

## Análise de Desempenho do Ambiente Virtual de Aprendizagem na Nuvem Privada Apache CloudStack

### Performance Evaluation of the Virtual Learning Environment in Apache CloudStack Private Cloud

Alison Vinicius Gomes da Silva\*<sup>1</sup>, Claudemir Jeremias de Lima<sup>1</sup>, Gustavo Rau de Almeida Callou<sup>1</sup>

<sup>1</sup>Departamento de Computação, Universidade Federal Rural de Pernambuco, Pernambuco, Brasil.

---

#### INFO ARTIGO

##### Palavras-chave:

Avaliação de Desempenho, Ambiente Virtual de Aprendizagem, Computação em Nuvem, Rede de Petri Estocástica.

---

#### RESUMO

A computação em nuvem é um paradigma que oferece recursos computacionais de forma dinâmica por meio da Internet. As universidades e escolas vêm adotando cada vez mais os Ambientes Virtuais de Aprendizagem (AVA) para facilitar a comunicação entre alunos e professores e, assim, existe uma necessidade de uma avaliação de desempenho do AVA em softwares de computação em nuvem. Dessa forma essa pesquisa visa propor uma estratégia baseada em modelos para a avaliação de desempenho do ambiente virtual de aprendizagem Moodle em nuvem privada. A metodologia para a realização das medições e modelagem do sistema foram desenvolvidos em rede de Petri estocástica. Este trabalho apresenta três estudos de casos realizados para ilustrar a aplicabilidade da metodologia e dos modelos propostos em um ambiente real, montado em laboratório, com o Moodle configurado em uma nuvem privada com Apache CloudStack. Os resultados mostram que o modelo de desempenho foi prático e eficiente para quantificar as métricas de interesse atestando a qualidade do serviço provido além de auxiliar os projetistas a estimar as configurações demandas para dar suporte ao sistema em diferentes situações e com diversos números de usuários.

---

#### ARTICLE INFO

##### Keywords:

Performance Evaluation, Virtual Learning Environment, Cloud Computing, Stochastic Petri Net.

---

#### ABSTRACT

Cloud computing is a paradigm that offers computing resources dynamically over the Internet. Universities and schools are increasingly adopting Virtual Learning Environments (VLE) to facilitate student-teacher communication, so there is a need for VLE benchmarking on cloud computing software. This paper performs the performance evaluation of the private cloud VLE. A methodology was proposed to perform the measurements and modeling of the system. The model was developed in stochastic petri net. This paper presents three case studies performed to illustrate the applicability of the proposed methodology and models in a real, lab-mounted environment with Moodle configured in a private cloud with Apache CloudStack. The results show that the performance model was practical and efficient to quantify the metrics of interest.

---

\* Correspondência para autor:

[alisonviniciusgs@gmail.com](mailto:alisonviniciusgs@gmail.com) (Silva, A. V. G.), [limasectma@gmail.com](mailto:limasectma@gmail.com) (Lima, C. J.), [gustavo.callou@ufrpe.br](mailto:gustavo.callou@ufrpe.br) (Callou, G.).

<http://dx.doi.org/10.21714/1679-18272019v17Esp.p120-133>

1679-1827 © 2019 Gest@o.org.

## 1. Introdução

Surgindo em meados de 1961, o termo computação em nuvem apresentava um modelo de computação oferecido com um serviço parecido com a distribuição de energia elétrica (MANSUR et al., 2010). A computação em nuvem é um conjunto de data centers virtualizados, que oferece serviço sob demanda, também conhecido como *pay per use* (pagar pelo que usar). Atualmente, grandes empresas utilizam a computação em nuvem para oferecer seus serviços, exemplos como a *Netflix*, *Google*, *Spotify* e *Uber*.

A computação em nuvem surgiu de uma necessidade que as empresas de comércio tinham em épocas de datas festivas como Natal, onde ocorria o aumento de recursos computacionais para conseguir atender a demanda requerida para essas datas. O problema era quando se passava essas datas comemorativas, os recursos extras que foram adquiridos ficavam ociosos e ocorreria um gasto desnecessário.

Então várias empresas atualmente, estão criando *softwares open source* para oferecer infraestrutura como serviço, com segurança, alta disponibilidade, confiabilidade e entre outros. Entre as empresas que oferece essa infraestrutura está o *Apache CloudStack*, *OpenStack* e *Eucalyptus*.

A computação em nuvem é um paradigma que é composto de *Grid computing* e *Utility computing* (KIM et al., 2010). A *Grid computing* é uma tecnologia utilizada para o desenvolvimento da computação em nuvem enquanto a *Utility computing* é um pacote de recursos computacional (XIONG; PERROS, 2009). As suas principais características são rápida elasticidade, alta disponibilidade, confiabilidade e baixo custo (BAUER; ADAMS, 2012).

Inicialmente, o ambiente virtual de aprendizagem (AVA) era usado para servir de comunicação entre aluno e professor no ensino a distância, atualmente atendem cursos presenciais e a distância onde também possuem vários recursos que ajudam na educação. As vantagens dos AVAs são a possibilidade de estudo a qualquer hora e qualquer lugar só necessitando de Internet para realizar o acesso, aprendizado coletivo onde vários alunos e professores trocam ideias em fóruns e chats e podendo, inclusive, elevar o ritmo de aprendizado.

Como o ambiente virtual de aprendizagem e o ambiente de computação em nuvem tem que garantir uma alta disponibilidade e oferecer uma boa performance para seus usuários. O gestor dos ambientes tem que planejar qual ambiente será utilizado e que atenda as demandas de desempenho e disponibilidade (DANTAS, 2018). A utilização de técnicas com medição, modelagem analítica e simulação podem ajudar o gestor na hora do planejamento, pois com essas avaliações será possível uma escolha que mais se ajuste a sua necessidade.

Dessa forma, as simulações para avaliação de desempenho tornam-se necessária para tomadas de decisões de forma proativa promovendo ações mais eficazes, principalmente em época de provas o acesso ao ambiente fica maior e com isso pode haver lentidão do sistema. No caso se a prova for online, haverá um grande fluxo de acessos de usuários, uploads e downloads de arquivos, o que pode gerar uma sobrecarga como é o caso do AVA da Universidade Federal Rural de Pernambuco (UFRPE).

Tal AVA supracitado, apresenta cerca de treze mil alunos matriculados na graduação e pós-graduação em 2017 (UFRPE, 2017), o uso do ambiente virtual de aprendizagem está cada vez mais presente no dia a dia dos docentes e discentes. Lançado em 2015 na UFRPE (UFRPE, 2015), o AVA aproxima a relação e a comunicação entre alunos e professores com recursos de fóruns, chats e atividades síncronas e assíncronas. O *Moodle* foi a ferramenta escolhida para ser adotada como AVA da UFRPE.

Contudo, para a utilização do *Moodle* hospedado em ambiente de nuvem privada, é importante realizar avaliações de desempenho, e assim atestar a qualidade do serviço provido. Dessa forma, auxiliar os projetistas a estimar as configurações demandas para dar suporte ao sistema em diferentes situações e com diversos números de usuários.

Dada a necessidade da realização de um estudo de desempenho sobre o AVA, o objetivo desse trabalho é propor uma estratégia baseada em modelos para a avaliação de desempenho do ambiente virtual de aprendizagem *Moodle* em nuvem privada.

Dentro dessa perspectiva, propõe medir o desempenho do Ambiente Virtual de Aprendizagem através de um conjunto métricas como vazão, tempo de resposta, utilização de CPU e memória e taxa de escrita em HD. A partir dessas métricas, visa gerar base de dados com os valores das medições em nuvem privada montada em laboratório e enfim propor modelo de desempenho para representar o Ambiente Virtual de Aprendizagem.

A metodologia dessa pesquisa foi a partir de três estudos de casos realizados para analisar a aplicabilidade da metodologia e dos modelos propostos com simulações baseadas em um ambiente real, montado em

laboratório, com o *Moodle* configurado em uma nuvem privada com *Apache CloudStack* para a realização das medições e modelagem do sistema em rede de *Petri* estocástica.

## 2. Trabalhos Relacionados

O estado da arte possui várias pesquisas relacionados com este trabalho. Alguns deles têm como foco a comparação de infraestruturas de nuvens, avaliação de desempenho do *Moodle*, modelagem para avaliar desempenho em nuvens privadas e avaliação de desempenho em banco de dados.

Sousa (SOUSA, 2015) realizou um planejamento para infraestruturas de nuvens privadas com uma solução integrada com metodologia, modelos de otimização utilizando meta heurística *Greed Randomized Search Procedure* (GRASP), modelos para geração de cenários de desempenho e custo e modelos de representação. Os modelos de representação são compostos de *Reliability Block Diagram* (RBD) e *Stochastic Petri Nets* (SPN) e com eles são realizados avaliação de desempenho e disponibilidade. Nos estudos de casos foram avaliados o impacto da variação de carga, da ocorrência de defeitos e atividades de reparo no ambiente de nuvem *Eucalyptus* utilizando vários cenários de configuração de *software* e de *hardware*.

Júnior e Tavares (JUNIOR; TAVARES, 2015) realizaram uma avaliação de desempenho e dependabilidade do *Moodle* na plataforma *OpenNebula*, utilizando cenários com 1 a 10 usuários simultâneos para acessar um quiz dentro do ambiente virtual de aprendizagem. Neste trabalho foram contabilizados os números de amostras, tempo médio de resposta, requisições por segundos e em *kilobytes* por segundos utilizando as ferramentas *JMeter* e *Syssat*. Como conclusão foi reportada a importância do dimensionamento de recursos computacionais fornecidos pela infraestrutura da nuvem para atender as necessidades do *Moodle*.

Os autores (MARON et al., 2014) avaliaram o desempenho das plataformas *OpenStack* e *OpenNebula* utilizando os *benchmarks Iperf* para avaliar a rede e coletar a vazão, *Iozone* para testar unidades de armazenamento com um arquivo de 100 MB analisando a escrita, reescrita, leitura e releitura, *Stream* para testes de memória, analisando as funções de *Add*, *Scale*, *Triad* e *Copy* e o *Linpack* para os testes de processador com uma matriz de 4000x4000 para ser executado pontos flutuantes. Ao final, foi apresentado um resumo comparativo do desempenho obtido para cada ambiente a partir dos *benchmarks* avaliados, onde foi mostrado que o *OpenStack* é recomendado para aplicações que utilizam somente o disco e *OpenNebula* é melhor para usar rede, memória e processador.

Torres (TORRES et al., 2016) propôs modelos em *Continuous Time Markov Chain* (CTMC), SPN e RBD para uma avaliação integrada de métricas de disponibilidade e desempenho em um serviço de armazenamento em uma nuvem privada gerenciada pelo ambiente *Eucalyptus*. O autor utilizou para as medições as ferramentas *JMeter* e *Apache benchmarking* para validação do modelo. Em seus testes de desempenho foram utilizadas métricas de vazão e utilização do sistema.

Os autores (JUNIOR et al., 2012) propõem desenvolver uma extensão para o *Moodle*, que visa fornecer um ambiente de aprendizagem ubíqua utilizando recomendação de objetos de aprendizagem. Foi utilizado um algoritmo genético para identificar quais objetos de aprendizagem serão recomendados. No estudo de caso o estudante realiza o *login*, responde ao um questionário para identificar suas preferências, depois escolhe qual é o seu dispositivo móvel, faz o *download* e instala o aplicativo em seu celular e quando entrar na aplicação terá mensagens com as recomendações de objetos de aprendizagem de acordo com seu perfil.

A Tabela 1 mostra as características dos cinco trabalhos relacionados e do trabalho proposto, destacando os seguintes aspectos: modelagem, ambiente de nuvem utilizado e Ambiente Virtual de Aprendizagem.

Tabela 1 – Características dos Trabalhos Relacionados.

Trabalhos	Modelagem	Ambiente de Nuvem	AVA
(SOUSA, 2015)	SPN e RBD	Eucalyptus	Não
(JUNIOR; TAVARES, 2015)	-	OpenNebula	Sim
(MARON et al., 2014)	-	OpenStack e OpenNebula	Não
(TORRES et al., 2016)	CTMC, SPN e RBD	Eucalyptus	Não
(JUNIOR et al., 2012)	-	-	Sim

Este Trabalho	SPN	CloudStack	Sim
---------------	-----	------------	-----

### 3. Fundamentação Teórica

Esta seção apresenta os conceitos básicos para um melhor entendimento deste trabalho. Primeiramente serão apresentados os conceitos sobre computação em nuvem. Em seguida, serão abordados os principais conceitos do AVA e avaliação de desempenho. Também são abordados os principais conceitos sobre rede de Petri e sua extensão, a rede de Petri estocástica. E por fim, apresenta-se a regressão linear.

#### 3.1 Computação em nuvem

O modelo de computação em nuvem foi desenvolvido com o objetivo de fornecer serviços de fácil acesso e de baixo custo e garantir características tais como disponibilidade e escalabilidade (RUSCHEL; ZANOTTO; MOTA, 2010).

A NIST (NIST, 2014) definiu a computação em nuvem como um modelo para acesso ubíquo, conveniente e sob demanda a uma rede compartilhada de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente adquiridos e liberados com mínimo esforço gerencial ou interação com o provedor de serviços. Existe três tipos de serviços de computação em nuvem definidos pela NIST (NIST, 2014):

- **Infraestrutura como Serviço (Infrastructure as a Service - IaaS):** É um modelo de serviço que fornece recursos de computação como hardware, armazenamento e rede. Podemos citar como exemplos de IaaS são o *Apache CloudStack*, *Amazon EC2* e *Google Cloud*.
- **Plataforma como Serviço (Platform as a Service - PaaS):** É um modelo de serviço onde o usuário pode utilizar para desenvolvimento e execução de aplicações. Os exemplos de PaaS são *Google Cloud Plataform*, *Microsoft Azure* e *Red Hat OpenShift*.
- **Software como Serviço (Software as a Service - SaaS):** É um modelo de serviço onde é fornecido aplicativos executados em uma infraestrutura de nuvem. Os exemplos de SaaS são *Google Apps*, *Dropbox* e *Microsoft Office 365*.

Além disso, também existe quatro tipos de modelos de computação em nuvem (NIST, 2014):

- **Nuvem pública:** a infraestrutura em nuvem é disponibilizada ao público geral.
- **Nuvem privada:** a infraestrutura em nuvem é operada exclusivamente para uma organização.
- **Nuvem comunitária:** a infraestrutura da nuvem é compartilhada por várias organizações.
- **Nuvem híbrida:** a infraestrutura da nuvem por uma composição de duas ou mais nuvens (privada, pública e comunitária).

Este trabalho vai abordar o uso da IaaS com o software Apache CloudStack. E será usado para o gerenciamento dos recursos da nuvem privada.

#### 3.2 Ambiente Virtual de Aprendizagem

Durante a década 90, com expansão e implementação de programas educacionais em meio online, os AVAs começaram a ganhar espaço e reconhecimento no âmbito educacional (BELUCE; OLIVEIRA, 2012). O AVA é uma ferramenta com dimensão pedagógica em que o professor disponibiliza vários recursos para o aluno, como textos, aulas, cronogramas e exercícios (HAGUENAUER; LIMA; FILHO, 2010).

Os AVAs possuem recursos de interação, comunicação e locais onde os participantes podem armazenar conteúdos utilizando várias mídias (SILVA, 2012). As principais características do AVA são a sua interatividade, hipertextualidade e conectividade (KENSKI, 2007).

Neste trabalho vai utilizar o AVA Moodle que é uma das ferramentas mais usadas atualmente e também é utilizada pela UFRPE tanto para os cursos presenciais como também para os cursos a distância (EaD).

#### 3.3 Avaliação de Desempenho

A avaliação de desempenho pode ser classificada com medição e modelagem (CALLOU et al., 2011; JAIN, 1991). Um dos principais propósitos para o desenvolvedor é fornecer alto desempenho com baixo custo (BARROS, 2018).

A medição de um sistema é o método mais direto para a avaliação de desempenho. As técnicas de medição apesar de fornecerem respostas exatas sobre o desempenho do sistema, não podem ser utilizadas se um novo sistema estiver sendo feito (TORRES et al., 2016). Já a modelagem analítica é utilizada para abstração do sistema. O modelo deve conter detalhes do sistema que são essenciais para o seu comportamento. E a simulação do modelo

dar resultados não tão precisos quanto os fornecidos pela medição, mas é possível calcular as estimativas (CALLOU et al., 2011).

### 3.4 Redes de Petri

A origem do conceito da rede de Petri (PN) ocorreu em 1962 na tese de doutorado de Carl Adam Petri na faculdade de Matemática e Física da Universidade Darmstadt na Alemanha (MURATA, 1989). A rede de Petri é uma técnica de especificação de sistemas que permite uma representação matemática e que é usada para modelar sistemas paralelos, concorrentes, assíncronos e não-determinísticos (MACIEL; LINS; CUNHA, 1996).

A rede de Petri é composta por 4 elementos: lugares, transições, arcos e *tokens*. Os lugares representam o estado do sistema. As transições são ações realizadas para alterar o estado do sistema. Para uma transição estar habilitada, é preciso que as pré-condições sejam satisfeitas. Os arcos representam o fluxo de *token* pela rede e os *tokens* representam o estado que o sistema se encontra. Graficamente, os lugares são representados por círculos, as transições, por barras, os arcos, por setas e os *tokens*, por pontos. A Figura 1 ilustra os elementos da rede de Petri.

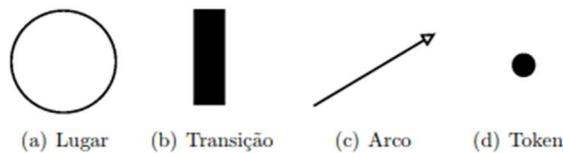


Figura 1 – Elementos da rede de Petri.

A representação formal de uma rede de Petri é uma 5-tupla  $PN = \{P, T, F, W, M_0\}$ .

- $P = \{p_1, p_2, \dots, p_n\}$  é um conjunto finito de lugares;
- $T = \{t_1, t_2, \dots, t_n\}$  é um conjunto finito de transições;
- $F \subseteq (P \times T) \cup (T \times P)$  é um conjunto de arcos;
- $W : F \rightarrow \{1, 2, 3, \dots\}$  é uma função de peso do arco;
- $M_0 : P \rightarrow \{0, 1, 2, 3, \dots\}$  é a marcação inicial,  $P \cap T = \emptyset$  e  $P \cup T \neq \emptyset$

A seguir, a Figura 2 mostra um exemplo de PN. O modelo apresenta o funcionamento de um sistema, onde os lugares representam o estado do sistema (*on* ou *off*) e as transições representam as ações que altera o estado do sistema (ligar e desligar). O modelo na Figura 2 (a) mostra que o sistema está ligado, pois possui *token* no lugar *on*. A única transição que pode ser disparada é desligar. Depois do disparo desta transição, o modelo passará para o estado desligado e possuirá um *token* no lugar *off* (Figura 2 (b)). Após isso, a transição habilitada para disparar é a Ligar, e o disparo desta fará o sistema voltar ao estado de ligado como na Figura 2 (a).

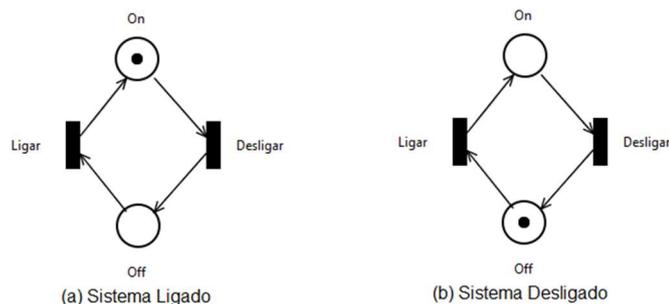


Figura 2 – Exemplo de uma rede de Petri.

As redes de Petri estocástica são uma extensão da rede de *Petri* utilizada para a modelagem de desempenho e dependabilidade (TORRES et al., 2016). Nas SPN são adicionados dois elementos, como pode ser visto na Figura 3, que são as transições estocásticas Figura 3 (a) e o arco inibidor Figura 3 (b). A transição estocástica utiliza tempos associados, de modo que o período de habilitação da transição corresponde ao período de execução da atividade. Já o arco inibidor desativa a transição se houver *tokens* no lugar de origem do arco inibidor, e vai ser ativada quando não houver *tokens* (SILVA, 2016).

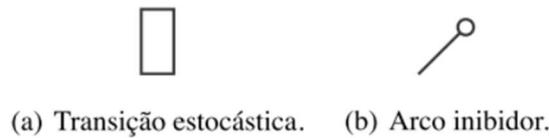


Figura 3 – Elemento da SPN.

Com os modelos em SPN é possível realizar cálculos de métricas de desempenho. Um exemplo dela é o tempo de resposta, que é o tempo entre o pedido de realização de um serviço pelo usuário e a chegada de resposta do pedido ao usuário (TRIVEDI et al., 1996).

### 3.5 Regressão Linear

A regressão linear é uma técnica estatística para relacionar uma variável x que é independente com uma variável y que é dependente (MATOS, 1995). A regressão linear simples é dada pela seguinte Equação 1.

$$y = a + bx \tag{1}$$

Onde y é a resposta de saída prevista, x é a variável de entrada e a e b são os parâmetros de regressão. O cálculo do parâmetro de regressão b é dada pela seguinte Equação 2.

$$b = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \tag{2}$$

Onde n é quantidade de pontos (x, y). Existe um grau de erro associado a regressão que é chamado de coeficiente de determinação e é representado por  $R^2$  (LILJA, 2000). Esse grau de erro varia entre zero e um. Quanto mais próximo o valor do coeficiente de determinação estiver de um, maior será a validade da regressão. O cálculo do coeficiente de determinação é dado pela seguinte Equação 3.

$$R^2 = \frac{b \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (y_i - \bar{y})^2} \tag{3}$$

## 4. Metodologia

Esta seção apresenta a metodologia usada para a avaliação de desempenho no ambiente virtual de aprendizagem em nuvem privada. Também são contextualizados as atividades realizadas e os objetos adquiridos em cada uma das atividades.

A Figura 4 ilustra a metodologia adotada para este trabalho, destacando as suas atividades que são: estudar o sistema, instalar e configurar a nuvem, instalar e configurar o AVA, criar plano de teste, realizar medição, analisar resultados, criar modelo, calcular regressão linear, simular o modelo e validar o modelo e os objetos adquiridos ao final de cada atividade. A seguir, cada fase da metodologia adotada será detalhada.

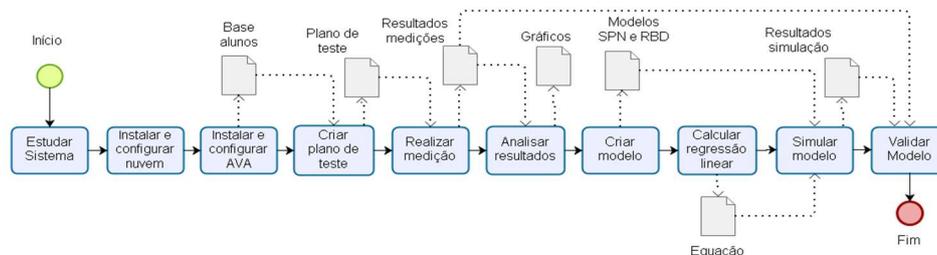


Figura 4 – Metodologia adotada.

**Estudar o sistema:** O objetivo desta fase foi realizar um estudo com a finalidade de se obter um entendimento da infraestrutura em nuvem *Apache CloudStack* e seus componentes, entendimento do funcionamento do ambiente virtual de aprendizagem *Moodle* que é utilizado na UFRPE, definição de métricas que serão utilizadas para avaliação do desempenho.

**Instalar e configurar a nuvem:** Esta atividade compreende a instalação e configuração da infraestrutura em nuvem com o *Apache CloudStack*. O ambiente foi montado em computador com a seguinte configuração: processador AMD A8-550B 3.2 GHz com 4 cores, 8 GB de memória RAM, 500 GB de armazenamento e sistema operacional CentOS 7. Também foi instanciado uma máquina virtual com a seguinte configuração: processador com 2 GHz e 2 cores, 4 GB de memória RAM, 20 GB de armazenamento e sistema operacional CentOS 7.

**Instalar e configurar o AVA:** Esta fase corresponde a instalação e configuração do ambiente virtual de aprendizagem Moodle na máquina virtual. Além disso, nesta atividade ainda ocorre a criação de uma base de alunos, professores, curso, turma e atividade. A base de alunos foi criada a partir do modelo de cadastro do Moodle, onde contém informações obrigatórias como: nome, E-mail e senha. Todos esses dados inseridos são fictícios.

**Criar plano de teste:** Um plano de teste na ferramenta de medição JMeter foi criado com a finalidade de se realizar experimentos no ambiente montado em laboratório. Nesta atividade será utilizada a base de dados de alunos que servirá como carga para o teste. Aqui foram definidas as seis atividades realizadas no ambiente que são: entrar no Moodle, realizar *login*, entrar no curso, entrar na atividade, realizar o *upload* de um arquivo de tamanho de 1 MB na atividade e realizar *logout*. Também foram definidos a quantidade de usuários virtuais e o tempo de inicialização de cada um.

**Realizar medição:** Esta atividade corresponde a medição utilizando a ferramenta JMeter e a ferramenta Vmstat, onde foi aplicado o plano de teste criado na atividade anterior. O JMeter foi responsável pela coleta de métricas como vazão e tempo de resposta, já o Vmstat mediu utilização da CPU, utilização da memória e taxa de escrita no HD. Os testes ocorreram trinta vezes para cada quantidade de clientes, que foi de um até quarenta clientes simultâneos e com chegada entre eles de 0,1 segundos. Essas repetições de testes são para se obter um padrão nos resultados. A Figura 5 ilustra a visão geral do ambiente de medição. O ambiente é composto de um cliente com a ferramenta JMeter para realizar as medições, uma máquina que será utilizada para ser o ambiente de nuvem que foi gerenciado pelo Apache CloudStack, uma VM onde foram instalados: o AVA Moodle e a ferramenta de medição Vmstat. E por fim, um *switch* Gigabit que foi responsável por fazer a ligação entre as máquinas.

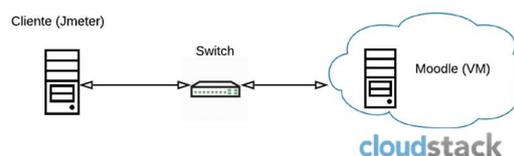


Figura 5 – Visão geral do ambiente de medição.

**Analisar resultados:** A análise dos resultados foi feita com cálculos estatísticos de média, desvio padrão, intervalo de confiança. E ao final são plotados os gráficos de comparação dos resultados com tempo de resposta, vazão, utilização de CPU, utilização da memória e taxa de escrita no HD.

**Criar modelo:** Esta fase corresponde à criação de modelo SPN de desempenho. Vale ressaltar que este modelo de desempenho em SPN é refinado na fase seguinte com a adoção de técnicas de regressão linear. O objetivo é conseguir representar o tempo diferente que o sistema demanda quando se trabalha com uma diferente quantidade de usuários no sistema. Sendo assim, o tempo de algumas atividades pode ser representado através de uma equação linear.

**Calcular regressão linear:** Realização do cálculo da regressão linear entre a quantidade de clientes no sistema e o tempo de atendimento. Ao final foi gerado uma equação que será utilizado pelo modelo de desempenho.

**Simular o modelo:** Será realizado a simulação com os mesmos cenários da medição no modelo SPN de desempenho. Sendo assim, o mesmo plano de testes realizado na medição foi adotado para a realização da simulação dos modelos propostos.

**Validar o modelo:** Para se realizar a validação do modelo proposto foi adotada a seguinte estratégia:

- I. Deve-se computar o intervalo de confiança para cada métrica de interesse tanto nos resultados do modelo, como também nos resultados da medição;
- II. Compara-se intervalos de confiança e, se estiverem sobrescrevendo, pode-se dizer que o modelo consegue representar o sistema modelado.

**5. Modelo**

Esta seção apresenta o modelo de desempenho para o Moodle em nuvem privada, mostrando as métricas, parâmetros e a regressão linear utilizada. A Figura 6 ilustra o modelo SPN de desempenho proposto para representar o comportamento do Moodle nos cenários analisados. O modelo representa o funcionamento da chegada de clientes até a resposta do AVA. Esse cenário mostra que um cliente que chega vai fazer n requisições. E, ao final, o sistema envia a resposta ao cliente.

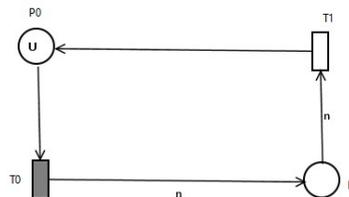


Figura 6 – Modelo de desempenho proposto.

O número de *tokens* presente no P0 representa a quantidade de clientes (U) que serão atendidos pelo sistema. A transição determinística T0 representa o tempo de chegada de usuário. Ao ser disparada a transição T0, um *token* é consumido de P0 e n *tokens* são gerados em P1. O lugar P1 representa a fila de requisições a serem atendidas pelo sistema. A transição T1 representa o tempo demandado para atender os n passos do cenário analisado. O disparo desta transição T1 faz com que n *tokens* sejam consumidos do lugar P1 e gera um *token* no lugar P0. Como pode ser visto na Figura 6, o arco que liga P0 a T0 tem peso um, onde representa a chegada de um cliente, já o arco que liga T0 a P1 tem peso n, que representa que o cliente vai realizar n atividades no Moodle, o arco que une P1 a T1 também possui peso n, pois o sistema realizará o atendimento dessas atividades e o arco que liga T1 a P0 possui peso 1 que significa que o sistema atendeu um cliente.

A Tabela 2 apresenta a descrição dos lugares do modelo e a Tabela 3 descreve as transições do modelo. Os parâmetros utilizados no *token* inicial do lugar P0 e nos tempos da transição serão explicados na subseção a seguir.

Tabela 2 - Descrição dos lugares do modelo.

Lugar	Nº Token Inicial	Descrição
P0	U	Quantidade de clientes que serão atendidos pelo sistema
P1	0	Fila de requisições a serem atendidas pelo sistema

Tabela 3 - Descrição das transições do modelo.

Transição	Tipo	Tempo	Tipo Serv.	Descrição
T0	Determinística	tCheg	Single Server	Chegada de n cliente no sistema
T1	Exponencial	tAten	Infinite Server	Atendimento de n cliente

### Parâmetros e Métricas do Modelo

O modelo possui três parâmetros que devem ser especificados para realizar a análise do modelo de desempenho. O modelo também possui duas métricas, uma é responsável por calcular o tempo de resposta, e a outra é destinada para o cálculo da vazão do sistema.

O primeiro parâmetro é  $t_{Cheg}$  da ferramenta que representa o tempo de chegada de clientes ao sistema. Este tempo é definido no ambiente da ferramenta JMeter na construção do plano de teste. Com o tempo de chegada é possível calcular a taxa de chegada de clientes que é dado por  $1/t_{Cheg}$ . O segundo parâmetro é  $U$  que representa a quantidade de clientes que serão atendidos pelo sistema. O terceiro e último parâmetro é  $t_{Aten}$  que é o tempo de atendimento de um cliente e é definido por uma equação de regressão linear que será apresentado na subseção a seguir.

A métrica do modelo  $tempResp$  é responsável pelo cálculo do tempo de resposta. O tempo de resposta representa o período entre o envio da requisição para o sistema até a chegada da resposta para o cliente (BARROS, 2018). Seguindo os conceitos da Lei de Little (TRIVEDI et al., 1996), a fórmula do tempo de resposta é definida na Equação 4.

$$tempResp = \frac{E\{\#P1\}}{P\{\#P0 > 0\} \times W(T0)} \quad (4)$$

Onde  $E\{\#P1\}$  indica o número médio de *tokens* no lugar P1. O  $P\{\#P0 > 0\}$  indica a probabilidade de ter *token* no lugar P0 multiplicado pela taxa de chegada de clientes ( $W$ ). Este tempo de resposta pode ser dado, por exemplo, em segundos.

A métrica do modelo  $ST$  é responsável pelo cálculo da vazão do sistema. A vazão representa a quantidade de requisições que o sistema atende durante um período de tempo. A fórmula da vazão é definida na Equação 5.

$$ST = \frac{E\{\#P1\}}{T1} \quad (5)$$

Onde  $E\{\#P1\}$  indica o número médio de *tokens* no lugar P1. O  $T1$  indica o tempo que o sistema leva para atender as requisições. A vazão é dada, por exemplo, por requisições por segundo.

### Regressão Linear

A Figura 7 apresenta a regressão linear entre a quantidade de clientes ( $x$ , independente) e o tempo de atendimento de um cliente ( $y$ , dependente).

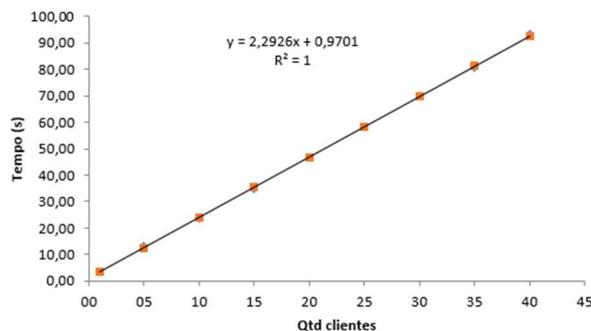


Figura 7 – Regressão Linear.

Pode-se notar que os valores de  $y$  previsto pela regressão está praticamente se sobrepondo os valores de tempo de atendimento feito pela medição e o valor da  $R^2$  foi de 0,9994. Assim,  $R^2$  bem próximo ao valor um, indica a validade da regressão. A regressão linear entre a quantidade de clientes e o tempo de atendimento de um cliente é dada pela Equação 6.

$$y = 2,2926 \times U + 0,9701 \tag{6}$$

Onde y é a previsão do tempo de atendimento de um cliente e U é a quantidade de clientes no sistema. A Equação 6 será utilizado na transição T1 do modelo de desempenho (Ver Figura 6).

### 6. Estudo de Caso

Esta seção apresenta três estudos de caso. O primeiro estudo de caso apresenta a avaliação da capacidade do sistema utilizando métricas de vazão e tempo de resposta. O segundo estudo de caso mostra-se os resultados das medições das métricas de utilização de CPU, utilização de memória e taxa de escrita no HD. O terceiro estudo de caso apresenta a validação do modelo de desempenho.

#### 6.1 Estudo de Caso I – Análise da Capacidade do Sistema

O estudo de caso apresentado nesta subseção avalia a capacidade do sistema Moodle em uma máquina virtual com duas configurações distintas. A primeira configuração tem um processador com 1GHz com 1 core e 1 GB de memória RAM, já a outra possui o mesmo processador da configuração anterior, mas altera o tamanho da memória RAM para 2GB. O teste realizado nessa etapa foi de vários clientes tentando somente acessar o Moodle. A Figura 8 mostra o resultado da comparação entre as duas configurações com a métrica da vazão.

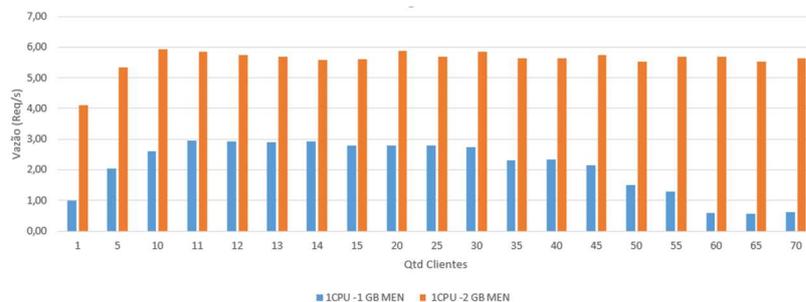


Figura 8 - Comparação da vazão.

Como pode-se notar na Figura 8, a máquina virtual que possui 1 GB de memória RAM teve um crescimento até onze clientes, manteve-se estável a vazão até trinta clientes e a partir de trinta e cinco clientes começou a ter queda. Enquanto a máquina virtual com 2 GB de memória RAM também teve um crescimento até dez clientes e durante o aumento da quantidade de clientes se manteve praticamente com a mesma vazão. A Figura 9 apresenta a comparação entre as duas configurações, mas utilizando a métrica do tempo de resposta.

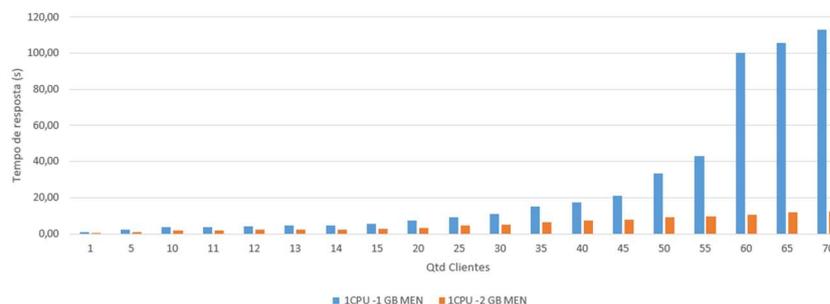


Figura 9 - Comparação do tempo de resposta.

Como pode-se notar na Figura 9 a máquina virtual que possui 1 GB de memória RAM apresentou valores inferiores a 20 segundos no tempo de resposta até 40 clientes e a partir dos 45 clientes percebe-se um aumento, onde ocorreu que o tempo mais que duplicou de 55 para 60 clientes. Já na segunda configuração o tempo de resposta tem valores inferiores a 20 segundos em todas as quantidades de clientes. Então, conclui-se que para ter um desempenho aceitável a máquina precisa de pelo menos 2 GB de memória RAM.

6.2 Estudo de Caso II – Análise da Medição de CPU, Memória e Escrita do HD

O estudo de caso apresentado nesta subseção avalia os resultados da medição das métricas de CPU, memória e escrita no HD. O objetivo de estudo foi analisar o comportamento do sistema simulando uma turma de quarenta alunos realizando as atividades propostas na Seção 4. A máquina virtual possui a mesma configuração que é mostrada na seção 4.

A Figura 10 mostra os resultados da medição da utilização da CPU. Como pode ser notado, um cliente utiliza quase 50% do processador. Nos outros casos o nível de utilização se estabiliza em torno dos 95%, pois o sistema entra em saturação. Com um maior número de clientes, o processador pode virar um gargalo para o sistema, e assim prejudicar o desempenho.

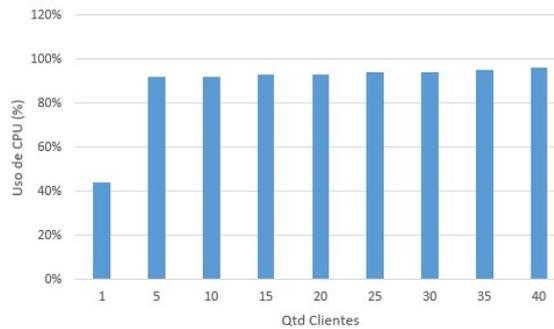


Figura 10 - Resultado da medição da utilização da CPU.

A Figura 11 mostra os resultados da medição da utilização da memória. À medida que a quantidade de clientes aumenta, a utilização da memória cresce junto. Mesmo com uma grande quantidade de clientes, o máximo de utilização de memória foi de 52%.

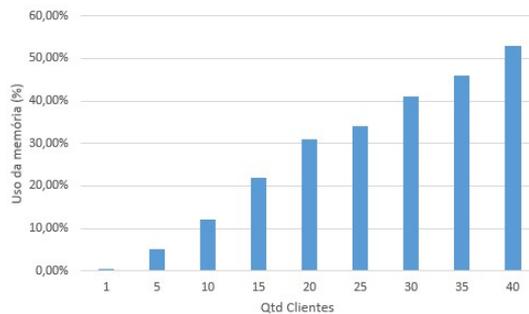


Figura 11 - Resultado da medição da utilização da memória.

A Figura 12 apresenta os resultados da medição da escrita do HD. A ferramenta *Vmstat* utiliza a unidade de medida de blocos para representar a quantidade de escrita do HD. Um bloco é que corresponde a 1024 bytes. Do mesmo jeito que acontece na utilização de memória, a quantidade de blocos escritos aumenta de acordo com a quantidade de clientes.

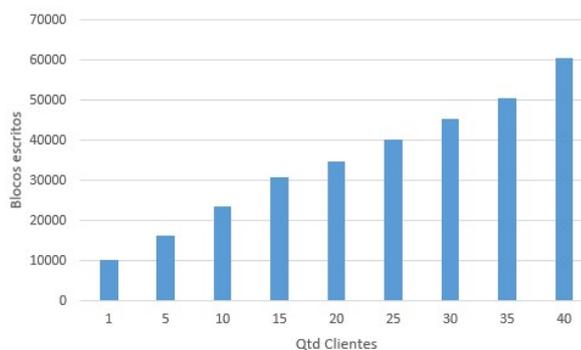


Figura 12 - Resultado da medição da escrita no HD.

### 6.3 Estudo de Caso II – Validação do Modelo

Esta subseção apresenta o estudo de caso para a validação do modelo de desempenho (ver Figura 6). A validação é um procedimento utilizado para certificar que os valores gerados pelo modelo apresentam coerência com os gerados pelo sistema real. Para ocorrer a validação do modelo, o mesmo deve ser executado nas mesmas condições impostas ao sistema. A validação desse modelo foi feita com a comparação do tempo de resposta da medição com o tempo de resposta da simulação estacionária do modelo. A configuração utilizada na máquina virtual é a mesma apresentada na seção 4.

A Figura 13 apresenta os resultados da medição e da modelagem. O nível de confiança dos resultados é de 95% e o erro relativo da simulação 10%. A Tabela 4 mostra em maiores detalhes os resultados da medição e da modelagem e seus respectivos intervalo de confiança. Para ocorrer a validação do modelo, os intervalos de confiança devem se sobrepor, como acontece na Tabela 4.

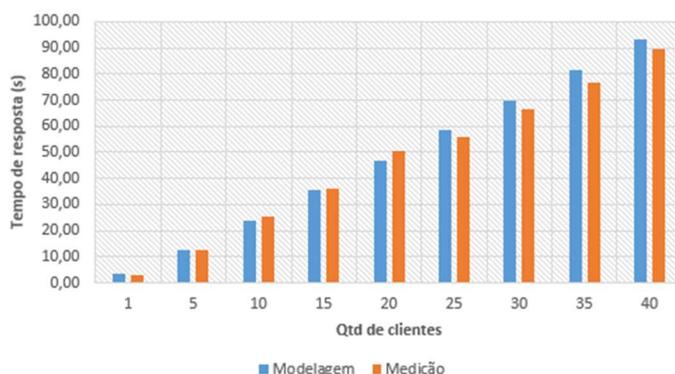


Figura 13 - Resultado da medição e modelagem.

Tabela 4 - Resultado para validação do modelo com tempo de resposta

Quantidade de Clientes	Medição		Modelagem	
	Tempo (s)	Intervalo de Confiança	Tempo (s)	Intervalo de Confiança
1	3,12	[2,73; 3,50]	3,27	[3,27; 3,28]
5	12,37	[10,29; 14,45]	12,39	[12,35; 12,44]
10	25,55	[22,54; 28,56]	23,97	[23,90; 24,04]
15	36,04	[35,20; 36,89]	35,47	[35,45; 35,48]
20	50,54	[46,96; 54,13]	46,78	[46,43; 47,13]
25	55,70	[53,25; 58,15]	58,52	[57,79; 59,25]
30	66,43	[62,80; 70,05]	69,55	[69,41; 69,68]
35	76,92	[73,27; 81,57]	81,51	[80,46; 82,26]
40	89,31	[79,95; 98,67]	93,01	[92,07; 93,96]

## 7. Conclusão

Na avaliação de desempenho, técnicas de modelagem e simulação são utilizadas para ajudar nesse processo. Este trabalho propôs um modelo SPN para avaliação de desempenho do ambiente virtual de aprendizagem Moodle em nuvem privada. Além disso, foram feitas medições das métricas de vazão, tempo de resposta, utilização de

CPU, utilização da memória e a quantidade de blocos inscritos no HD.

Contudo, espera-se que essa pesquisa possa levar o leitor a refletir sobre a importância das realizações de modelos para avaliar e medir o desempenho de Ambiente Virtual de Aprendizagem em nuvem privada favorecendo um planejamento que oferece uma infraestrutura como serviço, com segurança, alta disponibilidade, confiabilidade e entre outros.

## 8. Agradecimentos

Os autores gostariam de agradecer à FACEPE e ao CNPq pelo apoio financeiro para a realização desta pesquisa.

## Referências

- BARROS, J. R. A. de. Modelagem de Desempenho do Banco de Dados Cassandra. Dissertação (Mestrado) — Universidade Federal Rural de Pernambuco, 2018.
- BAUER, E.; ADAMS, R. Reliability and availability of cloud computing. [S.l.]: John Wiley & Sons, 2012.
- Babin, B. J.; Hardesty, D. M.; Suter, T. A. (2003) Color and shopping intentions: The intervening effect of price fairness and perceived affect. *Journal of Business Research*. 56(7), 541-551. [http://dx.doi.org/10.1016/S0148-2963\(01\)00246-6](http://dx.doi.org/10.1016/S0148-2963(01)00246-6)
- BELUCE, A. C.; OLIVEIRA, K. L. d. Ambientes virtuais de aprendizagem: Das estratégias de ensino às estratégias de aprendizagem. 2012.
- CALLOU, G. et al. Energy consumption and execution time estimation of embedded system applications. *Microprocessors and Microsystems*, Elsevier, v. 35, n. 4, p. 426–440, 2011.
- DANTAS, J. R. Planejamento de infraestrutura de nuvens computacionais para serviço de vod streaming considerando desempenho, disponibilidade e custo. Universidade Federal de Pernambuco, 2018.
- HAGUENAUER, C. J.; LIMA, L. G. R.; FILHO, F. C. Comunicação e interação em ambientes virtuais de aprendizagem. In: Anais do Congresso Internacional de Educação a Distância (CIED). Foz do Iguaçu-PR. [S.l.: s.n.], 2010.
- JAIN, R. The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling. [S.l.]: Wiley, 1991. I-XXVII, 1-685 p. (Wiley professional computing). ISBN 978-0-471-50336-1.
- JUNIOR, D. d. A. B. d. S.; TAVARES, E. A. G. Avaliação de desempenho e dependabilidade de infraestruturas da computação em nuvem. 2015.
- JÚNIOR, L. J. et al. Uma extensão do moodle para recomendação ubíqua de objetos de aprendizagem. *RENOTE*, v. 10, n. 3, 2012.
- KENSKI, V. M. Educação e tecnologias. [S.l.]: Papirus editora, 2007.
- KIM, H. et al. A trust evaluation model for qos guarantee in cloud systems. *International Journal of Grid and Distributed Computing*, v. 3, n. 1, p. 1–10, 2010.
- LILJA, D. J. Measuring Computer Performance: A Practitioner’s Guide. [S.l.]: Cambridge University Press, 2000.
- MACIEL, P.; LINS, R. D.; CUNHA, P. R. F. Introdução às redes de petri e aplicações.05 1996.
- MANSUR, A. F. U. et al. Novos rumos para a informática na educação pelo uso da computação em nuvem (cloud education): Um estudo de caso do google apps. In: Foz do Iguaçu: Anais do XVI Congresso Internacional ABED de Educação a Distância. [S.l.: s.n.], 2010. p. 35.
- MARON, C. A. et al. Avaliação e comparação do desempenho das ferramentas openstack e opennebula. 12th Escola Regional de Redes de Computadores (ERRC), p. 1–5, 2014.
- MATOS, M. A. Manual operacional para a regressão linear. Faculdade de Engenharia da Universidade do Porto, p. 63, 1995.
- MURATA, T. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, v. 77, n. 4, p. 541–580, April 1989. ISSN 0018-9219.
- NIST. NIST Cloud Computing Program. 2014. Disponível em: <http://www.nsti.gov/itl/cloud/index.cfm>.

- RUSCHEL, H.; ZANOTTO, M. S.; MOTA, W. d. Computação em nuvem. Curitiba, abr, p. 1–3, 2010.
- SILVA, B. A framework for availability, performance and survivability evaluation of disaster tolerant cloud computing systems. Universidade Federal de Pernambuco, 2016.
- SILVA, S. Acessibilidade digital em ambientes virtuais de aprendizagem. Revista GEINTEC Gestão, Inovação e Tecnologias, v. 2, n. 3, p. 245–254, 2012.
- SOUSA, E. T. G. Modelagem de desempenho, dependabilidade e custo para o planejamento de infraestruturas de nuvens privadas. Tese (Doutorado) — Federal University of Pernambuco, Recife, Brazil, 2015.
- TORRES, E. et al. Performance and availability evaluation of storage services in private cloud. In: 2016 11th Iberian Conference on Information Systems and Technologies (CISTI). [S.l.: s.n.], 2016. p. 1–6.
- TRIVEDI, K. S. et al. Reliability Analysis Techniques Explored Through a Communication Network Example. 1996.
- UFRPE. UFRPE lança Ambiente Virtual de Suporte à Aprendizagem. 2015. Disponível em: <http://ufrpe.br/br/content/ufrpe-lanca-ambiente-virtual-de-suporte-a-aprendizagem>.
- UFRPE. Relatório de Gestão. 2017. Disponível em: [http://www.proplan.ufrpe.br/sites/www.proplan.ufrpe.br/files/relatoriogestao\\_ufrpe\\_2017.pdf](http://www.proplan.ufrpe.br/sites/www.proplan.ufrpe.br/files/relatoriogestao_ufrpe_2017.pdf).
- XIONG, K.; PERROS, H. Service performance and analysis in cloud computing. In: 2009 Congress on Services - I. [S.l.: s.n.], 2009. p. 693–700. ISSN 2378-3818.