

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**KDCT: UMA FERRAMENTA PARA FACILITAR MIGRAÇÕES OFFLINE
DE MÁQUINAS VIRTUAIS DE OUTROS HIPERVISORES PARA O
KVM**

SÉRGIO COELHO DE CARVALHO

**Cajazeiras
2022**

SÉRGIO COELHO DE CARVALHO

**KDCT: UMA FERRAMENTA PARA FACILITAR MIGRAÇÕES OFFLINE DE
MÁQUINAS VIRTUAIS DE OUTROS HIPERVISORES PARA O KVM**

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. PhD. Francisco Daladier Marques Júnior.

Coorientador

Prof. Esp. Janderson Ferreira Dutra.

Cajazeiras

2022

IFPB / Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Suellen Conceição Ribeiro CRB-2218

C331k Carvalho, Sérgio Coelho de

Kdct: uma ferramenta para facilitar migrações offline de máquinas virtuais de outros hipervisores para o kvm / Sérgio Coelho de Carvalho. – Cajazeiras/PB: IFPB, 2022.

54f.:il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba-IFPB, Campus Cajazeiras. Cajazeiras, 2022.

Orientador(a): Prof. PhD. Francisco Daladier Marques Júnior; Coor.: Prof. Esp. Janderson Ferreira Dutra.

1. Informática. 2. Máquina Virtual. 3. KVM. 4. Conversão de Disco.

I. Carvalho, Sérgio Coelho de. II. Título

CDU: 004 C331k

ATA 33/2022 - CADS/UNINFO/DDE/DG/CZ/REITORIA/IFPB

**ATA DE DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO (TCC)
CURSO: ANÁLISE E DESENVOLVIMENTO DE SISTEMAS (ADS)**

Às 17h30 do dia 09 do mês de MAIO do ano de 2022, o(a) aluno(a) **SÉRGIO COELHO DE CARVALHO**, matrícula **201812010039**, apresentou, como parte dos requisitos para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, seu trabalho de conclusão de curso, tendo como título "**KDCT: UMA FERRAMENTA PARA FACILITAR A MIGRAÇÃO OFFLINE DE VMS DE OUTROS HIPERVISORES PARA O KVM**". Constituíram a banca examinadora os professores **Francisco Daladier Marques Júnior** (orientador), **Janderson Ferreira Dutra** (coorientador), **Fábio Abrantes Diniz** (examinador) e **Fábio Gomes de Andrade** (examinador).

Após a apresentação e as observações dos membros da Banca Examinadora, ficou definido que o trabalho foi considerado **APROVADO** com nota **95**, com a condição de que o (a) aluno (a) entregue, no prazo máximo de 30 dias, a versão final do trabalho com as correções sugeridas pelos membros da banca examinadora. Eu, **FÁBIO ABRANTES DINIZ**, Coordenador do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, lavrei a presente ata, que segue assinada digitalmente por mim e pelos membros da banca examinadora.

Cajazeiras, 12 de maio de 2022.

Documento assinado eletronicamente por:

- Francisco Daladier Marques Junior, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 12/05/2022 14:46:35.
- Fabio Abrantes Diniz, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 12/05/2022 15:05:36.
- Janderson Ferreira Dutra, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 12/05/2022 15:07:29.
- Fabio Gomes de Andrade, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 12/05/2022 15:39:25.
- Sergio Coelho de Carvalho, ALUNO (201812010039) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS, em 13/05/2022 15:08:09.

Este documento foi emitido pelo SUAP em 11/05/2022. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 293487
Código de Autenticação: 6091bf4ed8



Dedico este trabalho a Deus e a minha família, pois, sem eles eu não teria conseguido.

AGRADECIMENTOS

Agradeço primeiramente a Deus pela oportunidade da vida. A minha família, por estar sempre me apoiando e me motivando a continuar estudando para eu conseguir conquistar meus sonhos. A minha namorada, que do começo ao fim deste trabalho sempre me motivou e incentivou-me a continuar, dar o meu melhor e nunca desistir. Ao orientador professor PhD. Francisco Daladier Marques Júnior e ao coorientador professor Esp. Janderson Ferreira Dutra, pelas orientações, contribuições e motivações durante a realização deste trabalho. Aos demais professores do curso de Análise e Desenvolvimento de Sistemas do IFPB Campus Cajazeiras, pelos conhecimentos, conselhos e ensinamentos repassados durante todo o período do curso. A toda equipe do IFPB Campus Cajazeiras, que participaram de alguma forma dessa jornada de estudos. Agradeço ainda, a todas as pessoas que contribuíram de alguma maneira com toda a minha trajetória de estudos.

RESUMO

Mesmo com a constante evolução no paradigma da virtualização e com as diversas ferramentas disponíveis no mercado, que servem para gerenciar o hipervisor KVM, ainda existem dificuldades quando se deseja migrar uma máquina virtual, que seja oriunda de outro hipervisor para o KVM. Logo, é necessário verificar qual o tipo de formato de disco virtual esta máquina virtual está utilizando, a fim de saber se é possível migrar uma VM diretamente para o KVM. Caso a VM não utilize um formato de disco virtual nativo do KVM, para que seja importada para ele, o seu disco virtual deve ser convertido para o formato RAW ou QCOW2, pois são os formatos nativos do KVM e possuem uma melhor compatibilidade com o ambiente de virtualização KVM. Além do mais, como existem diversos hipervisores no mercado, os quais utilizam diversos formatos de discos virtuais, pode-se, gerar ainda mais incompatibilidades nos processos de migração offline de VMs desses hipervisores para o KVM. Visando contornar essas situações de incompatibilidade de conversão e exportação de discos virtuais para o KVM, este trabalho propõe o desenvolvimento de uma ferramenta de interface gráfica para o SO Linux, que visa auxiliar o processo de migração offline de máquinas virtuais oriundas de outros hipervisores, para o KVM. A ferramenta atuará na conversão dos discos virtuais oriundos de outros hipervisores para os formatos RAW e QCOW2, usados nativamente no KVM e ainda na exportação desses discos para os ambientes com KVM em que se deseja migrar as VMs. Além disso, a ferramenta funcionará independentemente de softwares de gerenciamento do KVM, bem como no sistema operacional Linux, por ser o ambiente nativo do KVM.

Palavras-chave: Virtualização. Máquinas Virtuais. Hipervisores. KVM. Conversão de Discos.

ABSTRACT

Even with the constant evolution in the virtualization paradigm and the various tools available on the market, which serve to manage Hypervisor KVM, there are still difficulties when you want to migrate a virtual machine that comes from another hypervisor to the KVM. Therefore, it is necessary to check which type of virtual disk format this virtual machine is using in order to know if it is possible to migrate a VM directly to the KVM. If VM does not use a native KVM virtual disk format, so that it is imported to it, your virtual disc should be converted to the RAW or QCOW2 format, as they are the native KVM formats and have better compatibility with the environment of virtualization kvm. Moreover, as there are several hypervisors on the market, which use various virtual disc formats, one can generate even more incompatibilities in the offline Migration processes of these hypervisors to KVM. In order to circumvent these situations of conversion incompatibility and export of virtual discs to KVM, this work proposes the development of a graphical interface tool for Linux, which aims to assist the offline migration process of virtual machines from other hypervisors, to the kvm. The tool will act to convert virtual discs from other hypervisors to RAW and QCOW2 formats, natively used in KVM and the export of these discs to the KVM environments where you want to migrate the VMs. In addition, the tool will work regardless of KVM management software, as well as in the Linux operating system, as it is the native KVM environment.

Keywords: Virtualization. Virtual Machine. Hypervisors. KVM. Disk Conversion.

LISTA DE FIGURAS

Figura 1 – Virtualização total e para-virtualização.	24
Figura 2 – Arquitetura dos hipervisores de tipo-I e de tipo-II.	27
Figura 3 – Arquitetura do <i>LibVirt</i>	31
Figura 4 – Diagrama de casos de uso da ferramenta KDCT.	33
Figura 5 – Arquitetura da ferramenta KDCT.	35
Figura 6 – Tela inicial da ferramenta com todas as dependências instaladas. . .	38
Figura 7 – Tela inicial da ferramenta faltando as dependências.	38
Figura 8 – Log de instalação de dependências.	39
Figura 9 – Log de desinstalação de dependências.	39
Figura 10 – Tela de conversão de discos virtuais com o <i>qemu-img</i>	40
Figura 11 – Exemplo de exportação de disco virtual em andamento.	41
Figura 12 – Exemplo de exportação de disco virtual.	42
Figura 13 – Exemplo de informações sobre o disco virtual selecionado.	43

LISTA DE QUADROS

Quadro 1 – Comparação entre ferramentas relacionadas.	22
Quadro 2 – Formatos de discos virtuais utilizados pelos hipervisores.	26
Quadro 3 – Comparação entre os formatos de disco virtuais RAW e QCOW2.	29
Quadro 4 – Requisitos da ferramenta KDCT.	34
Quadro 5 – UC1 - Instalar dependências.	48
Quadro 6 – UC2 - Escolher funcionalidade.	49
Quadro 7 – UC3 - Selecionar disco virtual.	49
Quadro 8 – UC4 - Selecionar diretório de destino.	50
Quadro 9 – UC5 - Processar conversão de disco.	50
Quadro 10 – UC6 - Preencher credenciais.	51
Quadro 11 – UC7 - Exportar disco virtual	52
Quadro 12 – UC8 - Desinstalar dependências.	53

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CTSS	<i>Compatible Time Sharing System</i>
E/S	Entrada/Saída
ENV	<i>Environment Variables</i>
GUI	<i>Graphical User Interface</i>
IBM	<i>International Business Machines Corporation</i>
KDCT	<i>KVM Disk Converter Tool</i>
KVM	<i>Kernel Virtual Machine</i>
MIT	<i>Massachusetts Institute of Technology</i>
MMV	Monitor de Máquinas Virtuais
OVA	<i>Open Virtual Appliance</i>
OVF	<i>Open Virtualization Format</i>
QCOW2	<i>QEMU Copy On Write Version 2</i>
RF	Requisito Funcional
RNF	Requisito Não Funcional
SCP	<i>Secure Copy</i>
SFTP	<i>Secure File Transfer Protocol</i>
SO	Sistema Operacional
SSH	<i>Secure Shell</i>
TI	Tecnologia da Informação
UC	<i>Use Case</i>
VMM	<i>Virtual Machine Monitor</i>
VMs	<i>Virtual Machines</i>
XaaS	<i>Everything as a Service</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Motivação	15
1.2	Objetivos	17
1.2.1	Objetivo Geral	17
1.2.2	Objetivos Específicos	17
1.3	Metodologia	17
1.4	Organização do Documento	18
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	Trabalhos Relacionados	19
2.2	Virtualização	23
2.3	Discos Virtuais	25
2.4	Hipervisores	26
2.5	Hipervisor KVM	28
2.6	Migração de Máquinas Virtuais	29
2.7	<i>LibVirt</i>	30
3	SOLUÇÃO PROPOSTA	32
3.1	Análise de Sistema	32
3.1.1	<i>Stakeholders</i>	32
3.1.2	Casos de uso	32
3.1.3	Requisitos	33
3.2	Estrutura da ferramenta	34
3.3	Implementação	37
4	CONCLUSÃO	44
4.1	Trabalhos Futuros	45
	REFERÊNCIAS	46

APÊNDICE A – CASOS DE USO	48
APÊNDICE B – REQUISITOS	54

1 INTRODUÇÃO

Em razão da evolução tecnológica, os ambientes virtualizados estão cada vez mais habituais nas universidades, empresas, companhias e organizações, trazendo muitas vantagens e benefícios para a prestação de serviços na Computação em Nuvem, por meio do paradigma de *Everything as a Service* (XaaS), onde tudo é um serviço.

A virtualização torna possível superar diversos problemas encontrados pelas equipes de tecnologia da informação (TI), oferecendo soluções que podem ser adequadas, de acordo com as necessidades dos utilizadores. Além disso, melhora a eficiência e rapidez dos processos, reduzindo também os riscos envolvidos (MACHADO, 2019).

Incluso no gigantesco domínio da virtualização, os hipervisores são *softwares* que permitem simular *hardwares* físicos, para que os sistemas operacionais (SOs) sejam instalados e gerenciados por ele. Existem dois tipos de hipervisores: a) os hipervisores diretamente sobre o *hardware* (também conhecidos como hipervisores de tipo-I ou *bare metal*) e b) os hipervisores que são dependentes do SO (conhecidos como hipervisores de tipo-II) (CACIATO, 2009), os quais são conhecidos também como *hosted* (CLEMENTINO et al., 2021).

Como exemplo de hipervisor de tipo-I, o *Kernel Virtual Machine* (KVM), traz para o Linux um completo ambiente de virtualização baseado em *kernel*, em que as máquinas virtuais são tratadas como processos legítimos do próprio SO Linux no *host* que está executando. Além disso, cada máquina virtual possui o seu próprio *kernel* e suas camadas de isolamento (MALISZEWSKI et al., 2021).

Os hipervisores costumam fornecer o módulo de armazenamento de dados das *Virtual Machines* (VMs) ou máquinas virtuais por meio de arquivos de discos virtuais que ficam armazenados nos discos rígidos da máquina real. Contudo, em consequência dos diversos hipervisores existentes no mercado, existem também diversos formatos de discos virtuais. Dessa forma, dificuldades podem ser encontradas durante o processo de migração *offline* de VMs oriundas de outros hipervisores para o hipervisor KVM, devido à incompatibilidade de discos virtuais. O motivo disso é que o KVM utiliza o formato RAW e *QEMU Copy On Write Version 2* (QCOW2) como formatos de discos padrões, já os outros hipervisores existentes utilizam outros diferentes formatos. Ou seja, para utilizar os formatos de discos virtuais padrões do KVM nas VMs migradas de outros hipervisores, os discos dessas VMs devem ser convertidos manualmente para RAW ou QCOW2.

Levando esse problema em consideração, o presente trabalho consiste em apresentar contribuições para auxiliar e facilitar o processo de migração offline de máquinas virtuais de outros hipervisores para infraestruturas de virtualização com KVM. Promovendo para o SO Linux, uma ferramenta de interface gráfica que atua na conversão e exportação de discos virtuais de máquinas virtuais oriundas de outros hipervisores, para que sejam utilizadas no KVM.

1.1 MOTIVAÇÃO

Apesar da evolução constante da tecnologia e do uso de serviços em ambientes virtualizados, ainda existem dificuldades que envolvem o processo de migração offline de VMs oriundas dos diferentes hipervisores existentes atualmente no mercado, para os ambientes de virtualização baseados no KVM.

Um problema existente na migração *offline* de VMs de outros hipervisores para o KVM é a incompatibilidade de discos virtuais que afeta por exemplo, a migração de uma máquina virtual criada no hipervisor *VirtualBox*, que possui o disco virtual no formato VDI, para o hipervisor KVM. Logo, existe um problema de compatibilidade, pois para tirar proveito de todas as funcionalidades e obter uma melhor compatibilidade com o KVM, é necessário utilizar os formatos de discos virtuais que são prioritários¹ no KVM, o QCOW2 e RAW. Portanto, para isso, é necessário realizar manualmente a conversão do disco virtual de tal VM do formato VDI, para o formato QCOW2 ou RAW.

O KVM utiliza e herda os formatos de discos virtuais do QEMU, suportando, assim, os mesmos formatos. No entanto, mesmo o KVM suportando diversos formatos de discos virtuais, segundo BUJOR; DOBRE (2013), os principais formatos são o RAW e QCOW2. Os formatos de discos RAW e QCOW2 possibilitam no KVM o uso de funcionalidades tais como: criptografia, compressão e até *snapshots*.

Uma situação a ser mencionada é que existem diversas ferramentas de gerenciamento para os ambientes que utilizam o KVM. Um exemplo desse tipo de ferramenta é a *virt-manager*². Outro exemplo é interface gráfica web para gerenciamento de servidores de virtualização *Cockpit*³. Porém, durante o período de elaboração deste trabalho, foi observado que ambas não disponibilizam a funcionalidade de conversão de discos virtuais.

¹ Disk Images - QEMU documentation. Disponível na URL: <<https://www.qemu.org/docs/master/system/images.html>>, acesso em: 05 jan. 2022.

² Manage virtual machines with virt-manager. Disponível na URL: <<https://virt-manager.org/>>, acesso em: 05 jan. 2022.

³ Cockpit Project. Disponível na URL: <<https://cockpit-project.org/>>, acesso em: 05 jan. 2022.

Para solucionar o problema de incompatibilidade de discos virtuais nos ambientes com KVM, ainda existem poucas ferramentas disponíveis para o SO Linux, que suportem a conversão de discos da maioria dos formatos dos hipervisores existentes no mercado para QCOW2 e RAW. Durante a elaboração deste trabalho, as principais ferramentas encontradas foram a *qemu-img*, do pacote *qemu-utils*⁴ e a *virt-v2v*⁵ que pertencem ao conjunto de ferramentas *Libguestfs*⁶. Dentre essas ferramentas, a *virt-v2v* é a única focada em converter discos de outros hipervisores para o KVM. No entanto, as tais ferramentas só funcionam via terminal, no qual as funcionalidades desejadas são executadas por meio de comandos e parâmetros, que podem ser alterados de acordo com o que se deseja realizar em tais ferramentas.

A usabilidade em ferramentas de linha de comandos pode ser mais complexa e mais demorada para alguns usuários, principalmente para os que são menos familiarizados com as ferramentas e com a interface de comandos. Pois, esses usuários podem necessitar ler a documentação das ferramentas, com o intuito de descobrir para que servem os comandos e os seus parâmetros, visando adequá-los de acordo com o tipo de conversão que deseja realizar.

Diante da escassez de ferramentas para o SO Linux que realizam conversões de discos virtuais para o KVM por meio de interfaces gráficas, percebe-se a relevância de desenvolver uma ferramenta de interface gráfica. Além disso, que tal ferramenta seja capaz de converter diferentes formatos de discos virtuais, oriundos de outros hipervisores e que também seja capaz de exportá-los diretamente do computador que está executando a ferramenta, para outros computadores e ambientes de virtualização com KVM.

O desenvolvimento da ferramenta proposta é intencionado para trazer para o SO Linux uma ferramenta que facilite por meio de uma interface gráfica, o processo de migração *offline* de VMs de outros hipervisores para o KVM e ainda para contemplar os usuários que preferem ferramentas de interfaces gráficas. Tal ferramenta deve funcionar independentemente de ferramentas de gerenciamento do KVM e ser compatível com o sistema operacional Linux, pois é o ambiente nativo do KVM.

⁴ Package: qemu-utils. Disponível na URL: <<https://packages.debian.org/pt-br/sid/qemu-utils>>, acesso em: 05 jan. 2022.

⁵ virt-v2v. Disponível na URL: <<https://libguestfs.org/virt-v2v.1.html>>, acesso em: 05 jan. 2022.

⁶ libguestfs, library for accessing and modifying VM disk images. Disponível na URL: <<https://libguestfs.org/>>, acesso em: 05 jan. 2022.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O objetivo geral é facilitar o processo de migração *offline* de máquinas virtuais de outros hipervisores, para ambientes que utilizam a infraestrutura de virtualização KVM, por meio de uma ferramenta de interface gráfica para o sistema operacional Linux.

1.2.2 Objetivos Específicos

São objetivos específicos deste trabalho:

- Investigar maneiras de migração de máquinas virtuais;
- Detectar os diferentes formatos de discos virtuais utilizados pelos principais hipervisores;
- Explorar as principais ferramentas de conversão de discos virtuais existentes;
- Publicação dos resultados obtidos com a ferramenta.

1.3 METODOLOGIA

São previstas durante o desenvolvimento deste trabalho de conclusão de curso, as seguintes atividades listadas:

- **Investigar as maneiras de migração de máquinas virtuais (A1):** Nesta etapa foram realizados aprofundamentos sobre as maneiras disponíveis de migração de máquinas virtuais;
- **Detectar os diferentes formatos de discos virtuais utilizados pelos principais hipervisores (A2):** Nesta etapa foram realizadas coletas de dados sobre os principais formatos de discos virtuais utilizados nos hipervisores mais usados atualmente;
- **Explorar as principais ferramentas de conversão de discos virtuais existentes (A3):** Nesta etapa foram realizados estudos sobre as principais ferramentas existentes, que realizam o processo de conversão de discos virtuais no SO Linux;
- **Realizar o levantamento de requisitos (A4):** Nesta etapa foi realizado o levantamento de requisitos para o desenvolvimento dessa ferramenta;

- **Implementar a ferramenta proposta (A5):** Durante esta etapa foi realizado o desenvolvimento da ferramenta *KVM Disk Converter Tool* (KDCT);
- **Publicação dos resultados obtidos da realização deste trabalho (A6):** Por fim, esta etapa trata da publicação dos resultados obtidos por meio da realização deste trabalho, seja em algum evento, revista ou periódico.

1.4 ORGANIZAÇÃO DO DOCUMENTO

Neste capítulo, estão localizadas as atividades necessárias para que os objetivos do desenvolvimento deste trabalho sejam alcançados, as motivações para a sua realização e a proposta de solução.

- O capítulo 2 apresenta todo o embasamento teórico que envolve o desenvolvimento deste trabalho, possibilitando um melhor entendimento dos conceitos teóricos ao leitor;
- No capítulo 3 é descrita toda a solução proposta por este trabalho, o que envolve etapas do desenvolvimento, como a de análise de projeto, implementação e as tecnologias utilizadas durante o desenvolvimento;
- Já o capítulo 4 se refere às considerações finais obtidas com o desenvolvimento deste trabalho, bem como, apresenta ideias para desenvolvimento de trabalhos futuros;
- Por fim, no apêndice A, localizam-se os casos de uso da ferramenta KDCT.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados conceitos teóricos que são necessários para uma melhor compreensão dos conteúdos, visando extinguir as dificuldades que podem ser encontradas durante o entendimento dos capítulos seguintes e que envolvem o desenvolvimento deste trabalho. Para tanto, serão abordados os seguintes tópicos: trabalhos relacionados, virtualização, discos virtuais, hipervisores, hipervisor KVM, migração de máquinas virtuais e o conjunto de ferramentas *LibVirt*.

2.1 TRABALHOS RELACIONADOS

Durante a realização do presente trabalho, foram encontradas poucas ferramentas para o SO Linux que realizam o processo de conversão de arquivos de discos virtuais. Contudo, as ferramentas encontradas: *qemu-img*, *virt-v2v* e *OVF Tool*, funcionam via linha de comandos. Além disso, quando se trata de ferramentas que sejam focadas migração para o ambiente KVM, a quantidade de ferramentas se reduz para uma ferramenta, que se trata da *virt-v2v*.

Dentro das ferramentas localizadas durante a elaboração deste trabalho, a *qemu-img* é a que mais se encontra documentações criadas por grandes instituições que utilizam soluções baseadas em Linux, como *Red Hat*¹, *OpenStack*² e *International Business Machines Corporation (IBM)*³.

Segundo FENN et al. (2010), *qemu-img*, é uma ferramenta que fornece a funcionalidade de conversão entre os mais populares formatos de discos virtuais. Tal ferramenta pertence ao pacote de ferramentas *qemu-utils*, é de código fonte aberto e funciona via linha de comandos. Para demonstrar o funcionamento das ferramentas relacionadas, exemplos de comandos necessários para realizar conversões de discos com tais ferramentas são exibidos a seguir, com os respectivos parâmetros e explicações.

No exemplo abaixo é demonstrado um comando de exemplo do *qemu-img*,

¹ CHAPTER 10. QEMU-IMG AND QEMU GUEST AGENT. Disponível na URL: <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/virtualization_administration_guide/chap-virtualization_administration_guide-tips_and_tricks>, acesso em: 18 jan. 2022.

² Converting between image formats. Disponível na URL: <<https://docs.openstack.org/image-guide/convert-images.html>>, acesso em: 18 jan. 2022.

³ QEMU image command. Disponível na URL: <<https://www.ibm.com/docs/en/linux-on-systems?topic=commands-qemu-image-command>>, acesso em: 18 jan. 2022.

necessário para converter um disco virtual do formato VDI para QCOW2.

qemu-img convert -p -f vdi -O qcow2 /home/user/Disco.vdi /home/user/-Disco.qcow2

Explicação do comando e seus respectivos parâmetros, para converter um arquivo de disco virtual do formato VDI para QCOW2, utilizando o *qemu-img*.

- **qemu-img** - Refere-se à invocação pelo nome da ferramenta;
- **convert** - Define a funcionalidade que deseja utilizar, que no caso é a de conversão de discos;
- **-p** - É utilizado para mostrar a porcentagem de progresso durante a conversão de discos.;
- **-f** - É usado para indicar qual o formato do disco que irá ser convertido;
- **vdi** - Indica que o arquivo de disco virtual que será convertido é de formato VDI.
- **-O** - Utiliza-se para indicar para qual formato o disco irá ser convertido;
- **qcow2** - Indica que o arquivo de disco virtual que será convertido para o formato QCOW2.
- **/home/user/Disco.vdi** - Indica a localização onde se encontra o arquivo de disco virtual que deseja-se converter;
- **/home/user/Disco.qcow2** - Define a localização onde se deseja salvar o arquivo de disco virtual que será convertido.

A ferramenta *virt-v2v*⁴ permite também converter discos virtuais no SO Linux. O diferencial de tal ferramenta é que ela é focada em converter discos virtuais para o KVM. Essa ferramenta pertence ao conjunto de ferramentas *Libguestfs*, que é de código fonte aberto e também funciona via linha de comandos.

Para converter um disco virtual utilizando o *virt-v2v* é necessário conhecer os parâmetros necessários para realizar o tipo de conversão desejada. No exemplo abaixo é demonstrado o comando para converter um arquivo de disco virtual do formato VDI para o formato QCOW2.

virt-v2v -i disk Disco.vdi -of qcow2 -o local -os /home/user/Discos

⁴ virt-v2v. Disponível na URL: <<https://libguestfs.org/virt-v2v.1.html>>, acesso em: 17 jan. 2022.

Observe a explicação do comando e dos parâmetros necessários para converter um arquivo de disco virtual do formato VDI para QCOW2:

- **virt-v2v** - Refere-se à invocação pelo nome da ferramenta;
- **-i** - Define o tipo de entrada;
- **disk** - Define que o método de entrada é um arquivo de disco virtual;
- **Disco.vdi** - Indica o nome do arquivo de disco virtual que se deseja converter;
- **-of** - Utiliza-se para indicar para qual formato o disco irá ser convertido;
- **qcow2** - Indica que o arquivo de disco virtual que será convertido para o formato QCOW2;
- **-o** - Define o tipo de saída;
- **local** - Define que o método de saída é local, onde o disco convertido será salvo localmente;
- **-os** - É utilizado para definir o diretório de saída;
- **/home/user/Discos** - Define a localização onde se deseja salvar o arquivo de disco virtual que será convertido.

Já a ferramenta *OVF Tool*⁵ desenvolvida pela *VMWare*, a qual dispõe de funcionalidades para realizar empacotamentos e modificações em arquivos de máquinas virtuais. Além disso, possibilita a importação e exportação de VMs utilizando o formato *Open Virtualization Format (OVF)*. Porém, tal ferramenta é focada nos hipervisores e formatos de discos utilizados principalmente pelos produtos da *VMWare*, como o formato de disco virtual VMDK.

No exemplo abaixo é demonstrado o comando necessário para extrair um pacote OVF de um arquivo *Open Virtual Appliance (OVA)* com a ferramenta *OVF Tool*.

```
ovftool /home/user/arquivo.ova /home/user/arquivo.ovf
```

Veja a explicação do comando e dos parâmetros necessários para extrair o pacote OVF de um arquivo OVA:

⁵ OVF Tool User's Guide. Disponível na URL: <<https://developer.vmware.com/docs/11747/ovf-tool-user-s-guide/>>, acesso em: 17 jan. 2022.

- **ovftool** - Refere-se à invocação pelo nome da ferramenta;
- **/home/user/arquivo.ova** - Define o arquivo OVA que irá ser extraído;
- **/home/user/arquivo.ovf** - Define o nome e onde o arquivo OVF irá ser extraído.

A ferramenta proposta é comparada no Quadro 1 com as ferramentas de conversão de discos que foram encontradas para o SO Linux. A comparação leva em consideração as funcionalidades de conversão de discos das ferramentas, em que a comparação foi feita com base em quatro critérios, 1. tipo de usabilidade da ferramenta; 2. os formatos de discos virtuais que a ferramenta aceita como entrada, para que sejam convertidos; 3. os formatos dos discos virtuais que são resultantes das conversões; e 4. os artefatos finais gerados nas conversões.

Quadro 1 – Comparação entre ferramentas relacionadas.

Nome:	Usabilidade:	Formatos de discos - entrada:	Formatos de discos - saída:	Artefato gerado:
KDCT	Interface gráfica.	QCOW2, QCOW RAW, VDI, VHD, VHDX, VMDK, QED.	QCOW2, RAW.	Disco virtual convertido.
<i>qemu-img</i>	Linha de comandos.	QCOW2, QCOW RAW, VDI, VHD, VHDX, VMDK, QED.	QCOW2, QCOW RAW, VDI, VHD, VHDX, VMDK, QED.	Disco virtual convertido.
<i>virt-v2v</i>	Linha de comandos.	QCOW2, QCOW RAW, VDI, VHD, VHDX, VMDK, QED.	QCOW2, RAW.	Disco virtual convertido; Arquivo XML de <i>domain</i> do LibVirt.
<i>OVF Tool</i>	Linha de comandos.	OVF, OVA, VMX, VI, vCloud, ISO, FLP, vApprun.	OVF, OVA, VMX, VI, vCloud, ISO, FLP, vApprun.	Arquivo de configurações das VMs; Disco virtual convertido.

Fonte: Elaborado pelo autor, 2022.

Analisando o Quadro 1, conclui-se que o diferencial da ferramenta proposta comparada com as outras ferramentas relacionadas, é justamente por se tratar de uma ferramenta contendo uma interface gráfica focada em converter discos virtuais, dos formatos: QCOW2, QCOW, RAW, VDI, VHD, VHDX, VMDK e QED, para os formatos que são utilizados nativamente pelo hipervisor KVM: QCOW2 e RAW.

2.2 VIRTUALIZAÇÃO

A virtualização é a principal ferramenta para permitir a prestação de serviços de *software* na atualidade, devido à popularização da computação em nuvem, no qual tudo é um serviço. Virtualização é uma tecnologia que envolve vários conceitos e pode abranger diversas áreas de estudo na ciência da computação, pois é capaz de proporcionar um uso mais eficiente de energia e dos recursos disponíveis nos dispositivos presentes em diferentes ambientes.

Pode-se afirmar que a concepção da virtualização teve início quando o cientista da computação Christopher Strachey fez a publicação de um artigo intitulado *Time Sharing Processing in Large Fast Computers* (CHRISTOPHER, 1959), na Conferência Internacional de Processamento de Informação em 1959, na cidade de Nova York. No artigo publicado, foi demonstrada uma nova maneira de utilização para computadores de grande porte, trazendo como benefício uma melhor utilização do *hardware* para os computadores (VERAS, 2011).

Levando em consideração as contribuições de Christopher, o *Massachusetts Institute of Technology* (MIT) desenvolveu um novo padrão chamado *Compatible Time Sharing System* (CTSS). Baseando-se nesse novo padrão, a IBM começou a utilizar o multi-processamento nos computadores, permitindo assim, que várias unidades de processamento funcionassem como uma só. Nesses computadores, o conceito de memória virtual foi utilizado como uma parte do SO, o que viabilizou a abstração e o mapeamento da memória real em memória virtual, surgindo assim, os primeiros meios de virtualização (VERAS, 2011).

A utilização da virtualização em ambientes da TI pode trazer vantagens para tais ambientes, tais como: a redução de custos que pode se dar devido a redução do uso de equipamentos físicos de *hardware* e a segurança que se dá devido a cada VM ser isolada uma da outra. No entanto, o uso da virtualização pode trazer desvantagens, como por exemplo para o desempenho, que se dá devido ao hipervisor ser mais um software executado na máquina, o que torna o uso do processamento maior (MATTOS, 2008).

Segundo CARISSIMI (2008), a ideia principal da virtualização é poder trocar ou ampliar algum recurso existente por outro, no qual de alguma maneira os recursos que irão ser substitutos imitem o comportamento dos recursos que serão substituídos.

A virtualização proporciona por meio dos hipervisores, que uma máquina física possa executar além do seu SO principal que possui acesso ao *hardware* real da

máquina, outros diversos SOs que são gerenciados pelos hipervisores. Os hipervisores permitem a instalação de SOs distintos em cada instância de máquina virtual (MATTOS, 2008).

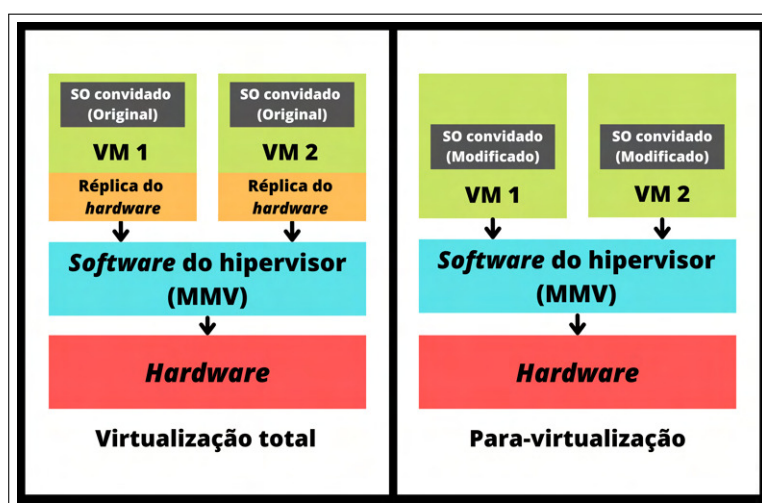
Na virtualização, o sistema operacional que está instalado na máquina física é chamado de *host operating system* (sistema operacional hospedeiro). Já os sistemas operacionais que são instalados nas máquinas virtuais são conhecidos como *guest operating system* (sistemas operacionais convidados). Cada máquina física pode executar um SO hospedeiro por vez, enquanto esse SO hospedeiro pode executar diversos SOs convidados (MATTOS, 2008).

Existem duas técnicas de virtualização que são utilizadas pelos hipervisores, a chamada virtualização completa, também conhecida como virtualização total e a para-virtualização (MATTOS, 2008).

A técnica de virtualização total consiste em executar o sistema operacional em uma réplica virtual do *hardware* físico da máquina hospedeira, desse modo os sistemas operacionais convidados que são executados no hipervisor não precisam sofrer alterações. Já na técnica chamada de para-virtualização, é necessário que o sistema operacional convidado seja modificado, para realizar chamadas ao hipervisor toda vez que uma instrução sensível possa alterar o estado do sistema operacional (MATTOS, 2008).

Na Figura 1 são exibidas as diferenças entre a técnica de virtualização total e a para-virtualização.

Figura 1 – Virtualização total e para-virtualização.



No lado esquerdo da Figura 1 é representada a técnica de virtualização total. Pode-se observar que é utilizada uma réplica virtual do *hardware* físico para executar o SO convidado sem que haja modificações, ou seja, o SO convidado permanece original. Já do lado direito da Figura 1 é demonstrada a técnica de para-virtualização, destaca-se que diferentemente da técnica de virtualização total, tal técnica não utiliza uma réplica virtual do *hardware* físico e além disso, utiliza um SO convidado modificado.

2.3 DISCOS VIRTUAIS

Os arquivos de imagens de discos virtuais podem ser reconhecidos pelo SO convidado da máquina virtual como um disco rígido e o tamanho desse disco rígido reconhecido pela máquina virtual é igual ao tamanho do arquivo de imagem de disco virtual que está armazenado no SO hospedeiro do computador *host*. Tais arquivos de imagens de discos virtuais podem ser armazenados em diferentes formatos existentes, porém, o formato que é considerado mais simples é o RAW, que também é conhecido como “cru”. Existem outros esquemas chamados de *copy-on-write*, que incluem por exemplo os seguintes formatos: QCOW2, VMDK, VDI e VHD (JOSHI et al., 2014).

No trabalho de RAITZ; PÉRICAS (2005), um disco virtual foi utilizado como maneira de armazenamento de dados na criação de uma VM, para implementar a proposta de seu trabalho. Segundo RAITZ; PÉRICAS (2005), o motivo disso foi facilitar a necessidade de realização de backup de dados, cópias da VM e ainda, eliminar a necessidade de particionamento de discos rígidos.

No Quadro 2 são exibidos exemplos de hipervisores que são utilizados no mercado. São eles: KVM⁶, *Oracle VirtualBox*⁷, *Microsoft Hyper-V*⁸ e *VMware vSphere*⁹. Além disso, são exibidos os respectivos formatos de discos virtuais, que são utilizados por padrão e que são recomendados por tais hipervisores.

⁶ Disk Images - QEMU Documentation. Disponível na URL: <<https://qemu-project.gitlab.io/qemu/system/images.html>>, acesso em: 25 jan. 2022.

⁷ Chapter 5. Virtual Storage. Disponível na URL: <<https://www.virtualbox.org/manual/ch05.html>>, acesso em: 25 jan. 2022.

⁸ Hyper-V Virtual Hard Disk Format Overview. Disponível na URL: <[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831446\(v=ws.11\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831446(v=ws.11))>, acesso em: 25 jan. 2022.

⁹ Virtual Disk Format 1.1. Disponível na URL: <<https://www.vmware.com/app/vmdk/?src=vmdk>>, acesso em: 25 jan. 2022.

Quadro 2 – Formatos de discos virtuais utilizados pelos hipervisores.

Hipervisores:	Tipo:	Formatos de discos virtuais:
KVM	Tipo-I	RAW, QCOW2, QED, QCOW, LUKS, VDI, VMDK, VHD, VHDX.
<i>VirtualBox</i>	Tipo-II	VDI, VDH, VMDK, HDD.
<i>Microsoft Hyper-V</i>	Tipo-I	VHDX, VHD.
<i>VMWare vSphere</i>	Tipo-I	VMDK.

Fonte: Elaborado pelo autor, 2022.

As informações contidas no Quadro 2 foram revelantes para o desenvolvimento da proposta, pois, contém dados sobre hipervisores que são muito utilizados no mercado. Além do mais, tais dados possuem o tipo dos hipervisores e os respectivos formatos de discos virtuais utilizados nos hipervisores, o que auxiliou na definição de quais formatos de discos virtuais deve ser compatível na funcionalidade de conversão de discos da ferramenta proposta.

2.4 HIPERVISORES

Também conhecidos como *Virtual Machine Monitor* (VMM) ou Monitor de Máquinas Virtuais (MMV), os hipervisores são um tipo *software*, que realizam o gerenciamento do acesso das máquinas virtuais ao *hardware* da máquina real no qual o hipervisor está instalado para controlar as máquinas virtuais. Sendo assim, os hipervisores são responsáveis por alocar os recursos necessários para cada máquina virtual, fazendo com que cada VM possua seus recursos e funcione de forma independente (GHANNOUM; RODRIGUES, 2018).

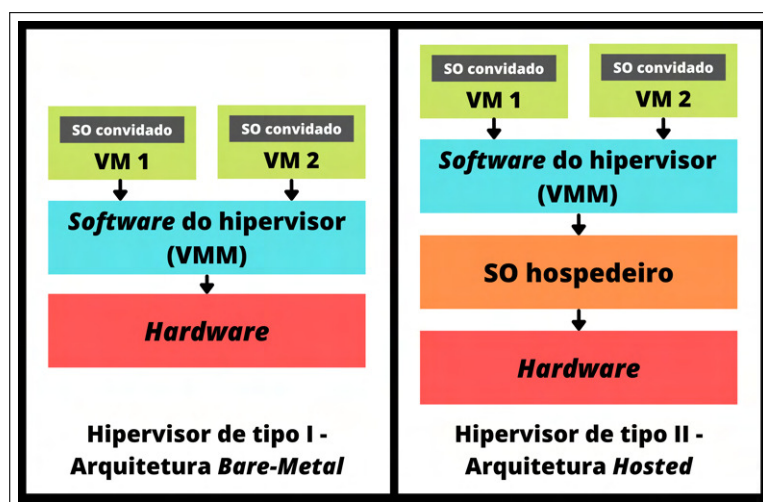
Existem dois tipos de hipervisores: a) hipervisores de tipo-I e b) hipervisores de tipo-II. Os hipervisores de tipo-I possuem um SO incorporado, que permite a instalação de outros SOs convidados, ou seja, eles são os próprios sistemas operacionais. Esses hipervisores são capazes de acessar e gerenciar totalmente o *hardware*. Já os hipervisores de tipo-II são dependentes de SOs, ou seja, são programas que são instalados e executados, rodando em uma camada de *software* acima dos SOs. Logo, precisam de um SO instalado na máquina; portanto, são executados concorrentemente com os demais *softwares* e recursos da máquina (CACIATO, 2009).

Portanto, hipervisores de tipo-II possuem um desempenho inferior comparado aos hipervisores de tipo-I, já que tal abordagem é executada em uma camada acima do *hardware*, não tendo acesso direto ao *hardware* físico da máquina (CACIATO, 2009).

Na Figura 2 as arquiteturas dos hipervisores de tipo-I e de tipo-II podem ser

observadas.

Figura 2 – Arquitetura dos hipervisores de tipo-I e de tipo-II.



Fonte: Elaborado pelo autor, 2022.

Conforme exibido na Figura 2, destaca-se que a principal diferença entre as duas abordagens de hipervisores é que do lado esquerdo da figura o *software* do hipervisor de tipo I, localiza-se acima do *hardware* do dispositivo, tendo acesso direto ao *hardware*. Já do lado direito da figura, o *software* do hipervisor de tipo II fica localizado uma camada acima do *hardware*, ou seja, acima do sistema operacional hospedeiro do dispositivo.

Existem atualmente no mercado diversas opções de hipervisores, sejam eles de tipo-I ou tipo-II. São opções disponíveis de hipervisores do tipo-I de acordo com os autores DINIZ (2019) e BARDEN (2021): KVM¹⁰, *Microsoft Hyper-V*¹¹, *Proxmox*¹², *VMware ESX Server*¹³ e *Xen Server*¹⁴. Já como hipervisores de tipo-II são opções disponíveis segundo LAUREANO; MAZIERO (2008) e BARDEN (2021): QEMU¹⁵,

¹⁰ KVM. Disponível na URL: <https://www.linux-kvm.org/page/Main_Page>, acesso em: 01 fev. 2022.

¹¹ Introdução ao Hyper-V no Windows 10. Disponível na URL: <<https://docs.microsoft.com/pt-br/virtualization/hyper-v-on-windows/about/>>, acesso em: 01 fev. 2022.

¹² Proxmox VE - Virtualization Management Platform. Disponível na URL: <<https://www.proxmox.com/en/proxmox-ve>>, acesso em: 01 fev. 2022.

¹³ VMware esX server. Disponível na URL: <https://www.vmware.com/pdf/esx_datasheet.pdf>, acesso em: 01 fev. 2022.

¹⁴ XenServer 7.0 Standard Edition. Disponível na URL: <<https://www.citrix.com/pt-br/downloads/citrix-hypervisor/product-software/xenserver-70-standard-edition.html>>, acesso em: 01 fev. 2022.

¹⁵ QEMU. Disponível na URL: <<https://www.qemu.org/>>, acesso em: 01 fev. 2022.

*VirtualBox*¹⁶ e o *VMware Workstation*¹⁷. Alguns destes hipervisores citados como exemplos são destaques entre os principais hipervisores existentes (MACHADO; SILVA, 2019).

2.5 HIPERVISOR KVM

Também conhecido como *Kernel Virtual Machine*, o KVM é um hipervisor que proporciona para o SO Linux uma plataforma de virtualização nativa. Por meio do KVM é possível criar e gerenciar máquinas virtuais, no qual cada VM possui suas próprias configurações de individuais. Tornando possível que o computador execute além do próprio Linux como SO convidado, outros diversos SOs como hospedeiros. Conforme CHE et al. (2010), o KVM se beneficia das vantagens do *kernel* padrão do SO Linux e ainda de tecnologias de virtualização assistidas por *hardware*.

O monitor de máquinas virtuais KVM é composto por dois componentes: o módulo *kernel* e o *user-space* (espaço de usuário). O módulo *kernel* é responsável pela virtualização dos recursos de *hardware*. Já o *user-space*, é responsável pela virtualização da Entrada/Saída (E/S), utilizando uma versão modificada do QEMU (emulador e virtualizador de máquinas), para simular a E/S. O módulo *kernel* implementa a virtualização de *hardware* utilizando o comando `/dev/kvm` e o *kill*, em que os SOs convidados podem possuir seus próprios espaços de endereços, que são alocados por meio do escalonador do próprio SO Linux (CHE et al., 2010).

No hipervisor KVM é possível utilizar diversos formatos de discos virtuais, porém segundo BUJOR; DOBRE (2013) os formatos principais são: RAW e QCOW2. Além disso tais formatos são utilizados nativamente no emulador QEMU¹⁸. Os formatos de discos virtuais RAW e QCOW2 são comparados no Quadro 3.

¹⁶ Oracle VM VirtualBox. Disponível na URL: <<https://www.virtualbox.org/>>, acesso em: 01 fev. 2022.

¹⁷ VMware Workstation Pro. Disponível na URL: <<https://www.vmware.com/br/products/workstation-pro.html>>, acesso em: 01 fev. 2022.

¹⁸ Disk Images - QEMU documentation. Disponível na URL: <<https://www.qemu.org/docs/master/system/images.html>>, acesso em: 01 fev. 2022.

Quadro 3 – Comparação entre os formatos de disco virtuais RAW e QCOW2.

Formato:	Tipos de alocação:	Características:
RAW	Pré-alocação realizada durante a criação do disco.	Estrutura simples, mais fácil de ser exportado e utilizado em outros hipervisores.
QCOW2	Alocação dinâmica de acordo com a necessidade de mais espaço (padrão). Também pode ser criado utilizando a pré-alocação durante a criação do disco.	Criptografia, compressão, <i>snapshots</i> .

Fonte: Elaborado pelo autor, 2022.

A comparação feita no Quadro 3 é revelante para o entendimento dos benefícios existentes ao se utilizar os formatos de disco RAW e QCOW2 nativos do KVM. Além de que, ressalta as principais diferenças entre ambos os formatos de discos. Em tal comparação, o formato QCOW2 destaca vantagem, pois, pode economizar espaço ao utilizar a alocação dinâmica e ainda há a possibilidade de utilização de criptografia, compressão e *snapshots*.

O formato de disco virtual RAW costuma oferecer um melhor desempenho, tanto na leitura, como na gravação de dados, pois nesse formato, os blocos de dados são sempre alocados. Além disso, os discos de formato RAW são alocados completamente na sua criação (BUJOR; DOBRE, 2013).

Apesar de haver a possibilidade de tornar possível a alocação completa na criação do disco, por padrão, nos discos de formato QCOW2 a alocação é de acordo com a necessidade de mais espaço. Além disso, o formato QCOW2 oferece algumas funcionalidades, como criptografia e compressão (BUJOR; DOBRE, 2013).

2.6 MIGRAÇÃO DE MÁQUINAS VIRTUAIS

O processo de migração de máquinas virtuais consiste em migrar as informações de uma máquina virtual, seja de um hipervisor para outro que esteja no mesmo computador ou de um hipervisor em um computador local para outro hipervisor em um computador remoto.

Segundo PATEL et al. (2014), existem duas técnicas de migração de máquinas virtuais: a) *hot* (quente), também chamada de *live* (migração ao vivo) e b) *cold* (fria), que é conhecida também como migração *offline* (não ao vivo).

A técnica de migração de VMs ao vivo corresponde ao processo de migração que é executado sem desligar a VM, além disso, o estado atual da VM é migrado, logo não é perdido. A primeira etapa de tal processo é a suspensão da VM que será migrada, depois o estado atual da VM que inclui os dados em memória é transferido para o computador de destino e finalmente, após isso, a VM é retomada da suspensão com o seu último estado restaurado no computador de destino. Já durante a execução da técnica de migração não ao vivo, a VM é interrompida e o seu estado atual é perdido (PATEL et al., 2014).

Na elaboração deste trabalho, a técnica de migração *offline* (não ao vivo) de VMs foi utilizada para realizar a migração de VMs de outros hipervisores para o KVM. Primeiramente a VM deve ser interrompida e logo depois, caso a VM esteja instalada em outro hipervisor e não esteja utilizando o formato de disco virtual RAW ou QCOW2, esse disco deve ser convertido para RAW ou QCOW2, para que tenha uma melhor compatibilidade com o KVM. Logo após ser convertido, o disco virtual deve ser exportado pela ferramenta para o dispositivo com o hipervisor KVM de destino. Após isso, finalmente a VM pode ser recriada no novo hipervisor, com o disco que foi convertido e exportado.

2.7 LIBVIRT

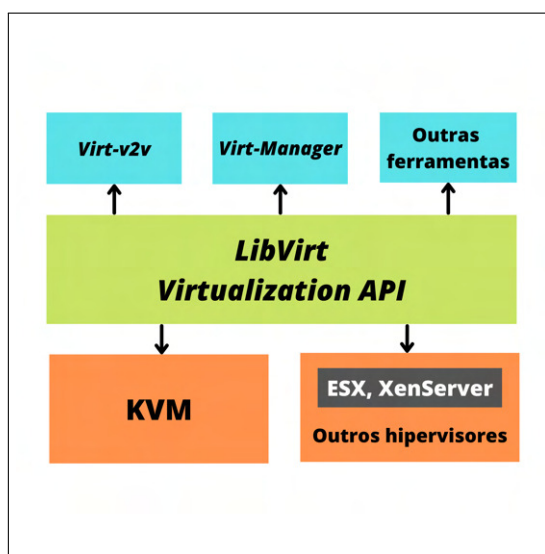
*LibVirt*¹⁹ fornece um conjunto de ferramentas de gerenciamento *open source*, para que sejam utilizadas em ambientes de virtualização. De acordo com BOLTE et al. (2010), o *LibVirt* é uma biblioteca desenvolvida na linguagem de programação C, que funciona como uma camada de abstração de diversas *Application Programming Interfaces* (APIs) de gerenciamento, que funciona para vários hipervisores.

O *LibVirt* trabalha implementando um *driver* remoto do lado do cliente, já do lado do servidor existe um *daemon*, que é utilizado para mapear as solicitações do cliente no lado do servidor. Esse *daemon* se chama *libvirtd*, ele recebe as solicitações encapsuladas do cliente e realiza uma chamada para o *driver* específico que se encaixe na solicitação (BOLTE et al., 2010). Atualmente os seus *drivers*²⁰ são compatíveis com diversos hipervisores, como: KVM, *VirtualBox*, *VMware ESX*, *VMware Workstation/Player* e *Xen Server*.

A Figura 3 ilustra a comunicação da API do *LibVirt*, que trabalha como um *middleware* entre os hipervisores e as ferramentas que utilizam o *libvirt*.

¹⁹ libvirt: The virtualization API. Disponível na URL: <<https://libvirt.org/>>, acesso em: 26 fev. 2022.

²⁰ libvirt: Internal drivers. Disponível na URL: <<https://libvirt.org/drivers.html>> acesso em: 26 fev. 2022.

Figura 3 – Arquitetura do *LibVirt*

Fonte: Elaborado pelo autor, 2022.

Observa-se na Figura 3, ferramentas que por meio do *LibVirt*, se comunicam com os hipervisores. No exemplo da Figura 3, são exibidas como exemplo, as ferramentas: *virt-v2v* e a *virt-manager* se comunicando por meio do *LibVirt* com os hipervisores *KVM*, *VMWare ESX*, *Xen Server* e demais hipervisores suportados. Além dessas ferramentas exibidas na Figura 3, existem outras diversas ferramentas disponíveis²¹ que utilizam o *LibVirt*.

²¹ libvirt: Applications using libvirt. Disponível na URL: <<https://libvirt.org/apps.html>> acesso em: 26 fev. 2022.

3 SOLUÇÃO PROPOSTA

Neste capítulo a solução proposta é apresentada. São expostas todas as fases durante a análise de sistema e desenvolvimento da solução, detalhando também as arquiteturas, métodos e as tecnologias que foram utilizadas durante o processo de desenvolvimento da ferramenta.

3.1 ANÁLISE DE SISTEMA

Nesta seção será apresentada a etapa de análise de sistema, que envolve *stakeholders*, casos de usos, diagramas e requisitos de *software*.

3.1.1 Stakeholders

A definição da palavra *stakeholder*, como a própria tradução retrata, significa parte interessada. Em um *software* as partes interessadas podem ser pessoas ou papéis que são afetados de algum meio por este *software* (SOMMERVILLE, 2011).

Neste projeto, os *stakeholders* são as pessoas interessadas em utilizar a ferramenta, ou seja, pessoas que desejam realizar a migração *offline* de VMs oriundas de outros hipervisores, para que sejam utilizadas em ambientes com um hipervisor KVM.

3.1.2 Casos de uso

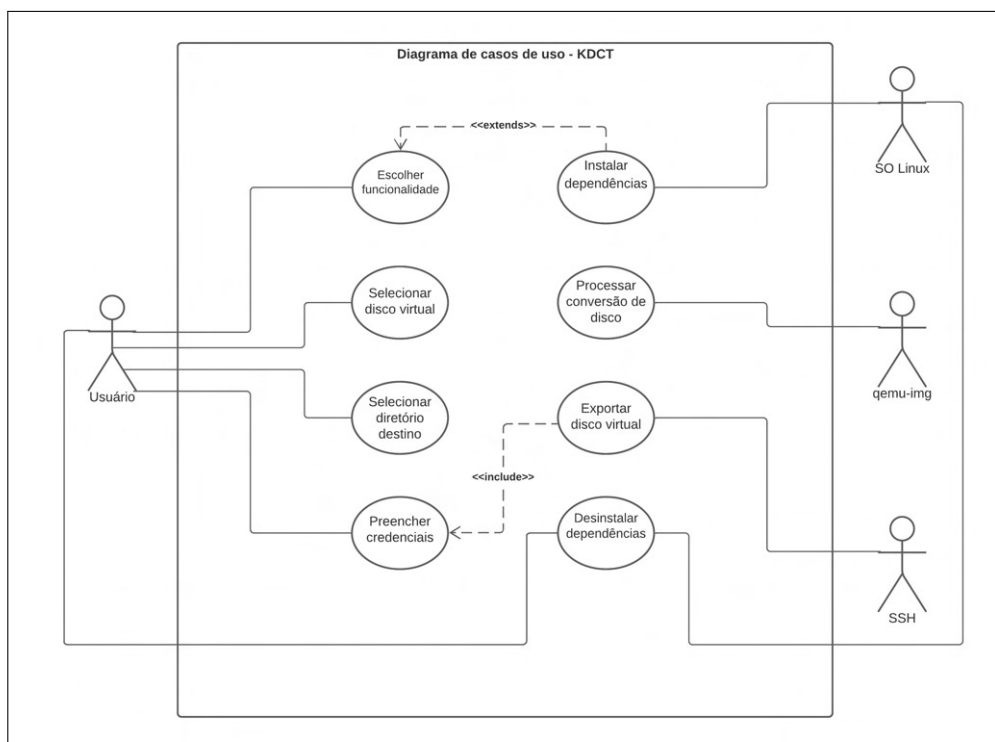
Os casos de usos são utilizados para descrever e identificar interações entre o sistema, seus usuários e ainda outros sistemas que também podem estar envolvidos nessas interações. Em um *Use Case* (UC) ou caso de uso, o nome do ator e da interação são identificados, mas eles também podem ser descritos de forma textual (SOMMERVILLE, 2011).

Existe ainda o diagrama de casos de uso, que permite uma representação gráfica do conjunto de casos de uso do sistema, contendo assim os possíveis atores do sistema e as possíveis interações que podem acontecer entre eles. No diagrama de casos de uso, os atores são representados como “palitos” em formatos de pessoas, já as interações são representadas como elipses e por fim, as ligações entre os atores e as interações são feitas utilizando linhas (SOMMERVILLE, 2011).

Na Figura 4 apresenta-se o diagrama de casos de uso da ferramenta KDCT,

esse diagrama foi desenvolvido utilizando a plataforma de desenvolvimento de gráficos e diagramas *LucidChart*¹, que é baseada na web e pode ser utilizada de maneira gratuita.

Figura 4 – Diagrama de casos de uso da ferramenta KDCT.



Fonte: Elaborado pelo autor, 2022.

Na Figura 4 são representadas as ligações entre os atores do sistema e as possíveis interações deles na ferramenta. Os atores dos casos de usos desta proposta são: usuário, SO Linux, *qemu-img* e SSH. Já as possíveis interações são: Escolher funcionalidade, instalar dependências, selecionar disco virtual, processar conversão de disco, selecionar diretório destino, exportar disco virtual, preencher credenciais e desinstalar dependências. Os casos de usos contidos na Figura 4 foram descritos de forma narrativa e são detalhados no apêndice A deste documento.

3.1.3 Requisitos

Requisitos de um sistema são utilizados para descrever suas funcionalidades, suas restrições e seus serviços, ou seja, o que o sistema deve e o que não deve fazer.

¹ Where seeing becomes doing. Disponível na URL: <<https://www.lucidchart.com>>, acesso em: 05 mar. 2022.

Eles também retratam quais são as necessidades dos usuários sobre os sistemas (SOMMERVILLE, 2011).

Os requisitos podem ser classificados como: a) Requisitos Funcionais (RF) e b) Requisitos Não Funcionais (RNF).

Os requisitos funcionais se referem a como o sistema deve se comportar a determinadas ações, quais funcionalidades o sistema deve possuir e o que ele deve realizar. Já os requisitos não funcionais dizem respeito às restrições que podem ser atribuídas ao sistema, às suas funcionalidades e serviços (SOMMERVILLE, 2011).

No Quadro 4 são exibidos os requisitos que foram definidos durante o processo de análise de requisitos para o desenvolvimento da ferramenta KDCT.

Quadro 4 – Requisitos da ferramenta KDCT.

Identificação:	Título:
RF_01	Instalar dependências.
RF_02	Compatibilidade de formatos de discos virtuais.
RF_03	Exibir informações do disco virtual.
RF_04	Converter discos virtuais para o KVM.
RF_05	Exportar discos virtuais.
RF_05	Desinstalar dependências.
RNF_01	Compatibilidade.
RNF_02	Facilidade de uso.
RNF_03	Padrões.
RNF_04	Implementação.

Fonte: Elaborado pelo autor, 2022.

Como requisitos funcionais observa-se os seguintes requisitos: Instalar dependências, compatibilidade de formatos de discos virtuais, exibir informações do disco virtual, converter discos virtuais para o KVM, exportar discos virtuais e desinstalar dependências. Já como requisitos não funcionais os seguintes: Compatibilidade, facilidade de uso, padrões e implementação. O detalhamento desses requisitos, incluindo suas descrições e prioridades encontram-se no apêndice B desse documento.

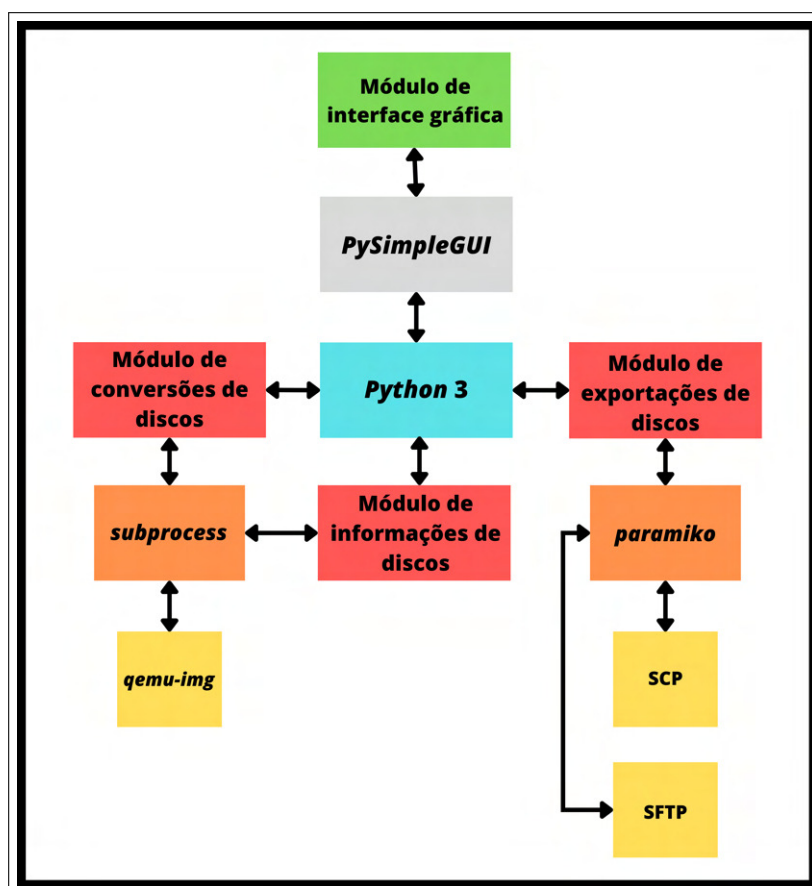
3.2 ESTRUTURA DA FERRAMENTA

O projeto arquitetural da ferramenta proposta é baseada no padrão de camadas. Muito utilizado nesse tipo de ferramenta em que os módulos se comunicam entre si e que utiliza-se a mesma linguagem no desenvolvimento.

A implementação do módulo de interface gráfica, do módulo de conversão de discos, do módulo de informações de discos e do módulo de exportação de discos da ferramenta foi feita utilizando a linguagem de programação *open source Python 3*².

Na Figura 5 a arquitetura geral da ferramenta KDCT é apresentada.

Figura 5 – Arquitetura da ferramenta KDCT.



Fonte: Elaborado pelo autor, 2022.

O módulo de interface gráfica da ferramenta KDCT foi implementado utilizando o *PySimpleGUI*³, uma biblioteca do *Python* que possibilita a criação de *Graphical User Interfaces* (GUIs) ou interfaces gráficas de uma maneira mais simples e facilitada, na qual com poucas linhas de código o desenvolvedor já consegue criar sua primeira interface gráfica funcional.

² Welcome to Python.org. Disponível na URL: <<https://www.python.org/>>, acesso em: 08 mar. 2022.

³ Python GUIs for Humans. Disponível na URL: <<http://www.pysimplegui.org/>>, acesso em: 15 mar. 2022.

A implementação do módulo de conversão de discos virtuais foi realizada por meio de uma integração da ferramenta KDCT com a ferramenta *qemu-img*. Tal implementação foi realizada dessa forma visando uma melhor integração da ferramenta KDCT com o ambiente de virtualização KVM no SO Linux, tirando proveito da ferramenta *qemu-img*, que vem presente no pacote *qemu-utils*⁴ que é instalado no computador no momento em que o pacote *qemu-kvm*⁵ que também é instalado durante a instalação do KVM. Ou seja, a ferramenta *qemu-img* está presente nos ambientes de virtualização com KVM no SO Linux.

Por utilizar a ferramenta *qemu-img* para processar as conversões de discos virtuais, a ferramenta KDCT possibilita converter os mesmos formatos de discos virtuais que a ferramenta *qemu-img*, que são: QCOW2, QCOW, RAW, VDI, VHD, VHDX, VMDK e QED. Esses formatos de discos podem ser convertidos para os formatos QCOW2 e RAW que são utilizados nativamente pelo KVM.

O módulo de conversão de discos virtuais funciona da seguinte maneira: As solicitações são realizadas por meio da interface gráfica da ferramenta KDCT e como a ferramenta *qemu-img* é do tipo que é executada via terminal, foi necessário utilizar o módulo *subprocess* da linguagem de programação *Python 3* para realizar toda a comunicação com a ferramenta *qemu-img*.

Sendo assim, após as informações como: formato de disco, diretório e formato em que o usuário deseja converter o disco serem preenchidas na interface gráfica da ferramenta KDCT, ao clicar no botão para converter o disco virtual, tais informações são repassadas para o módulo de conversão de discos virtuais da ferramenta KDCT, que monta automaticamente o comando chamado de *qemu-img convert*, necessário para realizar a conversão desejada na ferramenta *qemu-img*. Depois disso, o *subprocess* da linguagem *Python 3* é utilizado para criar um novo processo no SO da máquina para se comunicar com a ferramenta *qemu-img* e com isso, o comando de conversão gerado pela ferramenta KDCT é executado na ferramenta *qemu-img*, que por fim processa a conversão do disco que foi selecionado na ferramenta KDCT. Por fim, o usuário recebe informações com o status atual da conversão do disco na interface gráfica da ferramenta KDCT.

O módulo de informações sobre os arquivos de discos virtuais foi implementado utilizando o mesmo tipo de comunicação com a ferramenta *qemu-img* que é utilizada no módulo de conversão de discos virtuais. Porém, nesse módulo foi utilizado o comando

⁴ Package: qemu-utils. Disponível na URL: <<https://packages.debian.org/pt-br/sid/qemu-utils>>, acesso em: 15 mar. 2022.

⁵ Package: qemu-kvm. Disponível na URL: <<https://packages.debian.org/pt-br/sid/qemu-kvm>>, acesso em: 15 mar. 2022.

qemu-img info para obter as informações dos arquivos de discos virtuais e repassar para a interface gráfica da ferramenta KDCT.

Já na implementação do módulo de exportação de discos virtuais, foi utilizada a biblioteca para a linguagem *Python - Paramiko*⁶. Tal biblioteca possibilita implementar métodos de transferências de arquivos que utilizam protocolos do *Secure Shell* (SSH).

Para realizar a transferências dos arquivos de forma segura, foram utilizados dois protocolos que possibilitam isso, o *Secure File Transfer Protocol* (SFTP) e o *Secure Copy* (SCP). A implementação do protocolo SFTP foi realizada utilizando somente o *Paramiko*, porém na implementação do protocolo SCP foi necessária a utilização de outra biblioteca, chamada *scp*⁷, que em conjunto com o *Paramiko* possibilitou a implementação do protocolo.

Visando proteger a senha que deve ser informada pelo usuário para transferir os arquivos, foi utilizada a biblioteca *cryptocode*⁸, que criptografa e descriptografa *strings* de maneira simples e segura. O segredo que é utilizado para criptografar as *strings* deve ser inserido em um arquivo de *Environment Variables* (ENV), na raiz do projeto. A biblioteca *python-dotenv*⁹ foi utilizada para prover o funcionamento do arquivo *.env*.

3.3 IMPLEMENTAÇÃO

Nesta seção, é demonstrada a implementação das funcionalidades da ferramenta KDCT. Na Figura 6 é apresentada a tela inicial da ferramenta, na qual é possível acessar as funcionalidades de conversão e exportação de discos virtuais, também é possível instalar as dependências dessas funcionalidades.

⁶ Welcome to Paramiko. Disponível na URL: <<https://www.paramiko.org/>>, acesso em: 08 mar. 2022.

⁷ scp PyPI. Disponível na URL: <<https://pypi.org/project/scp/>>, acesso em: 15 mar. 2022.

⁸ cryptocode PyPI. Disponível na URL: <<https://pypi.org/project/cryptocode/>>, acesso em: 15 mar. 2022.

⁹ python-dotenv PyPI. Disponível na URL: <<https://pypi.org/project/python-dotenv/>>, acesso em: 15 mar. 2022.

Figura 6 – Tela inicial da ferramenta com todas as dependências instaladas.



Fonte: Elaborado pelo autor, 2022.

Caso as dependências para certa funcionalidade não estejam instaladas, o botão de acesso a funcionalidade ficará esmaecido, ou seja, desativado e o usuário precisará instalar as dependências e reiniciar a ferramenta para que o acesso a todas as funcionalidades seja liberado, isso é demonstrado na Figura 7.

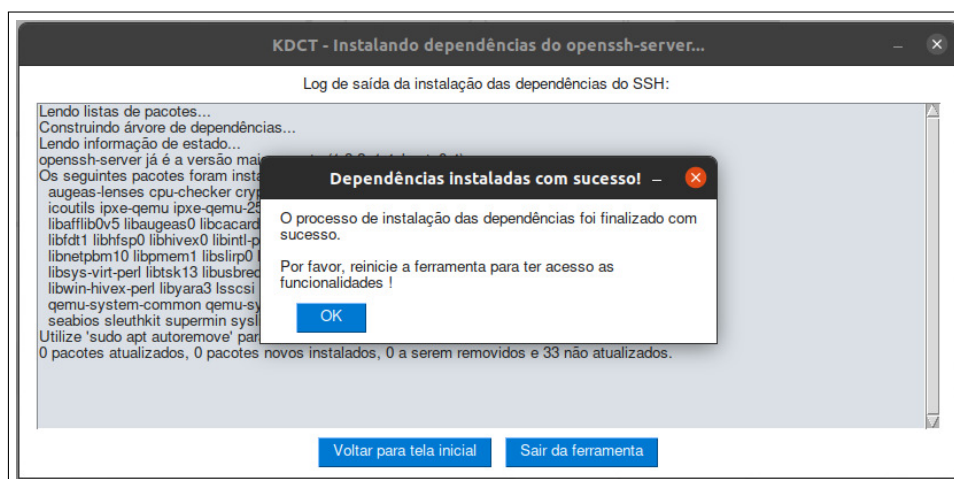
Figura 7 – Tela inicial da ferramenta faltando as dependências.



Fonte: Elaborado pelo autor, 2022.

O log do processo de instalação das dependências é exibido ao usuário durante a instalação, na qual o usuário poderá conferir os pacotes que foram instalados no sistema operacional do computador. O log de instalação é exibido na Figura 8.

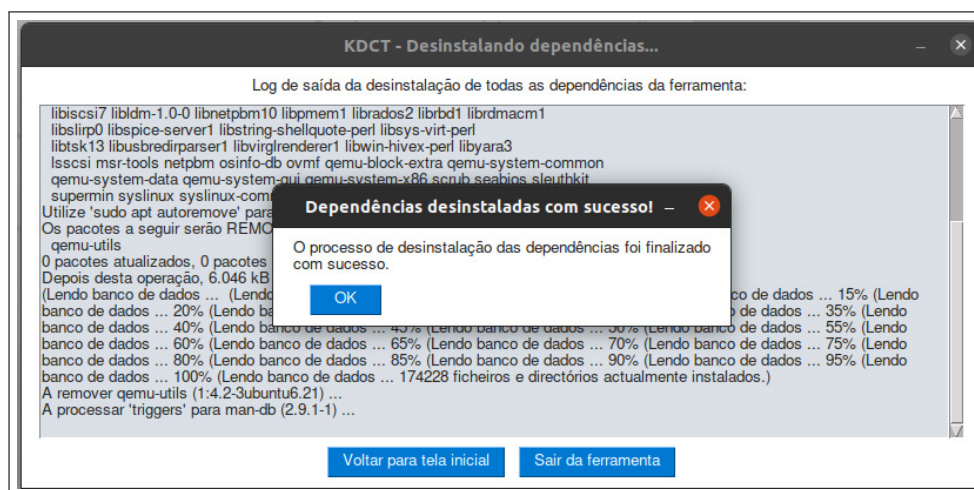
Figura 8 – Log de instalação de dependências.



Fonte: Elaborado pelo autor, 2022.

Da maneira que é possível instalar as dependências da ferramenta, também existe a possibilidade de desinstalá-las, caso o usuário deseje. A Figura 9 representa o processo de desinstalação das dependências.

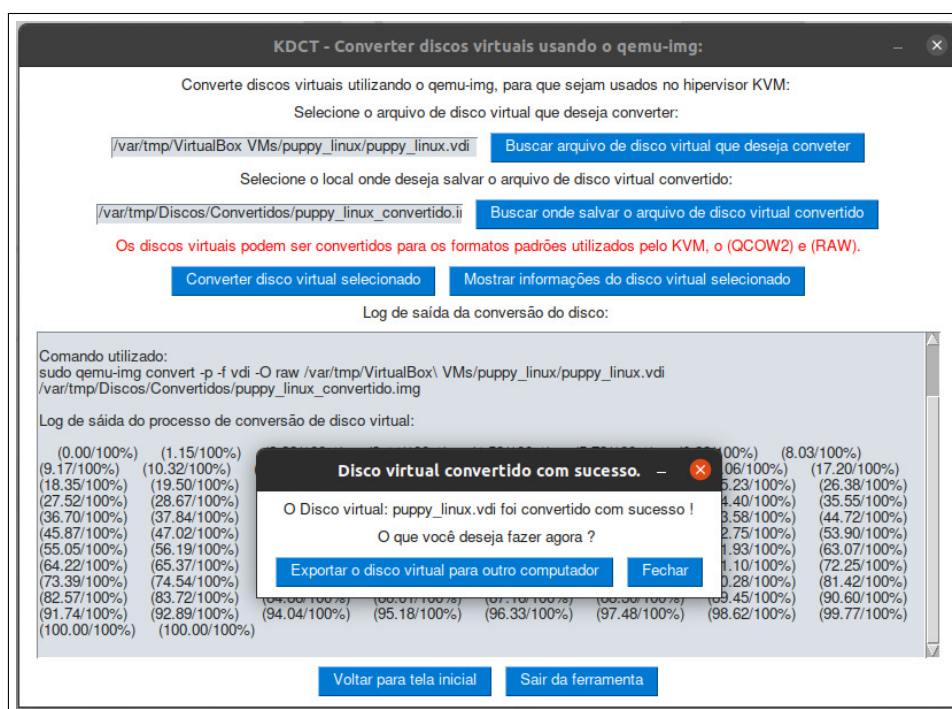
Figura 9 – Log de desinstalação de dependências.



Fonte: Elaborado pelo autor, 2022.

Com a funcionalidade de conversão de discos o usuário pode realizar as conversões dos diferentes formatos de discos virtuais, para os formatos utilizados nativamente pelo KVM, o RAW e QCOW2. Essas conversões são processadas por meio ferramenta externa *qemu-img*. A Figura 10 exibe a tela dessa funcionalidade.

Figura 10 – Tela de conversão de discos virtuais com o *qemu-img*.



Fonte: Elaborado pelo autor, 2022.

Na Figura 10, o usuário pode selecionar o arquivo de disco virtual que deseja converter, o local em que deseja salvar o disco que será convertido e para qual formato deseja converter o arquivo disco, o usuário pode escolher o formato RAW ou QCOW2. Além disso, o usuário consegue ver qual foi o comando utilizado para realizar a conversão no *qemu-img* e ainda obter informações sobre o disco virtual selecionado.

A Figura 11 exibe a funcionalidade de exportação de discos virtuais, na qual o usuário consegue selecionar o arquivo que deseja exportar e preencher o endereço IP, porta, nome e senha de usuário da máquina para qual deseja exportar o arquivo de disco virtual. Na tela de exportação de discos virtuais também é possível consultar

informações sobre o disco que o usuário selecionou. Além disso, o progresso da exportação é exibido durante o processo, no qual são mostrados a porcentagem e a quantidade de bytes que foram transferidos do total que o arquivo contém.

Figura 11 – Exemplo de exportação de disco virtual em andamento.

KDCT - Exportar arquivos de discos virtuais usando o SSH:

Exporta arquivos de discos virtuais que foram convertidos ou não, para que sejam usados em outros computadores com o hipervisor KVM:

Selecione o arquivo de disco virtual que deseja exportar:

Preencha as credenciais do computador de destino em que deseja exportar o disco virtual:

Digite o endereço IP do computador:

Digite a porta do computador:

Digite o nome do usuário do computador:

Digite a senha do usuário do computador:

Selecione o tipo de protocolo em que deseja usar para exportar o disco virtual:

SCP SFTP

Log de saída da exportação do disco virtual:

```

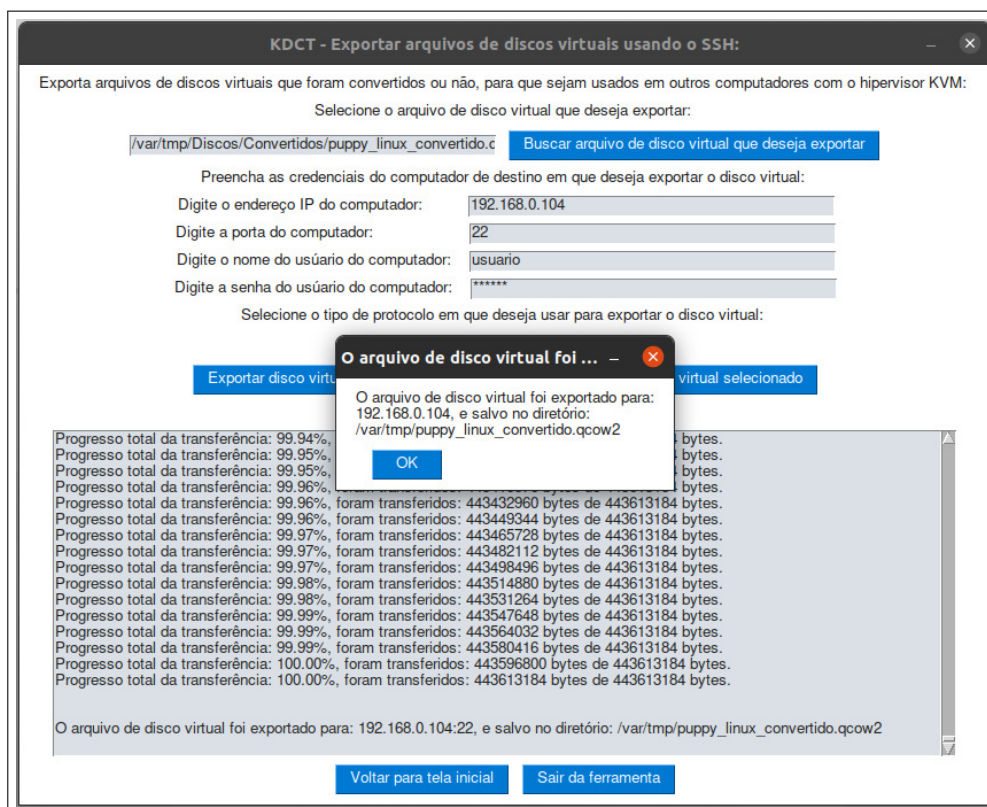
Progresso total da transferência: 4.34%, foram transferidos: 19251200 bytes de 443613184 bytes.
Progresso total da transferência: 4.34%, foram transferidos: 19267584 bytes de 443613184 bytes.
Progresso total da transferência: 4.35%, foram transferidos: 19283968 bytes de 443613184 bytes.
Progresso total da transferência: 4.35%, foram transferidos: 19300352 bytes de 443613184 bytes.
Progresso total da transferência: 4.35%, foram transferidos: 19316736 bytes de 443613184 bytes.
Progresso total da transferência: 4.36%, foram transferidos: 19333120 bytes de 443613184 bytes.
Progresso total da transferência: 4.36%, foram transferidos: 19349504 bytes de 443613184 bytes.
Progresso total da transferência: 4.37%, foram transferidos: 19365888 bytes de 443613184 bytes.
Progresso total da transferência: 4.37%, foram transferidos: 19382272 bytes de 443613184 bytes.
Progresso total da transferência: 4.37%, foram transferidos: 19398656 bytes de 443613184 bytes.
Progresso total da transferência: 4.38%, foram transferidos: 19415040 bytes de 443613184 bytes.
Progresso total da transferência: 4.38%, foram transferidos: 19431424 bytes de 443613184 bytes.
Progresso total da transferência: 4.38%, foram transferidos: 19447808 bytes de 443613184 bytes.
Progresso total da transferência: 4.39%, foram transferidos: 19464192 bytes de 443613184 bytes.
Progresso total da transferência: 4.39%, foram transferidos: 19480576 bytes de 443613184 bytes.
Progresso total da transferência: 4.40%, foram transferidos: 19496960 bytes de 443613184 bytes.
Progresso total da transferência: 4.40%, foram transferidos: 19496960 bytes de 443613184 bytes.
Progresso total da transferência: 4.40%, foram transferidos: 19513344 bytes de 443613184 bytes.
Progresso total da transferência: 4.40%, foram transferidos: 19529728 bytes de 443613184 bytes.
Progresso total da transferência: 4.41%, foram transferidos: 19546112 bytes de 443613184 bytes.

```

Fonte: Elaborado pelo autor, 2022.

Se a exportação tiver êxito, o *popup* mostrado na Figura 12 é exibido ao usuário, informando que exportação ocorreu com sucesso e o diretório em que o disco foi salvo no outro dispositivo. Além disso, caso ocorra algum erro durante ou antes de iniciar a exportação, *popups* de erros serão exibidos informando que tipo de problema ocorreu durante o processo.

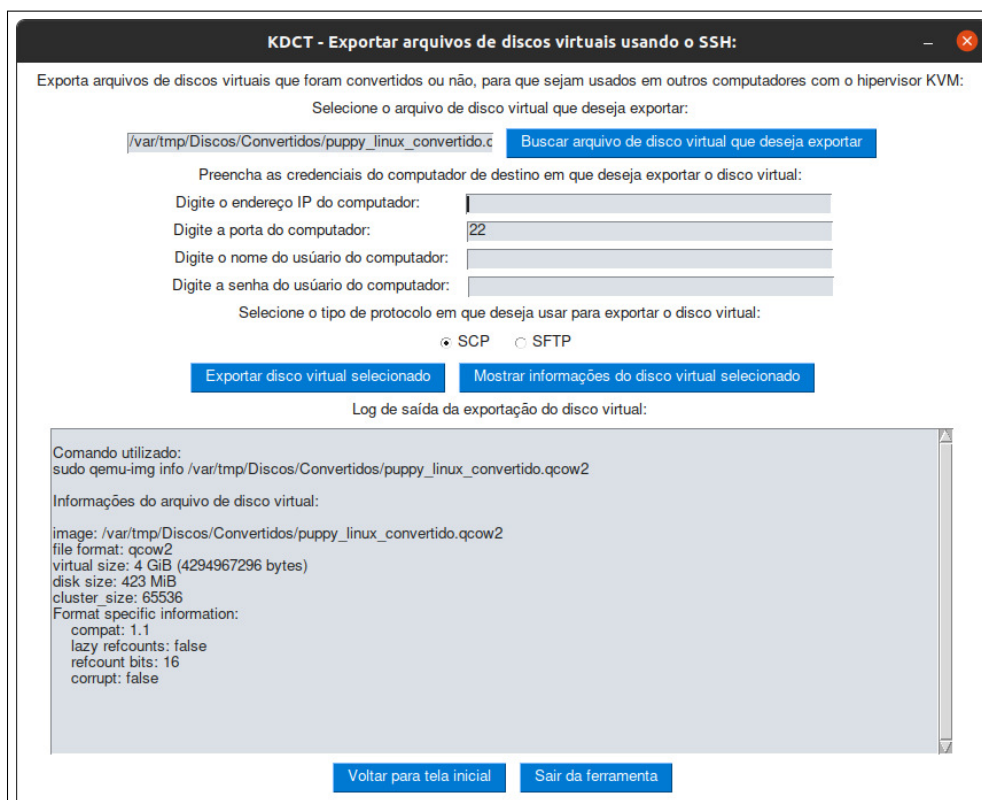
Figura 12 – Exemplo de exportação de disco virtual.



Fonte: Elaborado pelo autor, 2022.

Por meio da tela de conversão de discos, que é mostrada na Figura 10, como também na tela de exportação de discos na Figura 11, os usuários podem obter detalhes sobre os arquivos de discos virtuais selecionados. A Figura 13 mostra detalhes sobre um arquivo de disco que foi convertido pela ferramenta, do formato VDI para o formato QCOW2.

Figura 13 – Exemplo de informações sobre o disco virtual selecionado.



Fonte: Elaborado pelo autor, 2022.

As informações do disco virtual mostradas na Figura 13, são obtidos por meio da ferramenta *qemu-img* e podem variar de acordo com o formato do disco virtual e das funcionalidades suportadas por tal formato de disco. No exemplo da Figura 13, são listadas as seguintes informações sobre o disco QCOW2: *image*: que indica a localização do arquivo de disco virtual, *file format*: que indica o formato do arquivo de disco virtual, *virtual size*: que indica o tamanho virtual do arquivo de disco virtual, tamanho que é reconhecido como tamanho de um disco físico pelo SO convidado da VM que utiliza o disco e *cluster size*: que indica o tamanho do *cluster* do arquivo de disco virtual. Além disso, são listadas informações denominadas de *format specific information*, que são informações específicas que variam de acordo com o formato de disco. No exemplo são listadas as seguintes informações correspondentes ao formato QCOW2: *compat*, *lazy refcounts*, *refcount* e *corrupt*.

4 CONCLUSÃO

Após a identificação das dificuldades que envolvem o processo de migração *offline* de máquinas virtuais de outros hipervisores para ambientes com KVM, surgiu a ideia do desenvolvimento de uma ferramenta com interface gráfica para o SO Linux, que auxilie tal processo. Após a concepção dos objetivos deste trabalho e logo depois, a fase de análise de sistema, tornou possível a implementação da ferramenta proposta.

Com a utilização da ferramenta KDCT o usuário pode realizar migrações de VMs de outros hipervisores para o KVM, mediante da técnica de migração *offline* de VMs. Por meio da funcionalidade de conversão de discos virtuais, os discos oriundos de outros hipervisores podem ser convertidos para os formatos utilizados pelo KVM: RAW e QCOW2. E após a conversão dos discos para formatos nativos do KVM, o usuário pode ainda, exportá-los via rede para outro computador com KVM, no qual deseje utilizar os discos convertidos para recriar tais VMs. Com isso, espera-se que os usuários obtenham mais agilidade e facilidade durante a realização das migrações.

O uso das funcionalidades disponíveis na ferramenta traz mais prontidão e menos preocupações para seus usuários, pois, os mesmos não precisarão mais se preocupar em descobrir quais os comandos e parâmetros corretos utilizar para por exemplo, selecionar um arquivo de disco virtual oriundo do *VirtualBox* e convertê-lo do formato VDI para o formato QCOW2, para que seja posteriormente exportado para o KVM. Antes o usuário precisaria pesquisar uma ferramenta para realizar a conversão manualmente e, ainda, estudar os comandos e parâmetros utilizados para realizar tal processo. Além do mais, após a conversão, o usuário ainda precisaria de comandos SSH para transferir os arquivos manualmente para outro computador com KVM.

Já com a ferramenta KDCT, o usuário precisa apenas indicar onde se localiza o arquivo de disco virtual que deseja converter ou exportar, depois ele precisa indicar o formato no qual deseja converter o arquivo e onde deseja salvá-lo ou exportá-lo. O último passo é apenas clicar no botão para iniciar a conversão ou exportação do disco virtual e aguardar que o processo termine.

Por fim, conclui-se, que todas as funcionalidades propostas foram implementadas e todos objetivos foram alcançados, com exceção do objetivo referente a publicação dos resultados, que não foi possível alcançá-lo devido a limitação de tempo para realização do trabalho de conclusão de curso.

4.1 TRABALHOS FUTUROS

Baseando-se no paradigma de XaaS para implementações futuras, foi cogitado utilizar a ferramenta desenvolvida neste trabalho para prover um serviço de conversão de discos virtuais, que pode ser integrado em ambientes de computação nas nuvens. Um exemplo de uso deste serviço seria a integração com o *OpenStack Compute (nova)*¹, que é uma parte da arquitetura do *OpenStack* que é responsável pelo gerenciamento das VMs, no qual, a ferramenta fornecerá o serviço de conversão de discos virtuais para a realização do processo de migração *offline* de VMs para esses ambientes de nuvens *OpenStack*.

¹ OpenStack Compute (nova). Disponível na URL: <<https://docs.openstack.org/nova/latest/>>, acesso em: 05 abr. 2022.

REFERÊNCIAS

- BARDEN, C. d. O. Configuração de um ambiente escalável usando docker swarm para aplicações de big data. Universidade Federal de Santa Maria, 2021.
- BOLTE, M.; SIEVERS, M.; BIRKENHEUER, G.; NIEHÖRSTER, O.; BRINKMANN, A. Non-intrusive virtualization management using libvirt. In: **2010 Design, Automation Test in Europe Conference Exhibition (DATE 2010)**. [S.l.: s.n.], 2010. p. 574–579.
- BUJOR, d.-C.; DOBRE, R. Kvm io profiling. In: **2013 RoEduNet International Conference 12th Edition: Networking in Education and Research**. [S.l.: s.n.], 2013. p. 1–6.
- CACIATO, L. E. Virtualização e consolidação dos servidores do datacenter. **Centro de Computação da Universidade Estadual de Campinas—São Paulo, (23)**, 2009.
- CARISSIMI, A. Virtualização: da teoria a soluções. **26º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, p. 173–207, 2008.
- CHE, J.; SHI, C.; YU, Y.; LIN, W. A synthetical performance evaluation of openvz, xen and kvm. In: **2010 IEEE Asia-Pacific Services Computing Conference**. [S.l.: s.n.], 2010. p. 587–594.
- CHRISTOPHER, S. Time sharing in large fast computers. **Proceedings of the International Conference on Information processing**, Nova York, 1959.
- CLEMENTINO, D.; CARMIN, M.; DITTER, A.; ALVES, M. Proposta de mecanismo para replicação de comportamento de nuvem para pesquisa e reprodutibilidade. In: **Anais da XXI Escola Regional de Alto Desempenho da Região Sul**. Porto Alegre, RS, Brasil: SBC, 2021. p. 13–16. ISSN 2595-4164.
- DINIZ, L. **Migração do Xenserver para Proxmox e seus benefícios: um caso de uso no Campus Rio Pomba. 2018**. 2019.
- FENN, M.; GOASGUEN, S.; LAURET, J. Contextualization in practice: The clemson experience. **ACAT Proceedings**, 2010.
- GHANNOUM, R. G.; RODRIGUES, F. B. Virtualização de servidores: Vantagens e desvantagens. **Revista Mirante**, Anápolis, GO, Brasil, v. 11, n. 6, 2018.
- JOSHI, G.; SHINGADE, S.; SHIROLE, M. Empirical study of virtual disks performance with kvm on das. In: **2014 International Conference on Advances in Engineering Technology Research (ICAETR - 2014)**. [S.l.: s.n.], 2014. p. 1–8.
- LAUREANO, M. A. P.; MAZIERO, C. A. Virtualização: Conceitos e aplicações em segurança. **Livro-Texto de Minicursos SBSeg**, p. 1–50, 2008.
- MACHADO, J. P. M. **Virtualização de desktops VDI: o caso do ISCTE-IUL**. Tese (Doutorado) — Instituto Universitário de Lisboa, 2019.
- MACHADO, J. V. K.; SILVA, A. S. da. Virtualização de servidores com hyper-v. **PROJETOS E RELATÓRIOS DE ESTÁGIOS**, v. 1, n. 1, p. 1–58, 2019.

MALISZEWSKI, A.; VOGEL, A.; GRIEBLER, D.; SCHEPKE, C.; NAVAUX, P. Ambiente de nuvem computacional privada para teste e desenvolvimento de programas paralelos. **Sociedade Brasileira de Computação**, 2021.

MATTOS, D. M. F. Virtualização: Vmware e xen. **Universidade Federal do Rio de Janeiro**, 2008.

PATEL, P. D.; KARAMTA, M.; BHAVSAR, M.; POTDAR, M. Live virtual machine migration techniques in cloud computing: A survey. **International Journal of Computer Applications**, Citeseer, v. 86, n. 16, 2014.

RAITZ, L.; PÉRICAS, F. A. Utilização de máquinas virtuais para implantar um mecanismo transparente de detecção de intrusão em servidores web. **XIV Seminário de Computação**, Único, 2005.

SOMMERVILLE, I. **Engenharia de software-9ª Edição**. São Paulo, SP, Brasil: Pearson Prentice Hall, 2011. ISBN 9788579361081.

VERAS, M. **Virtualização**. [S.l.]: Brasport, 2011.

APÊNDICE A – CASOS DE USO

Quadro 5 – UC1 - Instalar dependências.

Caso de uso:	UC1 - Instalar dependências.
Descrição:	Permite instalar as dependências externas da ferramenta no SO do computador, para que seja possível utilizar todas as funcionalidades disponíveis na ferramenta.
Atores:	Usuário e SO Linux.
Pré-condição:	1- O usuário deve estar na tela inicial da ferramenta; 2- O computador necessita de conexão com a Internet, para que as dependências sejam baixadas e instaladas.
Fluxo normal:	1- O usuário clica no botão de instalar as dependências; 2- O usuário confirma que deseja instalar as dependências; 3- O processo de instalação das dependências é iniciado e as dependências são instaladas no SO Linux; 4- Um popup com a mensagem de confirmação de instalação concluída com sucesso é exibido na ferramenta; 5- O usuário deve reiniciar a ferramenta, para que o acesso às funcionalidades seja liberado.
Fluxo de exceção:	E1: O computador está sem acesso à Internet no momento: 1- Um popup com a mensagem informando que a instalação não foi concluída deve ser exibido na ferramenta.
Pós-condição:	O acesso a todas as funcionalidades da ferramenta é liberado.

Fonte: Elaborado pelo autor, 2022.

Quadro 6 – UC2 - Escolher funcionalidade.

Caso de uso:	UC2 - Escolher funcionalidade.
Descrição:	O usuário poderá escolher entre a funcionalidade de conversão ou exportação de discos virtuais.
Ator:	Usuário.
Pré-condição:	O usuário deve estar na tela inicial da ferramenta.
Fluxo normal:	1- O usuário escolhe qual funcionalidade irá utilizar no momento, se é a de conversão ou a de exportação de discos virtuais; 2- O usuário clica no botão relacionado a funcionalidade escolhida.
Fluxo secundário:	2 a) Os botões de acesso as funcionalidades estão desabilitados: 1- O usuário deve instalar as dependências da ferramenta para liberar o acesso às funcionalidades; 2- Retorna para o passo 1 do fluxo principal.
Pós-condição:	A tela da funcionalidade escolhida é exibida na ferramenta.

Fonte: Elaborado pelo autor, 2022.

Quadro 7 – UC3 - Selecionar disco virtual.

Caso de uso:	UC3 - Selecionar disco virtual.
Descrição:	O usuário poderá escolher qual disco virtual irá utilizar para realizar o processo de conversão ou exportação do mesmo.
Ator:	Usuário.
Pré-condição:	O usuário deve estar na tela de conversão ou de exportação de discos virtuais.
Fluxo normal:	1- O usuário escolhe com o seletor de arquivos da interface, qual disco virtual irá utilizar.
Fluxo de exceção:	E1: O disco selecionado é inválido: 1- Um popup de erro informando que o arquivo de disco virtual selecionado não é válido deve ser exibido.
Pós-condição:	O disco virtual escolhido pelo usuário poderá ser usado para realizar a conversão ou exportação.

Fonte: Elaborado pelo autor, 2022.

Quadro 8 – UC4 - Selecionar diretório de destino.

Caso de uso:	UC4 - Selecionar diretório de destino.
Descrição:	O usuário poderá escolher em qual diretório de destino deseja salvar o disco virtual que irá ser convertido.
Ator:	Usuário.
Pré-condição:	O usuário deve estar na tela de conversão de discos virtuais.
Fluxo normal:	1- O usuário escolhe com o seletor de diretórios, qual diretório deseja utilizar para salvar o arquivo de disco que irá ser convertido.
Fluxo de exceção:	E1: O diretório selecionado é inválido: 1- Um popup de erro informando que o diretório selecionado não é válido deve ser exibido.
Pós-condição:	O arquivo de disco virtual que será convertido poderá ser salvo no diretório de destino selecionado pelo usuário.

Fonte: Elaborado pelo autor, 2022.

Quadro 9 – UC5 - Processar conversão de disco.

Caso de uso:	UC5 - Processar conversão de disco.
Descrição:	Permite realizar o processamento da conversão dos arquivos de discos virtuais, utilizando a ferramenta de conversão <i>qemu-img</i> .
Ator:	<i>qemu-img</i> .
Pré-condição:	Todos os parâmetros necessários devem estar preenchidos na interface gráfica.
Fluxo normal:	1- Por meio dos parâmetros preenchidos na interface gráfica da ferramenta, o comando necessário para realizar a conversão de disco desejada com o <i>qemu-img</i> é gerado; 2- O comando gerado é enviado para a ferramenta <i>qemu-img</i> ; 3- A ferramenta de conversão <i>qemu-img</i> obtém o comando que foi gerado pela interface gráfica, para iniciar o processamento da conversão do disco; 4- A conversão é iniciada, de acordo com os parâmetros definidos na interface gráfica.
Fluxo de exceção:	E1: Falha no processo de conversão de disco virtual: 1- Um popup informando que a conversão do disco virtual não foi concluída com sucesso deve ser exibido na ferramenta, trazendo ainda o log do erro ocorrido durante o processo.
Pós-condição:	O disco virtual foi convertido para o formato especificado.

Fonte: Elaborado pelo autor, 2022.

Quadro 10 – UC6 - Preencher credenciais.

Caso de uso:	UC6 - Preencher credenciais.
Descrição:	O usuário poderá preencher as credenciais do computador em que deseja exportar os discos virtuais.
Ator:	Usuário.
Pré-condição:	O usuário deve estar na tela de exportação de discos virtuais.
Fluxo normal:	1- O usuário preenche o endereço IP do computador de destino; 2- O usuário preenche a porta do computador de destino; 3- O usuário preenche o nome de usuário do computador de destino; 4- O usuário preenche a senha do computador de destino.
Fluxo de exceção:	E1: Credenciais informadas inválidas: 1- Um popup de erro é exibido informando que não foi possível se conectar ao computador remoto devido uma falha de autenticação.
Pós-condição:	O disco virtual poderá ser exportado para o computador de destino, no qual é indicado pelas informações preenchidas pelo usuário.

Fonte: Elaborado pelo autor, 2022.

Quadro 11 – UC7 - Exportar disco virtual

Caso de uso:	UC7 - Exportar disco virtual
Descrição:	Permite realizar a exportação de arquivos de discos virtuais para outros computadores, utilizando o SSH.
Ator:	Usuário e SSH.
Pré-condição:	1- O usuário deve estar na tela de exportação de discos virtuais; 2- As credenciais do computador de destino devem estar preenchidas corretamente pelo usuário.
Fluxo normal:	1- O usuário clica no botão para iniciar o processo de exportação de disco; 2- O SSH estabelece a conexão com o computador de destino, utilizando as credenciais preenchidas pelo usuário; 3- Um popup com a mensagem de confirmação de exportação concluída com sucesso é exibido na interface da ferramenta.
Fluxo de exceção:	E1: Falha no processo de exportação do disco virtual: 1- Um popup de erro informando que a exportação do disco virtual não foi concluída com sucesso deve ser exibido na ferramenta, contendo ainda, o log do erro ocorrido no processo.
Pós-condição:	A exportação do disco virtual foi concluída com sucesso.

Fonte: Elaborado pelo autor, 2022.

Quadro 12 – UC8 - Desinstalar dependências.

Caso de uso:	UC8 - Desinstalar dependências.
Descrição:	O usuário poderá, quando desejar, desinstalar todas as dependências necessárias para o funcionamento da ferramenta.
Ator:	Usuário e SO Linux.
Pré-condição:	O usuário deve estar na tela inicial da ferramenta.
Fluxo normal:	<p>1- O usuário clica no botão de desinstalar as dependências;</p> <p>2- O usuário confirma que deseja desinstalar as dependências;</p> <p>3- O processo de desinstalação das dependências é iniciado e as dependências serão desinstaladas do SO Linux.</p> <p>4- Um popup deve ser exibido na ferramenta, com a mensagem de confirmação da desinstalação de todas as dependências da ferramenta.</p>
Fluxo de exceção:	<p>E1: Falha no processo de desinstalação das dependências:</p> <p>1- Um popup de erro com a mensagem informando que a instalação não foi concluída deve ser exibido na ferramenta.</p>
Pós-condição:	Todas as dependências externas da ferramenta foram desinstaladas do SO do computador no qual a ferramenta está em uso.

Fonte: Elaborado pelo autor, 2022.

APÊNDICE B – REQUISITOS

Requisitos funcionais:

- **RF_01 – Instalar as dependências do sistema:** A ferramenta deve possuir a funcionalidade de instalar todas as dependências necessárias no SO do computador, para a execução da mesma.
Prioridade: Essencial Importante Desejável
- **RF_02 – Compatibilidade de formatos de discos virtuais:** A ferramenta deve ler diferentes formatos de imagens de discos virtuais que forem selecionados pelo usuário.
Prioridade: Essencial Importante Desejável
- **RF_03 – Exibir informações do disco virtual:** A ferramenta deve exibir as informações do arquivo de disco virtual que foi selecionado pelo usuário.
Prioridade: Essencial Importante Desejável
- **RF_04 – Converter discos virtuais para o KVM:** A ferramenta deve comunicar-se com o *qemu-img*, para realizar o processo de conversão de discos virtuais, oferecendo suporte para os diferentes formatos de discos.
Prioridade: Essencial Importante Desejável
- **RF_05 – Exportar discos virtuais:** A ferramenta deve ser capaz de exportar para outros computadores os arquivos de discos virtuais que foram convertidos, utilizando as credenciais do usuário.
Prioridade: Essencial Importante Desejável
- **RF_06 – Desinstalar as dependências do sistema:** A ferramenta deve possuir a funcionalidade de desinstalar todas as dependências do SO do computador.
Prioridade: Essencial Importante Desejável

Requisitos não funcionais:

- **RNF_01 – Compatibilidade:** A ferramenta deve ser compatível com o sistema operacional Linux, ambiente nativo do KVM.
- **RNF_02 - Facilidade de uso:** A ferramenta deve possuir textos de ajuda em suas funcionalidades da interface gráfica.
- **RNF_03 - Padrões:** A ferramenta deve seguir os padrões de estilos de design da plataforma Linux.
- **RNF_04 – Implementação:** A ferramenta deve ser implementada usando a linguagem de programação Python, por ser uma linguagem que vem pré-instalada em diversos SOs Linux.

Documento Digitalizado Restrito

Documento de TCC.

Assunto: Documento de TCC.
Assinado por: Sergio Carvalho
Tipo do Documento: Tese
Situação: Finalizado
Nível de Acesso: Restrito
Hipótese Legal: Informação Pessoal (Art. 31 da Lei no 12.527/2011)
Tipo do Conferência: Cópia Simples

Documento assinado eletronicamente por:

- Sergio Coelho de Carvalho, ALUNO (201812010039) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS, em 08/06/2022 12:56:59.

Este documento foi armazenado no SUAP em 08/06/2022. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 541982

Código de Autenticação: 5daac1b4a1

