

Gerência e Alocação de Aulas Virtualizadas em Laboratórios de Informática: Uma Arquitetura Híbrida

Rodrigo de Brito Lira¹ , Thiago Gouveia¹ , Paulo Ditarso Maciel Jr.¹ 

¹Programa de Pós-Graduação em Tecnologia da Informação (PPGTI)
Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB)
Campus João Pessoa – Paraíba – Brazil

`lira.rodrigo@academico.ifpb.br`

`{thiago.gouveia, paulo.maciel}@ifpb.edu.br`

Abstract. *A gerência eficiente de laboratórios de informática nem sempre é uma realidade em instituições de pesquisa e ensino. A configuração de equipamentos, muitas vezes realizadas de forma manual, é uma tarefa custosa e suscetível a erros que podem inviabilizar o uso de tais laboratórios. Se por um lado é economicamente inviável depender de ferramentas comerciais para gerência destes ambientes, cujas soluções utilizam software e hardware proprietários; por outro lado é interessante fazer uso de equipamentos legados através de soluções abertas de virtualização. Este artigo propõe uma arquitetura híbrida para a realização de aulas virtualizadas em laboratórios de informática, onde a gerência é centralizada e a execução é distribuída em recursos existentes. Resultados preliminares indicam a viabilidade da solução proposta.*

Resumo. *The efficient management of computer labs is not always a reality in research and teaching institutions. The configuration of equipment, often performed manually, is a costly task and susceptible to errors that can make the use of such labs unfeasible. If, on the one hand, it is economically unfeasible to depend on commercial tools to manage these environments, whose solutions use proprietary software and hardware; on the other hand, it is interesting to make use of legacy equipment through open virtualization solutions. This article proposes a hybrid architecture for conducting virtualized classes in computer labs, where management is centralized and execution is distributed across existing resources. Preliminary results indicate the viability of the proposed solution.*

1. Introdução

O uso de tecnologias no cotidiano de trabalho das pessoas acelera o processo de transformação digital que muito se espera, principalmente em ambientes educacionais e de pesquisa. Esta realidade já está presente em diversas instituições de ensino, sobretudo na Rede Federal de Educação Profissional, Científica e Tecnológica¹ (RFEPCT), composta pelos mais de 600 Campi dos Institutos Federais espalhados em todo território nacional. Com um perfil de ensino profissionalizante dedicado à formação técnica de estudantes para o mercado de trabalho, encontra-se em tais instituições uma numerosa quantidade de laboratórios de aulas e pesquisa, que demandam uma gerência sazonal e repetitiva de seus equipamentos. Entende-se que o mal gerenciamento destes recursos ocasiona desperdício de dinheiro público. Além disso, percebe-se que uma utilização

¹ <http://portal.mec.gov.br/rede-federal-inicial/>

mais efetiva de computadores e demais dispositivos poderia alavancar um processo de transformação digital da maneira como os conteúdos são ministrados, como por exemplo em [Moreno et al. 2019], e compartilhados entre professores, estudantes ou mesmo instituições, como abordado em [Sampaio et al. 2015]. Contudo, administrar esses recursos de maneira eficaz, extraindo o máximo possível de sua utilização, não é uma atividade trivial e geralmente exige uma capacitação profissional da equipe responsável.

A grande variedade de ferramentas voltadas à administração de ambientes interligados através de redes de computadores traz grande complexidade à formação de um profissional com habilidades nesta área de conhecimento. O uso de soluções de gerência que empregam a virtualização de recursos buscam facilitar essa tarefa, porém demandam um alto custo financeiro e, geralmente, necessitam de *hardware* específico para o funcionamento. Estas soluções são conhecidas como *Virtual Desktop Infrastructure* (VDI) e baseiam-se em hipervisores de virtualização em um modelo de servidor remoto que hospeda praticamente toda a “computação” necessária. Os serviços são usualmente acessados via terminais simples, com quase nenhum poder computacional. Entretanto, esta não é a realidade das instituições de ensino brasileiras, uma vez que já existe um parque computacional cuja capacidade não pode ser desperdiçada. Se por um lado é economicamente inviável comprar soluções prontas que necessitam de *software* e *hardware* extras, por outro lado é indispensável utilizar os recursos legados da maneira mais eficiente possível, até mesmo como uma contrapartida aos impostos pagos pela população.

Neste contexto, esta pesquisa propõe uma arquitetura híbrida para gerenciar laboratórios de ensino e pesquisa em instituições de educação técnica e profissionalizante, denominada **vCLASS** (**virtual Classroom AS a Service**). Esta arquitetura é composta pela integração de soluções abertas (*open source*), caracterizada pela gerência centralizada dos recursos dos laboratórios e pela execução distribuída das aulas virtualizadas, nos dispositivos finais (estações de trabalho). Desta forma, além de orquestrar o uso dos laboratórios mais efetivamente, será possível utilizar os recursos legados de *hardware* de maneira mais produtiva. Em contrapartida às soluções proprietárias e centralizadas, que geralmente utilizam recursos de *hardware* e *software* próprios. Para isso, é necessário definir quais tecnologias abertas podem ser adaptadas para a **vCLASS** e como integrar de maneira eficiente as diferentes soluções encontradas. Dentre outros aspectos importantes desta pesquisa, destacam-se os testes de desempenho das ferramentas de virtualização, o desenvolvimento de uma solução centralizada de gerência dos laboratórios, a caracterização das aulas que serão ministradas, compartilhamento de conteúdos/soluções, “remotização” (uso remoto) dos laboratórios, dentre outros. Uma prova de conceito da arquitetura foi instanciada com o objetivo de validar as atividades de gerência em um laboratório de aula, além de comprovar que a solução customizada apresenta um desempenho adequado. Resultados preliminares indicam um desempenho de virtualização melhor que uma solução não-customizada e um tempo de configuração dos laboratórios satisfatoriamente reduzido, devido à automação das atividades necessárias.

Este documento está organizado em seis seções. A Seção 2 defini o problema estudado considerando as etapas realizadas para virtualização das aulas. A Seção 3 descreve os trabalhos relacionados encontrados na literatura. A arquitetura **vCLASS** é detalhada na Seção 4 e sua prova de conceito está descrita na Seção 5. Por fim, a Seção 6 apresenta as considerações finais.

2. Definição do Problema

Esta seção apresenta uma visão geral dos desafios da gerência de aulas virtualizadas em laboratórios de informática, com especial atenção ao cenário vivenciado na RFEPCT. O objetivo é estruturar o problema de modo a identificar pessoas e setores envolvidos, tarefas que devem ser realizadas, recursos de *hardware* e possíveis métricas capazes de indicar a qualidade das soluções encontradas.

O conjunto de ações necessário para uma boa condução das aulas virtualizadas em laboratórios de informática pode ser dividido em quatro etapas, como ilustrado na Figura 1. Essas etapas são realizadas em dois momentos distintos: o de preparação, no qual são tomadas todas as medidas para garantir a efetividade das aulas; e o período letivo, que corresponde à própria execução das práticas nos laboratórios. A Etapa 1, a ser realizada pelo(s) docente(s) encarregado(s) pelo componente curricular (aula), corresponde à preparação das práticas. Nesta fase, devem ser definidos os requisitos para a realização de cada exercício, tais como o sistema operacional, os aplicativos que precisam ser instalados, e suas respectivas configurações. Levando em consideração os requisitos definidos na etapa anterior, a Etapa 2 representa a alocação dos laboratórios às disciplinas, devendo ser realizada pelos responsáveis por gerenciar o ambiente físico da instituição. Na Etapa 3, a equipe de suporte é responsável por preparar os laboratórios e computadores para atender às demandas identificadas previamente. Por fim, a Etapa 4 é conduzida pelo professor e envolve os estudantes na realização das práticas propriamente ditas.

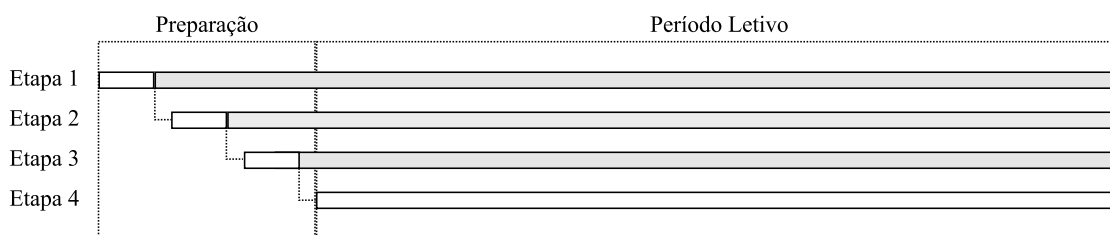


Figura 1: Etapas gerais da realização de aulas virtualizadas em laboratórios de informática.

Vale observar que as Etapas 1, 2 e 3 podem ser realizadas em duas ocasiões distintas, indicadas na Figura 1 pelas áreas em branco e cinza. O tempo em branco representa o momento natural para a realização das tarefas associadas à referida etapa. Em relação à Etapa 1, o professor pode querer adicionar, remover ou alterar alguma prática, de acordo com o *feedback* recebido das outras etapas. Problemas físicos ou de *hardware* também podem exigir a realocação dos laboratórios na Etapa 2, enquanto que a manutenção preventiva, o monitoramento, e as alterações realizadas nas Etapas 1 e 2 podem requerer ações adicionais na Etapa 3. A realização de tarefas em um momento posterior ao naturalmente esperado, representado pela área em cinza, pode ocasionar interrupções ou mesmo o impedimento de uso do laboratório.

Ressalta-se que cada etapa possui os seus próprios desafios e definir métricas para avaliar o seu sucesso não é uma tarefa simples. O bom andamento da Etapa 1 pode ser aferido pelo conjunto de ferramentas disponíveis, pela facilidade para preparação dos laboratórios e pela homogeneidade entre o ambiente de preparação e a sala de aula. Uma boa métrica para a Etapa 2 é o *hardware* necessário para alocação de todas as disciplina

e a flexibilidade para fazer alteração nas alocações. Por sua vez, os tempos de provisionamento e manutenção são bons indicativos de sucesso na Etapa 3. Por fim, o desempenho dos laboratórios virtuais se sobressai como a métrica mais importante da Etapa 4. Adicionalmente, a qualidade da experiência dos usuários pode ser medida através de questionários e entrevistas, oferecendo um indicador que permeia todas as etapas.

Em um cenário comum nos laboratórios da RFEPCT, *estações de trabalho* com razoável poder computacional possuem sistemas **Windows** e **Linux** em modo *dual boot*, capazes de executar o sistema hospedeiro e, quando necessário, máquinas virtuais para um ambiente mais “customizável” durante a aula. Antes do início do período letivo, os professores indicam quais aplicativos devem ser instalados em tais estações, que são formatadas e preparadas pelo setor de manutenção de acordo com a alocação dos laboratórios, que leva em conta apenas os requisitos de *hardware*. Estas ações comumente se repetem após o final de cada período letivo, geralmente realizadas de maneira manual e suscetível a erros, o que torna o processo custoso e lento. Finalmente, a realização das práticas requer que os estudantes instalem as máquinas virtuais ou baixem-nas pré-instaladas.

3. Trabalhos Relacionados

Esta seção apresenta, de forma não extensiva, um conjunto de trabalhos recentes relacionados à automatização do provisionamento de laboratórios para fins educacionais. Para referências mais antigas, o artigo [Krassmann et al. 2014] apresentam um mapeamento sistemático com foco na identificação de abordagens e metodologias de avaliação deste tipo de implementação. Foram encontradas 1.165 publicações, catorze sendo selecionadas para extração de dados. Os autores concluíram que a área ainda é pouco explorada.

No trabalho [Rangavittala et al. 2015], os autores apresentam um *framework multi-tenant* para provisionamento de serviços na nuvem com foco em ambientes educacionais construído em código aberto. Segundo os autores, a arquitetura *multi-tenant* é a característica chave do *framework*, que foi capaz de lidar com todas as demandas educacionais levantadas e suportar vinte usuários simultaneamente. Por sua vez, os autores em [Rubio et al. 2016] avaliam a aplicabilidade de ferramentas de gerenciamento de configuração para automatizar a criação de exercícios práticos no campo da cibersegurança. Utilizando as ferramentas **Puppet**² e **Packer**³, foram desenvolvidos modelos que geram cenários de teste com serviços vulneráveis e medidas de segurança implementadas. Foram avaliadas a flexibilidade da solução e a economia de tempo alcançada. O estudo conclui que o uso de ferramentas DevOps é uma opção viável tanto para o provisionamento de desafios de ensino em pequena escala, quanto para eventos de cibersegurança em grande escala.

Os autores em [Shukhman et al. 2017] descrevem o CERD (*Cloud Educational Resource Datacenter*), um centro de dados de recursos educacionais em nuvem baseado no modelo DaaS (*Desktop-as-a-Service*), como uma maneira de fornecer, para instituições educacionais, acesso remoto economicamente viável a *softwares* pagos. O trabalho propõe que a implementação real do CERD seja baseada na plataforma **OpenNebula**⁴. Um *framework* para gerência eficiente e em larga escala de laboratórios universitários, com o mínimo de administradores locais, é introduzido em [Masek et al. 2018]. O *framework* foi proposto com forte uso do **Ansible**⁵, uma ferramenta que simplifica tarefas

² <https://puppet.com/> ³ <https://packer.io/> ⁴ <https://opennebula.io/> ⁵ <https://ansible.com/>

complexas de orquestração e gerenciamento de configuração. Para tornar as funcionalidades mais acessíveis, foi criada uma aplicação *web*. O *framework* foi testado extensivamente na Universidade de Tecnologia de Brno, na República Tcheca, em laboratórios universitários selecionados.

Por sua vez, o trabalho [Huang 2019] discute como as tecnologias de virtualização podem fornecer exercícios práticos, reduzir amplamente a necessidade de recursos físicos, melhorar a eficácia do ensino e fornecer oportunidades únicas de avaliação que não estariam disponíveis com o uso apenas de computadores físicos. Na mesma linha, em [Rot et al. 2019], os autores descrevem o potencial da virtualização e suas aplicações em instituições de ensino superior, destacando a otimização da infraestrutura de TI e a redução dos custos de operação e manutenção relacionados. É apresentado um estudo de caso na Universidade de Economia de Wrocław, na qual foi implementada uma solução virtualizada com mais de 400 terminais em VDI. Foram utilizados seis servidores com dois processadores cada, totalizando 1.7 TB de memória RAM.

Recentemente, os autores do artigo [Genkov and Slavov 2021] descrevem uma abordagem de VDI utilizada para exercícios de laboratório em cursos de Computação e Engenharia de *software* na Universidade Técnica de Gabrovo. O sistema foi construído e implementado durante a pandemia para oferecer ensino a distância, possibilitando a utilização de vários sistemas operacionais simultaneamente, recuperação rápida de danos ao sistema e *backup* simplificado. O trabalho [Xiong et al. 2021] analisa e compara quatro das principais tecnologias de *desktop* na nuvem da atualidade: RDS (*Remote Desktop Service*), VDI, IDV (*Intelligent Desktop Virtualization*) e VOI (*Virtual OS Infrastructure*). Os autores propõem um esquema híbrido integrando VOI e VDI, provendo uma referência para a implementação em universidades. Na dissertação [Martins 2022], o autor aborda a implementação de sistemas de TI em instituições de ensino superior, destacando as limitações dos paradigmas tradicionais e os benefícios do uso de Computação em Nuvem e IaC (*Infrastructure-as-Code*) para simplificar a configuração dos sistemas, automatizar processos e reduzir custos. O trabalho propõe ainda uma arquitetura de nuvem privada usando **Openstack**⁶ e **Ansible**.

4. Arquitetura vCLASS

4.1. Módulos

A arquitetura híbrida proposta neste trabalho é composta por diversos módulos que interagem para o provimento dos serviços e que podem ser visualizados a partir de quatro seções: *Managers*, *Engine*, *Cache* e *Labs*. A Figura 2 ilustra como estes módulos estão dispostos na arquitetura e como é realizada a comunicação entre eles. A seção denominada *Managers* apresenta uma visão geral do sistema aos seus usuários, que podem ser administradores ou professores, através do **Módulo de Gerência (MGer)**. A seção *Engine* contempla cinco módulos da arquitetura híbrida, são eles: **Módulo de Imagens (MImg)**, **Módulo de Infraestrutura Física (MInfra)**, **Módulo de Configuração Virtual (Mvirt)**, **Módulo de Monitoramento (MMon)** e **Módulo de Autenticação e Perfis (MUser)**. A seção *Cache* é composta por um único componente, que é o **Módulo de Cache (MCache)**, replicado em pontos estratégicos para agilizar a transferência de imagens, quando necessário. Por fim, a seção *Labs* contém as **estações de trabalho** da arquitetura

⁶ <https://openstack.org/>

híbrida, onde serão executadas as aulas virtualizadas. Tais estações estarão habilitadas para opção de inicialização via rede e consistem de um Sistema Operacional (SO) customizado para a virtualização das aulas, com um componente de *cache* previamente configurado.

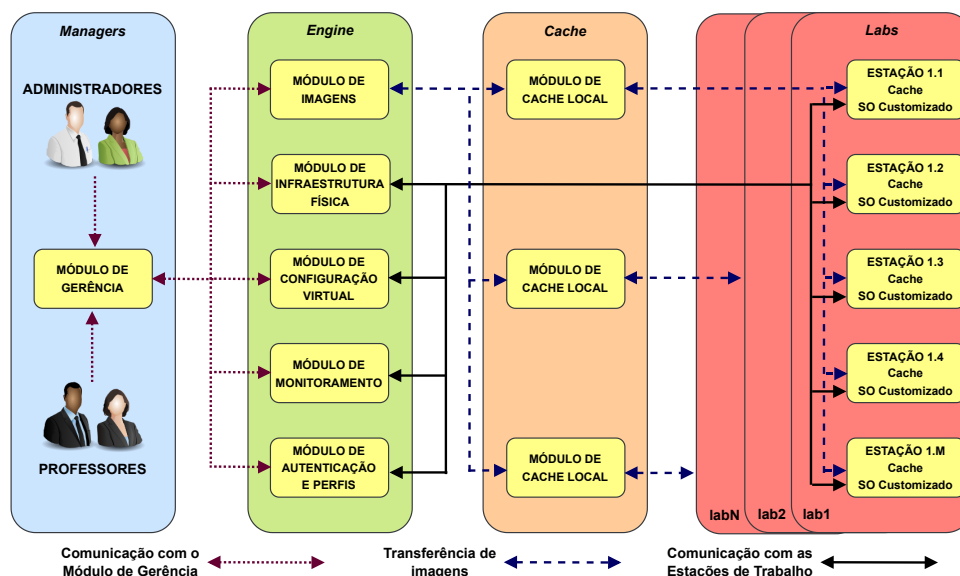


Figura 2: Arquitetura vCLASS.

A interação dos usuários com o módulo **MGer** é realizada por meio de uma interface Web e a comunicação com os demais módulos é implementada com uso de APIs (p.ex. via gRPC ou REST). A interação dos professores com este módulo ocorre estritamente para a elaboração ou atualização de imagens de suas respectivas aulas. Para tanto, poderão ser disponibilizados um serviço completo de criação de imagens via interface Web ou mesmo um formulário de requisitos a ser preenchido pelos professores menos familiarizados com ferramentas de virtualização. O objetivo é representar de forma simples suas requisições para as aulas virtualizadas, seja via formulário ou sistema Web.

O **MImg** disponibiliza um serviço para armazenar e distribuir as imagens primárias das máquinas virtuais, aquelas que são configuradas ou atualizadas pelos professores e definidas como a versão mais atual de suas aulas. Vale salientar que estas imagens são basicamente uma instanciação de um SO com os requisitos de *software* necessários para a aula. Apesar de não conter dados sensíveis, sugere-se algum mecanismo de criptografia no envio das imagens, pois previne a falsificação das mesmas. O **MInfra** será responsável por realizar a configuração *bare-metal* das estações de trabalho, através da distribuição e atualização do SO customizado para virtualização das aulas. Este último prover o ambiente necessário à execução das aulas virtualizadas, além de pré-configurar um componente de *cache* no dispositivo final. Além disso, deve ter compatibilidade com o **MVirt**, uma vez que este último é responsável por criar a infraestrutura virtual definida pelo **MGer**. Como requisitos funcionais de uma solução para o **MVirt**, espera-se o uso de uma linguagem de fácil entendimento e usabilidade, sem a dependência de agentes de execução e cuja comunicação seja criptografada. O **MMon** coleta informações de utilização das estações de trabalho com o intuito de verificar a “saúde” do *hardware* e exibe as informações através do **MGer**. Ao monitorar os recursos computacionais, como processador, memória, capacidade em disco e tráfego de rede, será possível agre-

gar informações para tentar antecipar problemas que inutilizem as estações de trabalho e, conseqüentemente, prejudiquem as aulas. O **MUser** é responsável por centralizar as configurações de usuários para acesso às estações e personalização de perfis. Para tanto, deverá interagir com soluções tradicionais de rede como **LDAP**, **CIFS** ou **NFS**. Através deste módulo será possível, por exemplo, mapear unidades de armazenamento em rede para os arquivos dos usuários ou estabelecer quotas e privilégios de maneira centralizada.

Como mencionado anteriormente, o **MCache** tem o intuito de agilizar a distribuição das imagens armazenando-as mais próximas das estações de trabalho, seja para uma implantação inicial do laboratório ou atualizações do mesmo. Na “ponta” da arquitetura híbrida, temos as estações de trabalho que executarão as devidas aulas virtualizadas. Um componente local do **MCache** terá a atribuição de verificar se a cópia local das máquinas virtuais são as mais recentes, ou se será necessário uma atualização.

4.2. Fluxos de Trabalho

Como intuito de esclarecer as interações e execuções de um sistema a partir da arquitetura híbrida proposta, alguns fluxos de trabalho são descritos nesta subseção.

A Figura 3 apresenta o fluxo de trabalho para a criação da imagem da aula virtualizada. Este fluxo está relacionado com a Etapa 1 descrita na Seção 2, na qual um professor é responsável por preparar a aula que será ministrada de maneira virtualizada. Ao interagir com o **MGer**, os professores podem verificar a existência de uma imagem adequada já disponibilizada, sem a necessidade de instalações personalizadas. Por outro lado, é possível especificar os requisitos de *software* necessários à aula, para que o **MImg** possa preparar a imagem desejada e salvar sua versão para distribuição. Em ambos os casos uma cópia da imagem pode ser distribuída para um ou vários módulos **MCache**.

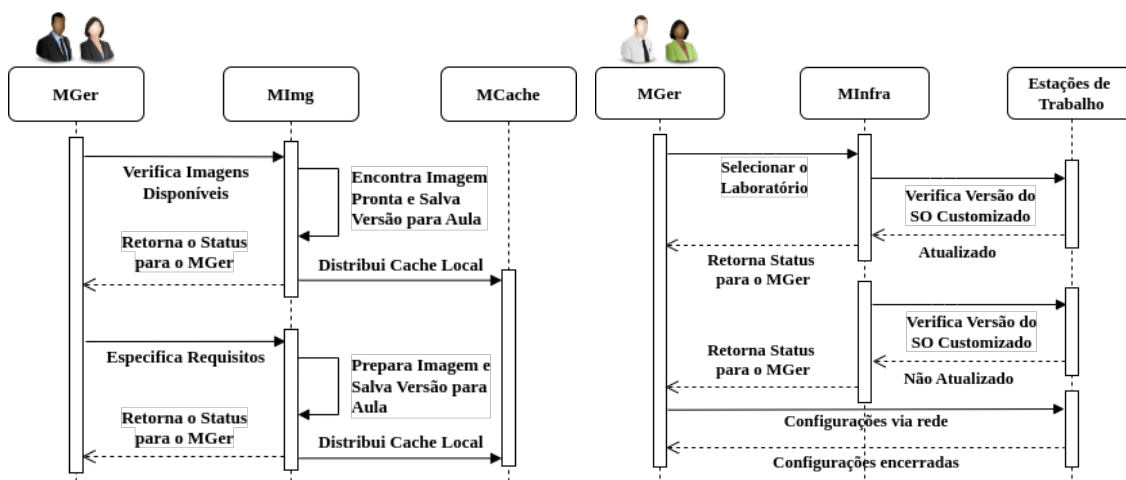


Figura 3: Criação de imagens.

Figura 4: Configuração *bare-metal*.

A Figura 4 apresenta um fluxo de trabalho relacionado à configuração da infraestrutura física, através da instalação dos SOs customizados via **MInfra**. Tal atividade é parte da Etapa 2 descrita na Seção 2, onde os administradores necessitam preparar o ambiente físico para execução das aulas e, quando necessário (p.ex. atualização do SO customizado), re-instalar o SO nos equipamentos. É importante observar que a **vCLASS** visa mitigar a necessidade de re-instalações físicas, uma vez que as aulas serão realizadas

de forma virtual. No fluxo em questão, Administradores utilizam o **MGer** para iniciar o processo e selecionar em qual laboratório será realizada a configuração *bare-metal*. Por sua vez, o **MInfra** é acionado para verificar se o SO customizado presente nas estações do referido laboratório está em sua versão mais atual ou não. Caso exista nova versão, o **MGer** comanda a configuração *bare-metal* do laboratório que consiste em formatar e instalar a nova versão do SO.

A Figura 5 demonstra o fluxo para instalação das aulas virtualizadas nas estações de trabalho. Caso exista, o ambiente será inicializado a partir de uma VM previamente configurada. Caso contrário, um conjunto de instruções será executado para realizar as configurações necessárias ao funcionamento do ambiente de aula. As fases deste fluxo consistem em: selecionar o laboratório específico por parte dos administradores; verificar se as versões das imagens estão atualizadas; e iniciar a configuração do ambiente virtual, onde pode ser preciso fazer o *download* de uma versão mais atual da aula virtualizada. As ações realizadas neste fluxo estão situadas na Etapa 3 definida na Seção 2.

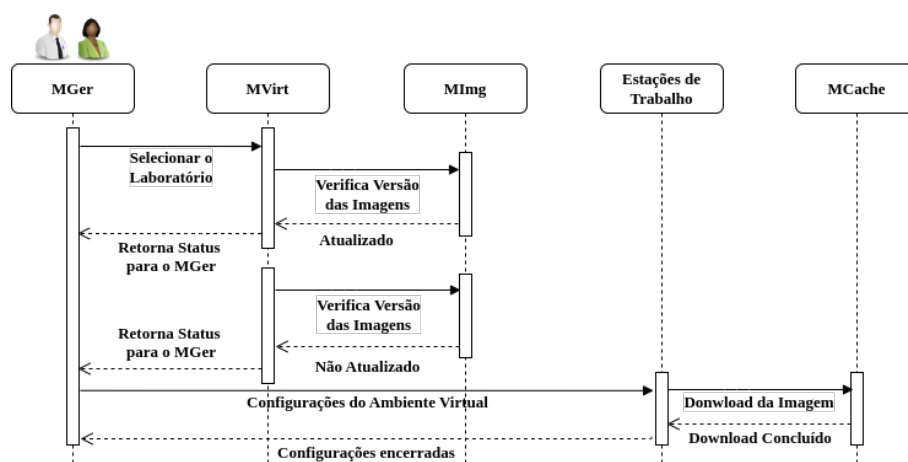


Figura 5: Fluxo de configuração do ambiente virtual.

A representação dos fluxos descritos acima é uma versão resumida de todas as atividades necessárias para tais configurações, onde detalhes dos processos e das interações com os demais módulos foram omitidos para efeito de simplificação.

5. Prova de Conceito

5.1. Descrição do Ambiente

Nesta subseção é descrito o ambiente de *software* e *hardware* utilizado para validar a prova de conceito (PoC - *Proof-of-Concept*) da arquitetura **vCLASS**. Mais especificamente, alguns módulos da arquitetura foram instanciados com ferramentas consolidadas em ambientes de IaC, com os quais foi possível validar o processo de configuração do ambiente virtual, bem como analisar o desempenho da solução customizada de virtualização.

Em um primeiro momento, buscou-se reproduzir um ambiente de configuração das aulas virtualizadas nas estações de trabalho. O módulo de armazenamento e transferência de imagens **MImg** foi instanciado como um servidor Web simples, com o uso do **Apache HTTP Server**⁷. Já o módulo **MVirt**, responsável por configurar o ambiente virtual, foi instanciado a partir da ferramenta **Ansible**. A solução **AWX**⁸ foi

⁷ <https://httpd.apache.org/> ⁸ <https://github.com/ansible/awx>

utilizada para representar as ações de configuração do **MGer** a partir de uma interface Web. Por fim, o SO customizado para as estações de trabalho consiste de um **Debian GNU/Linux** versão 11⁹, com uma seleção reduzida de pacotes, uma interface gráfica leve¹⁰ e personalizada¹¹, além do hipervisor **KVM**¹² (*Kernel-based Virtual Machine*).

A PoC foi configurada a partir de dois experimentos, descritos nas próximas subseções. O primeiro busca demonstrar a viabilidade da arquitetura proposta em configurar o ambiente de aulas virtuais em um tempo satisfatório. O segundo experimento visa apresentar uma análise de desempenho do SO customizado, com o objetivo de compará-lo com uma versão não customizada e com o ambiente físico. O intuito é demonstrar o quanto é possível melhorar em termos de desempenho em customizações futuras.

5.2. Configuração de Laboratório

Como mencionado anteriormente, o primeiro experimento visa comprovar a viabilidade das transferências de aulas virtualizadas no processo de preparação do laboratório para um semestre letivo. Ou seja, parte do processo de configuração do ambiente virtual representado pelo fluxo da Figura 5 e relacionado à Etapa 3 descrita na Seção 2.

Para efeito de simplificação, o ambiente utilizado neste experimento foi todo virtualizado. Foram utilizadas ao todo dezoito máquinas virtuais, quatro para as estações de trabalho, três por estação para representar as aulas virtualizadas, uma para o **MImg** e outra para o **MGer/MVirt**. Foram utilizadas as soluções **Apache HTTP Server**, **AWX** e **Ansible** para o **MImg**, o **MGer** e o **MVirt** respectivamente. Nas estações foi instalada uma versão customizada do SO **Debian GNU/Linux**. O experimento consiste em estimar o tempo necessário para transferir aulas virtualizadas, previamente preparadas e armazenadas no **MImg**, para as estações de trabalho. Por ser um experimento simplificado e virtualizado, optou-se por não instanciar o **MCache** e transferir as imagens diretamente do **MImg**. Para representar as aulas virtualizadas, foram utilizadas as versões “*vanilla*” do **Debian Desktop 11**, **Ubuntu Desktop 22.04** e **Windows 10**. A Tabela 1 apresenta os detalhes das configurações e resultados obtidos.

	Descrição	Tamanho	Tempo (min)
Aula 1	Debian Desktop 11	4,9G	01:18
Aula 2	Ubuntu Desktop 22.04	9,3G	02:52
Aula 3	Windows 10	11,0G	05:40
Total			09:50

Tabela 1: Tempo de transferência e inicialização das aulas virtualizadas.

Como pode ser observado no experimento, o tempo de transferência e inicialização das três aulas virtualizadas para execução em uma estação de trabalho foi de 9m50s. Simultaneamente, foi possível configurar as aulas nas quatro estações de trabalho disponíveis em aproximadamente 10 minutos. Como indicado, este tempo contabiliza não apenas a transferência entre o **MImg** e as estações de trabalho, mas também a inicialização de cada VM. Dessa forma, as aulas estarão prontas para uso, sem a necessidade da primeira (e geralmente mais demorada) execução ser realizada pelos estudantes.

⁹ <https://debian.org/> ¹⁰ <https://xlunch.org/> ¹¹ <http://openbox.org/> ¹² <https://linux-kvm.org/>

Os resultados obtidos neste experimento simplificado não objetivam retratar cenários realísticos, uma vez que vários fatores são desconsiderados, p.ex. a influência do tráfego em uma rede real. Contudo, acredita-se que este tempo é um indício satisfatório da viabilidade da arquitetura **vCLASS** e, certamente, representa um tempo muito inferior se comparado à atividade manual de instalar esses três SOs em uma máquina física do laboratório. Conjectura-se, portanto, que estas mesmas tarefas realizadas manualmente podem levar horas, ou mesmo dias, para finalizar um laboratório. A vantagem oferecida pela arquitetura **vCLASS** é automatizar todo esse processo de forma centralizada e ainda utilizar os recursos distribuídos de forma eficiente. No entanto, uma análise detalhada se faz necessária para demonstrar o uso da arquitetura completa em um ambiente empírico, inclusive utilizando o **MCache** para mitigar os efeitos do tráfego em rede.

5.3. Teste de Desempenho

Como uma forma de verificar o desempenho do SO customizado, foram realizados testes de *benchmark* com a suíte de *software* **OpenBenchmarking**¹³. O principal objetivo é comparar o desempenho em relação a uma versão não customizada do ambiente de virtualização, usando como parâmetro o desempenho dos testes em uma máquina física. O **OpenBenchmarking** fornece uma plataforma aberta de testes, onde foram selecionados com base na popularidade e interoperabilidade (utilização em **Windows** e **Linux**). Para verificar o desempenho de processamento (CPU) foi utilizado o **Blender**¹⁴ (versão 3.2.0), que estima o tempo de renderização de uma imagem a partir de uma modelagem 3D. Adicionalmente, foi utilizado o **t-test1**¹⁵ (versão 1.0.1) para teste de desempenho que mede o tempo de alocação em memória. Por fim, foi selecionado o **IOzone**¹⁶ (versão 1.9.6) para realizar testes de leitura e escrita em disco.

O experimento consistiu em executar 30 vezes cada um dos quatro testes, em um **Linux Debian Desktop 11**, seja em uma máquina física (instalação *bare-metal*), seja em uma VM executada a partir do SO customizado e de um SO **Linux** não customizado. Em cada um dos três ambientes (físico, customizado e não customizado), foram executados os *benchmarks* acima e os resultados estão dispostos na Figura 6. Para melhor entendimento da figura, os resultados dos ambientes com virtualização (customizado e não customizado) estão normalizados (diferença percentual) em relação aos obtidos na máquina física, uma vez que espera-se conseguir um desempenho melhor nesta última.

A primeira figura (6a), por exemplo, ilustra a diferença percentual entre o tempo de renderização na máquina física e os outros dois ambientes. Pode-se observar que, em praticamente todas as rodadas executadas, o ambiente customizado consegue renderizar com um tempo melhor do que o não customizado, chegando a ser melhor até mesmo que o ambiente físico em 3 rodadas (11, 25 e 29). A diferença percentual média é de 0,56%(±0,17%) para o SO customizado e 2,66%(±0,23%) para o não customizado. Ou seja, em média, o SO customizado leva 0,56% a mais de tempo para renderizar, se comparado ao tempo na máquina física. O resultado do teste de memória (6b) também é melhor para a versão customizada em relação a não customizada, com as diferenças médias em relação à máquina física de 15,98%(±0,29%) e 21,44%(±0,55%), respectivamente. Por fim, os resultados de leitura e escrita em disco (6c e 6d) medem a velocidade com que a ação é executada e, portanto, as diferenças percentuais são negativas, apontando que a

¹³ <https://openbenchmarking.org/>

¹⁴ <https://blender.org>

¹⁵ <https://locklessinc.com>

¹⁶ <http://iozone.org>

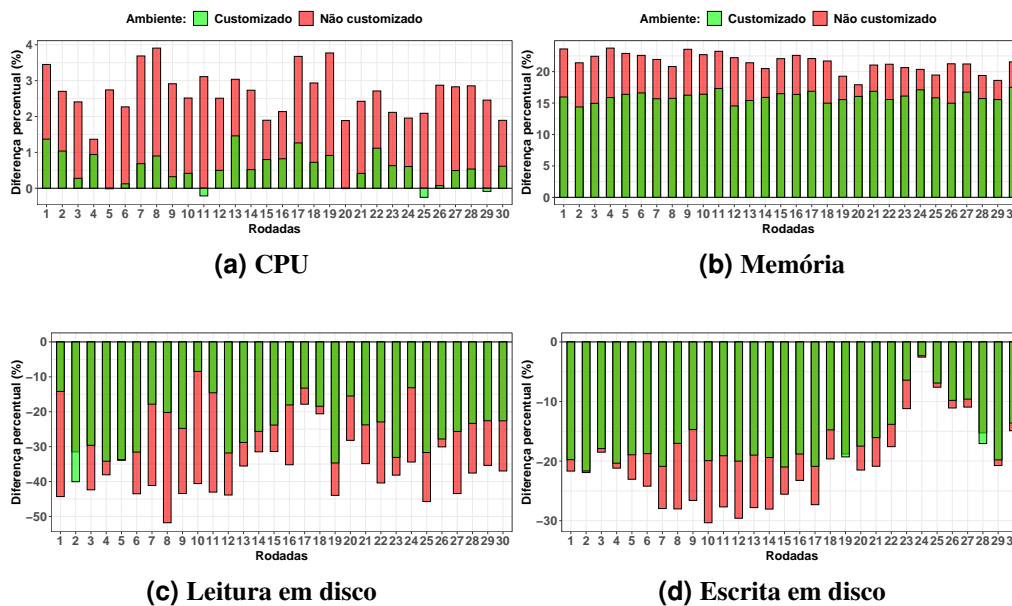


Figura 6: Resultados dos testes de *benchmark*.

máquina física é $y\%$ mais veloz do que as demais. Em resumo, o SO customizado é, em média, $24,18\%(\pm 2,91\%)$ mais lento que a máquina física para leitura em disco e o não customizado é $37,29\%(\pm 2,73\%)$. Já para a escrita em disco as diferenças médias são de $16,50\%(\pm 1,83\%)$ e $20,84\%(\pm 2,61\%)$. Pelos resultados obtidos, pode-se inferir que o SO customizado não foi afetado pela alterações realizadas e customizações futuras poderão melhorar ainda mais o seu desempenho. As médias e os respectivos erros possuem nível de confiança de 95% .

6. Conclusões

Este trabalho baseia-se na hipótese de que laboratórios de ensino e pesquisa podem ser gerenciados de maneira mais eficientes a partir de tecnologias abertas de virtualização e nuvem. Com isso, busca-se não apenas a economia na gestão de recursos públicos, mas também a efetiva utilização de equipamentos legados, tendo em vista que soluções proprietárias são dispendiosas e geralmente necessita de maquinário próprio. Este é o cenário em Institutos Federais de Educação Profissionalizante no Brasil, onde tal gerência é realizada de maneira repetida e ineficientemente. A virtualização é hoje a tecnologia principal que possibilita grandes áreas da computação, como por exemplo a de sistemas distribuídos e computação em nuvem. Imagina-se que a virtualização das aulas possibilitará o uso dos laboratórios de forma adequada, mitigando, por exemplo, o efeito dos problemas de *software* que comumente acontece. Para tanto, a arquitetura **vCLASS** se baseia em 3 premissas básicas. A primeira é o uso eficiente de recursos públicos legados, sem a necessidade de comprar soluções proprietárias de *hardware* e *software*. A segunda é a gerência centralizada de laboratórios de ensino e pesquisa, reduzindo o trabalho manual realizado pelas equipes responsáveis de maneira repetida a cada semestre letivo. Por fim, vislumbra-se que essa transformação digital do ambiente de ensino irá proporcionar um ecossistema onde professores e estudantes poderão compartilhar conteúdo (aulas, exercícios, práticas, etc) de maneira muito mais eficiente e adaptativa.

Como trabalho futuro, planeja-se uma investigação completa de soluções alternativas às ferramentas já utilizadas, bem como de novas ferramentas para compor a arquitetura do projeto, com o intuito de obter o melhor desempenho possível do ambiente virtualizado. Adicionalmente, pretende-se personalizar as imagens das máquinas virtuais para um menor tempo de implantação e considerar um ambiente empírico de avaliação.

Referências

- [Genkov and Slavov 2021] Genkov, D. and Slavov, M. (2021). Implementation of a virtual laboratory for computer oriented disciplines. In *2021 29th Telecommunications Forum (TELFOR)*, pages 1–4. IEEE.
- [Huang 2019] Huang, A. (2019). Teaching, learning, and assessment with virtualization technology. *Journal of Educational Technology Systems*, 47(4):523–538.
- [Krassmann et al. 2014] Krassmann, A. L. et al. (2014). Avaliação de implementações de computação em nuvem para ambientes educacionais: um mapeamento sistemático. *Novas Ideias em Informática Educativa. TISE*, pages 194–205.
- [Martins 2022] Martins, M. A. T. (2022). *Infrastructure as Code: Automatização do provisionamento e configuração de Infraestrutura e Serviços no Ensino Superior*. Masters dissertation, Instituto Politécnico de Viseu.
- [Masek et al. 2018] Masek, P. et al. (2018). Unleashing full potential of ansible framework: University labs administration. In *2018 22nd conference of open innovations association (FRUCT)*, pages 144–150. IEEE.
- [Moreno et al. 2019] Moreno, F. et al. (2019). Visual tabs: software para auxiliar o ensino de tabelas hash na disciplina de estrutura de dados. In *Anais do XLVI Seminário Integrado de Software e Hardware*, Porto Alegre, RS, Brasil. SBC.
- [Rangavittala et al. 2015] Rangavittala, S., Sanjay, H., and Salvi, S. (2015). Enhanced multi-tenant architecture for daas, paas, iaas and saas in edu-cloud: Simplifying the service provisioning in edu-cloud by multi-tenant architecture. In *Proc. of the Sixth International Conference on Computer and Communication Technology*, pages 51–56.
- [Rot et al. 2019] Rot, A., Chrobak, P., and Sobinska, M. (2019). Optimisation of the use of it infrastructure resources in an institution of higher education: A case study. In *2019 9th International Conference on Advanced Computer Information Technologies (ACIT)*, pages 171–174. IEEE.
- [Rubio et al. 2016] Rubio, M. S., Civera, G. L., and Herraiz, J. J. M. (2016). Automatic generation of virtual machines for security training. *IEEE Latin America Transactions*, 14(6):2795–2800.
- [Sampaio et al. 2015] Sampaio, F. et al. (2015). Labvad: Uma proposta de consórcio nacional para laboratórios didáticos a distância de ciências e robótica. In *Anais do XLII Seminário Integrado de Software e Hardware*, Porto Alegre, RS, Brasil. SBC.
- [Shukhman et al. 2017] Shukhman, A. et al. (2017). Cloud educational resource datacenter simulator. *Procedia Computer Science*, 103:543–548.
- [Xiong et al. 2021] Xiong, N., Zhou, S., Wu, Z., and Zhang, Z. (2021). Design and research of hybrid cloud desktop scheme in colleges and universities. In *MATEC Web of Conferences*, volume 336, page 05004. EDP Sciences.