



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
TECNÓLOGO EM REDES DE COMPUTADORES

WALAFI FERREIRA DA SILVA

**UTILIZANDO VIRTUALIZAÇÃO BASEADA EM CONTAINERS PARA CRIAÇÃO
DE LABORATÓRIOS PRÁTICOS DE DISCIPLINAS NA ÁREA DE TI**

QUIXADÁ – CEARÁ

2017

WALAFI FERREIRA DA SILVA

UTILIZANDO VIRTUALIZAÇÃO BASEADA EM CONTAINERS PARA CRIAÇÃO DE
LABORATÓRIOS PRÁTICOS DE DISCIPLINAS NA ÁREA DE TI

Monografia apresentada no curso de Redes de Computadores da Universidade Federal do Ceará, como requisito parcial à obtenção do título de tecnólogo em Redes de Computadores. Área de concentração: Computação.

Orientador: Prof. Dr. Paulo Antonio Leal Rego

QUIXADÁ – CEARÁ

2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S584u Silva, Walafi Ferreira da.
Utilizando virtualização baseada em containers para criação de laboratórios práticos de disciplinas na área de TI / Walafi Ferreira da Silva. – 2017.
44 f. : il.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Redes de Computadores, Quixadá, 2017.
Orientação: Prof. Dr. Paulo Antonio Leal Rego.

1. Containers. 2. Laboratórios- Prática. 3. Virtualização. I. Título.

CDD 004.6

WALAFI FERREIRA DA SILVA

UTILIZANDO VIRTUALIZAÇÃO BASEADA EM CONTAINERS PARA CRIAÇÃO DE
LABORATÓRIOS PRÁTICOS DE DISCIPLINAS NA ÁREA DE TI

Monografia apresentada no curso de Redes de Computadores da Universidade Federal do Ceará, como requisito parcial à obtenção do título de tecnólogo em Redes de Computadores. Área de concentração: Computação.

Aprovada em: ____/____/____

BANCA EXAMINADORA

Prof. Dr. Paulo Antonio Leal Rego (Orientador)
Universidade Federal do Ceará – UFC

Prof. Me. Bruno Góis Mateus
Universidade Federal do Ceará - UFC

Prof. Me. Antonio Rafael Braga
Universidade Federal do Ceará - UFC

Aos meus pais e familiares que me apoiaram a
todo momento.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter me dado forças nesse percurso árduo.

Aos meus pais Ana Iris Ferreira e Deuzemar Taveiro e a minha irmã Fernanda Ferreira por todo apoio, não somente durante a graduação, mas por toda vida.

Ao professor Dr. Paulo A. L. Rego por toda paciência e dedicação na orientação deste trabalho, que sempre se mostrou um profissional excelente e competente.

Aos professores da banca: Antonio Rafael Braga e Bruno Góis Mateus, por suas enriquecedoras sugestões.

Agradeço aos meus amigos Ana Lucia, Renan e Fábio por todo o apoio durante a graduação.

Aos colegas da turma de graduação, pelas reflexões, críticas e sugestões recebidas.

“Não sabendo que era impossível, foi lá e fez.”

(Jean Cocteau)

RESUMO

O desenvolvimento de práticas de laboratório em disciplinas na área de Tecnologia da Informação (TI) é uma forma importante para oferecer aos alunos um meio de obter experiência. Mas o problema por trás dessas práticas é que as instituições de ensino geralmente não possuem verba suficiente para disponibilizar uma estrutura dedicada para esses laboratórios. Dessa forma, a virtualização torna-se uma saída para se solucionar esse problema, possibilitando a execução de sistemas convidados sobre um anfitrião, reduzindo assim a infraestrutura física. Na virtualização baseada em hipervisores, um sistema operacional completo é executado sobre um hospedeiro, o que se torna um problema, pois nesse tipo de virtualização há um consumo excessivo de recursos do hospedeiro. Pensando nisso, surgiu a virtualização baseada em *containers*, que, diferente dos hipervisores, armazenam apenas as bibliotecas necessárias para executar uma aplicação, compartilhando o kernel com o hospedeiro, o que torna esse tipo de virtualização mais leve e eficiente. Sabendo disso, este trabalho tem por objetivo apresentar uma ferramenta que automatiza a criação de ambientes de estudo virtualizado para alunos da área de TI utilizando *containers*, que possibilita assim a criação, execução ou gerenciamento de um cenário para atividade prática de forma rápida e fácil. Os resultados obtidos a partir da avaliação quantitativa realizada mostram que a ferramenta desenvolvida agradou aos professores entrevistados e foi considerada um instrumento útil para o desenvolvimento das atividades de laboratório.

Palavras-chave: *Containers*. Laboratórios- Prática. Virtualização.

ABSTRACT

The development of laboratory practice in courses of Information Technology (IT) is an important way to offer students a means of gaining experience. But the problem behind these practices is that educational institutions generally do not have enough money to provide a dedicated structure for these laboratories. Thus, virtualization becomes a way to solve this problem by enabling the execution of guests on a host systems, thereby reducing the physical infrastructure. In-based virtualization hypervisors, a complete operating system runs on a host, which becomes a problem, because in this type of virtualization there is excessive consumption of the host resources. To solve this, containers based virtualization has emerged, on which, unlike the hypervisors, store only the necessary libraries to run an application, sharing the kernel with the host, which makes this type of virtualization lighter and efficient. Knowing this, this paper aims to present a tool that automates the creation of virtualised study environments for students in the area of IT using containers, which enables the creation, execution or management of a scenario for practical activity quickly and easily. The results obtained from the quantitative evaluation carried out show that the developed tool pleased the teachers interviewed and was considered a useful tool for the development of laboratory activities.

Keywords: Containers. Laboratories- Practice. Virtualization.

LISTA DE FIGURAS

Figura 1 – Virtualização total	15
Figura 2 – Paravirtualização	16
Figura 3 – Virtualização baseada em <i>containers</i>	17
Figura 4 – Arquitetura Docker	18
Figura 5 – Arquitetura da ferramenta	23
Figura 6 – Script de criação de cenário	24
Figura 7 – Cenário proposto em sala	24
Figura 8 – Esquema do banco de dados	26
Figura 9 – Tela de primeiro acesso ao servidor	26
Figura 10 – <i>Scripts</i> criados no <i>home</i> do usuário aluno	27
Figura 11 – Tela de acesso a máquina	27
Figura 12 – Opções da ferramenta	28
Figura 13 – Cenários disponíveis	28
Figura 14 – Dados do cenário	29
Figura 15 – Matrículas	30
Figura 16 – Gerenciamento de cenários	30
Figura 17 – Gerenciamento de <i>containers</i>	30
Figura 18 – Máquina de estado	32
Figura 19 – Tela de criação de cenário	32
Figura 20 – Tela de implementação de <i>script</i>	33
Figura 21 – Tela de máquinas de um novo cenário	33
Figura 22 – Formulário de novas máquinas	34
Figura 23 – Tela de redes disponíveis	34
Figura 24 – Formulário de novas redes	34
Figura 25 – Tela de dados do cenário	35

LISTA DE TABELAS

Tabela 1 – Questionário aluno.	38
Tabela 2 – Questionário professor.	38
Tabela 3 – Resultados: Questionário dos alunos	39
Tabela 4 – Resultados: Questionário dos professores	41

LISTA DE ABREVIATURAS E SIGLAS

CCNA	Cisco Certified Network Associate
MV	Máquina Virtual
NAT	Network Address Translation
TI	Tecnologia da Informação
VMM	Virtual Machine Monitor

SUMÁRIO

1	Introdução	12
1.1	Objetivos	13
1.1.1	<i>Objetivo geral</i>	13
1.1.2	<i>Objetivos específicos</i>	13
2	Fundamentação Teórica	14
2.1	Virtualização	14
2.1.1	<i>Virtualização total</i>	14
2.1.2	<i>Paravirtualização</i>	15
2.1.3	<i>Virtualização baseada em containers</i>	16
2.1.3.1	<i>Docker</i>	17
2.2	E-learning	18
3	Trabalhos Relacionados	21
4	Solução proposta	23
4.1	Arquitetura e componentes	23
4.1.1	<i>Acesso dos usuários alunos ao servidor e às máquinas</i>	25
4.2	Funcionalidades da ferramenta	27
4.2.1	<i>Instanciar cenários</i>	28
4.2.2	<i>Gerenciar cenários instanciados</i>	29
4.2.3	<i>Criar cenário</i>	31
4.3	Requisitos	35
5	Avaliação da ferramenta	37
5.1	Metodologia de avaliação	37
5.2	Resultados	38
5.3	Discussão dos resultados	39
6	Conclusão	42
	REFERÊNCIAS	43

1 INTRODUÇÃO

Nos últimos anos, a virtualização vem ganhando grande espaço na área de TI, sendo muito utilizada em servidores para fazer um melhor aproveitamento dos recursos do mesmo (GARCIA et al., 2016). Porém, a virtualização não se limita a isso, pois ela pode ser útil também em ambientes de uso doméstico por usuários comuns ou para testes em universidades (GARCIA et al., 2016).

Entre as diversas soluções de virtualização existentes, as mais utilizadas são: hipervisores, que funcionam como uma camada que abstrai o *hardware* das máquinas virtuais, o que resulta em sobrecarga em termos de virtualização de *hardware* e *drivers*; e a virtualização leve ou baseada em *container*, que oferece um nível diferente de abstração em termos de virtualização, isolamento e apresentam um melhor desempenho se comparado aos hipervisores (MORABITO; KJÄLLMAN; KOMU, 2015).

Tendo grande importância para os cursos da área de TI, as atividades de laboratório desempenham um papel importante, pois não só ajuda os alunos a compreenderem o que está sendo ensinado, mas também lhes proporciona habilidades de resolução de problemas e experiência necessárias para lidar com possíveis situações do mundo real (WANNOUS; NAKANO, 2010). Nos laboratórios de redes, por exemplo, os alunos podem acessar um conjunto de dispositivos de rede, com os quais ele pode construir e testar redes com diferentes estruturas e componentes.

Segundo Garcia et al. (2016), a implementação de laboratórios práticos para o ensino de redes de computadores vem se tornando cada vez mais comum no modelo de ensino atual. As práticas de laboratório devem simular eventos reais aos aprendizes e permitir que eles vivenciem experiências para o futuro. Entretanto, um grande problema que está por trás desses laboratórios é a estrutura física para as aulas práticas, que geralmente não suprem as necessidades. Pensando nisso, a virtualização surge como uma saída para contornar esse problema.

Para Carissimi (2008), a virtualização torna-se um meio para viabilização de práticas de laboratório e ensino em universidades. O autor ressalta que, em algumas disciplinas, há necessidade de se ter diversos sistemas operacionais diferentes em execução ao mesmo tempo e, na maior parte dos casos, é comum haver a necessidade de o aluno se ter acesso administrativo. Utilizando um ambiente virtualizado, o participante de uma disciplina pode executar diversos sistemas virtualizados simultaneamente e fazer todos os experimentos, que por serem executados de forma isolada, suas ações como administrador não afetam o sistema nativo.

Sabendo a ligeira vantagem que *containers* levam sobre os hipervisores e a necessidade de se utilizar virtualização para a criação de laboratórios virtuais para a prática de atividades de laboratório, o presente estudo tem por objetivo desenvolver uma solução, voltada aos professores e alunos da área de TI, para automatizar a criação de laboratórios virtuais para disciplinas da área, utilizando a tecnologia de virtualização baseada em *containers*.

Para o processo de desenvolvimento, primeiro foram observadas as práticas de laboratório executadas pelos professores em algumas disciplinas, e a partir daí foram elaborados requisitos necessários para o pleno funcionamento da ferramenta. Após concluída a implementação da solução, foram realizados testes e ao final de cada teste, foi realizada uma avaliação quantitativa sobre a ferramenta, para saber se os objetivos deste trabalho foram alcançados.

O restante do trabalho está organizado da seguinte maneira. Na Seção 1.1, são apresentados os objetivos geral e específicos. No Capítulo 2, é apresentada a fundamentação teórica. No Capítulo 3 apresenta os trabalhos relacionados. No Capítulo 4 é descrito a solução proposta. No Capítulo 5 é apresentado a avaliação da ferramenta. No Capítulo 6, é apresentado a conclusão.

1.1 Objetivos

1.1.1 Objetivo geral

Desenvolver uma solução que automatiza a criação de ambientes virtualizados para realização de laboratórios e práticas de disciplinas na área de TI.

1.1.2 Objetivos específicos

- a) Fazer levantamento bibliográfico sobre as ferramentas de *e-learning* e laboratórios virtuais, bem como suas características e funcionalidades;
- b) Levantar requisitos da solução a ser desenvolvida;
- c) Definir arquitetura e implementar a solução proposta;
- d) Fazer uma avaliação de usabilidade da solução desenvolvida.

2 FUNDAMENTAÇÃO TEÓRICA

As seções a seguir apresentam conceitos importantes relacionados a este trabalho.

2.1 Virtualização

Segundo Carissimi (2008), a virtualização é a técnica que permite particionar um único sistema computacional em vários outros, denominados de máquinas virtuais. Cada máquina virtual oferece um ambiente similar ao de uma máquina física, possibilitando assim que cada uma tenha seu próprio sistema operacional, aplicativos e serviços de rede.

O conceito de virtualização surgiu por volta dos anos 60 na IBM, no projeto M44/M44X, que visava avaliar o então recente conceito de compartilhamento de sistemas. Hoje, em um sistema virtualizado temos um *software* sendo executado no *host* denominado Virtual Machine Monitor (VMM) ou hipervisor (*hypervisor*). Esse software é responsável pela criação dos ambientes de computação chamados de máquinas virtuais (MVs) (SEO, 2009).

Com as MVs, além de apresentar uma economia de recursos de *hardware*, pode-se criar ambientes seguros e isolados para a execução de aplicações não-cofiáveis que permitem o monitoramento sem alteração no ambiente de produção, facilitando assim a migração de aplicações e servidores, e permitindo a simulação de *hardware* que o usuário não dispõe (SEO, 2009).

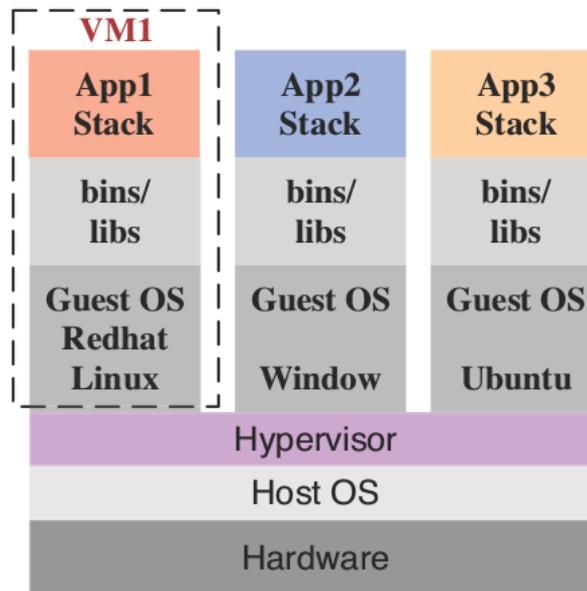
2.1.1 Virtualização total

O trabalho de Jersak (2014) salienta que na virtualização total, um sistema operacional sem alterações é executado sobre uma camada de *software*, chamada *hypervisor*, que abstrai completamente o *hardware* físico. Como não há alteração no sistema operacional, ele não está ciente que está sendo virtualizado, então o *hypervisor* torna-se necessário pois controla o acesso do sistema virtualizado aos dispositivos de *hardware* do hospedeiro.

A Figura 1 ilustra como está organizada a arquitetura da virtualização total. Nela é mostrada a camada chamada *hypervisor* que possui uma interação com as camadas do sistema operacional e *hardware* do *host*, que a partir dessa interação provem a disponibilidade de recursos para a camada executada acima, camada de sistemas operacionais visitantes (ZHANG; LU; PANDA, 2016).

Segundo Jersak (2014), é importante notar que esse processo de virtualização pode

Figura 1 – Virtualização total



Fonte – Adaptado de Zhang, Lu e Panda (2016)

ser bastante oneroso do ponto de vista de desempenho, já que o hipervisor deve emular todos os dispositivos da plataforma com detalhamento suficiente para que o sistema operacional possa manipulá-los em baixo nível. Segundo o mesmo autor, o processo de capturar e traduzir as instruções vindas do sistema virtualizado torna-se um fator relevante na redução do desempenho. Como exemplo de virtualização total existem: VMWare¹ e KVM² (MAURICIO, 2013).

2.1.2 Paravirtualização

A paravirtualização é uma abordagem alternativa que surgiu para contornar as desvantagens presentes na virtualização total sem tentar replicar exatamente o comportamento original do ambiente visitante. Essa abordagem necessita que o sistema operacional convidado seja modificado para ser executado no ambiente paravirtualizado (ANISETTI et al., 2007).

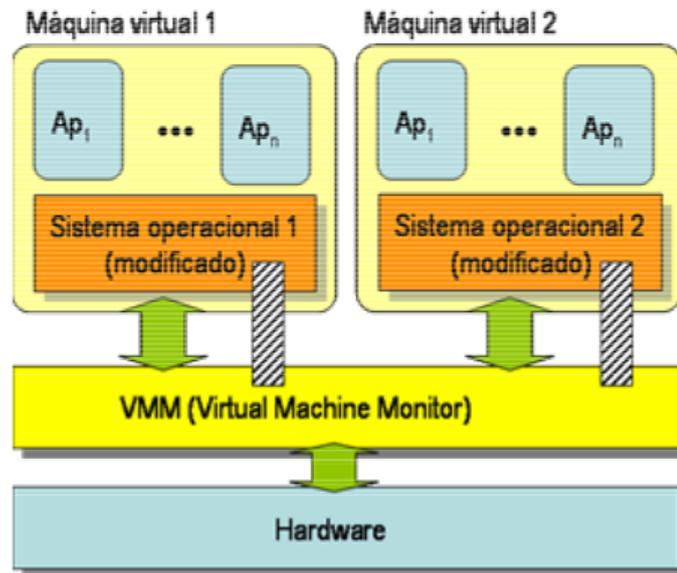
Nesse tipo de virtualização há uma modificação na infraestrutura de entrega de *hardware* para o sistema virtualizado para ficar mais semelhante possível ao *hardware* físico, mas não idêntica. Desse modo, alguns comandos podem ser enviados ao sistema operacional no *host* através de *drivers*, como ilustrado da Figura 2. O Xen³ pode ser utilizado como exemplo solução de paravirtualização mais utilizada (BERTOCHI; BELLEZI, 2016).

¹ <https://www.vmware.com/>

² <https://www.linux-kvm.org>

³ <https://xenserver.org/>

Figura 2 – Paravirtualização



Fonte – Carissimi (2008)

2.1.3 Virtualização baseada em containers

Também conhecida como virtualização a nível de sistema operacional, a virtualização baseada em *containers* surgiu como uma alternativa para a virtualização de *hipervisor* com o conceito de redução do *overhead* nas operações de entrada e saída em ambientes virtuais. *Containers* oferecem uma camada de virtualização leve, que promete um desempenho quase nativo (XAVIER et al., 2013).

Segundo Zhang, Lu e Panda (2016), na virtualização baseada em *container*, o *kernel* do *host* permite a execução de múltiplas instâncias isoladas, que compartilham o mesmo *kernel*, mas que executam uma pilha de *software* diferente (bibliotecas, serviços ou aplicações), como apresentado na Figura 3.

Containers são construídos a partir de duas tecnologia presentes no Linux. A primeira delas, o *cgroups* ou grupo de controle, é uma implementação do *kernel* do Linux que é utilizada para isolar o uso de diferentes tipos de recursos, tais como: memória, CPU e I/O de disco, sendo o recurso de memória o que mais afeta o desempenho de uma aplicação (ZHUANG et al., 2017). A segunda, o isolamento de *namespace*, pode ser utilizado para isolar processos em níveis de grupo diferente. Algumas das soluções baseados em *container* mais utilizadas são: Docker e LXC (ZHANG; LU; PANDA, 2016).

Segundo Morabito, Kjällman e Komu (2015), a abordagem da utilização de *kernel*

Figura 3 – Virtualização baseada em *containers*

Fonte – Fonte: Adaptado de Morabito (2017)

compartilhado presente na virtualização em *containers* apresenta uma desvantagem importante, como a impossibilidade de execução de *containers* baseado no Windows sobre um anfitrião Linux, já que *containers* compartilham o *kernel* do *host*.

Para este trabalho, será utilizado o Docker como solução de virtualização baseada em *containers*. Tal justificativa está descrita na próxima seção, explicando funcionalidades disponíveis no Docker.

2.1.3.1 Docker

Lançado no dia 25 de março de 2013 (MANU et al., 2016), o Docker⁴ é a plataforma de código aberto mais popular entre usuários e centros de TI para executar e gerenciar *containers*. Nele, pode-se executar uma ferramenta de *software*, empacotar como uma imagem, armazenar e compartilhar no repositório *Docker Hub* (AZAB, 2017).

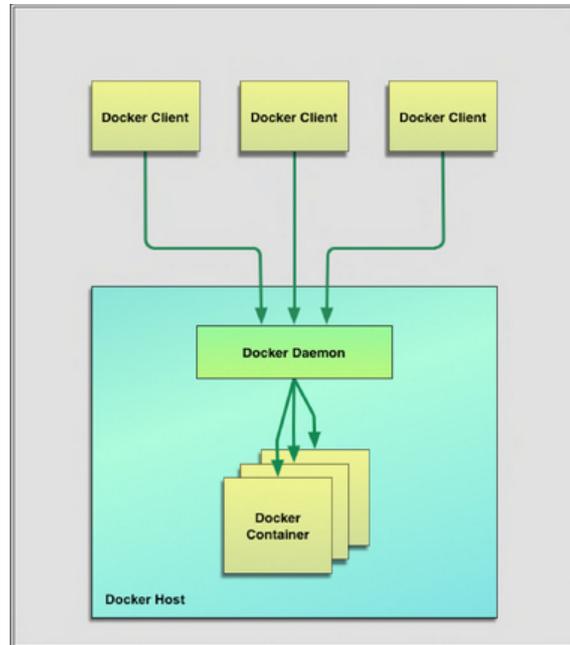
O Docker armazena as imagens construídas em registros, que podem ser públicos e privados. O *Docker Hub* é um repositório onde se pode armazenar e distribuir imagens, podendo ser aberta ao público (registro público) ou restrito a um grupo de pessoas (registro privado) (TURNBULL, 2014).

Docker possui uma arquitetura cliente-servidor, onde o cliente Docker provê uma interface que facilita a comunicação com o servidor Docker, também conhecido como *Docker Daemon* ou *Docker Engine*, que por sua vez faz todo o trabalho de gerenciamento dos *containers*

⁴ <https://www.docker.com/>

(TURNBULL, 2014). Tal arquitetura está ilustrada na Figura 4.

Figura 4 – Arquitetura Docker



Fonte – (TURNBULL, 2014)

De acordo com a documentação oficial⁵, Docker oferece funcionalidades importantes relacionadas ao gerenciamento de redes e volumes. Ligada ao gerenciamento de rede, Docker permite a criação de novas interfaces rede, especificando seu nome, tipo (*Bridge*, *None* ou *Host*), *range* de ip e *gateway*, para que seja utilizada em um *container*. Por padrão, todo *container*, ao ser executado, é vinculado a interface do tipo *bridge*. Sabendo disso, há a possibilidade de se especificar uma rede criada no ato execução do *container* ou possibilidade de se conectar uma segunda interface ao *container*. Já com gerenciamento de volumes, Docker permitir a criação um diretório para que seja montado e compartilhado entre a máquina física e *containers* (montagem *host-container*) ou entre *containers* (montagem entre *containers*), com o objetivo de fornecer recursos para a persistência e compartilhamento dos dados.

2.2 E-learning

Com o objetivo de apresentar uma aprendizagem de qualidade, acessível a todos e em qualquer lugar, foram criados ambientes virtuais chamados *e-learning* (MARTÍNEZ-TORRES et al., 2008). A ideia por trás desses ambientes é tornar mecanismo de aprendizagem mais flexíveis

⁵ <https://docs.docker.com/>

e abertos para que os alunos possam seguir uma aprendizagem individual, que se adéque às suas necessidades e interesses. Assim, Tryus e Kachala (2014) define *e-learning* como uma tecnologia educacional eficaz que permite substituir os métodos de aprendizagem tradicionais, reduzindo custos e intensificando a aprendizagem.

Para que isso seja possível, *e-learning* difunde conteúdo instrucional, experiências de aprendizagem entregues ou habilidade por meio de tecnologias eletrônicas na Internet, intranet ou extranet, quebrando limitação de tempo e espaço, e incluindo vários benefícios como a redução de custos (MARTÍNEZ-TORRES et al., 2008).

Entre os diversos portais de aprendizagem *e-learning* existentes, é possível destacar: Coursera, Udemy e Linux Academy. O projeto Coursera oferece cursos gratuitos online a diversas faculdades americanas. Os cursos incluem notas de aula, trabalhos de casa, testes e questionário de provas, além de ter cursos voltados para programação, onde os alunos podem enviar suas tarefas (TRYUS; KACHALA, 2014). O ambiente Udemy é uma plataforma de ensino e aprendizagem online onde qualquer pessoa pode construir e oferecer cursos gratuitamente ou pagos (DIAS; FRENEDOZO, 2015). Já o Linux Academy⁶ é um portal onde são oferecidos cursos avançados de treinamento e certificação online em Linux⁷, AWS⁸, OpenStack⁹ e DevOps¹⁰ para aprender novas habilidades e obter certificação. O Linux Academy cria ambientes virtuais, onde os alunos podem desempenhar atividades na nuvem, como se estivessem administrando toda uma infraestrutura virtualizada.

Para Ferreira et al. (2014), algumas ferramentas voltadas a aprendizagem que estão disponíveis na Internet são: Packet Tracer e Moodle. O Packet Tracer é uma ferramenta desenvolvida pela Cisco para simulação física e lógica de hardware Cisco, fornecendo ferramenta de visualização de enlaces, pacotes e tráfego de rede. Voltado para o ensino em Redes de Computadores com simulações baseadas nos níveis de conhecimentos exigidos para se obter a certificação CCNA da Cisco. O Moodle é um sistema de código aberto voltado ao gerenciamento de cursos. Nele é possível distribuir conteúdos como texto, imagens, livros, vídeos ou áudios e possibilita a criação de exercícios práticos, questionários e exames (KUMAR; GANKOTIYA; DUTTA, 2011).

Esse trabalho cita *e-learning* com o intuito de apresentar os novos meios de

⁶ <https://linuxacademy.com/>

⁷ <https://www.linux.com/>

⁸ <https://aws.amazon.com/>

⁹ <https://www.openstack.org/>

¹⁰ <https://aws.amazon.com/pt/devops/what-is-devops/>

aprendizagem, relacionando com os laboratórios práticos criados pelos mesmos, que é o mesmo objetivo desse trabalho.

3 TRABALHOS RELACIONADOS

Em Garcia et al. (2016), os autores procuram estudar a implementação de ambientes virtualizados para a prática de laboratórios em redes de computadores, mostrando como essas práticas são fundamentais para o desenvolvimento de um bom profissional da área. Para validar o estudo, foram elaborados dois experimentos que utilizam o VirtualBox¹ como ferramenta de virtualização. Os experimentos realizados focaram na provisão de serviços Web e Proxy, e os resultados obtidos apresentaram que o uso das máquinas virtuais se mostraram muito eficazes nos testes realizados, pois com eles foram atendidas todas as necessidades de instalação e configuração de sistema operacional..

Semelhante ao trabalho de Garcia et al. (2016), este trabalho também busca implementar ambientes virtualizados para prática de laboratório, mas em vez de usar virtualização total, será utilizada a virtualização baseada em *containers*, obtendo assim todas as vantagens oferecidas pelos mesmos.

Xu, Huang e Tsai (2014) apresentam uma plataforma de treinamento de laboratório virtual chamada V-Lab, onde é possível fornecer ambientes virtualizados, com os hipervisores Xen ou KVM, para experimentos práticos com essas tecnologias. Também é apresentado nesse trabalho que a plataforma V-Lab pode oferecer uma interface Web interativa para o gerenciamento dos recursos. Semelhante a este trabalho, nosso trabalho também visa oferecer uma plataforma para treinamento com uma interface Web para permitir o gerenciamento dos cenários, mas utilizando a tecnologia de *containers* para criar os ambientes virtualizados.

Buscando uma comparação detalhada do desempenho da virtualização baseada em hipervisores tradicionais e virtualização baseada em *containers*, o trabalho de Morabito, Kjällman e Komu (2015) pode ser usado como inspiração para este trabalho. Nele é feita uma avaliação profunda com uma série de aplicações *benchmark* utilizando como hipervisores tradicionais o KVM e OSv, e como containers o LXC e Docker, para medir a utilização recursos como de CPU, memória, I/O de disco e desempenho da rede, e constatou-se que a sobrecarga introduzida pelos *containers* é quase insignificante.

Baseado nos trabalhos acima relatados, o presente estudo visa a implementação de uma aplicação Web para o gerenciamento de ambientes virtualizados para facilitar as práticas de laboratórios, assim como relatado no trabalho de Xu, Huang e Tsai (2014), mas utilizando *containers*. O trabalho de Morabito, Kjällman e Komu (2015) auxiliou nossas escolhas, pois eles

¹ <https://www.virtualbox.org/>

mostram que a virtualização baseada em *containers* tem um desempenho superior a outros tipos de virtualização, além de possibilitarem a criação de várias instâncias, dado o menor tamanho das imagens se comparado com hipervisores.

4 SOLUÇÃO PROPOSTA

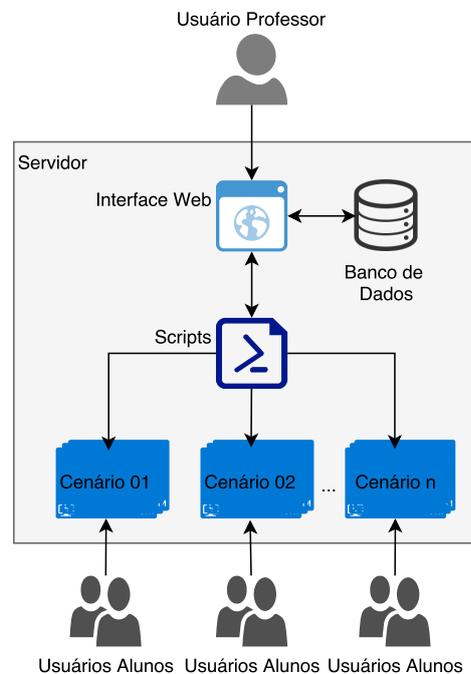
Neste Capítulo é apresentado a arquitetura e componentes da ferramenta, as funcionalidades implementadas, bem como os requisitos funcionais e não funcionais da solução proposta.

4.1 Arquitetura e componentes

As práticas de laboratório propostas pelos professores em diferentes disciplinas (e.g., Sistemas Operacionais Linux, Programação de Scripts e Segurança da Informação) foram observadas para a criação da solução proposta.

Uma visão geral da arquitetura da ferramenta pode ser observada na Figura 5. Nela, o usuário professor tem acesso a uma interface web, onde é possível criar *scripts* para a instanciação de seus próprios cenários ou instanciar cenários a partir dos *scripts* já existentes. Uma vez que um cenário foi criado, os usuários alunos podem acessar o servidor e, utilizando um *script* que é criado automaticamente no diretório *home* do mesmo, podem acessar as máquinas do cenário.

Figura 5 – Arquitetura da ferramenta



Fonte – Próprio autor (2017)

Dos componentes da ferramenta se tem:

- **Scripts:** Visando a automatização da instanciação de cenários, foram implementados

scripts, utilizando a linguagem *Shell Script*, que criam cenários para uma determinada lista de alunos. Um exemplo de *script* pode ser observado a Figura 6. Nele é criado um cenário que será explicado no próximo item.

Figura 6 – Script de criação de cenário

```
#!/bin/bash

sudo docker run --privileged -itd -h $Gateway --name $Gateway ubuntu:1.0 /bin/bash
sudo docker network connect Rede_Alunos $Gateway
sudo docker exec $Gateway ifconfig eth1 0

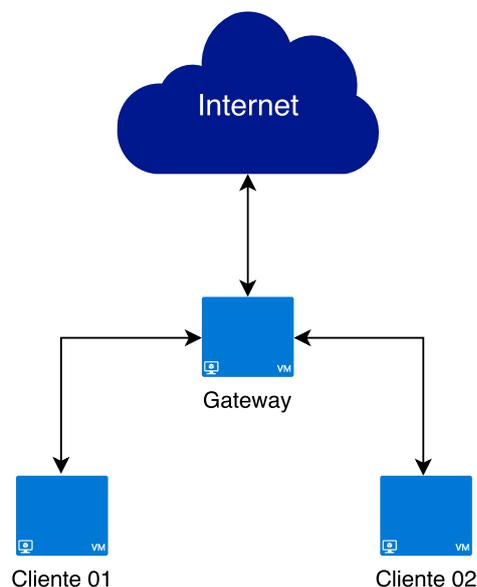
sudo docker run --privileged -itd -h $Maquina01 --name $Maquina01 --net Rede_Alunos ubuntu:1.0 /bin/bash
sudo docker exec $Maquina01 ifconfig eth0 0

sudo docker run --privileged -itd -h $Maquina02 --name $Maquina02 --net Rede_Alunos ubuntu:1.0 /bin/bash
sudo docker exec $Maquina02 ifconfig eth0 0
```

Fonte – Próprio autor (2017)

- **Cenários:** Resultado da execução de um *script*, um cenário é um conjunto de máquinas (*containers*) destinadas a um usuário aluno para a execução de uma determinada atividade em laboratório. Um exemplo de cenário pode ser visualizado na Figura 7. Nela, o *script* cria: uma máquina Gateway com uma interface de rede NAT e outra interface de rede conectada à rede local, sem acesso à Internet; e duas máquinas com apenas uma interface sem acesso à rede local. O objetivo desse cenário é fazer com que os alunos configurem a máquina Gateway, de modo que ela seja o roteador da rede local, sendo responsável por encaminhar os pacotes das outras duas máquinas, a fim de que elas tenham acesso à Internet.

Figura 7 – Cenário proposto em sala



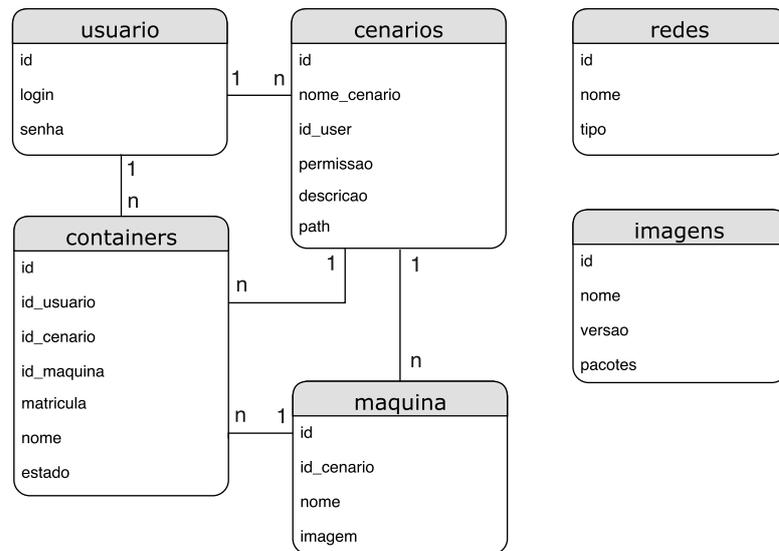
Fonte – Próprio autor (2017)

- **Interface Web:** Aplicação implementada utilizando a linguagem de programação PHP e executada em um servidor web Apache. Esta aplicação acessa o banco de dados, faz a leitura de dados armazenados no mesmo e executar comandos e *scripts* no servidor;
- **Banco de dados:** Onde se tem servidor um banco de dados *Mysql*. Na Figura 8, pode-se observar o diagrama com o esquema utilizado no banco de dados e nela as seguintes tabelas: (1) *usuarios*, que guarda informações de autenticação dos usuários professores à ferramenta, possuindo as seguintes colunas: identificador (*id*), *login* e senha do usuário professor. (2) *cenarios*, onde são armazenadas informações sobre os cenários criados por todos os usuários da ferramenta, como: identificador do cenário (*id*), nome do cenário (*nome_cenario*), identificador do usuário dono do cenário (*id_usuario*), a permissão do cenário (*permissao*), descrição do cenário (*descricao*) e o caminho (*path*) onde os *scripts* deste cenário estão armazenados no servidor. (3) *containers*, que é utilizada para guardar informações das máquinas de todos os cenários que estão em execução, onde temos as seguintes colunas: identificador da máquina (*id*), identificador do usuário que instanciou essa máquina (*id_usuario*), identificador do cenário a qual essa máquina pertence (*id_cenario*), a matrícula do aluno a quem essa máquina é destinada (*matricula*), nome da máquina (*nome*) e estado em que se encontra essa máquina (*estado*). (4) *maquina*, que é utilizada para armazenar informações sobre cada máquina de um cenário, como: identificador da máquina (*id*), identificador do cenário a qual ela pertence (*id_cenario*), o nome dado a essa máquina (*nome*) e a imagem que se é utilizada nesta máquina (*imagem*). (5) *imagens*, que é utilizada para guardar informações das imagens disponíveis no servidor para a criação de um cenário. Tal tabela contém as seguintes colunas: identificador da imagem (*id*), nome da imagem (*nome*), a versão da imagem (*versao*) e os pacotes que foram instalados nessa imagem (*pacotes*). (6) *redes*, que armazena informações das redes criadas pelos usuários professores e que estão disponíveis no servidor para serem utilizadas na construção de um cenário. Tal tabela tem as seguintes colunas: identificador da rede (*id*), nome da rede (*nome*) e seu tipo (*tipo*).

4.1.1 Acesso dos usuários alunos ao servidor e às máquinas

O acesso ao servidor por parte dos usuários alunos é realizado por meio de acesso remoto (SSH), sendo *login* e senha equivalentes a matrícula do aluno. Vale ressaltar que no primeiro acesso o usuário aluno é obrigado a alterar a senha, como o ilustrado na Figura 9.

Figura 8 – Esquema do banco de dados



Fonte – Próprio autor (2017)

Nela foi solicitado que o aluno, com a matrícula 380000, inserisse um *login* e senha, depois foi iniciado o processo de troca de senha, com o aluno repetindo sua senha, e logo após uma nova senha e a sua confirmação foi requisitada. Caso o processo tenha ocorrido com sucesso, será exigido que o aluno faça novamente a conexão ao servidor utilizando a nova senha.

Figura 9 – Tela de primeiro acesso ao servidor

```

walafi@walafi-PC:~$ ssh 380000@192.168.56.101
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.
ECDSA key fingerprint is SHA256:WwBZDT/1Dwy0ilmIUf41vIrPZZNLYNHU/1/6s7sFDM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.101' (ECDSA) to the list of known hosts.
380000@192.168.56.101's password:
You are required to change your password immediately (root enforced)
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-87-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

108 pacotes podem ser atualizados.
54 atualizações são atualizações de segurança.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

WARNING: Your password has expired.
You must change your password now and login again!
Mudando senha para 380000.
Senha UNIX (atual):
Digite a nova senha UNIX:
Redigite a nova senha UNIX:
passwd: password updated successfully
Connection to 192.168.56.101 closed.
  
```

Fonte – Próprio autor (2017)

Para cada máquina criada de um cenário, são inseridos *scripts* no diretório *home* de cada aluno, que o permite acessar a máquina, como o ilustrado na Figura 10. Nela, dois *scripts* relacionados ao cenário instanciado, estão disponíveis: um referente a uma máquina servidor e outro a uma máquina cliente. Para realizar o acesso a alguma máquina é necessário que o aluno execute o *script* da máquina desejada e a partir desta execução será pedido um *login* e senha de acesso. Nesse caso, *login* e senha foram configuradas para “*aluno*” e “*alunoufc*” respectivamente, sendo um padrão para todas as máquinas. Tal processo pode ser visualizado na Figura 11. Feito isso, o usuário aluno poderá desempenhar a prática de laboratório.

Figura 10 – *Scripts* criados no *home* do usuário aluno

```
380000@TCC-MV:~$ ls
Conecta1-380000-14-Servidor.sh  Conecta1-380000-15-Cliente.sh
380000@TCC-MV:~$ █
```

Fonte – Próprio autor (2017)

Figura 11 – Tela de acesso a máquina

```
1-380000-14-Servidor login: aluno
Password:
Welcome to Ubuntu 16.10 (GNU/Linux 4.4.0-87-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

aluno@1-380000-14-Servidor:~$ █
```

Fonte – Próprio autor (2017)

4.2 Funcionalidades da ferramenta

A Figura 12 apresenta as opções presentes na página inicial disponíveis ao professor após o processo de autenticação, entre elas estão: *Instanciar cenários*, *Gerenciar cenários instanciados* e *Criar cenário*, que são discutidas nas seções seguintes.

Figura 12 – Opções da ferramenta

Opções:

- Instanciar Cenário
- Gerenciar Cenários Instanciados
- Criar Cenário

Fonte – Próprio autor (2017)

4.2.1 Instanciar cenários

Nessa função da aplicação, é possível listar todos os cenários disponíveis ao usuário professor. A Figura 13 mostra a lista de cenários já disponíveis na aplicação para um usuário. Nela, algumas informações sobre os cenários são apresentadas, como: (1) *Nome do Cenário*, onde é exibido o nome de cada cenário; (2) *Permissão*, onde são mostradas as permissões de cada cenário, que pode ser público, significando que o cenário está disponível a todos que fazem uso da ferramenta, ou privado, permitindo o acesso a esse cenário a apenas o dono; (3) *Dono*, diz se o usuário atual é dono ou não do cenário; (4) *Descrição do Cenário*, onde é mostrada uma pequena descrição do cenário ou atividade desenvolvida; e por último, (5) *Situação*, que apresenta a situação atual do cenário, em uso ou disponível. Em caso de disponibilidade do cenário, é permitido ao usuário instanciar o mesmo clicando em qualquer lugar da linha referente a ele. Já em caso de uso do cenário, o usuário fica impedido de instanciar o mesmo cenário uma segunda vez afim de impossibilitar que o mesmo faça alterações no cenário em excursão.

Figura 13 – Cenários disponíveis

Cenários:

Nome do Cenário	Permissão	Dono	Descrição do Cenário	Situação
Cenário 01	Público	Sim	Cenário simples para prática em uma máquina	Disponível
Cenário 02	Público	Sim	Esse cenário contém três máquinas e o objetivo é configurar o encaminhamento de pacotes na máquina Gateway	Disponível
Cenário 03	Público	Sim	Cenário de treinamento para práticas com traceroute, tcpdump, compactação, ssh e scp	Disponível
Cenário 06	Público	Sim	Cenário para configuração de VPN	Disponível

Fonte – Próprio autor (2017)

Ao selecionar um cenário, o usuário é direcionado a uma nova página com

informações sobre o cenário selecionado como as apresentadas na Figura 14. Nela são exibidas informações, como: nome, descrição do cenário e o *script* de criação desse cenário. Ainda nessa página, três botões estão disponíveis: (1) *voltar*, onde o usuário pode utilizar para retornar à página anterior; (2) *Editar*, onde é possível fazer mudanças ou fazer uma cópia desse cenário. Essas duas funções são apresentadas apenas para usuário dono do cenário, para usuários não donos é possível apenas fazer uma cópia desse cenário para si, para poder editá-las. Vale ressaltar que ao clicar nesse botão o usuário é direcionado à página de criação de cenário, que será preenchida com dados do cenário antes selecionado. Tal funcionalidade dessa página será explicada na seção 4.2.3. (3) *Próximo*, onde o usuário pode clicar caso deseje prosseguir na instanciação do cenário e então o usuário é direcionado para a página de matrículas, com campos como o ilustrado na Figura 15. Nela, o usuário professor pode inserir um número referente à matrícula do aluno e adicioná-la clicando no botão *Add*. Nessa etapa, o usuário do aluno é criado caso o mesmo não conste no servidor, ou deletar clicando em *Del*, que faz com que a matrícula do aluno seja removida da lista. Por fim, o botão *Finalizar* inicia o processo de instanciação do cenário para cada matrícula referente ao aluno existe na lista e direciona o usuário professor para a página de gerenciamento de cenários instanciados, que será explicada na próxima seção.

Figura 14 – Dados do cenário



Fonte – Próprio autor (2017)

4.2.2 Gerenciar cenários instanciados

Após finalizar o processo de instanciação de um cenário, o usuário professor é direcionado para a página de Gerenciamento de cenários instanciados, que por sua vez também se faz acessível por meio da página principal na opção *Gerenciar cenários instanciados*. Essa página apresenta ao professor duas opções de listagem: (1) *Cenários*, como apresentado na

Figura 15 – Matrículas

Matrículas



Fonte – Próprio autor (2017)

Figura 16, exibe o nome, o estado em que se encontram as máquinas de um determinado cenário e ações disponíveis (botões: *Deletar*, *Stop*, *Start* e *Restart*); e (2) *Containers*, como apresentado na Figura 17, exibe mais detalhes sobre cada cenário, como: as matrículas de cada aluno, suas máquinas, o estado da mesma e ações que podem ser executados no cenário do aluno em questão (botões: *Deletar*, *Stop*, *Start* e *Restart*). Vale ressaltar que apenas nesta opção fica disponível ao professor a possibilidade de adicionar novos alunos em um cenário em execução, clicando no botão *Adicionar usuário*.

Figura 16 – Gerenciamento de cenários

Cenários		Containers
Nome do Cenário	Estado do Cenário	Ação
Cenário 01	Running: 2	Stop Deletar

Fonte – Próprio autor (2017)

Figura 17 – Gerenciamento de *containers*

Cenários		Containers
Cenário: Cenário 01		
Adicionar Aluno		
Aluno: 380011		
Nome do Container	Estado	
1-380011-1-Maquina	Running	
Stop Deletar		
Aluno: 380012		
Nome do Container	Estado	
1-380012-1-Maquina	Running	
Stop Deletar		

Fonte – Próprio autor (2017)

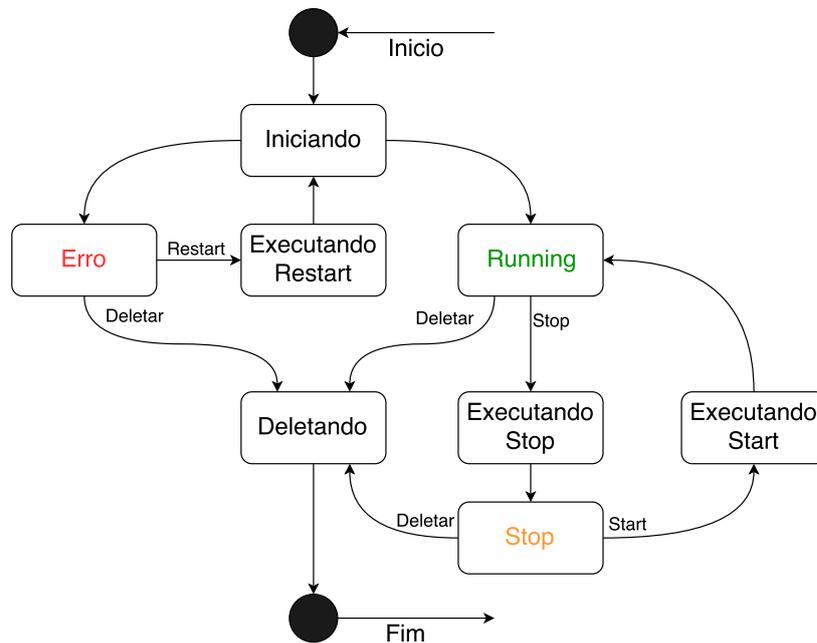
As ações disparadas pelos botões levam em conta o estado atual em que cada máquina de um determinado cenário se encontra. Para demonstrar os possíveis estados de uma máquina e os botões disponíveis, é apresentada a Figura 18. Nela, é apresentada um diagrama de estados e as ações que podem alterar cada estado. Inicialmente, o primeiro estado de uma máquina é (1) *Iniciando*. Nesse estado, as máquinas estão sendo instanciadas e nenhuma ação é disponível ao professor. Caso a inicialização seja mal sucedida, o estado da máquina muda para (2) *Erro*. Nesse estágio, as ações disponíveis para o usuário professor são: *Deletar* e *Restart*. Caso escolha *Deletar* as máquinas, o estado das mesmas passam para (3) *Deletando*, que deleta as máquinas escolhidas. Caso escolha *Restart*, as máquinas mudarão para o estado *Iniciando*. Caso a iniciação ocorra com sucesso, as máquinas passam para o estado (4) *Running*, que indica que as máquinas estão em execução e podem ser acessadas pelos usuários alunos. Nesse estado, as ações apresentadas ao professor são *Deletar* e *Stop*. Caso *Deletar* seja escolhida, as máquinas passam para o estado *Deletando* e são excluídas. Caso *Stop* seja escolhida, as máquinas passam para o estado (5) *Executando Stop*. Nessa etapa, nenhuma ação é disponível para o professor. Ao finalizar esse processo, as máquinas passam para o estado (6) *Stop*, que indica que as máquinas estão paradas e não podem ser acessadas pelos usuários alunos. Nessa etapa, duas ações estão disponíveis: *Deletar*, que coloca as máquinas no estado *Deletando* e as exclui, e *Start*, que põe as máquinas no estado (7) *Executando Start*, que por sua vez não possui nenhuma ação. Finalizado esse processo, as máquinas voltam para o estado *Running* e passam a está acessível para os alunos. Vale ressaltar que a qualquer momento uma máquina pode apresentar erro e passar para o estado de *Erro*.

4.2.3 Criar cenário

Buscando uma forma de possibilitar o usuário professor implementar seus próprios cenários para desenvolver laboratórios práticos, foi adicionado a opção *Criar cenário*. Uma visão inicial da página desta opção pode ser visualizada na Figura 19. Nela, algumas informações sobre o novo cenário devem ser obrigatoriamente preenchidas, como: (1) *Nome do Cenário*, que representa o nome que é dado ao cenário. (2) *Descrição do cenário*, nesse campo deve ser adicionada uma breve descrição do cenário ou atividade a ser desenvolvida. E por último, tem-se (3) *Privacidade*, que define a privacidade do cenário a ser criado, que por padrão é pré-selecionada como pública.

Feito todo o processo de preenchimento dos dados de um cenário, o usuário professor

Figura 18 – Máquina de estado



Fonte – Próprio autor (2017)

Figura 19 – Tela de criação de cenário

Criar Cenário

Fonte – Próprio autor (2017)

pode clicar em *Próximo* e a partir daí é apresentado ao mesmo uma nova tela, onde de fato o *script* de criação de um cenário será implementado, como a ilustrada na Figura 20. Nela, é apresentado um campo onde o usuário professor pode inserir suas linhas de código, como também é apresentado alguns botões com funções distintas, como:

- **Container:** Ao ser selecionada, essa opção abre uma tela sobre a página atual e somente através dela é possível adicionar ou remover uma máquina do cenário em criação. A Figura 21 ilustra essa tela. Nela, é exibida uma tabela com a lista de máquinas existentes no cenário, com as seguintes colunas: *Nome*, onde é exibido o nome referente a uma determinada máquina; *Imagem*, que contém o nome da imagem que está sendo utilizada por uma determinada máquina; e por fim, *ação*, que exhibe ao usuário professor um botão

Figura 20 – Tela de implementação de *script*

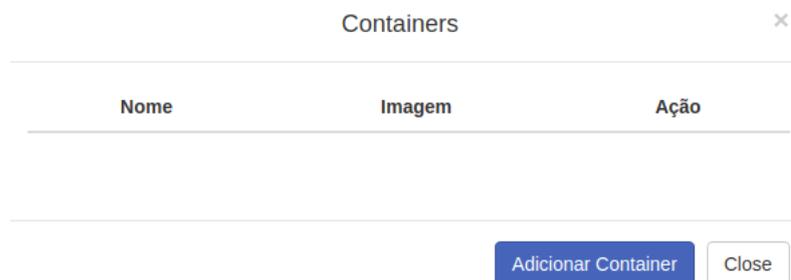
Criar Cenário



Fonte – Próprio autor (2017)

com o nome *deletar*, que possui a função de remover a máquina do cenário em construção. Outra funcionalidade importante é a possibilidade de inserir novas máquinas no cenário em desenvolvimento, através do botão *Adicionar Container*, que ao ser clicado apresenta um formulário para o seu preenchimento, como o exposto na Figura 22. Nele, informações sobre a máquina a ser criada é requisitada, como: *Nome*, um nome único que será atribuído a essa máquina; *Imagem*, imagem que será utilizada para a criação dessa máquina; *Rede Padrão*, que indica qual rede será utilizada por padrão nesta máquina; e *Permissão*, campo utilizado para definir privilégios do usuário aluno dentro da máquina. Feito o processo de preenchimento do formulário, fica então permitido adicionar essa máquina por meio do botão *Add*, ou então caso deseje limpar os dados isso poderá ser feito por meio do botão *Limpar*.

Figura 21 – Tela de máquinas de um novo cenário



Nome	Imagem	Ação

Fonte – Próprio autor (2017)

- **Redes:** Ao clicar nesse botão, uma nova tela é exibida ao usuário professor, e nela podem ser visualizadas as redes já criadas no sistema, como ilustrado na Figura 23. Na mesma tela, fica disponível a possibilidade de adicionar uma nova rede por meio do botão *Adicionar rede*, que ao ser clicado exibe um formulário como o apresentado na Figura 24. Nela, apenas o nome da nova rede é requerido. Vale ressaltar que as novas redes serão por padrão

Figura 22 – Formulário de novas máquinas

Fonte – Próprio autor (2017)

do tipo *bridge* e terão que ter nome único. Feito o preenchimento do campo nome com um nome único, torna-se possível adicionar a nova rede através do botão *Add*.

Figura 23 – Tela de redes disponíveis

Redes ×	
Nome	Tipo
bridge	bridge
host	host
none	null

Adicionar Rede
Close

Fonte – Próprio autor (2017)

Figura 24 – Formulário de novas redes

Fonte – Próprio autor (2017)

- **Voltar:** onde possibilita ao usuário professor voltar à página anterior de dados de um cenário.
- **Ajuda:** apresenta ao professor uma pequena ajuda de como criar seus cenários.
- **Salvar cenário:** Após toda implementação do *script* para a criação do cenário, deve-se utilizar o botão *Salvar cenário* para salvar o mesmo. Ao clicar neste botão, uma nova sobre tela aparece para o usuário professor com os dados finais do cenário como demonstrado

na Figura 25. Nela, informações sobre o cenário são reexibidas, como: nome, descrição, privacidade do cenário e uma tabela contendo os nomes e as imagens das máquinas adicionadas. Feito todo o processo de criação, o botão *Salvar* deve ser utilizado para de fato salvar o cenário. A partir daí, o usuário é direcionado para a tela de matrículas onde pode adicionar matrículas e instanciar o cenário para os alunos. Vale ressaltar que se não houver nenhuma máquina na lista de máquinas o usuário é impedido de salvar o cenário.

Figura 25 – Tela de dados do cenário



Dados do Cenário

Nome do Cenário:
teste

Descrição do Cenário:
este é um cenário de teste para o tcc

Privacidade:
 Público
 Privado

Nome	Imagem
Servidor	ubuntu:14.10

Salvar Close

Fonte – Próprio autor (2017)

4.3 Requisitos

Buscando demonstrar as tarefas e serviços oferecidos ou não pela ferramenta, foram definidos os requisitos funcionais e não funcionais da solução desenvolvida, que são:

- Requisitos funcionais:
 - a) O usuário professor deve poder instanciar um cenário;
 - b) O usuário professor deve poder parar, reexecutar, reiniciar ou deletar um cenário em execução;
 - c) O usuário professor deve poder criar seus próprios cenários;
 - d) O usuário professor dono do cenário deve poder editar ou fazer uma copia do cenário;

- e) O usuário professor não dono do cenário deve poder apenas fazer uma copia do cenário.
 - f) O usuário professor deve poder criar uma rede;
 - g) O usuário professor deve poder adicionar um usuário aluno a qualquer momento em um cenário em execução;
 - h) O usuário aluno deve poder acessar as máquinas quando o estado das mesmas for *Running*;
 - i) O usuário aluno não deve poder acessar as máquinas de um cenário quando o estado das mesmas forem *Stop*;
 - j) O usuário professor não deve poder instanciar um mesmo cenário que se encontra em uso.
- Requisitos não funcionais:
 - a) A ferramenta deve usar *Design* responsivo nas interfaces gráficas para que seja acessível em diferentes dispositivos;
 - b) A fim de garantir acesso seguro aos cenários, o usuário aluno deve utilizar um cliente SSH para acessar o servidor;
 - c) O usuário aluno deve mudar sua senha no primeiro acesso ao servidor por uma senha mais forte.

5 AVALIAÇÃO DA FERRAMENTA

Buscando uma forma de avaliar a ferramenta desenvolvida, foi proposto que professores e alunos respondessem um formulário com o objetivo de fazer uma avaliação quantitativa da ferramenta. Neste capítulo, é apresentada a metodologia utilizada para a realização dessa avaliação, bem como os resultados obtidos a partir da mesma e uma discussão geral sobre os resultados.

5.1 Metodologia de avaliação

Para a realização dos testes, a ferramenta, devidamente implementada, foi hospedada na Amazon¹ em uma máquina com 2 CPUs virtuais, 8 GB de memória ram e 40 GB de disco com sistema operacional *ubuntu 16.04.3 LTS*.

No primeiro momento, a ferramenta foi apresentada ao professor, mostrando todas as funcionalidades e funções disponíveis nela, e então foi pedido ao mesmo que realizasse um teste prévio criando um cenário que poderia ser utilizado em uma prática de laboratório. Então, com este mesmo cenário criado, foi solicitado ao professor que realizasse esta prática de laboratório com a turma de alunos.

Após o término da prática, foi aplicado um questionário para os alunos e outro para o professor, com o objetivo de realizar uma avaliação quantitativa da ferramenta. As perguntas aplicadas aos alunos e professores podem ser observadas na Tabela 1 e Tabela 2 respectivamente, bem como a justificativa de cada um delas. As alternativas para os formulários foram agrupadas em escalas de resposta diferentes, seguindo a seguinte estrutura: questões 1 e 2 do questionário aluno e questões de 1 a 6 do questionário do professor foram perguntados o nível de concordância (*concordo plenamente, concordo, discordo e discordo plenamente*); para a questão 7 do questionário professor com apenas dois itens, sim ou não; e para questão de 8 a 11 do mesmo questionário foram perguntados o nível de satisfação (*Muito satisfeito, Satisfeito, Nem satisfeito nem insatisfeito, insatisfeito e Muito insatisfeito*).

Vale ressaltar que perguntas subjetivas também foram feitas em ambos os formulários de maneira opcional, com intuito de saber as opiniões, dúvidas e dificuldades que todos os usuários tiveram ao utilizar a ferramenta.

¹ <https://aws.amazon.com/pt/>

Tabela 1 – Questionário aluno.

Pergunta 01	O acesso ao cenário (containers) é fácil.
Justificativa	Buscar saber que os alunos tiveram alguma dificuldade em acessar as máquinas do cenário.
Pergunta 02	É melhor utilizar a abordagem proposta do que usar máquinas virtuais do VirtualBox.
Justificativa	Buscar saber a aceitação do uso de <i>containers</i> utilizados na ferramenta.

Fonte – Produzido pelo autor

Tabela 2 – Questionário professor.

Pergunta 01	A organização da ferramenta propicia uma navegação dinâmica e fácil
Justificativa	Buscar saber se a organização da ferramenta está adequada para o desenvolvimento das atividades.
Pergunta 02	A ferramenta apresentada é um instrumento útil para a utilização em práticas de laboratório.
Justificativa	Saber se a ferramenta trará benefícios na realização de laboratórios práticos das disciplinas lecionadas por este professor.
Pergunta 03	As informações disponíveis na ferramenta são suficientes para a utilização da mesma.
Justificativa	Buscar saber se as informações disponíveis na ferramenta são suficientes para utilizar todas as funcionalidades da mesma.
Pergunta 04	A ferramenta dispõe de todas as funções necessárias para sua execução.
Justificativa	Saber se a ferramentas possui todas as funções implementadas para a excussão de laboratórios práticos.
Pergunta 05	A ferramenta é relevante para o campus.
Justificativa	Buscar saber se a ferramenta é uma contribuição direta para o ambiente universitário.
Pergunta 06	A construção de um cenário é fácil
Justificativa	Saber se a criação de um cenário é um fator critico sucesso da ferramenta em seu objetivo.
Pergunta 07	Você adotaria a ferramenta em alguma de suas disciplinas?
Justificativa	Saber se o professor incluiria a ferramenta desenvolvida como um instrumento que auxilie na curva de aprendizagem dos alunos.
Pergunta 08	De modo geral, o quão satisfeito você está com o tempo de criação de um cenário?
Justificativa	Saber se o professor está satisfeito com tempo em que se leva para criar um cenário.
Pergunta 09	De modo geral, o quão satisfeito você está com a instanciação de um cenário?
Justificativa	Saber se o professor está satisfeito com o tempo de execução dos <i>scripts</i> para se instanciar um cenário para uma determinada lista de alunos.
Pergunta 10	De modo geral, o quão satisfeito você está com o gerenciamento de cenários?
Justificativa	Buscar saber se as funcionalidades referentes ao gerenciamento dos cenário satisfizeram o professor.
Pergunta 11	Qual o seu nível de satisfação geral com a ferramenta?
Justificativa	Saber o quanto o conjunto da ferramenta satisfaz o professor.

Fonte – Produzido pelo autor

5.2 Resultados

Os questionários foram aplicados as turmas do curso de Redes de Computadores da Universidade Federal do Ceará, campus Quixadá: (1) turma do quarto semestre do curso - disciplina *Programação Script* e (2) turma do sexto semestre do curso - disciplina *Projeto Integrado de Redes de Computadores*, com vinte e um alunos e dois professores ao todo. Para a

Turma 1, o professor criou um cenário com duas máquinas, um cliente e um servidor, e propôs ao alunos que realizam a comunicação entre essas duas máquinas utilizando o NetCAT. Para a Turma 2, o professor optou por utilizar um cenário pronto, com uma máquina Gateway com uma interface de rede NAT e outra interface de rede conectada à rede local, sem acesso à Internet; e duas máquinas com apenas uma interface sem acesso à rede local. O objetivo desse cenário é fazer com que os alunos configurem a máquina Gateway, de modo que ela seja o roteador da rede local, sendo responsável por encaminhar os pacotes das outras duas máquinas, a fim de que elas tenham acesso à Internet. Os resultados do questionário dos alunos são apresentados na Tabela 3, e os resultados do questionário dos professores na Tabela 4.

Tabela 3 – Resultados: Questionário dos alunos

	Concordo plenamente	Pergunta 01		
		Concordo	Discordo	Discordo plenamente
Turma 1	12	1	0	0
Turma 2	6	2	0	0
Pergunta 02				
	Concordo plenamente	Concordo	Discordo	Discordo plenamente
Turma 1	9	3	1	0
Turma 2	7	1	0	0

Fonte – Produzido pelos autores

5.3 Discussão dos resultados

Analisando os resultados obtidos através da aplicação do questionário, observou-se o aluno, em sua maioria, avaliaram que o modo de acesso às máquinas é considerado fácil e que a solução baseada em *containers* é melhor do que utilizar hipervisores, como o VirtualBox.

Já os resultados colhidos por meio do questionário aplicados aos professores obtiveram resultados satisfatórios, como: (1) *Pergunta 1*: ambos os professores concordaram que a organização da ferramenta possibilita uma navegação dinâmica e fácil; (2) *Pergunta 2*: ambos os professores concordam plenamente que a ferramenta desenvolvida é um instrumento útil no desenvolvimento das práticas de laboratório, respondendo a umas das principais finalidades desta ferramenta; (3) *Pergunta 3*: um ponto que deixou a desejar foi o fato das informações disponíveis na ferramenta não serem suficientes para o desenvolvimento das atividades; (4) *Pergunta 4*: foi perguntado se a ferramenta possui todas as funções necessárias

para sua execução, e os professores responderam que concordam plenamente e concorda respectivamente, concluindo que as necessidades básicas são atendidas. (5) *Pergunta 5*: foi perguntado se a ferramenta é relevante para o campus e ambos professores concordaram plenamente quanto isso, implicando que esta ferramenta é uma contribuição direta para o meio acadêmico; (6) *Pergunta 6*: foi perguntado aos professores se a criação de um cenário é considerável fácil, o professor da Turma 1 discordou quanto a esse fato e o outro professor, da Turma 2, concordou quanto a isso, considerando então que a criação de um cenário não é tão fácil. (7) *Pergunta 7*: onde foi perguntado se os professores tem interesse em utilizar essa ferramenta em suas disciplinas e ambos os professores responderam que sim, levando a concluir que a ferramenta será útil. (8) *Pergunta 8*: foi perguntado o quanto satisfeitos os professores estão com o tempo de criação de um cenário e os mesmos responderam muito satisfeito e satisfeitos respectivamente, levando a crer que a criação de um cenário na ferramenta é muito rápido; (9) *Pergunta 9*: foi perguntado o quanto satisfeitos os professores estão com o tempo de instanciação de um cenário é rápido e eles responderam satisfeitos e muito satisfeito respectivamente, com isso acredita-se que a ferramenta seja eficiente nesse quesito; (10) *Pergunta 10*: foi perguntado o quanto satisfeito os professores estão com a funcionalidade de gerenciamento de cenário e os mesmos responderam satisfeitos e muito satisfeito respectivamente, levando a crer que a ferramenta cumpre todas as funcionalidades quanto a esse quesito; (11) *Pergunta 11*: foi perguntado, de modo geral, qual o nível de satisfação os professores tiveram da ferramenta, os mesmo responderam que estão satisfeitos, concluindo que a ferramenta é vista com bons olhos pelos professores e agradando os mesmo.

Vale ressaltar que os testes só puderam ser realizados em duas turmas por falta professores dispostos a testar a ferramenta com seus alunos.

Tabela 4 – Resultados: Questionário dos professores

		Pergunta 01				
		Concordo plenamente	Concordo	Discordo	Discordo plenamente	
Turma 1		0	1	0	0	
Turma 2		0	1	0	0	
		Pergunta 02				
		Concordo plenamente	Concordo	Discordo	Discordo plenamente	
Turma 1		1	0	0	0	
Turma 2		1	0	0	0	
		Pergunta 03				
		Concordo plenamente	Concordo	Discordo	Discordo plenamente	
Turma 1		0	0	1	0	
Turma 2		0	0	1	0	
		Pergunta 04				
		Concordo plenamente	Concordo	Discordo	Discordo plenamente	
Turma 1		1	0	0	0	
Turma 2		0	1	0	0	
		Pergunta 05				
		Concordo plenamente	Concordo	Discordo	Discordo plenamente	
Turma 1		1	0	0	0	
Turma 2		1	0	0	0	
		Pergunta 06				
		Concordo plenamente	Concordo	Discordo	Discordo plenamente	
Turma 1		0	1	0	0	
Turma 2		0	0	1	0	
		Pergunta 07				
		Sim	Não			
Turma 1		1	0			
Turma 2		1	0			
		Pergunta 08				
		Muito satisfeito	Satisfeito	Nem satisfeito nem insatisfeito	Insatisfeito	Muito insatisfeito
Turma 1		1	0	0	0	0
Turma 2		0	1	0	0	0
		Pergunta 09				
		Muito satisfeito	Satisfeito	Nem satisfeito nem insatisfeito	Insatisfeito	Muito insatisfeito
Turma 1		0	1	0	0	0
Turma 2		1	0	0	0	0
		Pergunta 10				
		Muito satisfeito	Satisfeito	Nem satisfeito nem insatisfeito	Insatisfeito	Muito insatisfeito
Turma 1		0	1	0	0	0
Turma 2		1	0	0	0	0
		Pergunta 11				
		Muito satisfeito	Satisfeito	Nem satisfeito nem insatisfeito	Insatisfeito	Muito insatisfeito
Turma 1		0	1	0	0	0
Turma 2		0	1	0	0	0

Fonte – Produzido pelos autores

6 CONCLUSÃO

Este trabalho teve por objetivo o desenvolvimento de uma ferramenta que auxilia aos professores dos cursos da área de TI nas práticas de laboratório utilizando virtualização baseada em *container*. Para isso, primeiro foi feito um levantamento das soluções de virtualização existente e mostrado o melhor desempenho apresentado na virtualização baseada em *containers* relacionada a virtualização baseada em *hypervisor*. Logo após, foram observadas as práticas realizadas pelos professores em laboratório e elaborado uma lista com as funcionalidades necessárias para o desenvolvimento de tais práticas para assim iniciar o processo de implementação da solução. O código da ferramenta desenvolvida será disponibilizado para a comunidade no repositório no *GitHub*¹.

A partir dos resultados obtidos por meio da avaliação quantitativa feita sobre a ferramenta, percebe-se que a mesma obteve resultados satisfatórios, sendo bem avaliada tanto por alunos, quanto por professores, alcançando o objetivo geral deste trabalho. Vale ressaltar que os testes só podaram ser realizados em duas turmas por falta de professores dispostos a testar a ferramenta desenvolvida com suas turmas.

Apesar dos testes terem ocorrido com sucesso, a ferramenta apresenta limitações que podem interferir nas práticas de laboratório, como: a falta de uma opção para o professor criar suas próprias imagens, ou a falta de uma saída de erro dos *scripts* de criação de cenário para quando ocorrer algum erro.

Como trabalhos futuros, fica a sugestão da realização de um *upgrade* na ferramenta, adicionando funcionalidades que irão ajudar ao professor na realização de um laboratório prático. Outra proposta é o uso de fato da ferramenta em alguma disciplina e verificar o quanto ela será útil no decorrer da disciplina.

¹ <https://github.com/Walafi02/TCC2Walafi>

REFERÊNCIAS

- ANISSETTI, M.; BELLANDI, V.; COLOMBO, A.; CREMONINI, M.; DAMIANI, E.; FRATI, F.; HOUNSOU, J. T.; REBECCANI, D. Learning computer networking on open paravirtual laboratories. **IEEE Transactions on Education**, IEEE, v. 50, n. 4, p. 302–311, 2007.
- AZAB, A. Enabling docker containers for high-performance and many-task computing. In: IEEE. **Cloud Engineering (IC2E), 2017 IEEE International Conference on**. [S.l.], 2017. p. 279–285.
- BERTOCHI, L. O.; BELLEZI, M. A. Virtualização de desktops em grandes ambientes. **Revista TIS**, v. 4, n. 2, 2016.
- CARISSIMI, A. Virtualização: da teoria a soluções. **Minicursos do Simpósio Brasileiro de Redes de Computadores–SBRC**, v. 2008, p. 173–207, 2008.
- DIAS, A. P.; FRENEDOZO, R. de C. **O uso da ferramenta fórum de discussão no ensino de ciências**. [S.l.: S.n], 2015.
- FERREIRA, K. H.; LIMA, R. W. de; CHAVES, J. O. M.; LIMA, M. de. Larc-virtual laboratory for practice development in the discipline of computer networks integrated with moodle. **International Journal of Recent Contributions from Engineering, Science & IT (iJES)**, v. 2, n. 1, p. 17–21, 2014.
- GARCIA, L.; ANTUNES, F.; LARA, D.; RIBEIRO, C. P. Utilização de ambientes virtualizados para ensino de servidores de redes de computadores. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2016. v. 27, n. 1, p. 90.
- JERSAK, L. C. **Mapeamento de máquinas virtuais em datacenters privados visando minimizar a interferência de desempenho**. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul, 2014.
- KUMAR, S.; GANKOTIYA, A. K.; DUTTA, K. A comparative study of moodle with other e-learning systems. In: IEEE. **Electronics Computer Technology (ICECT), 2011 3rd International Conference on**. [S.l.], 2011. v. 5, p. 414–418.
- MANU, A.; PATEL, J. K.; AKHTAR, S.; AGRAWAL, V.; MURTHY, K. B. S. Docker container security via heuristics-based multilateral security-conceptual and pragmatic study. In: IEEE. **Circuit, Power and Computing Technologies (ICCPCT), 2016 International Conference on**. [S.l.], 2016. p. 1–14.
- MARTÍNEZ-TORRES, M. R.; MARÍN, S. T.; GARCIA, F. B.; VAZQUEZ, S. G.; OLIVA, M. A.; TORRES, T. A technological acceptance of e-learning tools used in practical and laboratory teaching, according to the european higher education area 1. **Behaviour & Information Technology**, Taylor & Francis, v. 27, n. 6, p. 495–505, 2008.
- MAURICIO, L. A. F. **Avaliação de desempenho de plataformas de virtualização de redes**. Rio de Janeiro, RJ, Brasil, 2013. 78 p. Disponível em: <http://www.pel.uerj.br/bancodissertacoes/Dissertacao_Leopoldo_Mauricio.pdf>. Acesso em: 21 jun. 2017.

MORABITO, R. Virtualization on internet of things edge devices with container technologies: a performance evaluation. **IEEE Access**, IEEE, 2017.

MORABITO, R.; KJÄLLMAN, J.; KOMU, M. Hypervisors vs. lightweight virtualization: a performance comparison. In: IEEE. **Cloud Engineering (IC2E), 2015 IEEE International Conference on**. [S.l.], 2015. p. 386–393.

SEO, C. E. Virtualização—problemas e desafios. **IBM Linux Technology Center**, 2009.

TRYUS, Y.; KACHALA, T. **Cloud technologies in management and educational process of ukrainian technical universities**. [S.l.: S.n], p. 022–033, 2014.

TURNBULL, J. **The Docker Book: containerization is the new virtualization**. [S.l.]: James Turnbull, 2014.

WANNOUS, M.; NAKANO, H. Nvlab, a networking virtual web-based laboratory that implements virtualization and virtual network computing technologies. **IEEE Transactions on Learning Technologies**, IEEE, v. 3, n. 2, p. 129–138, 2010.

XAVIER, M. G.; NEVES, M. V.; ROSSI, F. D.; FERRETO, T. C.; LANGE, T.; ROSE, C. A. D. Performance evaluation of container-based virtualization for high performance computing environments. In: IEEE. **Parallel, Distributed and Network-Based Processing (PDP), 2013 21st Euromicro International Conference on**. [S.l.], 2013. p. 233–240.

XU, L.; HUANG, D.; TSAI, W.-T. Cloud-based virtual laboratory for network security education. **IEEE Transactions on Education**, IEEE, v. 57, n. 3, p. 145–150, 2014.

ZHANG, J.; LU, X.; PANDA, D. K. Performance characterization of hypervisor- and container-based virtualization for hpc on sr-ioV enabled infiniband clusters. In: IEEE. **Parallel and Distributed Processing Symposium Workshops, 2016 IEEE International**. [S.l.], 2016. p. 1777–1784.

ZHUANG, Z.; TRAN, C.; WENG, J.; RAMACHANDRA, H.; SRIDHARAN, B. Taming memory related performance pitfalls in linux cgroups. In: IEEE. **Computing, Networking and Communications (ICNC), 2017 International Conference on**. [S.l.], 2017. p. 531–535.