/  DEBIAN9\_IMG=debian-9-openstack-amd64.qcow2 #https://cdimage.debian.org/cdimage/openstack/archive/9.0.0-20170617/  WINXP\_IMG=  WIN7\_IMG=  WIN8\_IMG=  WIN10\_IMG=  WINSERVER08\_IMG=  WINSERVER12\_IMG=    # Tipos da VMS  TIPO\_LINUX=linux  TIPO\_WINDOWS=windows    PCM(){ PROMPT ;clear; PMENU;}  PVM(){ PROMPTVM ;clear; PVM;}    PROMPT()  { printf "${CE}\e[18;$(( ($(tput cols)-46 )/2+1))f<< PRESSIONE QUALQUER TECLA PARA CONTINUAR >>${FC}";  read -e -n1 PRMPT ;  local PRMPT ;  }    PROMPTVM()  { printf "${CE}\e[18;$(( ($(tput cols)-46 )/2+1))f<< PRESSIONE QUALQUER TECLA PARA CONTINUAR >>${FC}";  read -e -n1 PRMPT ;  local PRMPT ;  }    PMENU()  {  COLS;  }    MENU() {  tput clear  printf "${CE}\e[2;${POS2}f----------------------------------------"  printf "\e[3;${POS1}f-------------- MENU MAQUINAS --------------"  printf "\e[4;${POS}f---\e[4;${POS3}f---"  printf "\e[5;${POS}f---${CSM} [1] Criar [2] Listar ${CE}---"  printf "\e[6;${POS}f---${CSM} [3] Iniciar [4] Deletar ${CE}---"  printf "\e[7;${POS}f---${CSM} [5] Pausar [9] Exit ${CE}---"  #printf "\e[8;${POS}f---${CSM} [7] Help [8] ${CE}---"  printf "\e[8;${POS}f---\e[9;${POS3}f---"    printf "${CE}\e[9;${POS1}f------------------------------------------"  printf "\e[10;${POS2}f----------------------------------------\n\n"$FC""  read -p " OPÇãO: " -n1 OPC    case "$OPC" in  1) VM ;;  2) LISTAR ;;  3) INICIAR ;;  4) DELETAR ;;  5) PAUSAR ;;  9) echo -e "\n\n Saindo...\n";sleep 0.5; exit ;;  \*) MSGM ; PMENU ;;  esac  }    COLS() {  COLS=$(tput cols)  POS=$(((COLS-44)/2+1))  POS1=$((POS+1))  POS2=$((POS+2))  POS3=$((POS2+39))  (($COLS>=44)) && MENU || { echo -e "$CVA AJUSTE A TELA!\n No MÍNIMO 44 Colunas $FC" ; PCM ; }  }    MSGM() { echo -e "$CVA\n\n \" OPÇãO INVáLIDA! \" $FC" ; sleep 1 ; }    LISTAR() {  tput clear  virsh list --all    PCM  }    INICIAR() {  tput clear  virsh list --all    read -p "Nome da vm que deseja iniciar:" IN\_NAME    if virsh list --all | grep '$IN\_NAME'  then  virsh start $IN\_NAME  fi  PCM  }    PAUSAR() {  tput clear  virsh list --all    read -p "Nome da vm que deseja pausar/desligar: " ST\_NAME  virsh shutdown $ST\_NAME  PCM  }    DELETAR() {  tput clear    read -p "Nome da vm que deseja deletar: " DEL\_NAME    virsh destroy $DEL\_NAME    sleep 0.5;  PCM  }    VM() {  tput clear  printf "${CE}\e[2;${POS2}f-----------------------------------------"  printf "\e[3;${POS1}f-------------- MENU IMPORTAÇÃO -----------"  printf "\e[4;${POS}f---\e[4;${POS3}f---"  printf "\e[5;${POS}f---${CSM} [1] Debian [2] Windowns ${CE}---"  printf "\e[6;${POS}f---${CSM} [3] Kali [9] Cancelar ${CE}---"  #printf "\e[6;${POS}f---${CSM} [5] Win.Server 2012 [6] Win 2010 ${CE}---"  #printf "\e[7;${POS}f---${CSM} [7] Kali [8] Debian 8 ${CE}---"  #printf "\e[7;${POS}f---${CSM} [9] Cancelar ${CE}---"  printf "\e[7;${POS}f---\e[7;${POS3}f---"    printf "${CE}\e[9;${POS1}f------------------------------------------"  printf "\e[10;${POS2}f----------------------------------------\n\n"$FC""  read -p " OPÇãO: " -n1 OPCVM    case "$OPCVM" in  1) VMDEBIAN ;;  2) VMWIN ;;  3) KALI ;;  9) echo -e "\n\n Cancelando...\n";sleep 0.5; exit ;;  \*) MSGM ; PVM ;;  esac  }    KALI(){  tput clear    # Definição do virt-install debian 7.    TIPO=$TIPO\_LINUX  FORMATO=qcow2  VARIANTE=wheezy  PORTA=  LISTEN=  IMG\_DISK=$DIRETORIO/$KALI\_IMG  #MACADDRESS=openssl rand -hex 6 | sed 's/\(..\)/\1:/g; s/.$//'  #NETWORK\_PARAMS="bridge=${BRIDGE},model=virtio,mac=${MACADDRESS}"    # Inicio da configuração do usuario da maquina virtual.    read -p "Nome da maquina virtual:" NAME    READ() {  while :;do  read -p " Informe o TAMANHO d(o|a) $3 ($1 ~ $2): "  if (($REPLY>=$1 && $REPLY<=$2));then  echo -e " $3 OKAY ($1 ~ $2 GB)\n"  eval $3=$REPLY; break  else  echo -e " Tamanho D(o|a) $3 inválido\n Minimo:$1\n Máximo:$2"; continue ;  fi  done  }    # TAMANHO do disco virtual.  READ 1 50 DISCO    # TAMANHO da ram.  READ 1 3 MEMORIA    # Quantidade de CPU  READ 1 2 CPU    tput clear    echo -e "\n Os valores definidos são:\n DISCO:$DISCO\n MEMÓRIA:$MEMORIA\n CPU:$CPU\n"    #Criando disco da maquina.    CREATE\_IMG=$NAME.$FORMATO  qemu-img create -f $FORMATO $CREATE\_IMG ${DISCO}G  DISCO\_CRIADO=$CREATE\_IMG    # Comando Virt-install para instalar uma nova maquina virtual.    virt-install --import --name $NAME --memory $MEMORIA --vcpus $CPU --cpu host --disk $DISCO\_CRIADO,format=$FORMATO,bus=virtio --disk $IMG\_DISK,device=cdrom --network $NETWORK\_PARAMS --os-type=$TIPO --os-variant=$VARIANTE --graphics spice,port=$PORTA,listen=$LISTEN --noautoconsole    sleep 0.5  PCM  }    VMDEBIAN() {  tput clear  printf "${CE}\e[2;${POS2}f----------------------------------------"  printf "\e[3;${POS1}f---------------- MENU DEBIAN ----------------"  printf "\e[4;${POS}f---\e[4;${POS3}f---"  printf "\e[4;${POS}f---${CSM} [1] Debian 7 [2] Debian 8 ${CE}---"  printf "\e[5;${POS}f---${CSM} [3] Debian 9 [9] Cancelar ${CE}---"  #printf "\e[6;${POS}f---${CSM} [5] Win.Server 2012 [6] Win 2010 ${CE}---"  #printf "\e[7;${POS}f---${CSM} [7] Kali [8] Debian 8 ${CE}---"  printf "\e[6;${POS}f---${CSM} [9] Cancelar ${CE}---"  printf "\e[7;${POS}f---\e[7;${POS3}f---"    printf "${CE}\e[8;${POS1}f------------------------------------------"  printf "\e[8;${POS2}f----------------------------------------\n\n"$FC""  read -p " OPÇãO: " -n1 OPCVM    case "$OPCVM" in  1) DEBIAN7;;  2) DEBIAN8;;  3) DEBIAN9;;  9) echo -e "\n\n Cancelando...\n";sleep 0.5; exit ;;  \*) MSGM ; PVM ;;  esac  }    DEBIAN7(){  tput clear    # Definição do virt-install debian 7.    TIPO=$TIPO\_LINUX  FORMATO=qcow2  VARIANTE=wheezy  PORTA=  LISTEN=  IMG\_DISK=$DIRETORIO/$DEBIAN7\_IMG  #MACADDRESS=openssl rand -hex 6 | sed 's/\(..\)/\1:/g; s/.$//'  #NETWORK\_PARAMS="bridge=${BRIDGE},model=virtio,mac=${MACADDRESS}"    # Inicio da configuração do usuario da maquina virtual.    read -p "Nome da maquina virtual:" NAME    READ() {  while :;do  read -p " Informe o TAMANHO d(o|a) $3 ($1 ~ $2): "  if (($REPLY>=$1 && $REPLY<=$2));then  echo -e " $3 OKAY ($1 ~ $2 GB)\n"  eval $3=$REPLY; break  else  echo -e " Tamanho D(o|a) $3 inválido\n Minimo:$1\n Máximo:$2"; continue ;  fi  done  }    # TAMANHO do disco virtual.  READ 1 50 DISCO    # TAMANHO da ram.  READ 1 3 MEMORIA    # Quantidade de CPU  READ 1 2 CPU    tput clear    echo -e "\n Os valores definidos são:\n DISCO:$DISCO\n MEMÓRIA:$MEMORIA\n CPU:$CPU\n"    #Criando disco da maquina.    CREATE\_IMG=$NAME.$FORMATO  qemu-img create -f $FORMATO $CREATE\_IMG ${DISCO}G  DISCO\_CRIADO=$CREATE\_IMG    # Comando Virt-install para instalar uma nova maquina virtual.    virt-install --import --name $NAME --memory $MEMORIA --vcpus $CPU --cpu host --disk $DISCO\_CRIADO,format=$FORMATO,bus=virtio --disk $IMG\_DISK,device=cdrom --network $NETWORK\_PARAMS --os-type=$TIPO --os-variant=$VARIANTE --graphics spice,port=$PORTA,listen=$LISTEN --noautoconsole    sleep 0.5  PCM  }    DEBIAN8(){  tput clear    # Definição do virt-install debian 7.    TIPO=$TIPO\_LINUX  FORMATO=qcow2  VARIANTE=  PORTA=  LISTEN=  IMG\_DISK=$DIRETORIO/$DEBIAN8\_IMG  #MACADDRESS=openssl rand -hex 6 | sed 's/\(..\)/\1:/g; s/.$//'  #NETWORK\_PARAMS="bridge=${BRIDGE},model=virtio,mac=${MACADDRESS}"    # Inicio da configuração do usuario da maquina virtual.    read -p "Nome da maquina virtual:" NAME    READ() {  while :;do  read -p " Informe o TAMANHO d(o|a) $3 ($1 ~ $2): "  if (($REPLY>=$1 && $REPLY<=$2));then  echo -e " $3 OKAY ($1 ~ $2 GB)\n"  eval $3=$REPLY; break  else  echo -e " Tamanho D(o|a) $3 inválido\n Minimo:$1\n Máximo:$2"; continue ;  fi  done  }    # TAMANHO do disco virtual.  READ 1 50 DISCO    # TAMANHO da ram.  READ 1 3 MEMORIA    # Quantidade de CPU  READ 1 2 CPU    tput clear    echo -e "\n Os valores definidos são:\n DISCO:$DISCO\n MEMÓRIA:$MEMORIA\n CPU:$CPU\n"    #Criando disco da maquina.    CREATE\_IMG=$NAME.$FORMATO  qemu-img create -f $FORMATO $CREATE\_IMG ${DISCO}G  DISCO\_CRIADO=$CREATE\_IMG    # Comando Virt-install para instalar uma nova maquina virtual.    virt-install --import --name $NAME --memory $MEMORIA --vcpus $CPU --cpu host --disk $DISCO\_CRIADO,format=$FORMATO,bus=virtio --disk $IMG\_DISK,device=cdrom --network $NETWORK\_PARAMS --os-type=$TIPO --os-variant=$VARIANTE --graphics spice,port=$PORTA,listen=$LISTEN --noautoconsole    sleep 0.5  PCM  }    DEBIAN9(){  tput clear    # Definição do virt-install debian 7.    TIPO=$TIPO\_LINUX  FORMATO=qcow2  VARIANTE=  PORTA=  LISTEN=  IMG\_DISK=$DIRETORIO/$DEBIAN9\_IMG  #MACADDRESS=openssl rand -hex 6 | sed 's/\(..\)/\1:/g; s/.$//'  #NETWORK\_PARAMS="bridge=${BRIDGE},model=virtio,mac=${MACADDRESS}"    # Inicio da configuração do usuario da maquina virtual.    read -p "Nome da maquina virtual:" NAME    READ() {  while :;do  read -p " Informe o TAMANHO d(o|a) $3 ($1 ~ $2): "  if (($REPLY>=$1 && $REPLY<=$2));then  echo -e " $3 OKAY ($1 ~ $2 GB)\n"  eval $3=$REPLY; break  else  echo -e " Tamanho D(o|a) $3 inválido\n Minimo:$1\n Máximo:$2"; continue ;  fi  done  }    # TAMANHO do disco virtual.  READ 1 50 DISCO    # TAMANHO da ram.  READ 1 3 MEMORIA    # Quantidade de CPU  READ 1 2 CPU    tput clear    echo -e "\n Os valores definidos são:\n DISCO:$DISCO\n MEMÓRIA:$MEMORIA\n CPU:$CPU\n"    #Criando disco da maquina.    CREATE\_IMG=$NAME.$FORMATO  qemu-img create -f $FORMATO $CREATE\_IMG ${DISCO}G  DISCO\_CRIADO=$CREATE\_IMG    # Comando Virt-install para instalar uma nova maquina virtual.    virt-install --import --name $NAME --memory $MEMORIA --vcpus $CPU --cpu host --disk $DISCO\_CRIADO,format=$FORMATO,bus=virtio --disk $IMG\_DISK,device=cdrom --network $NETWORK\_PARAMS --os-type=$TIPO --os-variant=$VARIANTE --graphics spice,port=$PORTA,listen=$LISTEN --noautoconsole    sleep 0.5  PCM  }    # Configuração das VM winsdows.    VMWIN() {  tput clear  printf "${CE}\e[2;${POS2}f----------------------------------------------"  printf "\e[3;${POS1}f------------------ MENU WINDOWS -------------------"  printf "\e[4;${POS}f---\e[4;${POS3}f---"  printf "\e[5;${POS}f---${CSM} [1] Win XP [2] Win 2007 ${CE}---"  printf "\e[6;${POS}f---${CSM} [3] Win 2008 [4] Win 2010 ${CE}---"  printf "\e[7;${POS}f---${CSM} [5] Win.Server 2008 [6] Win.Server 2012 ${CE}---"  #printf "\e[7;${POS}f---${CSM} [7] Kali [8] Debian 8 ${CE}---"  printf "\e[8;${POS}f---${CSM} [9] Cancelar ${CE}---"  printf "\e[9;${POS}f---\e[9;${POS3}f---"    printf "${CE}\e[10;${POS1}f------------------------------------------"  printf "\e[11;${POS2}f----------------------------------------\n\n"$FC""  read -p " OPÇãO: " -n1 OPCVM    case "$OPCVM" in  1) WINXP;;  2) WIN7;;  3) WIN8;;  4) WIN10;;  5) WINSERVER08;;  6) WINSERVER12;;  9) echo -e "\n\n Cancelando...\n";sleep 0.5; exit ;;  \*) MSGM ; PVM ;;  esac  }    MSGC() { echo -e "$CVA\n\n \" OPÇÃO INVÁLIDA! \" $FC" ; sleep 0.25 ; }    WINXP(){  tput clear    # Definição do virt-install    TIPO=$TIPO\_WINDOWS  FORMATO=qcow2  VARIANTE=wheezy  PORTA=  LISTEN=  IMG\_DISK=$DIRETORIO/$WINXP\_IMG  #MACADDRESS=openssl rand -hex 6 | sed 's/\(..\)/\1:/g; s/.$//'  #NETWORK\_PARAMS="bridge=${BRIDGE},model=virtio,mac=${MACADDRESS}"    # Inicio da configuração do usuario da maquina virtual.    read -p "Nome da maquina virtual:" NAME    READ() {  while :;do  read -p " Informe o TAMANHO d(o|a) $3 ($1 ~ $2): "  if (($REPLY>=$1 && $REPLY<=$2));then  echo -e " $3 OKAY ($1 ~ $2 GB)\n"  eval $3=$REPLY; break  else  echo -e " Tamanho D(o|a) $3 inválido\n Minimo:$1\n Máximo:$2"; continue ;  fi  done  }    # TAMANHO do disco virtual.  READ 1 50 DISCO    # TAMANHO da ram.  READ 1 3 MEMORIA    # Quantidade de CPU  READ 1 2 CPU    tput clear    echo -e "\n Os valores definidos são:\n DISCO:$DISCO\n MEMÓRIA:$MEMORIA\n CPU:$CPU\n"    #Criando disco da maquina.    CREATE\_IMG=$NAME.$FORMATO  qemu-img create -f $FORMATO $CREATE\_IMG ${DISCO}G  DISCO\_CRIADO=$CREATE\_IMG    # Comando Virt-install para instalar uma nova maquina virtual.    virt-install --import --name $NAME --memory $MEMORIA --vcpus $CPU --cpu host --disk $DISCO\_CRIADO,format=$FORMATO,bus=virtio --disk $IMG\_DISK,device=cdrom --network $NETWORK\_PARAMS --os-type=$TIPO --os-variant=$VARIANTE --graphics spice,port=$PORTA,listen=$LISTEN --noautoconsole    sleep 0.5  PCM  }    WIN7(){  tput clear    # Definição do virt-install    TIPO=$TIPO\_WINDOWS  FORMATO=qcow2  VARIANTE=wheezy  PORTA=  LISTEN=  IMG\_DISK=$DIRETORIO/$WIN7\_IMG  #MACADDRESS=openssl rand -hex 6 | sed 's/\(..\)/\1:/g; s/.$//'  #NETWORK\_PARAMS="bridge=${BRIDGE},model=virtio,mac=${MACADDRESS}"    # Inicio da configuração do usuario da maquina virtual.    read -p "Nome da maquina virtual:" NAME    READ() {  while :;do  read -p " Informe o TAMANHO d(o|a) $3 ($1 ~ $2): "  if (($REPLY>=$1 && $REPLY<=$2));then  echo -e " $3 OKAY ($1 ~ $2 GB)\n"  eval $3=$REPLY; break  else  echo -e " Tamanho D(o|a) $3 inválido\n Minimo:$1\n Máximo:$2"; continue ;  fi  done  }    # TAMANHO do disco virtual.  READ 1 50 DISCO    # TAMANHO da ram.  READ 1 3 MEMORIA    # Quantidade de CPU  READ 1 2 CPU    tput clear    echo -e "\n Os valores definidos são:\n DISCO:$DISCO\n MEMÓRIA:$MEMORIA\n CPU:$CPU\n"    #Criando disco da maquina.    CREATE\_IMG=$NAME.$FORMATO  qemu-img create -f $FORMATO $CREATE\_IMG ${DISCO}G  DISCO\_CRIADO=$CREATE\_IMG    # Comando Virt-install para instalar uma nova maquina virtual.    virt-install --import --name $NAME --memory $MEMORIA --vcpus $CPU --cpu host --disk $DISCO\_CRIADO,format=$FORMATO,bus=virtio --disk $IMG\_DISK,device=cdrom --network $NETWORK\_PARAMS --os-type=$TIPO --os-variant=$VARIANTE --graphics spice,port=$PORTA,listen=$LISTEN --noautoconsole    sleep 0.5  PCM  }    WIN8(){  tput clear    # Definição do virt-install    TIPO=$TIPO\_WINDOWS  FORMATO=qcow2  VARIANTE=wheezy  PORTA=  LISTEN=  IMG\_DISK=$DIRETORIO/$WIN8\_IMG  #MACADDRESS=openssl rand -hex 6 | sed 's/\(..\)/\1:/g; s/.$//'  #NETWORK\_PARAMS="bridge=${BRIDGE},model=virtio,mac=${MACADDRESS}"    # Inicio da configuração do usuario da maquina virtual.    read -p "Nome da maquina virtual:" NAME    READ() {  while :;do  read -p " Informe o TAMANHO d(o|a) $3 ($1 ~ $2): "  if (($REPLY>=$1 && $REPLY<=$2));then  echo -e " $3 OKAY ($1 ~ $2 GB)\n"  eval $3=$REPLY; break  else  echo -e " Tamanho D(o|a) $3 inválido\n Minimo:$1\n Máximo:$2"; continue ;  fi  done  }    # TAMANHO do disco virtual.  READ 1 50 DISCO    # TAMANHO da ram.  READ 1 3 MEMORIA    # Quantidade de CPU  READ 1 2 CPU    tput clear    echo -e "\n Os valores definidos são:\n DISCO:$DISCO\n MEMÓRIA:$MEMORIA\n CPU:$CPU\n"    #Criando disco da maquina.    CREATE\_IMG=$NAME.$FORMATO  qemu-img create -f $FORMATO $CREATE\_IMG ${DISCO}G  DISCO\_CRIADO=$CREATE\_IMG    # Comando Virt-install para instalar uma nova maquina virtual.    virt-install --import --name $NAME --memory $MEMORIA --vcpus $CPU --cpu host --disk $DISCO\_CRIADO,format=$FORMATO,bus=virtio --disk $IMG\_DISK,device=cdrom --network $NETWORK\_PARAMS --os-type=$TIPO --os-variant=$VARIANTE --graphics spice,port=$PORTA,listen=$LISTEN --noautoconsole    sleep 0.5  PCM  }    WIN10(){  tput clear    # Definição do virt-install    TIPO=$TIPO\_WINDOWS  FORMATO=qcow2  VARIANTE=wheezy  PORTA=  LISTEN=  IMG\_DISK=$DIRETORIO/$WIN10\_IMG  #MACADDRESS=openssl rand -hex 6 | sed 's/\(..\)/\1:/g; s/.$//'  #NETWORK\_PARAMS="bridge=${BRIDGE},model=virtio,mac=${MACADDRESS}"    # Inicio da configuração do usuario da maquina virtual.    read -p "Nome da maquina virtual:" NAME    READ() {  while :;do  read -p " Informe o TAMANHO d(o|a) $3 ($1 ~ $2): "  if (($REPLY>=$1 && $REPLY<=$2));then  echo -e " $3 OKAY ($1 ~ $2 GB)\n"  eval $3=$REPLY; break  else  echo -e " Tamanho D(o|a) $3 inválido\n Minimo:$1\n Máximo:$2"; continue ;  fi  done  }    # TAMANHO do disco virtual.  READ 1 50 DISCO    # TAMANHO da ram.  READ 1 3 MEMORIA    # Quantidade de CPU  READ 1 2 CPU    tput clear    echo -e "\n Os valores definidos são:\n DISCO:$DISCO\n MEMÓRIA:$MEMORIA\n CPU:$CPU\n"    #Criando disco da maquina.    CREATE\_IMG=$NAME.$FORMATO  qemu-img create -f $FORMATO $CREATE\_IMG ${DISCO}G  DISCO\_CRIADO=$CREATE\_IMG    # Comando Virt-install para instalar uma nova maquina virtual.    virt-install --import --name $NAME --memory $MEMORIA --vcpus $CPU --cpu host --disk $DISCO\_CRIADO,format=$FORMATO,bus=virtio --disk $IMG\_DISK,device=cdrom --network $NETWORK\_PARAMS --os-type=$TIPO --os-variant=$VARIANTE --graphics spice,port=$PORTA,listen=$LISTEN --noautoconsole    sleep 0.5  PCM  }    WINSERVER08(){  tput clear    # Definição do virt-install    TIPO=$TIPO\_WINDOWS  FORMATO=qcow2  VARIANTE=wheezy  PORTA=  LISTEN=  IMG\_DISK=$DIRETORIO/$WINSERVER08\_IMG  #MACADDRESS=openssl rand -hex 6 | sed 's/\(..\)/\1:/g; s/.$//'  #NETWORK\_PARAMS="bridge=${BRIDGE},model=virtio,mac=${MACADDRESS}"    # Inicio da configuração do usuario da maquina virtual.    read -p "Nome da maquina virtual:" NAME    READ() {  while :;do  read -p " Informe o TAMANHO d(o|a) $3 ($1 ~ $2): "  if (($REPLY>=$1 && $REPLY<=$2));then  echo -e " $3 OKAY ($1 ~ $2 GB)\n"  eval $3=$REPLY; break  else  echo -e " Tamanho D(o|a) $3 inválido\n Minimo:$1\n Máximo:$2"; continue ;  fi  done  }    # TAMANHO do disco virtual.  READ 1 50 DISCO    # TAMANHO da ram.  READ 1 3 MEMORIA    # Quantidade de CPU  READ 1 2 CPU    tput clear    echo -e "\n Os valores definidos são:\n DISCO:$DISCO\n MEMÓRIA:$MEMORIA\n CPU:$CPU\n"    #Criando disco da maquina.    CREATE\_IMG=$NAME.$FORMATO  qemu-img create -f $FORMATO $CREATE\_IMG ${DISCO}G  DISCO\_CRIADO=$CREATE\_IMG    # Comando Virt-install para instalar uma nova maquina virtual.    virt-install --import --name $NAME --memory $MEMORIA --vcpus $CPU --cpu host --disk $DISCO\_CRIADO,format=$FORMATO,bus=virtio --disk $IMG\_DISK,device=cdrom --network $NETWORK\_PARAMS --os-type=$TIPO --os-variant=$VARIANTE --graphics spice,port=$PORTA,listen=$LISTEN --noautoconsole    sleep 0.5  PCM  }    WINSERVER12(){  tput clear    # Definição do virt-install    TIPO=$TIPO\_WINDOWS  FORMATO=qcow2  VARIANTE=  PORTA=  LISTEN=  IMG\_DISK=$DIRETORIO/$WINSERVER12\_IMG  #MACADDRESS=openssl rand -hex 6 | sed 's/\(..\)/\1:/g; s/.$//'  #NETWORK\_PARAMS="bridge=${BRIDGE},model=virtio,mac=${MACADDRESS}"    # Inicio da configuração do usuario da maquina virtual.    read -p "Nome da maquina virtual:" NAME    READ() {  while :;do  read -p " Informe o TAMANHO d(o|a) $3 ($1 ~ $2): "  if (($REPLY>=$1 && $REPLY<=$2));then  echo -e " $3 OKAY ($1 ~ $2 GB)\n"  eval $3=$REPLY; break  else  echo -e " Tamanho D(o|a) $3 inválido\n Minimo:$1\n Máximo:$2"; continue ;  fi  done  }    # TAMANHO do disco virtual.  READ 1 50 DISCO    # TAMANHO da ram.  READ 1 3 MEMORIA    # Quantidade de CPU  READ 1 2 CPU    tput clear    echo -e "\n Os valores definidos são:\n DISCO:$DISCO\n MEMÓRIA:$MEMORIA\n CPU:$CPU\n"    #Criando disco da maquina.    CREATE\_IMG=$NAME.$FORMATO  qemu-img create -f $FORMATO $CREATE\_IMG ${DISCO}G  DISCO\_CRIADO=$CREATE\_IMG    # Comando Virt-install para instalar uma nova maquina virtual.    virt-install --import --name $NAME --memory $MEMORIA --vcpus $CPU --cpu host --disk $DISCO\_CRIADO,format=$FORMATO,bus=virtio --disk $IMG\_DISK,device=cdrom --network $NETWORK\_PARAMS --os-type=$TIPO --os-variant=$VARIANTE --graphics spice,port=$PORTA,listen=$LISTEN --noautoconsole    sleep 0.5  PCM  }  PMENU |