

# DVL: Uma Ferramenta para Criação de Laboratórios Práticos de Disciplinas da Área de TI Utilizando Virtualização Baseada em Contêineres

Walafi F. da Silva<sup>1</sup>, Paulo A. L. Rego<sup>1</sup>, João M. U. de Alencar<sup>1</sup>,  
Antonio R. Braga<sup>1</sup> e Bruno G. Mateus<sup>1</sup>

<sup>1</sup>Curso de Redes de Computadores  
Universidade Federal do Ceará (UFC) - Campus Quixadá

walafi@alu.ufc.br, {pauloalr, joao.marcelo, rafaelbraga, brunomateus}@ufc.br

**Abstract.** *Practice labs in Information Technology (IT) classes are necessary because they enable students to practice the contents studied. However, many educational institutions do not have a budget for buying dedicated equipment to use in hands-on activities. Virtualization allows the execution of multiple guest systems on a host machine, reducing the physical infrastructure required. In this context, this work presents the DVL tool, which automates the creation of hands-on labs using Docker containers to quickly and easily create, execute and manage scenarios.*

**Resumo.** *As práticas de laboratório em disciplinas da área de Tecnologia da Informação (TI) são importantes pois possibilitam ao aluno colocar em prática os conteúdos estudados. Entretanto, muitas instituições de ensino não possuem orçamento para uma estrutura física dedicada para atividades práticas. A virtualização permite a execução de vários sistemas convidados sobre um hospedeiro, reduzindo a infraestrutura física necessária. Nesse contexto, este trabalho apresenta a ferramenta DVL (Docker Virtual Lab), que automatiza a criação de ambientes de estudo utilizando contêineres Docker para criar, executar e gerenciar cenários de forma rápida e acessível.*

## 1. Introdução

A virtualização ocupa posição de destaque na área de TI, sendo utilizada em servidores para um melhor aproveitamento dos recursos computacionais [Garcia et al. 2016]. Porém, a virtualização não se limita a facilitar a gerência do parque computacional empresarial, pois também é útil na manutenção de sistemas operacionais com configurações distintas coabitando estações de trabalho para usuários finais, ou na instanciação de ambientes para testes e simulações na academia. Entre as diversas soluções de virtualização existentes, as mais utilizadas são: hipervisores, que funcionam como uma camada que abstrai o *hardware* das máquinas virtuais, resultando em sobrecarga em termos de *hardware* e *drivers*; e a virtualização leve ou baseada em contêineres, que oferece um nível diferente de abstração e isolamento, apresentado uma menor sobrecarga se comparada aos hipervisores [Morabito et al. 2015].

As atividades de laboratório desempenham um papel importante em cursos e treinamentos da área de TI, pois não só ajudam aos alunos na compreensão dos

tópicos apresentados, mas também proporcionam habilidades de resolução de problemas e experiências necessárias para lidar com situações do mundo profissional [Wannous and Nakano 2010]. 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 devem simular eventos reais e permitir que os discentes vivenciem um amplo leque de experiências relacionadas ao mundo profissional e acadêmico. Entretanto, manter equipamentos dedicados para cada aluno apresenta alto custo financeiro, tanto com aquisição, quanto com manutenção e consumo energético.

A virtualização surge como uma alternativa para que instituições de ensino possam viabilizar práticas de laboratório e ensino [Carissimi 2008]. Em algumas disciplinas, há a necessidade de executar diversos sistemas operacionais ao mesmo tempo e, na maioria dos casos, é comum o aluno precisar de acesso administrativo ao sistema. Utilizando um ambiente virtualizado, é possível instanciar diversos sistemas e executar todos os comandos sem qualquer tipo de restrição, pois as ações de um usuário administrador não afetam o sistema nativo.

Este artigo apresenta a ferramenta DVL (*Docker Virtual Lab*), uma solução voltada a professores e alunos, que permite gerenciar laboratórios virtuais (chamados de cenários) de disciplinas da área de TI e automatizar a criação desses cenários, utilizando contêineres Docker. Com ela, os professores podem criar atividades práticas, com cenários compostos de várias máquinas, e instanciar rapidamente várias cópias do cenário, uma para cada aluno.

O restante deste artigo está organizado como segue: a Seção 2 apresenta alguns trabalhos relacionados, a Seção 3 apresenta a ferramenta e detalha as opções arquiteturais utilizadas no desenvolvimento da mesma. Já a Seção 4 descreve a demonstração planejada e apresenta os endereços para acesso ao código fonte, instruções para implantação e teste da ferramenta. Por fim, a Seção 5 conclui o artigo e apresenta os trabalhos futuros.

## 2. Trabalhos Relacionados

Técnicas de virtualização têm sido utilizadas em vários trabalhos para criar ambientes de estudo e treinamento. Em [Garcia et al. 2016], os autores defendem a criação de ambientes virtualizados para práticas de laboratório em redes de computadores. Para validar o estudo, foram elaborados experimentos utilizando o hipervisor VirtualBox<sup>1</sup> para ensinar os alunos a proverem diferentes serviços, como servidores *Web* e de *Proxy*. [Xu et al. 2014] apresenta uma plataforma de treinamento de laboratório virtual chamada V-Lab, na qual é possível criar ambientes para práticas utilizando os hipervisores Xen ou KVM. O V-Lab fornece uma interface *Web* para o gerenciamento dos recursos.

Diferente dos trabalhos mencionados, nossa solução utiliza virtualização baseada em contêineres para criar ambientes de estudo. Além disso, também disponibilizamos uma interface *Web* para facilitar o gerenciamento e instanciação dos cenários. A escolha da tecnologia de contêineres pode ser justificada com os resultados apresentados em [Morabito et al. 2015], que fez uma comparação detalhada do desempenho da virtualização baseada em hipervisores tradicionais e virtualização baseada em

---

<sup>1</sup><https://www.virtualbox.org/>

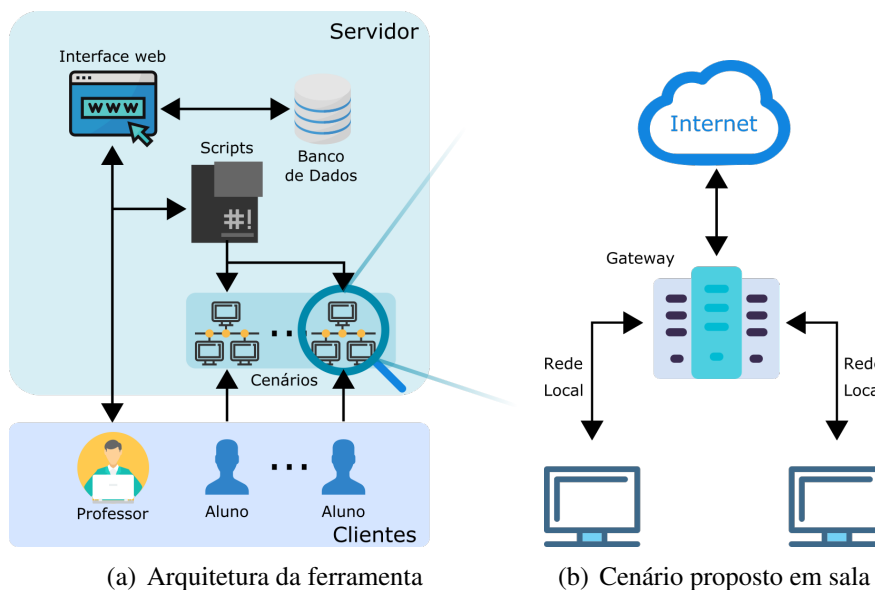
contêineres. Os autores utilizaram uma série de *benchmarks* para avaliar as ferramentas KVM, OSv, LXC e Docker, e mostraram que a virtualização baseada em contêineres tem um desempenho superior a outros tipos de virtualização, além de possibilitar a criação de um maior número de instâncias na mesma máquina física, devido ao menor tamanho das imagens quando comparadas com as de hipervisores.

### 3. DVL (Docker Virtual Lab)

Nesta seção, são apresentados os principais componentes e a arquitetura da ferramenta, as funcionalidades implementadas, bem como os requisitos funcionais e não funcionais da solução proposta.

#### 3.1. Arquitetura e componentes

A Figura 1(a) apresenta uma visão geral da arquitetura da ferramenta. Nela, é possível observar que o usuário professor tem acesso a uma interface web, onde é possível criar *scripts* para a instanciação de um cenário ou instanciar cenário a partir dos *scripts* já existentes. Uma vez criado o cenário, os usuários alunos podem acessar o servidor e acessar os contêineres do 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
```

(c) Script que define o cenário

**Figura 1. Aspectos da ferramenta**

Os principais componentes da ferramenta são: os *Scripts* que automatizam a instanciação de cenários, os *Cenários*, a Interface Web e o Banco de Dados. Os *Scripts* foram implementados utilizando a linguagem *Shell Script* e têm o papel de criar os

contêineres que fazem parte de um cenário, para um ou mais alunos. Um exemplo de *script* pode ser observado na Figura 1(c).

Um **Cenário** é um conjunto de contêineres destinados a um usuário aluno, para a execução de uma determinada atividade prática. Um exemplo de cenário pode ser visualizado na Figura 1(b). Nela, um contêiner Gateway é criado com uma interface de rede NAT e outra interface de rede conectada à rede local, sem acesso à Internet; e dois contêineres com apenas uma interface são conectados à rede local. Nessa atividade, os alunos devem configurar o Gateway, de modo que ele seja o roteador da rede local, permitindo o encaminhamento de pacotes dos outros dois contêineres, a fim de que eles também tenham acesso à Internet.

A **Interface Web** foi implementada utilizando a linguagem de programação PHP e executa em um servidor *Web Apache*. Ela é responsável por executar os *scripts* no servidor. E, por fim, o **Banco de dados MySQL** foi utilizado. Na Figura 2, pode-se observar o esquema utilizado no banco de dados e nele as seguintes tabelas: (1) *usuarios*, que guarda informações de autenticação dos usuários professores à ferramenta; (2) *cenarios*, onde são armazenadas informações sobre os cenários criados por todos os usuários da ferramenta; (3) *containers*, que é utilizada para guardar informações dos contêineres de todos os cenários que estão em execução; (4) *maquinas*, que é uma abstração utilizada para armazenar informações sobre cada contêiner de um cenário; (5) *imagens*, que armazena informações das imagens disponíveis no servidor para a criação de um cenário; (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.

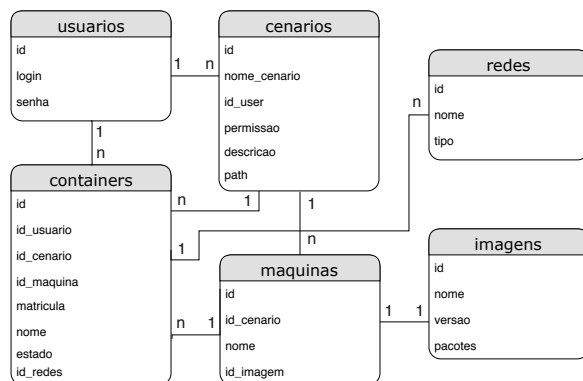


Figura 2. Esquema do banco de dados

### 3.1.1. Acesso dos usuários alunos ao servidor e aos contêineres

Os alunos podem acessar o servidor por meio de acesso remoto (SSH), sendo o nome de usuário e senha equivalentes ao número de matrícula do aluno. Vale ressaltar que ao realizar o primeiro acesso, o usuário aluno é forçado a alterar a senha.

Para cada contêiner de um cenário instanciado, são inseridos *scripts* no diretório *home* de cada aluno, permitindo assim, o acesso ao contêiner, como o ilustrado na Figura 3(a). Conforme mostra a figura, dois *scripts* relacionados ao cenário instanciado estão disponíveis: um referente a um contêiner servidor e outro a um contêiner cliente. Para

acessar um contêiner, o aluno deve executar o *script* correspondente ao contêiner desejado, e então lhe será solicitado nome de usuário e senha de acesso. Tal processo pode ser visualizado na Figura 3(b).

(a) *Scripts* criados no *home* do usuário aluno

(b) Tela de acesso ao contêiner

**Figura 3. Acesso do usuário via terminal**

### 3.2. Funcionalidades da ferramenta

Na página inicial do professor, aparecem três opções: *Instanciar cenários*, *Gerenciar cenários instanciados* e *Criar cenário*, que são discutidas nas seções seguintes.

#### 3.2.1. Instanciar cenários

Essa função lista todos os cenários disponíveis ao usuário professor, como mostra a Figura 4. Nela, pode-se observar os cenários e as colunas (1) *Nome do Cenário*, (2) *Permissão*, onde são mostradas as permissões de cada cenário (Público - cenário disponível a todos os professores cadastrados na ferramenta; ou Privado - apenas o dono pode acessar o cenário; (3) *Dono*, diz se o usuário atual é dono ou não do cenário; (4) *Descrição do Cenário*, e por último, (5) *Situação*, que apresenta a situação atual do cenário, em uso ou disponível. Em caso de disponibilidade, o professor pode instanciar ou modificar o cenário clicando em qualquer lugar da linha do cenário desejado. Contudo, se o cenário já estiver em uso, ele não pode ser modificado, apenas instanciado para novos alunos.

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

**Figura 4. Cenários disponíveis**

O usuário pode acessar as informações de um cenário, como mostra a Figura 5(a), ao selecioná-lo. Ainda nessa página, três botões estão disponíveis: *Voltar*, *Editar* e *Próximo*. Ao clicar em *Editar*, é possível fazer alterações ou uma cópia do cenário. As alterações podem ser feitas apenas pelo usuário dono do cenário, enquanto qualquer usuário pode fazer uma cópia do cenário. A página de edição é similar à página de criação de cenário - mais detalhes serão apresentados na Seção 3.2.3. Ao clicar em *Próximo*, o usuário segue o fluxo de instanciação de um cenário, sendo então direcionado à página de matrículas (Figura 5(b)). Nessa tela, o professor pode adicionar os números das matrículas dos alunos clicando no botão *Add*, enquanto o botão *Del* permite remover um aluno. O botão *Finalizar* inicia o processo de instanciação do cenário e redireciona o professor para a página de gerenciamento de cenários instanciados.

Nome: Cenário 01

Descrição: Cenário simples para prática em uma máquina

```
sudo docker run --privileged -itd -h $Maquina --name $Maquina ubuntu:1.0
```

Voltar Editar Próximo

(a) Dados do cenário

Nº da Matrícula do Aluno

Add Del

Finalizar

(b) Matrículas

Figura 5. Tela de cenários e de matrículas

### 3.2.2. Gerenciar cenários instanciados

Após instanciar um cenário, o professor é direcionado para a página *Gerenciar cenários instanciados*. Nela, duas opções de listagem estão disponíveis: (1) *Cenários* (Figura 6(a)), que exibe o nome, o estado em que se encontram as máquinas de um cenário e ações disponíveis (*Deletar*, *Stop*, *Start* e *Restart*); e (2) *Contêineres* (Figura 6(b)), que exibe mais detalhes de cada cenário, como: as matrículas dos aluno, suas máquinas, o estado da mesma e ações que podem ser executadas no cenário do aluno em questão (*Deletar*, *Stop*, *Start* e *Restart*). Apenas nesta opção fica disponível ao professor a possibilidade de adicionar novos alunos em um cenário em execução, clicando no botão *Adicionar aluno*.

Cenários		Contêineres	
Cenários		Contêineres	
Nome do Cenário	Estado do Cenário	Ação	
Cenário 1	Running: 1	Stop	Deletar

Cenários		Contêineres	
Cenário: Cenário 1		Adicionar Aluno	
Aluno: 380011			
Nome do Contêiner	Estado		
1-380011-1-Maquina	Running	Stop	Deletar

(a) Gerenciamento de cenários

(b) Gerenciamento de contêineres

Figura 6. Gerenciamento de cenário instanciado

As ações *Deletar*, *Stop*, *Start* e *Restart* levam em conta o estado atual de cada contêiner de um determinado cenário. A Figura 7 apresenta um diagrama para demonstrar os possíveis estados de um contêiner e os botões disponíveis. O primeiro estado é (1) *Iniciando*. Nesse estado, os contêineres estão sendo instanciados e nenhuma ação é disponível ao professor. Caso a inicialização seja mal sucedida, o estado do contêiner muda para (2) *Erro*. Nesse estágio, as ações disponíveis para o professor são: *Deletar* e *Restart*. Caso escolha *Deletar* os contêineres, o estado dos mesmos são alterados para (3) *Deletando*, que deleta os contêineres escolhidas. Caso a ação *Restart* seja selecionada, os contêineres mudarão para o estado *Iniciando*.

Caso a inicialização ocorra com sucesso, os contêineres passam para o estado (4) *Running*, onde estão em execução e podem ser acessados pelos alunos. Nesse estado, as ações apresentadas ao professor são *Deletar* e *Stop*. Caso a ação de *Deletar* seja escolhida, os contêineres passam para o estado *Deletando* e são excluídos. Caso *Stop* seja escolhida, os contêineres passam para o estado (5) *Executando Stop*, onde nenhuma ação

está disponível. Ao finalizar esse processo, os contêineres passam para o estado (6) *Stop*, que indica que eles estão parados e não podem ser acessados pelos alunos. Nessa etapa, duas ações estão disponíveis: *Deletar*, que coloca os contêineres no estado *Deletando* e os exclui, e *Start*, que põe muda os contêineres para o estado (7) *Executando Start*, que por sua vez não permite qualquer ação. Finalizado esse processo, os contêineres voltam para o estado *Running* e ficam acessíveis aos alunos. Vale ressaltar que, em caso de erro, em qualquer momento um contêiner pode passar para o estado *Erro*.

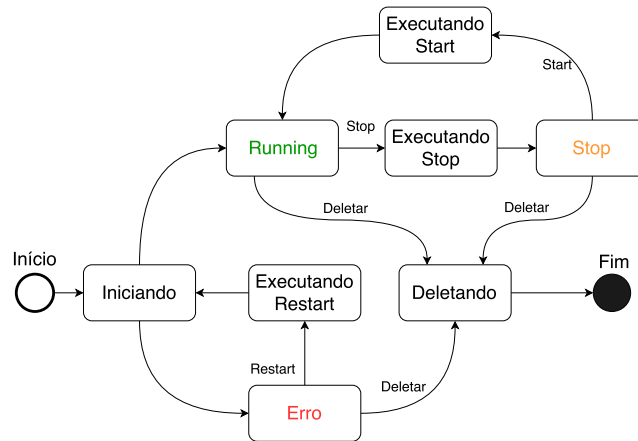


Figura 7. Diagrama com os possíveis estados dos contêineres

### 3.2.3. Criar cenário

A opção *Criar cenário* auxilia o professor a implementar cenários. Para isso, algumas informações sobre o novo cenário devem ser informadas, como mostra a Figura 8(a). Essas informações são: o *Nome do Cenário*, *Descrição do cenário* e *Privacidade*. Após o preenchimento das informações, o professor é direcionado à tela onde o *script* de criação de um cenário será implementado, como mostra a Figura 8(b). Nessa tela, o professor pode inserir suas linhas de código no campo reservado para este fim.

A tela de criação de cenário, intitulada "Dados do Cenário", contém os seguintes campos e controles:

- Um campo de texto para "Nome do Cenário".
- Um campo de texto para "Descrição do Cenário".
- Controles de privacidade: "Privacidade:" com opções "Público" (selecionada) e "Privado".
- Botões "Próximo" (verde) e "Limpar".

(a) Tela de criação de cenário

A tela de implementação de script, intitulada "Script", contém:

- Botões "Container" e "Redes" no topo.
- Um campo de texto grande para a inserção do script.
- Botões "Voltar", "Ajuda" e "Salvar Cenário" na base.

(b) Tela de implementação de *script*

Figura 8. Telas de configuração do cenário

Durante a criação do cenário, o professor pode listar todos os contêineres existentes no cenário. Por meio dessa funcionalidade, ele terá acesso a uma tabela com as seguintes informações: *Nome* (nome do contêiner); *Imagem* (nome da imagem que está

sendo utilizada por um determinado contêiner). Além disso, também é possível visualizar as redes já existentes no cenário e adicionar novas redes.

Após a configuração do cenário, o professor deve salvar o cenário. Durante essa etapa, é exibida uma tela para a conferência das informações. Após salvar o cenário, o professor é direcionado para a tela de matrículas, onde pode adicionar matrículas e instanciar o cenário para os alunos.

#### 4. Demonstração

O código fonte e um guia de instalação da ferramenta estão disponíveis para a comunidade no Github: <https://github.com/Walafi02/DVL>. Além disso, foram preparados dois vídeos: um com o guia de instalação (<https://youtu.be/eRcBf8hG98w>) e outro para apresentar as funcionalidades da ferramenta (<https://youtu.be/um97ebPY8Io>).

Durante o evento, será feita uma demonstração das funcionalidades da ferramenta, onde será criado um cenário com três contêineres: um Cliente, um Firewall e um Servidor Web. O objetivo do cenário é o aluno praticar a instalação de um servidor HTTP e configuração de um firewall Iptables, para que o Cliente possa acessar o Servidor Web.

#### 5. Conclusão

As práticas de laboratório em disciplinas na área de TI são importantes para prover experiência prática aos alunos. Como nem sempre existem equipamentos reais disponíveis para as práticas didáticas, este trabalho apresentou uma ferramenta para permitir a criação de ambientes de estudo virtuais utilizando a tecnologia de contêineres Docker.

A DVL gerencia a criação de cenários e automatiza a instanciação de contêineres para vários alunos, criando todos os recursos necessários para a execução das atividades de estudo. Como trabalhos futuros, pretende-se fazer uma avaliação de usabilidade e melhorar a interface da ferramenta, principalmente para facilitar a criação de cenários mais complexos.

#### Referências

- Carissimi, A. (2008). Virtualização: da teoria a soluções. *Minicursos do Simpósio Brasileiro de Redes de Computadores–SBRC*, 2008:173–207.
- Garcia, L., Antunes, F., Lara, D., and Ribeiro, C. P. (2016). Utilização de ambientes virtualizados para ensino de servidores de redes de computadores. In *Brazilian Symposium on Computers in Education (SBIE)*, volume 27, page 90.
- Morabito, R., Kjällman, J., and Komu, M. (2015). Hypervisors vs. lightweight virtualization: a performance comparison. In *Cloud Engineering (IC2E), 2015 IEEE International Conference on*, pages 386–393. IEEE.
- Wannous, M. and Nakano, H. (2010). Nvlab, a networking virtual web-based laboratory that implements virtualization and virtual network computing technologies. *IEEE Transactions on Learning Technologies*, 3(2):129–138.
- Xu, L., Huang, D., and Tsai, W.-T. (2014). Cloud-based virtual laboratory for network security education. *IEEE Transactions on Education*, 57(3):145–150.